M. S. RAMAIAH INSTITUTE OF TECHNOLOGY

Data Mining in Credit Scoring

By

Swagato Majumder (1MS14MCA49)

Taher Fakhruddin Makadam (1MS14MCA50)

Tasneem Khanum (1MS14MCA51)

NOVEMBER 2015

M S Ramaiah Institute of Technology
(Autonomous Institute Affiliated to VTU)
Bangalore – 54

Department of Master of Computer Application

M S Ramaiah Institute of Technology

(Autonomous Institute Affiliated to VTU)

CERTIFICATE

This is to certify that the project entitled **“ Data Mining in Credit Scoring”** has been completed by SWAGATO MAJUMDER, TAHER FAKHRUDDIN MAKADAM and TASNEEM KHANUM in 3rd semester of the degree of MCA – 2015 Examination, under our supervision and guidance.

(Assistant professor)

Dr.Manish Kumar

Acknowledgement

It gives us pleasure to present our project on **“Data Mining in Credit Scoring”.** The able guidance of our teaching staff department made this study possible. They have been a constant source of encouragement throughout the completion of this project.

We would sincerely like to thank **Mrs. Sailaja Kumar and Mr.Manish Kumar** for her help & support during the making of this project report. This report would not have been successful without the immense guidance from our guide & the valuable time that she has spent with us during our report development stages.

Swagato Majumder (1MS14MCA49)
Taher Fakhruddin Makadam (1MS14MCA50)
Tasneem Khanum (1MS14MCA51)

Contents

1. Abstract

A **credit score** is a numerical expression based on a level analysis of a person's credit files, to represent the creditworthiness of the person. A credit score is primarily based on credit report information typically sourced from credit bureaus.

Lenders, such as banks and credit card companies, use credit scores to evaluate the potential risk posed by lending money to consumers and to mitigate losses due to bad debt. Lenders use credit scores to determine who qualifies for a loan, at what interest rate, and what credit limits. Lenders also use credit scores to determine which customers are likely to bring in the most revenue. The use of credit or identity scoring prior to authorizing access or granting credit is an implementation of a trusted system.

Credit scoring is not limited to banks. Other organizations, such as mobile phone companies, insurance companies, landlords, and government departments employ the same techniques.

2. Introduction to Data Mining

2.1 Data Mining

Data mining field at the intersection of computer science and statistics, is the process that attempts to discover patterns in large data sets. It utilizes methods at the intersection of artificial intelligence, machine learning, statistics, and database systems.

The overall goal of the data mining process is to extract information from a data set and transform it into an understandable structure for further use. Aside from the raw analysis step, it involves database and data management aspects, data pre-processing, model and inference considerations, interestingness metrics, complexity considerations, post-processing of discovered structures, visualization, and online updating.

Data mining uses sophisticated mathematical algorithms to segment the data and evaluate the probability of future events. Data mining is also known as Knowledge Discovery in Data (KDD).

- Data Mining applies many older computational techniques from statistics, machine learning and pattern recognition

- Extract, transform, and load transaction data onto the data warehouse system.

- Store and manage the data in a multidimensional database system.

- Provide data access to business analysts and information technology professionals.

- Analyze the data by application software.

- Present the data in a useful format, such as a graph or table.

- The ultimate goal of data mining is prediction - and predictive data mining is the most common type of data mining and one that has the most direct business applications.

2.2 Advantages of Data Mining

Data mining is an important part of knowledge discovery process that analyzes large enormous set of data and gives us unknown, hidden and useful information and knowledge. Data mining has not only applied effectively in business environment but also in other fields such as weather forecast, medicine, transportation, healthcare, insurance, government and etc. Data mining brings a lot of advantages when using in a specific industry. Besides those advantages, data mining also has its own disadvantages as well such as privacy, security and misuse of information. We will examine the advantage of data mining in different industries in a greater detail.

1) **Marking/Retailing:** Data mining helps marketing companies to build models based on historical data to predict who will respond to new marketing campaign such as direct mail, online marketing campaign and etc. Through this prediction, marketers can have appropriate approach to sell profitable products to targeted customers with high satisfaction. Data mining brings a lot of benefit s to retail company in the same way as marketing. Through market basket analysis, the store can have an appropriate production arrangement in the way that customers can buy frequent buying products together with pleasant. In addition, it also help the retail company offers a certain discount for particular products what will attract customers.

2) **Banking/Crediting:** Data mining can assist financial institutions in areas such as credit reporting and loan information. Data mining gives financial institutions information about loan information and credit reporting. By building a model from previous customer's data with common characteristics, the bank and financial can estimate what are the god and/or bad loans and its risk level. In addition, data mining can help banks to detect fraudulent credit card transaction to help credit card's owner prevent their losses. For example, by examining previous customers with similar attributes, a bank can estimated the level of risk associated with each given loan.

3) **Law enforcement**: Data mining can aid law enforcers in identifying criminal suspects as well as apprehending these criminals by examining trends in location, crime type, habit, and other patterns of behaviors.

4) **Researchers:** Data mining can assist researchers by speeding up their data analyzing process; thus, allowing those more time to work on other projects.

5) **Manufacturing:** By applying data mining in operational engineering data, manufacturers can detect faulty equipment's and determine optimal control parameters. For example semi-conductor manufacturers had a challenge that even the conditions of manufacturing environments at different wafer production plants are similar, the quality of wafer are lot the same and some for unknown reasons even contain defects. Data mining has been applied to determine the ranges of control parameters that lead to the production of golden wafer. Then those optimal control parameters are used to manufacture wafers with desired quality.

6) **Customer Relationship Management:** A new business culture is developing today. Within it, the economics of customer relationships are changing in fundamental ways, and companies are facing the need to implement new solutions and strategies that address these changes. The concepts of mass production and mass marketing, first created during the Industrial Revolution, are being supplanted by new ideas in which customer relationships are the central business issue. Firms today are concerned with increasing customer value through analysis of the customer lifecycle. The tools and technologies of data warehousing, data mining, and other customer relationship management (CRM) techniques afford new opportunities for businesses to act on the concepts of relationship marketing.

7) **Governments:** Data mining helps government agency by digging and analysing records of financial transaction to build patterns that can detect money laundering or criminal activity.

2.3 Techniques used in Data Mining

Various algorithms and techniques like Classification, Clustering, Regression, Artificial Intelligence, Neural Networks, Association Rules, Decision Trees, Genetic Algorithm, Nearest Neighbor method etc., are used for knowledge discovery from databases.

2.3.1 Classification

Classification is the most commonly applied data mining technique, which employs a set of pre-classified examples to develop a model that can classify the population of records at large. Fraud detection and credit risk applications are particularly well suited to this type of analysis. This approach frequently employs decision tree or neural network-based classification algorithms. The data classification process involves learning and classification. In Learning the training data are analyzed by classification algorithm.

Types of classification models:

J48 Decision Tree Induction Algorithm

The J48 algorithm gives several options related to tree pruning. Many algorithms attempt to "prune", or simplify, their results. Pruning produces fewer, more easily interpreted results. More importantly, pruning can be used as a tool to correct for potential over fitting. The basic algorithm described above recursively classifies until each leaf is pure, meaning that the data has been categorized as close to perfectly as possible. This process ensures maximum accuracy on the training data, but it may create excessive rules that only describe particular idiosyncrasies of that data. When tested on new data, the rules may be less effective. Pruning always reduces the accuracy of a model on training data. This is because pruning employs various means to relax the specificity of the decision tree, hopefully improving its performance on test data. The overall concept is to gradually generalize a decision tree until it gains a balance of flexibility and accuracy.

J48 employs two pruning methods. The first is known as subtree replacement. This means that nodes in a decision tree may be replaced with a leaf -- basically reducing the number of tests along a certain path. This process starts from the leaves of the fully formed tree, and works backwards toward the root. The second type of pruning used in J48 is termed subtree rising.

Rule-based Classification

The rule-based classifiers learned model is represented as a set of IF-THEN rules. An IF-THEN rule is an expression of the form:

IF condition THEN conclusion.

A rule R can be assessed by its coverage and accuracy. Given a tuple, X, from a class labeled data set, D, let n covers be the number of tuples covered by R; n correct be the number of tuples correctly classified by R; and |D| be the number of tuples in D. We can define coverage accuracy R as

$$coverage(R) = \frac{n_{covers}}{|D|}$$

$$accuracy(R) = \frac{n_{correct}}{n_{covers}}.$$

We can build a rule-based classifier by extracting IF-THEN rules from a decision tree. To extract rules from a decision tree, one rule is created for each path from the root to a leaf node. Each splitting criterion along a given path is logically ANDed to form the rule antecedent (“IF” part). The leaf node holds the class prediction, forming the rule consequent (“THEN” part). Then, the rule set should be pruned. There are assorted methods to do this.

Bayesian Classification

Bayesian classifiers are statistical classifiers. They can predict class membership probabilities, such as the probability that a given tuple belongs to a particular Class. Bayesian classification is based on Bayes theorem, Studies comparing classification algorithms have found a simple Bayesian classifier known as the naive Bayesian classifier to be comparable in performance with decision tree and selected neural network classifiers. Bayesian classifiers have also exhibited high accuracy and speed when applied to large databases.

Naïve Bayesian classifiers assume that the effect of an attribute value on a given class is independent of the values of the other attributes. This assumption is called class conditional independence. It is made to simplify the computations involved and, in this sense, is considered “naïve.” Bayesian belief networks are graphical models, which unlike naïve Bayesian classifiers allow the representation of dependencies among subsets of attributes. Bayesian belief networks can also be used for classification.

So here the core formulation is

$$
\begin{aligned}
P(\mathbf{X}|C_i) &= \prod_{k=1}^{n} P(x_k|C_i) \\
&= P(x_1|C_i) \times P(x_2|C_i) \times \cdots \times P(x_n|C_i).
\end{aligned}
$$

Nearest-Neighbour Classifiers

One popular example method is k-Nearest-Neighbour Classifiers. Nearest-neighbour classifiers are based on learning by analogy, that is, by comparing a given test tuple with training tuples that are similar to it. The training tuples are described by n attributes. Each tuple represents a point in an n-dimensional space. In this way, all of the training tuples are stored in an n-dimensional pattern space. When given an unknown tuple, a k-nearest-neighbour classifier searches the pattern space for the k training tuples that are closest to the unknown tuple. These k training tuples are the k “nearest neighbours” of the unknown tuple. The unknown tuple is assigned the most common class among its k nearest neighbours.

Nearest-neighbour classifiers can also be used for prediction, that is, to return a real-valued prediction for a given unknown tuple. In this case, the classifier returns the average value of the real-valued labels associated with the k nearest neighbours of the unknown tuple.

Nearest-neighbour classifiers use distance-based comparisons that intrinsically assign equal weight to each attribute. They therefore can suffer from poor accuracy when given noisy or irrelevant attributes. The method, however, has been modified to incorporate attribute weighting and the pruning of noisy data tuples. The choice of a distance metric can be critical. The Manhattan (city block) distance or other distance measurements, may also be used. Nearest-neighbour classifiers can be extremely slow when classifying test tuples.

Artificial Neural Network

The word network in the term 'artificial neural network' refers to the inter–connections between the neurons in the different layers of each system. An example system has three layers. The first layer has input neurons, which send data via synapses to the second layer of neurons, and then via more synapses to the third layer of output neurons. More complex systems will have more layers of neurons with some having increased layers of input neurons and output neurons. The synapses store parameters called "weights" that manipulate the data in the calculations.

An ANN is typically defined by three types of parameters:

1. The interconnection pattern between different layers of neurons

2. The learning process for updating the weights of the interconnections

3. The activation function that converts a neuron's weighted input to its output activation.

4. Training a neural network model essentially means selecting one model from the set of allowed models (or, in a Bayesian framework, determining a distribution over the set of allowed models) that minimizes the cost criterion. There are numerous algorithms available for training neural network models; most of them can be viewed as a straightforward application of optimization theory and statistical estimation.

5. Most of the algorithms used in training artificial neural networks employ some form of gradient descent. This is done by simply taking the derivative of the cost function with respect to the network parameters and then changing those parameters in a gradient-related direction.

2.3.2 Association Analysis

In data mining, association rule learning is a popular and well researched method for discovering interesting relations between variables in large databases. It is intended to identify strong rules discovered in databases using different measures of interestingness. Based on the concept of strong rules,. introduced association rules for discovering regularities between products in large-scale transaction data recorded by point-of-sale (POS) systems in supermarkets. For example, the rule $\{\mathrm{onions}, \mathrm{potatoes}\} \Rightarrow \{\mathrm{burger}\}$ found in the sales data of a supermarket would indicate that if a customer buys onions and potatoes together, he or she is likely to also buy hamburger meat. Such information can be used as the basis for decisions about marketing activities such as, e.g., promotional pricing or product placements. In addition to the above example from market basket analysis association rules are employed today in many application areas including Web usage mining, intrusion detection and bioinformatics. As opposed to sequence mining, association rule learning typically does not consider the order of items either within a transaction or across transactions.

Association rules are usually required to satisfy a user-specified minimum support and a user-specified minimum confidence at the same time. Association rule generation is usually split up into two separate steps:

1. First, minimum support is applied to find all frequent item sets in a database.
2. Second, these frequent item sets and the minimum confidence constraint are used to form rules.

While the second step is straightforward, the first step needs more attention.

Many algorithms for generating association rules were presented over time.

Some well-known algorithms are Apriori, Eclat and FP-Growth, but they only do half the job, since they are algorithms for mining frequent item sets. Another step needs to be done after to generate rules from frequent item sets found in a database.

2.3.3 Cluster analysis

Cluster analysis or clustering is the task of grouping a set of objects in such a way that objects in the same group (called cluster) are more similar (in some sense or another) to each other than to those in other groups (clusters). It is a main task of explorative data mining, and a common technique for statistical data analysis used in many fields, including machine learning, pattern recognition, image analysis, information retrieval, and bioinformatics.

Cluster analysis itself is not one specific algorithm, but the general task to be solved. It can be achieved by various algorithms that differ significantly in their notion of what constitutes a cluster and how to efficiently find them. Popular notions of clusters include groups with low distances among the cluster members, dense areas of the data space, intervals or particular statistical distributions. Clustering can therefore be formulated as a multi-objective optimization problem. The appropriate clustering algorithm and parameter settings (including values such as the distance function to use, a density threshold or the number of expected clusters) depend on the individual data set and intended use of the results.

Besides the term clustering, there are a number of terms with similar meanings, including automatic classification, numerical
taxonomy, botryology and typological analysis. The subtle differences are often in the usage of the results: while in data mining, the resulting groups are the matter of interest, in automatic classification primarily their discriminative power is of interest.

Clustering algorithms can be categorized based on their cluster model, as listed

- Connectivity based clustering (hierarchical clustering)
- k-means clustering
- Distribution-based clustering

2.4 Tools for Data Mining

2.4.1 ORANGE:

Orange is a component-based data mining and machine learning software suite, featuring a visual programming front-end for explorative data analysis and visualization, and Python bindings and libraries for scripting. It includes a set of components for data pre-processing, feature scoring and filtering, modelling, model evaluation, and exploration techniques. It is implemented in C++ and Python. Its graphical user interface builds upon the cross-platform Qt framework. Orange is distributed free under the GPL. It is maintained and developed at the Bioinformatics Laboratory of the Faculty of Computer and Information Science, University of Ljubljana, Slovenia.

2.4.2 RAPID MINER:

Rapid Miner provides data mining and machine learning procedures including: data loading and transformation, data pre-processing and visualization, modelling, evaluation, and deployment. Rapid Miner is written in the Java programming language. It uses learning schemes and attribute evaluators from the Weka machine learning environment and statistical modelling schemes.

Rapid Miner provides a GUI to design an analytical pipeline (the "operator tree"). The GUI generates an XML (eXtensible Markup Language) file that defines the analytical processes the user wishes to apply to the data. Alternatively, the engine can be called from other programs or used as an API.

2.4.3 JHELP WORK:

It is create as an attempt to make a data analysis environment using open source packages with a comprehensible user interface and to create a tool competitive to commercial program.

2.4.4 KNIME:

KNIME allows users to visually create data flows (or pipelines), selectively execute some or all analysis steps, and later inspect the results, models, and interactive views. KNIME is written in Java and based on Eclipse and makes use of its extension mechanism to add plugins providing additional functionality.

1. KNIMEs core-architecture allows processing of large data volumes that are only limited by the available hard disk space (most other open source data analysis tools are working in main memory and are therefore limited to the available RAM). E.g. KNIME allows analysis of 300 million customer addresses, 20 million cell images and 10 million molecular structures.

2. Additional plugins allows the integration of methods for Text Mining, Image Mining, as well as time series analysis.

3. KNIME integrates various other Open-Source-projects, e.g. machine learning algorithms from Weka, the statistics package R, as well as LibSVM, JFreeChart, ImageJ, and the Chemistry Development Kit.

2.4.5 RATTLE:

Rattle provides considerable data mining functionality by exposing the power of the R Statistical Software through a graphical user interface. Rattle is also used as a teaching facility to learn the R software Language. There is a Log Code tab, which replicates the R code for any activity undertaken in the GUI, which can be copied and pasted. Rattle can be used for statistical analysis, or model generation. Rattle allows for the dataset to be partitioned into training, validation and testing. The dataset can be viewed and edited. There is also an option for scoring an external data file.

2.4.6 WEKA:

The Weka workbench contains a collection of visualization tools and algorithms for data analysis and predictive modelling, together with graphical user interfaces for easy access to this functionality. The original non-Java version of Weka was a TCL/TK front-end to (mostly third-party) modelling algorithms implemented in other programming languages, plus data pre-processing utilities in C, and aMakefile-based system for running machine learning experiments.

Advantages of Weka include:

- Free availability under the GNU General Public License
- Portability, since it is fully implemented in the Java programming language and thus runs on almost any modern computing platform
- A comprehensive collection of data preprocessing and modeling techniques
- Ease of use due to its graphical user interfaces

2.5 WEKA Tool Description

2.5.1 Basic Concepts

WEKA, formally called Waikato Environment for Knowledge Learning, is a computer program that was developed at the University of Waikato in New Zealand for the purpose of identifying information from raw data gathered from agricultural domains. WEKA supports many different standard data mining tasks such as data pre-processing, classification, clustering, regression, visualization and feature selection. The basic premise of the application is to utilize a computer application that can be trained to perform machine learning capabilities and derive useful information in the form of trends and patterns. WEKA is an open source application that is freely available under the GNU general public license agreement. Originally written in C the WEKA application has been completely rewritten in Java and is compatible with almost every computing platform.

It is user friendly with a graphical interface that allows for quick set up and Operation. WEKA operates on the predication that the user data is available as a flat file or relation, this means that each data object is described by a fixed number of attributes that usually are of a specific type, normal alpha-numeric or numeric values. The WEKA application allows novice users a tool to

Identify hidden information from database and file systems with simple to use options and visual interfaces.

The GUI Chooser consists of four buttons—one for each of the four major Weka applications—and four menus.

The buttons can be used to start the following applications:

- **Explorer** An environment for exploring data with WEKA (the rest of this documentation deals with this application in more detail).

- **Experimenter** An environment for performing experiments and conducting statistical tests between learning schemes.

- **KnowledgeFlow** This environment supports essentially the same functions as the Explorer but with a drag-and-drop interface. One advantage is that it supports incremental learning.

- **SimpleCLI** Provides a simple command-line interface that allows direct execution of WEKA commands for operating systems that do not provide their own command line interface.

At the very top of the window, just below the title bar, is a row of tabs. When the Explorer is first started only the first tab is active; the others are greyed out. This is because it is necessary to open (and potentially pre-process) a data set before starting to explore the data.

The tabs are as follows:

1. **Pre-process:** Choose and modify the data being acted on.

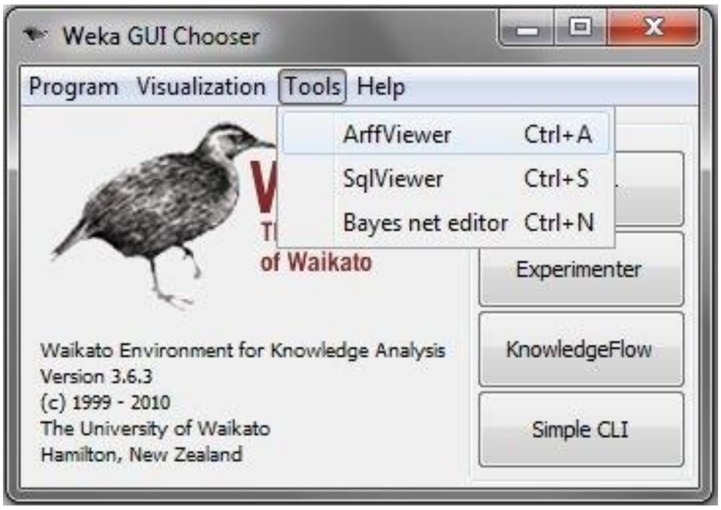
2. **Classify:** Train and test learning schemes that classify or perform regression.

3. **Cluster**: Learn clusters for the data.

4. **Associate**: Learn association rules for the data.

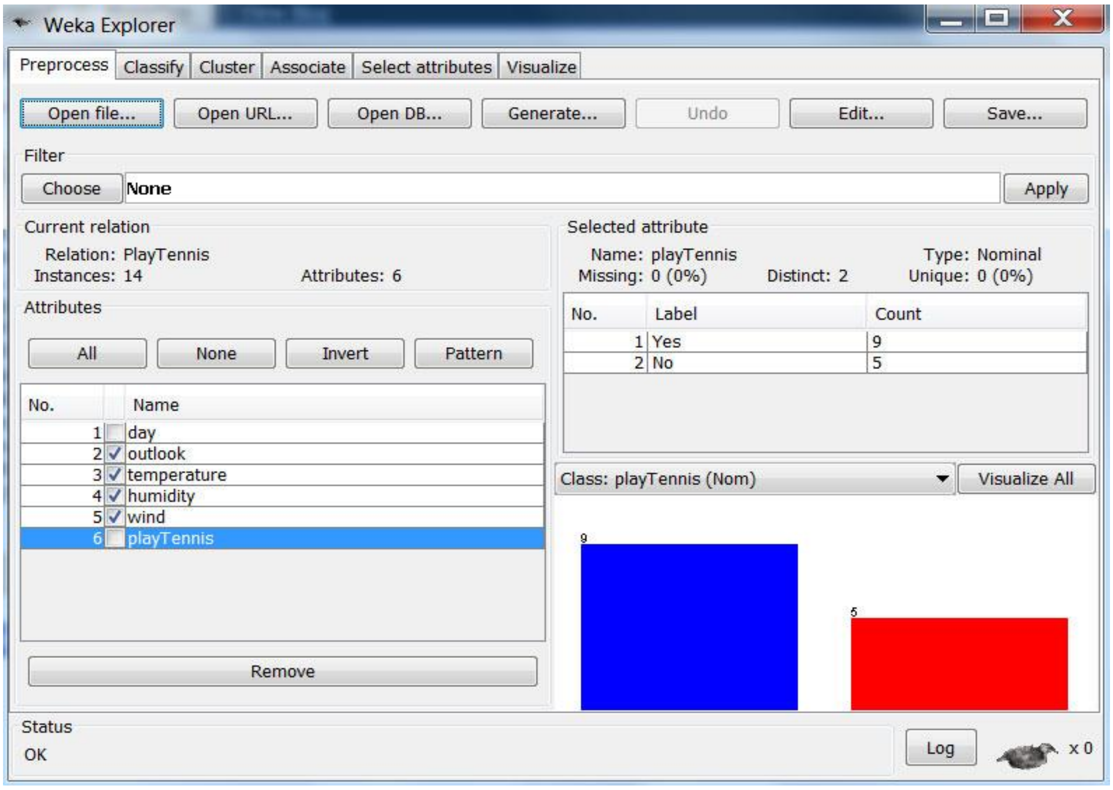
5. **Select attributes**: Select the most relevant attributes in the data.

6. **Visualize**: View an interactive 2D plot of the data.



Waikato Environment for Knowledge Analysis
Version 3.6.3
(c) 1999 - 2010
The University of Waikato
Hamilton, New Zealand

Fig 2.5.1(a): Weka GUI Chooser



Filter: None

Relation: PlayTennis
Instances: 14 Attributes: 6

Name: playTennis Type: Nominal
Missing: 0 (0%) Distinct: 2 Unique: 0 (0%)

No.		Name
1	□	day
2	☑	outlook
3	☑	temperature
4	☑	humidity
5	☑	wind
6	□	playTennis

No.	Label	Count
1	Yes	9
2	No	5

Class: playTennis (Nom)

Status
OK

Fig 2.5.1(b): Weka Explorer

2.5.2 Dataset

A set of data items, the dataset, is a very basic concept of machine learning. A Dataset is roughly equivalent to a two-dimensional spreadsheet or database table. In WEKA, it is implemented by the weka.core.Instances class. A dataset is a collection of examples, each one of class weka.core.Instance. Each Instance consists of a number of attributes, any of which can be nominal (= one of a predefined list of values), numeric (= a real or integer number) or a string (= an arbitrary long list of characters, enclosed in ”double quotes”).

2.5.3 Classifier

Any learning algorithm in WEKA is derived from the abstract weka.classifiers.Classifierclass. Surprisingly little is needed for a basic classifier: a routine which generates a classifier model from a training dataset (= buildClassifier) andanother routine which evaluates the generated model on an unseen test dataset(= classify Instance), or generates a probability distribution for all classes(= distributionForInstance).

A classifier model is an arbitrary complex mapping from all-but-one dataset Attributes to the class attribute. The specific form and creation of this mapping, or model, differs from classifier to classifier. For example, ZeroR’s (= weka.classifiers.rules.ZeroR) model just consists of a single value: the Most common class, or the median of all numeric values in case of predicting a Numeric value (= regression learning). ZeroR is a trivial classifier, but it gives a Lower bound on the performance of a given dataset which should be significantly improved by more complex classifiers. As such it is a reasonable test on how well the class can be predicted without considering the other attributes

The simplest case is using a training set and a test set which are mutually independent. This is referred to as hold-out estimate. To estimate variance in these performance estimates, hold-out estimates may be computed by repeatedly resampling the same dataset – i.e. randomly reordering it and then splitting it into training and test sets with a specific proportion of the examples, collecting all estimates on test data and computing average and standard deviation of accuracy.

3. Problem Description

3.1 Credit Scoring Data Set and Used Variables

Credit score is a number that banks use to determine whether you qualify for credit—and if so, how much interest they'll charge you. Insurance carriers and phone companies rely on the scores to decide if you're a good credit risk. A prospective boss or landlord may turn you down if your score doesn't measure up.

Your credit score represents your creditworthiness: how likely you will pay your bills and pay them on time. The Minneapolis-based Fair Isaac Corporation (better known as FICO) was the first to boil down your credit history

Attribute	Description
Age	Age of Customer.
Income per dependent	Income of the customer (Amount in thousands)
Monthly credit card exp	Monthly credit card expenditure (Amount in hundreds)
Own home	Whether the customer owns home (yes/no)
Self employed	Whether the customer is self-employed (yes/no)
Derogatory reports	No. of Derogatory reports (count)
Application accepted	Application accepted (yes/no)

This data set can be used for gathering useful results about the eligibility of the customer whether he/she can take credit or not. The class label signifies that a person's application is accepted or rejected.

As the above dataset attributes are dependent on each other we can use the classification technique to predict whether the customer is eligible for the credit or not.

3.2: Using Data Mining in data set

The WEKA(“Waikato Environment for Knowledge Analysis”) tool is used for data mining. Data mining fines valuable information hidden in large volume of data. Weka is a collection of machine learning algorithms for data mining tasks, written in java and it contains tools for data pre-processing, classification, regression, clustering, association rules, and visualization. The key features of weka are it is open source and platform independent. It provides many different algorithms for data mining and classification process. We have used 9 cross validation to minimize any bias in the process and improve the efficiency of the process.

3.3: Algorithm used for Analysis

Classification is the most commonly applied data mining technique, which employs a set of pre-classified examples to develop a model that can classify the population of records at large. Fraud detection and credit risk applications are particularly well suited to this type of analysis. This approach frequently employs decision tree or neural network-based classification algorithms. The data classification process involves learning and classification. In Learning the training data are analyzed by classification algorithm.

Types of classification models:

J48 Decision Tree Induction Algorithm

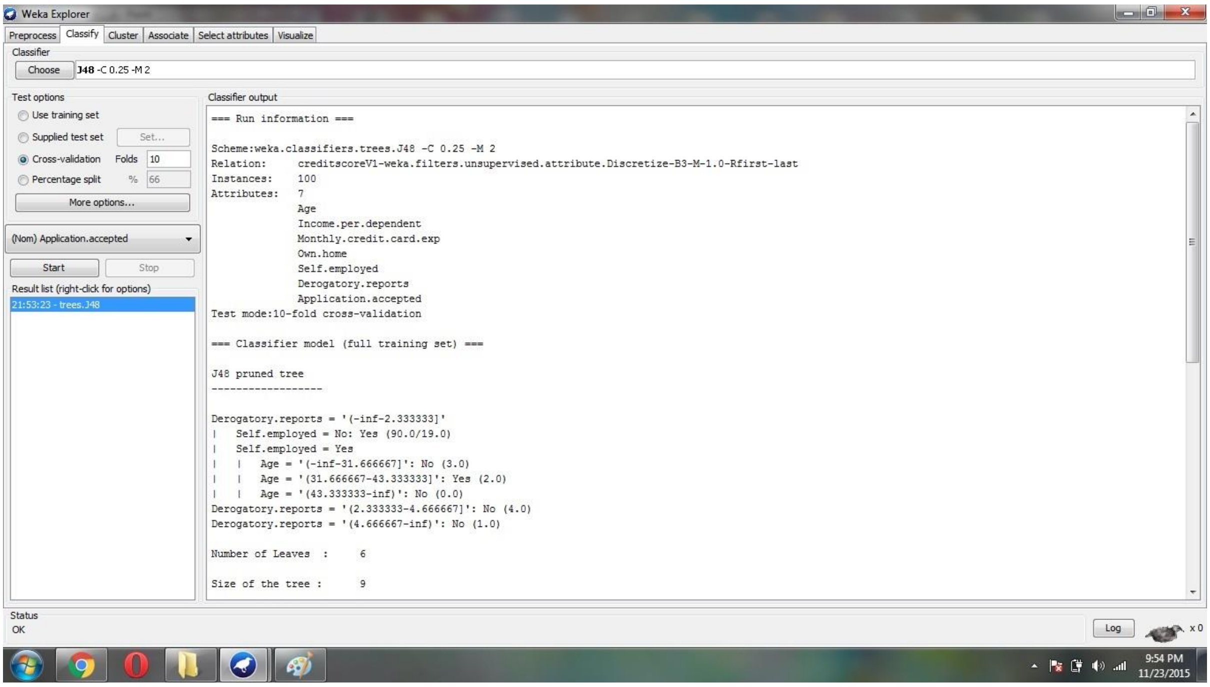
The J48 algorithm gives several options related to tree pruning. Many algorithms attempt to "prune", or simplify, their results. Pruning produces fewer, more easily interpreted results. More importantly, pruning can be used as a tool to correct for potential over fitting. The basic algorithm described above recursively classifies until each leaf is pure, meaning that the data has been categorized as close to perfectly as possible. This process ensures maximum accuracy on the training data, but it may create excessive rules that only describe particular idiosyncrasies of that data. When tested on new data, the rules may be less effective. Pruning always reduces the accuracy of a model on training data. This is because pruning employs various means to relax the specificity of the decision tree, hopefully improving its performance on test data. The overall concept is to gradually generalize a decision tree until it gains a balance of flexibility and accuracy.

J48 employs two pruning methods. The first is known as subtree replacement. This means that nodes in a decision tree may be replaced with a leaf -- basically

reducing the number of tests along a certain path. This process starts from the leaves of the fully formed tree, and works backwards toward the root. The second type of pruning used in J48 is termed subtree rising.

3.4: Results and Discussion

The result of our experiment is shown below:-



Classifier: J48 -C 0.25 -M 2

Test options

○ Use training set

○ Supplied test set

● Cross-validation Folds 10

○ Percentage split % 66

(Nom) Application.accepted

Result list (right-click for options)

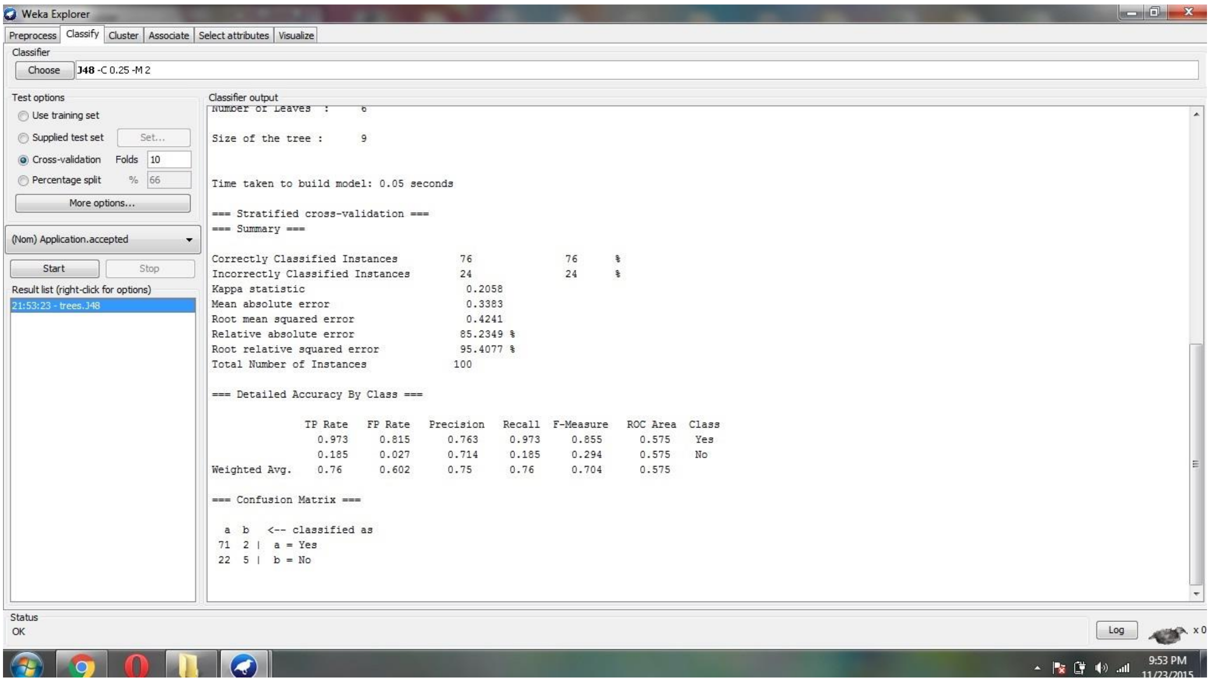
21:53:23 - trees.J48

Classifier output

```
=== Run information ===

Scheme:weka.classifiers.trees.J48 -C 0.25 -M 2
Relation:     creditscoreV1-weka.filters.unsupervised.attribute.Discretize-B3-M-1.0-Rfirst-last
Instances:    100
Attributes:   7
              Age
              Income.per.dependent
              Monthly.credit.card.exp
              Own.home
              Self.employed
              Derogatory.reports
              Application.accepted
Test mode:10-fold cross-validation

=== Classifier model (full training set) ===

J48 pruned tree
------------------

Derogatory.reports = '(-inf-2.333333]'
|   Self.employed = No: Yes (90.0/19.0)
|   Self.employed = Yes
|   |   Age = '(-inf-31.666667]': No (3.0)
|   |   Age = '(31.666667-43.333333]': Yes (2.0)
|   |   Age = '(43.333333-inf)': No (0.0)
Derogatory.reports = '(2.333333-4.666667]': No (4.0)
Derogatory.reports = '(4.666667-inf)': No (1.0)

Number of Leaves  :     6

Size of the tree :      9
```

Status: OK

Figure 1.1 Classifier Output



Classifier: J48 -C 0.25 -M 2

Test options

○ Use training set

○ Supplied test set

● Cross-validation Folds 10

○ Percentage split % 66

(Nom) Application.accepted

Result list (right-click for options)

21:53:23 - trees.J48

Classifier output

```
Number of Leaves  :     6

Size of the tree :      9


Time taken to build model: 0.05 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances          76               76      %
Incorrectly Classified Instances        24               24      %
Kappa statistic                          0.2058
Mean absolute error                      0.3383
Root mean squared error                  0.4241
Relative absolute error                 85.2349 %
Root relative squared error             95.4077 %
Total Number of Instances              100

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 0.973     0.815      0.763     0.973     0.855      0.575    Yes
                 0.185     0.027      0.714     0.185     0.294      0.575    No
Weighted Avg.    0.76      0.602      0.75      0.76      0.704      0.575

=== Confusion Matrix ===

  a  b   <-- classified as
 71  2 |  a = Yes
 22  5 |  b = No
```

Status

OK

Figure 1.2 Classifier Output

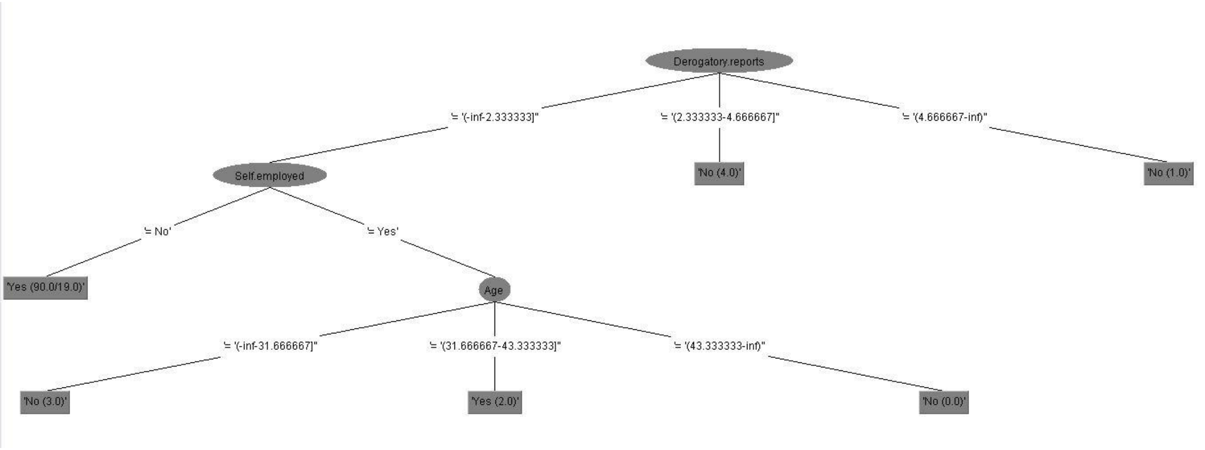


Figure 1.3 Tree Generate

4. Conclusion

From this report we conclude that the dataset “Credit Scoring” classifies the data set into whether the application of the customer is accepted or rejected based on Age, Income, Monthly Credit Card Expenditure, Own Home, Self Employed, and Derogatory Reports by using data mining classification technique. In our work we use the classification to enhance the prediction results of the given set of data .This data analysis results can be used for further research in enhancing the accuracy of the prediction system.

5. REFRENCES

1. "Data Mining Curriculum".

2. https://en.wikipedia.org/wiki/Credit_score.

3. http://facweb.cs.depaul.edu/mobasher/classes/ect584/WEKA/preprocess.html.

4. http://www.ijitee.org/attachments/File/v2i6/F0843052613.pdf

5. Introduction to Data Mining:-Pang-Ning Tan,Michael Steinbach,Vipin Kumar.

6. Clifton, Christopher (2010). "Encyclopædia Britannica: Definition of Data Mining".

7. Hastie, Trevor; Tibshirani, Robert; Friedman, Jerome (2009). "The Elements of Statistical Learning: Data Mining, Inference, and Prediction".