Comparison between Decision Tree and Random Forest towards recommendation system or engine

Submitted as a partial fulfillment of Bachelor of Technology in Computer Science & Engineering
of
Maulana Abul Kalam Azad University of Technology
(Formerly known as West Bengal University of Technology)

Project Report

Submitted by

Name of Students	University Roll No.
Pritam Roy	11600118037
Srijon Mallick	11600118017
Souvik Saha	11600118020
Rupak Pal	11600118032
Pritam Das	11600118038

Under the supervision of

Dr. S. S. Thakur
Associate Professor, Department of Computer Science and Engineering

Department of Computer Science & Engineering,
MCKV Institute of Engineering
243, G.T. Road(N)
Liluah, Howrah - 711204

Department of Computer Science & Engineering
MCKV Institute of Engineering
243, G. T. Road (N), Liluah, Howrah-711204

CERTIFICATE OF RECOMMENDATION

I hereby recommend that the thesis prepared under my supervision by Pritam Roy, Rupak Pal, Pritam Das, Srijon Mallick, Souvik Saha entitled Comparison between decision tree and random forest towards recommendation system or engine be accepted in partial fulfillment of the requirements for the degree of Bachelor of Technology in Computer Science & Engineering Department.

Mr. Avijit Bose
Assistant Professor & Head of the Department,
Computer Science & Engineering Department
MCKV Institute of Engineering, Howrah

Project guide
Dr. S. S. Thakur,
Associate Professor,
Computer Science &Engineering Department

MCKV Institute of Engineering
243, G. T. Road (N), Liluah Howrah-711204

Affiliated to

Maulana Abul Kalam Azad University of Technology
(Formerly known as West Bengal University of Technology)

CERTIFICATE

This is to certify that the project entitled Comparison between decision tree and random forest towards recommendation system or engine and submitted by

Name of students	University Roll No.
Pritam Roy	11600118037
Srijon Mallick	11600118017
Souvik Saha	11600118020
Rupak Pal	11600118032
Pritam Das	11600118038

has been carried out under the guidance of myself following the rules and regulations of the degree of Bachelor of Technology in Computer Science & Engineering of **Maulana Abul Kalam Azad University of Technology (**Formerly West Bengal University of Technology).

____

(Signature of the project guide)

Dr. S. S. Thakur,
Associate Professor,
Computer Science & Engineering Department

1. Pritam Roy
2. Srijon Mallick
3. Souvik Saha
4. Rupak Pal
5. Pritam Das,

MCKV Institute of Engineering
243, G. T. Road (N), Liluah Howrah-711204

Affiliated to
Maulana Abul Kalam Azad University of Technology
(Formerly known as West Bengal University of Technology)

CERTIFICATE OF APPROVAL

(B. Tech Degree in Computer Science & Engineering)

This project report is hereby approved as a creditable study of an engineering subject carried out and presented in a manner satisfactory to warrant its acceptance as a prerequisite to the degree for which it has been submitted. It is to be understood that by this approval, the undersigned do not necessarily endorse or approve any statement made; opinion expressed and conclusion drawn therein but approve the project report only for the purpose for which it has been submitted

COMMITTEE ON FINAL
EXAMINATION FOR
EVALUATION OF
PROJECT REPORT

1.
2.
3.
4.
5.

ACKNOWLEDGEMENT

We express our sincere gratitude to Dr. S.S. Thakur, Associate Professor, Department of Computer Science and Engineering, our project guide and Mr. Avijit Bose, Assistant Professor and Head of Department (CSE) for providing us their guidance and cooperation for the project. We also extend our sincere thanks to all other faculty members of Computer Science & Engineering Department and our friends for their support and encouragement. We will be failing in duty if we do not acknowledge with grateful thanks to the authors of the references and other literatures referred to in this project. Last but never the least we are very much thankful to our parents who guided and supported us in every step which we took.

CONTENTS

1. Abstract

Without a doubt, financial lending services hold a great amount of significance for any individual, business, or enterprise. As such services are required by an individual or a business to achieve or accomplish their goals and to compete with the giants of their fields. Financial loans are a major part of the primary source of capital not only in the emerging economies but also in the developed capital markets by both individuals and enterprises. The lending growth by the financial firms and the banks is considered the key factor for the inflation level and interest rate of any country which drives its economic growth and depicts its economic condition. The economic growth of the real economy is the primary roleof the financial firms. With such great importance and benefits of financial lending come some major issues and bottleneck problems. The most common and substantial issue in the domain of financial lending is the fair and successful lending of loans while keeping the ratio of loan defaulters to the least minimum value. In financial lending, the risk of loan defaulters can never be neutralized butcan be minimized. The purpose of this study is to provide a comparison between the Decision Tree and Random Forest algorithm toward a recommendation engine to predict the loan defaults. This kind of model becomes inevitable as the issue of bad loans is very much critical in the financial sector, especially in microfinancing banks of various underdeveloped and developed countries.

2. Introduction

Machine Learning (ML) pertains to the ability of data-driven models to “learn” information about a system directly from observed data without predetermining mechanistic relationships that govern the system. ML algorithms are able to adaptively improve their performance with each new data sample and discover hidden patterns in complex heterogeneous and high dimensional data. In different engineering domains, ML offers predictive models, such as Decision Trees (DTs), Random Forests (RFs), Support Vector Machines (SVMs), etc. Which are able to map highly non-linear heterogeneous input and output patterns even when physiological relationships between model variables could not be determined due to complexity, pathologies, or lack of biological understanding. They cope with missing values and are able to combine heterogeneous data types into a single model, whilst also performing an automatic principal feature selection. Combining multiple Decision Trees (DTs) in a Random Forests (RFs) maintains this interpretability, but offers state-of-the-art prediction accuracies. Machine learning is a branch of artificial intelligence that enables a machine to automatically learn from data, improve performance from experiences, and predict things without being explicitly programmed.

Tomas Pranckevičius, and Virginijus Marcinkevicius [1] investigated Naïve Bayes, Random Forest, Decision Tree, Support Vector Machines, and Logistic Regression classifiers implemented in Apache Spark, i.e., the in-memory intensive computing platform. They focused on comparing these classifiers by evaluating the classification accuracy, based on the size of training data sets, and the number of n-grams. In experiments, short texts for product-review data from Amazon were analyzed. A comparative analysis of decision tree algorithms and the random forest was done by Shiju Sathyadevanand Remya R. Nair [2]. Their paper focuses on the comparison of decision tree algorithms-ID3, and C4.5 which was then compared with random forest. Random forest is found to be superior to the other two based on accuracy. Apurva Datkhile et al. [3] defined a prototype using four different models- Naive Bayes, Random Forest, Logistic Regression, and Decision Tree Algorithm, which can be used by organizations to make a correct or correct decision to approve or reject a consumer's request for a loan. Sebastian Buschjägerand Katharina Morik [4] investigated implementations of decision trees and random forests for the classical von-Neumann computing architecture and custom circuits by the means of field-programmable gate arrays.

As in the past couple of decades, the decision-making for financial lending [5] has been very much influenced by information sharing and technological advancements. The technique of credit scoring is to evaluate different credit attributes by analyzing and classification to an individual and enterprise profile to assess the credit decision or to estimate the creditworthiness. Only credit scoring is not sufficient for financial lending because of such a massive number of loan defaulters. As financial analysts not only rely on the credit scores but also on their experience regarding the historical successful and unsuccessful cases as well for better decision making. Moreover, with such tremendous growth of financial lending and to improve the credit defaulter ratio, advanced statistical methods were introduced to fill the gap of underperforming credit scoring models. These advanced statistical models provided the alternative to the previous traditional statistical models which were based on the logistic regression and discriminate analysis In the more recent years, different researchers have also employed different data mining techniques for the loan defaulter predictions. A bank's profit or a loss depends to a large extent on loans i.e. whether the customers are paying back the loan or defaulting. By predicting the loan defaulters, the bank can reduce its Non-performing Assets. This makes the study of this phenomenon very important. Previous research in this era has shown that there are so many methods to study the problem of controlling loan default. But as the right predictions are very important for the maximization of profits, it is essential to study the nature of the different methods and their comparison. A very important approach in predictive analytics is used to study the problem of predicting loan defaulters (i) Collection of Data, (ii) Data Cleaning, and (iii) Performance Evaluation. Kadam et. al [6] studied the different aspects of loan forecasting and suggested using the Naive Bayes model for Loan approval. Lifeng Zhou and Hong Wang [7] proposed an Improved random forest that has greater accuracy. Tariq et al. [8] have provided comprehended research and developed a model to predict loan defaults. Kumar et al. [9] compared different machine learning algorithms to predict whether assigning a loan to a particular person will be safe or not.

We have proposed a study regarding the comparison of the Decision tree classifier and Random Forest algorithm towards the recommendation engine against the financial lending request. The purpose of this study is to provide comprehensive research and to develop a model to predict loan defaults.

3. Machine Learning

3.1. What is Machine Learning?

In the real-life scenario, a human being can learn or can gain knowledge from his/her experiences with his/her learning ability. So, we can easily train a human being. We have computers or machines which work on our instructions. But like a human, can we train a machine? The answer is yes, we can train a machine. A machine can also learn from its past experience. So here comes the role of Machine Learning.

Machine learning is a branch of artificial intelligence, that enables a machine to automatically learn from data, improve performance from experiences, and predict things without being explicitly programmed.

Two types of learning technique are there in machine learning:

1. Supervised Learning

2. Unsupervised Learning

3.2. Supervised Learning

Before exploring Supervised Learning, we need to be familiar with the two terminologies:

1. Train Dataset: Datasets having some labelled training data.The labelled data means some input data is already tagged with the correct output. This dataset is used to train a machine.

2. Test Dataset: Datasets having some unlabelled test data on which the machine predicts the output.

In supervised learning, the training data provided to the machines work as the supervisor that teaches the machines to predict the output correctly on test datasets. It applies the same concept as a student learns in the supervision of the teacher.

In the real-world, supervised learning can be used for Risk Assessment, Image classification, Fraud Detection, spam filtering, etc.

Categories of Supervised Machine Learning

Supervised machine learning can be classified into two types of problems, which are given below:

- Classification
- Regression

a) Classification

Classification algorithms are used to solve the classification problems in which the output variable is categorical, such as "Yes" or No, Male or Female, Red or Blue, etc. The classification algorithms predict the categories present in the dataset. Some real-world examples of classification algorithms are Spam Detection, Email filtering, etc.

Some popular classification algorithms are given below:

- Random Forest Algorithm
- Decision Tree Algorithm
- Logistic Regression Algorithm
- Support Vector Machine Algorithm

b) Regression

Regression algorithms are used to solve regression problems in which there is a linear relationship between input and output variables. These are used to predict continuous output variables, such as market trends, weather prediction, etc.

Some popular Regression algorithms are given below:

- Simple Linear Regression Algorithm
- Multivariate Regression Algorithm
- Decision Tree Algorithm
- Lasso Regression

3.3. Unsupervised Learning

In the previous topic, we learned supervised machine learning in which models are trained using labelled data under the supervision of training data. But there may be many cases in which we do not have labelled data and need to find the hidden patterns from the given dataset. So, to solve such types of cases in machine learning, we need unsupervised learning techniques.

Unsupervised learning is a type of machine learning in which models are trained using unlabelled dataset and are allowed to act on that data without any supervision.

4. Decision Tree

Decision Tree is a Supervised learning technique that can be used for both classification and Regression problems, but mostly it is preferred for solving Classification problems. It is a tree-structured classifier, where internal nodes represent the features of a dataset, branches represent the decision rules and each leaf node represents the outcome. Decision tree consists of two nodes, which are the Decision Node and Leaf Node. Decision nodes are used to make any decision and have multiple branches, whereas Leaf nodes are the output of those decisions and do not contain any further branches. The decisions or the test are performed on the basis of features of the given dataset.

It is a graphical representation for getting all the possible solutions to a problem/decision based on given conditions. A decision tree simply asks a question, and based on the answer (Yes/No), it further split the tree into subtrees. Below diagram explains the general structure of a decision tree:

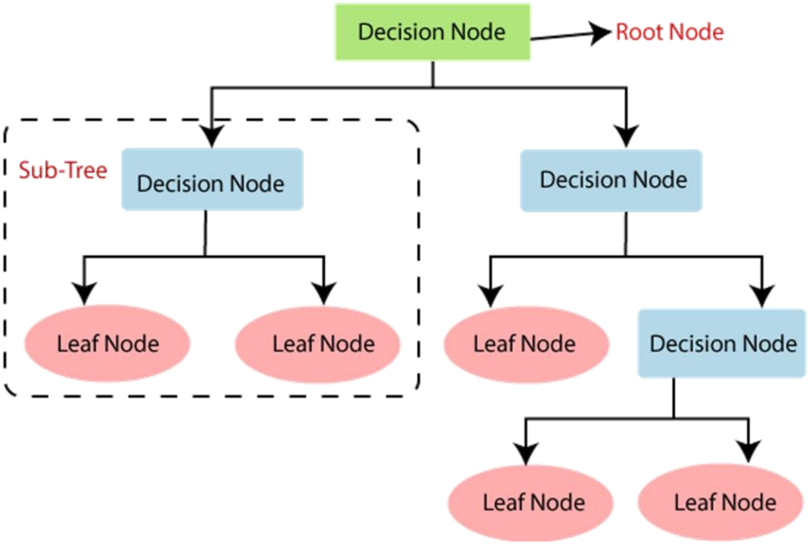


Figure 1 Decision Tree

4.1. Decision Tree Terminologies

- ❖ **Root Node:** Root node is from where the decision tree starts. It represents the entire dataset, which further gets divided into two or more homogeneous sets.

- **Leaf Node**: Leaf nodes are the final output node, and the tree cannot be segregated further after getting a leaf node.
- **Splitting:** Splitting is the process of dividing the decision node/root node into sub-nodes according to the given conditions.
- **Branch/Sub Tree:** A tree formed by splitting the tree.
- **Pruning:** Pruning is the process of removing the unwanted branches from the tree.
- **Parent/Child node:** The root node of the tree is called the parent node, and other nodes are called the child nodes

4.2. Why use Decision tree?

There are various algorithms in Machine learning, so choosing the best algorithm for the given dataset and problem is the main point to remember while creating a machine learning model. Below are the two reasons for using the Decision tree:

- Decision Trees usually mimic human thinking ability while making a decision, so it is easy to understand.
- The logic behind the decision tree can be easily understood because it shows a tree-like structure.

4.3. How does the Decision Tree algorithm Work?

In a decision tree, for predicting the class of the given dataset, the algorithm starts from the root node of the tree. This algorithm compares the values of root attribute with the record (real dataset) attribute and, based on the comparison, follows the branch and jumps to the next node.

For the next node, the algorithm again compares the attribute value with the other sub-nodes and move further. It continues the process until it reaches the leaf node of the tree. The complete process can be better understood using the below algorithm:

- **Step-1:** Begin the tree with the root node, says S, which contains the complete dataset.
- **Step-2:** Find the best attribute in the dataset using Attribute Selection Measure (ASM).
- **Step-3:** Divide the S into subsets that contains possible values for the best attributes.
- **Step-4:** Generate the decision tree node, which contains the best attribute.

- **Step-5:** Recursively make new decision trees using the subsets of the dataset created in step -3. Continue this process until a stage is reached where you cannot further classify the nodes and called the final node as a leaf node.

Example: Suppose there is a candidate who has a job offer and wants to decide whether he should accept the offer or not. So, to solve this problem, the decision tree starts with the root node (Salary attribute by ASM). The root node splits further into the next decision node (distance from the office) and one leaf node based on the corresponding labels. The next decision node further gets split into one decision node (Cab facility) and one leaf node. Finally, the decision node splits into two leaf nodes (Accepted offers and Declined offer). Consider the below diagram:

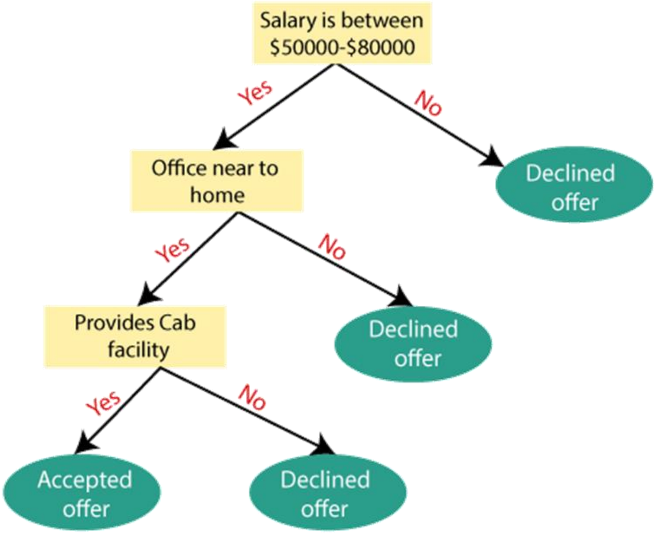


Figure 2 Working Of Decision Tree

4.4. Attribute Selection Measures

While implementing a Decision tree, the main issue arises that how to select the best attribute for the root node and for sub-nodes. So, to solve such problems there is a technique which is called as Attribute selection measure or ASM. By this measurement, we can easily select the best attribute for the nodes of the tree. There are two popular techniques for ASM, which are:

- Information Gain
- Gini Index

1. Information Gain:

- Information gain is the measurement of changes in entropy after the segmentation of a dataset based on an attribute.
- It calculates how much information a feature provides us about a class.
- According to the value of information gain, we split the node and build the decision tree.
- A decision tree algorithm always tries to maximize the value of information gain, and a node/attribute having the highest information gain is split first. It can be calculated using the below formula:

  Information Gain= Entropy(S)- [(Weighted Avg) *Entropy (each feature)

Entropy: Entropy is a metric to measure the impurity in a given attribute. It specifies randomness in data. Entropy can be calculated as:

Entropy(s)= -P(yes)log2 P(yes)- P(no) log2 P(no)

Where,

- S= Total number of samples
- P(yes)= probability of yes
- P(no)= probability of no

2. Gini Index:

- Gini index is a measure of impurity or purity used while creating a decision tree in the CART (Classification and Regression Tree) algorithm.
- An attribute with the low Gini index should be preferred as compared to the high Gini index.
- It only creates binary splits, and the CART algorithm uses the Gini index to create binary splits.
- Gini index can be calculated using the below formula:

  Gini Index= 1- $\sum_j P_j^2$

4.5. Advantages of the Decision Tree

- It is simple to understand as it follows the same process which a human follow while making any decision in real-life.
- It can be very useful for solving decision-related problems.
- It helps to think about all the possible outcomes for a problem.
- There is less requirement of data cleaning compared to other algorithms.

4.6. Disadvantages of the Decision Tree

- The decision tree contains lots of layers, which makes it complex.
- It may have an overfitting issue, which can be resolved using the Random Forest algorithm.
- For more class labels, the computational complexity of the decision tree may increase.

5. Random Forest

Random Forest is a popular machine learning algorithm that belongs to the supervised learning technique. It can be used for both Classification and Regression problems in ML. It is based on the concept of ensemble learning, which is a process of combining multiple classifiers to solve a complex problem and to improve the performance of the model.

As the name suggests, "Random Forest is a classifier that contains a number of decision trees on various subsets of the given dataset and takes the average to improve the predictive accuracy of that dataset." Instead of relying on one decision tree, the random forest takes the prediction from each tree and based on the majority votes of predictions, and it predicts the final output.

The greater number of trees in the forest leads to higher accuracy and prevents the problem of overfitting. The below diagram explains the working of the Random Forest algorithm:

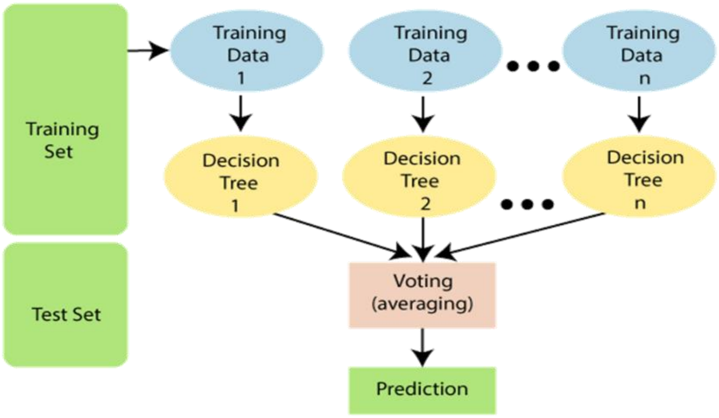


Figure 3 Random Forest

5.1. Assumptions for Random Forest

Since the random forest combines multiple trees to predict the class of the dataset, it is possible that some decision trees may predict the correct output, while others may not. But together, all the trees predict the correct output. Therefore, below are two assumptions for a better Random Forest classifier:

- There should be some actual values in the feature variable of the dataset so that the classifier can predict accurate results rather than a guessed result.
- The predictions from each tree must have very low correlations.

5.2. Why use Random Forest?

Below are some points that explain why we should use the Random Forest algorithm:

- It takes less training time as compared to other algorithms.
- It predicts output with high accuracy, even for the large dataset it runs efficiently.
- It can also maintain accuracy when a large proportion of data is missing.

5.3. How does Random Forest algorithm work?

Random Forest works in two-phase first is to create the random forest by combining N decision tree, and second is to make predictions for each tree created in the first phase.

The Working process can be explained in the below steps and diagram:

Step-1: Select random K data points from the training set.

Step-2: Build the decision trees associated with the selected data points (Subsets).

Step-3: Choose the number N for decision trees that you want to build.

Step-4: Repeat Step 1 & 2.

Step-5: For new data points, find the predictions of each decision tree, and assign the new data points to the category that wins the majority votes. The working of the algorithm can be better understood by the below example:

Example: Suppose there is a dataset that contains multiple fruit images. So, this dataset is given to the Random Forest classifier. The dataset is divided into subsets and given to each decision tree. During the training phase, each decision tree produces a prediction result, and when a new data point occurs, then based on the majority of results, the Random Forest classifier predicts the final decision. Consider the below image:

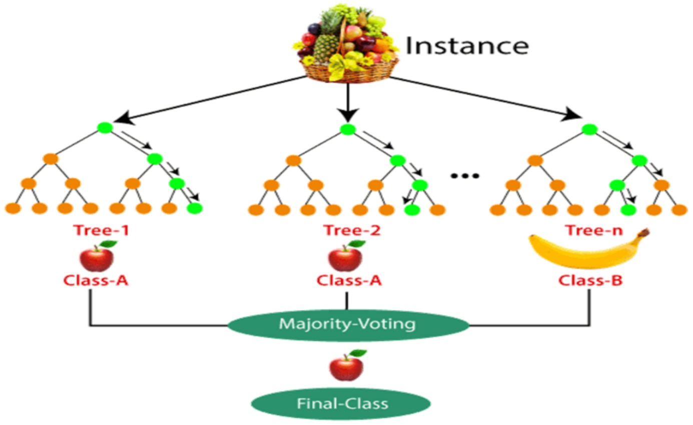


Figure 4 Working Of Random Forest

5.4. Advantages of Random Forest

- Random Forest is capable of performing both Classification and Regression tasks.
- It is capable of handling large datasets with high dimensionality.
- It enhances the accuracy of the model and prevents the overfitting issue.

5.5. Disadvantages of Random Forest

- Although random forest can be used for both classification and regression tasks, it is not more suitable for Regression tasks.

6. Methodology

Comparing machine learning approaches and methodologies to achieve the best prediction will be the main objective. We have selected Decision trees and Random forests as two of the fundamental Machine learning approaches ever known. Using them we will be generating loan default predictions to suggest the financial service providers' chance to make a more accurate decision. This section includes details of the sample data, dataset structure, and their individual means. Apart from that different data cleaning and pre-processing steps and measures will be reasoned. Shown below is the workflow of our methodology.

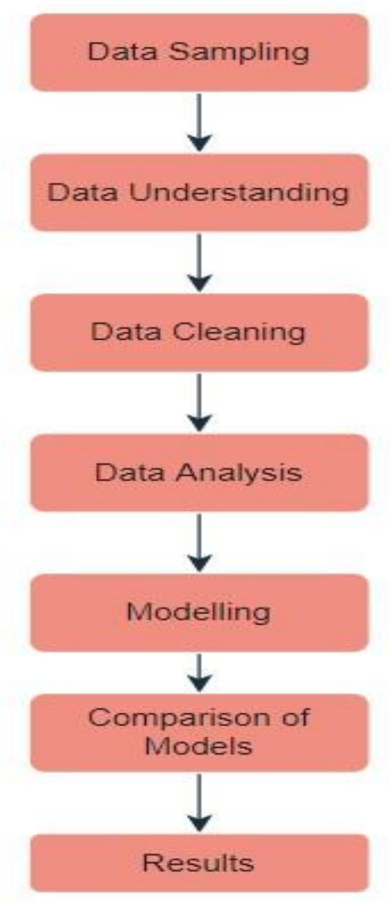


Figure 5 Workflow Diagram

6.1. Data Sampling

The process of collecting data from the entire population is a significantly difficult task and when the data is related to loans it is even harder to gather sufficient data to conduct any research. Manually recording the same would take years to compile and the individuals taking loans do not always share anything. Government organizations such as banks or insurance companies will never disclose any details. Even private organizations also do not share any with very few exceptions. While in this study, the data sample was obtained from the Lending Club organization 2015 database release [10] which is the only available valid data from an actual company.

6.2. Data Understanding

This phase can be considered as one of the most vital steps as in this phase, the understanding related to the structure of the data has to be developed which is very much substantial for model development. All this exploration of the data will improve the discovery process of the meaningful information and also helps in the identification of the anomalies in the data as well.

6.2.1. Dataset structure

Dataset initially had around 111 attributes with each attribute having 42542 data. For the development of the proposed system, the provided data is divided into two datasets each for the training purpose of the model and for the performance testing of the model. Both of the datasets have 12 attributes related to the feature of individuals although the testing dataset has no values in its class variable as they have to be predicted by the developed model. Furthermore, the attributes are comprised of five categorical, seven continuous including a class variable. Following table shows the description of each attribute of the dataset.

Attribute	Data Type	Description
LOAN_AMNT	Continuous	The listed amount of the loan applied for by the borrower. If at some point in time, the credit department reduces the loan amount, then it will be reflected in this value.

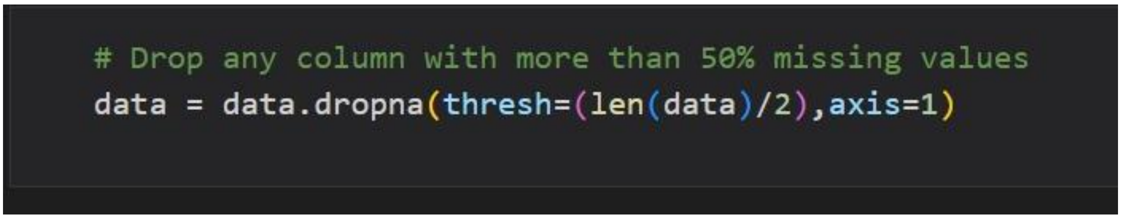
TERM	Categorical	The number of payments on the loan. Values are in months and can be either 36 or 60.
INT_RATE	Continuous	Interest Rate on the loan
GRADE	Categorical	LC assigned loan grade
EMP_LENGTH	Continuous	Employment length in years. Possible values are between 0 and 10 where 0 means less than one year and 10 means ten or more years.
HOME_OWNERSHIP	Categorical	The homeownership status is provided by the borrower during registration or obtained from the credit report. Our values are: RENT, OWN, MORTGAGE, OTHER
ANNUAL_INC	Continuous	The self-reported annual income provided by the borrower during registration.
VERIFICATION_STATUS	Categorical	Indicates if income was verified by LC, not verified, or if the income source was verified
DTI	Continuous	A ratio calculated using the borrower's total monthly debt payments on the total debt obligations, excluding mortgage and the requested LC loan, divided by the borrower's self-reported monthly income.
OPEN_ACC	Continuous	The number of open credit lines in the borrower's credit file.

TOTAL_ACC	Continuous	The total number of credit lines currently in the borrower's credit file
PURPOSE	Categorical	A category provided by the borrower for the loan request.
LOAN_STATUS	Categorical	Current status of the loan- Fully paid or default [Class Variable]

6.3. Data Cleaning

This step involves major tasks related to the modification of the data in the proposed study is the conversion of the data types, getting rid of unnecessary and unimportant data, and handling the missing values. As all the categorical attributes with character data types have to be converted into numeric for the purpose of the algorithm application. Moreover, all the missing values in the entire data have to be handled for improved and efficient modelling.

The first step in the modification process is related to the handling of the missing values in the dataset. Firstly we dropped those columns which had 50% or more missing values as they are of no use in modelling. Fig 6 shows the code for the procedure.



```
# Drop any column with more than 50% missing values
data = data.dropna(thresh=(len(data)/2),axis=1)
```

Figure 6 Code Snippet For Cleaning Missing Values

Apart from that the columns that has no significance or contribution in prediction has also been dropped. Also, the duplicate columns are removed. The removed columns are shown below.

['id','member_id','funded_amnt','funded_amnt_inv','installment','sub_grade','emp_title','url','desc','title','zip_code','mths_since_last_delinq','mths_since_last_record','out_prncp','out_prncp_inv','total_pymnt','total_pymnt_inv','total_rec_prncp','total_rec_int','total_rec_late_fee','recoveries','collection_recovery_fee','last_pymnt_d','last_pymnt_amnt','next_pymnt_d','last_credit_pull_d','collections_12_mths_ex_med','policy_code','mths_since_last_major_derog','earliest_cr _line','inq_last_6mths','pub_rec','revol_bal','acc_now_delinq','revol_util','initial_list_status','pub_rec_bankruptcies','chargeoff_within_12_mths','tax_liens']

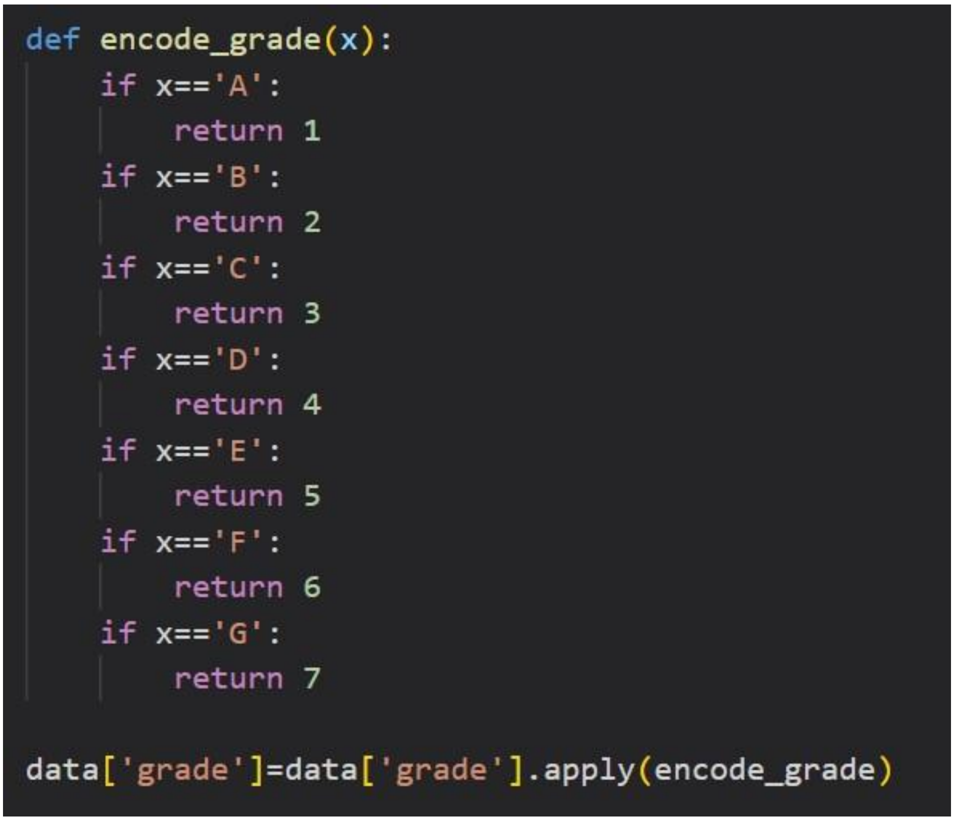
After all this steps we ended up with the necessary columns still having a small number of missing values as shown in the below Table 1.

Table 1 Attributes with Missing Values

Attributes	Number Of Missing Values
LOAN_AMNT	7
TERM	7
INT_RATE	7
GRADE	1119
EMP_LENGTH	7
HOME_OWNERSHIP	7
ANNUAL_INC	11
VERIFICATION_STATUS	7
DTI	7
OPEN_ACC	36
TOTAL_ACC	36
PURPOSE	7
LOAN_STATUS	7

Various techniques are available to handle these missing values although, in this study, these missing values are handled by the "Median" method. As in this method, each missing value is replaced by the median of that particular attribute and this process continues until all the missing values are handled. In certain cases, we have omitted the entire row with missing values as it will not affect the dataset.

The next step in the modification task is related to the data structure of the categorical attributes. As all the values in the categorical attributes are in the character format which cannot be incorporated during the implementation of the algorithm. To cope with this issue all the values in the categorical attributes are converted to numeric as factors. Once the categorical attribute instances are changed from character to numeric, the next step is to convert the data types of the character attributes to numeric.



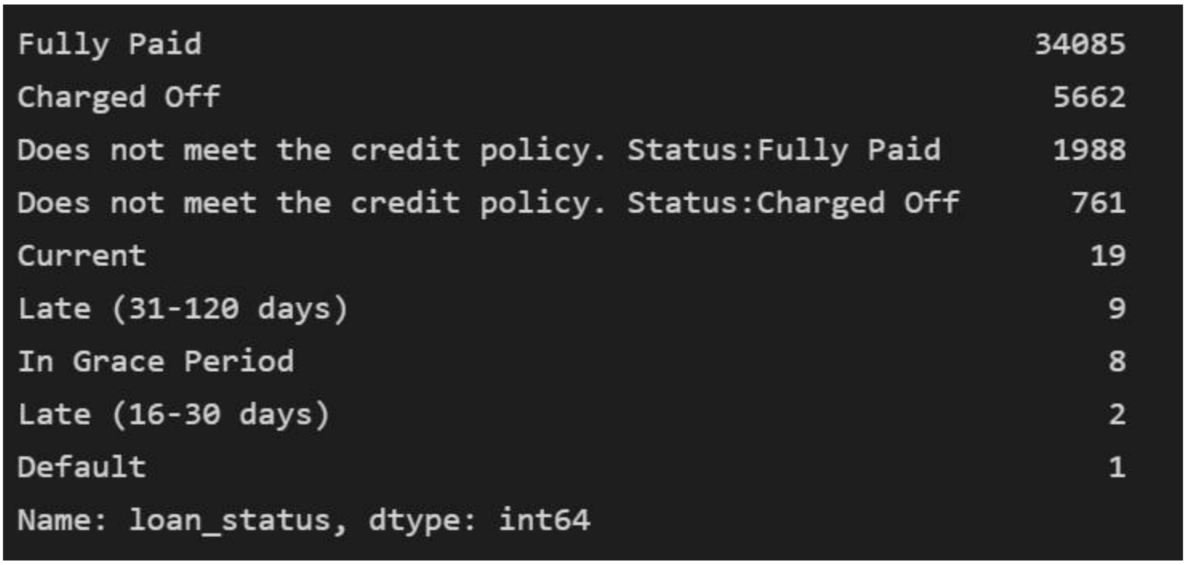
```python
def encode_grade(x):
    if x=='A':
        return 1
    if x=='B':
        return 2
    if x=='C':
        return 3
    if x=='D':
        return 4
    if x=='E':
        return 5
    if x=='F':
        return 6
    if x=='G':
        return 7

data['grade']=data['grade'].apply(encode_grade)
```

Figure 7 Encoding Grades

6.4. Data Analysis

Data Analysis (EDA) played an integral part in understanding the Lending Club dataset. It was vital to get familiar with different relationships within the data through different types of plots before moving towards classification. Analysing these relationships helped us with interpreting the outcomes of the models. Asking questions about these relationships provided us with additional knowledge about relationships that we may not have known existed. This section will further investigate data distribution and ask specific questions about the data lying within the dataset. Lending Club has nine categories of Loan Status. Figure 8 shows the count values for each Loan Status category.



```
Fully Paid                                             34085
Charged Off                                             5662
Does not meet the credit policy. Status:Fully Paid      1988
Does not meet the credit policy. Status:Charged Off      761
Current                                                   19
Late (31-120 days)                                         9
In Grace Period                                            8
Late (16-30 days)                                          2
Default                                                    1
Name: loan_status, dtype: int64
```

Figure 8 Loan Status Categories with Count

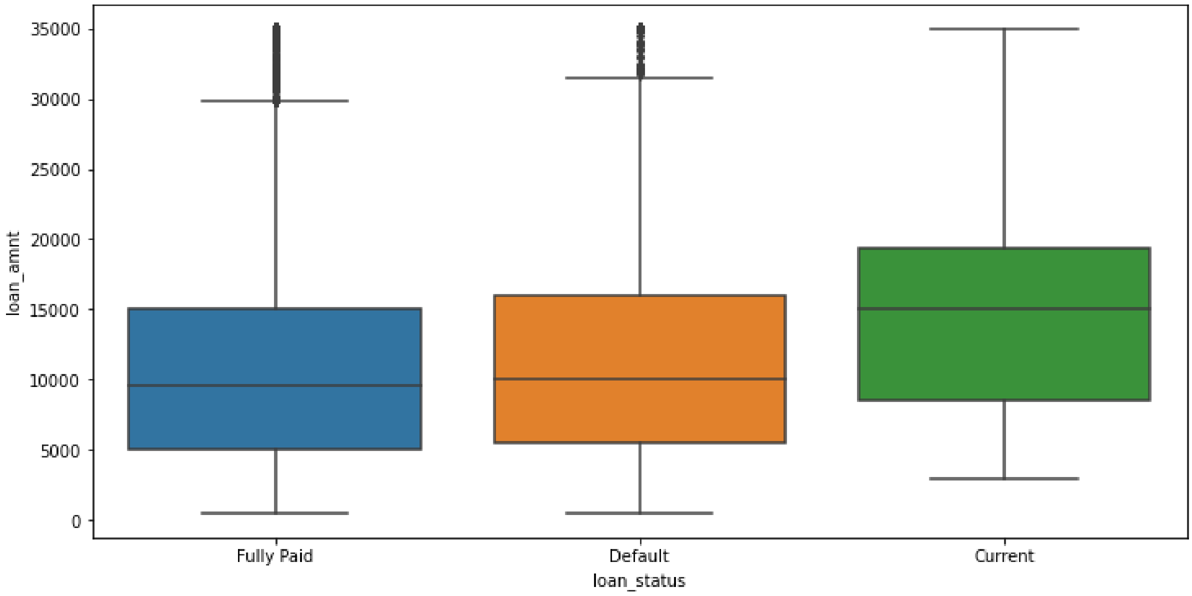
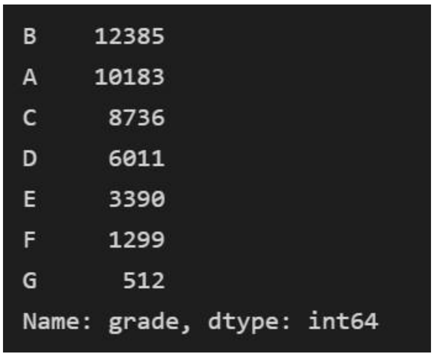


Figure 9 Box Plot For loan_amnt vs loan_status

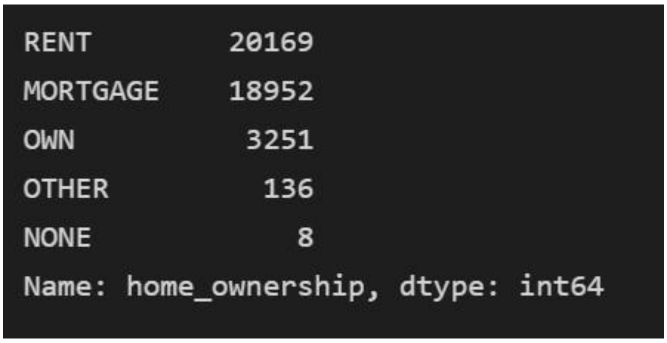
Lending Club has classified loans into seven grades, A - G. And each grade represents a certain level of risk associated as shown in the fig.10.



```
B    12385
A    10183
C     8736
D     6011
E     3390
F     1299
G      512
Name: grade, dtype: int64
```

Figure 10 Grades Categories with Count

Lending Club has also classified borrowers into six home-ownership types as shown in the fig.11.



```
RENT        20169
MORTGAGE    18952
OWN          3251
OTHER         136
NONE            8
Name: home_ownership, dtype: int64
```

Figure 11 Home Ownership Categories with Count

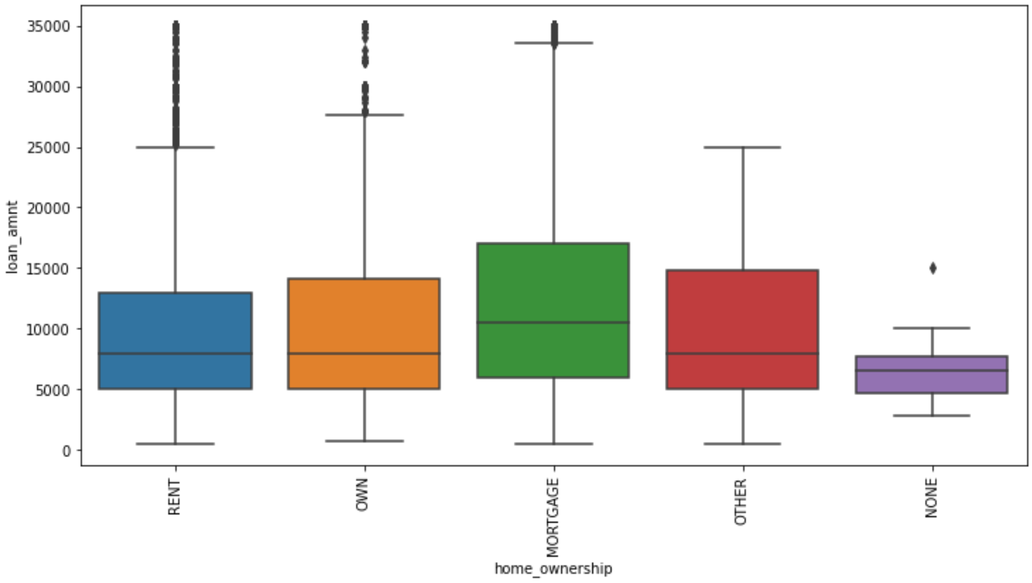
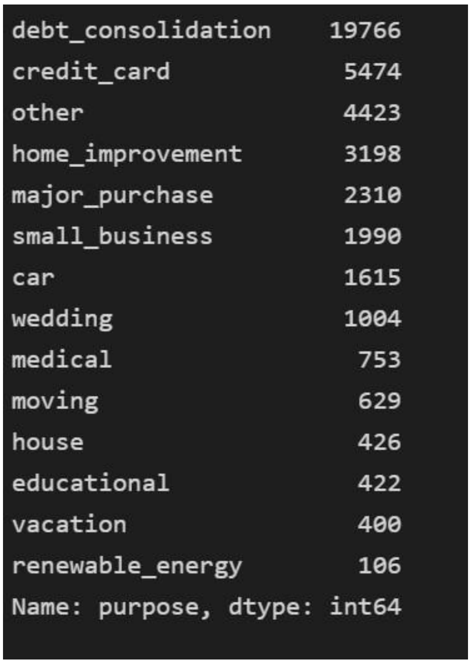


Figure 12 Box Plot For Home Ownership vs Loan Status

According to the figure below, there are 14 different purposes according to the Lending Club dataset individuals apply for loans as shown in the fig.13.



```
debt_consolidation    19766
credit_card            5474
other                  4423
home_improvement       3198
major_purchase         2310
small_business         1990
car                    1615
wedding                1004
medical                 753
moving                  629
house                   426
educational             422
vacation                400
renewable_energy        106
Name: purpose, dtype: int64
```

Figure 13 Purpose Categories with Count

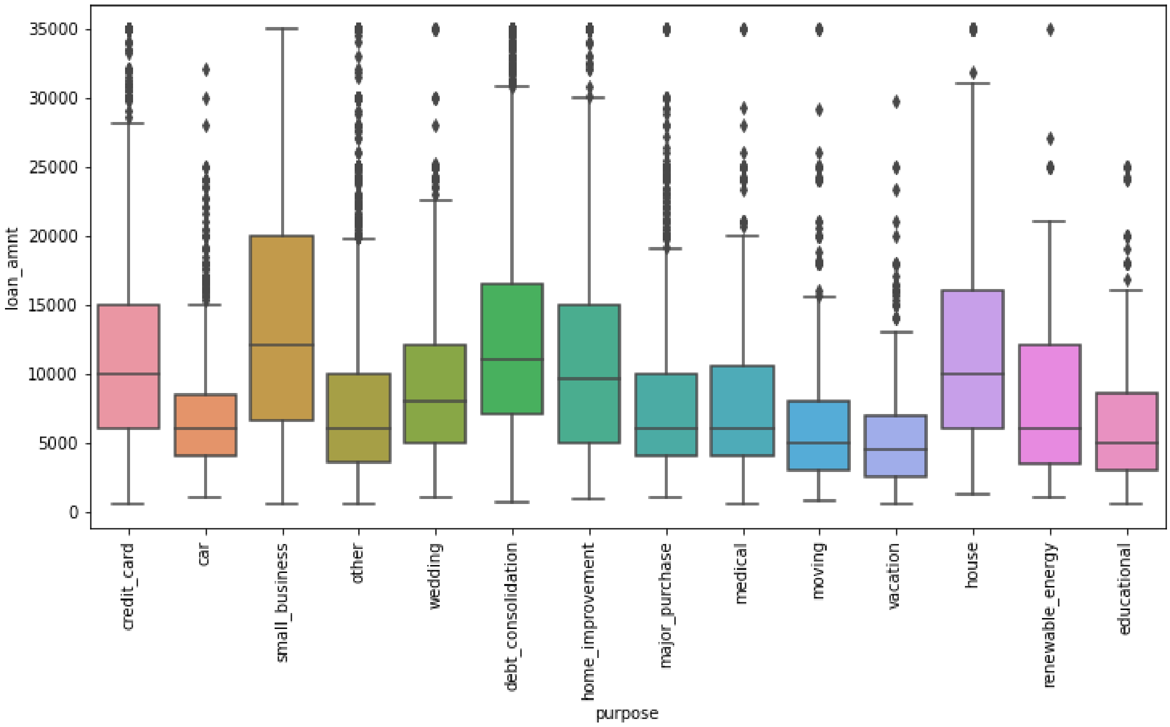


Figure 14 Box Plot For loan_amnt vs purpose

6.5. Modelling

Machine learning is about predicting and recognizing patterns and generate suitable results after understanding them. ML algorithms study patterns in data and learn from them. An ML model will learn and improve on each attempt. To gauge the effectiveness of a model, it's vital to split the data into training and test sets first. So before training our models, we split the Lending Club data into Training set which was 80% of the whole dataset and Test set which was the remaining 20%. Then it was important to implement a selection of performance metrics to the predictions made by our model. In this case, we tried to identify whether an individual is going to default on a loan or not. Model accuracy might not be the sole metric to identify how our model performed-the F1 score and confusion matrix should be important metrics to analyse as well. What is important is that the right performance measures are chosen for the right situations. We used 2 algorithms for our modelling purpose:

6.5.1. Decision Tree.

It was the first algorithm we used for the model. The first thing we did was import the necessary libraries using Scikit-Learn and create a variable for the decision tree classifier. And then fit the data accordingly to train the decision tree model followed by prediction on the test data.

6.5.2. Random Forest.

The second algorithm used for the model. The first step we did was import the necessary libraries using the Scikit-Learn library and create a variable for the random forest classifier. We set the estimator count to 100 for the random forest model. And then fit the data accordingly using the fit function on the classifier followed by prediction on the test data. We further compared the efficiency of the two models which will be shown in the next section.

6.6. Technology Used

This machine learning-based project is implemented using python because of its simple syntax, modular architecture, rich text processing tools and the ability to work on multiple operating systems. Python language is one of the most flexible languages and can be used for various purposes. The language is great to use when working with machine learning algorithms as it contains special libraries for machine learning.

Python packages:

Python is a high-level, interpreted, interactive and object-oriented scripting language. Python is designed to be highly readable. It uses English keywords frequently where as other languages use punctuation, and it has fewer syntactical constructions than other languages.

Numpy

NumPy is the fundamental package for scientific computing with Python. It contains among other things a powerful N-dimensional array-object, sophisticated(broadcasting) functions, tools for integrating C/C++ and Fortran code useful linear algebra, Fourier transform, and random number capabilities. Besides its obvious scientific uses, NumPy can also be used as an efficient multidimensional container of generic data. Arbitrary datatypes can be defined. This allows NumPy to seamlessly and speedily integrate with a wide variety of databases.

Matplotlib

Matplotlib is a Python 2D plotting library which produces publication quality figures in a variety of hardcopy formats and interactive environments across platforms. Matplotlib can be used in Python scripts, the Python and IPython shell, the jupyter notebook, web application servers, and four graphical user interface toolkits.

Pandas

Pandas is an open source, BSD-licensed library providing high performance, easy- to-use data structures and data analysis tools for the Python programming language. Pandas' library is well suited for data manipulation and analysis using python. In particular, it offers data structures and operations for manipulating numerical tables and time series.

Scikit-learn

Scikit-learn provides machine learning libraries for python some of the features of Scikit- learn includes:

- Simple and efficient tools for data mining and data analysis.
- Accessible to everybody, and reusable in various contexts.
- Built on NumPy, SciPy, and matplotlib.
- Open source, commercially usable –BSDlicense.

6.7. Software and Hardware Requirements

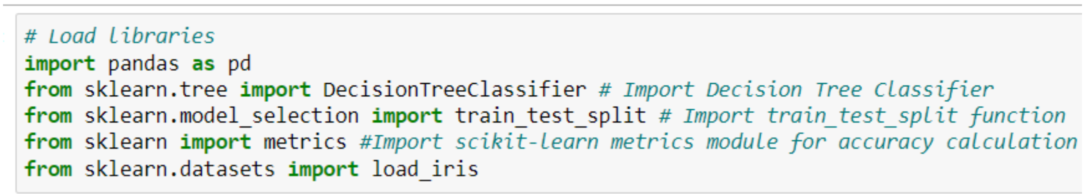
The experiments were conducted on Jupyter Notebook version 6.4.5present within Anaconda-3 version 4.10.3 with Python 3.8.12 running on an Intel Core TM i5 PC with 2.42 GHz CPU with 8GB RAM.

7. Result and Discussions

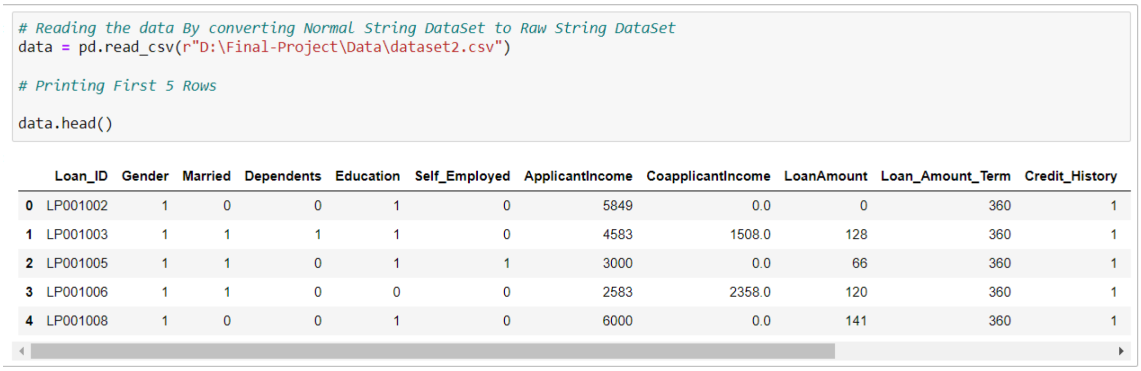
Once the model implementation from all the proposed techniques and validation has been satisfied. Testing data was incorporated in each model and the developed model was employed for the prediction of the loan approval by employing the trained model with Decision Tree (DT), and Random Forest (RF).

Decision Tree

Initially, we processed the dataset and put it up for the test with a decision tree classifier. Firstly, we import all the required packages and read them in the CSV file as shown below.



```
# Load libraries
import pandas as pd
from sklearn.tree import DecisionTreeClassifier # Import Decision Tree Classifier
from sklearn.model_selection import train_test_split # Import train_test_split function
from sklearn import metrics #Import scikit-learn metrics module for accuracy calculation
from sklearn.datasets import load_iris
```



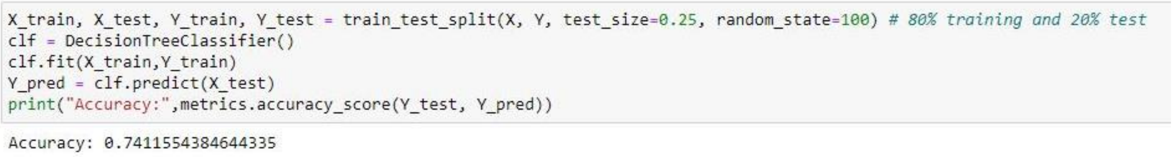
```
# Reading the data By converting Normal String DataSet to Raw String DataSet
data = pd.read_csv(r"D:\Final-Project\Data\dataset2.csv")

# Printing First 5 Rows

data.head()
```

	Loan_ID	Gender	Married	Dependents	Education	Self_Employed	ApplicantIncome	CoapplicantIncome	LoanAmount	Loan_Amount_Term	Credit_History
0	LP001002	1	0	0	1	0	5849	0.0	0	360	1
1	LP001003	1	1	1	1	0	4583	1508.0	128	360	1
2	LP001005	1	1	0	1	1	3000	0.0	66	360	1
3	LP001006	1	1	0	0	0	2583	2358.0	120	360	1
4	LP001008	1	0	0	1	0	6000	0.0	141	360	1

Now we will test the above data, in order to do that we split the data in an 80/20 ratio, where 80% of the data will be fed into the classifier to train it and 20% is kept aside to test its accuracy. The following snippets of code show the same.



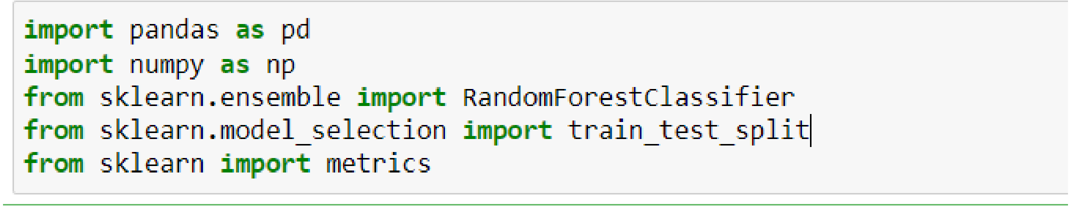
```
X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.25, random_state=100) # 80% training and 20% test
clf = DecisionTreeClassifier()
clf.fit(X_train,Y_train)
Y_pred = clf.predict(X_test)
print("Accuracy:",metrics.accuracy_score(Y_test, Y_pred))
```

```
Accuracy: 0.7411554384644335
```

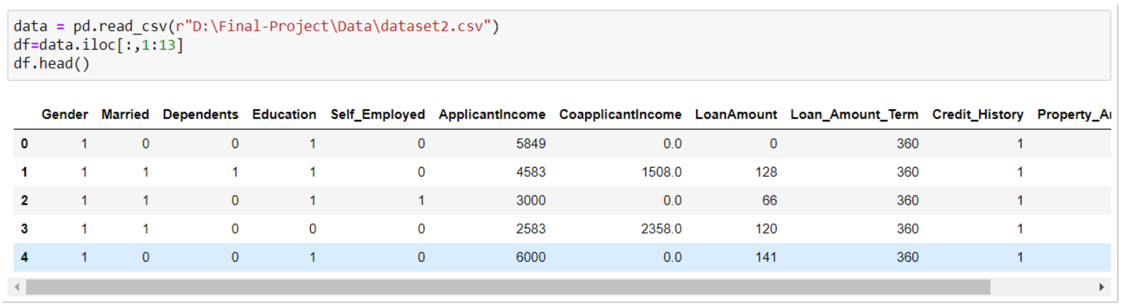
It is evident from the above image the decision tree classifier managed to get an accuracy rate of **74% (0.7411554384)** based on comparing actual test sets output values and predicted values.

Random Forest:

Initially we have processed the dataset and put it up for the test with random forest classifier. Firstly, we import all the required packages and read in the CSV file as shown below.



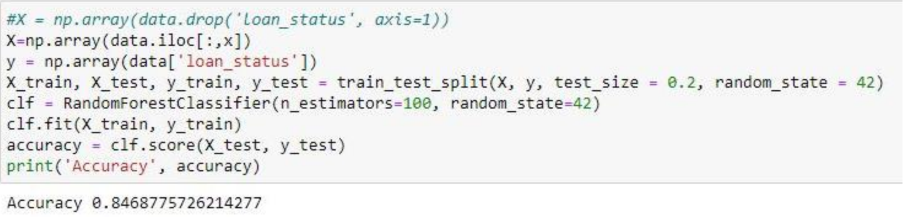
```
import pandas as pd
import numpy as np
from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import train_test_split
from sklearn import metrics
```



```
data = pd.read_csv(r"D:\Final-Project\Data\dataset2.csv")
df=data.iloc[:,1:13]
df.head()
```

	Gender	Married	Dependents	Education	Self_Employed	ApplicantIncome	CoapplicantIncome	LoanAmount	Loan_Amount_Term	Credit_History	Property_Ar
0	1	0	0	1	0	5849	0.0	0	360	1	
1	1	1	1	1	0	4583	1508.0	128	360	1	
2	1	1	0	1	1	3000	0.0	66	360	1	
3	1	1	0	0	0	2583	2358.0	120	360	1	
4	1	0	0	1	0	6000	0.0	141	360	1	

Similarly, here also we split the data in an 80/20 ratio, where 80% of the data will be fed into the classifier to train it and 20% is kept aside to test its accuracy. The following snippets of code show the same.



```
#X = np.array(data.drop('loan_status', axis=1))
X=np.array(data.iloc[:,x])
y = np.array(data['loan_status'])
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.2, random_state = 42)
clf = RandomForestClassifier(n_estimators=100, random_state=42)
clf.fit(X_train, y_train)
accuracy = clf.score(X_test, y_test)
print('Accuracy', accuracy)
```

```
Accuracy 0.846877572621427
```

Comparing actual test sets output values and predicted values Random Forest classifier managed to get an accuracy rate of **84% (0.8468775726).**

For assessment parameters, a confusion matrix will be applied for the model performance evaluation purpose. As the confusion matrix is a reliable method to summarize the performance of the classifier. The confusion matrix presents the number of predictions with respect to their correctness as the name suggested it shows how confused the classifier was during the prediction process. The tabular form is used in the confusion matrix (see Table 2) for the purpose of the description of the classifier.

Table 2 Confusion Matrix

		True Class		
Predicted Class		**Positive**	**Negative**	**Measures**
	Positive	True Positive (TP)	False Positive (FP)	Positive Predictive Value (PPV)= (TP/(TP+FP))
	Negative	False Negative (FN)	True Negative (TN)	Negative Predictive Value (NPV)= (TN/(TN+FN))
	Measures	Sensitivity TP/(TP+FN)	Specificity TN/(FP+TN)	Accuracy (TP+TN)/(TP+FP+TN+FN)

Table No.3 and Table No.4 shows the confusion matrix for Decision Tree Classifier and Random Forest.

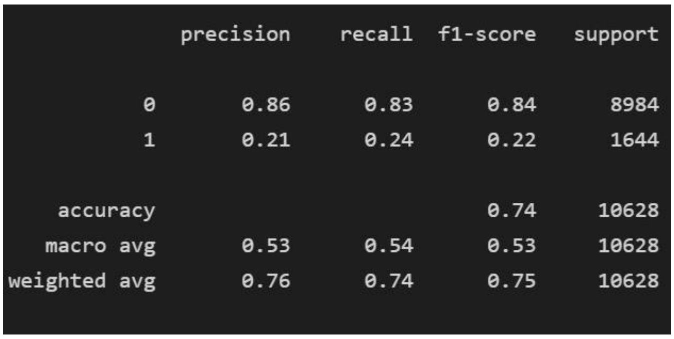
Table 3 Confusion Matrix For Decision Tree Classifier

		True Class	
Predicted Class		**Positive**	**Negative**
	Positive	7486	1498
	Negative	1253	391

Table 4 Confusion Matrix For Random Forest

		True Class	
Predicted Class		**Positive**	**Negative**
	Positive	7180	31
	Negative	1271	21

Another important assessment is the classification report which tells us about the precision, recall also known as sensitivity, f1-score. Precision denotes the percentage of correct outputs among all the returned outputs. Recall denotes the percentage of correct outputs among all the outputs that should be returned. F1-score is the harmonic mean of precision and recall. Classification report for both decision tree and random forest is shown below pictures (Fig.15 and Fig.16).

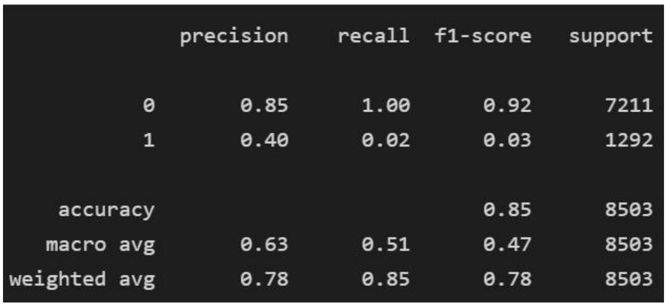


```
              precision    recall  f1-score   support

           0       0.86      0.83      0.84      8984
           1       0.21      0.24      0.22      1644

    accuracy                           0.74     10628
   macro avg       0.53      0.54      0.53     10628
weighted avg       0.76      0.74      0.75     10628
```

Figure 15 Classification Report For Decision Tree



```
              precision    recall  f1-score   support

           0       0.85      1.00      0.92      7211
           1       0.40      0.02      0.03      1292

    accuracy                           0.85      8503
   macro avg       0.63      0.51      0.47      8503
weighted avg       0.78      0.85      0.78      8503
```

Figure 16 Classification Report For Random Forest

From all the above evaluations and recorded results, it is evident that random forest has a greater accuracy in predicting the loan status over decision tree. In each and every respect random forest has a better chance of predicting whether the loan will be defaulted or not.

8. Conclusion

In our study, we have compared the two important machine learning algorithms and tested their ability for loan default prediction. Although each is able to predict the loan defaulter what we were looking for is the accuracy of our models. And from the observations, it seems that random forest has greater accuracy in doing the prediction. We started with a dataset that contained data of different attributes from which we carefully selected the best attributes that will help us to predict more accurate results.

This report aimed to explore, analyse, and suggest a machine learning algorithm to correctly identify whether a person, given certain attributes, has a high probability to default on a loan. This type of model could be used by Lending Club to identify certain financial traits of future borrowers that could have the potential to default and not pay back their loan by the designated time. The Random Forest Classifier provided us with an accuracy of 80% while the Decision

Tree method provided us with an accuracy of 74%. Hence, the Random Forest model appears to be a better option for such kind of data. Lending Club must be careful when identifying potential borrowers who fit certain criteria. For example, borrowers who do not own a home and are applying for a small business or wedding loan, this could be a negative combination that results in the borrower defaulting on a loan.

9. Future Scope

So far, with the limited data we have gathered, we have utilized the same to compare two important Machine learning algorithms. Judging by all respect possible, we found out that Random Forest has greater accuracy in predicting Loan status which implies loads will be paid or will be defaulted. We think that this basic recommendation system can solve the purpose to an extent. As of now we only propose the idea and procedure as a future possibility. However, as the data we required is not available at present implementation will not be possible. But maybe in the future, it can be implemented.

Now the broader idea behind the proposed recommendation system is that the system will take specific details about the loan and return a value or an amount up to which the system suggests to the lender that up to that limit the borrower will be able to pay back the loaned amount. Above that limit that it would not be possible for the borrower to pay back, simply the borrower will default. The value that the system will return can be considered a threshold value or amount. So far the knowledge we gathered from bank and lending companies' policies is that they provide the loan to the borrower based on the valuation of the stake, giving the borrower a percentage of the valuation predefined by the lender. However, they never judge whether the borrower can pay back the loaned amount. Even though the lenders know about the financial condition and other loans if there are any, still they lend the money. We suggest that lenders should judge the borrower's payback capacity based on certain criteria which will be fed into the recommendation system to give the output.

Based on the above information we propose the following procedure:

1. Take the recommendation criteria as inputs
2. Get the LTV ratio(percentage) as per the selected loan type [equation 1]
3. Calculate the maximum possible loan amount that the lender able to lend
4. Calculate the maximum possible payback amount the borrower can return

5. Render the features from the calculated and input data and set them into the ML algorithm
6. Based on the suggested threshold amount offer the loan to the borrower

The recommendation criteria contain several information about the borrower. Firstly, the type or purpose of the loan. A loan can be of many types such as home loans, agricultural loans, business loans, etc. Based on the type, the LTV ratio is calculated. The second criterion is the type of account in the bank. It can either be a savings or a current account. Savings for personal usage and current for business purposes and each has different loan approval criteria. The next criterion is the income details or payslip etc. to get the income amount. Next are ITR files from which expenditures will be calculated. Lastly stake valuation amount. Now all of this sums up as a dataset for a borrower which will be used to make the suggestion. Bank may require other documents to process the loan. But for our proposed system this data is only relevant for the recommendation.

Now next we need the LTV ratio calculation which will be based on the bank's predefined rates for different loan purposes. A loan-to-value (LTV) ratio in a loan is the percentage of the stake's value that a bank or financial institution can lend to a property buyer. The formula used is as follows:

$$\text{LTV Ratio (\%)} = (\text{Amount Borrowed/Stake's Value}) \times 100 \tag{1}$$

Now using this LTV ratio, we will get the maximum possible amount that the bank will be able to lend the borrower.

Now, the important task is to collect the data from income details and ITR files to get the income and gross expenditure that the individual uses up in his/her dealings for day-to-day purposes for personal needs and other purposes such as policy premiums, and tax, subscription bills, etc. From here we calculate the amount left with the individual to pay the loan premiums resulting in calculating the maximum possible payback amount.

Now we first add the calculated and derived data into the dataset that we discussed at the very onset of the Experimentation which will be passed along with the initial data inputs and use the same to run with the machine learning algorithms. With this, the threshold amount is generated as the suggestion to the lender whether to lend or not lend the amount to the borrower.

We studied through different relevant work but it seems our future scope is something promising if rightly implemented.

10. References

[1] Pranckevičius, Tomas and Virginijus Marcinkevicius. “Comparison of Naive Bayes, Random Forest, Decision Tree, Support Vector Machines, and Logistic Regression Classifiers for Text Reviews Classification.” *Balt. J. Mod. Comput.* 5 (2017): n. pag.

[2] Sathyadevan, Shiju and Remya R. Nair. “Comparative Analysis of Decision Tree Algorithms: ID3, C4.5 and Random Forest.” *CI 2015* (2015).

[3] Datkhile, Apurva, et al. "Statistical Modelling on Loan Default Prediction Using Different Models." *IJRESM* 3.3 (2020): 3-5.

[4] Buschjäger, Sebastian, and Katharina Morik. "Decision tree and random forest implementations for fast filtering of sensor data." *IEEE Transactions on Circuits and Systems I: Regular Papers* 65.1 (2017): 209-222.

[5] Ducai, M.T., 2012. The bank loans importance, information asymmetry and the impact of financial and economic crisis on corporate financing. Revista Tinerilor Economi¸sti (The Young Economists Journal), (18), pp.29–34.

[6] Kadam, A., et al. "Prediction for Loan Approval using Machine Learning Algorithm." International Research Journal of Engineering and Technology 8.04 (2021): 4089-4092.

[7] Zhou, Lifeng, and Hong Wang. "Loan default prediction on large imbalanced data using random forests." TELKOMNIKA Indonesian Journal of Electrical Engineering 10.6 (2012): 1519-1525.

[8] Tariq, Hafiz Ilyas, et al. "Loan default prediction model using sample, explore, modify, model, and assess (SEMMA)." Journal of Computational and Theoretical Nanoscience 16.8 (2019): 3489-3503.

[9] Arun, Kumar, Garg Ishan, and Kaur Sanmeet. "Loan approval prediction based on machine learning approach." IOSR J. Comput. Eng 18.3 (2016): 18-21.

[10] https://www.lendingclub.com/