A

Major-Project report

On

CUSTOMER LOAN PREDICTION ANALYSIS

Major-Project report submitted in partial fulfillment of the Requirement for the award of the degree of

Bachelor of Technology

In

Computer Science And Engineering

Submitted By

J.VENKATESH (HT NO:19D01A05E0)

Under the guidance of

Dr.D. SASHI RAJA SEKHAR

(Asst. Professor Dept. of C.S.E)

DEPARTMENT COMPUTER SCIENCE AND ENGINEERING

St. Mary’s Group of Institutions

(Affiliated to JNTU, Hyderabad)

Deshmukh (V), Hayathnagar (M), R.R. District-508284

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY

HYDERABAD – 72 (2019-2023)

St. Mary's Group of Institutions Hyderabad

(Approved by AICTE, Permitted by Govt. of T.S., Affiliated to JNTUH, Accredited by NBA, ISO 9001 : 2000 Certified)

Near Ramoji Film City, Behind Mount Opera, Deshmukhi (V), Pochampally (M) Nalgonda (Dt.), & Batasingaram (V),
Hayatnagar (M), Ranga Reddy (Dt.), Greater Hyderabad - 508 284. Under HMDA.

Date:

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

This is to certify that the project work "**CUSTOMER LOAN PREDICTION ANALYSIS**" is Submitted in partial fulfillment of the requirements for the award of the Degree **Bachelor of Technology** in **COMPUTER SCIENCE AND ENGINEERING** from **JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY HYDERABAD** is a bonafied work carried out under the guidance and supervision of **Dr.D. Shashi Raja Shekhar (Asst. Professor dept of CSE)**

S.No	Name of the Student	Hall Ticket No.
1	J.VENKATESH	19D01A05E0

INTERNAL GUIDE **Head of the Department**

EXTERNAL EXAMINAR

ACKNOWLEDGEMENT

This is an acknowledgement of the intensive drive and technical competence of many individuals who have contributed to the success of our project.

I am immensely thankful to my internal guide **Dr. D. Shashi Raja Shekhar, Asst. Professor Dept. of C.S.E,** for his valuable guidance and suggestions in each and every stage of this work, which helped me in completing this project work successfully.

We are obliged and grateful to my **HOD Mr.A.Narender, Asst.Professor** for his sagacious guidance in all respects and for his valuable guidance in each and every stage of this work, which helped me in completing this project work successfully.

My sincere thanks to our **Principal & Professor Dr.V. Goutham** St. Mary's **Group Of Institutions**, Hyderabad and to all my faculty members.

I am grateful to Chairman, St. Mary's Group of Institutions **DR. RAV. K.V.K.RAO** for granting me the permission for undergoing the practical training through development of this project in St. Mary's Group of Institutions, Hyderabad.

We are thankful to one and all who co-operated to us to complete my project successfully.

DECLARATION

I hereby declare that the results embodied in this dissertation entitled **"CUSTOMER LOAN PREDICTION ANALYSIS"** is carried out by me during the year 2019-2023 in partial fulfillment of the award of Degree of **Bachelor of Technology** in Computer Science & Engineering from **St. Mary's Group of Institutions Hyderabad.** I have not submitted the same to any other university or organization for the award of and other degree.

Place: Hyderabad

Students Name with Signature

Date:

J.VENKATESH
(HTNo:19D01A05E0)

ABSTRACT

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 I 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.

INDEX

CONTENTS **PAGE NO**

LIST OF FIGURES

1.INTRODUCTION

The Iris flower data set or Fisher's Iris data set is a multivariate data set introduced by the British statistician and biologist Ronald Fisher in his 1936 paper The use of multiple measurements in taxonomic problems as an example of linear discriminate analysis. It is sometimes called Anderson's Iris data set because Edgar Anderson collected the data to quantify the morphologic variation of Iris flowers of three related species. Two of the three species were collected in the Gaspé Peninsula "all from the same pasture, and picked on the same day and measured at the same time by the same person with the same apparatus". The data set consists of 50 samples from each of three species of Iris (Iris setosa, Iris virginica and Iris versicolor). Four features were measured from each sample: the length and the width of the sepals and petals, in centimeters. Based on the combination of these four features, Fisher developed a linear discriminant model to distinguish the species from each other. The use of this data set in cluster analysis however is not common, since the data set only contains two clusters with rather obvious separation. One of the clusters contains Iris setosa, while the other cluster contains both Iris virginica and Iris versicolor and is not separable without the species information Fisher used. This makes the data set a good example to explain the difference between supervised and unsupervised techniques in data mining: Fisher's linear discriminant model can only be obtained when the object species are known: class labels and clusters are not necessarily the same.

Nevertheless, all three species of Iris are separable in the projection on the nonlinear branching principal component. The data set is approximated by the closest tree with some penalty for the excessive number of nodes, bending and stretching. Then the so-called "metro map" is constructed. The data points are projected into the closest node. For each node the pie diagram of the projected points is prepared.

The area of the pie is proportional to the number of the projected points. It is clear from the diagram (left) that the absolute majority of the samples of the different Iris species belong to the different nodes. Only a small fraction of Iris-virginica is mixed with Iris-versicolor (the mixed blue-green nodes in the diagram). Therefore, the three species of Iris (Iris setosa, Iris virginica and Iris versicolor) are separable by the unsupervising procedures of nonlinear principal component analysis. To discriminate them, it is sufficient just to select the corresponding nodes on the principal tree.

2.LITERATURE SURVEY

Random forest is ensemble learning method for both classification and replaces issues. The advantage of random decision forest is reduce over fitting and helps to improve the accuracy and runs efficiently on a large datasets and work on both continuous and categorical values and predict analysis of data with help of test data.

Bhoomi Patel, Harshal Patil, Jovita Hembram, Shree Jaswal are used data mining methodology to predict the likely default from a dataset that contains information about home loan applications, thereby helping the banks for making better decisions in the future. Xin Li, Xianzhong Long, Guozi Sun, Geng Yang, and Huakang Li This paper mainly introduces the main application of LSTM-SVM model in user loan risk prediction, and elaborates the current economic background, traditional risk forecasting method. On this basis, the prediction methodology based on LSTM method and SVM method is proposed, and the prediction results are compared with the traditional algorithm, and the feasibility of the model is confirm. However, the LSTM-SVM method proposed in this paper actually has few limits and needs to be improved in future research .

Aakanksha, Tamara Denning, Vivek Srikumar, Sneha Kumar Kesera this paper is mainly used for voting classifier (combination of logistic regression, naïve bayes, SVM). They able to reduce the number of FP considerably. This work represents the group of generic passwords to reduce misclassification. Arutjothi present a new credit scoring model, which depends on the hybrid feature selection model and C4.5 classifier. This is depend on hybrid system not only has a strong mathematical basis, but also has higher accuracy and more benefits.

Mrunal Surve, Priya Shinde, Sandip Pandit, Pooja Thitme and Swati Sonawane in this paper, they mainly focus to identify and analyze the risk in giving a loan of commercial banks. To analyze risk in giving loan they have used data mining techniques. It includes analyzing and processing information from various agency/assets and summarize into valuable information . They have used C4.5 classification algorithm for predicting the risk percentage for an individual to give loans.

3.SYSTEM ANALYSIS

3.1 Existing System:

Machine Learning implementation is a very complex part in terms of Data analytics. Working on the data which deals with prediction and making the code to predict the future of out comes from the customer is challenging part.

3.2 Disadvantages of Existing System:

- Complexity in analyzing the data.
- Prediction is challenging task working in the model Coding is complex maintaining multiple methods.
- Libraries support was not that much familiar.

3.3 Proposed System:

Python has a is a good area for data analytical which helps us in analyzing the data with better models in data science. The libraries in python makes the predication for loan data and results with multiple terms considering all properties of the customer in terms of predicting.

3.4 Advantages:

- Libraries helps to analyse the data.
- Statistical and prediction is very easy comparing to existing technologies.
- Results will be accurate compared to other methodologies.

4.SYSTEM SPECIFICATIONS

SYSTEM REQUIREMENTS:

4.1 SOFTWARE REQUIREMENTS:

- Operating System : Windows 7 , Windows 8, (or higher versions)
- Language : Python 3.5
- Mozilla Firefox(or any browser)

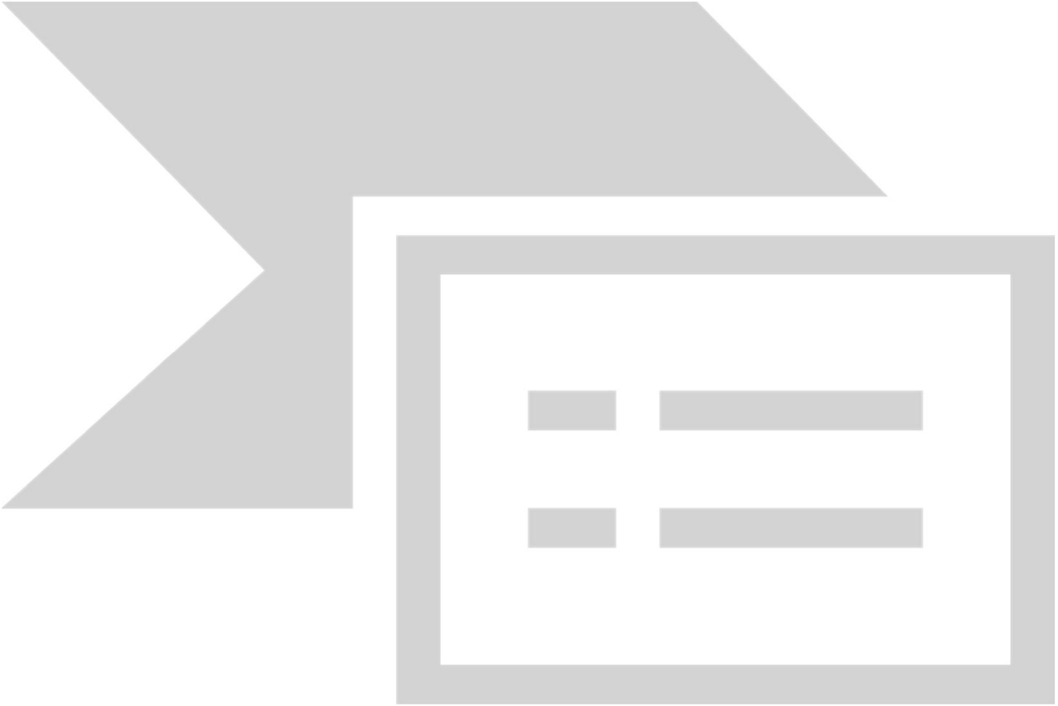
4.2 HARDWARE REQUIREMENTS:

- Processor : Pentium 3,Pentium 4 and higher
- RAM : 2GB/4GB RAM and higher
- Hard disk : 40GB and higher

Processor	**:**	**Pentium**
RAM	:	2GB
Hard disk	:	80GB

Fig: Hardware Components

5.SYSTEM ARCHITECTURE



6.SYSTEM DESIGN

The System Design Document describes the system requirements, operating environment, system and subsystem architecture, files and database design, input formats, output layouts, human-machine interfaces, detailed design, processing logic, and external interfaces.

This section describes the system in narrative form using non-technical terms. It should provide a high-level system architecture diagram showing a subsystem breakout of the system, if applicable. The high-level system architecture or subsystem diagrams should, if applicable, show interfaces to external systems. Supply a high-level context diagram for the system and subsystems, if applicable. Refer to the requirements trace ability matrix (RTM) in the Functional Requirements Document (FRD), to identify the allocation of the functional requirements into this design document.

This section describes any constraints in the system design (reference any trade-off analyses conducted such, as resource use versus productivity, or conflicts with other systems) and includes any assumptions made by the project team in developing the system design.
The organization code and title of the key points of contact (and alternates if appropriate) for the information system development effort. These points of contact should include the Project Manager, System Proponent, User Organization, Quality Assurance (QA) Manager, Security Manager, and Configuration Manager, as appropriate.

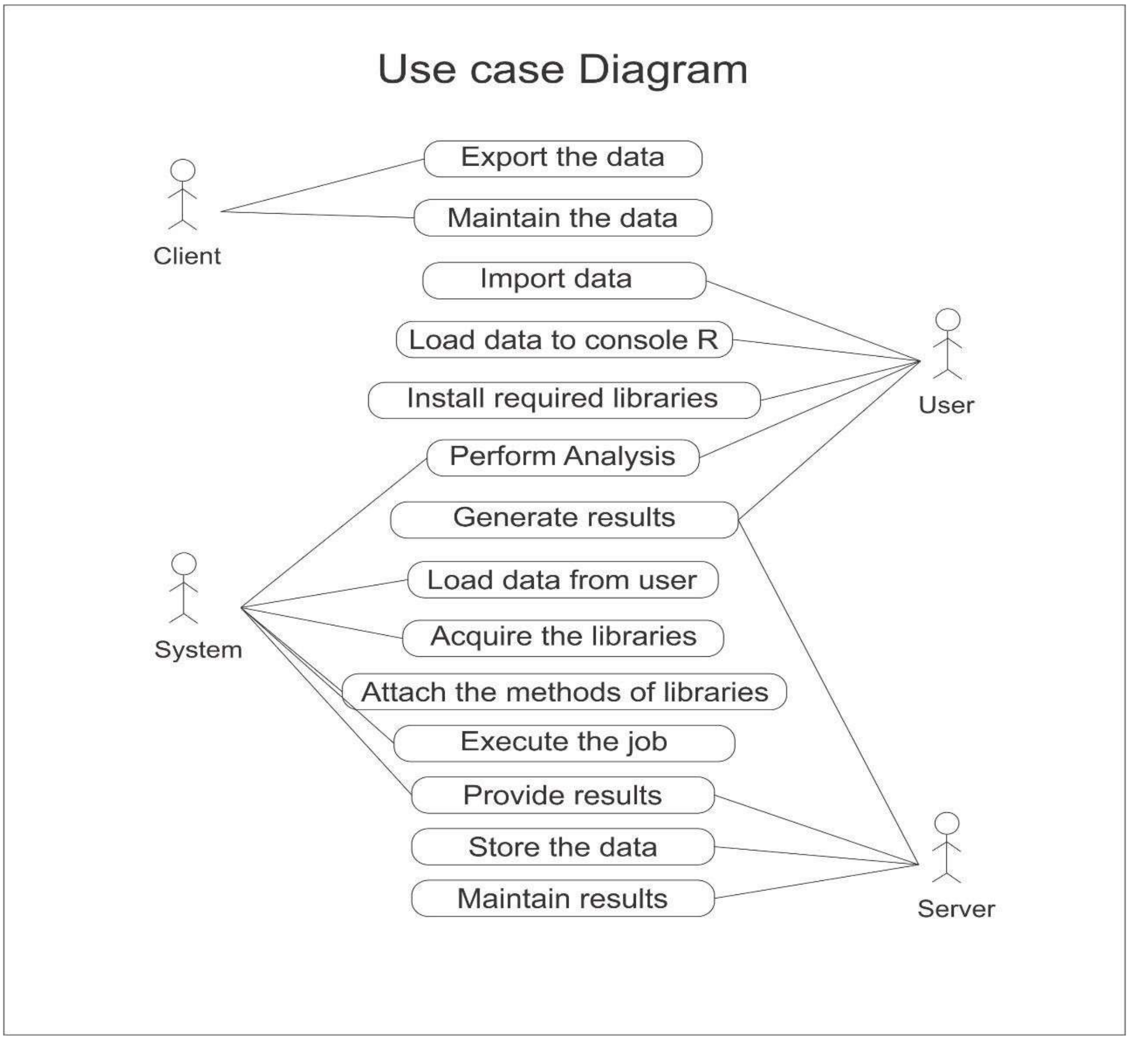
6.1 SYSTEM ARCHITECTURE:

6.2 UML Diagrams:

UML (Unified Modeling Language) is a standard vernacular for choosing, envisioning, making, and specifying the collectibles of programming structures. UML is a pictorial vernacular used to make programming blue prints. It is in like way used to exhibit non programming structures similarly like process stream in a gathering unit and so forth. UML is not a programming vernacular yet rather instruments can be utilized to make code in different tongues utilizing UML graphs. UML has an incite relationship with question composed examination and outline. UML expect a fundamental part in portraying trade viewpoints of a structure.

6.3 Use Case Diagram:

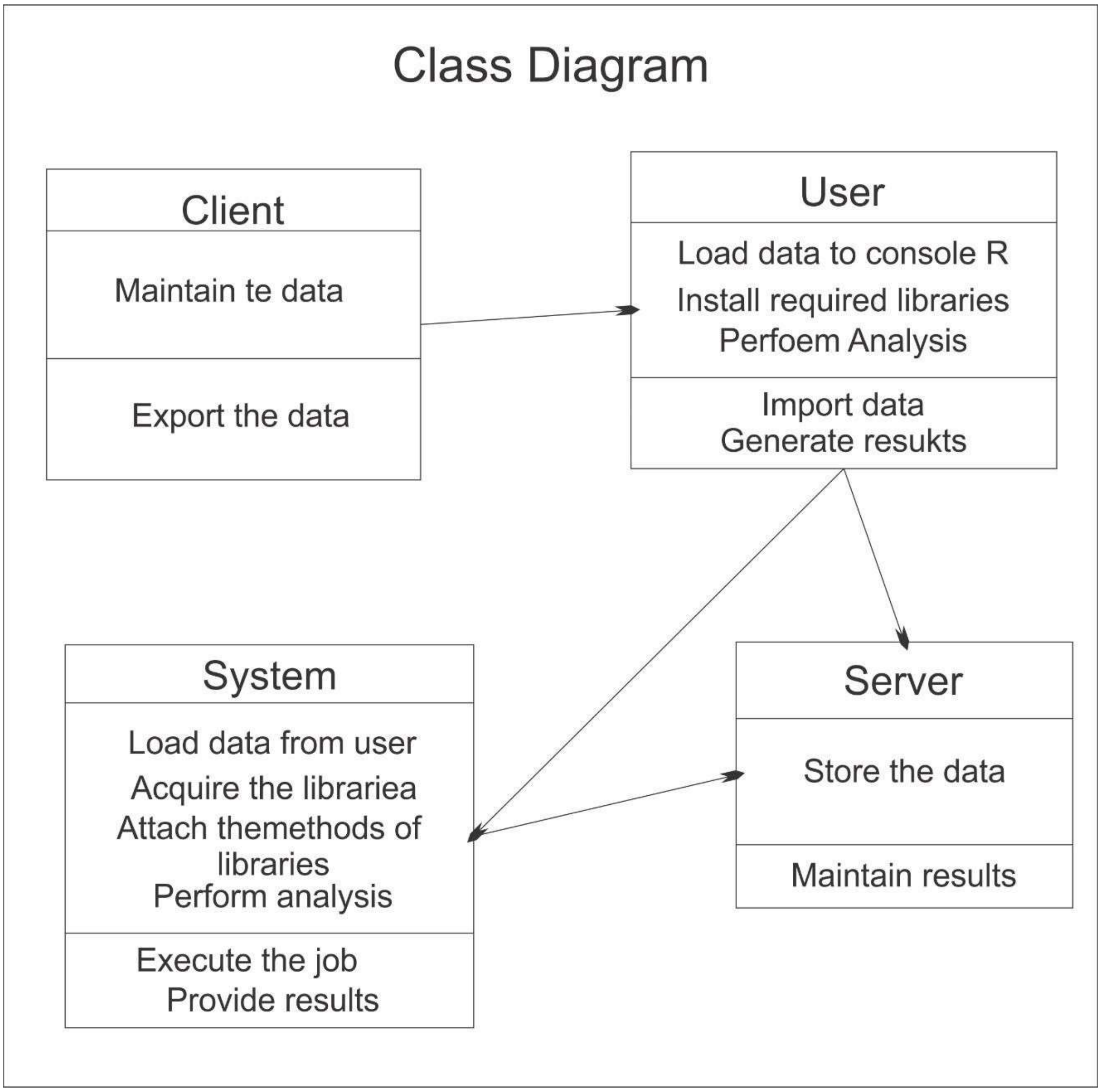
The use case graph is for demonstrating the direct of the structure. This chart contains the course of action of use cases, performing pros and their relationship. This chart might be utilized to address the static perspective of the structure.



In the above diagram, the performing specialists are customer, structure, client, server, Python and data cleaning. The client exchanges the data to the system which disengages the data into squares and gives the data to Python. By then Python does the data cleaning which is just performing data connection and data repairing, by then the results will be secured. These results can be seen using Python and can be secured in server for future reason. The gained results can be created as reports by the customer.

6.4 Class Diagram:

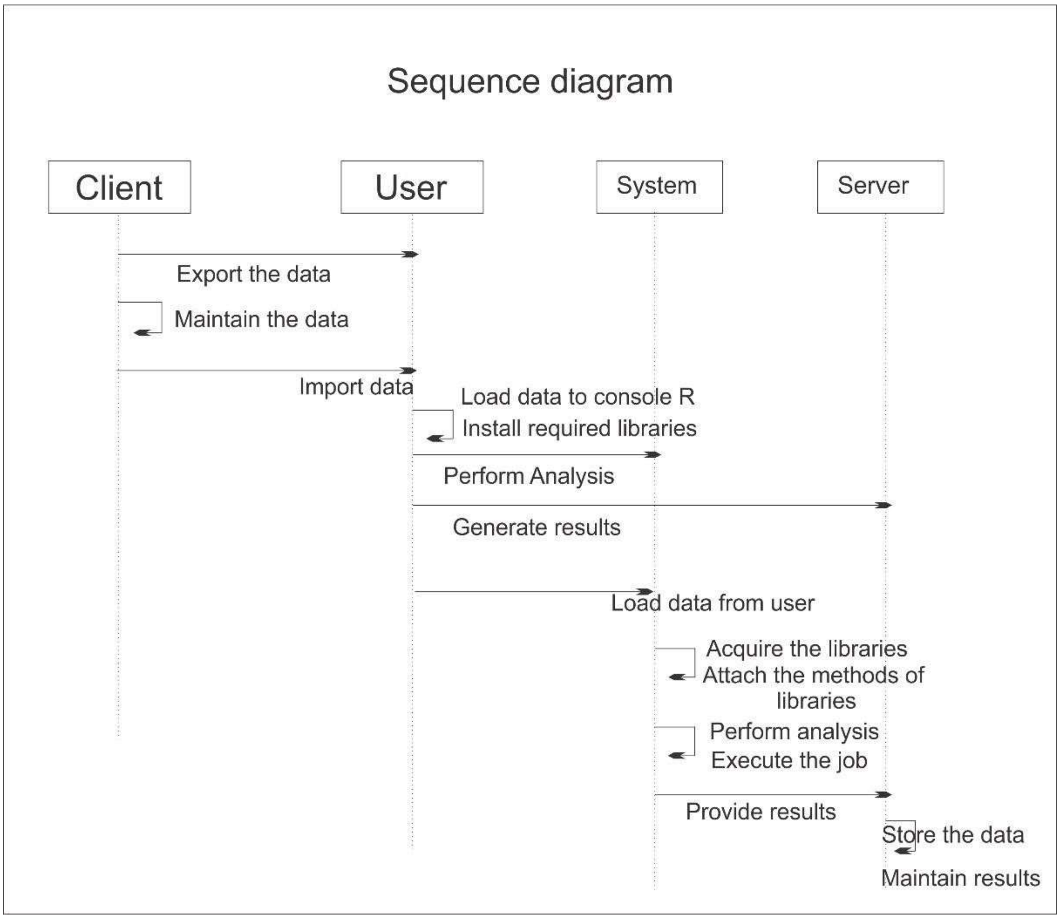
The class graph is the most normally pulled in layout UML. It addresses the static course of action perspective of the structure. It solidifies the strategy of classes, interfaces, joint attempts and their affiliations.



In the above class diagram, the relationship that is the dependence between each one of the classes is sketched out. Additionally, even the operations performed in each and every class is similarly appeared.

6.5 Sequence Diagram:

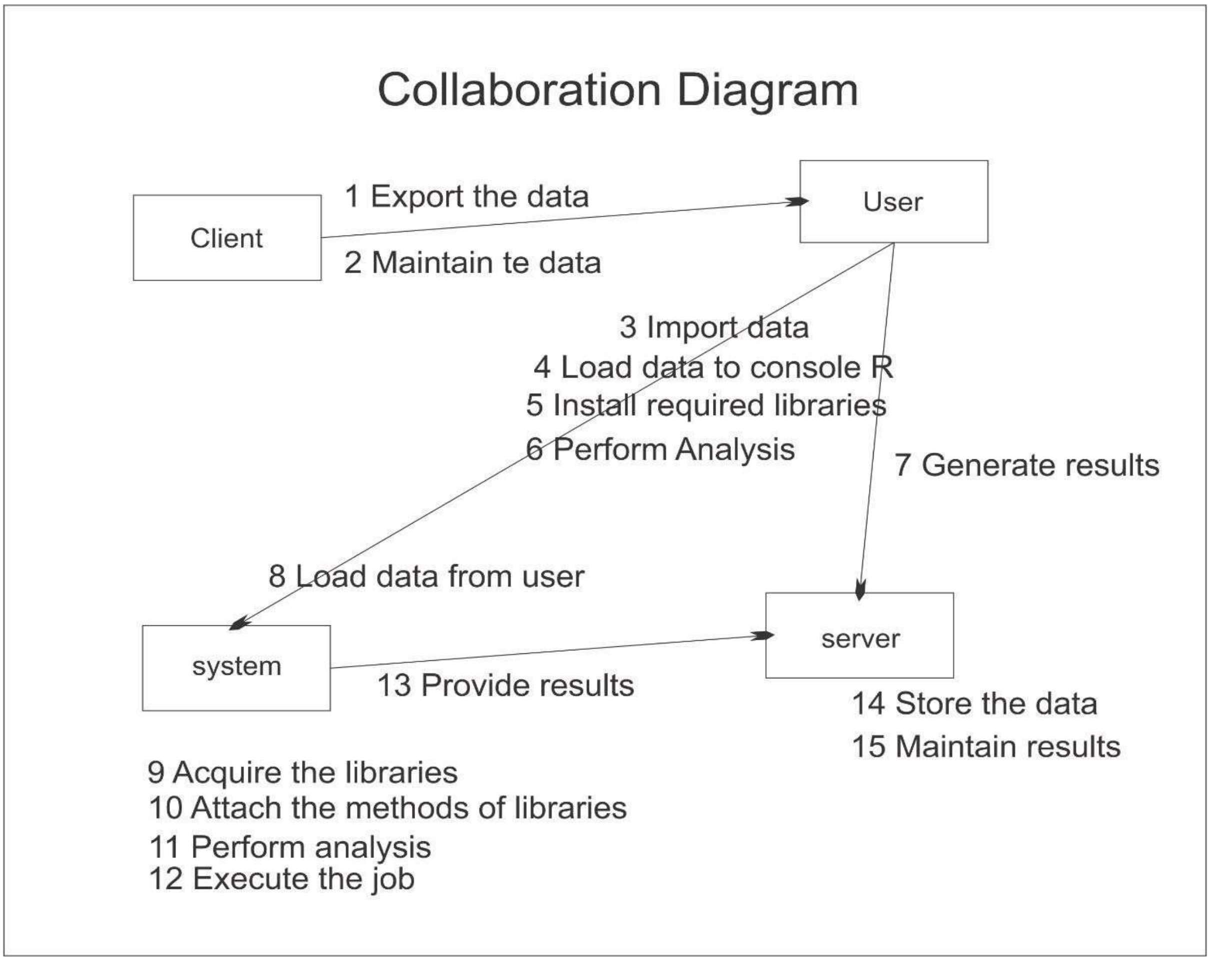
This is a cooperation design which tends to the time requesting of messages. It includes set of parts and the messages sent and gotten by the instance of parts. This chart is utilized to address the dynamic perspective of the structure.



A succession outline indicates question communications masterminded in time arrangement. In the above graph, there are five articles cooperating with each other. Each protest has a vertical dashed line which speaks to the presence of a question over some undefined time frame. This graph has additionally a tall, thin rectangle which is called center of control that demonstrates the timeframe amid which a protest is playing out an activity, either specifically or through a subordinate system.

6.6 Collaboration Diagram:

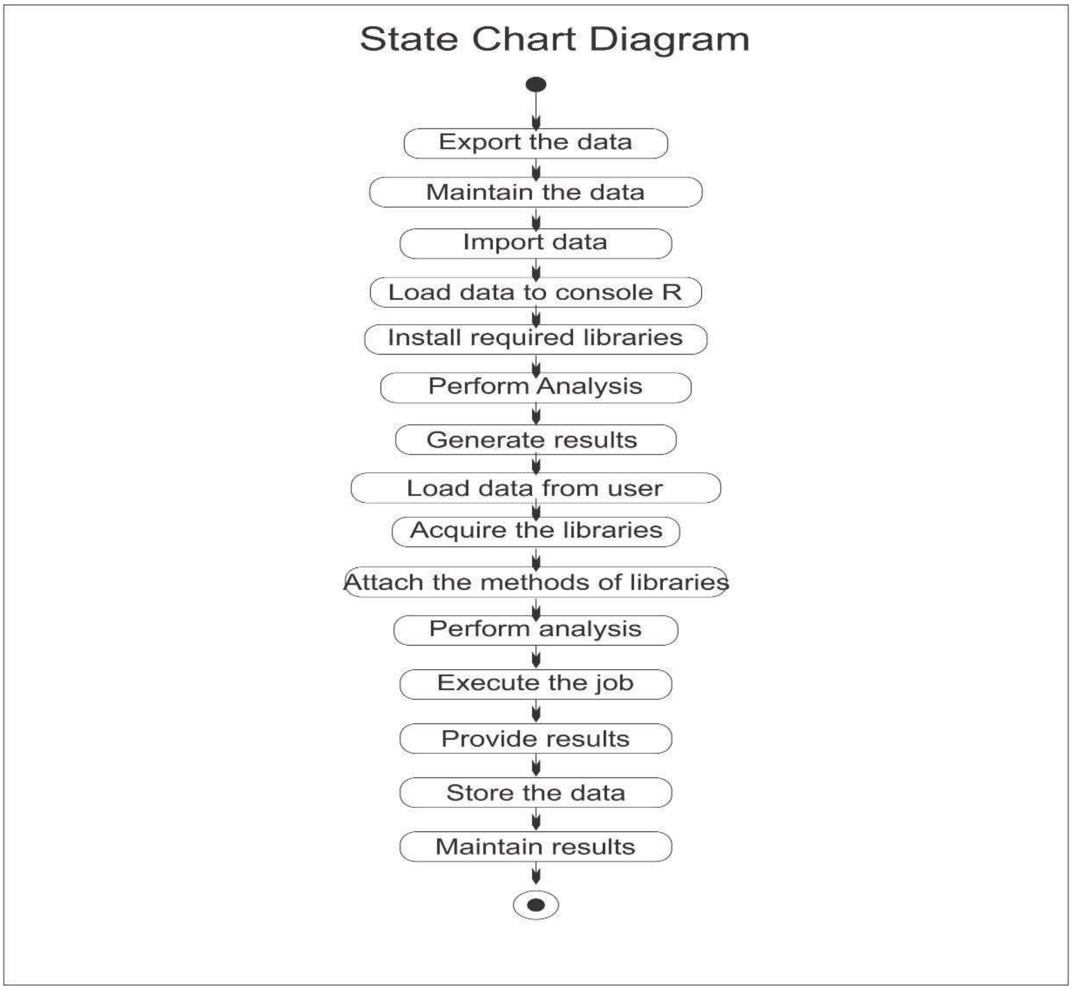
This is a support format, which tends to the principal relationship of articles that send and get messages. It incorporates set of parts, connectors that interface the parts and the messages sent and get by those parts. This graph is utilized to address the dynamic perspective of the framework.



The joint effort outline contains articles, way and arrangement number. In the above graph, there are five questions specifically customer, client, framework, Python and server. These items are connected to each other utilizing a way. A succession number show the time request of a message.

6.7 State Chart Diagram:

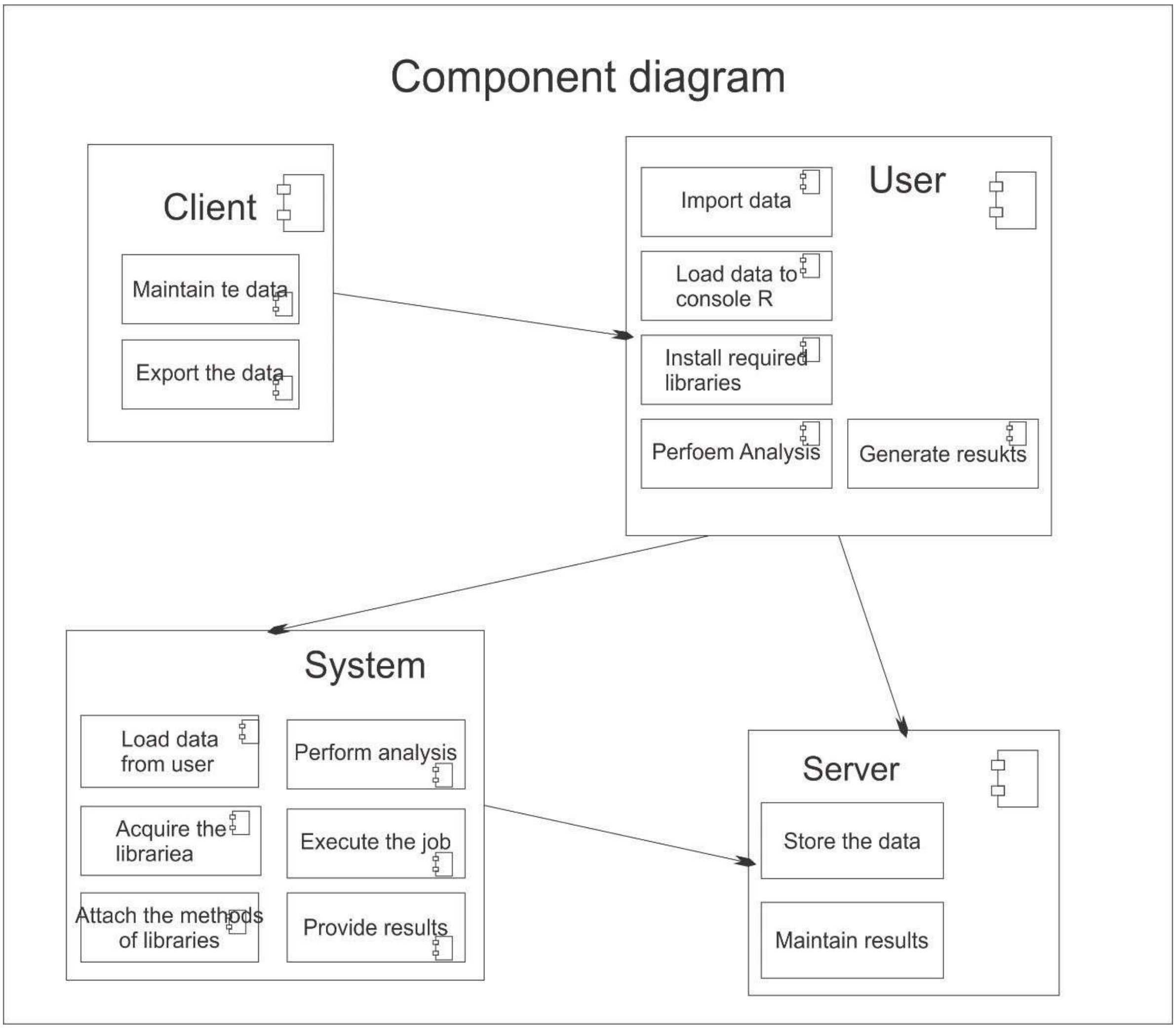
The state graph contains the game-plan of states, occasions and exercises. This graph is noteworthy for tending to the lead of the interface, class and made effort. The key centralization of state outline is to show the occasion sort out lead of the request. The state follows diagram the dynamic perspective of the framework.



A state outline graph contains two components called states and progress. States speak to circumstances amid the life of a question. We can without much of a stretch outline a state in Smart Draw by utilizing a rectangle with adjusted corners. Change is a strong bolt speaks to the way between various conditions of a question. Name the change with the occasion that activated it and the activity those outcomes from it.

6.8 Component Diagram:

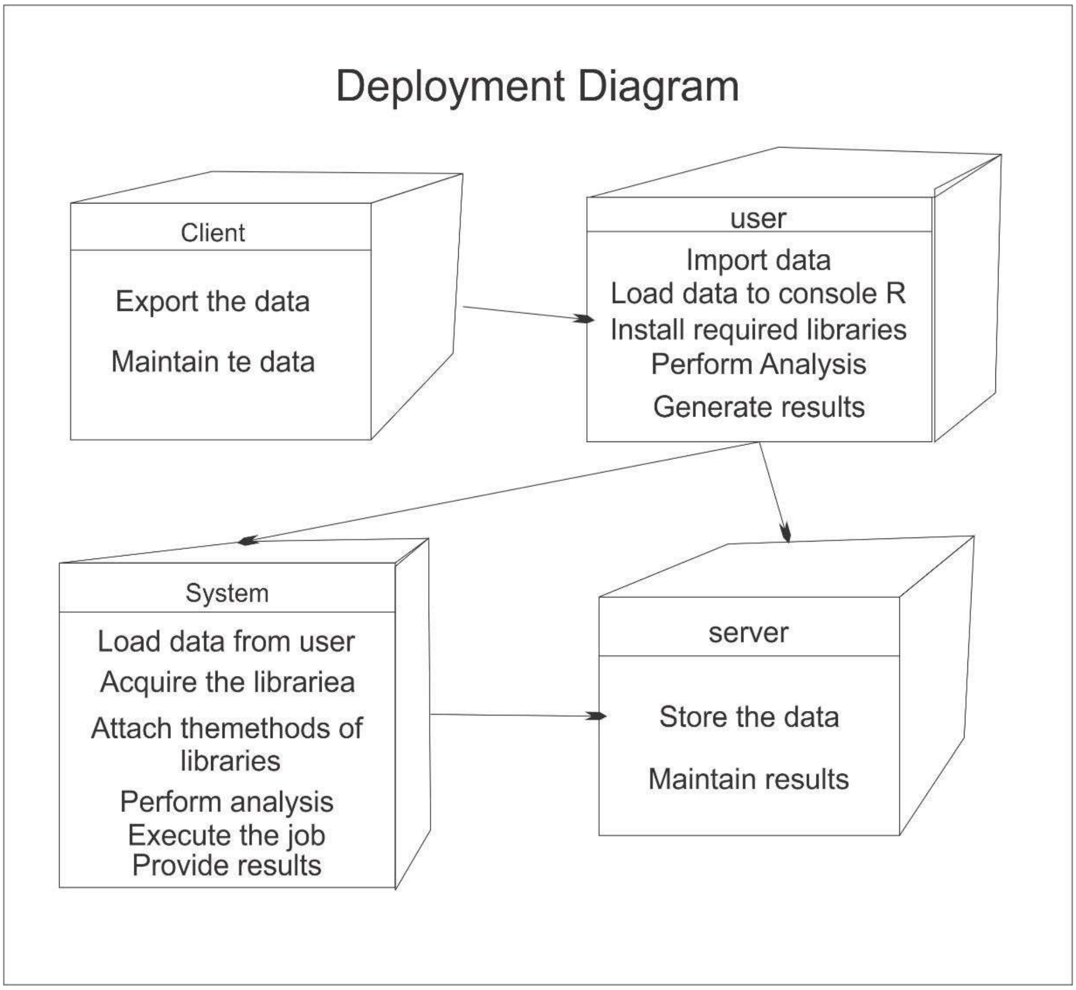
The imperative portion of part format is segment. This diagram demonstrates within parts, connectors and ports that understand the piece. Precisely when section is instantiated, duplicates of inside parts are besides instantiated.



A part outline is spoken to utilizing segment. A part is a physical building piece of the framework. It is spoken to as a rectangle with tab. Part outline portrays the inward handling of the venture. The information is sent to the Python where sqoop is utilized for information cleaning and the reports are produced utilizing hive.

6.9 Deployment Diagram:

The fundamental fragment in game-plan layout is a middle point. The strategy of focus focuses and their relationship with other is tended to utilizing sending plot. The sending outline is identified with the area diagram, that is one focus purpose obviously of activity format frequently includes no short of what one sections. This outline is in like way critical for tending to the static perspective of the framework.



An arrangement graph is spoken to utilizing hub. A hub is a physical asset that executes code parts. They are likewise used to portray run time handling of hubs. The information is sent to the Python where sqoop is utilized for information cleaning and the reports are produced utilizing hive.

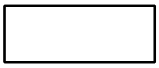
6.10 DATA FLOW DIAGRAMS:

An information stream design (DFD) is a graphical portrayal of the "stream" of information through a data framework, demonstrating its strategy edges. A DFD is a significant part of the time utilized as a preparatory stroll to make an overview of the framework, which can later be cleared up. DFDs can in like way be utilized for the depiction of information prepare. A DFD indicates what sort of data will be sense of duty regarding and yield from the structure, where the information will begin from and go to, and where the information will be secured. It doesn't demonstrate data about the organizing of process or data about whether strategy will work in game-plan or in parallel.

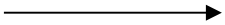
6.10.1 DFD Symbols:

In the DFD, there are four symbols

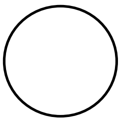
- A square defines a source or destination of system data.



- An arrow identifies data flow. It is the pipeline through which the information flows.



- A circle represents a process that transforms incoming data flow into outgoing data flow.



- An open rectangle is a data store, data at rest or a temporary repository of data.

6.10.1.1 Level 0: System input/ output level :

A level 0 DFD describes the system wide boundaries, dealing input to and output flow from the system and major processes.

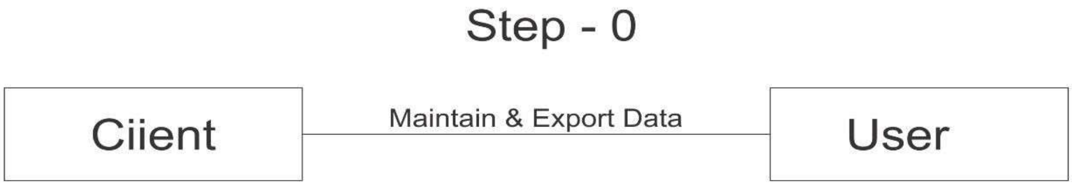


Fig: Level 0 DFD

DFD Level 0 is in like way called a Context Diagram. It's a urgent review of the entire structure or process being bankrupt down or appeared. It's required to be an at first watch, demonstrating the framework as a particular surprising state handle, with its relationship to outside substances.

6.10.1.2 Level 1: Sub system level data flow:

Level 1 DFD delineates the accompanying level of purposes of enthusiasm with the data stream between subsystems. The Level 1 DFD exhibits how the system is secluded into sub-structures (shapes), each of which oversees no less than one of the data streams to or from an outside pro, and which together give most of the helpfulness of the system as a rule.

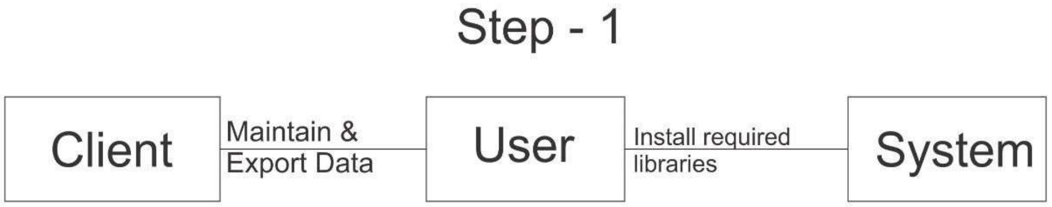


Fig: Level 1 DFD

6.10.1.3 Level 2: File level detail data flow :

Plausibility and danger examination are connected here from various perspectives. The level 2 DFD elucidates the fundamental level of understanding about the system's working.

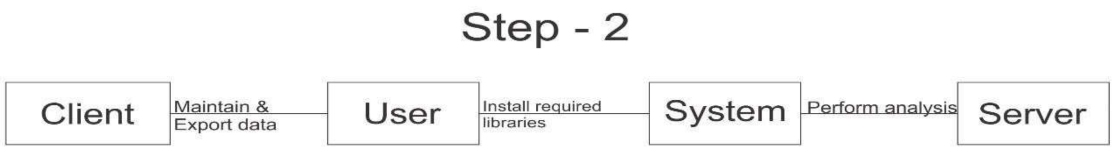


Fig: Level 2 DFD

6.10.1.4 Level 3:

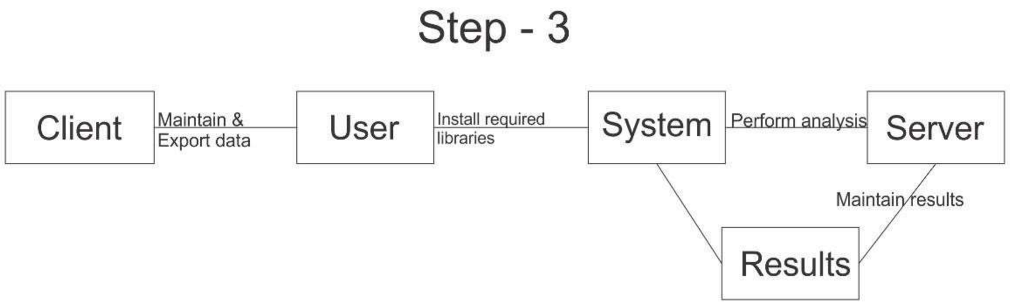


Fig: Level 3 DFD

7.IMPLEMENTATION MODULES

In Machine Learning, we are using semi-automated extraction of knowledge of data for identifying whether a loan would be approved or not . Classification could be a supervised learning within which the response is categorical that's its values area unit in finite unordered set. To easily the matter of classification, scikit learn are used. The praim primacy of this system is company need not has to maintain a ground team to validate and verify the customer records. They can easily check whether the loan has to be approved or not by this prediction model.

In this paper we try to develop user interface flexibly graphics concepts in mind, associated through a browser interface. Our goal is to implement machine learning model so as to classify, to the best potential degree of accuracy, master card fraud from a dataset gathered from Kaggle. once initial knowledge exploration, we have a tendency to knew we might implement a random forest model for best accuracy reports.

Random forest, as it was a good candidate for binary classification. Python sklearn library was used to implement the project, We used Kaggle datasets for Credit card fraud detection, using pandas to data frame for class ==0 for no fraud and class==1 for fraud, matplotlib for plotting the fraud and non fraud data, train_test_split for data extraction (Split arrays or matrices into random train and test subsets) and used Logistic Regression machine learning algorithm for fraud detection and print predicting score according to the algorithm. Finally C confusion matrix was plotted on true and predicted.

In this paper preprocessing is major part used sklearn method is MinMax scalar i.e., helps normalize the data. Model selection with help of cross validation, train/test split, kfold, GridSearchCV.

7.1 Model Selection:

Model selection is that method of selecting one in every of the models because the final model that addresses the issue. In there we have different steps. They are:

- Data filtering
- Data transformation
- Feature selection
- Feature engineering

For this process we have mainly two methods:

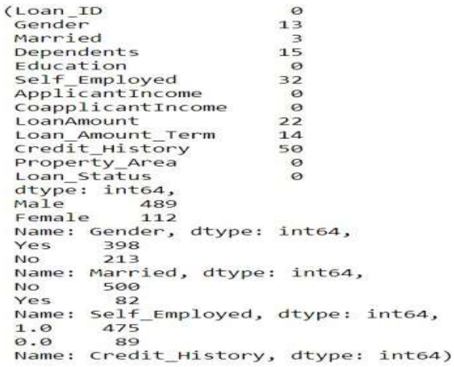
- Probabilistic model selection
- Resampling methods

In this paper we are using resampling methods such as cross validation, train/test split, Kflod, GridSearchCV .

7.2 PreProcessing:

Data mining methods are used in preprocessing for normalize the data which is collected from kaggle. There is a need to convert because dataset may have missing values, noisy data. So, we are using data mining method for cleaning method before using model selection process we are used preprocessing method for reduce the null values then recover the data with help of train/test split with help of MinMaxScalar.

MinMaxScalar, for each value in every feature MinMaxScalar cipher the minimum value within the feature then divided by the vary. The range is the distinction between the first most and original minimum. It preserves the shapes of the first original distribution.



```
(Loan_ID                 0
 Gender                 13
 Married                 3
 Dependents             15
 Education               0
 Self_Employed          32
 ApplicantIncome         0
 CoapplicantIncome       0
 LoanAmount             22
 Loan_Amount_Term       14
 Credit_History         50
 Property_Area           0
 Loan_Status             0
 dtype: int64,
 Male      489
 Female    112
 Name: Gender, dtype: int64,
 Yes    398
 No     213
 Name: Married, dtype: int64,
 No     500
 Yes     82
 Name: Self_Employed, dtype: int64,
 1.0    475
 0.0     89
 Name: Credit_History, dtype: int64)
```

7.3Feature Engineering:

It is the method of using domain data to extract options from data via data processing techniques. These features are wanted to improve the performance of machine learning algorithms. Feature engineering is thought-about as applied Machine learning itself. It is helping for import the models.

- import numpy as np
- import matplotlib.pyplot as plt

- import pandas as pd
- import seaborn as sn

7.4 Machine Learning Methods:

Machine learning is a subset of AI that trains machines with vast volumes of data to think and act like humans without being explicitly programmed. In this paper we are using supervised (Classification methods) methods Five machine learning classification models have been used for prediction of android applications. The models are available in python open source software. The brief details of each model are described below:

7.4.1 Decision Trees :

The basic algorithmic rule of call tree needs all attributes or options ought to be discredited. Feature choice relies on greatest info gain of options.

The data pictured in call tree will delineate within the kind of IF-THEN rules. This model is associate degree extension of C4.5 classification algorithms represented by Quinlan.

7.4.2 Random Forest :

Random forests are a classifying learning framework for characterization (and backslide) that work by building a very large number of Decision trees at planning time and yielding the class that's the mode of the classes surrender by individual trees.

7.4.3 Support Vector Machine :

Used SVM to build and train a model prepare a demonstrate utilizing human cell records, and classify cells to whether the tests are begin (mild state) or dangerous (evil state). Support vector machines are managed learning models that utilize affiliation R-learning calculation which analyze attributes and distinguished design information, utilized for application classification. SVM can beneficially perform a replace utilizing the kernel trick, verifiably mapping their inputs into high dimensional attribute spaces.

7.4.4 Logistic Regression :

Logistic regression is supervised learning classification algorithm (try to method connections and conditions between the target prediction output and input attributes) such that we are able to anticipate the yield values for new information based on those connections which it learned from the previous information sets .

7.4.5 K-nearest neighbor (KNN)

The KNN algorithm is a simple supervised machine learning algorithm that can be utilized to unravel both classification and replace issues. It is easy to implement and understand but significantly slows as the size of that data on use grows.

8.SAMPLE CODE

```
# CUSTOMER LOAN PREDICTION USING MACHINE LEARNING ALGORITHM
import pandas as pd
import numpy as np
import matplotlib as plt

df = pd.read_csv("import pandas as pd
import numpy as np
import matplotlib as plt

df = pd.read_csv("C:/Users/Shiva Reddy/Downloads/bhanu/Project14/data.csv")
df.head(10)
df.describe()
df['Property_Area'].value_counts()
import matplotlib.pyplot as plt
df['ApplicantIncome'].hist(bins=50)
plt.show()
df.boxplot(column='ApplicantIncome')
plt.show()
df.boxplot(column='ApplicantIncome', by = 'Education') plt.show()
df['LoanAmount'].hist(bins=50)
plt.show()
df.boxplot(column='LoanAmount')
plt.show()

temp1 = df['Credit_History'].value_counts(ascending=True)
temp2 = df.pivot_table(values='Loan_Status',index=['Credit_History'],aggfunc=lambda x:
x.map({'Y':1,'N':0}).mean())
print('Frequency Table for Credit
History:') print(temp1)

print('\nProbility of getting loan for each Credit History class:' )
print(temp2)
import matplotlib.pyplot as plt
fig = plt.figure(figsize=(8,4))
ax1 = fig.add_subplot(121)
ax1.set_xlabel('Credit_History')
ax1.set_ylabel('Count of Applicants')
ax1.set_title("Applicants by
Credit_History") temp1.plot(kind='bar')
plot.show()

ax2 = fig.add_subplot(122)
temp2.plot(kind = 'bar')
ax2.set_xlabel('Credit_History')
ax2.set_ylabel('Probability of getting loan')
ax2.set_title("Probability of getting loan by credit history")/data.csv")
df.head(10)
df.describe()
df['Property_Area'].value_counts()
import matplotlib.pyplot as plt
df['ApplicantIncome'].hist(bins=50)
plt.show()
df.boxplot(column='ApplicantIncome')
plt.show()
df.boxplot(column='ApplicantIncome', by = 'Education') plt.show()
df['LoanAmount'].hist(bins=50)
plt.show()
df.boxplot(column='LoanAmount')
plt.show()

temp1 = df['Credit_History'].value_counts(ascending=True)
temp2 = df.pivot_table(values='Loan_Status',index=['Credit_History'],aggfunc=lambda x:
x.map({'Y':1,'N':0}).mean())
print('Frequency Table for Credit History:') print(temp1)
print('\nProbility of getting loan for each Credit History class:' ) print(temp2)

import matplotlib.pyplot as plt
fig = plt.figure(figsize=(8,4))
ax1 = fig.add_subplot(121)
ax1.set_xlabel('Credit_History')
ax1.set_ylabel('Count of Applicants')
ax1.set_title("Applicants by
Credit_History")
temp1.plot(kind='bar') plt.show()
ax2            =
fig.add_subplot(122)
temp2.plot(kind = 'bar')
ax2.set_xlabel('Credit_History')
ax2.set_ylabel('Probability of getting loan')
ax2.set_title("Probability of getting loan by credit history")
```

9.SOFTWARE ENVIRONMENT

9.1 Big Data:

Bata data is a propelling term that depicts any voluminous measure of sorted out, semi-composed and unstructured data that can be burrowed for information. Though huge data doesn't suggest a specific sum, the term is much of the time used when discussing Petabytes and Exabyte's of data.

Bigdata is a term for informational collections that are so extensive or complex that customary information handling application programming is lacking to manage them. Gigantic data is used to depict a tremendous volume of data that is expansive to the point that it's difficult to process. The data is excessively colossal that outperforms current getting ready cutoff. Gigantic Data is an articulation used to mean an enormous volume of both sorted out and unstructured data that is so sweeping it is difficult to process using traditional database and programming frameworks. In most undertaking situations the volume of information is too enormous or it moves too quick or it surpasses current handling limit. Huge Data can possibly enable organizations to enhance operations and make speedier, more keen choices. This information, when caught, organized, controlled, put away, and investigated can enable an organization to increase helpful understanding to expand incomes, to get or hold clients, and enhance operations.

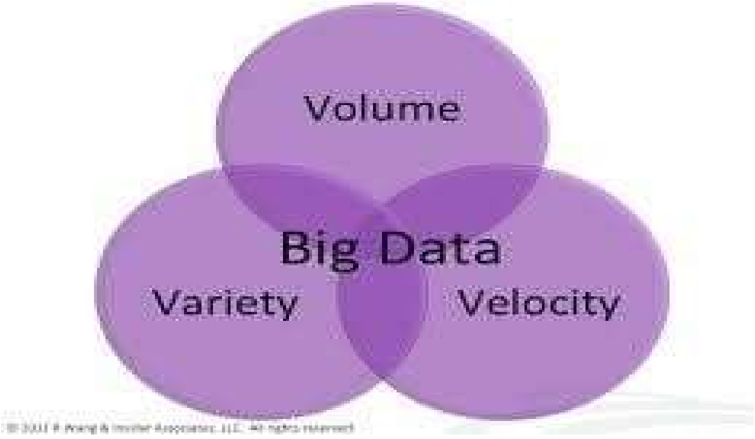


Fig: Big Data 3V's

Big data can be portrayed by 3Vs: the outrageous volume of information, the wide assortment of sorts of information and the speed at which the information must be must procedures

9.1.1 Volume:

Volume is the V most connected with huge information since, well, volume can be huge. Affiliations assemble data from an arrangement of sources, including business trades, web based systems administration and information from sensor or machine-to-machine data. Beforehand, securing it would've been an issue For example, facebook stores pictures of around 250 billions.

9.1.2 Velocity:

Speed is the measure of how quick the information is coming in. Information streams in at an extraordinary speed and should be managed in an opportune way. For instance, Facebook needs to deal with a torrent of photos consistently. It needs to ingest everything, process it, document it, and by one means or another, later, have the capacity to recover it.

9.1.3 Variety:

Data arrives in an extensive variety of associations – from sorted out, numeric data in standard databases to unstructured substance chronicles, email, video, sound, stock ticker data and cash related trades.

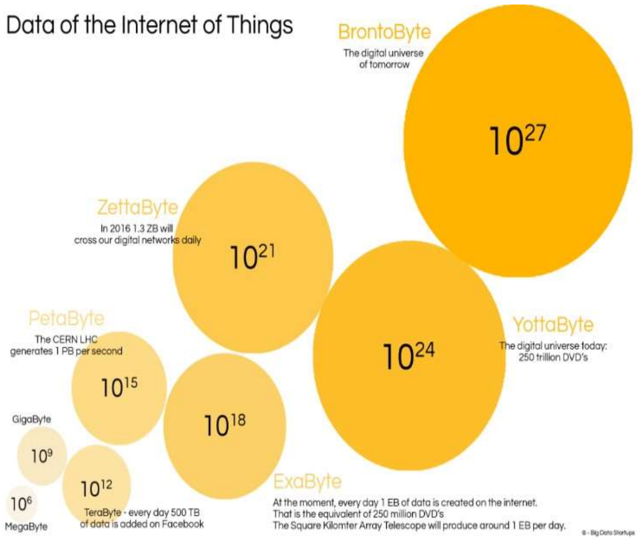


Fig : Data Measurements

9.2 About Python:

Python is a programming language, which means it'a a language both people and computers can understand. Python was developed by a Dutch software engineer named Guido van Rossum, who created the language to solve some problems he saw in computer languages of the time.

Python is an interpreted high-level programming language for general-purpose programming. Created by Guido van Rossum and first released in 1991, Python has a design philosophy that emphasizes code readability, and a syntax that allows programmers to express concepts in fewer lines of code, notably using significant whitespace. It provides constructs that enable clear programming on both small and large scales.

Python features a dynamic type system and automatic memory management. It supports multiple programming paradigms, including object-oriented, imperative, functional and procedural, and has a large and comprehensive standard library.

9.2.1 You Can Use Python for Pretty Much Anything:

One significant advantage of learning Python is that it's a general-purpose language that can be applied in a large variety of projects. Below are just some of the most common fields where Python has found its use:

- Data science
- Scientific and mathematical computing
- Web development
- Computer graphics

9.2.2 Python Is Widely Used in Data Science:

Python's ecosystem is growing over the years and it's more and more capable of the statistical analysis.

It's the best compromise between scale and sophistication (in terms od data processing. There are plenty of Python scientific packages for machine learning, natural language processing, complex data analysis and more. All of these factors make Python a great tool for scientific computing and a solid alternative for commercial packages such as MatLab. The most popular libraries and tools for data science are:

9.2.2.1 Pandas: a library for data manipulation and analysis. The library provides data structures and operations for manipulating numerical tables and time series.

9.2.2.2 NumPy: the fundamental package for scientific computing with Python, adding support for large, multi-dimensional arrays and matrices, along with a large library of highlevel mathematical functions to operate on these arrays.

9.2.2.3 SciPy: a library used by scientists, analysts, and engineers doing scientific computing and technical computing.

Being a free, cross-platform, general-purpose and high-level programming language, Python has been widely adopted by the scientific community. Scientists value Python for its precise and efficient syntax, relatively flat learning curve and the fact that it integrates well with other languages (e.g. C/C++).

As a result of this popularity there are plenty of Python scientific packages for data visualization, machine learning, natural language processing, complex data analysis and more. All of these factors make Python a great tool for scientific computing and a solid alternative for commercial packages such as MatLab.

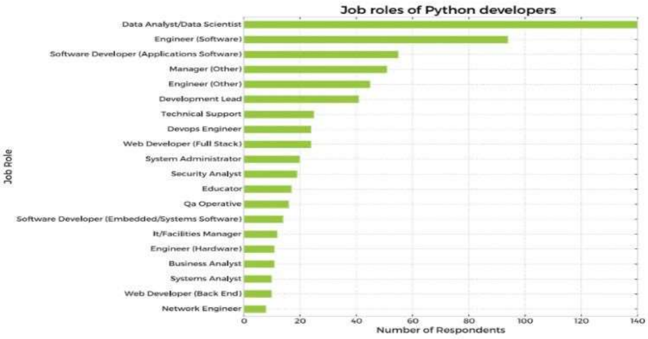


Fig: Graph

9.2.3 Here's our list of the most popular Python scientific libraries and tools:

9.2.3.1 Astropy:

The Astropy Project is a collection of packages designed for use in astronomy. The core astropy package contains functionality aimed at professional astronomers and astrophysicists, but may be useful to anyone developing astronomy software.

9.2.3.2 Biopython:

Biopython is a collection of non-commercial Python tools for computational biology and bioinformatics. It contains classes to represent biological sequences and sequence annotations, and it is able to read and write to a variety of file formats.

9.2.3.3 Cubes:

Cubes is a light-weight Python framework and set of tools for the development of reporting and analytical applications, Online Analytical Processing (OLAP), multidimensional analysis and browsing of aggregated data.

9.2.3.4 DEAP:

DEAP is an evolutionary computation framework for rapid prototyping and testing of ideas. It incorporates the data structures and tools required to implement most common evolutionary computation techniques such as genetic algorithm, genetic programming, evolution strategies, particle swarm optimization, differential evolution and estimation of distribution algorithm.

9.2.3.5 SCOOP:

SCOOP is a Python module for distributing concurrent parallel tasks on various environments, from heterogeneous grids of workstations to supercomputers.

9.2.3.6 PsychoPy:

PsychoPy is a package for the generation of experiments for neuroscience and experimental psychology. PsychoPy is designed to allow the presentation of stimuli and collection of data for a wide range of neuroscience, psychology and psychophysics experiments. **9.2.3.7 Pandas:**

Pandas is a library for data manipulation and analysis. The library provides data structures and operations for manipulating numerical tables and time series.

9.2.3.8 Mlpy:

Mlpy is a machine learning library built on top of NumPy/SciPy, the GNU Scientific Libraries. Mlpy provides a wide range of machine learning methods for supervised and unsupervised problems and it is aimed at finding a reasonable compromise between modularity, maintainability, reproducibility, usability and efficiency.

9.2.3.9 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 allows you to generate plots, histograms, power spectra, bar charts, errorcharts, scatterplots, and more.

9.2.3.10 NumPy:

NumPy is the fundamental package for scientific computing with Python, adding support for large, multi-dimensional arrays and matrices, along with a large library of high-level mathematical functions to operate on these arrays.

9.2.3.11 NetworkX:

NetworkX is a library for studying graphs which helps you create, manipulate, and study the structure, dynamics, and functions of complex networks.

9.2.3.12 TomoPy:

TomoPy is an open-sourced Python toolbox to perform tomographic data processing and image reconstruction tasks. TomoPy provides a collaborative framework for the analysis of synchrotron tomographic data with the goal to unify the effort of different facilities and beamlines performing similar tasks.

9.2.3.13 Theano:

Theano is a numerical computation Python library. Theano allows you to define, optimize, and evaluate mathematical expressions involving multi-dimensional arrays efficiently.

9.2.3.14 SymPy:

SymPy is a library for symbolic computation and includes features ranging from basic symbolic arithmetic to calculus, algebra, discrete mathematics and quantum physics. It provides computer algebra capabilities either as a standalone application, as a library to other applications, or live on the web.

9.2.3.15 SciPy:

SciPy is a library used by scientists, analysts, and engineers doing scientific computing and technical computing. SciPy contains modules for optimization, linear algebra, integration, interpolation, special functions, FFT, signal and image processing, ODE solvers and other tasks common in science and engineering.

9.2.3.16 Scikit-learn:

Scikit-learn is a machine learning library. It features various classification, regression and clustering algorithms including support vector machines, random forests, gradient boosting, k-means and DBSCAN, and is designed to interoperate with the Python numerical and scientific libraries NumPy and SciPy.

9.2.3.17 Scikit-image:

Scikit-image is a image processing library. It includes algorithms for segmentation, geometric transformations, color space manipulation, analysis, filtering, morphology, feature detection, and more.

9.2.3.18 ScientificPython:

ScientificPython is a collection of modules for scientific computing. It contains support for geometry, mathematical functions, statistics, physical units, IO, visualization, and parallelization.

9.2.3.19 SageMath:

SageMath is mathematical software with features covering many aspects of mathematics, including algebra, combinatorics, numerical mathematics, number theory, and calculus. SageMath uses the Python, supporting procedural, functional and object-oriented constructs.

9.2.3.20 Veusz:

Veusz is a scientific plotting and graphing package designed to produce publicationquality plots in popular vector formats, including PDF, PostScript and SVG.

9.2.3.21Graph-tool:

Graph-tool is a module for the manipulation and statistical analysis of graphs.

9.2.3.22 SunPy:

SunPy is a data-analysis environment specializing in providing the software necessary to analyze solar and heliospheric data in Python.

9.2.3.23 Bokeh:

Bokeh is a Python interactive visualization library that targets modern web browsers for presentation. Its goal is to provide elegant, concise construction of novel graphics in the style of D3.js, but also deliver this capability with high-performance interactivity over very large or streaming datasets. **9.2.3.24 TensorFlow:**

TensorFlow is an open source software library for machine learning across a range of tasks, developed by Google to meet their needs for systems capable of building and training neural networks to detect and decipher patterns and correlations, analogous to the learning and reasoning which humans use.

9.2.3.25 Nilearn:

Nilearn makes it easy to use many advanced machine learning, pattern recognition and multivariate statistical techniques on neuroimaging data for applications such as MVPA (Mutli-Voxel Pattern Analysis), decoding, predictive modelling, functional connectivity, brain parcellations, connectomes.

9.2.3.26 Dmelt:

DataMelt, or DMelt, is a software for numeric computation, statistics, analysis of large data volumes ("big data") and scientific visualization. The program can be used in many areas, such as natural sciences, engineering, modeling and analysis of financial markets. DMelt can be used with several scripting languages including Python/Jython, BeanShell, Groovy, Ruby, as well as with Java.

9.2.3.27 Python-weka-wrapper:

Weka is a suite of machine learning software written in Java, developed at the University of Waikato, New Zealand. It contains a collection of visualization tools and algorithms for data analysis and predictive modeling, together with graphical user interfaces for easy access to these functions..

9.2.3.28 Dask:

Dask is a flexible parallel computing library for analytic computing composed of two components: 1) dynamic task scheduling optimized for computation, optimized for interactive computational workloads, and 2) Big Data collections like parallel arrays, dataframes, and lists that extend common interfaces like NumPy, Pandas, or Python iterators to larger-thanmemory or distributed environments.

9.2.3.29 Python Saves Time:

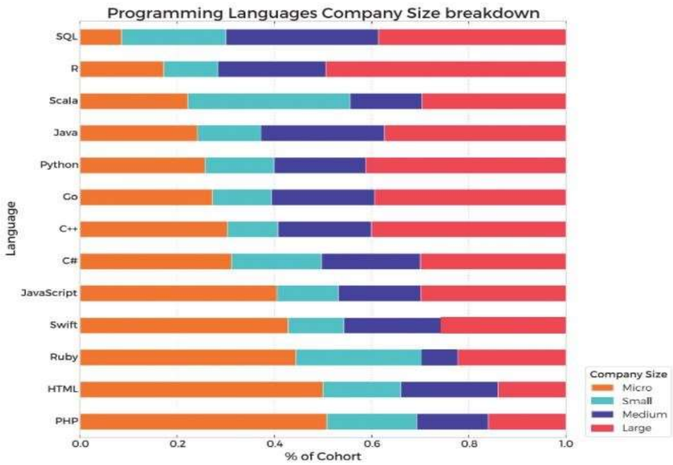
Even the classic “Hello, world” program illustrates this point:

```
print("Hello, world")
```

For comparison, this is what the same program looks like in Java: public

```
class HelloWorld {
   public static void main(String[] args) {
      System.out.println("Hello, world");
   }}
```

9.2.4 All the Big Names Use Python:



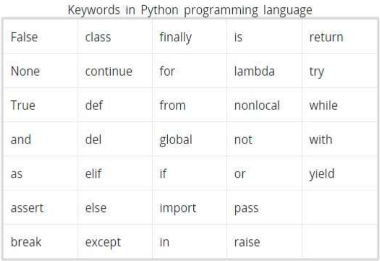
9.3 Python Keywords and Identifier:

Keywords are the reserved words in Python.

We cannot use a keyword as variable name, function name or any other identifier. They are used to define the syntax and structure of the Python language.

In Python, keywords are case sensitive.

There are 33 keywords in Python 3.3. This number can vary slightly in course of time. All the keywords except True, False and None are in lowercase and they must be written as it is. The list of all the keywords is given below.



Keywords in Python programming language

False	class	finally	is	return
None	continue	for	lambda	try
True	def	from	nonlocal	while
and	del	global	not	with
as	elif	if	or	yield
assert	else	import	pass	
break	except	in	raise	

Fig: KeyWords

Identifier is the name given to entities like class, functions, variables etc. in Python. It helps differentiating one entity from another.

9.3.1 Rules for writing identifiers:

Identifiers can be a combination of letters in lowercase (a to z) or uppercase (A to Z) or digits (0 to 9) or an underscore (_). Names like myClass, var_1 and print_this_to_screen, all are valid example.

An identifier cannot start with a digit. 1variable is invalid, but variable1 is perfectly fine. Keywords cannot be used as identifiers.

```
>>> global = 1
  File "<interactive input>", line 1     global = 1
          ^
```

SyntaxError: invalid syntax

We cannot use special symbols like !, @, #, $, % etc. in our identifier.

```
>>> a@ =
0
  File
"<interactive
input>", line
1
    a@
= 0

^
SyntaxError: invalid syntax
```

Identifier can be of any length.

9.4 Python

Python features a dynamic type system and automatic memory management. It supports multiple programming paradigms, including object-oriented, imperative, functional and procedural, and has a large and comprehensive standard library.

Python interpreters are available for many operating systems. C Python, the reference implementation of Python, is open source software and has a community-based development model, as do nearly all of its variant implementations. C Python is managed by the non-profit Python Software Foundation.

Fig: Python Logo

10.SYSTEM TESTING

10.1 Introduction To Testing:

Testing is a procedure, which uncovers blunders in the program. Programming testing is a basic component of programming quality affirmation and speaks to a definitive audit of determination, outline and coding. The expanding perceivability of programming as a framework component and chaperon costs related with a product disappointment are propelling variables for we arranged, through testing. Testing is the way toward executing a program with the plan of finding a mistake. The plan of tests for programming and other built items can be as trying as the underlying outline of the item itself It is the significant quality measure utilized amid programming improvement. Amid testing, the program is executed with an arrangement of experiments and the yield of the program for the experiments is assessed to decide whether the program is executing as it is relied upon to perform.

10.2 Testing Strategies:

A technique for programming testing coordinates the outline of programming experiments into an all around arranged arrangement of steps that outcome in fruitful improvement of the product. The procedure gives a guide that portrays the means to be taken, when, and how much exertion, time, and assets will be required. The procedure joins test arranging, experiment configuration, test execution, and test outcome gathering and assessment. The procedure gives direction to the specialist and an arrangement of points of reference for the chief. Due to time weights, advance must be quantifiable and issues must surface as ahead of schedule as would be prudent .

Keeping in mind the end goal to ensure that the framework does not have blunders, the distinctive levels of testing techniques that are connected at varying periods of programming improvement are:

10.3 Unit Testing:

Unit Testing is done on singular modules as they are finished and turned out to be executable. It is restricted just to the planner's prerequisites. It centers testing around the capacity or programming module. It Concentrates on the interior preparing rationale and information structures. It is rearranged when a module is composed with high union

- Reduces the quantity of experiments
- Allows mistakes to be all the more effectively anticipated and revealed**Black**

10.4 Black Box Testing:

It is otherwise called Functional testing. A product testing strategy whereby the inward workings of the thing being tried are not known by the analyzer. For instance, in a discovery test on a product outline the analyzer just knows the information sources and what the normal results ought to be and not how the program touches base at those yields. The analyzer does not ever inspect the programming code and does not require any further learning of the program other than its determinations. In this system some experiments are produced as information conditions that completely execute every single practical prerequisite for the program. This testing has been utilizations to discover mistakes in the accompanying classifications:

- Incorrect or missing capacities
- Interface blunders
- Errors in information structure or outside database get to
- Performance blunders
- Initialization and end blunders.

In this testing just the yield is checked for rightness.

10.5 White Box testing:

It is otherwise called Glass box, Structural, Clear box and Open box testing . A product testing procedure whereby express learning of the inner workings of the thing being tried are utilized to choose the test information. Not at all like discovery testing, white box testing utilizes particular learning of programming code to inspect yields. The test is precise just if the analyzer comprehends what the program should do. He or she would then be able to check whether the program veers from its expected objective. White box testing does not represent blunders caused by oversight, and all obvious code should likewise be discernable. For an entire programming examination, both white box and discovery tests are required.

In this the experiments are produced on the rationale of every module by drawing stream diagrams of that module and sensible choices are tried on every one of the cases. It has been utilizations to produce the experiments in the accompanying cases:

- Guarantee that every single free way have been Executed.

- Execute every single intelligent choice on their actual and false Sides.

10.6 Integration Testing:

Coordination testing guarantees that product and subsystems cooperate an entirety. It tests the interface of the considerable number of modules to ensure that the modules carry on legitimately when coordinated together. It is characterized as a deliberate procedure for developing the product engineering. In the meantime reconciliation is happening, lead tests to reveal blunders related with interfaces. Its Objective is to take unit tried modules and assemble a program structure in view of the recommended outline Two Approaches of Integration Testing

- Non-incremental Integration Testing
- Incremental Integration Testing

10.7 System Testing:

Framework testing includes in-house testing of the whole framework before conveyance to the client. Its point is to fulfill the client the framework meets all necessities of the customer's determinations. This testing assesses working of framework from client perspective, with the assistance of particular report. It doesn't require any inward learning of framework like plan or structure of code.

It contains utilitarian and non-useful zones of utilization/item. Framework Testing is known as a super arrangement of a wide range of testing as all the significant sorts of testing are shrouded in it. In spite of the fact that attention on sorts of testing may differ on the premise of item, association procedures, course of events and necessities. Framework Testing is the start of genuine testing where you test an item all in all and not a module/highlight.

10.8 Acceptance Testing:

Acknowledgment testing, a testing method performed to decide if the product framework has met the prerequisite particulars. The principle motivation behind this test is to assess the framework's consistence with the business necessities and check in the event that it is has met the required criteria for conveyance to end clients. It is a pre-conveyance testing in which whole framework is tried at customer's site on genuine information to discover blunders. The acknowledgment test bodies of evidence are executed against the test information or utilizing an acknowledgment test content and afterward the outcomes are contrasted and the normal ones.

The acknowledgment test exercises are completed in stages. Right off the bat, the essential tests are executed, and if the test outcomes are palatable then the execution of more intricate situations are done.

10.9 Test Approach:

A Test approach is the test system usage of a venture, characterizes how testing would be done. The decision of test methodologies or test technique is a standout amongst the most intense factor in the achievement of the test exertion and the precision of the test designs and gauges. Testing should be possible in two ways

- Bottom up approach
- Top down approach

10.9.1 Bottom up Approach:

Testing can be performed beginning from littlest and most reduced level modules and continuing each one in turn. In this approach testing is directed from sub module to primary module, if the fundamental module is not built up a transitory program called DRIVERS is utilized to recreate the principle module. At the point when base level modules are tried consideration swings to those on the following level that utilization the lower level ones they are tried exclusively and afterward connected with the already inspected bring down level modules

10.9.2 Top down Approach:

In this approach testing is directed from fundamental module to sub module. in the event that the sub module is not built up an impermanent program called STUB is utilized for mimic the sub module. This sort of testing begins from upper level modules. Since the nitty gritty exercises more often than not performed in the lower level schedules are not given stubs are composed. A stub is a module shell called by upper level module and that when achieved legitimately will restore a message to the calling module demonstrating that appropriate association happened.

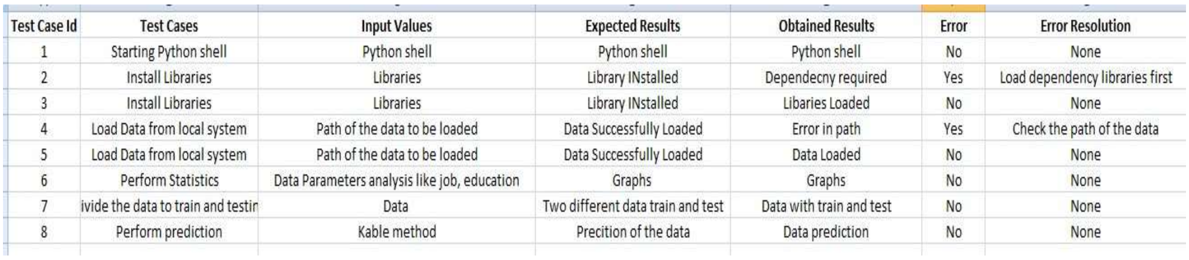
10.10 Validation:

The way toward assessing programming amid the improvement procedure or toward the finish of the advancement procedure to decide if it fulfills determined business prerequisites. Approval Testing guarantees that the item really addresses the customer's issues. It can likewise be characterized as to exhibit that the item satisfies its proposed utilize when sent on proper condition.

The framework has been tried and actualized effectively and along these lines guaranteed that every one of the prerequisites as recorded in the product necessities determination are totally satisfied.

10.11 Test Cases:

Experiments include an arrangement of steps, conditions and sources of info that can be utilized while performing testing undertakings. The principle expectation of this action is to guarantee whether a product passes or bombs as far as usefulness and different perspectives. The way toward creating experiments can likewise help discover issues in the prerequisites or plan of an application. Experiment goes about as the beginning stage for the test execution, and in the wake of applying an arrangement of information esteems, the application has a conclusive result and leaves the framework at some end point or otherwise called execution post condition.



Test Case Id	Test Cases	Input Values	Expected Results	Obtained Results	Error	Error Resolution
1	Starting Python shell	Python shell	Python shell	Python shell	No	None
2	Install Libraries	Libraries	Library INstalled	Dependecny required	Yes	Load dependency libraries first
3	Install Libraries	Libraries	Library INstalled	Libaries Loaded	No	None
4	Load Data from local system	Path of the data to be loaded	Data Successfully Loaded	Error in path	Yes	Check the path of the data
5	Load Data from local system	Path of the data to be loaded	Data Successfully Loaded	Data Loaded	No	None
6	Perform Statistics	Data Parameters analysis like job, education	Graphs	Graphs	No	None
7	ivide the data to train and testin	Data	Two different data train and test	Data with train and test	No	None
8	Perform prediction	Kable method	Precition of the data	Data prediction	No	None

Fig:Test Cases

11.SYSTEM STUDY

11.1 Feasibility Study:

A credibility contemplate expects to fair-mindedly and soundly uncover the qualities and inadequacies of a present business or proposed meander, openings and threats present in nature, the benefits required to bring through, and in the long run the prospects for advance. In its most clear terms, the two criteria to judge believability are incurred significant injury required and motivator to the fulfilled.

An inside and out arranged feasibility ponder should give a recorded establishment of the business or wander, a delineation of the thing or organization, accounting explanations, purposes of enthusiasm of the operations and organization, publicizing examination and game plans, budgetary data, authentic necessities and cost duties. All things considered, plausibility looks at go before specific change and wander utilization. There are three sorts of attainability

- Economical Feasibility
- Technical Feasibility
- Operational Feasibility

11.2 Economical feasibility:

The electronic structure manages the present existing system's data stream and technique absolutely and should make each one of the reports of the manual structure other than a substantial gathering of the other organization reports. It should be filled in as an electronic application with specific web server and database server. Advance a segment of the associated trades happen in different ranges. Open source programming like TOMCAT, JAVA, MySQL and Linux is used to restrict the cost for the Customer. No extraordinary wander need to manage the instrument.

11.3 Technical feasibility:

Surveying the particular probability is the trickiest bit of a believability consider. This is in light of the fact that, starting at the present moment, not a lot of point by point layout of the system, making it difficult to get to issues like execution, costs on (by excellence of the kind of development to be passed on) et cetera.

Different issues must be considered while doing a particular examination. Grasp the differing progressions required in the proposed system. Before starting the wander, we should be clear about what are the advances that are to be required for the change of the new system. Check whether the affiliation by and by has the required advancements. Is the required development open with the affiliation?

In case so is the utmost sufficient?

For instance – "Will the present printer have the ability to manage the new reports and structures required for the new system?"

11.4 Operational feasibility:

Proposed wanders are profitable just if they can be changed into information systems that will meet the affiliations working necessities. Simply communicated, this trial of probability asks with reference to whether the structure will work when it is made and presented. Are there genuine obstacles to Implementation? Here are questions that will help test the operational achievability of a wander.

- Is there sufficient help for the wander from organization from customers? In case the present structure is particularly cherished and used to the extent that individuals won't have the ability to see purposes behind change, there may be resistance.
- Are the present business methodologies qualified to the customer? If they are not, Users may welcome a change that will accomplish a more operational and supportive systems.
- Have the customer been locked in with the orchestrating and change of the wander? Early commitment decreases the chances of impenetrability to the structure.

12.CONCLUSION

In this study, a new method for recognizing sentiment in iris has been proposed. The analysis shows that overall sentiment (both in iris and text) is governed by little sentiment bearing terms. In order to exploit this fact, a new method that uses Keyword Spotting (KWS) to search for sentiment bearing terms in iris has been proposed. By focusing on the terms that impact decision and ignoring non-sentiment bearing words/phrases, the overall system is more immune to speech recognition errors. Additionally, a new method to create the sentiment bearing keyword list for KWS has also been proposed.

12.1 Future Enhancement:

Two of the three species were gathered in the Gaspé Peninsula "all from a similar field, and singled out that day and estimated in the meantime by a similar individual with a similar mechanical assembly.

13.BIBILOGRAPHY

REFERENCES:

[1] S. Johnson, ”Internet changes everything: Revolutionizing public participation and access to government information through the Internet”, Administrative Law Review, Vol. 50, No. 2 (Spring 1998) , pp. 277-337

[2] D. Chrysanthos. ”Strategic manipulation of internet opinion forums: Implications for consumers and firms.” Management Science 52.10 (2006): 1577-1593.

[3] M. Wollmer, et al. ”Youtube movie reviews: Sentiment analysis in an audio-visual context.” Intelligent Systems, IEEE (2013): pages 46-53.

[4] J. Naughton, ”The internet: is it changing the way we think?”, The Gaurdian, Saturday 14 August 2010

[5] G. Mishne and N. S. Glance, ”Predicting movie sales from blogger sentiment,” in AAAI 2006 Spring Symposium on Computational Approaches to Analyzing Weblogs, 2006.

[6] L. Barbosa, and J. Feng, ”Robust sentiment detection on twitter from biased and noisy data.”, in Proceedings of the International Conference on Computational Linguistics (COLING-2010). 2010.

[7] E. Cambria, N. Howard, Y. Xia, and T. S. Chua, ”Computational Intelligence for Big Social Data Analysis”, IEEE Computational Intelligence Magazine, 11(3), 8-9, 2016.

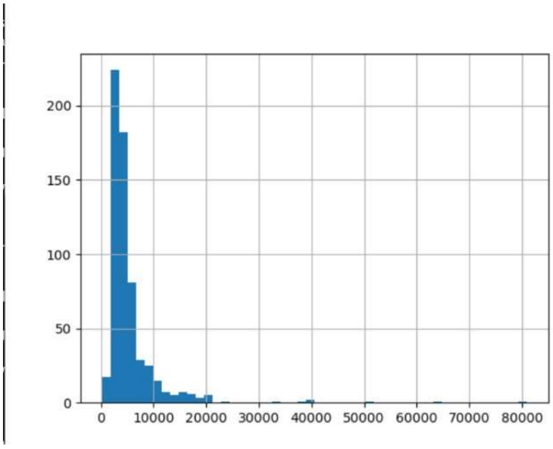
[8] E. Cambria, B. Schuller, Y. Xia, and B. White, ”New avenues in knowledge bases for natural language processing”, Knowledge-Based Systems, 108(C), 1-4, 2016.

[9] M. Bautin, L. Vijayarenu, and S. Skiena. ”International sentiment analysis for news and blogs.”, in Proceedings of the International AAAI Conference on Weblogs and Social Media (ICWSM-2008). 2008.

[10] I. Becker and V. Aharonson. ”Last but definitely not least: on the role of the last sentence in automatic polarity-classification.”, in Proceedings of the ACL 2010 Conference Short Papers. 2010.

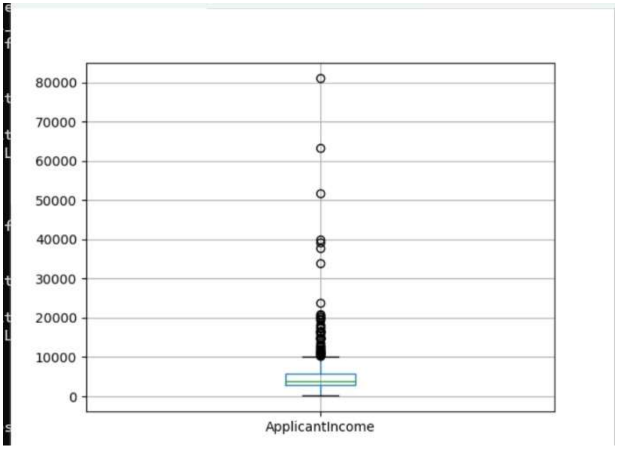
14.SCREENSHOTS

Prediction Graph



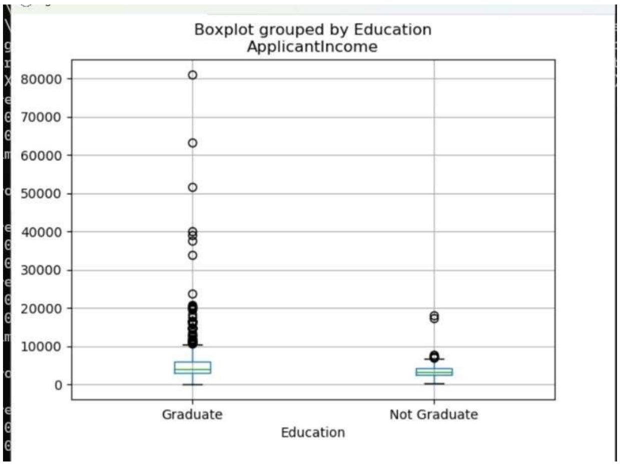
Fig

Application Income

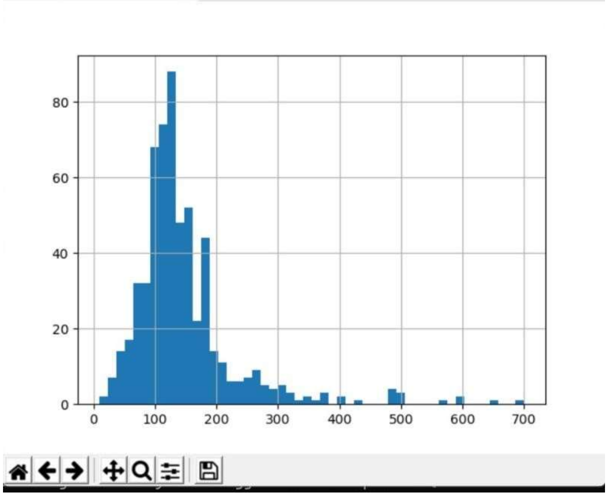


Fig

Box Plot Grounded by Education Application Income

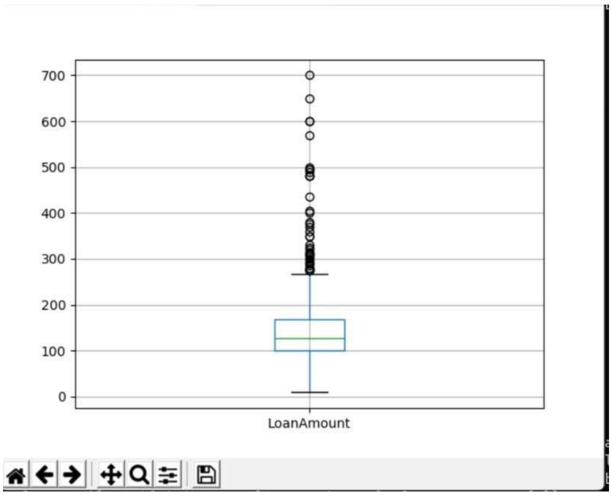


Prediction Graph After Application



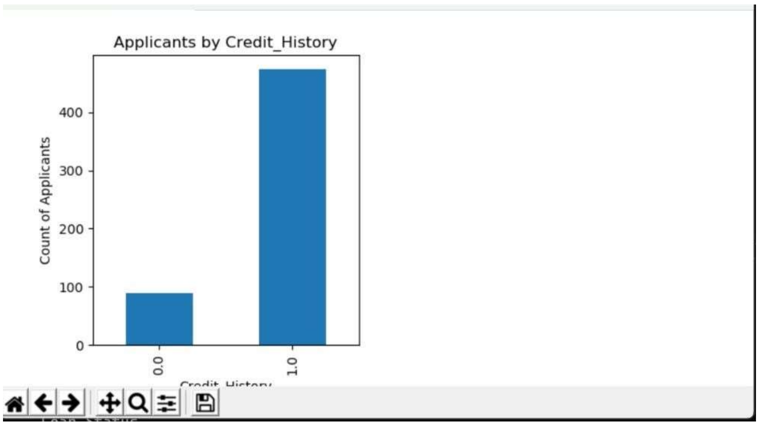
Fig

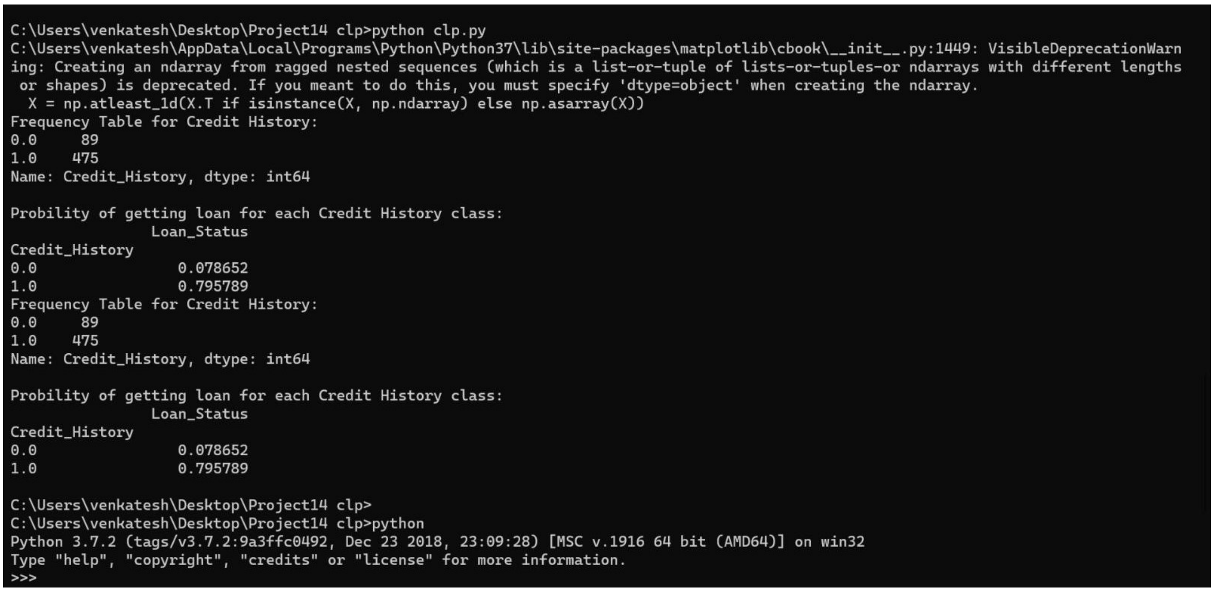
Loan Amount



Fig

Applicants by Credit_history





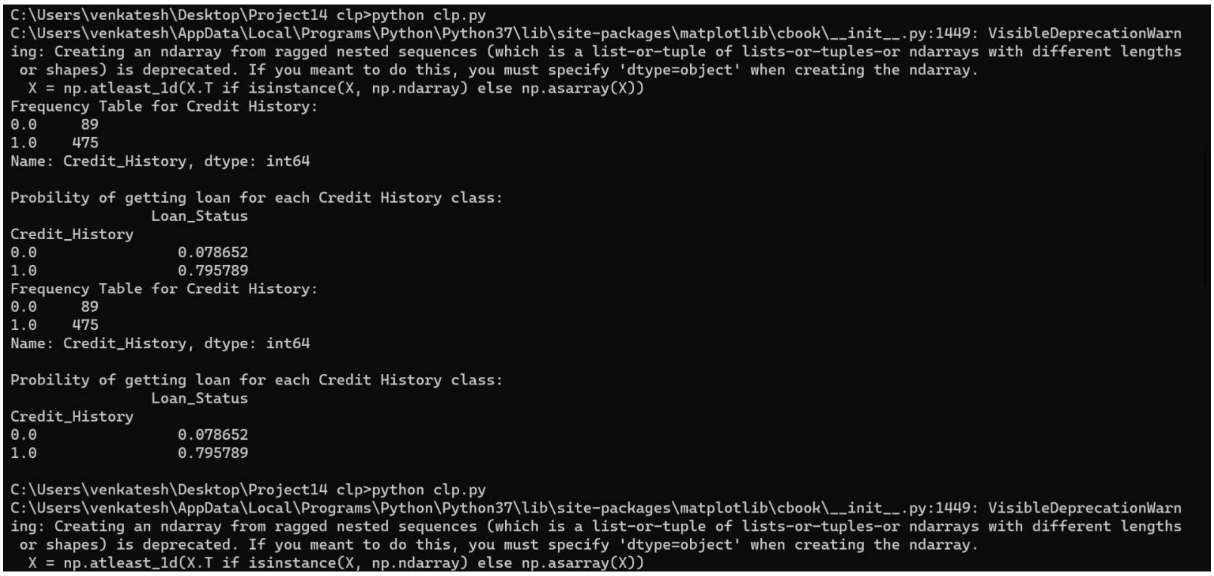
```
C:\Users\venkatesh\Desktop\Project14 clp>python clp.py
C:\Users\venkatesh\AppData\Local\Programs\Python\Python37\lib\site-packages\matplotlib\cbook\__init__.py:1449: VisibleDeprecationWarn
ing: Creating an ndarray from ragged nested sequences (which is a list-or-tuple of lists-or-tuples-or ndarrays with different lengths
 or shapes) is deprecated. If you meant to do this, you must specify 'dtype=object' when creating the ndarray.
  X = np.atleast_1d(X.T if isinstance(X, np.ndarray) else np.asarray(X))
Frequency Table for Credit History:
0.0     89
1.0    475
Name: Credit_History, dtype: int64

Probility of getting loan for each Credit History class:
                Loan_Status
Credit_History
0.0                0.078652
1.0                0.795789
Frequency Table for Credit History:
0.0     89
1.0    475
Name: Credit_History, dtype: int64

Probility of getting loan for each Credit History class:
                Loan_Status
Credit_History
0.0                0.078652
1.0                0.795789

C:\Users\venkatesh\Desktop\Project14 clp>
C:\Users\venkatesh\Desktop\Project14 clp>python
Python 3.7.2 (tags/v3.7.2:9a3ffc0492, Dec 23 2018, 23:09:28) [MSC v.1916 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

Fig



```
C:\Users\venkatesh\Desktop\Project14 clp>python clp.py
C:\Users\venkatesh\AppData\Local\Programs\Python\Python37\lib\site-packages\matplotlib\cbook\__init__.py:1449: VisibleDeprecationWarn
ing: Creating an ndarray from ragged nested sequences (which is a list-or-tuple of lists-or-tuples-or ndarrays with different lengths
 or shapes) is deprecated. If you meant to do this, you must specify 'dtype=object' when creating the ndarray.
  X = np.atleast_1d(X.T if isinstance(X, np.ndarray) else np.asarray(X))
Frequency Table for Credit History:
0.0     89
1.0    475
Name: Credit_History, dtype: int64

Probility of getting loan for each Credit History class:
                Loan_Status
Credit_History
0.0                0.078652
1.0                0.795789
Frequency Table for Credit History:
0.0     89
1.0    475
Name: Credit_History, dtype: int64

Probility of getting loan for each Credit History class:
                Loan_Status
Credit_History
0.0                0.078652
1.0                0.795789

C:\Users\venkatesh\Desktop\Project14 clp>python clp.py
C:\Users\venkatesh\AppData\Local\Programs\Python\Python37\lib\site-packages\matplotlib\cbook\__init__.py:1449: VisibleDeprecationWarn
ing: Creating an ndarray from ragged nested sequences (which is a list-or-tuple of lists-or-tuples-or ndarrays with different lengths
 or shapes) is deprecated. If you meant to do this, you must specify 'dtype=object' when creating the ndarray.
  X = np.atleast_1d(X.T if isinstance(X, np.ndarray) else np.asarray(X))
```

Fig