**Artificial Intelligence**

**CSE437**

**Final Project Report**

****

**Topic : Loan Prediction using Svm, Naive bayes and Random Forest  
Name and ID :**1. Maliha Tabassum - 19101212

2. MD. Rahat Khan- 20101616

3. Abdullah Al Fahim- 18201099  
**Section : 0**2

**Group no :** 17  
**Date of Submission :** 02.01.2023

**Department of Computer Science and Engineering   
BRAC University**

**Introduction:**

Finding if a person is actually eligible for getting the loan is actually a very tedious process. For this purpose, various types of investigation and background check is required. To make the task easier this project works to actually find if any person is eligible for getting the loan and if he/she is capable of repaying the loan. Thus it helps in accelerating the process of finding eligible borrowers of loan using Bank data.

**Methodology:**

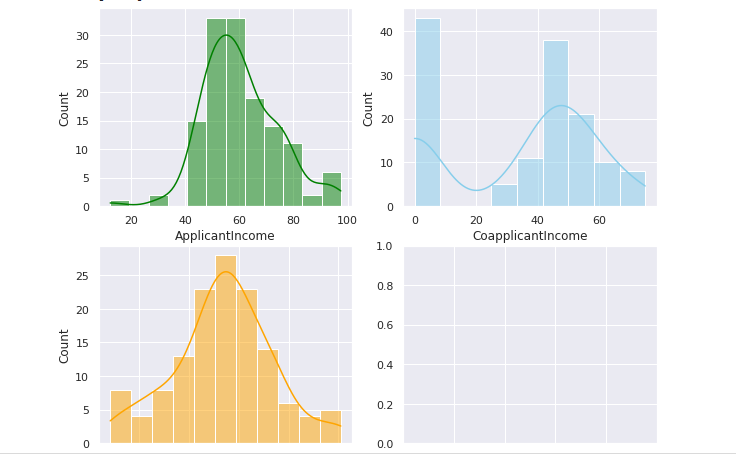
At first the dataset was selected and analyzed and then it was pre-processed using various techniques. Different data sets require different pre-processing techniques. And after proper pre-processing it is trained in model and tested. Which ultimately gives accuracy and visualization. Models used in this project are random forest, SVC and Gaussian Naive Bayes.

**Dataset Description:**

The dataset was selected from a website. It is a labeled and supervised dataset with 13 variables where there are 8 categorical variables, 4 continuous variables and 1 variable to accommodate the loan ID.

|  |  |  |
| --- | --- | --- |
| **Variable Name** | **Sample Data** | **Data Type** |
| Loan\_ID | LP001002; LP001003; | integer |
| Gender | Male; Female | integer |
| Married | Married; Not Married | integer |
| Dependents | 0; 1; 2; 3+ | integer |
| Education | Graduate; Undergraduate | integer |
| Self\_Employed | Yes; No | integer |
| ApplicantIncome | 5849; 4583; | float |
| CoapplicantIncome | 1508; 2358; | float |
| LoanAmount | 128; 66; | float |
| Loan\_Amount\_Term | 360; 120; | float |
| Credit\_History | 0; 1 | float |
| Property\_Area | Rural; Semiurban; Urban | integer |
| Loan\_Status | Y; N | integer |

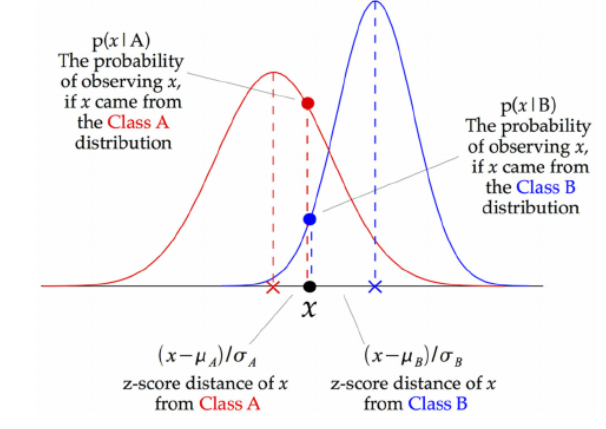
**Pre-Processing Techniques applied:**

1. **Dropping Unnecessary Values** Here, Some values are unnecessary like the loan\_ID which will not be used in finding out the accuracy .
2. **Data Imputation** Through data imputation we substitute or input a missing value in the data sheet.
   1. **Categorical Variables** Here, we have used ‘mode’ in order to perform data imputation as it is a categorical variable.
   2. **Numerical Variables.** Here, we have used ‘mean’ in order to perform data imputation as it is a numerical variable.
3. **One-hot Encoding** One-hot encoding is basically used to convert categorical value into numerical value in order to make it convenient to read for machine learning.
4. **Removing Outliers & Infinite values.** We use outliners to sort and cancel out the values that are out of range. This makes things easy for the calculation.
5. **Skewed Distribution Treatment.** Skewed distribution treatment is used to normalize the distribution of the data as we can see that some datas are positively skewed.  
     
   
6. **Features Separating** Dependent features are separated from the independent features.
7. **Smote Technique** Smote technique is used to remove the imbalance between accepted and rejected loans.
8. **Data Normalization** Data normalization will be performed to normalize the range of independent variables or features of data.
9. **Splitting Data Set** Data is splitted to 80% and 20% where 80% is used to train the data and 20% is used for testing.

**Models:**

**Gaussian Naive Bayes**

It is an approach to create a simple model where continuous valued features are distributed according to Gaussian(Normal) Distribution with no co-variance (independent dimensions) between dimensions. This model can be fit by simply finding the mean and standard deviation of the points within each label, which is all why needed to define such a distribution.



At every data point, the z-score distance between that point and each class-mean is calculated, namely the distance from the class mean divided by the standard deviation of that class.

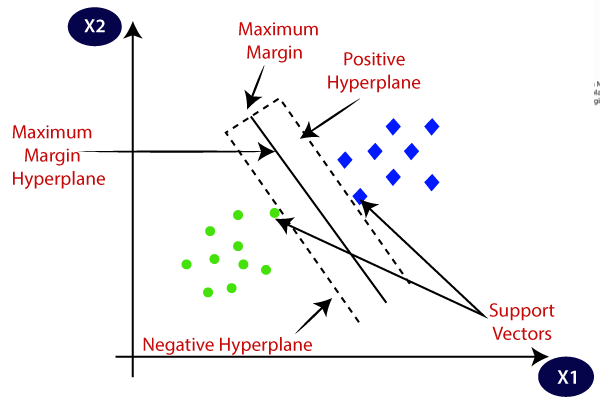
Thus, we see that the Gaussian Naive Bayes has a slightly different approach and can be used efficiently.

**SVM:**

Support Vector Machine or SVM is one of the most popular Supervised Learning algorithms, which is used for Classification as well as Regression problems. However, primarily, it is used for Classification problems in Machine Learning.

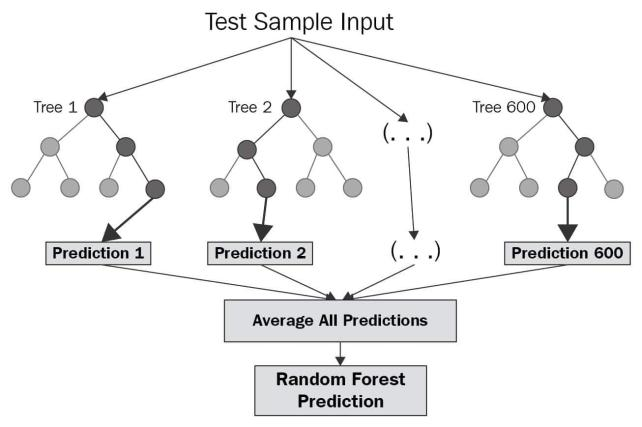
The goal of the SVM algorithm is to create the best line or decision boundary that can segregate n-dimensional space into classes so that we can easily put the new data point in the correct category in the future. This best decision boundary is called a hyperplane.

SVM chooses the extreme points/vectors that help in creating the hyperplane. These extreme cases are called support vectors, and hence the algorithm is termed as Support Vector Machine. Consider the below diagram in which there are two different categories that are classified using a decision boundary or hyperplane:



**Random Forest**

Random forests or random decision forests are an ensemble learning method for classification, regression and other tasks that operate by constructing a multitude of decision trees at training time.

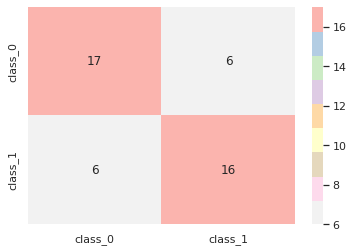


Random Forests, for a predefined number N, train N decision trees on the training data, with each tree being independent and random, and have each decision tree make its own prediction. The final prediction is either the average of all the predictions made by individual trees (Regression), or the most common prediction made by the trees (Classification)

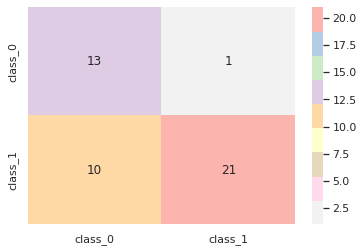
**Results:**

Confusion matrix A confusion matrix is a table that is used to describe the performance of a classification model on a set of test data for which the true values are known. It makes it easy to see if the model is confusing classes.

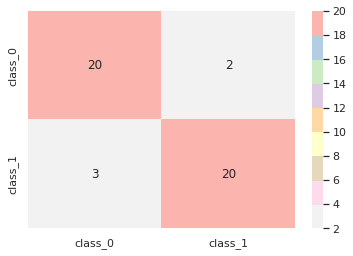
**Confusion matrix of Gaussian NB:**



**Confusion matrix of Support Vector Machine (SVM):**



**Confusion matrix of Random Forest:**



Based on the confusion matrices of all our models, we can see that all models have performed well and none has confused two classes.

**Accuracy score, F1 score, Precision score and Recall score:**

We can use some metrics like accuracy score, precision score, recall score, and F1 score to measure the performance of our models. The accuracy score is the ratio of correctly classified data instances and the total number of data instances. Accuracy is not a good metric when it comes to an unbalanced dataset. Precision is the ability of a classifier to label a negative sample as negative. The recall is the ability of the model to predict the positives out of samples that are true positives. F1 takes both precision and recall into account so if the precision score and recall score are both high, the F1 score will be high as well. For all metrics, 1 is the best result and 0 is the worst.

We have made a table with all the metrics that we had stored before. We can see the performances of the models at a glance from this table.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Model** | **Accuracy Score** | **F1 Score** | **Precision Score** | **Recall Score** |
| **Gaussian NB** | 0.73 | 0.74 | 0.74 | 0.74 |
| **Random Forest** | 0.89 | 0.89 | 0.91 | 0.87 |
| **SVM** | 0.76 | 0.70 | 0.93 | 0.57 |

**References:**

1. [4.4.3 Calculating the mode (statcan.gc.ca)](https://www150.statcan.gc.ca/n1/edu/power-pouvoir/ch11/mode/5214873-eng.htm)
2. [sklearn.preprocessing.MinMaxScaler — scikit-learn 1.1.2 documentation](https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.MinMaxScaler.html)
3. [SVM Machine Learning Tutorial – What is the Support Vector Machine Algorithm, Explained with Code Examples (freecodecamp.org)](https://www.freecodecamp.org/news/svm-machine-learning-tutorial-what-is-the-support-vector-machine-algorithm-explained-with-code-examples/#:~:text=SVMs%20are%20used%20in%20applications,linear%20and%20non%2Dlinear%20data.)
4. [SMOTE for Imbalanced Classification with Python (machinelearningmastery.com)](https://machinelearningmastery.com/smote-oversampling-for-imbalanced-classification/)
5. [Data Science in 5 Minutes: What is One Hot Encoding? (educative.io)](https://www.educative.io/blog/one-hot-encoding)