
# PESIT - BANGALORE SOUTH CAMPUS

**Hosur Road, Bengaluru - 560100**

Department of Electronics and Communication Engineering

**Mini Project Report**

**on**

# "LOAN ELIGIBILITY PREDICTION USING MACHINE LEARNING"

**JUNE-AUGUST 2021**

**BY**


| | |
|---|---|
| **KUSHAAL R** | **1PE17EC068** |
| **PUSHKAR PRAMOD WANI** | **1PE17EC093** |
| **PRANEETH M KASHYAP** | **1PE17EC089** |
| **SAGAR RAJ N** | **1PE17EC112** |

**UNDER THE GUIDANCE OF**

**Dr. Subash Kulkarni**
**Head Of Department,**
**Dept. of ECE,**
**PESIT-BSC**
**Bengaluru - 560100**


# CERTIFICATE

*THIS IS TO CERTIFY THAT THE MINI PROJECT WORK ENTITLED*

## "Loan Eligibility Prediction using Machine Learning"

*IS A BONAFIED WORK CARRIED OUT BY*


| | |
|---|---|
| **KUSHAAL R** | **1PE17EC068** |
| **PUSHKAR PRAMOD WANI** | **1PE17EC093** |
| **PRANEETH M KASHYAP** | **1PE17EC089** |
| **SAGAR RAJ N** | **1PE17EC112** |


In partial fulfillment for the completion of 8 th semester for Subject **Machine Learning - 17EC834** the program of study BE in Electronics and Communication under rules and regulations of PESIT-BSC, Bengaluru during the period June–August,2021. It is certified that all corrections/suggestions indicated for Assessment have been incorporated in the report. The mini project has been approved as it satisfies the 8 th Semester academic requirements in respect of mini project work.

| | |
|---|---|
| Signature of the Guide with the date | Signature with date and seal |
| **Dr. Subhash Kulkarni** | **Dr. Subhash Kulkarni** |
| **Head Of Department** | **Principal - PESIT-BSC** |

| **Name of the Examiners** | **Signature with Date** |
|---|---|
| 1. | |
| 2. | |

# DECLARATION

We , **Sagar Raj N , Praneeth M Kashyap , Pushkar Pramod Wani and Kushaal R** ,hereby declare that the dissertation entitled, '**Loan Eligibility Prediction using Machine Learning**', is the work done by us under the guidance of **Dr. Subash Kulkarni** , Head Of Department and is being submitted in partial fulfillment of the requirements for completion of 8 th Semester course work Machine Learning - 17EC834 in the Program of Study BE in Electronics and Communication.

**Place : BENGALURU**

**Date :**

**Name and Signature of the Candidates :**

1) SAGAR RAJ N

2) KUSHAAL R

3) PUSHKAR PRAMOD WANI

4) PRANEETH M KASHYAP

# ACKNOWLEDGEMENTS

The success of any task depends largely on the encouragement and guidelines of many others. We take this opportunity to express our gratitude to the people who have been instrumental in the successful completion of this Mini Project.

We would like to express our gratitude to **Dr. Subhash Kulkarni** Principal of PESIT-BSC for not only providing with the excellent facilities but also for offering his unending encouragement that has made this Mini Project a success.

We would thank again our guide , **Dr. Subhash Kulkarni**,Head Of Department, Department of Electronics and Communication who have being supporting through entire duration of the project.

And we also express our sincere thanks to all the Teaching and Non -Teaching Faculty for your support during the semester.

# Abstract

The accessibility and availability of huge amounts of data will be able to provide us useful knowledge if certain data mining techniques are applied on it. The main goal is to determine new patterns and then to interpret these patterns to deliver significant and useful information for the users. Since the number of parameters involved in predicting loan approval are vast and they have complex interdependence, machine learning is an appropriate tool to tackle the challenge of loan approval prediction. We implement various classifiers such as Logistic Regression, k-nearest neighbors, Random Forest, Gaussian Naive Bayes, Neural Networks etc.., and measure their performance in prediction. We also attempt to combine the classifiers using a Meta Classifier to produce greater predictive capability.

# Contents

# List of Figures

# Chapter 1

# Introduction

Banks have a variety of things to sell in our banking system, but their credit lines are their primary source of income. As a result, they will be able to profit from the interest on the loans they credit. The profitability or loss of a bank is mostly determined by loans, i.e. whether consumers repay the loan or default on it. The bank can lower its non-performing assets (NPAs) by anticipating loan defaulters. As a result, it's critical to investigate this occurrence. Previous research in this era has revealed that there are a plethora of approaches to studying the issue of loan default control.

However, because accurate forecasts are critical for profit maximisation, it is critical to investigate the nature of the various methodologies and compare them. In terms of loan forecasting, experimental studies revealed that the Naive Bayes model has the most consistent results without compromising on accuracy.

# Chapter 2

# System Design

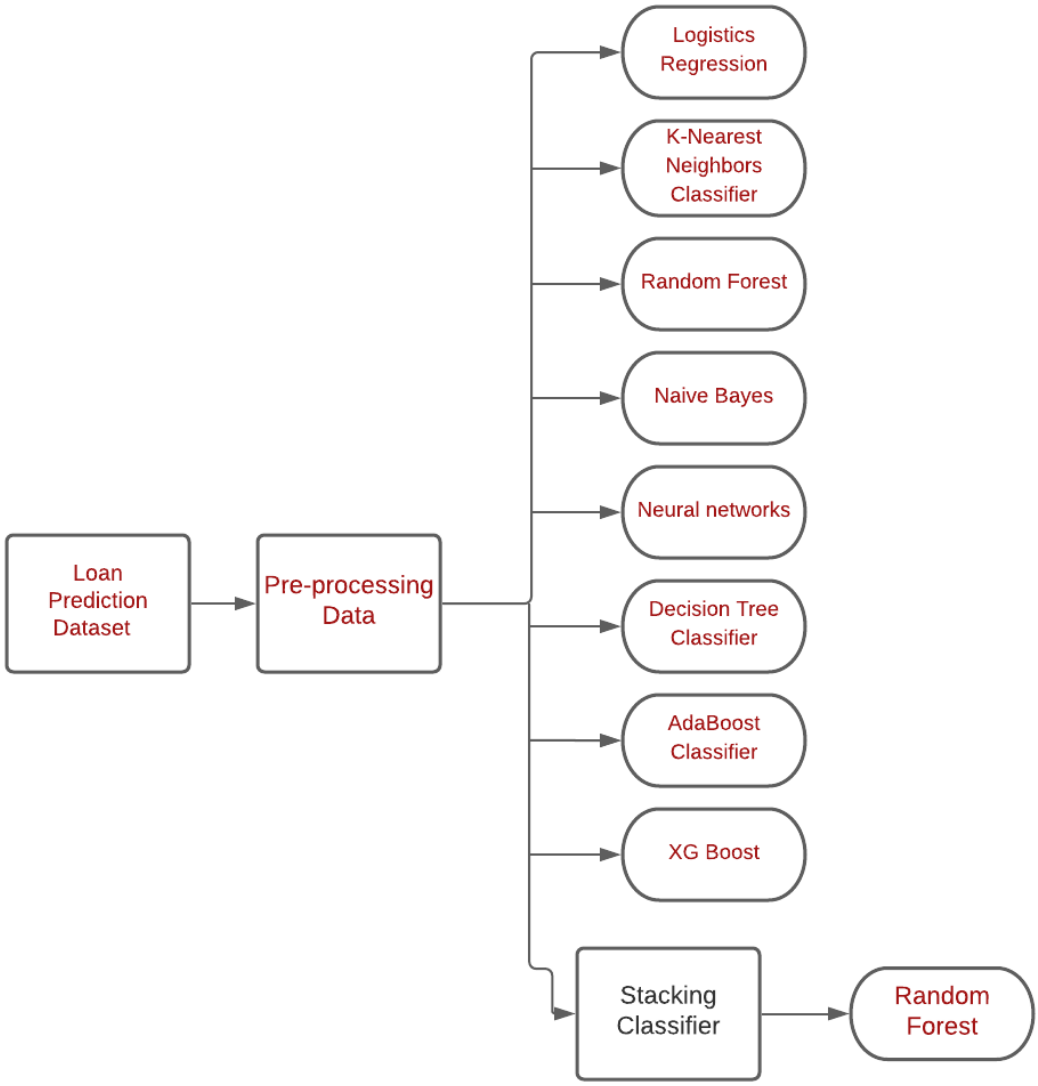


**Figure 2.1:** Proposed System Block diagram

# Brief Description of Classifiers used in our System

## 2.1 Types Of Classifiers

### 2.1.1 Logistics Regression

A linear model does not output probabilities, but it treats the classes as numbers (0 and 1) and fits the best hyperplane (for a single feature, it is a line) that minimizes the distances between the points and the hyperplane. So it simply interpolates between the points, and you cannot interpret it as probabilities.

A linear model also extrapolates and gives you values below zero and above one. This is a good sign that there might be a smarter approach to classification.

Since the predicted outcome is not a probability, but a linear interpolation between points, there is no meaningful threshold at which you can distinguish one class from the other.

### 2.1.2 K-Nearest Neighbors

K-Nearest Neighbors 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.It is widely disposable in real-life scenarios since it is non-parametric, meaning, it does not make any underlying assumptions about the distribution of data (as opposed to other algorithms such as GMM, which assume a Gaussian distribution of the given data).

In this classifier a cluster of labeled points are used to understand how the other points should be labelled. For labelling a new point it checks the already labelled points which could be closest to the point to be labelled, i.e closest to the neighbour. In this way depending on the votes of the neighbour the new point is labelled the same label which most of neighbours have. In in algorithm 'k' is the number of neighbours which are checked.

### 2.1.3 Decision Trees

This classification algorithm builds the regression models. These models are built in form of structure which is similar to tree - a tree like structure is created by this classifier. It keeps on dividing the data set into subsets and smaller subsets which develops an associated tree, incrementally. The decision tree is finally created which has decision nodes and leaf nodes. In this tree the leaf node will have details about the classification or the decision taken for classification whereas the decision will have branches. The highest decision node which will be at the top of the tree will correspond to the root node.

### 2.1.4 Random Forest Classifier

A random forest classifier. A random forest is a meta estimator that fits a number of decision tree classifiers on various sub-samples of the dataset and uses averaging to improve the predictive accuracy and control over-fitting.

### 2.1.5 Naive Bayes Classifier

This classifier can also be known as a Generative Learning Model. The classification here is based on Baye's Theorem, it assumes independent predictors. In simple words, this classifier will assume that the

existence of specific features in a class is not related to the existence of any other feature. If there is dependency among the features of each other or on the presence of other features, all of these will be considered as an independent contribution to the probability of the output. This classification algorithm is very much useful to large datasets and is very easy to use.

### 2.1.6 Neural Networks

As the name suggests this classifier has units known as neurons, which are arranged in layers that convert the input vector to relevant output. Each single neuron takes an input, this is most often a non-linear input, this is given to a function which is them passed to next layer to get the output. The input given to the first layer will act as an output for the next layer and so on, thus this classification algorithm follows a feed-forward method. But in this method there is no feedback to the previous layer, so weighting are also given to the signals passing through the neurons and the layers, these signal then are turned into a training phase this eventually then become a network to handle any particular problem.

### 2.1.7 XGBoost

Recently, the researches have come across an algorithm “XGBoost” and its usage is very useful for machine learning classification. It is very much fast and its performance is better as it is an execution of a boosted decision tree. This classification model is used to improve the performance of the model and also to improve the speed.

### 2.1.8 AdaBoost

Ada-boost or Adaptive Boosting is one of ensemble boosting classifier proposed by Yoav Freund and Robert Schapire in 1996. It combines multiple classifiers to increase the accuracy of classifiers. AdaBoost is an iterative ensemble method. AdaBoost classifier builds a strong classifier by combining multiple poorly performing classifiers so that you will get high accuracy strong classifier. The basic concept behind Adaboost is to set the weights of classifiers and training the data sample in each iteration such that it ensures the accurate predictions of unusual observations. Any machine learning algorithm can be used as base classifier if it accepts weights on the training set.

Adaboost should meet two conditions:
1)The classifier should be trained interactively on various weighed training examples.
2)In each iteration, it tries to provide an excellent fit for these examples by minimizing training error.

Light GBM use histogram based algorithm i.e it buckets continuous feature values into discrete bins which speeds up the training procedure and reduces the memory usage.It is capable of performing on par with XGBOOST with large datasets with a significant reduction in training time.

### 2.1.9 Stacking Classifier

Stacking Classifier is an ensemble learning technique that combines multiple base classification models predictions into a new data set, and treated as the input data for another classifier. This classifier employed to solve this problem. Stacking is often referred to as blending.In our case we have used Random Forest as a Meta-Classifier.

## 2.2 Numerical attributes of Loan Eligibility Prediction Dataset

| | ApplicantIncome | CoapplicantIncome | LoanAmount | Loan_Amount_Term | Credit_History |
|---|---|---|---|---|---|
| **count** | 614.000000 | 614.000000 | 592.000000 | 600.00000 | 564.000000 |
| **mean** | 5403.459283 | 1621.245798 | 146.412162 | 342.00000 | 0.842199 |
| **std** | 6109.041673 | 2926.248369 | 85.587325 | 65.12041 | 0.364878 |
| **min** | 150.000000 | 0.000000 | 9.000000 | 12.00000 | 0.000000 |
| **25%** | 2877.500000 | 0.000000 | 100.000000 | 360.00000 | 1.000000 |
| **50%** | 3812.500000 | 1188.500000 | 128.000000 | 360.00000 | 1.000000 |
| **75%** | 5795.000000 | 2297.250000 | 168.000000 | 360.00000 | 1.000000 |
| **max** | 81000.000000 | 41667.000000 | 700.000000 | 480.00000 | 1.000000 |

**Figure 2.2:** Numerical attributes of Loan Eligibility Prediction Dataset

### 2.2.1 Categorical and Other Attributes

1. Gender - (”Male” , ”Female”)
2. Marital Status - (”Yes” , ”No”)
3. Dependents - (0,1,2,3,3+)
4. Self Employed - (”Yes” , ”No”)
5. Education - (”Graduate” , ”Not-Graduate”)
6. Property Area - (”Urban” , ”RuraL”)

# Chapter 3

# Result Analysis

## 3.1 Accuracy of Models

```
Accuracy =  75.2688 % --> Model :  Decision Tree Classifier
Accuracy =  82.7957 % --> Model :  Naive Bayes
Accuracy =  78.4946 % --> Model :  Neural network
Accuracy =  78.4946 % --> Model :  Random Forest
Accuracy =  80.6452 % --> Model :  KNN
Accuracy =  82.7957 % --> Model :  Logistics Regression
Accuracy =  80.6452 % --> Model :  XG Boost Classifier
Accuracy =  81.7204 % --> Model :  AdaBoost classifier
Accuracy =  80.6452 % --> Model :  Stacking Classifier
```

**Figure 3.1:** Final Result

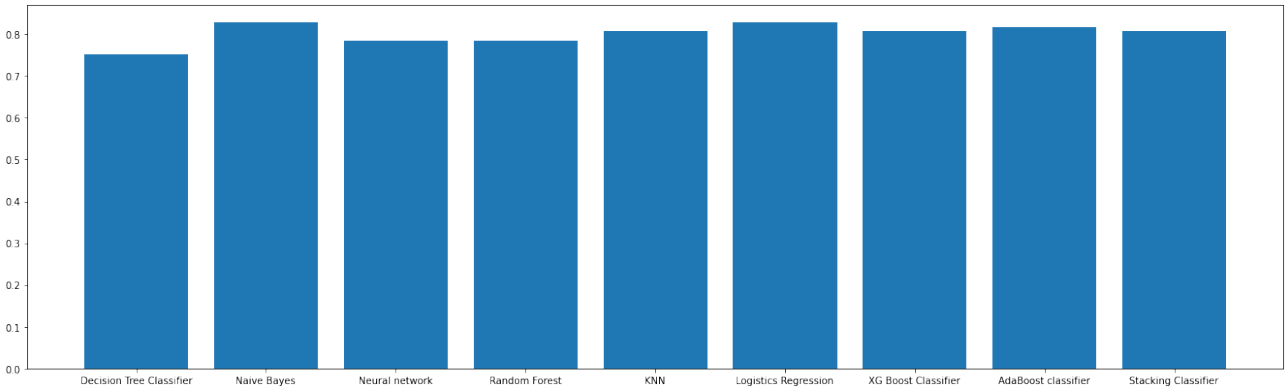


**Figure 3.2:** Bar Graph

## 3.2 Relative Predictions Comparison

```
(  Naive Bayes  ,  Random Forest  ) =  32
(  Naive Bayes  ,  Stacking Classifier  ) =  35
(  Naive Bayes  ,  KNN  ) =  11
(  Naive Bayes  ,  Decision Tree  ) =  47
(  Naive Bayes  ,  Neural Network  ) =  79
(  Naive Bayes  ,  Logistic Regression  ) =  2
(  Naive Bayes  ,  XG Boost  ) =  10
(  Naive Bayes  ,  Ada Boost  ) =  18
(  Random Forest  ,  Stacking Classifier  ) =  7
(  Random Forest  ,  KNN  ) =  27
(  Random Forest  ,  Decision Tree  ) =  55
(  Random Forest  ,  Neural Network  ) =  61
(  Random Forest  ,  Logistic Regression  ) =  32
(  Random Forest  ,  XG Boost  ) =  30
(  Random Forest  ,  Ada Boost  ) =  30
(  Stacking Classifier  ,  KNN  ) =  32
(  Stacking Classifier  ,  Decision Tree  ) =  56
(  Stacking Classifier  ,  Neural Network  ) =  60
(  Stacking Classifier  ,  Logistic Regression  ) =  35
(  Stacking Classifier  ,  XG Boost  ) =  31
(  Stacking Classifier  ,  Ada Boost  ) =  33
(  KNN  ,  Decision Tree  ) =  48
(  KNN  ,  Neural Network  ) =  70
(  KNN  ,  Logistic Regression  ) =  11
(  KNN  ,  XG Boost  ) =  13
(  KNN  ,  Ada Boost  ) =  19
(  Decision Tree  ,  Neural Network  ) =  84
(  Decision Tree  ,  Logistic Regression  ) =  47
(  Decision Tree  ,  XG Boost  ) =  43
(  Decision Tree  ,  Ada Boost  ) =  49
(  Neural Network  ,  Logistic Regression  ) =  79
(  Neural Network  ,  XG Boost  ) =  79
(  Neural Network  ,  Ada Boost  ) =  77
(  Logistic Regression  ,  XG Boost  ) =  10
(  Logistic Regression  ,  Ada Boost  ) =  18
(  XG Boost  ,  Ada Boost  ) =  12
```

**Figure 3.3:** Relative Number of Mismatches in Predictions

**Further, We used the above algorithms to predict loan eligibility on an unlabeled dataset of 367 samples. The relative predictions of the algorithms were then compared.**