**PREDICTION OF LOAN ELIGIBILITY OF THE KJDKJVLKLHJFVCUSTOMER**

A Project Report Submitted to Jawaharlal Nehru Technological University, Kakinada

In partial fulfillment of the requirements for the award of the degree of

**BACHELOR OF TECHNOLOGY**

Submitted By

**Student Name Roll No**

**C. KHYATHI 21MC1A0518**

**R. SREE VIDYA 21MC1A0596**

**M. MANJULA BAI 22MC5A0505**

**SH. FARHATH PARVEEN 21MC1A05A2**

Under the guidance of

**P. GOVINDARAJU**

 M. Tech, Assistant Professor.

**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING**

RK COLLEGE OF ENGINEERING

(Affiliated to Jawaharlal Nehru Technological University, Kakinada)

An ISO 9001-2015 Certified Institution

Kethanakonda (V), Ibrahimpatnam (Mandal) Vijawada-521456.

2024-2025

**RK COLLEGE OF ENGINEERING**

(Affiliated to Jawaharlal Nehru Technological University, Kakinada)

An ISO 9001-2015 Certified Institution

 Kethanakonda (V), Ibrahimpatnam (Mandal) Vijawada-521456.

**C E R T I F I C A T E**

This is to certify that the Project entitled **“Prediction of Loan Eligibility of the Customer”** Submitted by C. Khyathi, R. Sree Vidya, M. Manjula Bai, SH. Farhath Parveen Bearing Register Number 21MC1A0518,21MC1A0596,22MC5A0505,21MC1A05A2 in partial fulfillment for the award of degree of BACHELOR OF TECHNOLOGY of Jawaharlal Nehru Technological University, Kakinada during the Academic Year 2024-2025.

**PROJECT GUIDE HEAD OF THEDEPARTMENT**

**P. GovindaRaju, M. Tech, K. KoteswaraRao, M. Tech, Phd,**

**Assistant Professor H O D**

**EXTERNAL EXAMINER**

**DECLARATION**

We, C. Khyathi, R. Sree Vidya, M. Manjula Bai, SH. Farhath Parveen declare that this Project Report titled **“Prediction of Loan Eligibility of the Customer ”** has been prepared by me submitted to R.K College of Engineering, An ISO 9001-2015 Certified Institution Ibrahimpatnam, Vijayawada in partial fulfillment of the requirements for the award of the degree of Bachelor of Technology.

I also declare that the Project Report is the result of my team effort and that it has not formed the basis for any degree, diploma and other similar title was not submitted earlier for the award of the degree. I declare myself responsible for declaring the Project Work to submit in JNTU-Kakinada.

Date:

Place:

C. KHYATHI (21MC1A0518)

R. SREE VIDYA (21MC1A0596)

M. MANJULA BAI (22MC5A0505)

SH. FARHATH PARVEEN (21MC1A05A2)

**ACKNOWLEDGEMENT**

I express my sincere thanks to **Sri M.M. Kondaiah,** Chairman, RK College of Engineering, An ISO 9001-2015 Certified Institution for his kind support to pursue my Under Graduation Degree from the reputed College.

My thanks to **Dr M. Mahendranadh,** Secretary of RK College of Engineering, An ISO 9001-2015 Certified Institution for his moral support and encouraging me towards good academic record.

My sincere thanks to dynamic personality **Dr K. Ramakrishnaiah,** Principal, RK College of Engineering, for supporting me to complete my Project Work and I also express my thanks to

My sincere thanks to **Sri K. KotheswaraRao,** HOD for his kind support to complete my Project Work.

My heart full thanks to **Sri P. Govindaraju,** M. Tech, Assistant Professorfor encouraging and guiding me for bringing my Project Work in discipline manner for awarding my Under graduation degree from the JNTU-Kakinada.

If I forget my beloved friends who supported me to complete my Project Work. I thank all my friends supported for my Project Work.

**C. KHYATHI**

**R. SREE VIDYA**

**M. MANJULA BAI**

**SH. FARHATH PARVEEN**

**ABSTRACT**

With the increase in banking sector many people are applying for loans in bank. All these loans are not approvable. The main income of bank assets comes from gain earned from loans. The main objective of banks is to invest their assets in safe customers. Today many banks approve loan after many process of verification and validation but still there is no surety that selected customer is safe or not. Therefore it is important to apply various techniques in banking sector for selecting a customer who pays loan on time. In this report we use random forest algorithm for the classification of data. Random forests algorithm builds a model from trained dataset and this model is applied on test data and we get the required output.

**CONTENTS**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **S.NO.** | **CHAPTER** | | | **PAGE NO** |
|  | **ABSTRACT** | | | I |
|  | **LIST OF FIGURES** | | | II |
| **1** | **INTRODUCTION** | | | **1** |
| **2** | **LITERATURE SURVEY** | | | **2** |
| **3** | **SYSTEM ANALYSIS** | | | **6-15** |
|  | **3.1** | **EXISTING SYSTEM** | | **7** |
|  | **3.2** | **PROPOSED SYSTEM** | | **7** |
|  | **3.3** | **FUNCTIONAL REQUIREMENTS** | | **9** |
|  | **3.4** | **NON-FUNCTIONAL REQUIREMENTS** | | **10** |
|  | **3.5** | **FEASIBILITY STUDY** | | **13** |
|  | **3.6** | **REQUIREMENT SPECIFICATION** | | **15** |
| **4** | **SYSTEM DESIGN** | | | **16-29** |
|  | **4.1** | **MODULE DESCRIPTION** | | **17** |
|  | **4.2** | **SYSTEM ARCHITECTURE** | | **18** |
|  | **4.3** | **ALGORITHMS** | | **19** |
|  | **4.4** | **DATA SET DESCRIPTION** | | **23** |
|  | **4.5** | **UML DIAGRAMS** | | **24-29** |
|  |  | **4.5.1** | **Use case Diagram** | **25** |
|  |  | **4.5.2** | **Class Diagram** | **26** |
|  |  | **4.5.3** | **Sequence Diagram** | **27** |
|  |  | **4.5.4** | **Collaboration Diagram** | **28** |
|  |  | **4.5.5** | **Activity Diagram** | **29** |
| **5** | **TECHNOLOGY DESCRIPTIONS** | | | **30** |
|  | **5.1** | **Introduction to Python** | | **31-41** |
| **6** | **SAMPLE CODE** | | | **42-51** |
| **7** | **TESTING** | | | **52-57** |
|  | **7.1** | **Introduction** | | **53** |
|  | **7.2** | **Sample Test case Specification** | | **57** |
| **8** | **SCREENSHOTS** | | | **58-65** |
| **9** | **CONCLUSION** | | | **66-67** |
| **10** | **BIBLIOGRAPHY** | | | **68-70** |

**II**

**LIST OF FIGURES**

|  |  |  |
| --- | --- | --- |
| **S.NO** | **CHAPTER** | **PAGE NO** |
| 1 | ARCHITECTURE | 18 |
| 2 | 4.5.1. USE CASE DIAGRAM | 25 |
| 3 | 4.5.2. CLASS DIAGRAM | 26 |
| 4 | 4.5.3. SEQUENCE DIAGRAM | 27 |
| 5 | 4.5.4 COLLABORATION DIAGRAM | 28 |
| 6 | 4.5.5. ACTIVITY DIAGRAM | 29 |

**INTRODUCTION**

**1. INTRODUCTION:**

Loan Prediction is very helpful for employee of banks as well as for the applicant also. The aim of this Paper is to provide quick, immediate and easy way to choose the deserving applicants. Various Finance Company deals in all loans. They have presence across all urban, semi urban and rural areas. Customer first apply for loan after that company or bank validates the customer eligibility for loan. Company or bank wants to automate the loan eligibility process (real time) based on customer details provided while filling application form. These details are Gender, Marital Status, Education, Number of Dependents, Applicant Income, Loan Amount, Credit History and other. This project has taken the data of previous customers of various banks to whom on a set of parameters loan were approved. Sothe machine learning model is trained on that record to get accurate results. Our main objective of this project is to predict the safety of loan.

The Loan Eligibility Prediction project aims to develop a machine learning model that predicts whether a customer is eligible for a loan based on their credit history, income, employment status, and other relevant factors. Banks and financial institutions face challenges in determining the creditworthiness of loan applicants, leading to potential losses due to default or bad debts.

Objectives

1. Develop a predictive model to accurately determine loan eligibility.

2. Identify key factors influencing loan eligibility.

3. Improve loan approval process efficiency.

**LITERATURE SURVEY**

**2. LITERATURE SURVEY:**

**2.1] Loan Approval Prediction based on Machine Learning Approach**

With the enhancement in the banking sector lots of people are applying for bank loans but the bank has its limited assets which it has to grant to limited people only, so finding out to whom the loan can be granted which will be a safer option for the bank is a typical process. So in this paper we try to reduce this risk factor behind selecting the safe person so as to save lots of bank efforts and assets. This is done by mining the Big Data of the previous records of the people to whom the loan was granted before and on the basis of these records/experiences the machine was trained using the machine learning model which give the most accurate result. The main objective of this paper is to predict whether assigning the loan to particular person will be safe or not. This paper is divided into four sections

(i)Data Collection

(ii) Comparison of machine learning models on collected data

(iii) Training of system on most promising model

(iv) Testing

**2.2] Developing Prediction Model of Loan Risk in Banks using Data Mining**

Nowadays, There are many risks related to bank loans, for the bank and for those who get the loans. The analysis of risk in bank loans need understanding what is the meaning of risk. In addition, the number of transactions in banking sector is rapidly growing and huge data volumes are available which represent the customers behavior and the risks around loan are increased. Data Mining is one of the most motivating and vital area of research with the aim of extracting information from tremendous amount of accumulated data sets. In this paper a new model for classifying loan risk in banking sector by using data mining. The model has been built using data form banking sector to predict the status of loans. Three algorithms have been used to build the proposed model: j48, bayesNet and naiveBayes. By using Weka application, the model has been implemented and tested. The results has been discussed and a full comparison between algorithms was conducted. J48 was selected as best algorithm based on accuracy.

**2.3] Prediction of Loan Risk using NB and Support Vector Machine**

In recent years, banks and creditors face lots of challenges associated with the bank loan. In addition, banking transactions are growing rapidly and large data sets that represent customer behaviour and risks around debt burdens have increased. So, it became mandatory to know the risks related to bank loans. Each consumer knows what they want and what kind of business they would like to choose. Customer can simply find another bank if administrators are not ready to pay attention. Many data analysis technique exists that focus on loan risk. These techniques are oriented toward extraction of quantity and statistical data properties. They facilitate useful data explanations and help to gain better insights in the process behind data. While there can be traditional data analysis techniques lead to intuition knowledge, it is still made by human researchers. Banks and many investment companies are pioneers using data mining. The proposed work in data mining focused on using data from banking sector to predict the status of loans. Two classification algorithms Naïve Bayes and Support Vector Machine are discussed here. The results have been proven that the speed and accuracy has been improved when combined these two techniques.

**2.4] Loan Prediction by using Machine Learning Models**

Banks are making major part of profits through loans. Though lot of people are applying for loans. It’s hard to select the genuine applicant, who will repay the loan. While doing the process manually, lot of misconception may happen to select the genuine applicant. Therefore we are developing loan prediction system using machine learning, so the system automatically selects the eligible candidates. This is helpful to both bank staff and applicant. The time period for the sanction of loan will be drastically reduced. In this paper we are predicting the loan data by using some machine learning algorithms that is Decision Tree.

**2.5] Clustering Loan Applicants based on Risk Percentage using K-Means Clustering Techniques**

Data mining techniques uses some key ideas for data classification and prediction. Clustering techniques is used to place data items in to similar groups without prior knowledge of group definitions. Clustering provides efficient decision making by grouping large voluminous datasets in bank. Risk assessment is an important task of bank, as the increase and decrease of credit limits in bank depends largely to evaluate the risk properly. The key problem consists of identifying good and bad customer's status those who applied for loan. An improvised risk evaluation of Multi- dimensional Risk prediction clustering Algorithm is implemented to determine the good and bad loan applicants whether they are applicable or not. In order to increase the accuracy of risk, risk assessment is performed in primary and secondary levels. Hence for avoiding Redundancy, Association Rule is integrated. This method allows for finding the risk percentage to determine whether loan can be sanctioned to a customer or not. Finally it is proven that proposed method predicts the better accuracy and consumes less time than existing method.

**SYSTEM ANALYSIS**

**3. SYSTEM ANALYSIS:**

**3.1 EXISTING SYSTEM**

Till now loans are processed by various banks through pen and paperwork. When the large no of customers’ apply for bank loan these bank take lot of time to approve their loan. After approval of loan by the banks, there is no surety that the chosen applicant is capable of paying loan or not. Many banks use their own software’s for the loan approval. In existing system we use data mining algorithms for the loan approval; this is the old technique for the approval of loan. Mutiple data sets are combined and form a Generalised datasets, and different machine learning algorithms are applied to generate results. But these techniques are not up to the mark. Due to this huge banks are suffering from financial crises. To resolve this issue we introduce a new way for approval of loans.

**3.1.1Disadvantages of Existing system**

* Lot of paper work
* Manual maintenance
* Takes lot of time to approve loan
* Requires more man power

**3.2 PROPOSED SYSTEM**

Loan Approval System is software used for approval of loan in banking sector. In this proposed system we have used machine learning algorithm. Machine Learning is process in which a symmetric model is build from the existing dataset; this model is applied for the testing of the new dataset. The system consists of trained dataset and test dataset. The trained dataset is used for construction of model. This model is applied on testing dataset for the required result. In this project we are using machine learning algorithm called Random Forest to predict loan eligibility.

**3.2.1Advantages of Proposed System**

* This system is more secure and reliable
* System returns very accurate results

**3.3 FUNCTIONAL REQUIREMENTS**

In [software engineering](https://en.wikipedia.org/wiki/Software_engineering) and [systems engineering](https://en.wikipedia.org/wiki/Systems_engineering), a **functional requirement** defines a function of a [system](https://en.wikipedia.org/wiki/System) or its component, where a function is described as a specification of behavior between outputs and inputs.[[1]](https://en.wikipedia.org/wiki/Functional_requirement#cite_note-FultonAirborne17-1)

Functional requirements may involve calculations, technical details, data manipulation and processing, and other specific functionality that define what a system is supposed to accomplish.[[2]](https://en.wikipedia.org/wiki/Functional_requirement#cite_note-2) Behavioral requirements describe all the cases where the system uses the functional requirements, these are captured in [use cases](https://en.wikipedia.org/wiki/Use_case). Functional requirements are supported by [non-functional requirements](https://en.wikipedia.org/wiki/Non-functional_requirement) (also known as "quality requirements"), which impose constraints on the design or implementation (such as performance requirements, security, or reliability). Generally, functional requirements are expressed in the form "system must do <requirement>," while non-functional requirements take the form "system shall be <requirement>." The plan for implementing functional requirements is detailed in the system design, whereas *non-functional* requirements are detailed in the system architecture.

As defined in [requirements engineering](https://en.wikipedia.org/wiki/Requirements_analysis), functional requirements specify particular results of a system. This should be contrasted with non-functional requirements, which specify overall characteristics such as cost and [reliability](https://en.wikipedia.org/wiki/Reliability_engineering). Functional requirements drive the application architecture of a system, while non-functional requirements drive the technical architecture of a system.[[4]](https://en.wikipedia.org/wiki/Functional_requirement#cite_note-AdamsNon15-4)

In some cases a requirements analyst generates use cases after gathering and validating a set of functional requirements. The hierarchy of functional requirements collection and change, broadly speaking, is: user/[stakeholder](https://en.wikipedia.org/wiki/Project_stakeholder) request → analyze → use case → incorporate. Stakeholders make a request; systems engineers attempt to discuss, observe, and understand the aspects of the requirement; use cases, entity relationship diagrams, and other models are built to validate the requirement; and, if documented and approved, the requirement is implemented/incorporated.[[6]](https://en.wikipedia.org/wiki/Functional_requirement#cite_note-MITRESys14-6) Each use case illustrates behavioral scenarios through one or more functional requirements. Often, though, an analyst will begin by eliciting a set of use cases, from which the analyst can derive the functional requirements that must be implemented to allow a user to perform each use case.

**Data Collection**

1. Customer Data: Collect customer data (e.g., credit score, income, employment history).

2. Loan Data: Collect loan data (e.g., loan amount, interest rate, repayment terms).

**Data Preprocessing**

1. Data Cleaning: Clean and preprocess customer and loan data.

2. Feature Engineering: Extract relevant features from data (e.g., credit score, income-to-debt ratio).

**Random Forest Model**

1. Model Training: Train a Random Forest model using preprocessed data.

2. Model Evaluation: Evaluate model performance using metrics (e.g., accuracy, precision, recall).

**Prediction Service**

1. Prediction API: Develop an API for predicting loan eligibility.

2. Input Data: Receive customer data as input.

3. Eligibility Prediction: Use Random Forest model to predict loan eligibility.

**Result Display**

1. Result Page: Display loan eligibility result page.

2. Eligibility Status: Display eligibility status (e.g., eligible, not eligible).

3. Next Steps: Provide next steps for the user (e.g., loan application process).

**Admin Management**

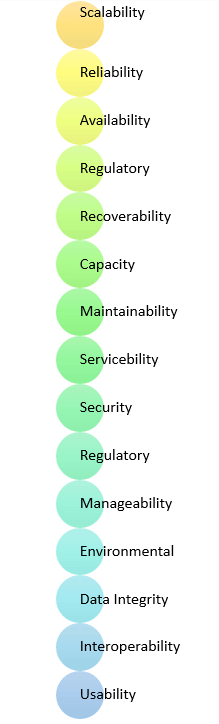
1. Admin Dashboard: Provide an admin dashboard for managing loan applications.

2. Model Monitoring: Monitor model performance.

**3.4 NON-FUNCTIONAL REQUIREMENTS**

(NFR) specifies the quality attribute of a software system. They judge the software system based on Responsiveness, Usability, Security, Portability and other non-functional standards that are critical to the success of the software system. Example of nonfunctional requirement, *“how fast does the website load?”* Failing to meet non-functional requirements can result in systems that fail to satisfy user needs.

Non-functional Requirements allows you to impose constraints or restrictions on the design of the system across the various agile backlogs. Example, the site should load in 3 seconds when the number of simultaneous users are > 10000. Description of non-functional requirements is just as critical as a functional requirement.



Non-Functional Requirements for Loan Eligibility Prediction System using Random Forest

**Performance Requirements**

1. Response Time: The system should respond within 2 seconds.

2. Throughput: The system should handle 100 concurrent requests.

3. Accuracy: The system should achieve an accuracy of 90% or higher.

**Security Requirements**

1. Data Encryption: The system should encrypt customer data.

2. Access Control: The system should implement access controls to ensure authorized access.

3. Compliance: The system should comply with relevant regulations (e.g., GDPR, CCPA).

**Usability Requirements**

1. User Interface: The system should have an intuitive and user-friendly interface.

2. Error Handling: The system should handle errors and exceptions gracefully.

3. Help and Support: The system should provide help and support resources.

**Scalability Requirements**

1. Horizontal Scaling: The system should be able to scale horizontally.

2. Vertical Scaling: The system should be able to scale vertically.

3. Load Balancing: The system should implement load balancing.

**Maintainability Requirements**

1. Code Quality: The system should have high-quality, maintainable code.

2. Testing: The system should have comprehensive testing.

3. Documentation: The system should have detailed documentation.

**Reliability Requirements**

1. Uptime: The system should have 99.9% uptime.

2. Downtime: The system should have minimal downtime.

3. Recovery: The system should have a disaster recovery plan.

**Compatibility Requirements**

1. Operating System: The system should be compatible with multiple operating systems.

2. Browser: The system should be compatible with multiple browsers.

3. Devices: The system should be compatible with multiple devices.

**Advantages of Non-Functional Requirement**

Benefits/pros of Non-functional testing are:

* The nonfunctional requirements ensure the software system follow legal and compliance rules.
* They ensure the reliability, availability, and performance of the software system
* They ensure good user experience and ease of operating the software.
* They help in formulating security policy of the software system.

**Disadvantages of Non-functional requirement**

Cons/drawbacks of Non-function requirement are:

* None functional requirement may affect the various high-level software subsystem
* They require special consideration during the software architecture/high-level design phase which increases costs.
* Their implementation does not usually map to the specific software sub-system,
* It is tough to modify non-functional once you pass the architecture phase.

**KEY LEARNING**

* A non-functional requirement defines the performance attribute of a software system.
* Types of Non-functional requirement are Scalability Capacity, Availability, Reliability, Recoverability, Data Integrity, etc.
* Example of Non Functional Requirement is Employees never allowed to update their salary information. Such attempt should be reported to the security administrator.
* Functional Requirement is a verb while Non-Functional Requirement is an attribute
* The advantage of Non-functional requirement is that it helps you to ensure good user experience and ease of operating the software
* The biggest disadvantage of Non-functional requirement is that it may affect the various high-level software subsystems.

**3.5 FEASIBILITY STUDY**

The feasibility of the project is analyzed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. For feasibility analysis, some understanding of the major requirements for the system is essential.

Three key considerations involved in the feasibility analysis are

* ECONOMICAL FEASIBILITY
* TECHNICAL FEASIBILITY
* SOCIAL FEASIBILITY

**ECONOMICAL FEASIBILITY**

This study is carried out to check the economic impact that the system will have on the organization. The amount of fund that the company can pour into the research and development of the system is limited. The expenditures must be justified. Thus the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased.

**TECHNICAL FEASIBILITY**

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands on the available technical resources. This will lead to high demands being placed on the client. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.

**SOCIAL FEASIBILITY**

The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity. The level of acceptance by the users solely depends on the methods that are employed to educate the user about the system and to make him familiar with it. His level of confidence must be raised so that he is also able to make some constructive criticism, which is welcomed, as he is the final user of the system.

**3.6 REQUIREMENT SPECIFICATION**

**3.6.1 Hardware Requirements**

➢ PROCESSOR - Intel Core I3/i5

➢ RAM - 4 GB

➢ HARD DISK - 500 GB

**3.6.2 Software Requirements**

* Operating system : Windows 8, Windows 10
* Coding Language : PYTHON

**SYSTEM DESIGN**

**4.SYSTEM DESIGN:**

**4.1MODULE DESCRIPTION**

**Data Collection**

The data is saved as a csv file as Loan data.csv and it is read and stored in the life data variable. Data Analysis verify the documents and forward the details to loan evaluator for approval or rejection. System approve the loan if documents are cleared and reject the loan if documents are not cleared Report is delivered to the applicant according to their status.

**Data Pre-processing**

Missing values inside the gathered data might lead to discrepancies. Data must be normalised in order to get better results from the algorithm. We need to eliminate the outliers and transform the variables in order to have a clearer picture. We utilise the chart feature to get to the bottom of these problems.

**Modelling**

Now we should train the model on the training dataset and test dataset. Thus it is chosen ML Models Here We are applied Machine learning algorithm applied. Such as Random Forest

**4.2 SYSTEM ARCHITECTURE:**

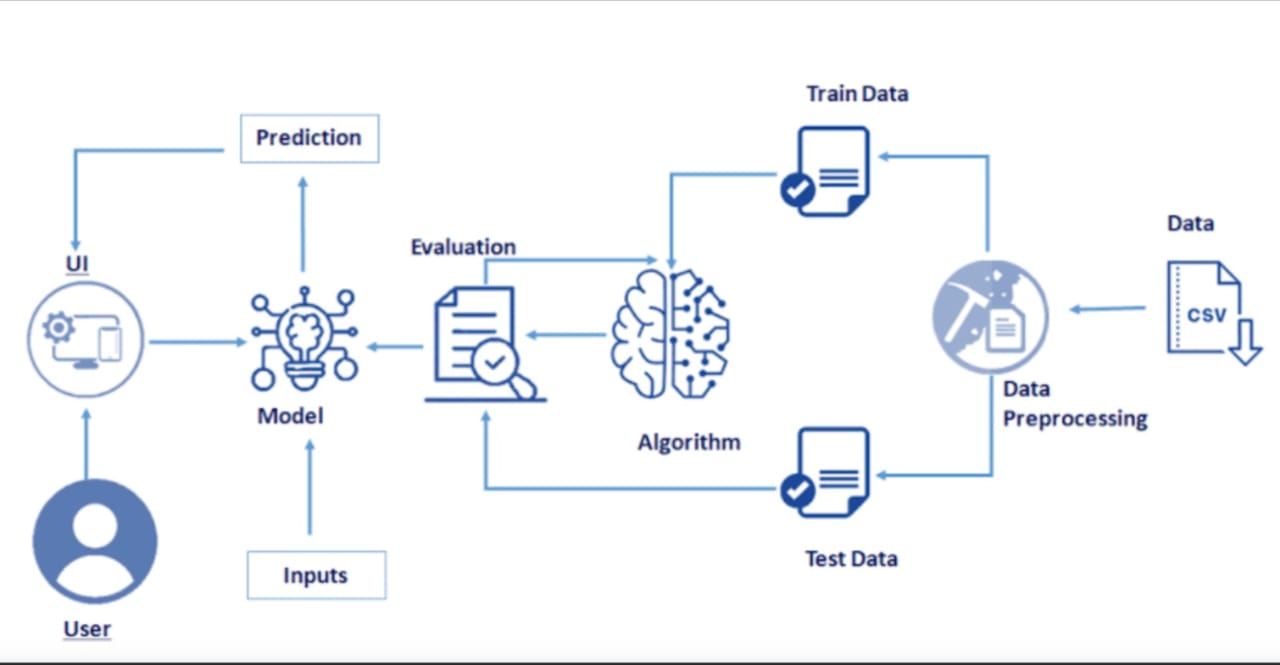


FIG1: Architecture

**4.3 ALGORITHM**

**Random Forest Algorithm**

1. Random Forest is a popular machine learning algorithm that belongs to the supervised learning technique.
2. 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.
3. 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.
4. 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.
5. 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:



**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:



**Python Implementation of Random Forest Algorithm**

Now we will implement the Random Forest Algorithm tree using Python. For this, we will use the same dataset "user\_data.csv", which we have used in previous classification models. By using the same dataset, we can compare the Random Forest classifier with other classification models such as [Decision tree Classifier,](https://www.tpointtech.com/machine-learning-decision-tree-classification-algorithm) [KNN,](https://www.tpointtech.com/k-nearest-neighbor-algorithm-for-machine-learning)[SVM,](https://www.tpointtech.com/machine-learning-support-vector-machine-algorithm) [Logistic Regression,](https://www.tpointtech.com/logistic-regression-in-machine-learning) etc.

Implementation Steps are given below:

1. Data Pre-processing step
2. Fitting the Random forest algorithm to the Training set
3. Predicting the test result
4. Test accuracy of the result (Creation of Confusion matrix)
5. Visualizing the test set result.

**1.Data Pre-Processing Step:**

Below is the code for the pre-processing step:

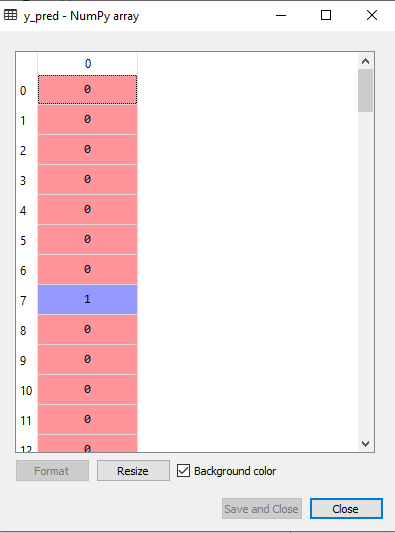
* Importing libraries
* Importing datasets
* Splitting the dataset into training and test set.
* feature Scaling

**2. Fitting the Random Forest algorithm to the training set:**

Now we will fit the Random forest algorithm to the training set. To fit it, we will import theRandom Forest Classifierclass from the sklearn.ensemble library.

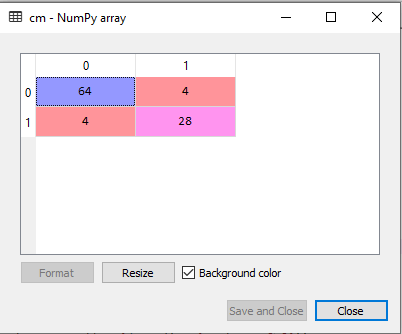
**3. Predicting the Test Set result:**

Since our model is fitted to the training set, so now we can predict the test result. For prediction, we will create a new prediction vector y\_pred.



**4. Creating the Confusion Matrix:**

Now we will create the confusion matrix to determine the correct and incorrect predictions.



**5. Visualizing the training Set result:**

Here we will visualize the training set result. To visualize the training set result we will plot a graph for the Randomforest classifier. The classifier will predict yes or No for the users who have either True or Not False.

So, in the Random Forest classifier, we have taken 10 trees that have predicted Yes or NO for the Purchased variable. The classifier took the majority of the predictions and provided the result.

**4.4 DATA SET DESCRIPTION**

A loan dataset typically contains financial records and associated information used to determine loan eligibility, including factors like income, loan amount, credit history, and loan status, often used for machine learning and data analysis to predict loan approval.

Here's a more detailed breakdown of what you might find in a loan dataset:

**Data Set Features:**

* **Applicant Information:**
  + **Demographics:**Gender, marital status, education level, number of dependents, age.
  + **Employment**: Employment status (e.g., employed, self-employed), job title, years of experience.
  + **Income:**Applicant income, co-applicant income, debt-to-income ratio.
* **Loan Details:**
  + **Loan Amount:**Requested loan amount, loan term (duration), loan type (e.g., personal, business, mortgage).
  + **Credit History:**Credit score, credit history length, number of credit inquiries.
  + **Repayment:**Debt-to-income ratio, revolving balance (unpaid credit card balance).
* **Loan Status:**
  + **Outcome:**Loan approval status (approved/rejected), current loan status (current, charged-off, fully paid).
  + **Repayment History:**Payment history, late payments, default status.
* **Other Relevant Information:**
  + **Property:**Property type (if applicable), property value.
  + **Location:**State/region, zip code.
  + **Homeownership:** Ownership status of the applicant's residence.

**4.5 UML DIAGRAMS**

UML stands for Unified Modeling Language. UML is a standardized general-purpose modeling language in the field of object-oriented software engineering. The standard is managed, and was created by, the Object Management Group.

The goal is for UML to become a common language for creating models of object oriented computer software. In its current form UML is comprised of two major components: a Meta-model and a notation. In the future, some form of method or process may also be added to; or associated with, UML.

The Unified Modeling Language is a standard language for specifying, Visualization, Constructing and documenting the artifacts of software system, as well as for business modeling and other non-software systems.

The UML represents a collection of best engineering practices that have proven successful in the modeling of large and complex systems.

The UML is a very important part of developing objects oriented software and the software development process. The UML uses mostly graphical notations to express the design of software projects.

**GOALS:**

The Primary goals in the design of the UML are as follows:

1. Provide users a ready-to-use, expressive visual modeling Language so that they can develop and exchange meaningful models.
2. Provide extendibility and specialization mechanisms to extend the core concepts.
3. Be independent of particular programming languages and development process.
4. Provide a formal basis for understanding the modeling language.
5. Encourage the growth of OO tools market.
6. Support higher level development concepts such as collaborations, frameworks, patterns and components.
7. Integrate best practices.

**4.5.1 USE CASE DIAGRAM**

A use case diagram in the Unified Modeling Language (UML) is a type of behavioral diagram defined by and created from a Use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases. The main purpose of a use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted



FIGURE 4.4.1 USECASE DIAGRAM

**4.5.2 CLASS DIAGRAM**

In software engineering, a class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among the classes. It explains which class contains information.



FIGURE 4.4.2 CLASS DIAGRAM

**4.5.3 SEQUENCE DIAGRAM**

A sequence diagram in Unified Modeling Language (UML) is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. Sequence diagrams are sometimes called event diagrams, event scenarios, and timing diagrams.

****

FIGURE 4.4.3 SEQUENCE DIAGRAM

**4.5.4 COLLABORATION DIAGRAM**

The collaboration diagram is used to show the relationship between the objects in a system. Both the sequence and the collaboration diagrams represent the same information but differently. Instead of showing the flow of messages, it depicts the architecture of the object residing in the system as it is based on object-oriented programming. An object consists of several features. Multiple objects present in the system are connected to each other. The collaboration diagram, which is also known as a communication diagram, is used to portray the object's architecture in the system.



4.4.4 COLLABORATION DIAGRAM

**4.5.5 ACTIVITY DIAGRAM**

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the Unified Modeling Language, activity diagrams can be used to describe the business and operational step-by-step workflows of components in a system. An activity diagram shows the overall flow of control.

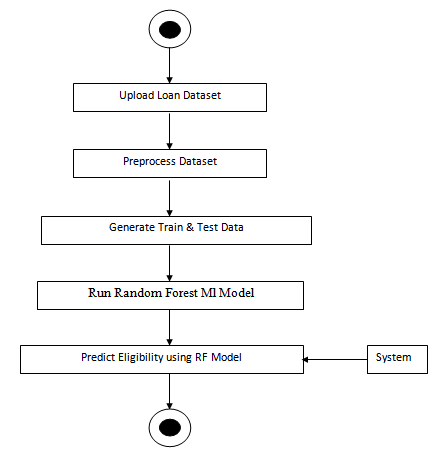


FIGURE 4.4.5 ACTIVITY DIAGRAM

**TECHNOLOGY DESCRIPTION**

**5.TECHNOLOGY DESCRIPTIONS:**

**5.1 Python Introduction**

**Python** is a general purpose, dynamic, high level and interpreted programming language. It supports Object Oriented programming approach to develop applications. It is simple and easy to learn and provides lots of high-level data structures.

Python is easy to learn yet powerful and versatile scripting language which makes it attractive for Application Development.

Python's syntax and dynamic typing with its interpreted nature, makes it an ideal language for scripting and rapid application development.

Python supports multiple programming pattern, including object oriented, imperative and functional or procedural programming styles.

Python is not intended to work on special area such as web programming. That is why it is known as multipurpose because it can be used with web, enterprise, 3D CAD etc.

We don't need to use data types to declare variable because it is dynamically typed so we can write a=10 to assign an integer value in an integer variable.

Python makes the development and debugging fast because there is no compilation step included in python development and edit-test-debug cycle is very fast.

# Python History

* Python laid its foundation in the late 1980s.
* The implementation of Python was started in the December 1989 by **Guido Van Rossum** at CWI in Netherland.
* In February 1991, van Rossum published the code (labeled version 0.9.0) to alt.sources.
* In 1994, Python 1.0 was released with new features like: lambda, map, filter, and reduce.
* Python 2.0 added new features like: list comprehensions, garbage collection system.
* On December 3, 2008, Python 3.0 (also called "Py3K") was released. It was designed to rectify fundamental flaw of the language.
* ABC programming language is said to be the predecessor of Python language which was capable of Exception Handling and interfacing with Amoeba Operating System.
* Python is influenced by following programming languages:
  + ABC language.
  + Modula-3

# Python Features

Python provides lots of features that are listed below.

#### 1) Easy to Learn and Use

Python is easy to learn and use. It is developer-friendly and high level programming language.

#### 2) Expressive Language

Python language is more expressive means that it is more understandable and readable.

#### 3) Interpreted Language

Python is an interpreted language i.e. interpreter executes the code line by line at a time. This makes debugging easy and thus suitable for beginners.

#### 4) Cross-platform Language

Python can run equally on different platforms such as Windows, Linux, Unix and Macintosh etc. So, we can say that Python is a portable language.

#### 5) Free and Open Source

Python language is freely available at [offical web address](https://www.python.org/).The source-code is also available. Therefore it is open source.

#### 6) Object-Oriented Language

Python supports object oriented language and concepts of classes and objects come into existence.

#### 7) Extensible

It implies that other languages such as C/C++ can be used to compile the code and thus it can be used further in our python code.

#### 8) Large Standard Library

Python has a large and broad library and prvides rich set of module and functions for rapid application development.

#### 9) GUI Programming Support

Graphical user interfaces can be developed using Python.

#### 10) Integrated

It can be easily integrated with languages like C, C++, JAVA etc.

# Python Applications

Python is known for its general purpose nature that makes it applicable in almost each domain of software development. Python as a whole can be used in any sphere of development.

Here, we are specifing applications areas where python can be applied.

#### 1) Web Applications

We can use Python to develop web applications. It provides libraries to handle internet protocols such as HTML and XML, JSON, Email processing, request, beautifulSoup, Feedparser etc. It also provides Frameworks such as Django, Pyramid, Flask etc to design and delelop web based applications. Some important developments are: PythonWikiEngines, Pocoo, PythonBlogSoftware etc.

#### 2) Desktop GUI Applications

Python provides Tk GUI library to develop user interface in python based application. Some other useful toolkits wxWidgets, Kivy, pyqt that are useable on several platforms. The Kivy is popular for writing multitouch applications.

#### 3) Software Development

Python is helpful for software development process. It works as a support language and can be used for build control and management, testing etc.

#### 4) Scientific and Numeric

Python is popular and widely used in scientific and numeric computing. Some useful library and package are SciPy, Pandas, IPython etc. SciPy is group of packages of engineering, science and mathematics.

#### 5) Business Applications

Python is used to build Bussiness applications like ERP and e-commerce systems. Tryton is a high level application platform.

#### 6) Console Based Application

We can use Python to develop console based applications. For example: **IPython**.

#### 7) Audio or Video based Applications

Python is awesome to perform multiple tasks and can be used to develop multimedia applications. Some of real applications are: TimPlayer, cplay etc.

#### 8) 3D CAD Applications

To create CAD application Fandango is a real application which provides full features of CAD.

#### 9) Enterprise Applications

Python can be used to create applications which can be used within an Enterprise or an Organization. Some real time applications are: OpenErp, Tryton, Picalo etc.

#### 10) Applications for Images

Using Python several application can be developed for image. Applications developed are: VPython, Gogh, imgSeek etc.

There are several such applications which can be developed using Python

# How to Install Python (Environment Set-up)

In this section of the tutorial, we will discuss the installation of python on various operating systems.

### Why Python

* Python works on different platforms (Windows, Mac, Linux, Raspberry Pi, etc).
* Python has a simple syntax similar to the English language.
* Python has syntax that allows developers to write programs with fewer lines than some other programming languages.
* Python runs on an interpreter system, meaning that code can be executed as soon as it is written. This means that prototyping can be very quick.
* Python can be treated in a procedural way, an object-orientated way or a functional way.

### Good to know

* The most recent major version of Python is Python 3, which we shall be using in this tutorial. However, Python 2, although not being updated with anything other than security updates, is still quite popular.
* In this tutorial Python will be written in a text editor. It is possible to write Python in an Integrated Development Environment, such as Thonny, Pycharm, Netbeans or Eclipse which are particularly useful when managing larger collections of Python files.

### Python Syntax compared to other programming languages

* Python was designed for readability, and has some similarities to the English language with influence from mathematics.
* Python uses new lines to complete a command, as opposed to other programming languages which often use semicolons or parentheses.
* Python relies on indentation, using whitespace, to define scope; such as the scope of loops, functions and classes. Other programming languages often use curly-brackets for this purpose.

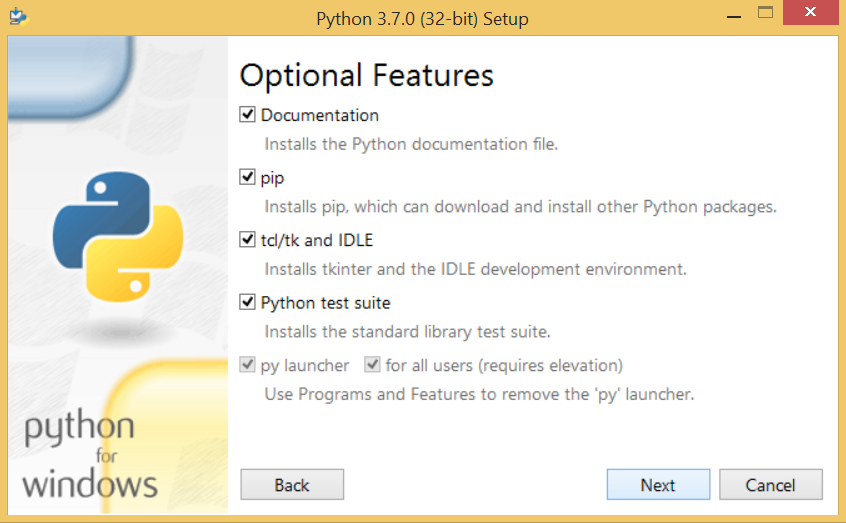
## Installation on Windows

Visit the link <https://www.python.org/downloads/> to download the latest release of Python. In this process, we will install Python 3.6.7 on our Windows operating system.

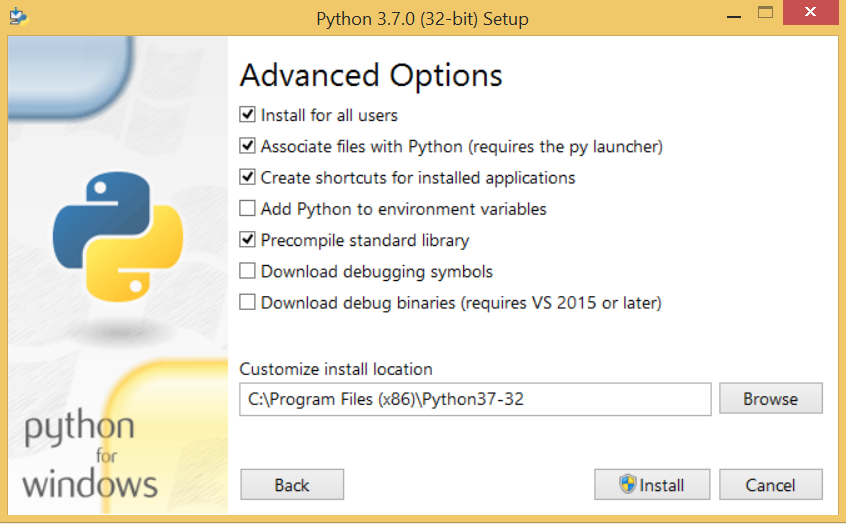
Double-click the executable file which is downloaded; the following window will open. Select Customize installation and proceed.

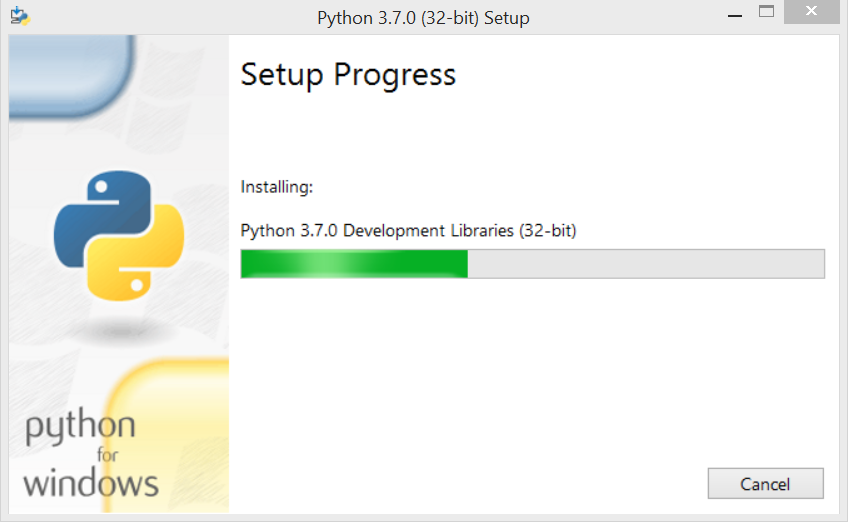
The following window shows all the optional features. All the features need to be installed and are checked by default; we need to click next to continue.

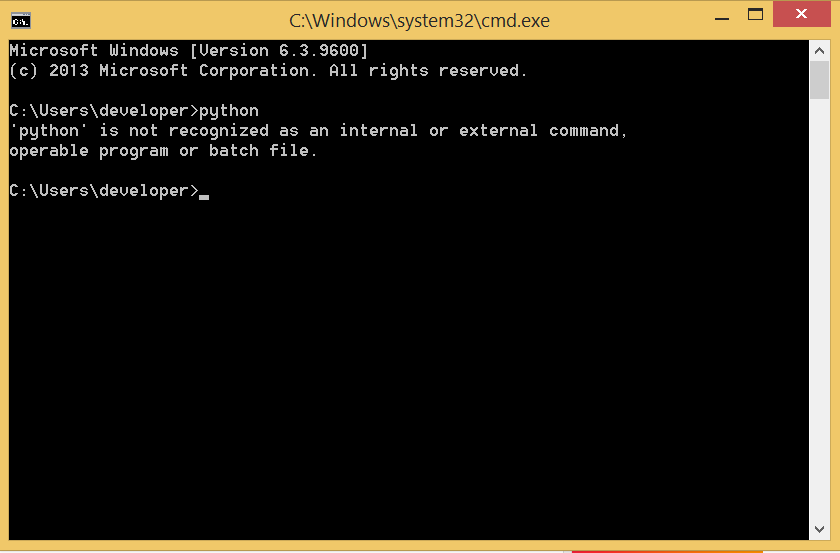
The following window shows a list of advanced options. Check all the options which you want to install and click next. Here, we must notice that the first check-box (install for all users) must be checked.



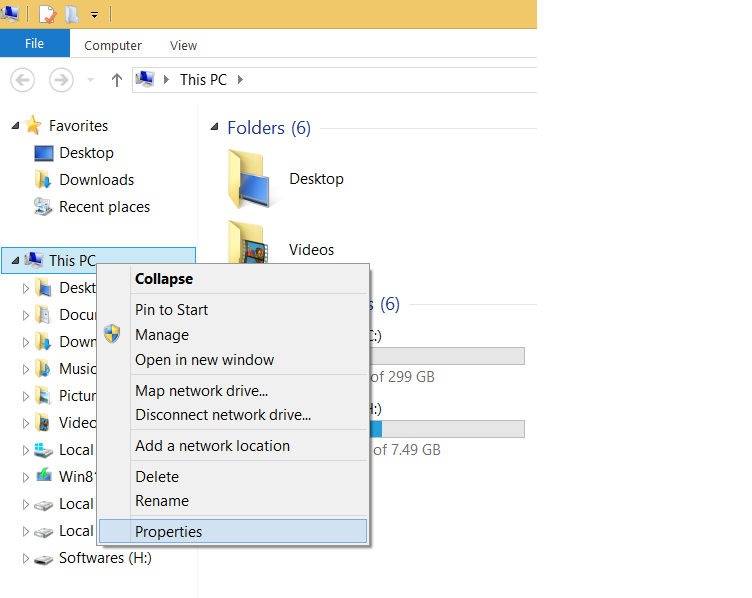
Now, we are ready to install python-3.6.6. Lets install it.

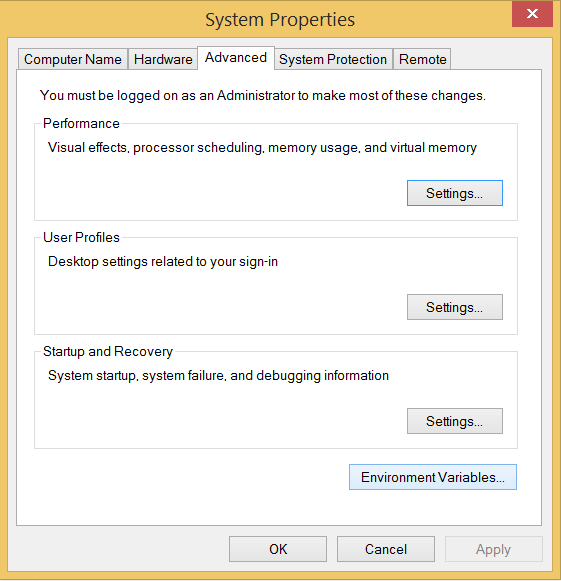


Now, try to run python on the command prompt. Type the command python in case of python2 or python3 in case of python3. It will show an error as given in the below image. It is because we haven't set the path.

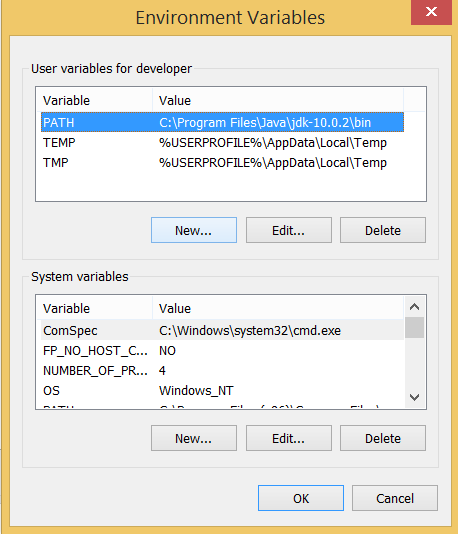


To set the path of python, we need to the right click on "my computer" and go to Properties → Advanced → Environment Variables.

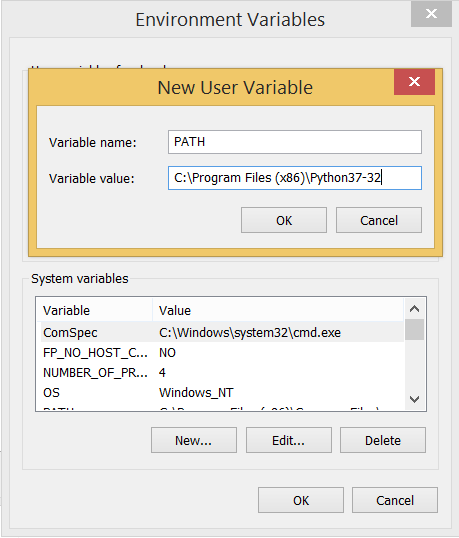




Add the new path variable in the user variable section.



Type PATH as the variable name and set the path to the installation directory of the python shown in the below image.



Now, the path is set, we are ready to run python on our local system. Restart CMD, and type python again. It will open the python interpreter shell where we can execute the python statements.

**Python Application like NumPy for Loan Eligibility Prediction**

NumPy is a library for working with arrays and mathematical operations in Python. Here's how it can be used in loan eligibility prediction:

**Key Features of NumPy**

1. Multi-dimensional arrays: NumPy provides support for large, multi-dimensional arrays and matrices.

2. Vectorized operations: NumPy allows for vectorized operations, making it faster and more efficient.

3. Mathematical functions: NumPy provides a wide range of mathematical functions for tasks like linear algebra, statistics, and more.

**Using NumPy in Loan Eligibility Prediction**

1. Data preprocessing: NumPy can be used to clean and preprocess customer data.

2. Feature engineering: NumPy can be used to extract relevant features from customer data.

3. Model implementation: NumPy can be used to implement machine learning models for loan eligibility prediction.

**Benefits of Using NumPy**

1. Faster and more efficient: NumPy's vectorized operations make it faster and more efficient than traditional Python code.

2. Easier data manipulation: NumPy's multi-dimensional arrays make it easier to manipulate and analyze large datasets.

3. Improved model performance: NumPy's mathematical functions can be used to improve the performance of machine learning models.

**Installation of NumPy**

If you have [Python](https://www.w3schools.com/python/default.asp) and [PIP](https://www.w3schools.com/python/python_pip.asp) already installed on a system, then installation of NumPy is very easy.

Install it using this command:

**C:\Users\*Your Name*>pip install numpy**

If this command fails, then use a python distribution that already has NumPy installed like, Anaconda, Spyder etc.

**Import NumPy:**

Once NumPy is installed, import it in your applications by adding the import keyword:

**import numpy**

Now NumPy is imported and ready to use.

**Python Application like Pandas for Loan Eligibility Prediction**

Pandas is a library for data manipulation and analysis in Python. Here's how it can be used in loan eligibility prediction:

**Key Features of Pandas**

1. Data structures: Pandas provides data structures like Series and DataFrames for data manipulation.

2. Data analysis: Pandas provides functions for data analysis, filtering, and grouping.

3. Data merging: Pandas provides functions for merging and joining datasets.

**Using Pandas in Loan Eligibility Prediction**

1. Data preprocessing: Pandas can be used to clean and preprocess customer data.

2. Feature engineering: Pandas can be used to extract relevant features from customer data.

3. Data analysis: Pandas can be used to analyze customer data and identify patterns.

**Benefits of Using Pandas**

1. Efficient data manipulation: Pandas provides efficient data structures and functions for data manipulation.

2. Easy data analysis: Pandas provides functions for data analysis and filtering.

3. Improved data insights: Pandas can be used to identify patterns and trends in customer.

**Installation of Pandas**

If you have [Python](https://www.w3schools.com/python/default.asp) and [PIP](https://www.w3schools.com/python/python_pip.asp) already installed on a system, then installation of Pandas is very easy.

Install it using this command:

**C:\Users\*Your Name*>pip install pandas**

**Import Pandas:**

Once Pandas is installed, import it in your applications by adding the import keyword:

**import pandas**

**SAMPLE CODING**

**6.** **SAMPLE CODE:**

from tkinter import messagebox

from tkinter import \*

from tkinter import simpledialog

import tkinter

from tkinter import filedialog

import matplotlib.pyplot as plt

import numpy as np

from tkinter.filedialog import askopenfilename

import pandas as pd

import seaborn as sns

from sklearn.metrics import accuracy\_score

from sklearn.model\_selection import train\_test\_split

from sklearn.preprocessing import normalize

from sklearn.ensemble import RandomForestClassifier

from sklearn.metrics import precision\_score

from sklearn.metrics import recall\_score

from sklearn.metrics import f1\_score

from sklearn.preprocessing import LabelEncoder

from sklearn.preprocessing import normalize

main = tkinter.Tk()

main.title("PREDICTION OF LOAN ELIGIBILITY OF THE CUSTOMER") #designing main screen

main.geometry("1300x1200")

global filename

precision = []

recall = []

fscore = []

accuracy = []

global X, Y

global dataset

global X\_train, X\_test, y\_train, y\_test

global le

global classifier

def upload(): #function to upload tweeter profile

global filename

global dataset

filename = filedialog.askopenfilename(initialdir="Dataset")

pathlabel.config(text=filename)

text.delete('1.0', END)

text.insert(END,filename+" loaded\n\n");

dataset = pd.read\_csv(filename)

dataset.fillna(0, inplace = True)

text.insert(END,str(dataset.head()))

dataset['Loan\_ID'] = dataset['Loan\_ID'].astype('str')

dataset['Gender'] = dataset['Gender'].astype('str')

dataset['Married'] = dataset['Married'].astype('str')

dataset['Education'] = dataset['Education'].astype('str')

dataset['Self\_Employed'] = dataset['Self\_Employed'].astype('str')

dataset['Property\_Area'] = dataset['Property\_Area'].astype('str')

dataset['Loan\_Status'] = dataset['Loan\_Status'].astype('str')

print(dataset.info())

sns.set\_style('dark')

dataset.plot(figsize=(18, 8))

plt.show()

def preprocess():

global dataset

global le

text.delete('1.0', END)

le = LabelEncoder()

dataset.drop(['Loan\_ID'], axis = 1,inplace=True)

dataset['Gender'] = pd.Series(le.fit\_transform(dataset['Gender']))

dataset['Married'] = pd.Series(le.fit\_transform(dataset['Married']))

dataset['Education'] = pd.Series(le.fit\_transform(dataset['Education']))

dataset['Self\_Employed'] = pd.Series(le.fit\_transform(dataset['Self\_Employed']))

dataset['Property\_Area'] = pd.Series(le.fit\_transform(dataset['Property\_Area']))

dataset['Loan\_Status'] = pd.Series(le.fit\_transform(dataset['Loan\_Status']))

text.insert(END,str(dataset.head()))

def splitDataset():

text.delete('1.0', END)

global filename

global dataset

global X, Y

global X\_train, X\_test, y\_train, y\_test

dataset = dataset.values

cols = dataset.shape[1]-1

X = dataset[:,0:cols]

Y = dataset[:,cols]

X = normalize(X)

print(Y)

indices = np.arange(X.shape[0])

np.random.shuffle(indices)

X = X[indices]

Y = Y[indices]

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, Y, test\_size=0.2)

text.insert(END,"Total records found in dataset are : "+str(X.shape[0])+"\n")

text.insert(END,"Total records used to train machine learning algorithms are : "+str(X\_train.shape[0])+"\n")

text.insert(END,"Total records used to test machine learning algorithms are : "+str(X\_test.shape[0])+"\n")

dataset = pd.read\_csv(filename)

plt.figure(figsize=(75,75))

sns.heatmap(dataset.corr(), annot = True)

plt.show()

def runRF():

global classifier

precision.clear()

recall.clear()

fscore.clear()

accuracy.clear()

text.delete('1.0', END)

global X\_train, X\_test, y\_train, y\_test

cls = RandomForestClassifier(n\_estimators=200,random\_state=0)

cls.fit(X\_train, y\_train)

predict = cls.predict(X\_test)

p = precision\_score(y\_test, predict,average='macro') \* 100

r = recall\_score(y\_test, predict,average='macro') \* 100

f = f1\_score(y\_test, predict,average='macro') \* 100

a = accuracy\_score(y\_test,predict)\*100

text.insert(END,'Random Forest Accuracy : '+str(a)+"\n")

text.insert(END,'Random Forest Precision : '+str(p)+"\n")

text.insert(END,'Random Forest Recall : '+str(r)+"\n")

text.insert(END,'Random Forest FSCORE : '+str(f)+"\n\n")

accuracy.append(a)

precision.append(p)

recall.append(r)

fscore.append(f)

classifier = cls

def predictEligibility():

global classifier

global le

text.delete('1.0', END)

testname = filedialog.askopenfilename(initialdir = "Dataset")

test = pd.read\_csv(testname)

test.fillna(0, inplace = True)

test['Loan\_ID'] = test['Loan\_ID'].astype('str')

test['Gender'] = test['Gender'].astype('str')

test['Married'] = test['Married'].astype('str')

test['Education'] = test['Education'].astype('str')

test['Self\_Employed'] = test['Self\_Employed'].astype('str')

test['Property\_Area'] = test['Property\_Area'].astype('str')

test.drop(['Loan\_ID'], axis = 1,inplace=True)

test['Gender'] = pd.Series(le.fit\_transform(test['Gender']))

test['Married'] = pd.Series(le.fit\_transform(test['Married']))

test['Education'] = pd.Series(le.fit\_transform(test['Education']))

test['Self\_Employed'] = pd.Series(le.fit\_transform(test['Self\_Employed']))

test['Property\_Area'] = pd.Series(le.fit\_transform(test['Property\_Area']))

test = test.values

test = normalize(test)

cols = test.shape[1]

test = test[:,0:cols]

predict = classifier.predict(test)

print(predict)

for i in range(len(predict)):

if predict[i] == 0:

text.insert(END,"Test Record : "+str(test[i])+" Sorry! Not Eligible for Loan\n\n")

else:

text.insert(END,"Test Record : "+str(test[i])+" Congratulation! You are Eligible for Loan\n\n")

def graph():

df = pd.DataFrame([['Random Forest','Precision',precision[0]],['Random Forest','Recall',recall[0]],['Random Forest','F1 Score',fscore[0]],['Random Forest','Accuracy',accuracy[0]],

],columns=['Parameters','Algorithm','Value'])

df.pivot("Parameters", "Algorithm", "Value").plot(kind='bar')

plt.show()

def close():

main.destroy()

font = ('times', 16, 'bold')

title = Label(main, text='PREDICTION OF LOAN ELIGIBILITY OF THE CUSTOMER')

title.config(bg='brown', fg='white')

title.config(font=font)

title.config(height=3, width=120)

title.place(x=0,y=5)

font1 = ('times', 13, 'bold')

uploadButton = Button(main, text="Upload Loan Dataset", command=upload)

uploadButton.place(x=50,y=100)

uploadButton.config(font=font1)

pathlabel = Label(main)

pathlabel.config(bg='brown', fg='white')

pathlabel.config(font=font1)

pathlabel.place(x=360,y=100)

preprocessButton = Button(main, text="Preprocess Dataset", command=preprocess)

preprocessButton.place(x=50,y=150)

preprocessButton.config(font=font1)

traintestButton = Button(main, text="Generate Train & Test Data", command=splitDataset)

traintestButton.place(x=340,y=150)

traintestButton.config(font=font1)

rfButton = Button(main, text="Run Random Forest ML Model", command=runRF)

rfButton.place(x=630,y=150)

rfButton.config(font=font1)

predictButton = Button(main, text="Predict Eligibility using RF Model", command=predictEligibility)

predictButton.place(x=920,y=150)

predictButton.config(font=font1)

graphButton = Button(main, text="Random Forest Performance Graph", command=graph)

graphButton.place(x=50,y=200)

graphButton.config(font=font1)

closeButton = Button(main, text="Exit", command=close)

closeButton.place(x=340,y=200)

closeButton.config(font=font1)

font1 = ('times', 12, 'bold')

text=Text(main,height=18,width=150)

scroll=Scrollbar(text)

text.configure(yscrollcommand=scroll.set)

text.place(x=10,y=250)

text.config(font=font1)

main.config(bg='brown')

main.mainloop()

**TESTING**

**7. TESTING:**

**7.1 INTRODUCTION**

In general, software engineers distinguish software faults from software failures. In case of a failure, the software does not do what the user expects. A fault is a programming error that may or may not actually manifest as a failure. A fault can also be described as an error in the correctness of the semantic of a computer program. A fault will become a failure if the exact computation conditions are met, one of them being that the faulty portion of computer software executes on the CPU. A fault can also turn into a failure when the software is ported to a different hardware platform or a different compiler, or when the software gets extended. Software testing is the technical investigation of the product under test to provide stakeholders with quality related information.

**System Testing and Implementation**

The purpose is to exercise the different parts of the module code to detect coding errors. After this the modules are gradually integrated into subsystems, which are then integrated themselves too eventually forming the entire system. During integration of module integration testing is performed. The goal of this is to detect designing errors, while focusing the interconnection between modules. After the system was put together, system testing is performed. Here the system is tested against the system requirements to see if all requirements were met and the system performs as specified by the requirements. Finally accepting testing is performed to demonstrate to the client for the operation of the system.

For the testing to be successful, proper selection of the test case is essential. There are two different approaches for selecting test case. The software or the module to be tested is treated as a black box, and the test cases are decided based on the specifications of the system or module. For this reason, this form of testing is also called “black box testing”.

The focus here is on testing the external behavior of the system. In structural testing the test cases are decided based on the logic of the module to be tested. A common approach here is to achieve some type of coverage of the statements in the code. The two forms of testing are complementary: one tests the external behavior, the other tests the internal structure.

Testing is an extremely critical and time-consuming activity. It requires proper planning of the overall testing process. Frequently the testing process starts with the test plan. This plan identifies all testing related activities that must be performed and specifies the schedule, allocates the resources, and specifies guidelines for testing. The test plan specifies conditions that should be tested; different units to be tested, and the manner in which the module will be integrated together. Then for different test unit, a test case specification document is produced, which lists all the different test cases, together with the expected outputs, that will be used for testing. During the testing of the unit the specified test cases are executed and the actual results are compared with the expected outputs. The final output of the testing phase is the testing report and the error report, or a set of such reports. Each test report contains a set of test cases and the result of executing the code with the test cases. The error report describes the errors encountered and the action taken to remove the error.

**Testing Techniques**

Testing is a process, which reveals errors in the program. It is the major quality measure employed during software development. During testing, the program is executed with a set of conditions known as test cases and the output is evaluated to determine whether the program is performing as expected. In order to make sure that the system does not have errors, the different levels of testing strategies that are applied at differing phases of software development are:

**Black Box Testing**

In this strategy some test cases are generated as input conditions that fully execute all functional requirements for the program. This testing has been uses to find errors in the following categories:

* Incorrect or missing functions
* Interface errors
* Errors in data structure or external database access
* Performance errors
* Initialization and termination errors.

In this testing only the output is checked for correctness. The logical flow of the data is not checked.

**White Box Testing**

In this testing, the test cases are generated on the logic of each module by drawing flow graphs of that module and logical decisions are tested on all the cases. It has been uses to generate the test cases in the following cases:

1. Guarantee that all independent paths have been executed.
2. Execute all logical decisions on their true and false sides
3. Execute all loops at their boundaries and within their operational
4. Execute internal data structures to ensure their validity.

**Testing Strategies**

**Unit testing**

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

This System consists of 3 modules. Those are Reputation module, route discovery module, audit module. Each module is taken as unit and tested. Identified errors are corrected and executable unit are obtained.

**Integration testing**

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

**System Testing**

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

**Functional Testing**

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

**Functional testing is centered on the following items**

Valid Input : identified classes of valid input must be accepted

Invalid Input : identified classes of invalid input must be rejected.

Functions : identified functions must be exercised

Output : identified classes of application outputs must be exercised.

Systems/Procedures : interfacing systems or procedures must be invoked

Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes, and successive processes.

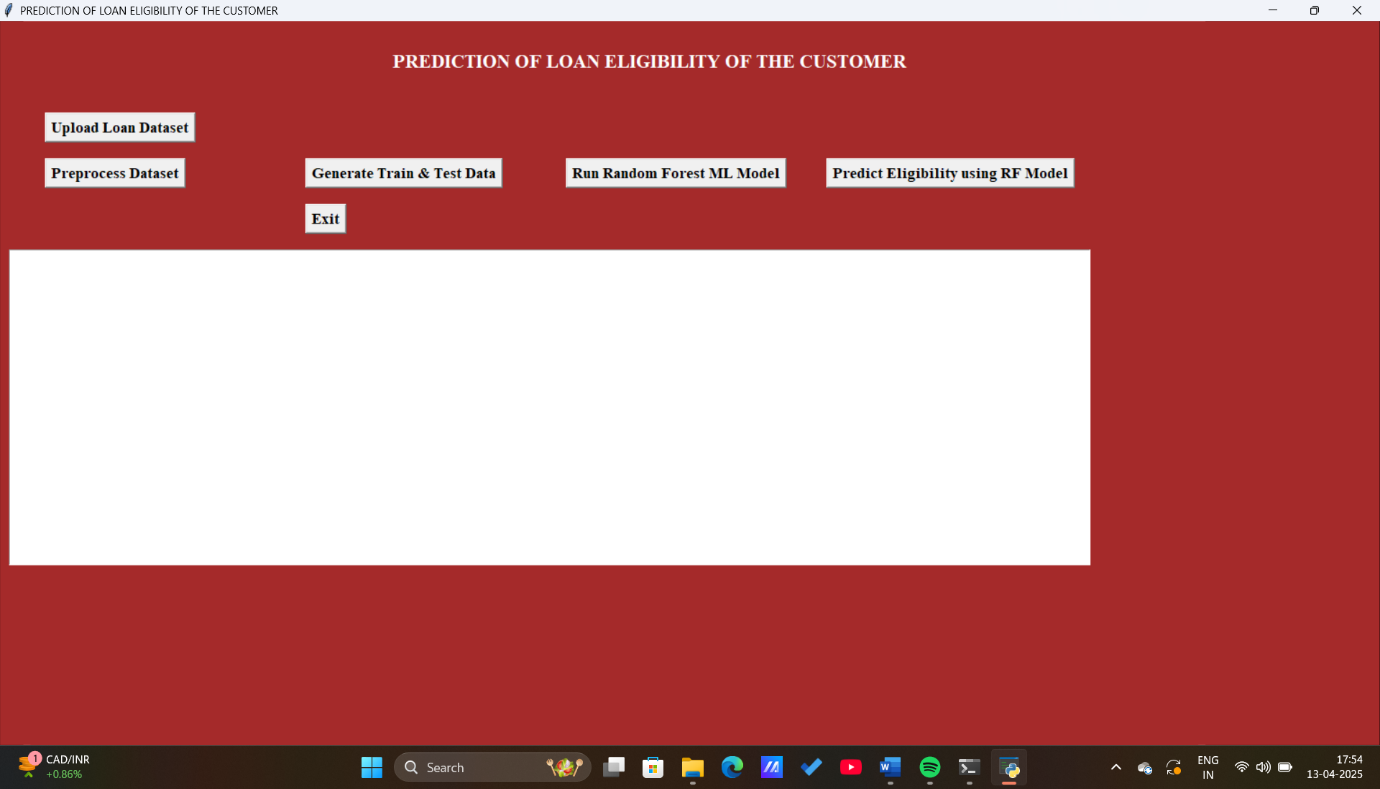
**7.2 SAMPLE TEST CASES SPECIFICATION**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Test | Test Case | Test Case | Test Steps | | | Test | Test |
| Case ID | Name | Description | Step | Expected | Actual | Case | Priority |
| 01 | Start the  Application | Host the  application  and test if it  starts  making sure  the required  software is  available | If it  doesn't  Start | We  cannot  run the  application | The  application  hosts  success. | High | High |
| 02 | Home Page | Check the  deployment  environment  for  properly  loading the  application. | If it  doesn’t  load. | We  cannot  access  the  application | The  application  is running  successfully  . | High | High |
| 03 | User  Mode | Verify the  working of  the  application  in freestyle  mode | If it  doesn’t  Respond | We  cannot  use the  Freestyle  mode. | The  application  displays the  Freestyle  Page | High | High |
| 04 | Data Input | Verify if the  application  takes input  and updates | If it fails  to take the  input or  store in  The  Database | We  cannot  proceed  further | The  application  updates the  input to application | High | High |

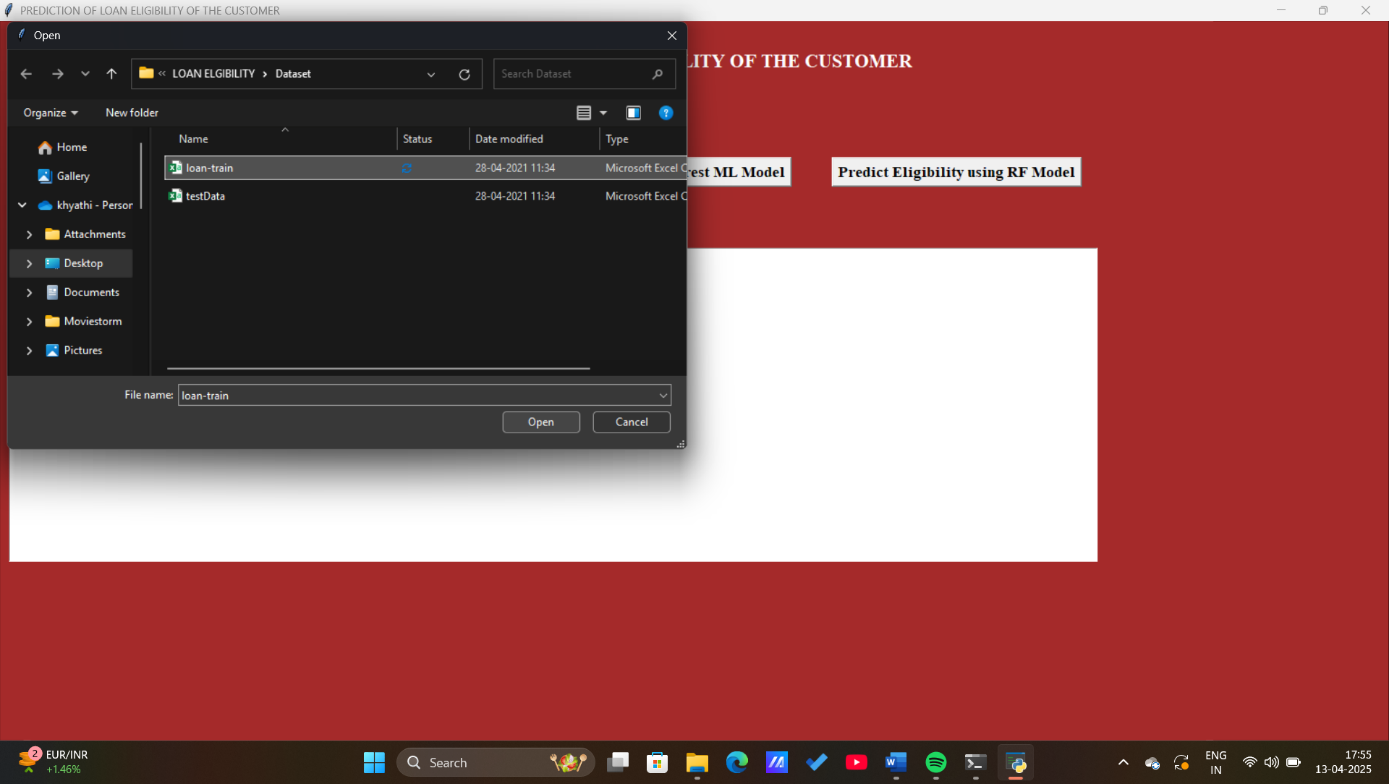
**SCREENSHOTS**

**8. SCREENSHOTS:**

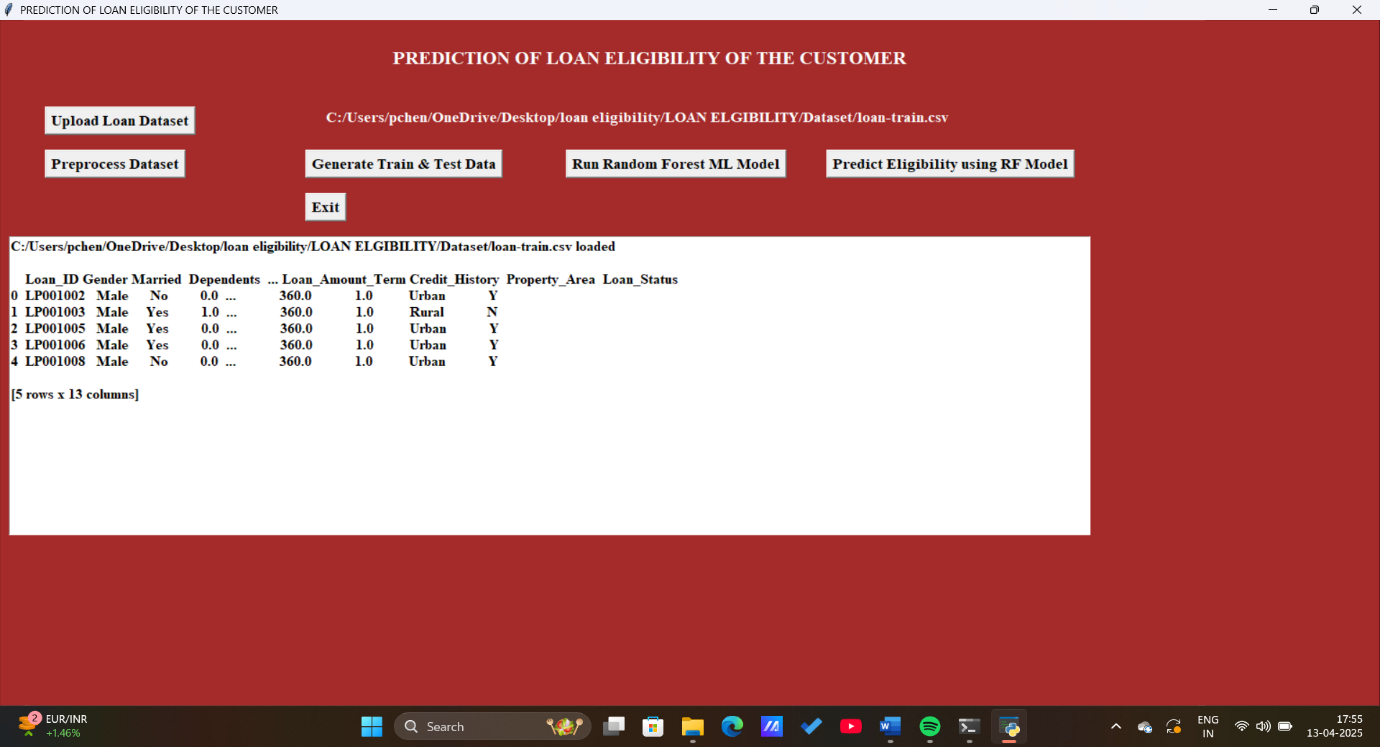
To run project double click on ‘run.bat’ file to get below screen



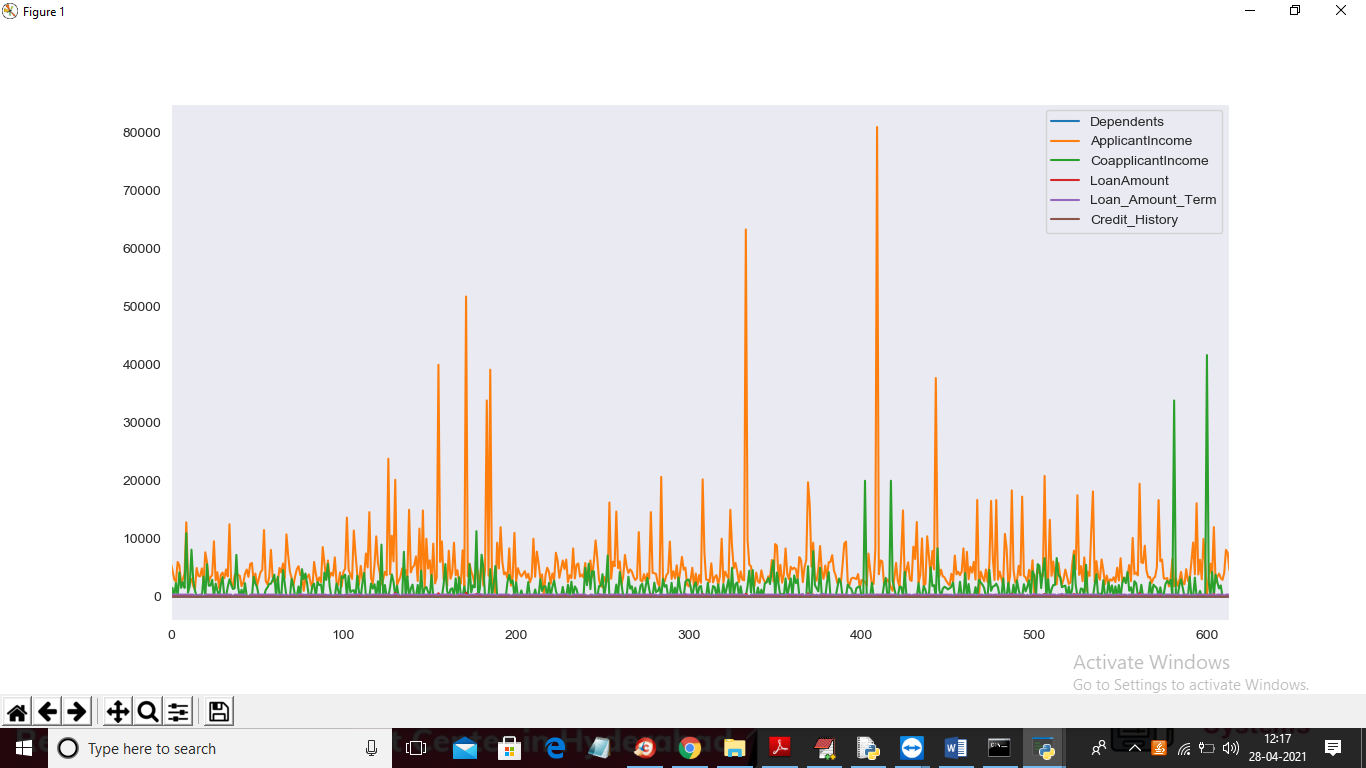
In above screen click on ‘Upload Loan Dataset’ button to load dataset



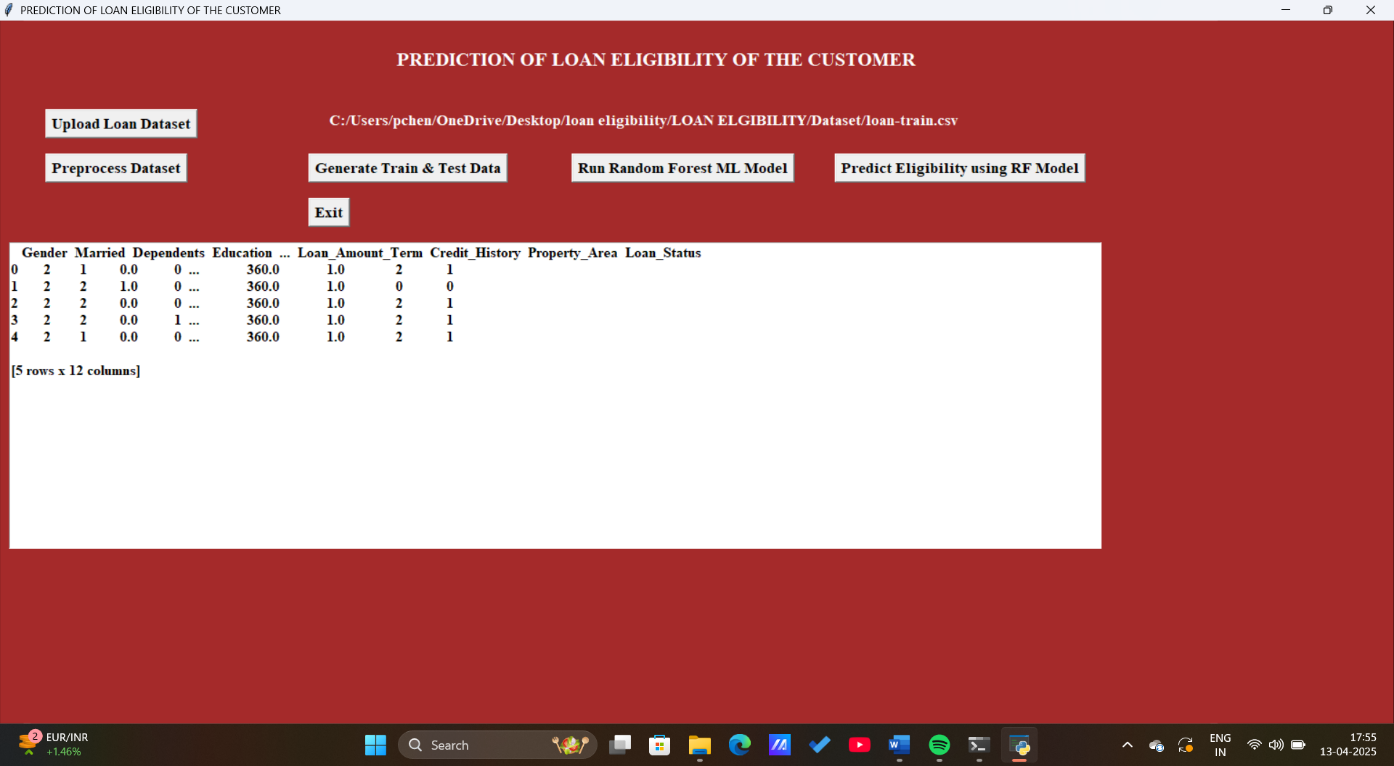
In above screen selecting and uploading ‘loan-train.csv’ file and then click on ‘Open’ button to load dataset and to get below screen



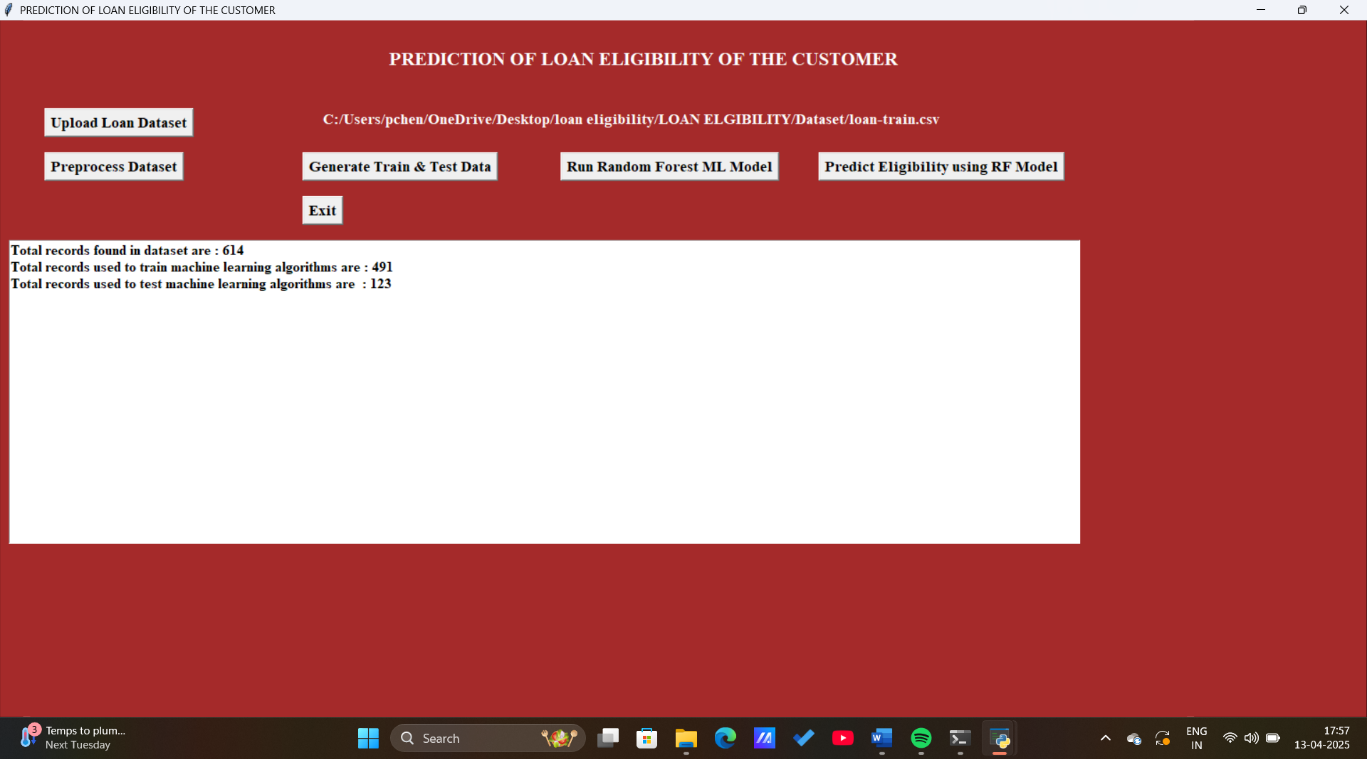
In above screen dataset loaded and all columns contains non-numeric values and machine learning will not accept non-numeric values so we need to convert all those values to numeric by assigning ID’s to them where MALE will replace with 0 and FEMALE will replace with 1 and below graph showing number of different values in dataset



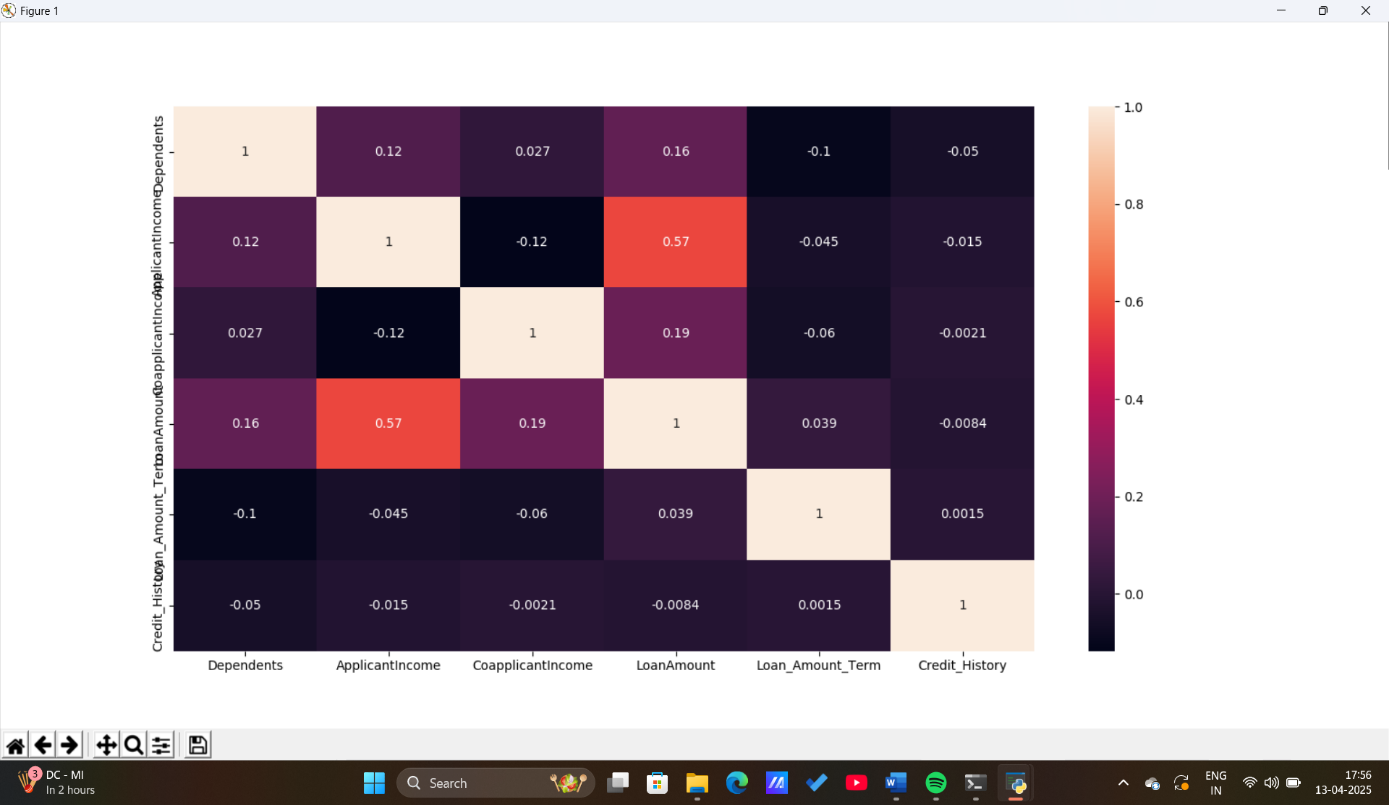
In above graph different colour lines represents counts of that column and you can see column names with colour in graph top right side. Now click on ‘Preprocess Dataset’ button to clean dataset



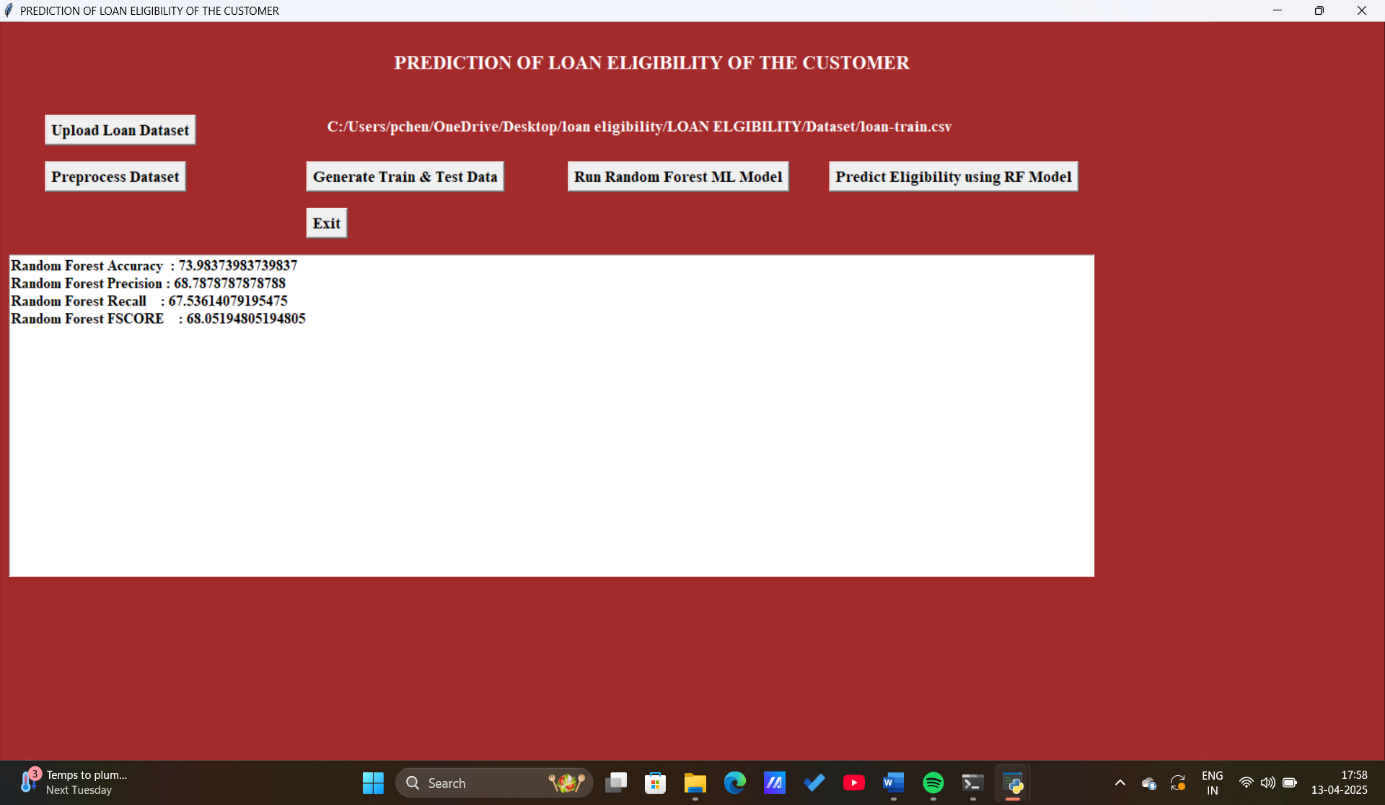
In above screen all non-numeric data is replace with numeric values and now click on ‘Generate Train & Test Data’ button to split dataset into train and test part



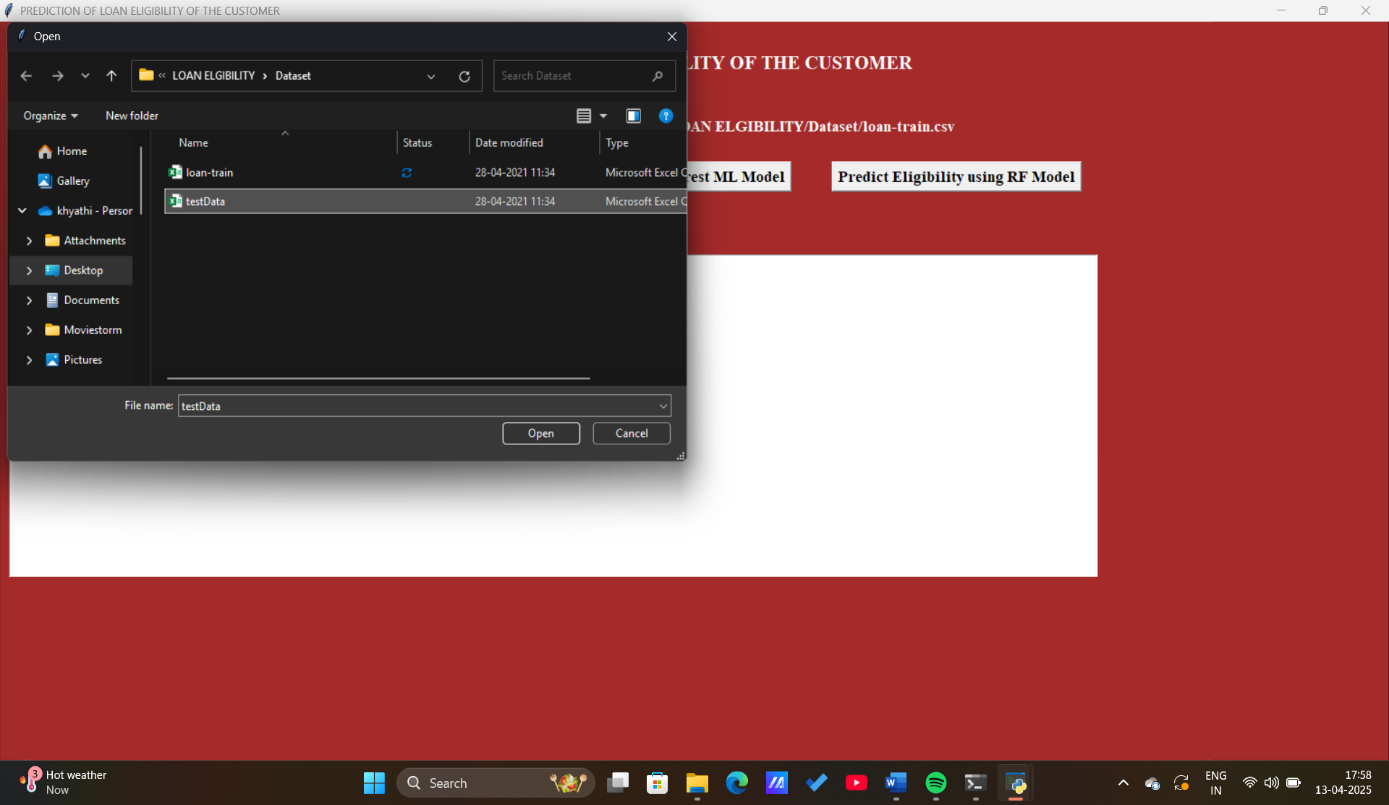
In above screen dataset contains 614 records and using 491 records to train ML and 123 records to test ML accuracy. In below graph we can see importance of each attribute with other attribute by using graph correlation metric



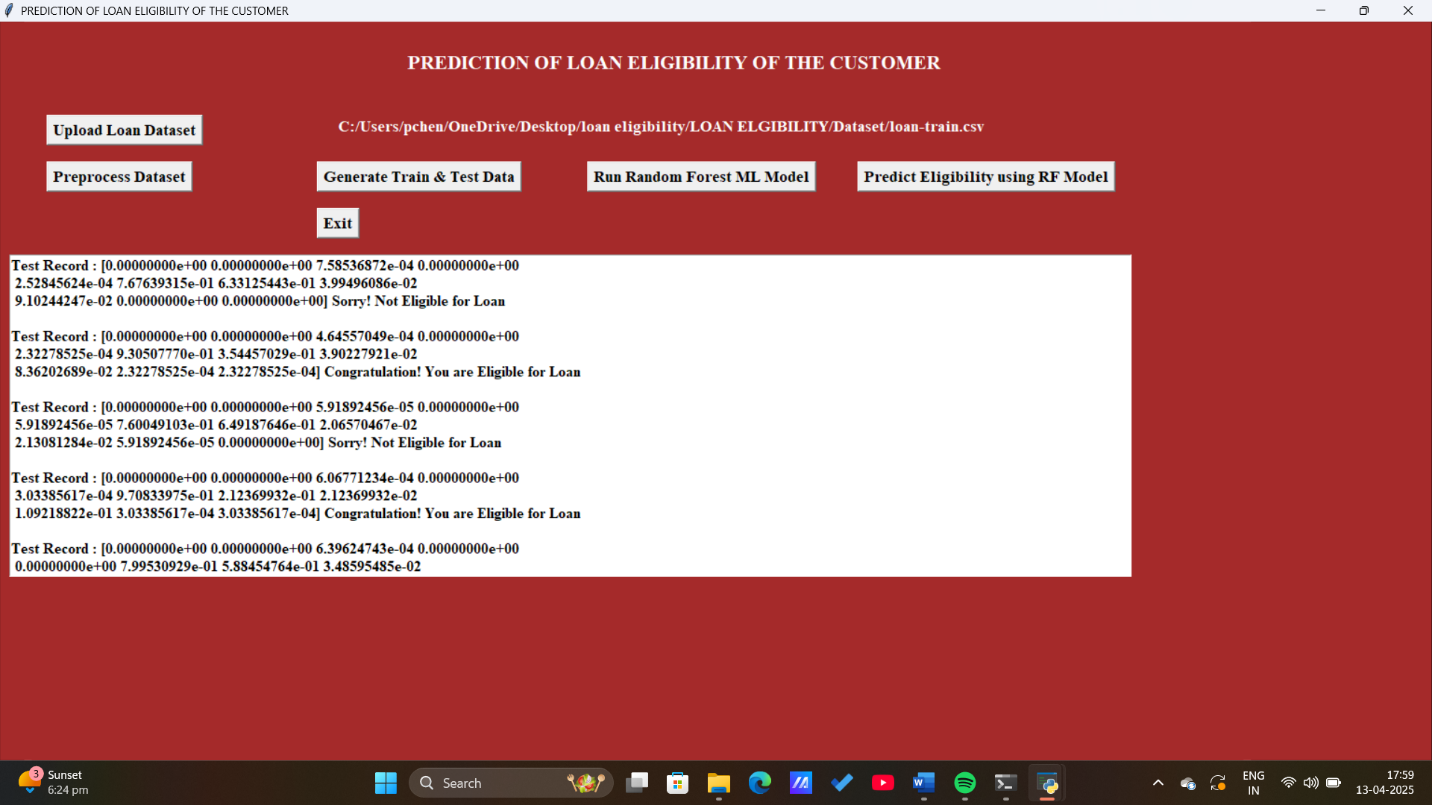
In above graph whatever column in x-axis and y-axis having value >0 will be consider as important features or column. Now click on ‘Run Random Forest Ml Model’ to build random forest model on above dataset



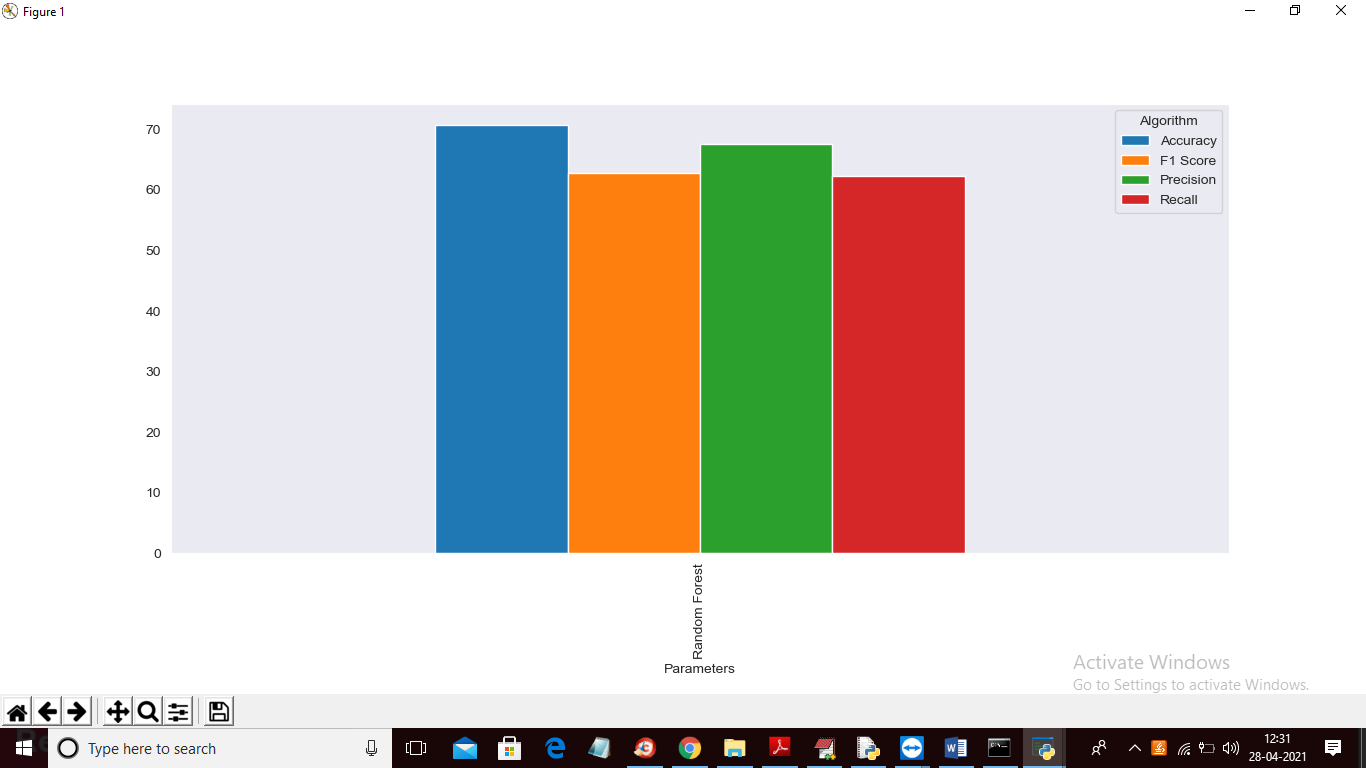
In above screen random forest model generated with 77% accuracy and we can see its precision, recall and FSCORE value and now click on ‘Predict Eligibility using RF Model’ button to upload test data and perform eligibility prediction



In above screen selecting and uploading ‘testData.csv’ file and then click on ‘Open’ button to load test data and then will get below prediction result



In above screen in square bracket we can see normalized test values and after square bracket we can see the prediction result as eligible or not eligible. You can scroll down above text area to view all predicted records and now click on ‘Random Forest Performance Graph’ button to get below graph



In above graph we can see accuracy, precision, recall and FSCORE values of random forest and graph y-axis represents %value where accuracy got 80% and Precision got 65%. Each metric bar colour name you can see from top right side

**CONCLUSION**

**9. CONCLUSION:**

From a proper analysis of positive points and constraints on the component, it can be safely concluded that the product is a highly efficient component. This application is working properly and meeting to all Banker requirements. This component can be easily plugged in many other systems. In this we are using machine learning algorithm called Random Forest to predict loan eligibility.

BIBLIOGRAPY

**10. BIBLIOGRAPHY**

[1] Kumar Arun, Garg Ishan, Kaur Sanmeet, ―Loan Approval Prediction based on Machine Learning Approach‖, IOSR Journal of Computer Engineering (IOSR-JCE), Vol. 18, Issue 3, pp. 79-81, Ver. I (May-Jun. 2016).

[2] Aboobyda Jafar Hamid and Tarig Mohammed Ahmed, ―Developing Prediction Model of Loan Risk in Banks using Data Mining‖, Machine Learning and Applications: An International Journal (MLAIJ), Vol.3, No.1, pp. 1-9, March 2016.

[3] S. Vimala, K.C. Sharmili, ―Prediction of Loan Risk using NB and Support Vector Machine‖, International Conference on Advancements in Computing Technologies (ICACT 2018), vol. 4, no. 2, pp. 110-113, 2018.

[4] Pidikiti Supriya, Myneedi Pavani, Nagarapu Saisushma, Namburi Vimala Kumari, kVikash,“Loan Prediction by using Machine Learning Models”, International Journal of Engineering and Techniques. Volume 5 Issue 2, Mar-Apr 2019

[5] Nikhil Madane, Siddharth Nanda, ”Loan Prediction using Decision tree”, Journal of the Gujrat Research History, Volume 21 Issue 14s, December 2019.

[6] Wei Li, Shuai Ding, Yi Chen, and Shanlin Yang, Heterogeneous Ensemble for Default Prediction of Peer-to-Peer Lending in China, Key Laboratory of Process Optimization and Intelligent DecisionMaking, Ministry of Education, Hefei University of Technology, Hefei 2009,

[7] Short-term prediction of Mortgage default using ensembled machine learning models, Jesse C.Sealand on july 20, 2018.

[8]Clustering Loan Applicants based on Risk Percentage using K-Means Clustering Techniques, Dr. K. Kavitha, International Journal of Advanced Research in Computer Science and Software Engineering.

[9] Toby Segaran, “Programming Collective Intelligence: Building Smart Web 2.0 Applications.” O’Reilly Media.

[10] Drew Conway and John Myles White,” Machine Learning for Hackers: Case Studies and Algorithms to Get you Started,” O’Reilly Media.

[11] Trevor Hastie, Robert Tibshirani, and Jerome Friedman,”The Elements of Statistical Learning: Data Mining, Inference, and Prediction,” Springer ,Kindle

[12] PhilHyo Jin Do ,Ho-Jin Choi, “Sentiment analysis of real-life situations using location, people and time as contextual features,” International Conference on Big Data and Smart Computing (BIGCOMP), pp. 39–42. IEEE, 2015.

[13] Bing Liu, “Sentiment Analysis and Opinion Mining,” Morgan & Claypool Publishers, May 2012.

[14] Bing Liu, “Sentiment Analysis: Mining Opinions, Sentiments, and Emotions,” Cambridge University Press, ISBN:978-1-107- 01789-4.

[15] Shiyang Liao, Junbo Wang, Ruiyun Yu, Koichi Sato, and Zixue Cheng, “CNN for situations understanding based on sentiment analysis of twitter data,” Procedia computer science, 111:376–381, 2017.CrossRef.

[16] K I Rahmani, M.A. Ansari, Amit Kumar Goel, “An Efficient Indexing Algorithm for CBIR,”IEEE- International Conference on Computational Intelligence & Communication Technology ,13-14 Feb 2015.

[17] Gurlove Singh, Amit Kumar Goel ,”Face Detection and Recognition System using Digital Image Processing” , 2nd International conference on Innovative Mechanism for Industry Application ICMIA 2020, 5-7 March 2020, IEEE Publisher.

**Certification**

