ANALYSIS AND APPLICATIONS OF DATA MINING ALGORITHMS

NESLİHAN DOĞAN

BOĞAZİÇİ UNIVERSITY

2010

ANALYSIS AND APPLICATIONS OF DATA MINING ALGORITHMS

Thesis submitted to the

Institute for Graduate Studies in the Social Sciences

in partial fulfillment of the requirements for the degree of

Master of Arts

in

Management Information Systems

by

NESLİHAN DOĞAN

Boğaziçi University

2010

Thesis Abstract

Neslihan Doğan, “Analysis and Applications of Data Mining Algorithms”

Classification algorithms are the most commonly used Data Mining models that are widely used to extract valuable knowledge from huge amounts of data. Comparing the classification algorithms has been interesting the data mining community for many years. The criteria to evaluate the classifiers are mostly the accuracy, complexity, robustness, scalability, integration, comprehensibility, stability and interestingness abilities of it. This thesis study is concerned with the accuracy, complexity and robustness of the classifiers. The data miner selects the model mostly with respect to its classification accuracy; therefore, the performance of each classifier plays a very crucial role. As complexity, the cpu time consumed by each classifier is implied in the study. The study firstly discusses the application of some classification models on multiple datasets in 3 stages: firstly implementing the algorithms on pure datasets, secondly implementing the algorithms on the same datasets where continuous numerical variables are discretised, thirdly implementing the algorithms on the same datasets where Principal Component Analysis is applied. On the results, the accuracies and complexities are compared. The relationship of dataset characteristics and implementation attributes between accuracy and complexity is also debated, and finally, a regression model is introduced for predicting the classifier accuracy and complexity with given dataset and implementation conditions. Finally, the study is also concerned with the robustness of the classifiers which is measured by repetitive experiments on noisy and cleaned datasets.

Tez Özeti

Neslihan Doğan, “Veri Madenciliği Algoritmalarının Analizi ve Uygulanması“

Sınıflandırma algoritmaları büyük veri setlerinden kıymetli bilginin elde edilmesi amacıyla kullanılan Veri Madenciliği modellerinden en yaygınıdır. Yıllardır, sınıflandırma algoritmalarının birbirleriyle karşılaştırılması veri madenciliği toplumunun ilgisini çekmektedir. Genel olarak modelleri karşılaştırma kriterleri modelin doğruluğu, karmaşıklığı, sağlamlığı, ölçeklenebilirliği, entegrasyonu, anlaşılabilirliği, istikrarlılığı ve ilgi çekiciliğidir. Bu çalışma sınıflandırma modellerinin doğruluk, zorluk ve sağlamlık özellikleriyle ilgilenmektedir. Veri madencisi genellikle modelini seçerken sınıflandırma doğruluk oranına göre karar verir, dolayısıyla her modelin doğruluğu önemli rol oynar. Bu çalışmada zorluk ile modelin harcadığı işlemci zamanı kastedilmektedir. Çalışma bazı sınıflandırma algoritmalarının çoklu veri setleri üzerinde 3 aşamalı deney sonuçlarını sunmaktadır: 1. Algoritmaların ham veri setleri üzerinde uygulanması, 2. Aynı algoritmaların veri setlerindeki sürekli sayıların münferit aralıklara dönüştürülmesinden sonra tekrar edilmesi, 3.Aynı algoritmaların veri setlerinde Ana Bileşenler Çözümlemesi yapılmasından sonra tekrar edilmesidir. Ortaya çıkan sonuçlara göre algoritmaların farklı deney aşamalarındaki doğruluk ve karmaşıklık dereceleri karşılaştırılmıştır. Ayrıca veri setlerinin karakteristikleri, ya da uygulama detayları ile doğruluk ya da zorluk arasındaki ilişkiler de incelenmiş ve son olarak da veri seti ve uygulama özellikleri ışığında bir sınıflandıma algoritmasının doğruluk ve karmaşıklık derecesini tahmin edebilecek bir regresyon modeli kurulmaya çalışılmıştır. Son olarak tez çalışması temizlenmiş ve temizlenmemiş veri setleri üzerinde tekrarlı deneylerle ölçülebilen sınıflayıcıların sağlamlığı kriteriyle de ilgilenmiştir.

ACKNOWLEDGEMENTS

I would like to thank my thesis advisor Assoc. Prof. Dr. Zuhal Tanrıkulu for guiding and facilitating my research activities. Her assistance in matters of both research and university bureaucracy has been greatly helpful. I would also like to thank to my thesis committee members: Assoc. Prof Dr. Sevinç Gulsecen and Asst. Prof. Dr. Özgür Döğerlioğlu for their contributions to my study. Also my special thanks go to Aylin Birsen Yılmas for her kindness and great work as the format editor. Lastly, I further would like to thank to all of my friends, my family and my spouse Çağrı Doğan for their continuous encouragement and support.

CONTENTS

CHAPTER I. INTRODUCTION 1

Data Mining Standards 2
Classification Algorithm Assessment 4
Data Preprocessing 5
Literature Review 6

CHAPTER 2. PURPOSE AND METHOD 9

Research Questions 10
Method 11
Data collection 13
Algorithms 16
Implementation 18

CHAPTER 3. EXPERIMENTAL RESULTS AND DISCUSSIONS 21

Research question 1: Does implementing the same classification algorithm on multiple datasets and with different implementation techniques result in different performance indicators? 22
Research question 2: Do the characteristics of the datasets affect the performance results of the classification algorithms? 38
Research question 3: Does binning the continuous numerical variables in the dataset into discreet intervals affect the classifier accuracy? 43
Research question 4: Does applying principal component analysis in the dataset affect the classifier accuracy? 44
Research question 5: Based on the results derived from the empirical results of this study (applying classifiers on various dataset with different implementation techniques), can a model to predict the performance of the classification algorithm be built? 46
Research question 6: Does implementing the same classification algorithm on multiple datasets and with different implementation techniques result in significantly different complexity? 48
Research question 7: Do the characteristics of the datasets affect the complexity of the classification algorithms? 53
Research question 8: Does binning the continuous numerical variables in the dataset into discreet intervals affect the classifier complexity? 56
Research question 9: Does applying principal component analysis in the dataset affect the classifier complexity? 57
Research question 10: Based on the results derived from the empirical results of this study (applying classifiers on various dataset with different implementation techniques), can a model to predict the complexity (consumed CPU time in seconds) of the classification algorithm be built? 59
Research question 11: Are the abilities of classifiers to handle missing or noisy data different? 62

CHAPTER 4. CONCLUSION 64

REFERENCES 70

TABLES

Table 1. Dataset Characteristics 14
Table 2. Accuracy Results / Pure Implementation and 10-Fold Cross Validation 24
Table 3. Accuracy Results / Pure Implementation and 66% Train-Test Split 24
Table 4. Accuracy Results / After Discretisation and 10-Fold Cross Validation 25
Table 5. Accuracy Results / After Discretisation and 66% Train-Test Split 25
Table 6. Accuracy Results / After PCA and 10-Fold Cross Validation..... 26
Table 7. Accuracy Results / After PCA and 66% Train-Test Split..... 26
Table 8. Complexity Results / Pure Implementation and 10-Fold Cross Validation. 27
Table 9. Complexity Results / Pure Implementation and 66% Train-Test Split..... 27
Table 10. Complexity Results / After Discretisation and 10-Fold Cross Validation. 28
Table 11. Complexity Results / After Discretisation and 66% Train-Test Split..... 28
Table 12. Complexity Results / After PCA and 10-Fold Cross Validation 29
Table 13. Complexity Results / After PCA and 66% Train-Test Split 29
Table 14. Overall Best Accuracy Results 30
Table 15. Best Accuracy Results for Each Dataset / Pure Implementations 31
Table 16. Best Accuracy Results for Each Dataset / After Discretisations 32
Table 17. Best Accuracy Results for Each Dataset / After PCA 34
Table 18. Overall Distribution of Classifiers Across Performance Intervals 35
Table 19. One Way Anova / Based on Pure Implementation Step Results 36
Table 20. One Way Anova / Based on Discretization Step Results 36
Table 21. One Way Anova / Based on PCA Step Results 36
Table 22. An Excerpt From the Results Dataset 39
Table 23. Correlation Between Accuracy and Number of Variables 40
Table 24. Correlation Between Accuracy and Number of Nominal Variables 40
Table 25. Correlation Between Accuracy and Number of Numerical Variables..... 41
Table 26. Correlation Between Accuracy and Number of Target Class Types 41
Table 27. Correlation Between Accuracy and Number of Instances 41
Table 28. Correlation Between Accuracy and Algorithm Type 42
Table 29. Correlation Between Accuracy and Validation Methods 42
Table 30. Correlation Between Accuracy and Discretisation 43
Table 31. Anova Results of Performance and Discretisation 43
Table 32. Correlation Between Accuracy and PCA 44
Table 33. Correlation Between Accuracy and Cumulative Variance in PCA 44
Table 34. Correlation Between Accuracy and Number of Components in PCA..... 45
Table 35. Anova Results of Performance and PCA 45
Table 36. Regression Results / Accuracy..... 47
Table 37. One Way Anova / Based on Pure Implementation Step Results 49
Table 38. One Way Anova / Based on Discretisation Step Results..... 50
Table 39. One Way Anova / Based on PCA Implementation Step Results..... 50
Table 40. One Way Anova / Based on Overall Implementation Step Results..... 52
Table 41. Overall Distribution of Classifiers Across Complexity Intervals 53
Table 42. Correlation Between Complexity and Number of Variables 54
Table 43. Correlation Between Complexity and Number of Nominal Variables 54
Table 44. Correlation Between Complexity and Number of Numerical Variables ... 55
Table 45. Correlation Between Complexity and Number of Target Class Types 55
Table 46. Correlation Between Complexity and Number of Instances 55
Table 47. Correlation Between Complexity and Validation Method 56
Table 48. Correlation Between Complexity and Algorithm Type 56
Table 49. Correlation Between Complexity and Discretisation 57

Table 50. Anova Results of Complexity and Discretisation 57
Table 51. Correlation Between Complexity and Number of Principal Components. 58
Table 52. Correlation Between Complexity and % of Cumulative Variance in PCA 58
Table 53. Correlation Between Complexity and PCA 59
Table 54. Anova Results of Complexity and PCA 59
Table 55. Regression Results / Complexity 61
Table 56. Robustness Comparison 63

FIGURES

Fig. 1 CRISP-DM phases 4
Fig. 2 Determining which test to use 14
Fig. 3 Methodological framework 15
Fig. 4 Anova mean plots / pure stage 35
Fig. 5 Anova mean plots / discretisation stage 37
Fig. 6 Anova mean plots / PCA stage 37
Fig. 7Anova mean plots / all trials 38
Fig. 8 Anova mean plots / pure implementations 50
Fig. 9 Anova mean plots / after discretisations 51
Fig. 10 Anova mean plots / after PCA 51
Fig. 11 Anova mean plots / all trials 52

CHAPTER I.

INTRODUCTION

Classification or prediction tasks are the most widely used types of data mining algorithms. Classification algorithms are supervised methods that discover the hidden associations between the target class and the independent variables (Maimon & Rokach, 2008). Supervised learning algorithms allow tags to be assigned to the observations so that an unobserved data can be categorized based on the training data (Han & Kamber, 2005). A task, a model structure, a score function, a search method and a data management method are the main components of each algorithm (Hand, Mannila & Smyt, 2001). Image and pattern recognition, medical diagnosis, loan approval, detecting faults or financial trends are among the well known examples of classification tasks are (Dunham, 2002).

In literature, it is obvious to see many example case studies in which the classification algorithms are utilized. Limanto, Cing and Watkins introduces a study where they use AIRS algorithm to understand the basic customer profiles and predict the customers who will most likely subscribe to 3G thus they aim to help telecommunication companies create a strategy for gaining customers (Limanto, Cing & Watkins, 2007). Another recent study introduces the implementation of large scale learning algorithms to estimate the bounce rate that is the fraction of customers who click on an advertisement but go on another task thus results in a poor return on investment (Sculley, Malkin, Basu, & Bayardo, 2009). The advantages and benefits of utilizing Artificial Neural Networks are debated on a study where authors try to predict the financial information manipulations (Küçükkocaoğlu, Benli & Küçüksözen,

2009). A similar research is proposed by authors who state the usefulness of Decision Trees, Neural Network and Bayesian Belief Network in determining the fraudulent financial statements (Kirkos, Spathis & Manolopoulos, 2007). Harper, Moy and Konstan proposes an interesting study where they compare the prediction abilities of learning algorithms and human beings on classifying the question types on certain question and answer websites as conversational or informational. Their algorithms are claimed to approach human performance (Harper, Moy & Konstan, 2009). The effects of product mix, personnel, physical conditions, and services on the customer satisfaction in large shopping areas are analysed by Artificial Neural Networks in another study (Tolon & Tosunoğlu, 2008).

Data Mining Standards

During the data mining projects different technologies and tools are used frequently for distinct purposes. This fact brings the importance of the integration and communication abilities of the tools. Emerging standards are also necessary for testing or comparing models. In the recent years a couple of data mining standards have matured and data mining tools and products are utilizing these standards. The main reason why many different standards exist today is that data mining is used in so many different ways and in so many different systems that each of these often require its own standards.

As Tang and Jamie state that there were no relational databases or SQL many years ago, and no standard to query different data sources (Tang & MacLennan, 2005). One of the first standards in the data mining space is the Predictive Model Markup Language (PMML) developed by Data Mining Group (DMG). PMML described standards of interchanging of data mining models among systems in a vendor-neutral

format. (Data Mining Group, 2010). Another emerging standard is web services for developing remote and distributes data mining applications by XMLA (Simba, 2010). MDX (Multidimensional Expression) is the most commonly used multi-dimensional expression language used by OLAP servers and BI communities (Tang & MacLennan, 2005). Privacy-preserving data mining (PPDM) is another emerging standard which is interested in the definition of privacy in data mining. Definitions of privacy can vary according to context, culture, and environment (Oliveira and Zaiane, 2004). Common Warehouse Metadata (CWM) is a widely accepted object oriented data mining standard which provides a model for representing data mining metadata in XML. Information and data mining tasks are represented in the form of form objects that are reusable, scalable and portable (Object Management Group, 2010). Java Data Mining (JDM) is another well-known and well-established object oriented data mining standard (Hornick, Marcade & Venkayala, 2007). The Cross Industry Standard Process for Data Mining (CRISP-DM) was a project to develop an industry- and tool-neutral data mining process model. It aims to make data mining tasks more manageable and reliable by standardizing the data mining phases, integrating and validating best practices from experts in diverse industries. In CRISP-DM standard life cycle of a data mining project consists of six phases as shown in Figure 1 (Cross Industry Standard Process for Data Mining, 2010).

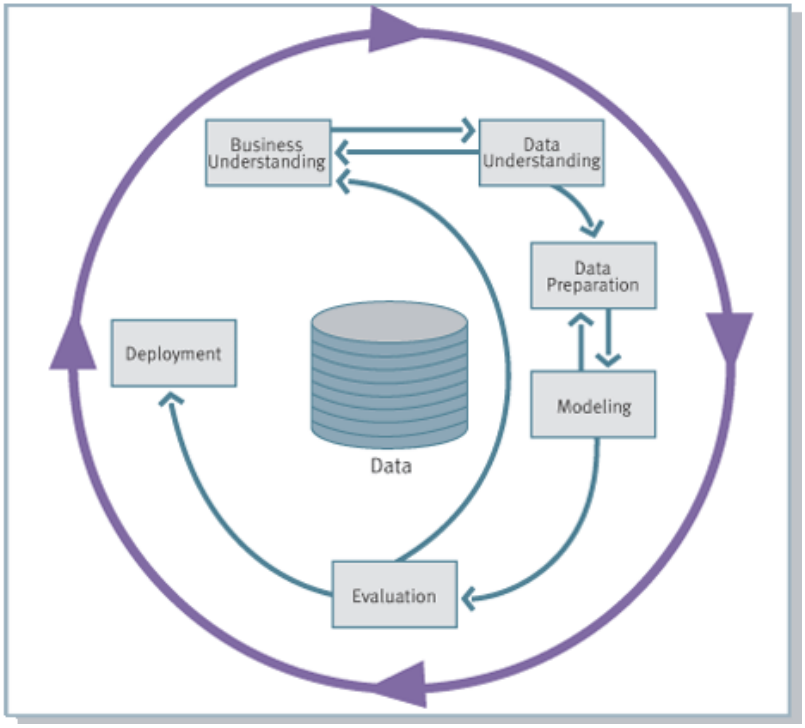

Fig. 1 CRISP-DM phases

Classification Algorithm Assessment

Before utilizing a model produced by a classification algorithm, it is assessed with respect to some criterion. The model will probably result in some errors therefore the data miner should take it into account while selecting a model (Cios, Pedrycz & Swiniarski, 2007). Accuracy, which is the percentage of instances that are correctly classified by the model, is the most commonly used decision criteria for model assessments (Han & Kamber, 2005).

However, there is also other criterion used to compare and evaluate the models. Berson defines the assessment concepts as accuracy, explanation and integration abilities (Berson, Smith & Thearling, 1999). Rokach introduces the comparison criterion as the generalization error of the model, the computational complexity that is the amount of CPU consumed by inducer, the comprehensibility that is the ability to

understand the model, the scalability that is the ability to run efficiently on larger databases, the robustness that is the ability to handle missing or noisy data, the stability that is the ability to produce repeatable results on different datasets and lastly the interestingness that is the ability to generate valid and new knowledge (Maimon & Rokach, 2008).

Data Preprocessing

Before implementing the classification algorithms it is recommended that the incomplete, noisy or inconsistent datasets are pre-processed to make the knowledge discovery process easier and more qualified. The most well known steps are summarization, cleaning, integrations and transformations, data and dimensionality reduction and discretization (Han & Kamber, 2005). Discretization and dimension reduction are within the scope of this study.

Data discretization techniques can be used to reduce the number of values for a given continuous variable by splitting the range of the variable into intervals. Binning, for example, is a type of discretization technique where variable is splitted into a particular number of bins. Dimension reduction is another pre-processing technique to obtain a reduced dataset representing the original dataset. The most commonly used dimension reduction technique is the PCA (Principal Component Analysis). “PCA searches for k n-dimensional orthogonal vectors that can best be used to represent the data where k<=n. The original data are thus projected onto a smaller space.” (Han & Kamber, 2005)

Literature Review

As data volume boosts in real life, it is becoming harder to make valuable and significant decisions, with respect to it. In those situations, Data Mining, that is used to extract the concealed knowledge from large amounts of data, is commonly used (Han & Kamber, 2005). The predictive power of the data mining classification algorithms has been appealing for many years. Many studies are concentrated on proposing a new classification model, comparing the models or important factors affecting the model's performance.

The literature contains many studies about algorithm comparisons. Quinlan states that it is not an easy task to declare that one algorithm is always superior to others, and links the abilities of models to task dependency. The study compares the decision tree with network algorithms and concludes that parallel type problems are not common for decision trees and sequential type problems are not suited to back-propagation (Quinlan, 1994). In an additional study, some algorithms as LARCKDNF, IEKDNF, LARC, BPRC and IE are compared on three tasks, and different results are stated for each task (Kaelbling, 1994). Hacker and Ahn have done another comparative experiment which is about eliciting user preferences. They compare many methods and recommend a new classifier called relative SVM, which outperforms others (Hacker & Ahn, 2009). The authors point at the useful data mining implications and try to understand "whether meaningful relationships can be found in the soil profile data at different locations". They use the data collected from the WA Department of Agriculture and Food (AGRIC) soils database and they also compare those data mining methods to existing statistical methods (Armstrong, Diepeveen & Maddern, 2007). The importance of feature selection is emphasised on a study where

the decision tree and regression methods are applied on a breastfeeding survey data (He et al., 2006). Another research implements Naïve Bayesian, Decision tree, KNN, NN and M5 to predict the lifetime prediction of metallic components, and it is stated that methods which can directly deal with continuous variables are performing better (Ge, Nayak, Xu & Li, 2006). AIRS algorithm is compared to other algorithms by Putten, Meng and Kok, and no significant evidence that it consistently outperforms the others has been found (Putten, Meng & Kok, 2008). Maindonald points at the difficulties and complexities about comparing the algorithms. He underlines the fact that users who are more expert with a specific model will have a tendency to have the best results with that model therefore the published performance results are very broad indicators and dependent on datasets. Moreover, he states the insufficiency of datasets from several of years to be compared about the changes in algorithm performances (Maindonald, 2006). Lastly Lim, Loh & Shih makes an experimental study about comparing classification type of algorithm accuracy and training times. They conclude in such a way that the mean of accuracies of algorithms are not significantly different from each other but the there is a huge difference between the mean of training times of classifiers (Lim, Loh & Shih, 2000)

In retrospect, some researchers tried to show the importance of datasets in classifications. A crucial point is introduced about the danger of using a single dataset for performance comparison, and tests are carried out for dynamic modifications of penalty and network architectures (Finnoff, Hergert & Zimmermann, 1995). A similar finding is stated as the performance results of learning algorithms are expected to deviate across different datasets; the study discussed data and implementation bias on time series datasets (Keogh & Kasetty, 2002).

The importance of implementation settings while running the algorithms have also been underlined by some authors in the literature. Keogh emphasizes the importance of implementation details such as parameter selections in algorithms and claims that algorithms should have few parameters or none (Keogh, Stefano & Ratanamahatana, 2004). Pitt points at a factor affecting the accuracy in their study that "the use of feature reduction algorithms on a large population survey database has shown that the use of the subset and attribute evaluation methods mostly results in an improvement in accuracy despite a reduction in the number of attributes" (Pitt & Nayak, 2007). And finally Howley has studied about the effects of data preprocessing steps on classifier accuracies. They compared the results of classifiers where no preprocessing step was applied, when techniques like normalisation or PCA was applied (Howley, Madden, O'Connell & Ryder, 2005).

As seen in the literature review, the data mining community is very interested in comparing different classification algorithms. As an example, Dogan and Tanrikulu proposes a comparative framework for evaluating classifier accuracies and they claim that classifier accuracies are not always the same on every dataset and performance is therefore significantly affected by dataset characteristics such as variable types or number of instances (Doğan & Tanrıkulu, 2010). This study is also concerned with classification performance and other factors affecting the accuracy with new perspectives and also concerned with other quality indicators such as the classifier complexity or robustness.

CHAPTER 2.

PURPOSE AND METHOD

The purpose of this study is to have a general idea about the accuracy and complexity of classification algorithms under given circumstances.

In literature, data mining community has been very interested in comparing classification type of algorithms but they usually compare the classifiers on a single dataset or they compare only a few of the algorithms not including the recent ones. It is not easy to find empirical results of how classifiers perform on different multiple datasets; therefore the basic concern about this study is the repetitive algorithm implementations on multiple datasets thus some idea about the effects of dataset characteristics on the performance can be derived from the study. The same concern is also valid for the complexity comparison that is the consumed cpu time by each classifier. Knowledge discovery process of data mining projects include the data pre-processing stage and recommends steps such as data cleaning, reductions, discretisations or component analysis if necessary. This study aims to find out if those data pre-processing activities have any effect on the classifier accuracy or model development time. Lastly the study aims to figure how robust the selected classifiers are. In order to understand their robustness, iterative implementations are done before and after cleaning noise in the datasets.

The details of the research questions and the methodological framework can be found in the following sections.

Research Questions

This study aims to compare the classification algorithm accuracies, complexities and robustness with respect to various datasets and implementation techniques. The research questions of this study are as follows:

1. Does implementing the same classification algorithm on multiple datasets and with different implementation techniques result in significantly different performance indicators?

2. Do the characteristics of the datasets affect the performance results of the classification algorithms?

3. Does binning the continuous numerical variables in the dataset into discreet intervals affect the classifier accuracy?

4. Does applying principal component analysis in the dataset affect the classifier accuracy?

5. Based on the results derived from the empirical results of this study (applying classifiers on various dataset with different implementation techniques), can a model to predict the performance of the classification algorithm be built?

6. Does implementing the same classification algorithm on multiple datasets and with different implementation techniques result in significantly different complexity?

7. Do the characteristics of the datasets affect the complexity of the classification algorithms?

8. Does binning the continuous numerical variables in the dataset into discreet intervals affect the classifier complexity?

9. Does applying principal component analysis in the dataset affect the classifier complexity?

10. Based on the results derived from the empirical results of this study (applying classifiers on various dataset with different implementation techniques), can a model to predict the complexity (consumed CPU time in seconds) of the classification algorithm be built?

11. Are the abilities of classifiers to handle missing or noisy data different (robustness)?

Method

In the implementation phase, 10 sample datasets have been used because research study is interested in application in multiple datasets. 13 classification algorithms have been selected to be implemented on the experimental datasets. WEKA (Waikato Environment for Knowledge Analysis) a popular suite of machine learning software has been used as a tool to run Naïve Bayesian, AIRS, Logistics, MLP, J48, AIRS2, AIRS2P, Clonogal and CSCA algorithms. SPSS has been used as a tool to run the Chaid, Ex-Chaid, CRT and Quest algorithms since they are available in SPSS. Data pre-processing steps have also been applied on the sample datasets. The

results of the implementations have been tabulated. Afterwards, a descriptive analysis and a one-way Anova test have been carried out to find answers to the first and sixth research questions. Correlation analysis has been conducted to answer the second, third, fourth, seventh, eighth and ninth research questions and lastly, regression models have been built to deal with the fifth and tenth research questions. In order to answer the eleventh research question, algorithms have been implemented on a noisy dataset before and after data cleaning and results containing the variances have been tabulated to see the robustness of the classifiers.

It is important to understand which statistical method is applied for answering which research question. The research questions of the thesis study can be classified in 3 groups. The first group of questions (Research Questions 1 and 6) are interested in finding some difference between algorithm types. The second group of questions (Research Questions 2, 3, 4, 7, 8, 9) are related to finding a possible relationship between a dependent variable (accuracy or complexity) and independent variables (dataset characteristics, dicretisation or PCA application). The last group of questions (Research Questions 5 and 10) are concerned with making up a dependent variable based on some independents. Once the nature of the research questions are deeply understood, it is crucial to choose the most appropriate statistical analysis method to find the answers. As Gamble states that the nature of the variables should be stated firstly. Are the variables numerical, ordinal, or nominal? Then the analyst needs to understand what is being looked for while choosing a test to use. For example, the statistical test could be helping about finding differences between some groups (difference tests) or relationship between variables (correlation tests) or a regression (regression tests) to make prediction. If difference tests are required, there is one more

step to decide on the necessary steps; is the difference looked for between or within groups? For example if it is a ‘between’ groups test then each participant belongs to only one of the groups; if it is a ‘within’ group test then each participant occurs in all of the groups. After deciding on variable type (numerical, ordinal, nominal), type of test (difference, correlation, regression), and within or between groups difference then the possible statistical tests come to the scene. Figure 2 shows this decision path very clearly (Gamble, 2001).

Considering the decision tree in Figure 2, the fact that all variables will be numerical after the experiments and the nature of the research questions of this study, it is obvious to see that the research questions 1 and 6 are trying to find any difference between algorithm types therefore conducting a one way Anova suits this need best. Research questions 2, 3, 4, 7, 8 and 9 are trying to find some correlations between dependent variables (accuracy or complexity) and independents (dataset or implementation specific attributes) therefore the best test seems to be Pearson’s Correlation. Finally, research questions 5 and 10 are trying to make a model to predict accuracy or complexities therefore a Linear Regression will fit this type of question perfectly. Figure 3 summarizes the methodological framework maintained during the research study.

Data collection

From the UCI Machine Learning Repository (UCI, 2010), sample datasets have been collected. The experimental datasets are Acute, Breast Cancer, CPU, Credits, Iris, Letters, Red wine, Segment, White wine and Wine. Table 1 summarises the attributes of each dataset.

Table 1. Dataset Characteristics

Dataset Name	Number of Variables	Number of Nominal Variables	Number of Numerical Variables	Target Class Types	Number of Instances
Acute	7	6	1	2	120
Breast Cancer	9	0	9	2	684
Cpu	6	0	6	3	210
Credits	15	9	6	2	653
Iris	4	0	4	3	150
Letters	16	0	16	26	20000
Segment	19	0	19	7	1500
Wineall	11	0	11	7	6497
Wine red	11	0	11	6	1599
Wine white	11	0	11	7	4898

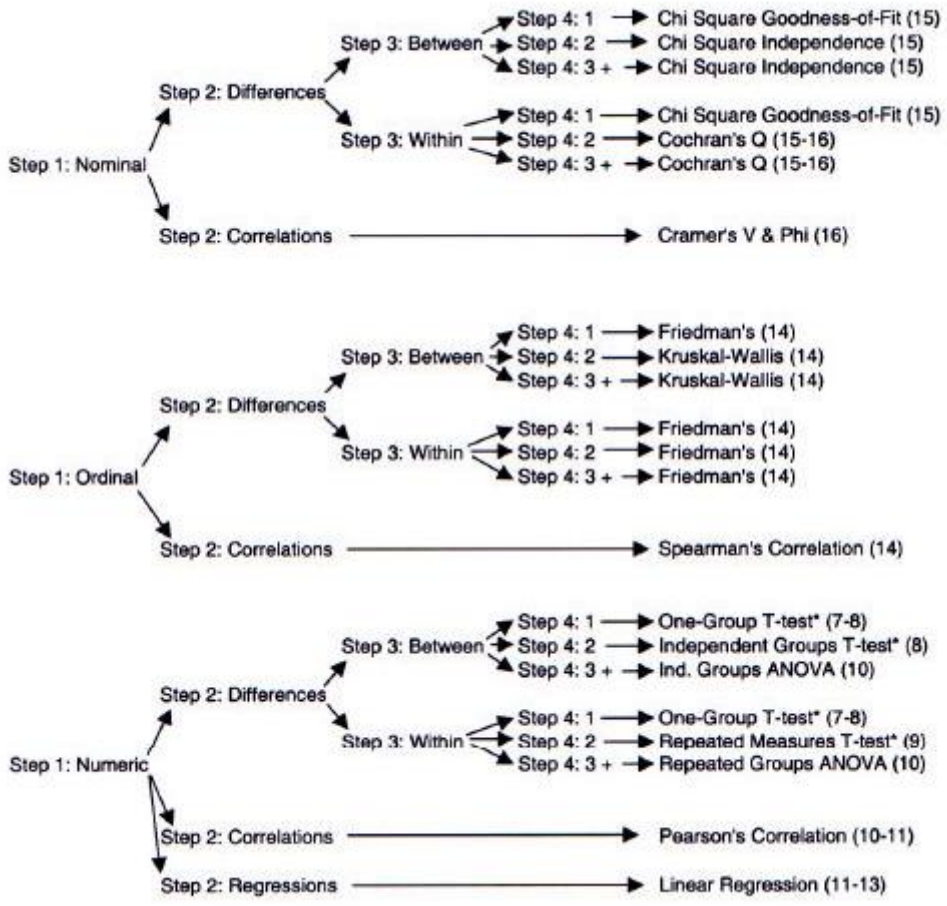

Fig. 2 Determining which test to use

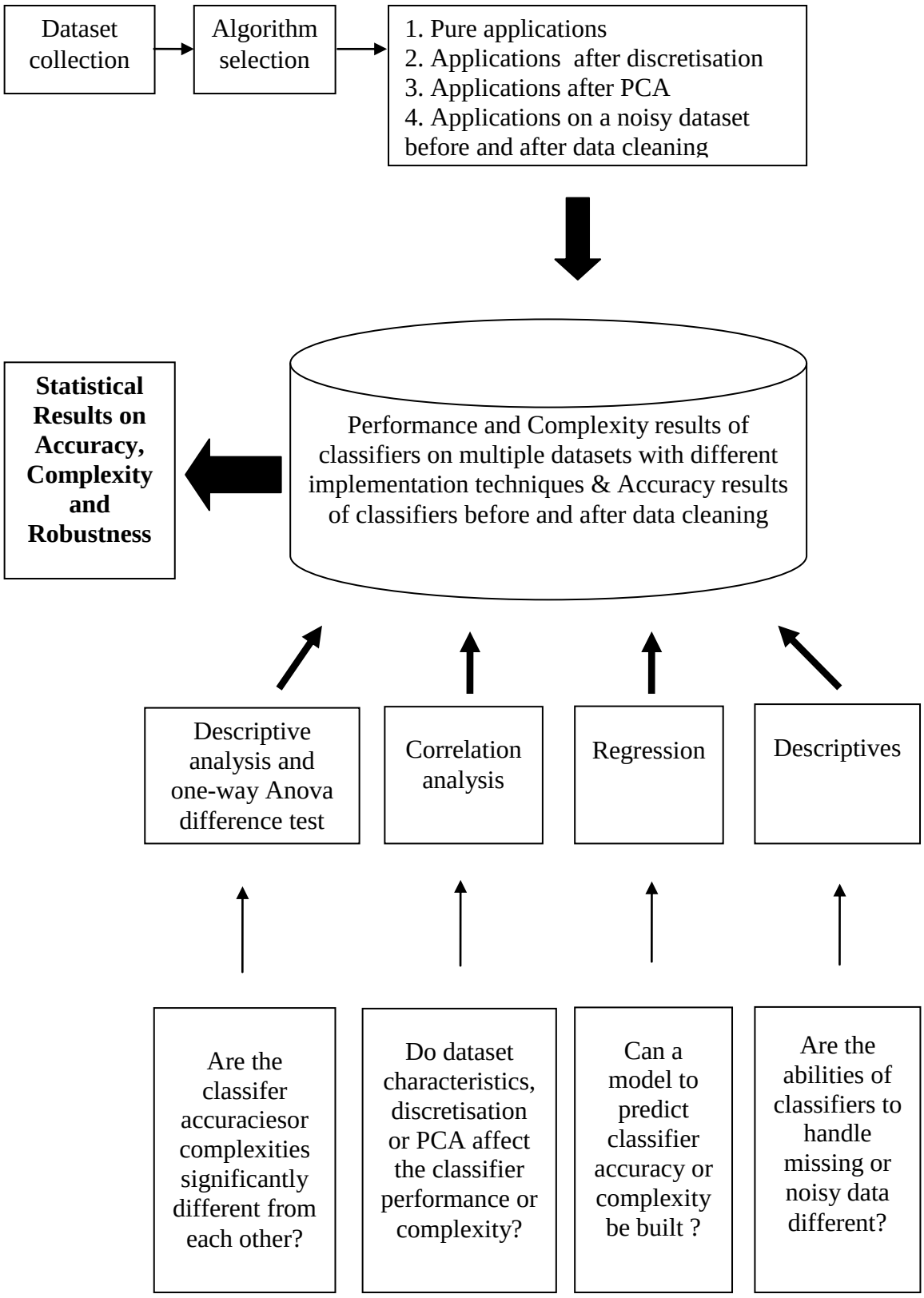

Fig. 3 Methodological framework

Algorithms

13 classification algorithms have been selected among many existing classification models within the scope of this study. The selected algorithms are Naïve Bayesian algorithm; J48, CHAID, Ex-CHAID, CRT and Quest decision tree algorithms; Multilayer perceptron (MLP); AIRS, AIRS2, AIRS2P, CSCA and Clonalg Artificial Immune Recognition Systems algorithms and Logistics Regression.

The Naïve Bayesian algorithm identifies the classification problem in accordance with to probabilistic phrases, and provides statistical methods to categorize the instances with respect to probabilities (Cios, Pedrycz & Swiniarski, 2007).

In Decision Tree algorithms, the classification procedure is condensed by a tree. After the model is constructed, it is applied to the whole database (Dunham, 2002). The CHAID algorithm cultivates the tree by locating the optimal splits until the stopping criteria is encountered with respect to the chi-squares (Berson, 1999). Chaid can deal with missing values and the outputs of the target function are discrete (Mitchell, 1997). Splitting and stopping steps in Exhaustive CHAID algorithm that was proposed by Biggs in 1991 are the same as those in CHAID. Merging step uses an exhaustive search process to merge any similar pair until a single pair remains. Also like CHAID, only nominal or ordinal categorical predictors are allowed, continuous predictors are first converted into ordinal predictors before using the algorithm (CART, 2010). J48 algorithm is a version of an earlier algorithm developed by J. Ross Quinlan, the very popular C4.5. J48 employs two pruning methods. The first is known as sub-tree replacement and the second is known as sub-tree raising (SPSS, 2010). QUEST is a binary-split decision tree algorithm for classification and data mining.

The objective of QUEST is similar to that of the CART algorithm. It uses an unbiased variable selection technique by default, uses imputation instead of surrogate splits to deal with missing values and can easily handle categorical predictor variables with many categories (Shih, 1997). CRT (Classification and Regression Trees) is another decision tree classifier which uses binary splits, first grows then prunes and uses Gini Index as splitting criteria and surrogates missing values (MONK, 2010). Badulescu points at the difficulty in selecting the best attribute while splitting the decision tree at the model induction phase and compares the performance of 29 different splitting measures claiming that the FSS Naïve Bayesian splits the attributes best (Badulescu, 2007).

Multilayer Perceptron is a type of artificial neural network algorithm which considers the human brain as the modelling tool (Cios, Pedrycz & Swiniarski, 2007). It provides a generic model for learning real, discrete and vector target values. The ability to understand the concealed model is tough and training times may be extensive (Mitchell, 1997).

As the human natural immune system differentiates and recalls the intruders, the AIRS algorithm is a cluster-based approach that understands the structure of the data and performs a k-nearest neighbour search. AIRS2 and AIRS2P are the extensions of existing AIRS algorithm with some technical differences (Putten, Meng & Kok, 2009). Another artificial immune system technique that is inspired by the functioning of the clonal selection theory of acquired immunity is Clonalg algorithm. It is inspired by "maintenance of a specific memory set, selection and cloning of most stimulated antibodies, death of non-stimulated antibodies, affinity maturation

(mutation), re-selection of clones proportional to affinity with antigen, and generation and maintenance of diversity". A variant implementation of Clonalg is called as Clonal Selection Classifier Algorithm (CSCA) that aims to maximize classification accuracy and minimize misclassification accuracy (Brownlee, 2005).

Logistic regression utilizes of independent variables to predict the probability of events by fitting the data to a logistic curve (Hand, Mannila,& Smyth, 2001). Each algorithm can make use of both numerical and categorical variables as inputs. They can handle target classes with more than two class types. Algorithms can also be referred to as classifiers or models.

Implementation

Before the implementation of algorithms, datasets were firstly cleaned from missing, noisy and incorrect data. Firstly, according to the missing data analysis, missing data have been removed from the datasets. Datasets were also cleaned to remove noisy data. Unnecessary space characters or other spelling mistakes were also cleaned in the datasets.

All 10 datasets (Acute, Breast Cancer, CPU, Credits, Iris, Letters, Red wine, Segment, White wine and Wine) have been used to run the 13 classification algorithms (Naïve Bayesian, CHAID, Ex-CHAID, CRT, Quest, J48, MLP, AIRS, AIRS2, AIRS2P, CSCA, Clonalg and Logistics algorithms). For all algorithms, splitting the data into train and test splits has been selected as the validation method. 66% of the data has been set as the training part and the rest has been set as the testing part since 1/3 of the dataset is commonly suggested to be split as the testing part. Then 10-fold

cross validation has been implemented on the same datasets for the selected algorithms. In other words, both splitting and 10-fold cross validation methods have been applied. This stage of the experiment referred as “pure implementation” resulted in 260 (10 datasets * 13 algorithms * 2 Validation methods) rows of accuracy and complexity values.

After the pure implementation phase, all continuously numerical variables in the datasets have been put into binned intervals within +/-1 standard deviation and saved as new variables. 13 algorithms again have been implemented on the pre-processed datasets made up of dicretised numerical variables. The second stage of the experiment referred as “after discretisation” resulted in another 260 rows of accuracy and complexity values (10 datasets * 13 algorithms * 2 Validation methods).

Following the second stage, principal components analysis has been conducted on the dataset. Components with eigenvalues over 1 have been set as components and saved as new variables. 13 algorithms again have been implemented on the pre-processed datasets that are made up of principal components. The third stage of the experiment referred as “after PCA” resulted in another 260 rows of accuracy and complexity values (10 datasets * 13 algorithms * 2 Validation methods).

In total, a dataset called as “results” have been obtained by those 780 rows of performance and complexity values derived from those 3 stages of the experiment. The 780 row dataset has been used to answer the first 10 research questions.

In order to answer the final research question, the ‘breast cancer’ dataset has been selected and algorithms have been run on it before and after cleaning the noisy instances and then the results have been compared.

WEKA and SPPS are the main components to run the selected algorithms. MS Excel also has been utilised to make some data pre-processing activities before implentations. SPPS has also been used to conduct all the statistical tests such as correlations, regression, Anova or descriptives. The result tables and figures are mostly extracted from the SPSS output files.

The environmental facts are also important for the nature of the experiments. All data mining algorithms and statistical tests have been conducted on a personal computer with the following configuration:

- Microsoft XP Professional Operating System with Service Pack 3
- Intel Core 2 Duo CPU
- 2.10 GHz
- 3 GB RAM
- 150 GB Harddisk

CHAPTER 3.

EXPERIMENTAL RESULTS AND DISCUSSIONS

In this section the performance and complexity results of each algorithm in each case will be discussed and research questions will be answered accordingly.

The percentage of instances correctly classified helps calculating the accuracy (Dunham, 2002) that is referred as the performance of classifiers throughout the study. Costs for wrong assignment, in other words, misclassification costs are not within the scope of this study.

The accuracy values of the multiple dataset implementations according to each classifier can be seen in Tables 2-7.

As Rokach and Maimon describes, complexity of a classifier is the computational resources used to train or test the model (Maimon & Rokach, 2008). In this study, complexity is referred as the CPU amount used by the classifiers during model building, in other words, the time observed in seconds to generate the classifier. The higher the values of time spent during modelling, the more complex the classifier is.

The complexity values of the multiple dataset implementations according to each classifier can be seen in Tables 8-13.

Research question 1: Does implementing the same classification algorithm on multiple datasets and with different implementation techniques result in different performance indicators?

Based on the findings of the empirical study, it can be seen in Tables 14-17 that the same classifier is not the best one for all datasets and always outperforms the other classifiers. For each dataset the best predictive classifier has been defined for each stage of the experiment. As Dogan and Tanrıkulu also claims in their study, a classifier cannot be said to outperform the others in every dataset (Doğan & Tanrıkulu, 2010).

According to Table 14, the overall best accuracy is obtained as 100% in the "Acute" dataset. The classifiers producing that rate of accuracy are Logistics, all immune system algorithms, MLP, Naive Bayesian, and only J48 from Decision tree algorithms.

Table 15 displays the detailed accuracy results for the best result cases of each dataset in pure implementations step. Logistics has the best performance for 'Acute' dataset; AIRS has the best accuracy for 'Acute', 'Iris', 'Letters' and 'Wine all' datasets; MLP has the best accuracy for 'Acute', 'Cpu', 'Segment' and 'Wine red' datasets. CHAID produces better performance only for the 'Credits' dataset; J48 has the best accuracy for 'Acute', 'Letters' and 'Wine white' datasets; AIRS2 has the best accuracy for 'Acute', and 'Segment' datasets; AIRS2P has the best accuracy for 'Acute' and 'Iris' datasets; CSCA has the best accuracy for only 'Acute' dataset; Clonalg has the best accuracy for 'Acute' and 'Breast cancer' datasets. Interestingly, Naïve Bayesian, Ex-CHAID, CRT and Quest have never produced the best result for a dataset from those classifiers which may mean they cannot handle continuous integer

variables and dense dimensionality as well as MLP, logistics or other immune system algorithms.

Table 16 displays the detailed accuracy results for the best result cases of each dataset after discretisation step. MLP has the best accuracy for ‘Acute’, ‘Cpu’ ,‘Iris’ and ‘Segment’ datasets; J48 has the best accuracy for ‘Acute’, ‘Iris’ , ‘Letters’, ‘Wine all, ‘Wine red’ and ‘Wine white’ datasets; AIRS2 has the best accuracy for ‘Acute’ and ‘Breast cancer’ datasets; CHAID has the best accuracy for again ‘Credits’ dataset; Clonalg has the best accuracy for ‘Acute’ and ‘Breast cancer’ datasets; AIRS, Logistics, Naive Bayesian, AIRS2P and CSCA has the best accuracy for only ‘Acute’ dataset. When the continuous variables are binned into intervals, J48 and Naive Bayesian started to predict better, it may depend on its ability to handle discrete values better. However Ex-CHAID, CRT or Quest still cannot predict as well as others.

Table 17 displays the detailed accuracy results for the best result cases of each dataset after PCA step. J48 has the best accuracy for ‘Acute’, ‘Letters’, ‘Segment’, ’Wine all’, ‘Wine white’ datasets; MLP has the best accuracy for ‘Acute’ and ‘Iris’ datasets; CRT has the best accuracy for ‘Breast cancer’ and ‘Wine red’ dataset; Naive Bayesian has the best accuracy for ‘Acute’, ‘Cpu’ and ‘Credits’ datasets; Logistics, AIRS, AIRS2, AIRS2P, Clonalg, CSCA has the best accuracy for only ‘Acute’ datasets. After PCA application, Naive Bayesian and CRT started to predict better thus they may be handling less amount of data and dimensions better. Decision tree algorithms other than CRT and J48 did not predict the best in any dataset.

Table 2. Accuracy Results / Pure Implementation and 10-Fold Cross Validation

	Acute	Breast cancer	CPU	Credit	Iris	Letter	Segment	Wine all	Wine red	Wine white
AIRS	100.0	96.2	94.7	82.5	95.3	86.5	88.0	86.5	51.3	48.3
AIRS2	100.0	97.1	97.1	85.6	95.3	83.8	86.5	48.9	51.4	51.1
AIRS2P	100.0	96.8	95.7	84.2	94.7	84.7	89.5	49.3	54.3	51.0
CHAID	91.7	93.0	80.9	86.4	66.7	53.8	78.7	53.8	59.4	54.6
Clonalg	98.3	95.9	90.0	53.4	95.3	12.5	64.7	38.0	45.5	39.8
CRT	91.7	92.7	80.9	87.4	66.7	36.8	89.3	56.4	62.0	54.1
CSCA	100.0	96.3	93.3	65.4	95.3	67.7	84.6	46.5	50.9	45.7
ExCHAID	91.7	93.0	80.9	86.4	66.7	54.1	78.7	54.8	59.1	54.7
J48	100.0	96.0	96.2	85.3	96.0	87.9	95.7	58.7	62.0	58.2
Logistics	100.0	96.8	97.1	86.1	96.0	77.4	95.9	77.4	59.8	53.7
MLP	100.0	96.0	97.2	82.7	97.3	82.2	96.7	54.9	60.7	55.2
Naïve bayes	95.8	96.3	89.5	78.3	96.0	64.0	81.1	64.0	55.0	44.3
Quest	85.0	91.2	74.6	84.8	66.7	23.7	83.0	53.2	46.3	52.0

Table 3. Accuracy Results / Pure Implementation and 66% Train-Test Split

	Acute	Breast cancer	CPU	Credit	Iris	Letter	Segment	Wine all	Wine red	Wine white
AIRS	100.0	96.1	94.4	78.5	98.0	83.6	87.1	83.6	51.8	46.0
AIRS2	100.0	96.6	90.3	82.5	94.1	82.1	97.3	47.4	50.7	49.2
AIRS2P	100.0	97.0	94.4	82.5	98.0	82.1	87.5	49.9	50.9	49.0
CHAID	61.9	93.2	74.2	88.1	23.5	54.5	57.4	54.5	57.7	53.9
Clonalg	100.0	97.4	86.1	58.3	96.1	10.4	64.3	43.3	41.7	28.3
CRT	58.1	87.7	81.2	81.1	30.0	25.4	81.8	52.5	44.4	53.8
CSCA	100.0	97.0	83.3	59.6	96.1	64.6	84.7	43.5	51.3	42.9
ExCHAID	60.4	94.0	76.1	85.9	29.5	54.4	57.4	49.5	54.2	51.3
J48	100.0	95.7	94.4	82.1	96.1	86.5	95.1	56.6	58.5	56.3
Logistics	100.0	97.0	94.4	83.0	92.2	77.0	95.5	77.0	57.9	52.3
MLP	100.0	96.6	97.2	79.7	98.0	82.8	97.3	56.3	62.5	51.2
Naïve bayes	95.1	96.1	87.3	75.3	94.1	64.4	81.4	64.4	52.0	43.2
Quest	58.1	87.7	76.3	81.1	30.0	25.4	80.6	52.5	44.4	44.0

Table 4. Accuracy Results / After Discretisation and 10-Fold Cross Validation

	Acute	Breast cancer	CPU	Credit	Iris	Letter	Segment	Wine all	Wine red	Wine white
AIRS	100.0	96.5	94.7	83.0	76.7	82.9	85.0	52.6	55.7	53.7
AIRS2	100.0	95.9	97.1	83.9	82.7	75.2	83.5	49.9	53.9	51.5
AIRS2P	100.0	95.8	95.7	84.1	86.0	75.1	85.1	50.3	53.2	51.6
CHAID	86.7	93.0	80.9	86.4	66.7	40.1	79.1	53.1	58.3	52.9
Clonalg	98.3	94.7	90.0	65.4	84.0	10.0	60.3	42.6	49.6	40.9
CRT	86.7	93.0	81.2	86.4	66.7	35.3	75.5	53.5	58.4	52.9
CSCA	100.0	96.5	93.3	81.2	86.7	62.8	86.7	50.5	54.3	50.5
ExCHAID	83.3	93.0	76.1	86.4	66.7	40.1	79.1	52.2	58.3	52.9
J48	100.0	95.5	96.2	85.8	90.7	83.6	89.5	55.7	59.3	55.5
Logistics	100.0	96.0	97.1	85.3	90.7	67.5	86.5	51.8	59.0	52.2
MLP	100.0	96.2	97.2	84.2	89.3	71.9	88.7	51.5	57.5	53.9
Naïve bayes	95.0	95.9	89.5	83.0	86.0	58.1	79.0	48.3	57.1	46.8
Quest	86.7	91.2	75.1	81.0	66.7	31.1	71.1	52.3	58.9	52.8

Table 5. Accuracy Results / After Discretisation and 66% Train-Test Split

	Acute	Breast cancer	CPU	Credit	Iris	Letter	Segment	Wine all	Wine red	Wine white
AIRS	100.0	96.1	94.4	81.2	76.5	81.5	86.3	50.8	54.2	52.7
AIRS2	100.0	97.4	90.3	81.2	74.5	73.3	82.9	48.4	53.7	51.4
AIRS2P	100.0	96.1	94.4	83.0	88.2	72.6	85.1	48.5	55.0	52.3
CHAID	55.6	93.2	74.2	88.1	25.5	39.4	80.4	51.8	58.1	51.9
Clonalg	100.0	97.4	86.1	75.8	82.4	11.1	68.4	41.4	46.0	32.8
CRT	56.8	94.2	81.2	86.3	28.6	33.0	73.3	51.5	57.1	49.8
CSCA	100.0	95.3	83.3	80.7	84.3	57.3	85.5	48.4	52.4	48.7
ExCHAID	55.6	91.9	76.1	85.9	55.6	39.1	75.2	49.3	57.3	51.9
J48	100.0	96.1	94.4	85.7	94.1	82.8	89.0	52.7	59.6	55.6
Logistics	100.0	97.0	94.4	82.5	94.1	66.5	88.4	52.6	56.8	51.1
MLP	100.0	97.0	97.2	76.6	94.1	70.2	89.8	51.7	58.1	49.5
Naïve bayes	100.0	94.8	87.3	82.5	86.3	57.0	80.0	48.6	55.9	46.3
Quest	58.1	87.7	76.3	83.4	25.0	28.6	71.5	51.7	57.4	51.1

Table 6. Accuracy Results / After PCA and 10-Fold Cross Validation

	Acute	Breast cancer	CPU	Credit	Iris	Letter	Segment	Wine all	Wine red	Wine white
AIRS	100.0	92.8	88.5	75.5	78.0	50.7	74.3	41.1	50.0	41.1
AIRS2	100.0	92.2	89.0	77.3	82.7	50.1	74.2	40.8	51.6	40.8
AIRS2P	100.0	94.3	89.0	77.0	80.0	54.5	78.0	44.2	53.6	44.5
CHAID	58.3	94.7	85.6	81.5	33.3	42.2	68.7	50.4	56.7	50.8
Clonalg	100.0	97.2	90.0	70.1	81.3	11.4	52.9	41.5	52.0	41.5
CRT	58.3	97.5	86.6	81.6	66.7	31.0	68.2	50.3	58.4	51.6
CSCA	100.0	97.2	87.1	79.8	84.0	52.0	79.0	47.6	56.3	47.6
ExCHAID	58.3	94.7	85.6	81.5	33.3	42.1	69.5	49.5	56.8	51.3
J48	100.0	97.5	88.5	80.7	84.7	64.7	84.7	51.9	56.8	51.9
Logistics	100.0	96.3	90.4	82.5	84.7	38.3	72.2	48.4	57.8	48.4
MLP	100.0	97.2	91.4	81.3	86.7	52.4	77.4	48.1	58.1	48.1
Naïve bayes	100.0	96.9	90.4	81.5	84.0	37.3	66.7	48.6	56.7	48.6
Quest	58.3	96.9	86.1	81.5	33.3	27.6	64.5	43.7	42.6	44.9

Table 7. Accuracy Results / After PCA and 66% Train-Test Split

	Acute	Breast cancer	CPU	Credit	Iris	Letter	Segment	Wine all	Wine red	Wine white
AIRS	100.0	91.8	84.7	74.9	82.4	49.8	74.7	41.9	52.4	41.9
AIRS2	100.0	97.0	90.3	71.7	86.3	48.5	73.5	41.4	51.7	41.4
AIRS2P	100.0	94.4	83.3	78.0	76.5	52.9	80.6	43.1	55.7	44.5
CHAID	55.6	95.3	74.2	71.5	25.5	35.0	58.9	47.8	52.2	47.0
Clonalg	100.0	96.6	93.1	76.2	80.4	10.6	58.6	44.3	46.7	44.3
CRT	58.1	97.8	81.2	80.7	28.6	29.4	73.3	47.3	52.0	48.2
CSCA	100.0	97.0	81.9	75.8	84.3	49.1	78.4	46.7	55.0	46.7
ExCHAID	56.8	95.7	76.1	75.3	24.4	38.0	56.9	44.9	53.7	49.5
J48	100.0	97.0	91.7	79.4	84.3	62.2	83.9	50.1	55.9	50.1
Logistics	100.0	97.0	86.1	80.7	86.3	38.1	73.9	48.9	57.7	48.9
MLP	100.0	97.4	93.1	80.2	88.2	52.8	74.7	48.7	54.8	48.7
Naïve bayes	100.0	96.6	93.1	84.3	86.3	37.1	66.3	48.6	55.7	48.6
Quest	46.7	95.6	76.3	81.6	25.0	25.8	56.6	48.1	41.0	45.1

Table 8. Complexity Results / Pure Implementation and 10-Fold Cross Validation

	Acute	Breast cancer	CPU	Credit	Iris	Letter	Segment	Wine all	Wine red	Wine white
AIRS	0.1	0.6	0.2	0.7	0.3	290.7	3.0	18.6	2.2	11.7
AIRS2	0.2	0.4	0.2	466.7	0.0	180.9	1.6	15.4	1.5	9.4
AIRS2P	0.1	0.3	0.1	696.6	0.2	145.3	1.3	11.8	1.0	6.7
CHAID	2.0	2.0	1.0	2.0	2.0	50.0	3.0	8.0	2.0	6.0
Clonalg	0.1	0.2	0.1	0.9	0.1	12.2	1.1	2.4	0.6	1.7
CRT	3.0	8.0	6.0	8.0	4.0	93.0	35.0	62.0	32.0	40.0
CSCA	0.0	2.6	0.3	15.6	0.0	14210.3	17.8	291.7	16.1	159.2
ExCHAID	2.0	2.0	1.0	1.0	1.0	87.0	3.0	8.0	3.0	7.0
J48	0.0	0.0	0.1	0.0	0.0	5.0	0.2	1.2	0.3	0.9
Logistics	0.0	0.0	0.1	0.1	0.0	427.7	35.0	5.2	0.9	4.1
MLP	0.4	1.9	0.6	23.0	0.4	1966.8	20.1	101.7	12.5	45.9
Naïve bayes	0.0	0.0	0.0	0.0	0.0	0.3	0.1	0.0	0.1	0.1
Quest	2.0	2.0	1.0	2.0	1.0	13.0	3.0	5.0	3.0	2.0

Table 9. Complexity Results / Pure Implementation and 66% Train-Test Split

	Acute	Breast cancer	CPU	Credit	Iris	Letter	Segment	Wine all	Wine red	Wine white
AIRS	0.1	0.7	0.2	0.7	0.2	1460.4	2.9	18.1	2.2	11.8
AIRS2	0.2	3.4	0.1	1432.0	0.1	683.7	19.2	15.3	1.4	9.4
AIRS2P	0.1	0.5	0.1	961.5	0.1	686.2	1.2	11.5	1.0	6.8
CHAID	1.0	1.0	1.0	1.0	1.0	12.0	2.0	3.0	2.0	3.0
Clonalg	0.1	0.2	0.1	1.2	0.1	36.6	0.8	2.3	0.7	1.8
CRT	2.0	2.0	1.0	2.0	2.0	13.0	8.0	6.0	4.0	7.0
CSCA	0.0	2.8	0.3	13.5	0.0	21296.4	18.4	434.2	15.1	164.5
ExCHAID	1.0	1.0	1.0	1.0	1.0	14.0	2.0	4.0	2.0	3.0
J48	0.0	0.0	0.0	0.0	0.0	16.6	0.1	1.3	0.1	0.8
Logistics	0.0	0.0	0.1	0.1	0.1	365.0	46.7	5.2	0.7	3.8
MLP	0.3	1.8	0.6	49.3	0.4	981.6	19.2	105.1	21.7	56.9
Naïve bayes	0.0	0.0	0.0	0.0	0.0	0.3	0.0	0.1	0.0	0.1
Quest	1.0	1.0	1.0	1.0	1.0	3.0	2.0	2.0	2.0	3.0

Table 10. Complexity Results / After Discretisation and 10-Fold Cross Validation

	Acute	Breast cancer	CPU	Credit	Iris	Letter	Segment	Wine all	Wine red	Wine white
AIRS	0.1	0.6	0.2	0.7	0.3	3355.5	3.8	136.5	11.2	56.0
AIRS2	0.2	0.5	0.2	358.1	0.1	61.7	1.6	87.6	4.9	48.1
AIRS2P	0.1	0.3	0.1	556.1	0.2	1955.1	1.3	51.1	3.8	36.4
CHAID	2.0	1.0	2.0	1.0	1.0	12.0	2.0	3.0	2.0	3.0
Clonalg	0.1	0.4	0.1	1.1	0.0	61.7	0.8	18.8	1.5	4.5
CRT	2.0	7.0	8.0	17.0	2.0	105.0	60.0	52.0	26.0	40.0
CSCA	0.0	2.5	0.3	9.7	0.2	47240.6	16.2	2862.8	52.6	639.6
ExCHAID	1.0	1.0	2.0	1.0	1.0	14.0	2.0	3.0	2.0	3.0
J48	0.0	0.1	0.1	0.1	0.0	47240.0	0.1	1.9	0.4	1.6
Logistics	0.0	0.2	0.1	0.2	0.1	3863.3	3.6	18.1	1.4	9.3
MLP	0.3	1.7	0.6	26.1	0.5	2518.8	19.3	102.9	46.2	169.8
Naïve bayes	0.0	0.1	0.0	0.0	0.0	1.8	0.1	0.1	0.0	0.1
Quest	1.0	1.0	1.0	11.0	1.0	13.0	2.0	4.0	3.0	3.0

Table 11. Complexity Results / After Discretisation and 66% Train-Test Split

	Acute	Breast cancer	CPU	Credit	Iris	Letter	Segment	Wine all	Wine red	Wine white
AIRS	0.1	0.5	0.2	1.0	0.2	3058.2	3.6	64.6	9.5	71.2
AIRS2	0.3	0.5	0.1	869.0	0.1	2108.3	1.6	45.7	5.3	39.6
AIRS2P	0.1	0.3	0.1	556.1	0.1	2007.1	1.2	64.1	3.6	30.3
CHAID	1.0	1.0	1.0	1.0	1.0	3.0	1.0	2.0	1.0	2.0
Clonalg	0.0	0.4	0.1	0.8	0.1	60.2	0.8	7.4	1.7	3.7
CRT	1.0	1.0	3.0	2.0	1.0	13.0	8.0	6.0	3.0	7.0
CSCA	0.0	2.6	0.3	10.4	0.2	45713.7	16.0	2997.1	51.1	652.5
ExCHAID	1.0	1.0	1.0	1.0	1.0	4.0	1.0	2.0	1.0	2.0
J48	0.0	0.0	0.0	0.0	0.0	29.6	0.2	4.3	0.2	2.4
Logistics	0.0	0.1	0.1	0.1	0.0	3748.6	3.6	9.1	1.4	7.5
MLP	0.3	1.8	0.6	44.5	0.3	1856.4	18.4	172.3	29.6	75.3
Naïve bayes	0.0	0.1	0.0	0.0	0.0	0.9	0.0	0.1	0.0	0.0
Quest	1.0	1.0	1.0	1.0	1.0	3.0	1.0	2.0	2.0	2.0

Table 12. Complexity Results / After PCA and 10-Fold Cross Validation

	Acute	Breast cancer	CPU	Credit	Iris	Letter	Segment	Wine all	Wine red	Wine white
AIRS	0.1	2.6	0.2	2.2	0.2	2066.7	1.9	269.5	19.7	39.2
AIRS2	0.1	0.3	0.1	0.7	0.0	998.6	0.7	201.3	6.7	48.1
AIRS2P	0.2	0.2	0.1	1.6	0.1	816.3	0.5	166.3	5.7	52.0
CHAID	1.0	0.5	1.0	1.0	1.0	15.0	3.0	3.0	2.0	2.0
Clonalg	0.0	0.5	0.1	0.5	0.0	28.5	0.6	33.1	1.5	7.6
CRT	3.0	3.0	2.0	15.0	2.0	61.0	12.0	34.0	16.0	30.0
CSCA	0.0	5.2	0.2	11.1	0.2	37362.0	11.4	3796.4	114.0	223.6
ExCHAID	1.0	0.5	1.0	1.0	1.0	19.0	1.0	4.0	2.0	2.0
J48	0.0	0.0	0.0	0.0	0.0	36.6	0.1	12.3	0.5	1.5
Logistics	0.0	0.1	0.1	0.0	0.0	1135.2	0.6	11.2	1.4	3.1
MLP	0.2	0.5	0.3	1.0	0.3	446.1	6.3	50.2	9.6	45.9
Naïve bayes	0.1	0.1	90.4	0.0	0.0	0.4	0.0	0.1	0.3	0.0
Quest	1.0	0.8	1.0	1.0	1.0	90.0	1.0	3.0	2.0	2.0

Table 13. Complexity Results / After PCA and 66% Train-Test Split

	Acute	Breast cancer	CPU	Credit	Iris	Letter	Segment	Wine all	Wine red	Wine white
AIRS	0.1	1.4	0.2	2.1	0.3	2113.4	1.9	174.5	11.0	43.8
AIRS2	0.0	0.4	0.1	1.1	0.0	1021.4	0.7	206.8	5.3	54.5
AIRS2P	0.1	0.3	0.1	1.0	0.0	822.9	0.6	176.9	6.8	51.6
CHAID	1.0	0.4	0.9	0.8	0.9	6.0	1.0	2.0	1.0	1.0
Clonalg	0.0	0.2	0.1	0.5	0.0	33.2	0.3	9.9	1.5	9.3
CRT	0.5	1.0	1.0	2.0	1.0	9.0	2.0	4.0	2.0	3.0
CSCA	0.0	4.5	0.2	10.8	0.1	33975.8	11.3	3715.3	125.9	2840.0
ExCHAID	0.5	0.4	0.9	0.8	0.9	7.0	1.0	2.0	1.0	1.0
J48	0.0	0.0	0.0	0.0	0.0	45.8	0.2	9.4	0.5	1.3
Logistics	0.0	0.0	0.0	0.0	0.0	3659.1	0.6	26.2	1.4	3.3
MLP	0.2	0.5	0.3	1.1	0.3	559.5	6.2	25.1	8.1	63.5
Naïve bayes	0.0	0.0	0.0	0.0	0.0	0.2	0.0	0.0	0.0	0.0
Quest	0.7	0.4	0.9	0.8	1.0	12.0	1.0	1.0	1.0	1.0

Table 14. Overall Best Accuracy Results

Dataset	Algorithm	Validation Method	Integer Variables Binned	PCA	Performance
Acute	Logistics	10fold	NO	N	100
Acute	AIRS	10fold	NO	N	100
Acute	MLP	10fold	NO	N	100
Acute	J48	10fold	NO	N	100
Acute	AIRS2	10fold	NO	N	100
Acute	AIRS2P	10fold	NO	N	100
Acute	CSCA	10fold	NO	N	100
Acute	logistics	traintestsplit	NO	N	100
Acute	AIRS	traintestsplit	NO	N	100
Acute	MLP	traintestsplit	NO	N	100
Acute	J48	traintestsplit	NO	N	100
Acute	AIRS2	traintestsplit	NO	N	100
Acute	AIRS2P	traintestsplit	NO	N	100
Acute	CSCA	traintestsplit	NO	N	100
Acute	Clonalg	traintestsplit	NO	N	100
Acute	Logistics	10fold	YES	N	100
Acute	AIRS	10fold	YES	N	100
Acute	MLP	10fold	YES	N	100
Acute	J48	10fold	YES	N	100
Acute	AIRS2	10fold	YES	N	100
Acute	AIRS2P	10fold	YES	N	100
Acute	CSCA	10fold	YES	N	100
Acute	Naïve bayes	traintestsplit	YES	N	100
Acute	Logistics	traintestsplit	YES	N	100
Acute	AIRS	traintestsplit	YES	N	100
Acute	MLP	traintestsplit	YES	N	100
Acute	J48	traintestsplit	YES	N	100
Acute	AIRS2	traintestsplit	YES	N	100
Acute	AIRS2P	traintestsplit	YES	N	100
Acute	CSCA	traintestsplit	YES	N	100
Acute	Clonalg	traintestsplit	YES	N	100
Acute	Naïve bayes	10fold	YES	Y	100
Acute	Logistics	10fold	YES	Y	100
Acute	AIRS	10fold	YES	Y	100
Acute	MLP	10fold	YES	Y	100
Acute	J48	10fold	YES	Y	100
Acute	AIRS2	10fold	YES	Y	100
Acute	AIRS2P	10fold	YES	Y	100
Acute	CSCA	10fold	YES	Y	100
Acute	Clonalg	10fold	YES	Y	100
Acute	Naïve bayes	traintestsplit	YES	Y	100
Acute	Logistics	traintestsplit	YES	Y	100
Acute	AIRS	traintestsplit	YES	Y	100
Acute	MLP	traintestsplit	YES	Y	100
Acute	J48	traintestsplit	YES	Y	100
Acute	AIRS2	traintestsplit	YES	Y	100
Acute	AIRS2P	traintestsplit	YES	Y	100
Acute	CSCA	traintestsplit	YES	Y	100
Acute	Clonalg	traintestsplit	YES	Y	100

Table 15. Best Accuracy Results for Each Dataset / Pure Implementations

Dataset Name	Algorithm Name	Validation Method	Performance
Acute	Logistics	10fold	100.0
Acute	AIRS	10fold	100.0
Acute	MLP	10fold	100.0
Acute	J48	10fold	100.0
Acute	AIRS2	10fold	100.0
Acute	AIRS2P	10fold	100.0
Acute	CSCA	10fold	100.0
Acute	Logistics	traintestsplit	100.0
Acute	AIRS	traintestsplit	100.0
Acute	MLP	traintestsplit	100.0
Acute	J48	traintestsplit	100.0
Acute	AIRS2	traintestsplit	100.0
Acute	AIRS2P	traintestsplit	100.0
Acute	CSCA	traintestsplit	100.0
Acute	Clonalg	traintestsplit	100.0
Breast cancer	Clonalg	traintestsplit	97.4
CPU	MLP	10fold	97.2
Credits	CHAID	traintestsplit	88.1
Iris	AIRS	traintestsplit	98.0
Iris	AIRS2P	traintestsplit	98.0
Letters	J48	10fold	87.9
Segment	MLP	traintestsplit	97.3
Segment	AIRS2	traintestsplit	97.3
Wine all	AIRS	10fold	86.5
Wine red	MLP	traintestsplit	62.5
Wine white	J48	10fold	58.2

The performance variable has been binned into intervals as LOW, MIDDLE, GOOD and VERY GOOD. Table 18 shows the distribution of each classifier across those performance intervals with respect to all stages of the experiment. Table 18 shows that the distribution of algorithms as AIRS, AIRS2, AIRS2P, J48, Naive Bayesian, Logistics, MLP and CSCA are mostly in Good or Very Good interval.

Table 16. Best Accuracy Results for Each Dataset / After Discretisations

Dataset Name	Algorithm Name	Validation Method	Performance
Acute	Logistics	10fold	100.0
Acute	AIRS	10fold	100.0
Acute	MLP	10fold	100.0
Acute	J48	10fold	100.0
Acute	AIRS2	10fold	100.0
Acute	AIRS2P	10fold	100.0
Acute	CSCA	10fold	100.0
Acute	Naïve bayes	traintestsplit	100.0
Acute	Logistics	traintestsplit	100.0
Acute	AIRS	traintestsplit	100.0
Acute	MLP	traintestsplit	100.0
Acute	J48	traintestsplit	100.0
Acute	AIRS2	traintestsplit	100.0
Acute	AIRS2P	traintestsplit	100.0
Acute	CSCA	traintestsplit	100.0
Acute	Clonalg	traintestsplit	100.0
Breast cancer	AIRS2	traintestsplit	97.4
Breast cancer	Clonalg	traintestsplit	97.4
CPU	MLP	10fold	97.2
Credits	CHAID	traintestsplit	88.1
Iris	MLP	traintestsplit	94.1
Iris	J48	traintestsplit	94.1
Letters	J48	10fold	83.6
Segment	MLP	traintestsplit	89.8
Wine all	J48	10fold	55.7
Wine red	J48	traintestsplit	59.6
Wine white	J48	traintestsplit	55.6

The basic concern of the first research question is to find if the classifiers have significantly different accuracies on multiple datasets. Even though the tables above show that none of the classifiers is dominant and different classifiers predict better in different circumstances; a one-way Anova test can help visualise the differences between classifier accuracies better. Table 19 shows that the mean of the prediction abilities of the classifiers on pure datasets are significantly different from each other and Figure 4 demonstrates this finding perfectly. According to Figure 4, the best

classifiers are Logistics, J48, AIRS, and MLP; the worst classifiers are Quest, Clonalg, CRT, Ex-CHAID and CHAID. Table 20 shows that the mean of the prediction abilities of the classifiers after discretisations are still significantly different from each other and Figure 5 demonstrates this finding perfectly. According to Figure 5, the top classifiers are J48, MLP, Logistics, AIRS, AIRS2P and AIRS2; the worst classifiers are still Quest, Clonalg, CRT, CHAID and Ex-CHAID. In the second stage of the experiment, a general tendency of the performance to drop is observable; however CSCA and Clonalg shows a tendency to increase and Quest is the most stabilized one. Table 21 shows that the mean of the prediction abilities of the classifiers after PCA are not significantly different from each other anymore since the mean of classifiers becomes closer to each one and Figure 6 demonstrates this finding perfectly. According to Figure 6, the top classifiers are J48, MLP, CSCA, Logistics, Naive Bayesian, AIRS2P, AIRS2 and AIRS; the worst classifiers are Quest, CHAID, Ex-CHAID, CRT and Clonalg. In the third stage of the experiment, still a general tendency of the performance to drop is observable; however Clonalg shows a tendency to increase; Naive Bayesian and CSCA are affected by PCA less. Although it was claimed that a classifier cannot be said to outperform the others in every dataset; with respect to all experimental trials, J48 shows the best performance on average of all datasets.

As it can be estimated, when all 780 trials are taken into account, the means of algorithms will be significantly different. Figure 7 shows this finding well and the best classifiers out of all trials are J48, MLP and Logistics. Some immune system type of algorithms (AIRS, AIRS2, AIRS2P, CSCA) and Naive Bayesian are also predicting

well. However Quest, Ex-CHAID, Clonalg, CHAID and CRT are the ones with the lowest predictive power.

Table 17. Best Accuracy Results for Each Dataset / After PCA

Dataset Name	Algorithm Name	Validation Method	Performance
Acute	Naïve bayes	10fold	100.0
Acute	Logistics	10fold	100.0
Acute	AIRS	10fold	100.0
Acute	MLP	10fold	100.0
Acute	J48	10fold	100.0
Acute	AIRS2	10fold	100.0
Acute	AIRS2P	10fold	100.0
Acute	CSCA	10fold	100.0
Acute	Clonalg	10fold	100.0
Acute	Naïve bayes	traintestsplit	100.0
Acute	Logistics	traintestsplit	100.0
Acute	AIRS	traintestsplit	100.0
Acute	MLP	traintestsplit	100.0
Acute	J48	traintestsplit	100.0
Acute	AIRS2	traintestsplit	100.0
Acute	AIRS2P	traintestsplit	100.0
Acute	CSCA	traintestsplit	100.0
Acute	Clonalg	traintestsplit	100.0
Breast cancer	CRT	traintestsplit	97.8
CPU	Naïve bayes	traintestsplit	93.1
Credits	Naïve bayes	traintestsplit	84.3
Iris	MLP	traintestsplit	88.2
Letters	J48	10fold	64.7
Segment	J48	10fold	84.7
Wine all	J48	10fold	51.9
Wine red	CRT	10fold	58.4
Wine white	J48	10fold	51.9

Table 18. Overall Distribution of Classifiers Across Performance Intervals

		Performance (Binned)				
		Low	Middle	Good	Very Good	Total
Algorithm Name	AIRS	8	10	25	17	60
	AIRS2	10	10	24	16	60
	AIRS2P	8	12	22	18	60
	CHAID	10	27	17	6	60
	Clonalg	23	11	11	15	60
	CRT	13	22	20	5	60
	CSCA	11	15	18	16	60
	ExCHAID	12	25	18	5	60
	J48	0	20	19	21	60
	Logistics	6	14	20	20	60
	MLP	5	16	17	22	60
	Naïve bayes	12	14	19	15	60
	Quest	20	17	21	2	60
Total		138	213	251	178	780

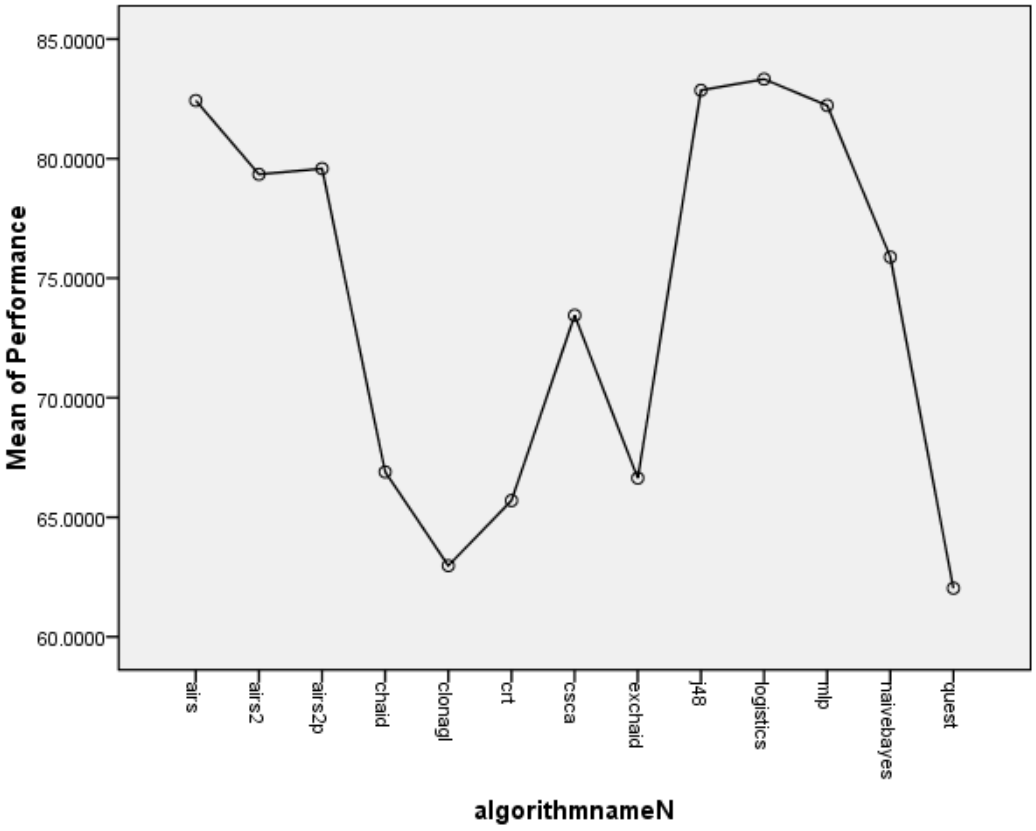

Fig. 4 Anova mean plots / pure stage

Table 19. One Way Anova / Based on Pure Implementation Step Results

Performance					
	Sum of Squares	df	Mean Square	F	Sig.
Between Groups	16120.371	12	1343.364	3.267	.000
Within Groups	101552.733	247	411.145		
Total	117673.104	259			

Table 20. One Way Anova / Based on Discretization Step Results

Performance					
	Sum of Squares	df	Mean Square	F	Sig.
Between Groups	10599.287	12	883.274	2.272	.009
Within Groups	96029.646	247	388.784		
Total	106628.933	259			

Table 21. One Way Anova / Based on PCA Step Results

Performance					
	Sum of Squares	df	Mean Square	F	Sig.
Between Groups	9726.389	12	810.532	1.766	.054
Within Groups	113371.250	247	458.993		
Total	123097.639	259			

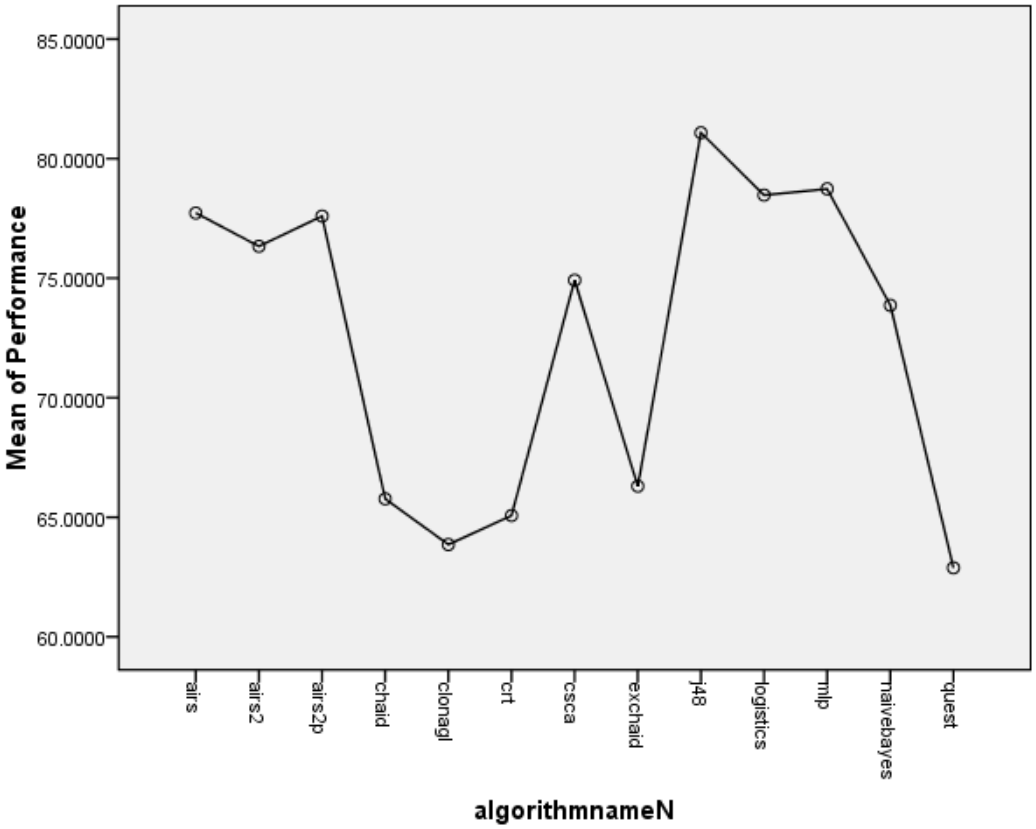

Fig. 5 Anova mean plots / discretisation stage

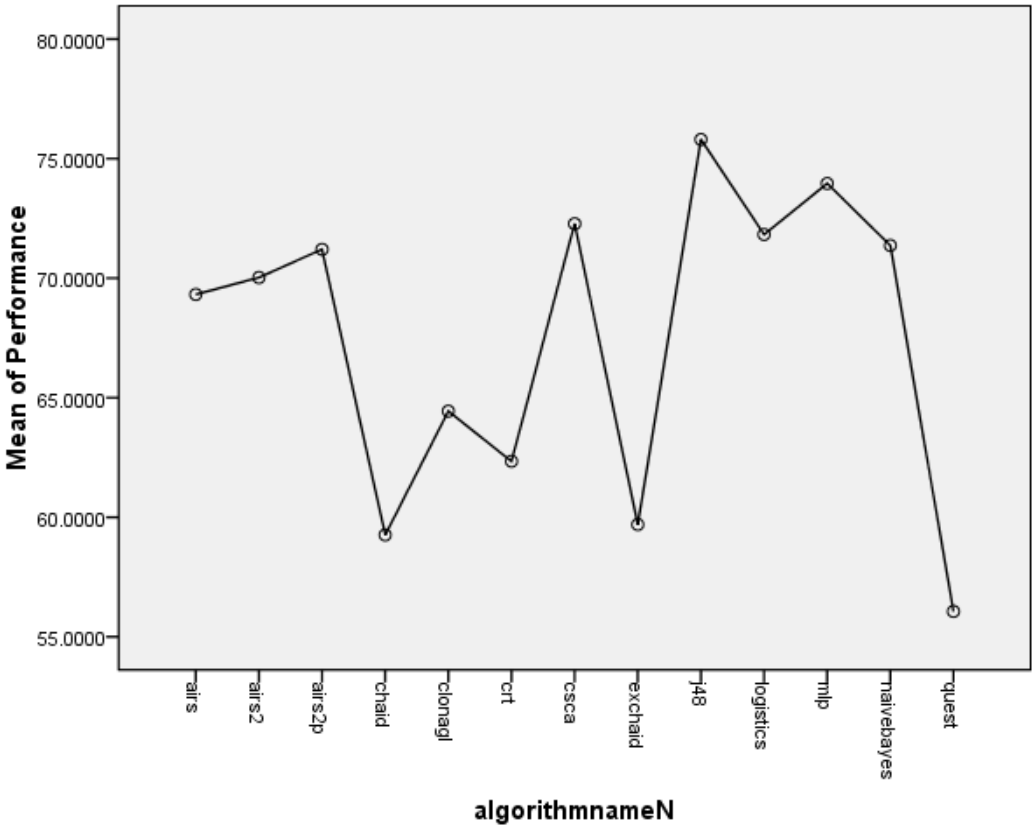

Fig. 6 Anova mean plots / PCA stage

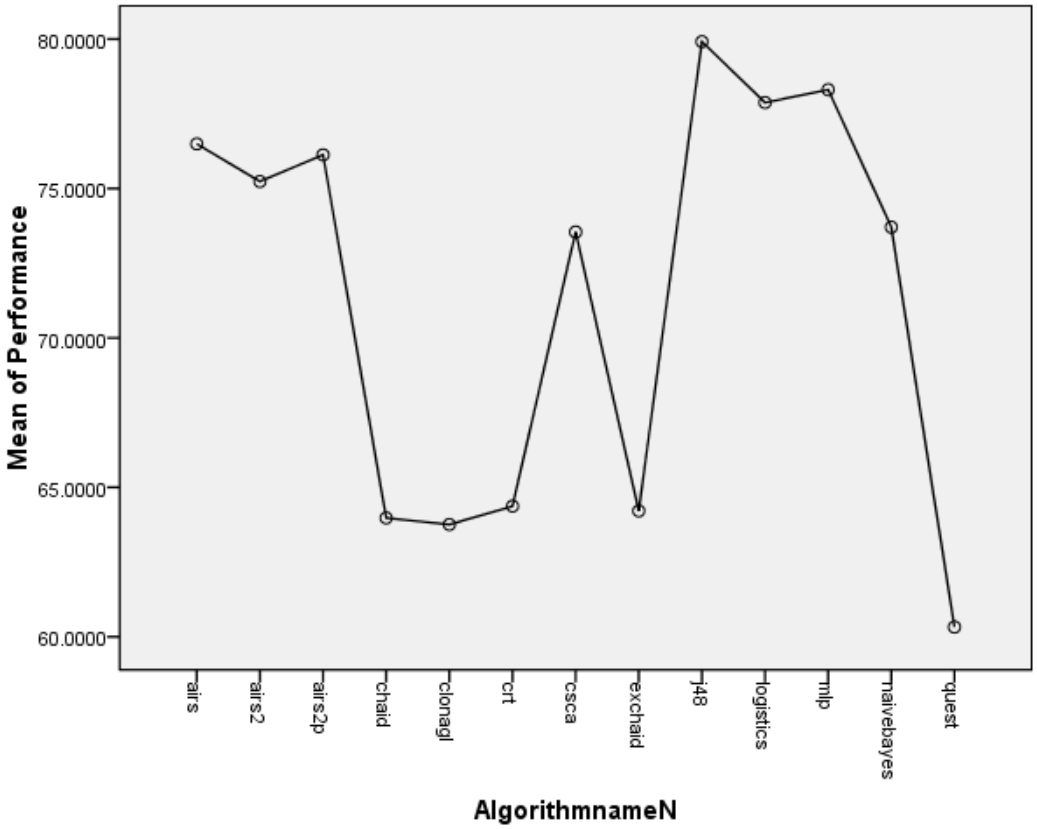

Fig. 7Anova mean plots / all trials

Research question 2: Do the characteristics of the datasets affect the performance results of the classification algorithms?

Once all of the iterations have been completed in the implementation step, a dataset of 780 rows including the combinations of the datasets, the algorithms, the validation methods, discretisation and pca application options with 13 columns for the variables have been obtained.

The first 13 fields except for the ‘Trial ID’ in Table 22 have been set as input variables, which are ‘dataset name’, ‘algorithm name’, ‘validation method’, ‘number of variables’, ‘number of nominal variables’, ‘number of numerical variables’,

'number of target class types', 'number of instances', 'Is PCA applied', 'integer variables binned', 'number of principal components' and '% of cumulative var. obtained in PCA' . The last fields in Table 22 shows the performance and complexity (cpu time used) variables, which are set as the dependent variable. Since the second research question is interested in dataset characteristics, independent variables have been defined based on dataset attributes such as number of variables, number of nominal variables, number of numerical variables, number of target class types and number of instances.

Table 22. An Excerpt From the Results Dataset

	TrialID	1	2	...	780
	Dataset Name	Acute	Iris		Cpu
	Algorithm Name	AIRS	CSCA		CSCA
	Validation Method	10fold	10fold		10fold
	Integer Variables Binned	YES	YES		NO
implementation attributes	PCA_Applied	Y	Y		N
	No Of Principal Components	3	1		0
	% of Cumulative Var. Obtained in PCA	83.198	70.99		0
dataset attributes	No Of Variables	7	4		6
	No Of Nominal Variables	6	0		0
	No Of Numerical Variables	1	4		6
	No Of Target Class Types	2	3		3
	No Of Instances	120	150		210
class	Performance	100%	84%		93.3%
	CPU Time	0.09sec	0.05sec		0.3sec

On the newly created dataset, which is referred to as the Results dataset, some kind of correlation analysis can be conducted in order to determine if any of the input variables affect the performance results significantly.

Firstly, in order to conduct the correlation analysis, all variables have been coded into numerical variables, and Z-score normalisations have been applied to them. SPSS has been used for implementation.

Table 23. Correlation Between Accuracy and Number of Variables

		Performance	Number of Variables
Performance	Pearson Correlation		
	Sig. (2-tailed)		
	N	780	780
Number of Variables	Pearson Correlation	-.237**	1
	Sig. (2-tailed)	.000	
	N	780	780

Table 24. Correlation Between Accuracy and Number of Nominal Variables

		Performance	Number of Nominal Variables
Performance	Pearson Correlation		
	Sig. (2-tailed)		
	N	780	780
Number of Nominal Variables	Pearson Correlation	.290**	1
	Sig. (2-tailed)	.000	
	N	780	780

Table 25. Correlation Between Accuracy and Number of Numerical Variables

		Performance	Number of Numerical Variables
Performance	Pearson Correlation		
	Sig. (2-tailed)		
	N	780	780
Number of Numerical Variables	Pearson Correlation	-.378**	1
	Sig. (2-tailed)	.000	
	N	780	780

Table 26. Correlation Between Accuracy and Number of Target Class Types

		Performance	Number of Target Class Types
Performance	Pearson Correlation		
	Sig. (2-tailed)		
	N	780	780
Number of Target Class Types	Pearson Correlation	-.340**	1
	Sig. (2-tailed)	.000	
	N	780	780

Table 27. Correlation Between Accuracy and Number of Instances

		Performance	Number of Instances
Performance	Pearson Correlation		
	Sig. (2-tailed)		
	N	780	780
Number of Instances	Pearson Correlation	-.480**	1
	Sig. (2-tailed)	.000	
	N	780	780

Table 28. Correlation Between Accuracy and Algorithm Type

		Performance	Algorithm Name
Performance	Pearson Correlation		
	Sig. (2-tailed)		
	N	780	780
Algorithm Name	Pearson Correlation	-.021	1
	Sig. (2-tailed)	.552	
	N	780	780

Table 29. Correlation Between Accuracy and Validation Methods

		Performance	Validation Method
Performance	Pearson Correlation		
	Sig. (2-tailed)		
	N	780	780
Validation Method	Pearson Correlation	-.064	1
	Sig. (2-tailed)	.075	
	N	780	780

According to Tables 23 to 29, some of the input variables have been found to be significantly correlated to the dependent variable, which is the performance of the classifier. Based on these results, the number of variables in the dataset (-.237 Pearson value), the number of numerical variables in the dataset (-.378 Pearson value), the number of instances in the dataset (-.480 Pearson value), the number of nominal variables in the dataset (.290 Pearson value) and the number of target class types (-.340 Pearson value) in the dataset have been found to go hand in hand with the classifier performance. On the other hand, algorithm name and validation method have been

found not to be significantly correlated to classifier accuracy. As a result, the answer to the second question can be concluded in such a way that most of the dataset characteristics can affect the classifier performance.

Research question 3: Does binning the continuous numerical variables in the dataset into discreet intervals affect the classifier accuracy?

Table 30. Correlation Between Accuracy and Discretisation

		Performance	Integer Variables Binned
Performance	Pearson Correlation		
	Sig. (2-tailed)		
	N	780	780
Integer Variables Binned	Pearson Correlation	-.091*	1
	Sig. (2-tailed)	.011	
	N	780	780

According to Table 30, the input variable ‘the integers are binned into intervals’ has been found to be significantly correlated to the dependent variable, which is the performance of the classifier (-.091 Pearson value). As a result, the answer to the third question can be concluded in such a way that discretisation of the continuous variables in the dataset can affect the classifier performance.

Table 31. Anova Results of Performance and Discretisation

Performance					
	Sum of Squares	df	Mean Square	F	Sig.
Between Groups	5829.0	1	5829.0	13.042	.000
Within Groups	347729.9	778	447.0		
Total	353559.0	779			

Moreover, Table 31 shows that the performance means of instances of which continuous variables have been discretised and the ones of which continuous variables have not been discretised are found to be significantly different with respect to the significance level of the one-way Anova test (.000).

Research question 4: Does applying principal component analysis in the dataset affect the classifier accuracy?

Table 32. Correlation Between Accuracy and PCA

		Performance	PCA applied
Performance	Pearson Correlation		
	Sig. (2-tailed)		
	N	780	780
PCA applied	Pearson Correlation	-.128**	1
	Sig. (2-tailed)	.000	
	N	780	780

Table 33. Correlation Between Accuracy and Cumulative Variance in PCA

		Performance	% of Cumulative Variance Obtained in PCA
Performance	Pearson Correlation		
	Sig. (2-tailed)		
	N	780	780
% of Cumulative Variance Obtained in PCA	Pearson Correlation	-.133**	1
	Sig. (2-tailed)	.000	
	N	780	780

Table 34. Correlation Between Accuracy and Number of Components in PCA

		Number of Principal Components	Performance
Number of Principal Components	Pearson Correlation		
	Sig. (2-tailed)		
	N	780	780
Performance	Pearson Correlation	-.282**	1
	Sig. (2-tailed)	.000	
	N	780	780

According to Tables 32 to 34, the input variables ‘PCA applied’ (-.128 Pearson value), ‘% of cumulative variance obtained in PCA’ (-.133 Pearson value) and ‘number of principal components’ (-.282 Pearson value) has been found to be significantly correlated to the dependent variable, which is the performance of the classifier. As a result, the answer to the forth question can be concluded in such a way that applying PCA in the dataset can affect the classifier performance.

Table 35. Anova Results of Performance and PCA

Performance					
	Sum of Squares	df	Mean Square	F	Sig.
Between Groups	2906.5	1	2906.5	6.449	.011
Within Groups	350652.5	778	450.7		
Total	353559.0	779			

On the other hand, Table 35 shows that the performance means of instances of which PCA has not been applied and the ones of which PCA has applied found to be significantly different with respect to the significance level of the one-way Anova test (.011).

Research question 5: Based on the results derived from the empirical results of this study (applying classifiers on various dataset with different implementation techniques), can a model to predict the performance of the classification algorithm be built?

Since there is a Results dataset containing the algorithm, dataset and implementation specific attributes in Table 22, it is possible to use these in a regression model and see their causal effects on the dependent performance variable. Due to finding the correlations between some of the selected independent and dependent performance variable in the previous research questions, it is essential to design a regression mode; therefore, a regression model has been developed to answer the last research question. According to the regression results, it is possible to build a model to predict the performance result. Equation 1 shows the regression function for predicting the performance.

The purpose of conducting a regression is to understand whether the coefficients on the independent variables are really different from 0; in other words, whether the independent variables are having an observable effect on the dependent variable. If coefficients are different than 0, this means the null hypothesis (the dependent not affected by the independents) can be rejected (Doğan & Tanrıkulu, 2010). Based on the regression function (1), some of the independent variables have been found to affect the dependent variable's performance. As a result, the number of principal components, % of cumulative variance obtained in PCA, number of target class types, number of instances, algorithm name, validation method and discretisation have a negative effect on performance. On the other hand, the number of variables,

number of nominal variables and PCA application has a positive effect on the performance.

Within a 95% confidence interval, p values in Table 36 should be close to or lower than 0.05 in order to be accepted as significant enough. With respect to p values (sig. column), the effect of the number of principal components, number of nominal variables, number of instances, validation method, and PCA application on performance is said to be more certain.

Table 36. Regression Results / Accuracy

Model		Unstandardized Coefficients		Standardized Coefficients		
		B	Std. Error	Beta	t	Sig.
1	Constant	.000	.029		.000	1.000
	Number of Principal Components	-.537	.065	-.537	-8.285	.000
	% of Cumulative Variance Obtained in PCA	-.260	.229	-.260	-1.136	.256
	Number of Variables	.044	.041	.044	1.063	.288
	Number of Nominal Variables	.299	.045	.299	6.712	.000
	Number of Target Class Types	-.150	.083	-.150	-1.797	.073
	Number of Instances	-.219	.074	-.219	-2.951	.003
	Algorithm Name	-.025	.029	-.025	-.845	.398
	Validation Method	-.064	.029	-.064	-2.194	.029
	Discretisation	-.035	.034	-.035	-1.050	.294
	PCA application	.608	.231	.608	2.633	.009

Performance =

-.537 * Number of Principal Components

-.260* % of Cumulative Variance Obtained in PCA

+.044* Number of Variables

+ .299* Number of Nominal Variables

-.150* Number of Target Class Types **(1)**

-.219* Number of Instances

-.025* Algorithm Name

-.064* Validation Method

-.035* Discretisation

+.608* PCA application

Research question 6: Does implementing the same classification algorithm on multiple datasets and with different implementation techniques result in significantly different complexity?

The basic concern of the sixth research question is to find if the classifiers have significantly different complexities on multiple datasets. A one-way Anova test can help visualise the differences between classifier accuracies perfectly. Table 37 shows that the mean of the complexitities of the classifiers on pure datasets are significantly different from each other (sig. 0.02) and Figure 8 demonstrates this finding perfectly. According to Figure 8, the most complex classifier is CSCA; MLP is slightly more complex than the rest of the classifiers and the remaining classifiers are all in low complexity nature.

Table 38 shows that the difference in the mean of the complexities of the classifiers after discretisations are not as significant as it was anymore and Figure 9 demonstrates this finding perfectly. According to Figure 9, an overall increase in the time spent is observable for all classifiers. The most complex classifier is still CSCA and the complexity of J48 is increased more as well; the rest of the classifiers are in similar low complexities. In the second stage of the experiment, a general tendency of the complexity to increase is observable.

Table 39 shows that the mean of the complexities of the classifiers after PCA are significantly different from each other and Figure 10 demonstrates this finding perfectly. According to Figure 10, the most complex classifiers is still CSCA and the other classifiers are in lower complexity. In the third stage of the experiment, a general tendency of the comlexity to increase is observable.

As it can be estimated, when all 780 trials are taken into account, the means of algorithms will be significantly different. Figure 11 and Table 41 shows this finding well and the most complex classifier is always CSCA out of all trials. Different implementation techniques do not significantly change the overall picture about the classifier complexities.

Table 37. One Way Anova / Based on Pure Implementation Step Results

CPUTime					
	Sum of Squares	df	Mean Square	F	Sig.
Between Groups	59519064	12	4959922	2	0.02
Within Groups	599185463	247	2425852		
Total	658704526	259			

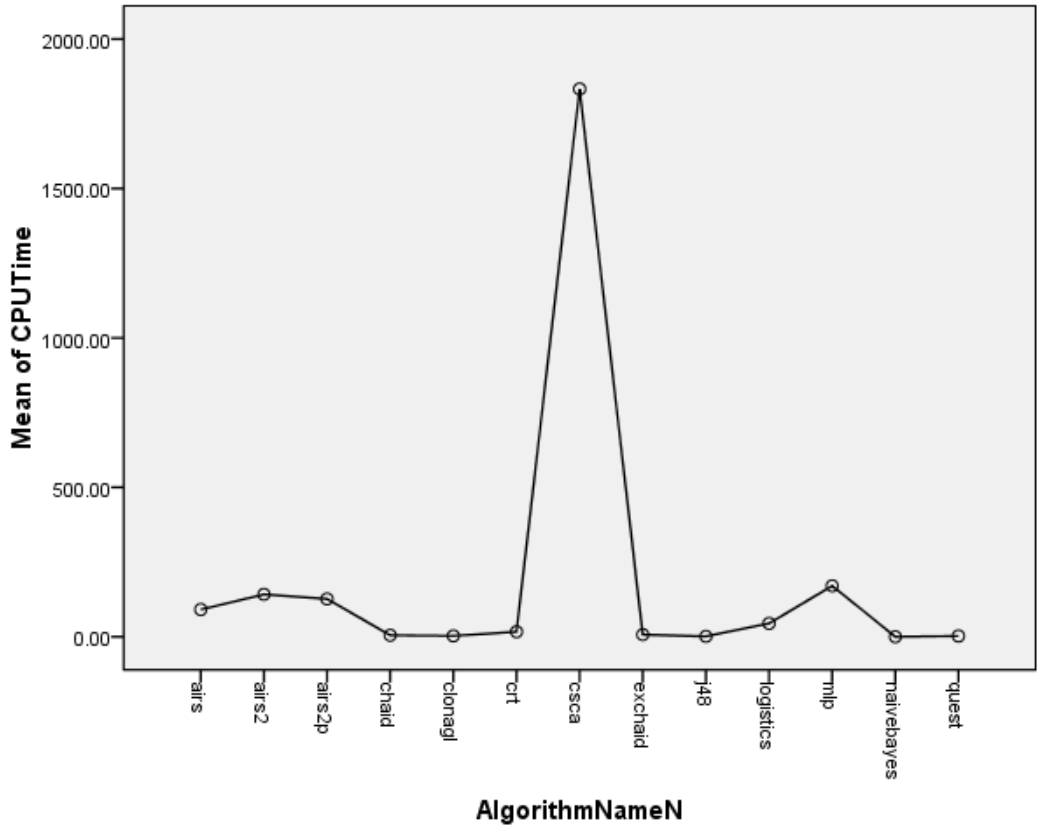

Fig. 8 Anova mean plots / pure implementations

Table 38. One Way Anova / Based on Discretisation Step Results

CPUTime					
	Sum of Squares	df	Mean Square	F	Sig.
Between Groups	503022223	12	41918519	2	0.06
Within Groups	6021358944	247	24377971		
Total	6524381167	259			

Table 39. One Way Anova / Based on PCA Implementation Step Results

CPUTime					
	Sum of Squares	df	Mean Square	F	Sig.
Between Groups	303576295	12	25298025	3	0.00
Within Groups	2273411220	247	9204094		
Total	2576987514	259			

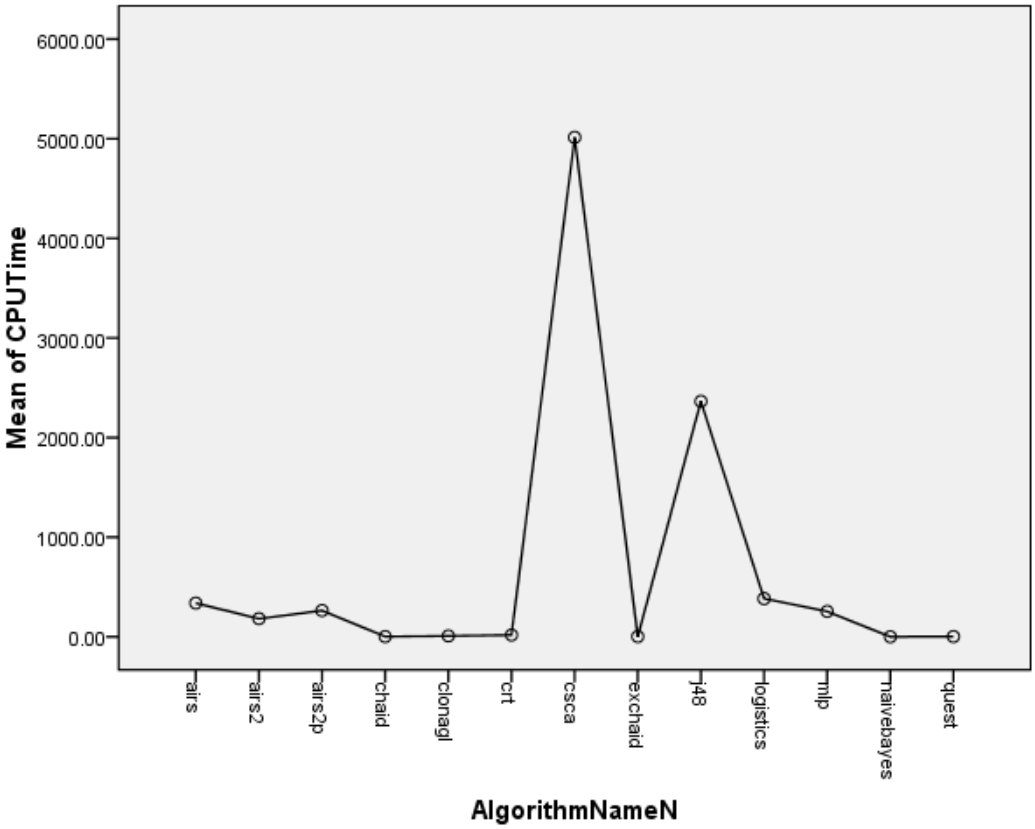

Fig. 9 Anova mean plots / after discretisations

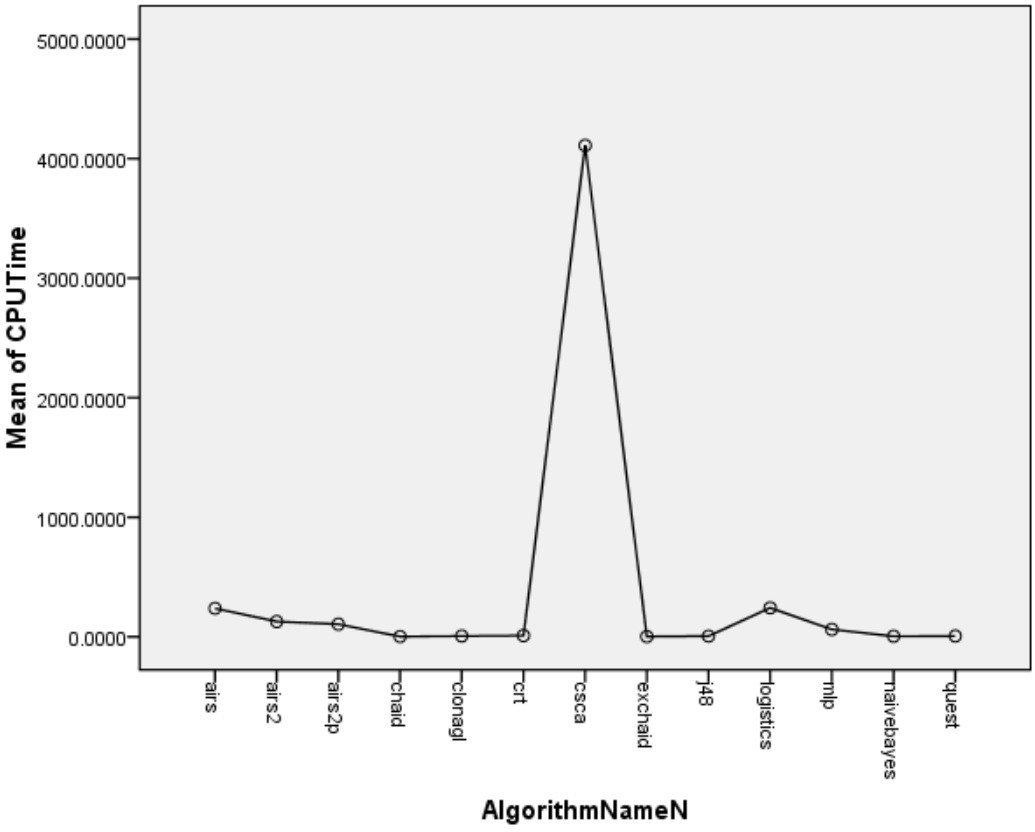

Fig. 10 Anova mean plots / after PCA

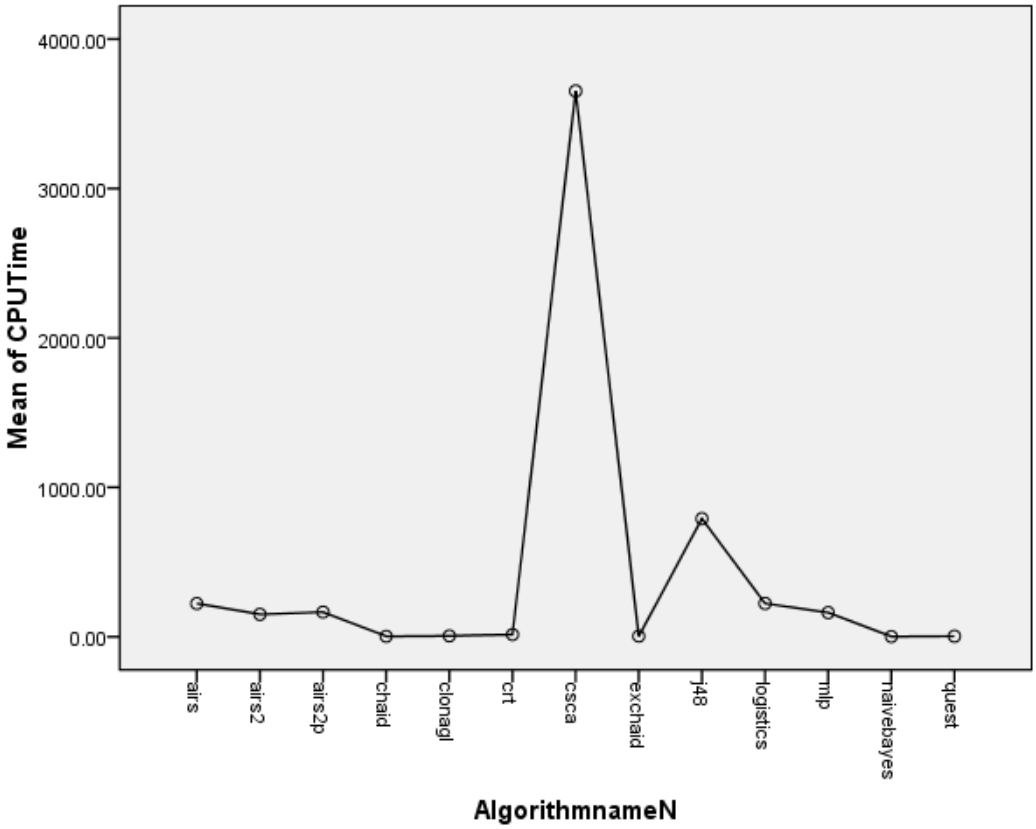

Fig. 11 Anova mean plots / all trials

Table 40. One Way Anova / Based on Overall Implementation Step Results

CPUTime					
	Sum of Squares	df	Mean Square	F	Sig.
Between Groups	713800388	12	59483366	5	0.00
Within Groups	9078189767	767	11835971		
Total	9791990156	779			

The complexity variable has been binned into intervals as LOW (up to 415 seconds), MIDDLE (415-3960 seconds) and HIGH (more than 3960 seconds). Table 41 shows the distribution of each classifier across those complexity intervals with respect to all stages of the experiment. Table 41 shows that only the CSCA and J48 appears to be in the high complexity part (HIGH cpu time interval) and algorithms are mostly in MIDDLE or LOW intervals.

Table 41. Overall Distribution of Classifiers Across Complexity Intervals

		CPU Time (seconds)			
		-3130 - 415	415-3960	3960+	Total
Algorithm Name	AIRS	55	5	0	60
	AIRS2	53	7	0	60
	AIRS2P	51	9	0	60
	CHAID	60	0	0	60
	Clonalg	60	0	0	60
	CRT	60	0	0	60
	CSCA	46	8	6	60
	ExCHAID	60	0	0	60
	J48	59	0	1	60
	Logistics	55	5	0	60
	MLP	54	6	0	60
	Naïve bayes	60	0	0	60
	Quest	60	0	0	60
Total		733	40	7	780

Research question 7: Do the characteristics of the datasets affect the complexity of the classification algorithms?

According to Tables 42 to 48, some of the input variables have been found to be significantly correlated to the dependent variable, which is the complexity of the classifier. Based on these results, the number of variables in the dataset (.124 Pearson value), the number of numerical variables in the dataset (.137 Pearson value), the number of target class types (.260 Pearson value) and the number of instances in the dataset (.300 Pearson value) have been found to go hand in hand with the classifier complexity. On the other hand, algorithm name, validation method and number of nominal variables have been found not to be significantly correlated to classifier complexity. As a result, the answer to the seventh question can be concluded in such a way that most of the dataset characteristics can affect the classifier complexity. Based on the results, the density of the instances, variables, target classes and numerical

variables in the dataset are expected to increase the cpu time consumed during the model implementation ultimately increasing the complexity of the classifiers.

Table 42. Correlation Between Complexity and Number of Variables

		CPU Time	Number of Variables
CPU Time	Pearson Correlation	1	.124**
	Sig. (2-tailed)		.001
	N	780	780
Number of Variables	Pearson Correlation	.124**	1
	Sig. (2-tailed)	.001	
	N	780	780

Table 43. Correlation Between Complexity and Number of Nominal Variables

		CPU Time	Number of Nominal Variables
CPU Time	Pearson Correlation	1	-.051
	Sig. (2-tailed)		.158
	N	780	780
Number of Nominal Variables	Pearson Correlation	-.051	1
	Sig. (2-tailed)	.158	
	N	780	780

Table 44. Correlation Between Complexity and Number of Numerical Variables

		CPU Time	Number of Numerical Variables
CPU Time	Pearson Correlation	1	.137**
	Sig. (2-tailed)		.000
	N	780	780
Number of Numerical Variables	Pearson Correlation	.137**	1
	Sig. (2-tailed)	.000	
	N	780	780

Table 45. Correlation Between Complexity and Number of Target Class Types

		CPU Time	Number of Target Class Types
CPU Time	Pearson Correlation	1	.260**
	Sig. (2-tailed)		.000
	N	780	780
Number of Target Class Types	Pearson Correlation	.260**	1
	Sig. (2-tailed)	.000	
	N	780	780

Table 46. Correlation Between Complexity and Number of Instances

		CPU Time	Number of Instances
CPU Time	Pearson Correlation	1	.300**
	Sig. (2-tailed)		.000
	N	780	780
Number of Instances	Pearson Correlation	.300**	1
	Sig. (2-tailed)	.000	
	N	780	780

Table 47. Correlation Between Complexity and Validation Method

		CPU Time	Validation Method
CPU Time	Pearson Correlation	1	-.013
	Sig. (2-tailed)		.711
	N	780	780
Validation Method	Pearson Correlation	-.013	1
	Sig. (2-tailed)	.711	
	N	780	780

Table 48. Correlation Between Complexity and Algorithm Type

		CPU Time	Algorithm Type
CPU Time	Pearson Correlation	1	.001
	Sig. (2-tailed)		.981
	N	780	780
Algorithm Type	Pearson Correlation	.001	1
	Sig. (2-tailed)	.981	
	N	780	780

Research question 8: Does binning the continuous numerical variables in the dataset into discreet intervals affect the classifier complexity?

According to Table 49, the input variable ‘the integers are binned into intervals’ has not been found to be significantly correlated to the dependent variable, which is the complexity of the classifier. As a result, the answer to the eighth question can be concluded in such a way that discretisation of the continuous variables in the dataset does not significantly affect the classifier modelling time.

Moreover, Table 50 shows that the complexity means of instances of which continuous variables have been discretised and the ones of which continuous variables have not been discretised are not found to be significantly different with respect to the significance level of the one-way Anova test (.21).

Table 49. Correlation Between Complexity and Discretisation

		CPU Time	Integers are binned
CPU Time	Pearson Correlation	1	.045
	Sig. (2-tailed)		.206
	N	780	780
Integers are binned	Pearson Correlation	.045	1
	Sig. (2-tailed)	.206	
	N	780	780

Table 50. Anova Results of Complexity and Discretisation

	Sum of Squares	df	Mean Square	F	Sig.
Between Groups	20148280	1	20148280	2	0.21
Within Groups	9771841876	778	12560208		
Total	9791990156	779			

Research question 9: Does applying principal component analysis in the dataset affect the classifier complexity?

According to Tables 51 to 53, none of the input variables ‘PCA applied’, ‘% of cumulative variance obtained in PCA’ or ‘number of principal components’ has been found to be significantly correlated to the dependent variable, which is the complexity of the classifier. As a result, the answer to the ninth question can be concluded in such

a way that applying PCA in the dataset cannot affect the classifier complexity significantly.

On the other hand, Table 54 shows that the complexity means of instances of which PCA has not been applied and the ones of which PCA has applied found to be not significantly different with respect to the significance level of the one-way Anova test (0.8).

Table 51. Correlation Between Complexity and Number of Principal Components

		CPU Time	Number of Principal Components
CPU Time	Pearson Correlation	1	.025
	Sig. (2-tailed)		.484
	N	780	780
Number of Principal Components	Pearson Correlation	.025	1
	Sig. (2-tailed)	.484	
	N	780	780

Table 52. Correlation Between Complexity and % of Cumulative Variance in PCA

		CPU Time	% Of Cumulative Variance Obtained in PCA
CPU Time	Pearson Correlation	1	-.007
	Sig. (2-tailed)		.844
	N	780	780
% Of Cumulative Variance Obtained in PCA	Pearson Correlation	-.007	1
	Sig. (2-tailed)	.844	
	N	780	780

Table 53. Correlation Between Complexity and PCA

		CPU Time	PCA Applied
CPU Time	Pearson Correlation	1	-.007
	Sig. (2-tailed)		.838
	N	780	780
PCA Applied	Pearson Correlation	-.007	1
	Sig. (2-tailed)	.838	
	N	780	780

Table 54. Anova Results of Complexity and PCA

	Sum of Squares	df	Mean Square	F	Sig.
Between Groups	527949	1	527949	0	0.8
Within Groups	9791462207	778	12585427		
Total	9791990156	779			

Research question 10: Based on the results derived from the empirical results of this study (applying classifiers on various dataset with different implementation techniques), can a model to predict the complexity (consumed CPU time in seconds) of the classification algorithm be built?

Due to finding the correlations between some of the selected independent and dependent performance variable in the previous research questions, it is essential to

design a regression mode; therefore, a regression model has been developed to answer the tenth research question.

According to the regression results, it is possible to build a model to predict the complexity result. Equation 2 shows the regression function for predicting the complexity.

Based on the regression function (2), some of the independent variables have been found to affect the complexity of the dependent variable. As a result, the number of principal components, the number of variables in datasets, validation method and PCA application has a negative effect on complexity. On the other hand, number of instances, algorithm name, discretisation, the number of target class types, % of cumulative variance obtained in PCA and number of nominal variables have a positive effect on the performance. However the negative effect here implies a reduction in complexity which means less time to build the model, and positive effect on the complexity means increased time to build the model.

Within a 95% confidence interval, p values in Table 55 should be close to or lower than 0.05 in order to be accepted as significant enough. With respect to p values (sig. column), only the effect of the number instances on complexity is said to be more certain.

Table 55. Regression Results / Complexity

Model		Unstandardized Coefficients		Standardized Coefficients	t	Sig.
		B	Std Error	Beta		
1	(Constant)	.000	.034		.000	1.000
	Number of Principal Components	-.073	.076	-.073	-.953	.341
	% Of Cumulative Varience Obtained in PCA	.110	.270	.110	.407	.684
	Number of Variables	-.032	.049	-.032	-.660	.509
	Number of Nominal Variables	.006	.052	.006	.110	.913
	Number of Target Class Types	.099	.098	.099	1.012	.312
	Number of Instances	.248	.087	.248	2.833	.005
	Validation Method	-.013	.034	-.013	-.388	.698
	Integers are binned	.065	.040	.065	1.652	.099
	PCA Applied	-.086	.272	-.086	-.318	.751
	Algorithm Name	.003	.034	.003	.090	.929

Complexity =

-0.953* Number of Principal Components

+0.407* % of Cumulative Variance Obtained in PCA

-0.660* Number of Variables

+0.110* Number of Nominal Variables

+1.012* Number of Target Class Types (2)

+2.833* Number of Instances

-0.388* Validation Method

+1.652* Discretisation

-0.318* PCA application

+0.090* Algorithm Name

Research question 11: Are the abilities of classifiers to handle missing or noisy data different?

In order to answer the last research question, 'Breast cancer' dataset has been selected to make the experiments. This dataset has some noise such as missing values. Firstly, all 13 algorithms with 10 fold cross validation have been applied on the 'Breast cancer' dataset and accuracy results are tabulated. Then, missing values were cleaned from 'Breast cancer' and the same algorithms were implemented over. The results of the two different steps can lead to determine which algorithms are more robust and which are not.

The first column in Table 56 shows the performance of classifiers before missing value analysis and the second column shows the accuracies after missing

values cleaned. It is obvious to see that most of the algorithms are able to handle missing values since the accuracies are quite better and only very slight increase in the performance is observed after missing value analysis. Moreover, AIRS and CSCA shows a different tendency to reduce the accuracy when missing values are cleaned. Certainly, more experiments should be conducted to draw more certain conclusions about classifier robustness.

Table 56. Robustness Comparison

	before MVA	afterMVA	Difference
Naïve bayes	96.0	96.3	-0.3
Logistics	96.6	96.8	-0.2
CHAID	92.7	93.0	-0.3
AIRS	97.0	96.2	0.8
MLP	95.3	96.0	-0.7
ExCHAID	92.7	93.0	-0.3
CRT	92.4	92.7	-0.3
Quest	91.0	91.2	-0.2
J48	94.6	96.0	-1.4
AIRS2	96.6	97.1	-0.5
AIRS2P	96.1	96.8	-0.6
CSCA	96.7	96.3	0.4
Clonalg	95.6	95.9	-0.3

CHAPTER 4.

CONCLUSION

Classification type of algorithms have been very popularly used by the data mining community and the prediction abilities of them or their complexities have been discussed for many years. When implemented efficiently and correctly data mining systems can be very crucial in many areas such as customer relationship management, fraud detection, credit evaluation, risk evaluations, medical treatment or disease detection, etc. Therefore the quality criterion like accuracy and complexity plays a crucial role in data mining projects while selecting a proper classifier.

In this study, CHAID, Ex-CHAID, CRT, Quest, J48, MLP, Logistics, AIRS, AIRS2, AIRSP, CSCA, Clonalg and Naïve Bayesian classification algorithms have been implemented on 10 different datasets.

According to the accuracy results, 'Acute' dataset is the easiest one to be predicted since most of the algorithms have their highest accuracy value on it. However none of the algorithms outperform the others in each dataset; therefore a algorithm may not be dominantly predicting the best in all domains and data miner should think about dataset bias as well.

J48, MLP and Logistics are the best predicting classifiers out of all trials on average. Immune system type of algorithms and Naive Bayesian are also predicting well. However CHAID, Ex-CHAID, CRT, Quest and Clonalg are the ones with the

lowest predictive power. Therefore they may not be suitable for datasets with high dimensionality and continuous integers.

The mean of the prediction abilities of the classifiers before and after discretisations are significantly different from each other. After discretisations, a general tendency of the performance to drop is observable; however CSCA and Clonalg shows a tendency to increase so that they are able to handle discrete values better and Quest is not affected a lot by the discretisation. After PCA, the mean of the prediction abilities of the classifiers are not significantly different from each other since the mean of classifiers becomes closer to each one. There is still a general tendency of the performance to drop; however Clonalg shows a tendency to increase; Naive Bayesian and CSCA are affected by PCA less. Considering the experimental results, J48 shows the best prediction ability for all stages on average. Thus, data analysts should be aware that some data pre-processing attempts may reduce the accuracy for some classifiers.

Therefore, conducting similar experiments may help data miners about which classifier to choose when. Based on the empirical findings, J48, MLP, Logistics and most immune system algorithms are producing quite robust accuracies following a similar pattern whether the dataset is pre-processed or not.

Another interest has been to find out the correlations between the accuracy results of classifiers and the dataset attributes. Based on the correlation analysis, the number of variables in the dataset, the number of numerical variables in the dataset and the number of instances in the dataset, the number of nominal variables in the

dataset and the number of target class types in the dataset have been found to go hand in hand with the classifier performance. The correlations between the accuracy results of classifiers and whether to discretise the continuous variables or not were also within the scope of the study. Based on the correlation analysis, the input variable 'the integers are binned into intervals' has been found to be significantly correlated to the dependent variable, which is the performance of the classifier. Another interest has been to find out the correlations between the accuracy results of classifiers and whether to apply PCA or not. Based on the correlation analysis, the input variable the input variables 'PCA applied', '% of cumulative variance obtained in PCA' and 'number of principal components' has been found to be significantly correlated to the dependent variable, which is the performance of the classifier. The statistical results show the fact that dataset characteristics, discretisation and PCA affect the classifier accuracy.

On the other hand, accuracy is not the only concern of the data analysts. The complexity that is the amount of cpu consumed by the classifier is another concern. The other research questions are related to the complexities of the algorithms. By complexity, the cpu time consumed has been implied in the study. Based on the Anova mean plots, CSCA algorithm has always found to be the most complex algorithm and J48 is a little bit more complex than the other algorithms and the rest are in similar complexity. Discretisation of continuous variables into intervals or PCA implementation has a general tendency to increase the cpu time but the mean of all cpu times are not changed significantly in either case. This part of the study gives a clear idea about the model development times, since the data analyst can understand that training time will last longer with CSCA based on these findings. Classifiers run at

times measured by seconds or minutes; however CSCA runs by hours or days sometimes. Moreover data analysts should also consider the fact that pre-processing may increase the complexity of the classifiers.

With respect to correlations between dataset characteristics and complexity; number of variables, number of numerical variables, number of target classes and number of instances can be said to significantly affect the complexities of classifiers. However discretisation or PCA implementation has no significant effect on the classifier complexity.

Based on the findings of this study, it can also be said that a regression model can be built to predict the performance and the complexity of a classifier on a given dataset with given implementation conditions. Lastly a robustness comparison conducted on a dataset before and after the missing values has been cleaned. Results show that the accuracy of algorithms does not reduce dramatically when noise is included too. Certainly, more experiments should be conducted to conclude more precisely about classifier robustness.

In this study, the factors affecting the classification algorithm performance and complexity have been underlined based on the empirical results of difference tests, correlation and regression studies. The fact that dataset characteristics and implementation details influence the accuracy or complexity of the algorithm cannot be denied. The deviation of algorithm accuracies across different datasets is observable. The means of accuracies with respect to discretisation or PCA also are significantly different. The business and academic community should take these

results into consideration, since establishing a knowledge discovery process on the same algorithm with the same implementation details may not always be certain and efficient. The model assessment and selection phase should be paid the utmost attention in an iterative manner, because any difference in dataset characteristics or pre-processing techniques can affect the model's accuracy or complexity, and switching to another classifier or changing the pre-processing technique may be a better decision. The regression model also gives some hints about the importance of a dataset, and that the accuracy or complexity can be predicted based on the instances or the field attributes of the dataset. This study can give idea about the expected accuracy and complexity of classifiers based on given dataset or pre-processing characteristics.

It is not an easy task to decide which classifier to use in a data mining problem; thus this study shows the importance of model selection and explains that an algorithm and a data pre-processing technique is not the best choice for all datasets. As Rokach & Maimon also claim that "no induction algorithm can be best in all possible domains" and they introduce the concept of "No Free Lunch Theorem" which says that "if one inducer is better than another in some domains, then there are necessarily other domains in which this relationship is reversed." (Maimon & Rokach, 2008). Data miners face the dilemma of which classifier to use and the situation gets harder when other criteria like comprehensibility or complexity are also concerned.

Certainly, conclusions are based on the scope of this study; therefore, increasing the scope may help to develop an extended framework for predicting the accuracy or the complexity of classifiers better. Obviously, there may be other factors influencing the accuracy or complexity of a model, thus input variables of the

regression function should be increased in the future. Moreover, there are other quality factors of classifiers to be discovered such as scalability, interestingness or comprehensibility. It is suggested that those other quality factors can be included to the study in the future. For example, if larger databases can be found then scalability of the classifiers can be tested in another study as well. Or some rule based or rule producing algorithms can be compared with respect to the interestingness measure, whether the models can produce new and valid knowledge or not.

REFERENCES

Armstrong L. J., Diepeveen D. & Maddern R. (2007). The application of data mining techniques to characterize agricultural soil profiles. Proc. 6th Australasian Data Mining Conference (AusDM'07), Gold Coast, Australia.

Badulescu L. A. (2007). The choice of the best attribute selection measure in Decision Tree induction. Annuals of University of Craiova, Math. Comp. Sci. Ser., 34(1), 88-93.

Berson A., Smith S., & Thearling K. (1999). Building Data Mining Applications for CRM. McGraw Hill.

Brownlee J. (2005). Clonal Selection theory & Clonalg , The Clonal Selection Classification Algorithm (CSCA): Technical Report. Retrieved January 1, 2010 from http://www.ict.swin.edu.au/personal/jbrownlee/2005/TR02-2005.pdf.

CART (2010). Machine Learning in Real World: CART [PowerPoint slides]. Retrieved January 1, 2010 from https://classshares.student.usp.ac.fj/IS421/dm8-decision-tree-cart.ppt.

Cios K. J., Pedrycz W., Swiniarski R. W. & Kurgan L. A. (2007). Data Mining A Knowledge Discovery Approach. USA: Springer.

Cross Industry Standard Process for Data Mining (2010). Process Model. Retrieved January 1, 2010 from http://www.crisp-dm.org/Process/index.htm.

Data Mining Group (2010). Data Mining Group. Retrieved January 1, 2010 from http://www.dmg.org/v4-0/GeneralStructure.html.

Doğan N., Tanrıkulu Z. (2010). A Comparative Framework for Evaluating Classification Algorithms. Proc. WCE 2010 International Data Mining and Knowledge Engineering (ICDMKE 2010).

Dunham M. H. (2002). Data Mining Introductory and Advanced Topics. New Jersey: Prentice Hall.

Finnoff W., Hergert F. & Zimmermann H. G. (1995). Improving model selection by dynamic regularization methods. In T. Petsche, S. J. Hanson, J. Shavlik. Computational Learning Theory and Natural Learning Systems: Selecting Good Models. (Ed.), (pp. 334–343). Cambridge: MIT Press.

Gamble, A. (2001). The Dummy's Guide to Data Analysis Using SPSS. Retrieved January 1, 2010 from http://www.aged.tamu.edu/research/readings/Research/2001SPSS_Guide.pdf.

Ge E., Nayak R., Xu Y. & Li Y. (2006). Data mining for lifetime prediction of metallic components. Proc. Fifth Australasian Data Mining Conference (AusDM2006), Sydney, Australia.

Hacker S. & Ahn L.v. (2009). Matchin: eliciting user preferences with an online game. CHI '09: Proceedings of the 27th international conference on Human factors in computing systems, New York, USA, ACM.

Han J. & Kamber M. (2005). Data Mining Concepts and Techniques. 2nd ed., Academic Press, Morgan Kaufmarm Publishers.

Hand D., Mannila H.& Smyth P. (2001). Principles of Data Mining. Cambridge: The MIT Press.

Harper F. M., Moy D. & Konstan J.A. (2009). Facts or Friends? Distinguishing Informational and Conversational Questions in Social Q&A Sites. CHI-2009, ACM.

He H., Jin H., Chen J., McAullay D., Li, J. & Fallon, T. (2006). Analysis of breast feeding data using data mining methods. Proc. Fifth Australasian Data Mining Conference (AusDM2006), Sydney, Australia.

Hornick M.F., Marcade E. & Venkayala S. (2007). Java Data Mining: Strategy, Standard, and Practice: A Practical Guide for architecture, design, and implementation. Morgan Kaufman.

Howley T., Madden M. G., O'Connell M. L. & Ryder A.G. (2005).The Effect of Principal Component Analysis on Machine Learning Accuracy with High Dimensional Spectral Data. Retrieved January 1, 2010 from http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.74.8032&rep=rep1&type=pdf.

Kaelbling L. P. (1994). Associative methods in reinforcement learning: an emprical study. In S. J. Hanson, T. Petsche, M. Kearns & R. L. Rivest. Computational Learning Theory and Natural Learning Systems: Intersection Between Theory and Experiment. (Ed.), (pp. 145–153). Cambridge: MIT Press.

Keogh E., Kasetty S., (2002) On the Need for Time Series Data mining benchmarks: a survey and empirical demonstration. Data Mining and Knowledge Discovery, 7, 349-371.

Keogh E., Stefano L. & Ratanamahatana C. A. (2004). Towards Parameter-Free Data Mining. KDD '04, Seattle, Washington, USA.

Kirkos E., Spathis C. & Manolopoulos Y. (2007). Data Mining techniques for the detection of fraudulent financial statements. Expert Systems with Applications, 32, 995–1003.

Küçükkocaoğlu G., Benli Y. K. & Küçüksözen C. (2009). Finansal Bilgi Manipülasyonunun Tespitinde Yapay Sinir Ağı Modelinin Kullanımı. IMKB, 36.

Lim T.S., Loh W.Y. & Shih Y.S. (2000). A Comparison of Prediction Accuracy, Complexity, and Training Time of Thirty-three Old and New Classification Algorithms. Machine Learning, 40, 203-229.

Limanto H.Y., Cing T.J. & Watkins A. (2007). An Immune systems Approach for classifying Mobile Phone Usage. International Journal of Data Warehousing and Mining, 3(2), 55-65.

Maimon O. & Rokach L. (2008). The Data Mining and Knowledge Discovery Handbook. USA: Springer.

Maindonald J. (2006). Data Mining Methodological Weaknesses and Suggested Fixes. Proc. Fifth Australasian Data Mining Conference (AusDM2006).

Mitchell T. M. (1997). Machine Learning. McGraw Hill.

MONK (2010). Analytics: Decision Tree Induction. Retrieved January 1, 2010 from http://gautam.lis.illinois.edu/monkmiddleware/public/analytics/decisiontree.html.

Object Management Group (2010). Whitepapers. Retrieved January 1, 2010 from http://www.omg.org/news/whitepapers/index.htm.

Oliveira S.R.M & Zaiane O.R. (2004). Toward Standardization in Privacy-Preserving Data Mining. Retrieved January 1, 2010 from http://webdocs.cs.ualberta.ca/~zaiane/postscript/dm-ssp04.pdf.

Pitt E., Nayak R. (2007). The Use of Various Data Mining and Feature Selection Methods in the Analysis of a Population Survey Dataset. Proc. 2nd International Workshop on Integrating Artificial Intelligence and Data Mining (AIDM 2007).

Putten P. v.d., Meng L., Kok J. N. (2008). Profiling novel classification algorithms: Artificial Immune System. Retrieved January 1, 2010 from http://www.liacs.nl/~putten/library/200809vdPuttenKokMengCIS.pdf.

Quinlan J. R. (1994). Comparing connectionist and symbolic learning methods. In S. J. Hanson, G. A. Drastal, & R. L. Rivest. (Ed.), Computational Learning Theory and Natural Learning Systems: Constraints and Prospect. (pp. 445–446). Cambridge: MIT Press.

Sculley D., Malkin R., Basu S. & Bayardo R. J. (2009). Predicting Bounce Rates in Sponsored Search Advertisements. Retrieved January 1, 2010 from http://www.bayardo.org/ps/kdd2009.pdf.

Shih Y. S. (1997). QUEST Classification Tree (version 1.9.2). Retrieved January 1, 2010 from http://www.stat.wisc.edu/~loh/quest.html.

Simba Technologies (2010). XML for Analysis. Retrieved January 1, 2010 from http://www.xmlforanalysis.com.

SPSS (2010). CHAID and Exhaustive CHAID Algorithms Retrieved January 1, 2010 from http://support.spss.com/ProductsExt/SPSS/Documentation/Statistics/algorithms/14.0/TREE-CHAID.pdf.

Tang Z. H. & MacLennan J. (2005). Data Mining with Microsoft SQL Server, Wiley.

Tolon M. & Tosunoğlu N.G. (2008). Alışveriş Merkezi Tüketicilerinin Tatmininin Yapay Sinir Ağları Yöntemiyle Ölçülmesi. Gazi Üniversitesi İktisadi ve İdari Bilimler Fakültesi Dergisi, 10(2), 247-259.

UCI Machine Learning Repository (2010). Retrieved January 1, 2010 from http://archive.ics.uci.edu/ml.