A PROJECT REPORT ON

LOAN APPROVAL PREDICTION SYSTEM USING MACHINE LEARNING

Submitted in partial fulfillment for the requirement of the award of

TRAINING

IN

Data Analytics, Machine Learning and AI using Python

Diginique Techlabs
Ideal for Geeks

Submitted By

Manideep Kalyanam (Chaitanya Bharathi Institute of Technology, Hyderabad)

Under the guidance of

Mr. Bipul Shahi

ACKNOWLEDGEMENT

My sincere gratitude and thanks towards my project paper guide Mr. Bipul Shahi, Corporate Trainer, Developer, Traveler!! IOT, Artificial Intelligence, Robotics, Cloud Computing, Android Apps!!!

It was only with his backing and support that I could complete the report. He provided me all sorts of help and corrected me if ever seemed to make mistakes. I have no such words to express my gratitude. I acknowledge my sincere gratitude to the lecturers, research scholars and the lab technicians for their valuable guidance and helping attitude even in their very busy schedule. And at last but not the least, I acknowledge my dearest parents for being such a nice source of encouragement and moral support that helped me tremendously in this aspect. I also declare to the best of my knowledge and belief that the Project Work has not been submitted anywhere else.

PROBLEM STATEMENT

A company wants to automate the loan eligibility process based on customer details provided while filling online application form. These details are Gender, Marital Status, Education, Number of Dependents, Income, Loan Amount, Credit History and others. To automate this process, we will build Machine Learning models (using different algorithms) and finally give a best fit model which can accurately predict which customer's loan it should approve and which to reject, in order to minimize the risk of loan default.

INTRODUCTION

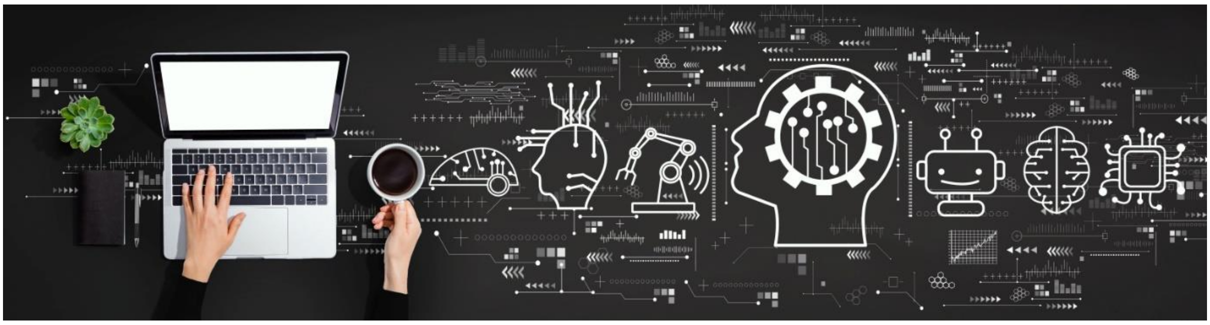
Loans have made our life easier, providing us the financial leverage that extends beyond our earnings. Be it Credit Card, Home Loan, Personal Loan, Vehicle loan etc., loans are the credit extended to us by lenders (usually a corporation, financial institution, or bank) on fulfilling certain key parameters. However, getting a loan in India can often be a tiresome process for the un-initiated, but not for individuals with a good credit score.

Distribution of the loans is the core business part of almost every bank. The main portion of the bank's assets directly comes from the profit earned from the loans distributed by the banks. The prime objective in banking environment is to invest their assets in safe hands where it is. Today many banks/financial companies approve loan after a regress process of verification and validation but still there is no surety whether the chosen applicant is the deserving right applicant out of all applicants. Through this system we can predict whether we can approve loan to that particular applicant or not and the whole process of validation of features is automated by machine learning technique. The disadvantage of this model is that it emphasizes different weights to each factor but in real life sometime loan can be approved on the basis of single strong factor only, which is not possible through this system.

Loan Prediction is very helpful for employees of banks as well as for the applicant also. The aim of this project is to provide quick, immediate and easy way to choose the deserving applicants. It can provide special advantages to the bank. The Loan Prediction System can automatically calculate the weight of each features taking part in loan processing and on new test data same features are processed with respect to their associated weight. A time limit can be set for the applicant to check whether his/her loan can be sanctioned or not.

Conceptual Approach

Machine learning:



Machine learning is a method of data analysis that automates analytical model building. It is a branch of artificial intelligence based on the idea that systems can learn from data, identify patterns and make decisions with minimal human intervention.

The process of learning begins with observations or data, such as examples, direct experience, or instruction, in order to look for patterns in data and make better decisions in the future based on the examples that we provide. **The primary aim is to allow the computers learn automatically** without human intervention or assistance and adjust actions accordingly.

Some machine learning methods:

Machine learning algorithms are often categorized as supervised or unsupervised.

- **Supervised machine learning algorithms** can apply what has been learned in the past to new data using labeled examples to predict future events. Starting from the analysis of a known training dataset, the learning algorithm produces an inferred function to make predictions about the output values. The system is able to provide targets for any new input after sufficient training. The learning algorithm can also compare its output with the correct, intended output and find errors in order to modify the model accordingly.

- **Unsupervised machine learning algorithms** are used when the information used to train is neither classified nor labeled. Unsupervised learning studies how systems can infer a function to describe a hidden structure from unlabeled data. The system doesn’t figure out the right output, but it explores the data and can draw inferences from datasets to describe hidden structures from unlabeled data.

- **Semi-supervised machine learning algorithms** fall somewhere in between supervised and unsupervised learning, since they use both labeled and unlabeled data for training – typically a small amount of labeled data and a large amount of unlabeled data. The systems that use this method are able to considerably improve learning accuracy. Usually, semi-supervised learning is chosen when the acquired labeled data requires skilled and relevant resources in order to train it / learn from it. Otherwise, acquiring unlabeled data generally doesn't require additional resources.

- **Reinforcement machine learning algorithms** is a learning method that interacts with its environment by producing actions and discovers errors or rewards. Trial and error search and delayed reward are the most relevant characteristics of reinforcement learning. This method allows machines and software agents to automatically determine the ideal behavior within a specific context in order to maximize its performance. Simple reward feedback is required for the agent to learn which action is best; this is known as the reinforcement signal.

Neural Networks:

A neural network is a type of machine learning which models itself after the human brain, creating an artificial neural network that via an algorithm allows the computer to learn by incorporating new data.

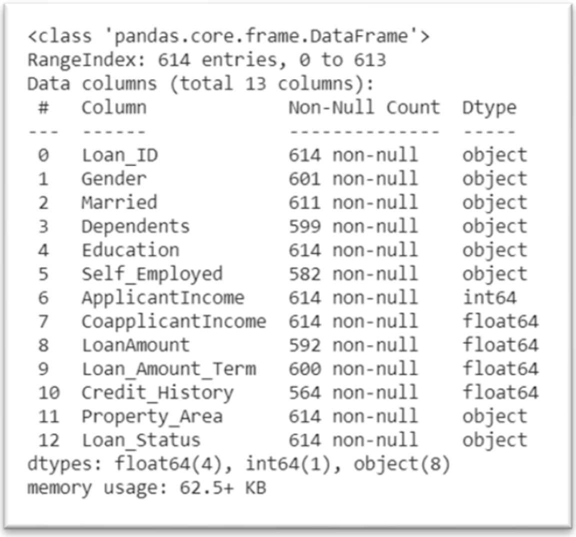
While there are plenty of artificial intelligence algorithms these days, neural networks are able to perform what has been termed deep learning. While the basic unit of the brain is the neuron, the essential building block of an artificial neural network is a perceptron which accomplishes simple signal processing, and these are then connected into a large mesh network.

The computer with the neural network is taught to do a task by having it analyse training examples, which have been previously labelled in advance. A common example of a task for a neural network using deep learning is an object recognition task, where the neural network is presented with a large number of objects of a certain type, such as a cat, or a street sign, and the computer, by analysing the recurring patterns in the presented images, learns to categorize new images.

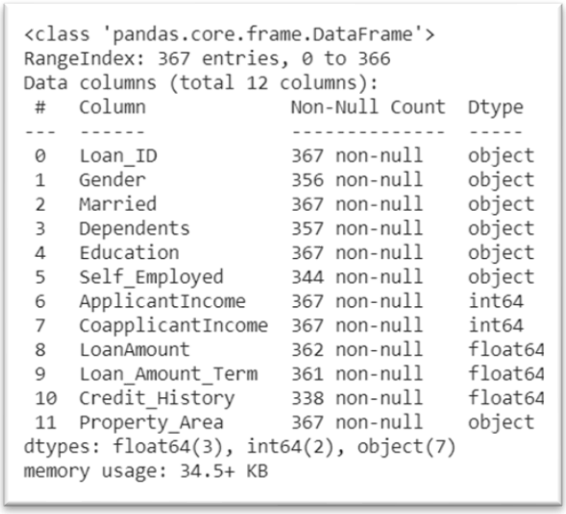
Dataset Description

Link for data: https://www.kaggle.com/sethirishabh/finance-company-loan-data

We will get two different datasets from the source. One is the Train_set and another one is Test_set. The information of the two datasets are given below:



```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 614 entries, 0 to 613
Data columns (total 13 columns):
 #   Column             Non-Null Count  Dtype
---  ------             --------------  -----
 0   Loan_ID            614 non-null    object
 1   Gender             601 non-null    object
 2   Married            611 non-null    object
 3   Dependents         599 non-null    object
 4   Education          614 non-null    object
 5   Self_Employed      582 non-null    object
 6   ApplicantIncome    614 non-null    int64
 7   CoapplicantIncome  614 non-null    float64
 8   LoanAmount         592 non-null    float64
 9   Loan_Amount_Term   600 non-null    float64
 10  Credit_History     564 non-null    float64
 11  Property_Area      614 non-null    object
 12  Loan_Status        614 non-null    object
dtypes: float64(4), int64(1), object(8)
memory usage: 62.5+ KB
```



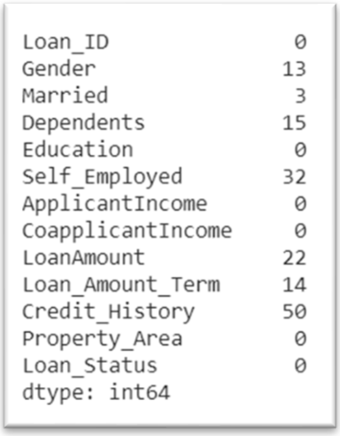
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 367 entries, 0 to 366
Data columns (total 12 columns):
 #   Column             Non-Null Count  Dtype
---  ------             --------------  -----
 0   Loan_ID            367 non-null    object
 1   Gender             356 non-null    object
 2   Married            367 non-null    object
 3   Dependents         357 non-null    object
 4   Education          367 non-null    object
 5   Self_Employed      344 non-null    object
 6   ApplicantIncome    367 non-null    int64
 7   CoapplicantIncome  367 non-null    int64
 8   LoanAmount         362 non-null    float64
 9   Loan_Amount_Term   361 non-null    float64
 10  Credit_History     338 non-null    float64
 11  Property_Area      367 non-null    object
dtypes: float64(3), int64(2), object(7)
memory usage: 34.5+ KB
```

(i) Train_set (ii) Test_set

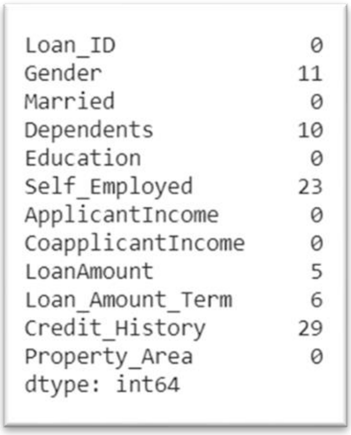
The datasets consist of customers' details like Gender, Marital Status, Education, Employment Status, Income, Loan Amount, Credit History, Loan Status and others as shown in the above figure.

Train_set consists of 614 rows and 13 columns. Test_set consists of 367 rows and 12 columns (i.e., It doesn't contain the label (Loan_Status)). Based on the features in Train_set we need to train our model and predict the label in Test_set.

The dataset consists of some null values. The null value count of Train_set and Test_set corresponding to the columns are given below:



```
Loan_ID               0
Gender               13
Married               3
Dependents           15
Education             0
Self_Employed        32
ApplicantIncome       0
CoapplicantIncome     0
LoanAmount           22
Loan_Amount_Term     14
Credit_History       50
Property_Area         0
Loan_Status           0
dtype: int64
```



```
Loan_ID               0
Gender               11
Married               0
Dependents           10
Education             0
Self_Employed        23
ApplicantIncome       0
CoapplicantIncome     0
LoanAmount            5
Loan_Amount_Term      6
Credit_History       29
Property_Area         0
dtype: int64
```

We will fill all the null values using different techniques while training our model.

About few columns:

Employment:
Lenders weigh your employment history and current engagement to ensure that your source of income is reliable. A lender wants to be certain that your employer is financially sound, with no history of outstanding or delay in paying employees their salaries.

Income:
Your income represents your repayment capacity. Lenders assess your income capacity in the backdrop of existing debt obligations, dependents, source, and duration.

Collateral (Property):
The collateral you provide to the bank while applying could help you secure the loan easier and sooner. As the loan amount is a percentage of the assessed value of the collateral, a high-value asset could mean more credit sanctioned for your use. So, based on Property area the value of asset is assessed.

Credit history: Your credit history is indicative of your future repayment behaviour, based on your pattern in settling past loans. It helps the lender to know if you will be punctual and regular with your payments.

We trained our model using some classification Machine learning algorithms.

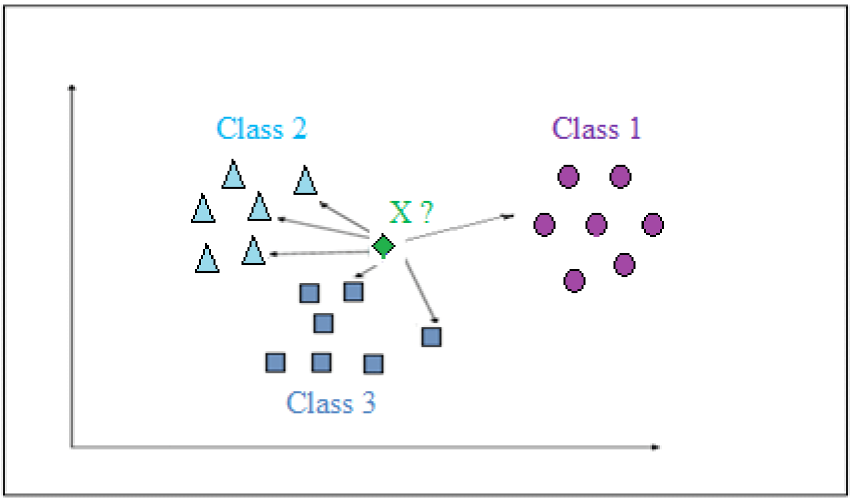
They are K-Nearest Neighbours, Logistic regression, Support vector machine, Random forest classifier and Gaussian Naïve Bayes.

K-Nearest Neighbours:

K-Nearest Neighbours is one of the most basic yet essential classification algorithms in Machine Learning. It belongs to the supervised learning domain and finds intense application in pattern recognition, data mining and intrusion detection.

The K-Nearest Neighbours (KNN) algorithm is a simple, easy-to-implement supervised machine learning algorithm that can be used to solve both classification and regression problems.

The KNN algorithm assumes that similar things exist in close proximity. In other words, similar things are near to each other. KNN captures the idea of similarity (sometimes called distance, proximity, or closeness) with some mathematics we might have learned in our childhood—calculating the distance between points on a graph. There are other ways of calculating distance, and one way might be preferable depending on the problem we are solving. However, the straight-line distance (also called the Euclidean distance) is a popular and familiar choice.



Logistic Regression:

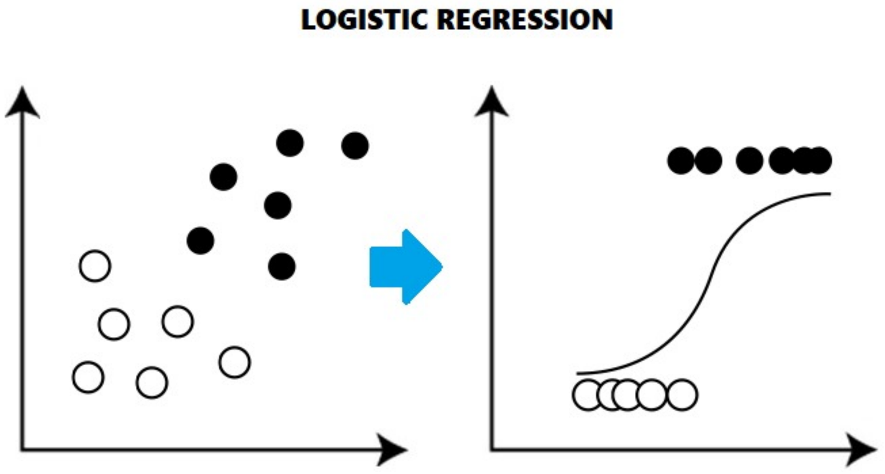
Logistic regression is one of the most popular Machine Learning algorithms, which comes under the Supervised Learning technique. It is used for predicting the categorical dependent variable using a given set of independent variables.

Logistic regression predicts the output of a categorical dependent variable. Therefore, the outcome must be a categorical or discrete value. It can be either Yes or No, 0 or 1, True or False, etc. but instead of giving the exact value as 0 and 1, it gives the probabilistic values which lie between 0 and 1.

Logistic Regression is much similar to the Linear Regression except that how they are used. Linear Regression is used for solving Regression problems, whereas Logistic regression is used for solving the classification problems.

In Logistic regression, instead of fitting a regression line, we fit an "S" shaped logistic function, which predicts two maximum values (0 or 1).

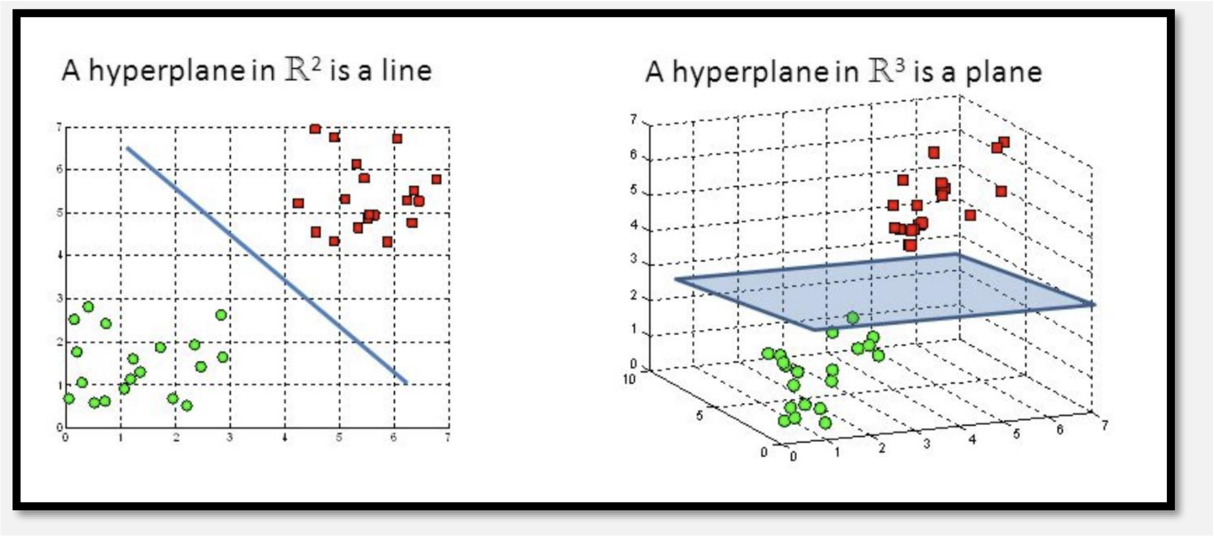
Logistic Regression is a significant machine learning algorithm because it has the ability to provide probabilities and classify new data using continuous and discrete datasets.



Support Vector Machine:

The objective of the support vector machine algorithm is to find a hyperplane in an N-dimensional space (N — the number of features) that distinctly classifies the data points.

To separate the two classes of data points, there are many possible hyperplanes that could be chosen. Our objective is to find a plane that has the maximum margin, i.e., the maximum distance between data points of both classes. Maximizing the margin distance provides some reinforcement so that future data points can be classified with more confidence.



Hyperplanes are decision boundaries that help classify the data points. Data points falling on either side of the hyperplane can be attributed to different classes. Also, the dimension of the hyperplane depends upon the number of features. If the number of input features is 2, then the hyperplane is just a line. If the number of input features is 3, then the hyperplane becomes a two-dimensional plane. It becomes difficult to imagine when the number of features exceeds three.

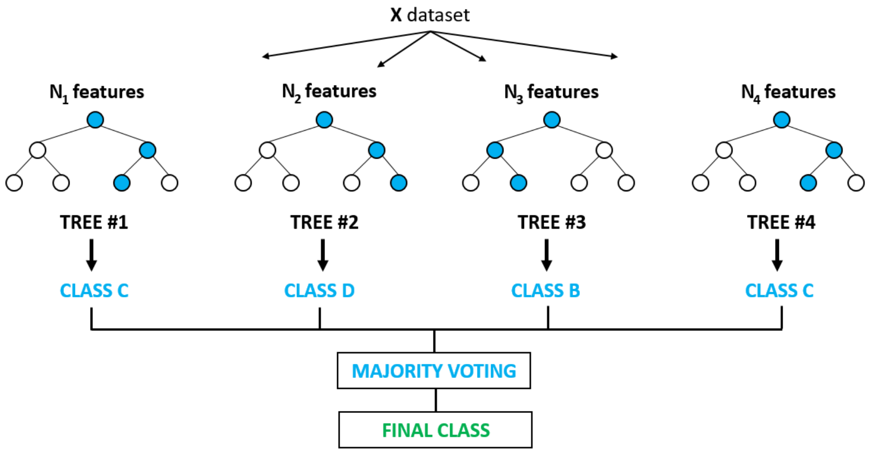
Support vectors are data points that are closer to the hyperplane and influence the position and orientation of the hyperplane. Using these support vectors, we maximize the margin of the classifier. Deleting the support vectors will change the position of the hyperplane. These are the points that help us build our SVM.

Random Forest Classifier:

Random forest is a supervised learning algorithm. The "forest" it builds, is an ensemble of decision trees, usually trained with the “bagging” method. The general idea of the bagging method is that a combination of learning models increases the overall result.

Random forest builds multiple decision trees and merges them together to get a more accurate and stable prediction.

Random forest in classification: Since classification is sometimes considered the building block of machine learning. Below you can see how a random forest would look like with four trees:



Random forest has nearly the same hyperparameters as a decision tree or a bagging classifier. Fortunately, there's no need to combine a decision tree with a bagging classifier because you can easily use the classifier-class of random forest. With random forest, you can also deal with regression tasks by using the algorithm's regressor.

Random forest adds additional randomness to the model, while growing the trees. Instead of searching for the most important feature while splitting a node, it searches for the best feature among a random subset of features. This results in a wide diversity that generally results in a better model.

Therefore, in random forest, only a random subset of the features is taken into consideration by the algorithm for splitting a node. You can even make trees more random by additionally using random thresholds for each feature rather than searching for the best possible thresholds (like a normal decision tree does).

Naive Bayes Algorithm:

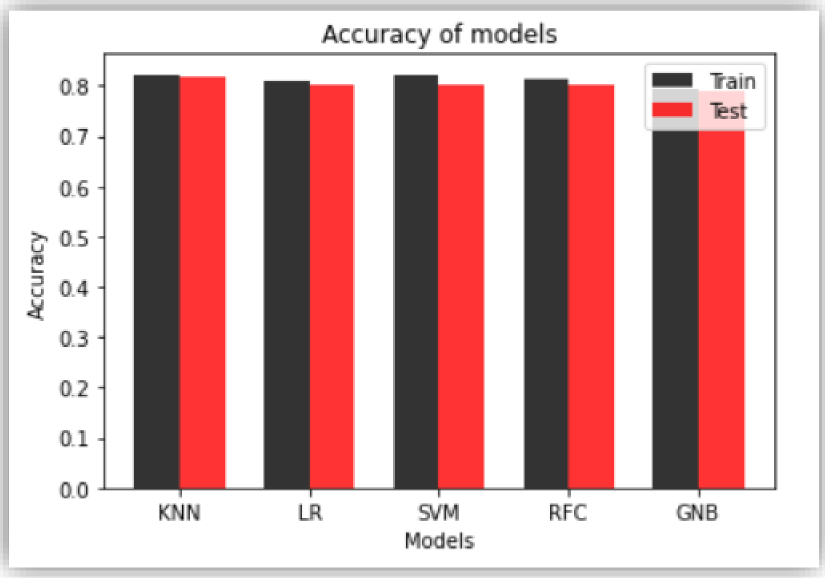
Naive Bayes Algorithm is one of the popular classification machine learning algorithms that helps to classify the data based upon the conditional probability values computation. It implements the Bayes theorem for the computation and used class levels represented as feature values or vectors of predictors for classification. Naive Bayes Algorithm is a fast algorithm for classification problems. This algorithm is a good fit for real-time prediction, multi-class prediction, recommendation system, text classification, and sentiment analysis use cases. Naive Bayes Algorithm can be built using Gaussian, Multinomial and Bernoulli distribution. This algorithm is scalable and easy to implement for the large data set.

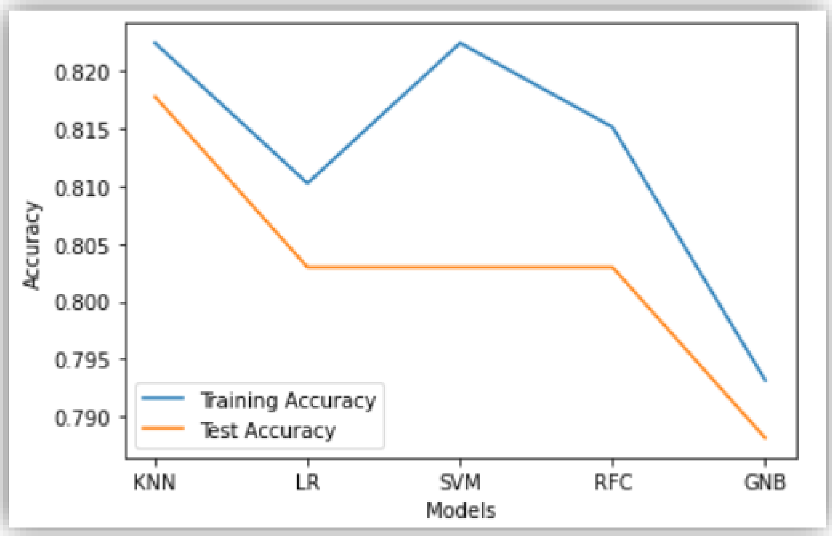
It helps to calculate the posterior probability P(c/x) using the prior probability of class P(c), the prior probability of predictor P(x) and the probability of predictor given class, also called as likelihood P(x/c).

The formula or equation to calculate posterior probability is:

- $P(c/x) = (P(x/c) * P(c)) / P(x)$

Comparison of Algorithms:





Model Name	Training Accuracy	Testing Accuracy
K-Nearest Neighbours	0.8223844282238443	0.8177339901477833
Logistic Regression	0.8102189781021898	0.8029556650246306
Support Vector Machine	0.8223844282238443	0.8029556650246306
Random Forest Classifier	0.8150851581508516	0.8029556650246306
Gaussian Naïve Bayes	0.7931873479318735	0.7881773399014779

Conclusion

In this project, we have practiced different machine learning techniques and different models for data training, attempting to achieve the highest accuracy of predicting Loan approval status. Thus, this study settled on classifying Loan status based on given Customers' details using five different algorithms and consequently testing its accuracy.

Comparing all the classification models, we conclude K-Nearest Neighbor is the preferred choice in terms of its high accuracy and computational efficiency. However, there is no single classifier that works best on all given problems. In case of KNN, by varying the number of neighbours (from 2 to 15) we found that the best training and test accuracies are found at n=7. Preprocessing such as Attribute reduction (12 reduced to 5) and normalization of the attributes are performed to reduce runtime and increase accuracy.

The general accuracies of different algorithms are mentioned in table of previous section. The overall highest Training accuracy – 82.23% and Test accuracy – 81.77% is achieved by KNN.

So, with this project we can reduce the risk factor behind selecting the safe person to whom the loan is to be approved so as to save lots of bank efforts and assets.

Links to follow:

Link to the repository:

https://github.com/manideep-kalyanam/Loan-approval-Prediction---ML

Link to .py and .ipynb files:

https://github.com/manideep-kalyanam/Loan-approval-Prediction---ML/tree/master/Code