# Modelling and Forecasting of Aspects of Credit Card Defaults

**Vaishali Singh**

***A dissertation submitted for the partial fulfilment of BS-MS dual degree in Science***

**Indian Institute of Science Education and Research(IISER) Mohali**
**April 2019**

## Certificate

This is to certify that the dissertation titled **“Modeling and Forecasting of Aspects of Credit Card Defaults ”** submitted by **Ms. Vaishali Singh** (Reg. No. MS14139) for the partial fulfilment of BS-MS dual degree programme of the Institute, has been examined by the thesis committee duly appointed by the Institute. The committee finds the work done by the candidate satisfactory and recommends that the report be accepted.

Dr. Shane D’Mello
(Supervisor)

Prof. Kanchan K. Jain
(Co-Supervisor)

Dr. Kapil Hari Paranjape

Dr. Neeraja Sahasrabudhe

Dr. Lingaraj Sahu

## Acknowledgement

I accept this open door with much joy to thank every one who have helped me in effective completion of my thesis. I might want to offer my most profound thanks to **Dr. Shane D'Mello** and **Prof. Kanchan K. Jain** for giving me the chance to work under their direction and for giving me opportune guidance to comprehend my objectives and last however not the least I might want to thank my establishment, Indian Institute of Science Education and Research Mohali.

## Declaration

The work presented in this dissertation has been carried out by me under the guidance of Dr. Shane D'Mello and Prof. Kanchan K Jain at the Indian Institute of Science Education and Research Mohali. This work has not been submitted in part or in full for a degree, a diploma, or a fellowship to any other university or institute. Whenever contributions of others are involved, every effort is made to indicate this clearly, with due acknowledgement of collaborative research and discussions. This thesis is a bonafide record of original work done by me and all sources listed within have been detailed in the bibliography and is cited properly.

Vaishali Singh
(Candidate)

Dated: April 25, 2019

In my capacity as the supervisor of the candidate's project work, I certify that the above statements by the candidate are true to the best of my knowledge.

Dr.Shane D'Mello
(Supervisor)

Prof. Kanchan K Jain
(Co-Supervisor)

# List of Figures

# Contents

# Abstract

The author of this thesis aims to reproduce and extend the work done by Vladimir G. Drugov [Dru] to understand the dynamics involved in the data set, make conclusions, provide best predictive model to predict future defaults and forecast monthly trends of credits through artificial intelligence. In finance, default is the failure of payment on debt by the due date. This thesis report is devoted to ”modelling and forecasting of aspects of credit card defaults” with the help of Data exploration by statistical visualisation techniques reproduced from The extended part which is research of the author is The data used is that of a credit card company [WEB] which has demographic and financial information of it's customers and status of default in their credit card payment.

The purpose of this study is to:

- Find impact of demographic and financial variables on the status of default.
- Find important variables responsible for defaults.
- Forecast pattern of unpaid credits of the customers.

# Chapter 1

# Introduction

Data is the new gold and has been backbone of many business firms. With the arrival of digital technology, a huge amount of data is generated every second. It could be structured or unstructured. It requires a good hand in programming and knowledge of statistics to make a sense out of such huge data. Data exploration through statistical analysis helps quantitative analysis of data to understand and study large amount of data which are difficult to analyze manually. This includes Mosaic plots, Box plots , Correlation plots etc.

This work focuses on to reproducing and extending the work done by Vladimir G. Drugov [Dru]. The extendion part includes using machine learning algorithms and methods to make a model which has help predicting future defaults given some demographical and financial factors. Also forecasting unpaid credit by the customers for the next month using time series analysis.

This work is especially interesting for researches working in the field of machine learning and artificial intelligence Another interest group for this research are international companies, especially business oriented ones. This can be additional interest to the credit card customers.

# Chapter 2

# Theoretical Framework

This chapter states analysis of essential theoretical foundations. In context of framework of experiment. Following questions are being answered:

- What are the pros of machine learning and it's different types?
- What are the commonly implemented tools in machine learning?
- What are the steps implemented in machine learning?
- What is the working behind machine learning algorithms?

To answer these questions, first section of this chapter deals with machine learning, and it's types. Afterwards, explanation of machine learning algorithms, tools and steps involved in machine learning.

## 2.1 Machine Learning

Machine Learning is a subset of Artificial Intelligence. It is a scientific study of algorithms and statistical tools that gives machines the capacity to master by not directly programmed. It involves training the machine by making a model using different algorithms which has mathematical foundation, and testing the level of training of the model by using a test set.

## 2.2 Machine Learning types

- **Supervised Machine Learning** includes tasks in which the algorithm already knows the input and output values. Input values are defined as the information in form of variables that the algorithm is given to use. Output values could be continuous or categorical depending on the problem and the data. Which means that machine already knows the structure of the data and the goal of these algorithms is to assign class/label/values to the new input data after being programmed(not explicitly).

- **Unsupervised Machine Learning**, in contrast of supervised machine learning uses data set which does not have output values. Such data does not have labels. This includes tasks which requires to find classes/labels within the data.

Since this work concerns the data which has output labels as defaults(1) and non default(0). Supervised machine learning algorithms will be focused on.

### 2.2.1 Machine learning outline the following steps

1. Import data set to be used in analysis : This involves extracting data from different databases or sources
2. Analyse, order and process the data: This requires cleaning the data to make it useful for learning. Which is dealing with missing values and outliers. The data is then reviewed graphically,showing relationship variables with each other.
3. Reduce the data dimension : It involves operations like removing extraneous variables, converting old variables and forming new variables which are important to include in the model.
4. Discover the data mining problem: This requires understanding the problem whether the task requires classification, prediction, clustering etc.
5. Partition the data: Split the data into training, validation, and test datasets.
6. Select the machine learning algorithm to be implemented.
7. Adopt algorithms to execute the chore : This involves trying different variables/setting within the algorithm.
8. Translate the aftereffects of the algorithm : This includes choosing the outstanding algorithm to convey, and examining the nal decision on the test information to know the execution of the model.
9. Send the model: This progression includes running the model on genuine records to deliver choices or activities.

## 2.3 Machine Learning Tools

A very important part in machine learning, is to understand how a computer program knows which of its results were correct and which contained mistakes. As mentioned in subsection 2.2.1 the evaluation of classification tasks is generally done by splitting the data set into a training set and a testing set. The idea is to train the machine learning algorithm on the training set, while the testing set is used to calculate performance of model on indicators to examine the performance of the algorithm. A common problem that machine learning algorithms faces is limited testing and training data. Therefore, overfitting and underfitting [Dra] can be a serious problem while performing these programs.

**Overfitting**

It refers to capturing of noise and patterns by the model which do not generalize well to unseen/test data. The model performs really well to the training set but fails to perform good on the test set.

**Underfitting**

It refers to not capturing enough patterns in the data by the model. The model performs really bad on training as well as testing dataset.

A common approach to solve this problem is cross validation.

### 2.3.1 Cross Validation

Cross validation is a technique used to assess how the results of statistical analysis(model) is good at generalizing to an independent test dataset. The goal is to test the model in training phase to avoid overfitting and underfitting of data.

There are some commonly used cross validation strategies that are used to avoid overfitting and underfitting, used according to the problem one wants to solve using the dataset with machine learning.

### 2.3.2 Different Validation Strategies

**Train/Test split or Holdout**

In this strategy, data is split into two sets: train and test set. where train and test data must not overlap.

**K-Fold**

In this strategy, ample data is given for training the model and also leaves ample data for validation. It is viewed as repeated holdout and then scores after K different holdouts are averaged.

**Leave one out**

It is a certain case of K fold. This strategy iterates through every sample in the dataset each time using (k-1) objects as train samples and 1 object as test set.

**Extra**

This strategy uses stratification with which similar target distribution is achieved over different folds when data is split.

### 2.3.3 Metrics to Evaluate Machine Learning Algorithm

After the validation of dataset, performance of model is validated. since every algorithm has its flaws and advantages.There is no perfect indicator for every machine learning algorithm. Though there some commonly used criterias for evaluating the performance of a machine [Mis] learning program, they are called evaluation metrics.

**Different types of evaluation metrics** [Mis]

**Classification Accuracy**

It is equivalent to the proportion of number of right expectations to the all out number of forecasts made.

$$Accuracy = \frac{number\ of\ right\ expectations}{Total\ number\ of\ forecasts\ made}$$

**Confusion Matrix**

It is a matrix that depicts execution of a model.

|  |  | Predicted |  |
|---|---|---|---|
|  |  | 0 | 1 |
| Actual | 0 | true positive | false positive |
|  | 1 | false negative | true negative |

- True Positives : When predicted and actual output are zeroes.
- True Negatives : When model predicted 1 and the actual output was 1.
- False Positives : When model predicted 0 and the actual output was 1.
- False Negatives : When model predicted 1 and the actual output was 0.

Accuracy of the model is calculated by averaging values across the main diagonal.

$$Accuracy = \frac{true\ negatives + true\ positives}{Total\ number\ of\ records}$$

**Precision**

It is the ratio of number of correct positive results to the number of positive results predicted by the classifier.

$$Precision = \frac{true\ positives}{false\ positives + true\ positives}$$

**Recall**

It is the ratio of number of right positive results to the number of all admissible samples.

$$Recall = \frac{true\ positives}{false\ negatvies + true\ positives}$$

### Mean Absolute Error

It is the normal of the contrast between the Actual Values and the Predicted Values.

$$Mean\ Absolute\ Error = \frac{1}{N}\sum_{i=1}^{k} |y_i - \hat{y}_i|$$

where, $y_i$ = predicted values, $\hat{y}_i$ = actual values and N = number of records.

### Mean Squared Error

It is given by

$$Mean\ squared\ Error = \frac{1}{N}\sum_{i=1}^{k}(y_i - \hat{y}_i)^2$$

# Chapter 3

# Machine Learning algorithms

There are numerous algorithms [SBY$^+$17] to create a prediction model. This section states and describes six different machine learning algorithms that are commonly used for classification problems: decision trees, random forest, Naïve Bayes classifier, k nearest neighbours,artificial neural networks and recurrent neural networks. Although they all predict a dependent variable based on independent variables, they are build using different mathematical methods.

## 3.1 Classification Trees

Trees,among the information driven strategies are the most straightforward and uncomplicated to describe. They are build on partitioning records into subsets by forming partition on predictors. These partitions form reasonable rules.The resulting subgroups in terms of the yield have to be further homogeneous by forming practical classification guidlines.

It is a classification tree algorithm that enables an easy understandable representation of model using a tree like plot and is one of the most common learning methods. It is one of the supervised learning algorithm in machine learning. It can be implemented for both categorical as well as Recursive partitioning of the predictor variables is the underline idea of classification.

### Recursive Partitioning

Suppose $x_1, ...., x_r$ are the independent predicting variables and the output associated with it is y. The role of Recursive partitioning is to divide different predictor variables into non intersecting multidimensional partitions. The predictor could be continuous,ordinal or binary. First, a predictor variable is chosen, $x_i$, and a point of $X_i$, $a_i$, is selected to partition it into two parts having records $X_i < a_i$ and $X_i \geq a_i$. In continuation of this process these parts is divided in same way again by selecting a predictor variable and a partition value for that variable. This gives three rectangular regions. Further this process is continued so that smaller and smaller partitions could b achieved. The aim is to partition the whole dataset, so that more and more homogeneous partitions could be achieved. Particular split is decided on the basis of of how much they reduce the heterogeneity which is also called impurity.

### Measure of Impurity

There are numerous strategies to calculate impurity. The two most popular measures of impurity are the Gini index and entropy.If labels of the feedback variable denoted by m = 1,2,...,l. Then, Gini index for a partition S is calculated by

$$Gini(S) = 1 - \sum_{m=1}^{l} p_m^2$$

where $p_m$ is the portion of records in partition S that are member of class m. Gini index has values between 0 to (l-1)/l .

Another way to find the impurity is by calculating entropy. The entropy for partition S is calculated by

$$entropy(S) = 1 - \sum_{m=1}^{l} p_m \log_2(p_m)$$

Entropy ranges between 0 to $\log_2(l)$

### Tree Structure

The representation of modelling through classification tress has a tree like structure which has nodes:

- Root Node: It represents the sample and gets further divided into homogeneous sets.
- Branch Node: It represents the represents the sub section of the entire tree.
- Leaf Node: The end node which doesn't split further.

In a classification Tree each attribute represents outcome of the provided data and each leaf node represents the decision taken.

## 3.2 Random Forest

Random Forest is a classification algorithm made using many Decision Trees. Ensemble models combine the results from different decision trees and gives the final decision output.The main characteristic of Random Forest is that it randomly selects samples from the dataset.

The idea behind random forests is:

- Produce numerous random trees,that too by replacing sample data.
- With help of a random training set of predicting variables at every step, deploy a classifying tree to every sample to get a random forest.
- Merge the classifications got from individual trees to get upgraded results. The class is assigned to a record on the basis of maximum votes received by a class from the numerous trees received.

## 3.3 Naive Bayes Classifier

Naive Bayes Classifier is a classification algorithm used for predictive modeling which is based on Bayes theorem. The basic assumption of this algorithm is that there is no dependence among the predictors/independent variables. Which means Naive Bayes classification assumes that there's no correlation among the predictors.

The principle behind Naive Bayes to classify each record is:

- Find out the records that belong to same predictor depiction.
- Find out what classes/labels all the records are member of and the class which is most frequent.
- Allot that class/label to the unseen record.

The method answers the question,"what is the probability of a record belonging to the class of interest?"

**Cutoff probability method**

- Setup a cutoff probability for the class/label of interest which one of the class.
- Look for training data with the identical predicting variables outline as the unseen data.
- Decide the probability that these records are member of the selected label.
- If that probability is higher than the cutoff probability, allot the unseen data value to the selected label.

**Conditional Probability**

To find probability of record being a member of class $q_i$ provided that the predictor values are $x_1$,$x_2$,$x_3$...,$x_r$. For a feedback with m classes $q_1$,$q_2$,$q_3$,...,$q_r$, and the predictor values $x_1$,$x_2$,$x_3$,...,$x_r$, one want to compute

$$P(q_i|x_1, x_2, ...., x_r)$$

from,

$$P(q_i|x_1, x_2..., x_r) = \frac{P(x_1, x_2, ..., x_r|q_i)P(q_i)}{P(x_1, ..., x_r|q_1)P(q_1) + .... + P(x_1, ..., x_r|q_m)P(q_m)}$$

## 3.4 K Nearest Neighbor(KNN)

KNN algorithm is utilized to recognize k number of records in the preparation dataset that are like another record that we wish to group. We at that point utilize closely same entries to arrange the unseen entry into a label, doling out the unseen entry to the dominating label between such neighbors. Signify the estimations of the indicators for unseen entry by $x_1$,$x_2$,...,$x_r$.

The idea that works behind KNN is to find k entries in the sample dataset used for training that are similar to an unseen entry of testing dataset that we want to classify. Later an unseen record is given label using the alike/close by entries by assignment a predominant class to the new new record among these similar records/neighbors.

**Determining Neighbors**

The k-nearest neighbors calculation is a characterization strategy is suspicions for type of the connection among different label enrollment say y and the indicators $X_1$,...,$X_r$. This strategy quantifies the distance between records dependent on their indicator esteems. The most well known proportion of separation is the Euclidean distance. If ($x_1$,$x_2$,...,$x_r$) and ($v_1$,$v_2$,...,$v_r$) are two entries then distance between them is calculated as

$$\sqrt{(x_1 - v_1)^2 + ..... + (x_r - v_r)^2}$$

**Steps Involved:**

- Firstly, separation between the unseen entry from test set and each entry in training set is calculated.
- On the basis of separation k separation entries that are closest are selected.
- The value assigned to the new entry is the mean of these k data points.

**Selecting K**

Value of K which gives the best classification performance is chosen. The records of training set is classified into validation data. Then error rates are computed based on the different choices of K. Then the accuracy of prediction in the validation data is examined resulting from different choices of K between 1 and 14. The value of K resulting the highest accuracy is chosen and then used to generate classifications on new records from testing data.

## 3.5 Artificial Neural Network

Artificial Neural Network is considered a part of deep learning in machine learning. It captures complex and complicated relation among the predicting variables and the output variable. It consists of nodes, also called neurons. It adapts weighted connections between them, while the network is getting trained. Depending on the input entries being received, activation function omits output value.

This consists of:

- Input Layer: It consists of nodes/neurons that accepts predictor entries and sequential values/neurons which receives input entry from the previous layer.
- Output Layer: It consists of output of nodes from an input layer.Which then becomes input for the next layer.
- Hidden Layer: It is the layer between input and output layer.

If the data consists of r number of predictors, then the input layer consists of r number of nodes. Output received from the hidden layer is collected as input for the output layer. Hidden layer is computed by doing weighted sum of the entries, to calculate the output from node m, predictors

$$x_1, ....., x_r$$

are used as entries and imputed by receiving weighted sum

$$\theta_m + \sum_{i=1}^{r} w_{im} x_i$$

,here $w_{1m}, w_{2m}, ...., w_{rm}$ are the weights that are first set randomly.Then they get adjusted as the network learns and $\theta_m$ is a constant known as bias that controls the level contribution of node m.Then the weighted sum is taken to a function f also called activation function. Then output through activation function is calculated by

$$output_m = f(\theta_m + \sum_{i=1}^{m} w_{im} x_i) = \frac{1}{1 + e^{-}(\theta_m + \sum_{i=1}^{m} w_{im} x_i)}$$

**Back Propagation of Error**

In neural networks back propagation is the process of using an error of the network to readjust the weights given to the hidden layers, which initially were set randomly.Which means that it calculates error by comparing the predicted values by the algorithm to the actual values. Then this error is used to readjust the weights given to the hidden layers.

If w is the old weight and n is the rate (values lies between 0 to 1) which specifies how quick the process is done by running different iterations to get the weights updated. Then the new weight $w_{new}$ is calculated by

$$w_{new} = w + \triangle w$$

Where,

$$\triangle w = n\frac{de}{dw}$$

$$e = error\ associated\ with\ the\ model$$

and $\frac{de}{dw}$ is called the gradient.

There are many other activation functions that can be used based on their pros and cons. But there are some properties that an activation function must follow. Activation function make the back-propagation conceivable since the inclinations are provided alongside the error to refresh the weight and biases. Without the differentiable non linear function, this would not be conceivable.

Which implies that the function must be differentiable and monotonic in nature.

**Types of Activation functions**[Abh]

**Linear**

1. **Identity**

   $$Function : f(x) = x$$

   $$Range = (-\infty, \infty)$$

   $$\dot{f}(x) = constant$$

   which means derivative is independent of input values. This implies each time we complete a back propagation, the gradient would not change. Furthermore, this is a major issue, we are not by any stretch of the imagination improving the error since the gradient is essentially unchanged. Furthermore, not simply that guess we are endeavoring to play out a convoluted assignment for which we need various layers in our system. Presently if each layer has a linear change, regardless of what number of layers we have the last yield is only a linear change of the info.

   **Non- Linear**

2. **Sigmoid/Logistic**

   $$Function : f(x) = \frac{1}{1 + e^{-x}}$$

   $$Range = (0, 1)$$

**Advantages**

- The function is differentiable.That implies, we can discover the slope at any two points.
- Sigmoid is monotonic(increasing) yet function derivative isn't.

**Disadvantages**

- It offers ascend to an issue of vanishing gradients(

$$\frac{de}{dw} <<<< 1$$

  resulting updated weights similar to old weights) , since the Y esteems will in general react less to changes in X
- Secondly , its yield isn't zero focused. It makes the gradient refreshes go excessively far in various ways. $0 <$ yield $< 1$, and it makes optimization difficult.
- function soak and finish gradient.

3. **Tan hyperbolic**

$$Function : f(x) = \frac{1 - e^{-2x}}{1 + e^{-2x}}$$

$$Range = (-1, 1)$$

**Advantages**

- Both tanh and it's derivative is monotonic.
- Easy Optimization

  **Disadvantages**
- Has problem of vanishing gradient.

4. **ReLU(Rectified Linear Unit)**

$$Function : f(x) = max(0, x)$$

$$Range = (0, \infty)$$

**Advantages**

- Both ReLU and its derivative are monotonic.
- It doesn't initiate all the neuron in the meantime because of it's functionality.
- Computation is easier.

  **Disadvantages**
- It does not yeild zero centric values.

In light of the properties of the issue we may probably settle on a superior decision for simple and snappier convergence of the Network.

- Sigmoid and it's mixes work better on account of characterization problems.
- Sigmoids and tanh are some of the time stayed away from because of the vanishing gradient issue.
- Tanh is maintained a strategic distance from more often than not because of dead neuron problem(giving same output from different hidden layers)
- ReLU actuation work is broadly utilized as it outputs better outcomes.

  ReLU should just be utilized in the hidden layers

## 3.6 Recurrent Neural Networks

Recurrent neural networks [Fra17] is the algorithm used for sequential data. It is an algorithm that remembers it's input. Sequential data is ordered data, in which related things follow each other. RNN runs the input information cycles through a loop. To make a decision it considers both current input as well as what it has learned while processing the previous inputs. For predictors $x_{t-1}, x_t, x_{t+1}, x_{t+2}$... and w is the weight that is first set randomly.$w_i$ are the weight vectors for hidden layers and $w_r$ is weight vectors of output layers.

Then the hidden layer at time stamp t is calculated as

$$h_t = g_h(w_i * x_t + w_r * h_{t-1} + \theta_h)$$

Then the output at time stamp t is calculated as

$$y_t = g_y(w_i * h_t + \theta_y)$$

where g is the activation function.

# Chapter 4

# Exploratory Data Analysis

## 4.1 Exploration of Predictors

This chapter explores the predictors in data through statistical visualisations and answer the question about impact of different demographical and financial predictor over defaults. It has portions that covers reproduction with some changes. The idea of plots is taken from [Dru] and performed by the author

**Data Sourse:**

The dataset is of a credit card company in Taiwan, Japan. Data set has information on default payments , credit limit , demography, Bill amounts and Bill payments of of the users from April to September 2005. Data has 30,000 customer records and 24 informative variables.

**Data Description:**

- ID: Customer identity of each user.
- SEX: Gender of the customer, 2 for female and 1 for male.
- LIMIT_BAL: Total amount of credit given.
- EDUCATION: Level of education of credit card users (graduate school, university,high school,others,unknown) represented as (1,2,3,4,5) respectively.
- AGE: Age of credit card users in years
- MARRIAGE: Marital status where married is labeled as 1, unmarried is labeled as 2 and unknown is labeled as 3
- PAY_AMT1-6: Payment amount from April to September.
- PAY_0: Repayment status from may to september where -1=pay duly, 1=payment delay for 1 month, 2=payment delay for 2 months, . . . 8=payment delay for 8 months, 9=payment delay for 9 months and above
- BILL_AMT2-6: Bill amount statement from april to august)

## 4.2 Computation

To perform data exploration, data in form of a csv file with input variables was set to read in R. After the input data was set to read. Data was cleaned by removing records with NA values. If NA values are of variable is more than 40% then whole variable can be removed. In the given data set because of lesser NA values in every variable, no variable was removed, only the records with NA were removed. Then the data cleaning is followed by checking for duplicate values. Since the given data did not have duplicate values. This process was skipped. Then to impute categorical variables as factors. Categorical variables were converted into factors. Then the following libraries were imported in R to make statistical visualizations.(full R code in appendix 5.1 )

- library(caret)
- library(dplyr)
- library(vcd)

### Visualizations

To begin with plots that visualize impact of demographic and financial variables on default. Certain changes were made. which includes grouping of age of customers to carry out better visualizations.

Since age of customers were given in the form of continuous variable, to make evaluation easy, different groups of ages were made: below 20-29(customers with age between 20 to 30), 30-39(customers with age between 30 to 39), 40-49(customers with age between 40 to 49) and above 50(customers with age above 50). Whereas the article followed [Dru] took different age bounds. Author preferred this age bounds to get insights of more age levels.

Then using package ”cotplot” in R mosaic plots for categorical variables: sex, age, education level , marital status and repayment status in relation with status of default were plotted For continuous variables : Bill amount, Credit limit and payment amount,box plots in relation with status of default were plotted.

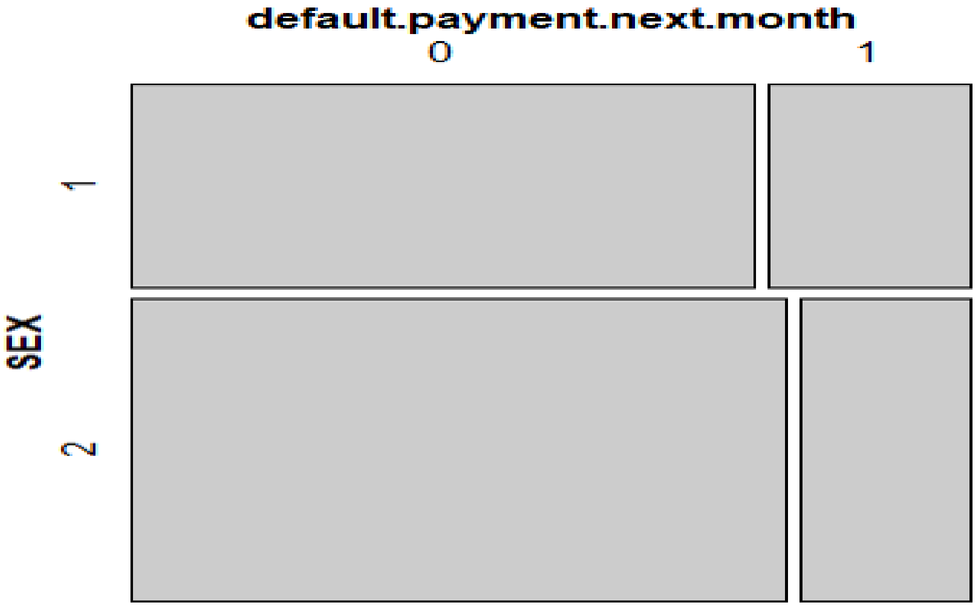

Figure 4.1: Mosaic plot between sex and status of default.

**Interpretation**

In mosaic plot shown above, sex of credit card client is represented as 1 and 2 where 1 is for men and 2 is for women. Status of Default is represented as 0 and 1, where 0 signifies non default and 1 represented as default.

It is clearly visual that men represented as 1, have more proportion default than women represented as 2. Which shows that credit card users who are men are defaulting more in comparison to credit card users who are women.

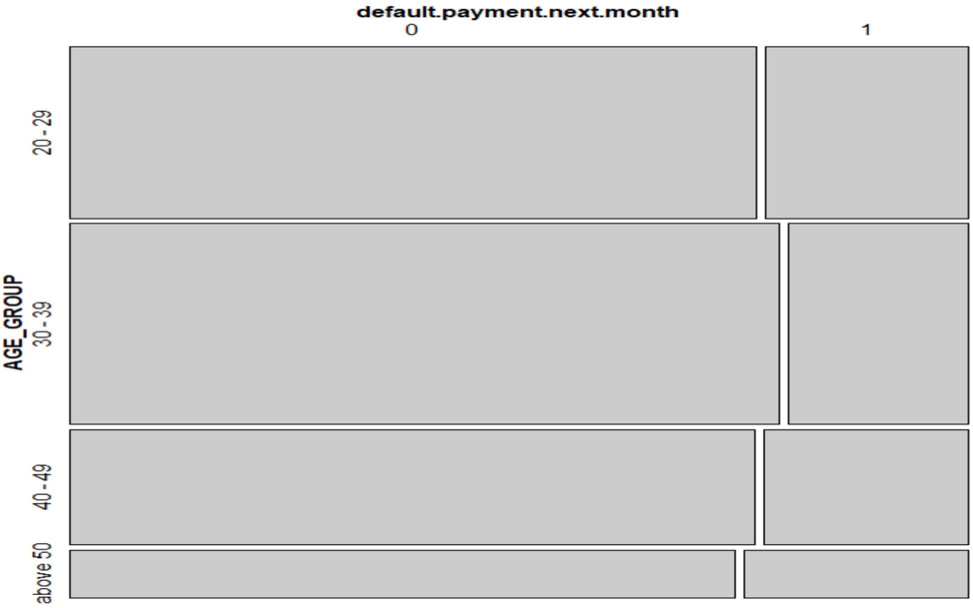

Figure 4.2: Mosaic plot between Age and status of default.

**Interpretation**

In the mosaic plot shown above, four age groups of credit card users is shown, 20-29(customers with age between 20 to 30), 30-39(customers with age between 30 to 39), 40-49(customers with age between 40 to 49) and above 50(customers with age above 50). Status of Default is represented as 0 and 1, where 0 signifies non default and 1 represented as default. It is clearly visual that clients of age lying between 20-29 which is represented as 20-29 are defaulting more in comparison to other age groups. Whereas clients of age lying between 30-39 are defaulting the least compared to the clients in other age groups.Whereas the article followed [Dru] took different age bounds. Author preferred this age bounds to get insights of more age levels.

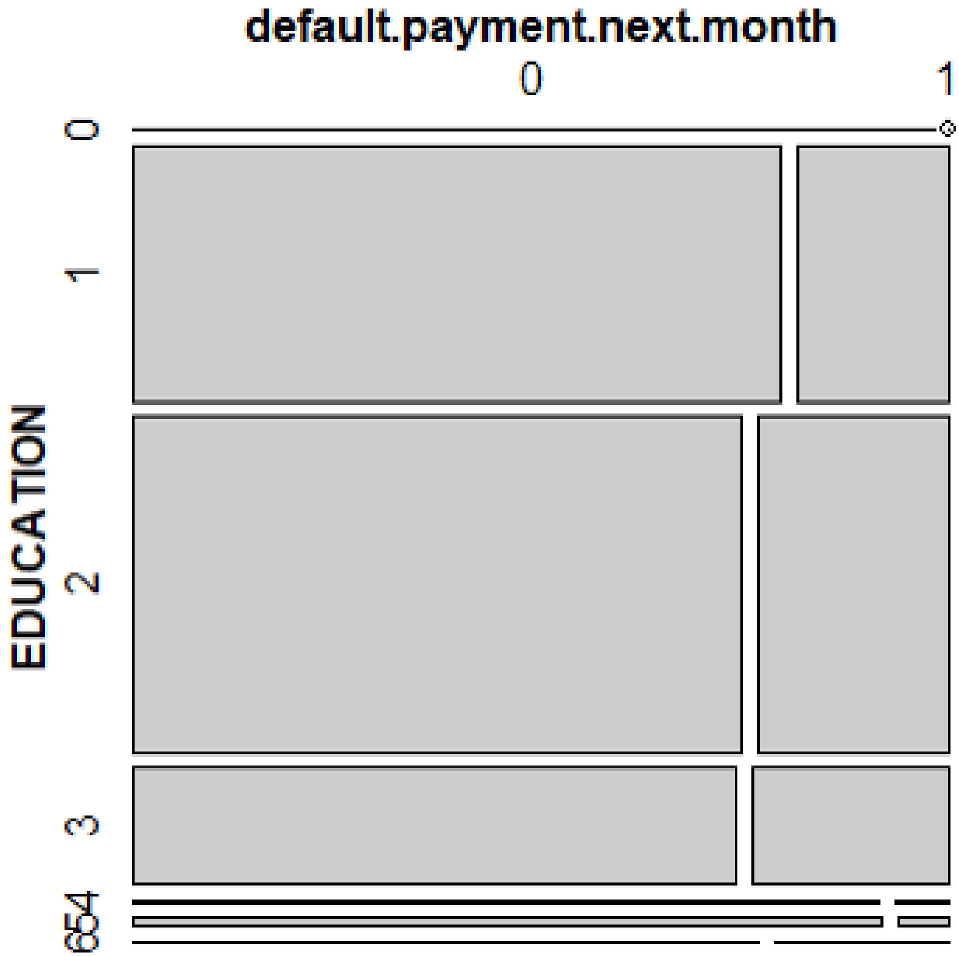

Figure 4.3: Mosaic plot between Education level and status of default.

**Interpretation**

In the mosaic plot shown above, education level of credit card users is represented by 1,2,3,4,5 and 6. Where 1 represents graduate level, 2 represents university, 3 represents high school, 5 represents others and 6 represents unknown.Status of Default is represents as 0 and 1, where 0 signifies non default and 1 represented as default. It is clearly visual that graduate clients who are represented by 1 are defaulting less in comparison to the clients with lower level of education status.

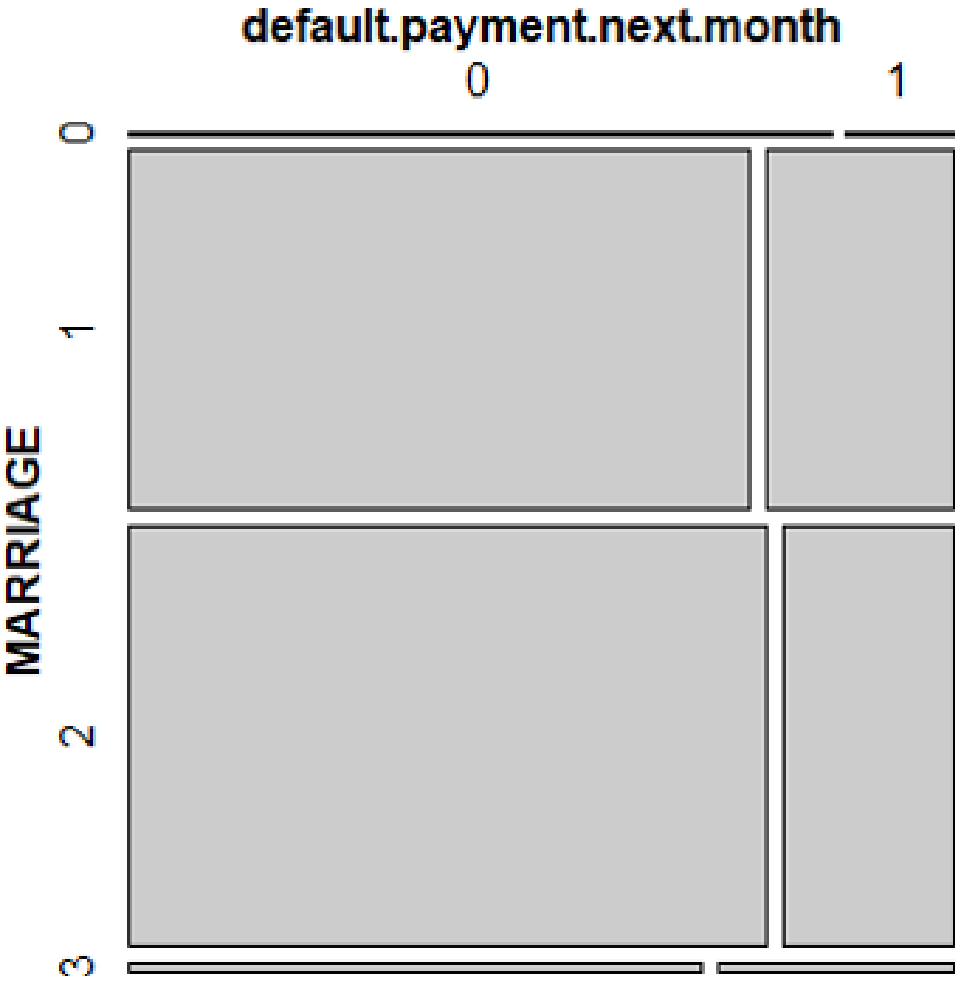

Figure 4.4: Mosaic plot between Marital status and status of default.

**Interpretation**

In mosaic plot shown above, marital status of credit card client is represented as 1,2 and 3, where 1 is for married and 2 is for unmarried and 3 is for unknown. Status of Default is represented as 0 and 1, where 0 signifies non default and 1 represented as default.

It is clearly visual that married clients represented as 1 are defaulting more in comparison to the unmarried clients represented as 2.

### Financial predictors

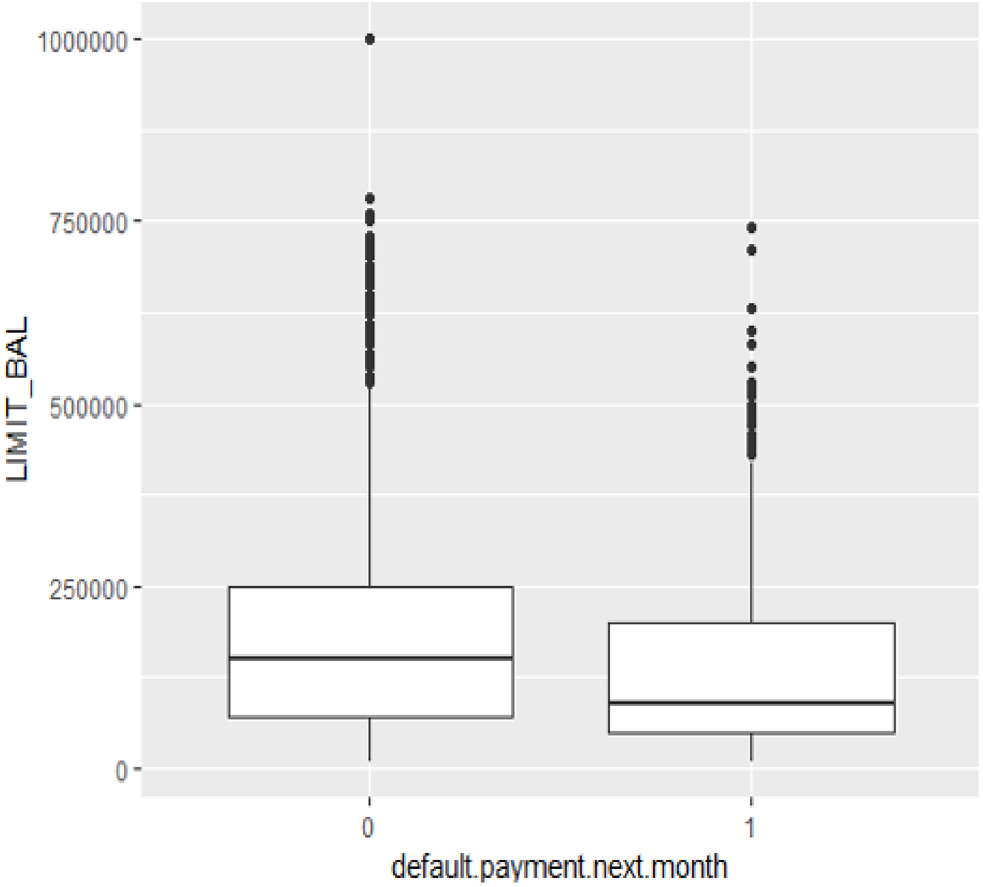

Figure 4.5: Box plot between credit card limit amount and status of default

### Interpretation

Box plots shown above represents amount of credit given to a client and status of default.Status of Default is represented as 0 and 1, where 0 signifies non default and 1 represented as default. It is clearly visual that clients with lesser credit card limit are defaulting more in comparison to the clients with higher credit card limit. Whereas the followed article [Dru] limits the range till 600000 and used a voilin style box plot. Author chose simple box plot to capture the full data and visualize the outliers.

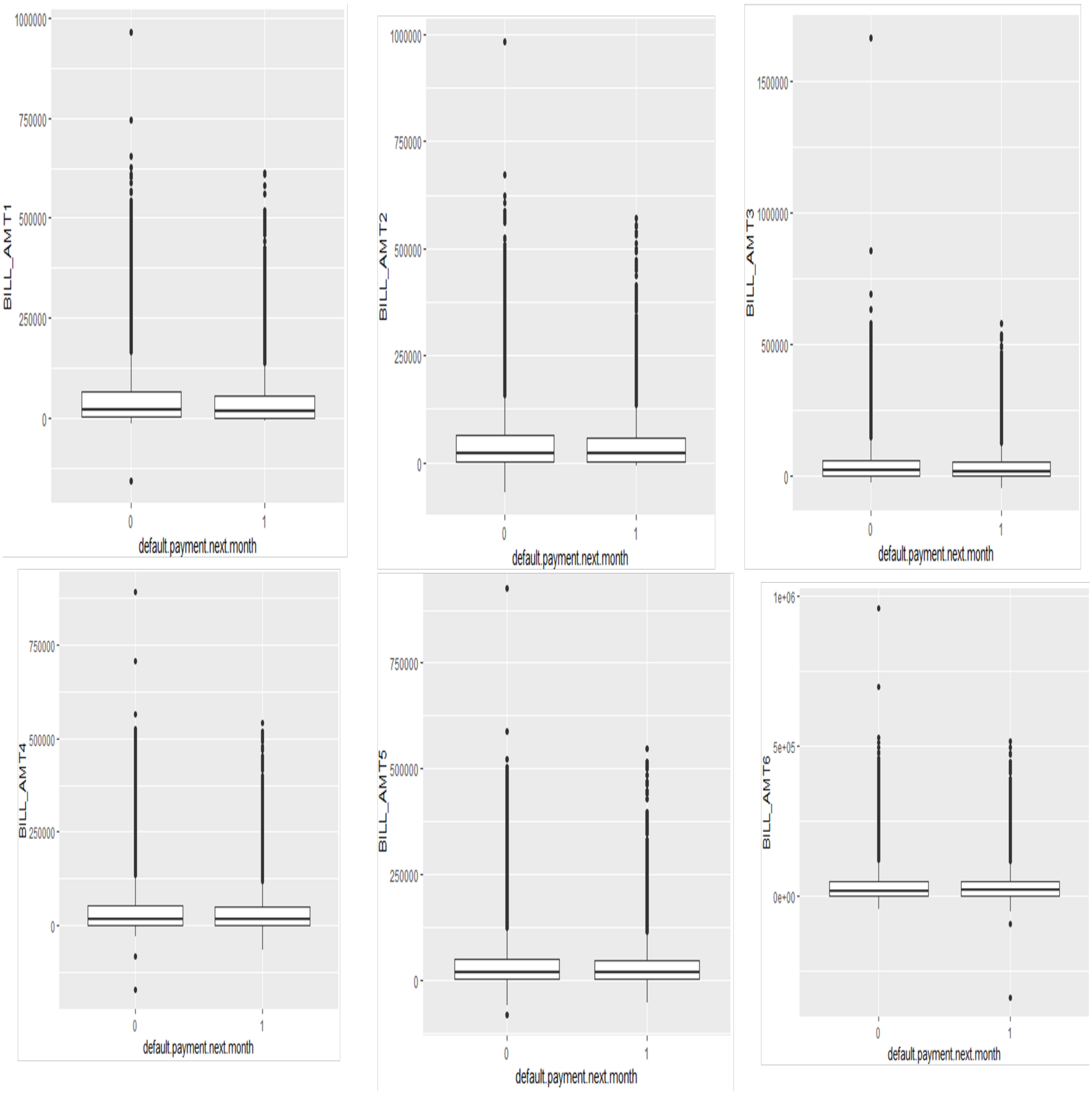

Figure 4.6: Box plot between Bill amount and status of default from April to September.

### Interpretation

Box plot shown represents amount of bill statement and status of default. Status of Default is represented as 0 and 1, where 0 signifies non default and 1 represented as default. It is clearly visual that in general clients with lesser difference between the bill amount owned in april and september are defaulting more in comparison to the clients with bigger difference between the bill amount owned in april and september. Whereas the followed article [Dru] limits the range till 400000 and used a voilin style box plot. Author chose simple box plot to capture the full data and visualize the outliers..

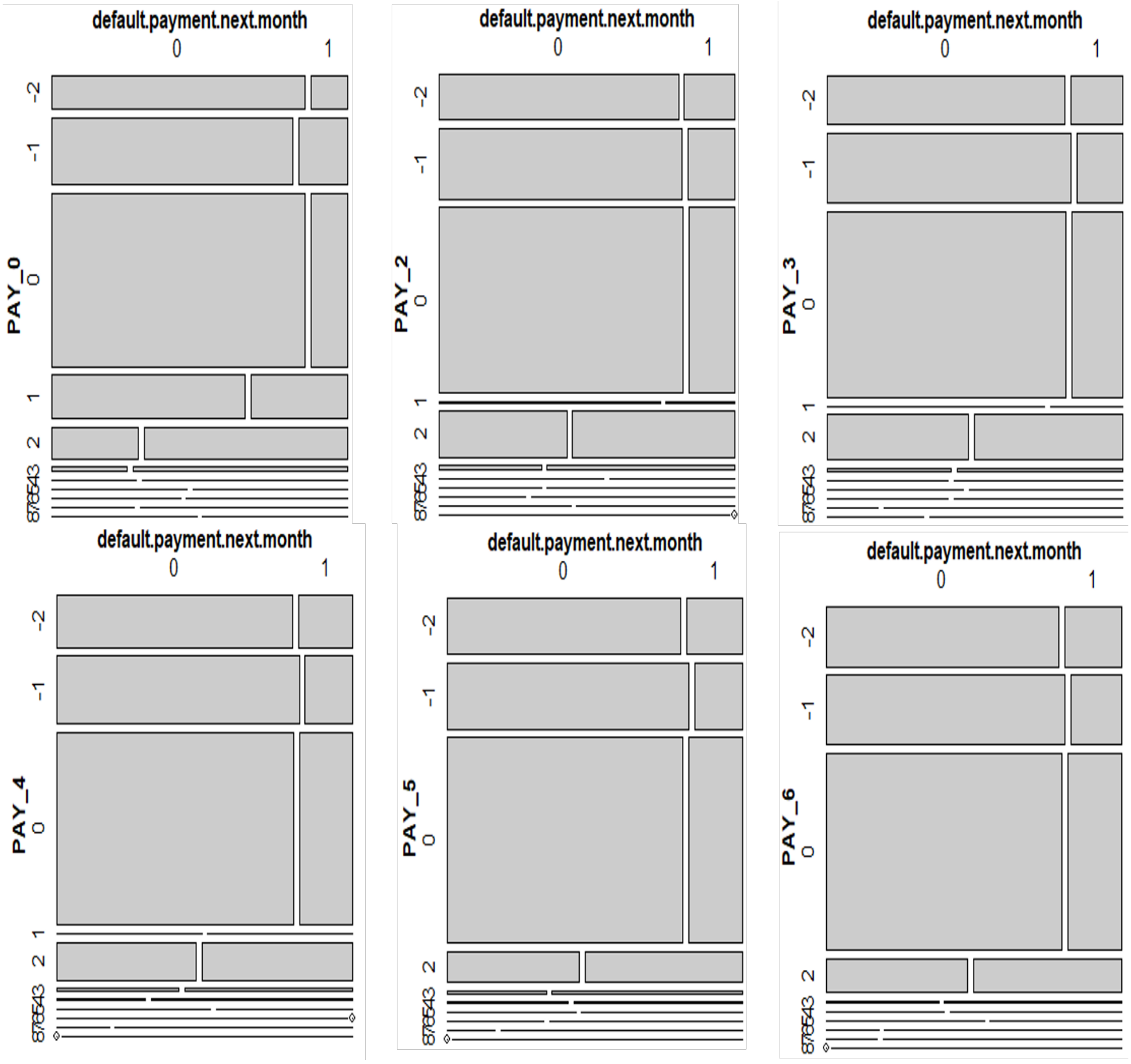

Figure 4.7: Mosaic between monthly delay in payment and status of default.

### Interpretation

In mosaic plot shown above, repayment status of credit card client is represented as $PAY\_0$,$PAY\_2$,$PAY\_3$,$PAY\_4$,$PAY\_5$ and $PAY\_6$, which repayment status of september, august, july, june, may, april respectively and pay duly is labeled as 0,payment delay for 1 month is labeled as 1, payment delay for 2 months is labeled as 2....,payment delay for 9 months and above is labeled as 9.Status of Default is represented as 0 and 1, where 0 signifies non default and 1 represented as default.

It is clearly visual that, even delay in payment for 1 month in any of the previous months, increases frequency of default of client.

# Chapter 5

# Predictive Modelling

## 5.1 Introduction

In this chapter, conclusion about predictors' impact on the default(dependent variable) made in the previous chapter will be checked. Also, by predictive modeling using machine learning algorithms for classification model will be made which can predict defaults. For this portion author did not take reference of paper cited in previous chapter. The computations and theory learned from [SBY$^{+}$17] in R

### 5.1.1 Implementation and Computation

Following the steps for machine learning mentioned in 2.2.1 , machine learning algorithms were implemented on the dataset(explained in previous chapter) in R.(full R code in appendix (6.1)

To begin with predictive modelling using machine learning algorithms, initial data cleaning steps were performed as mentioned in 2.2.1 . Next step was to normalize and standardize the data. which maps all variables to the same value range, to provide easy comparison between them.

**1. Decision Tree**

After implementation of decision tree algorithm in R,(full R code in appendix 6.2) decision tree was obtained to visualize the process and see important variables.

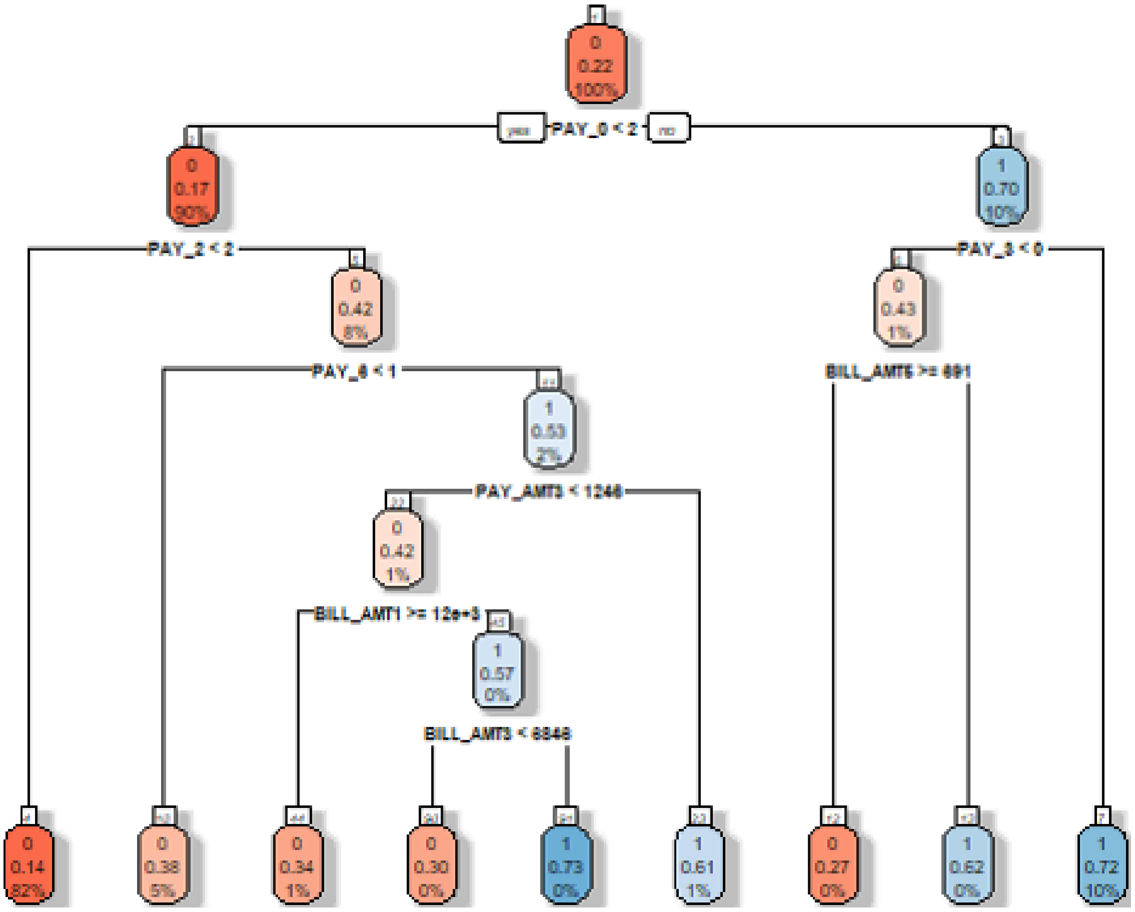

Figure 5.1: Decision Tree visualized by imputing the training dataset

Then confusion matrix was obtained after computing the algorithm to check accuracy of model in prediction of future defaults.

| | | Predicted | |
|---|---|---|---|
| | | 0 | 1 |
| Actual | 0 | 8906 | 486 |
| | 1 | 1638 | 970 |

Table 5.1: Confusion matrix from classification tree model

### Results

Decision tree gave 82.3% accuracy to predict the future defaults.

After analysis, decision tree showed algorithm can predict future defaults with accuracy 82.3% only by three predictors:

- PAY_0 Repayment status of September
- PAY_3 Repayment status of July
- PAY_2 Repayment status of August

### 2. Random Forest

Random forest algorithm was implemented in R(R code included in appendix 6.3) and random forest plot was obtained.

Then to find accuracy of the model ,confusion matrix was obtained after computing the algorithm.

| | | Predicted | |
|---|---|---|---|
| | | 0 | 1 |
| Actual | 0 | 8842 | 551 |
| | 1 | 1646 | 961 |

Random forest gave 81.6% accuracy to predict the future defaults.

Then to find out important variables which are responsible for prediction of future defaults, following plot was plotted.

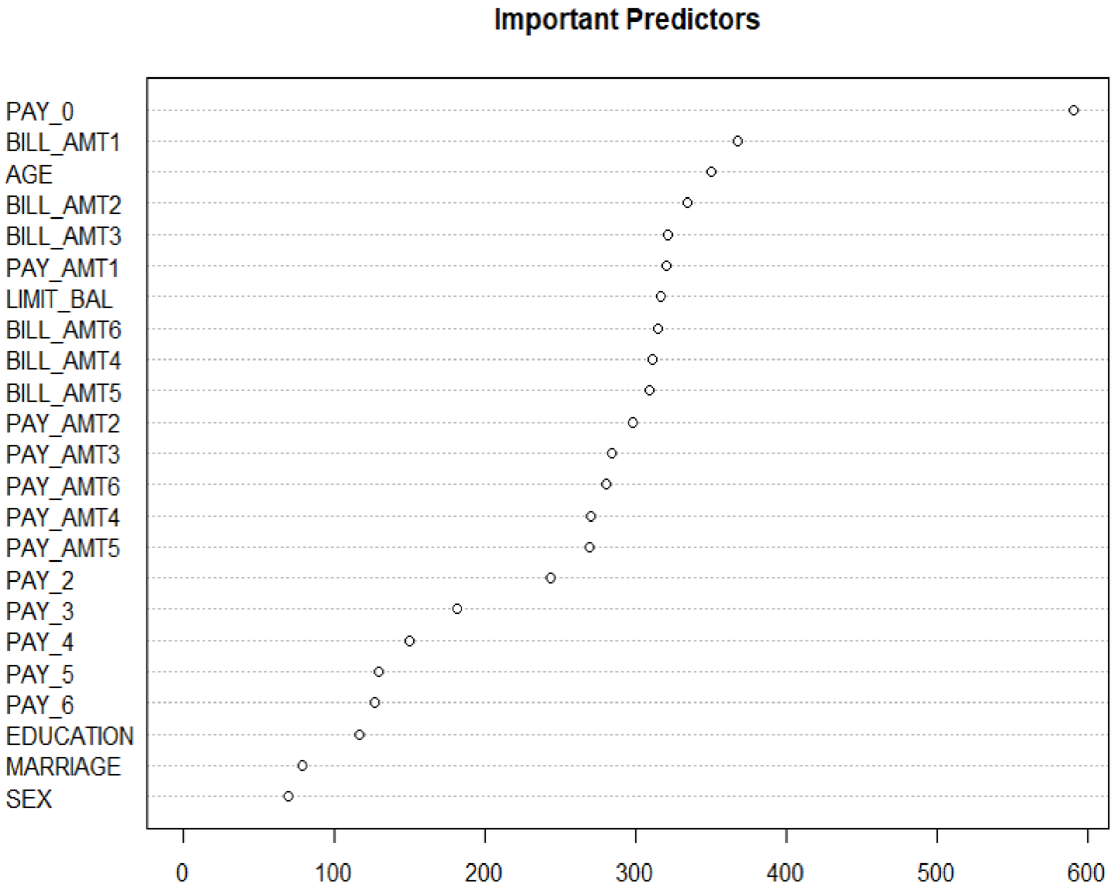

Figure 5.2: Variable importance plot through Random Forest.

### Interpretation

In above variable importance plot, scores resulted in the x-axis are computed by adding up the fall in Gini index, which is calculated by all the trees produced in the random forest classification.

Through analysis of variable importance plot, following important variables were obtained:

- Repayment status of September
- Amount of bill statement in september
- Age of credit card users in years
- Amount of bill statement in august
- Amount of bill statement in july

### 3.Naive Bayes

After implementation of Naive Bayes classifier on the dataset in R( full R code in apppendix 6.4 ),following confusion matrix was obtained to find the accuracy of the model.

| | | Predicted | |
|---|---|---|---|
| | | 0 | 1 |
| Actual | 0 | 3906 | 5486 |
| | 1 | 1049 | 1559 |

**Result**

Naive Bayes gave 45.5% accuracy to predict the future defaults.

**4.KNN**

On implementation of KNN in R(full code in appendix 6.4 ), following confusion matrix was obtained.

| | | Predicted | |
|---|---|---|---|
| | | 0 | 1 |
| Actual | 0 | 9011 | 2395 |
| | 1 | 335 | 259 |

**Result**

KNN gave 77.2% accuracy to predict the future defaults.

**5.Artificial Neural Network(ANN)**

On implementation of ANN in python(full code in appendix 6.5 ), following confusion matrix was obtained.

| | | Predicted | |
|---|---|---|---|
| | | 0 | 1 |
| Actual | 0 | 8903 | 505 |
| | 1 | 1610 | 982 |

**Result**

ANN gave 82.4% accuracy to predict the future defaults.

**5.Time Series Analysis using Recurrent Neural Network**

As mentioned in previous chapter recurrent neural network is an algorithm that works on sequential data. Given data had variables payment amount and bill amount from months april to august. To convert it into sequential data following conversions were made(full code in appendix 6.6) :

- First,variables with Payment amount were subtracted from bill amount according to their respective months, to create a variable named ”unpaid credit”.
- Then, data was transposed to create month a variable.

- Then a variable ,unpaid credit amount of five months (april to august) of a credit card user was selected randomly.
- Then a dataset with only months and unpaid credit was created to which RNN was implemented.

Above conversions were made to create a sequential data and predict unpaid credit by the credit card user for next two months by doing time series analysis using recurrent neural network( full code in appendix 6.6)

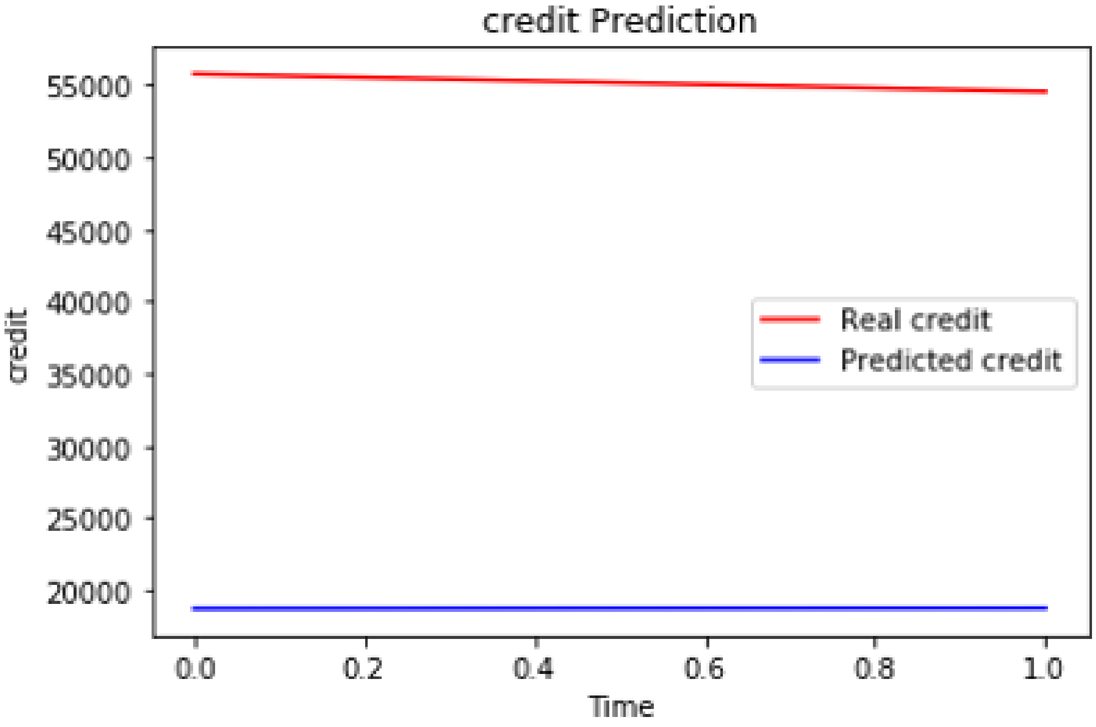

Figure 5.3: Pattern of unpaid credit for the next two months using RNN

# Chapter 6

# Conclusions and Results

This chapter summarizes the important conclusions and results obtained from exploratory data analysis and predictive modelling.

## 6.1 Impact of demographic and financial predictors on status of default

This section summarizes the impact demographic and financial predictors on status of default obtained from mosaic and box plots in exploratory data analysis. The conculsion made in this section are inspired from [Dru] with

- **PAY_0-6**(Repayment status from may to september, 2005)

  Clients who are skipping even a month of payment are getting in the category of default

- **Education:**(Education levels of clients) Clients with education level 1, 4 and 5 are Defaulting more.

- **Marriage:**(Marital status of clients) Married clients are higher in number for default.

- **Sex:**(Gender of clients) Male clients are higher in number for default compared to female clients.

- **PAY_AMT1-6:**(Payment amount from may to september) Clients with lesser amount paid in previous are defaulting more.

- **LIMIT_BAL:** (Credit limit of customer's credit card)

  Clients with lesser credit limit are defaulting more.

- **BILL_AMT2-6:**( Bill amount statement april to june)

  Clients with lesser amount bill are defaulting more.

## 6.2 Important Variables

After the Analysis of decision tree and random forest, following variables turned out to be most and least important for the prediction of status of defaults.

### 6.2.1 Most important variables

- PAY_0 (repayment status of september)
- BILL_AMT1-3(bill amount of july, august and september)
- AGE (age of credit card users in years )
- PAY_AMT1 (payment amount of september)
- LIMIT_BAL(total credit limit given to the credit card users)

### 6.2.2 Least important variables

- **PAY_2-6** (repayment status from april to august)
- **Education**(level of education of credit card users)
- **Marriage**(marital status of credit card users)
- **Sex**(gender of credit card users)

## 6.3 Best machine learning models

After computing various machine learning algorithms

- Artificial Neural Network was strongest classifier with accuracy 82.5%.
- Second Strongest classifier was the decision tree with accuracy 82.3%.
- Random Forest showed 81.6% accuracy.
- KNN showed 77.2% accuracy.
- Naïve Bayes was weakest classifier with 45.5% accuracy.

## 6.4 Result from RNN

Predicted unpaid credit of the client chosen for next two months that is october and november was 18726.824 and 18747.613 respectively.

# Appendix

### 5.1

```
”ddata <- read.csv(file.choose(),header = TRUE)

head(ddata)

#remove na columns

ddata <- ddata[, !apply(is.na(ddata), all)]

ddata$default.payment.next.month <-
as.factor(ddata$default.payment.next.month)
install.packages(”caret”)

library(caret)

library(dplyr)

newdata <- ddata %>%
newdata <- ddata %>%
mutate(AGE_GROUP = ifelse(AGE < 20, ”below 20”,
                            ifelse(AGE < 30, ”20 - 30”,
                                   ifelse(AGE < 40, ”30 - 40”,
                                          ifelse(AGE < 50, ”40 - 50”,
                                          ”above 50”)))))

install.packages(”vcd”)

library(vcd)

cotabplot(~ AGE_GROUP + default.payment.next.month ,
          data = newdata)

cotabplot(~ SEX + default.payment.next.month ,
          data = newdata)

cotabplot(~ EDUCATION + default.payment.next.month ,
          data = newdata)

cotabplot(~ MARRIAGE + default.payment.next.month ,
          data = newdata)

cotabplot(~ PAY_0 + default.payment.next.month ,
          data = newdata)
```

```
cotabplot(~ PAY_2 + default.payment.next.month ,
          data = newdata)

cotabplot(~ PAY_3 + default.payment.next.month ,
          data = newdata)

cotabplot(~ PAY_4 + default.payment.next.month ,
          data = newdata)

cotabplot(~ PAY_5 + default.payment.next.month ,
          data = newdata)

cotabplot(~ PAY_6 + default.payment.next.month ,
          data = newdata)

library(ggplot2)

ggplot(data = ddata, aes(default.payment.next.month, LIMIT_BAL))
+ geom_boxplot()

ggplot(data = ddata, aes(default.payment.next.month,BILL_AMT1))
+ geom_boxplot()

ggplot(data = ddata, aes(default.payment.next.month,BILL_AMT1))
+ geom_boxplot()

ggplot(data = ddata, aes(default.payment.next.month,BILL_AMT2))
+ geom_boxplot()

ggplot(data = ddata, aes(default.payment.next.month,BILL_AMT3))
+ geom_boxplot()

ggplot(data = ddata, aes(default.payment.next.month,BILL_AMT4))
+ geom_boxplot()

ggplot(data = ddata, aes(default.payment.next.month,BILL_AMT5))
+ geom_boxplot()

ggplot(data = ddata, aes(default.payment.next.month,BILL_AMT6))
+ geom_boxplot()

ggplot(data = ddata, aes(default.payment.next.month,PAY_AMT1))
+ geom_boxplot()

ggplot(data = ddata, aes(default.payment.next.month,PAY_AMT2))
+ geom_boxplot()

ggplot(data = ddata, aes(default.payment.next.month,PAY_AMT3))
```

```
+ geom_boxplot()

ggplot(data = ddata, aes(default.payment.next.month,PAY_AMT4))
+ geom_boxplot()

ggplot(data = ddata, aes(default.payment.next.month,PAY_AMT5))
+ geom_boxplot()

ggplot(data = ddata, aes(default.payment.next.month,PAY_AMT6))
+ geom_boxplot()"
```

### 6.1

```
”#decisiontrees
ddata <- read.csv(file.choose(),header = TRUE)
#remove na columns
ddata <- ddata[, !apply(is.na(ddata), all)]
#remove na columns
ddata <- ddata[!apply(is.na(ddata), all),]
#conversion
ddata$default.payment.next.month <-
as.factor(ddata$default.payment.next.month)
#partition
dpartdf <- sample(1:nrow(ddata),
0.6*nrow(ddata), replace = F)
dtrain <- ddata[dpartdf,]
dtest <- ddata[-dpartdf,]
#modeling
library(rpart)
dmod <- rpart(default.payment.next.month ~., data = dtrain)
dmod
#makingtree
plot(dmod,margin = 0.1)
text(dmod,use.n = TRUE, pretty = TRUE,Cex= 0.1)
#prediction
dpred <- predict(dmod, newdata = dtest, type = ”class”)
dpred
#confusionmatrix
table(”Actual Value” = dpred,”Predicted Value”
= dtest$default.payment.next.month)”
```

### 6.2

```
”#randomforest
rdata <- read.csv(file.choose(),header = TRUE)
#remove na columns
rdata <- rdata[, !apply(is.na(rdata),all)]
#remove na columns
rdata <- rdata[!apply(is.na(rdata),all),]
#conversion
rdata$default.payment.next.month <-
as.factor(rdata$default.payment.next.month)
#partition
rpartdf <- sample(1:nrow(rdata),0.6*nrow(rdata), replace = F)
rtrain <- rdata[rpartdf,]
rtest <- rdata[-dpartdf,]
#modeling
install.packages(”randomForest”)
library(randomForest)
rmod <- randomForest(default.payment.next.month ~. ,data = rtrain)
rmod
#get list of important variables
rmod$importance
#prediction
rpred <- predict(rmod, newdata = rtest, type = ”class”)
rpred
#confusionmatrix
table(”Actual Value” = rpred,
”Predicted Value” = rtest$default.payment.next.month)”
```

### 6.3

```
"#naivebayes
iidata <- read.csv(file.choose(), header = TRUE)
#remove na columns
iidata <- iidata[, !apply(is.na(iidata),2,all)]
#remove na columns
iidata <- iidata[!apply(is.na(iidata),2,all),]
#normalization
iidata[,1:4] <- scale(iidata[,1:4],center = T, scale = T)
#partition
iipartdf <- sample(1:nrow(iidata),0.6*nrow(iidata), replace = F)
iitrain <- iitrain <- iidata[iipartdf,]
iitest <- iidata[-iipartdf,]
#modeling
library(e1071)
nmod <- naiveBayes(class ~., iitrain)
attributes(nmod)
nmod
nmod$tables
nmod$apriori
#prediction
nmodtest <- predict(nmod,iitest[,-5], type = "class")
#confusionmatrix
table("Actual Value" = nmodtest,"Predicted Value" = iitest$class)"
```

### 6.4

```
”idata <- read.csv(file.choose(), header = TRUE)
head(idata)
idata <- idata[, !apply(is.na(idata),2,all)]

#normalization
idata[,1:4] <- scale(idata[,1:4],center = T, scale = T)
#partition
ipartdf <- sample(1:nrow(idata),0.6*nrow(idata), replace = F)
itrain <- itrain <- idata[ipartdf,]
itest <- idata[-ipartdf,]
#modeling
library(class)
modk <- knn(train= itrain[,1:4], test = itest[,1:4],
cl = itrain$class , k = 5)
summary(modk)
#classificationmatrix
table(”Actual Value” = modk,”Predicted Value” = itest$class)

#miscalssificationerror
mean(mod!= itest$class)
#choosingKvalue
modtrain = NULL
modtest = NULL
errtrain = NULL
errtest = NULL
for(i in 1:20) {
  modtrain = knn(train = itrain[,1:4],
  test = itrain[,1:4],
                 cl = itrain$class , k = i)
  modtest = knn(train = itrain[,1:4],test = itest[,1:4],
                cl = itrain$class , k = i)
  errtrain[i] = 100*mean(modtrain!= itrain$class)
  errtest[i] = 100*mean(modtest!= itest$class)
}
dfp = data.frame(”value of k” = 1:20,”error training” = errtrain,
                 ”error validation” = errtest)
round(dfp, digits = 2)
range(dfp$error.validation)
plot(dfp$value.of.k,dfp$error.validation,las = 1,type = ”l”,
     xlab = ”value_of_k”, ylab = ”validation error”,
     xlim = c(0,2), ylim = c(0,8))
range(dfp$value.of.k,dfp$error.training)
#bestk
min(errtest)
bestk = dfp[which.min(errtest),1]; bestk

#classificationmatrix
```

```
table("Actual Value" = cctest$default.payment.next.month,
"Predicted Value" = modkcc)"
```

### 6.4

```
”import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
dataset = pd.read_csv('credithesis.csv')
X = dataset.iloc[:, 1:24].values
y = dataset.iloc[:, 24].values
from sklearn.preprocessing import LabelEncoder,
OneHotEncoder
onehotencoder = OneHotEncoder(categorical_features = [2])
X = onehotencoder.fit_transform(X).toarray()
X = X[:, 2:]
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size = 0.4, random_state = 0)
from sklearn.preprocessing import StandardScaler
sc = StandardScaler()
X_train = sc.fit_transform(X_train)
X_test = sc.transform(X_test)
import keras
from keras.models import Sequential
from keras.layers import Dense

classifier = Sequential()
classifier.add(Dense(units = 12, kernel_initializer = 'uniform',
activation = 'relu'))
classifier.add(Dense(units = 12, kernel_initializer = 'uniform',
activation = 'relu'))

classifier.add(Dense(units = 1, kernel_initializer = 'uniform',
activation = 'sigmoid'))
classifier.compile(optimizer =
'adam', loss = 'binary_crossentropy',
metrics = ['accuracy'])
classifier.fit(X_train, y_train,
batch_size = 32, epochs = 100)

y_pred = classifier.predict(X_test)
y_pred = (y_pred > 0.5)
from sklearn.metrics import confusion_matrix

cm = confusion_matrix(y_test, y_pred)”
```

### 6.5

```
"import pandas as pd
import numpy as np
import matplotlib.pylab as plt
%matplotlib inline
from matplotlib.pylab import rcParams
rcParams['figure.figsize'] = 15, 6
d1 = pd.read_csv('creditdefault.csv')
pd.read_csv('creditvvv.csv')
d1 = d1.drop(d1.columns[1],axis=1)
d1 = d1.drop(d1.columns[1],axis=1)
d1 = d1.drop(d1.columns[1],axis=1)
d1 = d1.drop(d1.columns[1],axis=1)
d1 = d1.drop(d1.columns[1],axis=1)
d1 = d1.drop(d1.columns[1],axis=1)
d1 = d1.drop(d1.columns[1],axis=1)
d1 = d1.drop(d1.columns[1],axis=1)
d1 = d1.drop(d1.columns[0],axis=1)
d1.to_csv('creditvv.csv')
identity = d2.iloc[:,0]
datat = pd.concat([identity,credit1,
credit2, credit3, credit4, credit5], axis = 1)
datat.to_csv("newdefault.csv")
data2 = pd.read_csv('newdefault6.csv')
data3.to_csv('ddata7.csv')
data4 = data3.iloc[:,:8]
data5 = pd.read_csv('ddata11.csv',
parse_dates=['Month'], index_col='Month')"
```

### 6.6

```
"import numpy as np
import matplotlib.pyplot as plt
import pandas as pd

dataset_train = pd.read_csv('credit_train.csv')
training_set = dataset_train.iloc[:, 1:2].values
dataset_total
len(dataset_total)
from sklearn.preprocessing import MinMaxScaler
sc = MinMaxScaler(feature_range = (0, 1))
training_set_scaled = sc.fit_transform(training_set)
X_train = []
y_train = []
for i in range(2, 3):
    X_train.append(training_set_scaled[i-2:i, 0])
    y_train.append(training_set_scaled[i, 0])
X_train, y_train = np.array(X_train), np.array(y_train)
X_train = np.reshape(X_train, (X_train.shape[0],
X_train.shape[1], 1))
from keras.models import Sequential
from keras.layers import Dense
from keras.layers import LSTM
from keras.layers import Dropout
regressor = Sequential()
regressor.add(LSTM(units = 50, return_sequences = True,
input_shape = (X_train.shape[1], 1)))
regressor.add(Dropout(0.2))
regressor.add(LSTM(units = 50, return_sequences = True))
regressor.add(Dropout(0.2))
regressor.add(LSTM(units = 50, return_sequences = True))
regressor.add(Dropout(0.2))
regressor.add(LSTM(units = 50))
regressor.add(Dropout(0.2))
regressor.add(Dense(units = 1))
regressor.compile(optimizer = 'adam', loss = 'mean_squared_error')
regressor.fit(X_train, y_train, epochs = 100, batch_size = 32)
dataset_total = pd.concat((dataset_train['credit'],
dataset_test['credit']), axis = 0)
inputs = dataset_total[len(dataset_total) -
len(dataset_test) - 2:].values
inputs = inputs.reshape(-1,1)
inputs = sc.transform(inputs)
X_test = []
for i in range(2, 4):
    X_test.append(inputs[i-2:i, 0])
X_test = np.array(X_test)
X_test = np.reshape(X_test, (X_test.shape[0], X_test.shape[1], 1))
predicted_credit = regressor.predict(X_test)
predicted_credit = sc.inverse_transform(predicted_credit)
from sklearn.preprocessing import MinMaxScaler
sc = MinMaxScaler(feature_range = (0, 50000))



plt.plot(real_credit, color = 'red',
label = 'Real credit')
plt.plot(predicted_credit,
color = 'blue', label = 'Predicted credit')
plt.title('credit Prediction')
plt.xlabel('Time')
plt.ylabel('credit')
plt.legend()

plt.show()"
```

# Bibliography

[Abh] Abhigoku, *Activation functions and its types in artifical neural network*, https://medium.com/@abhigoku10/activation-functions-and-its-types-in-artifical-neural-network-14511f3080a8.

[Dra] Georgios Drakos, *Cross-validation*, https://towardsdatascience.com/cross-validation-70289113a072.

[Dru] Vladimir G. Drugov, *Default payments of credit card clients in taiwan from 2005*, https://rstudio-pubs-static.s3.amazonaws.com/281390_8a4ea1f1d23043479814ec4a38dbbfd9.html.

[Fra17] Chollet Francois, *Deep learning with python*, 2017.

[Mis] Aditya Mishara, *Metrics to evaluate your machine learning algorithm*, https://towardsdatascience.com/metrics-to-evaluate-your-machine-learning-algorithm-f10ba6e38234.

[SBY+17] Galit Shmueli, Peter C Bruce, Inbal Yahav, Nitin R Patel, and Kenneth C Lichtendahl Jr, *Data mining for business analytics: concepts, techniques, and applications in r*, John Wiley & Sons, 2017.

[WEB] https://archive.ics.uci.edu/ml/machine-learning-databases/00350/.