Loan Approval Prediction

Bhargava Gaggainpali

DSC680 - Applied Data Science

bgaggainpali/bgaggainpali_DSC680 (github.com)

Business problem & Hypothesis

To purchase a house, or a car we always look forward to take a loan from the bank. Taking a loan on many other things has become part of daily life and we need to know the important factors how the whole process work through the process.

In financial industry, Loan deciding on a loan application, whether to approve or not is a two-edged sword. Bank should not lose business by denying a legitimate customer, who can repay. Also, it should not approve loan to in-eligible customer. Banks are playing important role in challenging times like now, with COVID pandemic across the globe.

I have selected the topic, as I was interested in knowing the process of loan approval and the key factors in it. As I explore more about the domain, I understand that it's not same set of rules which is being followed across domain. And each subdomain like Home loan approval, Car loan approval and such classification have basic approval structure, but the factors which influence are different.

Solution Method

I see this problem as a classification issue, where we should try to understand and able to predict the customers, who have high Loan approval eligibility. Supervised machine learning algorithm to work on the classification problem to be trained with algorithms like:

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

Start with loading data into a data frame and then understand the data, then perform Exploratory Data Analysis (EDA) on the data set. EDA involves doing Univariate and

Bivariate Analysis, identify missing values and outliers and fill the gaps with appropriate values. In the next step, building model with starting from logistic regression and observe the accuracy of the model. When the accuracy of the of the model is not high, then planning to use Decision Tree and Random Forest to achieve higher accuracy.

Technical approach involves understanding the data by drawing multiple bar charts to observe the target variable with respect to each of the variable. Build a heat map to understand the relationship between variables. Build model using the different algorithms and observe the accuracy of the model, evaluate the accuracy of the model by building confusion matrix.

Data

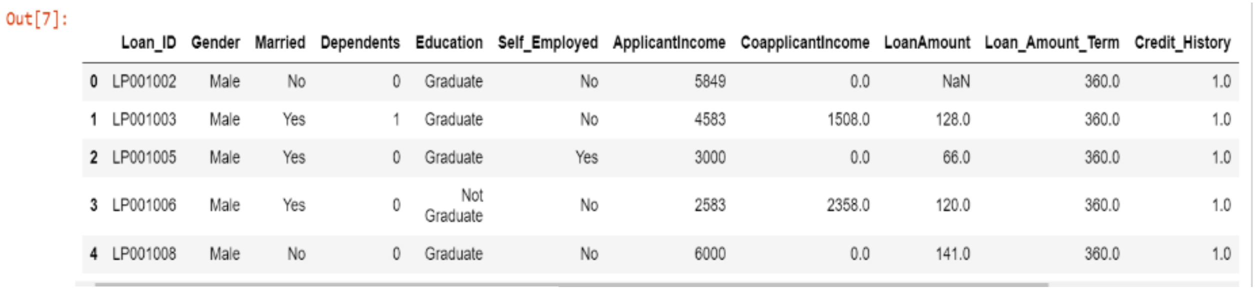
I have identified LoanApprovalPrediction.csv as source for my work, below is the Kaggle link. Initially when I looked at the dataset, I had questions about the variables as they are more of general kind and was guessing that if would serve my purpose of analysis. when closely observed the stats, it surprised me as the amount of value we can retrieve from such data. I am satisfied with the dataset which I have selected.

https://www.kaggle.com/premptk/loan-approval-prediction-model/data?select=LoanApprovalPrediction.csv

Variable	**Description**
Loan_ID	Unique Loan ID
Gender	Male/ Female
Married	Applicant married (Y/N)
Dependents	Number of dependents
Education	Applicant Education (Graduate/ Under Graduate)

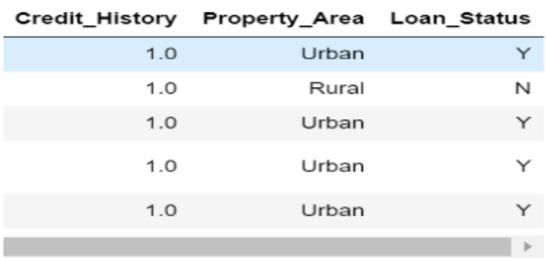
Self_Employed	Self employed (Y/N)
ApplicantIncome	Applicant income
CoapplicantIncome	Coapplicant income
LoanAmount	Loan amount in thousands
Loan_Amount_Term	Term of loan in months
Credit_History	credit history meets guidelines
Property_Area	Urban/ Semi Urban/ Rural
Loan_Status	Loan approved (Y/N)

Initial Observations

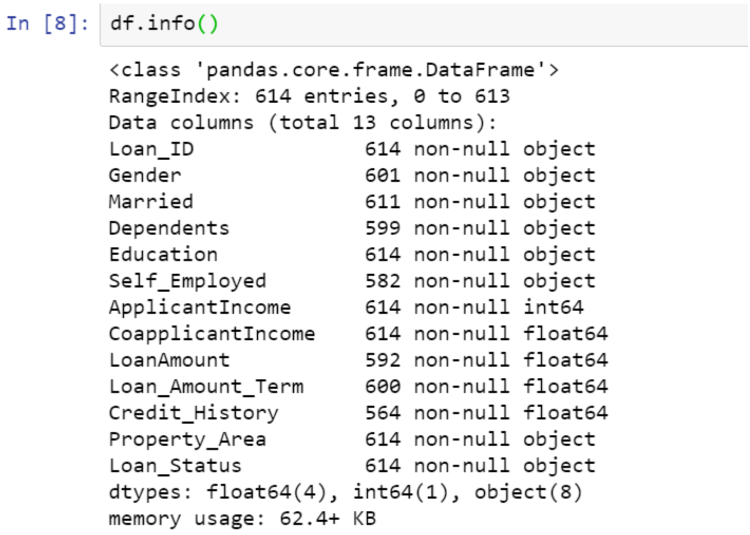


Out[7]:

	Loan_ID	Gender	Married	Dependents	Education	Self_Employed	ApplicantIncome	CoapplicantIncome	LoanAmount	Loan_Amount_Term	Credit_History
0	LP001002	Male	No	0	Graduate	No	5849	0.0	NaN	360.0	1.0
1	LP001003	Male	Yes	1	Graduate	No	4583	1508.0	128.0	360.0	1.0
2	LP001005	Male	Yes	0	Graduate	Yes	3000	0.0	66.0	360.0	1.0
3	LP001006	Male	Yes	0	Not Graduate	No	2583	2358.0	120.0	360.0	1.0
4	LP001008	Male	No	0	Graduate	No	6000	0.0	141.0	360.0	1.0



Credit_History	Property_Area	Loan_Status
1.0	Urban	Y
1.0	Rural	N
1.0	Urban	Y
1.0	Urban	Y
1.0	Urban	Y



```
In [8]: df.info()

        <class 'pandas.core.frame.DataFrame'>
        RangeIndex: 614 entries, 0 to 613
        Data columns (total 13 columns):
        Loan_ID              614 non-null object
        Gender               601 non-null object
        Married              611 non-null object
        Dependents           599 non-null object
        Education            614 non-null object
        Self_Employed        582 non-null object
        ApplicantIncome      614 non-null int64
        CoapplicantIncome    614 non-null float64
        LoanAmount           592 non-null float64
        Loan_Amount_Term     600 non-null float64
        Credit_History       564 non-null float64
        Property_Area        614 non-null object
        Loan_Status          614 non-null object
        dtypes: float64(4), int64(1), object(8)
        memory usage: 62.4+ KB
```

Categorical Features: Based on the data, (Yes/No or Male/Female) below are categorical variables.

Gender, Married,

Self_Employed,

Credit_History,

Loan_Status.

Ordinal Features: Based on the data with inherent hierarchy, below are ordinal variables.

Dependents,

Education and

Property_Area

Numerical Features: Based on the numerical data, below are numerical variables.

ApplicantIncome,

Co-applicantIncome,

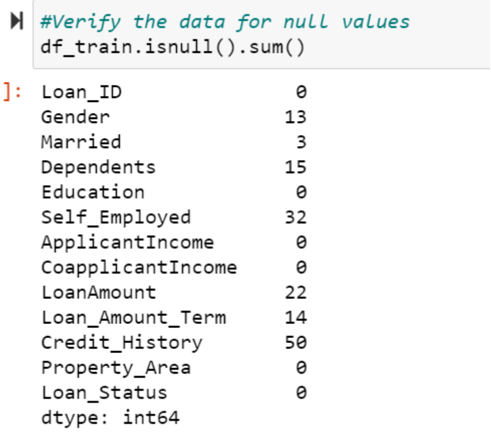
LoanAmount,

Loan_Amount_Term

Data Cleaning and Handling Missing Values

Below is the summary on missing values in each of the variables, Credit_History is the variable with maximum 50 missing values, then Self_Employed has 32 missing values.

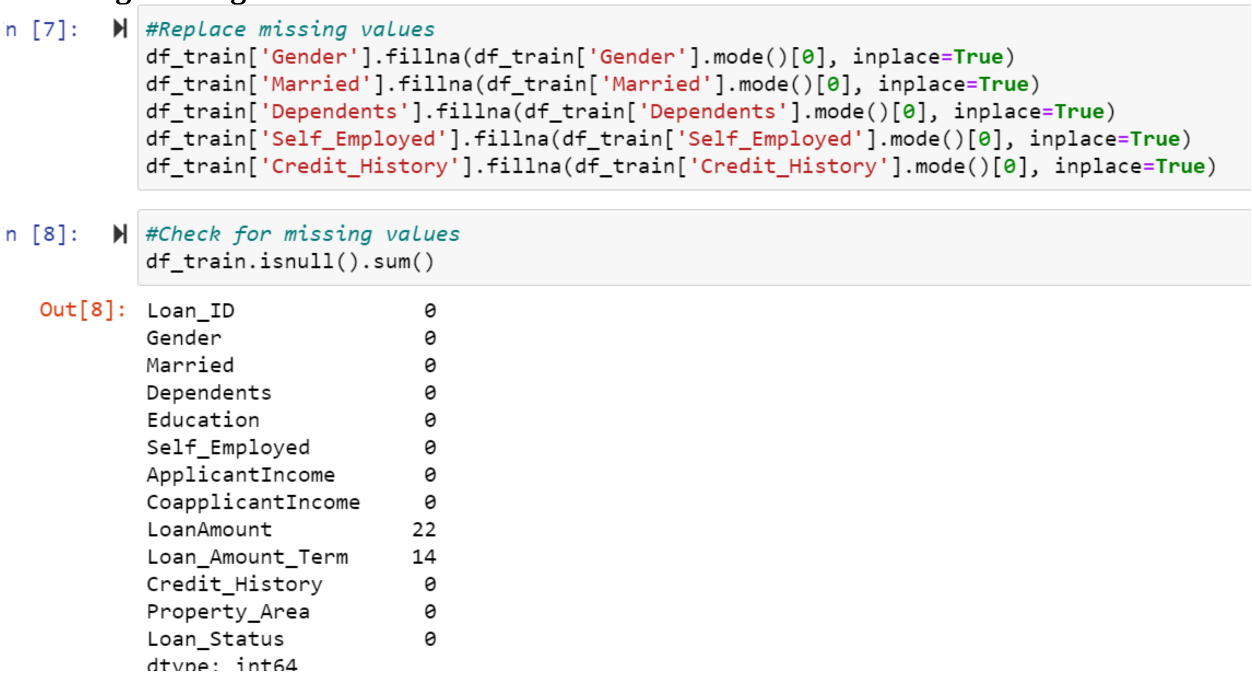
Missing Values in data, before handling:



```
#Verify the data for null values
df_train.isnull().sum()
```

```
]: Loan_ID              0
   Gender              13
   Married              3
   Dependents          15
   Education            0
   Self_Employed       32
   ApplicantIncome      0
   CoapplicantIncome    0
   LoanAmount          22
   Loan_Amount_Term    14
   Credit_History      50
   Property_Area        0
   Loan_Status          0
   dtype: int64
```

Handling Missing values:



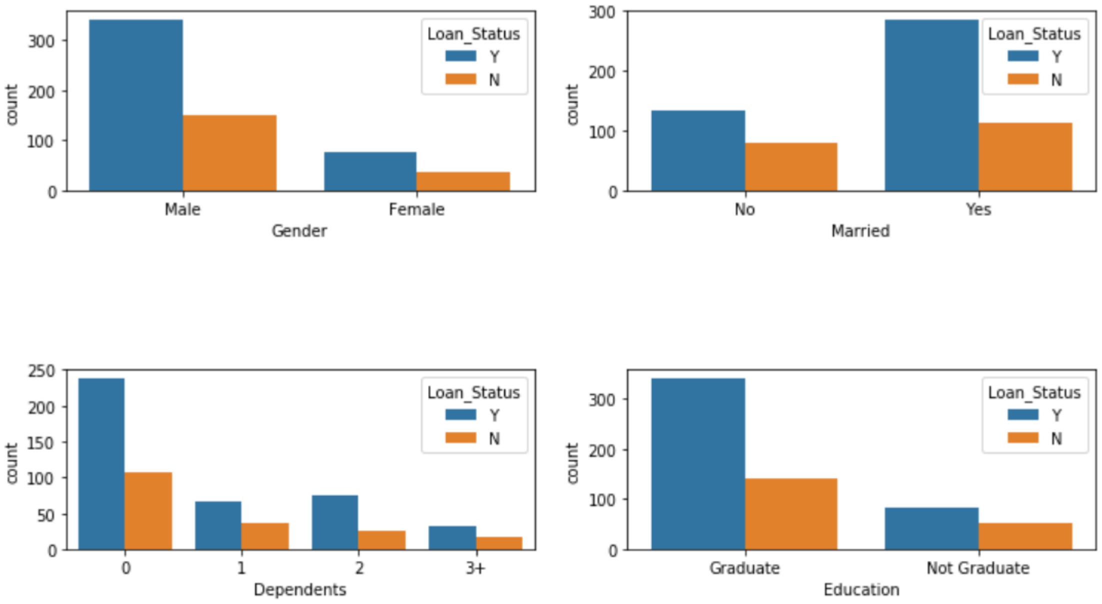
```
n [7]:  #Replace missing values
        df_train['Gender'].fillna(df_train['Gender'].mode()[0], inplace=True)
        df_train['Married'].fillna(df_train['Married'].mode()[0], inplace=True)
        df_train['Dependents'].fillna(df_train['Dependents'].mode()[0], inplace=True)
        df_train['Self_Employed'].fillna(df_train['Self_Employed'].mode()[0], inplace=True)
        df_train['Credit_History'].fillna(df_train['Credit_History'].mode()[0], inplace=True)
```

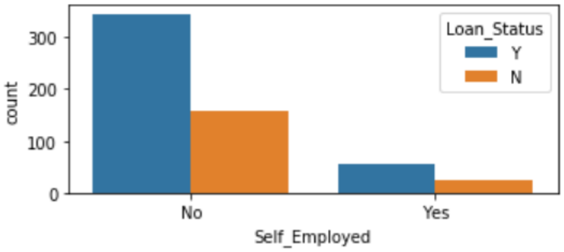
```
n [8]:  #Check for missing values
        df_train.isnull().sum()
```

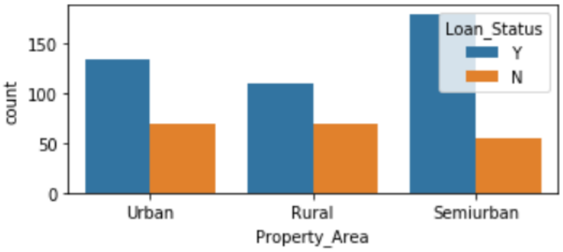
```
Out[8]: Loan_ID              0
        Gender               0
        Married              0
        Dependents           0
        Education            0
        Self_Employed        0
        ApplicantIncome      0
        CoapplicantIncome    0
        LoanAmount          22
        Loan_Amount_Term    14
        Credit_History       0
        Property_Area        0
        Loan_Status          0
        dtype: int64
```

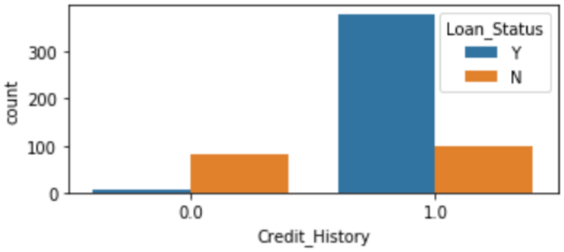
Exploratory Data Analysis

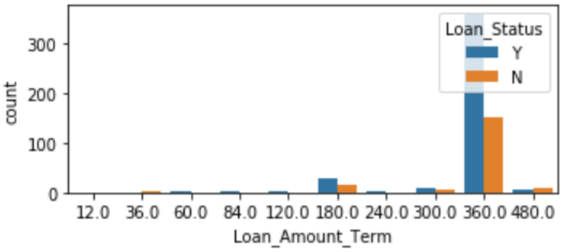
After initial analysis of looking at the dataset values and the basic stats, I had to change my focus on considering many factors. Initially was under the impression that, Loan approval depends on education, applicant income and limited factors. I saw surprising stats when I used visualizations to give clear idea on how each factor has its effect on the Loan approval process. I had to increase my research questions to explore and include more variables, than initially prepared. Its based on the initial analysis using visualization.











Few observations based on the above plots.

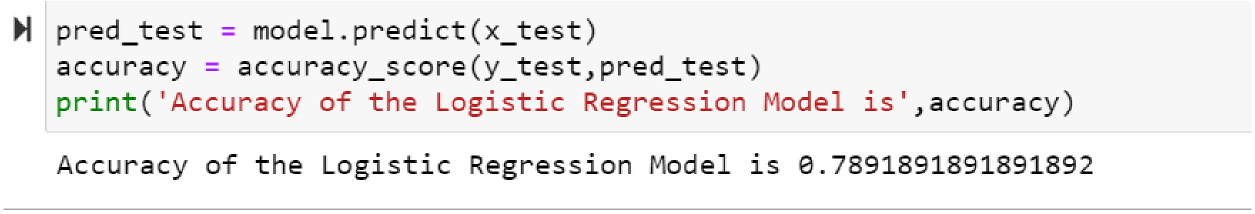
- **Gender**: There are around 300 Men and 100+ Women in the dataset.
- **Marital Status**: Around 65% of the population in the dataset is Marred. Also, married applicants are more likely to be granted loans.
- **Dependents**: Most of the loan applicants have zero dependents and are also likely to accepted for loan.
- **Education**: About 80% of the population is Graduate and graduates have higher probability of loan approval
- **Employment**: 80% of population are not self employed.
- **Property Area**: Majority applicants are from Semi-urban and also likely to be granted loans.
- **Credit_History**: Applicants with credit history are far more likely to be accepted.
- **Loan Amount Term**: Majority of the loans taken are for 360 Months (30 years).

Model Deployment

As part of the current project, four models were developed after data preparation steps. Data is split in the ratio of 70:30 for train and test, i.e. 70% of the data is fed to the model to understand the patterns and remembering the outcome, later 30% of the data is used to validate the prediction results.

Below are four models

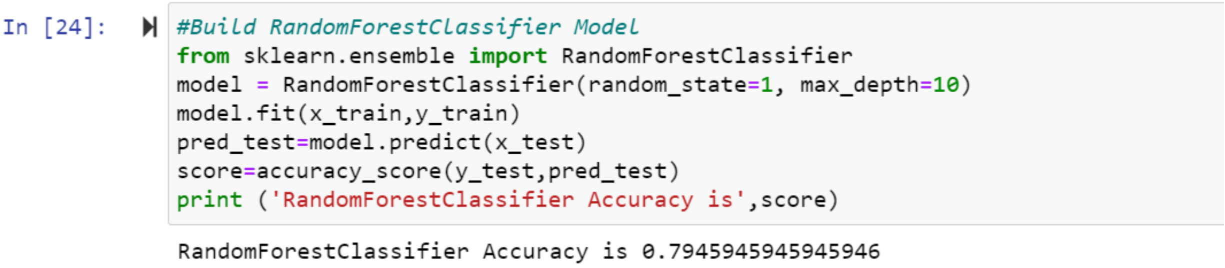
- ***Logistic Regression***



```
pred_test = model.predict(x_test)
accuracy = accuracy_score(y_test,pred_test)
print('Accuracy of the Logistic Regression Model is',accuracy)
```

```
Accuracy of the Logistic Regression Model is 0.7891891891891892
```

- ***Random Forest***

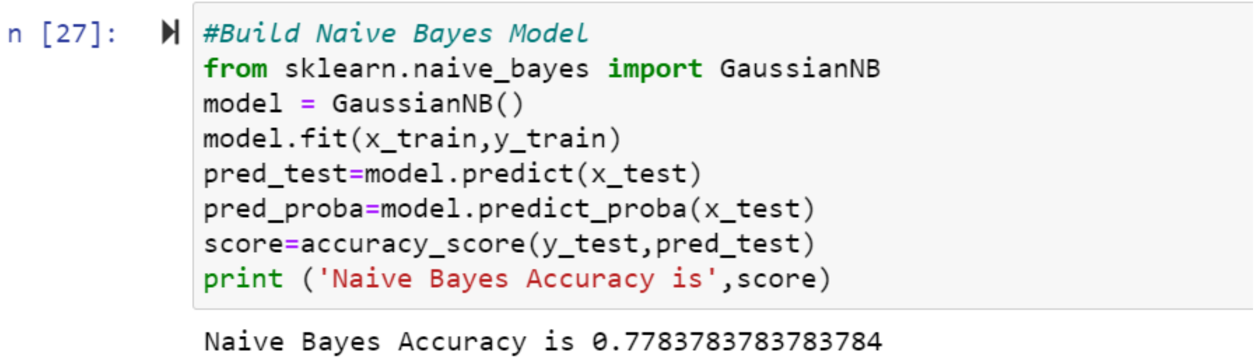


In [24]:

```
#Build RandomForestClassifier Model
from sklearn.ensemble import RandomForestClassifier
model = RandomForestClassifier(random_state=1, max_depth=10)
model.fit(x_train,y_train)
pred_test=model.predict(x_test)
score=accuracy_score(y_test,pred_test)
print ('RandomForestClassifier Accuracy is',score)
```

```
RandomForestClassifier Accuracy is 0.7945945945945946
```

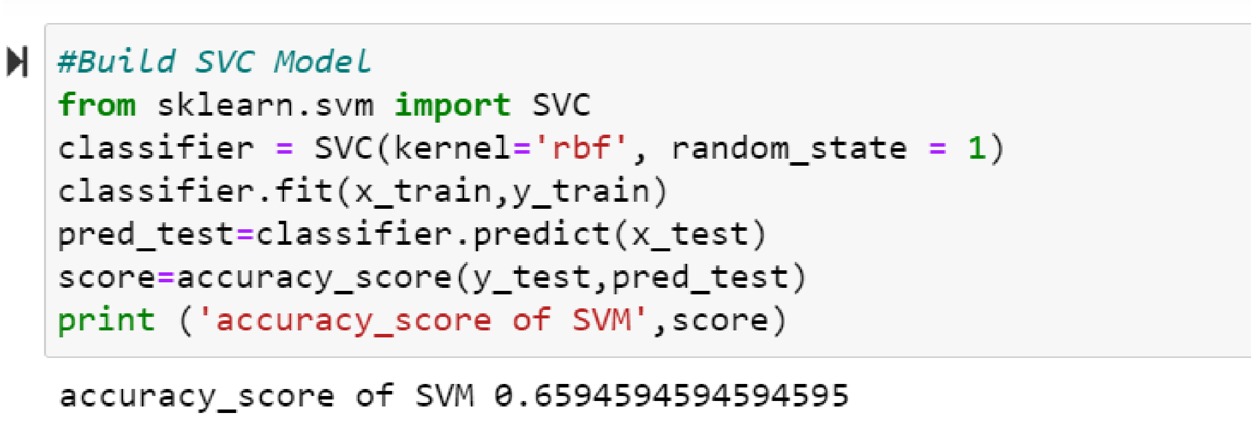
- ***Naïve Bayes***



```
#Build Naive Bayes Model
from sklearn.naive_bayes import GaussianNB
model = GaussianNB()
model.fit(x_train,y_train)
pred_test=model.predict(x_test)
pred_proba=model.predict_proba(x_test)
score=accuracy_score(y_test,pred_test)
print ('Naive Bayes Accuracy is',score)
```

```
Naive Bayes Accuracy is 0.7783783783783784
```

- ***SVM***

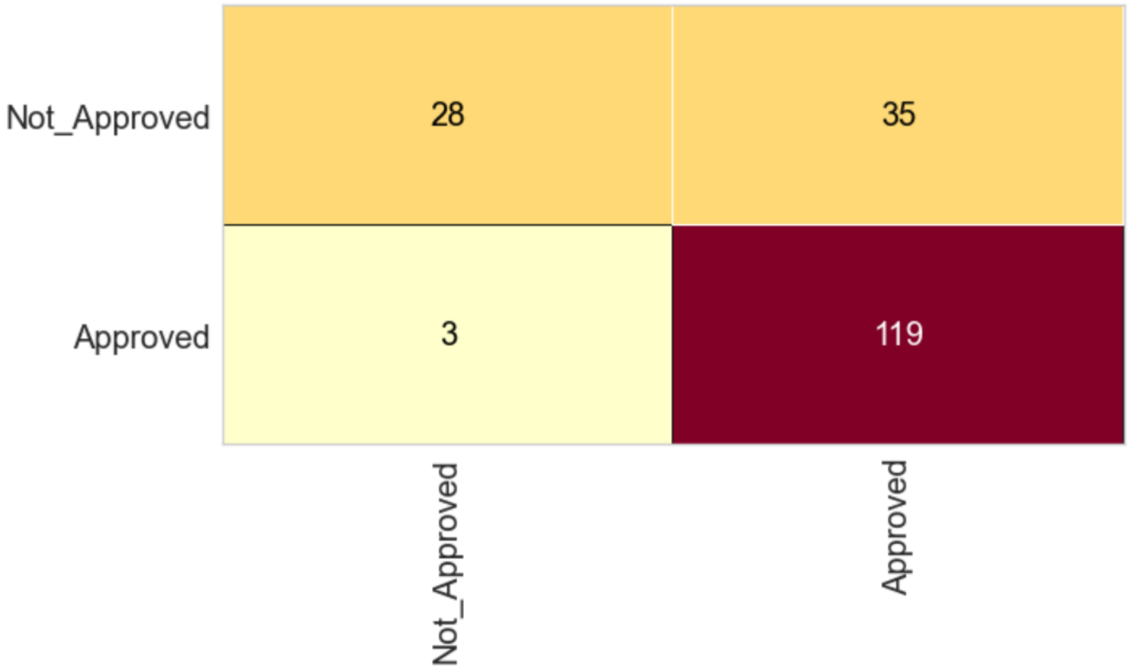


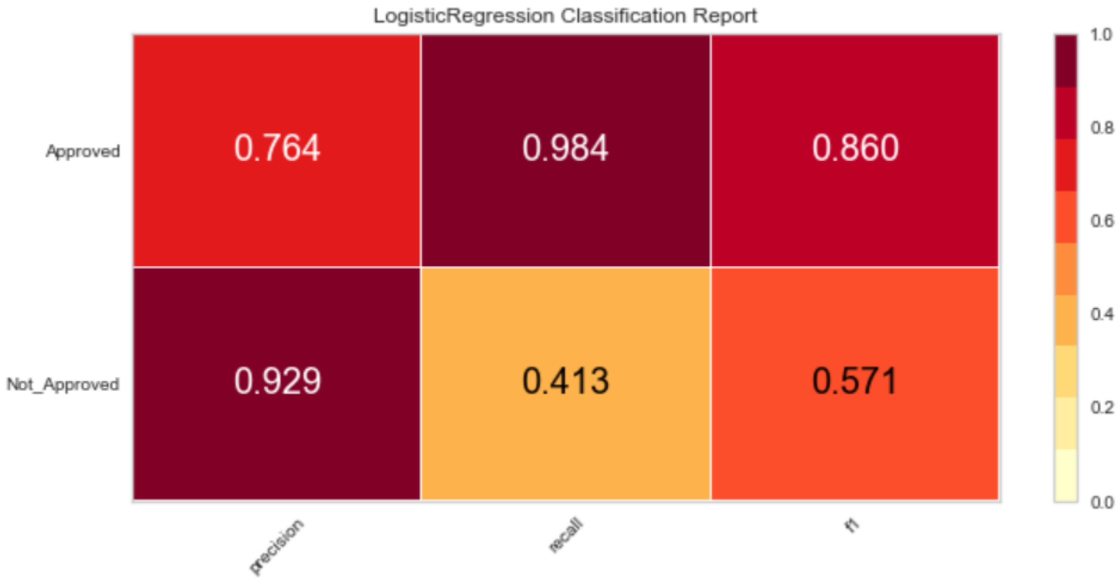
```
#Build SVC Model
from sklearn.svm import SVC
classifier = SVC(kernel='rbf', random_state = 1)
classifier.fit(x_train,y_train)
pred_test=classifier.predict(x_test)
score=accuracy_score(y_test,pred_test)
print ('accuracy_score of SVM',score)
```

```
accuracy_score of SVM 0.6594594594594595
```

Testing and Evaluation

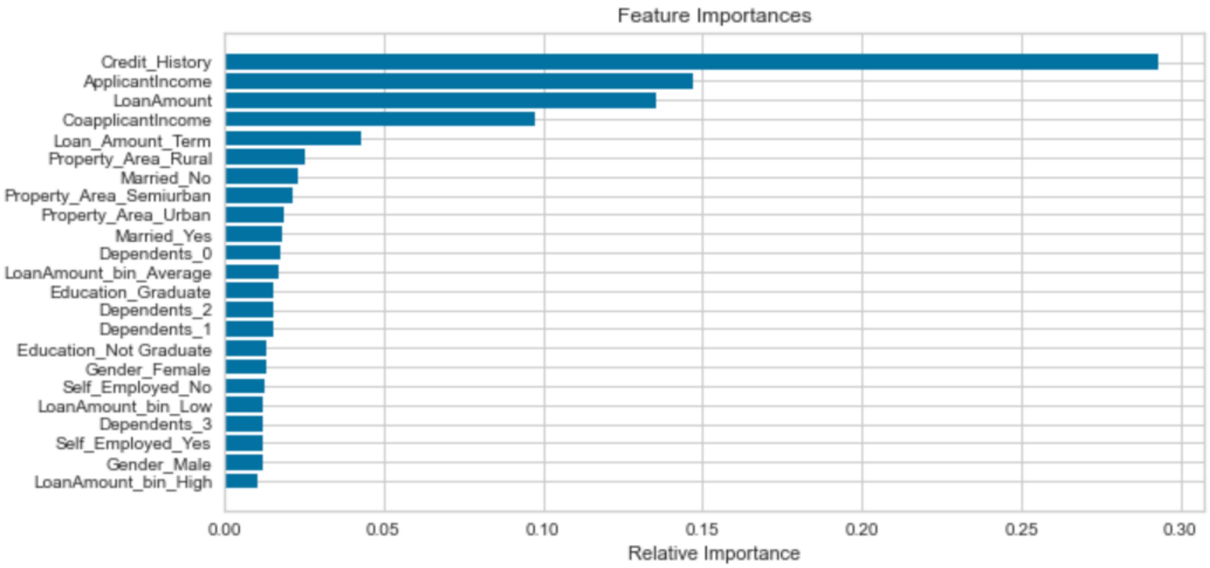
After completing Model building using different algorithms, evaluate the accuracy of the model by building confusion matrix. As part of this project confusion matrix is built for each of the models, below is confusion matrix built on Logistic Regression Model.


Not_Approved
Approved
28
35
3
119
Not_Approved
Approved


LogisticRegression Classification Report
Approved
Not_Approved
0.764
0.984
0.860
0.929
0.413
0.571
precision
recall
f1
1.0
0.8
0.6
0.4
0.2
0.0

Conclusion

Below chart shows that credit history is most important factor in deciding the loan application followed by applicant income and loan amount.



Future Analysis Questions

I would like to continue my analysis and try to explore further to find answers to the given questions.

1. With the COVID situation in place, are these features still valid to predict the Loan approval process.
2. As with the different type of Loans like Home Loan vs Car Loan, would these features stay in common across industry.
3. Will the feature importance change with respect to geographic location?

Reference:

1. Kumar Arun, Garg Ishan, Kaur Sanmeet – 2016 - IOSR Journal of Computer Engineering (IOSR-JCE) - Loan Approval Prediction based on Machine Learning Approach. 4. 18-21.pdf (iosrjournals.org)

2. Mohammad Ahmad Sheikh; Amit Kumar Goel; Tapas Kumar - 2020 – IEEE - An Approach for Prediction of Loan Approval using Machine Learning Algorithm | IEEE Conference Publication | IEEE Xplore

3. Rajiv Kumar, Vinod Jain, Prem Sagar Sharma, Shashank Awasthi, Gopal Jha – 2019 - Prediction of Loan Approval using Machine Learning | International Journal of Advanced Science and Technology (sersc.org)

4. Rahul Shukla - Sep 19, 2020 - Prediction of Loan Approval with Machine Learning https://medium.com/@rahulshuklawork/prediction-of-loan-approval-with-machine-learning-539cbd2aad31

5. Mridul Bhandari - Sep 14, 2020 - How to predict Loan Eligibility using Machine Learning Models Build predictive models to automate the process of targeting the right applicants.https://towardsdatascience.com/predict-loan-eligibility-using-machine-learning-models-7a14ef904057