

PROFESSIONAL TRAINING REPORT

Sathyabama Institute of Science and Technology (Deemed to be University)

Submitted in partial fulfillment of the requirements for
the award of Bachelor of Engineering Degree in
Computer Science and Engineering

By

NAME: SIBI PRASHANTH
(37110713)



**DEPARTMENT OF COMPUTER SCIENCE AND
ENGINEERING SCHOOL OF COMPUTING
SATHYABAMA INSTITUTE OF SCIENCE AND
TECHNOLOGY JEPPIAAR NAGAR, RAJIV GANDHI SALAI,
CHENNAI – 600119, TAMILNADU**

MARCH 2021



SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY)

Accredited with Grade “A” by NAAC

(Established under Section 3 of UGC Act, 1956)

JEPPIAAR NAGAR, RAJIV GANDHI SALAI, CHENNAI– 600119

www.sathyabamauniversity.ac.in



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of **Sibi Prashanth C (37110713)** who carried out the project entitled “**Performance Analysis of Loan default Predictor**” under my supervision from November (2020) to March (2021).

Internal Guide Name

Dr Mercy Paul Selvan M.E,Ph.D

Head of the Department

Dr.L.Lakshmanan M.E,Ph.D

Dr.S.Vigneshwari M.E., Ph.D.,

SUBMITTED FOR VIVA VOCE EXAMINATION HELD ON

Internal Examiner

External Examiner

DECLARATION

I, Sibi Prashanth C hereby declare that the Project Report entitle Performance analysis of Loan default predictors, done by me under the guidance of Dr. Mercy Paul Selvan M.E.,Ph.d is submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering degree in Computer Science and Engineering.

DATE:

PLACE:

SIGNATURE OF THE CANDIDATE:

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to **Board of Management** of **SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. T.Sasikala M.E., Ph.D., Dean**, School of Computing , **Dr.S.Vigneshwari M.E., Ph.D., and Dr.L.Lakshmanan M.E., Ph.D.,** Heads of the Department of Computer Science and Engineering for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide **Dr. Mercy Paul Selvan M.E.,Ph.d.,** for her valuable guidance, suggestions and constant encouragement paved way for the successful completion of my project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the **Department of Computer Science and Engineering** who were helpful in many ways for the completion of the project.

Certificate



Python for Data Science and AI

ISSUED TO

Sibi Prashanth C



Issued on: 25 APR 2020

Issued by: Coursera | Authorized by: IBM

Verify: <https://www.youracclaim.com/go/3bM1x3dD>

ABSTRACT

The data set used in this project was provided by IBM

This entire project was coded using Python in the Jupyter Notebook. This project is about performance analysis of loan default predictor model, which was created using 4 of the following algorithms: Logistic Regression, KNN, Decision Tree, and Support Vector Machine. First the data sets are read. Then data visualisation is done to get some insights of the data, and some data normalization for data consistency. Then starts the model building. The model is trained using and then validated using the training data set. The model is tested using a part of the training data set, called the validation set. Then our 4 models are evaluated using the following evaluation metrics: Jacard Index, F1 Score. Then the performance of those 4 models, are displayed in a tabular form. The above is using the training data set. Then the 4 models, are given a completely new data, it hasn't seen before, and its performance is tested on previously unseen data and the scores are again displayed.

The goal is to make use of data which contains various details about a client's Transactional history, and other information about him, and the model will predict if a particular customer, will be able to pay of his loans or not.

LIST OF FIGURES

Fig Name	Page No
Fig 3.1: First 5 rows of the training data set	5
Fig 3.2: First 5 rows of the testing data set	5
Fig 3.3: Decision Tree	8
Fig 3.4: SVM	8
Fig 3.5: Jaccard index	9
Fig 4.1: Scores of the Baseline Model	11
Fig 4.2: Scores of the entire data set	11
Fig 4.3: Bar plot of the 5 scores	12

LIST OF TABLES

Table No and Name	Page No
Table 1.1: Data Set Description	2
Table 4.1: Scores of models on training data set	2
Table 4.2: Scores of models on testing data set	3

Contents

PROFESSIONAL TRAINING REPORT	i
NAME: SIBI PRASHANTH	i
(DEEMED TO BE UNIVERSITY)	ii
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING	ii
SUBMITTED FOR VIVA VOCE EXAMINATION HELD ON	ii
DATE:	iii
ABSTRACT	vi
LIST OF FIGURES	vii
LIST OF TABLES	viii
1.INTRODUCTION	1
WHAT IS MACHINE LEARNING?	1
APPLICATIONS OF MACHINE LEARNING?	1
THE FOUR ALGORITHMS USED.	1
THE 2 EVALUATION METRICS USED.	1
AIM AND SCOPE OF THE PRESENT INVESTIGATION	2
2. LITERATURE SURVEY	4
3.METHODOLOGY	5
PACKAGES USED	5
PROCEDURE	5
3.2.1- DATA READING AND ANALYSIS	5
3.2.2- DATA PREPROCESSING AND NORMALISATION	6
ALGORITHMS EXPLAINED:	6
MODEL BUILDING	9
EVALUATION METRICS	9
MODEL TESTING	10
4.RESULTS AND DISCUSSION	11
5. SUMMARY	13
REFERENCES:	14
APPENDIX	15
SNAP SHOTS:	15

1.INTRODUCTION

WHAT IS MACHINE LEARNING?

It is a branch of artificial intelligence that allows the computer to learn from past experiences, and think by themselves using past experience/data without having to explicitly code. Its used in data science, where insights and patterns are learned from past data. These insights are applied when solving a future problem.

APPLICATIONS OF MACHINE LEARNING?

Machine Learning is used in:

1. Spam detection in our email applications
2. Display of target ads to particular users, based on his past searches in the Internet.
3. Movie recommendations in Netflix .
4. Route generation in maps.
5. Music recommendation in spotify

THE FOUR ALGORITHMS USED.

The following 4 algorithms are used :

1. Logistic Regression
2. KNN
3. Decision Tree
4. Support Vector Machines

THE 2 EVALUATION METRICS USED.

1. Jaccard Index or Jaccard Score
2. F1 Score

AIM AND SCOPE OF THE PRESENT INVESTIGATION

Now whenever a customer approaches a bank for loans, they don't just hand over the amount of money the customer asked. The bank determines some factors, before sanctioning the loan, using the customer information and his previous transactional history. The data set used in this contains the information about the customer, with regard to the amount in principal he got the loan for, and his other details. So 4 models are built, which will be trained with this data set, and evaluated.

Initially the model is built with a part of the training data set, and its performance is evaluated using the validation set. Then it's tested with completely unseen data and its performance is again evaluated. We use the following three evaluation metrics: LogLoss, F1 Score, Jaccard Index.

Field	Description
Loan_status	Whether a loan is paid off or in collection
Principal	Basic principal loan amount at the
Terms	Origination terms which can be weekly (7 days), biweekly, and monthly payoff schedule
Effective_date	When the loan got originated and took effects
Due_date	Since it's one-time payoff schedule, each loan has one single due date
Age	Age of applicant
Education	Education of applicant

Gender	The gender of applicant
--------	-------------------------

Table 1.1: Data Set Description

2. LITERATURE SURVEY

This section talks about works , that has been done previously relating to the same domain. Loan defaults can cause losses to banks, as a result it's a very crucial factor. Earlier days not much data were there , and banks were just utilizing certain people in order to determine customers with good credit score. Nowadays with the abundance in data available and powerfull tools, which are helpful to banks and finance firms for searching good customers. There has been an increase in demands for loans, luckily there have been strides made in the fields of data science. Espacially loan default prediction, has more focus spent on it for research and model building.

In the fast moving world of technology and innovation, the process of loan default predictions has been automated, especially with the help of advancements in data science like mentioned earlier.

[1] Used Decision Tree and Random Forest and got an accuracy score of 73 % and 80%.

[3] Concluded that decision tree(95% accuracy) performs better than Logistic Regression(73% accuracy)

[4] Talked about algorithms such as SVM, Logistic Regression and KNN

[5] Conculoded that:

- Low income applicants more likely to pack back loans, than high income applicants.
- Applicants not meeting eligilibty criteria are less likely to pack back loans.

[6] Used decision tree and managed to achieve score of 81%.

3.

METHODOLOGY

PACKAGES USED

- o Numpy: Has advanced mathematical functions.
- o Pandas: package dealing with creation and manipulation of data sets using data frames.
- o Matplotlib: for plotting various graphs.
- o Sklearn: To build the logistic regression model, decision tree, Support Vector machine and decision trees.

PROCEDURE

3.2.1- DATA READING AND ANALYSIS

First the data set called 'loan_train' is read into a data frame. We then display the first 5 rows of it.

	Unnamed: 0	Unnamed: 0.1	loan_status	Principal	terms	effective_date	due_date	age	education	Gender
0	0	0	PAIDOFF	1000	30	9/8/2016	10/7/2016	45	High School or Below	male
1	2	2	PAIDOFF	1000	30	9/8/2016	10/7/2016	33	Bechalor	female
2	3	3	PAIDOFF	1000	15	9/8/2016	9/22/2016	27	college	male
3	4	4	PAIDOFF	1000	30	9/9/2016	10/8/2016	28	college	female
4	6	6	PAIDOFF	1000	30	9/9/2016	10/8/2016	29	college	male

Fig 3.1: First 5 rows of the training data set

We see there are 2 unnamed columns here, which will be dropped later on.

Next the format of the columns 'effective_date' and 'due_date' , is changed to date time object.

	Unnamed: 0	Unnamed: 0.1	loan_status	Principal	terms	effective_date	due_date	age	education	Gender
0	0	0	PAIDOFF	1000	30	2016-09-08	2016-10-07	45	High School or Below	male
1	2	2	PAIDOFF	1000	30	2016-09-08	2016-10-07	33	Bechalor	female
2	3	3	PAIDOFF	1000	15	2016-09-08	2016-09-22	27	college	male
3	4	4	PAIDOFF	1000	30	2016-09-09	2016-10-08	28	college	female
4	6	6	PAIDOFF	1000	30	2016-09-09	2016-10-08	29	college	male

Fig 3.2: First 5 rows of the testing data set

Using the column 'effective_date' a new custom column is created called 'day_of_week'. This column, represents the day in the week, loan was sanctioned. Its values range from 1 to 7.

Then some basic data analysis is between, to get an overview of the data. Both univariate and Bivariate analysis performed. From the Bi Variate analysis between the columns 'day_of_week' and 'loan_status', and we find out that people who have taken loans at end of week(day 5 and above) are less likely to pay it off , so a new column called 'weekend' is created. Through feature binarization, whenever the day of week is 3 or lower, the 'weekend' column is set to 1, else its is set to 0.

3.2.2- DATA PREPROCESSING AND NORMALISATION

We have a set of categorical columns, so they have to be converted to numbers for model building in the later stages. Those are columns like 'Gender' , 'Education' , and 'loan_status'. 'Gender' and 'Loan_Status' are converted into 1's and 0s, each denoting the either class of their respective column. Then to the 'education' column, they are converted into dummy variables. A new column, is then created called 'deadline' is created which calculates the difference in no between 'effective_date' and 'due_date'.

Now the features are extracted. The features are the set of columns, which are the independent variables or the predictor variables are extracted. Then the target variable is stored separately.

Finally the values of the features , are normalized between 0 and 1. So as to improve the efficiency of the model.

ALGORITHMS EXPLAINED:

1. Logistic Regression:

“Logistic regression is a statistical model that in its basic form uses a logistic function to model a binary dependent variable, although many more complex extensions exist. In **logistic regression**^[1] is estimating the parameters of a logistic model. Mathematically, a binary logistic model has a dependent variable with two possible values, such as pass/fail which is represented by an indicator variable, where the two values are labelled "0" and "1". In the logistic model, the log-odds (the logarithm of the odds) for the value labelled "1" is a linear combination of one or more independent variables ("predictors"); the independent variables can each be a binary variable (two classes, coded by an indicator variable) or a continuous variable (any real value).” From: Wikipedia.

Logistic regression is usually used in cases, when our problem at hand is a classification problem. Where we classify an input (here where its customer related data) into a different output category. If the no of the o/p categories is 2 its called binary classification. Common examples are, when we want to determine if a person will test positive for a particular disease, and in this case, if he will be able to pay of his loan (Yes or no).

2. K Nearest Neighbors

This algorithm classes a particular input point into a label(output) based on its position with respect to its nearest neighbours. Assuming that the input points of multiple co-ordinates are plotted on a N-dimensional graph. Suppose there already a set of 'n' points where some points are mapped into an output '1' and the others mapped to an output '0', and a new point, 'n+1'th point is plotted in the graph. Suppose if the new point is plotted closer to the points mapped as '1' this new point will also be mapped as '1', and vice versa. This was just a simple example. However, when more and more points are

mapped into the graph, much advanced calculations are made, into categorising the new points into a particular label.

3. Decision Trees

It's a Machine Learning algorithm, that uses a tree like flow chart structure, where each node represents a decision or test, and the branches from that node represent the outcome of that particular test. There are 3 types of nodes in a decision tree:

- Decision Nodes- The node which contains a test or condition.
- Chance Nodes- the outcome of a particular decision node.
- End Nodes- the final nodes representing the final output.IT
-

Here below is an example of a simple decision tree, on weather we should play tennis or not.

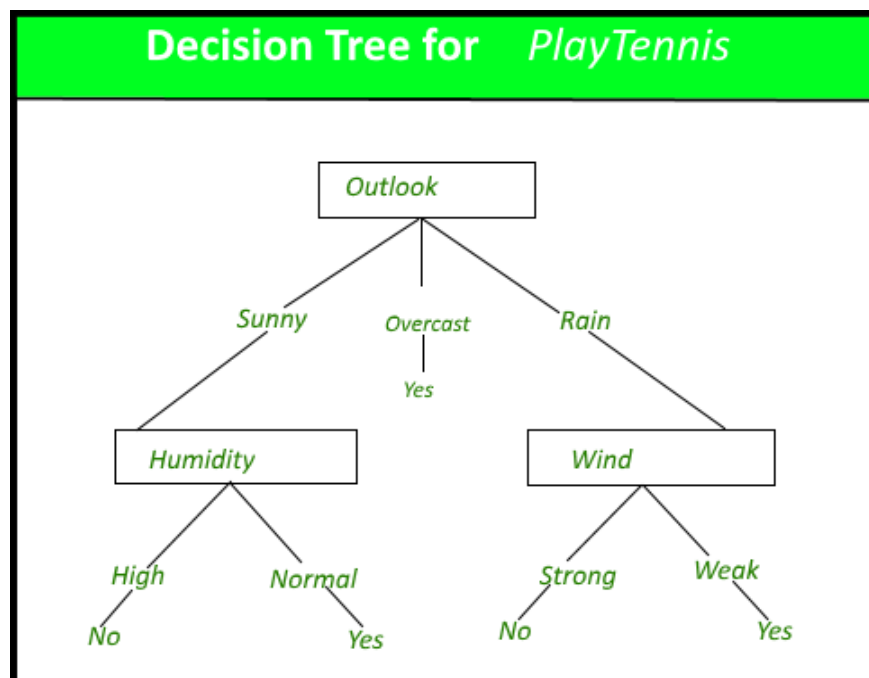


Fig 3.3: Decision Tree

4. Support Vector Machine

It's a linear algorithm that is used for both classification problems as well as regression problems. It can be used for linear and nonlinear problems. It creates a hyper plane, which divides the input points into corresponding output labels.

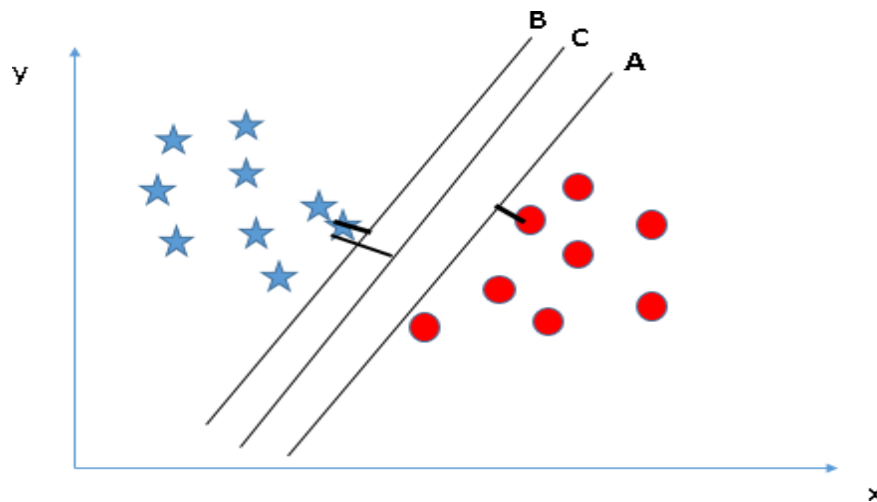


Fig 3.4: Support Vector Machine

MODEL BUILDING

From the Sklearn modules, each of the algorithms are imported. Then the models are each implemented one by one.

First to the object of Knn model, the training data is passed, and the predictions made by the model is stored in a list. This list is compared to the actual values of the target variable. Then using the Jacard Index, and F1 score this model is evaluated using Jacard Index, F1 Score and Log Loss and its scores are noted.

This same procedure is applie for all the models, and the score for each model is noted down. Finally we get a performance results of each of those 4 models,

EVALUATION METRICS

1. Jaccard Index: It's a statistic or evaluation metric, which is used to denote how similar 2 sets are. Shows the similarity in percentage among the total data, between those two sets. Its values range from 0 to 100.

Its denote by J and its represented by the below formula.

$$J = (A \cap B) / (A \cup B)$$

Where A and B, represents the sets compared.

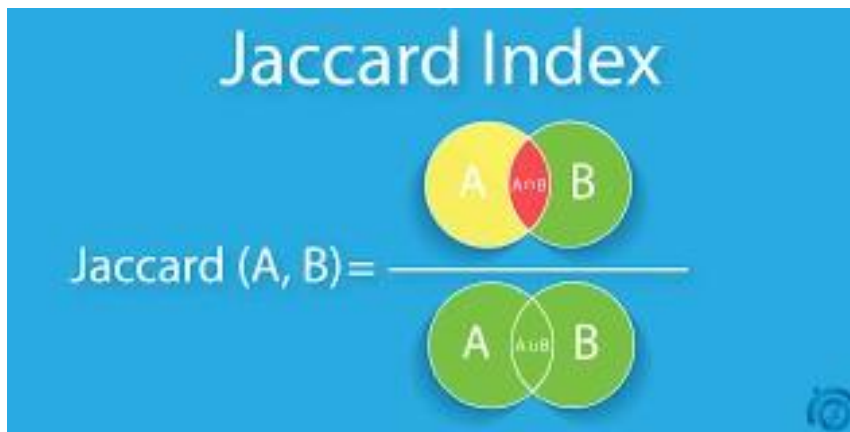


Fig 3.5: Jaccard Index

2. F1 Score: Its an evaluation metric, which measures the accuracy of the model. Its used mainly for Binary Classification Models, whose output is labels are of the from (1 or 0) or (True or False). It used two metrics called Precision and Recal to calculate the accuracy.

Its given by the following formula.

$$F1 = 2 * (Precision + Recall) / (Precision * Recall)$$

$$Precision = (TP) / (TP + FP)$$

$$Recall = (TP) / (TP + FN)$$

Precision- tells us, the percentage of cases where our model predicted as postivie (in this case Paid Off), are actually positive.

Recall - tells us, among the actual positive values present in the data set, how many values are predicted as Postive.

TP- The positive values predicted predicted by the model, that are actually positive.

FP- The true values predicted by the model, that are actually negative.

FN- The false values predicted by the model, that are actually positive.

The range of F1 score is between 0 and 1.

MODEL TESTING

Finally the models are again given, the features of a completely new data set, the testing data set previously unknown by the model. The Labels of each, category are saved separately. Just like the procedure followed while using the training set, a similar procedure followed when using the testing data set.

Similarly the performance of the 4 models , are noted down and compared with the previous set, to check if there is any difference in performance, like overfitting etc.

4. RESULTS AND DISCUSSION

Conclusions from basic data analysis:

- 86 % of female pay there loans while 73 % of males pay there loan.
- People who get the loan at the end of the week don't pay it off.
- People pay of the loan, in the 9th month , rather than paying it in 11th and 12th.

Performance of the 4 algorithms:

K Nearest Neighbors:

To determin the value of K , a loop is run from the values 3 to 12. To determine the best value.

The best value of K = 8, from the iterations

Performance of the models with the training data set.

F1 Score	Jaccard Index	
KNN	0.77	0.79
Decision Tree	0.92	0.93
SVM	0.66	0.75
Logistic Regression	0.7	0.76

Table 4.1: Scores of models on training data set

From the testing data set.

	F1 Score	Jaccard Index
KNN	0.71	0.74
Decision Tree	0.76	0.76
SVM	0.69	0.76
Logistic Regression	0.7	0.78

Table 4.2: Scores of models on testing data set

The scores off Logistic Regression model and the SVM model, have improved.

However, the scores of the Decision tree model has surprisingly reduced by a significant value, implying overfitting.

5. SUMMARY

This entire project was coded in python. As a result the data sets were successfully read, data was successfully normalized. Among the 4 models, the decision tree performed exceptionally well on the training data set, with an F1 score and Jaccard Index of 0.92 and 0.93 respectively. However on the testing set, its score reduced considerably implying overfitting. On the contrary the logistic regression model and svm showed improved performance, on the testing data.

REFERENCES:

- [1] : Mehul Madaan et al 2021 IOP Conf. Ser.: Mater. Sci. Eng. 1022 012042
- [2]: Ahmed M S I and Rajaleximi P R 2019 An empirical study on credit scoring and credit scorecard for financial institutions Int. Journal of Advanced Research in Computer Engineering & Technol. (IJARCET) 8 275–9
- [3]: Zhu L, Qiu D, Ergu D, Ying C and Liu K 2019 A study on predicting loan default based on the random forest algorithm The 7th Int. Conf. on Information Technol. and Quantitative Management (ITQM) 162 pp 503–13
- [4] Breeden J L 2020 A survey of machine learning in credit risk
- [5] Madane N and Nanda S 2019 Loan prediction analysis using decision tree Journal of The Gujarat Research Society 21 p p 214–21
- [6] Supriya P, Pavani M, Saisushma N, Kumari N V and Vikas K 2019 Loan prediction by using machine learning models Int. Journal of Engineering and Techniques 5 pp144–8
- [7] Shoumo S Z H, Dhruva M I M, Hossain S, Ghani N H, Arif H and Islam S 2019 Application of machine

learning in credit risk assessment: a prelude to smart banking TENCON 2019 – 2019 IEEE Region 10 Conf. (TENCON) pp 2023–8

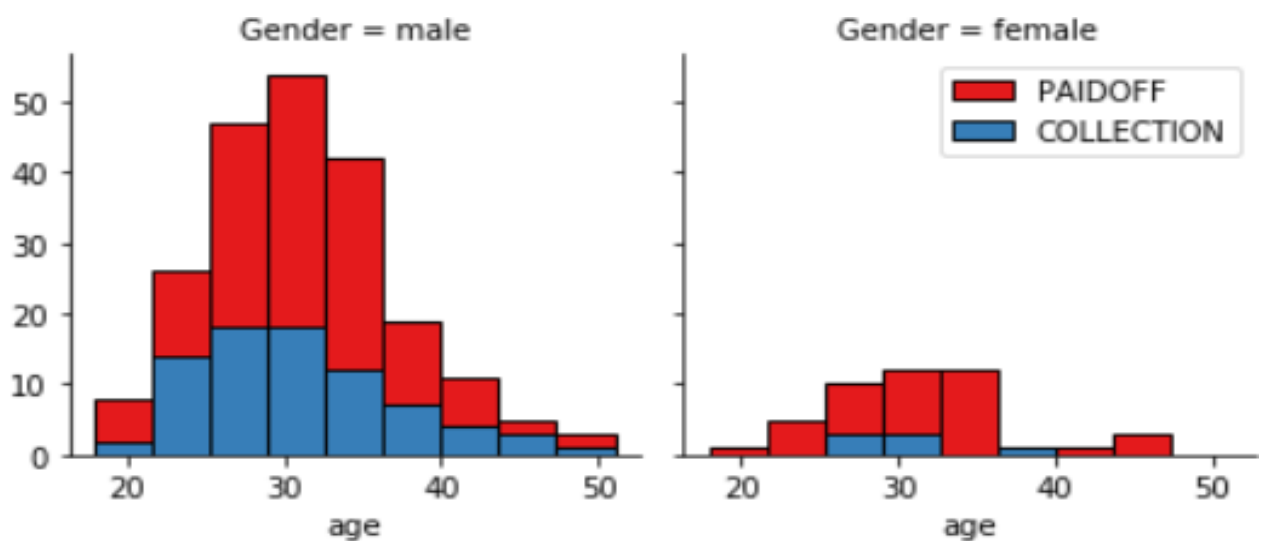
[8] Hamid A J and Ahmed T M 2016 Developing prediction model of loan risk in banks using data mining Machine Learning and Applications: An Int. Journal (MLAIJ) 3 pp 1–9

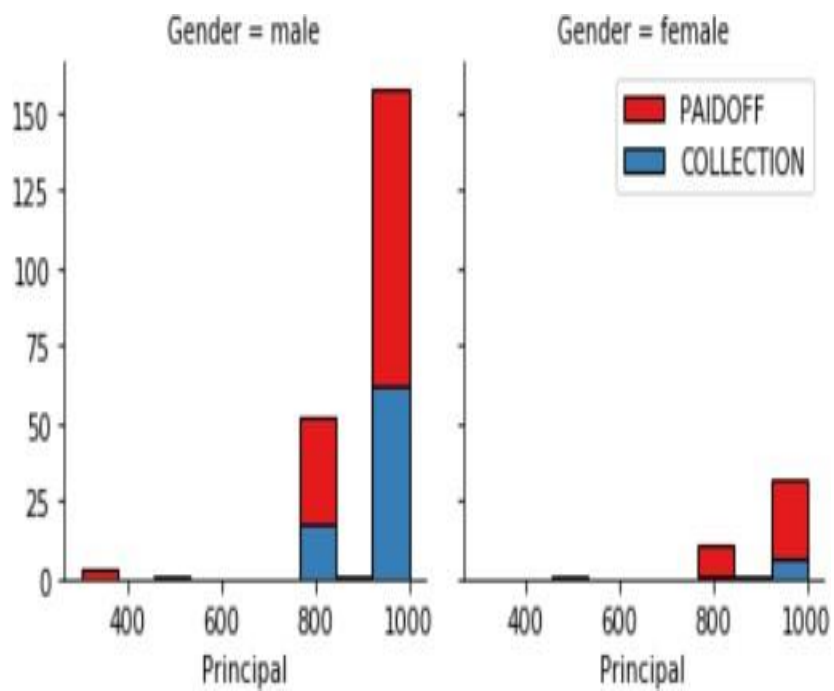
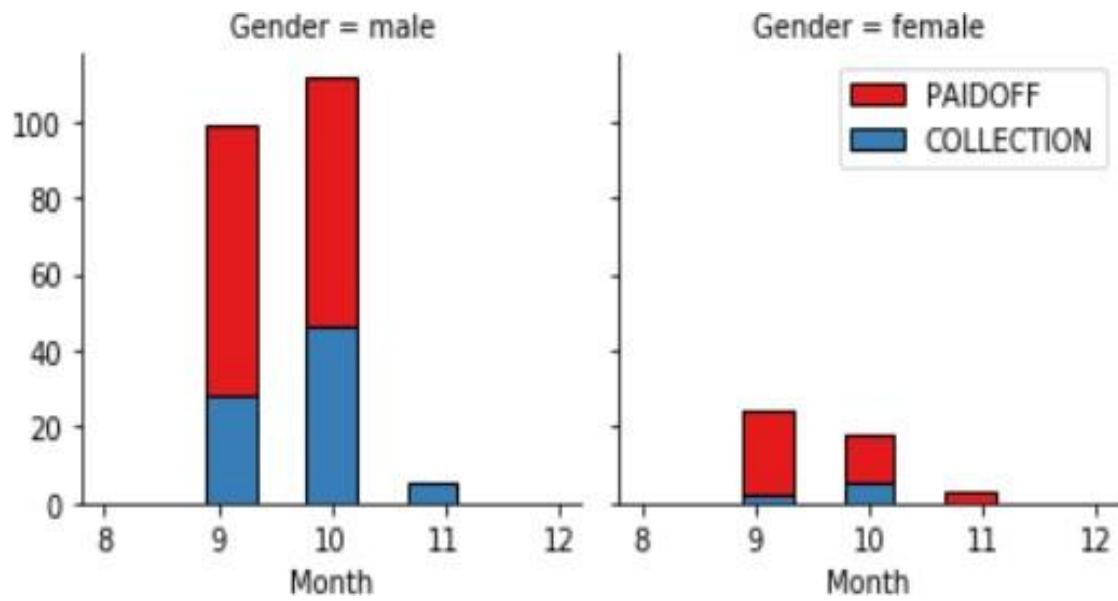
[9] Li M, Mickel A and Taylor S 2018, “Should this loan be approved or denied?”: a large dataset with class assignment guidelines Journal of Statistics Education 26 pp

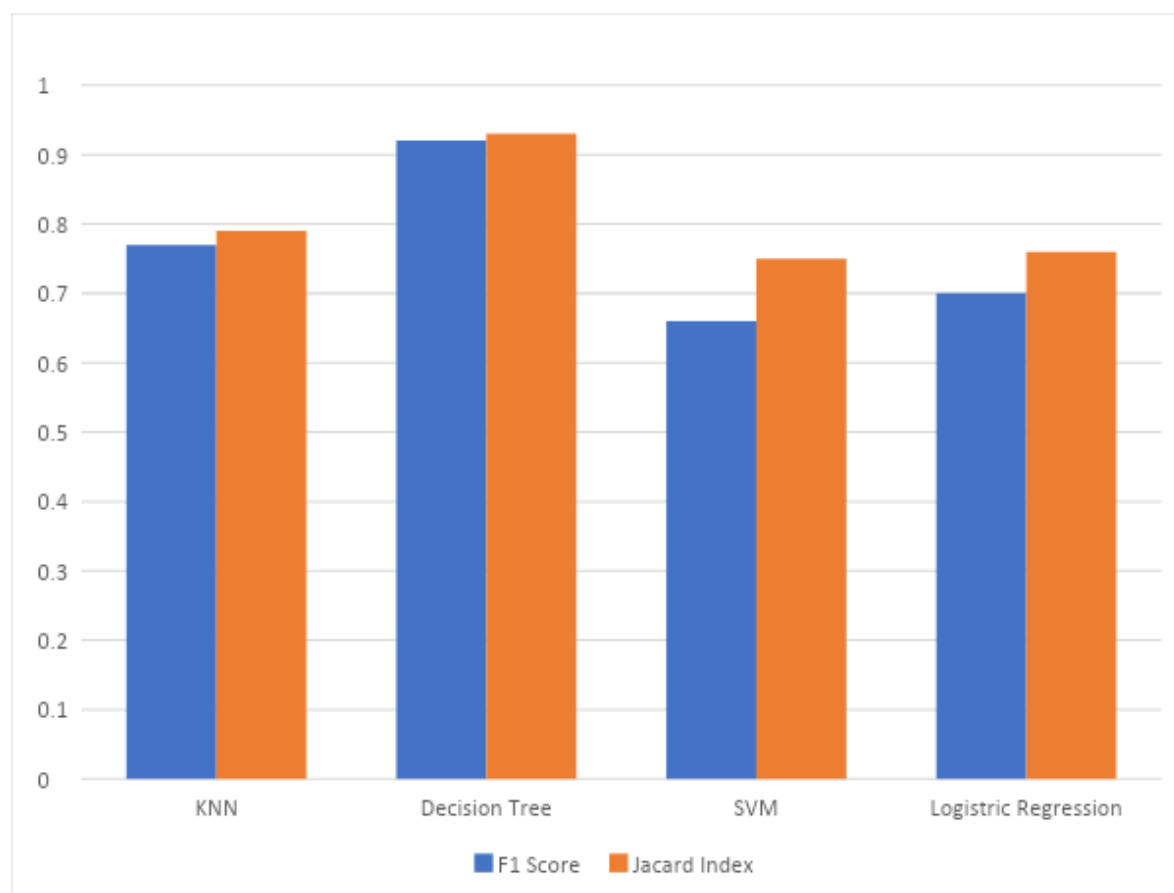
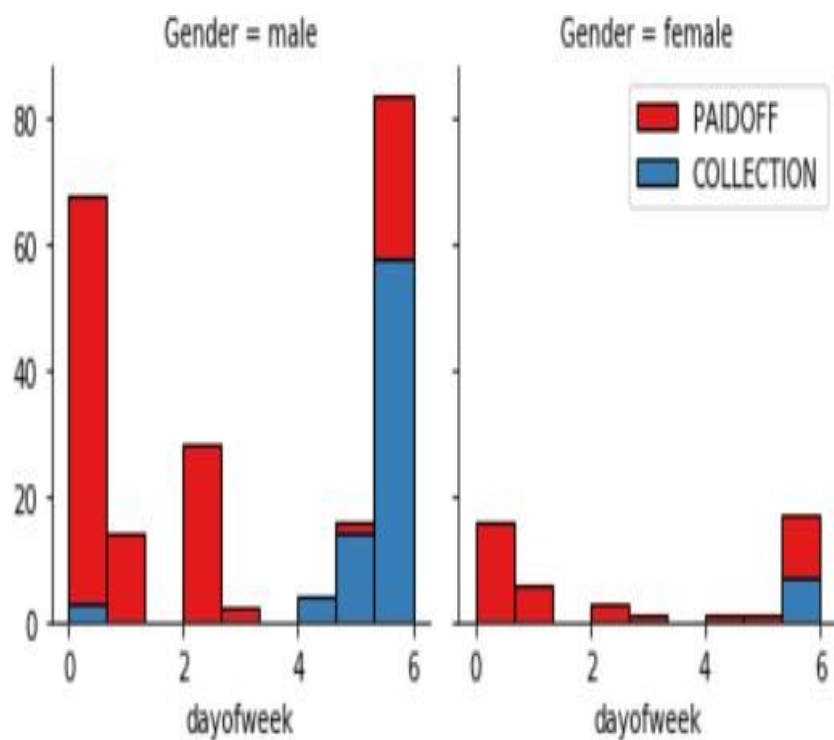
[10] Kacheria A, Shivakumar N, Sawkar S and Gupta A 2016 Loan sanctioning prediction system Int. Journal of Soft Computing and Engineering (IJSCE) 6 pp 50–3

APPENDIX

SNAP SHOTS:







S

```
# import itertools
import numpy as np
import matplotlib.pyplot as plt
from matplotlib.ticker import NullFormatter
import pandas as pd
import numpy as np
import matplotlib.ticker as ticker
from sklearn import preprocessing
%matplotlib inline
```

```
df=pd.read_csv('loan_train.csv')
df.head()
```

```
#Convert to date time object
```

```
df['due_date']=pd.to_datetime(df['due_date'])
df['effective_date']=pd.to_datetime(df['effective_date'])
```

```
#Visualising and pre processing
```

```
df['loan_status'].value_counts()
```

```
df['loan_status'].value_counts().plot.bar()
```

```
import seaborn as sns

bins = np.linspace(df.Principal.min(), df.Principal.max(), 10)
g = sns.FacetGrid(df, col="Gender", hue="loan_status", palette="Set1", col_wrap=
g.map(plt.hist, 'Principal', bins=bins, ec="k")

g.axes[-1].legend()
plt.show()
```

```
bins = np.linspace(df.age.min(), df.age.max(), 10)
g = sns.FacetGrid(df, col="Gender", hue="loan_status", palette="Set1", col_wrap=
g.map(plt.hist, 'age', bins=bins, ec="k")

g.axes[-1].legend()
plt.show()
```

```
df['dayofweek'] = df['effective_date'].dt.dayofweek
bins = np.linspace(df.dayofweek.min(), df.dayofweek.max(), 10)
g = sns.FacetGrid(df, col="Gender", hue="loan_status", palette="Set1", col_wrap=
g.map(plt.hist, 'dayofweek', bins=bins, ec="k")
g.axes[-1].legend()
plt.show()
```

```
df['Gender'].replace(to_replace=['male','female'], value=[0,1],inplace=True)
df.head()
```

```
df['deadline']=df['due_date']-df['effective_date']
df.head()
```

```
df['deadline']=df['deadline'].dt.days
df.head(3)
```

```
Feature = df[['Principal','terms','age','Gender','weekend','dayofweek']]
Feature = pd.concat([Feature,pd.get_dummies(df['education'])], axis=1)
Feature.drop(['Master or Above'], axis = 1,inplace=True)
Feature.head()
```

```
X = Feature
X[0:5]
```

```
# We encode our labels as 0 and 1 s
y = df['loan_status'].replace(to_replace=['PAIDOFF','COLLECTION'], value=[0,1]).values
y[0:5]
```

```
X= preprocessing.StandardScaler().fit_transform(X)
X[0:5]
```

```
# import scoring methods
from sklearn.metrics import jaccard_similarity_score
from sklearn.metrics import f1_score
from sklearn.metrics import log_loss
from sklearn.model_selection import cross_val_score
```

```
# our data for training (used '_train' just for improving readability)
X_train, y_train=X,y
```

```
# a dictionary for keeping all scores of the classifiers
trainScores={}
```

K N N

```
from sklearn.neighbors import KNeighborsClassifier
bestScore=0.0
accList=[]

for k in range(3,12):
    clf_knn=KNeighborsClassifier(n_neighbors=k,algorithm='auto')
    scores=cross_val_score(clf_knn,X,y,cv=k)
    scores=scores.mean()
    accList.append(scores)

    if scores>bestScore:
        bestScore=scores
        best_clf=clf_knn
        bestK=k

print("Best K is :",bestK,"| Cross validation Accuracy :",bestScore)
clf_knn=best_clf
```

```
clf_knn.fit(X_train,y_train)
y_pred=best_clf.predict(X_train)
```

```
trainScores['KNN-jaccard']=jaccard_similarity_score(y_train, y_pred)
trainScores['KNN-f1-score']=f1_score(y_train, y_pred, average='weighted')
```

DECISION TREE

```
from sklearn import tree
```

```
clf_tree=tree.DecisionTreeClassifier()
clf_tree=clf_tree.fit(X_train,y_train)
```

```
y_pred=clf_tree.predict(X_train)
```

```
trainScores['Tree-jaccard']=jaccard_similarity_score(y_train, y_pred)
trainScores['Tree-F1-Score']=f1_score(y_train, y_pred, average='weighted')
```

```
trainScores
```

SUPPORT VECTOR MACHINE

```
y_train=y_train.astype(float)
```

```
from sklearn import svm

clf_svm=svm.LinearSVC(random_state=7)
clf_svm.fit(X_train, y_train)

y_pred=clf_svm.predict(X_train)
```

```
trainScores['SVM-jaccard']=jaccard_similarity_score(y_train, y_pred)
trainScores['SVM-f1-score']=f1_score(y_train, y_pred, average='weighted')
```

```
trainScores
```

```
#Logistic Regression
```

```
from sklearn.linear_model import LogisticRegression

clf_log = LogisticRegression(random_state=0, solver='lbfgs',
                             multi_class='multinomial')
clf_log.fit(X_train, y_train)

y_pred=clf_log.predict(X_train)
y_proba=clf_log.predict_proba(X_train)
```

```
trainScores['LogReg-jaccard']=jaccard_similarity_score(y_train, y_pred)
trainScores['LogReg-f1-score']=f1_score(y_train, y_pred, average='weighted')
```