Machine Learning Project

Harshal Yaravalkar

Background

- **Data Set Details:**

Getting a loan approved requires a complex mix of factors not the least of which is a steady income! So this ML project aims to create a model that will classify if loan can be approved to the user based on various factors such as the user's marital status, income, education, employment prospects, number of dependents, etc.

- The dataset provides details about all these factors which can then be used to create an ML model that demonstrates if loan can be approved.
- Train dataset has shape (614,13)
- Test dataset has shape (367,12)

Dataset

- In this Loan Status Prediction dataset, we have the data of applicants who previously applied for the loan based on the property which is a Property Loan.
- The bank will decide whether to give a loan to the applicant based on some factors such as Applicant Income, Loan Amount, previous Credit History, Co-applicant Income, etc…
- Our goal is to build a Machine Learning Model to predict the loan to be approved or to be rejected for an applicant.

Dataset Features

- **Loan_ID**: A unique loan ID
- **Gender**: Either male or female.
- **Married**: Weather Married(yes) or Not Married(No).
- **Dependents**: Number of persons depending on the client.
- **Education**: Applicant Education(Graduate or Undergraduate).
- **Self_Employed**: Self-employed (Yes/No).
- **ApplicantIncome**: Applicant income.
- **CoapplicantIncome**: Co-applicant income.
- **LoanAmount**: Loan amount in thousands.
- **Loan_Amount_Term**: Terms of the loan in months.
- **Credit_History**: Credit history meets guidelines.
- **Property_Area**: Applicants are living in either Urban, Semi-Urban or Rural
- **Loan_Status**: Loan approved (Y/N).

Goal

- In this project, we are going to classify an individual whether he/she can get the loan amount based on his/her Income, Education, Working Experience, Loan taken previously, and many more factors. Let’s get more into it by looking at the data.

Requirements

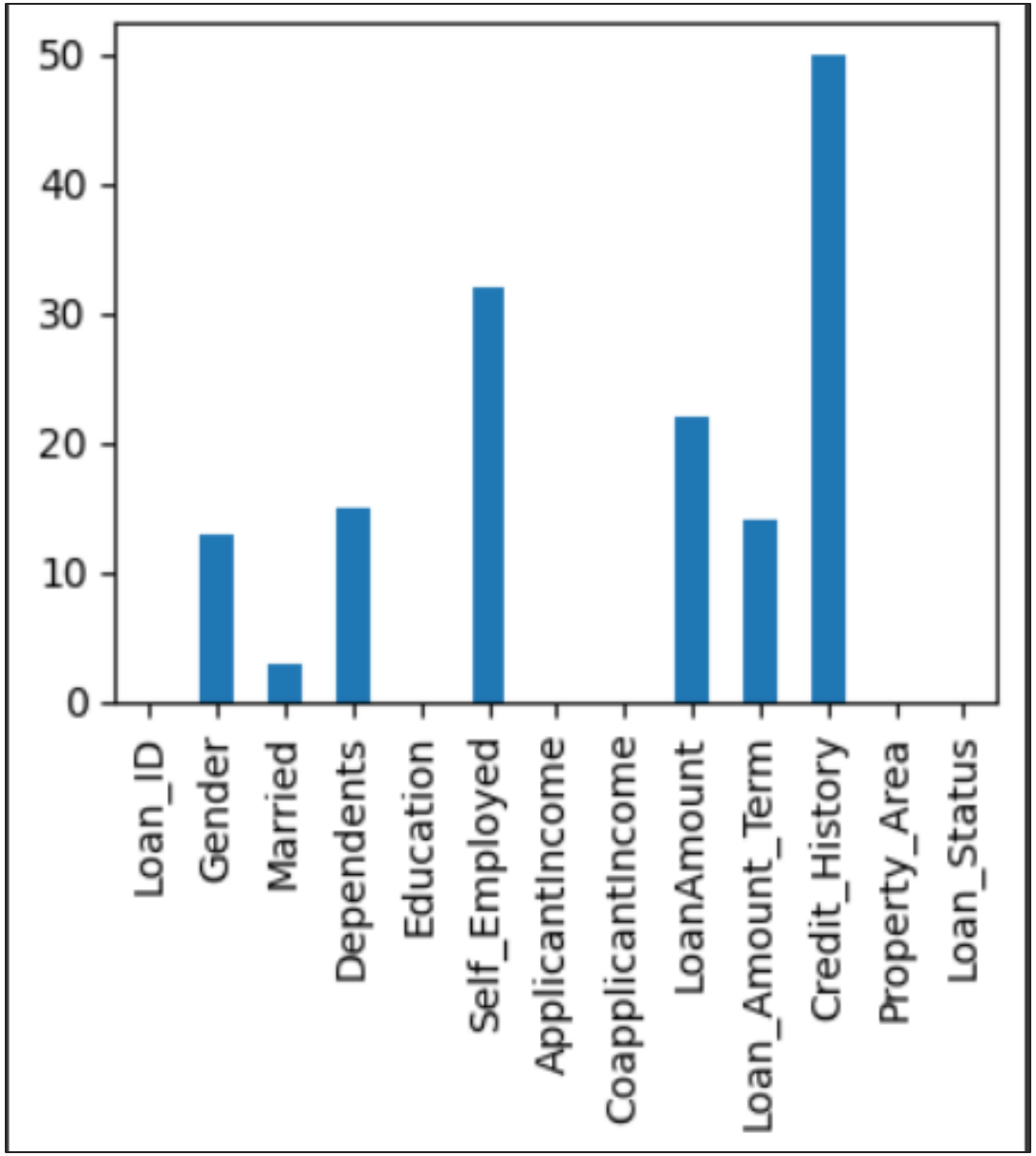
- Importing required libraries and modules. We will be using these requirements to modify, visualize data and clean it for training.

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
warnings.filterwarnings('ignore')
```

```
df_train = pd.read_csv("/content/train.csv")
df_test = pd.read_csv("/content/test.csv")
```

Handling Null data

- Plotted number of null values in each column.
- The null values of Features Gender, Self_Employed, Loan_Amount_term, Credit_History, Loan_Amount are filled with mode and mean using fillna function.
- The column Dependents has values 0,1,2,3+ in string data type to make them numerical values. They were replaced with 0,1,2,3 using replace function.
- “Married” column has only few missing values so those rows were dropped.
- Loan_ID column is also dropped as it is not relevant regarding our predictions.



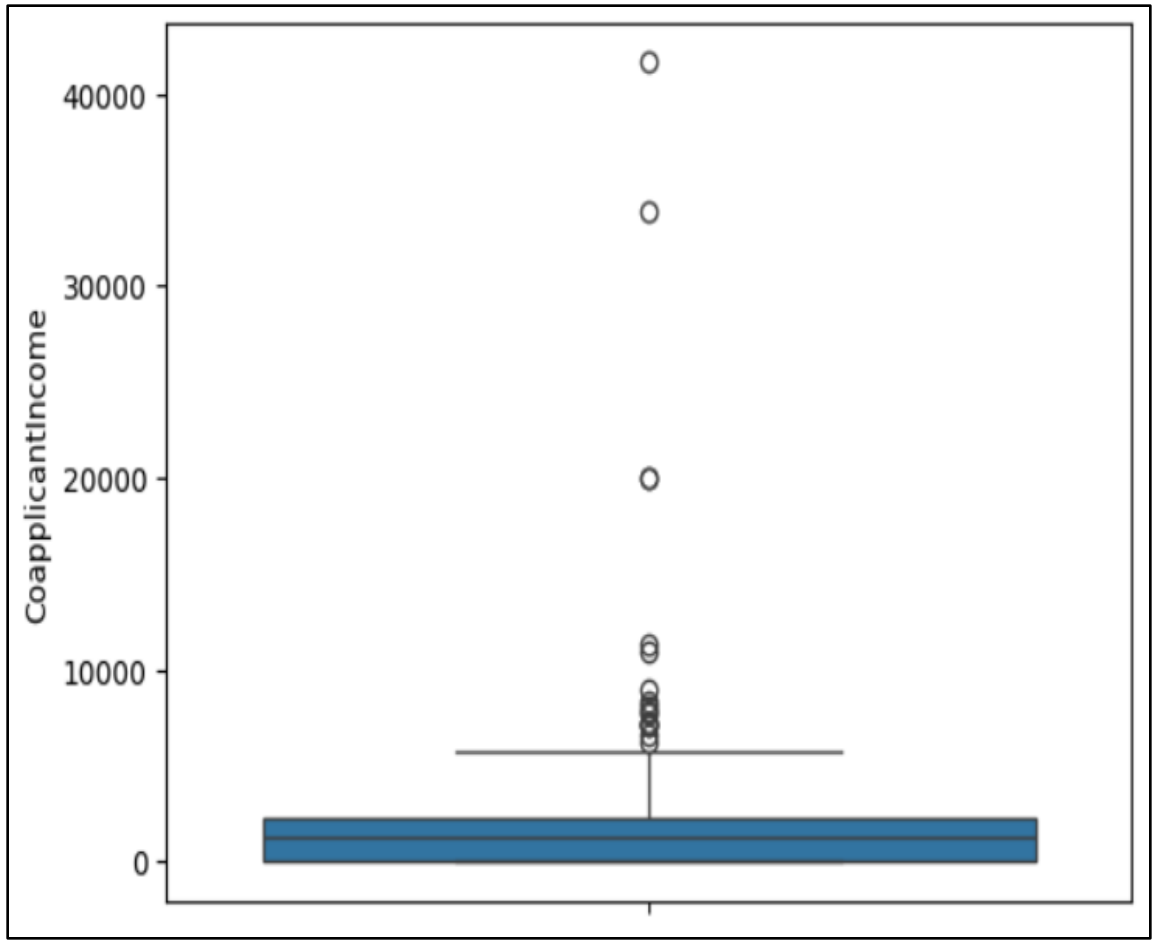
Handling outliers

As observed in correlation, columns “Applicant Income” and “Coapplicant Income” are relevant and has higher correlational significance in dataset. That’s why we only drop rows from these two features to remove outliers.

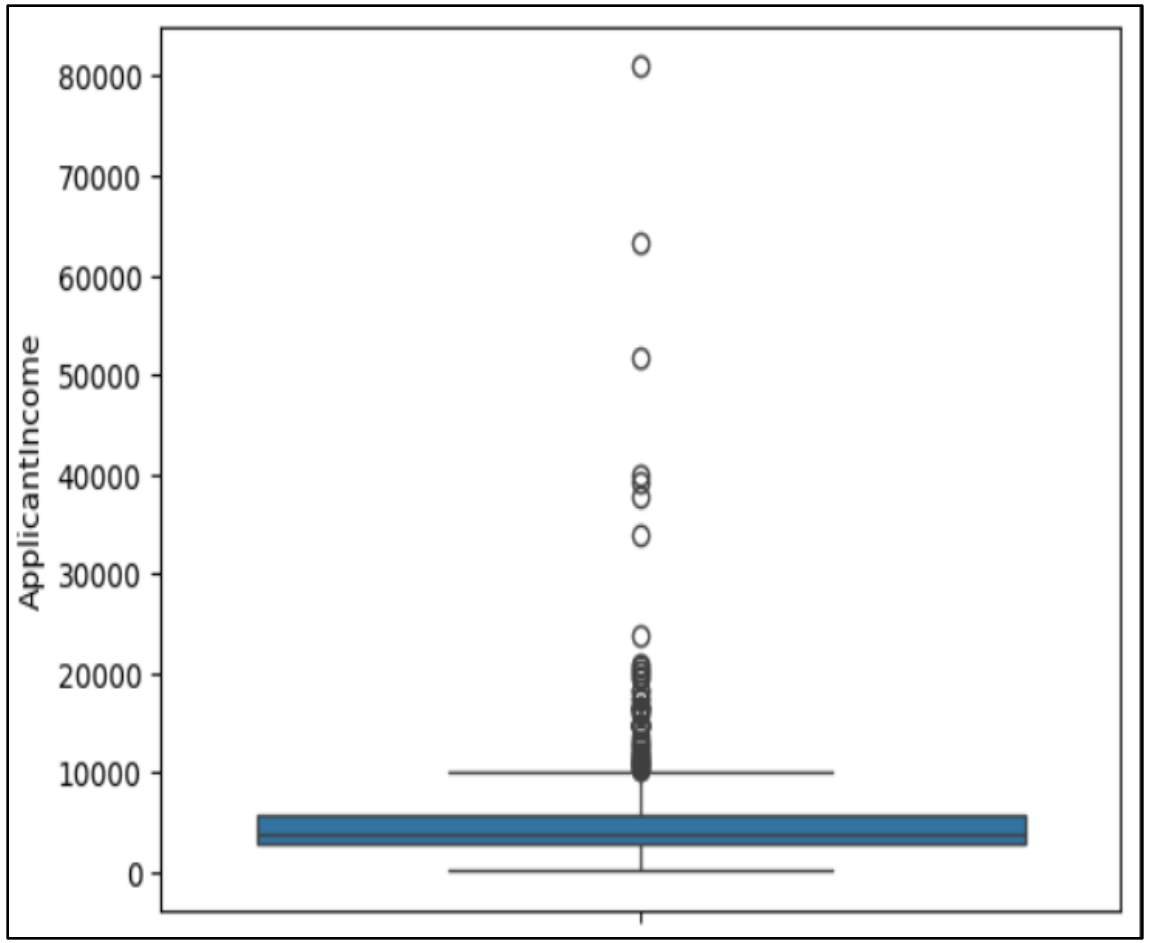
	Dependents	ApplicantIncome	CoapplicantIncome	LoanAmount	Loan_Amount_Term	Credit_History
Dependents	1.000000	0.118901	0.028788	0.163692	-0.103619	-0.038736
ApplicantIncome	0.118901	1.000000	-0.116266	0.566037	-0.044361	-0.018900
CoapplicantIncome	0.028788	-0.116266	1.000000	0.188151	-0.059135	0.012093
LoanAmount	0.163692	0.566037	0.188151	1.000000	0.036135	-0.001631
Loan_Amount_Term	-0.103619	-0.044361	-0.059135	0.036135	1.000000	-0.005291
Credit_History	-0.038736	-0.018900	0.012093	-0.001631	-0.005291	1.000000

Outliers

CoApplicant's Income outliers



Applicant's Income outliers



To Find Outliers

- To find fences to identify outliers I used Interquartile range(IQR) tells you the range of the middle half of your dataset. You can use the IQR to create “fences” around your data and then define outliers as any values that fall outside those fences.

Calculate IQR	→	IQR = Q3 – Q1
Calculate your upper fence	→	Q3 + (1.5 * IQR)
Calculate your lower fence	→	Q1 – (1.5 * IQR)

- All values that fall outside these fences are outliers.

Outliers

Applicant Income Outliers

- Q3,Q1 = 5790,2875
- IQR = 5790 – 2875 = 2915
- UF = 5790 + (1.5 * 2915) = 10162
- LF = 2875 – (1.5 * 2915) = 1497.5
- Dropped the rows where

10162 < applicant's income < 1497.5

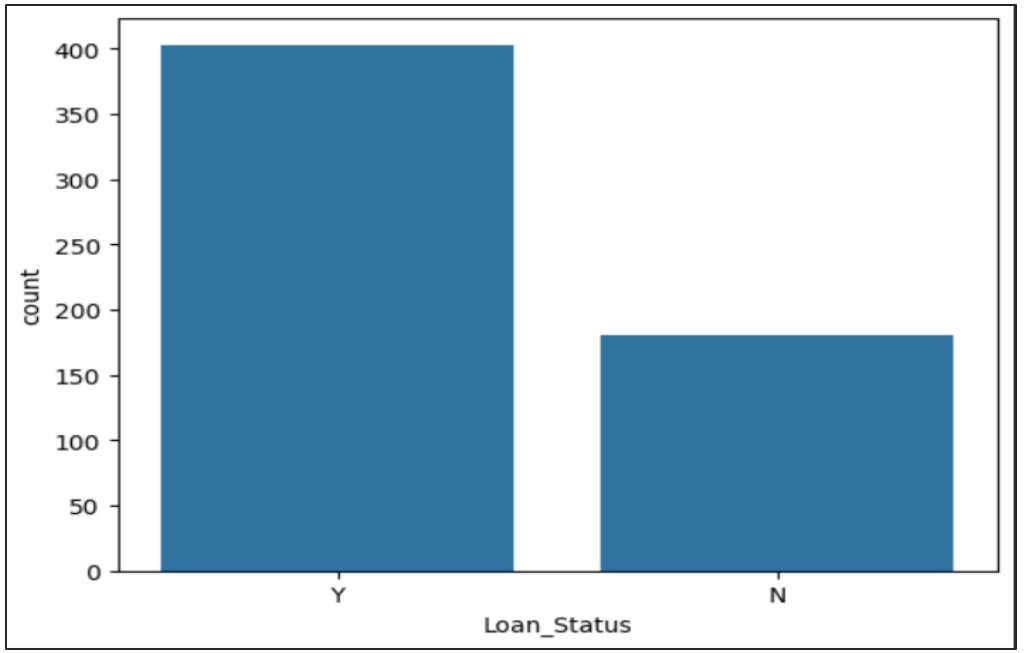
Co-Applicant Income Outliers

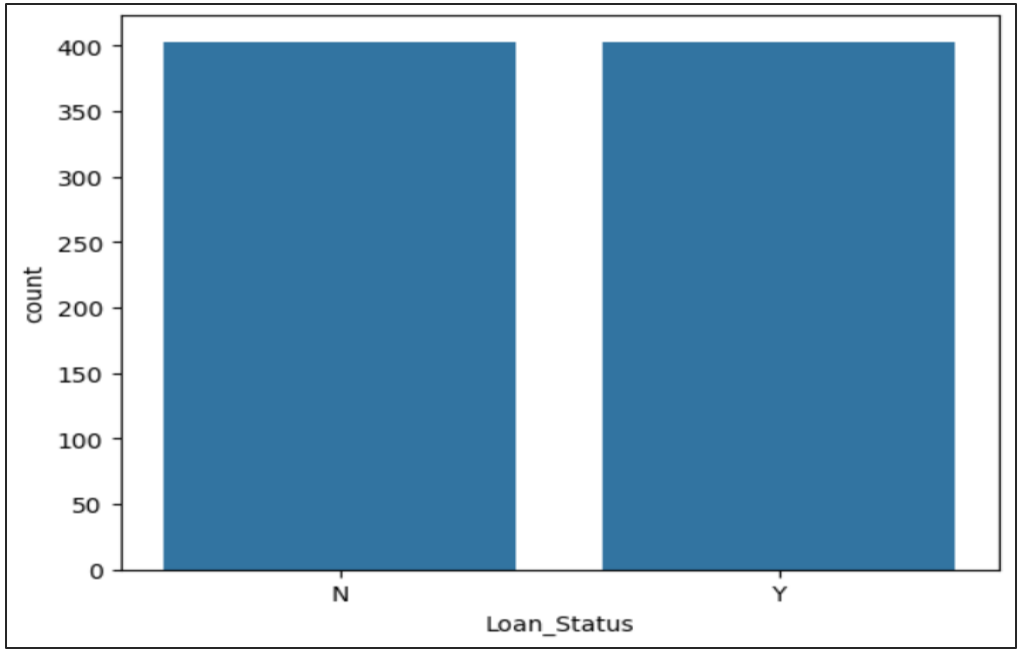
- Q3, Q1 = 2318, 0
- IQR = 2318 – 0 = 2318
- UF = 2318 + (1.5 * 2318) = 5795
- LF = 0 – (1.5 * 2318) = 3477
- Dropped the rows where

5795< Coapplicant's income < 3477

Handling unbalanced data

- In this dataset, distribution of Y and N classes in Loan Status is unbalanced.
- This imbalance data can lower the accuracy of classification predictions.
- To handle this unbalanced data I used resample from sklearn.utils
- Resampling technique is used to upsample or downsample the minority or majority class.
- I upsampled the data so that the minority class matches with the majority class.
- After resampling both Y and N are equally distributed.





Encoding, Splitting and Scaling data

- Using label encoder I encoded features Gender, Married, Education, Self-employed, and property area.
- Then split data into dependent and independent variables x and y.
- Then applying train_test_split to x and y variables xtrain, xtest, ytrain,ytest are obtained for training and testing.
- Apply Scaling on xtrain and xtest. I used standard scaler for scaling.

Applying Algorithms

- This is a Binary classification problem. I applied various classification algorithms such as KNN classifier, KNN using gridsearch with cross-validation, RandomForest, SVC, Logistic Regression.
- Then check their accuracy with classification Report and Confusion Matrix applied on predictions on xtest and ytest data
- Each algorithm gives various accuracy percentage by examining these percentage and confusion matrix we can decide best algorithm for deolyment.

- KNeighborsClassifier

```
[[61 20]
 [14 67]]
              precision    recall  f1-score   support

           N       0.81      0.75      0.78        81
           Y       0.77      0.83      0.80        81

    accuracy                           0.79       162
   macro avg       0.79      0.79      0.79       162
weighted avg       0.79      0.79      0.79       162
```

- Random Forest Classifier

```
[[75  6]
 [ 5 76]]
              precision    recall  f1-score   support

           N       0.94      0.93      0.93        81
           Y       0.93      0.94      0.93        81

    accuracy                           0.93       162
   macro avg       0.93      0.93      0.93       162
weighted avg       0.93      0.93      0.93       162
```

- KNeighborsClassifier with cross validation using Grid Search

```
[[74  7]
 [11 70]]
              precision    recall  f1-score   support

           N       0.87      0.91      0.89        81
           Y       0.91      0.86      0.89        81

    accuracy                           0.89       162
   macro avg       0.89      0.89      0.89       162
weighted avg       0.89      0.89      0.89       162
```

- Support Vector Machine

```
[[44 37]
 [13 68]]
              precision    recall  f1-score   support

           N       0.77      0.54      0.64        81
           Y       0.65      0.84      0.73        81

    accuracy                           0.69       162
   macro avg       0.71      0.69      0.68       162
weighted avg       0.71      0.69      0.68       162
```

Trained Models

- Overall all the models did perform well in classification but Random Forest is performing the best.
  - KNN Classifier – 0.79
  - KNN with cross-validation using GridSearch – 0.89
  - Random Forest – 0.93
  - SVC – 0.69
  - Logistic Regression – 0.77

- SVC model has performed poorly compared to other models.

Deployment

- I trained the model using random Forest as it gives highest accuracy in all algorithms.
- Streamlit is used for the deployment of Machine Learning model.
- For that I wrote a python file in which all the previously mentioned steps are written and then a pickle file is created for object serialization. Pickle serialization saves objects into a file in byte format while deserialization is the reverse of serialization.
- Then a simple function for accepting inputs and predict output prediction.
- To take inputs from a user and display output a HTML page is coded.
- Using streamlit functions deployed the model.