**Data Analytics to acknowledge & verify the customer documentation for minimizing the Bank Fraud**

Financial institutions within the world need data analytics to carve a robust bank fraud detection technique. Advanced analytics provides the tools necessary for banks to acknowledge and act on suspicious patterns, quickly notify customers of fraud incidents and position themselves for faster settlements.

Traditionally, different financial institutions like bank have attempted fraud detection with manual practices, or rule-based solutions, which have been restricted in their success. A rule-based approach means that a complex set of standards for highlighting suspicious operations has to be established and reviewed manually.

While this can be effective in discovering anomalies, which adapt to known patterns, it is not capable of identifying fraud which follows modern, or unfamiliar patterns. This gives criminals the encouragement to develop ever more advanced techniques to go around the rules, and they themselves are leveraging new tools to achieve it.

**Problem Definition**

To recognize the purpose of the problem and the prediction target, we must define the project objectives appropriately. Therefore, to proceed with an analytical approach, we have to acknowledge the obstacles first. In the above project, we have used the machine learning models to predict & acknowledge the machine regarding the verification of the customer’s documentation. This will train the machine to minimize the Bank Fraud.

Following are the steps we have taken to complete the prediction;

1. ***Importing the data***
2. ***Cleaning and preparing the data***
3. ***Exploring and visualizing the data***
4. ***Splitting dataset into train\_set and test\_set***
5. ***Finding: accuracy score,classification report,confusion matrix,auc,roc curve***
6. ***Using varous algorithms:***

***a. KNeighborsClassifier b. SVC c. LogisticRegression d. DecisionTreeClassifier e. GaussianNB***

1. ***Printing confusion matrix and auc***

**Importing the data**

Importing of data is done for the use of Python Standard Library that has provided the functions to interact and concluded. Python has bundle of libraries for different tasks. Importing is not only a matter of using external libraries, but it also allows you to keep your code clean and organized.

**Cleaning and preparing the data**

After importing the dataset, it has been read and null value has been analysed. Heatmap has been used to find all the null values in the dataset. Even null values are found in the dataframe. Cleaning of data has been done as for example;

"Creadit\_History" column is replaced with column median value

"LoanAmount" column is replaced with column median value

"Loan\_Amount\_Term" column is replaced with column median value

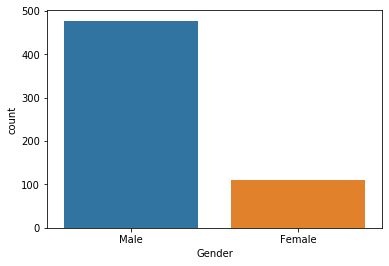
Apart from replacing the data, some od the columns have been dropped to get the maximum accuracy level.

**Exploring and visualizing the data**

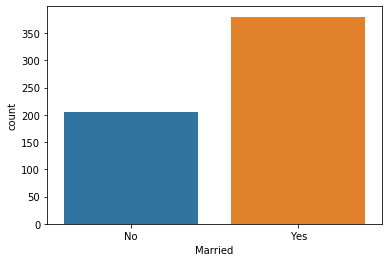
Once again, we have checked NaN value. From analysis point of view, now there is no NAN value.

Count plot has been used to visualize the column data.

As for example; we have visualized Gender in countplot. The following graph shows us the analysis;



Again, we have visualized Married in countplot. The graphical representation is given below;



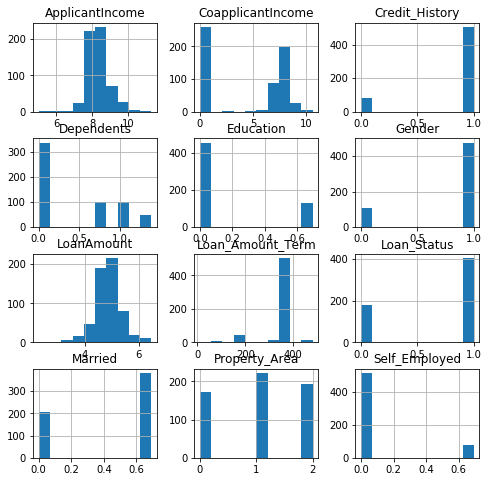
We havedropped unnecessary column like "Loan\_ID". Then, we have Create new data frame (ds\_str) for object dtype datas i.e.;

ds\_str=ds.drop(['ApplicantIncome','CoapplicantIncome','LoanAmount','Loan\_Amount\_Term','Credit\_History'],axis=1)

ds\_str.head()

**Label Encoder processing** has been used to change object to int data type. Firstly, we have created the new dataframe(ds\_str) for processing object columns into int datatype. Again, new dataframe (ds\_new) has been created for all int columns from main dataframe(ds). We have used LabelEncoder to convert and process the object columns into int for desired target. Lastly, both dataframes (ds,ds\_str) have been concated into ds\_new dataframe.

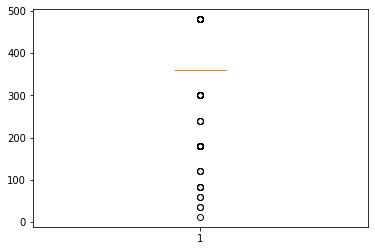
**Using hist plot** to visualize each columns distribution. Some examples are given below for a detailed understanding.



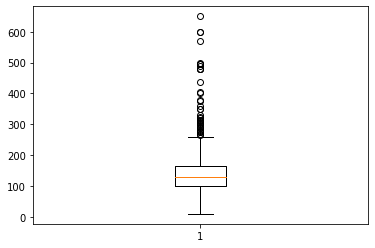
**Finding of Outliers**

We have thoroughly checked distribution of column and relationship between others. Then, accordingly we have dandled the data outliers, worked out for correlation columns and skewness within it. We have used boxplot and found many outliers in columns. Some of them with their graphical representations are as follows;

**Outliers in Credit History**



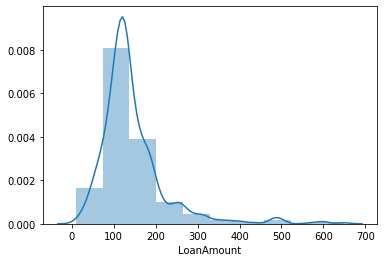
**Outliers in Loan\_amt\_term**



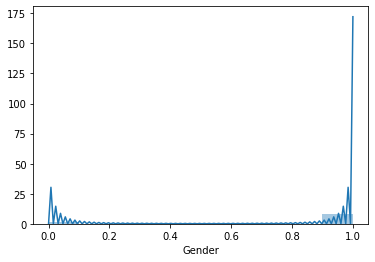
**Checking od Skewness of Data**

We have checked the distribution of columns data and found there is uneven skewness of different columns as for examples;

**Graphical representation of Skewness of data in Loan amount.**



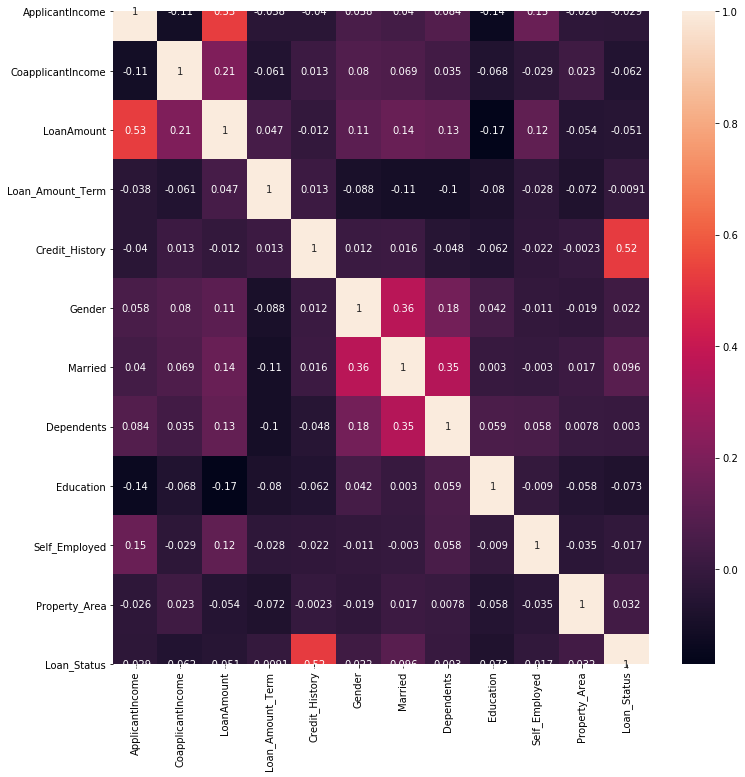
**Graphical representation of skewness of data in Gender.**



According to the graph we have found the data are alleviating and deteriorating which shows the skewness of data.

**Checking Correlation columns**

Using heatmap to check correlated columns. We have found some of the columns are highly correlated (CoapplicantIncome ,Credit\_History ,Education



Checking data skewness and Handling skewness of data by using numpy.log1p method i.e.;

ds\_new['ApplicantIncome']=np.log1p(ds\_new['ApplicantIncome'])

ds\_new['CoapplicantIncome']=np.log1p(ds\_new['CoapplicantIncome'])

ds\_new['LoanAmount']=np.log1p(ds\_new['LoanAmount'])

ds\_new['Married']=np.log1p(ds\_new['Married'])

ds\_new['Education']=np.log1p(ds\_new['Education'])

ds\_new['Self\_Employed']=np.log1p(ds\_new['Self\_Employed'])

ds\_new['Dependents']=np.log1p(ds\_new['Dependents'])

**Splitting dataset into train\_set and test\_set**

Separated the input and output data for classification models

x=ds\_new.drop(['Gender','Married','Education','Self\_Employed','Dependents',"CoapplicantIncome"],axis=1)

print(x.shape)

y=ds\_new['Loan\_Status']

print(y.shape)

(586, 6)

(586,)

Scaled input data for model and separated data in trained test for model

print(y\_train.shape,y\_test.shape)

(468,) (118,)

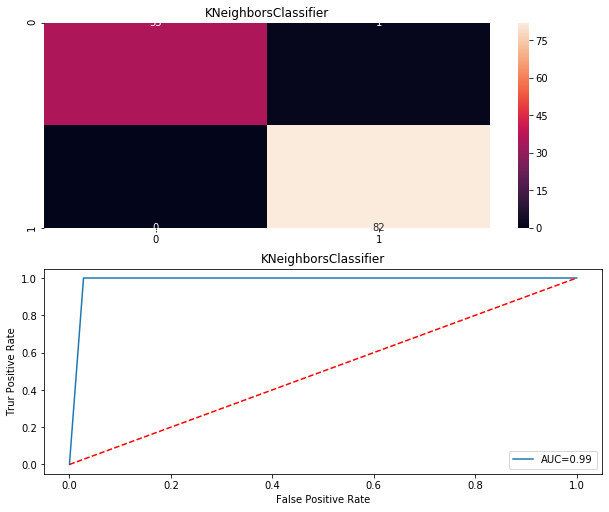
Checking done for train test shape and algorithms in variables has been declared.

**Finding: accuracy score,classification report,confusion matrix,auc,roc curve**

We have created anonymous program for flowing actions:

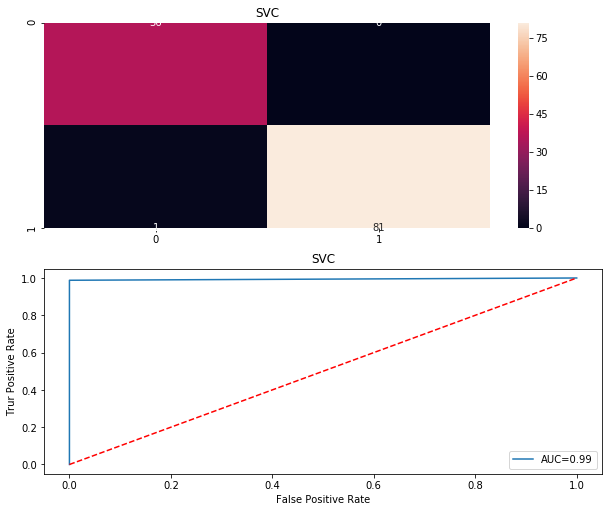
In this research, five models are implemented for loan predictions. They are, (1) KNeighborClassifier, Logistic Regression (LR), (2) Decision Tree (DT) (3) Support Vector Classifier (SVC), (4) , we have run the different algorithms after applying all the previous process of importing, cleaning the dataset. The different algorithms have been shown using its graphical representation;

**Graphical Representation of KNeighborClassifier Algorithm**



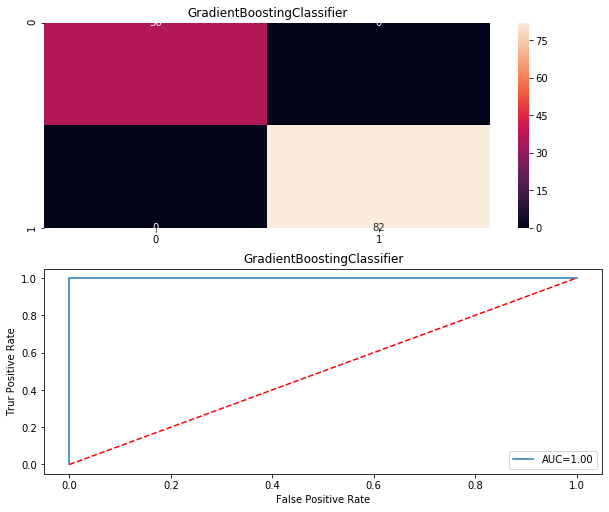
It is a classification method. In the given algorithm, we have plotted each data item as a point in n-dimensional space (where n is number of features we have) with the value of each feature is being the value of a specific coordinate.

**Graphical Representation of SVC Algorithm**



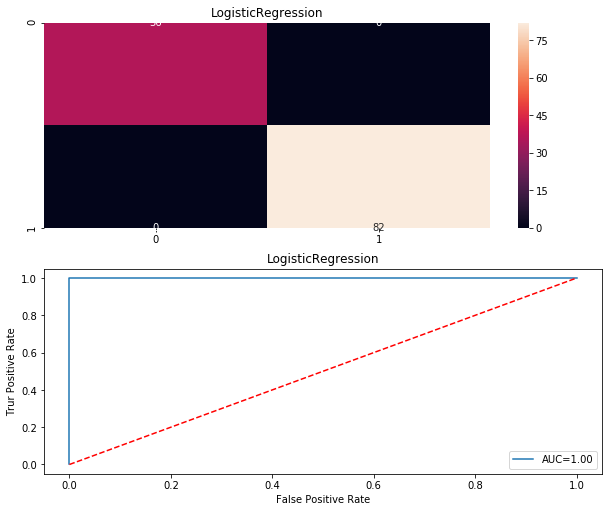
The objective of a Linear SVC (Support Vector Classifier) is to ensemble the information/ data one has provided, this has returned the "best fit" hyperplane which divides and categorizes, the dataset. From there, after getting the hyperplane, you will later fetched some features to our classifier to ascertain what the "predicted" class is.

**Graphical Representation of GradientBoostingClassifier Algorithm**



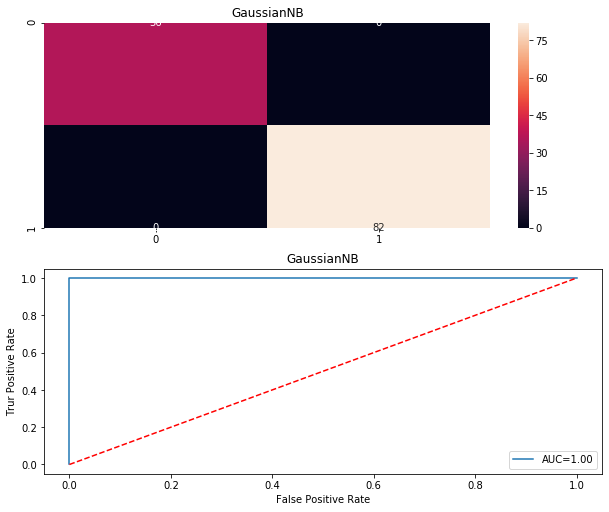
GradientBoostingClassifier (GBC) is a boosting algorithm used when we deal with plenty of data to make a prediction with high prediction power. Boosting is an ensemble of learning algorithms that combines the prediction of several base estimators in order to improve robustness over estimator. For developing a strong predictor, GBC unites the multiple weak or average predictors.

**Graphical Representation of LogisticRegression Algorithm**



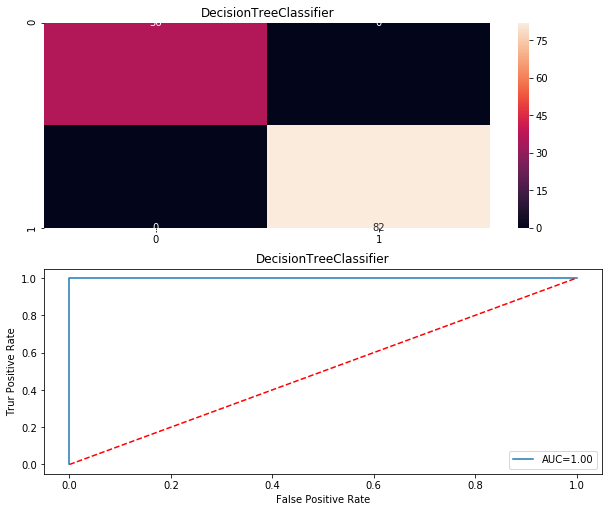
LogisticRegression is utilized to estimate discrete values or Binary values like 0 or 1, yes or no, true or false ) based on given set of independent variable(s). In simple words, it predicts the probability of occurrence of an event. Its output values lie between 0 and 1 as it predicts the probability.

**Graphical Representation of GaussianNB Algorithm**



GaussianNB is utilise in classification and it assumes that features follow a normal distribution. According to a normal (or Gaussian) distribution, a typical assumption is that the continuous values associated with each class are distributed while dealing with continuous data. This algorithm is very firmly known for multi class prediction feature. Here the probability of multiple classes of target variable easily get predicted using GaussianNB.

**Graphical Representation of DecisionTreeClassifier Algorithm**



It is a type of learning algorithm which is mostly utilized for classification dilemmas. Remarkably, it works for both categorical and continuous dependent variables. This is completed which was based on the most important attributes/ independent variables to make as distinct groups as possible.

**Printing confusion matrix and auc**

After applying the different algorithm for prediction, we must check accuracy score of algorithms. We have got the diverse accuracy score after using the different algorithms.

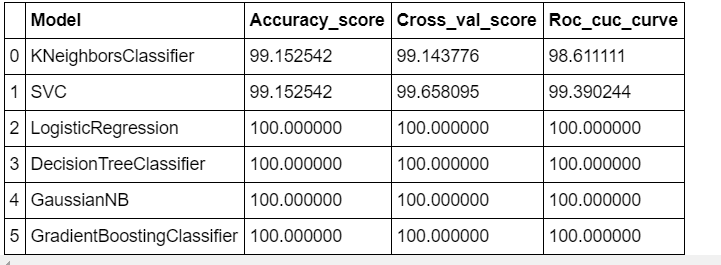
Simultaneously, we have checked the value of Cross Val Score of all the applied algorithms.

**Then,** we have found the roc auc score.

**Lastly**, we have found the classification report

1. Visualize confusion matrix of each algorithm
2. Visualize auc curve of each algorithm

Different algorithms scores in table format;



From above tabular representation of different algorithm, we have found that the LogisticRegression, DecisionTreeClassifier, GaussianNB, GradientBoostingClassifier algorithm model "accuracy score,Cross\_val\_score ,Roc\_cuc\_curve" slightly better than other algorithms. And as we have to choose one of the best, so we are choosing the LogisticRegression model.

# Conclusion:

Depending upon the given features that were used as the inputs to the machine learning to predict whether to give loan to a person or not. We have used the EDA process to find out the correlated columns and data analysis. And later we have used the different classification algorithms like KNeighborsClassifier, SVC, LogisticRegression, DecisionTreeClassifier, GradientBoostingClassifier & GaussianNB.

After using all the above approaches, we have concluded that LogisticRegression provides the highest level of accuracy score for machine learning i.e., approx... 100%.