A Project Report on

Loan Prediction Using ML

Project

Submitted in partial fulfillment of the requirements for
the award of the degree of

Master of Science in Data Science and Big Data Analytics

in

Data Science and Big Data Analytics

by

Devang Vasant Vasava

41680

Under the Guidance of

Prof. Esmita Gupta

DEPARTMENT OF INFORMATION TECHNOLOGY
B. K. Birla College of Arts, Science and Commerce (Autonomous), Kalyan
B. K. Birla College Road, Near RTO, Kalyan

UNIVERSITY OF MUMBAI

ACADEMIC YEAR 2021-2022

Acknowledgement

This Project Report entitled ***"Loan Prediction Using ML Project"*** Submitted by **"Devang Vasant Vasava*" (41680)*** is approved for the partial fulfillment of the requirement for the award of the degreeof ***Master of Science*** in **Data Science and Big Data Analytics** from ***University of Mumbai.***

Prof. Esmita Gupta
Co-Guide Guide

Prof. Esmita Gupta
Head, Department of Information Technology

Place: B. K. Birla College (Autonomous), Kalyan
Date:

CERTIFICATE

This is to certify that the project entitled ***"Loan Prediction Using ML Project"*** submitted by **"Devang Vasant Vasava"** ***(41680)*** for the partial fulfillment of the requirement for award of a degree ***Masterof Science*** in ***Data Science and Big Data Analytics***, to the University of Mumbai, is a bonafide workcarried out during academic year 2021-2022.

Prof. Esmita Gupta
Guide

Co-Guide

Prof. Esmita Gupta
Head Department of IT

Dr. Avinash Patil
Principal

External Examiner(s)

1.

2.

Place: B. K. Birla College (Autonomous), Kalyan
Date:

Declaration

I declare that this written submission represents my ideas in my own words and where others' ideasor words have been included, I have adequately cited and referenced the original sources. I also declare that I have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in my submission. I understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

_____

(Signature)

Devang Vasant Vasava, student Id: - 41680

Date:

ABSTRACT

The cost of assets is increasing day by day and the capital required to purchase an entire asset is very high. So purchasing it out of your savings is not possible. The easiest way to get the required funds is to apply for a loan. But taking a loan is a very time consuming process. The application has to go through a lot of stages and it's still not necessary that it will be approved. To decrease the approval time and to decrease the risk associated with the loan many loan prediction models were introduced. The aim of this project was to compare the various Loan Prediction Models and show which is the best one with the least amount of error and could be used by banks in real world to predict if the loan should be approved or not taking the risk factor in mind. After comparing and analysing the models, it was found that the prediction model based on Random Forest proved to be the most accurate and fitting of them all. This can be useful in reducing the time and manpower required to approve loans and filter out the perfect candidates for providing loans.

In our banking system, banks have many products to sell but main source of income of any banks is on its credit line. So they can earn from interest of those loans which they credits. A bank's profit or a loss depends to a large extent on loans i.e. whether the customers are paying back the loan or defaulting. By predicting the loan defaulters, the bank can reduce its Non-performing Assets. This makes the study of this phenomenon very important. Previous research in this era has shown that there are so many methods to study the problem of controlling loan default. But as the right predictions are very important for the maximization of profits, it is essential to study the nature of the different methods and their comparison. A very important approach in predictive analytics is used to study the problem of predicting loan defaulters (i) Collection of Data, (ii) Data Cleaning and (iii) Performance Evaluation. Experimental tests found that the Naïve Bayes model has better performance than other models in terms of loan forecasting.

INDEX

TABLE OF CONTENTS

CHAPTER 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. Dream housing 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, 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. So the 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. To predict loan safety, the SVM and Naïve bayes algorithm are used. First the data is cleaned so as to avoid the missing values in the data set**.**

In the present scenario, a loan needs to be approved manually by a representative of the bank which means that person will be responsible for whether the person is eligible for the loan or not and also calculating the risk associated with it. As it is done by a human it is a time consuming process and is susceptible to errors. If the loan is not repaid, then it accounts as a loss to the bank and banks earn most of their profits by the interest paid to them. If the banks lose too much money, then it will result in a banking crisis. These banking crisis affects the economy of the country. So it is very important that the loan should be approved with the least amount of error in risk calculation while taking up as the least time possible. So a loan prediction model is required that can predict quickly whether the loan can be passed or not with the least amount of risk possible.

Plan of Implementation: -

The project can be broken down into 7 main steps which are as follows:

1. Understand the dataset.
2. Clean the data.
3. Analyse the columns to be Features.
4. Process the features as required by the model/algorithm.
5. Train the model/algorithm on training data.
6. Test the model/algorithm on testing data.
7. Tune the model/algorithm for higher accuracy.

Motivation: -

Loan approval is a very important process for banking organizations. The system approved or reject the loan applications. Recovery of loans is a major contributing parameter in the financial statements of a bank. It is very difficult to predict the possibility of payment of loan by the customer. Using Machine learning we predict the loan approval.

Overview: -

For banking organizations, loan approval and risk assessment which is related is a very complex and significant process which is needs a high effort for relevant employee or manager to take a decision, because of manual or traditional methods that used in banks. The banking industry still needs a more precise method of predictive modeling for several problems. In general, for financial institutions and especially for banks forecasting credit defaulters is a hard challenge. The primary role of the current systems is to accept, or sending loan application to a specific level of approval to be studied and it is very difficult to foresee the probability of the borrower for paying the due dues amount without using methods to predict. Machine learning (ML) techniques and the algorithm that belongs to are a very amazing and promising technique in predicting for a large amount of data. Our research proposed to study three machine learning algorithms [1], Decision Tree (DT), Logistic Regression (LR), and Random Forest (RF), by using real data collected from Quds Bank with a variables that cover credit restriction and regulator instructions. The algorithm has been implemented to predict the loan approval of customers and the output tested in terms of the predicted accuracy.

Existing System: -

In the existing system there is a formula to calculate the projected score and a win predictor based on the win percentage of a Bank loan. These techniques won't give accurate results because they are based on perceptions and predictions based on a particular instant.

Existing system purely depends on how the Loan deals with the very important factors that influence the outcome.

Proposed System: -

The following shows the pseudo code for the proposed loan prediction method

1. Load the data
2. Determine the training and testing data
3. Data cleaning and pre-processing.

a) Fill the missing values with mean values regarding numerical values.
b) Fill the missing values with mode values regarding categorical variables.c) Outlier treatment.

4. Apply the modelling for prediction

a) Removing the load identifier
b) Create the target variable (based on the requirement).
In this approach, target variable is loan-status
c) Create a dummy variable for categorical variable (if required) and
split the training and testing data for validation.
d) Apply the model: NB method, SVM method

5. Determine the accuracy followed by confusion Matrix.

CHAPTER 2

PROBLEM STATEMENT AND OBJECTIVE: -

PROBLEM STATEMENT: -

Banks, Housing Finance Companies and some NBFC deal in various types of loans like housing loan, personal loan, business loan etc in all over the part of countries. These companies have existence in Rural, Semi Urban and Urban areas. After applying loan by customer these companies validates the eligibility of customers to get the loan or not. This project provides a solution to automate this process by employing machine learning algorithm. So the customer will fill an online loan application form. This form consist details like Sex, Marital Status, Qualification, Details of Dependents, Annual Income, Amount of Loan, Credit History of Applicant and others.

OBJECTIVE: -

The objective of the problem is to pick out which customer will be able to pay the debt and which customer is likely will not be able to pay the debts. Clearly we have to create a classification model here. We have to use algorithms like logistic regression, decision tree or random forest. We need to create a model that is accurate and the error percentage should be less. The main objective of this project is to predict whether assigning the loan to particular person will be safe or not. ... In this paper we are predict the loan data by using some machine learning algorithms they are classification, logic regression, Decision Tree and gradient boosting.

- ❑ A classification model is run on data attempting to classify whether the person or client is eligible for get loan from any bank with good accuracy of statement.
- ❑ Our objectives included some points about this Loan Status Prediction.

CHAPTER 3

LITERATURE SURVEY: -

1." Loan Approval Prediction based on Machine Learning Approach" Author- Kumar Arun, Garg Ishan, Kaur Sanmeet Year- 2018The 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."Exploring the Machine Learning Algorithm for Prediction the Loan Sanctioning Process" Author- E. Chandra Blessie, R. Rekha - Year- 2019 Extending credits to corporates and individuals for the smooth functioning of growing economies like India is inevitable. As increasing number of customers apply for loans in the banks and non- banking financial companies (NBFC), it is really challenging for banks and NBFCs with limited capital to device a standard resolution and safe procedure to lend money to its borrowers for their financial needs. Inaddition, in recent times NBFC inventories have suffered a significant downfall in terms of the stock price. It has contributed to a contagion that has also spread to other financial stocks, adversely affecting the benchmark in recent times.In this paper, an attempt is made to condense the risk involved in selecting the suitable person who could repay the loan on time thereby keeping the bank's nonperforming assets (NPA) on the hold. This is achieved by feeding the past records of the customer who acquired loans from the bank into a trained machine learning model which could yield an accurate result. The prime focus of the paper is to determine whether or not it will be safe to allocate the loan to a particular person. This paper has the following sections (i) Collection of Data, (ii) Data Cleaning and (iii) Performance Evaluation. Experimental tests found that the Naïve Bayes model has better performance Evaluation. Experimental tests found that the Naïve Bayes model has better performance than other models in terms of loan forecasting.

3."Loan Prediction using machine learning model" Year- 2019whether or not it will be safe to allocate the loan to a particular person. This paper has the following sections (i) Collection of Data, (ii) Data Cleaning and (iii) Performance Evaluation. Experimental tests found that the Naïve Bayes model has better performance than other models in terms of loan forecasting.With the enhancement in the banking sector lots of people are applying for bankloans 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 project 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 mostaccurate result

CHAPTER 4

TECHNOLOGIES AND ALGORITHMS: -

Anaconda: -

Anaconda is a distribution of the Python and R programming languages for scientific computing, that aims to simplify package management and deployment. The distribution includes data-science packages suitable for Windows, Linux, and macOS. License: Freemium (Miniconda and the Individual Edition are free software, but the other editions are software as a service). Developer(s): Anaconda, Inc. (previously Continuum Analytics) Initial release: 0.8.0/17 July 2012; 9 years ago. Operating system: Windows, macOS, Linux. Stable release: 2021.11 / 17 November 2021; 2 months ago. Written in: Python. anaconda is a fairly sophisticated installer. It supports installation from local and remote sources such as CDs and DVDs, images stored on a hard drive, NFS, HTTP, and FTP. Installation can be scripted with kickstart to provide a fully unattended installation that can be duplicated on scores of machines

Anaconda is a distribution of the Python and R programming languages for scientific computing (data science, machine learning applications, large-scale data processing, predictive analytics, etc.), that aims to simplify package management and deployment. The distribution includes data-science packages suitable for Windows, Linux, and macOS. It is developed and maintained by Anaconda, Inc., which was founded by Peter Wang and Travis Oliphant in 2012. As an Anaconda, Inc. product, it is also known as Anaconda Distribution or Anaconda Individual Edition, while other products from the company are Anaconda Team Edition and Anaconda Enterprise Edition, both of which are not free. Package versions in Anaconda are managed by the package management system conda.

This package manager was spun out as a separate open-source package as it ended up being useful on its own and for things other than Python. There is also a small, bootstrap version of Anaconda called Miniconda, which includes only conda, Python, the packages they depend on, and a small number of other packages. Anaconda distribution comes with over 250 packages automatically installed, and over 7,500 additional open-source packages can be installed from PyPI as well as the conda package and virtual environment manager. It also includes a GUI, Anaconda Navigator, as a graphical alternative to the command-line interface (CLI).

The big difference between conda and the pip package manager is in how package dependencies are managed, which is a significant challenge for Python data science and the reason conda exists. Before version 20.3, when pip installed a package, it automatically installed any dependent Python packages without checking if these conflict with previously installed packages. It would install a package and any of its dependencies regardless of the state of the existing installation.[13] Because of this, a user with a working installation of, for example, TensorFlow, could find that it stopped working having used pip to install a different package that requires a different version of the dependent NumPy library than the one used by TensorFlow. In some cases, the package would appear to work but produce different results in detail. While pip has since implemented consistent dependency resolution,[13] this difference accounts for a historical differentiation of the conda package manager. In contrast, conda analyses the current environment including everything currently installed, and,

together with any version limitations specified (e.g. the user may wish to have TensorFlow version 2,0 or higher), works out how to install a compatible set of dependencies, and shows a warning if this cannot be done. Open-source packages can be individually installed from the Anaconda repository, Anaconda Cloud (anaconda.org), or the user's own private repository or mirror, using the conda install command. Anaconda, Inc. compiles and builds the packages available in the Anaconda repository itself, and provides binaries for Windows 32/64 bit, Linux 64 bit and MacOS 64-bit. Anything available on PyPI may be installed into a conda environment using pip, and conda will keep track of what it has installed itself and what pip has installed.

Custom packages can be made using the conda build command, and can be shared with others by uploading them to Anaconda Cloud, PyPI or other repositories. The default installation of Anaconda2 includes Python 2.7 and Anaconda3 includes Python 3.7. However, it is possible to create new environments that include any version of Python packaged with conda.

Anaconda Navigator: -

Anaconda Navigator is a desktop graphical user interface (GUI) included in Anaconda distribution that allows users to launch applications and manage conda packages, environments and channels without using command-line commands. Navigator can search for packages on Anaconda Cloud or in a local Anaconda Repository, install them in an environment, run the packages and update them. It is available for Windows, macOS and Linux.

The following applications are available by default in Navigator:

- JupyterLab
- Jupyter Notebook
- QtConsole
- Spyder Glue
- Orange
- RStudio
- Visual Studio

Jupyter Notebook: -

The Jupyter Notebook is an open-source web application that allows you to create and share documents that contain live code, equations, visualizations, and narrative text. Its uses include data cleaning and transformation, numerical simulation, statistical modelling, data visualization, machine learning, and much more. Jupyter Notebook (formerly IPython Notebooks) is a web-based interactive computational environment for creating Jupyter notebook documents. The “notebook” term can colloquially make reference to many different entities, mainly the Jupyter web application, Jupyter Python web server, or Jupyter document format depending on context.

According to the official website of, Project Jupyter exists to develop opensource software, open-standards, and services for interactive computing across dozens of programming languages. Jupyter Notebook Book is an open-source project for building books and documents from computational material. It allows the user to construct the content in

mixture of Markdown, an extended version of Markdown called MyST, Maths & Equations using Math Jax, Jupyter Notebooks, restructured Text, the output of running Jupyter Notebooks at build time. Multiple output formats can be produced (currently single files, multipage HTML web pages and PDF files). Now that we have a brief understanding of the concept of Project Jupyter and the Jupyter Notebooks, and we have established that these Notebooks are quite revolutionary in the modern ages, let us proceed to understand the more in-depth details of this amazing interactive environment in the next sections.

Python Programming: -

Python is used for web development, AI, machine learning, operating systems, mobile application development, and video games. A successor to the ABC programming language, Python is a high level, dynamically typed language developed by Guido Van Rossum in the early 1980s Python is an interpreted high-level general-purpose programming language. Its design philosophy emphasizes code readability with its use of significant indentation. Its language constructs and object-oriented approach aim to help programmers write clear, logical code for small- and large-scale projects. Python is dynamically-typed and garbage-collected. It supports multiple programming paradigms, including structured (particularly, procedural), object-oriented and functional programming. It is often described as a "batteries included" language due to its comprehensive standard library.

Guido van Rossum began working on Python in the late 1980s, as a successor to the ABC programming language, and first released it in 1991 as Python 0.9.0. Python 2.0 was released in 2000 and introduced new features such as list comprehensions, cycle-detecting garbage collection, reference counting, and Unicode support. Python 3.0, released in 2008, was a major revision that is not completely backward compatible with earlier versions. Python 2 was discontinued with version 2.7.18 in 2020. Python consistently ranks as one of the most popular programming languages.

History of Python: -

Python was conceived in the late 1980s by Guido van Rossum at Centrum Wickenden & Informatica (CWI) in the Netherlands as a successor to the ABC programming language, which was inspired by SETL capable of exception handling and interfacing with the Amoeba operating system.[10] Its implementation began in December 1989.[40] Van Rossum shouldered sole responsibility for the project, as the lead developer, until 12 July 2018, when he announced his "permanent vacation" from his responsibilities as Python's "benevolent dictator for life", a title the Python community bestowed upon him to reflect his long-term commitment as the project's chief decision-maker. In January 2019, active Python core developers elected a five-member Steering Council to lead the project.

Python 2.0 was released on 16 October 2000, with many major new features. Python 3.0, released on 3 December 2008, with many of its major features backported to Python 2.6.x and 2.7. x. Releases of Python 3 include the 2to3 utility, which automates the translation of Python 2 code to Python 3. Python 2.7's end-of-life was initially set for 2015, then postponed to 2020 out of concern that a large body of existing code could not easily be forward-ported to Python 3. No further security patches or other improvements will be released for it. With Python 2's end-of-life, only Python 3.6.x and later are supported. Python 3.9.2 and 3.8.8 were

expedited as all versions of Python (including 2.7) had security issues leading to possible remote code execution and web cache poisoning

Python Library: -

• Object-oriented signals that JavaScript's power to exert control over an HTML page is based on manipulating objects within that page.

• If you are familiar with object-oriented programming, you will be aware of some of the power that this can bring to the coding environment.

NumPy Library: -

NumPy (pronounced /ˈnʌmpaɪ/ (NUM-py) or sometimes /ˈnʌmpi/ (NUM-pee)) is a library for the Python programming language, adding support for large, multi-dimensional arrays and matrices, along with a large collection of high-level mathematical functions to operate on these arrays. The ancestor of NumPy, Numeric, was originally created by Jim Hugunin with contributions from several other developers. In 2005, Travis Oliphant created NumPy by incorporating features of the competing Numarray into Numeric, with extensive modifications. NumPy is open-source software and has many contributors. NumPy is a NumFOCUS fiscally sponsored project.

Pandas Library: -

Pandas is a software library written for the Python programming language for data manipulation and analysis. In particular, it offers data structures and operations for manipulating numerical tables and time series. It is free software released under the three-clause BSD license. The name is derived from the term "panel data", an econometrics term for data sets that include observations over multiple time periods for the same individuals. Its name is a play on the phrase "Python data analysis" itself. Wes McKinney started building what would become pandas at AQR Capital while he was a researcher there from 2007 to 2010.

Matplotlib Library: -

Matplotlib is a plotting library for the Python programming language and its numerical mathematics extension NumPy. It provides an object-oriented API for embedding plots into applications using general-purpose GUI toolkits like Tkinter, python, Qt, or GTK. There is also a procedural "pylab" interface based on a state machine (like OpenGL), designed to closely resemble that of MATLAB, though its use is discouraged. SciPy makes use of Matplotlib.

Matplotlib was originally written by John D. Hunter. Since then, it has an active development community and is distributed under a BSD-style license. Michael Droettboom was nominated as matplotlib's lead developer shortly before John Hunter's death in August 2012 and was further joined by Thomas Caswell. Matplotlib is a Num FOCUS fiscally sponsored project.

Matplotlib 2.0.x supports Python versions 2.7 through 3.10. Python 3 support started with Matplotlib 1.2. Matplotlib 1.4 is the last version to support Python 2.6. Matplotlib has pledged not to support Python 2 past 2020 by signing the Python 3 Statement.

Seaborn Library: -

Seaborn is a Python data visualization library based on matplotlib. It provides a high-level interface for drawing attractive and informative statistical graphics. For a brief introduction to the ideas behind the library, you can read the introductory notes or the paper. Visit the installation page to see how you can download the package and get started with it. You can browse the example gallery to see some of the things that you can do with seaborn, and then check out the tutorial or API reference to find out how. To see the code or report a bug, please visit the GitHub repository. General support questions are most at home on stack overflow or discourse, which have dedicated channels for seaborn.

SCIKIT LEARN: -

Scikit-learn (Sklearn) is the most useful and robust library for machine learning in Python. It provides a selection of efficient tools for machine learning and statistical modelling including classification, regression, clustering and dimensionality reduction via a consistence interface in Python. This library, which is largely written in Python, is built upon NumPy, SciPy and Matplotlib.

It was originally called ***scikits. learn*** and was initially developed by David Cournapeau as a Google summer of code project in 2007. Later, in 2010, Fabian Pedregosa, Gael Varoquaux, Alexandre Gramfort, and Vincent Michel, from FIRCA (French Institute for Research in Computer Science and Automation), took this project at another level and made the first public release (v0.1 beta) on 1st Feb. 2010.

MS Excel: -

Microsoft Excel is a spreadsheet program included in the Microsoft Office suite of applications. .... Spreadsheets present tables of values arranged in rows and columns that can be manipulated mathematically using both basic and complex arithmetic operations and functions Relational Database.

Heroku: -

Heroku is a cloud platform as a service (PaaS) supporting several programming languages. One of the first cloud platforms, Heroku has been in development since June 2007, when it supported only the Ruby programming language, but now supports Java, Node.js, Scala, Clojure, Python, PHP, and Go. For this reason, Heroku is said to be a polyglot platform as it has features for a developer to build, run and scale applications in a similar manner across most languages. Heroku was acquired by Salesforce in 2010 for $212 million.

History of Heroku: -

Heroku was initially developed by James Linden Baum, Adam Wiggins, and Orion Henry for supporting projects that were compatible with the Ruby programming platform known as Rack. The prototype development took around six months. Later on, Heroku faced setbacks because of lack of proper market customers as many app developers used their own tools and environment. [citation needed] In January 2009, a new platform was launched which was built almost from scratch after a three-month effort. In October 2009, Byron Sebastian joined

Heroku as CEO. On December 8, 2010, Salesforce.com acquired Heroku as a wholly owned subsidiary of Salesforce.com. On July 12, 2011, Yukihiro "Matz" Matsumoto, the chief designer of the Ruby programming language, joined the company as Chief Architect, Ruby. That same month, Heroku added support for Node.js and Clojure. On September 15, 2011, Heroku and Facebook introduced Heroku for Facebook. At present Heroku supports Redis databases in addition to its standard PostgreSQL.

Etymology: -

The name "Heroku" is a portmanteau of "heroic" and "haiku". The Japanese theme is a nod to Matz for creating Ruby. The name itself is pronounced similarly to the Japanese word meaning “widely” (Heroku), though the creators of Heroku did not want the name of their project to have a particular meaning, in Japanese or any other language, and so chose to invent a name.

Heroku Architecture: -

Applications that are run on Heroku typically have a unique domain used to route HTTP requests to the correct application container or dyno. Each of the dynos are spread across a "dyno grid" which consists of several servers. Heroku's Git server handles application repository pushes from permitted users. All Heroku services are hosted on Amazon's EC2 cloud-computing platform.

Flask Server: -

Flask is a web framework, it’s a Python module that lets you develop web applications easily. It’s has a small and easy-to-extend core: it’s a microframework that doesn’t include an ORM (Object Relational Manager) or such features. Flask is a web application framework written in Python. It was developed by Armin Ronacher, who led a team of international Python enthusiasts called Poocco. Flask is based on the Werkzeg WSGI toolkit and the Jinja2 template engine.

WSGI: -

The Web Server Gateway Interface (Web Server Gateway Interface, WSGI) has been used as a standard for Python web application development. WSGI is the specification of a common interface between web servers and web applications.

WERKZEUG: -

Werkzeug is a WSGI toolkit that implements requests, response objects, and utility functions. This enables a web frame to be built on it. The Flask framework uses Werkzeg as one of its bases.

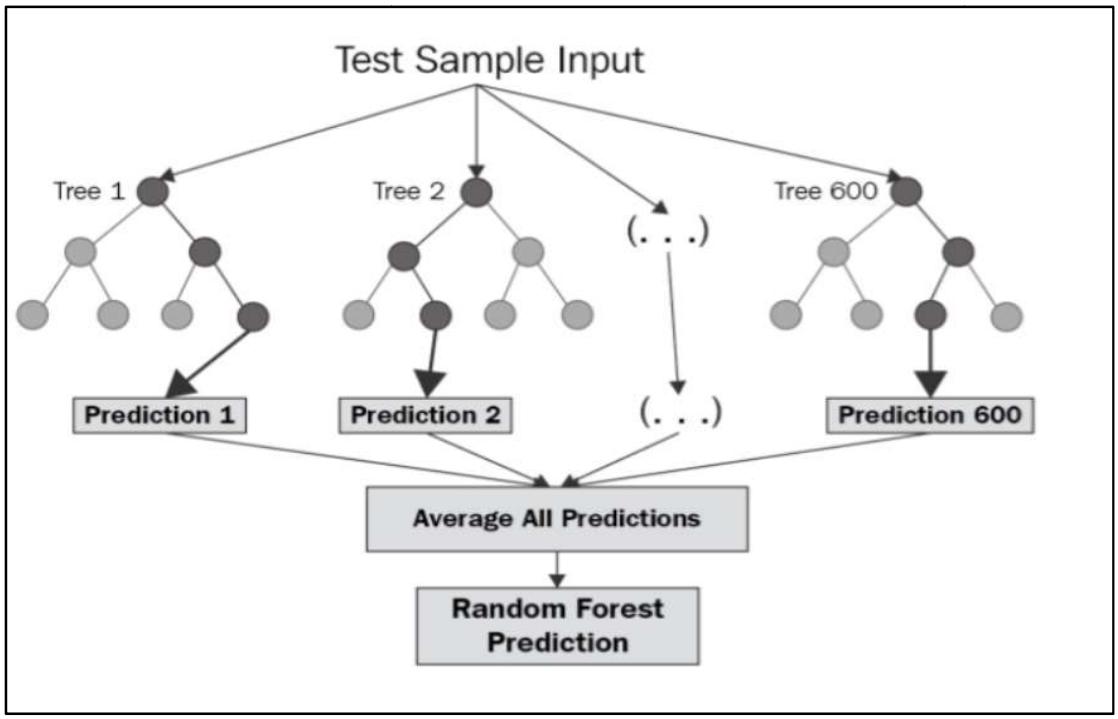
JINJA2: -

jinja2 is a popular template engine for Python. A web template system combines a template with a specific data source to render a dynamic web page.

RANDOM FOREST REGRESSION: -

Random Forest Regression is a supervised learning algorithm that uses ensemble learning method for regression. Ensemble learning method is a technique that combines predictions from multiple machine learning algorithms to make a more accurate prediction than a single model.

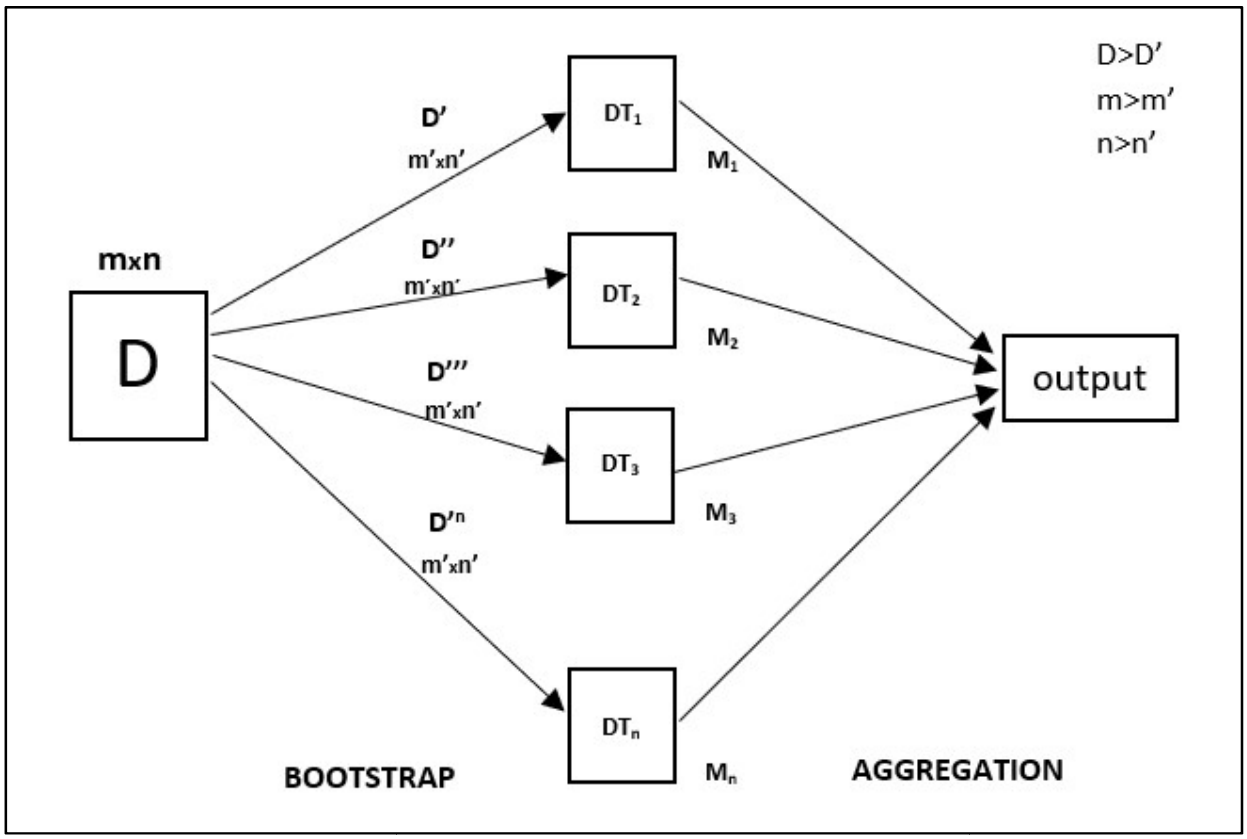
A Random Forest Regression model is powerful and accurate. It usually performs great on many problems, including features with non-linear relationships. Disadvantages, however, include the following: there is no interpretability, overfitting may easily occur, we must choose the number of trees to include in the model.



The diagram above shows the structure of a Random Forest. You can notice that the trees run in parallel with no interaction amongst them. A Random Forest operates by constructing several decision trees during training time and outputting the mean of the classes as the prediction of all the trees. To get a better understanding of the Random Forest algorithm, let's walk through the steps:

1. Pick at random *k* data points from the training set.
2.Build a decision tree associated to these *k* data points.
3.Choose the number *N* of trees you want to build and repeat steps 1 and 2.
4.For a new data point, make each one of your *N*-tree trees predict the value of *y* for the data point in question and assign the new data point to the average across all of the predicted *y* values.

Every decision tree has high variance, but when we combine all of them together in parallel then the resultant variance is low as each decision tree gets perfectly trained on that particular sample data, and hence the output doesn't depend on one decision tree but on multiple decision trees. In the case of a classification problem, the final output is taken by using the majority voting classifier. In the case of a regression problem, the final output is the mean of all the outputs. This part is called **Aggregation**.



CHAPTER 5

SOFTWARE REQUIREMENTS SPECIFICATIONS: -

System configurations the software requirement specification can produce at the culmination of the analysis task. The function and performance allocated to software as part of system engineering are refined by established a complete information description, a detailed functional description, a representation of system behaviour, and indication of performance and design constrain, appropriate validate criteria, and other information pertinent to requirements.

Software Requirements: -

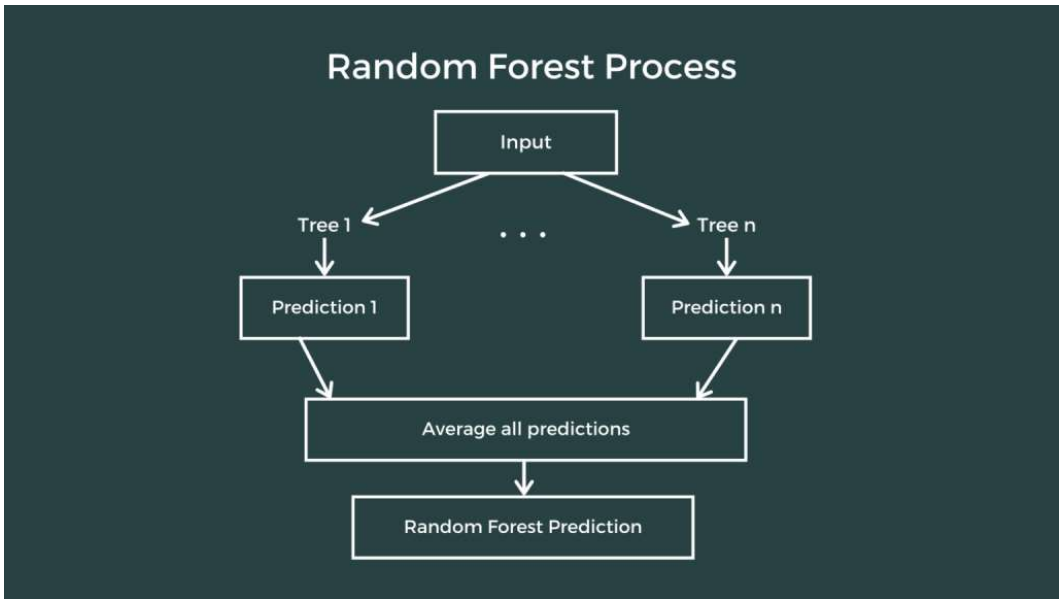
- Coding Language: Python.
- Front-End: Jupyter Notebook.
- Data Base: MS Excel.

Hardware Requirement: -

- Operating system: Windows 10 Home Single Language.
- System: Pentium IV 2.4 GHz.
- Hard Disk: 1TB.
- Ram: 8GB.

Functional Requirement: -

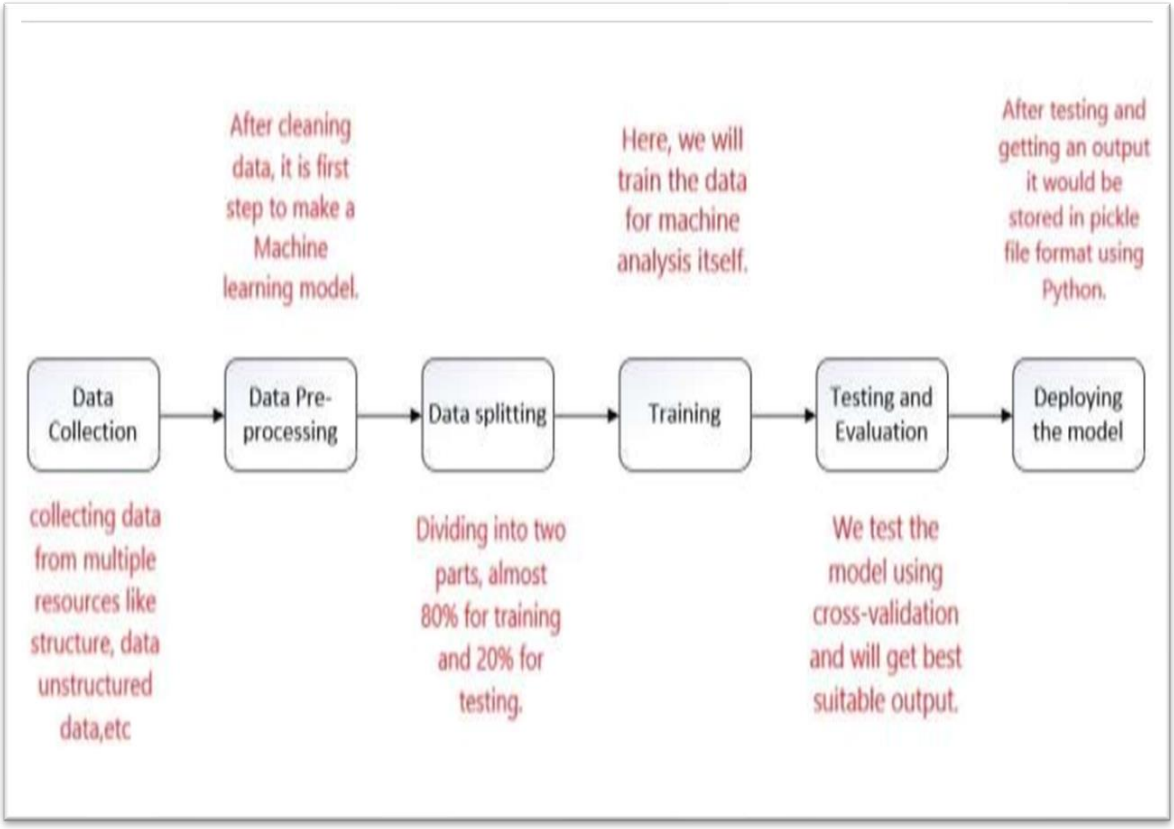
1. The System must provide the predicted Loan status .
2. The System must have an easy-to-use interface for the system for all the users.
3. The admin must be able to modify/update the dataset.
4. The dataset of the Loan Status must be available for the system.



CHAPTER 6

METHODOLOGY: -

Proposed Model: -



- **Data collection: -**

Solving machine learning problems firstly we require raw data because without raw data we cannot do machine learning problems. raw data we get from further discussion of the problem with client and data scientist team we focus on data that is a data integration and data integration is a very difficult task because we collect data from multiple resources like structure data unstructured data, web scraping, etc. collected data stored in data warehouse and we get data from a data warehouse.

Loan_ID	Gender	Married	Dependents	Education	Self_Employed	ApplicantIncome	CoapplicantIncome	LoanAmount	Loan_Amount_Term	Credit_History	Property_Area
LP001015	Male	Yes	0	Graduate	No	5720	0	110	360	1	Urban
LP001022	Male	Yes	1	Graduate	No	3076	1500	126	360	1	Urban
LP001031	Male	Yes	2	Graduate	No	5000	1800	208	360	1	Urban
LP001035	Male	Yes	2	Graduate	No	2340	2546	100	360		Urban
LP001051	Male	No	0	Not Graduate	No	3276	0	78	360	1	Urban
LP001054	Male	Yes	0	Not Graduate	Yes	2165	3422	152	360	1	Urban
LP001055	Female	No	1	Not Graduate	No	2226	0	59	360	1	Semiurban
LP001056	Male	Yes	2	Not Graduate	No	3881	0	147	360	0	Rural
LP001059	Male	Yes	2	Graduate		13633	0	280	240	1	Urban
LP001067	Male	No	0	Not Graduate	No	2400	2400	123	360	1	Semiurban
LP001078	Male	No	0	Not Graduate	No	3091	0	90	360	1	Urban
LP001082	Male	Yes	1	Graduate		2185	1516	162	360	1	Semiurban
LP001083	Male	No	3+	Graduate	No	4166	0	40	180		Urban
LP001094	Male	Yes	2	Graduate		12173	0	166	360	0	Semiurban
LP001096	Female	No	0	Graduate	No	4666	0	124	360	1	Semiurban
LP001099	Male	No	1	Graduate	No	5667	0	131	360	1	Urban
LP001105	Male	Yes	2	Graduate	No	4583	2916	200	360	1	Urban
LP001107	Male	Yes	3+	Graduate	No	3786	333	126	360	1	Semiurban
LP001108	Male	Yes	0	Graduate	No	9226	7916	300	360	1	Urban

- **Data Pre-processing: -**

  Data pre-processing is a process of preparing the raw data and making it suitable for a machine learning model. It is the first and crucial step while creating a machine learning model. When creating a machine learning project, it is not 0always a case that we come across the clean and formatted data. And while doing any operation with data, it is mandatory to clean it and put in a formatted way. So for this, we use data pre-processing task.

- **Data splitting: -**

  The train-test split procedure is used to estimate the performance of machine learning algorithms when they are used to make predictions on data not used to train the model. In this step data is split for training and testing almost 80% of data is for training and 20% for testing is a basic rule in the machine learning.

- **Training: -**

  In this step, we do training data for machine analysis itself and we do another step is to validate training data because training data set will produce either overfitting or under the fitting problem that means false positive output or true negative output that means overfitting means when you go new area and 1st person give disrespect and you considering all people are same this is.

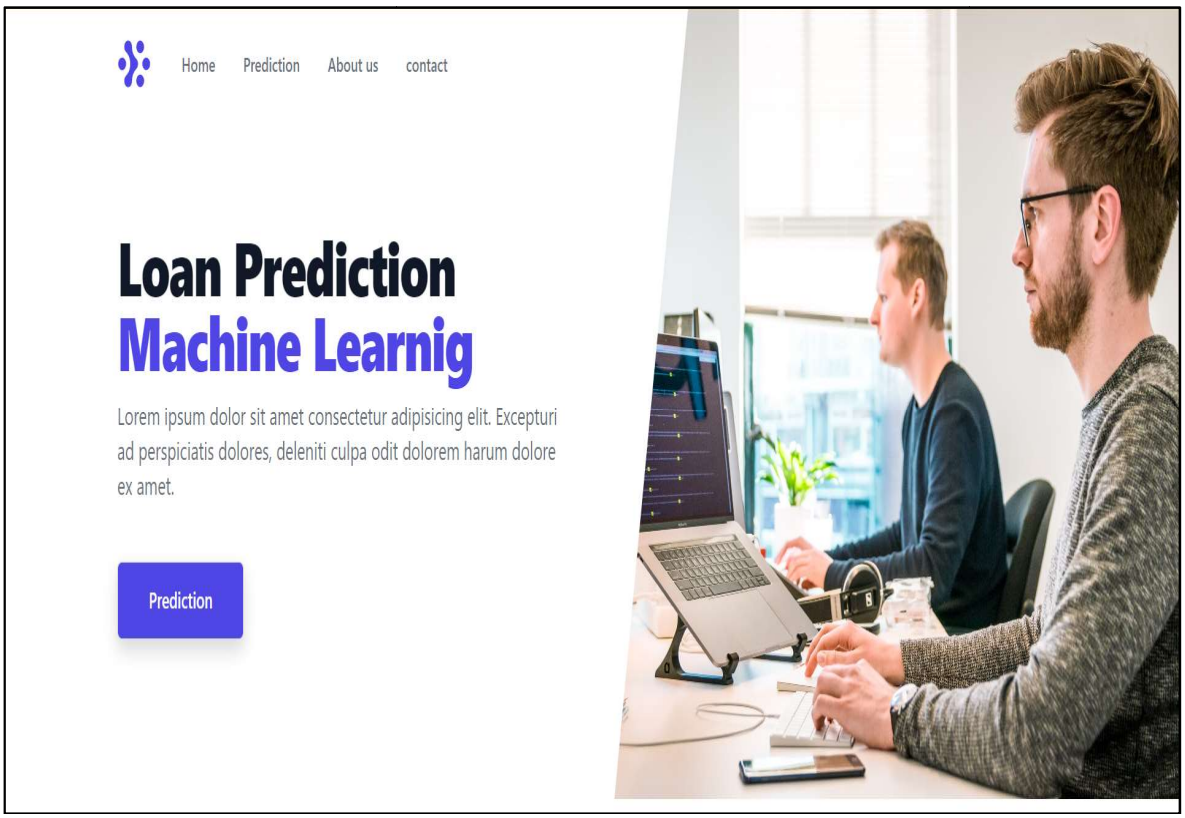
- **Testing and Evaluation: -**
  In the testing phase, we test the model using cross-validation, we check the model is well or not and going is right or not, there are some techniques of cross-validation and we use confusion matrix for checking model performance. We will test in all algorithms and will get best suitable output.

CHAPTER 7

APPENDIX: -

Now we are ready to run our model on our local machine. Open the command prompt and first change directory to the folder where we have the project saved. Then run **python app.py.**

Now on your browser open http://127.0.0.1:5000/ and run the application. Below are the images of the User Interface.



OUTPUT: -

You will need a procfile and requirements file. I have provided those in the GitHub link at the end of this article. The procfile will contain: *web: gunicorn app:app*

For creating a requirement file type this code on cmd in your virtual environment:

```
$ pip freeze > requirements.txt
```

We will deploy this project on Heroku platform.

1. Register on Heroku.
2. Upload project on GitHub.
3. Log in to your Heroku Dashboard.
4. Click on new/create new app.
5. Give an app name, choose region and click on create.
6. Then go onto the deploy section and connect your app to GitHub.
7. Click deploy project.

CHAPTER 8

SOURCE CODE: -

app.py code: -

```
 # save this as app.py
from flask import Flask, escape, request, render_template
import pickle
import numpy as np

app = Flask(__name__)
model = pickle.load(open('model.pkl', 'rb'))

@app.route('/')
def home():
    return render_template("index.html")
@app.route('/predict', methods=['GET', 'POST'])
def predict():
    if request.method ==  'POST':
        gender = request.form['gender']
        married = request.form['married']
        dependents = request.form['dependents']
        education = request.form['education']
        employed = request.form['employed']
        credit = float(request.form['credit'])
        area = request.form['area']
        ApplicantIncome = float(request.form['ApplicantIncome'])
        CoapplicantIncome = float(request.form['CoapplicantIncome'])
        LoanAmount = float(request.form['LoanAmount'])
        Loan_Amount_Term = float(request.form['Loan_Amount_Term'])

        # gender
        if (gender == "Male"):
            male=1
        else:
            male=0

        # married
        if(married=="Yes"):
            married_yes = 1
        else:
            married_yes=0
```

```
    # dependents
    if(dependents=='1'):
    dependents_1 = 1
    dependents_2 = 0
    dependents_3 = 0
    elif(dependents == '2'):
    dependents_1 = 0
    dependents_2 = 1
    dependents_3 = 0
    elif(dependents=="3+"):
    dependents_1 = 0
    dependents_2 = 0
    dependents_3 = 1
    else:
    dependents_1 = 0
    dependents_2 = 0
    dependents_3 = 0

    # education
    if (education=="Not Graduate"):
    not_graduate=1
    else:
    not_graduate=0

    # employed
    if (employed == "Yes"):
    employed_yes=1
    else:
    employed_yes=0

    # property area
if(area=="Semiurban"):
    semiurban=1
    urban=0
    elif(area=="Urban"):
    semiurban=0
    urban=1
    else:
    semiurban=0
    urban=0
```

```
 ApplicantIncomelog = np.log(ApplicantIncome)
      totalincomelog = np.log(ApplicantIncome+CoapplicantIncome)
      LoanAmountlog = np.log(LoanAmount)
      Loan_Amount_Termlog = np.log(Loan_Amount_Term)

      prediction = model.predict([[credit, ApplicantIncomelog,LoanAmountlog, Loan_Amount_Termlog,
totalincomelog, male, married_yes, dependents_1, dependents_2, dependents_3, not_graduate,
employed_yes,semiurban, urban ]])

      # print(prediction)

      if(prediction=="N"):
         prediction="oops,sorry you cant't capanle for any loan....."
      else:
         prediction="Congratualations,You are capable for the loan....."


      return render_template("prediction.html", prediction_text="loan status is :-  {}".format(prediction))




   else:
      return render_template("prediction.html")



if __name__ == "__main__":
   app.run(debug=True)
```

CHAPTER 9

RESULTS: -

I have just compared 2 algorithms here Logistic Regression and Random Forest Regression.

The first algorithm we will look at is **Logistic Regression**. We have used **GridSearchCV** for hyperparameter tuning.

The second used is Random Forest Regression. I have used **RandomizedSearchCV** for hyperparameter tuning.

In [61]:
```
# randomforest classifier
from sklearn.ensemble import RandomForestClassifier
model = RandomForestClassifier()

model.fit(x_train, y_train)
```

Out[61]:
```
RandomForestClassifier(bootstrap=True, ccp_alpha=0.0, class_weight=None,
                       criterion='gini', max_depth=None, max_features='auto',
                       max_leaf_nodes=None, max_samples=None,
                       min_impurity_decrease=0.0, min_impurity_split=None,
                       min_samples_leaf=1, min_samples_split=2,
                       min_weight_fraction_leaf=0.0, n_estimators=100,
                       n_jobs=None, oob_score=False, random_state=None,
                       verbose=0, warm_start=False)
```

In [62]:
```
print("Accuracy is", model.score(x_test, y_test)*100)
```

```
Accuracy is 79.22077922077922
```

In [63]:
```
# decision tree classifier
from sklearn.tree import DecisionTreeClassifier
model2 = DecisionTreeClassifier()
model2.fit(x_train, y_train)
print("Accuracy is", model2.score(x_test, y_test)*100)
```

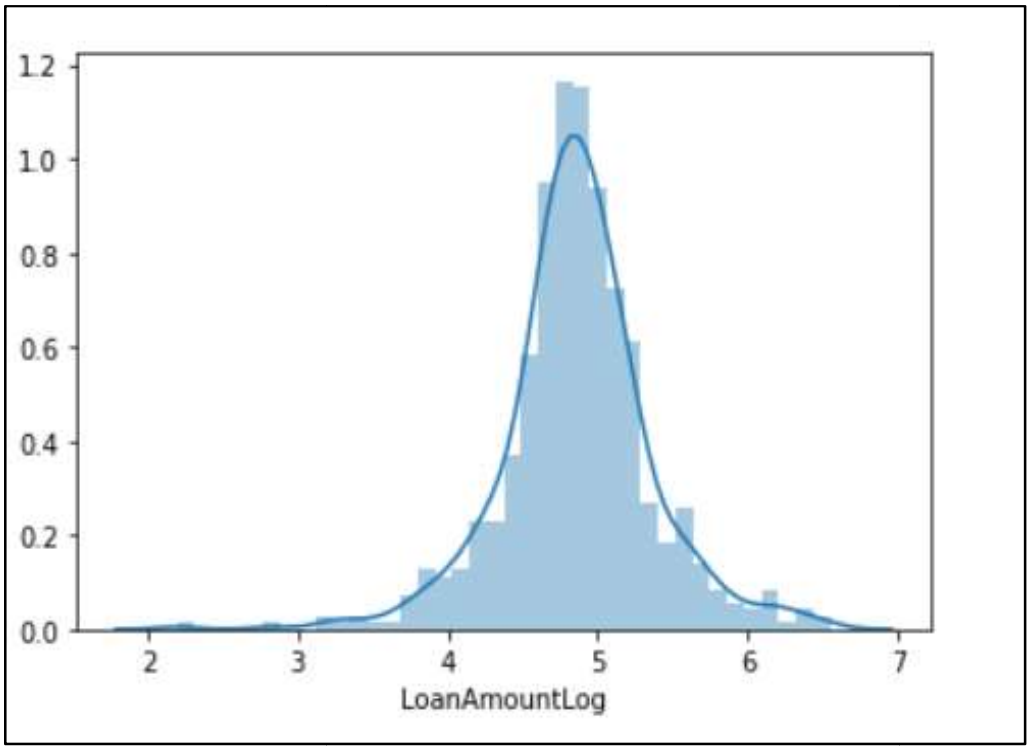
```
Accuracy is 70.77922077922078
```

In [64]:
```
# logistic regression
from sklearn.linear_model import LogisticRegression
model3 = LogisticRegression()
model3.fit(x_train, y_train)
print("Accuracy is", model3.score(x_test, y_test)*100)
```

```
Accuracy is 77.27272727272727
```

1. Random Forest Model: -

Evaluating the Random Forest Regression model using **Distplot** and **Sklearn Metrics:**



By analyzing the above models, we can conclude that Random Forest Regression works better onour dataset as it had lower error metric values than Logistic Regression.

CHAPTER 10

CONCLUSION: -

The predictive models based on Logistic Regression, Decision Tree and Random Forest, give the accuracy as 80.945%, 93.648% and 83.388% whereas the cross-validation is found to be 80.945%, 72.213% and 80.130% respectively. This shows that for the given dataset, the accuracy of model based on decision tree is highest but random forest is better at generalization even though it's cross validation is not much higher than logistic regression.

FEATURE SCOPE: -

- In future, this model can be used to compare various machine learning algorithm generated prediction models and the model which will give higher accuracy will be chosen as the prediction model. This paper work can be extended to higher level in future. Predictive model for loans that uses machine learning algorithms, where the results from each graph of the paper can be taken as individual criteria for the machine learning algorithm.

CHAPTER 11

BIBLOGRAPHY: -

We express our sincere gratitude to all those people who helped us in gathering the information while preparing this project. To prepare this project we required information regarding how to develop efficient & proper software on Loan Prediction system.

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.

- https://towardsdatascience.com/
- https://geeksforgeeks.org/
- https://github.com/
- https://www.kaggle.com/
- https://w3schools.com/python/

REFERENCES: -

[1] Vaidya and Ashlesha, Predictive and probabilistic approach using logistic regression: Application to prediction of loan approval, 2017 8th International Conference on Computing, Communication and Networking Technologies (ICCCNT). IEEE, 2017.

[2] Amin, Rafik Khairul and Yuliant Sibaroni, Implementation of decision tree using C4. 5 algorithm in decision making of loan application by debtor (Case study: Bank pasar of Yogyakarta Special Region), 2015 3rd International Conference on Information and Communication Technology (ICoICT). IEEE, 2015.

[3] Arora, Nisha and Pankaj Deep Kaur, A Bolasso based consistent feature selection enabled random forest classification algorithm: An application to credit risk assessment, Applied Soft Computing 86 (2020), 105936.

[4] Yang, Baoan, et al, An early warning system for loan risk assessment using artificial neural networks, Knowledge-Based Systems 14.5-6 (2001), 303-306.

[5] Metawa, Noura, M. Kabir Hassan and Mohamed Elhoseny, Genetic algorithm based model for optimizing bank lending decisions, Expert Systems with Applications 80 (2017), 75-82.

[6] Hassan, Amira Kamil Ibrahim and Ajith Abraham. "Modeling consumer loan default prediction using ensemble neural networks, 2013 International Conference On Computing, Electrical And Electronic Engineering (ICCEEE). IEEE, 2013.