

Bank Loan Default Prediction using Machine Learning Algorithm



Satyajit Saha
Dublin Business School
M.SC Data Analytics (2019-2020)

Declaration

I hereby declare that apart from the explicit references made by others to the craft, the substance of the thesis is unique and has not been submitted in entire or to some extent in other colleges and some other degree. This project is my very own and contains nothing which is the result of work done in a group or a joint effort with others.

Satyajit Saha
12.02.2019

Abstract

Loan default cause huge loss for bank so they much attention to curb the bad loans and apply various methods to detect the customer behaviour of their customer.

In this data challenge, we will be working with 8 different datasets which is collected from Kaggle and as a Data Analyst / Data Scientist, we will be performing below tasks-

- Use Python to connect and upload the datasets
- Read the tables into Panda data frame
- Data cleaning
- Pre-process data for Machine Learning
- Use Rapid miner for sorting the algorithm
- Train ML algorithm to predict the default customer
- Evaluate model performance
- Using Tableau for Data Visualization

Chapter I

1. INTRODUCTION

Defaulting on loan [1] is failure of repayment of a loan. It is breach of contract as a mortgage or some type of loan are supposed to be paid in certain time. From consumer point of view, when they are not able to repay the loan they received, it's called default on a loan. But from corporate perspective a default is a failure by the corporation to pay the interest or other debt security owned by corporation debtholders.

A default on a loan obligation can make serious consequences [2, 3] and affect the credit score [4, 5] of the individual or company that defaults on the loan. If the bank decides to call in a loan, then normal discussion takes place with concerned stakeholders. Upon satisfaction review and promise, banks lend money to individuals. Analysis of default on a load is therefore of paramount importance for banks and financial institutions as well as insurance companies [6-10]. To enter into the loan default analysis, we need to be acquainted with the following concepts:

a. Bank Loan

A bank loan is the lending money by a bank to individuals, organizations, corporations etc. The borrower is usually liable to pay the interest until the debt is repaid and also to repay the principal amount. The interest rates depend on the banking institutes and it could be fixed or variable.

b. Default

The term default means fail to meet the legal obligation of the financial institution such as when home buyer fails to make mortgage payment, or a corporation fails to pay a bond which has reached maturity.

1.1 Bad loan globally

India has the worst loan ratio out of 10 major economy in world. Gross non-performing assets (NPAs) at Indian banks continue to rise from 8.93% in September 2017 to peaking at 10.3% in March (2018), with the bad debt burden at state banks surging by nearly a fifth from end-December levels brought about by rising slippages from the metals and power sector.

Other countries in the top ten only have low single-digit bad loan ratios with Brazil at 3.6%, France at 3.1%. Both Germany and China have bad loan ratios of 1.7%.

Reserve Bank of India said (Dec 2018) said that the ratio for the bank fell for the first in India since 2015. \$190 billion pile of soured and stressed debt has cast the future of some lenders in doubt and curbed investments (Source Economic Times)

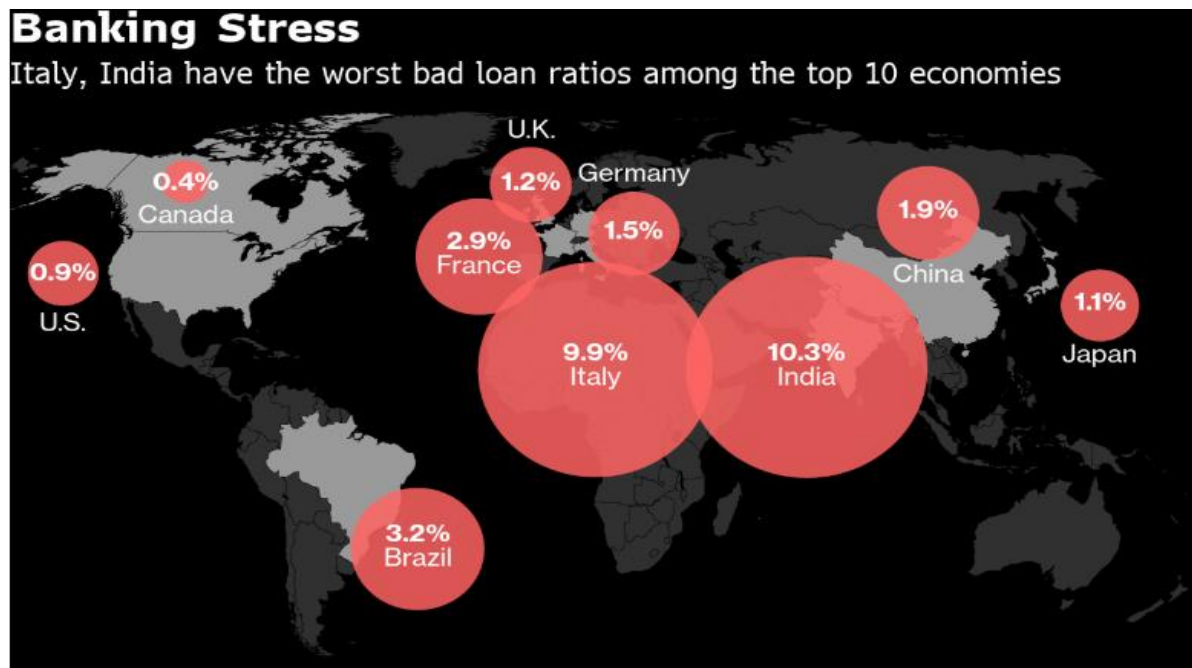


FIG. 1.1. Banking Stress Globally [11]

Source-IMF's Financial Soundness Indicators

Supervisory authorities primarily concerned with protecting depositor's interest via ensuring that financial institutions can survive under business as usual conditions and sufficiently immune to any adverse market shocks. To distinguish between strong and weak bank, supervisory authority makes use of early warning, expert system or statistical modelling. The outcome of this analysis can drive the imposition of targeted regulatory measures. These measures can take the form of pre-emptive corrective actions addressing vulnerabilities of weaker banks and as a result increase their chance of sustainability.

Between 1934 and 2014 there were 4069 banks in the United States that failed or received financial assistance from FDIC. More precisely 3483 banks failed or were assisted by the Central Bank from 1980 to 2014 following the deregulation of the US banking system in 1980's, notwithstanding the considerable efforts made by supervisory authorities in identifying vulnerable financial institutions, according to the FDIC records.

Based on this statistic it is evident prompt action and early measurement was not taken to avoid the mishap. It is essential that all risk drivers and relevant information should be combined into a single measure, representing each bank's financial strength. Reflecting

in a single and easy score a bank's overall risk could prove to be a difficult task due to the big bulk of information that is currently collected by supervisory authorities.

Here we will focus on predicting customer behaviour and mapping the likely to be default customer. So, the financial institutions such as bank can take measure at early stage and avoid the bad loan.

It will also reduce the operation cost and enhance revenue of the bank. Therefore, it is necessary to avoid bad loans. Also, it will ensure better compliance and reinforce security.

Here I will be using supervised classification models for predicting customer behaviour and loan default predictions. Based on accuracy model, I will be choosing appropriate algorithm.

Keywords: loan default, mortgage, non-performing assets

1.2 Literature Review

Several studies have explored the usage of Machine Learning in bank loan default prediction and I have taken references of their work. Also, I will highlight their algorithms and method to predict the customer behaviour analysis.

Jia [11] used Logistics regression algorithm for bad bank loan prediction. Initially he made comparison between Random Forest, Logistic Regression and XG boosting. Then based on ROC curve he choosed logistic regression. A Bayesian interface-based analogous to support vector machine (SVM)has been considered by Huang et. al., (2011) [12]. However large drawback of this approach was too much computational complexity.

Puvvada et. al., [13] implemented Recursive Feature Elimination (RFE)using Logistic Regression model to get the best 10 features. On ROC curve they made comparison among logistic regression, SVM, MLP and K nearest neighbours. They have chosen logistic regression for the accuracy on final draft. Galindo (2000) [14] used decision- tree model on mortgage-loan data for credit risk assessment. Coal (2010) [15] examined the defaults of US commercial bank that occurred in 2009 by examining CAMEL ratio as well as portfolio variables such as real estate, loan and Mortgages which proved to be important indicator.

Manjeet Kumar et. al.,[16] used Multilayer Perceptron Neural Network approach for default prediction. They have used PCA to reduce number of dimensions to overcome the problems of under-fitting as well as reduce the time and complexity required to train as well as generating output. Ensemble models have been used for loan prediction and reducing the error. An ensemble is a supervised learning algorithm comprising many weak learners which will become strong learner when working together.

Amira Kamil et. al.,[17] constructed a loan default predication model using three several neural network training algorithms. The aim is to test accuracy using attribute filter

technique and develop a model called ensemble model by combining the results of those three algorithms. The experiment did on several parameters like training time, MSE, R, iteration for comparison. The best algorithm was Levenberg-Marquardt because it had largest R and the slowest algorithm is One Step Secant (OSS). For the accuracy purpose, the filtering function was applied on original dataset that produced two another dataset. Then for each data set different training algorithm of neural network is applied and the filtering function gave the better model among all the models.

Ghadge [18] develops the artificial neural network model for predict the credit risk of a bank. Altunbas et. al. (2012) [19] demonstrated that a strong deposit base and diversification of income sources were the key characteristics of a business models that typically relate to significantly reduced default risk. Berger (2013) [20] showed that capital (either total equity or regulatory capital), had a positive impact on the survival probabilities and market shares of small banks, during all time horizons.

Wanke et al. (2015) [21] showed that, along the typical CAMELS proxies, bank contextual information, such as ownership type, country of origin, bank type and operating system (Islamic or conventional), also have a significant impact on efficiency. Chiaramonte et al. (2015) illustrated that z-score3 is, at least, as effective as CAMELS variables, with the advance of being less demanding in terms of data, while it shows an increased efficiency on more sophisticated business models

Haling (2006) [22] introduced a two-step survival time procedure that combines a multi-period logit model and a survival time model and focused on 50 variables covering information regarding bank characteristics, credit risk of the loan book, capital structure, profitability, management quality and macroeconomics. Kolari et al. (2002) [23] introduced the parametric approach of trait recognition to develop early warning systems and incorporated variables related to a number of different bank characteristics, including size, profitability, capitalization, credit risk, liquidity, liabilities and diversification. Lal (2014) [24] focused on profitability factors during a stress period.

Aida Krichene (2015) [25] made analysis on credit risk with K- nearest neighbour. A ROC curve had been plotted to assess the performance of the model. The result shows that KNN model with 95.6% accuracy was the best model with cash flow information. Alina et. al. [26] published an article on credit risk modelling for company's default prediction using neural networks (2016). The paper asses the default risk on a sample of 3000 companies applying for credit to an International bank operating in Romania. Based on old credit history the author has distributed the companies in seven class, using and adapting standard and poor categories. Here the author has performed the estimation using first using logit regression and then ANN (Artificial Neural Network). Later on, the results were compared with Standard and Poor's transition matrix for 2010. The results were compared in terms of accuracy power and arguments were given for choosing ANN model.

Glorfled (2010) [27] proposed a systematic way to make an optimal design of high-performance model for neural network estimating credit value related to commercial loan. As per the author the neural network was able to classify loan application 75 % correctly. A study by A.J.F. Loux and A.J.F Feelders [28] jointly conducted a study on personal loan by using Data Mining technique. It was carried out into the customers of ABN AMRO bank, Netherlands. Historical data of the clients and their return back activities and behaviours are used to predict whether the customer will be default or not. Data Mining technique was used to assess the personal information.

Jozef Zurada [29] used decision tree and neural network in credit risk evolution. Glorflied [30] presented a powerful approach. They aim to develop neural network approach for assessing the commercial loan. Here the developed neural network model was able to classify the 75% data correctly.

Michale Johnson [31] studied broadly the usage of data mining technique in banking sectors. They built a prediction model if the customer will pay the amount after getting the loan or not, using neural network and classification.

Wolfgang Härdle et. al.[32] predicted bankruptcy with Support Vector Machine. As per their research support vector machine are capable of extracting useful information from financial data. Here SVMs produced better classification result than any other parametric method. SVM is a classification method which is based on statistical theory. It is already successfully applied on optical character recognition, medical diagnosis and text classification. Moreover, SVM do not rely heavily on heuristic and have a flexible structure. Similarly, Bo Wang [33] also implemented SVM classifier to research on housing loan credit. According to the authors SVM is a good classifier for binary classification technique and the result possess stronger robustness. Their research shows that SVM can resolve unbalance problem efficiently.

Peter Addo [34] analysis on credit risk using Deep Learning models. They implemented ANN method for analysing the credit fraud.

Suresh Ramakrishnan et. al [35] made analysis on Corporate Default Prediction using Ensemble Classification. The key idea of Ensemble classification is combining different approach. They applied Adaboost algorithm. Adaboost applies the classification system repeatedly to the training data but learning attention is focused on adaptive weight.

Frydman (1985) [36] used decision tree model for default prediction. Using this model, they marked failed and non-failed firms based on country level factors. Quinlan (1996) [37] noted that decision tree model can deal with noise and non-systematic error as well in the value of features. There are other studies also who predicted default prediction using the same method i.e. Messier Jr. (1988) [38], Pompe and Feelders (1997) [39]

Shin (2002) [40] suggested a model using genetic algorithms technique. However recently some of the mainstream commercial firm using ANN model for predicting default analysing. For example, Moody's public firm and many other public bank and financial institution developed ANN model for default prediction (Atiya,2001) [41]

In a major study on default prediction Gestel (2005) [42] employed SVM and logistic regression. They combined both algorithms for better result which is necessary for ratings bank.

Abellan (2012) [43] showed that using ensemble method on a special type of decision trees, called CDTs (Credal Decision Tree), provides a suitable tool for the classification task.

Myers (1963) [44] implemented a multi-stage technique in which they employed two stage discriminate analyse model. They reported that second stage identifier model identified 70% more bad case than the first stage model. Lin (2002) [45] said that there is 3 % of improvement when employing a logistic model, followed by a neural network. However, there are limited study where different technique or algorithms have been compared within credit risk.

Key Words-SVM, KNN, Deep Learning, Neural Network, KNN, Ensemble Model

Chapter II

METHODOLOGY

The primary focus of this study is to predict the customers default on loans using machine learning algorithms [12, 13] using Python programming language. To this end, we are going to use Random Forest, SVM, LR, NN algorithms to predict the loan default risk among customers.

2.1 Random Forest Model

Random forest is popular classification model algorithm. It was invented by Bierman in 2000 and later on it gained much popularity. It is being widely used in different sphere these days. The main reason for wide adaption of this algorithm is accurate excellency among other supervised classified model. One of the main features of this algorithm is, it can easily fit with big dataset. It also can handle large number of input variable without strict correlation. Random Forest (RF) provides valuable information about input parameter interaction which is unlikely available with other supervised algorithm such as SVM or Neural Network model.

Furthermore, Random Forest can be useful to produce forecasts and offer proximities among the pairs which is important for clustering either under a supervised setup or unsupervised set up. One significant feature of this method is that this model provides consistency in performance as number of tress get increase the efficiency and flexibility embedded in the structure of Random Forest leads to enhance the performance in classification problem.

Random forest model is widely used in academic world and in the finance industry to model time series and to analysing recurring pattern for improving prediction accuracy.

Observation that are not selected in original dataset are called off-bag observations. They are used to estimate error rate and estimate feature importance.

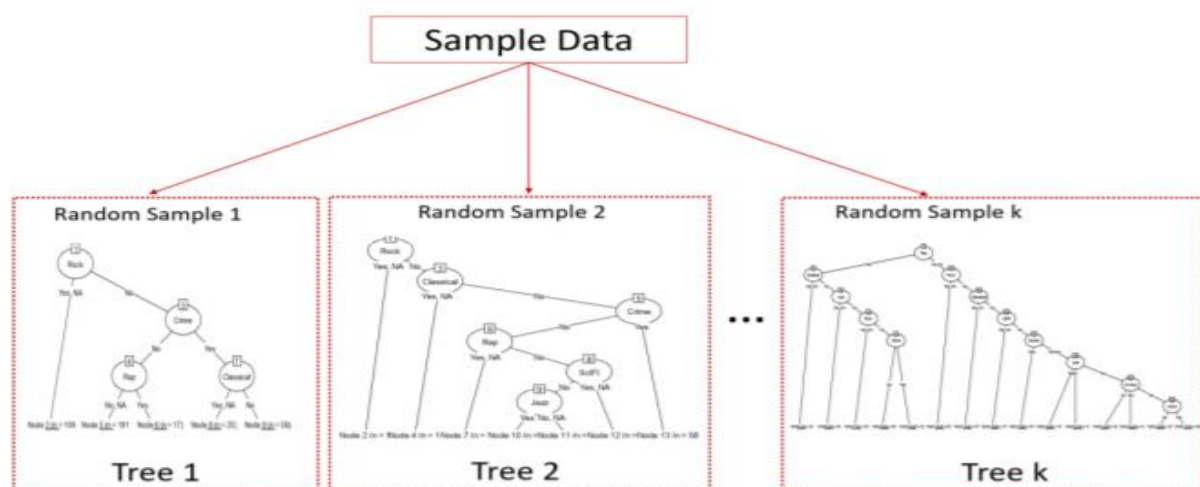


FIG. 2.2.1 Random Forest Algorithm

As illustrated on above diagram each tree is different here and this is because at each node the best split is determined by randomly selecting features. This is result in forest of decision.

2.1.1 Advantage of Random Forest

Below are some advantages of Random forest model-

- They are easy to build and execute. Also provide very good result
- It reduces classification and regression error
- It is competitive with Neural Networks and Support vector Machines (SVM)
- They required minimum data preparation.
- Missing values are handled
- They run efficiently in large database with many features
- Overfitting is not the problem here
- Bag error rate tends to be very accurate

2.1.2 Limitation of Random Forest

Below are some limitation of random forest model-

- They are compatible when relationship between the target variable and feature variable is linear.
- They are biased with large number of classed
- Classification rule generated by Random Forest are generally incomprehensible.
- Sometime usage of RF is challenging where clarity on rule generation and variable interact is important.

2.2 Artificial Neural Network

Artificial Neural Network is one of major computational algorithms. It conceived the behaviour of biological system composed of "Neurons". It enables machine learning of understanding pattern recognition. These are presented as interconnected Neurons which can compute value from input.

Here Neuron systems are connected through nodes. Our brain is consist of million of neurons. It sends and manage signal in form of electrical and chemical signals. These neurons are connected with synapses which allows Neurons to pass signal.

ANN includes large number of process units and these units works together to process the information. After that it generate meaningful results from it. Neural Network algorithm is used for data classification after careful training. It also used for pattern recognition.

2.2.1 Neural Network layers:

ANN consists of three layers: -

- Input layer
- Hidden layer
- Output layer

2.2.1.1 Input layers

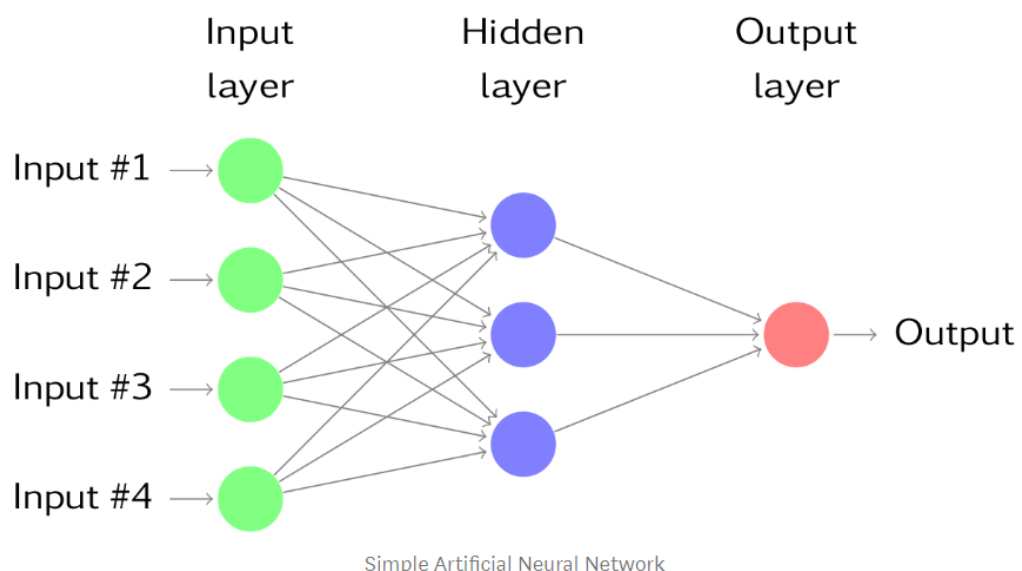
It is designed to receive input values of explanatory attributes for each observation. Generally, the number of input layer is equal to the explanatory variable. Here input layers are connected one or more hidden layers. Nodes of input layers do not change any data; they just receive the data and send across to all the hidden nodes.

2.2.1.2 Hidden layers

This layer decides the activity of hidden unit. The hidden layer transforms the input value inside hidden network. In hidden layer actual processing is done by weighted connections. There might be one or more hidden layers.

2.2.1.3 Output Layers

The hidden layers are linked with output layers. Output layers get connections from hidden layers or input layers. It returns an output value. In classification problem there is generally only one node. The active node of output layer changes the data to produce output values.

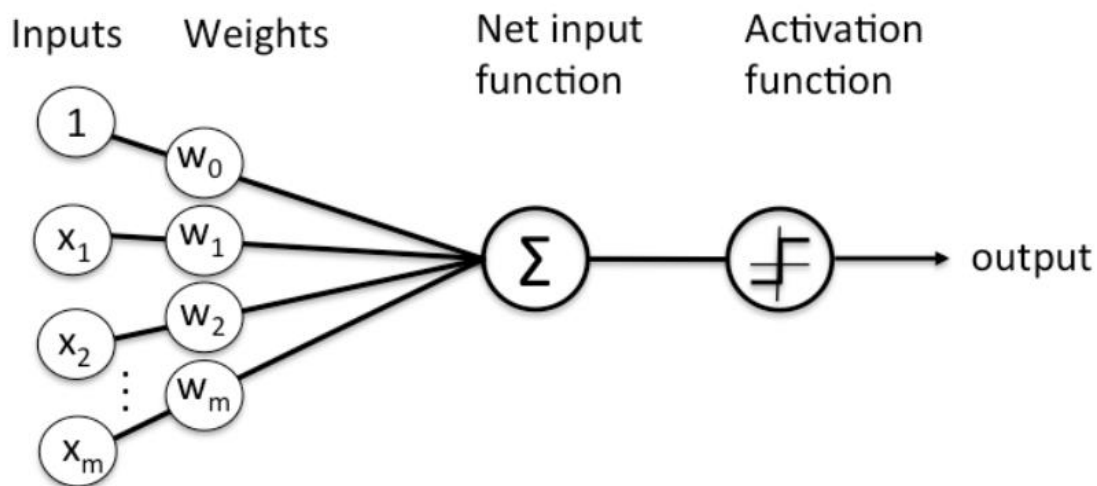


2.2.2 Neural Network Structure

Neural network consists of different layers and elementary units. We can simply break the structure into two layers: An input layer and an output layer. Each unit in input layer has a single input and single output.

There could be more than one output unit. By adding one hidden layer between input layers and output layers, it increases the predictive power of Neural Network model. But here hidden layer should be small as much as possible. Therefore, it ensures that it does not store any information in its node and generalise it to avoid overfitting.

Overfitting may occur sometime. This happen due to the small size of learning model in relation to the complexity of the model.



2.2.3 Advantages of Neural Network

Below are some advantages of Neural Network-

- The ability work with inadequate knowledge. After ANN training, the data may produce output without even complete information.
- It has fault tolerance as damaging one or more cell does not prevent it from generating output.
- It has parallel processing ability. Therefore, it can process one or more job at same time.
- ANN have distributed memory.
- It has ability to train machines.

2.2.4 Limitation of Neural Network

Below is some limitation of ANN model-

- Sometimes it's difficult to explain the function of ANN model. This is one of major problem as when it gives problem solution, it does not give a clue as to how and why.
- The duration of network is unknown.
- There is no assurance of proper network structure. The appropriate network structure is established by trial and error.
- For parallel data processing they need improved hardware system. So, it has hardware dependency.
- They are often referred as 'black-box' model as it provides very little insights how it functions. The user just needs to feed it and watch it train and wait for the output.

2.3 Support Vector Machine

The Support Vector Machine (SVM) is famous supervised classification method which can be used for both regression and classification. It works by making hyperplanes in a multidimensional space and then this hyperplane separates the sample data into different groups.

2.3.1 Hyperplane Classifier

Classification algorithms based on drawing lines to distinguish between objects of different class membership are known as hyperplane classifiers.

2.3.2 Optimal Hyperplane

An optimal hyperplane is one that makes the maximum distance from either of the classes. To make an optimal hyperplane, SVM employs an iterative training algorithm to maximize the distance margin around the separating hyperplane.

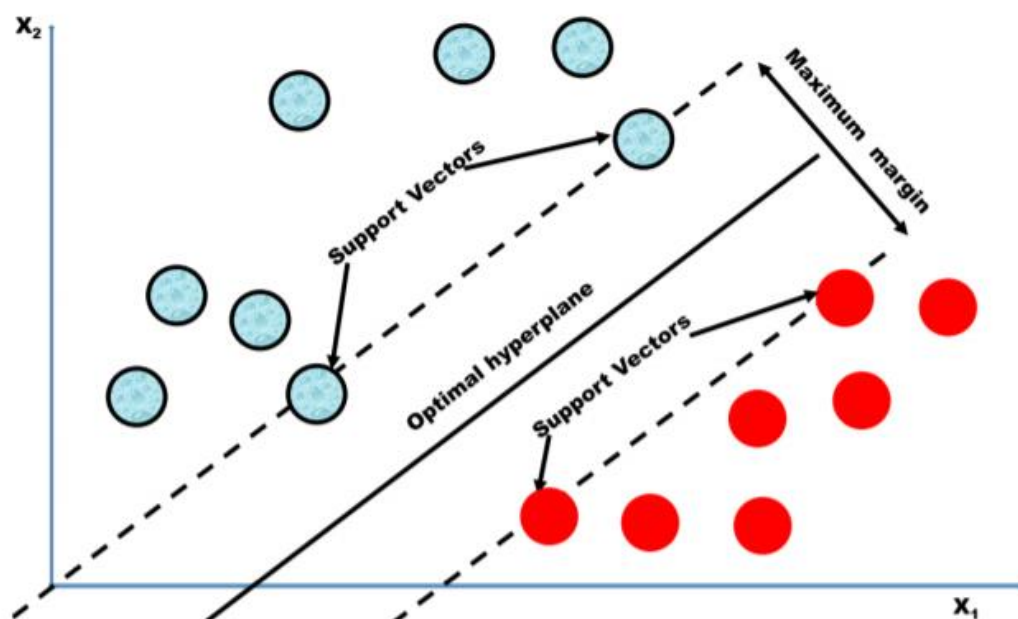


Fig- 2.4.1 Optimal Hyperplane

Hyperplanes are decision boundaries that classify the data points. Data points falling either side of the decision boundary can be regarded as separate classes. Moreover, the dimension of the hyperplane depends on the number of features. If the number of features is two then the hyperplane would be a simple line. If the number of input features is three then the hyperplane becomes a two-

dimensional model. However, it becomes hard to imagine when the number of features exceeds three.

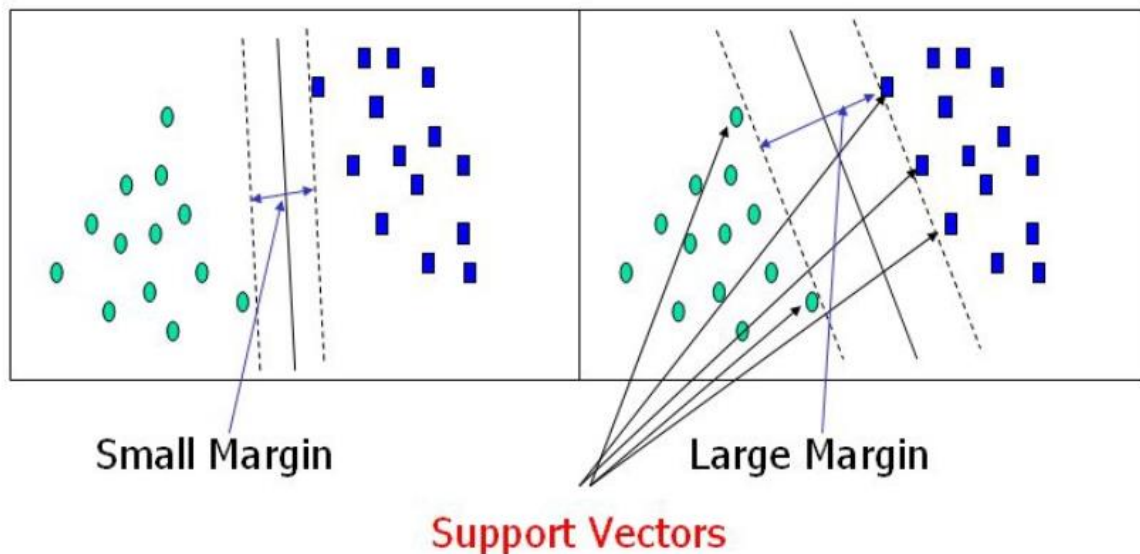


Fig.2.4.2- Support Vectors

2.3.3 Advantages of Support Vector Machine

Below are the advantages of Support Vector Machine (SVM)-

- This model comes with some theoretical guarantee regarding performance.
- This model does not suffer from high dimensionality.
- It is very popular method in text classification where high dimensional spaces are the norm
- Unlike neural network, they do not get trapped in local minima.

2.3.4 Limitation of Support Vector Machine

Below are some limitation of SVM model-

- Sometimes SVM model parameters can be difficult to interpret.
- They have high algorithm complexity and difficult to explain.
- SVM performance is very sensitive with the choice of cost parameter.
- SVM consume memory at high intense.
- It takes long time to train the data

2.4 Logistic Regression

Logistic regression is one of statistical technique that is adopted by Machine Learning. It is used to assign to observation to discrete set of classes. Unlike linear regression which only support

continuous numeric value, logistic regression transforms the output variable using sigmoid function to return a value which could be mapped two or more discrete class.

2.4.1 Types of Logistic Regression

There are three types of logistic regression-

- Binary
- Multi
- Ordinal

2.4.1.1 Binary Logistic Regression

When user wants to predict the value into two class, it is called binary logistic regression. Let's assume if you want to map the students if they passed or failed on given data.

To map the predicted values sigmoid function is used. This function maps the real value into 0 or 1.

Below is the equation-

$$S(z) = \frac{1}{1 + e^{-z}}$$

Where S(z) is the output between 0 and 1 (Probability estimation)

Z= input of the function

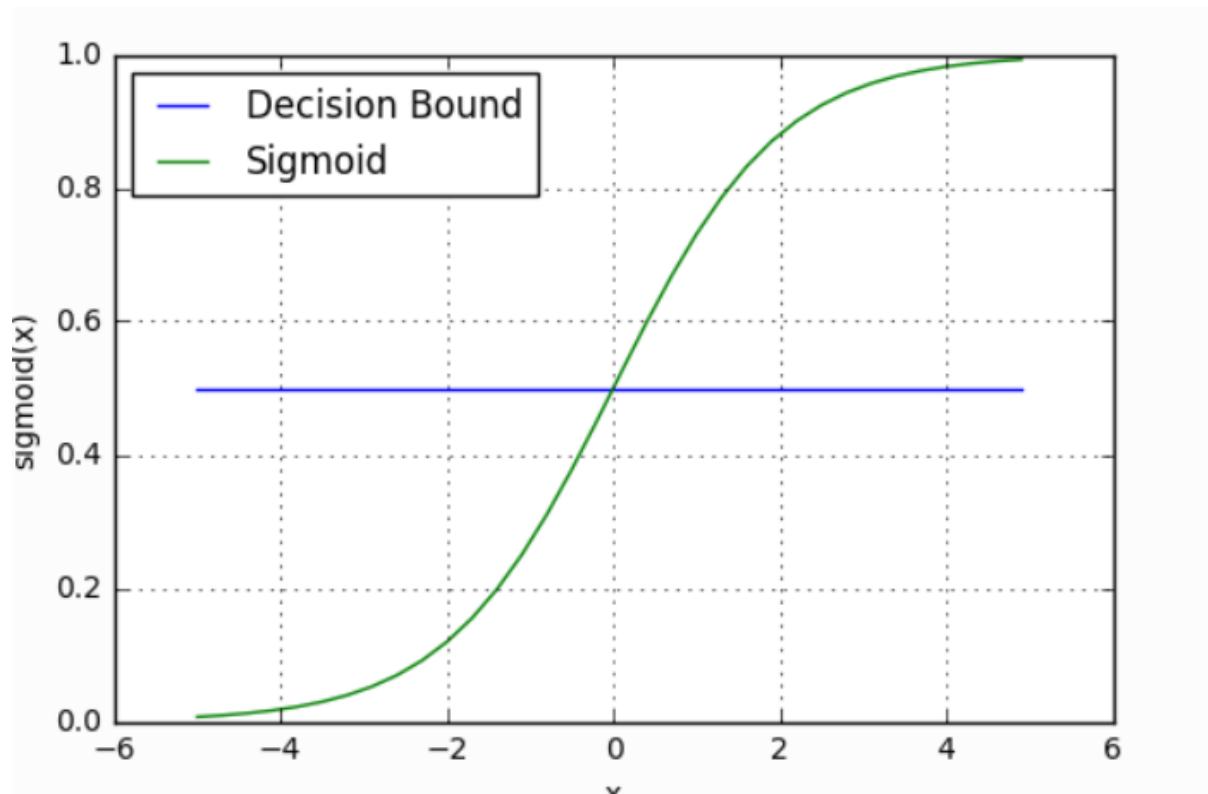
E= natural log

Decision Boundary

Here our current prediction function returns the value anything between 0 and 1. In order to make this discrete class we have to identify the threshold. So, the value above the threshold would be one class and below the value of threshold would be another class. So we may classify as below-

$p \geq 0.5$, class=1

$p < 0.5$, class=0



2.5.1 Fig- Decision Boundary

Making Predictions

Now using sigmoid function and decision boundary we can make prediction of class. The value more towards the decision boundary it would be 'true' else it would be 'false'. We can call the 'true' class as class 1 and it is denoted by $P(\text{class}=1)$.

2.4.1.2 Multiclass Logistic Regression

Given a binary classification algorithm there are two common approach for multi class regression: One Vs rest and One vs one. Each approach has pros and cons. However best approach depends on the dataset.

Here instead of $y=0,1$ we have to expand the definition so $y=0, 1, \dots, n$. So basically, we have to run binary classification multiple times for each class.

One drawback of one vs rest is, there are lot of classes, each binary classifier sees a highly imbalanced dataset which may impact the performance.

2.4.1.3 Ordinal Logistic Regression

Ordinal regression is used to predict the behaviour of dependent variables with a set of independent variables. The dependent variables are the order response category and on the other hand independent variable could be continuous or categorical variable.

2.5.2 Advantages of Logistic Regression

Below are some advantages of logistic regression-

- Logistic regression is more robust as the independent variable don't have to be normally distributed or need to have equal variance in each group.
- It can handle non-linear effects
- Here independent variables do not need to have interval
- There is no homogeneity about the variable assumption.
- User can add explicit power terms
- Independent variables do not need to be unbounded.

2.5.3 Limitation of Logistic Regression

Here below are some limitation of logistic regression-

- Logistic regression basically based on identifying independent variable but in case researcher/ user identify the wrong independent variable then model will not have predictive value.
- Logistic regression model works well with categorical outcome and also with multinomial outcomes. However logistic regression cannot work with continuous outcomes.
- Logistic regression requires each data points need to be independent with respect to other data points. If observations are related with each other then the model tend to overweight the significance of those observation. This is one of the major drawbacks as many researchers looks for multiple observation of same individual.
- Logistic regression is vulnerable to overfitting.

2.6 Project Workflow

Below we have made the project workflow chart which we can refer while implementing the methodology. The project is classified into different stage and each stage is elaborated briefly.

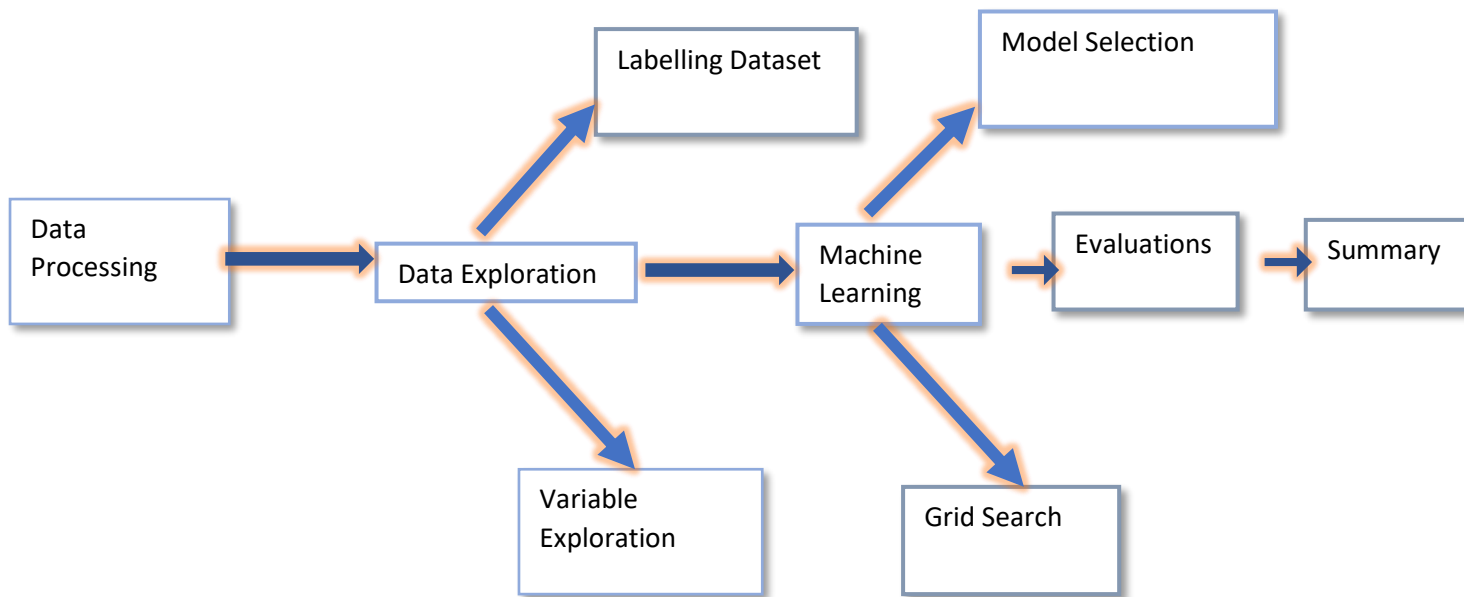


FIG. 2.6 Project Workflow

Below we have discussed each step involved into this process.

2.6.1 Data Processing

In this stage we need to identify main tables and understand the dataset. If we have multiple tables and columns then we may need to drop some unnecessary columns. Also, we need to upload the data set into Database server or the tool that may be useful for cleaning the dataset. Here I will be using MS server as database system and will be using R and Python for cleaning the dataset.

2.6.2 Data Exploration

In this stage we need to explore the variable and identify the dependent and independent / target variable by analysing the dataset. Here target variable needs to split into Binary format as we want to categories the data (as default or not default).

We also can use the other variable for analysing the dataset and can be useful for mapping the dataset. Therefore, it will give us clear understanding about the dataset.

2.6.3 Model Selection

Based on dataset we need to choose the correct supervised algorithm. I will be using ROC Curve for shortlisting the algorithm for implementing into given dataset. Here all the supervised model may have some limitation considering given dataset. However, we need to use the correct algorithm that may have maximum accuracy and not over-fitting with the dataset.

2.6.4 Evaluation

Based on correct algorithm we will evaluate the dataset. For that purpose, we need to split the data into training dataset and test dataset. We can also measure the variable importance individually by plotting the bar chart.

We can also take few variables from the variable importance chart to visualize the decision boundaries. Two variables are needed as a pair each time to plot the boundaries.

2.6.5 Summary

Here we can draw our conclusion based on our analysis. Also, we can put suggestion across to overcome the observed limitation in order to get better accuracy in future.

Chapter III

IMPLEMENTATION

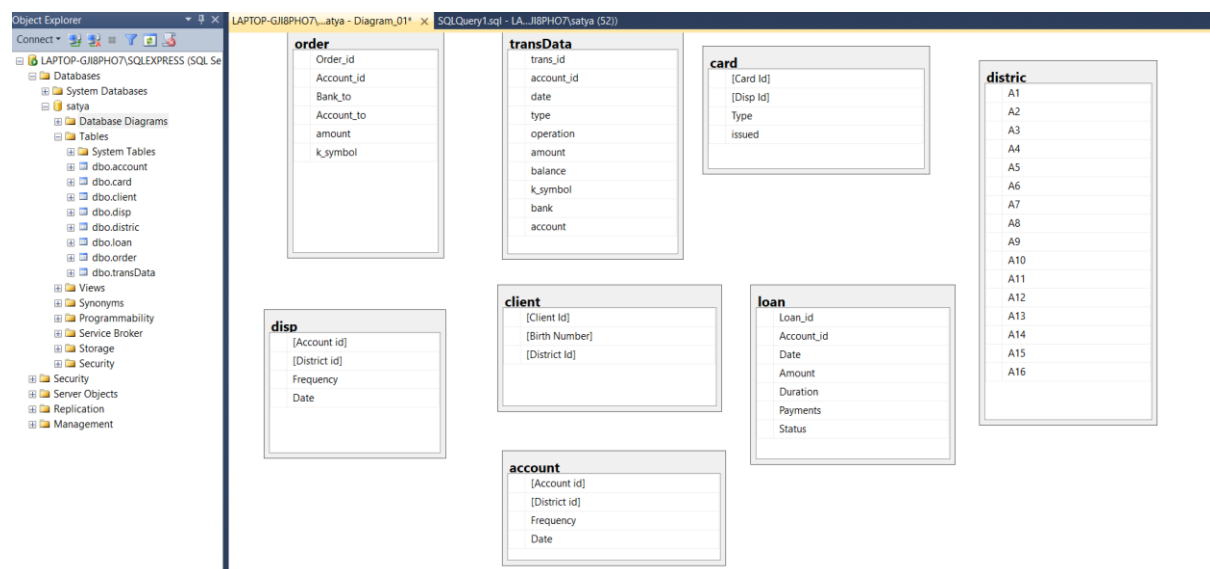
3.1 About Dataset

Dataset Name- Bank Loan Data Default

Link-<https://www.kaggle.com/zhunqiang/bank-loan-default-prediction/data>

Here dataset contains several tables with plenty of information about the different bank accounts of the customers such as loans, transactions and credit cards.

The main purpose is to predict customer behaviour about loan for each account. Therefore, the most important table is “loan”. Also, after careful analysis I believe “order”, “trans” and “card” contain useful information for my purpose. Also, I need to use “Account” and disposition to combine them together. I have made database diagram consisting all the tables.

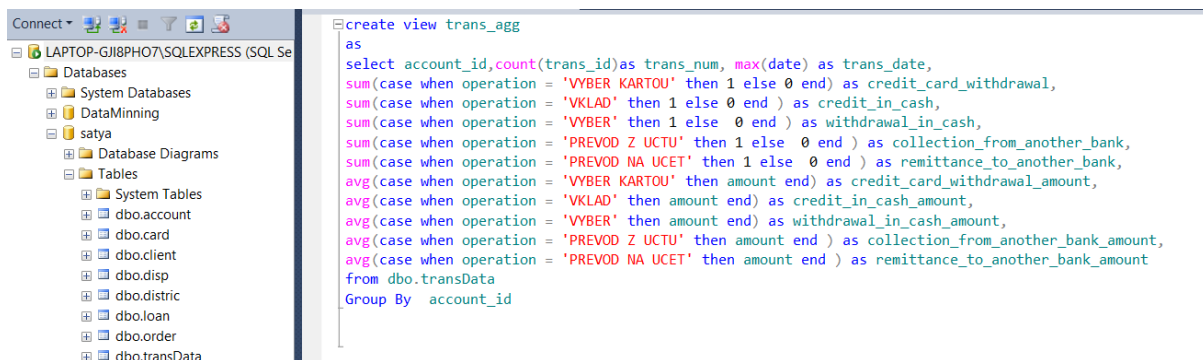


The columns “status” in table “loan” is target variable here, which stands for the customers’ loan behaviours. Then I should select useful features for the model. The standard is based on relevant to our target and some business sense. For example, the

columns “bank to” and “account to” have nothing to do with loan behaviours, so I drop them.

3.2 Data Preparation

Here many are categorical variable which needs to convert into numeric variable. Usually it is part of feature engineering and can be completed using python. But I need to use some aggregate function which can be done using MS SQL Server and thus cannot be applied on categorical variable. So, I conducted this step in SQL Server. Below is the query which I have used –



```
create view trans_agg
as
select account_id, count(trans_id) as trans_num, max(date) as trans_date,
sum(case when operation = 'VYBER KARTOU' then 1 else 0 end) as credit_card_withdrawal,
sum(case when operation = 'VKLAD' then 1 else 0 end) as credit_in_cash,
sum(case when operation = 'VYBER' then 1 else 0 end) as withdrawal_in_cash,
sum(case when operation = 'PREVOD Z UCTU' then 1 else 0 end) as collection_from_another_bank,
sum(case when operation = 'PREVOD NA UCET' then 1 else 0 end) as remittance_to_another_bank,
avg(case when operation = 'VYBER KARTOU' then amount end) as credit_card_withdrawal_amount,
avg(case when operation = 'VKLAD' then amount end) as credit_in_cash_amount,
avg(case when operation = 'VYBER' then amount end) as withdrawal_in_cash_amount,
avg(case when operation = 'PREVOD Z UCTU' then amount end) as collection_from_another_bank_amount,
avg(case when operation = 'PREVOD NA UCET' then amount end) as remittance_to_another_bank_amount
from dbo.transData
Group By account_id
```

After that feature engineering and model fitting need to be done using Python. I have connected SQL Server with python and then imported the tables using Pandas data frame. Below is the query-

```
import pyodbc
conn = pyodbc.connect('Driver={SQL Server};'
                      'Server=LAPTOP-GJI8PH07\SQLEXPRESS;'
                      'Database=satya;'
                      'Trusted_Connection=yes;')

cursor = conn.cursor()
cursor.execute('select * from dbo.trans_agg')

for row in cursor:
    print(row)
```

Later on, I found there is some missing value in “Loan Monthly Payment” that means customer did not make the loan payment monthly. Therefore, I replaced the missing value with Zero instead of mean or median.

Below query I am using to update and imports different tables into Python-

```
# import and update table of account
account = pd.read_csv(
    "S:\Project_Thesis\Data\account.asc",
    sep=";",
    delimiter=None,
    header="infer",
    names=None,
)

account.date = account.date.apply(lambda x: pd.to_datetime(str(x), format="%y%m%d"))
account.head()
```

```
# import and update table loan
loan = pd.read_csv(
    "S:\Project_Thesis\Data\loan.asc",
    sep=";",
    delimiter=None,
    header="infer",
    names=None,
    low_memory=False,
)
loan.date = loan.date.apply(lambda x: pd.to_datetime(str(x), format="%y%m%d"))
loan.head()
```

3.3 Data Labelling

In the original dataset Loan table and under the status column there are four categories. Where,

A: -Stands for those customers who finished contract and no outstanding amount is pending.

B: - Stands for those customers who finished contracts but did not pay the loan.

C: - Stands for those customers who are still paying the amount and found no issue.

D: -Stands for those customers who are in debt and not able to pay the amount.

Instead of multiclassification in supervised model, a binary model would be useful to predict if a customer is default or not. Consequently, I have marked A and C as "0", they are not default and B and D as "1" which represents the default loan. This is how I have converted categorical value into numeric value.

Below is the query I used for above purpose-

```
# import and update table card
loan = pd.read_csv(
    "S:\Project_Thesis\Data\loan.asc",
    sep=";",
    delimiter=None,
    header="infer",
    names=None,
)

loan.status = loan.status.map({"B": 1, "D": 1, "A": 0, "C": 0})
loan.head()
```

	loan_id	account_id	date	amount	duration	payments	status
0	5314	1787	930705	96396	12	8033.0	1
1	5316	1801	930711	165960	36	4610.0	0
2	6863	9188	930728	127080	60	2118.0	0
3	5325	1843	930803	105804	36	2939.0	0
4	7240	11013	930906	274740	60	4579.0	0

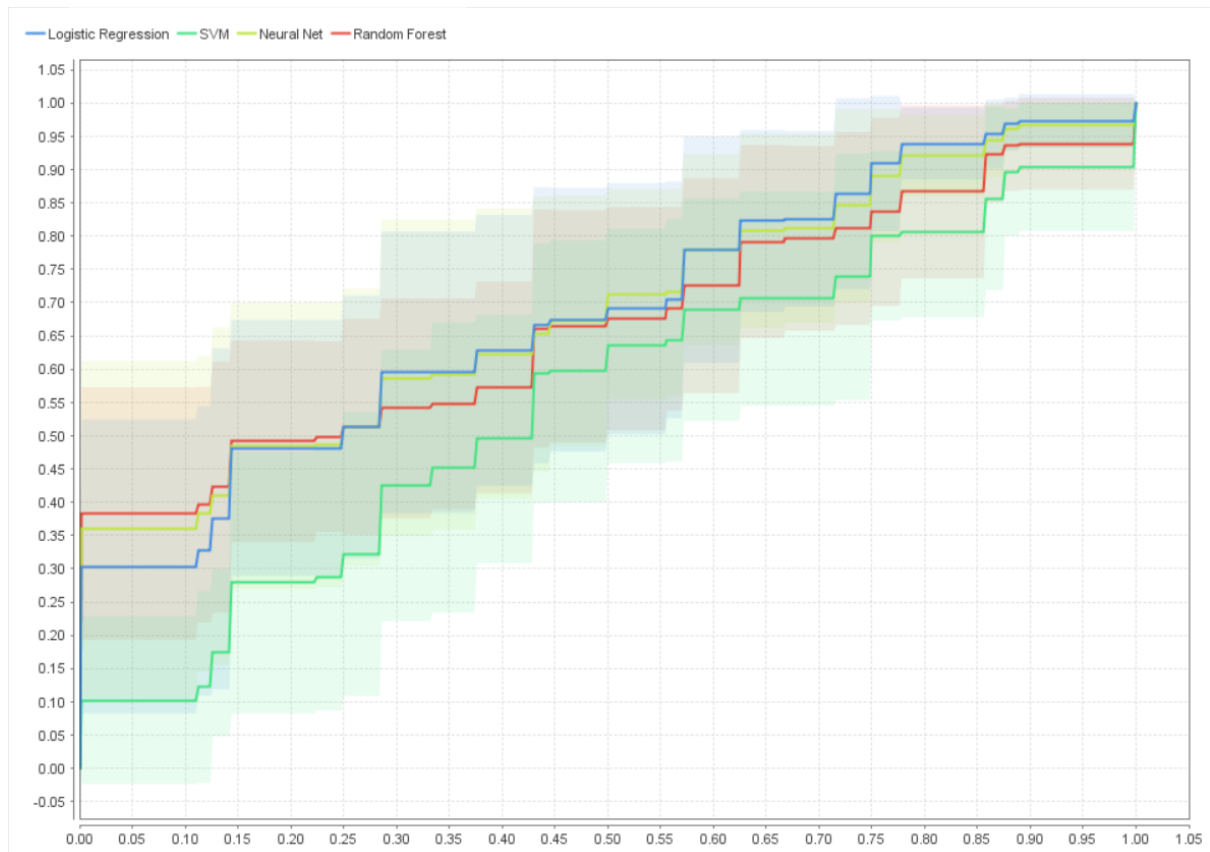
Later on, I have exported the data into my local system for further analysis

```
##export the data in csv format
export_csv = loan.to_csv (r'S:\Project_Thesis\Jup\export_loan.csv', index = None, header=True)
```


3.4 ROC Curve

Here we have used ROC Curves to get overview about the suitable models. Out of all the supervised models we will be using the algorithm which would provide high accuracy and low fixed error. We need Rapid Miner tool for above purpose.

Higher the AOC, better the model is. Here predicting value is limited 0 to 1.



As we can see on above ROC curve diagram that logistic regression and Neural Network model would have higher accuracy based on my given data. We may verify the same after executing the algorithm in Python.

3.5 Model Evaluation

3.5.1 Drafting Model: Random Forest

```
: ##import module
import pandas as pd
from sklearn.datasets import load_digits
df1 = pd.read_csv("S:\Project_Thesis\Jup\export_loan.csv")
df1.head()
```

	loan_id	account_id	date	amount	duration	payments	status
0	5314	1787	930705	96396	12	8033.0	1
1	5316	1801	930711	165960	36	4610.0	0
2	6863	9188	930728	127080	60	2118.0	0
3	5325	1843	930803	105804	36	2939.0	0
4	7240	11013	930906	274740	60	4579.0	0

```
: # split the data into X and Y axis
from sklearn.model_selection import train_test_split

X=df1[['loan_id', 'account_id', 'payments']] # Features
y=df1['status'] # Labels

# Split dataset into training set and test set
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3) # 70% tr
```

```
: ## Train model on training set
from sklearn.ensemble import RandomForestClassifier
model= RandomForestClassifier (n_estimators=10)
model.fit(X_train,y_train)
```

```

## Train model on training set
from sklearn.ensemble import RandomForestClassifier
model= RandomForestClassifier (n_estimators=10)
model.fit(X_train,y_train)

RandomForestClassifier(bootstrap=True, class_weight=None, criterion='gini',
                        max_depth=None, max_features='auto', max_leaf_nodes=None,
                        min_impurity_decrease=0.0, min_impurity_split=None,
                        min_samples_leaf=1, min_samples_split=2,
                        min_weight_fraction_leaf=0.0, n_estimators=10, n_jobs=1,
                        oob_score=False, random_state=None, verbose=0,
                        warm_start=False)

model.score(X_test,y_test)

0.8731707317073171

y_predicted = model.predict(X_test)

from sklearn.metrics import confusion_matrix
cm= confusion_matrix (y_test, y_predicted)
cm

array([[178,   7],
       [ 19,   1]], dtype=int64)

```

3.5.2 Drafting Model: Artificial Neural Network

In [68]: *## Using Neural Network Algorith*

```

import numpy
import pandas as pd

```

In [65]: *##import the dataset*

```

df1 = pd.read_csv("S:\Project_Thesis\Jup\export_loan.csv")
df1.head()

```

Out[65]:

	loan_id	account_id	date	amount	duration	payments	status
0	5314	1787	930705	96396	12	8033.0	1
1	5316	1801	930711	165960	36	4610.0	0
2	6863	9188	930728	127080	60	2118.0	0
3	5325	1843	930803	105804	36	2939.0	0
4	7240	11013	930906	274740	60	4579.0	0

In [66]: *## import keras*

```

from keras.models import Sequential
from keras.layers import Dense

```

In [69]: *##random seed for reproducibility*

```

numpy.random.seed(7)

```

```
In [4]: ## split the dataset into train and test dataset
from sklearn.model_selection import train_test_split
X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size = 0.3, random_state = 0)
```

```
In [5]: ##Feature Scaling
##Feature Scaling

from sklearn.preprocessing import StandardScaler
sc = StandardScaler()
X_train = sc.fit_transform(X_train)
X_test = sc.transform(X_test)
```

```
In [7]: import warnings
warnings.filterwarnings("ignore", category=FutureWarning)
```

```
In [8]: ##Import Keras Library
import keras
from keras.models import Sequential
from keras.layers import Dense
```

```
In [17]: ##create model,add dense layer one by one specifying activation function
model = Sequential()
model.add (Dense(12,input_dim =6, activation='relu')) ## Input Layer required input
model.add (Dense(8,activation = 'relu'))
model.add (Dense(1,activation = 'sigmoid'))## sigmoid instead of relu for final prediction
```

```
In [18]: ##compiling neural network
model.compile(loss="binary_crossentropy", optimizer= "adam", metrics = ['accuracy'])
```

```
In [19]: ## call the function to fit with the dataset (training the network)
model.fit (X_train,Y_train, epochs = 1000, batch_size = 10 )
```

```
Epoch 1/1000
477/477 [=====] - 3s 6ms/step - loss: 0.5537 - acc: 0.8407
Epoch 2/1000
477/477 [=====] - 0s 147us/step - loss: 0.4503 - acc: 0.8931
Epoch 3/1000
477/477 [=====] - 0s 174us/step - loss: 0.3882 - acc: 0.8952
Epoch 4/1000
477/477 [=====] - 0s 165us/step - loss: 0.3538 - acc: 0.8952
Epoch 5/1000
477/477 [=====] - 0s 163us/step - loss: 0.3357 - acc: 0.8952
Epoch 6/1000
477/477 [=====] - 0s 154us/step - loss: 0.3252 - acc: 0.8952
Epoch 7/1000
477/477 [=====] - 0s 179us/step - loss: 0.3185 - acc: 0.8952
Epoch 8/1000
477/477 [=====] - 0s 156us/step - loss: 0.3124 - acc: 0.8952
Epoch 9/1000
477/477 [=====] - 0s 160us/step - loss: 0.3096 - acc: 0.8952
Epoch 10/1000
477/477 [=====] - 0s 155us/step - loss: 0.3060 - acc: 0.8952
```

```
In [21]: ##predicting the Test set results
Y_pred=model.predict (X_test)
Y_pred=(Y_pred > 0.5)
```

```
In [30]: ## Creating the confusin matrix
from sklearn import metrics
cnf_matrix = metrics.confusion_matrix(Y_test, Y_pred)
```

```
In [31]: print(cnf_matrix)

[[165  14]
 [ 23   3]]
```

```
In [32]: print("Accuracy:",metrics.accuracy_score(Y_test, Y_pred))
print("Precision:",metrics.precision_score(Y_test, Y_pred))
print("Recall:",metrics.recall_score(Y_test, Y_pred))

Accuracy: 0.8195121951219512
Precision: 0.17647058823529413
Recall: 0.11538461538461539
```

3.5.3 Drafting Model: SVM

```
In [55]: ##Importing Libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
```

```
In [2]: ##importing the Dataset
df1 = pd.read_csv("S:\Project_Thesis\Jup\export_loan.csv")
df1.head()
```

```
Out[2]:
```

	loan_id	account_id	date	amount	duration	payments	status
0	5314	1787	930705	96396	12	8033.0	1
1	5316	1801	930711	165960	36	4610.0	0
2	6863	9188	930728	127080	60	2118.0	0
3	5325	1843	930803	105804	36	2939.0	0
4	7240	11013	930906	274740	60	4579.0	0

```
In [3]: ##Exploratory data analysis
df1.shape
```

```
Out[3]: (682, 7)
```

```
In [12]: ##Dividing the dataset into training and test dataset
X=df1[['loan_id', 'account_id', 'payments']] # Features
y=df1['status'] # Labels
```

```
In [12]: ##Dividing the dataset into training and test dataset
X=df1[['loan_id', 'account_id', 'payments']] # Features
y=df1['status'] # Labels
```

```
In [13]: from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.20) ##splitting dataset into 80 (train) and 20(test set)
```

```
In [14]: ##Training the algorithm
from sklearn.svm import SVC
svclassifier = SVC(kernel='linear')
svclassifier.fit(X_train, y_train)
```

```
Out[14]: SVC(C=1.0, cache_size=200, class_weight=None, coef0=0.0,
decision_function_shape='ovr', degree=3, gamma='auto', kernel='linear',
max_iter=1, probability=False, random_state=None, shrinking=True,
tol=0.001, verbose=False)
```

```
In [15]: ##making prediction
y_pred = svclassifier.predict(X_test)
```

```
In [16]: ##Evaluating the algorithm
from sklearn.metrics import classification_report, confusion_matrix
print(confusion_matrix(y_test,y_pred))
print(classification_report(y_test,y_pred))
```

```
[[120  0]
 [ 17  0]]
```

```
##Evaluating the algorithm
from sklearn.metrics import classification_report, confusion_matrix
print(confusion_matrix(y_test,y_pred))
print(classification_report(y_test,y_pred))
```

```
[[120  0]
 [ 17  0]]
```

	precision	recall	f1-score	support
0	0.88	1.00	0.93	120
1	0.00	0.00	0.00	17
avg / total	0.77	0.88	0.82	137

```
C:\Users\satya\Anaconda3\lib\site-packages\sklearn\metrics\classification.py:1135: UndefinedMetricWarning: Precision and F-score are ill-defined and being set to 0.0 in labels with no predicted samples.
'precision', 'predicted', average, warn_for)
```

3.5.4 Drafting Model: Logistic Regression

```
In [19]: ##Classification Using Logistic Regression
##import Library and dataset
import pandas as pd
```

```
In [20]: ##importing the Dataset
df1 = pd.read_csv("S:\Project_Thesis\Jup\export_loan.csv")
df1.head()
```

```
Out[20]:
```

	loan_id	account_id	date	amount	duration	payments	status
0	5314	1787	930705	96396	12	8033.0	1
1	5316	1801	930711	165960	36	4610.0	0
2	6863	9188	930728	127080	60	2118.0	0
3	5325	1843	930803	105804	36	2939.0	0
4	7240	11013	930906	274740	60	4579.0	0

```
In [21]: ##Explotary analysis
df1.shape
```

```
Out[21]: (682, 7)
```

```
In [23]: ##splitting the dataset into feature and variable
feature_cols = ['loan_id', 'account_id', 'date', 'amount', 'duration', 'payments']
X = df1[feature_cols] # Features
y = df1.status # Target variable
```

```
In [24]: ##splitting the data
## # split X and y into training and testing sets
from sklearn.cross_validation import train_test_split
X_train,X_test,y_train,y_test=train_test_split(X,y,test_size=0.30,random_state=0)
```

C:\Users\satya\Anaconda3\lib\site-packages\sklearn\cross_validation.py:41: DeprecationWarning: This module was deprecated in version 0.18 in favor of the model_selection module into which all the refactored classes and functions are moved. Also note that the interface of the new CV iterators are different from that of this module. This module will be removed in 0.20.
"This module will be removed in 0.20.", DeprecationWarning)

```
In [25]: ##Model development and predictions
### import the class
from sklearn.linear_model import LogisticRegression
# instantiate the model (using the default parameters)
logreg = LogisticRegression()
# fit the model with data
logreg.fit(X_train,y_train)
#
y_pred=logreg.predict(X_test)
```

```
In [26]: ##Model Evaluation using confusion matrix
# import the metrics class
from sklearn import metrics
cnf_matrix = metrics.confusion_matrix(y_test, y_pred)
cnf_matrix
```

```
Out[26]: array([[179,    0],
               [ 26,    0]], dtype=int64)
```

```
Out[26]: array([[179,    0],
               [ 26,    0]], dtype=int64)
```

```
In [27]: print(classification_report(y_test,y_pred))
```

	precision	recall	f1-score	support
0	0.87	1.00	0.93	179
1	0.00	0.00	0.00	26
avg / total	0.76	0.87	0.81	205

C:\Users\satya\Anaconda3\lib\site-packages\sklearn\metrics\classification.py:1135: UndefinedMetricWarning: Precision and F-score are ill-defined and being set to 0.0 in labels with no predicted samples.
'precision', 'predicted', average, warn_for)

```
In [ ]:
```

Chapter IV

4.Result and Discussion

4.1 Visualizing Confusion Matrix Using Heatmap

We can now visualize the confusion matrix of each algorithm using Matplotlib and seaborn. Therefore, we have created confusion matrix using Heatmap.

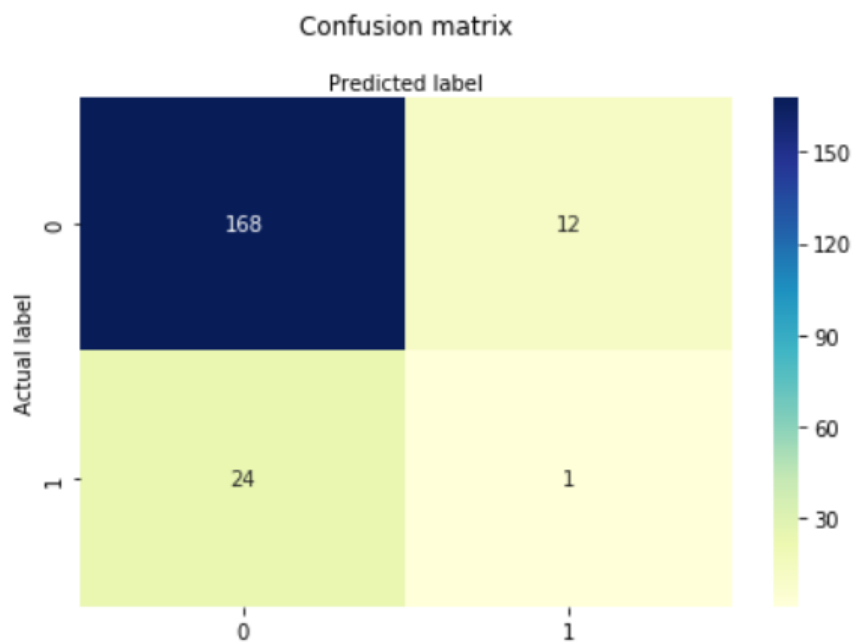


Fig: Random Forest Classification Confusion Matrix

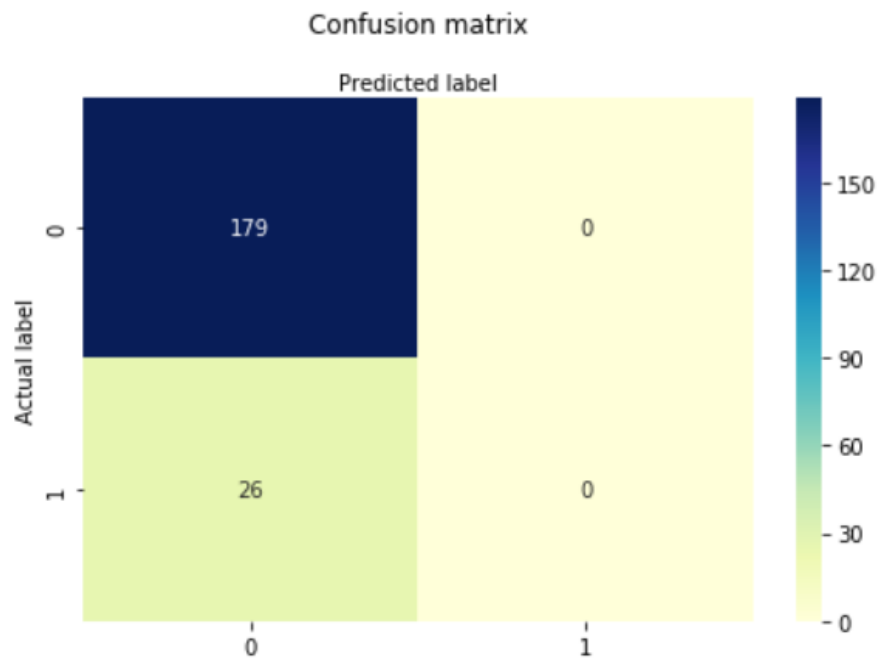
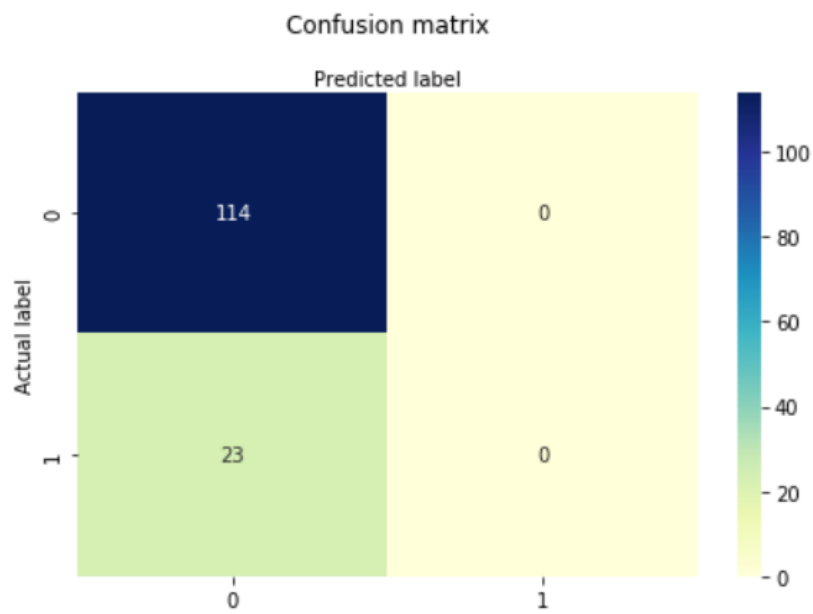


Fig: Logistic Regression Confusion Matrix

Fig: SVM Confusion Matrix



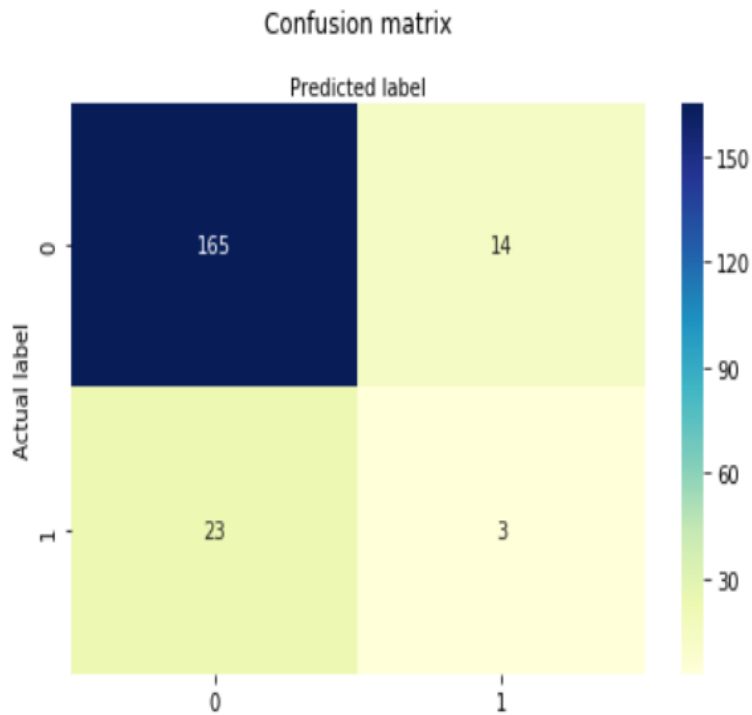


Fig: ANN Confusion Matrix

4.2 Machine Learning algorithm comparison

Here we have made comparison among the four algorithms on same datasets. We have used binary classification of target variable.

10-K cross validation is used to evaluate each algorithm and configured with same random seeds. So, the same splits are performed on the datasets and each algorithm is evaluated same way.

Below codes have been used-

```
In [4]: ##Load Dataset
df1 = pd.read_csv("S:\Project_Thesis\Jup\export_loan.csv")
df1.head()
```

```
Out[4]:
```

	loan_id	account_id	date	amount	duration	payments	status
0	5314	1787	930705	96396	12	8033.0	1
1	5316	1801	930711	165960	36	4610.0	0
2	6863	9188	930728	127080	60	2118.0	0
3	5325	1843	930803	105804	36	2939.0	0
4	7240	11013	930906	274740	60	4579.0	0

```
In [5]: #split dataset in features and target variable
X=df1[['loan_id', 'account_id', 'payments']] # Features
y=df1['status'] # Labels
```

```
In [6]: #Split into train and test set using sklearn model_selection
from sklearn.model_selection import train_test_split
```

```
In [8]: X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.30, random_state=7)
```

```
In [9]: # Feature Scaling
from sklearn.preprocessing import StandardScaler
sc = StandardScaler()
X_train = sc.fit_transform(X_train)
X_test = sc.transform(X_test)
```

```
In [15]: from sklearn.metrics import classification_report
from sklearn.metrics import confusion_matrix
from sklearn.metrics import accuracy_score
from keras.models import Sequential
from sklearn.linear_model import LogisticRegression
from sklearn.ensemble import RandomForestClassifier
from sklearn.svm import SVC
from sklearn.neural_network import MLPClassifier
##mlp= MLPClassifier(hidden_layer_sizes=(10,10,10), max_iter= 1000)
from sklearn.model_selection import cross_val_score
from sklearn.model_selection import KFold
algorithms = []
algorithms.append(('LR', LogisticRegression()))
algorithms.append(('RFC', RandomForestClassifier()))
algorithms.append(('SVM', SVC()))
algorithms.append(('ANN', MLPClassifier()))
```

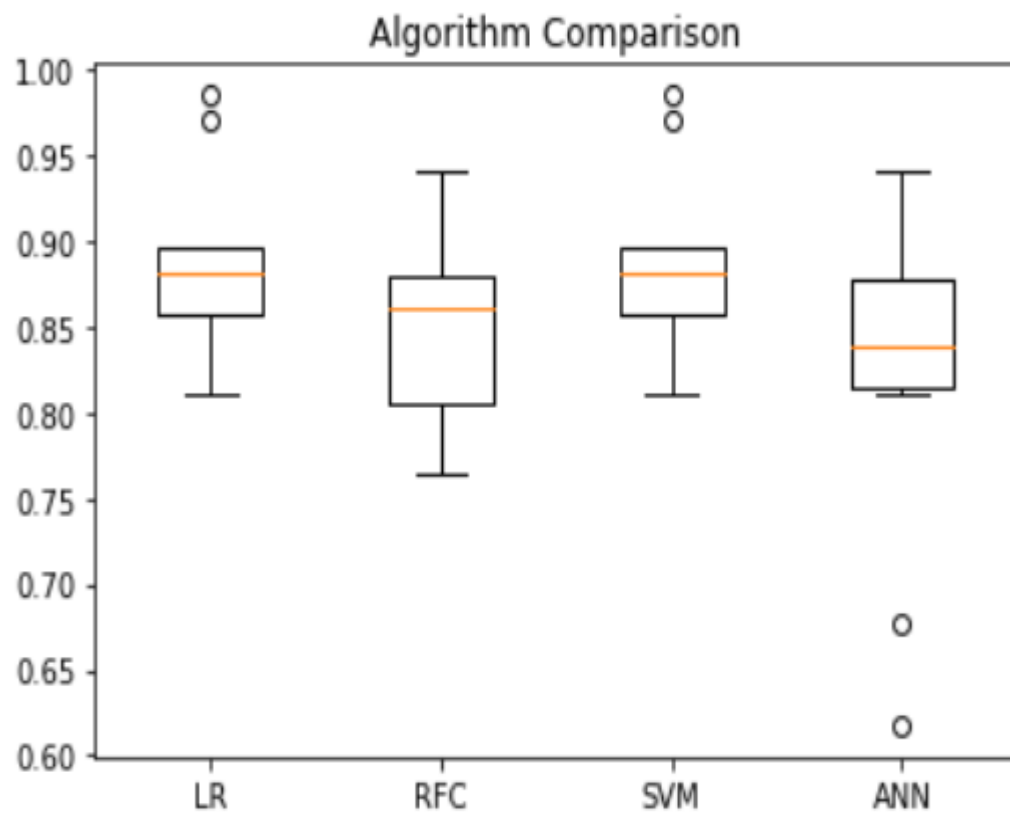
```
In [10]: import warnings
warnings.filterwarnings("ignore", category=FutureWarning)
```

```
In [16]: results = []
names = []
scoring = 'accuracy'
for name, algorithm in algorithms:
    k_fold_validation = KFold(n_splits=10, random_state=1)
    cv_results =cross_val_score(algorithm, X, y, cv=k_fold_validation, scoring='accuracy')
    results.append(cv_results)
    names.append(name)
    msg = "%s| Mean=%f STD=%f" % (name, cv_results.mean(), cv_results.std())
    print(msg)
```

```
In [16]: results = []
names = []
scoring = 'accuracy'
for name, algorithm in algorithms:
    k_fold_validation = KFold(n_splits=10, random_state=1)
    cv_results =cross_val_score(algorithm, X, y, cv=k_fold_validation, scoring='accuracy')
    results.append(cv_results)
    names.append(name)
    msg = "%s| Mean=%f STD=%f" % (name, cv_results.mean(), cv_results.std())
    print(msg)

LR| Mean=0.888725 STD=0.051725
RFC| Mean=0.850512 STD=0.053808
SVM| Mean=0.888725 STD=0.051725
ANN| Mean=0.819629 STD=0.094427
```

```
In [17]: from matplotlib import pyplot
pyplot.boxplot(results,labels=names)
pyplot.title('Algorithm Comparison')
pyplot.show()
```



From this result it is obvious that Logistic Regression and SVM are the perhaps worthy for further study on this problem.

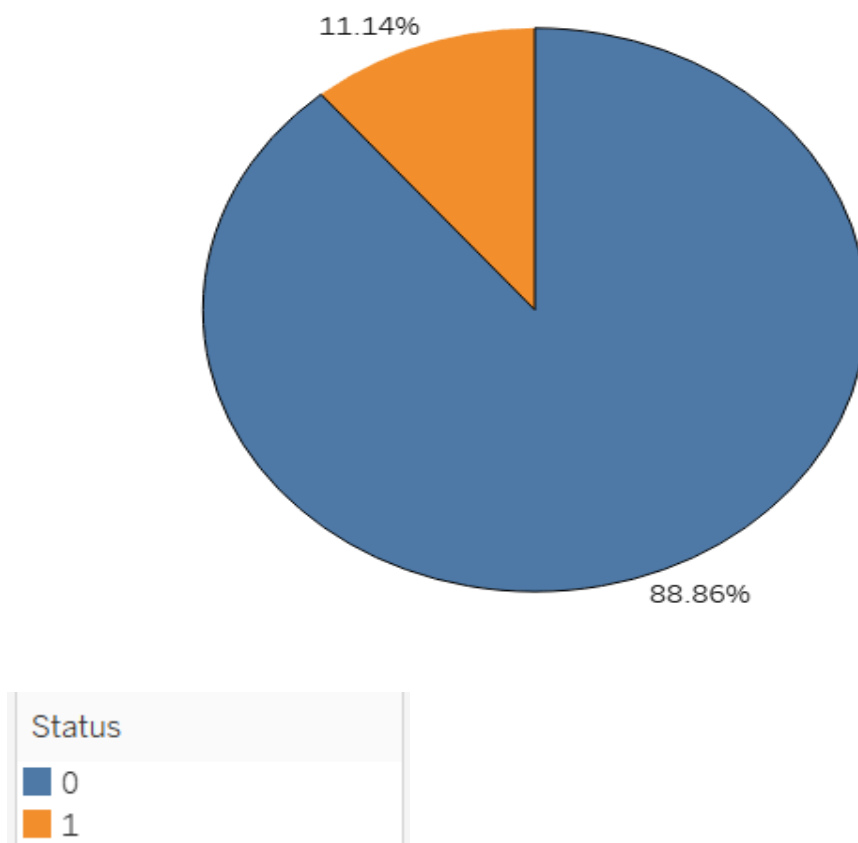
We know that the selection of the best algorithm depends upon or ability to interpret the dataset and analysis however. But being able to compare these algorithm gives us view of what could be the best algorithm to use.

4.3 Variable Exploration

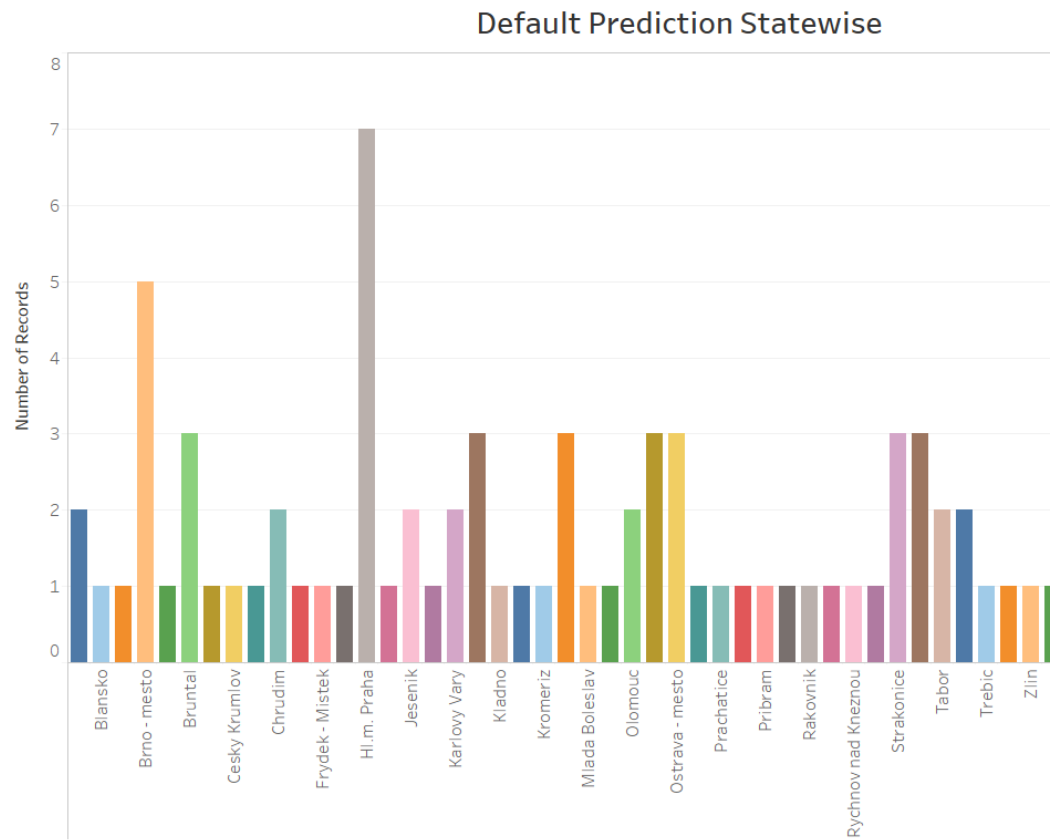
4.3.1 Loan Status Proportion

Below is the total number of defaulters against the non- default customer. There are around 11.143 % customers are labelled as default. We have depicted the same using Pie Chart

Loan Status Proportion



4.3.2 Loan Default District Wise

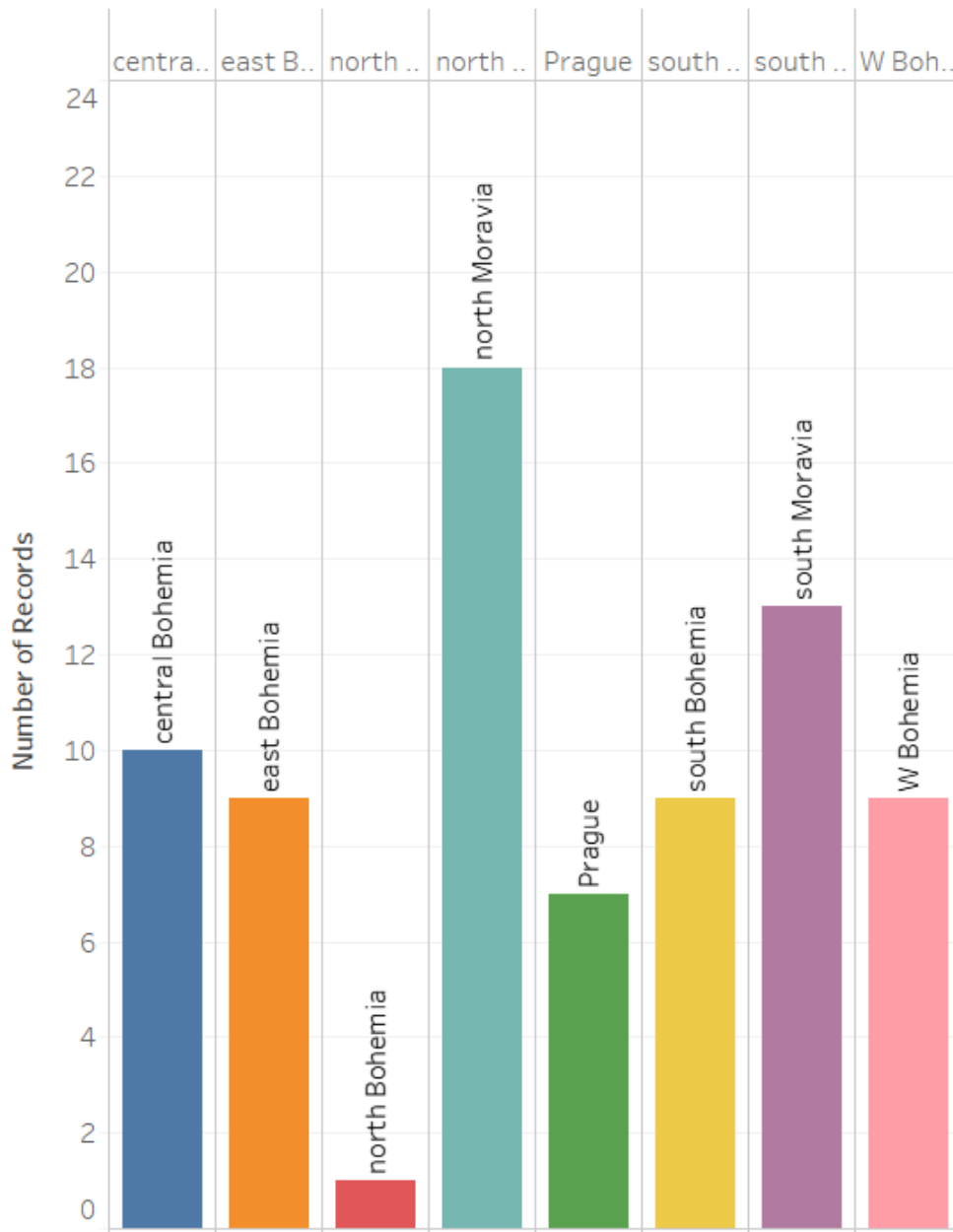


We can see from above bar graph that H.M.Praha district is one which have maximum number of default customers belongs. So, the graph shows the distribution of bank loan defaulters district wise.

Also Brno-mesto is second largest district which have most number of defaulters.

4.3.3 Loan Default Region Wise

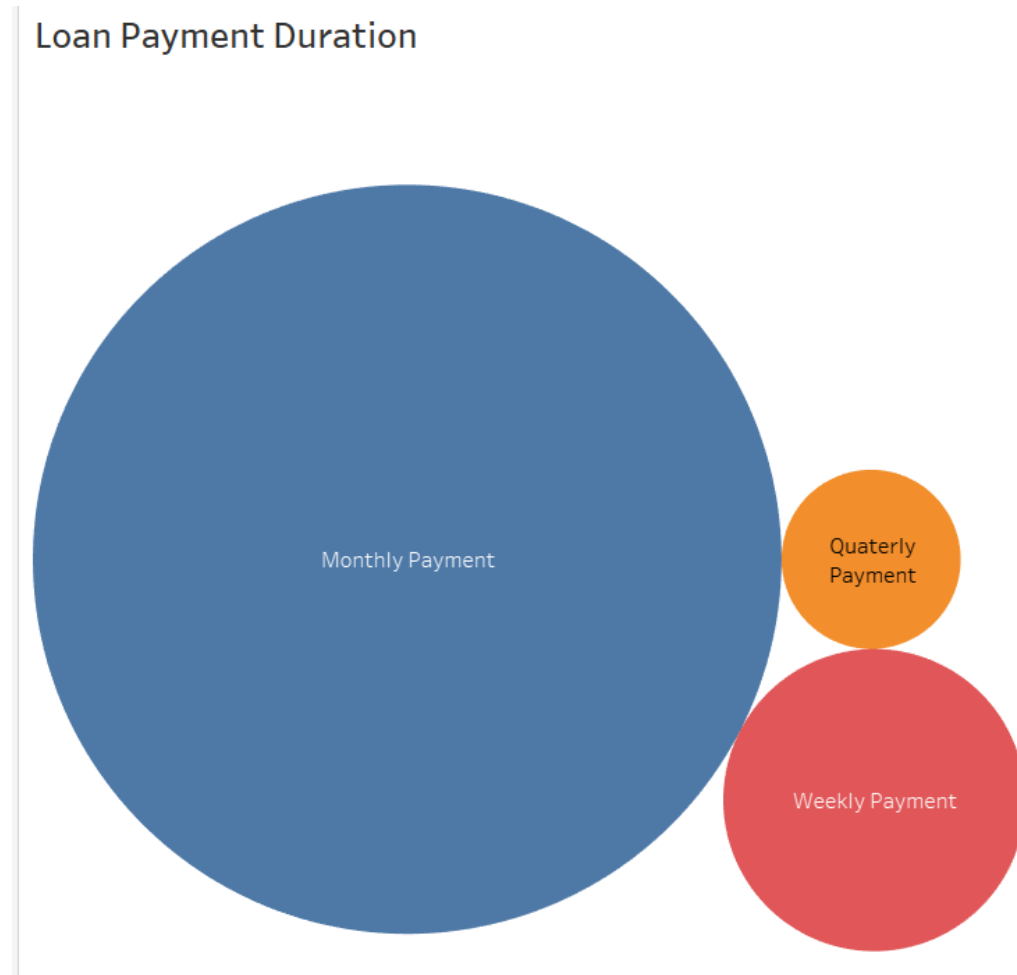
Loan Default Region Wise



We can see that North Moravia has the greatest number of bank default customer. Apart from that South Moravia and Central Bohemia also having large number of default customer.

Same time North Bohemia is having lowest bank default customers.

4.3.4 Loan Payment Duration

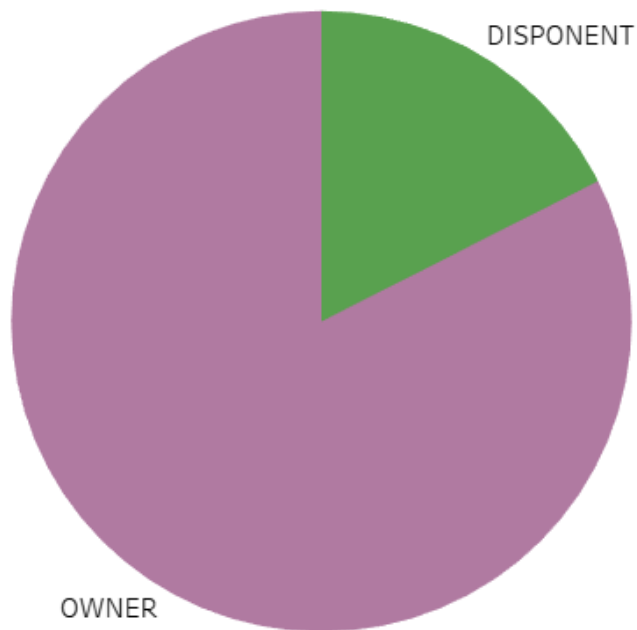


We can see from above graph that customers mainly preferred for monthly instalment payment rather than quarter payment or weekly payment.

So the number is high for monthly payment and least customer opted for quarterly payment.

4.3.5 Loan Dispense Type

Dispense Type



Looking at the above figure it can be said that loan disbursement amount is maximum given to the owner rather than deponent type.

Therefore number of owner is high.

Chapter V

5.Future Work and Conclusion:

Here we have done the classification using Binary Classification method (0 and 1). To improve this, we can use the probabilities predicted by model and set the threshold by ourselves. Here threshold could be based on several factors like business objectives. However, it could be different scenario in case to case.

The decision maker of bank wants to control the loss at acceptable level so they may use relatively low threshold. This mean more customers will be regarded as potential bad customer or likely default customer. Then their profile could be verified by credit management team.

In this way bank could detect the default behaviour in the early stage and conduct the corresponding actions to reduce the possible loss.

References

- [1] S.T. Trautmann, R. Vlahu, Strategic loan defaults and coordination: An experimental analysis, *Journal of Banking & Finance*, 37 (2013) 747-760.
- [2] D.E. Christman, Multiple realities: Characteristics of loan defaulters at a two-year public institution, *Community college review*, 27 (2000) 16-32.
- [3] R.J. Miles, M.C. Ozdogan, S.K. Song, C.D. Heard, Syndicated lending update: Defaulting lender issues, *Banking LJ*, 126 (2009) 165.
- [4] R. DeYoung, D. Glennon, P. Nigro, Borrower–lender distance, credit scoring, and loan performance: Evidence from informational-opaque small business borrowers, *Journal of Financial Intermediation*, 17 (2008) 113-143.
- [5] A. Blöchliger, M. Leippold, Economic benefit of powerful credit scoring, *Journal of Banking & Finance*, 30 (2006) 851-873.
- [6] E.I. Altman, H.J. Suggitt, Default rates in the syndicated bank loan market: A mortality analysis, *Journal of Banking & Finance*, 24 (2000) 229-253.
- [7] J. Dermine, C.N. de Carvalho, Bank loan losses-given-default: A case study, *Journal of Banking & Finance*, 30 (2006) 1219-1243.
- [8] T.O. Berge, K.G. Boye, An analysis of banks' problem loans, DOI (2007).
- [9] H.D. Khieu, D.J. Mullineaux, H.-C. Yi, The determinants of bank loan recovery rates, *Journal of Banking & Finance*, 36 (2012) 923-933.
- [10] G. Pennacchi, Deposit insurance, bank regulation, and financial system risks, *Journal of Monetary Economics*, 53 (2006) 1-30.
- [11] DOI <https://www.imf.org/external/np/sta/fsi/eng/fsi.htm>.
- [12] A.E. Khandani, A.J. Kim, A.W. Lo, Consumer credit-risk models via machine-learning algorithms, *Journal of Banking & Finance*, 34 (2010) 2767-2787.
- [13] R.E. Turkson, E.Y. Baagyere, G.E. Wenya, A machine learning approach for predicting bank credit worthiness, 2016 Third International Conference on Artificial Intelligence and Pattern Recognition (AIPR), IEEE, 2016, pp. 1-7.
- [14] G Galindo and P Tamayo , Credit Risk Assessment using statistical and machine learning : Basic Methodology and Risk modelling application
- [15] Coal, A Bank Loan default Prediction : A comparative model analysis
- [16] Manjet Kumar, Brijesh Patel, Harsh Kantawala, Credit risk analysis using machine learning algorithm

- [17] A Hasan, A Abraham, Modelling loan default prediction using ensemble neural network
- [18] A Ghadge, A Survey on Ensemble Model for loan prediction
- [19] Altunbas, Predicting Bank insolvencies using Machine Learning Technique.
- [20] Berger, M-Calder, Determinants of SME Loan Default
- [21] Peter Wanke, Barros, Azad, Predicting performance in ASEAN Bank :An integrated fuzzy MCDM-Neural Network Approach
- [22] Haling, A comparison and an application to credit rating transitions
- [23] James Kolari, D Glennon, Shin, Caputo, Predicting large US bank failures
- [24] Lal, Predicting Bank loan default using logistic regression
- [25] Aida Krichene, Using a Naïve Bayesian classifier methodology for loan risk assessment
- [26] Alina M dima, Simona V, Credit risk modelling for companies default prediction Using neural network
- [27 & 30] Glorfeld, Loan default prediction in Ukraine Retail Banking
- [28] A.J.F Feelders, A.J.F. Loux, Developing prediction model of loan risk in bank using Data Mining
- [29] Jozef Zurada, M Zuarada, Validating loan granting decision and predicting default rates on consumer loan
- [31] Michale Johnson, Bank loan default Prediction using Supervised classification algorithm
- [32] Wolfgang Härdle, R Moro, L Hoffman, Learning Machine Supporting Bankruptcy Prediction
- [33] Bo Wang, An Empirical Study on Loan default predictions model
- [34] Peter Addo, D Guegan, B Hassani, Credit Risk Analysing Using Machine and Deep Learning models
- [35] Suresh Ramakrishnan, M Bekri, M Mirzaei, Corporate Default Prediction with Adaboost and bagging classifiers
- [36] Frydman, Loan default prediction by combining soft information extracted from descriptive text in online peer to peer lending
- [37] Quinlan, Using a Naïve Bayesian Classification approach
- [38] Messier, Judgement and decision making research in accounting and auditing
- [39] Pompe and Feelders, Bank loan default prediction in Romania
- [40] Shin, Bank -firm relationship in default prediction models. An analysis on sample of Italian firms
- [41] Aamir Atiya, Credit risk predicting using neural network
- [42] Gestel, Corporate Loan default prediction :An overview of methodologies and applications

[43] Abellan, A comparative study on base classifier in ensemble methods for credit scoring

[44] Myers, Soft information and default prediction in Cooperative and Social Bank

[45] Lin, Tsai,Cheng,The Consumer loan default predicting model- An application of DEA -DA and neural network