TECHNICAL DESIGN DOCUMENT

Loan Approval Prediction using Machine Learning

Work By:

Rimi Mondal

Table of Content

Introduction

This predictive model project involves building a model that can predict future outcomes based on historical data. Predictive model is a process that uses statistical algorithms and machine learning techniques to analyse historical data and make predictions about future events.

It involves with various steps:

- Define Problem
- Collection and preparation of data
- Exploration of data
- Introduce feature engineering
- Various model selection and training
- Model Evaluation
- Model Deployment

Overall, this model is mostly used in financial sector.

In the loan eligibility tutorial from scratch to advance is a comprehensive guide that teaches the process of building a loan eligibility model from the beginning to the end, covering both the theoretical and practical aspects of the project.

Overall, a loan eligibility tutorial from scratch to advance will provide a comprehensive guide for building a loan eligibility model that can help financial institutions make data-driven decisions and improve their business processes.

Data Set and Coding

Coding Language Use

For the project we use Python as a coding language because it has a vast standard library that includes machine learning and data analysis so it can be beneficial and easy for produce data in best precise manner.

Benefits of Python:

- Easy to use
- Large standard library
- Cross-Platform Compatibility
- High-Level Language
- Dynamic Typing
- Extensive Support for Libraries
- Ideal for Rapid Prototyping
- Great for Data Science and Machine Learning

- Open-Source and Free

Applications of the Project

- Generate the idea
- Download the data from Kaggle
- Upload the CSV files in Python
- Cleaning the Data
- Identify the important features
- Generate the outcomes

These above steps help to make perfect predictive model.

Decision Tree Model

1. Import the necessary libraries: To perform decision tree in Python, we need to import the necessary libraries like pandas, numpy, and scikit-learn. These libraries provide the tools and functions to build and train decision tree models.
2. Load the data: We need to load the data into Python using the pandas library. The data should be in a tabular format with the predictor variables and the target variable in separate columns.
3. Pre-process the data: We need to pre-process the data to handle missing values, encode categorical variables, and scale the data if required. This step is important to ensure that the data is suitable for building a decision tree model.
4. Split the data: We need to split the data into training and testing sets. The training set is used to train the decision tree model, and the testing set is used to evaluate the performance of the model.
5. Build the decision tree model: We need to build the decision tree model using the scikit-learn library. We need to specify the hyperparameters of the decision tree model, such as the maximum depth of the tree, the criterion for splitting the nodes, and the minimum number of samples required to split a node.
6. Train the model: We need to train the decision tree model using the training data. The model will learn the patterns in the training data and use them to make predictions on new data.
7. Evaluate the model: We need to evaluate the performance of the decision tree model using the testing data. We can use metrics such as accuracy, precision, recall, and F1-score to evaluate the performance of the model.

8. Visualize the tree: We can visualize the decision tree using the graphviz library. This step is optional but can be useful to understand the structure of the decision tree and the rules it uses to make predictions.
9. Use the model for prediction: Once the model is trained and evaluated, we can use it to make predictions on new data. We can pass the new data to the model, and it will use the rules learned from the training data to make predictions on the new data.

Various Libraries

- **Import pandas as pd:** It is a statement in Python that imports the pandas library and assigns it the alias "pd". This allows us to use the functions and classes provided by pandas for data analysis and manipulation in our Python code.

- **Import numpy as np:** It is a powerful numerical computing library for Python that provides support for arrays and matrices, as well as a wide range of mathematical functions and operations on arrays.

- **Import seaborn as sns:** It is a data visualization library for Python that is built on top of the matplotlib library. Seaborn provides a high-level interface for creating informative and attractive statistical graphics. It supports a variety of plots such as scatter plots, line plots, bar plots, histograms, heatmaps, and more.

- **Import matplotlib.pyplot as plt:** Matplotlib is a data visualization library for Python that provides a wide range of plotting functions and tools. The pyplot module of Matplotlib provides a simple and convenient interface for creating various types of plots such as line plots, scatter plots, histograms, bar plots, and more.

- **Import plotly.express as px:** Plotly Express is a data visualization library for Python that provides a simple and convenient interface for creating interactive plots. It is built on top of the Plotly library and provides a high-level API for creating a wide range of plots such as scatter plots, line plots, bar plots, pie charts, and more.

- **Sklearn feature selection import RFE:** The scikit-learn library is a widely used machine learning library for Python that provides a range of functions and tools for classification, regression, clustering, and other tasks. The feature selection module of scikit-learn provides functions and algorithms for selecting the most relevant features from a dataset for use in machine learning models.

- **Sklearn.model_selection import train_test_split:** This train_test_split function is a utility function for splitting a dataset into random training and testing sets. It takes as input the dataset to be split, the proportion of the dataset to be used for testing, and an optional random seed value for reproducibility.

Data Type and Size

Data Type and Size:

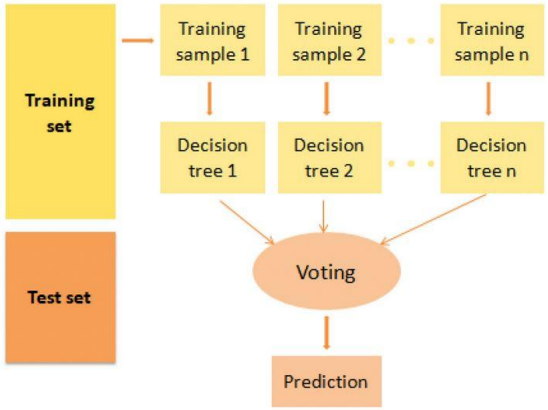
In the project two types of data use train and test data. The size of the data in the train dataset is 614 entries and 13 rows and in the test dataset 367 entries and 12 rows.

```
Total (614, 13) Columns and Rows in the Train Dataset
Total 614 Rows in the Train Dataset
Total 13 Rows in the Train Dataset

-------------------------------------------------------------------------------
-----------------------------------

Total (367, 12) Columns and Rows in the Test Dataset
Total 367 Rows in the Test Dataset
Total 12 Rows in the Test Dataset
```

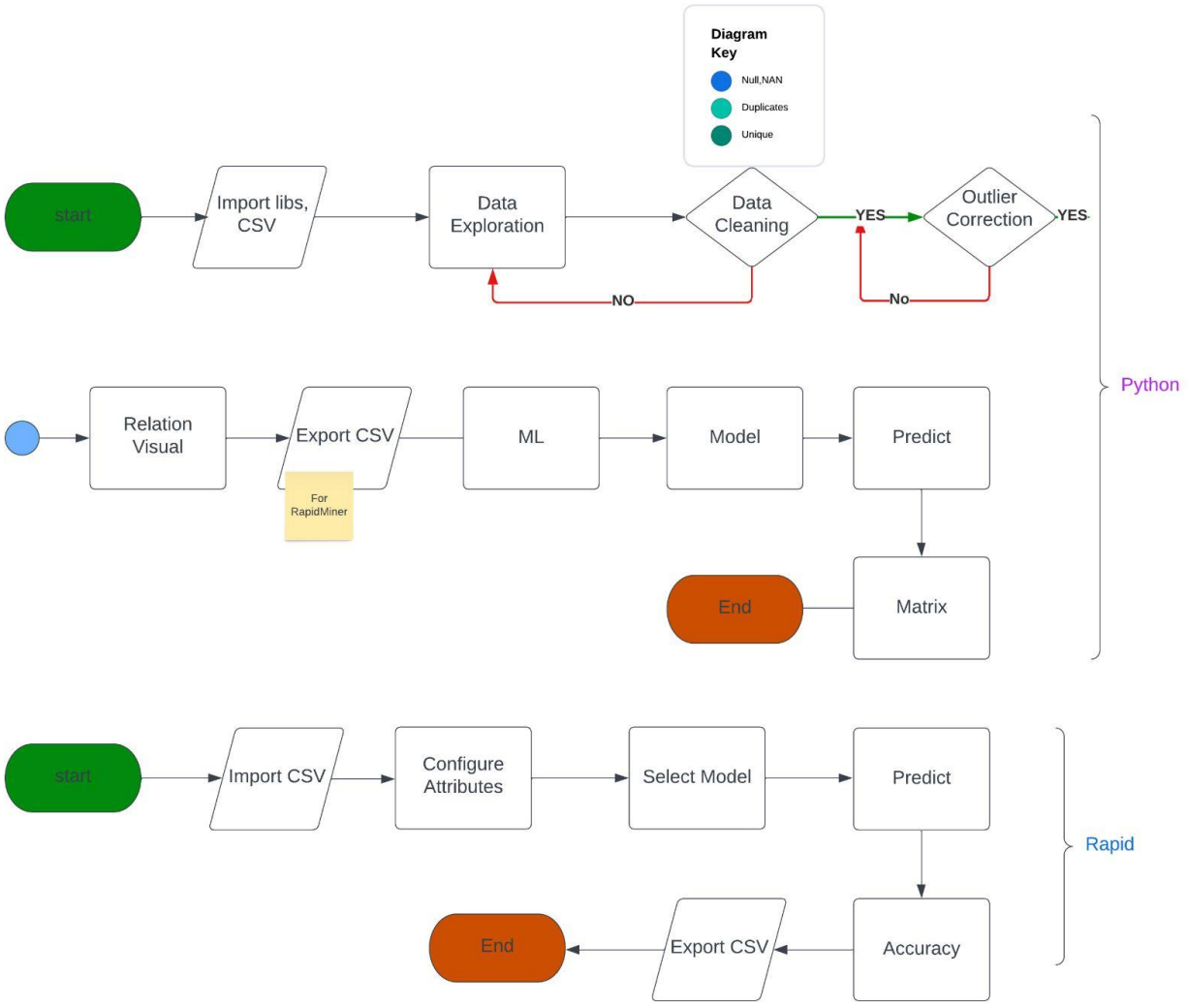
1. Load_Id: It is unique loan identifier in string format.
2. Gender: Define gender of loan lenders in string format.
3. Married: Marital status of loan lenders in string format.
4. Dependents: It's Boolean dataset.
5. Self_Employed: Define employment status of loan lender.
6. ApplicantIncome: Income of loan lenders.
7. CoapplicantIncome: It mention loan lender's partner income.
8. LoanAmount: It mention amount of loan sanctioned.
9. Loan_Amount_Term: It in the days of the loan.

Flowchart

Random Forest Chart



BPMN Flowchart:



Cost Estimation

Hardware costs: The project requires a computer with sufficient processing power and memory to handle the data processing and training. Assuming a mid-range laptop or desktop, the cost could be around CAD 1000.

Software costs: The project requires several software tools and libraries, including Python, Jupyter Notebook, pandas, numpy, seaborn, matplotlib.pyplot, plotly.express, and scikit-learn. These tools and libraries are mostly open-source and free, but there may be some minor costs associated with setting up the development environment. Assuming minor costs, the software cost could be around CAD 200.

Data costs: The project uses publicly available data, so there are no direct costs associated with acquiring the data.

Training costs: The project requires knowledge and skills in data science and machine learning, which can be acquired through self-study or by taking courses. Assuming self-study, the training cost could be around CAD 0.

Time costs: The project requires a significant amount of time to complete, including problem definition, data collection and preparation, data exploration, feature engineering, model selection and training, model evaluation, and model deployment. Assuming a reasonable hourly rate of CAD 50, the time cost could be around CAD 1250.

Overall, the estimated cost of the project is around CAD 2500. Keep in mind that this is just an estimate, and the actual cost could vary depending on the specific hardware and software requirements, as well as the time and training costs.

References

https://www.kaggle.com/code/vikasukani/loan-eligibility-prediction-machine-learning/notebook

https://www.kaggle.com/code/talhabu/loan-eligibility-tutorial-from-scratch-to-advance

https://pandas.pydata.org/docs/

https://seaborn.pydata.org/

https://matplotlib.org/

https://plotly.com/python/

https://docs.rapidminer.com/

https://www.geeksforgeeks.org/