DESIGN AND DEVELOPMENT OF DATA MINING MODELS FOR THE PREDICTION OF MANPOWER PLACEMENT IN THE TECHNICAL DOMAIN

Thesis submitted to

Cochin University of Science and Technology

in fulfilment of the requirements for

the award of the degree of

Doctor of Philosophy

in Computer Science and Engineering

by

Mr. Sudheep Elayidom. M

Under the supervision of

Dr. Sumam Mary Idicula
Professor, Department of Computer Science

DEPARTMENT OF COMPUTER SCIENCE
COCHIN UNIVERSITY OF SCIENCE AND TECHNOLOGY
KOCHI 682 022

2012

CERTIFICATE

This is to certify that the thesis, entitled "**DESIGN AND DEVELOPMENT OF DATA MINING MODELS FOR THE PREDICTION OF MANPOWER PLACEMENT IN THE TECHNICAL DOMAIN**" submitted to Cochin University of Science and Technology, in partial fulfilment of the requirements for the award of the Degree of Doctor of Philosophy in Computer Science and Engineering is a record of original research work done by Mr. Sudheep Elayidom.M (REG NO:3438), during the period (2007-2012) of his study in Department of Computer Science at Cochin University of Science and Technology, under my supervision and guidance and the thesis has not formed the basis for the award of any Degree/ Diploma/Associateship/Fellowship or other similar title to any candidate of any university.

Signature of the guide

DECLARATION

I, Sudheep Elayidom, M. hereby declare that the thesis, entitled "**DESIGN AND DEVELOPMENT OF DATA MINING MODELS FOR THE PREDICTION OF MANPOWER PLACEMENT IN THE TECHNICAL DOMAIN**" submitted to Cochin University of Science and Technology, in partial fulfilment of the requirements for the award of the Degree of Doctor of Philosophy in Computer Science and Engineering is a record of original research work done by me during the period 2007-2012 under the guidance of Dr. Sumam Mary Idicula, Professor, Department of Computer Science, Cochin University of Science and Technology, and it has not formed the basis for the award of any Degree/Diploma/Associateship/Fellowship or other similar title to any candidate of any university.

Signature of the Candidate

ACKNOWLEDGEMENT

Without God, I am nothing, it was He who began this work, gave me timely intuitions, walked along with me throughout the research work and enabled me to complete it successfully.

I am deeply indebted to my research supervisor **Dr. Sumam Mary Idicula**, Professor, Department of Computer Science, Cochin University of Science and Technology for her sincere and effective guidance. More than a supervisor she has been my mentor throughout my research work. Her timely suggestions based on her experiences on the research methodologies and directions have really made my path much easier. Without her knowledge, patience and support, this thesis would not have taken up this form.

I express my sincere thanks to **Dr. K. Poulose Jacob**, Head, Department of Computer Science, Cochin University of Science and Technology, for his support for the official procedures involved in the PhD program in the department.

I wish to express my heartfelt gratitude to **Dr. Asha Gopalakrishnan, Dr P.G. Sankaran and Ms. Sreelakshmi**, Department of Statistics, Cochin University of Science and Technology who gave me directions to have a good foundation in statistics in the beginning phases of my research work.

It's a great privilege to extend my deep sense of gratitude and sincere thanks to **Mr. Joseph Alexander and his staff** at the Nodal center, Cochin University of Science and Technology for providing me authentic data for my research work.

I would like to extend my sincere thanks to my Head and Principal **Dr. David Peter. S.,** for giving me the spare time for doing part time research work amidst official duties.

I would like to offer my sincere thanks to **Dr. B. Kannan, Ms. Simily, Dr. Philip Samuel, Dr. Sheena Mathew, Dr. Rekha. K. James and Dr. Shahana T.K**, faculties of Cochin University of science and technology for providing their suggestions on various research directions and thesis preparation.

Finally I would like to thank all my family members, friends, teaching and non teaching staff of Cochin University of Science and Technology who have given me support during my long research period.

Sudheep Elayidom. M

ABSTRACT

Data mining is one of the hottest research areas nowadays as it has got wide variety of applications in common man's life to make the world a better place to live. It is all about finding interesting hidden patterns in a huge history data base. As an example, from a sales data base, one can find an interesting pattern like "people who buy magazines tend to buy news papers also" using data mining. Now in the sales point of view the advantage is that one can place these things together in the shop to increase sales. In this research work, data mining is effectively applied to a domain called placement chance prediction, since taking wise career decision is so crucial for anybody for sure. In India technical manpower analysis is carried out by an organization named National Technical Manpower Information System (NTMIS), established in 1983-84 by India's Ministry of Education & Culture. The NTMIS comprises of a lead centre in the IAMR, New Delhi, and 21 nodal centres located at different parts of the country. The Kerala State Nodal Centre is located at Cochin University of Science and Technology.

In Nodal Centre, they collect placement information by sending postal questionnaire to passed out students on a regular basis. From this raw data available in the nodal centre, a history data base was prepared. Each record in this data base includes entrance rank ranges, reservation, Sector, Sex, and a particular engineering.

From each such combination of attributes from the history data base of student records, corresponding placement chances is computed and stored in the history data base. From this data, various popular data mining models are built and tested. These models can be used to predict the most suitable branch for a particular new student with one of the above combination of criteria.

Also a detailed performance comparison of the various data mining models is done.

This research work proposes to use a combination of data mining models namely a hybrid stacking ensemble for better predictions. A strategy to predict the overall absorption rate for various branches as well as the time it takes for all the students of a particular branch to get placed etc are also proposed.

Finally, this research work puts forward a new data mining algorithm namely C 4.5 * stat for numeric data sets which has been proved to have competent accuracy over standard benchmarking data sets called UCI data sets. It also proposes an optimization strategy called parameter tuning to improve the standard C 4.5 algorithm.

As a summary this research work passes through all four dimensions for a typical data mining research work, namely application to a domain, development of classifier models, optimization and ensemble methods.

CONTENTS

LIST OF TABLES

LIST OF FIGURES

NOTATIONS

ACC	Accuracy of a data mining model
TP	The recall or true positive rate
FP	The false positive rate
TN	The true negative rate
FN	The true negative rate
ROC	Receiver operator Characteristic test
ANOVA	Analysis of variance
O	Big-Oh notation for time complexity
σ^2	Statistical Variance
μ	Statistical Mean
sgn()	signum function to extract sign of real numbers

Chapter 1

Introduction

This chapter explains the motivation behind this research work by blending data mining techniques into a social science problem like employment prediction. It also shows the contribution of this research work to the field of data mining and concludes with a description on organisation of this thesis.

1.1 Background

Any science or technology gets the real meaning when common man in the society can make use out of it and the world becomes a better world. All the researches that happen in the world have or should have such a strong motivation to help the human kind to have a better life. As all of us know a person's success in life in a long run is largely dependent on the wise decision he/she makes regarding which career one should settle down. Usually in Indian context, this decision is made when one chooses an undergraduate course, during an early phase of one's life. In India, the usual culture existing in the society unlike developed countries like U.S or U.K is that they pursue their normal studies up to tenth standard where they study almost all subjects. Then in levels 11^{th} and 12^{th} they focus more on mathematics, biology or humanities. After their 12^{th} standard, they write a common entrance examination for entry into engineering courses and another examination for entry into medical courses. Admission to engineering and medical courses depends on the rank a student gets in the common entrance exam conducted by state/central government. In this context, using the past and present data available and various computational techniques developed, "how can a teenager be helped in choosing his future career option?" is a relevant question. Hence guiding a student to the most prospective path is a social science problem of high relevance.

1.2 Motivation and scope

Today academic institutions and their auxiliary centres maintain huge amount of information regarding their academic activities like student performance and placement history. Data mining can be very effectively used in analysing such socially relevant data. In one of the existing works in literature, student retention analysis has been carried out in which, a system has been developed to find out those students who will actually finish the course and pass out from an institution. This information is useful in places, where there are many cases of students leaving an institution without completing the course creating waste of institutional resources [20]. In another work, questions like “which are the courses that are usually selected by top performers?” etc are addressed [30].

But it can be seen that any notable work that uses this valuable information for placement chance prediction has not been reported.

This placement history if properly maintained, are valuable sources of information which can give light to questions like “what is the prospect of getting a placement if a particular branch of study is selected?”

This research work is an attempt to help the prospective students to make wise career decisions using data mining technology. Popular Data mining models like decision trees, neural networks, Naïve Bayes classifier etc. are used. Effort is taken to improve the overall performance of these models and to find an optimum model for this particular problem. A decision is made based on data like Entrance Rank, Gender (M/F), Sector (Rural/Urban), Reservation category (General, OBC (Other Backward Castes), SC/ST (Scheduled castes / tribes) and branch (Civil Engineering, Computer Science

and Engineering and so on). As explained in chapter 3, during data pre-processing, the importance of above attributes in deciding the target classes was scientifically analyzed using a statistical technique called chi-square analysis. Using this technique many other irrelevant attributes like “loans taken by student”, “special training undergone by student” etc were filtered of since they were not the deciding factors for the target classes. It can be scientifically as well as socially verified that these attributes are relevant as far as placement chances are concerned. For example, attributes like reservation, background etc have a strong impact on getting a placement in the current social set up. In India for certain types of technical jobs they prefer males than females as well as certain branches are having higher degree of placement. Also a person’s entrance rank is the figure based on which one gets admitted to an institution. A classification model was built which can predict the placement chances for a particular student’s input, which is very valuable information for a student to choose a particular branch. A procedure to predict the overall absorption rate for a branch in a future year was developed. Absorption rate means the total percentage of students who got placement among the passed out students for a particular branch in a particular year. Also a procedure to find waiting time to get placement for all of the passed out students for any branch in a year, was addressed. Absorption rate and waiting time calculations were the two sub problems addressed in this research work.

1.3 Objective

As pointed out in earlier section, this research work is an attempt to help the prospective young students to choose a bright career. The domain selected for this research work is the engineering courses offered in Kerala. In Cochin University of science and technology, Kerala, India, there is a centre

called “Nodal Centre” which is dedicated and responsible for the systematic collection of data from passed out students from various engineering colleges in Kerala State. This authentic information is trusted and used by the AICTE (All India Council for Technical education) an institute, responsible for the management of engineering education in India.

The main research question for this work is how efficient data mining models can be designed and developed for the prediction of manpower placement in the technical domain. The answer to this can be found out by answering the following sub questions.

1. From the large volume of data available in the nodal centre, how to extract only the data suitable for this research work?
2. How can the dimensionality of the data be reduced?
3. What types of mining algorithms should be used?
4. How can the performance evaluation of the models be done?
5. What are the steps to be taken for improving the performance of the models?
6. What are the possible offshoots of this research work?

1.4 Dataset

As pointed out in previous section the dataset for this research work is the systematic data collected from the passed out engineering students of Kerala state by the Nodal centre in Cochin University of Science and Technology, Kerala, India.

Every year the nodal centre sends postal questionnaire to the passed out students containing important questions like “whether placed or not”, reservation, residing demographic information and so on. This data is collected and prepared in MS EXCEL format and then ported to various

data/data base formats for data analysis activities. Usually the technical analysis of the data is done after 4 years from the year a particular batch of students has passed out. For e.g. in 2011, nodal centre analyses the data of students passed in 2007. This is to ensure that by this settling time, majority of the students are well placed or might have opted for higher studies. The data collection form in Nodal centre that is used for collecting data from the passed out students is shown in Appendix A.

Even though this particular research work considers the engineering domain, the data mining models and procedures used may be extended to other domains like medical, financial etc for predicting the employment chances in these domains too. Depending on the history data, the relevant attributes and best models performances may vary.

1.5 Contribution

As pointed out at the beginning of this chapter, the worth of any research work is judged by the value it has contributed to the common man's life. In this context this research work analyses the effectiveness of data mining techniques to guide the prospective students to choose a bright technical career.

This work also probed whether combing data mining models could give a better performance. It was proved that combining classifiers give better performance than selecting the single best among base level classifiers. Also a hybrid ensemble classifier having best performance suitable for this domain was developed.

Apart from the original goal, a new algorithm for handling numeric data sets for decision tree construction and an optimization strategy for decision trees was developed. Also some useful analysis that may help the

government in planning and governance were also carried out. They include lot of statistical information like which are the best engineering branches that are in demand, whether to increase student intakes next year, how many students are placed in a particular branch in a particular year, how many will be placed in a particular branch in a coming year, what can be the approximate waiting time for getting a placement in a particular branch after graduation etc. Currently data mining research works proceed mainly in 4 directions.

1. Applying data mining principles to a new application domain
2. Developing a new data mining algorithm/classifier
3. Optimizing existing models
4. Combining classifiers called meta learners

It can be seen that this research work has made contributions in each of these data mining research directions.

1.6 Structure of the thesis

The layout of the thesis is as follows:

- Chapter 1 gives an introduction to this research work giving thrust on motivation, objective, data sets used and the main contributions of this work.
- Chapter 2 is a literature survey on the various data mining models and algorithms that are popular in different application areas.
- Chapter 3 discusses the various data pre-processing techniques used in this research work and the data portability issues.
- Chapter 4 describes the experimental set up for the data mining models used. The classifiers were developed using different file formats, software packages etc, and are discussed in detail in this

chapter. Also it describes a parameter optimization strategy for C 4.5 decision tree algorithm.

- Chapter 5 deals with performance evaluation of the various models developed.
- Chapter 6 describes the hybrid stacking ensemble approach for improving classifier accuracy. A discussion on the problem of selecting the best base level classifier is also given in this chapter.
- Chapter 7 discusses the absorption rate prediction technique which can give some light for the government in deciding the student intake for each branch.
- Chapter 8 describes the development of a new algorithm namely C 4.5 * stat for numeric datasets.
- Chapter 9 concludes the thesis and puts forward few future directions in which this research work may be extended later.

1.7 Chapter summary

The background, motivation, scope and objectives of the work are clearly specified in this chapter. Technical manpower details supplied by Nodal Centre, Cochin University of science and technology, Kerala, India were used as the data sets for this research work. The chapter concludes by giving the contribution of this research work.

Chapter 2

Literature review on data mining research

A literature survey on the research and developments in the data mining domain is given in this chapter. The chapter is organised as individual sections for each of the popular data mining models and respective literature is given in each section.

2.1 Data mining concepts

Data mining is a collection of techniques for efficient automated discovery of previously unknown, valid, novel, useful and understandable patterns in large databases. The patterns must be actionable so that they may be used in an enterprise's decision making process. It is usually used by business intelligence organizations, and financial analysts, but it is increasingly used in the sciences to extract information from the enormous data sets generated by modern experimental and observational methods [6].

A typical example for a data mining scenario may be "In the context of a super market, if a mining analysis observes that people who buy pen tend to buy pencil too, then for better business results the seller can place pens and pencils together."

Data mining strategies can be grouped as follows:

- **Classification**- Here the given data instance has to be classified into one of the target classes which are already known or defined [19, 20]. One of the examples can be whether a customer has to be classified as a trustworthy customer or a defaulter in a credit card transaction data base, given his various demographic and previous purchase characteristics.
- **Estimation**- Like classification, the purpose of an estimation model is to determine a value for an unknown output attribute. However, unlike classification, the output attribute for an estimation problem are numeric rather than categorical. An example can be "Estimate the salary of an individual who owns a sports car?"
- **Prediction**- It is not easy to differentiate prediction from classification or estimation. The only difference is that rather than determining the

current behaviour, the predictive model predicts a future outcome. The output attribute can be categorical or numeric. An example can be "Predict next week's closing price for the Dow Jones Industrial Average". [53, 65] explains the construction of a decision tree and its predictive applications.

- **Association rule mining** -Here interesting hidden rules called association rules in a large transactional data base is mined out. For e.g. the rule {milk, butter->biscuit} provides the information that whenever milk and butter are purchased together biscuit is also purchased, such that these items can be placed together for sales to increase the overall sales of each of the items [2, 40].
- **Clustering**- Clustering is a special type of classification in which the target classes are unknown. For e.g. given 100 customers they have to be classified based on certain similarity criteria and it is not preconceived which are those classes to which the customers should finally be grouped into.

The main application areas of data mining are in Business analytics, Bioinformatics [33, 34, and 64], Web data analysis, text analysis, social science problems, biometric data analysis and many other domains where there is scope for hidden information retrieval [42]. Some of the challenges in front of the data mining researchers are the handling of complex and voluminous data, distributed data mining, managing high dimensional data and model optimization problems.

In the coming sections the various stages occurring in a typical data mining problem are explained. The various data mining models that are

commonly applied to various problem domains are also discussed in detail in the coming sections.

2.2 Data mining stages

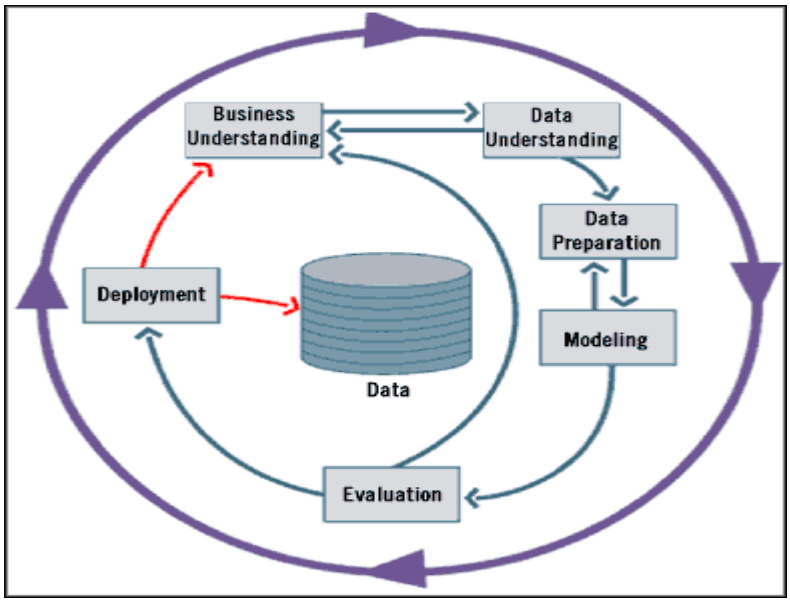


Figure 2.1: Data mining stages

Any data mining work may involve various stages as shown in Figure 2.1. Business understanding involves understanding the domain for which the data mining has to be performed. The various domains can be financial domain, educational data domain and so on. Once the domain is understood properly, the domain data has to be understood next as shown in figure 2.1. Here relevant data in the needed format will be collected and understood.

Data preparation or pre-processing is an important step in which the data is made suitable for processing. This involves cleaning data, data transformations, selecting subsets of records etc. When data is prepared there

are two stages, namely selection and transformation. Data selection means selecting data which are useful for the data mining purpose. It is done by selecting required attributes from the database by performing a query. Data transformation or data expression is the process of converting the raw data into required format which is acceptable by data mining system. For e.g., Symbolic data types are converted into numerical form or categorical form.

Data modelling involves building the models like decision tree, neural network etc from above pre-processed data.

2.3 Data mining models

There are many popular models that can be effectively used in different data mining problems. Decision trees, neural networks, Naive Bayes classifier, Lazy learners, Support vector machines, and regression based classifiers are few among them. Depending upon the type of application, nature of data and attributes, one can decide which can be the most suited model. Still there is no clear cut answer to the question of which is the best data mining model. One can only say for a particular application one model is better than the other.

2.3.1 Decision trees

The decision tree is a popular classification method. It is a tree like structure where each internal node denotes a decision on an attribute value. Each branch represents an outcome of the decision and the tree leaves represent the classes [54, 56]. Decision tree is a model that is both predictive and descriptive. A decision tree displays relationships found in the training data.

In data mining and machine learning, a decision tree is a predictive model; that is, a mapping from observations about an item to conclusions about its target value [62]. More descriptive names for such tree models are classification tree (discrete outcome) or regression tree (continuous outcome). In these tree structures, leaves represent classifications and branches represent conjunctions of features that lead to those classifications. The machine learning technique for inducing a decision tree from data is called decision tree learning.

Given a set of examples (training data) described by some set of attributes (ex. Sex, rank, background) the goal of the algorithm is to learn the decision function stored in the data and then use it to classify new inputs. The discriminative power of each attribute that can best split the dataset is done either using the concept of information gain or Gini index. Popular decision tree algorithms like ID3, C4.5, C5 etc use information gain to select the next best attribute whereas popular packages like CART, IBM intelligent miner etc use the Gini index concept.

Information gain

A decision tree can be constructed top-down using the information gain in the following way:

1. Let the set of training data be S. Continuous attributes if any, should be made discrete, before proceeding further. Once this is done put all of S in a single tree node.
2. If all instances in S are in same class, then stop
3. Split the next node by selecting an attribute A , for which there is maximum information gain
4. Split the node according to the values of A

5. Stop if either of the following conditions is met, otherwise continue with step 3:
   (a) If this partition divides the data into subsets that belong to a single class and no other node needs splitting.
   (b) If there are no remaining attributes on which the sample may be further divided.

In order to build a decision tree, it is needed to be able to distinguish between 'important' attributes, and attributes which contribute little to the overall decision process. Intuitively, the attribute which will yield the most information should become our first decision node. Then, the attribute is removed and this process is repeated recursively until all examples are classified. This procedure can be translated into the following steps:

$$I(P(v_1), \ldots, P(v_i)) = \sum_{i=1}^{n} -P(v_i) log_2 P(v_i) \quad \textbf{- (2.1)}$$

$$Gain(Attribute) = I\left(\frac{p}{p+n}, \frac{n}{p+n}\right) - \sum_{i=1}^{v} \frac{p_i + n_i}{p+n} I\left(\frac{p_i}{p_i+n_i}, \frac{n_i}{p_i+n_i}\right) \quad \textbf{- (2.2)}$$

First, the formula calculates the information required to classify the original dataset (p - # of positive examples, n - # of negative examples). Then, the dataset is split on the selected attribute (with *v* choices) and the information gain is calculated. This process is repeated for every attribute, and the one with the highest information gain is chosen to be the decision node.

Even though algorithms like ID3, C4.5, C5 etc uses information gain concepts; there are few differences between them. [6] Gives a comparison of various decision tree algorithms and their performances.

C4.5 handles both continuous and discrete attributes. In order to handle continuous attributes, C4.5 creates a threshold and then splits the list into those

whose attribute value is above the threshold and those that are less than or equal to it. In order to handle training data with missing attribute values, C4.5 allows attribute values to be marked as "?" for missing. Missing attribute values are simply not used in gain or entropy calculations. C4.5 uses pruning concepts. The algorithm goes back through the tree once it's been created and attempts to remove branches that do not help by replacing them with leaf nodes. C5.0 is significantly faster and memory efficient than C4.5. It also uses advanced concepts like boosting, which is later discussed in the thesis.

Algorithms based on information gain theory tend to favour those attributes that have more values where as those based on Gini index tend to be weak when number of target classes are more. Hence Nowadays, researchers are trying to optimize the decision tree performances by using techniques like pre-pruning(removing useless branches of the tree as the tree is built), post pruning(removing useless branches of the tree after the tree is built) etc.

Other variants of decision tree algorithms include CS4 [34, 35], as well as Bagging [9], Boosting [17, 47], and Random forests [10]. A node is removed only if the resulting pruned tree performs no worse than the original, over the cross validation set [63]. Since the performance is measured on validation set, this pruning strategy suffers from the disadvantage that the actual tree is based on less data. However, in practice, C4.5 makes some estimate of error based on training data itself, using the upper bound of a confidence interval (by default is 25%) on the re-substitution error. The estimated error of the leaf is within one standard deviation of the estimated error of the node. Besides reduced error pruning, C4.5 also provides another pruning option known as sub tree raising. In sub tree raising, an internal node might be replaced by one of nodes below and samples will be redistributed. A

detailed illustration on how C4.5 conducts its post-pruning is given in [44, 55]. Other algorithms for decision tree induction include ID3 (predecessor of C4.5) [42], C5.0 (successor of C4.5), CART (classification and regression trees) [8], LMDT (Linear Machine Decision Trees) [63], OC1 (oblique classifier) and so on.

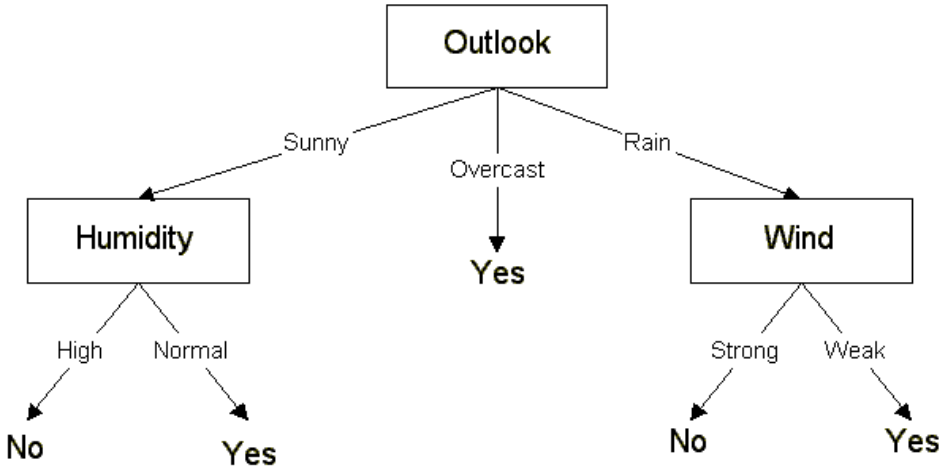


Figure 2.2: A Sample decision tree-Partial view

Figure 2.2 shows the partial view of a sample decision tree. One of the decision rule it provides is, if Outlook is sunny and humidity is normal, one can go ahead to play.

Usually, when a decision tree is built from the training set, it may be over fitted, which means that the tree performs well for training data only. Its performance will not be that good with unseen data. So one can "Shave off" nodes and branches of a decision tree, essentially replacing a whole sub tree by a leaf node, if it can be established that the expected error rate in the sub tree is greater than that in a single leaf. This makes the classifier simpler.

Pruning is a technique to make an over fitted decision tree simpler and more general. In post pruning, after the tree is fully grown, with some test data, some branches are removed and smaller trees are derived. Now the subset tree with minimum error and simplicity is selected as the final tree.

Another approach is called pre-pruning in which the tree construction is halted early. Essentially a node is not split if this would result in the goodness measure of tree falling below a threshold. It is however, quite difficult to choose an appropriate threshold. The classic decision tree algorithm named C4.5 was proposed by Quinlan [44]. Majority of the research works in decision trees are concerned with the improvement in the performance using optimization techniques such as pruning. [20] Reports a work dealing with understanding student data using data mining. Here decision tree algorithms are used for predicting graduation, and for finding factors that lead to graduation.

[22] Provides an overview about how data mining and knowledge discovery in databases are related to each other and to other fields, such as machine learning, statistics, and databases. [8] Suggests methods to classify objects or predict outcomes by selecting from a large number of variables, the most important ones in determining the outcome variable. [31] Discusses how performance evaluation of a model can be done by using confusion matrix, which contains information about actual and predicted classifications done by a classification system.

In a typical classification task, data is represented as a table of samples, which are also known as instances. Each sample is described by a fixed number of features which are known as attributes and a label that indicated its

class [27]. [52] Describes a special technique that uses genetic algorithms for attribute analysis.

Data mining can extract implicit, previously unknown and potentially useful information from data [32, 62]. It is a learning process, achieved by building computer programs to seek regularities or patterns from data automatically. Machine learning provides the technical basis of data mining. Classification learning is a generalization of concept learning [63]. The task of concept learning is to acquire the definition of a general category given a set of positive and negative training examples of the category [37]. Thus, it infers a Boolean-valued function from the training instances. As a more general form of concept learning, classification learning can deal with more than two class instances. In practice, the learning process of classification is to find models that can separate instances in the different classes using the information provided by training instances.

2.3.2 Neural networks

Neural networks offer a mathematical model that attempts to mimic the human brain [5, 15]. Knowledge is represented as a layered set of interconnected processors, which are called neurons. Each node has a weighted connection to other nodes in adjacent layers. Individual nodes take the input received from connected nodes and use the weights together with a simple function to compute output values. Learning in neural networks is accomplished by network connection weight changes while a set of input instances is repeatedly passed through the network. Once trained, an unknown instance passing through the network is classified according to the values seen at the output layer. [51,57] surveys existing work on neural network construction, attempting to identify the important issues involved, directions

the work has taken and the current state of the art. Typically, a neural network model is having a configuration as shown in figure 2.3 in its basic form. Neurons only fire when input is bigger than some threshold. It should, however, be noted that firing doesn't get bigger as the stimulus increases, it is an all or nothing arrangement [28].

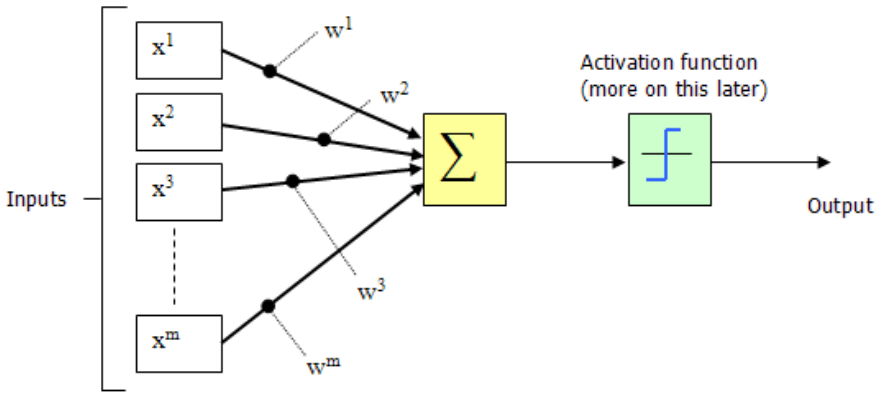


Figure 2.3: A neural network configuration

Suppose a firing rate is there at each neuron. Also suppose that a neuron connects with **m** other neurons and so receives m-many inputs "x_1 …. … x_m. This configuration is actually called a **Perceptron**.

In 1962, Rosenblatt proposed the perceptron model. It was one of the earliest neural network models. A Perceptron models a neuron by taking a weighted sum of inputs and sending the output 1, if the sum is greater than some adjustable threshold value otherwise it sends 0. This is the all or nothing spiking described in the previous paragraph. It is also called an activation function.

The inputs **(x_1, x_2, x_3 …x_m)** and connection weights **(w_1, w_2, w_3…w_m)** in Figure 2.4 are typically real values, both positive (+) and negative (-). If the

feature of some x_i tends to cause the perceptron to fire, the weight w_i will be positive; if the feature x_i inhibits the perceptron, the weight w_i will be negative. The perceptron itself consists of weights, the summation processor, and an activation function, and an adjustable threshold processor (called bias). For convenience the normal practice is to treat the bias, as just another input. Figure 2.4 illustrates the revised configuration with bias.

The bias can be thought of as the propensity (a tendency towards a particular way of behaving) of the perceptron to fire irrespective of its inputs. The perceptron configuration network shown in Figure 2.4 fires if the weighted sum > 0, or in mathematical terms, it can be represented as in (2.3)

$$\text{Weighted sum} = \sum_{i=1}^{m} bias + (w^i x^i) \qquad \textbf{- (2.3)}$$

Activation function: The activation usually uses one of the following functions.

Sigmoid function: The stronger the input is, the faster the neuron fires. The sigmoid is also very useful in multi-layer networks, as the sigmoid curve allows for differentiation (which is required in Back Propagation training of multi layer networks). In mathematical terms, it can be represented as in (2.4)

$$\mathbf{f(x) = 1/(1+e^{-x})} \qquad \textbf{- (2.4)}$$

Step function: A step function is a basic on/off type function, if 0>x then 0, else if x>=0 then 1. Hence depending on the type of input, output and problem domain, suited functions are adopted at respective layers.

Learning can be of two types. Supervised and unsupervised. As an example of supervised learning one can consider a real world example of a

baby learning to recognise a chair. He is taught with many objects that are chairs and that are not chairs. After this training, when a new object is shown, he can correctly identify it as a chair or not. This is exactly the idea behind the perceptron. As an example of unsupervised learning, one can consider a six months old baby recognising his mother. Here a supervisor does not exist. All classification algorithms are examples of supervised learning. But clustering is unsupervised learning, where a model does not exist based on which classification has to be performed.

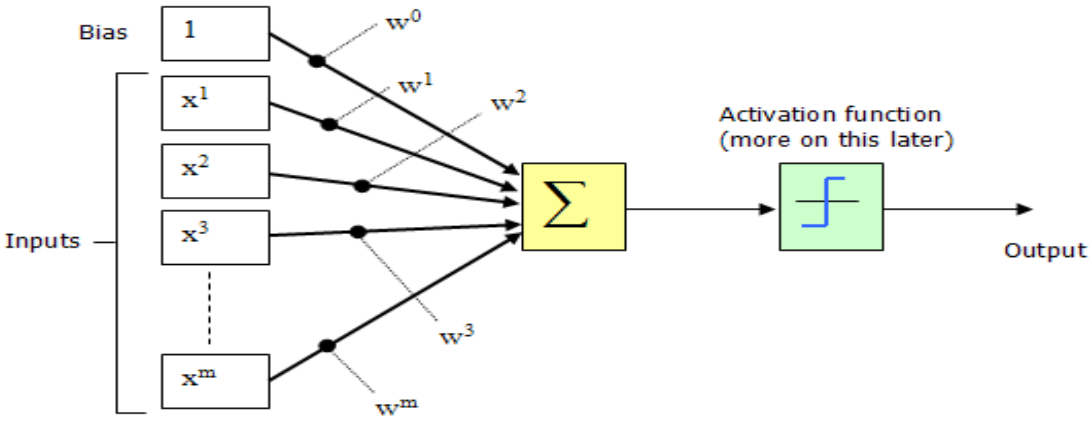


Figure 2.4: Artificial Neuron configurations, with bias as additional Input

Perceptron learning: The Perceptron is a single layer neural network whose weights and biases are trained to produce a correct target vector when presented with the corresponding input vector. The training technique used is called the perceptron learning rule. The perceptron generated great interest due to its ability to generalize from its training vectors and work with randomly distributed connections. Perceptrons are especially suited for simple problems in pattern classification. Suppose the data can be separated perfectly into two

groups using a hyper plane, it is said to be linearly separable [67]. If the data is linearly separable, the perceptron learning rule can be applied which is given below.

The Learning rule: The perceptron is trained to respond to each input vector with a corresponding target output of either 0 or 1. The learning rule converges on a solution in finite time if a solution exists.

The learning rule can be summarized in the following two equations:

$$b = b + [T - A] \quad \textbf{- (2.5)}$$

For all inputs i:

$$W(i) = W(i) + [T - A] * P(i) \quad \textbf{- (2.6)}$$

Where **W** is the vector of weights, **P** is the input vector presented to the network, **T** is the correct result that the neuron should have shown, **A** is the actual output of the neuron, and **b** is the bias.

Training: Vectors from a training set are presented to the network one after another. If the network's output is correct, no change is made. Otherwise, the weights and biases are updated using the perceptron learning rule (as shown above). When each epoch (an entire pass through all of the input training vectors is called an epoch) of the training set has occurred without error, training is complete.

When the training is completed, if any input training vector is presented to the network and it will respond with the correct output vector. If a vector, **P**, not in the training set is presented to the network, the network will tend to exhibit generalization by responding with an output similar to target vectors for input vectors close to the previously unseen input vector **P**. The

transfer function used in the Hidden layer is Log- Sigmoid while that in the output layer is Pure Linear.

Neural networks are very good in classification and regression tasks where the attributes have missing values and also when the attribute values are categorical in nature [28, 67]. The accuracy observed is very good, but the only bottle neck is the extra training time and complexity in the learning process when the number of training set examples seems very high. [15, 51] Describe how neural networks can be applied in data mining. There are some algorithms for extracting comprehensible representations from neural networks. [5] Describes research to generalize and extend the capabilities of these algorithms. The application of the data mining technology based on neural network is vast. One such area of application is in the design of mechanical structure. [57] Introduces one such application of the data mining based on neural network to analyze the effects of structural technological parameters on stress in the weld region of the shield engine rotor in a submarine.

[70] Explains an application of neural networks in study of proteins. In that work, global adaptive techniques from multiple alignments are used for prediction of Beta-turns. This also introduces global adaptive techniques like Conjugate gradient method, Preconditioned Conjugate gradient method etc. An approach to discover symbolic classification rules using neural networks is discussed in [67]. Here, first the network is trained to achieve the required accuracy rate, and then activation values of the hidden units in the network are analyzed. Classification rules are generated using the result of this analysis.

Back propagation in multi layer perceptrons: Among the several neural network architectures, for supervised classification, feed forward multilayer

network trained with back propagation algorithm is the most popular. Neural networks, in which signal flows from input to output (forward direction) are called, feed forward neural networks. A single layer neural network can solve linearly separable problems only. But when the problem to be solved is more complex, a multilayer feed forward neural network can be used. Here there can be hidden layers other than input and output layers. There is a layer of weights between two adjacent levels of units (input, hidden or output). Back propagation is the training method most often used with feed-forward multi layer networks, typically when the problem to be solved is a non linear classification problem. In this algorithm, the error at the output layer is propagated back to adjust the weights of network connections. It starts by making weight modifications at the output layer and then moving backward through the hidden layers. Conjugate gradient algorithms are one of the popular back propagation algorithms which are explained in section 4.3.

[28] Proposes a 3 layer feed forward network to select the input attributes that are most useful for discriminating classes in a given set of input patterns. This is particularly helpful in feature selection. [60, 61] describe practical applications of neural networks in patient data analysis and [69] describes application of radial basis functions neural networks in protein sequence classification.

2.3.3 Naive Bayes classifier

This classifier offers a simple yet powerful supervised classification technique. The model assumes all input attributes to be of equal importance and independent of one another. Naive Bayes classifier is based on the classical Bayes theorem presented in 1763 which works on the probability theory. In simple terms, a naive Bayes classifier assumes that the presence (or

absence) of a particular feature of a class is unrelated to the presence (or absence) of any other feature. Even though these assumptions are likely to be false, Bayes classifier still works quite well in practice.

Depending on the precise nature of the probability model, Naive Bayes classifiers can be trained very efficiently in a supervised learning setting. In many practical applications, parameter estimation for Naive Bayes model uses the method of maximum likelihood.

Despite its simplicity, Naive Bayes can often outperform more sophisticated classification methods. An advantage of the Naive Bayes classifier is that it requires a small amount of training data to estimate the parameters (means and variances of the variables) necessary for classification. Because independent variables are assumed, only the variances of the variables for each class need to be determined and not the entire covariance matrix.

The classifier is based on Bayes theorem, which is stated as:

$$P(A|B) = P(B|A)*P(A)/P(B) \quad \textbf{- (2.7)}$$

Where

- P (*A*) is the prior probability or marginal probability of *A*. It is "prior" in the sense that it does not take into account any information about *B*.
- P (A|B) is the conditional probability of A, given B. It is also called the posterior probability because it is derived from or depends upon the specified value of B.
- P (B|A) is the conditional probability of B given A.
- P (B) is the prior or marginal probability of B, and acts as a normalizing constant. Bayes' theorem in this form gives a mathematical representation of how the conditional probability of event A given B is related to the converse conditional probability of B given A. Similarly numeric data

can be dealt with in a similar manner provided that the probability density function representing the distribution of the data is known. For example, suppose the training data contains a continuous attribute, *x*. First the data has to be segmented by the class, and then compute the mean and variance of *x* in each class. Let μ_c be the mean of the values in *x* associated with class *c*, and let σ_c^2 be the variance of the values in *x* associated with class *c*. Then, the probability of some value given a class, $P(x = v \mid c)$, can be computed by plugging *v* into the equation for a Normal distribution parameterized by μ_c and σ_c^2 . Equation (2.8) is used for numeric data assuming numeric data follows normal distribution.

$$P(x = v|c) = \frac{1}{\sqrt{2\pi\sigma_c^2}} e^{-\frac{(v-\mu_c)^2}{2\sigma_c^2}} \quad \text{- (2.8)}$$

The Naive Bayes classifier is illustrated with an example in section 2.3.3.1.

2.3.3.1 Example

Training Examples

Day	Outlook	Temperature	Humidity	Wind	PlayTennis
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

Figure 2.5: Training example for Naive Bayes classifier

Consider the data in figure 2.5 and assume one want to compute whether tennis can be played? (This is called as hypothesis H). Given the following weather conditions:

Outlook="sunny" Temp="cool" Humidity="high" Wind="Strong"

The above attribute conditions together can be called as evidence E.

It has to be found out whether tennis can be played. Hence one has to compute P(Playtennis=Yes|E).

Using the formulae in (2.7),

P(Playtennis=Yes|E)=P(E|Playtennis=yes)*P(Playtennis=yes)/P(E)

P(E|Playtennis=yes) is computed as follows:

P(Outlook=sunny|Playtennis=Y)*P(Temp=cool|palytennis=yes)*

P(Humidity="high|Playtennis=yes)*P(Wind="Strong|Playtennis=yes).

Which equals 2/9*3/9*3/9*3/9 = 0.008.

Also P(Playtennis=yes)=9/14=0.642

So P(Playtennis=Yes|E)=0.008*0.642/P(E)=0.005/P(E)

Similarly P(Playtennis=No|E), by applying values becomes 0.021/P(E)

HENCE ONE CAN NOT PLAY TENNIS!!!!!!

The same technique may be applied to different data sets and there are many packages like SPSS, WEKA etc that support Bayes classifier. Bayes classifier is so popular owing to its simplicity and efficiency. The classifier is explained in detail in [62]. [38] Describes the possibilities of real world applications of ensemble classifiers that involve Bayesian classifiers. In [16] a hybrid approach of classifiers that involves Naive Bayes classifier is also discussed.

Usually there are two active research areas in this comparison of classifier performances in a domain. One is optimizing a single classifier

performance by improvement of single classifier algorithm. The second approach is combining classifiers to improve accuracy.

Another important application area of data mining is association rule mining. It is all about finding interesting patterns usually purchase patterns in a sales transactional data bases. [4] Represents an application of association rules techniques in the data mining domain.

Another approach in improving classifier performances was in studies of applying special algorithms like genetic algorithms [52] and fuzzy logic [12] concepts into classifiers, which were found to be successful in improving accuracies. The section 2.3.4 presents a literature survey on the research that has happened in the domain of ensemble of classifiers.

2.3.4 Ensemble of classifiers

An ensemble of classifiers is an approach in which several classifiers are combined together to improve the overall classifier performance. It can be done in two ways, homogenous way in which same classifiers are combined and heterogeneous or hybrid in which different classifiers are combined.

"Whether an ensemble of homogenous or heterogeneous classifiers yields good performance" is always been a debatable question. [36] Proposes that depending on a particular application, an optimal combination of heterogeneous classifiers seems to perform better than homogenous classifiers.

[13, 50, 58] explain the possibilities of combining data mining models to get better results. In this work, the classifier performance is improved using the stacking approach. There are many strategies for combining classifiers like voting [7], bagging and boosting each of which may not involve much learning in the meta or combining phase. Stacking is a parallel combination of classifiers in which all the classifiers are executed parallel and learning takes

place at the meta level. Decision on which model or algorithm performs best at the meta level for a given problem is an active research area.

When only the best classifier among the base level classifiers is selected, the valuable information provided by other classifiers is ignored. In classifier ensembles which are also known as combiners or committees, the base level classifier performances are combined in some way such as voting or stacking.

It is found that stacking method is particularly suitable for combining multiple different types of models. Instead of selecting one specific model out of multiple ones, the stacking method combines the models by using their output information as inputs into a new space. Stacking then generalizes the guesses in that new space. The outputs of the base level classifiers are used to train a meta classifier. In this next level, it is ensured that the training data has accurately completed the learning process. For example, if a classifier consistently misclassifies instances from one region as a result of incorrectly learning the feature space of that region, the meta classifier may be able to discover this problem. This is the clear advantage of stacking over other methods like voting where no such learning takes place from the output of base level classifiers. Using the learned behaviours of other classifiers, it can improve such training deficiencies. Other bright future research areas in ensemble methods can be design and development of distributed, parallel and agent based ensemble methods for better classifier performances as pointed out in [16].

Ensemble of decision trees: Ensemble methods are learning algorithms that construct a set of classifiers and then classify new samples by taking a vote of their predictions [17]. Generally speaking, an ensemble method can increase

predictive performance over a single classifier. In [18], Dietterich explains why ensemble methods are efficient compared to a single classifier within the ensemble. Besides, plenty of experimental comparisons are performed to show significant effectiveness of ensemble methods in improving the accuracy of single base classifiers [3, 7, 9, 24, 43 and 46].

The original ensemble method is Bayesian averaging [17]. But bagging (bootstrap aggregation) and boosting are two of most popular techniques for constructing ensembles. [24] Explains the principle of ensemble of decision trees.

Bagging of decision trees: The technique of bagging (derived from bootstrap aggregation) was coined by Breiman [9], who investigated the properties of bagging theoretically and empirically for both classification and numeric prediction. Breiman had also proposed a classification algorithm namely random forest, as a variant of conventional decision tree algorithm, which is included in the WEKA data mining package [10].

Bagging of trees combines several tree predictors trained on bootstrap samples of the training data and gives prediction by taking majority vote. In bagging, given a training set with samples, a new training set is obtained by drawing samples uniformly with replacement. When there is a limited amount of training samples, bagging attempts to neutralize the instability of single decision tree classifier by randomly deleting some samples and replicating others. The instability inherent in learning algorithms means that small changes to the training set cause large changes in the learned classifier.

Boosting of decision trees: Unlike bagging, in boosting, every new tree that is generated, are influenced by the performance of those built previously. Boosting encourages new trees to become “experts” for samples handled

incorrectly by earlier ones [55]. When making classification, boosting weights a tree's contribution by its performance, rather than giving equal weight to all trees which is adopted by bagging. There are many variants on the idea of boosting. The version introduced below is called AdaBoostM1 which was developed by Freund and Schapire [23] and designed specifically for classification. The AdaBoostM1 algorithm maintains a set of weights over the training data set and adjusts these weights after iterations of the base classifier. The adjustments increase the weight of samples that are misclassified and decrease the weight of samples that are properly classified. By weighting samples, the decision trees are forced to concentrate on those samples with high weight. There are two ways that AdaBoostM1 manipulates these weights to construct a new training set to feed to the decision tree classifier [55]. One way is called boosting by sampling, in which samples are drawn with replacement using probability proportional to their weights. Another way is boosting by weighting, in which the presence of sample weights changes the error calculation of tree classifier. That is, using the sum of the weights of the misclassified samples divided by the total weight of all samples, instead of the fraction of samples that are misclassified. [48, 49] give introduction and applications of boosting. In [43], C4.5 decision tree induction algorithm is implemented to deal with weighted samples

CS4— a new method of ensemble of decision trees: CS4 stands for cascading-and-sharing for ensemble of decision trees. It is a newly developed classification algorithm based on an ensemble of decision trees. The main idea of this method is to use different top-ranked features as the root node of a decision tree in an ensemble (also named as a committee) [35]. Different from bagging or boosting which uses bootstrapped data, CS4 always builds decision

trees using exactly the same set of training samples. The difference between this algorithm and Dietterich's randomization trees is also very clear. That is, the root node features of CS4 induced trees are different from each other while every member of a committee of randomized trees always shares the same root node feature (the random selection of the splitting feature is only applied to internal nodes). On the other hand, compared with the random forests method which selects splitting features randomly, CS4 picks up root node features according to their rank order of certain measurement (such as entropy, gain ratio). Thus, CS4 is claimed as a novel ensemble tree method. Breiman noted in [9] that most of the improvement from bagging is evident within ten replications. Therefore, 20 is set (default value is 10) as the number of bagging iterations for Bagging classifier, the number of maximum boost iterations for AdaBoostM1, and the number of trees in the forest for Random forests algorithm

Example applications: Classifier ensembles have wide applications ranging from simple applications to remote sensing. [25, 26] explain ensemble of classifiers which classifies very high resolution remote sensing images from urban areas.

[11] Describes a practical application of ensemble of classifiers in the domain of intrusion detection in mobile ad-hoc networks. There they use ensemble of classifiers for predicting intrusion attempts. In [66] ensemble based classification methods were applied to spam filtering. [59] Describes a cost sensitive learning approach which resembles ensemble methods, for recurrence prediction of breast cancer.

All the ensemble algorithms discussed in this section are based on ‘‘passive’’ combining, in that the classification decisions are made based on

static classifier outputs, assuming all the data is available at a centralized location at the same time. Distributed classifier ensembles using ‘‘active’’ or agent-based methods can overcome this difficulty by allowing agents to decide on a final ensemble classification based on multiple factors like classification and confidence [1]. Another application is in face recognition. Chawla and Bowyer [14] addressed the standard problem of face recognition but under different lighting conditions and with different facial expressions. The authors of [41] decided to theoretically examine which combiners do the best job of optimizing the Equal Error Rate (EER), which is the error rate of a classifier when the false acceptance rate is equal to the false rejection rate. [56, 62] explain the practical methods of how to use Weka for advanced data mining tasks including how to include java libraries in Weka.

2.3.5 Other data mining models

Even though popular classification models like decision trees, neural networks and Naive Bayes classifiers are described in detail in the above section, it is worth mentioning about data mining models, which are quite popular among researchers owing to its efficiency and range of applications. Some of them are given below:

2.3.5.1 Lazy classifiers

Lazy learners store the training instances and do no real work until classification time. *IB1* is a basic instance-based learner which finds the training instance closest in Euclidean distance to the given test instance and predicts the same class as this training instance. If several instances qualify as the closest, the first one found is used. *IBk* is a *k*-nearest-neighbor classifier that uses the same distance metric. The number of nearest neighbors (default *k* is 1) can be specified explicitly or determined automatically using leave-one-

out cross-validation, subject to an upper limit given by the specified value. Predictions from more than one neighbor can be weighted according to their distance from the test instance, and two different formulas are implemented for converting the distance into a weight. The number of training instances kept by the classifier can be restricted by setting the window size option. As new training instances are added, the oldest ones are removed to maintain the number of training instances at this size. *KStar* is a nearest neighbor method with a generalized distance function based on transformations.

2.3.5.2 Association rules

They are really not that different from classification rules except that they can predict any attribute, not just the class, and this gives them the freedom to predict combinations of attributes too. Also, association rules are not intended to be used together as a set, as classification rules are. Different association rules express different regularities that underlie the dataset, and they generally predict different things.

Because so many different association rules can be derived from even a tiny dataset, interest is restricted to those that apply to a reasonably large number of instances and have a reasonably high accuracy on the instances to which they apply to. The coverage of an association rule is the number of instances for which it predicts correctly. This is often called its support. Its accuracy is often called confidence. It is the number of instances that it predicts correctly, expressed as a proportion of all instances to which it applies.

For example, for the rule:

If temperature = cool then humidity = normal, coverage is the number of days that are both cool and normal, and the accuracy is the proportion of

cool days that have normal humidity. It is usual to specify minimum coverage and accuracy values and to seek only those rules whose coverage and accuracy are within these specified limits. The concept of association rules are mentioned here owing to its high applicability in sales data bases.

2.3.5.3 Machine learning and statistics

Data mining can be considered as a confluence of statistics and machine learning. In truth, one should not look for a dividing line between machine learning and statistics because there is a continuum. Some derive from the skills taught in standard statistics courses, and others are more closely associated with the kind of machine learning that has arisen out of computer science. Historically, the two sides have had rather different traditions. While statistics is more concerned with testing hypotheses, machine learning is more concerned with formulating the process of generalization as a search through possible hypotheses. But this is a gross oversimplification. Statistics is far more than hypothesis testing, and many machine learning techniques do not involve any searching at all. In the past, many similar methods were developed in parallel in machine learning and statistics. One is decision tree induction. Four statisticians (Breiman et al. 1984) published a book on Classification and regression trees in the mid-1980s, and throughout the 1970s and early 1980s a prominent machine learning researcher, J. Ross Quinlan, was developing a system for inferring classification trees from examples. These two independent projects produced quite similar methods for generating trees from examples, and the researchers became aware of one another's work much later only. Most learning algorithms use statistical tests when constructing rules or trees and for correcting models that are "over fitted" in that they depend too strongly on the details of the particular

examples used to produce them. Statistical tests are used to validate machine learning models and to evaluate machine learning algorithms. From the literature survey it could easily be concluded that statistics and data mining are very much correlated topics and helps each other in future developments of respective fields.

2.3.6 The ethics of data mining

The use of data, particularly data about people, has serious ethical implications in data mining. Practitioners of data mining techniques must act responsibly by making themselves aware of the ethical issues that surround their particular application. When applied to people, data mining is frequently used to discriminate like who gets the loan, who gets the special offer, and so on. Certain kinds of discrimination like racial, sexual, religious, and so on are not only unethical but also illegal. However, the complexity of the situation depends on the application. Using sexual and racial information for medical diagnosis is certainly ethical, but using the same information when mining loan payment behaviour is not. Even when sensitive information is discarded, there is a risk that models will be built that rely on variables that can be shown to substitute for racial or sexual characteristics. For example, people frequently live in areas that are associated with particular ethnic identities. So using an area code in a data mining study runs the risk of building models that are based on race, even racial information is explicitly excluded from the data. It is widely accepted that before people make a decision to provide personal information they need to know how it will be used and what it will be used for, what steps will be taken to protect its confidentiality and integrity, what the consequences of supplying or withholding the information are, and any rights of redress they may have. Whenever such information is collected,

individuals should be told these things, not in legalistic small print but straightforwardly in plain language they can understand. The potential use of data mining techniques may stretch far beyond what was conceived when the data was originally collected. This creates a serious problem. It is necessary to determine the conditions under which the data was collected and for what purposes it may be used. Surprising things emerge from data mining. For example, it was reported that one of the leading consumer groups in France has found that people with red cars are more likely to default on their car loans. What is the status of such a "discovery"? What information is it based on? Under what conditions was that information collected? In what ways is it ethical to use it? Clearly, insurance companies are in the business of discriminating among people based on stereotypes like young males pay heavily for automobile insurance. But such stereotypes are not based solely on statistical correlations; they also involve common-sense knowledge about the world. Whether the preceding finding says something about the kind of person who chooses a red car, or whether it should be discarded as an irrelevancy, is a matter of human judgment based on knowledge of the world rather than on purely statistical criteria. When presented with data, it is needed to ask who is permitted to have access to it, for what purpose it was collected, and what kind of conclusions is legitimate to draw from it. The ethical dimension raises tough questions for those involved in practical data mining. It is necessary to consider the norms of the community that is used to dealing with the kind of data involved, standards that may have evolved over decades or centuries but ones that may not be known to the information specialist. In addition to community standards for the use of data, logical and scientific standards must be adhered to when drawing conclusions from it. If one comes up with

conclusions (such as red car owners being greater credit risks), one need to attach caveats to them and back them up with arguments other than purely statistical ones. The point is that data mining is just a tool in the whole process. It is people who take the results, along with other knowledge, and decide what action to be taken. Of course, anyone who uses advanced technologies should consider the wisdom of what they are doing. So the conclusion that can be drawn is like when data mining is conducted in a particular domain, one should seriously address the question whether the objective of mining is useful for the mankind and is there anything non-ethical hidden behind the rules extracted from the data mining process.

2.4 Chapter summary

This chapter gives an introduction to the various data mining strategies popularly used. Models like decision tree, neural network and Naive Bayes classifier are described in detail. Important research works carried out using these models are reviewed in this chapter. Decision trees are very simple to interpret and work faster. Neural networks manage missing values and categorical values very efficiently. Naive Bayes insist that their numeric data should be normally distributed. From the discussions, it can be concluded that for data mining one cannot find a single classifier that outperforms every other but rather one can find a classifier that performs well for a particular domain.

Chapter 3

Data pre-processing

*This chapter explains the data pre-processing that was carried out for this research work. Systematic data collection happens in Nodal **c**entre from passed out students. This data happened to be in old technologies like FoxBASE data base formats which have to be converted to the formats needed for respective data mining models which are described in the coming sections.*

3.1 Data

The data used in this work was the data supplied by the Kochi nodal centre of National Technical Manpower Information System (NTMIS). Data is compiled by the nodal centre from the feedback given by graduates, post graduates, and diploma holders in engineering from various engineering colleges and polytechnics located within the state. This survey of technical manpower information was originally done by the Board of Apprenticeship Training (BOAT) for various individual establishments. The collected data was entered into FoxBASE data base system, which is a pretty old data base technology and this practice was there in Nodal centre since its inception. This format has to be completely ported to respective formats needed for various data mining models. A prediction model was prepared from training data during the year 2000-2002 and tested with data from the year 2003. In nodal centre during 2000-2002 data records of 6096 students were collected for analysis and during 2003, 2428 records were collected for analysis. From this, processed records were prepared as explained in section 3.2.2. In Nodal centre they conduct data collection and analysis of passed out students on a 4 years back basis. That is, in year N, they conduct data processing of year N-4. Data of year 2000-2002 was used as training set. Later, for further verification, the model was tested on 2008 data also, with its previous three years data (2005-2007) as training data. Still it was showing good accuracy comparable to that of 2003 test data.

3.2 Data pre-processing

The success of any data mining process depends highly on data selected for operation. For a classification problem, attributes that are having very good discriminative power should be selected. Chi-square statistical test can select the most important attributes that can decide the target classes in a classification problem. In this work the SPSS software was used for conducting a chi-square statistical test.

3.2.1 Chi-square (χ^2) analysis for dimensionality reduction

	X	**Y**	**Totals**
A	$r_1 * c_1$ / total	$r_1 * c_2$ / total	r_1
B	$r_2 * c_1$ / total	$r_2 * c_2$ / total	r_2
Totals	c_1	c_2	*total*

Table 3.1: Typical Cross tabulation

Table 3.1 shows tabulation with two attribute values A and B as row variables and two class labels X and Y as column variables. The goal is to decide whether the column class attribute is dependent on row attribute. If the two variables were indeed independent one would expect the values in the table cells to look as in table 3.1, according to Pierson chi-square test.

The test starts with a null hypothesis that the row and column variables are independent, or unrelated. If that assumption was true one would expect that the values in the cells of the table are balanced. To determine what is meant by balanced, consider the tabulation data in example as in table 3.2, with two variables, sector and chance of this work, for example. It is to be decided on

whether there is a relation between sectors (Rural/Urban) and chance (E, G, A, P), or whether the two variables are independent of each other.

	E	G	A	P	Totals
Rural	324	194	68	*168*	*754*
Urban	416	114	69	*192*	*791*
Totals	*740*	*308*	*137*	*360*	*1545*

Table 3.2: Cross tabulation table 2

Now it is to be understood what kind of distribution in the various cells one would expect if the two variables were independent. Suppose the expected value of sector=Rural and chance=excellent has to be computed as shown in table 3.3, applying the formulae in table 3.1. It is 754*740/1545=361. Similarly the expected values of all cells are to be computed. Now the chi square value is given by the equation (3.1).

$$\chi^2 = \sum_{i=1}^{r} \sum_{j=1}^{c} (O_{ij} - E_{ij})^2 / E_{ij} \qquad \text{-(3.1)}$$

So the difference between expected value and actual value has to be squared and divided by the expected value. This has to be repeated for each cell and added up finally to get a chi-square value of 32.95.

In statistics, the number of degrees of freedom is the number of values in the final calculation of a statistic that are free to vary. In a chi-square test, the degrees of freedom is given by (r-1)*(c-1) which is (2-1)*(4-1) =3 in this case. Now in a statistical table called, chi-square table, the degrees of freedom row labelled 3 has to be located and the corresponding cell critical value which is close to the chi-square value 32.95 towards its right also has to be located. For that cell find the corresponding probability value which is nothing but its corresponding column label. Chi-square test says that if this probability value is less than 0.05, the variables are not independent or the null hypothesis may be rejected. In our work, this value was observed to be much below 0.05 and was close to 0. Hence it was concluded that the variables are dependent.

	E	**G**	**A**	**P**	**Totals**
Rural	754*740/1545 =361	754*308/1545 =150	754*137/1545 =67	*754*360/1545 =176*	*754*
Urban	791*740/1545 =379	791*308/1545 =158	791*137/1545 =70	*791*360/1545 =184*	*791*
Totals	*740*	*308*	*137*	*360*	*1545*

Table 3.3: Cross tabulation table with expected values

List of attributes extracted and found to be the deciding attributes by chi-square test: They are rank, category, sex, sector and branch. The significance of each of these attributes is given below:

RANK: Rank secured by candidates in the engineering entrance exam, which is numeric in nature. This being a continuous valued entity, it is discretized into intervals of 200 and only rank ranges like (1-200), (201-300) etc are used for data mining purposes.

CATEGORY: It tells about the social background of students. It can take any value from the set {General, SC/ST, OBC}.

SEX:{M/F}

SECTOR: It tells about the demographic background. The possible values are {Urban, Rural}

BRANCH: Range {A-J}, each engineering branch is assigned a letter. For e.g. A for CS and so on

ACTIVITY: The placement chances for a particular attributes combination. It can take one among the set (Excellent, Good, Average and poor} as its value. This is the output class to be predicted. The same dataset and attributes were used for all the models for comparing their performances and choosing optimum models.

Statistics and data mining are two closely related fields [45]. Hence lot of researches are still happening in the domain of better attribute dependency analysis techniques related to the fields of statistics and data mining. One such work is described in [68].

3.2.2 Data cleansing techniques

The cleansing stage makes data more consistent using checks and validations. Cleansing stage converts the data to standard formats by fixing data ranges, encoding etc to standard file format like excel sheets or data base formats depending on the model used. For replacing missing values for numeric attributes, arithmetic mean substitution can be used. For categorical data mode substitutions may be used, where the most repeated value is substituted. For e.g. if the data for marital status is missing in data corresponding to college students, it can be replaced with "unmarried" since that is the most usual case.

Data transformation or normalisation: Some of the common types of transformations are ***attribute transformations*** like for e.g. '0' may be used for Male and '1' may be used for female and decimal scaling In decimal scaling, for the given data X={2,8,10,13.17,20}, applying the scaling transformation to the range [0,1] , {0.1,0.4,0.5,0.65,0.85,1.0} is obtained, by dividing with the maximum value 20. But the most popular data normalization techniques used is the ***MIN-MAX normalization*** where data is converted to a new range [0, 1] by applying common formulae to all data as given in equation 3.2. It is sometimes called change of origin and scale transformation where data values are changed by translating their origin by adding/subtracting a constant as well by scaling down the range to another value. In ***smoothing transformations***, the data values are adjusted to another value like in ceil (2.5) =3, a rounding value transformation. In ***scrubbing transformations***, whenever a replacement is needed, text strings are found and replaced. For e.g. if due to a data entry error, if all names "Williams" was entered as "Bill", the entries can be corrected with a find-replace function.

Since this research work had used the data very carefully collected by the nodal centre, the scope for data cleaning was very limited. The nodal centre had used concepts like mode substitutions, table look up functions etc for data cleaning. The main data pre-processing function that was done in this work was the data normalization and attribute transformations. It was done for the neural network modeling. Also the computation of placement chances for each of the attribute permutations (for e.g. corresponding to rank (200-399), sex-female, sector= rural, background=OBC, how many have got placement in civil engineering) and preparation of the final data sheet for data mining are worth mentioning. Though the data sets used for all the models were same, they were tested using different file formats and software tools. The respective data set preparations are described in the following sections.

3.2.3 Data set for decision trees

The database provided by Nodal Centre, was in FoxBase format. The data mining problem using decision trees was planned to be implemented as a web based application. Hence it had to be converted to some DBMS used for web sites like mySQL to make the approach efficient and faster. First FoxBase data was converted to CSV files (Comma Separated files) and this file was loaded to MS Excel. Then this Excel format was converted to mySQL format using xls-mySQL converter. The individual database files (DBF format) for the years 2000-2003 were obtained and converted to mySQL format as explained above for decision tree model. All the data mining models for prediction were created using data from years 2000-2002 and tested using data of year 2003. Due to the difference in the software packages used, different file formats were required for different models. Decision trees could be implemented in WEKA too, which is explained in chapter 4.

As explained in section 3.2.1, the most decisive attributes for this research work turned out to be rank, category, sex, sector and branch. Now for training of the data mining models, a new table was created which had the above attributes as decisive attributes. For example a typical record had the following attribute values.

RANK (1-200) SECTOR (U) SEX (M), CATEGORY (GEN) BRANCH (A), ACTIVITY

The activity attribute is related to the percentage of students who actually got placement in the history data base, who belongs to the particular combination of attribute values as shown above.

The Probability (P) for the above combination of attribute values is the number of students having above attribute values who actually got placed divided by total number of students with these same attribute criteria.

The activity or placement chance for a particular attribute combination is obtained by the following rules:

IF P>=95 THEN Chance='E',

If P>=75 && P<95 THEN Chance='G'

If P>=50 && P<75 THEN Chance='A';

Else Chance='P';

Where E, G, A, P stand for Excellent, Good, Average & Poor respectively. Similarly the placement chances for different attribute value combinations was computed and stored as different records.

Branch	Sector	Sex	Rank	Reservation	Chance
CS	RURAL	M	1-200	GEN	E
CS	RURAL	F	201-400	GEN	A
EC	RURAL	M	401- 600	OBC	G
EC	URBAN	F	601- 800	OBC	P
(Continues.....)					

Table 3.4: Sample Partial Dataset used to construct Decision Tree

A data sheet similar to the one given in table 3.4 was prepared. From this data sheet the decision tree was constructed using the decision tree construction algorithm C4.5. The test data set consisted of 1063 such records each representing one particular combination of input attributes say for e.g., Branch=CS, Sex=F, Rank (201-400), BG=RURAL, RES= GEN and its corresponding placement chance as one among(E,G,A,P).

3.2.4 Data set for Naive Bayes classifier

For prediction problem using naive Bayes classifier a package called WEKA was used. WEKA is a collection of machine learning algorithms for data mining tasks. WEKA contains tools for data pre-processing, classification, regression, clustering, association rules, and visualization. It is also well-suited for developing new machine learning algorithms.

All of WEKA's techniques are predicated on the assumption that the data is available as a single flat file or relation, where each data point is described by a fixed number of attributes (normally, numeric or nominal but some other types are also supported).

WEKA's main user interface is the Explorer, but essentially the same functionality can be accessed through the Component-Based Knowledge Flow interface and from the command line. There is also the Experimenter, which allows the systematic comparison of the predictive performance of WEKA's machine learning algorithms on a collection of datasets.

The Explorer interface has several panels that give access to the main components of the workbench. The Pre-process panel has facilities for importing data from a database, a CSV file, etc., and for pre-processing this data using a so-called filtering algorithm. These filters can be used to transform the data (e.g., turning numeric attributes into discrete ones) and make it possible to delete instances and attributes according to specific criteria. WEKA input must be in arff format (Attribute related file format).

In WEKA software, a relation name is represented as
–@relation student
A list of attribute definitions
–@attribute RANK numeric
@attribute SECTOR {U, R}
@attribute SEX {F, M}
@attribute CATEGORY {GEN, OBC, SC, ST}
@attribute BRANCH {A, B, C, D, E, F, G, H, I, J}
@attribute CHANCE {E, P, A, G}
– The last attribute is the class to be predicted
A list of data elements can be given as
@data 1, U, F, GEN, D, E

The same datasets used in decision trees are used in naive Bayes problem also. The initial data processing is same as described in the decision trees section. Sample data is shown in table 3.4. Only difference is that for data porting from excel format to arff format, WEKA package was used as shown in Figure 3.1. It is a popular tool developed by University of Waikato, New Zealand.

Figure 3.1: The Knowledge Flow interface in WEKA

3.2.5 Data set for neural networks

MATLAB was used for modelling the neural network in the context of this problem. The same original dataset from year 2000-2002, which was used in other models were also used here. Though WEKA supports Neural Network modelling, MATLAB offers more scripting and modelling features for neural networks. Later at the point of experimenting with ensemble, it was implemented in WEKA too.

Once the most appropriate raw input data is selected, it must be pre processed to produce accurate forecasts. Transformation and normalization are two widely used pre-processing methods. A typical transformation operation

involves converting categorical to numeric data. Normalization involves distributing the data evenly and scaling down into an acceptable range for the model.

Attribute	Range	Mapped to
RANK	1 to 4000	0 to 1
SEX	1 to 2	0 to 1
CATEGORY	1 to 4	0 to 1
SECTOR	1 to 2	0 to 1
BRANCH	A to J	0 to 1
ACTIVITY	1 to 4	One of the four values 'E', 'G', 'A' and 'P'.

Table 3.5: Attribute values mapped to 0 to 1 scale

Data normalization: Equation 3.2 was used for transforming each data value D to I.

$$I = Imin + (Imax-Imin) * (D-Dmin) / (Dmax-Dmin) \quad \textbf{- (3.2)}$$

Linear scaling requires the minimum and maximum values associated with each input. These values are Dmin and Dmax, respectively. Imin to Imax is the input range required for the network. Dmin and Dmax are determined by the attribute values.

Since the neural network accepts input in the range either -1 to 1 or 0 to 1, all the input and output data are mapped to data between 0 and 1. Table 3.5 shows the actual range of the attribute values and the normalized range.

Table 3.6 shows a snippet of sample data used to train neural network.

The output data used for training is derived from ACTIVITY (Placement chance) attribute. Instead of representing the output on 0 to 1 scale, a 4 value code was assigned with each record. A code value of 1000 represents a 'excellent' chances of getting placement for the student, a code value of 0100, 0010, 0001 represents 'good', 'average' and 'poor' chances of placement of a student respectively. The interpretation of placement chances is same as that of decision trees and Naive Bayes classifier and is shown in table 3.7.

Sex	Reservation	Location	Rank	Branch
0	0	1	0.72	0.47
1	0	1	0.72	0.47
0	1	0	0.59	0.33
0	0	1	0.4	0.66
0	1	0	0.72	0.47
1	1	1	0.27	0.38

Table 3.6: Snippet of the sample data used to train the neural network.

Range of probability	Output code	Chances of placement
P >= 0.95	1000	Excellent
0.75 <= P < 0.95	0100	Good
0.50 <= P < 0.75	0010	Average
P < 0.50	0001	Poor

Table 3.7: Assigning output code to records

3.3 Chapter summary

This chapter described the various data pre-processing steps that were adopted for this research work. In Nodal centre the data collection process is very systematic since its inception. In Nodal centre they conduct data collection and analysis of passed out students on a 4 year back basis. That is, in year N, they do processing of year N-4. Data of the year 2000-2002 was used as training set and data of the year 2003 was used as test set for this work. Also to further verify the results, data of the year 2008 was used as test set with its previous three years data (2005-2007) as training set. The results were good and comparable with the case of the model with 2003 test data. Even though the same training and test data were used for all the models, their individual data format requirements varied and in this chapter the data formats for the individual models and corresponding transformations were discussed.

Chapter 4

Experimenting with data mining models

This chapter explains the experimental set up and the procedure used for each of the data mining models. In each of the sub sections, the tools, input file formats and the experimental procedure used for respective models are described.

4.1 Experimental back ground

As explained in chapter 1, the main goal of this work is to propose an efficient data mining model for employment chance prediction. For this, the effectiveness of each of the three models for this problem was to be tested. As already known, each of the models have their own data formats and experimental setup. As nodal centre had a plan to have a web based application which can help the students to choose a good career, the first thought was to start experimenting with decision tree model as a web application. Hence Web based tools like PHP, mySQL etc were used to implement the same. Decision trees can also be implemented and optimised using WEKA software package. So it was implemented in WEKA as the parameters was to be optimized using a parameter tuning strategy to improve accuracy. Another popular model to be implemented was neural networks, for which MATLAB was used. MATLAB was chosen considering its very good scripting and modelling features for neural networks. Later at the stage of ensemble experiments, the neural network was implemented in WEKA too. Naive Bayes classifier was modelled and tested in WEKA itself. It was ensured that the same training data sets and test data sets were used for all the three models. As explained in chapter 3, data from the year 2000 to 2002 was used as training set and the year 2003 data was used as test set. The results were again verified with test data of 2008 and its previous 3 year data as training data.

In the following sections, the modelling and experimental background for each of the three models is described. In the next chapter the detailed performance comparison of these models with respect to various evaluation criteria are described.

4.2 Decision tree

As described in chapter 3, the initial database provided by Nodal Centre, was in FoxBASE format. Later it was converted to mySQL to make the approach efficient and faster. First FoxBase data was converted to CSV files (Comma Separated files) and this file was loaded to MS Excel. Then from this Excel format using xls-mySQL converter, it was converted to mySQL format. Important attributes were found using chi square attribute analysis and they were Rank, Sex, Reservation (General, OBC, SC/ST), category (Rural/Urban) and Branch. The most popular decision tree algorithm using information gain was completely implemented through PHP programming. The tree induction algorithm used was C 4.5 based on information gain concept, and is described in detail in section 2.3.1.

A new table was created in which the placement chance for each possible input combination was stored. For e.g., for RANK (1-200) SECTOR (U) SEX (M) CATEGORY (GEN), the percentage of students who had actually got placement in history database was computed. If this percentage is greater than 95%, this combination was classified as “Excellent” chance. This decision tree is physically stored as an adjacency list. As an example, the adjacency list corresponding to the partial view of the decision tree given in figure 4.1 is showed in table 4.1.

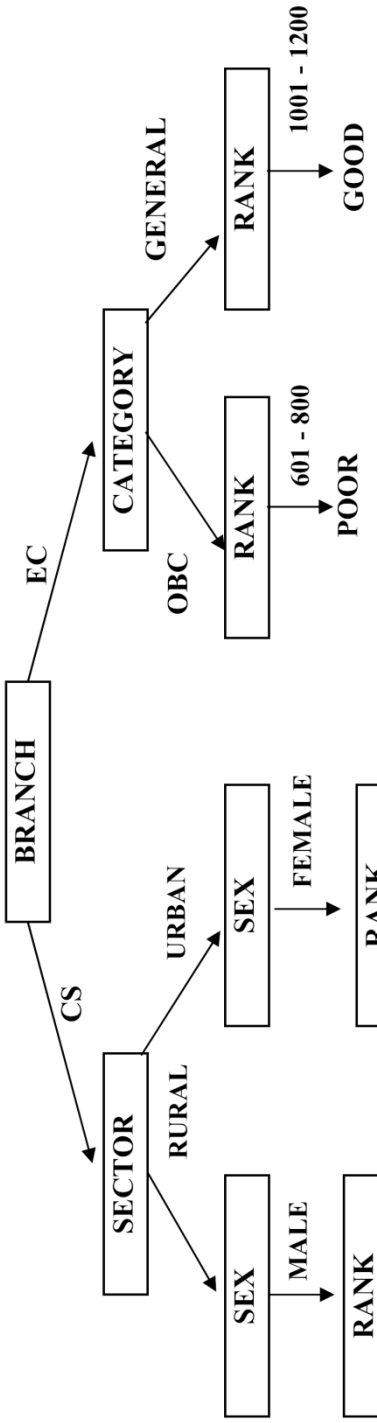


Figure 4.1: A decision tree-Partial view

Id	Type	Node	Parent
1	BRANCH	CS	0
2	SECTOR	RURAL	1
3	SEX	MALE	2
4	RANK	1	3
5	ACTIVITY	EXCELLENT	4
6	SECTOR	URBAN	1
7	SEX	FEMALE	6
8	RANK	2	7
9	ACTIVITY	AVERAGE	8
10	BRANCH	EC	0
11	CATEGORY	OBC	10
12	RANK	4	11
13	ACTIVITY	POOR	12
14	CATEGORY	GENERAL	10
15	RANK	6	14
16	ACTIVITY	GOOD	15

Table 4.1: Adjacency list a partial view

The adjacency table is actually implemented as a database table in mySQL. The concept is very similar to the data structure adjacency list. In this every node in the tree is a record in the table. In order to store a tree like structure with links between records, a special link called parent was used to designate which is the parent of a particular node in the tree. For this data, if SEX is found to be the attribute with max gain when the decision tree algorithm is running, it is added to the Adjacency list and the database is split

into unique set of records with common values for sex. This process is recursively repeated for all cases. For all iterations, the best attribute field would be appended to the adjacency list. One key point here is that the attribute field names, the results etc are all treated as nodes and the list can identify the node only by its corresponding TYPE. The ID field stores the unique id number of the node, while the parent stores the id of the parent of the node.

As an example the partial view of the adjacency list in table 4.1 may be interpreted as if branch=CS, sector=Rural, sex=male and rank is between 1 and 199(which is using the code "1"), there is excellent placement chance.

When this work is implemented as a web application, the user enters his search criteria through the user interface screen and then a query string is constructed, which are parsed to get individual attributes which are used to search the decision tree.

For example consider the following query,

What is the placement chance for BRANCH (CS) SECTOR(R) SEX (M) and rank is between 1 and 200?

The query proceeds from ID 1 as in the adjacency list in table 4.1 and the algorithm searches for all the nodes which have ID 1 as parent. The algorithm will find ID =2 and ID=6 as children nodes. It then finds that ID=2 is the right path which needs to be taken to arrive at the result. This process continues and the Chance value is found at ID=5 as E (Excellent).

Thus the decision tree was constructed and it was tested with data from year 2003. The performance analysis of this model is described in chapter 5. As discussed earlier, decision tree algorithms are also available with WEKA

data mining package. Now in order to optimize its performance using a parameter tuning strategy, it was implemented in WEKA too.

4.2.1 A Parameter optimization strategy

"How to improve the performance of classification algorithms?" is an important research issue. To accomplish this, a researcher can proceed along several paths. In this work, an approach called "Parameter tuning" is used to improve the performance of decision trees. The result obtained indicates that parameter tuning offers evident variability in classifier performance. Here the parameter tuning of the J48 algorithm (Implementation of the classical C4.5 decision tree algorithm) in the popular data mining package WEKA is considered.

The C4.5 algorithm (J48 in WEKA) induces pruned or unpruned decision tree classifiers. Pruning is all about optimizing a tree structure by removing some branches which may not contribute much to classifier accuracy. There are two types of pruning, namely pre-pruning and post pruning. In pre-pruning before a new branch is added to the tree, evaluations are made to see whether this branch will improve accuracy. In post pruning, the tree is first created and then branches are removed stage by stage, if they do not improve classifier accuracy beyond a predefined threshold.

J48 parameter tuning possibilities include the type of pruning, the confidence threshold for pruning, the number of folds used for reduced error pruning and the minimum number of instances per leaf. Sub tree raising is a post-pruning technique that raises the sub tree of the most popular branch. Reduced-error pruning is performed by splitting the data set into a training and validation set. The smallest version of the most accurate sub tree is then constructed by greedily removing the node that less improves the validation

set accuracy. The J48 default configuration can be summarized as follows. The Pruning is set as “true” and the technique used is sub tree raising. The confidence threshold for pruning is 0.25, the minimum instances per leaf parameter are set to 2 and the number of folds for reduced error pruning value is set to 3.

Confidence threshold	Minimum instances per leaf	Folds for pruning	Pruning	Reduced error pruning	Sub-tree raising	accuracy
0.3	4	5	True	False	False	77
0.4	4	5	True	False	True	77.4
0.3	3	4	True	False	True	77.4
0.4	4	3	True	False	True	77.4
0.5	5	3	True	False	True	78.57
0.25	2	3	True	False	True	79.39 *Default
0.5	5	3	True	True	False	79.39
0.4	7	5	True	False	True	80.3

Table 4.2: Improvement in the accuracy of C 4.5 algorithm through parameter tuning

These parameters are varied step by step and by combinations. The accuracies are recorded for every combination of the parameters and a sub set of these combinations is shown in table 4.2. Figure 4.2 shows the WEKA screenshot for the parameter optimization of C 4.5 algorithm. It can be observed that best accuracy of 80.3 is obtained for the combination (Confidence Threshold=0.4, Minimum instances per leaf=7, Folds for pruning=5, Pruning=true, Reduced error pruning=false and sub tree raising =true). The performance analysis corresponding to this decision tree is described in chapter 5.

4.3 Neural networks

Neural networks are widely used for pattern recognition and they try to mimic biological neural networks in their functioning. As explained in chapter 3, it is this model that used data normalisation techniques extensively. This is required because neural networks expect its categorical inputs in the range {0, 1}. A supervised learning algorithm was used to build this model. MATLAB was used to model the neural network.

One of the most popular neural network models is Perceptron, but this model is limited to the classification of linearly separable vectors. Since the data used for this work was not linearly separable, a multi layer perceptron network was considered. The training algorithm used was back propagation algorithm. An appropriate model of BP neural network was selected and repeatedly trained with the input data until the error reduced to a fairly low value. At the end of training, a set of thresholds and weights which determined the architecture of the neural network were obtained.

A back propagation neural network model consisting of three layers was developed. The number of input neurons was 5, which depended upon the number of the input attributes. The number of neurons used in hidden layer was 5; this number was fixed based on observations. The transfer function used in the Hidden layer was Log- Sigmoid while that in the output layer was Pure Linear.

Training was done by the Conjugate Gradient algorithm, Powell-Beale Restarts, one of whose application is given in [70]. This algorithm provided faster convergence in comparison to conventional basic Back propagation algorithm by performing a search along the conjugate direction to determine the step size which minimizes the performance function along that line.

Conjugate Gradient Algorithms: The basic back propagation algorithm adjusts the weights in the steepest descent direction (negative of the gradient). This is the direction in which the performance function is decreasing most rapidly. It turns out that, although the function decreases most rapidly along the negative of the gradient, this does not necessarily produce the fastest convergence. In the conjugate gradient algorithms a search is performed along conjugate directions, which produces generally faster convergence than steepest descent directions. In most of the training algorithms, a learning rate is used to determine the length of the weight update (step size).

In most of the conjugate gradient algorithms, the step size is adjusted every iteration. A search is made along the conjugate gradient direction to determine the step size which will minimize the performance function along that line. Any of the search functions can be used interchangeably with a variety of the training functions. Some search functions are best suited to certain training functions, although the optimum choice can vary according to the specific application. An appropriate default search function is assigned to each training function, which can be modified by the user.

Powell-Beale Restarts: For all conjugate gradient algorithms, the search direction will be periodically reset to the negative of the gradient. The standard reset point occurs when the number of iterations is equal to the number of network parameters (weights and biases), but there are other reset methods which can improve the efficiency of training. One such reset method was proposed by Powell in 1977, based on an earlier version proposed by Beale. For this technique, search will be restarted if there is very little orthogonality left between the current gradient and the previous gradient.

This is tested with the following inequality:

$$\left|\mathbf{g}_{k-1}^{T}\mathbf{g}_{k}\right| \geq 0.2\left\|\mathbf{g}_{k}\right\|^{2} \quad \text{- (4.1)}$$

If this condition is satisfied, the search direction is reset to the negative of the gradient.

Thus a MLP network with Powell-Beale-Restarts was effectively implemented and its detailed evaluation is discussed in the next chapter.

4.4 Naive Bayes classifier

This classifier was implemented using WEKA 3.4 package. Naïve Bayes algorithms are available in the classify pane of this package and using default settings and supplying a separate training and test set option, the experiment can be conducted. The same training and test data sets were used as the case of other models. Weka contains tools for data pre-processing, modeling, testing and visualization. It is also well-suited for developing new machine learning schemes. The Initial database provided by Nodal Center was loaded to MS Excel and converted to CSV files (Comma Separated files) .This file was loaded in Weka Knowledge Flow interface and converted into ARFF files (Attribute-Relation File Format), which is the WEKA standard file format.

An experimental setup for using WEKA to model classifiers is shown in figure 4.3

Figure 4.2: Optimizing parameters of the C 4.5 decision tree algorithm in WEKA

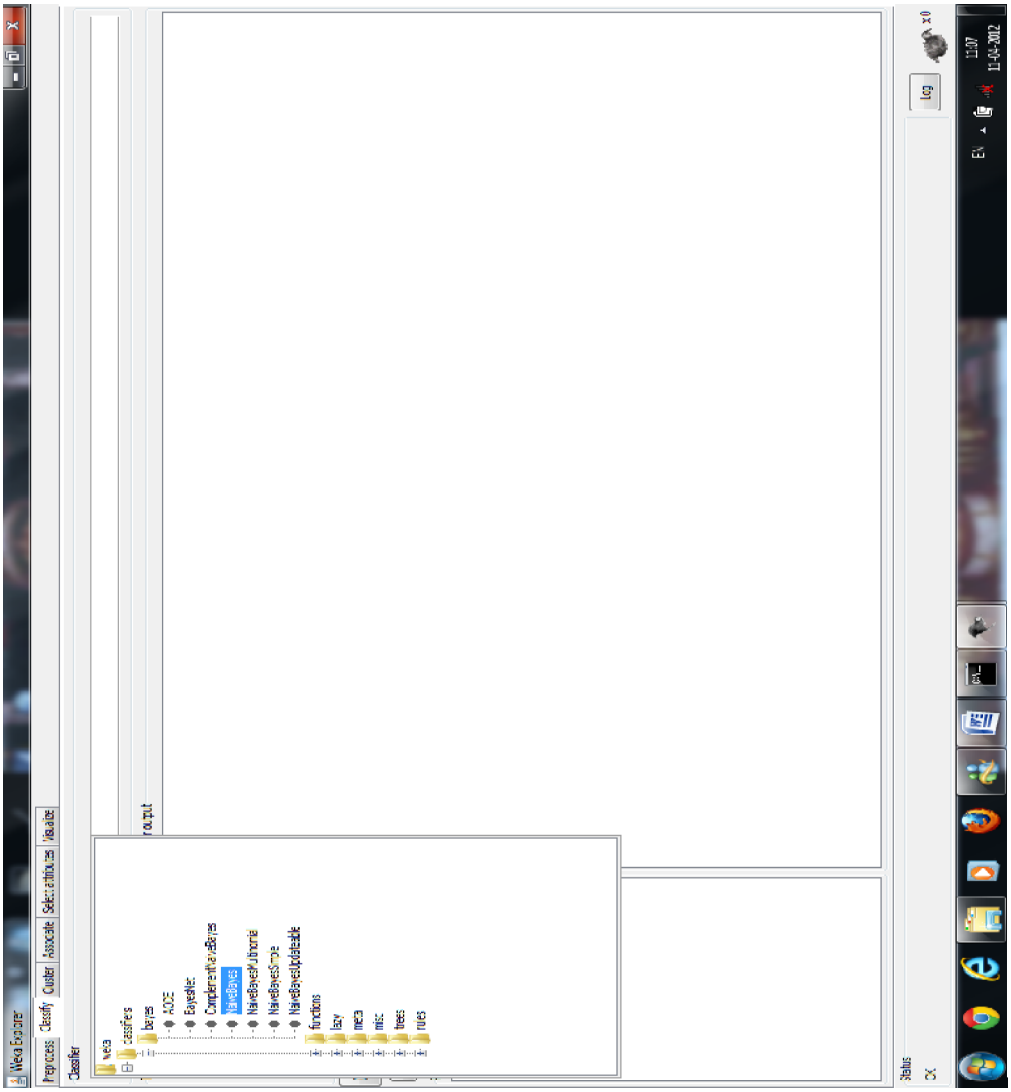


Figure 4.3: WEKA screen shot for choosing classifiers for modelling

4.5 Chapter summary

In this chapter, the modelling and testing setups for each of the data mining models were described. Naive Bayes classifier was implemented using WEKA 3.4 package where as neural network was implemented using MATLAB. MATLAB offers good scripting and modelling features for neural networks. Decision tree model was developed as a web application, with the aim that later it may be used as a web site for the public. It was also implemented using WEKA package. Using a parameter tuning strategy, the various parameters of the C 4.5 decision tree algorithm were optimized to improve the accuracy. At the time of ensemble experimentation, all the three models were implemented using WEKA too. All the models were trained using the data for the years 2000 to 2002 and tested with the data for the year 2003. These models were again verified using year 2008 test data and its previous three years as training data. In the next chapter the testing and comparison of performances of these models are described in detail.

Chapter 5

Performance evaluation of the data mining models

This chapter explains the theory and practice of various model evaluation mechanisms in data mining. Performance analysis is mainly based on confusion matrix. Also various statistical measures such as accuracy, ROC area etc used to analyse performances are described here.

-

5.1 Performance comparison strategies

In any branch of science, it is almost a common requirement that performance of various models have to be compared with each other to understand the suitability of a model to a given problem. In data mining also it is a common requirement and in this work "***Confusion matrix***" was used for this purpose. From the confusion matrix, various statistical measures are analysed and inferences are drawn. In this chapter, further descriptions are divided into two main parts. The first part explains the theoretical background behind confusion matrix and its analysis. The second part explains its application to this problem. In the data collected from the nodal centre there were records from 2000-2002, which were used for training and records in 2003, used for testing. They were converted to 1063 input combination records as described in chapter 3. To use 10 fold cross validation effectively, the input records should be usually less than 1000. As numbers of test records are greater than 1000 in this case, holdout method using separate training and test set is used.

5.2 Theoretical background for performance analysis

A ***confusion matrix*** is a simple performance analysis tool typically used in supervised learning. It is used to represent the test result of a prediction model. Each column of the matrix represents the instances in a predicted class, while each row represents the instances in an actual class. One benefit of a confusion matrix is that it is easy to see if the system is confusing two classes (i.e. commonly mislabelling one as another).

In the table 5.1, a confusion matrix is shown, for which, the various values and related equations are described. Few of these equations are very relevant for performance analysis.

Confusion matrix		**Predicted**	
		Negative	**Positive**
Actual	**Negative**	a	b
	Positive	c	d

Table 5.1: A Typical Confusion matrix

The entries in the confusion matrix have the following meaning in the context of a data mining problem:

- *a* is the number of **correct** predictions that an instance is **negative**,
- *b* is the number of **incorrect** predictions that an instance is **positive**,
- *c* is the number of **incorrect** of predictions that an instance **negative**,
- *d* is the number of **correct** predictions that an instance is **positive**.

Several standard terms are defined for the 2 class matrix:

The ***accuracy*** (*AC*) is the proportion of the total number of predictions that were correct. It is determined using the equation:

$$AC = \frac{a+d}{a+b+c+d} \quad \textbf{- (5.1)}$$

The ***recall*** or ***true positive*** (*TP*) ***rate*** is the proportion of positive cases that were correctly identified, as calculated using the equation:

$$TP = \frac{d}{c+d} \quad \textbf{- (5.2)}$$

The ***false positive*** (*FP*) ***rate*** is the proportion of negatives cases that were incorrectly classified as positive, as calculated using the equation:

$$FP = \frac{b}{a+b} \quad \textbf{- (5.3)}$$

The ***true negative*** (*TN*) ***rate*** is defined as the proportion of negatives cases that were classified correctly, as calculated using the equation:

$$TN = \frac{a}{a+b} \quad \textbf{- (5.4)}$$

The ***false negative*** (*FN*) ***rate*** is the proportion of positives cases that were incorrectly classified as negative, as calculated using the equation:

$$FN = \frac{c}{c+d} \quad \textbf{- (5.5)}$$

Finally, ***precision*** (*P*) is the proportion of the predicted positive cases that were correct, as calculated using the equation:

$$P = \frac{d}{b+d} \quad \textbf{- (5.6)}$$

The above concepts for a two-class problem can be extended to a multi class problem by focusing one of the classes as positive at a time and the rest as negative. The average of these parameters like precision, recall etc for individual classes becomes the final values of the entire model.

5.2.1 ROC (Receiver operator characteristic test)

ROC is a plot of the true positive rate against the false positive rate for the different possible cut points of a diagnostic test.

An ROC curve demonstrates several things:

1. It shows the tradeoffs between sensitivity and specificity (any increase in sensitivity will be accompanied by a decrease in specificity).
2. The closer the curve follows the left-hand border and then the top border of the ROC space, the more accurate the test.
3. The closer the curve comes to the 45-degree diagonal of the ROC space, the less accurate the test.
4. The area under the curve is a measure of text accuracy.

The graph below shows three ROC curves representing excellent, good, and worthless tests plotted on the same graph. The accuracy of the test depends on how well the test separates the group being tested into positive and negative cases. Accuracy is measured by the area under the ROC curve. An area of 1 represents a perfect test; an area of .5 represents a worthless test. A rough guide for classifying the accuracy of a diagnostic test is the traditional academic point system: A sample plot is shown in figure 5.1.

- .90-1 = excellent (A)
- .80-.90 = good (B)
- .70-.80 = fair (C)
- .60-.70 = poor (D)
- .50-.60 = fail (F)

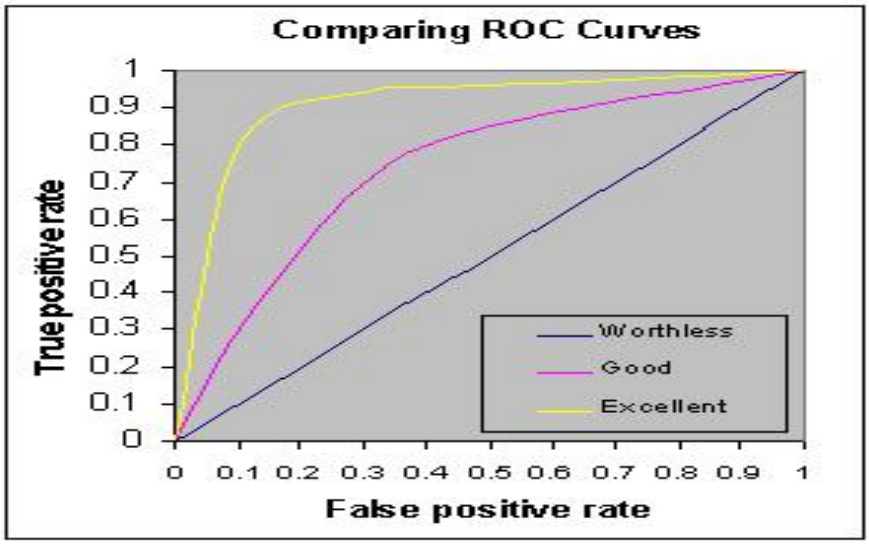


Figure 5.1: ROC Curve characteristics

From the data mining literature and text books like [62], it is observed that every data mining model is having their own pros and cons and suitable applications which can be summarised as follows. This is a general comparison.

A detailed comparison for this particular problem is given in coming sections. A comparison of the three classifiers with the most important attributes like speed, accuracy etc are shown in the table 5.2. It can be seen that each of these classifiers is good in its own sense. In coming sections of the thesis, the practical application of these models to a social science problem like employment prediction is given.

From this table of classifier performance comparison, some interesting characteristics can be observed. There are basically two types of classifiers, namely algorithm based and probability based. Both have got their own pros and cons. In this thesis, three classifiers are given more concentration, as these are the three that perform best on categorical type of input. The table 5.2

shows a summary comparison of algorithms based on experiences from various data mining applications in the literature [62]. It is assumed that their performance ranges are to be categorized as good, very good, best etc.

The efficiency of a data mining model depends on many factors as shown in table 5.2. Its ability to handle categorical data (for example sex, reservation type etc) as well as data with missing values etc are crucial in the selection of the model.

	Decision Tree	**Neural Networks**	**Naive Bayes**
Supervised	✓	✓	✓
Basic Algorithm	C 4.5/CART	Back propagation	Probability based (Bayes theorem)
Accuracy	Very good	Very good	Good
Task	Classification and regression	Classification and regression	Classification and prediction
Speed	Best	Good	Good
Handling missing values	Good	Best	Good
Handling categorical values	Good	Best	Good
Distribution of data	Not must	Not must	For numeric data normal distribution is must

Table 5.2 Comparison of classifier performances against various data mining concepts

5.3 Performance analysis techniques applied to this problem

Testing was conducted separately for each of the data mining models and performance parameters were computed separately. The same training and test sets were used for all the models. The data of years 2000-2002 records were used for building the models and year 2003 records were used for testing.

5.3.1 Testing for the neural network based prediction

The knowledge discovered is expressed in the form of confusion matrix as shown below in table 5.3.

Confusion matrix					
		Predicted			
		E	**G**	**A**	**P**
Actual	**E**	401	4	26	72
	G	6	6	1	1
	A	9	2	410	5
	P	84	1	4	31

Table 5.3: Confusion Matrix (Neural network)

Class	Precision	Recall
E	0.80	0.79
G	0.46	0.43
A	0.93	0.96
P	0.28	0.26

TABLE 5.4: class wise precision/recall values for neural networks model

Table 5.4 shows the precision and recall values for each of the classes observed for the neural network model. The average of these respective values is used for final performance comparison.

The accuracy is given by (5.1),

$$AC = 848/1063 = \mathbf{0.797}$$

The experimental back ground for each of these models is described in detail in chapter 4. In order to further verify the effectiveness of this modelling approach, this model was tested using data of year 2008. It was observed that the model is giving comparable results when it is trained with its previous consecutive three years data as training data (2005-2007). But when the test data is from a year which is far away from the year of training data, the model may not give optimum results, considering the natural fluctuations of employment opportunities for a branch. Hence it is recommended that for best performance of this modelling approach for prediction of any current year n, at most its previous (n-3) consecutive year's data may be used as training data.

5.3.2 Testing based on the decision tree based prediction

Confusion matrix					
		Predicted			
		E	**G**	**A**	**P**
Actual	**E**	416	7	6	74
	G	7	4	1	2
	A	11	4	404	7
	P	86	3	1	30

Table 5.5: Confusion Matrix (Decision tree)

The confusion matrix obtained for the decision tree work is shown in table 5.5. Table 5.6 shows the class wise precision/recall values observed for decision tree model.

Class	Precision	Recall
E	0.80	0.83
G	0.22	0.29
A	0.98	0.95
P	0.27	0.25

Table 5.6: class wise precision/recall values for decision tree model

The accuracy is given by (5.1),

$$AC = 854/1063 = \mathbf{0.803}$$

5.3.3 Testing based on the Naive Bayes based prediction

Confusion matrix					
		Predicted			
		E	G	A	P
Actual	E	496	0	2	5
	G	5	3	4	2
	A	80	30	248	68
	P	20	1	2	97

Table 5.7: Confusion Matrix (Naive Bayes)

Class	Precision	Recall
E	0.82	0.99
G	0.1	0.21
A	0.97	0.58
P	0.56	0.8

Table 5.8: class wise precision/recall values for Naive Bayes classifier model

The confusion matrix obtained is shown in table 5.7. Table 5.8 shows the class wise precision/recall values for Naive Bayes classifier model. The accuracy is given by (5.1),

$$AC = 844/1063 = \mathbf{0.794}$$

5.4 Comparison

Model	Accuracy	Precision	Recall	ROC Area
Neural Network	79.70%	0.62	0.61	0.76
Decision tree	80.30%	0.57	0.58	0.77
Naïve Bayes	79.40%	0.61	0.65	0.79

Table 5.9: Summary of model comparisons

Table 5.9 shows the complete summary of the performance comparisons of the three data mining models used for this research work. As pointed out earlier no proven results are there in data mining domain to say that a particular model is superior to any other model for all applications. As shown in table 5.2, neural networks may be very useful in situations where data is having lot of missing values and are categorical in nature, where as in applications requiring higher speeds of classifications, decision trees may be useful. Still accuracies may be comparable for both the models. Hence one can only say one model is better for a particular application compared to others, rather than generalising for all types of applications [63].

Similarly, in this work too, it can be verified that the three models gave similar results in terms of accuracy and their performances are comparable. There are other performance measures like precision, true positive, ROC etc which are described in section 5.2 of this chapter.

From the summary table it can be verified that the accuracy of the decision tree is slightly higher than that of the other models where as Naive Bayes classifier is better in terms of all other parameters like recall and ROC characteristics. It is well known that Naive Bayes classifier is a concept that is simpler to visualize and implement where as neural network modelling may become a bit complex as number of records, attributes and target classes becomes more complex. Their training time is proportional to the size of the target data set and attributes. But nowadays with the advent of latest data mining software packages like WEKA, SPSS, MATLAB etc testing and implementation of data mining models have become more clear and straight forward and many other visualisations and analysis are possible with these packages. As it can be verified that all the popular models show comparable

performances in the domain of employment chance prediction, the next possible path of research in this work can be how the classifier performance can be improved further. This is explained in the chapter 6.

5.5 Chapter summary

In this chapter, the performances of three popular data mining models namely decision tree, neural networks and Naive Bayes classifier were analysed based on various statistical measures. Confusion matrix was used for classifier performance analysis and the confusion matrix for each of the models were shown and discussed. Performance measurements namely accuracy, recall, precision and ROC area were used for performance comparison. It was found that all the three models were comparable according to performance parameters. To further verify the effectiveness of these models, they were tested using data of year 2008 using training data of its previous three years (2005-2007). It was observed that the models showed comparably good accuracies as with 2003 test data sets. Also it is recommended that the models will give optimal results of prediction for any year, provided it is trained with its previous three year's data. As usual at this stage of research work the next natural direction of the research work was to understand how the classifier performances can be improved further, which is described in chapter 6 that describes the ensemble approach.

Chapter 6

The stacking ensemble approach

This chapter proposes the stacking ensemble approach for combining different data mining classifiers to get better performance. Other combination techniques like voting, bagging etc are also described and a comparative description showing how stacking is having advantages over others is also shown.

6.1 An introduction to combined classifier approach

In previous chapters, different classifiers that are very popular owing to their simplicity and accuracy were discussed. Since the last few years the researches in data mining are to another direction called meta learners. Here the scientific community tried to address the question" Whether a combined classifier model gives a better performance than selecting the single base level model with best accuracy?"

There are many answers to the above questions. Scientists tried to think about various possibilities in which classifiers can be combined and their performances are compared with the best among the base level classifiers from which they are made. In this research work, a study was conducted to find how ensemble of classifiers improves model performance. In the following section some of the most common model combining approaches that exist in the data mining, are analysed.

6.2 Popular ways of combining classifiers

In our day to day life, when crucial decisions are made in a meeting, a voting among the members present in the meeting is conducted when the opinions of the members conflict with each other. This principle of "***voting***" can be applied to data mining also. In voting scheme, when classifiers are combined, the class assigned to a test instance will be the one suggested by most of the base level classifiers involved in the ensemble. Bagging and boosting are the variants of the voting schemes.

Bagging is a voting scheme in which n models, usually of same type, are constructed. For an unknown instance, each model's predictions are

recorded [7, 9]. That class is assigned which is having the maximum vote among the predictions from models.

Boosting is very similar to bagging in which only the model construction phase differs. Here the instances which are often misclassified are allowed to participate in training more number of times. There will be n classifiers which themselves will have individual weights for their accuracies. Finally, that class is assigned which is having maximum weight [7, 47]. An example is Adaboost algorithm. Bagging is better than boosting as boosting suffers from over fitting. Over fitting is that phenomenon where the model performs well only for the training data. This is because it knows the training data better and it does not know much about unknown data.

There are 2 approaches for combining models. One of them uses ***voting*** in which the class predicted by majority of the models is selected, whereas in ***stacking*** the predictions by each different model is given as input for a meta level classifier whose output is the final class. Whether it is voting or stacking, there are two ways of making an ensemble. They are ***Homogenous ensemble*** where all classifiers are of same type and ***heterogeneous ensemble*** where the classifiers are different.

The basic difference between stacking and voting is that in voting no learning takes place at the meta level, as the final classification is decided by the majority of votes casted by the base level classifiers whereas in stacking learning takes place at the meta level.

6.3 Stacking framework-a detailed view

Stacking is the combining process of multiple classifiers generated by different learning algorithms $L_1 \ldots L_n$ on a single dataset. In the first phase a set of base level classifiers C_1, $C_2 \ldots C_n$ is generated. In the second phase a meta level classifier is developed by combining the base level classifier.

The WEKA data mining package can be used for implementing and testing the stacking approach. Meta learning is a separate area in the data mining domain and is usually a part of ensemble methods which are one of the hottest research fields. The following section analyses the impact of meta learning in data mining.

6.4 Impact of ensemble data mining models

A methodology on how models can be combined for customer behavior is described in [21]. Companies are eager to learn about their customer behavior using data mining technologies. But the diverse requirements of such companies make it difficult to select the most effective algorithm for the given problem. Recently, a movement towards combining multiple classifiers has emerged to improve classification results. In [21], a method for the prediction of the customer's purchase behavior by combining multiple classifiers based on genetic algorithm is proposed.

One approach in combining models is called the Meta decision trees, which deals with combining a single type of classifier called decision trees. [55] Introduces Meta decision trees (MDTs) as a novel method for combining multiple models. Instead of giving a prediction, MDT leaves specify which model should be used to obtain a prediction.

In this work, the focus is on how classifier performance can be improved using the stacking approach. While conventional data mining

research focuses on how the performance of a single model can be improved, this work focuses on how heterogeneous classifiers can be combined to improve classifier performance. It was also observed that this approach yields better accuracy in the domain of employment prediction problems. In [71], it is being observed that when one prepares ensemble, the number of base level classifiers is not much influencing, and usually researchers select 3 or 7 at random depending on the type of applications. In this research, three classifiers namely decision tree, neural network, and Naive Bayes classifier were selected for making an ensemble. They have been tested individually as explained in chapter 5 and their ensemble is explained in this chapter.

There are many strategies for combing classifiers like voting, bagging and boosting each of which may not involve much learning in the Meta or combing phase. Stacking is a parallel combination of classifiers in which all the classifiers are executed parallel and learning takes place at the Meta level. To decide which model or algorithm performs best at the Meta level for a given problem, is also an active research area, which is addressed in this thesis. It is always a debate that whether an ensemble of homogenous or heterogeneous classifiers yields good performance. [36] Proposes that depending on a particular application an optimal combination of heterogeneous classifiers seems to perform better than the homogenous classifiers.

When only the best classifier among the base level classifiers is selected, the valuable information provided by other classifiers is being ignored. In classifier ensembles which are also known as combiners or committees, the base level classifier performances are combined in some way such as voting or stacking.

Research has shown that combining a set of simple classifiers may result in better classification in comparison to any single sophisticated classifier [18, 39, and 59]. In [17], Dietterich gave three fundamental reasons for why ensemble methods are able to outperform any single classifier within the ensemble — in terms of statistical, computational and representational issues. Besides, plenty of experimental comparisons have been performed to show significant effectiveness of ensemble. Assume there are several different, but equally good, training data sets. A classifier algorithm is biased for a particular input *x* if, when trained on each of these data sets, it is systematically incorrect when predicting the correct output for *x*. An algorithm has high variance for a particular input *x* if it predicts different output values when trained on different training sets. The prediction error of a learned classifier is related to the sum of the bias and the variance of the learning algorithm. Usually there is a trade-off between bias and variance. A learning algorithm with low bias must be "flexible" so that it can fit the data well. But if the learning algorithm is too flexible, it will fit each training data set differently, and hence have high variance. Mathematically, classifier ensembles provide an extra degree of freedom in the classical bias/variance trade off, allowing solutions that would be difficult (if not impossible) to reach with only a single classifier.

6.5 Mathematical insight into stacking ensemble

If an ensemble has M base models having an error rate e < 1/2 and if the base models' errors are independent, then the probability that the ensemble makes an error is the probability that more than M/2 base models misclassify the example. The simple idea behind stacking is that if an input–output pair

(x, y) is left out of the training set of h_i, after training is completed for h_i, the output y can still be used to assess the model's error. In fact, since (x, y) was not in the training set of h_i, $h_i(x)$ may differ from the desired output y. A new classifier then can be trained to estimate this discrepancy, given by $y - h_i(x)$. In essence, a second classifier is trained to learn the error the first classifier has made. Adding the estimated errors to the outputs of the first classifier can provide an improved final classification decision [38].

6.6 Stacking ensemble framework applied in this work

In this work, keeping the three base level classifiers as same, various meta level classifiers were tested and it was observed that multi response modal trees(M5') meta level classifier performed best among others. Although numerous data mining algorithms have been developed, a major concern in constructing ensembles is how to select appropriate data mining algorithms as ensemble components.

A challenge is that there is not a single algorithm that can outperform any other algorithms in all data mining tasks, i.e. there is no global optimum solution in selecting data mining algorithms although much effort is devoted to this area. An ROC (Receiver Operating Characteristics) analysis based approach was approved to evaluate the performance of different classifiers. For every classifier, its TP (True Positive) and FP (False Positive) are calculated and mapped to a two dimensional space with FP on the x-axis and TP on the y-axis. The most efficient classifiers should lie on the convex hull of this ROC plot since they represent the most efficient TP and FP trade off. The three base level classifiers decision tree (ROC=0.77, ACC=0.803), neural networks (ROC=0.76, ACC=0.797) and naive Bayes (ROC=0.79,

ACC=0.794) are the best choices among base level classifiers for this domain. It was clear that by combing classifiers with stacking using an algorithm namely multi response model trees (M5'), an accuracy of (82.2) was observed which is better than selecting best among base level classifier accuracy in this work.

The table 6.1 shows the summary.

	Base level classifiers			**Meta level classifiers**		
Classifier	**Decision tree**	**Neural Network**	**Naïve Bayes**	**Bagging**	**Regression**	**M5'**
Accuracy	80.3	79.7	79.4	79.2	78	**82.2**
Precision	0.57	0.62	0.61	0.75	0.78	**0.87**
Recall	0.58	0.61	0.65	0.9	0.85	**0.76**
ROC	0.77	0.76	0.79	0.88	0.88	**0.89**

Table 6.1: Summary performances with base & Meta level classifier

Multi response model trees: When decision trees are constructed, if linear regression principles are also adopted, for each of the m target classes, m regression equations are formed. This concept is adopted in an algorithm namely M5 by Quinlan [71]. Given a new example x to classify, LR_j (x) is calculated for all j, and the class k is predicted with maximum LR_k (x). In multi response modal trees, instead of m linear equations, m model trees are induced. Model trees combine a conventional decision tree with the possibility of linear regression functions at the leaves and also it is capable of dealing with continuous attributes. This representation is relatively perspicuous

because the decision structure is clear and the regression functions do not normally involve many variables. M5' algorithm uses this concept and gives a better performance when used at the Meta level of the stacking ensemble for this domain. This is illustrated in figure 6.1. The M5' algorithm is implemented in Weka package as M5P algorithm. The experimental set up in Weka for ensemble learning is shown in figure 6.2.

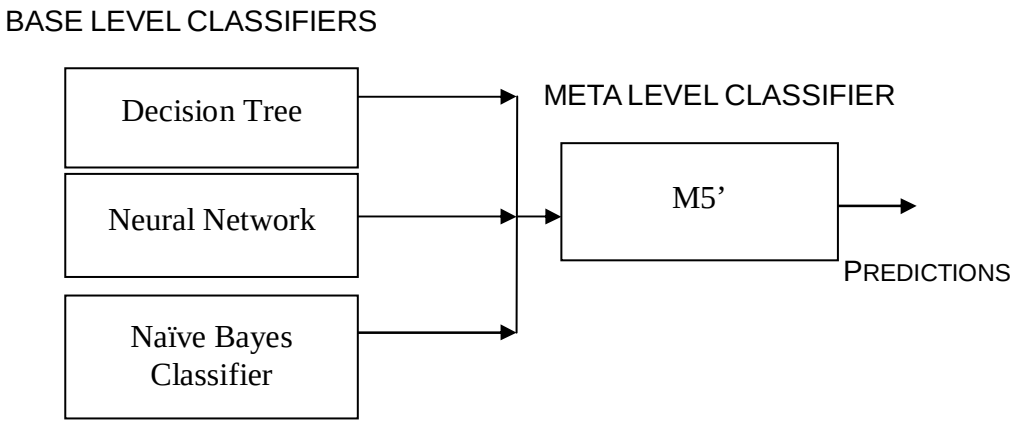


Fig 6.1: Model ensemble using stacking

Some Meta level algorithms expect only two class problems. So, in order to test those possibilities., the 4-class problem may be modelled as a 2- class problem, by combining attributes "Excellent" and "Good" as "Excellent" and "Average" and "Poor" as "Poor" . Then experiments on this two class problem showed that with the same base level classifiers, the voted perceptron algorithm was giving better performance in terms of accuracy (82%).

The Voted Perceptron algorithm: In the voted-perceptron algorithm, more information is stored during training and then it uses this elaborate information to generate better predictions on the test data. The algorithm is detailed below.

The information maintained during training was the list of all prediction vectors that were generated after each and every mistake. For each such vector, the number of iterations it survived was counted until the next mistake was made; this count was referred as the weight of the prediction vector. To calculate a prediction, the binary prediction of each one of the prediction vectors was computed and all these predictions were combined by a weighted majority vote. The weights used are the survival times described above. This makes intuitive sense as good prediction vectors tend to survive for a long time and thus have larger weight in the majority vote.

The algorithm:

Input: A labelled training set $<(x_1,y_1)\ldots.(x_m, y_m)>$ where $x_1 \ldots x_m$ are feature vector instances and $y_1 \ldots y_m$ are class labels to which the training instances have to be classified, T is the no of epochs.

Output: A list of weighted perceptrons $<(w_1,c_1)\ldots(w_k,c_k)>$ where $w_1 \ldots w_k$ are the prediction vectors and $c_1 \ldots c_k$ are the weights.

K=0

$w_1 = 0$

$c_1 = 0$

Repeat T times

For i = 1 to m

If (x_i, y_i) is misclassified:

$w_{k+1} = w_k + y_i x_i$

$c_{k+1} = 1$

$k = k + 1$

Else

$c_k = c_k + 1$

At the end, a collection of linear separators w_0, w_1, w_2, etc, along with survival times: c_n = amount of time that w_n survived is returned. This c_n is a good measure of the reliability of w_n. To classify a test point x, use a weighted majority vote: y'=sgn(S) where S is the sign function which returns output to a range as one of the values {0, 1,-1}. This is shown in equation (6.1).

$$\mathbf{Y'=}\ \operatorname{sgn}\left\{\sum_{n=0}^{N} c_n\ \operatorname{sgn}(w_n \cdot x)\right\} \qquad \textbf{- (6.1)}$$

6.7 Advantages of stacking ensemble methods

Stacking takes place in two phases. In the first phase each of the base level classifiers takes part in the j- fold cross validation training where a vector is returned in the form $<(y'_0 \ldots y'_m), y_j>$ where y'_m is the predicted output of the m^{th} classifier and y_j is the expected output for the same . In the second phase this input is given for the Meta learning algorithm which adjusts the errors in such a way that the classification of the combined model is optimized. This process is repeated for k-fold cross validation to get the final stacked generalization model. It is found that stacking method is particularly better suited for combining multiple different types of models. Stacked generalization provides a way for this situation which is more sophisticated than winner-takes-all approach [16, 39]. Instead of selecting one specific generalization out of multiple ones, the stacking method combines them by using their output information as inputs into a new space. Stacking then generalizes the guesses in that new space. The winner-takes-all combination approach is a special case of stacked generalization. The simple voting approaches have their obvious limitations due to their abilities in capturing only linear relationships. In stacking, an ensemble of classifiers is first trained

using bootstrapped samples of the training data, producing level-0 classifiers. The outputs of the base level classifiers are then used to train a Meta classifier. The goal of this next level is to ensure that the training data has accurately completed the learning process. For example, if a classifier consistently misclassified instances from one region as a result of incorrectly learning the feature space of that region, the Meta classifier may be able to discover this problem. Using the learned behaviours of other classifiers, it can improve such training deficiencies [16].

It is always an active research area that whether combining data mining models gives better performance than selecting that model with best accuracy among base level classifiers. In this research also, in pursuit for finding the best model suitable for this problem, this possibility was explored. In combining of the models usually the models in level 0(base level classifiers) are operated in parallel and combined with another level classifier called as Meta level classifier. In this work, using decision tree, neural network and Naive Bayes classifier as the base level classifiers, various Meta level classifiers have been tested and it was observed that multi response model tree Meta level classifier performed best among others. Although numerous data mining algorithms have been developed, a major concern in constructing ensembles is how to select appropriate data mining algorithms as ensemble components. As pointed out earlier numerous research works are being taken place in the field of ensemble of classifiers and many of them are proposing different types of classifiers at the base level and Meta level depending on the type of application. This research work is also giving light into the field of data mining research by proposing an efficient combination of base and Meta level classifiers for a social science problem like employment prediction.

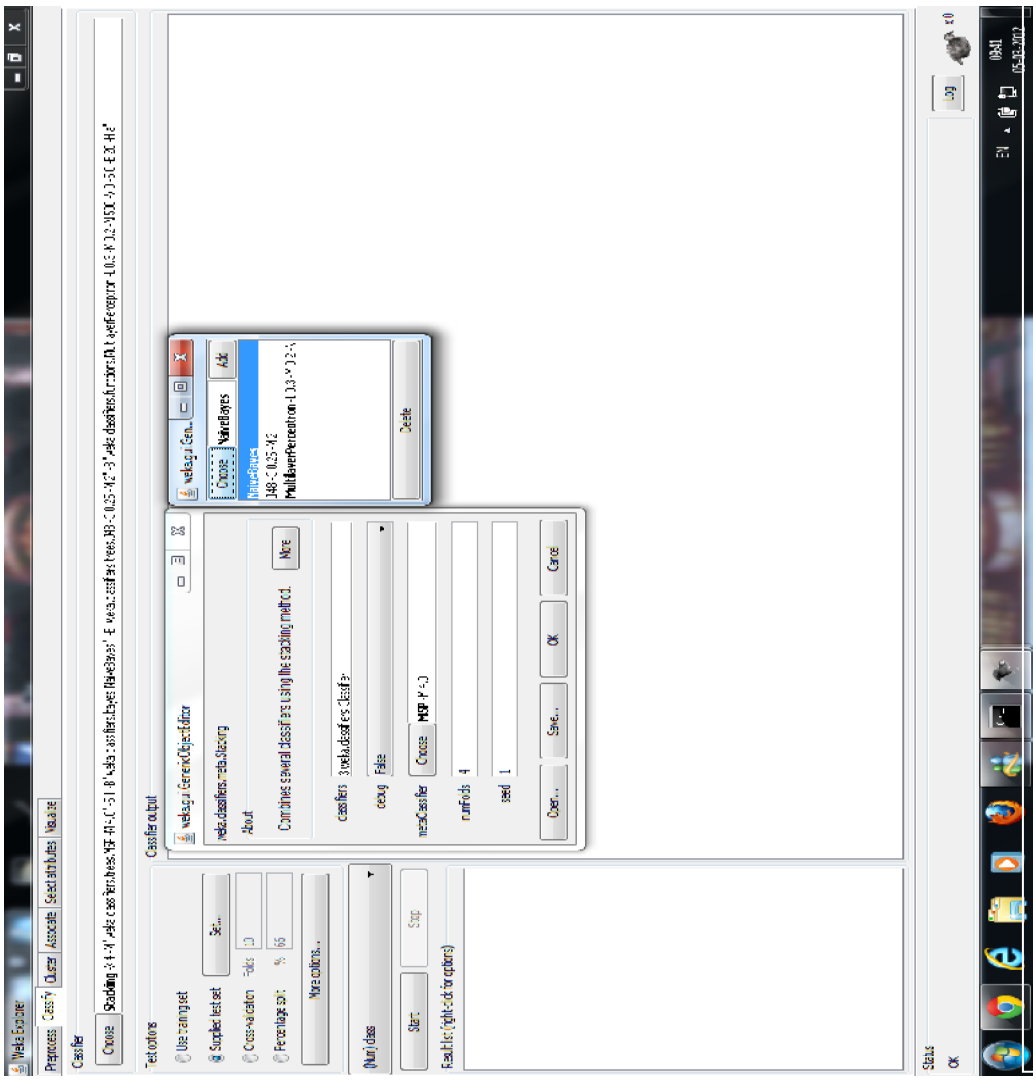


Figure 6.2: Weka Experimental set up for ensemble learning

6.8 Chapter summary

This chapter suggests the need for making an ensemble of classifiers and the various methods for making it. One of the hottest questions in front of data mining researchers today is "Whether a combined classifier model gives better performance than the best among the base level classifiers". This important question is tried to be addressed in this thesis. For exploring this there are two approaches-voting based techniques and stacking based techniques. The basic difference between stacking and voting is that in voting no learning takes place at the Meta level, as the final classification is by votes casted by the base level classifiers, whereas in stacking, learning takes place in the Meta level. A Meta level is the level at which the base level classifiers are combined using an algorithm. In this work, the base level classifiers are stacked with many meta learning algorithms like bagging, regression based algorithms etc, and it was observed that when multi response model tree algorithm is used at the Meta level the model is giving much better performance. Hence through this work, it is strongly suggested that a combined ensemble approach gives a better performance than selecting the best base level classifier, which also confirms few other research results that have happened in other functional domains [11, 25, and 38].

Chapter 7

Absorption rate prediction

This chapter describes absorption rate prediction, a derivative of the research work explained in previous chapters. Here the percentage of passed out students in a branch who may get a placement in any coming year is computed. Also the approximate time they have to wait for 100% placement, i.e. when the whole students passed from a branch in a particular year get placed, is computed. This will serve as a useful input for the government for deciding whether to increase or decrease the intake of a particular branch.

7.1 Absorption rate prediction

In this work, for any engineering branch of study, the percentage of students who may get a placement in any future year is predicted, using statistical regression techniques. The approximate waiting time for a batch of students of any branch for getting a total placement is also computed. This subtask is called waiting time prediction. So the subtasks can be listed as follows:

- Predict the job absorption rate for a particular branch i.e. to figure out the percentage of students of a particular branch that will be placed in a particular branch in a particular year in the future based on the existing trends. The concept of regression is used to formulate equations to determine the percentage of students who are placed. Input is the year wise final absorption rate for every branch.
- From the history data, placement rate status of a particular batch for a period of every 3 months for each year is calculated. From this data one should be able to predict the time needed to attain 100% placement for the given batch. The concepts of trend lines are used to provide accurate results.

The outcome of these subtasks will help the NTMIS to determine which branches need more intakes, since they have good placement rates and also will help in drawing attention to the branches which are lagging in terms of placement.

7.2 Time series analysis

Statistics regarding placement of students has become one much sought after information by the students all over the world. Data mining that works on statistical techniques can be used for this purpose. Visualisation techniques like histograms and pie chart can be effectively used for the display of statistical information. Predicting percentage of placement and placement time has lack of accuracy other than being a tedious job. The need is to develop an interface that uses a database of statistical data and predicts accurately the time in which 100% placement will be achieved for a particular branch of students who have passed out in a particular year. The waiting time prediction is useful in the sense that more the waiting time for a branch, more it indicates that intake for the coming years should be reduced.

In this work, time series based statistical data mining techniques are used to predict job absorption rate for a discipline as a whole. Each time series describes a phenomenon as a function of time. For example, daily stock prices could be used to describe the fluctuations in the stock market. In general, for a time series *X* with *n* observations, *X* is represented as in (7.1)

$$X = (v_1 ,t_1), (v_2 ,t_2), ..., (v_n ,t_n) \quad \textbf{- (7.1)}$$

Where v_i and t_i, are the observation value and its time stamp, respectively. The data that is used in this work to compute the absorption rate is a time series data, where absorption rates for different years are used. Many researchers are working on applications that use time series analysis. Data mining on time series data is essential nowadays to improve the business analysis and forecasting. Several Works are already done in analyzing time series data like astronomical data [29], business data etc. But nowadays using sophisticated

statistical packages like Table Curve, SAS, SPSS etc, regression equations can be obtained as used in this work.

7.3 Curve fitting

Curve fitting is finding a curve which has the best fit to a series of data points. A first degree polynomial y = ax + b will connect any two points. A second degree polynomial, ($y = ax^2+bx+c$) will exactly fit three points. Curve fitting to a straight line: $y = ax + b$, is called "linear regression". For curve fitting either to a straight line or polynomial function, the best-fit coefficients has to be found out in one step. Figure 7.1 shows the idea behind line fitting.

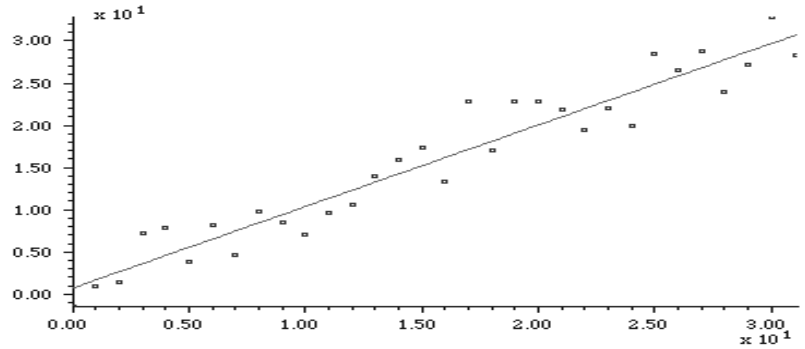


Figure 7.1: Line passing through common points

7.4 Key concepts used in regression analysis

7.4.1 Linear regression

Linear regression analyzes the relationship between two variables, X and Y. For each data point, both X and Y are known and the best straight line through the data has to be found out. In some situations, the slope and/or intercept may have a scientific meaning. In other cases, the linear regression

line can be used as a standard curve to find new values of X from Y, or Y from X.

In general, the goal of linear regression is to find the line that best predicts Y from X. Linear regression does this by finding the line that minimizes the sum of the squares of the vertical distances of the points from the line. Note that linear regression does not test whether the data are linear (except via the runs test). It assumes that the data are linear, and finds the slope and intercept of the straight line that best fits the data.

7.4.2 Non-linear regression

Nonlinear regression is a general technique to fit a curve through the data. It fits data to any equation that defines Y as a function of X and one or more parameters. It finds the values of the parameters that generate the curve that comes closest to the data (minimizes the sum of the squares of the vertical distances between data points and curve). Except for a few special cases, it is not possible to directly derive an equation to compute the best-fit values for the data. Instead nonlinear regression requires a computationally intensive, iterative approach. Non-Linear Regression uses matrix algebra and it fits a mathematical model to the underlying data.

7.4.3 R-squared value

R-Squared is a statistical term denoting how good one term is at predicting another. If R-Squared is 1.0 then given the value of one term, the value of another term can be perfectly predicted. The value 0 does not help in prediction. More generally, a higher value of R-Squared means that one term can be predicted from another in a better way. R-Squared is most often used in linear regression. Given a set of data points, linear regression gives a formula for the line most closely matching those points. It also gives an R-Squared

value to say how well the resulting line matches the original data points. In this work, table curve package is used to compute these values.

7.4.4 Trend lines

Patterns of gradual change in a condition, output, process, average or general tendency of a series of data points that move in a certain direction over time can be represented by a line or by a curve on a graph. Graphical representation of time series data (information in sequence over time) showing the curve that reveals a general pattern of change is known as Trend Chart. Trend line can be defined as a straight or curved line in a trend chart that indicates the general pattern or direction of a time series data. It may be drawn visually by connecting the actual data points or (more frequently) by using statistical techniques such as 'exponential smoothing' or 'moving averages. 'Trend line forecasting is a simple technique of estimating future values of a time series data by extending the trend line into future. This simple method is used for predicting 100% placement. Trend lines can offer great insight, but if used improperly, they can also produce false signals. Other items, such as horizontal support, resistance levels, peak-and-trough analysis etc should be employed to validate trend line breaks. While trend lines have become a very popular aspect of technical analysis, they are merely one tool for establishing, analyzing, and confirming a trend.

7.5 Input data Processing

The Initial database provided by Nodal Centre, was in FoxBase format. It was to be converted to some latest DBMS like mySQL to make the approach efficient and faster. First FoxBase data was converted to CSV files (Comma Separated files) and this file was loaded to MS Excel. Then from this Excel format using xls-mySQL converter it was converted to mySQL format.

For Absorption-rate prediction (degree and diploma separate), the data for years 1983-2008 was used. The data had two attributes namely year and absorption percentage (For e.g., (1983, 95%), (1984, 73%) and so on). For waiting time prediction, the data had three attributes namely year, month, and absorption percentage up to that month. To prepare this data, the three attributes extracted were Roll no of the candidate, Month and year at which he/she joined the Company as shown in table 7.1. The database created using these three attributes was then fed to an xls-sql converter and finally a mysql database was created. These attributes were fed into mysql through sql queries and each of these entities was put under the database of the respective branch. Sorting was performed, first using the joining year followed by the joining month to prepare the final attributes (year, month, absorption percentage up to that month) from these extracted attributes. A partial view of the data set after processing, which was used for waiting time prediction is shown in table 7.2.

Roll No	Month of getting placement	Year of getting placement
10483	7	2001
10508	7	2001
10473	8	2001
10476	9	2001
10482	9	2001
10480	10	2001
10492	11	2001

Table 7.1: A Partial view of the data set with attributes extracted from raw data for waiting time prediction

Year of getting placement	Month of getting placement	Percentage of placement
2001	8	15
2001	11	22
2001	13	26
2001	16	33
2001	18	41

Table 7.2: A Partial view of the data set after processing for waiting time prediction

7.6 Logic

7.6.1 Absorption rate prediction

The attributes like year and the percentage of students absorbed into job that year were extracted from the database. A graph was plotted using these two fields. A pictorial representation of the trends for the Electronics and Communication branch is given in figure 7.2. An equation for this graph was obtained for predicting the absorption rate in the future. A package called "TableCurve" was used for this purpose. TableCurve helps researchers to find the ideal model for even the most complex data. Next, all possibilities for fitting into different graph shapes like linear and non linear equations has to be considered. Each of those graphs was matched (overlapped) with the original graph and the graph which was most resembling and including the maximum points was chosen. The equation to be selected was chosen based on best R-squared value returned by the software. This method is very similar to the non-linear regression method. The equation thus obtained can be used to calculate the percentage of students absorbed into a job for the year which is input to it.

7.6.2 Validation

Validation of the methods were conducted in such a way that all the available year (1983-2008) values were substituted in the equation and predicted absorption rates were compared with observed absorption rates .

The following equation shows the equation of the fitted curve returned by the package Table curve for the Electronics and communication branch. The data set used for prediction is shown in table 7.3 and the coefficients of the obtained equation are shown in table 7.4. The accuracy observed was 87.5%, assuming a 3% variation among actual and observed values. The graph obtained for the above equation is shown in figure 7.2.

$$Y = a+b\cos(x)+c\sin(x)+d\cos(2x)+e\sin(2x)+f\cos(3x)+g\sin(3x)+h\cos(4x) +i\sin(4x)+j\cos(5x)+k\sin(5x)+l\cos(6x)+m\sin(6x). \quad \textbf{- (7.2)}$$

Similarly the absorption rate prediction for civil and computer science branches are shown in Appendix - B.

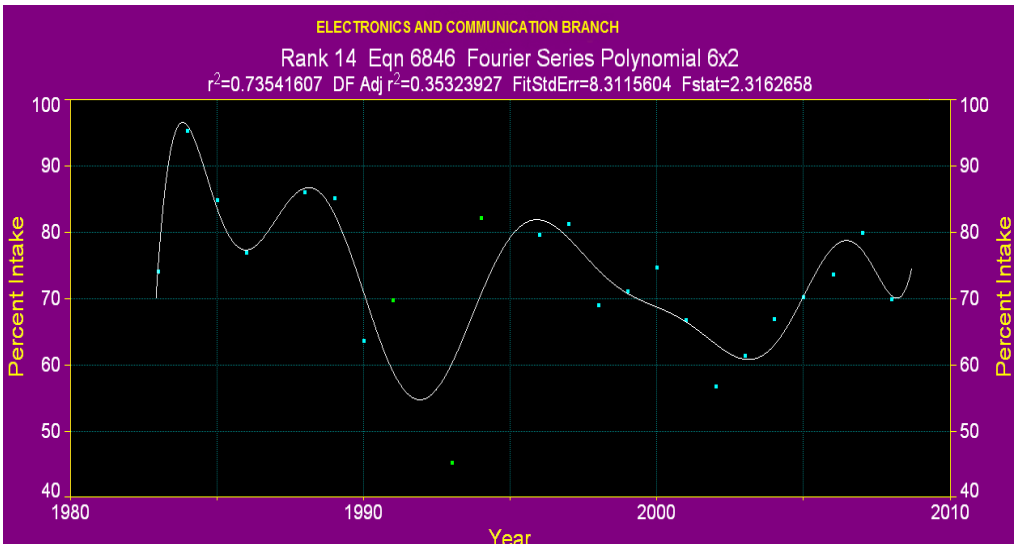


Figure 7.2: Absorption rate trend curve for the Electronics and communication branch

The values of the coefficients of the equation are also shown. The R-Square value observed has been 0.74. The X values used are years and Y values used shows the total placement percentage of students passed in a particular year for this branch. Y Predict shows the value predicted by the fitted equation.

X Value	Y Value	Y Predict
1983	74.200000	74.017976
1984	95.500000	96.233889
1985	85.000000	84.022007
1986	77.000000	77.356317
1988	86.200000	86.673699
1989	85.300000	83.045199
1990	63.700000	71.293640
1991	69.900000	59.234849
1993	45.300000	59.933734
1994	82.300000	70.346374
1996	79.700000	81.931194
1997	81.300000	79.175669
1998	69.100000	74.431900
1999	71.200000	70.844679
2000	74.800000	68.777072
2001	66.800000	66.535418
2002	56.800000	63.241354
2003	61.500000	60.865410
2004	67.000000	62.905311
2005	70.300000	70.168489
2006	73.700000	77.612444
2007	80.000000	77.404846
2008	70.000000	70.548529

Table 7.3: Data used for absorption rate prediction of Electronics and Communication branch

Coefficients	Value
a	142.7937045
b	-1437.15001
c	-250.352700
d	-462.026922
e	2136.199433
f	1902.553209
g	565.3156406
h	459.5054449
i	-1150.76208
j	-463.668472
k	-235.286195
l	-67.9889743
m	94.26857602

Table 7.4: Values of coefficients of equation obtained for absorption rate prediction of Electronics and Communication branch

7.7 Waiting –time prediction

The waiting time prediction was done by plotting the placement rate on the Y axis and time period in months in X axis. The variable Y depends on X given by the equation (7.3)

$$Y = mX + c \quad \textbf{- (7.3)}$$

The major issue encountered while plotting the values was that a straight line was never obtained whenever a graph was plotted for a given branch's database. To get a 'trend line' for the plotted graphs, a mathematical library called chartdirector was used. The library had inbuilt functions which provided the slope and the y – intercept of the trend line. A trend line is a straight line that fits a number of data points computed using linear regression. The library functions perform linear regression analysis on the data points, and represent the result as a best fit straight line. In linear regression analysis, the data points are assumed to be related by (7.4)

$$y = m * x + c + err \quad \textbf{- (7.4)}$$

The uncertainties of the data point are contributed by the uncertainties in m, c, and err. The uncertainties of the data points are represented visually as a prediction band around the regression line. For example, a 95% prediction band means there is 95% probability that a data point will be in that band. The prediction band is always wider than the confidence band. It is because the uncertainties of the regression line are contributed by m and c, while the uncertainties of the data points are contributed by m, c and err. The err term makes the data points less certain than the regression line.

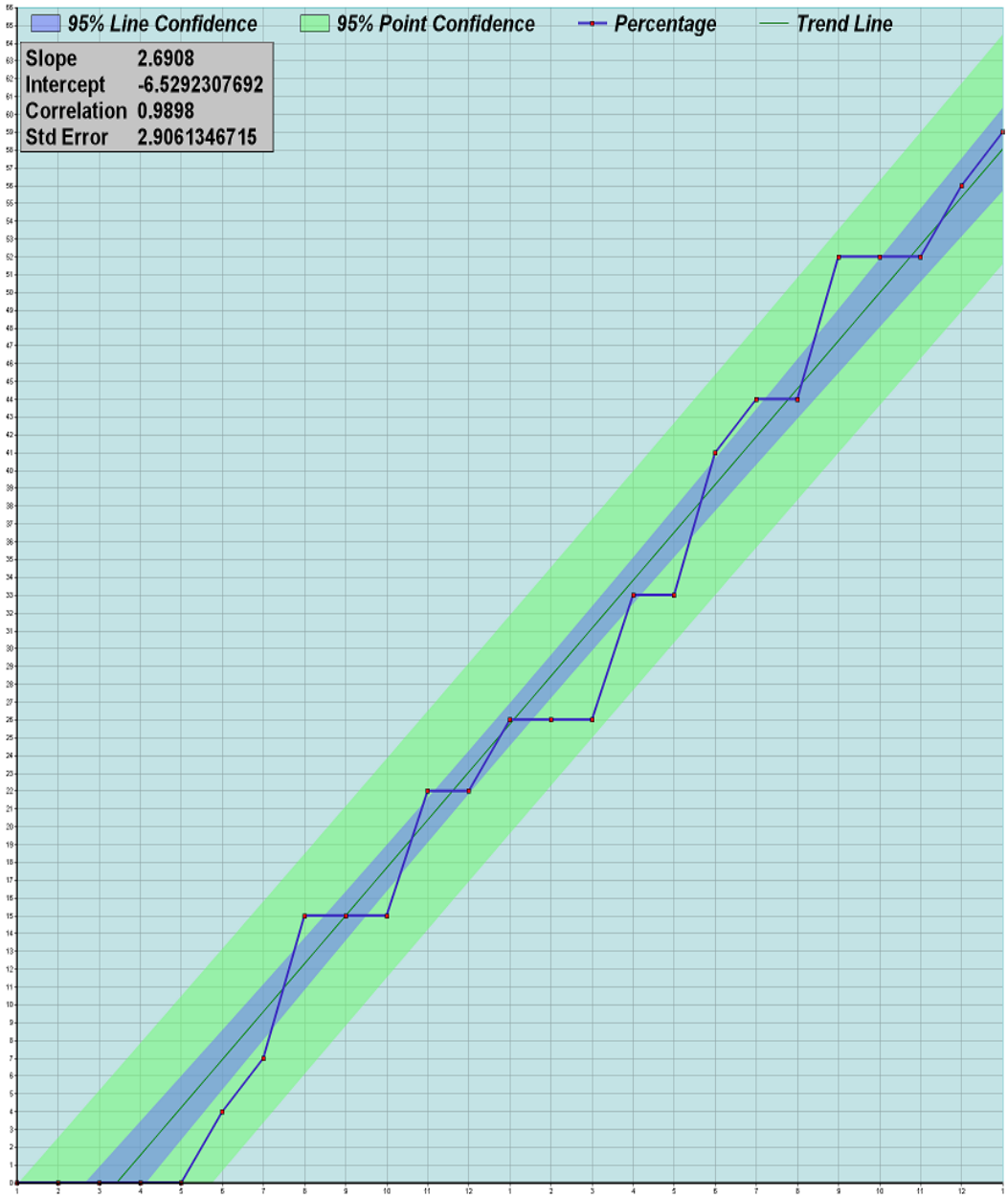


Figure 7.3: Representation of placement rates Vs month to calculate 100% case

Figure 7.3 shows a graph plotting the waiting time prediction trendline. In that the Y axis is placement rate and the X-axis is months. After a unit of 12 months, the month values in the X-axis of the graph restarts from 1. The figure also shows the 95% confidence band and prediction band of the work. This means it is 95 % sure that the curve lies within the confidence band and 95% of any new points will be in the prediction band.

7.7.1 Validation

The method used to find 100% placement could be verified by reversing the procedure with a known result. Absorption rates of actual data (y value) were substituted in these trend line equations computed by the software and predicted time taken(x value) for that particular absorption rate was computed for available data of each year. It was observed that the time computed for a particular absorption rate percentage in the above method was same as the observed value with a small deviation of 3%, since the database of absorption rates follows a trend. Hence mathematically it can be said that it will be correct for 100% absorption time also.

7.8 Chapter summary

In this chapter the two related works of this research work namely absorption rate prediction and waiting time prediction are discussed. In absorption rate prediction the input is year wise absorption rate for each branch. The absorption rate for any branch for a future year is to be predicted using time series analysis. In waiting time prediction using regression analysis, the time needed for getting the full students of a batch to get placements is computed. This is also called 100% placement waiting time.

Chapter 8

The C 4.5*stat algorithm

*This chapter explains a new algorithm namely C 4.5*stat for numeric data sets. It is a variant of the C 4.5 algorithm and it uses variance instead of information gain for the tree construction.*

8.1 Motivation

Decision tree is an important classification technique which can support different data types for the attributes. While handling numeric data sets, they are first converted to categorical types and then classified using information gain. Information gain is a very popular and useful concept which tells how good an attribute is for predicting the class of each of the training data. But this process is computationally intensive for large data sets. Also popular decision tree algorithms like ID3 cannot handle numeric data sets. This work proposes statistical variance and statistical mean as alternatives to information gain to split nodes in completely numerical data sets. The new algorithm namely C 4.5*stat was proved to be competent with C4.5, against the standard UCI benchmarking datasets. This algorithm was developed as a related work of the main research work. However, as this algorithm was intended for numeric data sets, it was proved against standard UCI datasets and not on the placement history datasets used in this work. The specific advantages of this proposed new algorithm are,

- It avoids the computation of information gain for large numeric data sets. Instead it uses statistical variance, which is much easier to compute. The accuracy is still competent with standard C 4.5.
- It avoids the conversion of large numeric data sets to categorical data sets, which is a time consuming task.
- For data sets with more number of attributes it is advantageous in terms of time complexity.

8.2 C4.5*stat algorithm

1. Let the set of training data be S. Put all of S in a single tree node.
2. If all instances in S are in same class, then stop
3. Split the next node by selecting an attribute A, for which there is minimum Statistical Variance. Put the split point as the Statistical Mean of the current subset of data.
4. Stop if either of the following conditions is met, otherwise continue with step 3:
   (a) If this partition divides the data into subsets that belong to a single class and no other node needs splitting.
   (b) If there are no remaining attributes on which the sample may be further divided.

In conventional decision tree algorithms like C4.5, the splitting will be done based on the maximum information gain concept. But here the statistical variance is used, which is defined as follows:

In general, the ***population variance*** of a *finite* population of size *N* is given by equation (8.1)

$$\sigma^2 = \frac{1}{N}\sum_{i=1}^{N}(x_i - \mu)^2 \qquad \textbf{- (8.1)}$$

Where μ is the population mean as given by equation (8.2):

$$\mu = \frac{1}{N}\sum_{i=1}^{N}x_i \qquad \textbf{- (8.2)}$$

Here the assumption is that, if a subset of the data is having low variance then there is a chance that they converge to a particular class in minimum number of iterations as there is minimum variation in the data for that attribute.

8.3 Implementation

For implementing the algorithm, the popular open source data mining package WEKA 3.4 was used. There program for C4.5*stat algorithm was coded using JAVA, by integrating WEKA with Netbeans 7.0 IDE. Many Weka core classes were reused giving more accurate and systematic implementation. The advantage of using Netbeans IDE was that one can view the test results immediately in the IDE, just as that available in WEKA. A screenshot of the Netbeans integration with Weka for developing new algorithms for classifiers is shown in figure 8.1.

8.4 Validation

The validation of C 4.5*Stat algorithm was done against the universally accepted UCI benchmark datasets. Cross validation using 4 folds was used as a standard testing strategy. UCI repository has got different data sets out of which there are 7 purely numeric data sets that are supplied with Weka data mining suite. They were used for testing the java implementation of the new C4.5*stat algorithm. There are classifiers that can handle numeric data sets. Among them Neural Network, Naïve Bayes classifier and C4.5 gave comparably good accuracies whose performances are also recorded in this work.

8.5 Validation results

Table 8.1 shows the summary of the test results. Here some of the existing decision tree algorithms having reasonably good accuracy and computing time are considered for comparison. It is clear that compared to other decision tree algorithms, C 4.5*stat have clear advantages in terms of accuracy over these data sets. The next task was to compare its accuracy with

other popular classification algorithms for numeric data sets namely C 4.5, neural networks and Naive Bayes classifier. In table 8.2, these comparisons are shown. It is observed that C 4.5*stat algorithm has competent accuracy. In statistics, to further generalise the result for a larger population, there are many hypothesis testing strategies. The statistical technique namely ANOVA was used here. In statistics, **analysis of variance** (**ANOVA**) is a collection of statistical models, and their associated procedures, for testing a hypothesis. In its simplest form ANOVA provides a statistical test of whether or not the means of several groups are all equal, and therefore generalizes *t*-test to more than two groups.

Dataset	ADTree	REPTree	RandomTree	C 4.5 * stat
Iris	NA	96	94	95.3
Segment	NA	94.81	89.13	94.07
Diabetes	73.17	73.56	68.09	74.47
Letter	NA	NA	82.875	81.34
Breast-cancer	95.7	94.7	93.84	95.13
Glass	NA	66.82	62.61	67.28
Labor	82.45	68.42	85.96	82
Mean	**83.77**	**82.385**	**82.35**	**84.23**

Table 8.1: Accuracies of various decision tree algorithms on UCI Data sets

Dataset	C4.5*stat	C 4.5	Neural Network	Naïve Bayes
Iris	95.3	95.3	95.3	94.66
Segment	94.07	96.17	95.18	77.03
Diabetes	74.47	73.3	75.78	75.78
Letter	81.34	86.59	82.56	63.99
Breast-cancer	95.13	95.56	96.28	95.99
Glass	67.28	66.82	67.28	45.32
Labor	82	73.68	89.47	92.98
Mean	**84.23**	**83.92**	**85.97**	**77.96**

Table 8.2: Accuracies of various classification algorithms on UCI Data sets

ANOVA testing is useful when the means of groups of data to be compared are two or more. A t-test can be used to compare two means. A multiple t-test can also be used to compare more than two means. But the procedure may become more complicated as one has to first construct many pairs among many groups and then proceed with pair wise analysis. But in ANOVA, the procedure is much straight forward. In ANOVA, one starts with the assumption or the null hypothesis that the means are equal.

So in this work, the null hypothesis considered was that the mean of accuracies of all the algorithms namely C 4.5*stat, C 4.5, neural networks and Naive Bayes classifiers were comparable. Then after the test, the most important value to be observed is the p-value of the ANOVA test. According to ANOVA testing if this value is greater than 0.5, the null hypothesis can be accepted. But in this case it was 0.665. Hence the null hypothesis can be accepted and accuracies were concluded to be competent with each other. Neural networks showed a slightly higher accuracy, but lengthy and complex training time can be its limiting factor.

The time complexity of standard decision tree algorithm is O (mn^2), where m is the number of records and n is the number of attributes [72]. This is because there are total m records itself among all nodes in a particular level at a time and for computing information gain, it has to consider each of the n attributes. So in a particular level, the complexity is O (mn). In the worst case, there will be a split corresponding to each of the n attributes. So altogether it becomes like O (mn^2) in the worst case. But here as the numeric data are split based on statistical mean the number of levels in the worst case is $\log_2 m$. So the time complexity becomes O ($mn\log_2 m$). So as the numbers of attributes become very high, which is common in huge data sets like bioinformatics data, this algorithm will have an edge in terms of time.

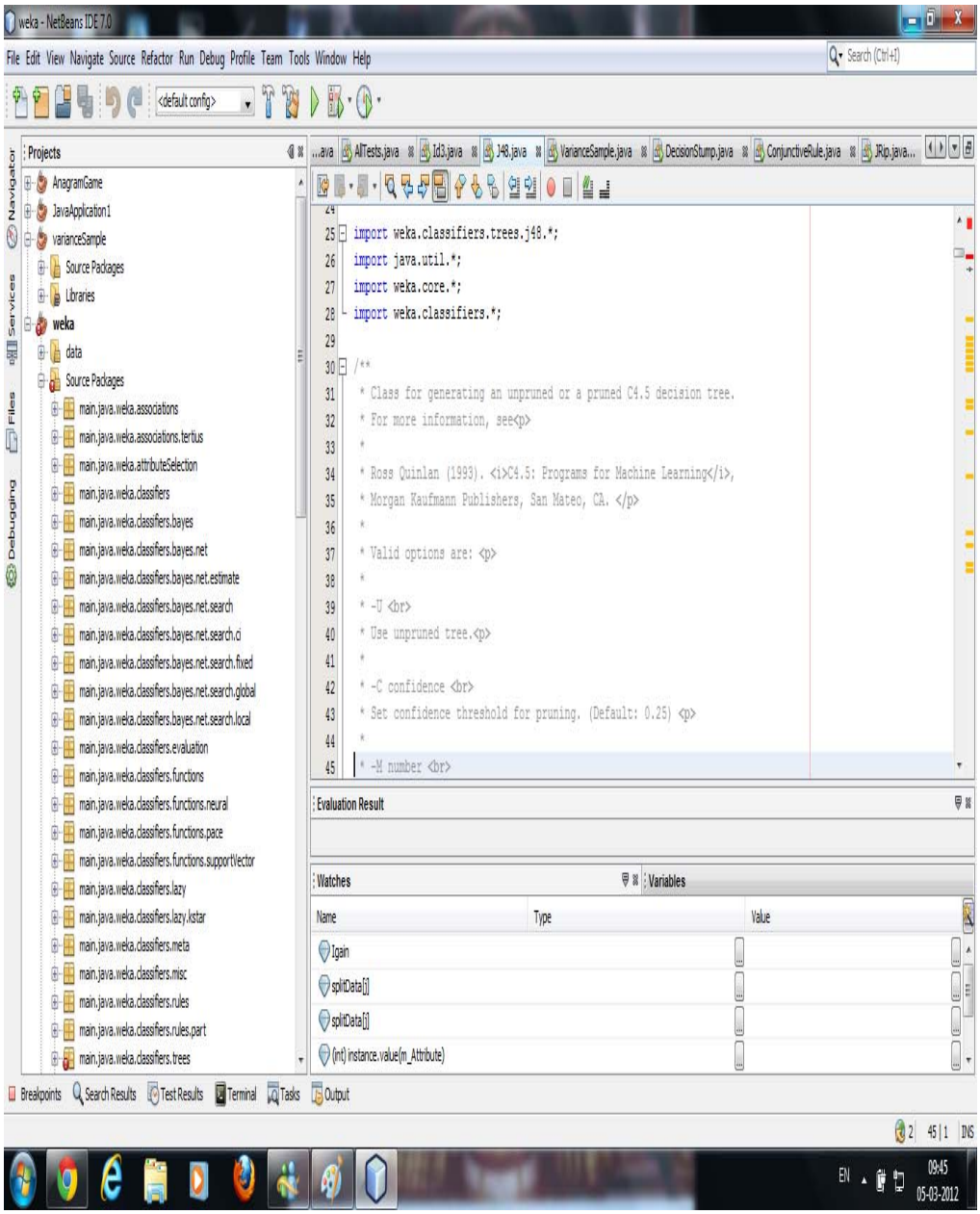


Figure 8.1: Integrating WEKA with Netbeans IDE for developing new classifiers

8.6 Chapter summary

In this chapter a new algorithm namely C 4.5 * stat for numeric data sets is described. This algorithm is found to be competent in accuracy with its information gain counterpart C 4.5. Instead of information gain, this algorithm uses statistical variance. An improvement in computing time is also found, when the number of attributes of the data set increases.

Chapter 9

Conclusions and future scope

This chapter discusses various inferences that are drawn from this research work and further directions in which this work can be extended.

9.1 Conclusions

This research work is about how data mining techniques can be effectively used to solve a social science problem namely employment chance prediction. The data mining models developed are capable of predicting chances of employment for a student choosing a particular branch for his engineering studies. It starts from attribute analysis for determining the most decisive attributes those can best decide the placement chance. The authentic data supplied from nodal centre at Cochin University of Science and Technology, was used for the study.

Three popular data mining models were considered for the study. They were decision trees, neural networks and Naive Bayes classifier. The models were built from the training data of years 2000-2002 and tested using test data of year 2003. Further verification of the models was done using data of year 2008 for which the models were built using the three previous year data (2005-2007). The performance of the models was compared using various statistical measures like accuracy, precision, recall, ROC and so on. It was concluded that these three model performances are comparable with each other, though Naive Bayes classifier is found to give better values for ROC area, recall etc. It is also observed that the model gives best performance when it is being trained with previous three year data, and tested with the data of the subsequent year.

As an offshoot of this research work, two other problems namely absorption rate prediction and waiting time prediction were also addressed. In the former case, the year wise placement rates for various branches for different years were used. The absorption rate of any branch in any coming year was predicted using regression or time series analysis. In waiting time

prediction, a methodology to predict the average time students have to wait until the 100% of the students passed in that year gets a placement, was developed. Both these methodologies provide valid input to the government on deciding whether to increase or decrease the student intake in coming years for a particular branch. Currently nodal centre is doing more research in this direction and similar study on courses other than engineering is also under active consideration.

In addition to the development of basic classifiers like decision trees, neural networks and Naive Bayes classifier, a study on how to improve the performance of the models was also conducted. Ensemble methods were considered for that. One or more homogenous or heterogeneous models can be an ensemble. Homogenous ensembles are in place for a long time as it is simple to experiment using any one tool. With the advent of software packages like WEKA, the testing of heterogeneous ensemble methods has become possible nowadays using the meta learning add-ons in the package. M5' is the meta learning algorithm used in this work for developing ensemble. It was inferred that combing the classifiers gives a better performance than selecting the best base level classifier.

Also this research work developed a new data mining algorithm namely C 4.5 * stat for numeric data sets which had proved to have competent accuracy over standard benchmark data sets called UCI data sets. This algorithm may have an edge for huge data sets where number of attributes is very high as in the case of bioinformatics data. Also the effect of parameter tuning for optimization of decision trees was also analysed.

During the progress of this research work, many interesting inferences as well as many possible future directions, to which this work can be extended, were identified. They are described in the coming section.

9.2 Future directions

National Technical Manpower Information System (NTMIS) was set up by the Ministry of Human Resource Development, Government of India to provide up-to-date and meaningful manpower requirement information on a continuing basis to plan for technical manpower development scientifically. The nodal centre at Cochin University of Science and Technology is one among the 20 nodal centres located at different parts of the country. This scheme is funded by AICTE and efforts are taken by the AICTE to strengthen it as an up-to-date and dynamic national data base.

If the data mining models developed as a part of this research work are implemented at the nodal centre, an efficient analysis and usage of the authentic national data can be achieved. In future, this can be used by many admission seekers to analyse and choose the most suitable engineering branch for them.

This research work was carried out by analysing the data of students belonging to Kerala state alone. It will be of great benefit to the country if this research work can be spanned to the whole country or at the international level such that the data may reflect different cultures, different characteristic as a whole and can be generalised in a better way.

In this work, the data of passed out students from the engineering domain was considered. Similarly one can analyse the data from other domains like medical, legal and other P.G studies to get some interesting

hidden information. Since the nature of data may be different, the deciding attributes and the data mining models may vary giving rise to more contributions to the data mining field.

Another area to which this research work can be extended is association rule mining where new interesting hidden patterns can be found from the data. Assuming a student is opting for computer science branch and is given admission in a particular college, it can be investigated whether he has better chances of placement. Finding out the information about dropouts from engineering courses and finding the reasons for it can be a useful future work. Similarly one can investigate whether going for higher studies is giving any extra push for the employment chance of a student. As a summary when data mining techniques are effectively applied over authentic, up-to-date, national level data base, many hidden and useful information can be retrieved, which can be efficiently used both at government and general public level for future planning policies.

BIBILIOGRAPHY

[1] A. Agogino, K. Tumer, "Efficient agent-based cluster ensembles," In Proceedings of the Fifth International Joint Conference on Autonomous Agents and Multi-Agent Systems, pp. 1079 – 1086, Hakodate, Japan 2006.

[2] H. Alaaeldin, "Association mining of dependency between time series," in Proceedings of SPIE Vol. 4384, pp. 291-301, 2008.

[3] E. Alfaro, N. García, M. Gámez, D. Elizondo, Bankruptcy, "Forecasting: an empirical comparison of AdaBoost and neural networks," Decision Support Systems, 45 (1), pp 110–122, 2008.

[4] Anandavalli, Ghose, Gauthaman, "Mining spatial gene expression data using association rules," CSC -IJCSS, 3(5), pp. 351-357, 2009.

[5] Antony Browne, Brian.D.Hudson, David.C.Whitley, Martyn.G.Ford, Philip Picton, "Biological data mining with neural networks: implementation and application of a flexible decision tree extraction algorithm to genomic problem domains," Elsevier Journal of Neuro computing, 57(1), pp. 275-293, March 2004.

[6] Anyanwn, Shiva, "Comparative analysis of serial decision tree classification algorithms," CSC- IJCSS, 3(3), pp. 230-240, 2009.

[7] E. Bauer and R. Kohavi. An empirical comparison of voting classification algorithms: Bagging, boosting, and variants. Machine Learning, 36(1-2), pp. 105–139, 1999.

[8] L. Breiman, J.H. Friedman, R.A. Olshen, and C.J. Stone, "Classification and regression trees," Wadsworth International Group, 1984.

[9] L. Breiman, “Bagging predictors,” ACM journal of Machine Learning, 24(2), pp. 123–140, 1996.

[10] L. Breiman, “Random forests,” ACM journal of Machine Learning, 45(1), pp. 5–32, 2001.

[11] J.B.D. Cabrera, C. Gutierrez, R.K. Mehra, “Ensemble methods for anomaly detection and distributed intrusion detection in mobile ad-hoc networks,” Information Fusion (Special Issue on Applications of Ensemble Methods), 9(1), pp. 96-119, 2008.

[12] Cezary Z. Janikow, “Fuzzy decision trees: issues and methods,” IEEE Transactions on Systems, Man, and Cybernetics, Part B, 28(1), pp. 1-14, 1998.

[13] P.K. Chan, S.J. Stolfo, “A comparative evaluation of voting and meta-learning on partitioned data,” In Proceedings of the International Conference on Machine Learning, pp. 90-98, California, USA, 1995.

[14] N.Chawla, K.Bowyer, ”Designing multiple classifier systems for face recognition,” In Proceedings of the Sixth International Workshop on Multiple Classifier Systems, Springer, Volume 3541/2005, pp. 982-984, 2005.

[15] Cynthia Krieger, “Neural Networks in Data Mining”, CiteSeerX, 1996. [Online].Available: http://www.cs.uml.edu/~ckrieger/user/Neural_Networks.pdf.

[16] Dan Zhu, “A hybrid approach for efficient ensembles,” Science Direct Decision support systems, 48(1), pp. 480-487, 2010.

[17] T.G. Dietterich, “Ensemble methods in machine learning,” In Proceedings of First international workshop on multiple classifier systems, pp. 1–15, Cagliari, Italy, 2000.

[18] T.G. Dietterich, “An experimental comparison of three methods for constructing ensembles of decision trees: Bagging, boosting, and randomization”, Springer Machine Learning, 40(2), pp. 139-157, 2000.

[19] S. Dudoit, J. Fridlyand, and T.P. Speed, “Comparison of discrimination methods for the classification of tumours using gene expression data,” Journal of the American Statistical Association, 97(457), pp. 77–87, 2002.

[20] Elizabeth Murray, “Using Decision Trees to Understand Student Data,” In Proceedings of the 22nd ICML, Bonn, Germany, 2005. [Online].Available:http://www.cs.ou.edu/~amy/courses/cs5973_fall2005/Murray_Final_Paper.pdf.

[21] Eunju Kim, Wooju Kim, Yillbyung Lee, “Combination of multiple classifiers for the customer’s purchase behaviour prediction,” Elsevier Decision Support Systems, 34(1), pp. 167– 175, 2002.

[22] U. Fayyad, R Uthurusamy, “From Data Mining to Knowledge Discovery in Databases,” AI Magazine, 17(3), pp. 37-54, 1996.

[23] Y. Freund, R. E. Schapire, “Experiments with a new boosting algorithm,” In Proceedings of 13th International Conference on Machine Learning, pp. 148-156, Bari, Italy, 1996.

[24] M.Gashler, C. Giraud-Carrier, T. Martinez, “Decision tree ensemble: small heterogeneous is better than large homogeneous,” In Proceedings of the Seventh International Conference on Machine Learning and Applications, pp. 900-905, San Diego, California, 2008.

[25] G. Giacinto and F. Roli, " Ensembles of neural network classifiers for remote sensing images", In Proceedings of the European Symposium on Intelligent Techniques, pp. 166-170, Bari, Italy, 1997.

[26] P.A. Gislason, J.A. Benediktsson, J.R. Sveinsson, “Decision fusion for the classification of urban remote sensing images,” Pattern Recognition Letters, 27 (1), pp. 294–300, 2006.

[27] M.A. Hall, “Correlation-based feature selection for machine learning,” PhD thesis, Department of Computer Science, University of Waikato, Hamilto, New Zealand, 1998.

[28] Hongjun Lu, Rudy Setiono, Huan Liu, “Effective Data Mining Using Neural Networks,” IEEE TKDE, 8(6), pp. 957-961, 1996.

[29] Jefery D Scargles, “Studies in Astronomical time series analysis,” The astrophysical journal 263(1), pp. 835-853, 1982.

[30] Jinlong Wang, Shunyao Wu, Yang Jiao, Huy Quan Vu, "Study on Student Score Based on Data Mining,” JCIT, 5(6), pp. 171-179, 2010.

[31] R. Kohavi, F. Provost, Editorial for the Special Issue on application of machine learning and the knowledge of discovery process, Springer Machine Learning 30(1), pp. 271-274, 1998.

[32] M. Kamber, J. Han, “Data mining: concepts and techniques” Maurgan Kaufmann, 2001.

[33] J. Li , H. Su, H. Chen, B. Futscher, "Optimal Search-Based Gene Subset Selection for Gene array Cancer Classification," IEEE Transactions on information technology in biomedicine, 11(4), pp. 398-405, 2007

[34] J. Li, H. Liu, S.-K. Ng, and L.Wong, “Discovery of significant rules for classifying cancer diagnosis data,” Journal of Bioinformatics, 19(2), pp. 93–102, 2003.

[35] J. Li, H. Liu, “Ensembles of cascading trees,” In Proceedings of 3rd IEEE International Conference on Data Mining, pp. 585–588, Florida, USA, 2003.

[36] G. Magdalena, Tadeusz Lasota, Bogdan Trawiński and Krzysztof Trawiński, “Comparison of Bagging, Boosting and stacking ensembles Applied to Real Estate Appraisal,” Intelligent Information and Database Systems , Lecture Notes in Computer Science, 5991/2010, pp. 340-350, 2010.

[37] T. Mitchell, M. Palatucci, "Classification in Very High Dimensional Problems with Handfuls of Examples," Principles and Practice of Knowledge Discovery in Databases (ECML/PKDD), Springer, September, 2007.

[38] Nikunj.C.Oza, Kagan Tumer, “Classifier ensembles: Select real-world applications,” Science Direct Information Fusion, 9(1), pp. 4-20, 2008.

[39] D. Optiz , R. Maclin, “Popular ensemble methods: an empirical study,” Journal of Artificial Intelligence Research, 11(1), pp. 169–198, 1999 .

[40] Oyelade, Oladipupo, Olufunke. “Knowledge Discovery from students result repository: Association Rules mining approach,” CSC- IJCSS, 4(2), pp. 199-207, 2010.

[41] N. Poh, S. Bengio, “An investigation of f-ratio client-dependent normalisation on biometric authentication tasks,” In Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing, pp. 721–724, Philadelphia, USA, 2005.

[42] J.R. Quinlan, “Induction of decision trees,” Springer Machine Learning, 1(1), pp.81–106, 1986.

[43] J.R. Quinlan, "Bagging, boosting, and C4.5," In Proceedings of 13th National Conference on Artificial Intelligence, pp. 725-730, 1996.

[44] J.R. Quinlan, "C4.5: program for machine learning," Morgan Kaufmann, 1993.

[45] R. Sandy, "Statistics for Business and Economics," McGraw-Hill, 1989.

[46] R.E. Schapire, "The boosting approach to machine learning: An overview," In Proceedings of MSRI Workshop on Nonlinear Estimation and Classification, 2002. [Online].Available: http://www.cs.princeton.edu/courses/archive/spring10/cos424/papers/Schapire2003.pdf.

[47] R.E. Schapire, Yoav Freund, "A short introduction to boosting," Journal of Japanese Society for Artificial Intelligence, 14(5), pp. 771-780, September 1999.

[48] R.E. Schapire, "Theoretical views of boosting**",** In Proceedings of Computational Learning Theory: Fourth European Conference, pp. 1-10, Nordkirchen, Germany, 1999.

[49] R.E. Schapire, Peter Stone, David McAllester, Michael L. Littman, A. János. Csirik, "Modelling auction price uncertainty using boosting-based conditional density estimation**,"** In Proceedings of the Nineteenth International Conference on machine learning, pp. 546-553, Sydney, Australia, 2002.

[50] Sholomo Hershkop, Salvatore J.Stolfo, "Combining Email models for False Positive Reduction", In Proceedings of ACM international conference on knowledge discovery in data bases, pp. 98-107, Chicago, USA, 2005.

[51] Shuxiang Xu, "Data Mining Using Higher Order Neural Network Models With Adaptive Neuron Activation Functions," IJACT, 2(4), pp. 168-177, 2010.

[52] M.M.S Silva, F.E.B Otero, A A Freitas, J C NIevola, "Genetic Programming for Attribute Construction in Data Mining," Springer Lecture Notes in Computer Science, Volume 2610/2003, pp. 384-393, 2003.

[53] Sreerama K. Murthy, "Automatic Construction of Decision Trees from Data: A Multi-Disciplinary Survey," Springer Data Mining and Knowledge Discovery, 2(4), pp. 345-389, 1998.

[54] C.J. Thornton, "Techniques in Computational Learning", Chapman and Hall, 1992.

[55] Todorovski, Sašo Džeroski, "Combining multiple models with meta decision trees," Principles of Data Mining and Knowledge Discovery, Lecture Notes in Computer Science, Volume 1910/2000, pp. 69-84, 2000.

[56] P.E. Utgoff and C.E. Brodley, "Linear machine decision trees," Technical report, Department of Computer Science, University of Massachusetts, Amherst, MA., 1991.

[57] L. Wang, T. Z. Sui, "Application of Data Mining Technology Based on Neural Network in the Engineering," In the Proceedings of International Conference WiCom 2007, pp. 5544 - 5547, Shanghai, China, 2007.

[58] C.P. Wei, H.C. Chen, T.H. Cheng, "Effective spam filtering: single-class learning and ensemble approach," Science Direct Decision Support Systems, 45 (3), pp. 491–503, 2008.

[59] C.P. Wei, H. Chang, Ci-Wei Lan and T. H. Cheng, "Cost sensitive learning for recurrence prediction of breast cancer," In Proceedings of the 14th Pacific Asia Conference on Information Systems, Taipei, Taiwan, 2010.
[Online].Available: http://www.pacis-net.org/file/2010/S28-01.pdf.

[60] C.P. Wei, O. R. Liu Sheng, P. Hu and N. Chang, "Automated Learning of Patient Image Retrieval Knowledge: Neural Networks versus Inductive Decision Trees," Science Direct Decision Support Systems, 30(2), pp. 105-124, 2000.

[61] C.P Wei , O. R. Liu Sheng, P. Hu, "Neural Network Learning for Intelligent Patient Image Retrievals," *IEEE Intelligent Systems"*, 13(1), pp. 49-57, 1998.

[62] I.J Witten, E. Frank. "Data Mining: Practical Machine Learning Tools and Techniques with Java Implementation," Morgan Kaufmann, 2000.

[63] I.J Witten, E. Frank, "Data Mining: Practical Machine Learning Tools and Techniques," Elsevier, 2011.

[64] I.H. Witten, E. Frank, M. Hall, L. Trigg, G. Holmes, "Data mining in bioinformatics using Weka," Oxford University Press Bioinformatics, 20(15), pp. 2479-2481, 2004.

[65] E. Wolfers, Zitzewitz, "Prediction Markets," SIEPR discussion paper, 2004, [Online].Available: http://wwwsiepr.stanford.edu/repec/sip/03-025.pdf.

[66] J. Wang, Ke Gao, Yang Jiao and Gang Li, "Study on Ensemble Classification Methods towards Spam Filtering" In Proceedings of ADMA, pp. 314-325, Beijing, China , 2009.

[67] Yoav Freund, Robert E. Schapire, “Large Margin Classification using Perceptron Algorithm,” Springer Machine Learning, 37(3), pp. 277-296, 1999.

[68] Yong Shi, "A Dimension Reduction Approach Using Shrinking for Multi-Dimensional Data Analysis," IJIIP, 1(2), pp. 86-98, 2010.

[69] Zarita Zainuddin, Maragatham Kumar**, “**Radial Basis Function Neural Networks in Protein Sequence Classification*,”* Malaysian Journal of Mathematical Sciences, 2(2), pp. 185-194, 2008.

[70] Zarita Zainuddin, Chan Siow Cheng, Lye Weng Kit, “Prediction of B-Turns Using Global Adaptive Techniques from Multiple Alignments in Neural Networks,” Malaysian Journal of Mathematical Sciences, 2(2), pp. 185-194, 2008.

[71] B. Zenko, S. Dzeroski, “Is combining classifiers Better than selecting the best one”, Springer Machine learning, 54(3), pp. 255-273, 2004.

[72] Zhang Harry, Su Jiang, “ A fast decision tree learning algorithm”, In Proceedings of the 21st national conference on Artificial intelligence, pp. 500-505, Boston, USA, 2006.

PUBLICATIONS

A). Publications in International refereed journals.

1. Sudheep Elayidom, Sumam Mary Idicula, and Joseph Alexander “Design and performance analysis of data mining techniques based on decision trees and Naive Bayes classifier for employment chance prediction”, Journal of Convergence in Information Technology, 6 (5), pp. 89-98, 2011.
2. Sudheep Elayidom, Sumam Mary Idicula, and Joseph Alexander “A Generalized data mining framework for employment prediction problems”, International Journal of Computer Applications. 31(3), pp. 40-47, 2011.
3. Sudheep Elayidom, Sumam Mary Idicula, and Joseph Alexander “A Hybrid ensemble framework for employment prediction problems”, Bioinfo Advances in Computational Research Journal, 3 (1), pp. 25-30, 2011.
4. Sudheep Elayidom, Sumam Mary Idicula, and Joseph Alexander “Analysis and implementation of data mining techniques using Naive-Bayes classifier and neural networks” Global Journal of Computer Science and Technology, 10 (10), pp. 20-25, 2010.
5. Sudheep Elayidom, Sumam Mary Idicula, and Joseph Alexander” Applying statistical dependency analysis techniques in a data mining domain, CSC-International Journal of Data Engineering Journal, 1(2), pp. 14-24, 2010.
6. Sudheep Elayidom, Dr. Sumam Mary Idicula, and Joseph Alexander "Applying data mining using statistical techniques for career selection", International Journal of Recent Trends in Engineering-Issue on Computer Science, 1 (1), pp. 446-449, 2009.

B). Papers in International Conferences.

1. Sudheep Elayidom, Sumam Mary Idicula, and Joseph Alexander "Applying data mining techniques for placement chance prediction". In Proceedings of international conference on advances in computing, control and telecommunication technologies, pp.669-671, Kerala, India, 2009.
[Online].Available:http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber= 5375893 &tag=1.

2. Sudheep Elayidom, Sumam Mary Idicula, and Joseph Alexander "Comparison of data mining techniques based on decision trees and neural networks for placement chance prediction". In Proceedings of International conference on Computational Engineering Practices & Techniques, pp. 173-178, Kerala, India, 2010.

APPENDIX - A

INSTITUTE OF APPLIED MANPOWER RESEARCH
SECTOR A7, INSTITUTIONAL AREA, DELHI - 110 040
NATIONAL TECHNICAL MANPOWER INFORMATION SYSTEM
(Scheme Funded by the AICTE)

(GRADUATE FOLLOW - UP QUESTIONNAIRE)

Please tick relevant box by putting (✓) and write in words wherever line is provided.

Address of Nodal Centre: Nodal Centre (NTMIS)
Cochin University of Science and Technology, Kochi - 682 022.

Sl. No..... Year of Passing.....

I. IDENTIFICATION PARTICULARS

a) Name of the Candidate .....
Address .....
City .....District.....State.....
Telephone No. & STD Code (Res.).....Mob.....
Email .....

b) Name of the Institution.....
Address.....
City.....District*.....State.....

c) Result declared : Month..... Year .....

d) Type of Award : Diploma [ ] Post Diploma [ ] Degree [ ]
P.G. Diploma [ ] P.G. Degree [ ] Ph.D. [ ]

i) Speciality and branch of study.....

ii) Division/grade .....Aggregate percentage of total marks.....

e) Nature of the Course : Full time [ ] Part Time [ ]

II. GENERAL PARTICULARS

1. Date of Birth : dd/mm/yyyy [.....] / [.....] [.....]
2. Sex : Male [ ] Female [ ]
3. Whether you belong to any reserved category : SC [ ] ST [ ] OBC [ ]
4. Name of the State you belong to .....
5. Whether you belong to : Rural Area [ ] Urban Area [ ]
6. Are you registered with Employment Exchange / University Employment Bureau ? Yes [ ] No [ ]
7. Have you undergone any training after completing your diploma / degree ? Yes [ ] No [ ]
   If yes, please indicate the following

Type of Training	Duration (months)
Apprenticeship Training	.....
Company Training	.....
Other (Specify)	.....

8. Have you taken any further technical education after completing your diploma / degree ? Yes [ ] No [ ]
   If Yes, Please specify : Post Diploma [ ] Degree [ ] P.G. Diploma [ ]
   P.G. Degree [ ] Ph.D. [ ]
9. GATE Score if any..... 10. Annual family income Rs.....

* ENTRANCE RANK WAS COLLECTED SEPARATELY FROM OFFICE OF ENTRANCE COMMISSIONER

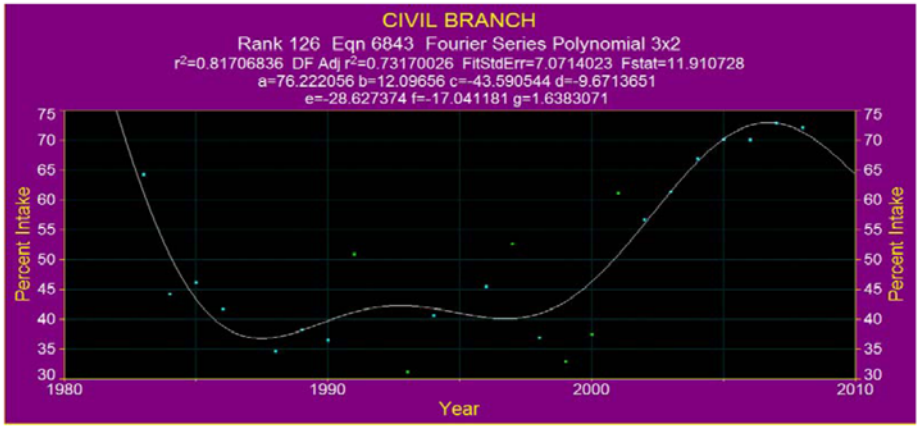
APPENDIX - B

Absorption rate prediction for civil engineering branch

Fourier Series Polynomial
y=a+bcos(x)+csin(x)+dcos(2x)+esin(2x)+fcos(3x)+gsin(3x)

X Value	Y Value	Y Predict
1983	64.4	60.82868
1984	44.3	51.46608
1985	46.3	44.19309
1986	41.8	39.36494
1988	34.7	36.58764
1989	38.3	37.65014
1990	36.6	39.41071
1991	51	41.1713
1993	31.3	42.80425
1994	40.7	42.40597
1996	45.6	40.51098
1997	52.8	40.04931
1998	37	40.60972
1999	33	42.53941
2000	37.5	45.9358
2001	61.3	50.61192
2002	56.8	56.11931
2003	61.5	61.82474
2004	67	67.02663
2005	70.3	71.08938
2006	70.2	73.56995
2007	73	74.31222
2008	72.2	73.49003

a	76.22205561
b	12.09656033
c	-43.5905437
d	-9.67136511
e	-28.6273743
f	-17.0411808
g	1.638307072



Absorption rate prediction for Computer science branch

Polynomial equation $y=a+bx+cx^2+dx^3+ex^4+fx^5+gx^6+hx^7+ix^8+jx^9+kx^{10}$

X Value	Y value	Y predict
1988	67.5	66.59
1989	68.8	70.8
1990	75.9	74.7
1991	79.1	78.2
1993	81.3	83.3
1994	82.4	84.8
1996	93.9	85.8
1997	85.5	85.2
1998	81.2	84.1
1999	77.6	82.7
2000	83.4	81.0
2001	80.1	79.3
2002	77.8	77.8
2003	77	76.9
2004	77	76.9
2005	78.3	78.1
2006	81.2	81.1
2007	86.2	86.1
2008	93.7	93.8

a	8.38E+08
b	4439.620063
c	-679.78598
d	0.019142818
e	0.000111911
f	3.43E-08
g	-6.35E-12
h	5.82E-16
i	-2.96E-18
j	-2.16E-21
k	9.98E-25