Loan Default Prediction Using Machine Learning

By Susmitha Kondamudi

About Project

The Data set contains the customers information through which we have to predict the Default status by building a Machine Learning Model

Step-By-Step Process

The Data set has to be processed for any null values, outliers, skewness

the Data has....
Gender 208
Employment_Status 94

The null values in Employment Status will be dropped and the null values of Gender will be filled using prediction model

As Gender is a Class point, we train

Classification Model

With accuracy evaluation I found

Decision tree classifier

to be the best fit Model

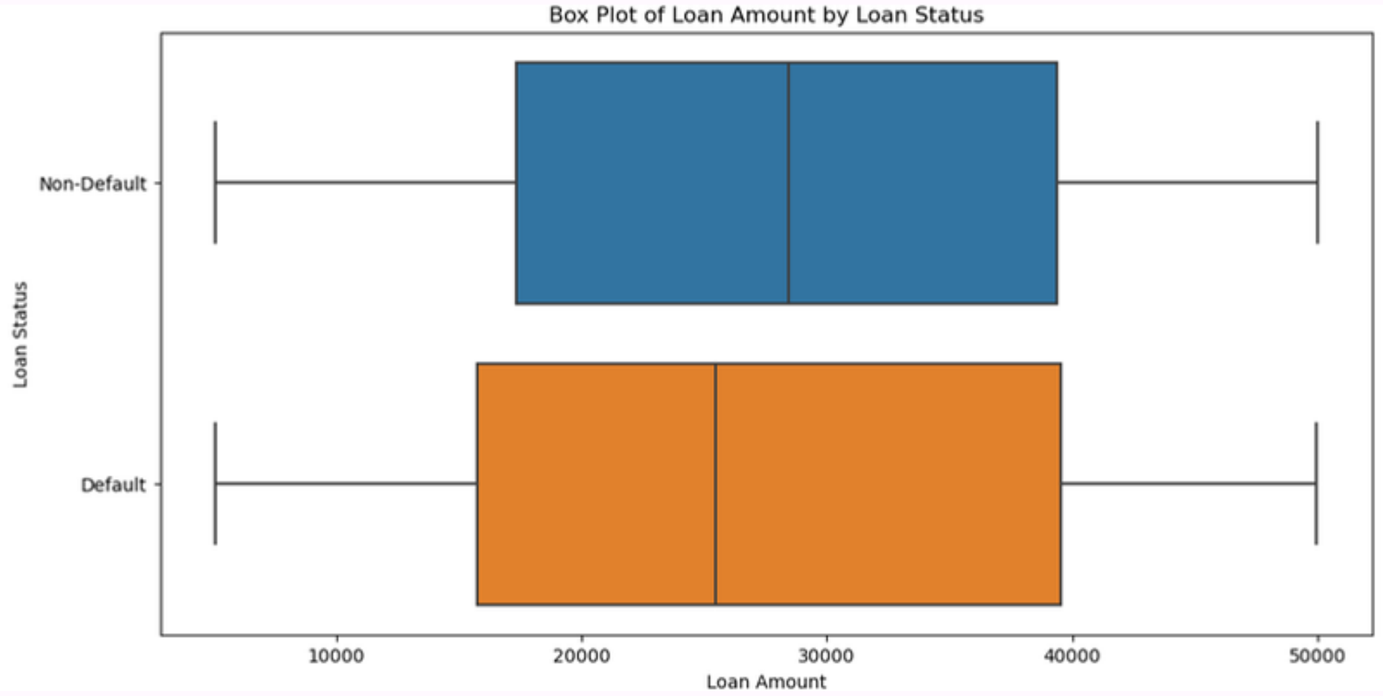
filling Nan values of Gender

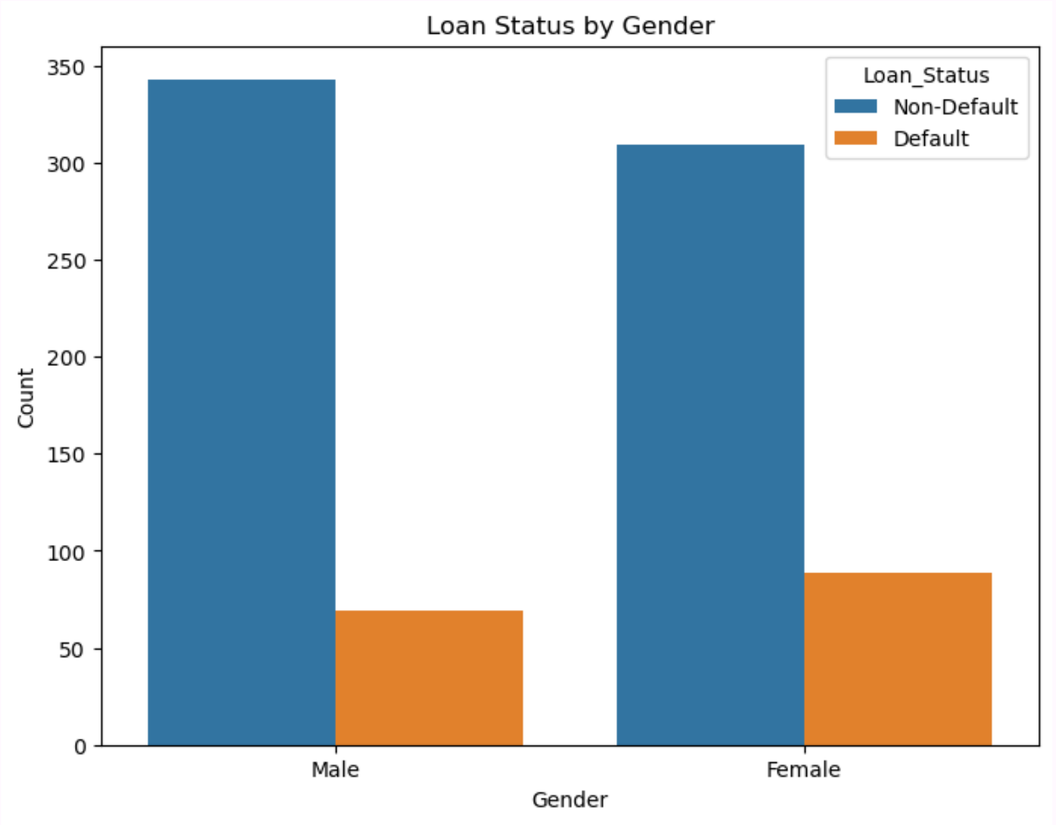
```
# Predict missing values
predicted_values = dt.predict(X_pred)

# Fill missing values in prediction data with predicted values
prediction_data['Gender'] = predicted_values

# Concatenate training and prediction data
filled_data = pd.concat([train_data, predictio
```

EDA


Box Plot of Loan Amount by Loan Status
Non-Default
Default
Loan Status
10000
20000
30000
40000
50000
Loan Amount

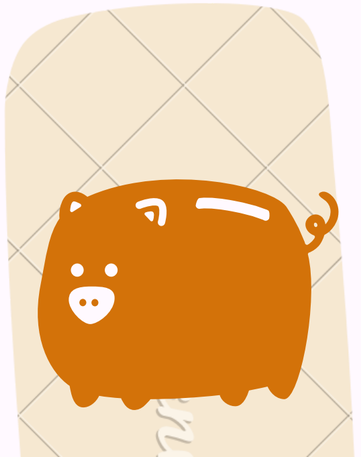

Loan Status by Gender
Loan_Status
Non-Default
Default
350
300
250
200
150
100
50
0
Count
Male
Female
Gender

Build a Classification Model to predict Loan Default Status

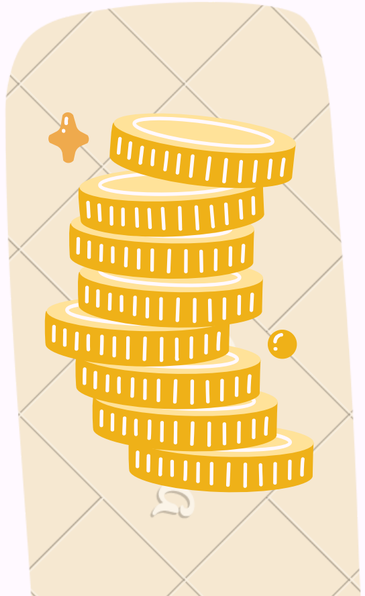
Default status as dependent variable and following as independent variables

Credit Score, Age, Gender

Debt_to_Income_ Ratio



Income



Employment Status

Interest_Rate, Loan Duration Months

Gradient Boosting Classifier is found to be the best fit for prediction

```
new_sample = np.array([[55, 15000, 250, 0.62, 23999, 57466, 12.13
                        'Male', 'Employed', 'Urban']])
    new_sample_gender = ohe.transform(new_sample[:, [8]]).toarray(
new_sample_employeeStatus = ohe2.transform(new_sample[:, [9]]).to
    new_sample_Location = ohe3.transform(new_sample[:, [10]]).toarray()
  new_sample = np.concatenate((new_sample[:, [0, 1, 2, 3, 4, 5, 6, 7]],
new_sample_gender, new_sample_employeeStatus, new_sample_Location), axis=1)
        new_sample_scaled = scaler.transform(new_sample)
            new_pred = gb.predict(new_sample_scaled)
    decoded_pred = decode_labels(new_pred, label_encoder)

    print('Loan_Default_prediction is:', decoded_pred)
```

Pickle the model and build a streamlit application

THANK YOU