Predictive Modelling Approach to Data-driven Computational Psychiatry

Wajdi Alghamdi

Data Science & Soft Computing Lab, and

Department of Computing, Goldsmiths, University of London.

First Supervisor Dr Daniel Stamate

Data Science & Soft Computing Lab, and

Department of Computing, Goldsmiths, University of London.

Second Supervisor Dr Daniel Stahl

Department of Biostatistics & Health Informatics, Institute of Psychiatry, Psychology and Neuroscience, King's College London.

This dissertation is submitted for the degree of Doctor of Philosophy

September 2018

DECLARATION

I declare that this thesis has been entirely composed by myself and that it has not been submitted, in whole or in part, in any previous application for a degree. The work presented is my own and was developed as joint research with other collaborators as per listed publications, except where I state otherwise by reference or acknowledgement. Parts of this work have been published in:

- The 15th IEEE International Conference on Machine Learning and Applications, Anaheim, California, USA, 2016.
- The 16th IEEE International Conference on Machine Learning and Applications, Cancun, Mexico, 2017.
- The 12th Annual Conference on Health Informatics meets eHealth, Schönbrunn Palace, Vienna, Austria, 2018.
- The 17th International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems. Cádiz, Spain, 2018.
- The 14th International Conference on Artificial Intelligence Applications and Innovations, Rhodes, Greece, 2018.

Wajdi Alghamdi

ACKNOWLEDGEMENTS

First of all, and most importantly, I would like to thank my supervisor, **Dr Daniel Stamate**, for his patient guidance and encouragement, and for the extremely useful advice and research insights and ideas, he has helped me with throughout my time as his PhD student. I have been extremely blessed to have a supervisor who cared so much about my research and academic development, and who allocated a lot of his time to guide me – I am really grateful. Due to his continuous support, I benefited of valuable accesses to clinical datasets for my research, to excellent computing facilities made available in his research lab for analyzing these data, and of the excellent work connections with the Institute of Psychiatry, Psychology and Neuroscience (IoPPN) at King's College London, the top research institution of this profile in Europe and worldwide. My PhD thesis' achievements were possible also due to the fruitful collaboration of Dr Stamate's team in the Data Science & Soft Computing Lab in which my work was developed, with the world-class Psychiatry and Statistics research partners from IoPPN. Special thanks go to my co-supervisor, **Dr Daniel Stahl**, from the Department of Biostatistics and Health Informatics of IoPPN, for all his support in my research work, for his extremely useful advices and for the amazing talks that he invited me to attend in the Machine Learning Journal Club he organises at King's College London. The insightful discussions with **Dr Marta di Forti** and **Prof Sir Robin Murray** from IoPPN, and their support with data and excellent comments while I was preparing my first publications on predictive modelling with applications in mental health, were very valuable and I thank them for it.

I would like to thank also the members of the Data Science & Soft Computing Lab, and of the Department of Computing at Goldsmiths, University of London. Here I greatly benefited from a very good work atmosphere, and of the excellent computing facilities for Data Analytics and Machine Learning modelling.

Last but not least, I thank my father **Prof Mohamad Alghamdi** – I am beyond grateful to him for everything he has done and continues to do for me in his special ways. I honestly cannot thank him enough for always providing me with a never-ending supply of support, reassurance, and love. My gratitude to him is therefore beyond words.

Thank you for all your support and energy, and for everything you have done along the way. This thesis would undoubtedly not have come to be as it is without your continuous presence.

Wajdi Alghamdi

ABSTRACT

This dissertation contributes with novel predictive modelling approaches to data-driven computational psychiatry and offers alternative analyses frameworks to the standard statistical analyses in psychiatric research. In particular, this document advances research in medical data mining, especially psychiatry, via two phases. In the first phase, this document promotes research by proposing synergistic machine learning and statistical approaches for detecting patterns and developing predictive models in clinical psychiatry data to classify diseases, predict treatment outcomes or improve treatment selections. In particular, these data-driven approaches are built upon several machine learning techniques whose predictive models have been pre-processed, trained, optimised, post-processed and tested in novel computationally intensive frameworks. In the second phase, this document advances research in medical data mining by proposing several novel extensions in the area of data classification by offering a novel decision tree algorithm, which we call PIDT, based on parameterised impurities and statistical pruning approaches toward building more accurate decision trees classifiers and developing new ensemble-based classification methods. In particular, the experimental results show that by building predictive models with the novel PIDT algorithm, these models primarily led to better performance regarding accuracy and tree size than those built with traditional decision trees. The contributions of the proposed dissertation can be summarised as follow. Firstly, several statistical and machine learning algorithms, plus techniques to improve these algorithms, are explored. Secondly, prediction modelling and pattern detection approaches for the first-episode psychosis associated with cannabis use are developed. Thirdly, a new computationally intensive machine learning framework for understanding the link between cannabis use and first-episode psychosis was introduced. Then, complementary and equally sophisticated prediction models for the first-episode psychosis associated with cannabis use were developed using artificial neural networks and deep learning within the proposed novel computationally intensive framework. Lastly, an efficient novel decision tree algorithm (PIDT) based on novel parameterised impurities and statistical pruning approaches is proposed and tested with several medical datasets. These contributions can be used to guide future theory, experiment, and treatment development in medical data mining, especially psychiatry.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

Chapter 1 Introduction

1.1 Research context and motivation

From the day life existed, decision making has been a part of the evolution of the human species. Humans make most of their decisions based on information and their experiences. To make an accurate decision they usually look for patterns in their past experiments, and then decide on their best action. At present, more data and experiences have become available due to the massive increase in computers' abilities.

In many domains, where data is growing rapidly, there is always useful hidden information that needs to be extracted. For example, current patient records are stored on a regular basis, and this data may be used in extracting patterns for diseases, or for estimating health risk automatically, etc. Therefore, there is a need for knowledge discovery methods to be devised and applied to such data [1].

The process of deriving knowledge from data has long existed, and was previously based on traditional statistical analyses and interpretations. These analyses and interpretations for data domains such as health, finance, business, and marketing usually rely on the specialists' skills to read into the data, which makes these analyses and interpretations costly and often only able to produce limited results. As adequate computer capacity and the volume of data increase, traditional approaches are often inappropriate. In this context, modern machine learning techniques are suited to improve the quality of these analyses and interpretations significantly.

The field of machine learning has advanced at a tremendous pace in recent years, with advanced predictive techniques being developed and improved upon. In order for these technologies to become truly refined, they must be applied to a variety of fields and subsequently challenged to find relevant solutions [2]. One such area of application is the field of medical research, which has a broad range of potential uses for machine learning [3] [4] [5].

Recently, machine learning techniques have emerged as a promising approach to medical prediction. For instance, recent works have sought to compare a variety of algorithms in predicting patient survival after breast cancer [6] and surgery for hypocellularity carcinoma [7]. In addition, machine learning techniques have proven their abil-

ity in predicting mental diseases such as Alzheimer's [8]. These studies suggest that machine learning can provide medical research with powerful techniques beyond the traditional statistical approaches mostly used in this area, such as statistical tests, linear, and logistic regression. In biomedical engineering, several recent papers have explored the potential for classification algorithms to detect disease [9] [10]. This has led to the publication of additional guidance for medical researchers on how to interpret and question such findings [11]. Last but not least, there is tremendous interest in current interdisciplinary research into exploiting the power of machine and statistical learning to enable further progress in the new and promising area of precision medicine, in which predictive modelling plays a key role in forecasting treatment outcomes, and thus decisively contributes to optimising and personalising treatments for patients [5] [12].

A promising new approach is the use of computational modelling approaches to psychiatry [4]. Computational psychiatry has made it possible to combine enormous levels and types of computation with several types of data in an effort to advance classification of mental disease, predict treatment outcomes, and improve treatment selection [13] [14].

Computational psychiatry is considered an essential area of research, yet there are still many difficult tasks that need to be carried out precisely and efficiently. Most studies in medical research (such as psychiatry) are only explanatory research and do not involve risk prediction modelling using machine learning algorithms. Moreover, incomplete or inconsistent records, as well as the methodologies used (based mostly on conventional and straightforward statistical methods as pointed out above) limit many existing studies. These methods are traditionally well recognised and used in medical research (such as psychiatry), but in many situations do not match the great potential of modern machine learning methods.

Motivated by the above discussion, the present work is devoted to proposing synergistic statistical and machine learning approaches to medical data mining and precision medicine in the area of psychiatric research. In particular, this dissertation proposes a predictive modelling approach to data-driven computational psychiatry.

1.2 Thesis statement

This document advances research in data mining via two phases. In the first phase, this dissertation focuses on developing a synergistic statistical and machine learning approach to medical data mining and precision medicine to improve patient care. In particular, this work proposes novel prediction modelling and pattern detection approaches for the first-episode psychosis associated with cannabis use. A significant effort in this study was the data pre-processing due to inherent challenges present in data collected in a case-control study involving many missing values, multiple encodings of related information, and a significantly large number of variables, etc. The innovative approaches are built upon several machine learning techniques whose predictive models have been optimised in a computationally intensive framework. Then, a new computationally intensive machine learning framework for understanding the link between cannabis use and first-episode psychosis was introduced. Finally, prediction models for the first-episode psychosis associated with cannabis use were developed using artificial neural networks and deep learning with the proposed novel computationally intensive framework.

In the second phase, the dissertation focuses on developing new machine learning algorithms that are particularly suitable for medical research. In particular, we propose novel and enhanced algorithms that produce models with high explanatory power, such as decision trees based on new families of impurities and statistical pruning approaches. The novel decision tree algorithm called PIDT, which is based on parameterised impurities and statistical pruning approaches, is proposed and tested with several medical datasets.

1.3 Aims and objectives

The aim of this thesis is to develop predictive modelling data-driven approaches to computational psychiatry to advance classification of mental disease, predict treatment outcomes, or improve treatment selection. To this end, the thesis proposes synergistic statistical and machine learning approaches to medical data mining and precision medicine in the area of psychiatric research. It also proposes new machine learning algorithms that have high explanatory power and are particularly suitable for medical research.

To this end, several medical datasets are presented to report the advantages and the disadvantages of the machine and statistical learning algorithms. This includes the clinical psychiatry data (first-episode psychosis - cannabis clinical dataset) introduced to build novel predictions models and to detect new patterns in patients' data [15]. The clinical psychiatry data was used to develop predictive modelling data-driven approach to computational psychiatry to advance classifications of mental disease. In order to accomplish this aim, the following objectives should be met.

> Objective 1: systematically review and analyse the area of computational psychiatry and the challenges involved.
>
> Objective 2: Derive statistical and machine learning methods for predictive modelling of mental diseases, focusing on psychosis associated with cannabis use, and investigate their use and performance.
>
> Objective 3: Propose approaches to further enhance the predictive value of the derived methods.
>
> Objective 4: Derive methods with high explanatory power suitable for computational psychiatry research.

The above objectives are associated with some important research questions that the thesis aims to answer.

I. Can one use a clinical data to build prediction models for mental diseases such as the first-episode psychosis associated with cannabis use? This refers to Objective 2.
II. How can one improve the prediction of mental illness in the presence of a significantly large number of missing values and unbalanced classes in case-control data? What is the variability of predictions in the presence of missing values? Are these prediction models stable enough? These refer to Objective 3.
III. Do some predictors, such as cannabis use attributes, in clinical psychiatry data [15] have predictive information for mental illness, such as the first episode psychosis? What is the predictive value of these predictors on mental illness, such as first episode psychosis? These refer to Objective 2 and 3.
IV. Can the prediction models be improved with post-processing techniques? Can the prediction models be improved via optimising the cut-off point selection on the ROC curve? These refer to Objective 3.

V. Can one develop a new machine learning algorithm that has high explanatory power, such as decision trees, for medical research? This refers to Objective 4.

VI. How attribute selection should be done and what impurity measures should be used? How overfitting can be avoided? These refer to Objective 4.

1.4 Methodology

Different research problems involve different research methodologies. The main categories of research approaches that were used in the thesis are experimental research, build research, process research and simulation research. These research strategies are described as follow. The first methodology, which is mainly used, is the experimental methodology which uses experiments designed to test hypothesis. It often involves record keeping, experimental setup design, and experimental results reporting. All the experiments and results should be reproducible. The second methodology is the build methodology which often encompasses designing the software system, reusing components, choosing an adequate programming language, and considering testing all the time. The third methodology is the process methodology which often includes software process, methodological issues, and cognitive modelling. The final methodology is the simulation methodology which uses computer simulations to address question difficult to answer in the real application.

This thesis includes several comparative studies which usually employ several techniques, and try to find which one is better. Answering this question should be for a given purpose, which is not necessarily absolute ranking, such as proposing predictive modelling approaches to data-driven computational psychiatry. Other research questions like “where are the differences?” and “What are the trade-offs?” need to be answered as well in this research method.

The above research method should be applied to a “clinical psychiatry data” and typically compared in the form of a table.

1.5 Structure of the thesis and contributions

The thesis is organised as follows. Background information on the machine learning modelling procedure was used to establish the results in this work, and strategies to improve

the procedures are provided in Chapter 2. The overall data processing framework/pipeline is proposed in Chapter 3 as well as a series of standard results, definitions, and models of existing statistical and machine learning algorithms related to this study are reviewed. In Chapter 4, novel prediction modelling and pattern detection approaches for first-episode psychosis associated with cannabis use are derived. A new machine learning framework for understanding the link between cannabis use and first-episode psychosis is presented in Chapter 5. More powerful prediction models for first-episode psychosis associated with cannabis use via neural networks and deep learning, are provided in Chapter 6. A novel decision tree algorithm PIDT based on new families of impurity measures and statistical pruning approaches for building optimised decision trees are introduced and used to understand successfully the link between cannabis use and first-episode psychosis in Chapter 7. In Chapter 8, conclusions are drawn, and future research directions are discussed.

Overall, the thesis is organised into six chapters and a common conclusion. The main contents of each chapter are briefly outlined below.

Chapter 2 – Background and problem. Background information on the machine learning modelling procedure is used to establish the results in this work, and strategies to improve them are set out in this chapter. This includes information on the data used in this dissertation, data preparation techniques, resampling techniques, and estimating the model performances.

Chapter 3 – Methodology. This chapter presents the general data processing framework/pipeline first, and then it will allow customising/tailoring this framework to fit the needs of each chapter. The proposed framework can be tailored to the needs of a particular dataset, or to answer a specific research question, by using a particular method/technique. Also, a series of standard results, definitions, and models of the existing statistical and machine learning algorithms related to this study are provided.

Chapter 4 - Novel prediction modelling and pattern detection approaches for the first-episode psychosis associated with cannabis use. The predictive value of cannabis-related variables concerning first-episode psychosis is demonstrated in this chapter by showing that there is a statistically significant difference between the performance of the

predictive models built with and without cannabis variables. We were inspired in this approach by the Granger causality techniques [16], which are used to demonstrate that some variables have predictive information on other variables in a regression context, as opposed to classification, which is mainly the case in our framework. Moreover, we investigate how different patterns of cannabis use relate to new cases of psychosis, via association analysis and Bayesian techniques such as Apriori and Bayesian Networks, respectively.

Chapter 5 - A new machine learning framework for understanding the link between cannabis use and first-episode psychosis. This chapter proposes a refined machine learning framework for understanding the links between cannabis use and first episode psychosis. The novel framework concerns extracting predictive patterns from clinical data using optimised and post-processed models based on Gaussian processes and support vector machines algorithms. The cannabis use attributes' predictive power is investigated and we demonstrate statistically and with ROC analysis that their presence in the dataset enhances the prediction performance of the models with respect to models built on data without these specific attributes.

Chapter 6 – Predicting first-episode psychosis associated with cannabis use with artificial neural networks and deep learning. This chapter proposes a novel machine learning approach, based on neural networks and deep learning algorithms, to developing highly accurate predictive models for the onset of first-episode psychosis. Our approach is also based on a novel methodology of optimising and post-processing the predictive models in a computationally intensive framework. A study of the trade-off between the volume of the data and the extent of uncertainty due to missing values, both of which influence predictive performance, enhanced this approach. The performance capabilities of the predictive models are enhanced and evaluated by a methodology consisting of novel model optimisation and testing, which integrates a phase of model tuning, a phase of model post-processing with ROC optimisation based on maximum accuracy, Youden and top-left methods, and a model evaluation with the k-fold cross-testing novel methodology (explained in the previous chapter). We further extended our framework by investigating cannabis use attributes' predictive power and demonstrating statistically that their presence in the dataset enhances the prediction performance of the artificial

neural networks presented in this chapter. Finally, the model stability is explored via simulations with 2000 repetitions of the model building and evaluation experiments.

Chapter 7 - PIDT: A novel decision tree algorithm based on parameterised impurities and statistical pruning approaches. This chapter presents novel splitting attribute selection criteria based on some families of parameterised impurities that we propose here to be used in the construction of optimal decision trees. These criteria rely on families of strict concave functions that define the new generalised parameterised impurity measures that we applied in devising and implementing our PIDT novel decision tree algorithm. This chapter also proposes the S-condition based on statistical permutation tests, whose purpose is to ensure that the reduction in impurity, or gain, for the selected attribute is statistically significant. The idea behind proposing such algorithms is to build accurate prediction models that are easy to interpret and explain to psychiatry experts.

Chapter 8 - Conclusion and directions for future work. The main results are summarised, and future research directions are discussed.

Chapter 2 Background and problem definition

2.1 Data-driven computational psychiatry

Machine learning algorithms have already begun to prove their particular capabilities in and contributions to medical research and applications [4]. In particular, machine learning techniques have been successfully used in diagnosing psychosis [17], analysing diabetic patients' data [18] [19], classifying leukaemia [20], and detecting heart conditions in electrocardiogram (ECG) data [21], etc. These studies show that machine learning has proven to be capable of dealing with challenging medical data, in particular with the ambiguous nature of the ECG signal data, for which machine learning algorithms show outstanding results compared to other methods [20] [21].

These days, more health care providers are replacing traditional paper notes with electronic patient records. In addition, the use of advanced technologies, such as computers, personal digital assistants, smartphones, etc., has enabled information to become more available and accurate [3]. This led to a tremendous increase in the electronic health data, creating a promising basis for applying machine learning algorithms to extract insights from data.

Currently, machine learning algorithms are in the process of revolutionising health. In the same way as machine learning has made an enormous difference to business and industry, it will just as undoubtedly enhance medical research and improve the practice of healthcare providers. The medical field is considered a critical area of research, yet there are still many difficult tasks that need to be carried out precisely and efficiently. The future success of health sector planning, and of health care in general, will be in the adoption of intelligent systems where robotics and machine learning intersect. In order for health sector planning to catch up with this fast-changing environment, machine learning must be at the core of most strategies. For example, new developments in psychiatry concern the so-called data-driven computational psychiatry, which relies heavily on the use of machine learning [4]. Data-driven computational psychiatry has made it possible to combine enormous levels and types of computation with several types of data in an

effort to advance classification of mental disease, predict treatment outcomes, or improve treatment selection [13] [14].

Most studies in medical research (such as psychiatry) so far are only explanatory research and do not comprise risk prediction modelling using machine learning algorithms. In addition, many existing studies are limited by incomplete or inconsistent records, but also by the methodologies used, which are based mostly on conventional and straightforward statistical methods. These methods are traditionally well recognised and used in medical research (such as psychiatry), but in many situations, they do not match the large potential of the modern machine learning methods.

In this chapter, we discuss and summarise different machine and statistical learning techniques that are suitable for use in the medical field, especially in psychiatry. Medical research involves many problems that benefit from analysing data based on techniques of data pre-processing, predictive modelling, clustering, and so on. In predictive modelling in particular, the task is to predict the outcome associated with a particular patient given a feature vector describing that patient. In clustering, patients are grouped because they share similar characteristics, and in data pre-processing operations such as feature selection, the task is to select the most relevant attributes to predict the outcome for a patient [2].

Many of these data pre-processing algorithms are described in this chapter. However, we should note that no single algorithm is superior to others in all the problems. The algorithm needs to match the structure and the particularities of the problem at hand, in order to obtain useful information or an accurate model. The ultimate aim is to develop models that use predictors or known features to create predictive models that will be utilised for predicting the output [22]. However, choosing the suitable algorithm and developing a model are not the only aspects we need to consider; other data mining phases such as data pre-processing and model post-processing are also involved. Therefore, extracting knowledge from data involves all these phases of processing.

The first stage, which is the pre-processing stage, has three sub-processes: data filtering, data cleaning, and data transformation and projection. Data filtering is responsible for the selection process of relevant data to be analysed. Data cleaning involves handling data problems such as treating missing values, smoothing noise in data, removing outliers, etc. Data transformation is responsible for aggregation, normalisation, and

unit conversion, and helps to speed up the process, improve performance, and decrease problem complexity.

The processing stage consists of some sub-processes such as model generation, tuning and building, and evaluating the output model. Model generation and tuning are some of the most critical sub-processes. The model generation and tuning are iterative processes comprising three steps: choosing the algorithm and its parameters, building the model, and evaluating the model. The goal of this process is to find the best parameter values for the model and thus assess the performance of an algorithm for the problem at hand [23].

The last stage is the post-processing stage, which is responsible for knowledge presentation and improving the model performance. Knowledge presentation is used to display the extracted knowledge comprehensively. Finally, based on the results from the entire data mining process, the best performing model is applied to the current problem.

The essential question when dealing with machine learning is not whether a learning algorithm is favoured over others, but under which conditions a certain developed prediction model can significantly outperform others for a given application problem. This chapter provides a literature review of several strategies to improve these statistical and machine learning prediction algorithms. Moreover, the several datasets used in this work are outlined in this chapter. Synthetically generated datasets are used to report the advantages and the disadvantages of several machine and statistical learning algorithms. In addition, clinical data was used to build novel predictions models and detect new patterns in the data.

The above techniques are used in the remainder of the thesis to propose novel synergistic machine learning and statistical approaches to pattern detection and to develop predictive models for research questions such as predicting first episode psychosis.

2.2 Sources of datasets

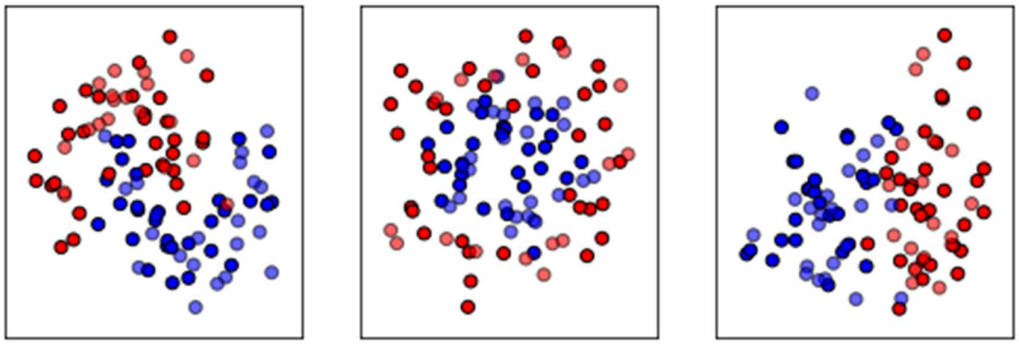
Several datasets were used throughout the document. Randomly generated datasets were used in the literature to report the advantages and the disadvantages of several machine and statistical learning algorithms on various types of datasets. Other datasets such as clinical data were used to build novel prediction models and to detect new patterns in that

particular dataset. Finally, some public data was used to validate the decision tree classifiers produced with the novel PIDT algorithm that we propose in chapter 7. The selected datasets can be grouped into synthetically generated datasets, clinical datasets, and public datasets.

2.2.1 Generated datasets

Three synthetic datasets were generated to illustrate the nature of decision boundaries of different classifiers and to give an overview of how different classifiers perform on various synthetic datasets using a package called scikit-learn 0.19.1 (October 2017) [24]. Each of the three datasets has 100 samples. Each sample has two input attributes and one output attribute that represents the class membership of each sample. The three generated datasets are:

- **Moons dataset,** which contains samples in the form of two interleaving half-circles.
- **Circles dataset,** which contains samples in the form of a larger circle containing a smaller circle.
- **Linear dataset**, which contains samples that are linearly separable.



Moons Dataset Circles Dataset Linear Dataset

Figure 2:1 Generated datasets.

Each of these three datasets has some noise added in order to simulate real situations in which data and classifications are not perfect. Figure 2:1 presents the data points for the three generated data sets. The plots show training points in solid colours and testing points in semi-transparent colours.

2.2.2 First-episode psychosis - cannabis clinical dataset

This clinical dataset is a part of a case-control study at the inpatient units of the South London and Maudsley (SLaM) NHS Foundation Trust [15]. The dataset is also used in training and optimising the predictive models for first-episode psychosis in chapters 4, 5, 6, and 7. The clinical data consists of 1106 records, including patients and controls. Those described as patients were patients of the trust who at one time presented with first-episode psychosis; controls were recruited from the local area through the internet, newspaper advertising, and by distributing leaflets. Each record refers to a participant of the study and has 255 possible attributes, which were divided into four categories. The first category consists of demographic attributes that represent general features such as gender, race, and level of education. Secondly, drug-related attributes contain information on the use of non-cannabis drugs such as tobacco, stimulants, and alcohol. The third category is formed by genetic attributes. The final category contains cannabis-related attributes such as the duration of use, initial date of use, frequency, and cannabis type, etc. (See Appendix 2).

2.2.3 Public datasets

Five public datasets are used in chapter 7 to illustrate the performance of different impurity measures as splitting criteria for the decision trees built with our newly proposed PIDT algorithm. These datasets are as follows:

1. Breast Cancer Wisconsin (original) dataset from the UCI Machine Learning Repository [25]. This dataset comprises 569 observations and 30 numeric attributes. Each observation is in one of the two classes, malignant or benign.
2. Pima Indians Diabetes dataset from the UCI Machine Learning Repository [25]. Ten measures (variables) were obtained for each of n = 442 diabetes patients over one year. The goal is a quantitative measure of disease progression after one year.
3. Hepatitis dataset from the UCI Machine Learning Repository [25]. The hepatitis dataset contains 155 examples of hepatitis patients, described by 19 numeric and nominal attributes. Of these cases, 123 correspond to the patients who survived treatment (‘live’) and 32 examples of mortalities (‘die’).

4. The Primary Tumour dataset from the University Medical Centre, Institute of Oncology, Ljubljana, Yugoslavia [25]. The primary tumour dataset has 21 concepts and 17 attributes, and 207 out of 339 examples contain at least one missing value.
5. The glass identification dataset from the UCI Machine Learning Repository [25] comprises data representing a study of classification of types of glass, motivated by criminological investigations. At the scene of the crime, the glass left can be used as evidence, if correctly identified. This dataset contains 10 attributes regarding several glass types (multi-class).

2.3 Data preparation

Data preparation techniques refer to the process of adding, deleting, or transforming data. Data preparation can influence improving a model's predictive ability. The choice of the predictive modelling techniques determines which strategies to apply. Some models, such as tree-based models, have the capacity to deal with numeric and nominal attributes. Others, like support vector machines, do not. In addition, some models, such as distance-based models, like k-nearest neighbour (k-NN), are very insensitive to the characteristics of the predictor data. Others, like linear regression, are not.

Prior to applying any statistical and machine learning algorithms, significant work effort is usually involved in the data pre-processing in order to deal with the challenges present in the data sets. This section reviews several approaches to overcome these challenges in the data.

2.3.1 Dealing with missing values

In real-life data sets, the most common problem is incomplete data. Many applications have missing data for a variety of reasons. Sometimes the data collection was done improperly. In some cases, participants interrupted their participation in a study. In other cases, the value for an attribute is unavailable. All these situations generate missing values. A limited number of machine learning algorithms, such as C4.5 decision trees or Naive Bayes, can handle internally missing values, but the vast majority require preliminary treatment of the missing values. Several methods exist for handling missing data. Some methodologies remove attributes or records that have a high percentage of missing

values [26]. Other methodologies treat the missing values by imputing them. This section presents techniques to deal with missing data, such as embedded methods for missing data, filtering missing data, and imputing missing data.

Decision tree prediction approaches have robust methods for handling incomplete data, such as C4.5 [27]. In the training stage, the impurity of each attribute is adjusted by a factor depending on the number of available values in the training set for the same attribute. The classification and regression trees algorithm (CART) employs the surrogate variables splitting (SVS) technique. This method is for use during the prediction phase only [26]. The recursive partitioning and regressing trees (RPART) approach contains an extension of the previous methods to handle missing data during the training stage [28].

Filtering techniques remove incomplete data parts. The list-wise deletion method removes all incomplete instances so that it is suitable for the prediction models to use the remaining complete instances [29]. These methods may be practicable when the number of missing values is quite small compared with the remaining data set. There are two ways to filter and discard data with missing values. The first way is to keep complete instances only. The second method consists of determining the percentage of missing data on each record and attribute and deleting the record and/or attributes with missing data percentages below the specified threshold. Unfortunately, essential attributes could be discarded during this process. Therefore, extra care should be taken before removing any attribute.

Imputation techniques assume that there is a relation between the attributes, so the objective is to apply prediction models to infer the missing values in one attribute using the existing values of other attributes in the dataset. This process is repeated until it produces a complete data set. Imputation replaces the missing data in a deterministic or stochastic way [1]. In the deterministic case, the missing value is replaced by a uniquely inferred value. In the stochastic case, the missing value is replaced by a random value from some distribution. Imputation methods may be global or local, depending on the volume of data.

On the one hand, global imputation techniques are of two main kinds: missing attribute and non-missing attribute methods. In the missing attribute method, new values are calculated for the missing data items based on analysing the existing values for the

attributes, by using mean, median, or mode. Although this technique has some bias towards a standard deviation, if we use a non-deterministic mean imputation method, we will get better performance because it produces random disturbance to the average. The main drawbacks of this approach are the potential generation of inconsistent data and the complexity of the computation. In the non-missing attribute methodologies, we assume that there is a correlation between missing and non-missing values. These methodologies use the correlation to predict the missing data. Imputation by regression treats the missing data as a target attribute, and it performs regression to input missing values [30]. However, this method also has some disadvantages. When selecting a suitable regression model, for example, only one value is derived for each missing data, which fails to represent the uncertainty associated with missing values. In 1987, Rubin proposed a new technique of imputation to overcome the uncertainty problem in linear regression. Multiple imputations consist of three stages: produce m-complete data sets through single imputation, analysis of each of the m-data sets, and the combination of the results of the m-analyses into the result [31].

On the other hand, there are many techniques for local imputations. Local imputation methods do not have a theoretical formulation but have been implemented in practice [32] [33] [34]; the imputation uses a supervised learning technique such as K-NN and bagging trees. The k-NN imputation uses the k-NN algorithm to estimate and substitute missing data. It tries to find similar records for the current record and to impute the missing value from the corresponding values of the neighbouring records. The main benefit of this approach is that it can predict both discrete attributes and continuous attributes. It uses the most common value for discrete attributes and the mean value for numeric attributes. Other algorithms, such as bagging trees, have also confirmed their ability in many applications in practice [35]. In the bagging tree imputation, the algorithm treats the attribute that contains missing values as an output attribute, and it builds the tree using the remaining attributes. Then, the algorithm imputes the missing values in the output attribute using the built tree. The imputation iterates through the attributes until there are no missing values.

2.3.2 Centring and scaling

In both statistics and machine learning, centring and scaling numeric attributes are often crucial prior to building prediction models, in order for a particular model to perform accurately. Centring and scaling are regularly adapted to support the numerical stability of the computations during building prediction models. For instance, the K-NN algorithm requires the attributes to have a standard scale to be developed accurately, since it depends directly on measuring the distance between records. In addition, these manipulations are needed when regression models are being generated because if the predictors have several units and ranges, the final model will have disproportionate coefficients, which makes it difficult to interpret. Moreover, centring and scaling could be employed first when building penalised models such as lasso regression and ridge regression, since the penalty in these methods is calculated based on the estimated coefficients.

To centre an attribute, the mean value of that attribute is subtracted from all values of the attribute. As a result of this process, the attribute will have a mean value that is equal to zero. Then, to scale the attribute, the values of the attribute are divided by the calculated standard deviation. As a result of this process, the attribute will have a standard deviation equal to one. To centre and scale an attribute x the following equation is employed on each data point r, where r ranges from 1 to n.

$$x_{rc}^{*} = \frac{x_{rc} - \bar{x}_c}{\sigma_c}$$

Above x_{rc}^{*} is the new value of the predictor c for the r^{th} data point, $\bar{x}_c$ is the mean of the n values of the predictor c, and σ_c is the standard deviation of the n values of the predictor c. The only disadvantage of these transformations is the loss of the real values of the individual records since the data are no longer available in the original range or units.

2.3.3 Resolving outliers

Outliers are samples that are abnormally far from the mainstream of the data. In both statistics and machine learning, dealing with these outliers is necessary before building accurate prediction models. Outliers can mislead the training process resulting in prolonged training times and less accurate models. Some predictive models, such as tree-

based prediction models and SVM, are resistant to outliers. However, other models such as logistic regression and neural networks are not.

One way to deal with the outliers is to detect them and then to delete them. Outliers can be recognised easily by looking at some plots such as the box plot. A single box plot for one attribute is shown in Figure 2:2, where the plot shows the median, the first and the third quartile, and the outliers. When some samples are suspected to be outliers, these samples should be removed from the dataset, especially if the used prediction model is considered sensitive to outliers. However extra care should be taken when removing samples, especially if the dataset size is small, otherwise sensitive data may be wasted.

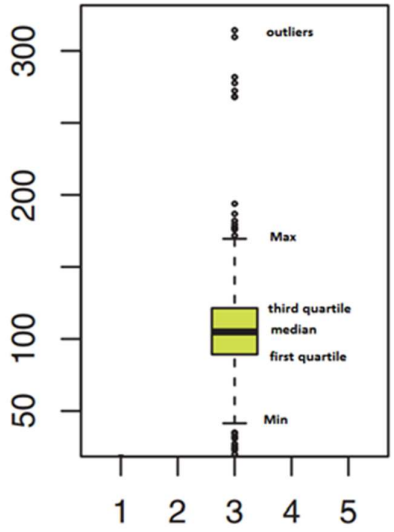


Figure 2:2 Box plot for a single attribute.

Another way to deal with the outliers is to transfer the attributes using the spatial sign [36]. The spatial sign projects the attribute values into a multidimensional sphere. This process will have the effect of making all the samples have the same distance from the centre of the sphere. Mathematically, each sample is divided by its squared norm, and the equation is as follows:

$$x_{rc}^{*} = \frac{x_{rc}}{\sum_{i=1}^{m} x_{ri}^{2}}$$

Where x_{rc}^{*} is the new value of the c^{th} predictor for r^{th} data point, where r range from $1\ to\ n$ and where c ranges from $1\ to\ m$. This approach is intended to measure the distances between the attributes. Therefore, it is important to centre and scale the attribute applying the above approach.

2.4 Feature selection

Feature selection methods are recommended when the predictor's number is too large compared to the sample size, resulting in the model scoring high accuracy on the training data but performing very poorly on the test data [37]. In the feature selection process, we select specific predictors to avoid the problem of over-fitting [38]. Mostly, the feature selection reduces the dimension of data, which also speeds up the data mining process, decreases computational cost, and overcomes over-fitting. However, reducing some attributes may cause loss of information and might lead to worse results. In 2007, Nilsson mentioned the two main categories of feature selection problems, which are finding the optimal predictive attributes for building efficient prediction models and finding all the relevant attributes for the class attribute [39].

Feature selection algorithms have some fundamental processes that affect the nature of the search for the best attributes, such as the starting point, the search organisation, the evaluation strategy, and the stopping criterion [40].

The point of departure is choosing a point in the predictors subset to begin the search. The selection point may be taken with no predictors, with all predictors, or somewhere in the middle. If the selection point is selected with no predictors, this methodology will start by adding predictors and proceeding forward through the search space. If the selection point is chosen with all predictors, it will start removing the predictors and proceed backwards through the search space. Finally, if the selection point is in the middle, the methodology will start adding predictors and proceed outwards from that point.

Secondly, the search organisation is the search strategy that may be an exhaustive search or a heuristic search. In the exhaustive search, the methodology starts with a small number of predictors. With x initial predictors there exist 2^X possible subsets. Although a heuristic search is more feasible than an exhaustive search and gives good results, it cannot guarantee to find the optimal attribute subset.

Thirdly, the evaluation process is the primary factor that differentiates between different feature-selection algorithms. The evaluation process is done by filters or wrapper techniques. Filters are used to remove the undesirable predictors of the data before learning begins. Wrapper techniques use a combination of induction algorithms and statistical re-sampling techniques [41].

Fourthly, the stopping process has to be determined by the feature selector methods. Feature selectors may stop adding or removing predictors depending on an evaluation strategy. If the alternative predictor does not improve upon the merit of a current predictor subset, it will stop. An alternative option is to continue generating predictor subsets until the opposite end of the search space and then choose the best subset.

There are two main categories of feature selection. The first type is to find the best subset of predictive features, which helps produce efficient prediction models. The second type is to find all the relevant predictors for the class attribute, which could be achieved by performing a ranking on the attributes according to their predictive powers. Predictive power measures are done by first computing the performance of the classifier built with every single variable, by computing statistic measures such as correlation coefficient or by applying information theory measures such as the mutual information [37].

2.4.1 Search strategies

Search strategies apply a complete search for the best predictors subset according to the evaluation function used. Heuristic search procedures consist of efficient ways to provide solution quality and decrease the search complexity. Many techniques of search strategies in this category take into consideration the remaining predictors for selection/rejection for any iteration. Therefore, these techniques are relatively fast. Many search techniques could be used for the search strategies, such as greedy hill climbing, stepwise bi-directional search, best-first search, and genetic algorithms. Genetic algorithms consider global changes and usually reach the optimal global solution. Greedy hill climbing search considers the local modification to the predictor's subset, and it can determine a locally optimal solution. Best-first search considers local modification and allows backtracking along the search path.

Greedy hill climbing is a simple search technique, which examines the local changes to the current predictor's subset. Local changes add or delete a single predictor from the subset. If the algorithm starts adding predictors, it will make a forward selection. However, if the algorithm will delete a feature, it will do a backwards elimination [42] [43]. Another option is the stepwise bi-directional search, which uses both adding and deleting predictors. In this technique, the search algorithm considers all possible local

changes to the current subset and selects the best one, or it may choose the first real improvement. If one change is accepted, it will not reconsider once again.

Best first search is an artificial intelligence search technique [44]. It is similar to greedy hill climbing in allowing backtrack along the search path. In addition, the best first moves in the search space will be the current predictor set. However, this technique is not like the greedy hill climbing. When the search seems less promising, it will backtrack to the more promising subset. Furthermore, this technique will explore the entire search space and will use the stop process to limit the number of subsets that result in no improvement.

Genetic algorithms are based on the idea of natural selection in biology [45]. Genetic algorithms and all evolutionary computation algorithms are an iterative process. The new generation is produced by applying genetic processes such as crossover, and mutation to the current generation. Mutation is done by changing the values in the subset randomly. However, the crossover is produced by combining two predictors from a pair of subsets into a new subset. The genetic processes are applied based on the value of their fitness, which is evaluated by evaluation techniques. After the assessment process, the better subsets will have a good chance to be used to create the new subsets through crossover and mutation. The algorithm uses a population of solutions, which updates over time to avoid trapping in a local minimum solution. In the feature selection process, the solution is represented by the fixed binary string. The value of each position in the binary string represents the status of a particular feature that may be presence or absence.

2.4.2 Feature selection filters

The primary goal of feature selection is to get efficient attributes to build accurate prediction models. In addition, feature selection could be used to minimise the probability of error (Bayesian). The filters employ feature selection regardless of the type of classifier but depend on the properties of the data distribution itself. There are many algorithms for the filtering process, such as RELIEF [46], Las Vegas Filter (LVF) [47], FOCUS [48], correlation-based filter (CFS) [49], and principal component analysis (PCA), as well as many statistical methods based on hypothesis tests.

In 1991, FOCUS was one of the earliest multivariate filters. The main drawback is that it cannot handle noisy data and it has a predisposition towards over-fitting [48]. In

1992, Kira proposed the RELIEF technique based on the nearest neighbour learner methodology [46]. The main drawback is that there is no methodology for choosing the neighbour sample size. In 1996, Liu used a probabilistically guided random search to explore the attribute subspace and proposed a method called LVF [47].

In 2010, Halalai et al. proposed a new filter to select those attributes that have a strong correlation with the target attribute and a weak correlation between each other [49]. Finally, PCA could be employed for feature selection. PCA has proven its ability in a large variety of applications including image processing and so on [50].

2.4.2.1 *Consistency filters*

In 1991, Almuallim and Dieterich proposed a new technique called FOCUS, which was designed for the Boolean domain [48]. FOCUS searches for predictor subsets until it finds the minimum combination of predictors that divide the training into the purist classes. The output will be a combination of predictor values in each class. The final predictor subset will be processed by an induction decision tree ID3 [51]. In 1994, Caruanna and Freitag claimed that there are two main difficulties with FOCUS [52]. Firstly, the FOCUS approach proposes to achieve consistency in the training data. However, the search process may become difficult because many of the predictors are needed to keep consistency. Secondly, this method produces a strong bias towards consistency, because the algorithm will continue to add predictors to fix a single inconsistency. In 1996, Liu and Setiono proposed a new algorithm similar to FOCUS, called Las Vegas Filter [47]. LVF works by generating a random subset S from the predictor subset during each round of execution. The inconsistency rate described by S is compared to the inconsistency rate of the best subset. If the new subset S is consistent with the best subset, then S will be the new best subset. The inconsistency rate is calculated in two steps. Firstly, inconsistency count is the number of instances occurring in the group minus the number of instances occurring in the group with the most common class value. Secondly, the overall inconsistency is the sum of the inconsistency counts in all groups of a matching instance divided by the total number of instances. Liu and Setiono mentioned that due to the randomness of LVF, the longer it is allowed to execute the better the result. They tested the algorithm on two large datasets: the first has 65,000 instances described by 59 attributes, and the second has

5,909 instances described by 81 attributes. LVF achieved a good result and reduced the dataset by more than half in both cases.

2.4.2.2 *Instance-based learning filter for feature selection*

In 1992, Kira and Rendell proposed their new algorithm RELIEF, which uses instance-based learning to assign an appropriate weight to each predictor [46]. The weights of the predictors are used to distinguish between the class values. It uses the weights to reorder the predictors; the weights beyond the user-specified threshold are used to create the final subset. The algorithm randomly selects sample instances from the training data. Each instance will do two operations: nearest hit and nearest miss. In other words, it will find the closest instance in the same class (nearest hit) and the nearest instance in the counter class (nearest miss). The attribute's weight is updated according to the value of the instance in the closest hit and nearest miss as shown in the next equation:

$$W_c = W_c - \frac{diff(C,R,H)^2}{m} - \frac{diff(C,R,M)^2}{m}$$

As shown in the above equation, the weight of attribute C is updated, where R is a randomly sampled instance, H is the nearest hit, M is the nearest miss, and m is the number of randomly sampled instances. The difference function $diff$ is a Boolean function for nominal attributes, and it is used to test the existence of the difference between two instances for a given attribute; it assigns 1 if the values are different, or 0 if the values are the same. In the case of continues attributes, the diff function has a value between [0, 1. The output of all weights will be between the interval [-1, 1].

In 1994, Kononenko modified the RELIEF algorithm to work on multiple classes [53]. In 1997, Scherf and Brauer proposed a new instance-based technique called Euclidean Based Feature Selection (EUBAFES) [54]. EUBAFES is similar to RELIEF for determining separated clusters by reinforcing similarities between instances occurring in the same class while decreasing the similarity between instances of different classes.

2.4.2.3 *Learning algorithm as a filter for another learning algorithm*

In these filter approaches, researchers used a particular learning algorithm as a filter to determine the best predictor subsets for a primary learning algorithm. In 1995, Cardie used the decision tree algorithm for the feature selection process [55]. K-NN classifier

has been used with the decision trees. This hybrid system achieved a better result than decision trees alone. In 1996, Singh and Provan used a greedy oblivious decision tree algorithm during the feature selection process to construct a Bayesian network [56]. The results showed that the Bayesian network combined with the oblivious decision tree algorithm outperformed the classical Bayesian network only.

In 1995, Holmes and Nevill-Manning used Holte's 1R system to calculate the predictive accuracy of individual predictors [57]. This technique is done without any searching; however, it depends on the user selecting the desired predictors from the ranked list. If we split the data into training data and testing data, the 1R method will be able to calculate a prediction accuracy for each rule and each feature on the training data. The predictors will be reordered due to prediction scores; the highest ranked predictors will be selected with any learning algorithm.

In 1995, Pfahringer used a program to get the decision table majority (DTM) classifiers to choose predictors [58]. DTM classifiers are a type of nearest neighbour classifiers and are produced by greedy searching for search space. DTMs provide highly recommended results when all predictors are nominal. Pfahringer used the concept of minimum description length (MDL) [59]. MDL was used to calculate the cost of encoding a decision table. Other learning algorithms use the predictors that are produced in the final determination table.

2.4.2.4*Principal component analysis*

Principal component analysis (PCA) methodology exchanges the set of the original attribute with a new subset of uncorrelated attributes that represent most of the data [60]. If an attribute misleads the prediction process, it is considered as being noise. A classifier would perform better if the noise in the data were removed. PCA can also be seen as one of the approaches for removing noise from the data. It assumes that directions in the data space along which data varies least are mostly due to noise. PCA is a way of detecting patterns in data by highlighting their similarities and differences between them.

Another advantage of principal component analysis is that once these patterns in the data are found, the data can be compressed, and the number of predictors can be reduced. An example of the use of the principal component analysis is seen in [50], where

Nedevschi et al. employ principal component analysis for feature selection and extraction in image processing with impact on the application's performance.

2.4.2.5 *Discretisation*

In 1995, Setiono and Liu claimed that discretisation could be used in the feature selection process for numeric predictors, and proposed a new algorithm called Chi^2 [61]. A statistical test that can test out ratios is the chi-square or goodness of fit test. For example, Chi^2 is important for any genetic experiment, it can decide if the data fits any of the Mendelian ratios. The formula for Chi^2 test is as follows:

$$x^2 = \frac{1}{d}\sum_{r=1}^{n}\frac{(O_r - E_r)^2}{E_r}$$

Above O_r are the observed values, E_r are the expected values and d is the degrees of freedom. If the predictor can be discretised to a single value, it can be removed from the data. The Chi^2 technique uses the chi-square statistic x^2 test to determine when adjacent intervals should be merged. The extent of the merging process is manipulated by a set x^2 threshold. Setiono and Liu reported a good result after discretisation of domains containing numeric and nominal predictors by using C4.5 [61]. Note that discretisation can be used in general as a transformation of numeric into nominal predictors.

2.4.3 Wrapper methods

Wrapper techniques are different from the filter methods; they search for the optimal subset by using an empirical risk estimate for a particular classifier [62] [63]. Wrapper techniques consist of three main stages, which are the generation procedure, evaluation, and validation procedure. The generation step is a search technology used to select a subset of predictors from the original predictor set. The evaluation stage will measure subsequently the quality of a subset, which we get from the first stage. The selected predictor depends on the evaluation function that has been used. The validation stage validates the selected subset through comparisons obtained from other predictor generations and selection procedures. The last stage is used to identify the best performance from the first two procedures.

2.4.3.1 *Wrappers for decision tree learners*

In 1974, Allen proposed a way to obtain predictors given a credible criterion of a good prediction [64]. In 1994, John et al. supported Allen's approach and suggested using it in the general framework of feature selection in machine learning [65]. John et al. assume that the relevant predictors have two options: strongly relevant and weakly relevant. Otherwise, the predictor should be irrelevant. They claim that the wrapper can find the relevant predictors. When all predictors are considered, predictor x_c will be strongly relevant, if the probability distribution of class values changes when x_c is eliminated. On the other hand, predictor x_c will be weakly relevant, if the probability distribution of class values does not change when x_c is eliminated.

Many research works have been conducted to improve the performance of C4.5. They have also sought to reduce the size of the decision tree. In 1993, Quinlan used the forward selection and backwards elimination search. However, the results show that there is no significant improvement in C4.5 [27]. In 1994, Caruanna and Freitag applied some greedy search, backwards elimination, forward selection, and stepwise bi-directional search with ID3 [52]. Finally, in 1995 Vafaie and De Jong applied the genetic algorithm in a wrapper frame to improve the performance of decision tree learners [66]. This approach comprises two genetic algorithms; the first for feature selection, and the second for constructing inductive models. The results show that this technique improved the performance in many cases. In 1996, Cherkauer et al. proposed a modern technology to improve the accuracy of decision trees, called SET-Gen [67]. The SET-Gen approach uses a genetic search algorithm. The results show that the resulted decision trees are small and accurate.

2.4.3.2 *Wrappers for instance-based learning*

In 1997, Domingos proposed a context-sensitive wrapper approach to feature selection for instance-based learners, called RC [68]. The predictors may be relevant only in the restricted area, relevant given only specific values, or irrelevant. When predictors are calculated globally, the irrelevant aspects of these sorts of predictors may affect the usefulness of the instance-based learner. The RC algorithm can detect and make use of context-sensitive predictors. RC selects a different set of predictors for each instance in training set by using a backwards search and a cross-validation to estimate the accuracy.

RC finds the nearest neighbour in the same class for each instance in the training set and deletes those predictors in which the two differ. Accuracy is estimated by cross-validation. If accuracy is improved, the modified instance will be accepted. Otherwise, the instance will return to its original state and be deactivated. The process of selection continues unit all instances have an active state. The results show that the RC algorithm achieves superior results over standard wrapper feature selectors using forward and backwards search strategies with the instance-based learner. However, when the predictors are globally relevant or irrelevant, then RC will be the same as the standard wrapper feature selection. In addition, classical wrapper techniques can detect globally irrelevant predictors more easily than RC. Furthermore, the wrapper with the RC technique achieves substandard performance on the database that contains many instances.

2.4.3.3 *Wrappers for Naive Bayes classifiers*

It is well known that the Naive Bayes classifier assumes all attributes' probability distributions are conditionally independent. In 1994, Langley and Sage claimed in case of redundant predictors that the Naive Bayes classifier's performance could be improved by removing these predictors [62]. A forward search was applied to the Bayes classifier for the feature selection process. The results show a good increase the learning rate. In 1995, Pazzani combined feature selection and constructive inductive in a wrapper framework to improve the performance of Bayes classifiers [69]. He tested the new technique and found an improvement in the Naive Bayes classifier. In 1995, Kohavi reported an improvement in the Bayes classifier using wrapper-based feature selection [70].

2.5 Estimating the model performance

This section gives details of several performance measures. These performance measures are used to assess the quality of machine learning approaches and prediction models.

The confusion matrix is one of the most used performance measures. The confusion matrix is used to analyse how well the classifier can recognise different classes [71]. A general representation of the confusion matrix is shown in Table 2:1. TP and TN mean the classifier gave a true prediction, while FP and FN mean the classifier gave a

false prediction. The confusion matrix is used to produce other performance measurements such as accuracy, recall, precision, F1, Kappa, etc. These performance measures are discussed in detail in the following subsections.

	Yes	**No**	**Total**
Yes	TP	FN	P
No	FP	TN	N
Total	$\bar{P}$	$\bar{N}$	P + N

Table 2:1 The Confusion Matrix and the evaluation measures: true positive TP, true negative TN, false positive FP, false negative FN, positive P, and negative samples N.

2.5.1 Accuracy and error rate

Accuracy is one of the most well-known performance assessment techniques for prediction problems. The accuracy of the model is defined as the rate of correctly classified instances. It can be calculated from the confusion matrix as follows:

$$Accuracy = \frac{TP + TN}{P + N}$$

Although most prediction algorithms are using accuracy to measure their performance, sometimes the accuracy may be a misleading performance measure. For example, if we have a dataset that has an output (class) attribute very skewed such that instances are distributed as 80% belonging to class A and 20% to class B, if the two classes have equal importance, then the algorithm that has predicted all instances in class A will have 80% accuracy. In this case, we would prefer an algorithm with less accuracy, but that can predict some of the instances in class B.

The error rate, which is also the misclassification rate, is just the complement of the accuracy 1-accuracy. Besides, it could be computed from the confusion matrix as follows:

$$Error\ Rate = \frac{FP + FN}{P + N}$$

2.5.2 Recall

Recall is known as sensitivity in the medical field or as the true positive rate. Recall measures the proportion of the actual positives that are correctly classified [1]. For instance, recall may refer to the percentage of sick patients who were correctly classified. Recall could be calculated from the confusion matrix as follows:

$$Recall\ (\ sensitivity)\ = \frac{TP}{TP + FN}$$

We note that the recall of the negative class is called specificity, and this is a symmetrical measure with respect to sensitivity if we change the focus on the negative class.

2.5.3 Precision

Precision is defined as the proportion of the true positives against all the positive results including false positive. For example, precision refers to the percentage of sick patients who were correctly classified as having a particular disease among the total of people who were actually sick. Precision is calculated from the confusion matrix as follows:

$$Precision\ = \frac{TP}{TP + FP}$$

2.5.4 F-measure

F-measure is the harmonic mean of precision and recall and is known as F-score or F1 score. F1 is calculated from the precision and recall as follows:

$$F1\ = 2 * \frac{precision \times recall}{precision + recall}$$

The F-measure is used to measure the effectiveness of a classifier. It ignores the TN, which can vary without affecting the statistic.

2.5.5 Area under the ROC curve

The receiver operating characteristic (ROC) curve graphically displays the trade-off between the true positive rate and the false positive rate of a classifier. The ROC curve is created by building a graph in which TP is plotted along the y-axis and FP is plotted along the x-axis as shown in Figure 2:3.

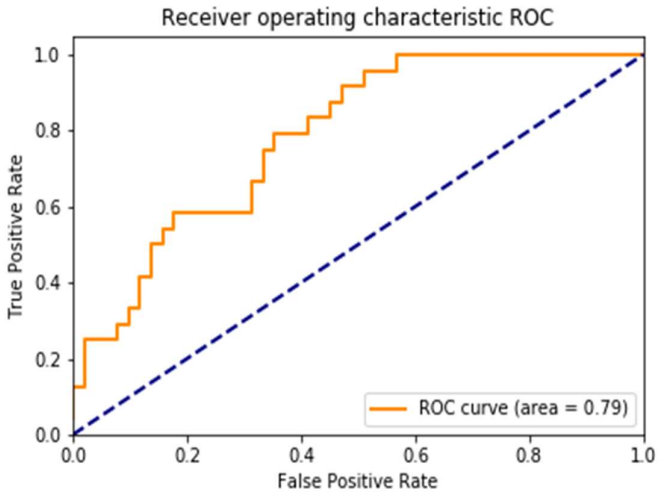


Figure 2:3 The ROC curve.

AUC is the area under the ROC curve with a value between 0 and 1 [72]. Note that, because random guessing produces the diagonal dashed line between (0, 0) and (1, 1), which is a curve corresponding to an AUC of 0.5, no authentic classifier should have an AUC value of 0.5 or less. The AUC is equivalent to the Wilcoxon test of ranks [73]. AUC is usually used for model comparison. Note that some representations of the ROC curve display the sensitivity on the y-axis and (1-specificity) on the x-axis, which is entirely equivalent to the representation in Figure 2:3 that uses, for illustration of varieties of representations, alternative names for the same quantities.

2.5.6 Kappa

Kappa, which is also called Cohen's Kappa, is a statistical measure that assesses the interrater agreement for categorical items [2]. Kappa takes into account the accuracy that could be possibly occurring by chance. The Kappa equation is as follows:

$$Kappa = \frac{O - E}{1 - E}$$

Above O is the obsereved accuracy and E is the expected accuracy. Kappa values range between -1 and 1. When Kappa equals to 0, this means there is no agreement between the predicted and the actual classes. In contrast, when Kappa has a value of 1, it shows excellent concordance of the model prediction and the observed classes.

When the class distributions are equivalent, the overall accuracy and Kappa are proportional. Depending on the context, Kappa values within 0.30 to 0.50 indicate reasonable agreement [2]. However, if Kappa was less than 0.30, it indicates that the model's performance occurs mostly by chance only and the accuracy does not reflect how good the model is.

2.6 Resampling techniques

In most cases resampling methods for estimating model performance are similar. A portion of samples are used to build a model, and the remaining samples are used to evaluate the model performance. This process is iterated several times, and the results are combined and summarised. The variation in the methodologies depends on which way the samples were chosen for training and testing. This section gives an overview of the most used resampling techniques in the field of machine learning.

2.6.1 Cross-validation method/leave one out validation

In the simplest cross-validation approach, the data is divided into two parts: a training set and a testing set. This operation is called two-fold validation. Each part of the details used once for training and once for testing. Han et al. provided an overview of a technique called k-fold cross-validation [1]. In general, the cross-validation method partitions the data into equal sized k subsets. One subset is used for testing, and the rest of the sets are used for training. This process is repeated until each subset is tested once. Finally, we can measure the performance either by the average of measures calculated in each iteration or by the basis of the overall number of correct predictions from all iterations. Figure 2:4 shows a simple representation of the five-fold cross-validation methodology.

The leave one out cross validation (LOOCV) approach is considered to be a particular case of the cross-validation method [74] in which each instance is used once as the test case, and the rest of instances are used as the training set. In other words, if our dataset contains n instances, then (n-1) instances are used for the training set and the nth instance for the test case. This method is most often used for small datasets because of its expensive nature. The studies by Guyon [37] and Kohavi [74] use LOOCV to validate their models where they have proven its ability.

Testing Data	Training Data	Training Data	Training Data	Training Data	K = 1
Training Data	Testing Data	Training Data	Training Data	Training Data	K = 2
Training Data	Training Data	Testing Data	Training Data	Training Data	K = 3
Training Data	Training Data	Training Data	Testing Data	Training Data	K = 4
Training Data	Training Data	Training Data	Training Data	Testing Data	K = 5

Figure 2:4 Five-fold cross-validation.

2.6.2 Monte Carlo cross-validation

Repeated cross-validation or Monte Carlo cross-validation is a technique that simply creates multiple splits of the data for training and testing the models [2]. Non-repeated validation could introduce some bias, especially in small datasets. Repeating the sampling a number of times is essential because this decreases the uncertainty of the calculated performance.

For example, by splitting the data into two random parts, we can easily introduce some bias. However, by repeating the sampling 100 times, we can at least ensure that we covered more data with less bias since the randomness method controls the proportion of the data going into each subset. However, at some point, the increase in the number of iterations will cause it to lose its effectiveness. In addition, it is also good to note that when the number of iteration increases, the cost of the method increases as well.

2.6.3 Holdout and random subsampling

In the holdout method, the original data set is divided into two parts: a training set and a test set [1]. The sets can be randomly selected for instance as 50% for each set, or 2/3 for training and 1/3 for the test set, depending on the choice of the analyst. The training set is used to build up the prediction model, and the test data is used to test its performance. Selecting a subset of individuals from the whole dataset is known as a sampling process. Random subsamples are selected several times, and the performance will be measured each time, such as accuracy as the average of the measures obtained in each iteration. The holdout technique is usually applied to large datasets.

2.6.4 Bootstrap method

Han et al. presented another evaluation approach called bootstrap technique [1]. In the bootstrap approach, the training data is sampled uniformly with replacement, which means each time an instance is selected it is equally likely to be chosen again and added back to the training set. Hence, bootstrap allows the instances to be selected more than once. There are many existing bootstrap approaches such as 0.632 bootstrap. In 0.632 bootstrap the data will be sampled n times, where n is the number of instances in the original dataset. The training dataset will comprise in average 63.2% of the original instances, and the rest of the instances, 36.8% in average, will form the test set. The accuracy of the overall model can be shown as:

$$Acc\ (M) = \sum(0.632 \times Acc\ (M_i)_{testset} + 0.368\ \times Acc\ (M_i)_{trainset})$$

2.6.5 Boosting and AdaBoost

In 1995, Freund and Schapire proposed a new AdaBoost algorithm [75]. The boosting approach has been used to improve the performance of any learning algorithm. Boosting is trying to run a weak learner such as decision trees and prediction rule on different training data [1]. These weak classifiers are merged into a new single stronger classifier. The main purpose is to achieve higher accuracy than the accuracy of the weak learner's classifiers. The fundamental idea of the AdaBoost algorithm is to assign a weight to each example of the training set. In the first round, all weights are equal. The weights of the correctly classified instances are decreased. However, the weights of all misclassified instances are increased. This process introduces a series of classifiers that complement one another. In 1996, Freund and Schapire described two versions of the AdaBoost algorithm, which are AdaBoost.M1 and AdaBoost.M2 [75].The boosting process improves the performance because of two main reasons. Firstly, the final classifier has an error in the training set smaller than the original classifiers. Secondly, the variance of the newly combined classifiers is also less than the variance produced by the weak learner's classifier.

2.6.6 Bagging

In the bagging approach proposed by Breiman in 1996, multiple classifiers are built concurrently. Each classifier is trained from instances taken with replacement from the training set [76]. In the default case, the sample size is equal to the size of the original training

set. Due to the process of sampling with replacement, some of the original instances may appear more than once, and some may not be included. Bagging is similar to boosting in its purpose; both are used to improve the accuracy of a classifier. The improvement is executed by producing different classifiers and combining multiple models. Both of them use voting to combine the outputs of various predictions of the same type. In boosting, each classifier is influenced by the performance of those built before, and instances are chosen with a probability proportional to their weight. In bagging, each instance is selected with equal probability.

2.7 Model tuning

Most of the machine learning models have some parameters that need to be estimated and/or optimised in order for the models to be applied accurately. For example, when applying the k-NN prediction model, we need to choose the best values for k. The goal here is to find the best value for k that is used for the neighbours without overfitting the models or getting less accurate models. This type of technique is called tuning the model parameters.

There are several methods for tuning the parameters. The most common method so far is to define a grid of values and estimate the model across these values; the value that scores the highest performance is considered the optimal value.

Other strategies for tuning the parameters genetic algorithms [77] and simple search methods [2] also exist. These methods choose the best values for tuning parameters via assessing a large number of models and a defined set of tuning parameters, when model performance can be efficiently calculated. Other studies also show different comparisons of tuning such as Cohen [78]. In practice, grid optimisation can produce comparable results at lower cost.

2.8 Conclusion

A promising new approach is the use of predictive modelling approaches to data-driven computational psychiatry. Computational psychiatry has made it possible to combine enormous levels and types of computation with several types of data in an effort to advance classification of mental disease, predict treatment outcomes, or improve treatment selection [13]. Background information on the machine learning modelling procedure

used to establish the results in this dissertation, as well as strategies to improve them, are described in this chapter. First, several medical datasets were presented to report the advantages and the disadvantages of the machine and statistical learning algorithms. This includes clinical psychiatry data (first-episode psychosis - cannabis clinical dataset) used in our approach to build novel predictions models and to detect new patterns in patients' data. The latest data was used to develop predictive modelling data-driven approach to computational psychiatry to advance classification of mental disease. Then, several data preparation techniques were reviewed in order to deal with the challenges present in the data sets such as dealing with missing values, resolving outliers, and feature selection. Afterwards, several performance measures and resampling methods for estimating model performance were reviewed in order to assess the quality of the prediction models. Finally, this chapter concluded with methods for tuning the prediction models parameters.

Chapter 3 Methodology

3.1 Data processing pipeline

Data pipeline is a set of data processing components connected in series, where the output of one component is the input of the next one. The data pipeline describes the information flow within a framework as a series of steps needed to generate useful insights from data.

This section presents the general data processing framework/pipeline that should elegantly handle different types of data, different data mining tasks, and different types of patterns/models. Choosing the suitable algorithm and developing a model are not the only aspects we need to consider during building prediction models; other data mining phases such as data pre-processing and model post-processing are also involved. Therefore, extracting knowledge from data involves several stages namely; data preparation, data pre-processing, data processing and data post-processing.

The first stage, which is the data preparation stage has several sub-processes such as Rationalisation, Refinement, Random shuffling, Stratified Sampling, Cross-validation methods, K fold cross testing and Monte Carlo simulation. The second stage is the data pre-processing stage include several phases such as missing values imputation, feature selection, balancing classes, cantering, scaling, etc. The processing stage consists of some sub-processes such as model generation, tuning and building, and evaluating the output model. Model generation and tuning are some of the most critical sub-processes. The model generation and tuning are iterative processes comprising three steps: choosing the algorithm and its parameters, building the model, and evaluating the model. The goal of this process is to find the best parameter values for the model and thus assess the performance of an algorithm for the problem at hand [23]. The last stage is the post-processing stage, which is responsible for knowledge presentation and, improving the model performance.

Figure 3:1 gives an overview of the data processing pipeline used through the thesis. The first two stages were explained in detail in Chapter 2, the rest of the stages will be illustrated in this chapter, focusing on statistical and machine learning methods that are a key component of the proposed synergistic approach.

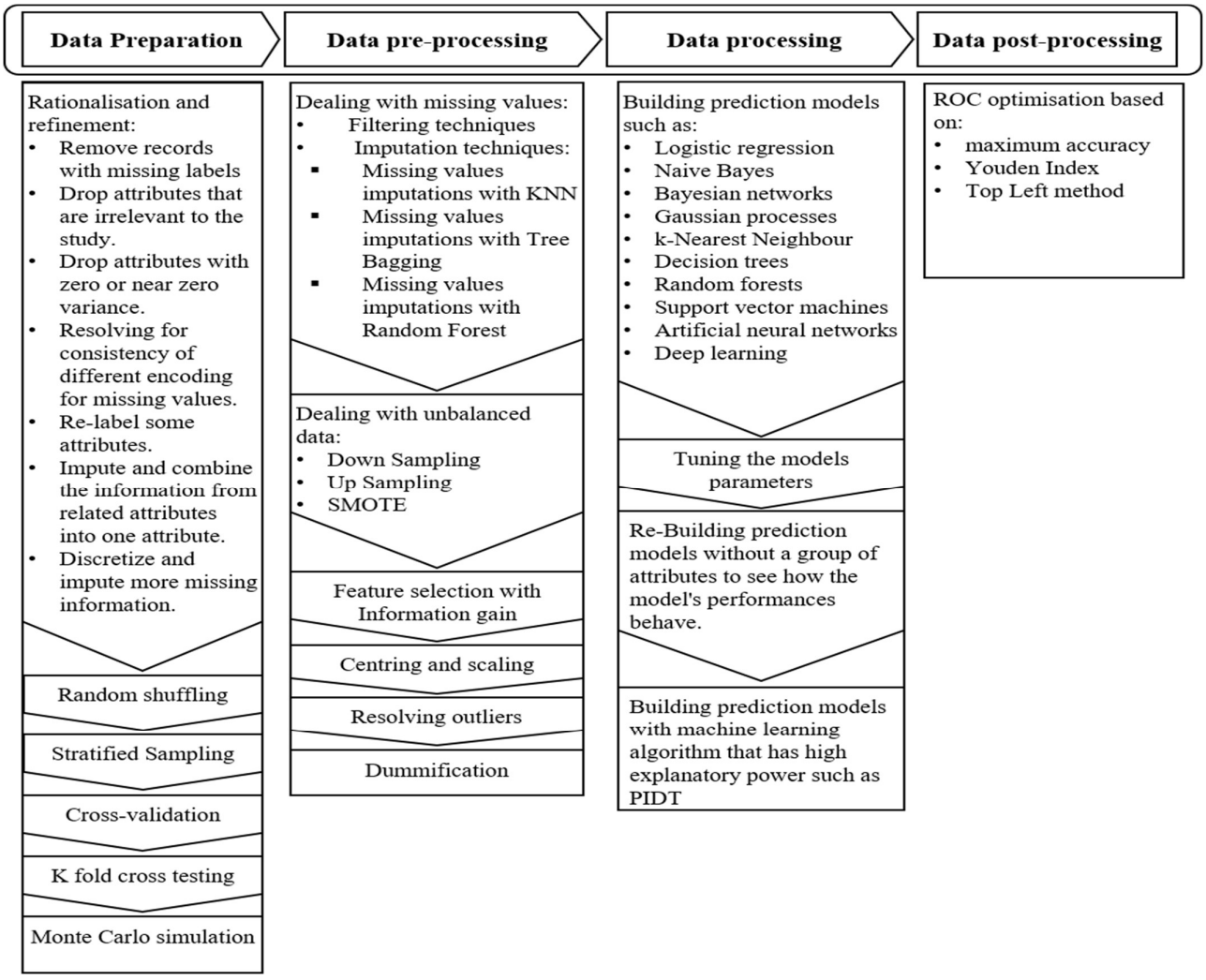


Figure 3:1 Data processing pipeline.

3.2 Statistical and machine learning models

Machine learning algorithms have already begun to prove their particular capabilities and contribution to medical research and applications [4]. In particular, machine learning techniques have been successfully used in diagnosing psychosis [17], analysing diabetic patients' data [18] [19], classifying leukaemia [20], and detecting heart conditions in ECG data [21], etc. These studies show that machine learning has proven to be capable of dealing with challenging medical data, in particular with the ambiguous nature of the ECG

signal data, for which machine learning algorithms show outstanding results compared to other methods [20] [21].

In this chapter, we discuss and summarise different machine and statistical learning techniques that are suitable for use in the medical field especially in psychiatry. Medical research involves many problems that benefit from analysing data based on techniques of data pre-processing, predictive modelling, clustering, and so on. In particular, in predictive modelling, the task is to predict the outcome associated with a particular patient, given a feature vector describing that patient. In clustering, patients are grouped because they share similar characteristics, and in data pre-processing operations such as feature selection, the task is to select the most relevant attributes to predict the outcome for a patient [2].

In the last three decades, many algorithms have been proposed in the machine learning research field, such as decision trees, support vector machines, and deep learning networks. Other techniques were also produced in the field of statistics, including methods for understanding the relationship between input and output variables; such techniques have also been adopted and improved by the machine learning community [79]. This chapter provides a literature review of the existing algorithms for predictive modelling in the fields of statistical and machine learning.

Mathematical formulations:

Let X be an n $\times$ m data matrix. We denote the r^{th} row vector of X by X_r, and the c^{th} column vector of X by X_c. Rows are also called records or data points, while columns are also called attributes or features. Since we do not restrict the data domain of X, the scale of this domain's features can be categorical or numerical. For each data point X_r, we have a label y_r. For classification problems, we assume a set of known class labels Y, so $y_r \in Y$. Let D be the set of labelled data $D = \{(X_r, y_r)\}_{r=1}^{n}$. During the classification task, the goal is to predict the labels of new data points by training a classifier on D. For regression problems, Y is a numeric set and the purpose of a learner is to predict a numeric value for the outcome variable.

3.3 Linear regression

Linear regression is the most traditional and popular methodology in both statistical and machine learning. Linear regression was initially proposed in the field of statistics as a method for understanding the linear relationship between input and output variables, and it was largely adopted as a machine learning technique [79]. Linear regression assumes a linear relationship between the input variables X and the output variable Y. This means that Y can be calculated from a linear aggregation of the input variables X [80]. Figure 3:2(a) shows a simple linear regression model, which is when there is only one input variable. Otherwise, if there are multiple input variables, the method is referred to as multivariate linear regression [81] as shown in Figure 3:2(b). The simple linear regression model can be presented as:

$$Y = \beta_0 + \beta_1 x$$

Here β_0 and β_1 are the coefficients that we must estimate from the training data. Briefly, the estimated coefficients could be presented as follows:

$$\beta_1 = \frac{\sum_{r=1}^{n}(x_r - \bar{x})(y_r - \bar{y})}{\sum_{r=1}^{n}(x_r - \bar{x})^2}$$

$$\beta_0 = \bar{y} - \beta_1 \bar{x}$$

Another way to calculate the estimation of the coefficients is to calculate the statistical properties of the data such as variance and covariance.

$$variance = \frac{\sum_{r=1}^{n}(x_r - \bar{x})^2}{n-1}$$

$$covariance = \frac{\sum_{r=1}^{n}(x_r - \bar{x})(y_r - \bar{y})}{n-1}$$

By calculating the statistical properties, the estimation of the coefficients can be simplified to:

$$\beta_1 = \frac{covariance\ (x, y)}{variance\ (x)}$$

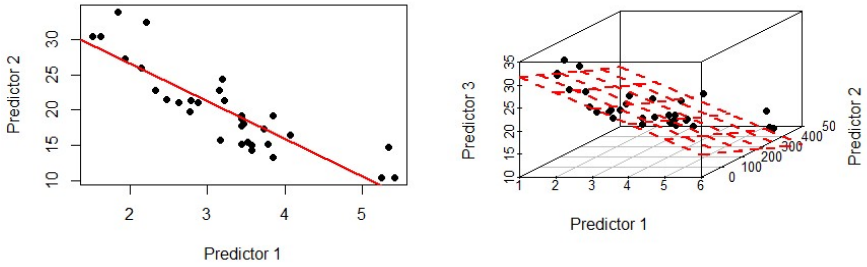
$$\beta_0 = \bar{y} - \beta_1 \bar{x}$$

Linear regression is a procedure where a straight line is used to model the association between the input and output [2]. If we have more than two dimensions, this straight line will be considered to be a hyperplane. Predictions are obtained by using a combination of the input values to predict the output value. Each input attribute x is

associated with a coefficient β, and the goal of the learning algorithm is to determine the coefficients that result in accurate predictions Y.

$$Y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots$$

Coefficients can be found using methods such as stochastic gradient descent. Gradient Descent is the process of minimising a function following the slope or gradient of that function [81]. In machine learning, a technique called stochastic gradient descent is used to evaluate the coefficients in such a way as to minimise the error of a model on the training data.



(a) Simple linear regression (b) Multivariate linear regression

Figure 3:2 Linear regression.

The way this optimisation algorithm works is to consider each training record one at a time. First, the model forms a prediction for a training record and calculates the error. Then, the model is updated to reduce the error for the next prediction. This process is repeated for some number of iterations. The equation for updating the coefficients (β) at each iteration in machine learning language is as follows:

$$\beta = \beta - learning\ rate\ \times error \times x$$

Here β is the coefficient being optimised, $learning\ rate$ is a learning rate that must be specified (e.g., 0.05), $error$ is the prediction error for the model on the training data attributed to the coefficient, and x is the input value. This optimisation algorithm is used to find the set of coefficients in a model that produces the least error for the model on the training data set. Figure 3:3 shows a simple estimation of how the linear regression model will perform on three synthetic datasets using scikit-learn 0.19.1 (October 2017) [24]. The plots show training points in solid colours, testing points in semi-transparent colours and the accuracy of each model is in bold black.

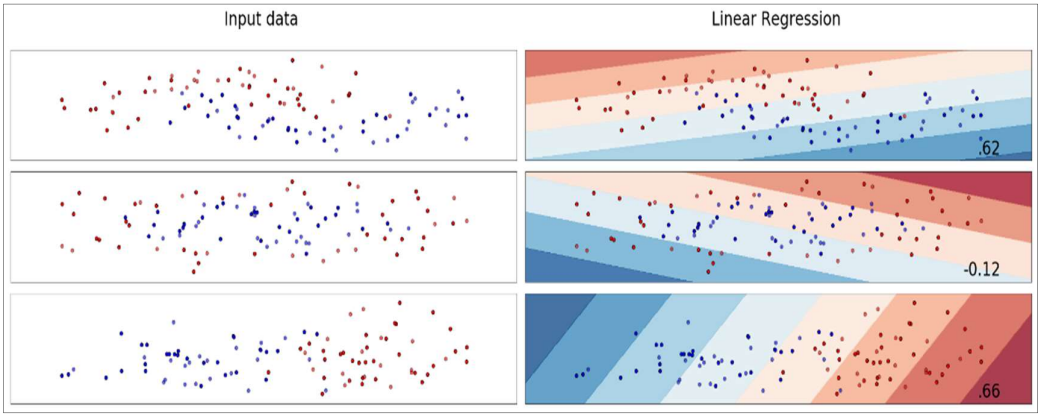


Figure 3:3 Linear regression on generated datasets.

3.4 Penalised models

Usually, the coefficients produced by the least squares regression are unbiased. However, by allowing the parameter estimates to be biased, the variance will be lesser, and it is possible to produce models with smaller MSEs. One popular method for creating biased regression models is controlling (or regularising) the parameter estimates such as in ridge regression, lasso regression, and elastic net regression. Controlling the parameter estimates can be accomplished by adding a penalty to the sum of the squared errors when the parameter estimates become significant.

One popular penalised model is the ridge regression, which adds a penalty, proportional to the sum of the squared coefficients, to the sum of the squared residuals as follows:

$$SSE_{L1} = \sum_{r=1}^{n}(y_r - \hat{y}_r)^2 + \lambda \sum_{c=1}^{m}\beta_c^2$$

Here y_r is the r^{th} the observed value of the outcome and $\hat{y}_r$ is the predicted outcome of r^{th} data point, $r = 1 \ldots n$, and λ is the model's parameter.

A compelling alternative to ridge regression is the least absolute shrinkage and selection operator model, which called the lasso regression. Lasso regression adds a penalty similar to the ridge regression penalty:

$$SSE_{L2} = \sum_{r=1}^{n}(y_r - \hat{y}_r)^2 + \lambda \sum_{c=1}^{m}|\beta_c|$$

This type of regularisation has been extended to many other methods, such as linear discriminant analysis [82] [83], PLS [84], and PCA [85].

A combination approach of the lasso regression and ridge regression is the elastic net regression [86]. This adds both types of penalties and could produce models that are more accurate. Elastic net regression is represented as follows:

$$SSE_{L2} = \sum_{r=1}^{n}(y_r - \hat{y}_r)^2 + \lambda_1 \sum_{c=1}^{m}\beta_c^2 + \lambda_2 \sum_{c=1}^{m}|\beta_c|$$

Both ordinary and penalised regression models are quite popular. On the one hand, ordinary linear regression finds parameter estimates that have a minimum bias. On the contrary, ridge regression, lasso regression, and elastic net regression find estimates that produce lower variance.

3.5 Logistic regression

Logistic regression is another popular linear prediction algorithm for two-class problems, which is easy to implement and understand. In addition, it has proven its ability for getting good results on a wide variety of prediction problems.

The logistic function is the core of the logistic regression classifier. Logistic regression is very much like linear regression. It linearly combines the input values (X) using the coefficient values to predict an output value ($\hat{y}$) which is given by:

$$\hat{y} = \frac{e^{\beta_0+\beta_1 x_1}}{1 + e^{\beta_0+\beta_1 x_1}}$$

Here e is the base of the natural logarithm, β_0 is the intercept term, and β_1 is the coefficient of the single input value (x_1). The$\hat{y}$ prediction is a real value between 0 and 1 that needs to be rounded to 0 or 1.

Each column in the input data has an associated β coefficient that must be estimated from the training data. The actual representation of the model are the coefficients in the equation. Similar to linear regression, logistic regression employs gradient descent to update the coefficients. For each gradient descent iteration, the coefficient β is updated using the equation:

$$\beta = \beta + (Learning\ rate \times (y - \hat{y}) \times \hat{y} \times (1 - \hat{y}) \times x)$$

Figure 3:4 shows a simple estimation of how the logistic regression model will perform on three synthetic datasets namely moons dataset, circles dataset, linear dataset using a package called scikit-learn 0.19.1 (October 2017) [24]. Each of these three da-

tasets has some noise added in order to simulate real situations in which data and classifications are not perfect. The plots show training points in solid colours, testing points in semi-transparent colours and the accuracy of each model is in bold black. Figure 3:4 shows that logistic regression can efficiently deal with linear and moon datasets. However, it fails to separate the circle data.

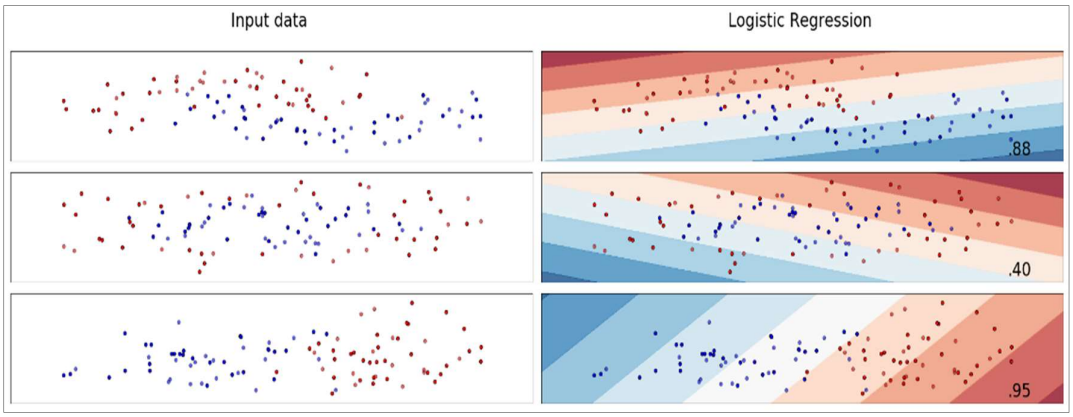


Figure 3:4 Logistic regression on generated datasets.

3.6 Naive Bayes

The Naive Bayes technique is a method that uses the probabilities of each attribute belonging to each class to make a prediction. Naive Bayes is based on using Bayes' theorem with strong independence assumptions between the attributes. Simply, the model assumes that all of the predictors are conditionally independent of the others. Bayes' rule is used to compute the posterior probability distribution of the example's classification. Bayes' rule is stated as:

$$P(y|X) = \frac{P(X|y)\,P(y)}{P(X)}$$

Here $P(y|X)$ represents the probability of the class given the provided data. Naive Bayes is a simple but surprisingly powerful predictive modelling technique with many advantages. It is fast to train and classify, and it is insensitive to irrelevant features. Figure 3:5 shows a simple illustration of how the model performs on three synthetic datasets. In practice, the Naive Bayes classifier works well even when the independence assumption does not hold.

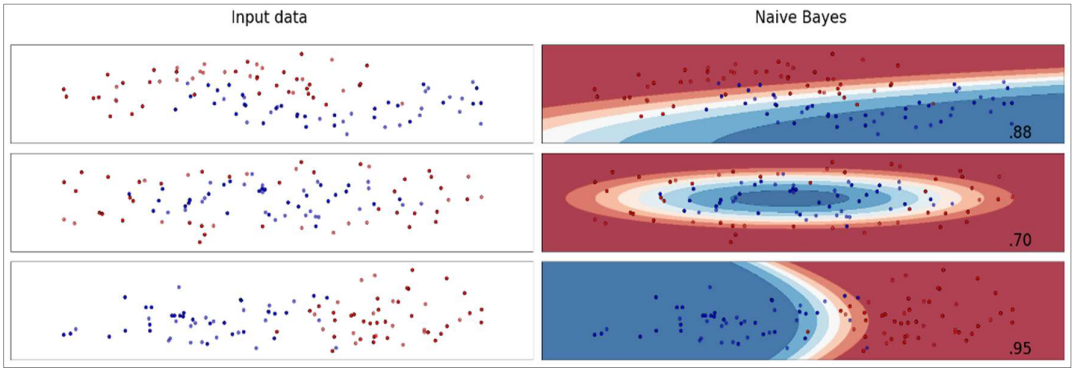


Figure 3:5 Naive Bayes on generated datasets.

3.7 Bayesian networks

A Bayesian network is a directed acyclic graph where each node represents a random variable that is linked with edges that represent direct dependencies among the nodes [87]. The network consists of nodes representing the random variables, edges between pairs of nodes representing the causal relationship of these nodes, and a conditional probability distribution in each of the nodes [87]. The structure of the network should capture the qualitative relationships between variables. In particular, two nodes should be connected directly if one affects or causes the other. The constructed directed acyclic graph has to include conditional probability distributions by the Bayes' rule for each node in the graph. For nominal attributes, we can represent the conditional probability distributions as a table that lists the probability that parent node takes on each of its different values for each combination of values of its children nodes.

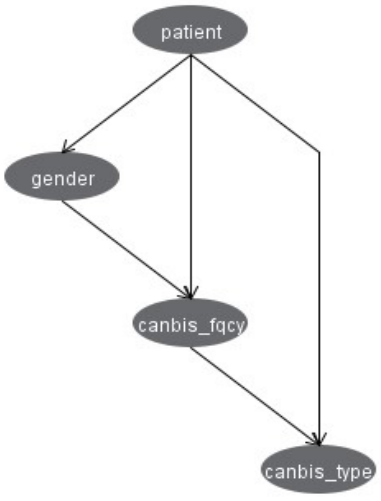


Figure 3:6 Bayesian network

Figure 3:6 illustrates a simple Bayesian network for the cannabis – first episode psychosis dataset introduced earlier in this chapter. The figure shows that first-episode patients are influenced by gender, cannabis frequency and cannabis type variables. In addition, the conditional probability distributions are shown in Table 3:1 and Table 3:2. It is important to note that this is just one possible structure for the problem; we consider alternative network structures in chapter 4.

Patient	Cannabis frequency	Skunk	Never used	Hash
patients	Daily	0.737	0.132	0.132
patients	Only at weekends	0.589	0.084	0.327
patients	Never used	0.042	0.937	0.022
controls	Daily	0.369	0.329	0.302
controls	Only at weekends	0.273	0.507	0.22
controls	Never used	0.032	0.91	0.058

Table 3:1 Probability distribution table for cannabis type

Patient	gender	Only at weekend	Never used
patients	0.374	0.053	0.573
patients	0.176	0.171	0.654
controls	0.13	0.276	0.594
controls	0.082	0.222	0.696

Table 3:2 Probability distribution table for cannabis frequency

3.8 Linear and quadratic discriminant analysis

Linear discriminant analysis and quadratic discriminant analysis are two classification techniques that concern a linear and a quadratic decision surface, respectively. These classifiers are not complicated because they can be directly calculated, and they have no hyperparameters to optimise. Linear discriminant analysis (LDA) and quadratic discriminant analysis (QDA) have proven their ability in practice with many applications.

Both LDA and QDA can be determined from simple probabilistic calculations that model the class conditional distribution of the data $P(X|y=k)$ for each class k [2]. Predictions can then be retrieved by using Bayes' rule:

$$P(y=k|X) = \frac{P(X|y=k)\,P(y=k)}{P(X)}$$

$$= \frac{P(X|y=k)\,P(y=k)}{\sum_l P(X|y=l)\,P(y=l)}$$

Then, the class k will be chosen to maximise this conditional probability. In particular, for linear and quadratic discriminant analysis, $P(x|y)$ is modelled as a multivariate Gaussian distribution with density:

$$P(X|y=k) = \frac{1}{(2\pi)^n \,|\sum_k|^{1/2}} \, e^{\left(-\frac{1}{2}(X-\mu_k)^t \sum_k^{-1}(X-\mu_k)\right)}$$

While training the models on the training dataset, we need to estimate the class priors $P(y=k)$ (by the proportion of instances of class k), the class means μ_k (by the sample class means) and the covariance matrices (either by the empirical sample class covariance matrices, or by regularised estimators).

In the case of LDA, the Gaussians for each class are assumed to share the same covariance matrix: $\sum_k = \sum$ for all k. This leads to linear decision surfaces in between, as can be seen by comparing the log-probability ratios $\log\left(\frac{P(y=k|X)}{P(y=l|X)}\right)$

$$\log\left(\frac{P(y=k|X)}{P(y=l|X)}\right) = 0 \;\leftrightarrow (\mu_k - \mu_l)\sum X = \frac{1}{2}(u_k^t \sum \mu_k - \mu_l^t \sum \mu_l)$$

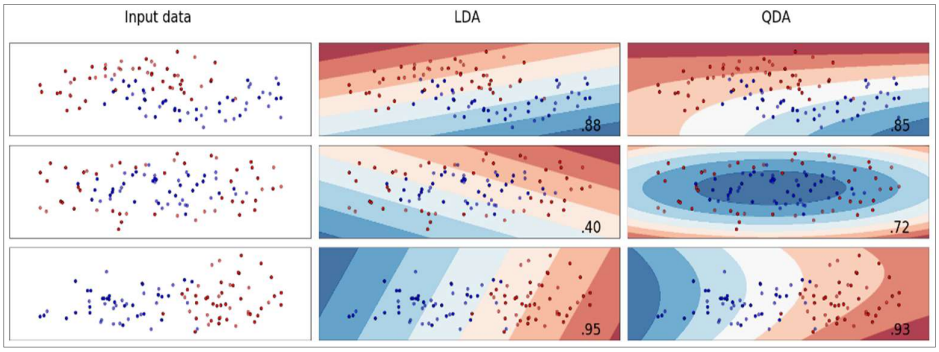


Figure 3:7 LDA and QDA on generated datasets

In the case of QDA, there are no assumptions on the covariance matrices $\sum_k$ of the Gaussians, leading to quadratic decision surfaces. If in the QDA model one assumes that the covariance matrices are diagonal, then the inputs are assumed to be conditionally independent in each class, and the resulting classifier is equivalent to the Gaussian Naive Bayes classifier. Figure 3:7 shows an illustration of how the discriminant analysis models will perform on three generated datasets namely moons dataset, circles dataset, linear dataset using a package called scikit-learn 0.19.1 (October 2017) [24] The plots show

training points in solid colours, testing points in semi-transparent colours and the accuracy of each model is in bold black.

3.9 Gaussian processes

The Gaussian process technique is one of the latest utilised methods in machine learning applications [88]. With Gaussian process, instead of creating a single function using an optimised weight vector, we yield a distribution of all possible functions given the training data [89]. To do this, we produce any collection of random variables where a random subset of variables has a joint Gaussian distribution. This technique is built on the assumption that all the attributes have Gaussian distributions. Gaussian process empowers the model training by using the covariance matrix of the joint distribution [89].

When predicting a value, the approach generates a Gaussian distribution with a covariance matrix produced by a kernel function. The kernel function describes how a feature y of a function changes depending on how all other features x change. In a Gaussian process, the model trains on parameters of the kernel function, instead of weight vector as traditional regressions. The most common kernel functions are the sigmoid and squared exponential covariance with all its various forms. Given a dataset $D = \{(X_r, y_r)\}_r^n = 1$, with binary class labels $y_r \in \{-1, +1\}$, we infer class label probabilities at new points. Figure 3:8 illustrates a Gaussian process predictive model. Note that the best predictive probabilities relay on the relative density of the two classes, and not on the absolute density.

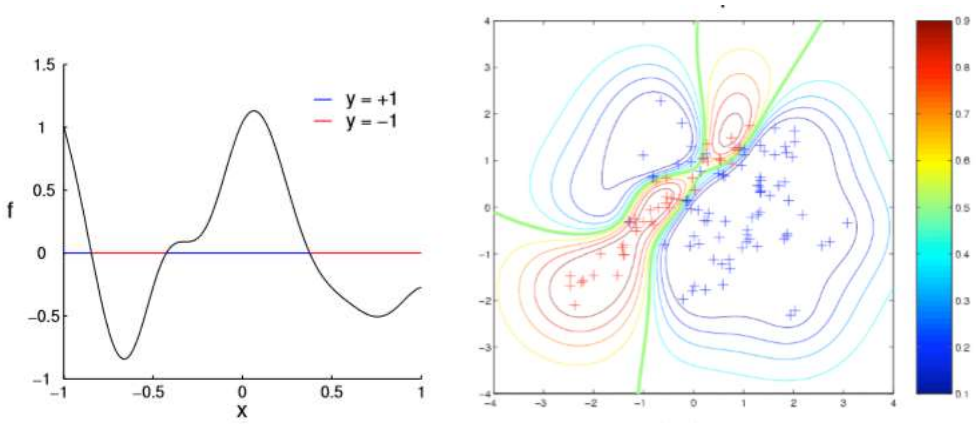


Figure 3:8 Gaussian Process classifier

3.10 k-Nearest Neighbour

K-Nearest Neighbour (k-NN, lazy learning method) is one of the instance learning techniques. Instance learning methods have three major properties. Firstly, during the learning process, they store all training data. Secondly, every new query is answered by comparing the new case to the training data. Finally, for each new case, a searching process in the training data for a similar case is executed. The k-NN prediction approach provides a typical simple example of methods working on non-parametric data (for which there is no prior knowledge of the statistical distribution of the data). The k-NN prediction process assumes the class of an instance is the same as the class of the nearest instance. K-NN uses a similarity metric to measure the proximity of an instance to another. Hence, the instances that have close proximity are classified in the same class [90].

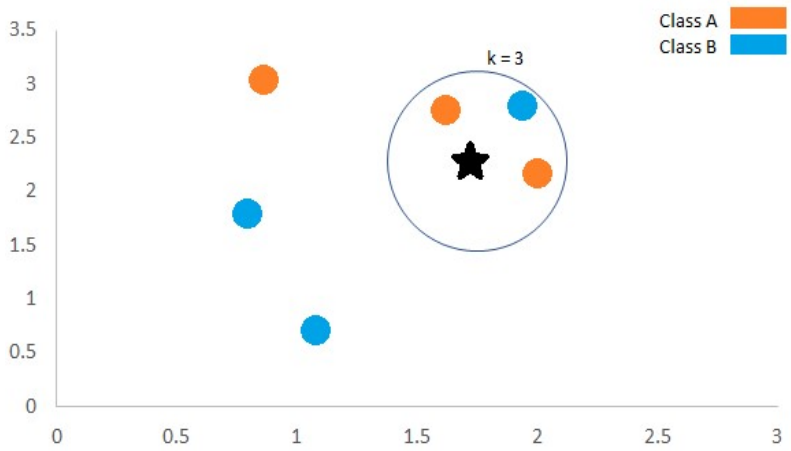


Figure 3:9 K-Nearest Neighbour

K-Nearest Neighbour rule

The k-NN algorithm is a very straightforward method. The entire training dataset is stored. When a prediction is required, the k-most similar records to a new record from the training dataset are then located. From these neighbours, a summarised prediction is made. The similarity between records can be measured in many different ways. A problem or data-specific method can be used. Generally, with tabular data, a good starting point is the Euclidean distance as shown in Figure 3:9.

Once the neighbours are discovered, the summary prediction can be made by returning the most common outcome or taking the average. As such, k-NN can be used for prediction problems. Figure 3:10 shows an illustration of how the model performs on three simple datasets. The plots show training points in solid colours, testing points in semi-transparent colours and the accuracy of each model is in bold black.

In the k-NN rule, the K nearest instances are considered. In this approach, distance measures play the most important rule. Some of the most used measures are:

Euclidean distance: This is defined for real values and is based on the following equation.

$$d(x', x'') = \sqrt{\sum_{r=1}^{m} (x'_r - x''_r)^2}$$

Minkowski distance: This is defined for real values and a given q, and is computed utilising the following equation.

$$d(x', x'') = \left(\sum_{r=1}^{m} |x'_r - x''_r|^q\right)^{1/q}$$

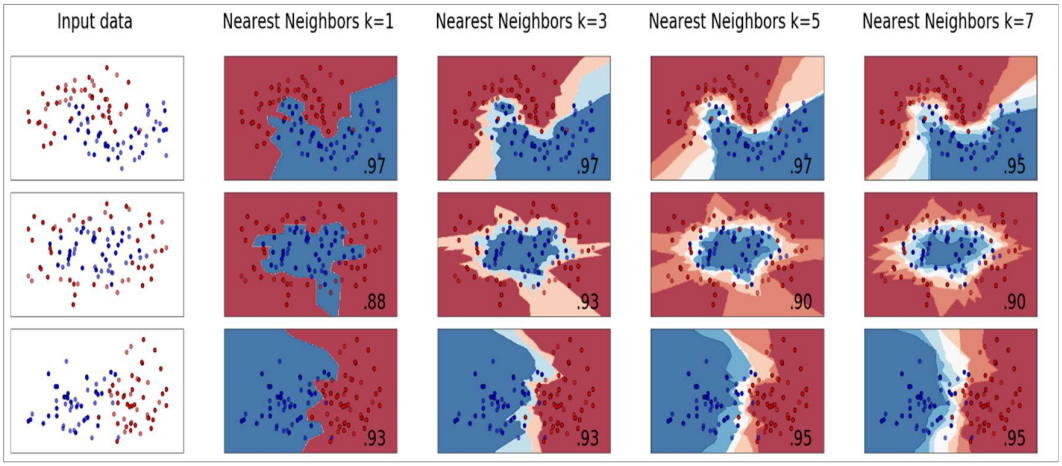


Figure 3:10 K-Nearest Neighbour on generated datasets

3.11 Decision trees

Decision trees are a popular and widely used machine learning technique that implement the 'divide and conquer' approach [1]. Decision trees can be used for classification or

regression tasks, as an efficient non-parametric (from the point of the data) method. It has a hierarchical data structure, and the input space is split into local regions to predict the dependent variable [91]. The representation of a decision tree can be written as G= (V, E). It consists of the finite set of nodes (V) and a set of edges E. According to [91] the graph must be directed, which means the edges must be ordered pairs of vertices (v, w). The graph must be acyclic, which means it has no cycles. There is only one root node, which does not have any edge enter, and every other node has only one entering edge. There is exactly one path (sequence of edges) from the root node to any other node. The node that does not have a descendant is called leaf (terminal node). Otherwise, it is called an internal node, except for the root node as shown in Figure 3:11.

The root and internal nodes represent a test over a given data set attributes, and the edges correspond to the possible outcomes of the test. Terminal nodes can hold class labels for a classification process, continuous values for a regression process, or even models produced by other learning algorithms. The process of getting a prediction outcome starts from the root node. It navigates through the decision tree and follows the edges according to the results from the attribute tests. The prediction result will be received at the leaf nodes.

The decision tree has another important definition, which is the concept of depth and breadth. The average number of levels in the decision tree is known as the average depth of the tree. The average number of nodes in each level of the tree is known as the average breadth. The new definition helps us to assess the tree complexity. If the values of depth and breadth are high, then the complexity of the decision tree will also be high. The creation process of the optimal decision tree is considered to be a complex and a hard task. In 1976, Hayfil reported that producing a minimal binary tree, regarding the expected number of tests, is an NP-complete problem. In 1996, Hancock et al. showed that constructing a minimal decision tree consistent with the training set is an NP-complete problem [92]. In 2000, Zantema and Bodlaender reported that finding a minimal decision tree for a given decision tree is an NP-complete problem [93]. These research works claimed that building an optimal decision tree using a brute-force approach is a complicated issue. Efficient techniques have been developed as heuristics to overcome the problem of growing decision trees. Such heuristics are based on a top-down induction method for instance.

One of the important works conducted within a top-down induction approach is Hunt's concept learning system framework (CLS) [94]. CLS is used for the prediction process to minimise the cost of classifying an object. In other words, it can minimise the cost of determining the value of a certain attribute exhibited by the object, and the cost of classifying an object to the right class if it was originally classified to the wrong class. Hunt's technique can be recursively represented in two steps. Assume X to be the set of training instances associated with node t and the class labels represented by k = $\{k_1, k_2, k_3, \ldots k_i\}$ [79]. If all the instances in X_t belong to the same class k_t $t$hen t is a leaf node that will be labelled k_t. If X_t contains the instance that belongs to more than one class, the attribute test condition is done to partition the instances into smaller subsets. A child node is created for each outcome of the test condition, and the instances will be distributed across the nodes based on the results. Then the algorithm is recursively applied to each child node [94].

Hunt's algorithm is almost the basis for all top-down decision tree induction algorithms. Many improvements have been made to Hunt's algorithm. The most critical problem is the stopping criterion; as mentioned in the algorithm the stopping point needs all leaf nodes to be pure. This issue may cause an over-fitting problem. To solve this problem, we try to find the minimum level of impurity that could be reached. In addition, we can process a pruning step after the tree has been grown. Another critical question is how to select the attribute test condition to partition the instances into smaller subsets.

The process of creation of the decision tree is to produce patterns to help us classify unseen data in the future. These patterns are placed in the root node in which the tree starts growing. The most important point in creating top-down induction decision trees is selecting attributes for splitting a node into subsets. There are two common types of decision trees: univariate (axis-parallel) and multivariate (oblique). The univariate's goal is to choose the attributes that better discriminate the input data [95]. There are many univariate criteria. Some criteria are based on the origin of the measure, such as information theory, dependence, and distance. Some criteria are based on the measuring structure, such as impurity-based criteria, normalised impurity-based criteria, and binary criteria. The multivariate's goal is to find a combination of attributes that have a good dis-

criminatory power. Many of the multivariate splitting criteria are based on the linear combination of the input attributes. The best linear combination can be performed using a greedy search, linear programming, linear discrimination analysis, etc.

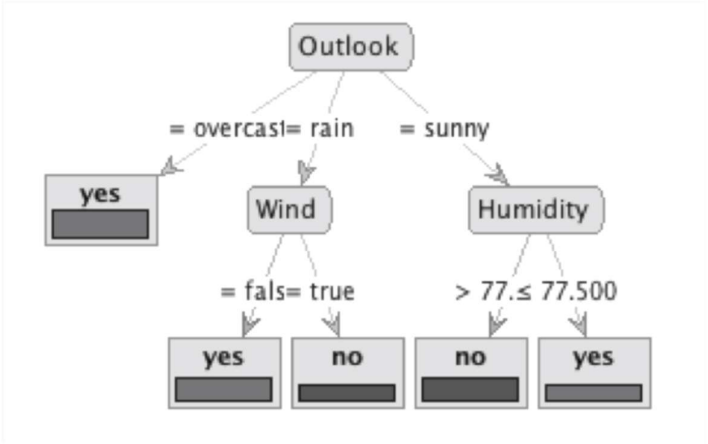


Figure 3:11 Decision tree for the weather problem

Decision tree learning algorithms

There are many top-down decision trees algorithms such as CART [26], ID3 [51], and C4.5 [27]. There are two important concepts, namely the growing phase and pruning phase. Some of the algorithms apply the two concepts, such as C4.5 and CART. However, the other inducers implement the growing phase. All these algorithms are top-down induced using the divide and conquer concept. The selection of the most suitable function is made by some splitting measure as explained before. The splitting continues until either no more splitting is possible or the stopping criteria is satisfied.

A decision tree is a classifier based on the attribute values pairs [1]. Each node in the decision tree represents an attribute in the dataset. The branches represent the values that the attribute (node) can take. The prediction process starts at the root node and then sorts them based on their attribute values. The decision tree in Figure 3:11 gives an example of how decision trees classify some weather data to decide if it is a good day for playing golf or not.

The top-down greedy search algorithms for building a decision tree recursively divide the training data into subsets based on the attribute that will best classify the training data, which initially forms the root node of the decision tree. The algorithm is then repeated recursively on each partition of the divided data to create a sub-tree. At each level in the partition process, a statistical measure is evaluated to find the best attribute.

The standard procedure for building the decision tree is to check all attributes in the training set for the attribute that helps the most in reducing uncertainty in taking a decision. Entropy is a unique function that satisfies the four axioms of uncertainty [96]. It represents the average amount of information when coding each class into a code word.

In our weather example, there are four predictor variables (outlook, temperature, wind, and humidity), which are used to predict the class variable. For each node in the decision tree, we have to decide which attribute is being used to split it. We have to determine whether we will split the node or turn it into a leaf node. Figure 3:11 shows the decision tree for the weather problem. There are many ways for determining an attribute used for splitting nodes, such as information gain, Gini index, gain ratio, purity, likelihood-ratio, chi-squared statistics, and other univariate splitting criteria. The most common methods are based on information gain and Gini index.

Information Gain

Information gain was proposed by Quinlan and is based on the entropy concept [51]. If the uncertainty in the system increases, it will be more difficult to predict an outcome generated by the system. In other words, if we have three balls of different colours, the chance of guessing the colour of a randomly drawn ball is 1/3. However, if we have ten balls of different colours, the chance of guessing the colour of a randomly drawn ball is 1/10. When the uncertainty increases, the amount of information needed to reach a decision (guess better) will increase. The average amount of information can be calculated through a mathematical measure called entropy. It means that if we have a high value of the entropy, it will indicate more information and in turn more uncertainty. The mathematical equation used to calculate entropy is:

$$H(x) = \sum_{i=1}^{k} Pr(x_i)\ log(Pr(x_i))$$

Where Pr is a probability distribution. If we have 20 instances in our weather problem, the entropy at the root node will be:

$$Entropy\ at\ root = -\frac{5}{20} \times log_2\left(\frac{5}{20}\right) - \frac{15}{20} \times log_2\left(\frac{5}{20}\right) = 0.811\ bit$$

In the next step, we will determine the attribute that we can use for the split in a node. This will be the attribute that corresponds to the largest reduction in entropy, which defines the so-called Information Gain (IG). IG is calculated mathematically as follows:

$$IG\ (attribute_x) = entropy\ (Current\ node) - \sum_i^n \Pr(Child\ node)_i \times\ entropy(Child\ node)_i$$

Assume we have five instances in sunny, four occurring in overcast, and five instances in rainy classes. IG for outlook is 0.247 bits, for temperature is 0.029 bits, for windy is 0.048 bits, and for humidity is 0.152 bits. Finally, we will select for the split in the current node (the decision tree root in our example) the attribute that leads to a maximum IG, that is, the outlook attribute.

Gini index

Gini index is another conventional approach for selecting the attribute to split; it was proposed by Breiman et al. [26]. The mathematical equation giving the Gini index is:

$$Gini(x) = 1 - (Pr_1^2 + Pr_2^2 + Pr_3^2 + \cdots + Pr_k^2)$$

Where *Pr*s are the relative frequencies of classes. We should calculate the Gini index for each possible attribute in relation to a specific node. The mathematical equation is:

$$Gini(attribute_x) = \sum_i^n \Pr(Child\ node)_i \times\ Gini(Child\ node)_i$$

The Gini index value for outlook is 0.446, for humidity is 0.367, for windy is 0.428 and for temperature is 0.4403. The best Gini value is the smallest one, which corresponds to humidity, so this is used to split the root node. It is clear that choosing the root node using information gain is different from when using the Gini index. However, in both (and other splitting criteria) cases, the tree grows recursively by finding the roots of the subtrees, as proceeded above.

3.11.1 C4.5 algorithm

In 1993, the C4.5 algorithm was proposed by Quinlan [27]. To grow the decision tree recursively, the algorithm uses the information gain or the gain ration splitting criteria.

The resulting decision tree can also be converted into a set of rules. Each rule represents a path from the root node to a leaf. The algorithm tries to generalise the rule by removing any of its conditions that help to improve the estimated accuracy of the rule. The new rules are sorted by their accuracy and will be used in the stored sequence when classifying a new example. The accuracy of each rule is calculated on the training dataset. The accuracy is estimated for the rule from the training examples that it covers.

Figure 3:12 shows a simple illustration of how the decision tree models perform on several simple datasets.

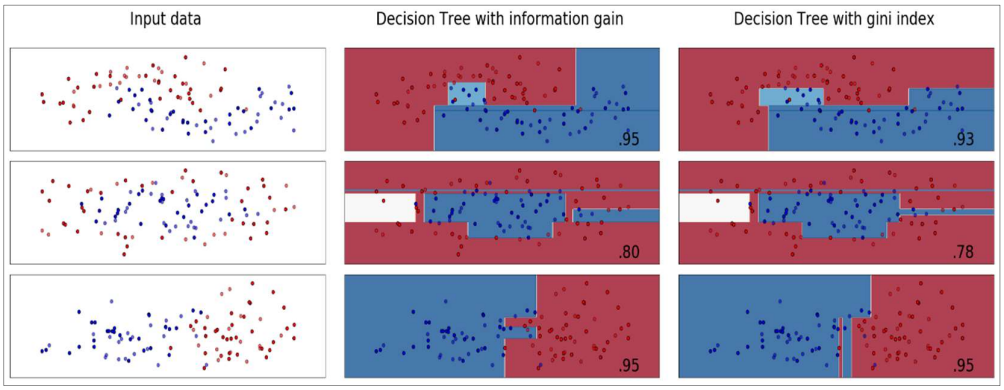


Figure 3:12 Decision tree classifiers on generated datasets

The C4.5 algorithm (and its commercial extension C5.0) is equipped with mechanisms to prevent or reduce the over-fitting problem [23]. Two common approaches have been used for this. Firstly, the algorithm tries to stop training before reaching a point at which it perfectly fits the training data. Secondly, the algorithm may prune the induction decision tree, which is the most commonly used remedy [77]. Pruning is used to discard parts of a prediction model that may cause a random variation in the training sample. Applying the pruning methods makes the model more comprehensible to the user and leads to a better accuracy on new data. The mechanism that is used during the pruning process should be efficient enough to distinguish between parts of the classifier. Statistical tests can be used to determine whether an observed effect is genuine or there may be fluctuations in the sampling process. This, in turn, will help to decide to prune.

3.11.2 CART, CHAID, and QUEST algorithms

Some decision tree approaches have been proposed to predict categorical and continuous variables, such as classification and regression tree (CART), chi-squared automatic interaction detector (CHAID), and quick, unbiased, efficient statistical trees (QUEST) algorithms. A classification and regression tree (CART) is a binary partitioning decision tree that evaluates relationships between data to produce a model for the future data [97].

Unlike CART, CHAID was proposed to build non-binary decision trees, and it is used only for categorical variables. The CHAID approach applies merging and testing of independent variables to deal with the missing data, which may lead to an increase of the computation time. Therefore, if we analyse large datasets, CHAID will need more computational time. However, QUEST can create the binary decision tree faster than any other technique. In addition, QUEST requires ample memory space when it deals with large datasets.

The CART algorithm can handle missing values. It has been used extensively in the medical diagnosis field [98]. Such an example is an application of learning a decision tree for the severity of heart attacks and the appropriate treatment for patients [99].

The CART algorithm uses the splitting criterion known as the Gini index. CHAID uses the chi-squared test as a splitting criterion in selecting an attribute in a node, and Bonferroni corrections to account for multiple testing [100]. QUEST algorithm, in turn, makes use of the ANOVA F test for each numeric predictor, and the chi-squared test for each categorical predictor, when selecting the best attribute for a split.

3.11.3 Random forests

Random forests implement an updated version of bagging trees. As in bagging, the algorithm constructs a number of decision trees on the bootstrapped versions of the training dataset. However, a random sample of k predictors (k being fixed) is chosen to compete in a node of the tree. k is a parameter which can be tuned, but by default, it is equal to the square root of the total number of predictors P when we have a classification task, or its equal to the third of the total number of predictors of P when it is a regression task.

There are multiple random forests approaches that differ from each other in the way they introduce random perturbation into the induction procedure. In 1990, Kwok and

Carter proposed the first randomised induction algorithms [100]. They claim that the average multiple decision trees with different structures give better results than any of the decision trees in the forest.

In 1994, Breiman aggregated multiple versions of an estimator into an ensemble to lead to a better accuracy [76]. The output from his process produces a lower error. In 1995, Dietterich and Kong proposed a new approach to randomise the choice of the best split at a given node [101]. The output showed better results than bagging in a low noise setting. However, when the noise is important, bagging achieved better results. In 1997, Amit et al. proposed a randomised variant of the tree algorithm, which consisted of searching for the best split at each node over a random sample of the predictors [102]. In 1998, Ho improved on Amit's approach. He proposed the random subspace (RS) approach to building a decision forest whose trees are built using random subsets of the predictors. This method achieved the best performance over the conventional random forests methods [95].

```
Select the number of models to build, m
for i = 1 to m do
        Generate a bootstrap sample of the original data
        Train a tree model on this sample
        for each split do
                Randomly select k (< P) of the original predictors
                Select the best predictor among the k predictors and
                partition the data
                end
        Use typical tree model stopping criteria to determine when a tree is
        complete (but do not prune)
        end
```

Figure 3:13 Basic random forests.

Random forests algorithms present a few parameters whose influence has been extensively studied, as in [103], and are now well understood. The three most important ones are: (1) the number of trees, (2) the number k of predictors competing in a node, and (3) the tree depth. Usually, a large number of trees ensures the convergence of the ensemble-based model – and in practice at least 500 models should be developed; this number can be increased until the performance converges. In most cases of tasks at hand, random forest models are optimised according to parameter k. Random forests models show more

considerable stability and a lower variance in prediction than single tree models [104]. A simple random forests algorithm version is described in Figure 3:13. In addition, a comparison between random forest models and AdaBoost models on simply generated datasets is shown in Figure 2:14. This illustrates a tendency of the superiority of the AdaBoost models, which is often encountered in practice. However, random forests models tend to produce good results on highly dimensional datasets, due to the particularity of selecting random samples of predictors to compete in each node, which contributes to a good stability of these ensemble models.

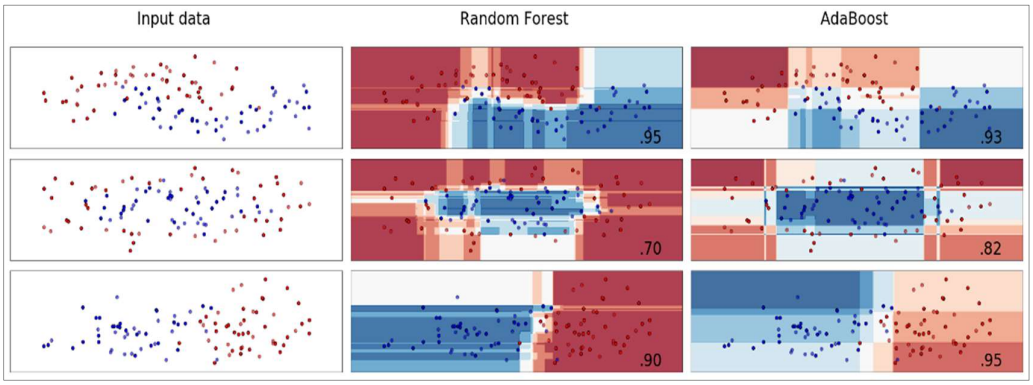


Figure 3:14 Random forest and AdaBoost on generated datasets

3.12 Support vector machines

Support vector machine (SVM) is a state of the art technique in the machine learning field, which has also been used in many medical applications to improve methods for detecting diseases in clinical settings [105] [106]. Moreover, SVM has demonstrated high performance in solving classification problems in bioinformatics [107] [108].

In this algorithm, each sample in the dataset is plotted as a point in a p-dimensional space, and the best hyperplane that separates the classes accurately is chosen [38]. The hyperplane is defined mathematically by the following equation:

$$\beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots \beta_m x_m = 0$$

A prediction is made by using the hyperplane equation with the values of the new point. If the left-hand expression in the equation returns a value greater than 0, the point will be classified to the first class (Y_r = +1). Similarly, if the left hand expression returns a value less than 0, the point will be classified to the second class (Y_r= -1).

To identify the best hyperplane, the SVM algorithm identifies the so-called support vectors, which are points in the dataset that influence the separation hyperplane. Indeed, by maximising the distances between the nearest data points (the support vectors) and the hyperplane, it is possible to choose the right hyperplane. This distance between the closest data point and the hyperplane is called Margin M [81]. Mathematically the margin M and the model are determined as follows:

$$\underset{\beta_0,\beta_1,\dots,\beta_m}{\text{maximising}}\ M$$

$$subject\ to \sum_{c=1}^{m} \beta_c^2 = 1,$$

$$y_r(\beta_0 + \beta_1 x_1 + \cdots + \beta_c x_{rc}) \geq M, \qquad \forall\, r = 1, \dots, n$$

The optimal hyperplane that can separate the classes is the hyperplane that has the largest margin, which is called the maximal-margin hyperplane.

However, most of the time data points cannot be separated with a hyperplane in real situations. To overcome this problem, the SVM algorithm allows some points in the training data to violate the separating line, which leads to the concept of soft margin classifier, mathematically formulated as follows:

$$\underset{\beta_0,\beta_1,\dots,\beta_m\ and\ \varepsilon_1,\dots,\varepsilon_r}{\text{maximising}}\ M$$

$$subject\ to \sum_{c=1}^{m} \beta_c^2 = 1,$$

$$y_r(\beta_0 + \beta_1 x_1 + \cdots + \beta_c x_{rc}) \geq M\,(1 - \varepsilon_r),$$

$$\varepsilon \geq 0, \sum_{r=1}^{n} \varepsilon_r \leq C,$$

The tuning parameter C represents the allowed amount of violation of the margin. When C gets smaller, the model fits better the training data, which leads to higher variance and lower bias. When C gets larger, the model is more flexible which leads to lower variance and higher bias.

SVM technique also comprises capabilities to perform classification using a non-linear boundary and can do that efficiently by employing the kernel trick. A kernel intuitively describes the similarity between data points and the support vectors. In linear SVM, the dot product is the employed similarity measure. Other kernels can be used to transform the input space into higher dimensions, such as the polynomial kernel and the radial kernel. The use of more complex kernels allows different hyperplanes to separate the classes that are non-linearly separable in the original space.

3.12.1 Linear SVM

In linear SVM, the dot product is the kernel and can be written as:

$$K(x, x_r) = \sum_{c=1}^{m} (x * x_r)$$

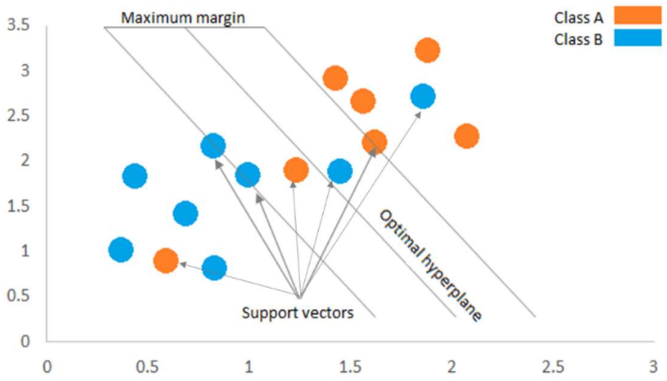


Figure 3:15 A linear Support Vector Machine.

3.12.2 Polynomial SVM

In polynomial SVM, instead of the dot product, the polynomial kernel is employed as follows:

$$K(x, x_r) = (1 + \sum_{c=1}^{m} (x * x_r))^d$$

Here the degree d of the polynomial must be specified. When d is equal to 1, the kernel is linear. The polynomial kernel allows for curved lines to fit the input space.

3.12.3 Radial SVM

Another popular kernel is the radial kernel defined as:

$$K(x, x_r) = \exp(-\gamma \sum_{c=1}^{m} (x - x_r)^2)$$

Here gamma is a parameter whose values can determine the model built with the radial kernel to create complex separation regions within the feature space as shown in Figure 3:17. Also, Figure 3:16 shows an example of a SVM with a radial basis kernel. In addition, Figure 3:17 illustrates a comparison between SVM with linear, radial and polynomial kernels on three generated datasets.

SVM is easily extendable for multi-classes classification by decomposition of this problem in 2-class classifications sub-problems. SVM is also extended to perform regression, a technique with similar mathematical formulations. Finally, we mention that in practice, SVM can be tuned by using grids on the parameters. For instance, for a radial kernel, one can build a grid of values for γ and C and fit several models and selecting the one that performs the best in a cross validation.

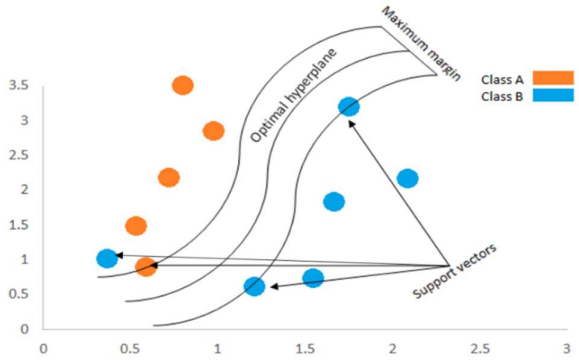


Figure 3:16 Support Vector Machine with a radial kernel.

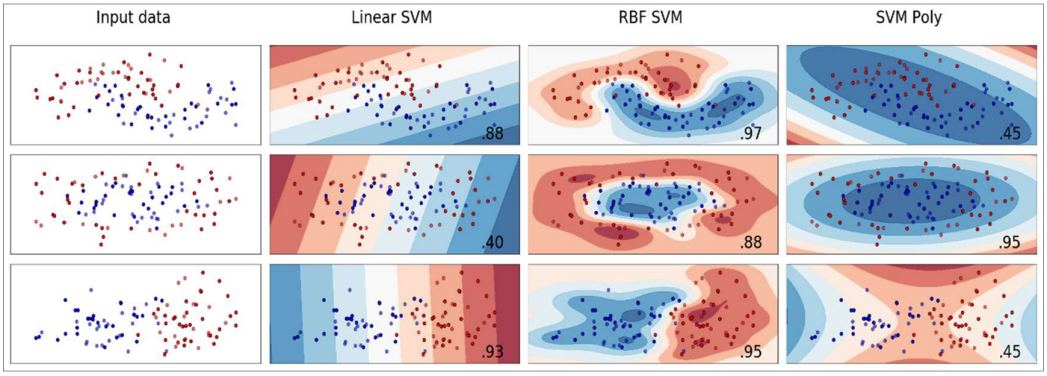


Figure 3:17 Support Vector Machines on several generated datasets.

3.13 Artificial neural networks

Artificial neural networks (ANN) are broadly applicable machine learning models that are motivated by the biological neural networks [109]. ANN investigate the functional relationship between the input variables and output variable.

Frank Rosenblatt at the Cornell Aeronautical Laboratory invented the perceptron, which is the simplest neural network. A perceptron follows the 'feed-forward' model, meaning that inputs are sent into the neuron, processed, and a result is output [110].

The most common issue with neural networks is the time consumed for model training. Neural networks can learn their weights and biases using the gradient descent algorithm or backpropagation algorithm [2].

Figure 3:18 illustrates a neural network consisting of an input layer with two input nodes, a hidden layer with two nodes and an output layer with a single node. Each layer is connected to other layers from both sides. Some of them are input units that receive information from the data that they will attempt to learn from. Output units are on the opposite side of the network. Between the input units and output units are one or more layers of hidden units. The units are connected by edges (synapses) labelled with a number called a weight. The larger the weight (in absolute value), the more influence one unit has on another. Figure 3:19 illustrates how neural networks perform on some generated datasets.

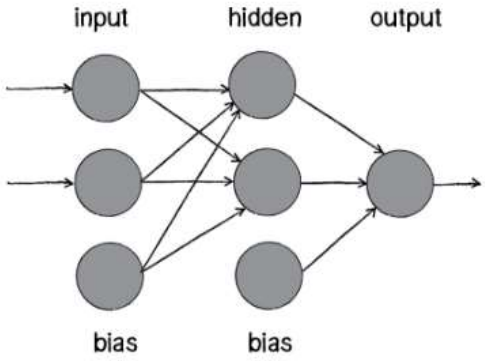


Figure 3:18 Typical architecture of Artificial Neural Networks.

There are many categories of ANN used in research and practical applications. The most well-known ANN are one-layer neural networks, model-averaged neural networks [111], multi-layer perceptron [110] [112], radial basis function network [113], penalised multinomial regression [114], etc.

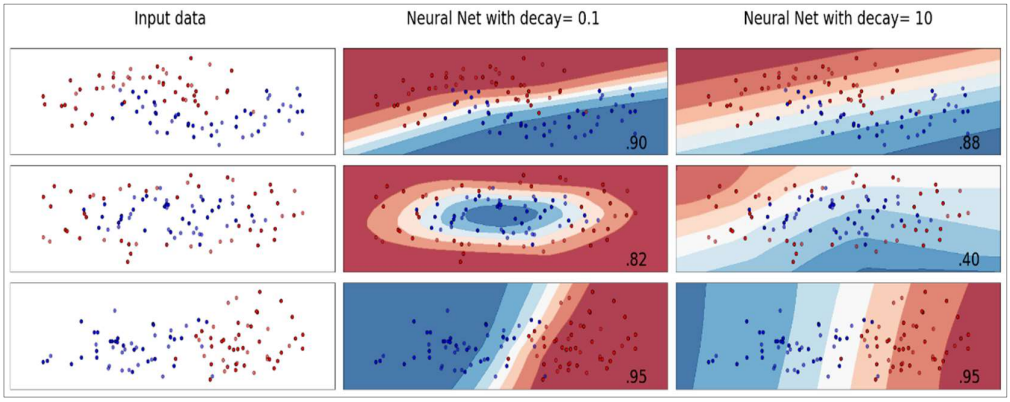


Figure 3:19 Neural networks on generated datasets.

3.13.1 Deep learning

Deep learning is a new family of machine learning methods based on learning representations of data. Deep learning was developed based on sophisticated algorithms that model high-level features and extract those constructs from data by using a complex neural networks architecture [115].

In recent years, Deep learning has generated much excitement in machine learning research community and industry. Various deep learning architectures such as deep neural networks have been applied to fields like speech and audio recognition, natural language processing, and bioinformatics, where they have been shown to provide state-of-the-art results on most of the tasks [116] [117].

Deep learning usually involves many parameters, but it was designed to reduce the number of parameters the user has to specify by applying features selection and early stopping techniques. Variable importance of neural networks models is notoriously difficult to compute, and there are many pitfalls. Deep learning selects the attributes that best suit the model. The early stopping is usually set to let it end training automatically once a performance such as the area under the curve does not improve (specifically, if the moving average of length two does not improve by at least 1% for five consecutive scoring events). However, there are a few parameters that still need to be tuned, such as the number and sizes of hidden layers, the number of epochs, the activation function, and the generalisation techniques penalties L1 and L2.

While sigmoids have been used historically as activation functions for neural networks, deep learning implements further activation methods, such as tanh, rectifier,

and maxout. In addition, the L1 and L2 parameters could be tuned to prevent overfilling of the model.

Recently, deep neural networks have attracted widespread attention, mainly by defeating alternative machine learning methods such as SVM [38] in numerous critical applications, such as classifying Alzheimer's disease [10] and classifying AD/MCI patients [118]. While SVM is still a fairly popular technique within the machine learning community, deep learning is gaining considerable attention in this community [119] [120] [121]. Deep learning methods are a type of representation learning methods, which means that they can automatically identify the optimal representation of the raw data without requiring prior feature selection.

3.14 Conclusion

This chapter provides pipeline for the data processing used through the thesis. It also gives a literature review of the existing algorithms for predictive modelling in the fields of statistical and machine learning. These algorithms include a wide range of algorithms such as linear regression which is the most traditional and popular methodology in both statistical and machine learning, Naive Bayes, Gaussian process, k-nearest neighbour, decision trees, random forests, support vector machine, artificial neural networks, and deep learning.

Statistical and machine learning algorithms offer great promise in helping organisations uncover patterns hidden in medical data that can be used to improve understanding, prediction and treatment of different illness. However, these statistical and machine learning algorithms need to be guided by users who understand the data, and the general nature of the analytical methods involved. Building models is only one step in knowledge discovery. It is vital to collect and prepare the data properly as well as to check the models with experts. The best model is often found after building models of several different types, or by trying different technologies or algorithms.

Chapter 4 Novel prediction modelling and pattern detection approaches for the first-episode psychosis associated with cannabis use

Over the last two decades, a significant body of research has established a link between cannabis use and psychotic outcomes. In this chapter, we aim to propose a novel synergistic machine learning and statistical approach to pattern detection and to develop predictive models for the onset of first-episode psychosis. The data used has been gathered from real cases in cooperation with a medical research institution, and comprises a wide set of variables including demographics, and drug-related, as well as several variables specifically related to cannabis use. Our approach is built upon several machine learning techniques whose predictive models have been optimised in a computationally intensive framework. The ability of these models to predict first-episode psychosis has been extensively tested through large-scale Monte Carlo simulations. Our results show that boosted classification trees outperform other models in this context and have significant predictive ability despite a large number of missing values in the data. Additionally, association analysis and Bayesian techniques were applied to investigate how different patterns of cannabis use relate to new cases of psychosis.

4.1 Problem description

A number of US states have already legalised or are in the process of legalising the use of cannabis. Some other countries such as Uruguay have previously done so. Moreover, cannabis is currently one of the most used illicit drugs in the world [122]. However, research established a significant link between cannabis use and psychotic symptoms, and that cannabis use is the most preventable risk factor for psychotic disorders [123] [124]. In this context, any harm caused by cannabis use, in particular in connection to psychosis, should be quantified.

As such, more recently researchers sought to understand whether specific patterns of cannabis use (such as potency [125] [126] or age [127]) relate to a higher risk of psychotic disorders. One study estimated that nearly a quarter of all new psychosis patients in South London (UK) could be attributed to the use of high-potency, skunk-like cannabis [15]. The same study estimated that the risk of experiencing psychotic disorders is roughly three times higher for those who are daily users of cannabis and over five times for those using high potency cannabis daily, compared to those who are not users.

However, there is still scope for further understanding of the links between patterns of cannabis use and psychosis [128]. Most existing studies are only explanatory research strategies and not risk prediction modelling using machine learning algorithms. Moreover, many studies are limited by incomplete or inconsistent records, or a lack of detailed variables, but also by the methodologies used, which are based mainly on a number of conventional statistical techniques, such as hypotheses formulation and verification via statistical tests, logistic regression modelling, etc. These methods are traditionally well recognised and used in medical research, but in many situations, they do not match the enormous potential of the modern machine learning methods.

In this chapter, we propose a novel synergistic machine learning and statistical approach to pattern detection and to developing predictive models for the onset of first-episode psychosis. The dataset on which we based our study was collected from previously conducted medical studies as described in [15]. It comprises a broad set of variables including demographics, drug-related, as well as several other variables with specific information on the participants' history of cannabis use (see Appendix B).

Prior to the prediction modelling, a significant effort in our work was involved in the data pre-processing due to inherent challenges present in data collected in a case-

control study involving many missing values, multiple encodings of related information, a significantly large number of variables, etc. The prediction modelling phase consisted of investigating several machine learning techniques, such as k-nearest neighbours (k-NN), support vector machine (SVM) with different kernels, decision trees, bagged trees, boosted classification trees, extreme gradient boosting and random forests (RF), whose predictive models have been optimised in a computationally intensive framework. The ability of these models to predict first-episode psychosis, which is a novelty and one of the contributions of this study, has been extensively verified through large-scale Monte Carlo simulations in the same computationally intensive framework. Our results show that boosted classification trees outperform other models in this context and have a good predictive ability despite a large number of missing values in the data.

Then, the predictive value of cannabis-related variables with respect to first-episode psychosis is demonstrated in this work by showing that there is a statistically significant difference between the performance of the predictive models built with and without cannabis variables. We were inspired by this approach, proposed and implemented here, by the Granger causality techniques [16]. These are used to demonstrate that some variables have predictive information on other variables in a regression context, as opposed to classification, which is mainly the case in our framework.

Moreover, we investigate how different patterns of cannabis use relate to new cases of psychosis, via association analysis and Bayesian techniques such as Apriori and Bayesian Networks, respectively. Finally, we extended our approach by applying different cuts of the data sets to the selected prediction model (boosted classification trees) to examine how the prediction performances' variation evolves.

The remainder of the chapter is organised as follows: Section 2 presents our approach to predicting first episode psychosis, based on experimenting with various machine learning algorithms and on computational intensive model optimisations. The section also includes the data pre-processing and investigates the outcomes of the extensive Monte Carlo simulations in order to study the variation of the model performances that may have also been affected by the presence of a high proportion of missing values in the data. In Section 3, we build optimised prediction models without the cannabis attributes to study if there is a statistically significant difference between the predictive power obtained with and without the cannabis attributes. Further, we

investigate and discuss further relationships between the cannabis variables and first episode psychosis. Finally, the directions for future work and the conclusion are presented in Section 4.

4.2 Predicting first-episode psychosis: a computationally intensive approach

The data used to develop our novel approach to predicting first episode psychosis is part of a case-control study at the inpatient units of the South London and Maudsley (SLaM) NHS Foundation Trust [15]. The clinical data consists of 1106 records, including 489 patients, 370 controls and 247 unlabelled records. Those described as patients were patients of the Trust who at one time presented with first-episode psychosis; controls were recruited from the local area through the internet, newspaper advertising, and by distributing leaflets. Each record refers to a participant of the study and has 255 possible attributes, which were divided into four categories. The first category consists of demographic attributes that represent general features such as gender, race, and level of education. Secondly, drug-related attributes contain information on the use of non-cannabis drugs such as tobacco, stimulants and alcohol. The third category is formed of genetic attributes which were removed from the analysis for the purpose of this study. The final category contains cannabis-related attributes, such as the duration of use, initial date of use, frequency, and cannabis type, etc. A complete data dictionary is represented in Appendix 2.

In order to build our approach to predicting first episode psychosis, the data required a set of pre-processing transformations, including feature selection, data sampling, data type conversions as needed by training certain types of models, and missing value imputation.

Prediction modelling consisted of considering various machine learning techniques which are suitable for this classification problem and the dataset, including K-nearest neighbours, support vector machine with different kernels, decision trees, bagged trees, boosted classification trees, extreme gradient boosting and random forests. The models were evaluated based on accuracy, the area under the ROC curve, precision, sensitivity, specificity, and Cohen's kappa statistic. All experiments, including model training and optimisation based on repeated cross-validations, and extended Monte Carlo

simulations based on split validations to investigate the stability of the performances of the models*, were conducted in a computationally intensive framework, using the packages R 3.2.3 (December, 2015) [129], RapidMiner 6.5 (Jan, 2016) [130], WEKA 3.6.15 (December, 2015) [71] and Apache Spark 2.0 (June, 2016) [131], by performing a parallel processing on a data science cluster of 11 servers based on Xeon processors and 832GB of RAM.

4.2.1 Data pre-processing

Data pre-processing was performed before modelling in order to rationalise the complexity of the data and prepare the data for use. The pre-processing consisted of the stages of rationalisation and refinement.

4.2.1.1 *Rationalisation*

The work of this stage sought to perform a high-level simplification of the dataset, and included several steps:

First, records that were missing critical data were removed from the dataset. This included records with missing labels (i.e. no specification of patient versus control group), as well as records for which all cannabis-related variables were missing.

Secondly, specific variables were removed from the data. This primarily involved variables that were deemed to be irrelevant to the study (such as those related to individual IDs of the study participants), and also included variables which were outside the scope of the current study (for example, certain gene-related variables). In addition, any numeric predictors that had zero or near-zero variance were dropped.

Lastly, we sought to make the encoding of missing values consistent across the dataset. Prior to this step, values including 66, 99, and -99 all represented cases with missing values. All such indicators were replaced with a consistent missing value indicator, NA.

*The study of the variability of the performances of the predictive models was required by the extensive imputations due to the presence of many missing values in the dataset, since we decided to include a superset of the attributes in the analysis.

4.2.1.2 *Refinement*

This stage requires several steps. First, the variables were re-labelled to provide more intuitive descriptions of the data contained within. Then, the variable types were made consistent across the dataset. In some instances, this involved converting characters into factor variables. In others, it involved taking an integer value, converting to a factor variable, and then labelling each category with its meaning. For example, cannabis frequency originally contained values of NA, 0, 1, and 2. This was converted to a factor variable where 0 was *Never used*, 1 was labelled *Less Than Daily*, and 2 was *Daily*.

In multiple situations, some variables had a similar meaning to other variables, yet there were often missing values for some records in some of these variables. Accordingly, a process of imputation was used to combine the information from related variables into one effectively. For example, two fields described alcohol use, but were inconsistently present across the records. These were combined in a way that created one variable that reflected whether the subject was a user of alcohol. This process was used to generate value-reacher and value-consistent fields related to alcohol use, tobacco use, employment history, and subjects' age.

After that, any attribute that contained more than 50% of missing values was dropped from the study. We then removed any record for which more than 70% of the remaining attributes contained missing values. The resulting dataset, after the transformations above, contained 777 records and 29 attributes. The 777 records are divided into 451 patients and 332 controls. A summary of some of these fields – specifically the ones that relate to cannabis use such as type, frequency, age first use, and duration – are seen in Table 4:1. The complete data dictionary is included in Appendix 2.

Finally, numerical attributes were discretised into several intervals. Discretising numerical attributes is a necessary step with this dataset to avoid creating inaccurate data during the missing values imputation process. This could occur because of participants, who never consumed cannabis, have missing values in most cannabis-related attributes. Such missing values could be replaced with inaccurate values during the missing values imputation process instead of the real value 'never used', which does not exist in the data. To avoid such circumstances a new interval called 'never used' was added directly to the categorical attributes. However, in the case of numeric attributes such as duration, these

attributes were discretised into different intervals, then a new interval called 'never used' was added.

Attribute	Description
Lifetime cannabis user	Ever used cannabis: yes or no
Age first used cannabis	Age upon the first use of cannabis: 7 to 50
Age first used cannabis under 15	Age less than 15 when first used cannabis: yes, no or never used
Age first cannabis used under 14	Age less than 14 when first used cannabis: yes, no or never used
Current cannabis user	Current cannabis user: yes or no
Cannabis frequency	Pattern of cannabis use: never used, only at weekends or daily
Cannabis measure	Cannabis usage measure: none, hash less than once per week, hash on weekends, hash daily, skunk less than once per week, skunk on weekends, skunk daily
Cannabis type	Cannabis type: never used, hash or skunk
Duration	Cannabis use duration: 0 to 41 (months)

Table 4:1 Cannabis use attributes in the analysed dataset

Different discretising methods, such as by user specification, entropy, size and frequency were applied to the dataset. After comparing the discretising methods, we concluded that in order to build models that gain higher performance in predicting first episode psychosis, some numerical attributes were discretised by frequency and others were discretised by user specification. User specifications were suggested by previous studies [15] [126] [132].

4.2.1.3 *Missing values and imputation*

Although the dataset was pre-processed and attributes with more than 50% of missing values have been removed from the dataset, it still contains a large number of missing values. Of the 777 records, only 22.5% are complete records. This volume of missing information makes modelling more challenging, but often this is the reality in medical and social research. In this chapter, we used a superset of the variables that were explored in [15] in order to examine the efficacy of machine learning to deal with a significant number of missing values present in the whole collected dataset. A plot of the proportion of missing values is shown in Figure 4:1. Only two variables – the output attribute *patient* and the input attribute *lifetime_cannabis_user* – are populated completely. On the other extreme, the attribute *children* is missing values in 48% of the records. Missing values

can exist in medical data sets for many reasons. For example, the participants may be unable to fully complete a survey, or may want to abstain from answering certain questions, or do not attend follow-up appointments, etc. Alternatively, researchers may decide to add or remove certain attributes from the data collection process over time. Missing values mostly need to be imputed before applying pattern discovery techniques.

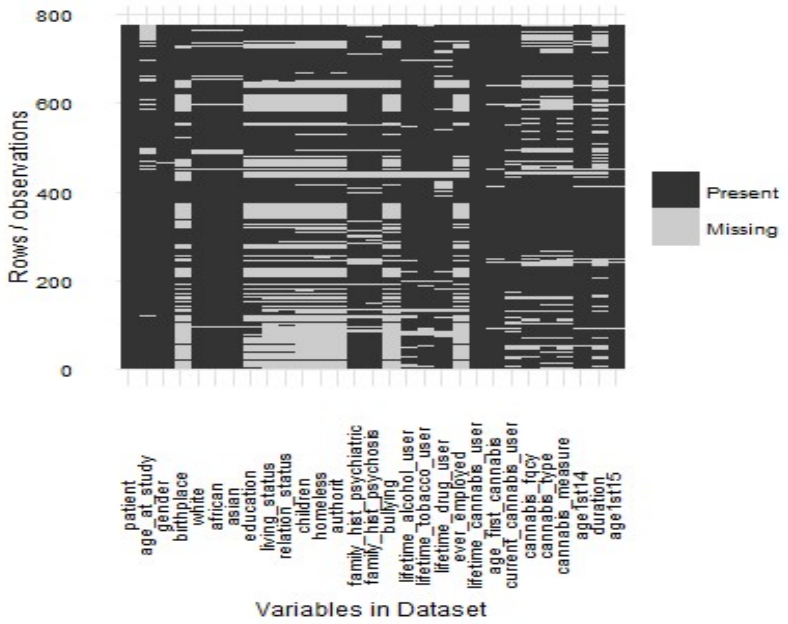


Figure 4:1 Summary of the ratio of missing values for each attribute

However, the predictive power of the data may depend significantly on the way missing values are treated. While some machine learning algorithms, such as decision trees [133], have the capability to handle missing data outright, most machine learning algorithms do not. Usually, in medical research applications, missing values are imputed using a supervised learning technique, such as k-nearest neighbour, after suitable scaling to balance the contribution of the numeric attributes. These imputation techniques do not have theoretical formulations but have been much implemented in practice [134] [135]. In this work, we opt for the tree bagging imputation from the caret 6.0 package (January 2016) [136] to impute the missing values in the training data sets which are generated in the cross-validations, or in the repeated experiments of the Monte Carlo simulations. This imputation process is thus repetitive, becomes a part of each model training and is therefore evaluated as part of each model's performance metrics. We should also note that imputations performed with k-nearest neighbour from the caret 6.0 package (January

2016) [136] led finally to slightly weaker predictive results than those based on tree bagging imputations, although the latter are more computationally costly as being based on an ensemble technique. This of course matters in a computationally intensive framework comprising an intensive model optimisation and extensive Monte Carlo simulations. As such the use of adequate computing power is the solution, and we benefited of it to handle this aspect.

As a final transformation of the data, since some prediction modelling algorithms, such as support vector machines, work only with numeric data, we transformed the input nominal variables into dummy variables, obtaining a dataset of 91 variables.

4.2.2 Training and optimising predictive models

For the purpose of developing optimised predictive models for first episode psychosis, we have considered a variety of suitable classification algorithms from the caret 6.0 package (January 2016) [136], from simple to state-of-the-art, including C5.0 decision tree [2] [133], boosted classification trees [137], bagged CART [76], random forests [104], support vector machine [2], k-nearest neighbours [138] and eXtreme gradient boosting [139].

In the view of model optimisation, the values of the parameters for each of the algorithms have been controlled by suitably chosen grids. Predictive models have been fitted, in a 10-cross-validation procedure, on each training set after tree bagging imputations of missing values on the same training set, and have been tested on each test set. The best performance models with their parameters have been selected for subsequent comparison. For instance, the random forests algorithm has been tuned for a fixed number of 500 trees and up to 44 attributes, and a value of 30 attributes was ultimately selected for optimal performance. Similarly, the support vector machine algorithm has been tuned for different kernels and values for the cost and gamma (also called sigma in an R package such as the caret 6.0 package (January 2016) [136]) parameters. The best model was obtained on the radial (RBF) kernel, with the cost 16384, and sigma $3.052e^{-05}$.

Model	Optimised Parameter(s)
C5 Decision Tree	Iter = 70, Model = rules, Winnow = False
Boosted Trees	Iter = 100, MaxDepth= 5, Nu = 0.1
eXtreme Gradient Boosting	Nrounds = 50, MaxDepth = 3, Eta= 0.3, Gamma= 0, Colsample= 0.8, MinChild =1
Bagged CART	None
Random Forests	Mtry = 30
SVM (Linear)	Cost = 16
SVM(Radial)	Cost = 16384, Gamma = $3.05e^{-05}$
SVM (Poly)	Cost = 64, Degree = 1, Scale= 0.1
k-NN	K = 5

Table 4:2 Summary of parameters tuned for each model.

A summary of the models with the chosen optimised parameters is shown in Table 4:2. The key performance measure was the accuracy (the rate of accurate classification), and we also monitored Kohen's kappa statistic (the agreement between actual values and predictions, adjusted for what could be expected from pure chance) [140]. Summary tables of initial estimations of accuracy and kappa for all models can be seen in Table 4:3 and Table 4:4, respectively. These results are the average outcome of the optimisation procedure explained above based on a ten-cross validation. Based on these results, the models that were selected for further analysis were boosted classification trees (AdaBoost), random forests, and support vector machine with the radial kernel.

Model	Min.	Median	Mean	Max.
SVM (Radial)	0.7179	0.7806	0.7824	0.859
AdaBoost	0.7403	0.7807	0.785	0.8462
Random Forests	0.7143		0.7798	0.8462
xgbTree	0.7143	0.7756	0.7786	0.8333
C5 Decision Tree	0.7273	0.7677	0.7682	0.8077
Bagged CART	0.6753	0.7565	0.754	0.8333
SVM (Linear)	0.6923	0.7628	0.7619	0.8205
SVM (Poly)	0.6923	0.7628	0.7657	0.8333
k-NN	0.6623	0.6815	0.6923	0.7692

Table 4:3 Initial estimation of model accuracy.

Model	Min.	Median	Mean	Max.
SVM(Radial)	0.4659	0.5479	0.5519	0.7052
AdaBoost	0.5266	0.5501	0.5586	0.677
Random Forests	0.4898	0.5479	0.5465	0.6823
xgbTree	0.5087	0.5384	0.5432	0.66
C5 Decision Tree	0.5014	0.5232	0.5211	0.5979
Bagged CART	0.4398	0.5032	0.495	0.6653
SVM (Linear)	0.4594	0.509	0.5078	0.6293
SVM (Poly)	0.4594	0.509	0.516	0.6544
k-NN	0.3191	0.3393	0.3564	0.5074

Table 4:4 Initial estimation of model kappa.

4.2.3 Monte Carlo simulations

Due to expected potential variations of the predictive models' performance, depending on the datasets for training and testing, but in particular, due to the uncertainties introduced by the missing values in the data, we conducted extensive Monte Carlo simulations to study these variations, and thus the stability of the models.

On each training set, a tree bagging imputation was performed prior to fitting a model with its corresponding optimal parameters. The models' performances consisting of accuracy, precision, sensitivity, specificity, area under the curve, and kappa were estimated on the test set in each iteration. The aggregation of all iterations formed various distributions of the above performance measures. Figure 4:2 gives a summary of the implemented methodology.

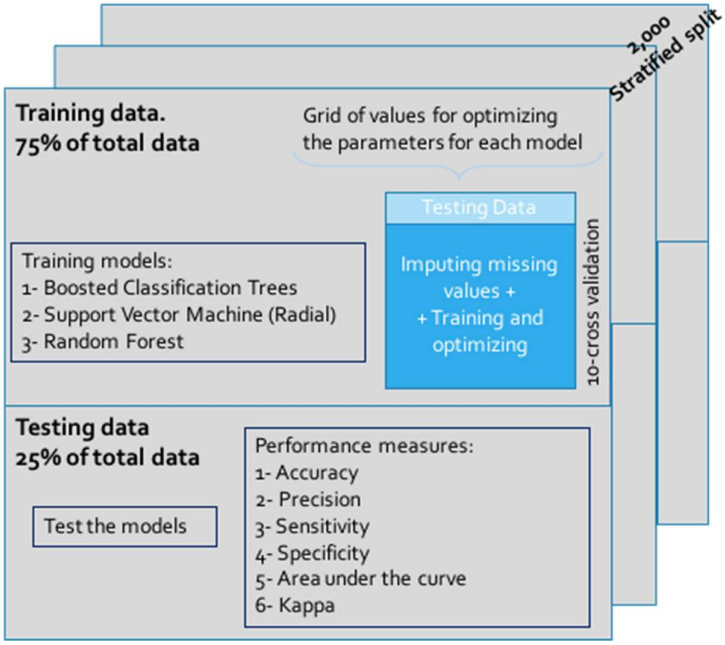
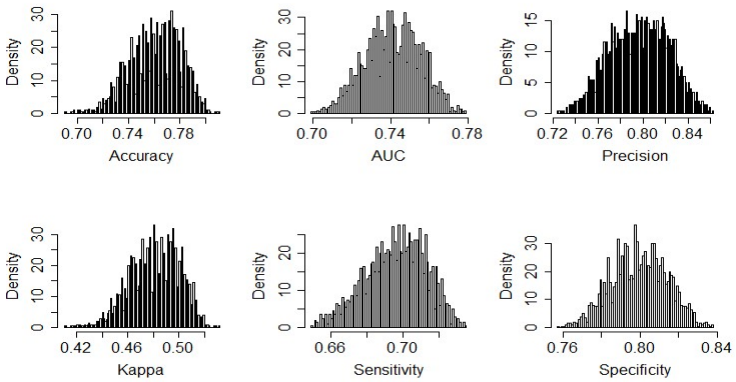
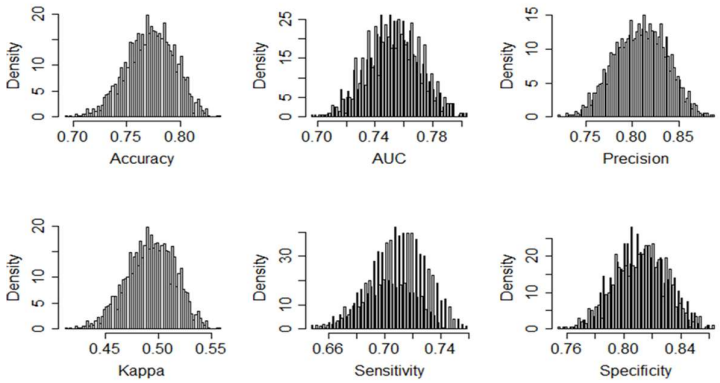


Figure 4:2 Summary of the implemented methodology

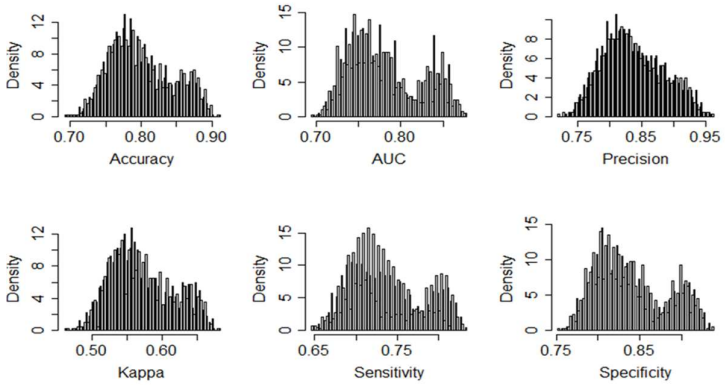
Figures 4:3 a, 4:3 b, and 4:3 c show histogram plots of the Monte Carlo simulations for the highest-performing models: support vector machine with the radial kernel, random forests, and boosted classification trees, respectively. Radial support vector machine models (see Figure 4:3 a) had a mean accuracy of 0.76, with a 95% confidence interval of [0.72, 0.80]. These models achieved a high level of mean specificity of 0.81 (95% CI [0.77, 0.84]). In contrast, random forests models (see Figure 4:3b) scored a mean accuracy of 0.78 (95% CI [0.75, 0.81]). Finally, boosted classification trees models scored the highest performance among all proposed models (see Figure 4:3). These models achieved a mean accuracy of 79.2% (95% CI [0.72, 0.84]) and a mean kappa of 0.56 (95% CI [0.48, 0.64]). The latter shows significant predictive information of the input attributes over first episode psychosis. We note a good predictive power and stability of these models, based on an acceptable level of variation of their performance measures evaluated across extensive Monte Carlo experiments.



(a) Monte Carlo simulation for Radial SVM



(b) Monte Carlo simulation for random forests



(c) Monte Carlo simulation for boosted classification tree

Figure 4:3 Monte Carlo simulations.

As mentioned before, a significant proportion of this variation may be explained by the uncertainties in the data, represented by the high proportion of missing values still present in the dataset after the removal of the attributes with more than 50% of missing values.

4.3 Cannabis attributes' predictive information over first-episode psychosis

After performing Monte Carlo simulations, the best performing models were further analysed to better understand the predictive power of the cannabis-related attributes over first episode psychosis. Moreover, we investigated the link between cannabis-related attributes and first episode psychosis via association analysis and Bayesian inference-based techniques.

4.3.1 Predicting first-episode psychosis without cannabis attributes

We re-fit best performing models with the three chosen algorithms, but this time with the cannabis-related attributes represented in Table 4:1 removed from the dataset; the performances obtained with and without the cannabis-related attributes are compared using student's t-test. That is, the predictive value of cannabis-related attributes concerning first-episode psychosis is demonstrated by showing that there is a statistically significant difference between the performances of the predictive models built with and without the cannabis variables. We were inspired in this approach we propose here by the Granger causality techniques, which are used to demonstrate that some variables have predictive information on other variables in a regression context (as opposed to classification in this case) [16].

Our analysis shows that the accuracy of all models decreased by around 6% if the cannabis-related attributes are removed from the process of building the predictive models. If we compare, for instance, the accuracies of the best two random forests models obtained on the data sets with and without the cannabis attributes, the p-value obtained for the one-tailed t-test was 0.00006. We conclude that the model with cannabis attributes has higher predictive accuracy. In other words, the additional cannabis variables jointly account for predictive information over first episode psychosis.

4.3.2 Cannabis use and first-episode psychosis associations

To further explore the link between the cannabis attributes and first episode psychosis, we look into detecting patterns in data with association analysis and Bayesian inference techniques such as Apriori [141] and Bayesian Networks [87] using WEKA 3.6.15 (December 2015) [71].

1- Cannabis user=Yes & Cannabis=Daily & Cannabis type=Skunk → Patient
(conf=.85, 95%CI[.773,.899]
2- Cannabis user=Yes & age1st14=Yes & Cannabis type=Skunk → Patient
(conf=.81, 95%CI[.723,.87]
3- Cannabis user=Yes & Cannabis type=Skunk → Patient
(conf=.79, 95%CI[.72,.839]
4- Cannabis user=Yes & Cannabis=Daily → Patient
(conf=.74, 95%CI[.67,.793]
5- Cannabis user=Yes & Cannabis=Daily & ageFirstCannabis=15 → Patient
(conf=.73, 95%CI[.643,.805]
6- Cannabis user=Yes & Duration =above 6 → Patient
(conf=.71, 95%CI[.634,.778]

Figure 4:4 Top association rules.

A repetitive fine-tuning of Apriori led to the detection of the top six rules, represented in Figure 4:4. The quality of these rules is expressed by their confidence estimates, and by 95% confidence intervals for these estimates. The rules represent patterns in the general local population in the mentioned area, since the data sample is representative of this area's population. The first rule states that if a participant were a cannabis user who consumes skunk daily, then there is an 85% likelihood that this participant is a first-episode psychosis patient. This rule shows evidence of a strong association between using high potency cannabis such as skunk daily, and first episode psychosis. If this type of cannabis is used daily or less often, then the likelihood of first-episode psychosis decreases from 85% to 79%, as expressed by rules 1 and 3 together. For a general type of cannabis which is used daily, the likelihood of a user to be a first-episode psychosis patient decreases from 85% to 74%, as expressed by rules 1 and 4 together. Rule number 2 supports findings from [15] [126] by associating the age of the first use of high potency cannabis (skunk) with the psychosis onset. Rule number 5, having 73% confidence, is consistent with findings from [142] regarding the onset of psychosis among cannabis users in relation to a cannabis consumption starting at age 15 and younger. Finally, rule number 6 expresses the finding that if a participant were a cannabis user who has used

cannabis daily for at least 6 months, then there is a 71% likelihood that this participant is a first-episode psychosis patient. The rules discovered in Figure 4:4 have presented to experts. They have confirmed the reliability of the discovered roles, as well as the novelty of rule 6 that associate the duration of the cannabis use with first episode psychosis.

Recently, Bayesian networks have been used in psychiatry as a decision model for supporting the diagnosis of dementia, Alzheimer's disease, and mild cognitive impairment [143]. We have applied this machine learning technique to detect further the interaction between first episode psychosis and the cannabis attributes. As such only cannabis related attributes were used as predictors in fitting the Bayesian network model, whose DAG is depicted in Figure 4:5. The model details suggest that duration and cannabis type are among the most predictive attributes. The Bayesian network probability distributions show that subjects who started using cannabis by age 15, and consumed cannabis daily for more than six months, are more than twice as likely to be patients rather than controls. On the other hand, subjects who started using cannabis by age 15 and consumed cannabis only at the weekend for more than six months, increase their chance by 1.5 times to be patients rather than controls. In addition, the model confirms that subjects who used skunk daily are twice as likely to be patients rather than controls. These findings further support the idea that cannabis use could lead to the onset of first-episode psychosis.

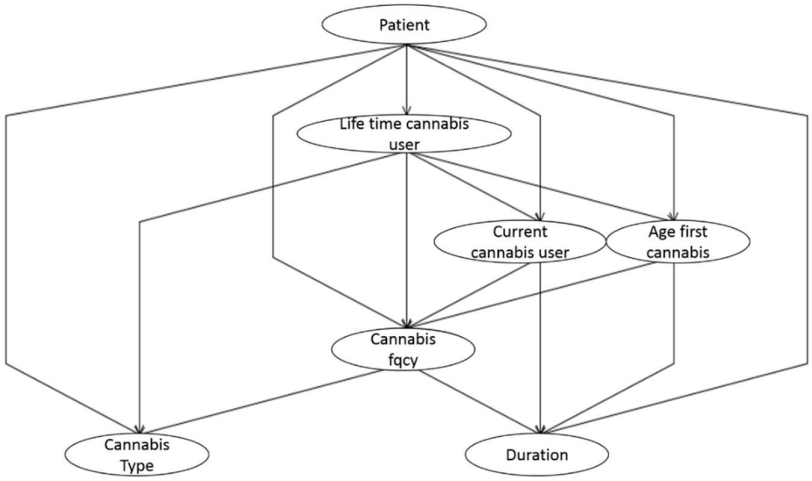


Figure 4:5 Bayesian Network for cannabis variables.

Several studies have linked cannabis use to the significant increase in the risk of psychiatric disorders, but the patterns causing the onset of first-episode psychosis are not always easy to determine. The present study contributed to several findings, some of which support previous works such as [15], and some others express novel/ previously unreported links between cannabis use and first-episode psychosis.

4.3.3 Cannabis use duration and first-episode psychosis associations

As far as we are aware, associating cannabis use duration with first-episode psychosis is one of the unreported links between cannabis use and first-episode psychosis. This led us to investigate further the relationship between the duration of cannabis use and first-episode psychosis. To do so, we first removed any participant who never consumed cannabis to determine the patterns of participants who consume cannabis. Then, we detected and removed ten outliers using Euclidian distance on the cannabis use duration attribute. Finally, the duration attribute was discretised into the same intervals that were applied above. Figure 4:6 shows a histogram of the cannabis use duration attribute after the preprocessing. It shows that participants who consumed cannabis for less than three months are 1.6 times more likely to be patients than being controls. In addition, by increasing the duration of cannabis use by the participants to six months, Figure 4:6 shows that the participants will be 2.3 times more likely to be patients than controls. Moreover, if we increase the duration of the cannabis use in the participants for more than six months, the participants will be 2.6 times more likely to be patients than controls. Finally, we conclude that by increasing the duration of cannabis use we detect an increase in the likelihood of being a patient with first-episode psychosis.

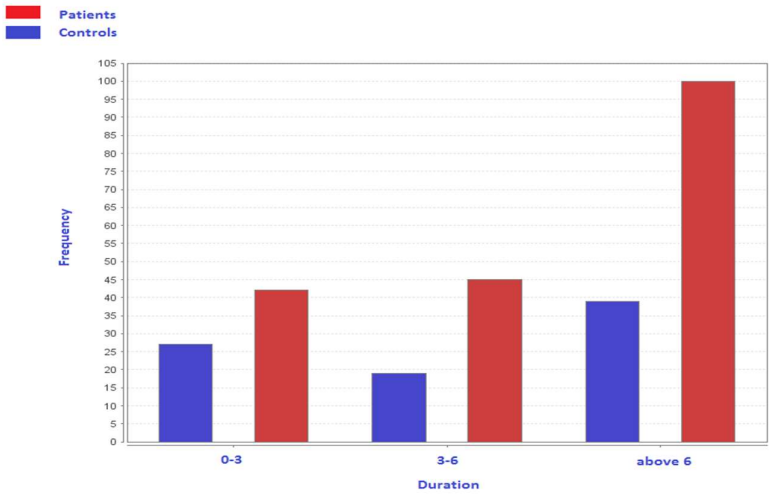


Figure 4:6 Histogram of the cannabis use duration attribute.

4.4 Conclusion

The aim of this chapter has been to propose a novel synergistic machine learning and statistical approach to pattern detection and to developing predictive models for the onset of first-episode psychosis. To our knowledge, previous studies on the link between cannabis use and first-episode psychosis investigated this highly important (and further to be revealed) relationship via conventional statistical methodologies and techniques and did not tackle the predictability of this condition in relation to cannabis use.

One direction of research explored in the subsequent chapters consists of investigating the impact of including sampling methods and post-processing techniques in order to enhance the prediction performance. Moreover, the ensuing chapters investigate into the first-episode psychosis predictiveness enhancements by considering Gaussian processes, neural networks, and deep learning approaches.

Chapter 5 A new machine learning framework for understanding the link between cannabis use and first-episode psychosis

Recently, several studies have begun to investigate the existence of links between cannabis use and psychotic disorders. This chapter proposes a refined machine learning framework for understanding the links between cannabis use and first episode psychosis. The novel framework concerns extracting predictive patterns from clinical data using optimised and post-processed models based on Gaussian processes and support vector machines algorithms. The cannabis use attributes' predictive power is investigated, and we demonstrate statistically and with ROC analysis that their presence in the dataset enhances the prediction performance of the models with respect to models built on data without these specific attributes.

5.1 Problem description

Several studies, including Chapter 4, have begun to investigate the existence of links between cannabis use and psychotic disorders. In this chapter, we propose a new computational psychiatry and machine learning framework based on developing optimised models for predicting the onset of first-episode psychosis with Gaussian processes (GP) and support vector machines (SVM) from the caret 6.0 package (January 2016) [136].

This chapter differs from the previous one by introducing a novel framework and using more advanced machine learning algorithms that employ different kernel methodologies such as GP and SVM with linear, radial, and polynomial kernels. Furthermore, the framework we present here integrates data pre-processing, model tuning, and model post-processing with receiver operating characteristic (ROC) optimisation based on the maximum accuracy cut-off threshold, and model evaluation with k-fold cross-testing. This sequence of enumerated phases is repeated 2000 times for each GP and SVM with radial and polynomial kernels to study the potential variation of the performances of the resulting models. We also investigate the cannabis use attributes' predictive power by establishing statistically that their presence in the dataset augments the models' performances.

The aim is to develop a predictive modelling approach to help in understanding the link between first-episode psychosis and cannabis use. The dataset on which we based our study was collected by psychiatry practitioners and was used in chapter 4 [15]. It comprises an extensive set of variables, including demographic, drug-related and other variables, with specific information on participants' history of cannabis use, some of which are illustrated in Table 4:1.

5.2 Methods

5.2.1 Data pre-processing

The quality of data may significantly affect the performance of the predictive models [2]. In order to help improve the quality of the data and, consequently, of the predictive mod-

els, the clinical data is pre-processed. Data pre-processing usually deals with the preparation and transformation of the initial dataset. In this chapter, we applied numerous pre-processing techniques such as missing values imputation, class balancing, and feature selection to improve the efficiency and the ease of the modelling process.

Firstly, in term of missing values imputation, we applied random forest imputation from the RandomForest 4.0-7 package (June 2015) [144]. Although this method is computationally expensive, it enhanced the predictive power of the final models.

Secondly, the synthetic minority over-sampling technique (SMOTE) from the DMwR 0.4.1 package (August 2013) [145] was selected to treat the unbalanced classes that existed in the data. SMOTE chooses a data point randomly from the minority class, determines the K nearest neighbours to that point and then uses these neighbours to generate new synthetic data points using slight alterations. Our analysis used five neighbours. The results show that SMOTE leads to an increase in both the area under the ROC curve (AUC) and the accuracy.

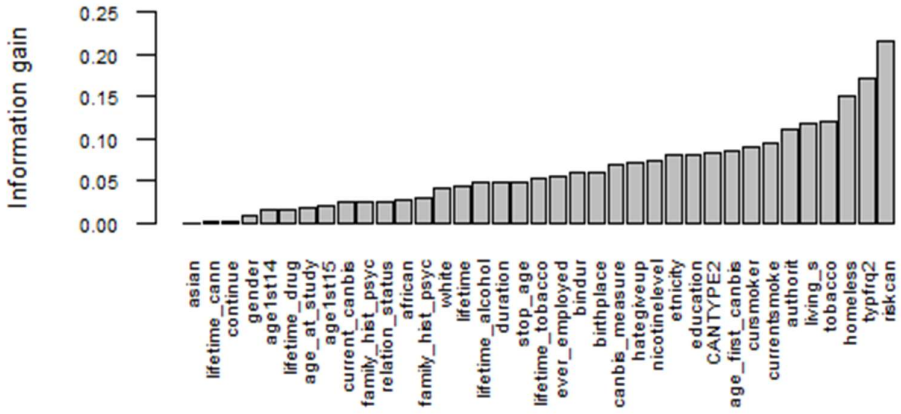


Figure 5:1 Attributes' predictive power with respect to Information Gain.

Finally, we applied a feature selection technique based on the information gain [79]. To do so, we evaluated the information gain for each attribute with respect to the class. Such techniques are often used with forward selection or backward elimination, which considers only removing the feature subset with least ranking values. In this chapter, we apply information gain to filter out the attribute that does not have predictive power regarding information gain. Figure 5:1 illustrates the attribute predictive powers concerning the information gain. Figure 5:1 shows that attributes such as riskcan and

typefrq2, which are cannabis measures, are the highest ranked attributes while attributes such as gender are among the least ranked attributes. Initially, this indicates that some of the cannabis use attributes have more predictive power than some of the demographic attributes.

5.2.2 Predictive modelling

To develop optimised predictive models for first-episode psychosis, we controlled the values of the parameters for each of the considered algorithms using chosen grids. Predictive models have been fitted in a five-fold cross-validation procedure on each training set after pre-processing techniques were applied on the same training set and have been tested on each test set. Models based on SVM and GP were optimised to maximise AUC.

First, SVM models were tuned with different kernels such as SVM with the radial kernel (SVMR) and SVM with the polynomial kernel (SVMP). The optimal SVM models were obtained with SVMR after tuning the parameters cost and gamma over 10 values. The optimal values for cost and gamma were 16 and 0.004, respectively.

Then, GP models were tuned with different kernels such as GP with the radial kernel (GPR) and GP with the polynomial kernel (GPP). The optimal GP models were obtained with GPP, with the parameters degree and scale, and tuned over 10 values. The optimal values for degree and cost were three and 0.01, respectively.

5.2.3 Predictive model post-processing

ROC curves allow visual analyses of the trade-offs between a predictive model's sensitivity and specificity regarding various probability cut-offs. The curve is obtained by measuring the sensitivity and specificity of the predictive model at every cutting point and plotting the sensitivity against 1-specificity. Figure 5:2 shows the ROC curves obtained for two of our predictive models, namely SVMR and GPR. The curve shows that SVMR performs better than the other model regarding the evaluation dataset.

Several methods exist for finding a new cut-off threshold on the ROC curve. In this chapter, we find the point on the ROC curve corresponding to the highest accuracy.

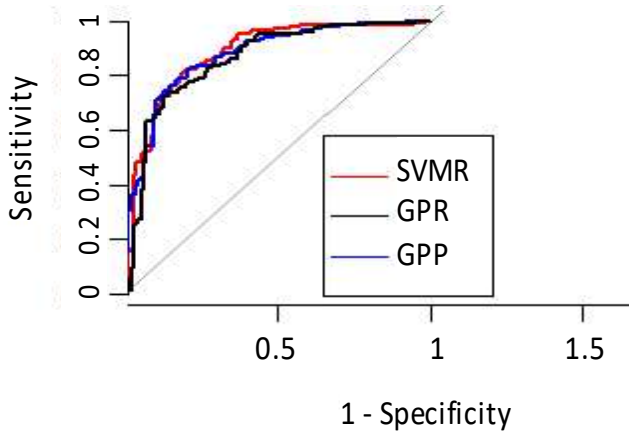


Figure 5:2 ROC curves for 3 models: SVMR, GPP and GPR.

5.2.4 Overall modelling procedure

The modelling procedure used, which is based on pre-processing, model optimisation and post-processing, was successfully used in [17]. Figure 5:3 gives a summary of the implemented methodology. First, the dataset is randomly split, with stratification, in 60% and 40% parts denoted here by D1 and D2, respectively. Then, D1 is used for training and for optimising the model, as explained in subsection 5.2.2. Different pre-processing methods that were explained in subsection 5.2.1 were appropriately integrated into the cross-validation. Finally, to further enhance the model performance, the post-processing and model evaluation methods were applied to the optimised model using k-fold cross testing on the D2 dataset as shown in Figure 5:3. In the k-fold cross testing procedure, we produce k post-processed model variants of the original optimised model. First, we create k-stratified folds of the D2 dataset. Then, k-1 folds are used to find an alternative probability cut-off on the ROC curve with one of the three specific methods presented in subsection 5.2.3, thus obtaining a post-processed model variant. The remaining one-fold is scored with the post-processed model variant based on the newly found cut-off point. This procedure enhances the predictive models and ensures that the data for scoring are always distinct.

The main advantage of framework introduced here over other methods such as the nested cross-validation is that the new framework integrates two inner cross validations that act jointly instead of one inner cross-validation. This ensures keeping separate the data used for building optimised prediction, the data used for post-processing with receiver operating characteristic (ROC) optimisation, and the data used for scoring.

Another advantage (mentioned above) is that the new framework integrates data pre-processing, model tuning, and model post-processing with receiver operating characteristic (ROC) optimisation based on the maximum accuracy cut-off threshold, and model evaluation with k-fold cross-testing. Nested cross validations usually lack data that is used for model post-processing optimisation.

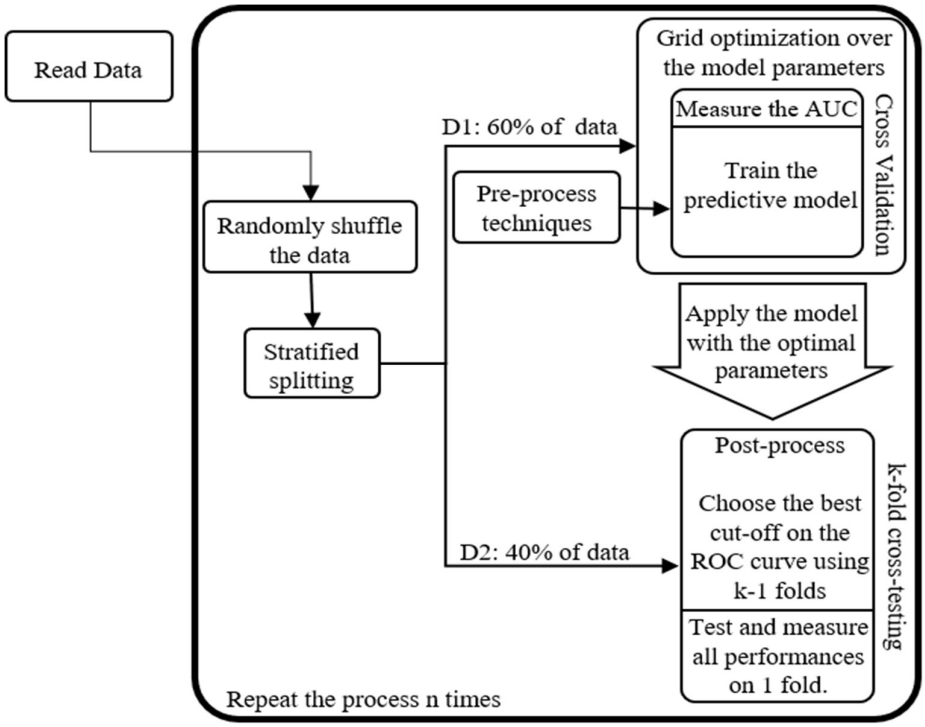


Figure 5:3 Summary of the implemented methodology with the k-fold cross-testing method.

5.3 Results

Due to expected potential variations of the predictive models' performances, we conducted extensive repeated experiments simulations to study these variations and the models' stabilities. The simulations consisted of 2000 iterations of the procedure explained in section 5.2.4. The models' performances concerning the accuracy, AUC, sensitivity, and specificity were evaluated for each iteration.

The aggregation of all iterations yielded various distributions of the above performance measures. These distributions were then visualised using box plots in Figure 5:4 to capture the models' performance capability and stability.

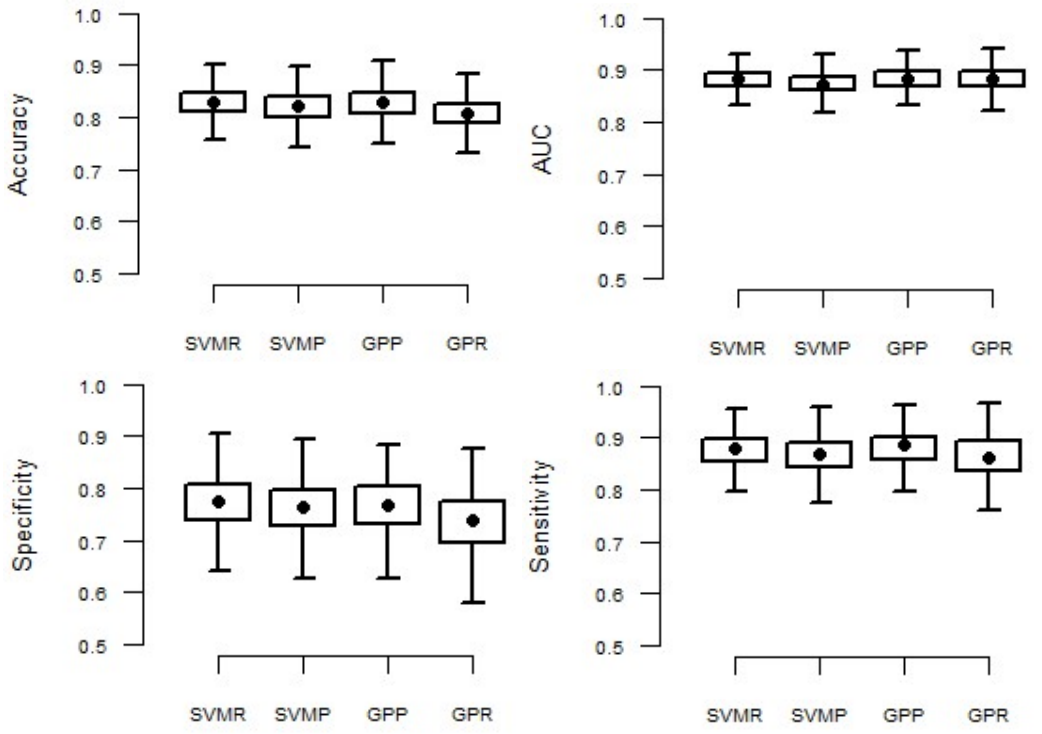


Figure 5:4 2000 repeated experiments simulations on Support Vector Machines with Radial (SVMR) and Polynomial kernels (SVMP) and Gaussian Processes with Radial (GPR) and Polynomial kernels (GPR).

In addition, estimations of the predictive models' performances regarding means and standard deviation (SD) are shown in Table 5:1. We report results regarding models that are post-processed with ROC optimisation based on the largest accuracy cut-off method, as explained in section 5.2. The results show that SVMR achieved the best results with a mean accuracy of 0.83 (95% CI [0.78, 0.88]) and a mean sensitivity of 0.87 (95% CI [0.81, 0.93]), similar to the results achieved by GPP. The rest of the predictive models scored a mean accuracy of 82%, which is better than all the performances reported by our work in chapter 4. We interpret this performance improvement by the fact that our methodology presented in the current chapter was significantly enhanced and became more sophisticated with the addition of the approach we proposed explicitly for model post-processing and evaluation with the novel k-fold cross testing procedure explained in subsection 5.2.4.

Overall, we find that the models, especially SVMR and GPP, have good predictive power and stability, based on an acceptable level of variation in their performance measures evaluated across extensive repeated experiment simulations. In addition, the

results indicate that the performance differences between the different methods for selecting the ROC cutting points are not significant regarding the four performances.

Model	Accuracy		AUC		Sensitivity		Specificity	
	Mean	SD	mean	SD	mean	SD	mean	SD
SVMR	0.83	0.03	0.88	0.02	0.88	0.02	0.77	0.04
SVMP	0.82	0.02	0.87	0.02	0.86	0.03	0.76	0.05
GPR	0.81	0.03	0.86	0.04	0.86	0.04	0.74	0.05
GPP	0.83	0.02	0.88	0.02	0.88	0.03	0.76	0.04

Table 5:1 Estimations of the predictive models' performances.

After performing the repeated experiment simulations, we further investigated the predictive models in order to better comprehend the predictive power of the cannabis-related attributes over first-episode psychosis via statistical tests. To do so, we re-fitted our performing models but removed the cannabis-related attributes, represented in Table 4:1, from the dataset. Then, we compared the performances of the models built with and without the cannabis-related attributes using the t-test. We demonstrated the predictive value of cannabis-related attributes with respect to first-episode psychosis by showing that there is a statistically significant difference between the performances of the predictive models built with and without the cannabis variables.

Our analysis showed that the accuracy of SVMR decreased by 6% if the cannabis-related attributes were dropped from the process of building the predictive models, as shown in the right-hand image in Figure 5:5. If we compare, for instance, the accuracies of the SVMR models built on the data sets with and without the cannabis use attributes, the p-value obtained for the one-tailed t-test was 0.0006. This means that the predictive models with cannabis attributes have higher predictive accuracy than the models that were built without the cannabis attributes. This leads us to conclude that the additional cannabis variables jointly account for predictive information on first-episode psychosis. These results are consistent with findings from chapter 4. In addition, we demonstrated that there is a significant difference between the ROC curves of the predictive models built with and without the cannabis variables, as shown in the left-hand image in Figure 5:5. This also confirms the idea that the predictive models with cannabis attributes have higher predictive power than the models that were built without the cannabis attributes.

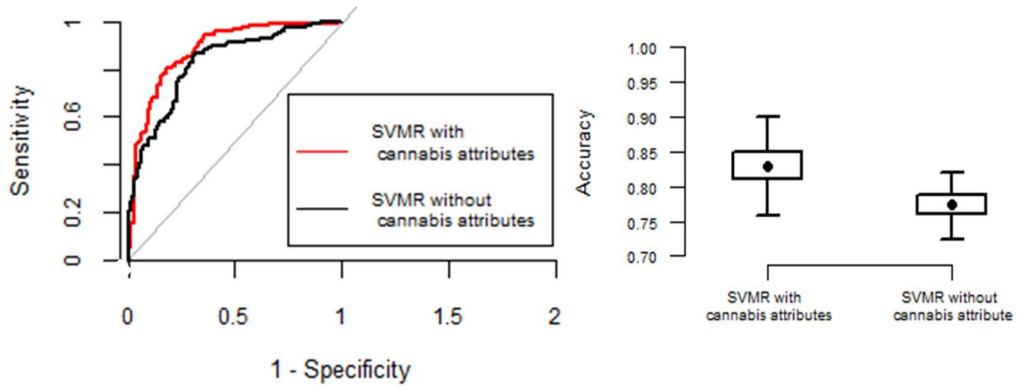


Figure 5:5 Left: ROC curves for optimised SVMR, with and without the cannabis attributes. Right: boxplots for 2000 repeated experiments simulations for optimised SVMR, with and without the cannabis attributes.

5.4 Conclusion

The advent of machine learning has so far proved to be of prime importance and capability in various fields, and recently in medical research and healthcare. This chapter proposes a novel computational psychiatry and machine learning framework based on developing predictive models for the onset of first-episode psychosis in the presence of clinical data including also cannabis use. We explored two types of state of the art machine learning algorithms, namely Gaussian processes and support vector machines. Models are tuned and further optimised via post-processing and evaluated with a k-fold cross testing methodology – a novelty introduced in our research studies [17] and presented in this chapter. To study the variation of the performances of the prediction models, the framework incorporates 2000 repetitions of the model building, optimising, and testing sequence. Experimental results show that the two machine learning algorithms lead to comparable models, with a slight advantage for support vector machines ahead of Gaussian processes, an advantage that is not statistically significant.

Our best models score an average accuracy of 83%, which is above all of the accuracy performances we achieved in previous studies, such as chapter 4. This chapter extends on previous work as in chapter 4 by proposing a new machine learning framework based on a novel methodology in which models are post-processed based on ROC optimisation, and evaluated with the recent method of k-fold cross testing which adopted from [17]. Moreover, in this new methodology, we developed optimised models with other powerful techniques such as Gaussian processes not addressed in chapter 4. We also

demonstrate statistically that the best models' performances decrease if cannabis attributes are removed from the analysis. This fact is also confirmed and illustrated by ROC analysis.

A significant proportion of the models' performance variation may be explained by the uncertainties present in the data, represented by the high proportion of missing values. Chapter 4 and chapter 5 propose a fix cutting off for the missing values percentage followed by imputations to replace the missing values. However, it would be interesting to investigate further in the next chapter how this prediction performances variation evolves by limiting the uncertainty in the data. Another direction of research that will be explored next chapter consists of investigating into the first-episode psychosis predictiveness enhancements by considering artificial neural networks and deep learning approaches.

Chapter 6 Predicting first-episode psychosis associated with cannabis use with artificial neural networks and deep learning

In recent years, a number of studies have investigated the existence of links between cannabis use and psychotic disorder. More recently, artificial neural networks and in particular deep learning have made a revolutionary step in pattern recognition and machine learning. This chapter proposes a novel machine learning approach, based on neural networks and deep learning algorithms, to developing highly accurate predictive models for the onset of first-episode psychosis. Our approach is also based on a novel methodology of optimising and post-processing the predictive models in a computationally intensive framework. A study of the trade-off between the volume of the data and the extent of uncertainty due to missing values, both of which influence predictive performance, enhanced this approach. The performance capabilities of the predictive models are enhanced and evaluated by a methodology consisting of novel model optimisation and testing, which integrates a phase of model tuning, a phase of model post-processing with ROC optimisation based on maximum accuracy, Youden and top-left methods, and a model evaluation with the k-fold cross-testing novel methodology (explained in the previous chapter). We further extend our framework by investigating the cannabis use attributes' predictive power and demonstrating statistically that their presence in the dataset enhances the prediction performance for the artificial neural networks presented in this chapter. Finally, the model stability is explored via simulations with 2000 repetitions of the model building and evaluation experiments. The results show that the average accuracy in predicting first-episode psychosis achieved by our models in intensive Monte Carlo simulation is about 89%, which represents a significant improvement with respect to the already sophisticated prediction modelling we introduced in the previous chapters. We account for this significant increase in prediction performance to the most powerful state of the art algorithms existing today in machine learning, i.e. deep learning algorithms.

6.1 Problem description

Artificial neural networks and in particular deep learning have revolutionised many areas that use machine learning. In this chapter, we propose a novel machine learning approach based on neural networks and deep learning techniques to develop predictive models for the onset of first-episode psychosis. The dataset on which we based our study was collected by psychiatry practitioners and has been used in previously conducted studies, such as Chapter 4 and Chapter 5.

Our approach features a gradual control of the limitation of the uncertainty present in the data due to missing values that are usually inherent in clinical datasets due to patients missing appointments, patients not reporting all details, etc. This feature involves considering different thresholds for allowed levels of missingness (per attributes and per records) in the data sets that we call 'cutting points', in order to examine how the prediction models' performances may vary with these thresholds. Our approach is also based on a novel methodology of optimising and post-processing the predictive models in a computationally intensive framework. Furthermore, we extend our approach by encapsulating a novel post-processing k-fold cross-testing method, which is a contribution of our work described in chapter 4, in order to further optimise and test these models. The results show that the accuracy in predicting first-episode psychosis achieved by our best models in intensive Monte Carlo simulation falls between 85.13% and 91.54%, with an average of about 89%.

The rest of this chapter is organised as follows: Section 2 presents our methodology for predicting first-episode psychosis, based on artificial neural networks, and deep learning. Our novel framework is also based on a novel methodology of optimising and post-processing the predictive models in a computationally intensive framework. A novel study of the trade-off between the volume of the data and the extent of uncertainty due to missing values, both of which influence predictive performance, enhanced this approach. Finally, we extended our approach by encapsulating the novel post-processing k-fold cross-testing method in order to further optimise and test these models. In section 3, we investigate the outcomes of the extensive Monte Carlo simulations in order to study the variation of the models' performances. The section also builds optimised prediction models without the cannabis attributes to determine whether there is a statistically significant

difference with respect to the performances of the neural network models using the cannabis attributes. Finally, the conclusion of this chapter is presented in section 4.

6.2 Methods

6.2.1 A trade-off between the extent of missing values and the dataset size

A trade-off between the extent of missing values present in the dataset and the dataset size needed to be investigated from the point of view of the predictive power of the models that can be built on the dataset. The intuition is that by using a larger subset of the available dataset in the analysis, one would obtain a positive effect on the performance of predictive models (since more data is used to build the models). Nevertheless, this larger subset may also encapsulate more uncertainty due to the presence of more missing values, which usually has a negative effect on the predictive models (even with imputation). Therefore, different cutting points, defined as the thresholds for the percentage of missing values (or level of missingness) allowed in attributes and records, respectively, were considered in order to study the variation of the predictive power of subsets of the dataset. Attributes and records presenting some levels of missingness up to the respective cutting points or thresholds, respectively, were kept in the dataset, and the remaining ones were removed. The considered cutting points for the records were 10%, 20%, and 100%. For instance, 30% in this grid means that we keep only the records that have up to 30% missing values in the dataset (and 100% means practically that all records are kept in the dataset). Moreover, the cutting points for the attributes were identified by first determining the percentage of missing values for each attribute, and then ordering these percentages and splitting them into 20 equal groups. The extreme values in each group formed the cutting points for the attributes.

Overall, these cutting points were applied to the dataset and compared with the performance of single-layer neural network tuned models in an attempt to determine optimal cutting points, which were those for which these models had the highest accuracy. Once these cutting points were determined, they were applied, and a final dataset was

thus obtained as the outcome of a trade-off between the extent of missing values present in the dataset, and the dataset size.

How exactly did we proceed to obtain this final dataset? Note that we do not perform a full optimisation on all pairs of cutting points for attributes and records to determine this final dataset (because training and tuning neural networks is a computationally expensive procedure), but merely apply a heuristic in our framework. Initially, we search for an optimal value among all the attribute cutting points, and apply it to the dataset. In our case, this was 92%. Then, we applied different record cutting points on the resulting dataset following the grid mentioned above, and we determined the best cutting point, which was 70% in our case. To compare the cutting points and select the best ones, the criterion was the accuracy of the single-layer neural network from the caret 6.0 package (January 2016) [136] which has been tuned on the training set (70% of the data), in a 5-fold cross-validation procedure, on a 10x10 grid for the number of hidden units, and decay values to prevent overfitting with regularisation methods. Random forest imputations of missing values were applied. The models' performances consisting of accuracy and kappa were estimated on the test set (30% of the data).

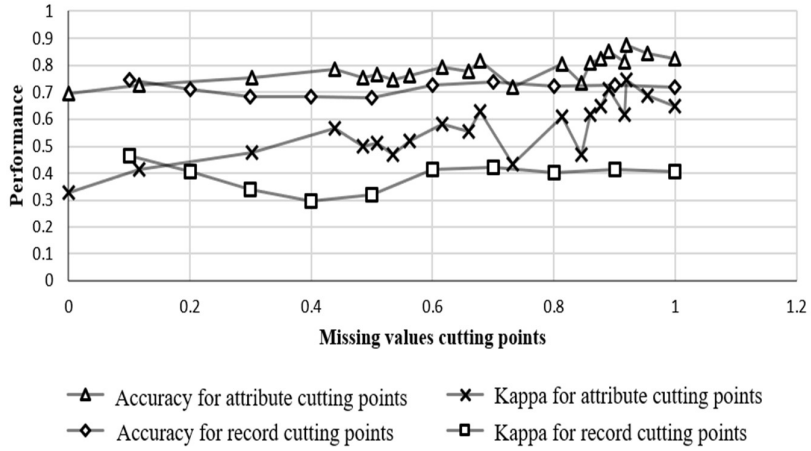


Figure 6:1 Model performance for record and attribute cutting points.

Figure 6:1 illustrates the process in which we observed a decrease in the performance when all the attributes were included or when the cannabis attributes were not present in the obtained dataset.

By applying the determined 92% cutting point for the attributes and 70% cutting point for the records to the original dataset, we obtained 107 attributes and 628 records

divided into 360 patients and 268 controls, on which the main phase of predictive modelling with various algorithms was developed and presented in what follows. We note that the proportion of controls and patients in the final dataset are approximately the same as in the original dataset, indicating that the current dataset is representative.

6.2.2 Missing values imputation

The predictive power of the data may depend significantly on the way that missing values are treated. Some machine learning algorithms, such as decision trees [1], have the capability to handle missing data outright. However, most machine learning algorithms do not have the capability to handle missing data. In many situations, missing values are imputed using a supervised learning technique, such as k-nearest neighbour (k-NN). These imputation techniques do not have theoretical formulations but are often applied in practice [2]. Several imputation techniques, such as the k-NN imputation, the tree bagging imputation from the caret 6.0 package (January 2016) [136] and the random forest imputation from the RandomForest 4.0-7 package (June 2015) [144] were considered in this chapter. The last method, although it was the most computationally expensive, produced the best results regarding the performance of the final predictive models in this context of the obtained dataset.

6.2.3 Training and optimising (tuning) predictive models

For the purpose of developing optimised predictive models for first-episode psychosis, the values of the parameters for each of the considered algorithms were controlled by chosen grids. Predictive models were fitted, in a 5-fold cross-validation procedure, on each training set after pre-processing techniques were applied on the same training set and have been tested on each test set. Models based on neural networks with a single-layer, neural networks with multi-hidden-layers, and deep networks, were optimised (tuned) based on maximising AUC, the area under the ROC curve.

The single-layer neural network was tuned over 10 values of the size (i.e. the number of hidden units) and 10 values of the decay (i.e. the weight decay), which is the parameter in the penalisation method for model regularisation to avoid overfitting, similar to the penalisation method in ridge regression, based on the L2 norm [2]. The optimal values were 3 and 0.01, respectively. The neural networks model with multi-hidden layers

was tuned over 10 values for each of the 3 hidden layers (i.e. 10 values for the number of hidden units in each layer), and 10 values for the decay. The optimal values were 5, 5, and 5 for the 3 layers, and 0.01 for decay, respectively.

As for the deep networks, we employed the H2O's deep learning software version 3.10.3.6 (June 2017), which is based on a multi-layer feedforward artificial neural networks model trained with stochastic gradient descent using back-propagation [146]. The deep networks usually contain a large number of hidden layers consisting of neurons with tanh, rectifier, and maxout activation functions. This type of model has many parameters, but it was designed to reduce the number of parameters that the researcher has to specify by applying feature selection and early stopping techniques. We employed deep networks using the method of Gedeon [147] to select the best attributes. In our experiments, the early stopping was set to let it stop automatically once the area under the curve AUC stops improving, in particular, when AUC does not improve by at least 1% for 10 consecutive scoring events.

Furthermore, a grid optimisation was used with the parameters that need to be tuned, such as the activation function, the number and sizes of the hidden layers, the number of epochs, and the 2 parameters corresponding to the L1 and L2 regularisations (i.e. the decays) for preventing overfitting.

The models were tuned overall activation functions, and over 3, 4, …, 25 layers and 30, 35, …, 50 layers. The number of units in each layer had the values 50, 100,…, 250. In addition, we used the values 2, 3, 5, and 10 for tuning the number of epochs. Finally, the parameters for the L1 and L2 regularisations were each tuned over the values 10^{-1}, 10^{-2},…, 10^{-10}.

After performing the proposed techniques, the optimal values selected for the deep learning model are rectifier as an activation function, five epochs, and eight hidden layers of 200 neurons each. As for the L1 and L2 parameters, the optimal values were 10^{-4} and 10^{-5}, respectively.

6.2.4 Treating unbalanced classes

When there is a priori knowledge of a class imbalance, one direct method to reduce the imbalance's influence on model training is to select training set samples with roughly

equal event rates [2]. Treating data imbalances usually leads to better prediction models and a better trade-off between sensitivity and specificity.

In this chapter, we considered three sampling approaches to sub-sample the training data in a manner that mitigated the imbalance problem. The first approach was downsampling, in which we sampled (without replacement) the majority class to be the same size as the minority class. The second method was upsampling, in which we sampled (with replacement) the minority class to be the same size as the majority class. The last approach was the synthetic minority over-sampling technique (SMOTE) [145]. SMOTE selects a data point randomly from the minority class, determines the k-NN to that point and then uses these neighbours to generate new synthetic data points using slight alterations. Our analysis of neural networks and deep networks modelling in this chapter used five neighbours. The results show that the up-sampling procedure yielded no real improvement in the AUC or in the accuracy performances. Simple downsampling of the data also had no positive effect on the model performances. However, SMOTE with neural networks models led to an increase in both the AUC and the accuracy.

As mentioned before, data balancing supports a good trade-off between sensitivity and specificity. Another method that helps to balance sensitivity and specificity, or a good trade-off between the two performances, is model post-processing through the determination of new cut-off points on the ROC curves [2]. Our framework used three such methods, which can be seen as post-processing optimisations of the models. The first method found the point on the ROC curve closest to the top-left corner of the ROC plot, which represents the perfect model (100% sensitivity and 100% specificity). The second method is Youden's J index [148], which corresponds to the point on the ROC curve furthest from the main diagonal of the ROC plot. The third method, 'maximum accuracy', found the cut-off, which is the point with the highest model accuracy.

To further improve the model performance, a specially designed post-processing procedure and model evaluation were adapted in our modelling procedure. First, the dataset was stratified split randomly into 60% training data and 40% evaluation data. Then, the training data is used for training and for optimising the model, as explained in Section 5.2.3, in a cross-validation fashion, with AUC as the optimisation criterion, with and without class balancing. Different pre-processing methods such as missing values imputation and sampling methods as explained above were appropriately integrated into the

cross-validation. The optimal model obtained on the training data was then applied to the evaluation dataset in a specially designed post-processing procedure, the so-called k-fold cross-testing method presented in chapter 4 and in [17].

More precisely, in the k-fold cross-testing method, we produce k post-processed model variants of the original optimised model. First, we create k-stratified folds of the evaluation dataset. Then, k-1 folds are used to find an alternative probability cut-off on the ROC curve with one of the three specific methods presented above (top-left, Youden, and largest accuracy), obtaining a post-processed model variant. The remaining one-fold is scored with the post-processed model variant based on the newly found cut-off point. Finally, the whole procedure is repeated until all folds are used for scoring at their turn, then the predictions are integrated, and the model performance is measured on the complete evaluation dataset. We note here as an essential remark that in each such iteration of the procedure, the ROC optimisation data (the k-1 folds) and the scored data (the remaining fold) are always distinct, so the data for model post-processing and the data for scoring are always distinct [17].

6.2.5 Increasing model performance via optimised cut-off point selection on the ROC curve

As pointed out in the previous chapters, ROC curves allow visual analyses of the trade-offs between a predictive model's sensitivity and specificity regarding various probability cut-offs. The left-hand image in Figure 6:2 shows the ROC curves obtained for both the single-layer neural network and the multi-layer neural networks. They suggest that the multi-layer neural networks model performs better than the single-layer neural network on the evaluation dataset.

Multiple methods exist for finding a new probability cut-off on the ROC curve. First, one can find the point on the ROC curve that is closest to the perfect model (100% sensitivity and 100% specificity), which is the point with the shortest distance from the point (0, 1) as shown in the right-hand image in Figure 6:2. To find the shortest distance, $[(1 - \text{sensitivity})^2 + (1 - \text{specificity})^2]$ was calculated and minimised [149]. Another approach for finding an optimal cut-off point on the ROC curve is to find the largest distance from the diagonal to the ROC curve as shown in the right-hand image in Figure 6:2. This

is the point with the largest value for the Youden index, which is defined as (sensitivity + specificity -1) [148]. These are the two most popular methods for establishing the optimal cut-off [2] [150]. We used both of these methods, as well as the maximum accuracy approach, which determines the point on the ROC curve corresponding to the greatest accuracy (the blue point in Figure 6:2, right). In our analysis, the optimal cutting point was derived from independent sets, rather than from the training set or the evaluation sets, as shown previously. This is particularly important, especially for smaller datasets.

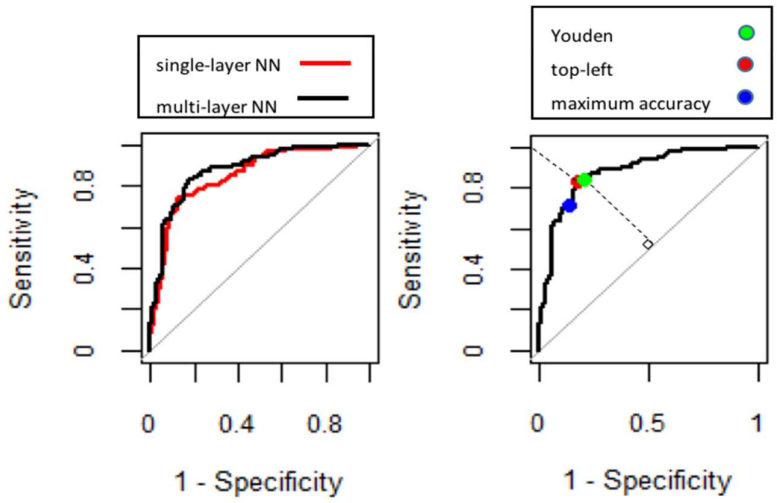


Figure 6:2 Left: ROC curves for 2 of our optimised neural network (NN) models: single-layer NN and multi-layer NN. Right: ROC optimisation post-processing of the multi-layer NN model, with 3 optimal cutting points: maximum accuracy, Youden and top-left methods.

6.2.6 Monte Carlo simulations with neural networks and deep learning

As in the previous chapters, we conducted extensive Monte Carlo simulations to study the stability of the neural network and deep network models. In particular, the simulations for each single-layer neural network, multi-layer neural networks and deep networks consisted of 2,000 iterations of the procedure included in the bold contour box of Figures 6:4 and 6:5. The models' performances consisting of accuracy, sensitivity, specificity, and kappa were evaluated in each iteration. The aggregation of all iterations formed various

distributions of the above performance measures. These distributions were visualised using histograms to capture the performance capability and stability of models, as shown in the results section. Figure 6:3 illustrates a summary of the implemented methodology with the k-fold cross-testing method and, a trade-off between the extent of missing values and the dataset size.

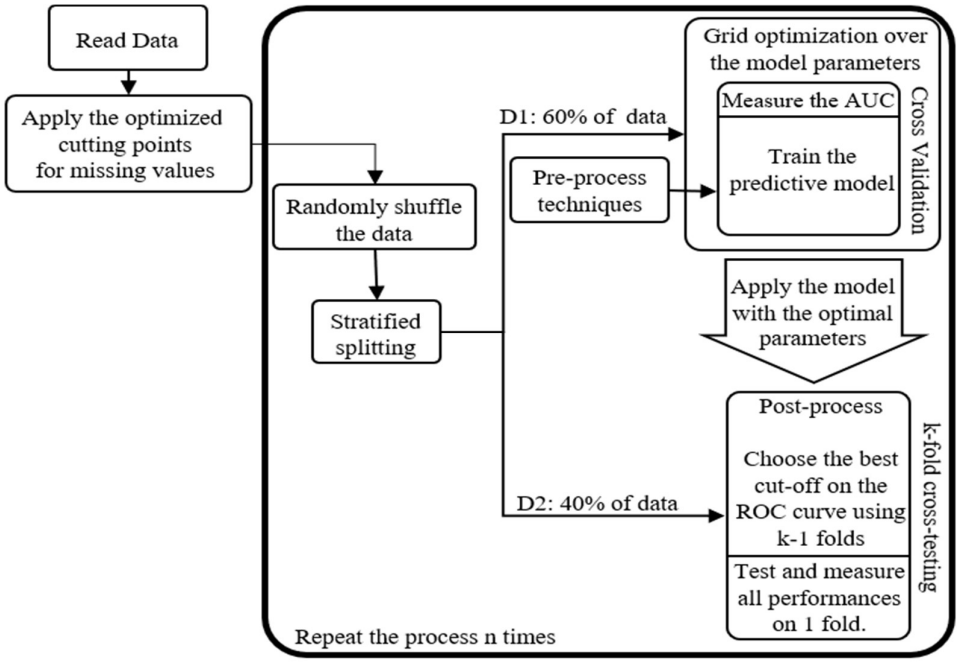


Figure 6:3 Summary of the implemented methodology with the k-fold cross-testing method and, a trade-off between the extent of missing values and the dataset size.

6.3 Results and discussion

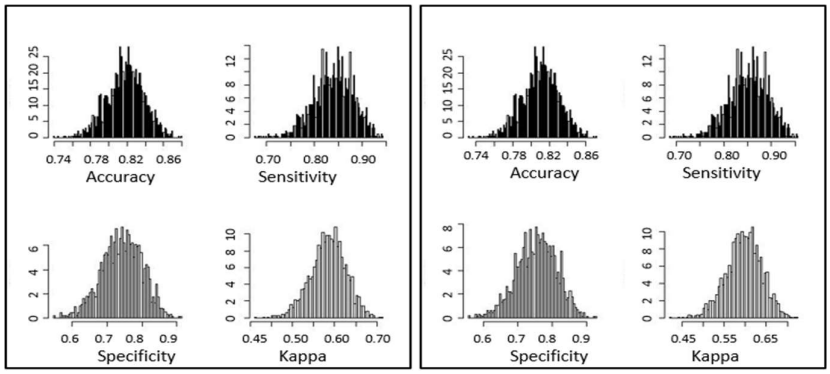
We present here the performances obtained with our approach to predicting first-episode psychosis, investigated with Monte Carlo simulations conducted on an 11 server cluster running R 3.3.3 (March, 2017) and specific libraries for neural network and deep learning namely caret 6.0 (January 2016) [136] and H2O 3.10.3.6 (June 2017), respectively. We should note that in this section we only report results regarding models that either are not post-processed, or are post-processed with ROC optimisation based on the largest accuracy cut-off method only. The other two methods for post-processing, namely top-left and Youden, led to comparable results.

The results presented in Table 6:1 show that the single-layer neural network scored a mean accuracy of 0.80 (95% CI [0.76, 0.84]) and a mean sensitivity of 0.84 (95%

CI [0.76, 0.91]). In addition, the multi-layers neural networks achieved a mean accuracy of 0.81 (95% CI [0.77, 0.85]) and a mean sensitivity of 0.85 (95% CI [0.77, 0.92]). Figure 6:5 shows histogram plots of the Monte Carlo simulations for single and multi-layer neural networks with post-processing and performances evaluated with the k-fold cross-testing method [17]. The results indicate that the difference between single and multi-layer neural networks is not significant regarding the four performances. Overall, we remark on a good predictive power and stability of these models.

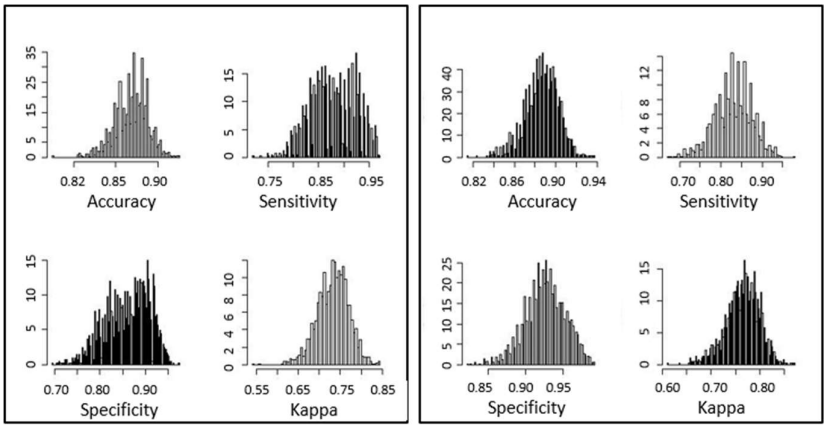
Model	Accuracy	Kappa	Sensitivity	Specificity
Single-layers neural network	0.80	0.59	0.84	0.74
Multi-layers neural networks	0.81	0.60	0.85	0.75
Deep networks	0.89	0.76	0.83	0.93

Table 6:1 Estimations of the predictive models' performances.



(a) Multi-layers neural networks (b) Single layer neural networks

Figure 6:4 2000 Monte Carlo simulation for neural networks.



(a) Deep network without post-processing (b) Deep network with post-processing

Figure 6:5 2000 Monte Carlo simulation for deep networks.

6.3.1 Attributes' predictive power with respect to neural networks models and the t-test, and with the ROC approach

In this subsection, we evaluate the predictive power of the attributes (including cannabis-related predictors) with respect to the first-episode psychosis.

6.3.1.1 *Student's t-test*

We now use the t-test to evaluate the predicting effect of the cannabis-related attributes on first episode psychosis with respect to the neural networks models developed in this chapter. Concretely, we show a statistically significant difference between the performances of the predictive models built with and without the cannabis variables.

As such, our analysis showed that the models' accuracy decreased by 5% for single-layer neural network and by 6% for the multi-neural networks and deep learning, if the cannabis-related attributes were removed from the process of building the predictive models. Then, we compared the accuracies of the single-layer neural network models built on the datasets with and without the cannabis-related attributes using the one-tailed t-test. The p-value obtained for the t-test was7.1×10^{-203}. As for the multi-layer neural networks models built on the datasets with and without the cannabis use attributes, the p-value obtained for the one-tailed t-test was2.1×10^{-37}. Finally, the p-value with a value of 4.2×10^{-17} was obtained for the one-tailed t-test when deep learning models were applied with and without the cannabis attributes.

This means that the predictive models with cannabis attributes have higher predictive accuracy than the models that were built without the cannabis attributes. In other words, the additional cannabis variables jointly account for predictive information over first-episode psychosis also based on the neural network models. These results are consistent with results of the analyses presented in the previous chapters based on overall distinct methodologies and learning algorithms.

6.3.1.2 *Ranking attributes' importance with the ROC curve approach*

For a further understanding of which variables affect first-episode psychosis, we conducted an analysis using the ROC curve approach [2]. We measured the individual importance of all attributes in the dataset to discover those that yield significant improvements in the model predictive power. To do so, the ROC curve is considered in relation to each attribute. Then, a series of cut-offs is applied to the data to predict the class. The sensitivity and specificity are calculated for each cut-off, and the ROC curve is computed. Finally, the area under the curve is used as a measure of variable importance. Table 6:2 shows the top 10 attributes ranked by the ROC curve approach.

Attribute	Importance
riskcan0.1	100.00
TotCANTYPE2.1	90.38
type_use.hash	88.53
totfreq2.1	87.74
duration.3	85.84
Education. University_professional qualifications	85.55
Totfreq2.2	84.24
bullying.no	78.55
white	72.89
homeless.1	72.86

Table 6:2 ROC curve attribute importance.

The results in Table 6:2 support prior evidence that cannabis attributes, such as the type of the cannabis used and the frequency of usage, have significant power in predicting first-episode psychosis. For example, the results in Table 6:2 support findings from [15] by associating the type of cannabis, especially high-potency cannabis, with the onset of psychosis. In addition, duration.3 in Table 6:2, which represents the duration of cannabis use, is consistent with findings from chapter 4.

6.4 Conclusion

The aim of this chapter has been to propose a novel machine learning approach to developing predictive models for the onset of first-episode psychosis with neural net-

works and deep learning. To our knowledge, previous studies on the link between cannabis use and first-episode psychosis investigated this highly important relationship via conventional statistical methodologies and techniques and did not tackle the predictability of this condition in relation to cannabis use. An exception is constituted by our contributions in chapter 4 and chapter 5, which are the first studies to predict first episode-psychosis using machine learning. They are based on support vector machines, bagged trees, boosted classification trees, eXtreme gradient boosting, and random forests. However, the accuracies in chapter 4 and chapter 5 were around 80%, and as such, less than all neural network and deep network models' performances achieved in this chapter.

In this chapter, we successfully classified first-episode psychosis from normal control with 89% accuracy (the highest performance) using deep learning. This solution proves the high potential in psychiatry of the applicability of machine learning, and enables researchers and medical doctors to evaluate the risk of and to predict first-episode psychosis, with its potential impacts on allocating medical attention and treatment more efficiently in an optimised way.

Our approach features a gradual control of the limitation of the uncertainty present in the data by investigating a trade-off between the extent of missing values entailing uncertainty, and the dataset size. Moreover, due to expected potential variations of the predictive models' performances due to the uncertainties resulting from the remaining missing values in the data, we conducted extensive Monte Carlo simulations to study these variations and the stability of the models.

Our methodology included novel contributions not only in the pre-processing and model optimisation phases, but also in model post-processing with ROC optimisation using three methods for finding the best probability cut-off, which, on one hand increase model performance, and on the other hand lead to balancing the sensitivity and specificity. The latter constitutes an issue usually when datasets are unbalanced. This methodology also incorporated our k-fold cross testing solution, a novel method we presented and employed in chapter 5.

Chapter 7 PIDT: A novel decision tree algorithm based on parameterised impurities and statistical pruning approaches

In the process of constructing a decision tree, the criteria for selecting the splitting attributes influence the performance of the model produced by the decision tree algorithm. The most well-known criteria, such as Shannon entropy and Gini index, suffer from the lack of adaptability to the datasets. This chapter presents novel splitting attribute selection criteria based on some families of parameterised impurities that we proposed here to be used in the construction of optimal decision trees. These criteria rely on families of strict concave functions that define the new generalised parameterised impurity measures that we applied in devising and implementing our PIDT novel decision tree algorithm. This chapter also proposes the S-condition based on statistical permutation tests, whose purpose is to ensure that the reduction in impurity, or gain, for the selected attribute is statistically significant. We implemented the S-pruning procedure based on the S-condition, to prevent model overfitting. These methods were evaluated on a number of simulated and benchmark datasets. Experimental results suggest that by tuning the parameters of the impurity measures and by using our S-pruning method, we obtain better decision tree classifiers with the PIDT algorithm.

7.1 Problem description

The decision tree algorithm is a highly efficient algorithm used in machine learning and data mining; the model produced by the algorithm is easy to understand and interpret, and it offers accurate results in abbreviated time. Different versions of the decision tree algorithm have been introduced in the last few decades, and it remains an attractive research domain within the field of machine learning. Such algorithms are useful in numerous contexts within pattern recognition and machine learning applications. In the medical field, for instance, decision trees have been employed to diagnose heart disease patients [151] and to predict patients who may suffer from psychosis as in chapter 3.

A decision tree algorithm simulates a tree assembly [1]. A decision tree consists of nodes that are connected via branches. The decision tree begins with a single root node and ends with a number of leaf/decision nodes; the nodes in between are the internal nodes.

In classification trees, each leaf node is labelled with a particular class. Each node that is not a leaf node applies a test on a particular attribute, and each branch represents a result of the test. The nodes are selected from the top level based on the attribute-selection measure [71]. For example, ID3 algorithm [51] and its extended version C4.5 [27] use information gain (which is based on Shannon entropy) to construct the decision tree; the element with the highest gain is taken as the root node, and the dataset is divided based on the root element values. Again, the information gain is calculated for all the internal nodes separately, and the process is repeated until leaf nodes are reached.

Unlike most machine learning algorithms, decision trees perform local feature selection on different sets of features. The selected feature should be the feature that shows the largest reduction in the uncertainty at the node [79]. The dataset may then be partitioned accordingly into sub-nodes. This procedure is applied recursively until it meets any stopping criterion, such as the minimum number of instances or the maximum tree depth. Choosing the splitting and stopping criteria are two open problems in decision tree algorithms.

To address the first issue, many decision tree algorithms have proposed different impurity measures as a splitting criterion. Most decision tree algorithms are based on the information gain function for choosing the best attribute for splitting the data at each node that is not a leaf node. For instance, the ID3 and C4.5 algorithms are based on Shannon

entropy [79], while the classification and regression tree CART algorithm is based on the Gini index [26]. However, one drawback of this kind of approach is that these types of impurity measures are based on only one fixed concave function for assessing the impurity in the datasets' class distributions, which means they suffer from a lack of adaptability to various datasets.

Many studies have investigated the importance of the split criterion [152] [153]. These studies have concluded that the choice of impurity measure does have some influence on the decision tree's efficacy. Inspired by these studies, we have proposed several novel splitting criteria based on parameterised families of strict concave functions that may be used as impurity measures. As such, we propose new parameterised impurities including parameterised entropy (PE), parameterised Gini (PG), parameterised Tsallis (PT), parameterised Renyi (PR), as well as parameterised AlphaBeta impurity (ABI) and parameterised GiniEntropy (GE) impurity. Their purpose will consist of being mostly reduced in a node after a split, which will dictate the choice of the most suitable attribute in that node. These methods indeed provide an innovative approach to improved decision tree performance, as this chapter shows.

As for the second problem, most practical decision tree implementations use a 'greedy' approach to grow the tree. Such algorithms would usually suffer from overfitting the dataset [79], and additional mechanisms are needed to prevent this. Several stopping criteria have been introduced to overcome this issue, such as setting the minimum value of the information gain to grow the tree with a C4.5 algorithm [71]. A number of recent papers have used permutation tests for different machine learning problems, such as studying the classifier performance [154], or in the feature selection process [155]. With the model overfitting problem in mind, in this chapter we proposed the S-condition based on statistical permutation tests, whose purpose is to ensure that the reduction in impurity, or gain, for the selected attribute in a node of the decision tree is statistically significant, and that the observed gain is unlikely to be at least that high just by chance. Moreover, we implemented the S-pruning procedure based on the S-condition to prevent model overfitting.

We integrate the use of our novel families of parameterised impurities for the attribute selection with the S-pruning procedure, and with the optimisation of the parameters of the impurity via cross-validation according to the accuracy performance, in a new decision tree algorithm that we call PIDT (parameterised impurity decision tree).

The remainder of this chapter is organised as follows. Section 2 introduces the mathematical formulations and the general requirements for the impurity measures, as well as the novel parameterised impurity measures that we propose for use in selecting the splitting attributes in our PIDT algorithm. Section 3 introduces our S-condition and S-pruning procedure based on permutation tests, which enhance the PIDT algorithm to prevent model overfitting. Section 4 experimentally investigates the proposed parameterised impurity measures and compares them with conventional impurity functions, based on the performances obtained by the PIDT and conventional decision tree algorithms on a number of benchmarks and generated datasets. Finally, Section 5 presents conclusions and offers directions for future work.

7.2 Impurity measures

As mentioned above, a decision tree algorithm splits the dataset sample (at each node that is not a leaf node) into two or more sets based on the attribute that scores the highest gain (i.e. reduction in impurity) [156]. In the previous section, we mentioned two conventional impurities mostly used in decision tree algorithms, namely Shannon entropy and Gini index. However, there are also other impurities that are presented in the literature such as Tsallis [157], and Renyi [156]. A different work also proposed a generalisation of conditional entropy [158]. Considering these different studies based on a generalisation of conditional entropy, various impurity measures suggest that the choice of the impurity measure influences the decision tree's effectiveness. In the following sub-sections, we provide the mathematical formulations of and the criteria for functions defined on discrete probabilistic distributions, to be impurity measures.

7.2.1 Mathematical formulations

Let X be an n × m data matrix. We denote the r-th row vector of x by x_r, and the c-th column vector of x by x_c. Rows are also called records or data points, while columns are also called attributes or features. Since we do not restrict the data domain of X, the scale

of this domain's features can be categorical or numerical. For each data point x_r, we have a class label y_r. We assume a set of known class labels Y, so $y_r \in$ Y. Let D be the set of labelled data $D = \{(X_r, y_r)\}_{r=1}^{n}$. During the classification task, the goal is to predict the labels of new data points by training a classifier on D. Now, let k be the total number of data entries in a node, and ki be the number of data entries classified as class i. Then $p_i = {k_i}/{k}$ is the ratio of instances classified as i and estimates the probability of class i in the dataset in that node.

The primary purpose of the impurity measures is to express the degree of mixture of various classes in a dataset and then to help to define how well the classes are separated via a split in a node. As such, in general, an impurity measure should satisfy specific requirements. Breiman et al. [26] suggested that an impurity measure is a function *Imp* whose argument is a vector of probabilities from a discrete probability distribution (given by the class proportions in a dataset), which satisfies the following properties:

Property A: Strict concavity $Imp'' < 0$.

Property B: Maximality $Imp' = 0\ for\ \left(p_i = {1}/{k}\right) for\ i\ = 1, \ldots, k.$

Property C: Minimality $Imp = 0 \leftrightarrow \exists i\ | pi\ = 1.$

These properties state that the impurity function should be a strictly concave function; they also express details of the maximum and minimum points of the function. Both Shannon entropy and Gini index, which are defined below, meet the impurity-based criteria:

$$Entropy\ (D) = E(D)\ = -\sum_{i=1}^{k} p_i * log(p_i)$$

$$Gini\ (D) = G(D)\ = 1 - \sum_{i=1}^{k} p_i^2$$

Several authors compared the behaviour of Gini index and Shannon entropy to determine which performs better; they concluded that it is not possible to decide which one leads to higher accuracies of the produced decision trees since the two measures have only about 2% disagreement in most cases [154]. Note that both Gini index and Shannon entropy are based on one strict concave function each, and as such they might not have

the flexibility in adapting to various datasets. We have also considered Renyi entropy and Tsallis entropy, both of which generalise Shannon entropy. They are described by the following formulas, respectively:

$$Renyi\ (D)\ =\ R(D)\ = \frac{1}{1-\gamma} * log(\sum_{i=1}^{k} p_i^{\gamma}) \qquad where\ \gamma\ >\ 0\ and\ \gamma\ \neq\ 1$$

$$Tsallis\ (D) =\ T(D) = \frac{1-\sum_{i=1}^{k} p_i^{\gamma}}{1-\gamma} \qquad where\ \gamma\ >\ 0\ and\ \gamma\ \neq\ 1$$

In the next subsection, we propose several families of generalised parameterised impurity measures based on the requirements suggested by Breiman et al. [26] and outlined above, and we introduce our new PIDT algorithm employing these impurities.

7.2.2 Parameterised impurity measures

As mentioned previously, the novel parameterised impurity measures (proposed below) are used to select the attribute that has the most effect on reducing the impurity by splitting the dataset in a node of the decision tree.

Our first proposed family of parameterised impurities is the parameterised entropy PE, which is formulated below, and is illustrated in Figure 6:1 for the case of the 2 class problems (the x-axis represents the probability of one class).

$$\boldsymbol{PE\ (D) =\ E(D)^{\alpha}} \qquad \boldsymbol{where\ \alpha\ \in\ (0, 1]}$$

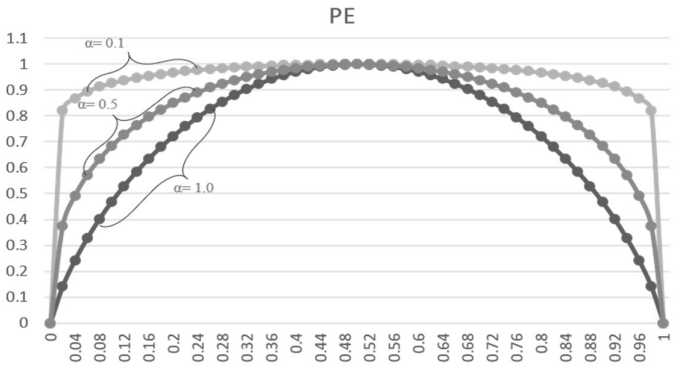


Figure 7:1 Parameterised entropy (PE) with different values for α.

The interval of variation for the parameter α, i.e. (0, 1], was chosen to allow, on the one hand, a large diversity of shapes of the graph of the impurity PE, and on the other hand, to mathematically ensure the concavity of the impurity. The other requirements inspired by Breiman's work [26], to which we referred in the previous subsection, are also met.

Figure 7:1 illustrates the impact of α on the shape of the PE curve. In particular, α =1 corresponds to the conventional Shannon entropy, while smaller positive values for α have an effect of diminishing the curvature of the PE curve around its middle section (the second derivative's absolute value tends to decrease in that area), and of gradually transforming the curve and make it tends to a plateau for small values of the parameter (for illustration see the curve for α = 0.1 in Figure 7:1). Intuitively, these changes in the shape of the PE curve suggest potential changes in choosing attributes in a split node of the decision tree, and this was confirmed experimentally when we implemented our framework. This situation happens because the process may give preference to different class probability distributions in the data subsets that are issued from the split. Parameter α clearly influences that splits will be created in the decision tree, and as such it influences the model learnt from the data, and allowed it to have more flexibility in adapting to the data than in the case of a fixed impurity such as the conventional Shannon entropy.

In the same manner, parameterised Gini, parameterised Renyi, and parameterised Tsallis are defined by using the following formulas:

$$\boldsymbol{PG}\ (\boldsymbol{D}) = \boldsymbol{G}(\boldsymbol{D})^{\boldsymbol{\alpha}} \qquad \boldsymbol{where}\ \boldsymbol{\alpha} \in (\mathbf{0}, \mathbf{1}]$$

$$\boldsymbol{PR}\ (\boldsymbol{D}) = \boldsymbol{R}(\boldsymbol{D})^{\boldsymbol{\alpha}} \qquad \boldsymbol{where}\ \boldsymbol{\alpha} \in (\mathbf{0}, \mathbf{1}]$$

$$\boldsymbol{PT}\ (\boldsymbol{D}) = \boldsymbol{T}(\boldsymbol{D})^{\boldsymbol{\alpha}} \qquad \boldsymbol{where}\ \boldsymbol{\alpha} \in (\mathbf{0}, \mathbf{1}]$$

Note that since the concave functions that define the conventional Shannon entropy and Gini index are generalised by the proposed families of parameterised impurities PE and PG, respectively, the use of these families of impurities is expected, roughly speaking, to produce comparable or better decision trees in most cases than those based on the conventional entropy and Gini index.

We now define two more families of parameterised impurities based on two parameters α and β.

$$\boldsymbol{GE}\ (\boldsymbol{D}) = \boldsymbol{G}(\boldsymbol{D})^{\boldsymbol{\alpha}} + \boldsymbol{E}(\boldsymbol{D})^{\boldsymbol{\beta}} \qquad \boldsymbol{where}\ \boldsymbol{\alpha}\ \boldsymbol{and}\ \boldsymbol{\beta} \in (\mathbf{0}, \mathbf{1}]$$

$$\boldsymbol{ABI}\ (\boldsymbol{D}) = \sum_{\boldsymbol{i}=\mathbf{1}}^{\boldsymbol{k}} \boldsymbol{p}_{\boldsymbol{i}}^{\propto} * (\mathbf{1} - \boldsymbol{p}_{\boldsymbol{i}})^{\boldsymbol{\beta}} \qquad \boldsymbol{where}\ \boldsymbol{\alpha}\ \boldsymbol{and}\ \boldsymbol{\beta} \in (\mathbf{0}, \mathbf{1}]$$

Note that GE combines arbitrary positive and not larger than 1 powers of the Gini index and of the conventional Shannon entropy, generalising these impurities, and offering further flexibility by using two parameters. Through the use of the two parameters, ABI family generalises the Gini index and also offers further flexibility in expressing various shapes of impurity. Note also that both GE and ABI fulfil, mathematically speaking, the requirements of impurity inspired by Breiman et al. [26].

Figure 7:2 illustrates, for the case of 2 class problems, the parameterised families of impurities PE and PG for various values of parameter α (see the top half), and the parameterised family of impurities GE for various values of parameters α and β (see the bottom half).

The above-parameterised impurity families are used in our novel decision tree algorithm, PIDT. In particular, the impurities define the criterion for selecting the best attributes in the nodes of the decision tree based on the largest decrease in impurity, from the dataset in the parent node to the datasets in the child nodes. This difference is the so-called gain, and are defined precisely in the next section when the statistical S-condition will be introduced. The PIDT algorithm uses one single selected family of parameterised impurities for a tree induction, and optimises the parameters of the impurity in a cross-validation fashion with respect to the accuracy performance.

In the next section, we develop an enhancement of the process of growing the decision tree with the PIDT algorithm and based on a novel statistical pruning procedure S-pruning that we introduce here as a useful tool to prevent overfitting problems.

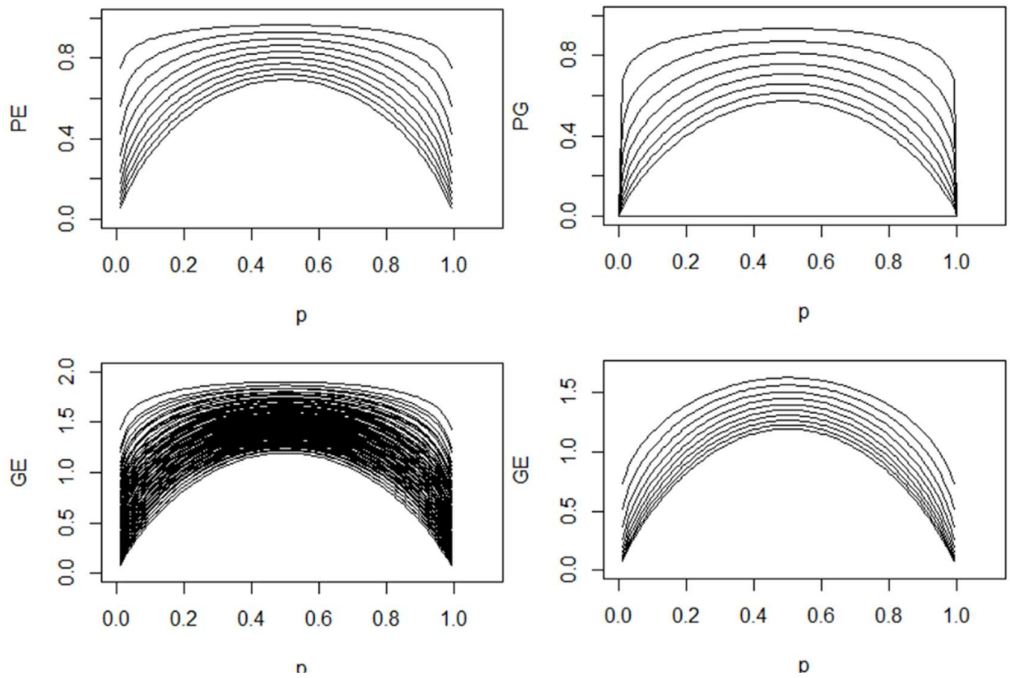


Figure 7:2 Novel parameterised impurity measures PE, PG (top), and GE (bottom).

7.3 S-pruning

Roughly speaking, the novel S-pruning procedure we describe here terminates some of the branches of the decision tree based on the outcome of a statistical test. In particular, this pruning method only allows the attributes that have a significant predictive power to split the node and grow the tree. Stopping the development of a branch is based on a certain condition, named here the S-condition.

7.3.1 S-condition

Let X_c be the attribute with the highest gain G in a node N. Roughly speaking, G is expressed by the reduction in impurity after the split with attribute X_c in the node N. More precisely, the gain is defined in the same way as the information gain for the conventional Shannon entropy in C4.5 algorithm [27]. The impurity is measured in the dataset before the split, and in the resulting data subsets for each child after the split. The impurities in all these data subsets are averaged with weights derived as the fractions represented by the data subsets out of the dataset before the split. The impurity weighted average is then

subtracted from the impurity of the dataset before the split, and the result defines the gain G mentioned above. The gain is non-negative for all attributes due to the concavity property of the impurity. Moreover, a higher gain may indicate a higher predictive power for an attribute. However, we want to ensure that a higher gain does not occur by chance. The S-condition defined here is a statistical mechanism to check this.

Let D be the dataset in node N. Then shuffle (i.e. randomly permute) the labels in dataset D and measure again the gain for X_c. Do this t times so that a vector V of t gain values is built. The S- condition is satisfied if and only if G is smaller than the q quantile of vector V. When the S-condition is satisfied, the branch in node N stops growing and N becomes a terminal node. This defines the S-pruning procedures.

Overall, the logic behind the S-condition is that if the gain G is smaller than the q quantile (for instance for a value q such as 0.95 or 0.9) of a vector V of t gain values (for instance t = 1000) obtained for X_c using random labels (since they are shuffled or randomly permuted), then X_c is not considered to have predictive power according to the data D in that node N. The values of t and s = 1-q must be specified by the user, where t is the number of label permutations (and thus equal to the number of gain values collected), and the value of s is the significance level (such as in the statistical tests). A smaller s will encourage more pruning. Intuitively, s indicates how likely it is that the gain of the selected attribute X_c would have been acceptably high just by chance. Another relevant quantity here is the p-value, defined experimentally as the fraction of cases in which the gain obtained with the random labels was higher than or equal to the gain obtained with the original labels of the records in D. Therefore, if the p-value is small enough (e.g., the p-value is smaller than or equal to the significance level s = 0.1 or 0.05), then we can say that the gain of the selected attribute in the original data is indeed significantly better and, in consequence, that the gain is too high to have occurred just by chance. That is, the null hypothesis of the permutation test is rejected in this case. As such, attribute X_c is considered to have significant predictive power, and the split takes place. Note that the S-condition does not hold in this case.

On the other hand, if the p-value is larger than the significance level s, or in other words, the S-condition holds, this means that the gain for the selected attribute is not large enough to indicate predictive power, so the development of that branch is stopped.

Note also that higher q (or equivalently smaller s) results in oversimplified trees, whereas the opposite results in reduced pruning and larger trees. Because of using the S-pruning procedure, fewer nodes are expanded during the building phase, and thus constructing the decision tree is simplified. In addition, the decision tree has the advantage of avoiding overfitting while it is being built.

7.4 Comparison of decision tree classifiers with various impurity measures

We now compare several impurity measures with respect to their impact on the decision tree induction, including the conventional impurities such as Shannon entropy and Gini index, and the new parameterised families of impurities introduced here. We argue that the conventional impurities mentioned above have their flexibility limitations when used with various datasets. We also argue that, due to their flexibility, the parameterised families of impurities are better suited to the purpose of class separation. We test our novel S-pruning procedure introduced in the previous section. Finally, we demonstrate empirically that the proposed PIDT algorithm indeed produces better decision trees than the algorithms that use merely the conventional entropy and Gini index impurity measures.

This section also investigates the performance of decision trees as a result of parameter optimisation. To investigate the usefulness of the novel parameterised impurity functions, we tested them on different datasets and compared them with the conventional impurities mentioned above. To optimise the parameters of an impurity family, a grid search over a parameter space with 5-fold cross-validation was used to select the values of the best parameters.

7.4.1 Predicting first-episode psychosis with the PIDT algorithm

We chose the open-source library Weka (Waikato Environment For Knowledge Analysis) version 3.6.15 (December 2015) [71] as a starting point in implementing our PIDT algorithm with the S-pruning method option and parameter optimisation for the families of parameterised impurities above. In particular, the tree builder code was modified and extended to support the conventional impurities Shannon entropy and Gini index, as well

as Tsallis and Renyi, and we implemented the new families of parameterised impurity measures introduced in this chapter. The S-pruning method was also added. The PIDT software allows users to specify the family of impurities and values for their relevant parameters, or to choose the optimisation of these parameters. It allows specifying the significance level s and the number of permutations t when the S-pruning method is enabled.

The data used to develop our novel approach to predicting first episode psychosis is part of a case-control study at the inpatient units of the South London and Maudsley (SLaM) NHS Foundation Trust [15]. In particular, 5-fold cross-validation and was performed with and the S-pruning method. On the training set, SMOTE were applied to treat the unbalanced classes in the data. In addition, random forest imputations were performed on the training data prior to fitting a model with its corresponding optimal parameters.

The PIDT algorithm was run for different impurity measures and values for α, β, γ parameters (whichever apply), and significance level s. The parameter space for α and β was 0.05, 0.1,..., 0.95, 1.0; for γ the values were 0.1, 0.2,…, 0.9, 1.5, 2.0,…, 5.0; and the considered significance level s values were 0.01, 0.05, 0.1, 0.15 and 0.2.

Finally, the best-performing models with their parameters were chosen for the final comparison on the separate test datasets. The chosen optimised parameters were PE as an impurity function with 0.2 for both parameters α and significance level s. The best results scored regarding accuracy, kappa, sensitivity and specificity are 0.817, 0.629, 0.817 and 0.82 respectively. The results demonstrate that the novel PIDT algorithm could be used to construct more efficient predictive models for the first episode psychosis compared with the predictive one built in chapter 4.

Although the obtained results were weaker than all neural network and deep network models scored in chapter 6, PIDT still provides powerful prediction models with several advantages that neural networks lack. One advantage is the knowledge representation of decision trees such as PIDT is easy to interpret and explain to experts, which is not the case for neural networks. Therefore, the decision-making process itself can be easily validated by an expert. Another advantage of PIDT that it implicitly performs feature selection and require relatively little effort from users for data preparation compared to neural networks. Finally, PIDT algorithm learns very fast and neural networks learn

relatively slow. Because of these reasons, building PIDT is especially appropriate to support the decision-making process in medical prediction such as predicting first-episode psychosis.

7.4.2 Additional experimental analysis

In this section, more experiments were conducted with PIDT on five real datasets as well as on two simulated datasets. PIDT algorithm was applied to seven datasets, of which five were real public datasets and two were simulated datasets with different characteristics.

The real datasets from the University of California–Irvine (UCI) machine learning repository [25] that were provided to illustrate the performance of different impurity measures included the diagnostic Wisconsin breast cancer dataset, the diabetes dataset, the glass identification dataset, and a medical dataset for hepatitis and primary tumours [159]. Two datasets were also generated using simulation techniques in particular, based on Guyon's proposed approach employed in various researches [159] [160] [161] [162] [163]. The simulated datasets contain a few thousand samples and different numbers of classes.

The PIDT algorithm was run for different impurity measures and values for α, β, γ parameters (whichever apply), and significance level s. The parameter space for α and β was 0.05, 0.1,..., 0.95, 1.0; for γ the values were 0.1, 0.2, …, 0.9, 1.5, 2.0,…, 5.0; and the considered significance level s values were 0.01, 0.05, and 0.1. Finally, the best-performing models with their parameters were chosen for the final comparison on the separate test datasets. Table 7:2 shows a summary of the models built with the chosen optimised parameters, while Table 6:1 provides the summary of the models built by using conventional impurities. Bold fonts in Table 7:2 show the best results scored regarding the chosen dataset. The results demonstrate that the parameterised entropy (PE) could be used to construct more efficient decision trees compared with the conventional entropy impurity and Gini index impurity. In particular, PE led to better results when it was applied to the S-pruning method on most datasets. By looking at Table 7:1 and Table 7:2, we observe that the accuracy generally improved, and the number of nodes decreased for the models produced by the PIDT algorithm.

Dataset	Decision tree with entropy		Decision tree with Gini	
	Accuracy	No. nodes	Accuracy	No. nodes
Breast cancer	0.654	67	0.654	76
Pima diabetes	0.736	119	0.724	135
Hepatitis	0.807	21	0.794	25
Primary tumour	0.434	60	0.363	57
Glass	0.626	39	0.556	55
Simulated data 1	0.721	33	0.668	67
Simulated data 2	0.612	188	0.601	157

Table 7:1 Assessing decision trees built with conventional impurity performances.

In particular, it is interesting to observe that the accuracy tended to improve depending on the dataset, thus confirming that this performance could be affected by the method used for selecting attributes during the tree construction. Regarding tree size, this was diminished for most datasets. The best reduction was achieved for the Pima diabetes database, where the size of the tree was reduced ten times compared to the standard tree algorithm – which used entropy (as shown in Table 7:1) – and was comparable to the tree size discussed in [158]. Although the decision tree approach may not provide considerable performance improvements against neural network-based approaches, its value in terms of presenting knowledge and providing insight to medical practitioners is an important, high-value attribute in computational psychiatry context. Figures 7:3 and 7:4 show example of pruned and unpruned decision tree for pima diabetes. We also note that our results for the hepatitis dataset produced more accurate results and a smaller tree compared with the results presented in [158]. Overall, PE and PR impurities, in conjunction with activating the S-pruning procedure, produce more accurate results and yield much smaller trees for most of the datasets as shown in Figures 7:5 and 7:6 that gives glass example of pruned and unpruned decision tree for dataset.

Dataset	PIDT								
	Accuracy	No. nodes	Parameters						
			Impurity	α	β	γ	S-pruning	*s*	Permuta-tions
Breast cancer	**0.731**	91	PG	0.5	-	-	no	-	-
	0.720	**29**	PR	1	-	0.5	yes	0.05	1000
Pima diabetes	0.734	**11**	PE	0.5	-	-	yes	0.05	1000
Hepatitis	**0.839**	23	PE	0.3	-	-	no	-	-
	0.807	**7**	PE	0.3	-	-	yes	0.05	1000
Primary tumour	0.434	60	PE	1	-	-	no	-	-
Glass	**0.636**	**27**	PE	0.6	-	-	no	-	-
Simulated data 1	0.721	**7**	PE	0.8	0	0	yes	0.05	1000
Simulated data 2	**0.693**	**157**	GE	1	0.4	-	no	-	-

Table 7:2 Assessing decision trees built with the PIDT algorithm with parameter optimisation, and with and without S-pruning procedure activated. "-" mean values do not apply.

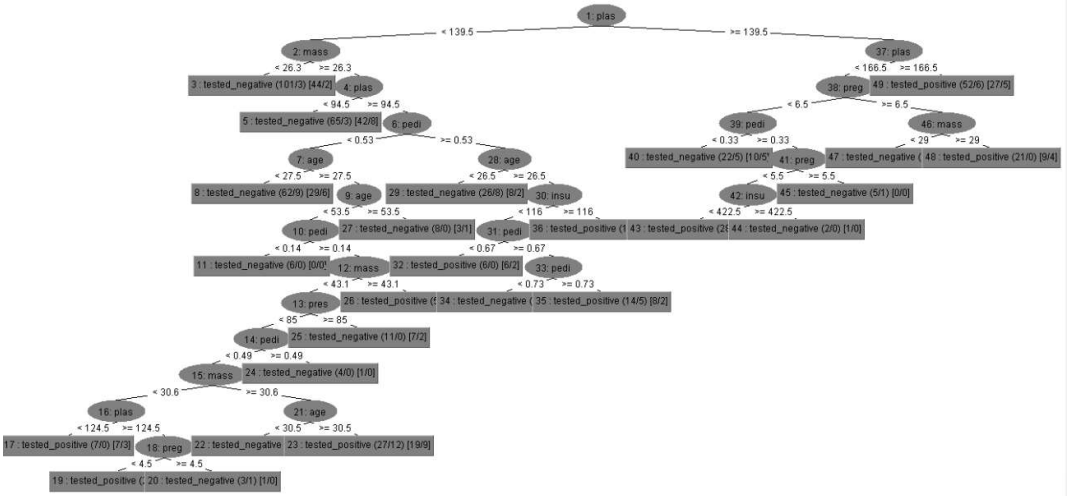


Figure 7:3 Unpruned decision tree for Pima diabetes

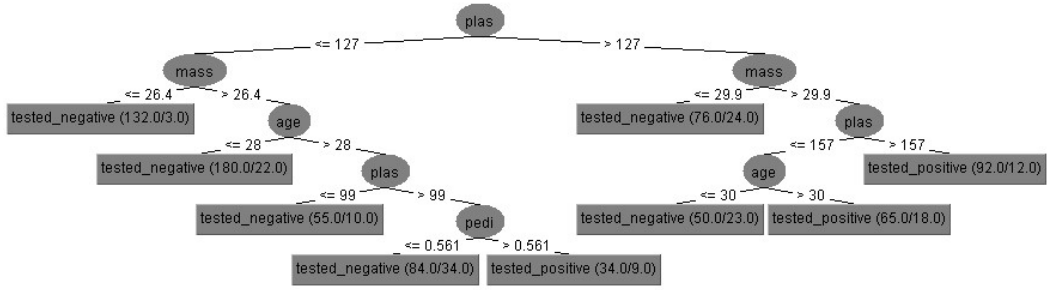


Figure 7:4 Pruned decision tree for Pima diabetes

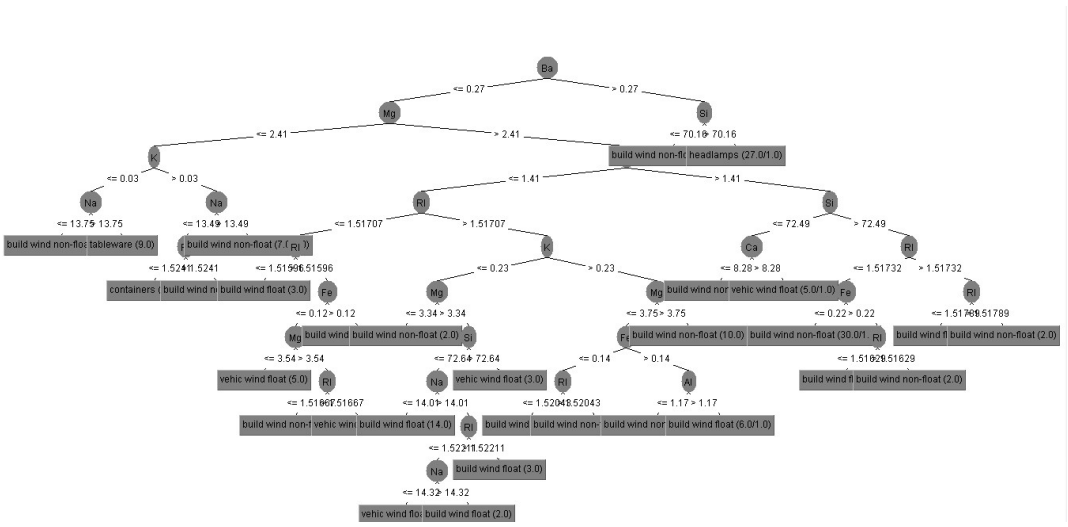


Figure 7:5 Unpruned decision tree for glass dataset

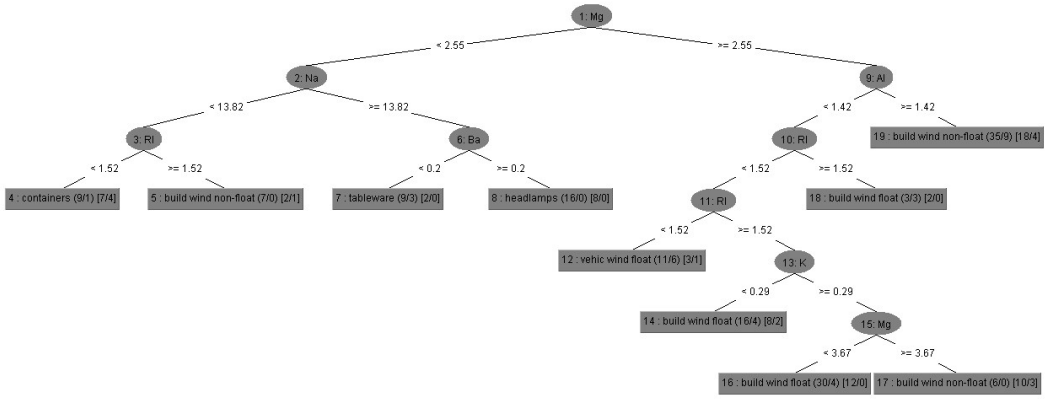


Figure 7:6 Pruned decision tree for glass dataset

7.5 Conclusion

The aim of this chapter has been to propose a novel decision tree algorithm based on parameterised impurities and statistical pruning approaches (PIDT). The novel contributions of this chapter are the following. First, it presented novel splitting attribute selection criteria based on families of parameterised impurities. These criteria rely on families of strict concave functions that define the new generalised parameterised impurity measures that we applied in devising and implementing our PIDT novel decision tree algorithm. Then, the chapter proposed the S-condition based on statistical permutation tests, whose purpose is to ensure that the reduction in impurity, or increase in gain, for the selected

attribute is statistically significant. We implemented the pruning procedure based on the S-condition, to prevent model overfitting.

This chapter proposed and tested the innovative approach to building optimised classification trees using novel parameterised impurity measures that generalise conventional impurities such as Shannon entropy and Gini index. The experiments were conducted on five real datasets as well as on two simulated datasets. The results show that by building decision trees using parameterised impurity measures with optimal values for their parameters, the predictive models primarily led to better performance in terms of accuracy than those built with traditional entropy impurity and Gini impurity. Forevermore, as an application for our research, we have considered using decision trees with the novel PIDT algorithm to build predictive models for first episode psychosis. The results demonstrate that the parameterised entropy (PE) could be used to construct more efficient predictive models compared with the predictive models built in chapter 3. However, PIDT may not provide considerable performance improvements against neural network-based approaches, its value in terms of presenting knowledge and providing insight to medical practitioners is an important, high-value attribute in computational psychiatry context

The novel S-pruning method based on permutation tests was also tested here to overcome the overfitting problem and to produce smaller decision trees. The proposed impurity measures gained significance and produced much smaller trees when they were applied with the S-pruning procedure enabled. However, if the significance level s for S-pruning is set too small, it may result in oversimplified trees, which do not fit data well enough. The idea is that the significance level for the S-pruning needs adjustments to the data and the problem at hand.

Ongoing research regards the extension of our PIDT algorithm to ensemble-based techniques such as novel algorithms for random forests and boosting with decision trees built with the new families of parameterised impurities and statistical pruning, which will incorporate other enhancements currently under investigation.

Future applications of the new algorithms concern the prediction of dementia risk, a topic that has received considerable interest recently due to limited existing research and the immense potential in the prevention and reduction of huge medical and social expenditures worldwide. Such research developments are currently taking place in the research lab in which this dissertation work was produced.

Chapter 8 Conclusion and directions for future work

8.1 Conclusion

The objective of this thesis is to propose novel predictive modelling approaches to data-driven computational psychiatry. In particular, this document advances research in medical data mining via two phases. In the first phase, this dissertation advances research in data mining, mainly medical data mining, by proposing a novel prediction modelling and pattern detection approaches for the first-episode psychosis associated with cannabis use. This phase is built upon several machine learning techniques whose predictive models have been trained, optimised, and tested in a computationally intensive framework. They exhibited a good predictive power and stability based on an acceptable level of variation of their performance measures evaluated across extensive experiments encapsulated in a series of large-scale Monte Carlo simulations. Moreover, the link between cannabis-related attributes and first-episode psychosis were investigated via association analysis and Bayesian inference-based techniques.

This phase also proposes a novel machine learning approach to developing predictive models for the onset of first-episode psychosis using artificial neural networks and deep networks. Our approach features a gradual control of the limitation of the uncertainty present in the data by investigating a trade-off between the extent of missing values entailing uncertainty, and the dataset size.

Moreover, several sampling methods and several methods for choosing the optimal cutting point on the ROC curve to improve and evaluate the prediction models' performances were proposed in novel methodologies, which reflect our contribution to the state of the art methods in predictive modelling on the one hand and to interdisciplinary computational psychiatry research on the other.

As our best results, we successfully classified first-episode psychosis from standard control with 89% accuracy using state of the art methods in classification based on deep learning. This solution proves the high potential in psychiatry of the applicability of machine learning, in particular of deep learning, and enables researchers and medical doctors to evaluate the risk of and to predict first-episode psychosis, with great potential

towards developing personalised medicine in this direction, and in optimising medical attention and treatment.

In the second phase, this dissertation advances research in data mining by proposing several novel extensions in the area of data classification by proposing innovative parameterised impurity measures toward building more accurate decision trees classifiers with potential in developing new ensemble-based classification algorithms. The objective of this second phase is to propose new machine learning algorithms that are particularly suitable for medical research, in particular novel and enhanced techniques that produce models with high explanatory power such as decision trees or enhanced decision tree ensemble based algorithms that are able to perform well with highly dimensional data such as genotype data with millions of features as SNPs, which are increasingly present in medical research. An approach to building optimised classification trees using novel parameterised and generalised impurity measures and statistical pruning methods, was proposed and tested here with very good results. In addition, the results demonstrate that our novel approach with the parameterised entropy and statistical pruning could successfully be used to construct efficient predictive models for the onset of first episode psychosis.

8.2 Future work

Ongoing research regards the extension of our PIDT algorithm to ensemble-based techniques such as novel algorithms on random forests and boosting with decision trees built with the new families of parameterised impurities and statistical pruning, which will incorporate other enhancements currently under investigation.

Future applications of the new algorithms concern the prediction of dementia risk. In the United Kingdom alone, there are currently almost 1 million people living with dementia. There is no cure yet, and the condition has higher health and social care costs than cancer, strokes, and chronic heart disease considered together. Recent estimates show that dementia expenditure in the UK is £26 billion per year. Current thinking also suggests that 35% of cases of dementia could be prevented if predicted in advance and doctors and their patients take informed action [164].

Such research developments in predicting dementia are currently taking place in the Data Science & Soft Computing Lab in which this dissertation work was produced.

We expect that the novel methodologies presented in this thesis as a contribution to the predictive modelling field, with applicability to first episode psychosis prediction, will also be extended, adapted and largely applied in this computational psychiatry research topic of large interest concerning the prediction of the risk of dementia.

REFERENCES

[1] J. Han, M. Kamber and J. Pei, Data Mining: Concepts and Techniques, Elsevier, 2011.

[2] M. Kuhn and K. Johnson, Applied Predictive Modeling, Springer, 2013.

[3] G. Paliouras, V. Karkaletsis and C. D. Spyropoulos, Machine Learning and Its Applications, Springer, 2001.

[4] Q. J. M. Huys, T. V. Maia and M. J. Frank, "Computational psychiatry as a bridge from neuroscience to clinical applications," *Nature Neuroscience,* vol. 19, no. 3, pp. 404-413, 2016.

[5] I. Raquel, D. Stahl and P. McGuffin, "Machine learning, statistical learning and the future of biological research in psychiatry," *Psychological Medicine,* vol. 46, no. 12, pp. 2455-2465, 2016.

[6] S. Huang, J. Loh, J. Tsai, M. Houg and H. Shi, "Predictive model for 5-year mortality after breast cancer surgery in Taiwan residents," *Chinese Journal of Cancer,* vol. 36, no. 23, 2017.

[7] H.-Y. Shi, W.-T. Hung, K.-T. Lee, S.-C. Wang, W.-H. Ho, S.-C. Chang, J.-J. Wang, D.-P. Sun, H.-H. Lee and C.-C. Chiu, "Artificial neural network model for predicting 5-year mortality after surgery for hepatocellular carcinoma and performance comparison with logistic regression model: a nationwide Taiwan database study," in *Third International Conference on Innovations in Bio-Inspired Computing and Applications*, Kaohsiung, Taiwan, 2012.

[8] J. Shi, S. Zhang, M. Tang, C. Ma, J. Zhao, T. Li, X. Liu and Y. Sun, "Mutation screening and association study of the neprilysin gene in sporadic Alzheimer's disease in Chinese persons," *The Journals of Gerontology Series A: Biological Sciences and Medical Sciences,* vol. 60, no. 3, pp. 301-306, 2005.

[9] E. Elveren and N. Yumuşak, "Tuberculosis disease diagnosis using artificial neural network trained with a genetic algorithm," *Journal of Medical Systems,* vol. 35, no. 3, pp. 329-332, 2011.

[10] S. Liu, S. Liu, W. Cai, S. Pujol, R. Kikinis and D. Feng, "Early diagnosis of Alzheimer's disease with deep learning," in *The 11th International Symposium on Biomedical Imaging*, Beijing, China, 2014.

[11] K. R. Foster, R. Koprowski and J. D. Skufca, "Machine learning, medical diagnosis, and biomedical engineering research - commentary," *BioMedical Engineering OnLine,* vol. 13, no. 94, 2014.

[12] S. Kapur, A. G. Phillips and T. R. Insel, "Why has it taken so long for biological psychiatry to develop clinical tests and what to do about it?," *Journal of Molecular Psychiatry,* vol. 17, no. 12, p. 1174–1179, 2012.

[13] K. Friston, K. Stephan and R. Montague, "Computational psychiatry: the brain as a phantastic organ," *Lancet Psychiatry,* vol. 1, no. 2, pp. 148-158, 2014.

[14] R. Adams, J. Quentin and R. Jonathan, "Computational Psychiatry: towards a mathematically informed understanding of mental illness," *Journal of Neurology, Neurosurgery & Psychiatry,* vol. 87, no. 1, pp. 53-63, 2016.

[15] M. DiForti, A. Marconi, E. Carra, S. Fraietta, A. Trotta, M. Bonomo, F. Bianconi, J. OConnor, M. Russo, S. Stilo, T. Marques, V. Mondelli, P. Dazzan, C. Pariante, A. David, F. Gaughran, Z. Atakan, C. Iyegbe, J. Powell, C. Morgan, M. Lynskey and R. Murray, "Proportion of patients in south London with first-episode psychosis attributable to use of high potency cannabis: a case-control study," *The Lancet Psychiatry,* pp. 233-238, 2015.

[16] B. Schelter, M. Winterhalder and J. Timmer, Granger causality: Basic theory and application to neuroscience (handbook of time series analysis), Wiley-VCH, 2006.

[17] A. Katrinecz, D. Stamate, W. Alghamdi, D. Stahl, ESM-MERGE Group Investigators, P. Delespaul, J. van Os and S. Guloksuz, "Predicting Psychosis Using the Experience Sampling Method with Mobile Applications.," in *16th IEEE International Conference on Machine Learning and Applications (IEEE ICMLA'17)*, 2017.

[18] J. Fernandez de Canete, S. Gonzalez-Perez and J. Ramos-Diaz, "Artificial neural networks for closed loop control of in silico and ad hoc type 1 diabetes," *Computer Methods and Programs in Biomedicine,* vol. 106, no. 1, pp. 55-66, 2012.

[19] M. Catalogna, E. Cohen, S. Fishman, Z. Halpern, U. Nevo and E. Ben-Jacob, "Artificial neural networks based controller for glucose monitoring during clamp test," *PLoS ONE,* vol. 7, no. 8, 2012.

[20] M. Adjouadi, M. Ayala, M. Cabrerizo, N. Zong, G. Lizarraga and M. Rossman, "Classification of leukaemia blood samples using neural networks," *Annals of Biomedical Engineering,* vol. 38, no. 4, pp. 1473-1482, 2010.

[21] Y. Yan, X. Qin, Y. Wu, N. Zhang, J. Fan and L. Wang, "A restricted Boltzmann machine based two-lead electrocardiography classification," in *The 12th International Conference on Wearable and Implantable Body Sensor Networks*, 2015.

[22] F. Murtagh, Data Science Foundations: Geometry and Topology of Complex Hierarchic Systems and Big Data Analytics, Chapman and Hall/CRC , 2017.

[23] D. Hand, H. Mannila and P. Smyth, Principles of Data Mining, MIT Press, 2001.

[24] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel and B. Thirion, "Scikit-learn: Machine Learning in Python," *Journal of Machine Learning Research,* vol. 12, pp. 2825-2830, 2011.

[25] UCI machine learning repository: Data Sets, [Online]. Available: https://archive.ics.uci.edu/ml/datasets.html. [Accessed 21 11 2018].

[26] L. Breiman, J. Friedman, R. Olshen and C. Stone, Classification and Regression Trees, Chapman & Hall/CRC, 1984.

[27] J. R. Quinlan, C4.5: Programs for Machine Learning, Morgan Kaufmann, 1993.

[28] E. Atkinson and M. Therneau, "An introduction to recursive partitioning using the RPART routines," *Rochester: Mayo Foundation,* 2000.

[29] P. Allison, Missing data, the sage handbook of quantitative methods in psychology, Thousand Oaks, CA, USA: Sage, 2009.

[30] M. Magnani, "Techniques for dealing with missing data in knowledge discovery tasks," *Obtido,* vol. 115, no. 01, 2004.

[31] D. Rubin, Multiple Imputation for Nonresponse in Surveys, vol. 81, John Wiley & Sons, 2004.

[32] J. Grzymala-Busse, W. Grzymala-Busse and L. Goodwin, "A comparison of three closest fit approaches to missing attribute values in preterm birth data," *International Journal of Intelligent Systems,* vol. 17, no. 2, pp. 125-134, 2002.

[33] G. Batista and M. Monard, "An analysis of four missing data treatment methods for supervised learning," *Applied Artificial Intelligence,* vol. 17, no. 5-6, pp. 519-533, 2003.

[34] G. Gediga and I. Duntsch, "Maximum consistency of incomplete data via noninvasive imputation," *Artificial intelligence Review,* vol. 19, no. 1, pp. 93-107, 2003.

[35] B. Twala, "n empirical comparison of techniques for handling incomplete data using decision trees," *Applied Artificial Intelligence,* vol. 23, no. 5, pp. 373-405, 2009.

[36] S. Serneels, E. Nolf and P. Espen, "Spatial sign preprocessing: a simple way to impart moderate robustness to multivariate estimators," *Journal of Chemical Information and Modeling,* vol. 46, no. 3, pp. 1402-1409, 2006.

[37] I. Guyon and A. Elisseeff, "An introduction to variable and feature selection," *Journal of Machine Learning Research,* vol. 3, pp. 1157-1182, 2003.

[38] I. Guyon, J. Weston, S. Barnhill and V. Vapnik, " Gene selection for cancer classification using support vector machines," *Machine Learning,* vol. 46, no. 1-3, pp. 389-422, 2002.

[39] R. Nilsson, " Statistical feature selection, with applications in life science," Linkoping University, 2007.

[40] P. Langley, "Selection of relevant features in machine learning," 1994.

[41] A. Kohavi and G. John, "Wrappers for feature subset selection," *Artificial Intelligence,* vol. 97, no. 1-2, pp. 273-324, 1996.

[42] C. Girish and S. Ferat, "A survey on feature selection methods," *Computers & Electrical Engineering,* vol. 40, no. 1, pp. 16-28, 2014.

[43] A. Miller, Subset Selection in Regression, Chapman and Hall/CRC, 2002.

[44] G. Cestnik, I. Konenenko and I. Bratko, "Assistant 86: A knowledge elicitation tool for sophisticated users," in *The 2Nd European Conference on European Working Session on Learning*, 1987.

[45] J. Holland, Adaptation in Natural and Artificial Systems, MIT Press, 1992.

[46] K. Kira and L. Rendell, "The feature selection problem - traditional methods and a new algorithm," in *In Proceedings of Ninth National Conference on AI*, 1992.

[47] H. Liu and R. Setiono, "A probabilistic approach to feature selection – a filter solution," in *The International Conference on Machine Learning*, 1996.

[48] H. Almuallim and T. Dietterich, "Learning with many irrelevant features," in *The Ninth National Conference on AI*, 1991.

[49] R. Halalai, C. Lemnaru and R. Potolea, "Distributed community detection in social networks with genetic algorithms," in *The 6th International Conference on Intelligent Computer Communication and Processing*, 2010.

[50] S. Nedevschi, S. Bota and C. Tomiuc, "Stereo-based pedestrian detection for collision-avoidance applications," in *Transactions on Intelligent Transportation Systems*, 2009.

[51] R. Quinlan, "Induction of decision trees," *Machine Learning,* vol. 1, no. 1, pp. 81-106, 1986.

[52] R. Caruana and D. Freitag, "Greedy attribute selection," in *The Eleventh International Conference in Machine Learning*, 1994.

[53] I. Kononenko, "Estimating attributes analysis and extensions of relief," in *The European Conference on Machine Learning*, 1994.

[54] M. Scherf and W. Brauer, "Feature selection by means of a feature weighting approach," Inst. für Informatik, 1997.

[55] C. Cardie, "Using decision trees to improve cased-based learning," The First International Conference on Knowledge Discovery and Data Mining, 1995.

[56] M. Singh and G. Provan, "Efficient learning of selective Bayesian network classifiers," The Thirteenth International Conference on Machine Learning, 1996.

[57] G. Holmes and C. Nevill-Manning, "Feature selection via the discovery of simple classification rules," Symposium on Intelligent Data Analysis, 1995.

[58] B. Pfahringer, "Compression-based feature subset selection," The IJCAI-95 Workshop on Data Engineering for Inductive Learning, 1995.

[59] J. Rissanen, "Modeling by shortest data description," *Automatica,* pp. 465-471, 1987.

[60] I. Jolliffe, Principal component analysis, Springer, 2002.

[61] R. Setiono and H. Liu, "Feature selection and discretization of numeric attributes," in *The Seventh International Conference on Tools with Artificial Intelligence*, 1995.

[62] P. Langley and S. Sage, "Induction of selective Bayesian classifiers," in *The Tenth Conference on Uncertainty in Artificial Intelligence*, 1994.

[63] R. Kohaviand and D. Sommerfield, "Feature subset selection using the wrapper method: Overfitting and dynamic search space topology," in *The First International Conference on Knowledge Discovery and Data Mining*, 1995.

[64] D. Allen, "The relationship between variable selection and data augmentation and a method for prediction," *Technometrics,* vol. 16, no. 1, pp. 125-127, 1974.

[65] G. John, R. Kohavi and P. Pfleger, "Irrelevant features and the subset selection problem," in *The Eleventh International Conference in Machine Learning*, 1994.

[66] H. Vafaie and K. De Jong, "Genetic algorithms as a tool for restructuring feature space representations," in *In Proceedings of the International Conference on Tools with A.I*, 1995.

[67] K. Cherkauer and J. Shavlik, "Growing simpler decision trees to facilitate knowledge discovery," in *In Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, 1996.

[68] P. Domingos, "Context-sensitive feature selection for lazy learners," *Artificial Intelligence Review,* pp. 227-253, 1997.

[69] M. Pazzani, "Searching for dependencies in Bayesian classifiers," in *In Proceedings of the Fifth International Workshop on AI and Statistics*, 1995.

[70] R. Kohavi, "Wrappers for performance enhancement and oblivious decision graphs," Stanford University, Computer Science Department, 1995.

[71] I. Witten, E. Frank, M. A. Hall and C. J. Pal, Data Mining: Practical machine learning tools and techniques, Morgan Kaufmann, 2016.

[72] T. Fawcett, "An introduction to ROC analysis," *Pattern Recognition Letters,* vol. 27, no. 8, pp. 861-874, 2006.

[73] S. Mason and N. Grahm, "Areas beneath the relative operating characteristics (roc) and relative operating levels (rol) curves: Statistical significance and interpretation," *Quarterly Journal of the Royal Meteorological Society,* pp. 2145-2166, 2002.

[74] R. Kohavi, "A study of cross-validation and bootstrap for accuracy estimation and model selection," in *The Fourteenth International Joint Conference on Artificial Intelligence*, 1995.

[75] Y. Freund and R. Schapire, "Experiments with a new boosting algorithm," in *Machine Learning: Proceedings of the Thirteenth International Conference*, 1996.

[76] L. Breiman, "Bagging predictors," *Machine Learning,* 1996.

[77] A. M. Molinaro, R. Simon and R. M. Pfeiffer, "Prediction error estimation a comparison of resampling methods," *Bioinformatics,* vol. 21, no. 15, pp. 3301-3307, 2005.

[78] G. Cohen, M. Hilario, C. Pellegrini and A. Geissbuhler, "SVM modelling via a hybrid genetic strategy: a healthcare application," *Studies in health technology and informatics,* vol. 116, pp. 193-198, 2005.

[79] P. Tan, M. Steinbach and V. Kumar, Introduction To Data Mining, Pearson, 2016.

[80] S. Russell and P. Norvig, Artificial intelligence: a modern approach, Boston: Pearson, 2016.

[81] G. James, D. Witten, T. Hastie and R. Tibshirani, An introduction to statistical learning: With applications in r, Springer , 2014.

[82] L. Clemmensen, T. Hastie, D. Witten and B. Ersboll, "Sparse discriminant analysis," *Technometrics,* 2011.

[83] P. Xanthopoulos, P. M. Pardalos and T. B. Trafalis, "Linear Discriminant Analysis," in *Robust data mining*.

[84] H. Chun and S. Keles, "Sparse partial least squares regression for simultaneous dimension reduction and variable selection," *Journal of the Royal Statistical Society,* vol. 72, no. 1, pp. 3-25, 2010.

[85] I. Jolliffe, N. Trendafilov and M. Uddin, "A modified principal component technique based on the lasso," *Journal of Computational and Graphical Statistics,* vol. 12, no. 3, 2003.

[86] H. Zou, T. Hastie and R. Tibshirani, "Sparse principal component analysis," *Journal of Computational and Graphical Statistics,* 2006.

[87] K. Shefali, "An evolutionary Bayesian network learning algorithm using feature subset selection for Bayesian network classifiers," *International Journal of Computer Applications,* vol. 135, no. 13, pp. 1-8, 2016.

[88] M. Kanagawa, P. Hennig, D. Sejdinovic and B. Sriperumbudur, "Gaussian Processes and Kernel Methods: A Review on Connections and Equivalences," *Arxiv e-prints,* vol. arXiv:1805.08845v1, 2018.

[89] M. Kuss and E. Rasmussen, "Assessing approximate inference for binary Gaussian process classification," *Journal of machine learning research,* vol. 6, pp. 1679-1704, 2005.

[90] T. Cover and P. Hart, "Nearest neighbour pattern classification," in *TransactionsonInformation Theory*, 1967.

[91] E. Alpaydin, Introduction to machine learning, MIT Press, 2010.

[92] T. Hancock, T. Jiang, M. Li and J. Tromp, "Lower bounds on learning decision lists and trees," *Information and Computation,* vol. 126, no. 2, pp. 21-27, 1996.

[93] H. Zantema and H. Bodlaender, " Finding small equivalent decision trees is hard," *International Journal of Foundations of Computer Science,* pp. 343-354, 2000.

[94] H. Hunt, E. Marin and J. Stone, Experiments in induction, Academic Press, 1966.

[95] L. Rokach and O. Maimon, Data Mining with Decision Trees, World Scientific, 2008.

[96] C. Shannon, "A mathematical theory of communication," *Bell Syst. Tech. J,* pp. 379-423, 1986.

[97] Y. Li, "Predicting materials properties and behaviour using classification and regression trees," *Materials Science and Engineering,* vol. 1, no. 2, pp. 261-268, 2006.

[98] S. Caetanoa, J. Aires, M. Daszykowskia and Y. Heydena, "Prediction of enantioselectivity using chirality codes and classification and regression trees," *Analytica Chimica Acta,* vol. 544, no. 1-2, pp. 315-326, 2005.

[99] A. Andryashin, "Financial applications of classification and regression trees," Center of Applied Statistics and Economics, Humboldt University, Berlin, 2005.

[100] S. Kwok and C. Carter, "Multiple decision trees," in *the Fourth Annual Conference on Uncertainty in Artificial Intelligence*, 2013.

[101] D. Dietterich and E. Kong, "Machine learning bias, statistical bias, and statistical variance of decision tree algorithms," 1995.

[102] Y. Amit, D. Geman and K. Wilder, "Joint induction of shape features and tree classifiers," in *Pattern Analysis and Machine Intelligence*, 1997.

[103] A. Criminisi, J. Shotton and E. Konukoglu, "Decision forests for classification, regression, density estimation, manifold learning and semi-supervised learning," 2011.

[104] L. Breiman, "Random forest," *Machine Learning,* vol. 45, no. 1, pp. 5-32, 2001.

[105] I. Maglogiannis, E. Loukis, E. Zafiropoulos and A. Stasis, "Support vectors machine-based identification of heart valve diseases using heart sounds," *Computer methods and programs in biomedicine,* vol. 95, no. 1, pp. 47-61, 2009.

[106] R. Thurston, K. Matthews, J. Hernandez and F. De La Torre, "Improving the performance of physiologic hot flash measures with support vector machines," *Psychophysiology,* vol. 46, no. 3, pp. 285-292, 2009.

[107] A. Chu, H. Ahn, B. Halwan and B. Kalminand, "A decision support system to facilitate management of patients with acute gastrointestinal bleeding," *Artificial Intelligence in Medicine,* vol. 42, no. 3, pp. 247-259, 2008.

[108] S. Rice, G. Nenadic and B. Stapley, "Mining protein function from text using term-based support vector machines," *BMC Bioinformatics,* vol. 6, no. 1, p. S22, 2005.

[109] N. Nikolaev and H. Iba, Adaptive learning of polynomial networks: genetic programming, backpropagation and Bayesian methods, Springer, 2006.

[110] F. Gunther, "Neuralnet: Training of neural networks," *The R Journal,* vol. 2, no. 1, pp. 30-38, 2010.

[111] B. Ripley, Pattern recognition and neural networks, Cambridge University Press, 1996.

[112] S. Russell and P. Norvig, "A modern, agent-oriented approach to introductory artificial intelligence," *ACM SIGART Bulletin,* vol. 6, no. 2, pp. 24-26, 1995.

[113] M. Buhmann, Radial Basis Functions, Cambridge University Press, 2003.

[114] S. Bull, J. Lewinger and S. Lee, "Penalized maximum likelihood estimation for multinomial logistic regression using the Jeffreys prior," 2005.

[115] J. Schmidhuber, " Deep learning in neural networks: An overview," *Neural Networks,* vol. 61, pp. 85-117, 2015.

[116] G. Hinton, L. Deng, D. Yu and G. Dahl, "Deep neural networks for acoustic modelling in speech recognition: The shared views of four research groups," *IEEE Signal Processing Magazine,* vol. 29, no. 6, pp. 82-97, 2012.

[117] N. Morgan, "Deep and wide: Multiple layers in automatic speech recognition," *IEEE Transactions on Audio, Speech, and Language Processing,* vol. 20, no. 1, pp. 7-13, 2012.

[118] F. Li, L. Tran and K. Thung, " Robust deep learning for improved classification of ad/mci patients," *International Workshop on Machine Learning in Medical Imaging,* 2014.

[119] M. Arbabshirani, S. Plis, J. Sui and V. Calhoun, "Single subject prediction of brain disorders in neuroimaging: promises and pitfalls," *Neuroimage,* vol. 145, pp. 137-165, 2016.

[120] V. Calhoun and J. Sui, "Multimodal fusion of brain imaging data a key to finding the missing links in complex mental illness," *Biological psychiatry: cognitive neuroscience and neuroimaging,* vol. 1, no. 3, pp. 230-244, 2016.

[121] P. Glauner, "Deep convolutional neural networks for smile recognition," *arXiv preprint ,* 2015.

[122] "United Nations office on drugs and crime, world drug report," 2016.

[123] O. Van, M. Bak and H. Hanssen, " Cannabis use and psychosis: A longitudinal population based study," *American Journal of Epidemiology,* vol. 156, no. 4, 2002.

[124] T. Moore, S. Zammit and A. Lingford-Hughes, "Cannabis use and risk of psychotic or affective mental health outcomes: a systematic review," *The Lancet Psychiatry,* vol. 370, no. 9584, pp. 319-328, 2007.

[125] R. Murray, H. Quigley and D. Quattrone, " Traditional marijuana, high-potency cannabis and synthetic cannabinoids: Increasing risk for psychosis," *World Psychiatry,* vol. 15, no. 3, pp. 195-204, 2016.

[126] M. Di Forti, C. Morgan and P. Dazzan, "High-potency cannabis and the risk of psychosis.," *The British Journal of Psychiatry,* vol. 195, no. 6, pp. 488-491, 2009.

[127] S. Dragt, D. Nieman and F. Schultze-Lutte, " Cannabis use and age at onset of symptoms in subjects at clinical high risk for psychosis," *Acta Psychiatrica Scandinavica,* vol. 125, no. 1, pp. 45-53, 2011.

[128] R. Radhakrishnan, S. Wilkinson and D. Souza, "Gone to pot: A review of the association between cannabis and psychosis.," *Frontiers in Psychiatry,* vol. 5, p. 54, 2014.

[129] R. C. Team, "R: A Language and Environment for Statistical Computing," R Foundation for Statistical Computing, [Online]. Available: https://www.R-project.org/.

[130] M. Hofmann and R. Klinkenberg, RapidMiner: Data mining use cases and business analytics applications, Chapman & Hall/CRC Data Mining and Knowledge Discovery Series, 2013.

[131] M. Zaharia, R. S. Xin, P. Wendell, T. Das, M. Armbrust, A. Dave, X. Meng, J. Rosen, S. Venkataraman, M. J. Franklin, A. Ghodsi, J. Gonzalez, S. Shenker and I. Stoica, "Apache Spark: A Unified Engine for Big Data Processing," *Commun. ACM,* vol. 59, no. 11, pp. 56-65, 2016.

[132] D. Fergusson, L. Horwood and E. Ridder, "Tests of causal linkages between cannabis use and psychotic symptoms," *Addiction,* vol. 100, no. 3, pp. 354-366, 2005.

[133] S. Salzberg, "C4.5: Programs for machine learning by Ross Quinlan," *Machine Learning,* vol. 16, no. 3, pp. 235-240, 1994.

[134] G. Busse, W. Busse and L. Goodwin, " A comparison of three closest fit approaches to missing attribute values in preterm birth data," *International Journal of Intelligent Systems,* vol. 17, no. 2, pp. 125-134, 2002.

[135] G. Batista and M. Monard, "An analysis of four missing data treatment methods for supervised learning," *Applied Artificial Intelligence,* vol. 17, no. 5-6, pp. 519-533, 2003.

[136] M. Kuhn, J. Wing, S. Weston, A. Williams and e. al., "caret: Classification and Regression Training," 2017.

[137] D. Mease, A. Wyner and A. Buja, "Boosted classification trees and class probability/quantile estimation," *Journal of Machine Learning Research,* vol. 8, no. 1, pp. 409-439, 2007.

[138] G. Shakhnarovich, T. Darrell and P. Indyk, Nearest-neighbor methods in learning and vision: theory and practice, MIT Press, 2006.

[139] J. Friedman, "Greedy function approximation: a gradient boosting machine," *Annals of Statistics,* vol. 29, no. 5, pp. 1189-1232, 2001.

[140] J. Cohen, " A coefficient of agreement for nominal scales," *Educational and Psychological Measurement,* vol. 20, no. 1, pp. 37-46, 1960.

[141] S. Sarawagi, S. Thomas and R. Agrawal, "Integrating association rule mining with relational database systems," *ACM SIGMOD,* vol. 27, no. 2, pp. 343-354, 1998.

[142] M. Di Fort, H. Sallis and F. Allegri, "Daily use, especially of high-potency cannabis, drives the earlier onset of psychosis in cannabis users," *Schizophrenia Bulletin,* vol. 40, no. 6, pp. 1509-1517, 2013.

[143] F. Seixas, B. Zadrozny and J. Laks, "A Bayesian network decision model for supporting the diagnosis of dementia, Alzheimers disease and mild cognitive impairment," *Computers in Biology and Medicine,* vol. 51, no. 1, pp. 140-158, 2014.

[144] A. Liaw and M. Wiener, "Classification and Regression by randomForest," *R News,* vol. 2, no. 3, pp. 18-22, 2002.

[145] N. Qazi and K. Raza, "Effect of Feature Selection, SMOTE and Under Sampling on Class Imbalance Classification," in *2012 UKSim 14th*, 2012.

[146] A. Candel, V. Parmar, E. LeDell and A. Arora, " Deep Learning with H2O," 2015. [Online]. Available: http://h2o.ai/resources.

[147] S. Aiello and E. Eckstrand, "Machine Learning with R and H2O," 2016. [Online].

[148] D. Bohning and W. Bohning, "Revisiting Youden's index as a useful measure of the misclassification error in a meta-analysis of diagnostic studies," *Statistical Methods in Medical Research,* vol. 17, no. 6, pp. 543-554, 2008.

[149] M. Pepe, The Statistical Evaluation of Medical Tests for Classification and Prediction, Oxford University Press, 2003.

[150] N. Perkins and F. Schisterman, "The inconsistency of optimal cutpoints obtained using two criteria based on the receiver operating characteristic curve," *Am J Epidemiol,* vol. 163, no. 7, pp. 670-675, 2006.

[151] M. Shouman and T. Turner, "Using decision tree for diagnosing heart disease patients," in *the Ninth Australasian Data Mining Conference*, 2011.

[152] W. Buntine and T. Niblett, "A further comparison of splitting rules for decision-tree induction," *Machine Learning,* vol. 8, no. 1, pp. 75-85, 1992.

[153] W. Liu and A. White, "The importance of attribute selection measures in decision tree induction," *Machine Learning,* vol. 15, no. 1, pp. 25-41, 1994.

[154] M. Ojala and G. Garriga, " Permutation tests for studying classifier performance," *Journal of Machine Learning Research,* vol. 11, no. 1, pp. 1833-1863, 2010.

[155] P. Good, Permutation tests: a practical guide to resampling methods for testing hypotheses, Springer Science & Business Media, 2013.

[156] T. Maszczyk and W. Duch, "Comparison of Shannon, Renyi and Tsallis entropy used in decision trees," in *International Conference on Artificial Intelligence and Soft Computing*, 2008.

[157] L. Raileanu and K. Stoffel, "Theoretical comparison between the Gini index and information gain criteria," in *Annals of Mathematics and Artificial Intelligence*, 2004.

[158] C. Tsallis, R. Mendes and A. R. Plastino, "The role of constraints within generalised non-extensive statistics," *Physica A: Statistical Mechanics and its Applications,* vol. 261, no. 3-4, pp. 534-554, 1998.

[159] I. Guyon, J. Li and T. Mader, "Competitive baseline methods set new standards for the nips 2003 feature selection benchmark," *Pattern recognition letters,* vol. 12, no. 1, pp. 1438-1444, 2007.

[160] I. Guyon, " Design of experiments of the nips 2003 variable selection benchmark," 2003.

[161] I. Guyon, S. Gunn, A. Ben-Hur and G. Dror, "Result analysis of the nips 2003 feature selection challenge," *Advances in neural information processing systems,* pp. 545-552, 2005.

[162] H. Osman, "Correlation-based feature ranking for online classification," in *Systems, Man and Cybernetics*, 2009.

[163] I. Guyon and A. Elisseeff, "An introduction to feature extraction, Feature extraction," *Feature extraction,* pp. 1-25, 2006.

[164] G. Livingston, A. Sommerlad, V. Orgete, S. Codtafreda, J. Huntley, D. Ames, C. Ballard and A. Bums, "Dementia prevention, intervention, and care," *The Lancet,* vol. 390, no. 10113, pp. 2673-2734, 2017.

Appendix 1

Wajdi Alghamdi - Publications

(* indicates joint-first authors)

- PIDT: A Novel Decision Tree Algorithm Based on Parameterised Impurities and Statistical Pruning Approaches, Proc. 14th International Conference on Artificial Intelligence Applications and Innovations, AIAI 2018, to appear in Springer IFIP
  Daniel Stamate, **Wajdi Alghamdi***, Daniel Stahl, Doina Logofatu and Alexander Zamyatin

- A New Machine Learning Framework for Understanding the Link between Cannabis Use and First-Episode Psychosis, Proc. eHealth2018 12th Annual Conference on Health Informatics meets eHealth, to appear in Health Technologies and Informatics series
  Wajdi Alghamdi, Daniel Stamate, Daniel Stahl, Alexander Zamyatin, Robin Murray, Marta Di Forti

- Can Artificial Neural Networks Predict Psychiatric Conditions Associated with Cannabis Use? Proc. 14th International Conference on Artificial Intelligence Applications and Innovations, AIAI 2018, to appear in Springer IFIP
  Daniel Stamate, **Wajdi Alghamdi***, Daniel Stahl, Alexander Zamyatin, Robin Murray and Marta di Forti

- Predicting First-Episode Psychosis Associated with Cannabis Use with Artificial Neural Networks and Deep Learning, Proc. 17th International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems, IPMU 2018, to appear in Springer
  Daniel Stamate, **Wajdi Alghamdi*,** Daniel Stahl, Ida Pu, Fionn Murtagh, Danielle Belgrave, Robin Murray and Marta di Forti

- Predicting Psychosis Using the Experience Sampling Method with Mobile Applications.
  Proceedings of 16th IEEE International Conference on Machine Learning and Applications (IEEE ICMLA'17), 2017, Publisher: IEEE, DOI: 10.1109/ICMLA.2017.00-84
  Andrea Katrinecz, Daniel Stamate, **Wajdi Alghamdi,** Daniel Stahl, ESM-MERGE Group Investigators, Philippe Delespaul, Jim van Os, Sinan Guloksuz.

- A prediction Modelling and Pattern Detection Approach for the First-Episode Psychosis Associated to Cannabis Use. Proceedings of 15th IEEE International Conference on Machine Learning and Applications (IEEE ICMLA'16), 2016, Publisher: IEEE, DOI: 10.1109/ICMLA.2016.0148
  Wajdi Alghamdi, Daniel Stamate, Katherine Vang, Daniel Stahl, Marco Colizzi, Giada Tripoli, Diego Quattrone, Olesya Ajnakina, Robin M. Murray and Marta Di Forti

- A Statistical Learning Approach to Predicting the First Episode Psychosis Associated to High Potency Cannabis Use. Poster at 1st Prediction Modelling in Psychiatric Research Workshop (UK-PMPR), 2016, **Wajdi Alghamdi,** Daniel Stamate, Katherine Vang, Daniel Stahl, Marco Colizzi, Giada Tripoli, Diego Quattrone, Olesya Ajnakina, Robin M. Murray and Marta Di Forti

Appendix 2

First-episode psychosis - cannabis clinical dataset
Data Dictionary (glossary)

Field Name	New Name	Description	Value Code
patient	patient	Primary label for classification: whether or not a subject is a patient with first episode psychosis.	patient control
ageon	age_at_study	Age at the time of the study.	range1 [0 - 24.5] range2 [24.5 -33.5] range3 [33.5 - 61]
gender	gender	Gender of subject.	female male
birth_pl	birthplace	Was the subject was born in the UK.	non-UK born UK born
white	white	Describe the ethnicity (from Genetic)	range1 [0 - 0.086] range2 [0.086 - 0.811] range3 [0.811 - 1]
african	african	Describe the ethnicity (from Genetic)	range1 [0 - 0.033] range2 [0.033 - 0.178] range3 [0.178 - 1]
asian	asian	Describe the ethnicity (from Genetic)	range1 [0 - 0.023] range2 [0.023 - 0.817] range3 [0.817 - 1]
level_ed	education	Level of education of the subject.	no qualification GCSE/O levels vocational/college A levels university/professional

living_s	living status	Living situation of the subject.	1 : Living alone 2-7: Other statuses
relation	relationship_status	Relationship status of subject.	1: Single 2: Married/ living with some-one 3: In a steady relationship 4: Divorced, separated 5: Widowed
children	children	Number of children	Integer (range from 0-3)
homeless	homeless	homeless duration	0 = Never 1 = 6 months ago 2 = 1 years 3 = 5+ years
authorit	authorit	authority care	0 = Never 1 = 6 months ago 2 = 1 years 3 = 5+ years
family_h	family_hist_psychiatric	Does the subject have a family history of psychiatric illness?	No Yes Dementia
psych_y	family_hist_psychosis	Does the subject have a family history of psychosis?	0: No 1: Yes
bullying	bullying	Have the participant experi-enced bullying	0: No 1: Yes
alcohol	lifetime_alcohol_user	Has the subject has ever used alcohol.	0: No 1: Yes
copy_tob	lifetime_tobacco_user	Has the subject has ever used tobacco.	0: No 1: Yes
other_dr	lifetime_drug_user	Has the subject has ever used drugs other than cannabis.	0: No 1: Yes

employmentever	ever_employed	Has the subject has ever been in employment.	0: No 1: Yes
copy_can	lifetime_cannabis_user	Has the subject has ever used cannabis.	0: No 1: Yes
copy_age	age_first_cannabis	Age of subject when first used cannabis.	range1 [0.0 - 15] range2 [16 - 25] range3 [25 - 60] Never Used
current	current_cannabis_user	Did the subject use cannabis within the prior month?	Never Used 0: No 1: Yes
totfreq2	cannabis_fqcy	How frequently the subject uses cannabis.	0: Never Used 1: Only At Weekends 2: Daily
TotCANTYPE2	Cannabis_type	The type of cannabis used by the subject.	0: Never Used 1: Hash 2: Skunk
riskcan0	cannabis_measure	A combination of cannabis_fqcy and cannabis_type	0: Non User 1: Hash At Weekends 2: Hash Less Than Daily 3: Hash Daily 4: Skunk At Weekends 5: Skunk Less Than Daily 6: Skunk Daily
duration	duration	How long has the subject used cannabis?(in months)	Decimal (range from 0-41).
agecabuse14	age1st14	Age of subject when first used cannabis.	Never Used 0: No 1: Yes
age1st15	age1st15	Was the subject aged 15 or under when he first used cannabis?	Never Used 0: No 1: Yes