Final Project Report Submission

REGEX Software

Loan Status Prediction by using selected Machine Learning Algorithm.

By: Poulami Bakshi
Arshiya Moin

Guided by:
Ashutosh Pandey

Introduction:

In finance, a **loan** is the lending of money by one or more individuals, organizations, or other entities to other individuals, organizations, etc. The recipient (i.e. the borrower) incurs a debt, and is usually liable to pay interest on that debt until it is repaid, and also to repay the principal amount borrowed. To read more check out Wikipedia. The whole process of ascertaining if a burrower would pay back loans might be tedious hence the need to automate the procedure.

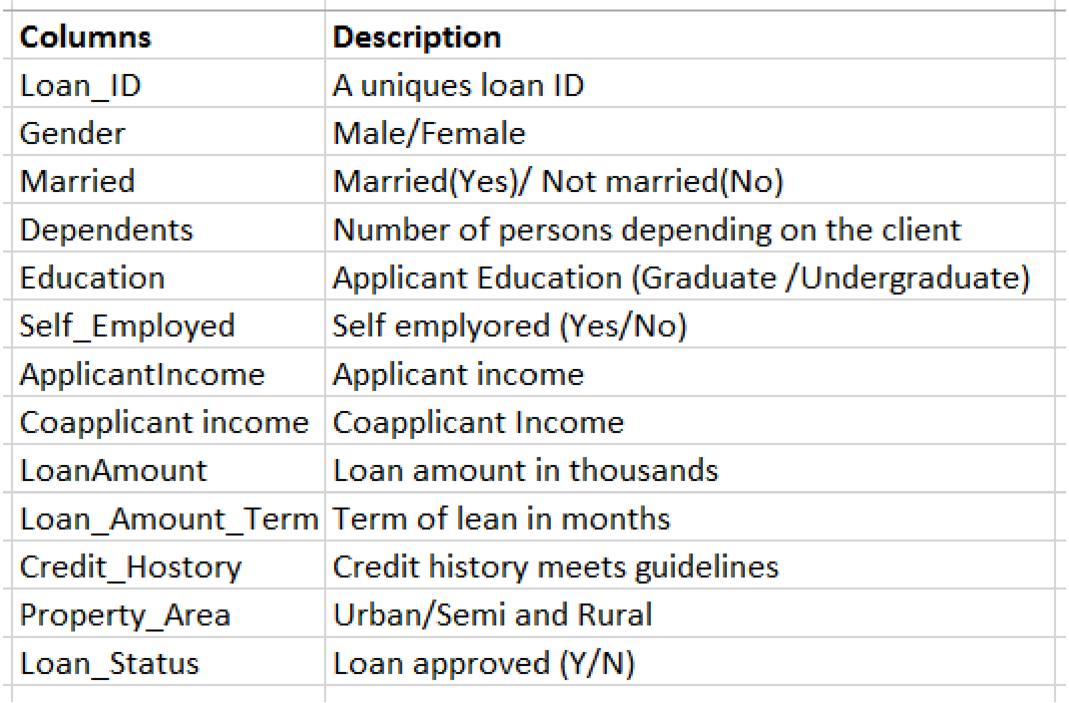
The problem at hand: The major aim of this project is to predict which of the customers will have their loan paid or not. Therefore, this is a supervised classification problem to be trained with algorithms like:

1. Logistic Regression
2. Decision Tree
3. Random Forest

Note: The machine learning classifier that can be used is not limited to the aforementioned. Other models like XGBoost, CatBoost, and the likes can be applied in the training of the model. The choice of these three algorithms is sequel upon the desire to keep the model explanatory of itself and also, the dataset is small.

Source of Data: The dataset for this project is retrieved from Kaggle, the home of Data Science.

Details Of Datasets:



Columns	Description
Loan_ID	A uniques loan ID
Gender	Male/Female
Married	Married(Yes)/ Not married(No)
Dependents	Number of persons depending on the client
Education	Applicant Education (Graduate /Undergraduate)
Self_Employed	Self emplyored (Yes/No)
ApplicantIncome	Applicant income
Coapplicant income	Coapplicant Income
LoanAmount	Loan amount in thousands
Loan_Amount_Term	Term of lean in months
Credit_Hostory	Credit history meets guidelines
Property_Area	Urban/Semi and Rural
Loan_Status	Loan approved (Y/N)

This table shows the variable names and their corresponding description.

Tools/Skills Used:

1. Python programing
2. Jupyter Notebook
3. Pandas
4. Numpy
5. Matplotlib
6. Seaborn
7. Exploratory Data Analytics
8. Feature Engineering

9. Data Visualization

10. Sciklearn

11. Machine Learning Algorithm

Reading the Data:

We are using the Pandas library to load the CSV file in the Jupyter Notebook. Pandas is one of the tools in Machine Learning which is used for data cleaning and analysis. It has features that are used for exploring, cleaning, transforming, and visualizing from data.

head():- Returns the first 5 rows of the Dataframe. To override the default, you may insert a value between the parenthesis to change the number of rows returned. Example: df. head(10) will return 10 rows.

Data Analysis:

Matplotlib and Seaborn are the two libraries that are been used in this model. Matplotlib is mainly deployed for basic plotting. Visualization using Matplotlib generally consists of bars, pies, lines, scatter plots, and so on. Seaborn, on the other hand, provides a variety of visualization patterns. It uses fewer syntax and has easily interesting default themes.

This model applying countplot() from seaborn to show the visualizing count of the target feature.

countplot(): Show the counts of observations in each categorical bin using bars. A count plot can be thought of as a histogram across a categorical, instead of quantitative, variable. The basic API and options are identical to those for barplot(), so you can compare counts across nested variables.

Pairplot(): A pairplot plots pairwise relationships in a dataset. The pairplot function creates a grid of Axes such that each variable in data will be shared in the y-axis across a single row and in the x-axis across a single column.

To plot multiple pairwise bivariate distributions in a dataset, you can use the pairplot() function. This shows the relationship for (n,2) combination of variables in a DataFrame as a matrix of plots, and the diagonal plots are the univariate plots.

Data Cleaning:

It is very important to clean the data because it contains many unwanted columns unwanted data outliers null values nan columns and many more. Data cleaning refers to identifying and correcting errors in the dataset that may negatively impact a predictive model. Data cleaning is used to refer to all kinds of tasks and activities to detect and repair errors in the data.

Hence, using some basics like:

df.columns, df.isnull().sum(), df.drop().

isnull(). sum(). sum() returns the number of missing values in the data set. A simple way to deal with data containing missing values is to skip rows with missing values in the dataset.

The drop() function is used to drop specified labels from rows or columns. Remove rows or columns by specifying label names and corresponding axis, or by specifying directly index or column names. When using a multi-index, labels on different levels can be removed by specifying the level.

df.unique(): The unique() function is used to find the unique elements of an array. Returns the sorted unique elements of an array. The indices of the unique array that reconstruct the input array. the number of times each unique value comes up in the input array.

Scaling the data:

Centering and scaling happen independently on each feature by computing the relevant statistics on the samples in the training set. Mean and standard deviation are then stored to be used on later data using transform.

Standardization of a dataset is a common requirement for many machine learning estimators: they might behave badly if the individual features do not more or less look like standard normally distributed data (e.g. Gaussian with 0 mean and unit variance).

For instance, many elements used in the objective function of a learning algorithm (such as the RBF kernel of Support Vector Machines or the L1 and L2 regularizers of linear models) assume that all features are centered around 0 and have variance in the same order. If a feature has a variance that is orders of magnitude larger than others, it might dominate the objective function and make the estimator unable to learn from other features correctly as expected.

Modeling:

train_test_split() is a function in Sklearn model selection for splitting data arrays into two subsets: for training data and for testing data. With this function, you don't need to divide the dataset manually.

By default, Sklearn train_test_split will make random partitions for the two subsets. However, you can also specify a random state for the operation.

Decision Trees are a type of Supervised Machine Learning (that is you explain what the input is and what the corresponding output is in the training data) where the data is continuously split according to a certain parameter. The tree can be explained by two entities, namely decision nodes and leaves. The leaves are the decisions or the final outcomes. And the decision nodes are where the data is split.

An example of a decision tree can be explained using the above binary tree. Let's say you want to predict whether a person is fit given their information like age, eating habits, and physical activity, etc. The decision nodes here are questions like 'What's the age?', 'Does he exercise?', 'Does he eat a lot of pizzas'? And the leaves, which are outcomes like either 'fit', or 'unfit'. In this case, this was a binary classification problem (a yes-no type problem). There are two main types of Decision Trees:

Classification trees (Yes/No types).

What we've seen above is an example of a classification tree, where the outcome was a variable like 'fit' or 'unfit'. Here the decision variable is Categorical.

Regression trees (Continuous data types)

A **confusion matrix** is a table that is often used to describe the performance of a classification model (or "classifier") on a set of test data for which the true values are known. The confusion matrix itself is relatively simple to understand, but the related terminology can be confusing. Confusion Matrix is a useful machine learning method that allows you to measure Recall, Precision, Accuracy, and AUC-ROC curve. Below given is an example to know the terms True Positive, True Negative, False Negative, and True Negative. True Positive: You projected positive and its turns out to be true.

The **classification report** is about key metrics in a classification problem. You'll have precision, recall, f1-score, and support for each class you're trying to find. The recall means "how many of this class you find over the whole number of elements of this class".

Model accuracy is defined as the number of classifications a model correctly predicts divided by the total number of predictions made. It's a way of assessing the performance of a model, but certainly not the only way.

Conclusion:

From the Exploratory Data Analysis, we could generate insight from the data. How each of the features relates to the target. Also, it can be seen from the evaluation of three models that Logistic Regression performed better than others, Random Forest did better than Decision Tree.