Bank Loan Personal Modelling using Classification Algorithms of Machine Learning

Duggirala Akhil

Third Year Btech.CSE,Rajiv Gandhi University of Knowledge Technologies - IIIT Srikakulam, AP, India

Abstract: ***Bank Loan is a major burden worldwide in the 21st century. Human services are overpowering national and corporate spending plans because of asymptomatic loans including education loans, business loans, etc. Consequently, there is an urgent requirement for early solution of such loans balances and finding the persons who are capable of buying the personal loan. The information which is gathered by data analysis of banks is utilizing by applying different blends of calculations and algorithms for the early-stage prediction of whether the person is able to pay the loan. Machine Learning is one of the slanting innovations utilized in numerical circles far and wide in many fields. In this research, we compared the accuracy of machine learning algorithms that could be used for predictive analysis of loan clearance and predicting overall risks associated with it. The proposed experiment is based on a standard machine learning algorithms such as Logistic Regression, Random Forest, K-Nearest Neighbors(KNN), Support Vector Machine (SVM), Decision Tree and Gradient Boosting. Most of the entities in this world are related in one or another way, at times finding a relationship between entities can help you make valuable decisions. Likewise, I will attempt to utilize this information as a model that predicts the person is able to pay the loan amount or not. Moreover, the data analysis is carried out in Python using Google Colab in order to validate the accuracy of all the Algorithms.***

Keywords: Machine learning, Data Analysis, Classification algorithms, Bank Loan

1. Introduction

Banking is presently the leading problem of worldwide. As expected 69 percent of adults – 3.8 billion people have bank accounts. Education Loans [1], Business Loans are related to the Banking sector. Thus, a feasible and accurate prediction of bank-related issues is very important. National banks collect data on various bank-related issues. These data can be visualized using various machine learning techniques to gain useful insights. Data collected is very huge and, many times, this data can be noisy with outliers. As the datasets are too huge for human minds to visualize, this process can be easily explored using various machine learning techniques. Thus, these algorithms have become very useful, in recent times, to predict the presence or absence of bank-related issues accurately. To begin with, the work we are using different types of techniques and algorithms. In this paper, the classification of machine learning techniques and algorithms are used to increase the accuracy rate. In Machine learning, classification algorithms are a supervised learning approach in which the computer learns from the input data and learn from it. This data collection may basically be bi-class (like recognizing whether the individual is male or female or that the mail is spam or non-spam) or it might be multi-class. Here are the names of classification algorithms which we are going to implement and compare the accuracy in this research:

1) Linear Classifiers: Logistic Regression
2) K-Nearest Neighbor
3) Support Vector Machine (SVM)
4) Decision Trees
5) Random Forest
6) Naive Bayes
7) Gradient Boosting
8) Neural Networks

2. Background of the Study

Banking is the most important organization in present scenario. because most of the transactions became online. The bank is a governmental organization that is liable to government policies. The Bank is having the six types of bank loans: Education Loans, Loans against PPF, Business Loans, Personal Loans, Gold Loans, and Vehicle Loans [2]. The dataset used for the logistic regression analysis is available on the Kaggle website, from an ongoing bank loan personal modelling. The classification goal of this study is to predict whether the customer has the liable to take a loan. The bank loan personal modelling dataset consists of 5,000 records of customer's data and 14 attributes. The data analysis is carried out in Python programming by using Jupyter Lab which is a more flexible and powerful data science application software.

3. Methodology

3.1 Workflow of building Machine Learning Model

Figure 1 indicates the steps followed in order to build the model in machine learning.

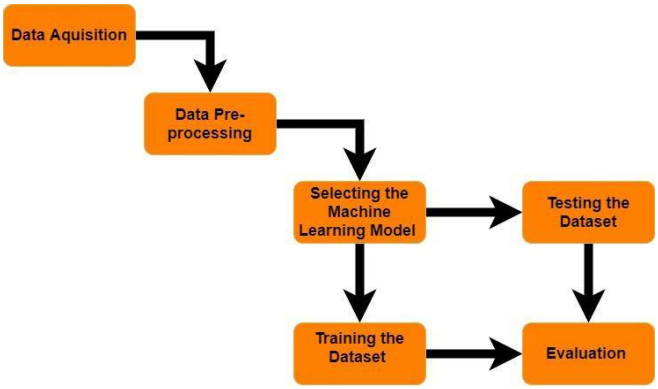


Figure 1: Workflow of building Machine Learning Model

3.2 Data Acquisition

The dataset which is used for classification is collected from the Kaggle website.

3.3 Data Pre-Processing

Data Preprocessing helps to build up an accurate Machine Learning model. Data pre-processing is the process of transforming the data. It will remove all the NAN, NULL values from our data and brings the data into a format where a machine can parse it easily. This process is also called as Data Wrangling. This includes the identification of missing data, noisy data, inconsistent data, and null values.

3.4 Proposed System

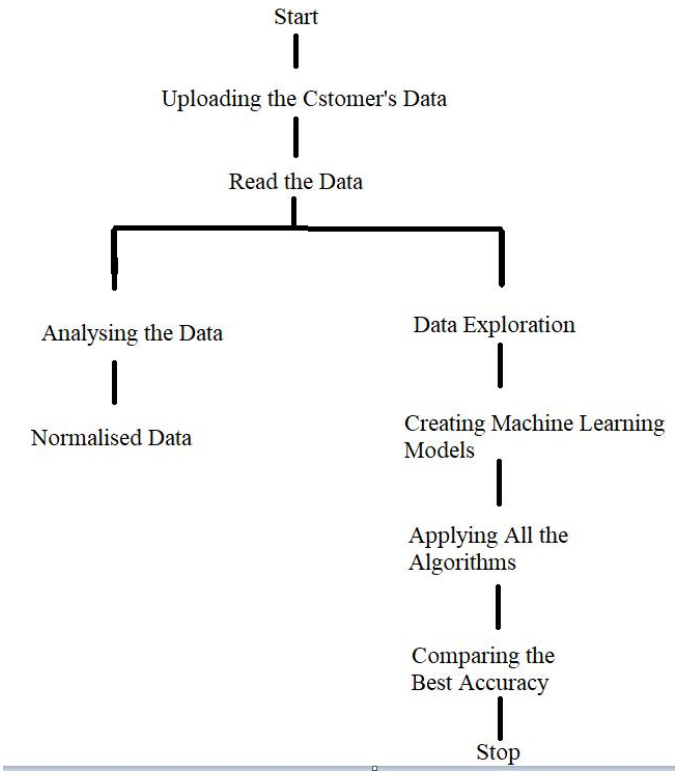


Figure 2: Proposed System

3.5 Select Machine Learning Model

Then the pre-processed data is identified for the analysis. We will be using the Classification Algorithms of machine learning to compare the best accuracies of all algorithms.

a) Input Variables of the study
The dataset consists of 14 columns. Machine Learning model is based on the identification of Dependent Variable.

Age : Customer's age

Experience: Number of years of professional experience

Income: Annual income of the customer

Zip Code : Home Address Zip Code

Family : Family size of the customer

CCAvg: Average spending on credit cards per month

Education: Education Level 1.Undergraduate 2. Graduate 3. Advanced Professional

Mortgage: Value of house mortgage if any

Securities Account: Does the customer have a securities account with the bank?

CD Account : Does the customer have a cetificate of deposit (CD) account with the bank?

Online: Does the customer use internet banking facilities?

CreditCard : Does the customer uses a credit card issued by UniversalBank?

Personal Loan : Did this customer accept the personal loan offered in the last compaing?x

Figure 3: Description of Input Variables

4. Data Analysis

Data Analysis was carried out on the Google Colab.

4.1 Importing the Libraries

Here we have loaded the data into Google Colab to build a machine learning model. In addition to that, the required libraries are used as supportive applications for carrying out analysis on the data .

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt        #visualisation
import seaborn as sns                  #Visualisation
%matplotlib inline
```

4.2 Reading the Dataset

	ID	Age	Experience	Income	ZIP Code	Family	CCAvg	Education	Mortgage	Personal Loan	Securities Account	CD Account	Online
ID	1.000000	-0.008473	-0.008326	-0.017695	0.013432	-0.016797	-0.024675	0.021463	-0.013920	-0.024801	-0.016972	-0.006909	-0.002528
Age	-0.008473	1.000000	0.994215	-0.055269	-0.029216	-0.046418	-0.052012	0.041334	-0.012539	-0.007726	-0.000436	0.008043	0.013702
Experience	-0.008326	0.994215	1.000000	-0.046574	-0.028626	-0.052563	-0.050077	0.013152	-0.010582	-0.007413	-0.001232	0.010353	0.013898
Income	-0.017695	-0.055269	-0.046574	1.000000	-0.016410	-0.157501	0.645984	-0.187524	0.206806	0.502462	-0.002616	0.169738	0.014206
ZIP Code	0.013432	-0.029216	-0.028626	-0.016410	1.000000	0.011778	-0.004061	-0.017377	0.007383	0.000107	0.004704	0.019972	0.016990
Family	-0.016797	-0.046418	-0.052563	-0.157501	0.011778	1.000000	-0.109275	0.064929	-0.020445	0.061367	0.019994	0.014110	0.010354
CCAvg	-0.024675	-0.052012	-0.050077	0.645984	-0.004061	-0.109275	1.000000	-0.136124	0.109905	0.366889	0.015086	0.136534	-0.003611
Education	0.021463	0.041334	0.013152	-0.187524	-0.017377	0.064929	-0.136124	1.000000	-0.033327	0.136722	-0.010812	0.013934	-0.015004
Mortgage	-0.013920	-0.012539	-0.010582	0.206806	0.007383	-0.020445	0.109905	-0.033327	1.000000	0.142095	-0.005411	0.089311	-0.005995
Personal Loan	-0.024801	-0.007726	-0.007413	0.502462	0.000107	0.061367	0.366889	0.136722	0.142095	1.000000	0.021954	0.316355	0.006278
Securities Account	-0.016972	-0.000436	-0.001232	-0.002616	0.004704	0.019994	0.015086	-0.010812	-0.005411	0.021954	1.000000	0.317034	0.012627
CD Account	-0.006909	0.008043	0.010353	0.169738	0.019972	0.014110	0.136534	0.013934	0.089311	0.316355	0.317034	1.000000	0.175880
Online	-0.002528	0.013702	0.013898	0.014206	0.016990	0.010354	-0.003611	-0.015004	-0.005995	0.006278	0.012627	0.175880	1.000000
CreditCard	0.017028	0.007681	0.008967	-0.002385	0.007691	0.011588	-0.006689	-0.011014	-0.007231	0.002802	-0.015028	0.278644	0.004210

4.3 Data Pre-processing

The number of missing values are identified for cleaning an existing dataset. The total percentage of missing values in the column was identified using the Pandas Data Frame. The total number of rows with missing values is zero(0). It has used the Pandas dropna() method which was used to analyze the drop rows/columns with Null values.

```
dataset = dataset.dropna()
```

After applying the Pandas dropna() method which was used to analyze the drop rows/columns with Null values we can note by the below.

```
Age                 5000
Income              5000
ZIP Code            5000
Family              5000
CCAvg               5000
Education           5000
Mortgage            5000
Personal Loan       5000
Securities Account  5000
CD Account          5000
Online              5000
CreditCard          5000
dtype: int64
```

Dropping Irrelevant Columns

This step is certainly needed in EDA because sometimes there would be columns that we never use and in such cases dropping is useful. In this case, the columns such as Experience and Age may not be very relevant.
To remove the columns we have to find the correletion between Experience and Age. The correlation can be found below.

```
experience = dataset['Experience']
age = dataset['Age']
correlation = experience.corr(age)
correlation
```

```
0.9942148569683321
```

Finding Correlation by Pearson Method

```
corr = dataset.corr(method = 'pearson')
corr
```

	ID	Age	Experience	Income	ZIP Code	Family	CCAvg	Education	Mortgage	Personal Loan	Securities Account	CD Account	Online	Cr
ID	1.000000	-0.008473	-0.008326	-0.017695	0.013432	-0.016797	-0.024675	0.021463	-0.013920	-0.024801	-0.016972	-0.006909	-0.002528	
Age	-0.008473	1.000000	0.994215	-0.055269	-0.029216	-0.046418	-0.052012	0.041334	-0.012539	-0.007726	-0.000436	0.008043	0.013702	
Experience	-0.008326	0.994215	1.000000	-0.046574	-0.028626	-0.052563	-0.050077	0.013152	-0.010582	-0.007413	-0.001232	0.010353	0.013898	
Income	-0.017695	-0.055269	-0.046574	1.000000	-0.016410	-0.157501	0.645984	-0.187524	0.206806	0.502462	-0.002616	0.169738	0.014206	
ZIP Code	0.013432	-0.029216	-0.028626	-0.016410	1.000000	0.011778	-0.004061	-0.017377	0.007383	0.000107	0.004704	0.019972	0.016990	
Family	-0.016797	-0.046418	-0.052563	-0.157501	0.011778	1.000000	-0.109275	0.064929	-0.020445	0.061367	0.019994	0.014110	0.010354	
CCAvg	-0.024675	-0.052012	-0.050077	0.645984	-0.004061	-0.109275	1.000000	-0.136124	0.109905	0.366889	0.015086	0.136534	-0.003611	
Education	0.021463	0.041334	0.013152	-0.187524	-0.017377	0.064929	-0.136124	1.000000	-0.033327	0.136722	-0.010812	0.013934	-0.015004	
Mortgage	-0.013920	-0.012539	-0.010582	0.206806	0.007383	-0.020445	0.109905	-0.033327	1.000000	0.142095	-0.005411	0.089311	-0.005995	
Personal Loan	-0.024801	-0.007726	-0.007413	0.502462	0.000107	0.061367	0.366889	0.136722	0.142095	1.000000	0.021954	0.316355	0.006278	
Securities Account	-0.016972	-0.000436	-0.001232	-0.002616	0.004704	0.019994	0.015086	-0.010812	-0.005411	0.021954	1.000000	0.317034	0.012627	
CD Account	-0.006909	0.008043	0.010353	0.169738	0.019972	0.014110	0.136534	0.013934	0.089311	0.316355	0.317034	1.000000	0.175880	
Online	-0.002528	0.013702	0.013898	0.014206	0.016990	0.010354	-0.003611	-0.015004	-0.005995	0.006278	0.012627	0.175880	1.000000	
CreditCard	0.017028	0.007681	0.008967	-0.002385	0.007691	0.011588	-0.006689	-0.011014	-0.007231	0.002802	-0.015028	0.278644	0.004210	

We can find that Age, Experience columns are highly correlated so we can use only one column.

Dropping columns Experience and Id

The Id of a person will not help to build the model.
The Experience is highly correlated to Age. So we will drop Experience and Id columns.

```
dataset =  dataset.drop(['ID','Experience'],axis=1)
dataset.head(5)
```

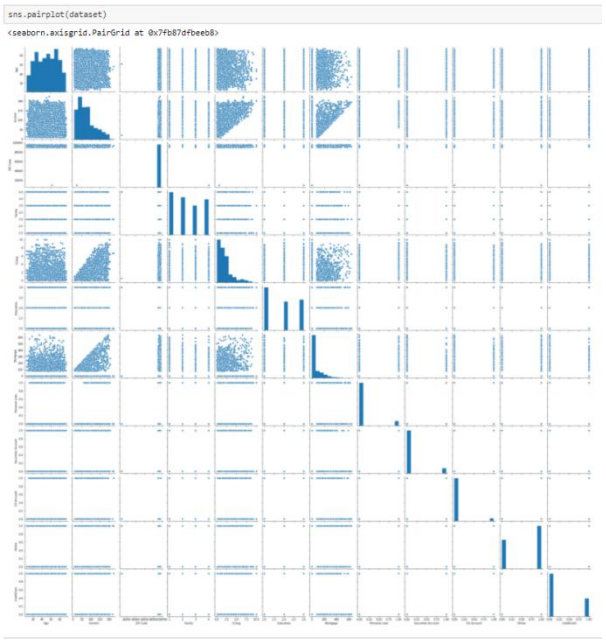
	Age	Income	ZIP Code	Family	CCAvg	Education	Mortgage	Personal Loan	Securities Account	CD Account	Online	CreditCard
0	25	49	91107	4	1.6	1	0	0	1	0	0	0
1	45	34	90089	3	1.5	1	0	0	1	0	0	0
2	39	11	94720	1	1.0	1	0	0	0	0	0	0
3	35	100	94112	1	2.7	2	0	0	0	0	0	0
4	35	45	91330	4	1.0	2	0	0	0	0	0	1

4.4 Visualization of data by Pair Plot

The following visualization derived through the Google Colab for display predicators.

```
sns.pairplot(dataset)
```

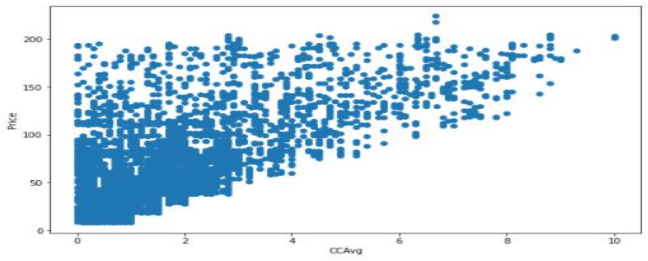
```
<seaborn.axisgrid.PairGrid at 0x7fb87dfbeeb8>
```



4.5 Visualization of data by Scatter Plot

The following visualization derived through the Google colab for displaying the CCAvg and Income.

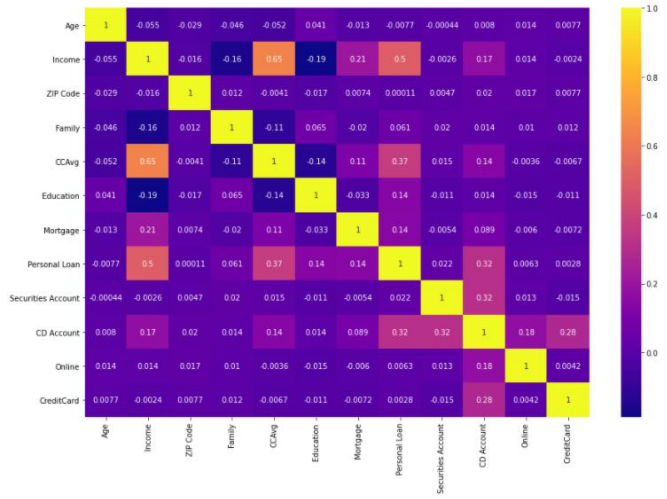
```
fig, ax = plt.subplots(figsize = (10,6))
ax.scatter(dataset['CCAvg'],dataset['Income'])
ax.set_xlabel('CCAvg')
ax.set_ylabel('Price')
plt.show()
```



Heatmap to find the correlation on dataset

```
fig,ax = plt.subplots(figsize = (15,10))
sns.heatmap(dataset.corr(),cmap='plasma',annot=True)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fb881cbae48>
```

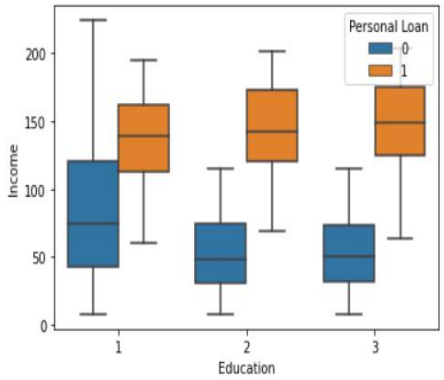


We can find that both CCAvg and Income are highly correlated.

Visualization of People by Income with boxplot

```
sns.boxplot(x='Education',y='Income',hue='Personal Loan', data = dataset)
```

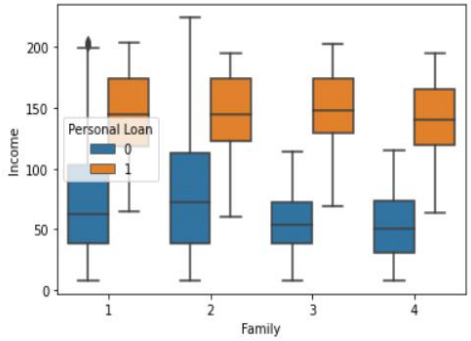
```
<matplotlib.axes._subplots.AxesSubplot at 0x7fb881f29358>
```



Visualization of Family by Income using Box plot.

```
sns.boxplot(x='Family',y='Income',hue='Personal Loan', data = dataset)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fb881fe2b00>
```



Transformation of Feature Variables

We found that both CCAvg and Income are highly correlated, so we have to scale the both attributes by importing PowerTransformer and using the method 'Yeo-Johnson'.

```
from sklearn.preprocessing import PowerTransformer
pt = PowerTransformer(method='yeo-johnson',standardize=False)
pt.fit(x['Income'].values.reshape(-1,1))
sns.distplot(pt.transform(x['Income'].values.reshape(-1,1)))
```

```
pt = PowerTransformer(method='yeo-johnson',standardize=False)
pt.fit(x['CCAvg'].values.reshape(-1,1))
sns.distplot(pt.transform(x['CCAvg'].values.reshape(-1,1)))
```

We should also transform the mortgage column as it as more 0 values.

```
x['MortgageN'] = pd.cut(x['Mortgage'],bins=[0,100,200,300,400,500,600,700],labels=[0,1,2,3,4,5,6],include_lowest=True)
x.drop('Mortgage',axis=1,inplace=True)
```

4.6 Training and Testing the Datasets

We have used a sci-kit learn library to split the dataset into training and testing sets

```
x = dataset[['Age','Income','Family','CCAvg','Online','CreditCard','Education', 'Mortgage', 'Securities Account', 'CD Account']]
y = dataset['Personal Loan'].values
```

Now we have to standardize the data and split the data in the ratio 70:30 for training and testing.

```
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler

scx = StandardScaler()
scy = StandardScaler()
x_train,x_test,y_train,y_test = train_test_split(x,y,test_size = 0.3, random_state = 0,stratify = y)
scx.fit_transform(x_train)
scx.transform(x_test)
```

5. Implementation of Classification Algorithms

5.1 Linear Regression: Logistic Regression

Logistic regression is a classification algorithm, used when the value of the variable is categorical in nature.The logistic regression is also known as the sigmoid function which helps in the easy representation in graphs. In this algorithm, the data should be imported and then trained. By using equation the logistic regression algorithm.

```
from sklearn.linear_model import LogisticRegression
logreg=LogisticRegression()
logreg.fit(x_train,y_train)
y_pred=logreg.predict(x_test)
```

Confusion matrix of Logistic Regression

```
from sklearn.metrics import confusion_matrix
print(confusion_matrix(y_test,y_pred))
```

```
[[1331   25]
 [  58   86]]
```

Accuracy of Logistic Regression

```
LRAccuracy = metrics.accuracy_score(y_test,y_pred) * 100
print(LRAccuracy)
```

```
94.46666666666667
```

We can find the Accuracy score for the test data is 94%. But in confusion matrix , the false negative are very much that is above 50%..

5.2 K-Nearest Neighbors (KNN)

The K-Nearest Neighbors algorithm is a simple, supervised machine learning algorithm that can be used to solve problems. Here we are using for predicting whether the customer is able to take loan or not, using a dataset acquired from Kaggle. The major drawback of KNN is it becoming significantly slows as the size of that data increases. It is used for classification and regression types of problems. Classification is done based on the majority vote to its neighbors. KNN is instance-based learning .

```
from sklearn.neighbors import KNeighborsClassifier
neigh = KNeighborsClassifier(n_neighbors=3)
neigh.fit(x_train,y_train)
y_pred = neigh.predict(x_test)
```

Confusion matrix of K-Nearest Neighbors

```
from sklearn.metrics import confusion_matrix
print(confusion_matrix(y_test,y_pred))
```

```
[[1324   32]
 [  77   67]]
```

Accuracy of K-Nearest Neighbors

```
KNNAccuracy = metrics.accuracy_score(y_test,y_pred)* 100
print(KNNAccuracy)
```

```
92.73333333333333
```

We can find the Accuracy score for the test data is 92%. But in confusion matrix , the false negative are 77 .

5.3 Support Vector Machine(SVM)

Support Vector Machine is an extremely popular supervised machine learning technique (having a pre-defined target variable) that can be used as a classifier as well as a predictor. A Support Vector Machine model represents the training data points as points in the feature space.

```
SVMAccuracy 94.97142857142858
```

The Accuracy of Support Vector Machine Algorithm is 94%.

Decision Tree

Decision Tree Algorithm is known as the supervised learning algorithm. Moreover, in supervised learning algorithms, the decision tree algorithm can be used for solving regression and classification problems. The Decision tree Algorithm is a decision support tool that uses a tree-like model. The goal of using a Decision Tree is to create a training model that can use to predict the target variable by learning simple decision rules inferred from training data .

```
print("DTAccuracy ",metrics.accuracy_score(y_test,y_pred)*100)
```

```
DTAccuracy  92.73333333333333
```

The Accuracy of Decision Tree Algorithm is 92%.

Random Forest

Random Forest Algorithm is used for supervised and classification, but mostly it's used for classification problems. It generates decision trees based on data samples and then gets the prediction from each of them. After prediction, it selects the most suitable solution by means of voting. It is an aggregate method that is better than a single decision tree because it decreases the over-fitting by averaging the result.

Confusion matrix of Random Forest

```
from sklearn.metrics import confusion_matrix
print(confusion_matrix(y_test,y_pred))
```

```
[[1353    3]
 [  15  129]]
```

Accuracy of Random Forest

```
print("RFAccuracy",metrics.accuracy_score(y_test,y_pred))
```

```
RFAccuracy 0.9886666666666667
```

The Accuracy of Random Forest is 98.8% with a false negatives of 15.

Naive Bayes

A Naive Bayes Classifier is a supervised machine-learning algorithm which assumes that features are statistically independent. Regardless of this assumption, it has proven itself to be a classifier with good results. Naive Bayes Classifier uses Baye's Theorem.

```
print("NBAccuracy",metrics.accuracy_score(y_test,y_pred))
```

```
NBAccuracy 0.8873333333333333
```

The Accuracy of Navie Baye's is 88 %.

Gradient Boosting

Gradient boosing is one of the most powerful machine learning algorithms. Essentially, the model creates a bunch of weak models and takes the best components of each one using gradient descent. Think of this like random forrest but instead of just averaging all the values, the model picks the best branches of each tree.

Confusion Matrix of Gradient Boosting

```
from sklearn.metrics import confusion_matrix
print(confusion_matrix(y_test,y_pred))
```

```
[[1347    9]
 [  12  132]]
```

Accuracy of Gradient Boosting

```
print("GBAccuracy",metrics.accuracy_score(y_test,y_pred) * 100)
```

```
GBAccuracy 98.6
```

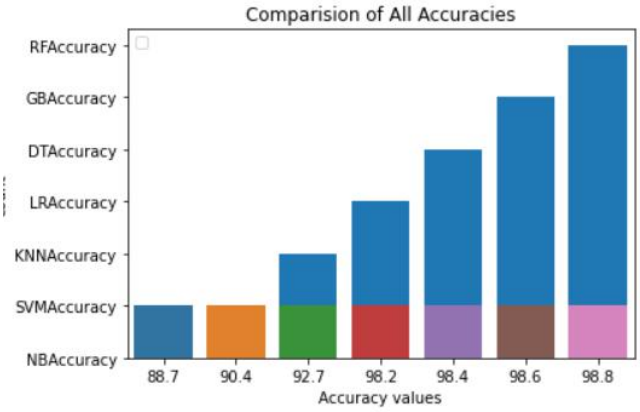
The Accuary of Gradient Boosting is 98.6% and false negative are 12 which is less among all the models.

6. Comparison of the best Algorithm by Bar Graph

6.1 Accuracy bar graph of all the Algorithms

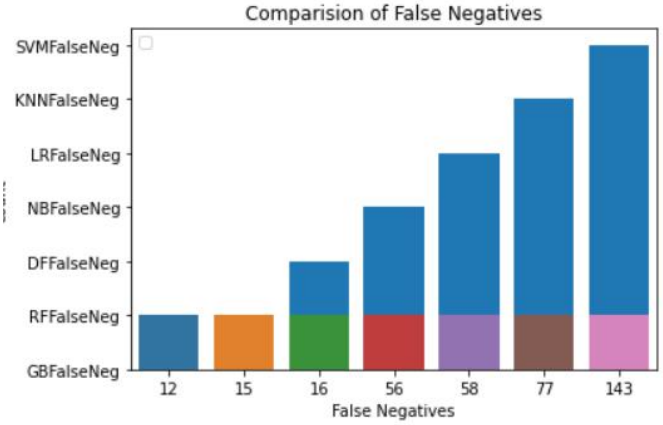
By analyzing all the seven Classification Algorithm of machine learning for the prediction of the customer liablity to take loan. It comes to know that Gradient Boosting is the most efficient algorithm out of all seven algorithms, as it has 98.6 % accuracy with 12 false negatives as shown below.

```
LRAccuracy :  94.46666666666667  False Negatives 58
DTAccuracy : 98.2  False Negatives 16
RFAccuracy : 98.86666666666667  False Negatives 15
SVMAccuracy : 90.46666666666667  False Negatives 143
NBAccuracy : 88.73333333333333  False Negatives 56
KNNAccuracy : 92.73333333333333  False Negatives 77
GBAccuracy : 98.6  False Negatives 12
```



From the above analysis we can say that RandomForest has the highest accuracies among all the model built.

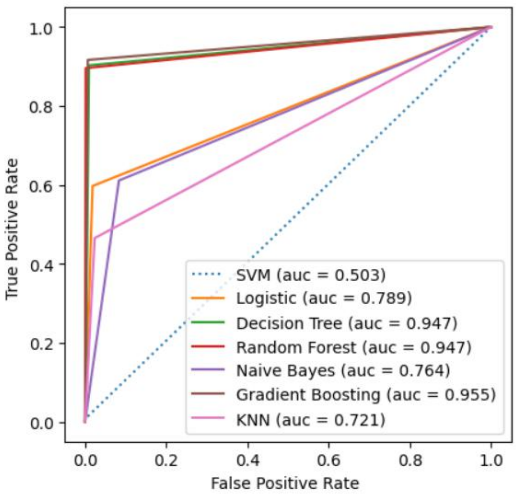
Comparing the False Negatives



Here Gradient Boosting has very less False negatives when compared with all the models.

6.2 ROC curve for Bank Modelling.

The Receiver Operating Characteristic (ROC) curve is a simple plot used to visualize the performance of a binary classifier. Good classification accuracy models should have significantly more true positives than false positives at all thresholds. Area Under the Curve (AUC) quantifies the model classification accuracy.



From the Figure we can find the Area under the curve is more for the Gradient Boosting with auc = 0.955.

7. Conclusion

The main aim of this research is to compare the accuracy of all the classification algorithms to evaluate whether the person is able to take the loan or not .We come to know that Gradient Boosting is a more appropriate algorithm to predict the customer who are capable. The primary motive of this research is the prediction of the customer with a high rate of accuracy by comparing all seven classification algorithms. Further, the accuracy of the Gradient Boosting model is 98.6 which is best out of all seven algorithms. The value under the AUC curve is 95 which is greater among all the model which is more satisfactory.

8. Future Work

Nowadays most of the data is computerized, the data is distributed everywhere but we're not utilizing it properly. By Analyzing the available data we can also use for unknown patterns. The motive of this future work is to predicting the customers who are liable to take the loan with high rate of accuracy by using the Classification Algorithms of Machine Learning. For predicting this with the help of different parameters, we can use Logistic Regression, Support Vector Machine, KNN, Decision Tree, Random Forest, Navie Bayes,Gradient Boosting, sklearn in machine learning Algorithm. Moreover, the model could be improved by using more data and techniques. The future scope of the paper is the prediction of the customer with all the parameters by using advanced techniques, with a high rate of accuracy and algorithms in less time complexity.

References

[1] V.V. Ramalingam*, Ayantan Dandapath, M Karthik Raja; Heart disease prediction using machine learning techniques : a survey – March 2018