A
Project Report
On

LOAN APPROVAL PREDICTION

By

Amrita Chatterjee
(Roll No: 30018021010
Reg No: 213001818010010)

Protyush Jana
(Roll No: 30018021007
Reg No: 213001818010007)

Adrija Karmakar
(Roll No: 30018021017
Reg No: 213001818010017)

Submitted in Partial Fulfillment of the Requirement for the Degree of

MASTER OF SCIENCE IN APPLIED STATISTICS

DEPARTMENT OF APPLIED STATISTICS
MAULANA ABUL KALAM AZAD UNIVERSITY OF TECHNOLOGY (WBUT)
WEST BENGAL, NADIA- 741249
JUNE, 2022

LOAN APPROVAL PREDICTION

Project work submitted by

Amrita Chatterjee

Protyush Jana

Adrija Karmakar

Under the Supervision of

Mr. Taranga Mukherjee

Assistant Professor

Department of Applied Statistics

Maulana Abul Kalam Azad University of Technology (WBUT)

West Bengal, Nadia 741249

DEPARTMENT OF APPLIED STATISTICS

MAULANA ABUL KALAM AZAD UNIVERSITY OF TECHNOLOGY (WBUT)

WEST BENGAL, NADIA- 741249

JUNE, 2022

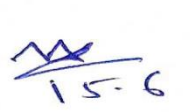
MAULANA ABUL KALAM AZAD UNIVERSITY OF TECHNOLOGY (WBUT)

WEST BENGAL, NADIA- 741249

CERTIFICATE

I hereby forward this project thesis entitled "**LOAN APPROVAL PREDICTION**" by *Amrita Chatterjee* (Roll No: 30018021010; Reg No: 213001818010010), *Protyush Jana* (Roll No: 30018021007; Reg No: 213001818010007) *& Adrija Karmakar* (Roll No: 30018021017; Reg No: 213001818010017)of 2021-23 in partial fulfillment of the requirement for the degree of MASTER IN APPLIED STATISTICS AND ANALYTICS of the DEPARTMENT OF APPLIED STATISTICS, MAULANA ABUL KALAM AZAD UNIVERSITY OF TECHNOLOGY (WBUT), WEST BENGAL, NADIA- 741249.

This project thesis has been completed under my guidance in the Department of Statistics, Maulana Abul Kalam Azad University of Technology (WBUT), West Bengal.



TARANGA MUKHERJEE
Supervisor
Assistant Professor
Department of Applied Statistics
Maulana Abul Kalam Azad University of Technology
West Bengal

Countersigned:

ANWESHA SENGUPTA
Professor & Head of the Department
Department of Applied Statistics
Maulana Abul Kalam Azad University of Technology
West Bengal, Nadia 741249

ACKNOWLEDGEMENT

Acknowledgment is not simply a ritual. It is an expression of heartfelt gratitude and indebtedness to all those who have been associated with the development of the thesis.

I would like to express my profound and deep sense of gratitude to my guide *Prof. (Mr.) Taranga Mukherjee* for his unending help, guidance, and suggestions without which this thesis would not have been a reality. He has acted as a friend, philosopher, and guide to me. I own great indebtedness for his untiring effort thoughtful the period of my project work.

I express my sincere thanks to *Prof. Anwesha Sengupta*, Head of the Department of Applied Science, Maulana Abul Kalam Azad University of Technology, Kalyani, West Bengal, India, for his cooperation extended during the period of my project work.

I am great fully indebted to *Prof. Prasanta Narayan Dutta*, Course Coordinator, and *Prof. Sukhendu Samajdar*, Director, Department of Applied Science, Maulana Abul Kalam Azad University of Technology, Kalyani, West Bengal, India, for their great inspiration and encouragement for my work and for providing me a favorable environment in the form of infrastructural facilities for the research work.

I am greatly indebted to all the faculty members, who often took pains and stood by me in adverse circumstances. Without their encouragement and inspiration, it was not possible for me to complete this project.

Finally, my earnest thanks go to my friends who were always beside me when I needed them without any excuses and made these three years worthwhile.

Amrita Chatterjee

(Amrita Chatterjee)

Protyush Jana

(Protyush Jana)

Adrija Karmakar

(Adrija Karmakar)

CONTENTS

INTRODUCTION

The two most pressing issues in the banking sector are: 1) How risky is the borrower? 2) Should we lend to the borrower given the risk?

The response to the first question dictates the borrower's interest rate. Interest rate, among other things (such as the time value of money), tests the riskiness of the borrower, i.e., the higher the interest rate, the riskier the borrower. We will then decide whether the applicant is suitable for the loan based on the interest rate.

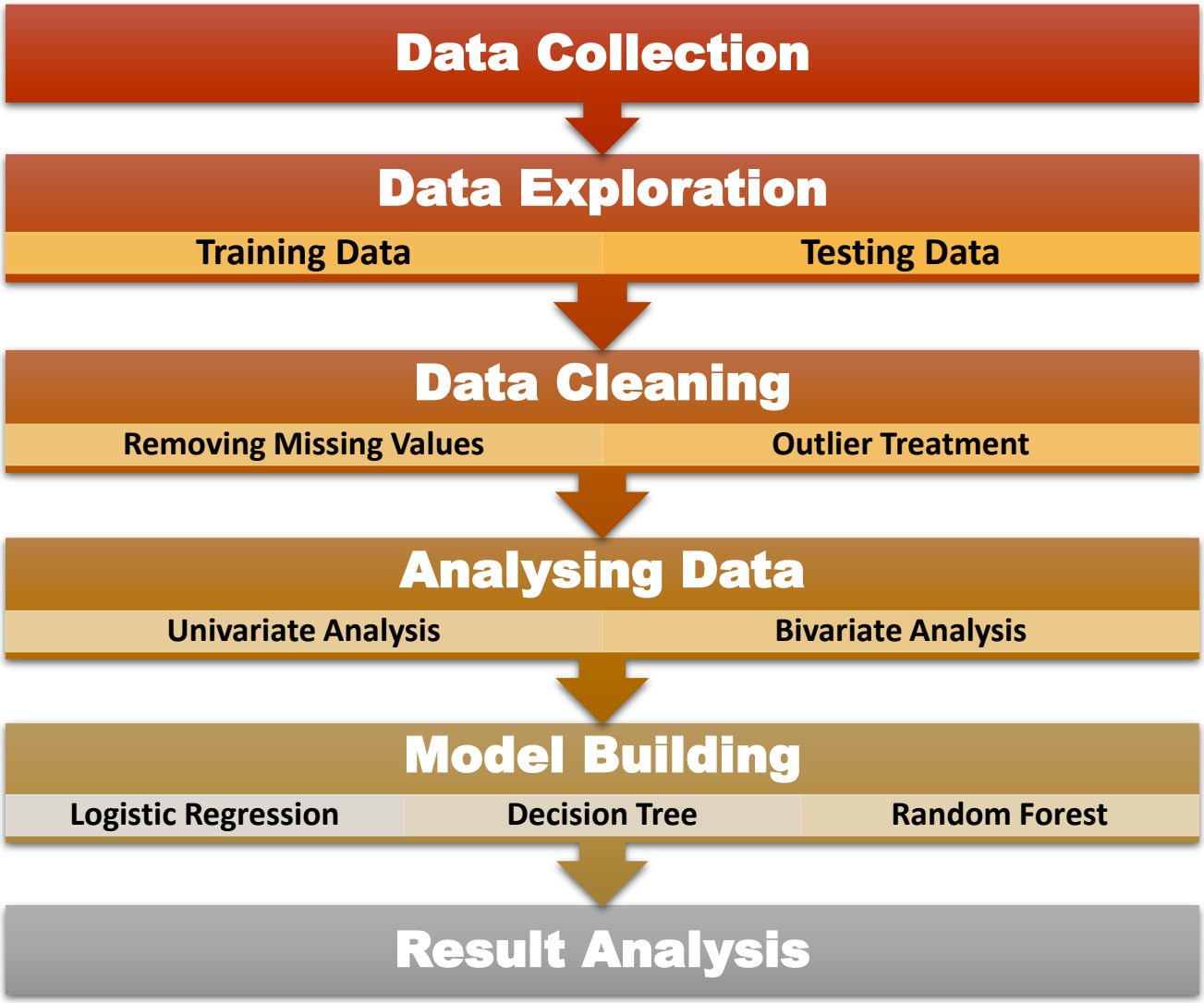
Lenders (investors) make loans to creditors in return for the guarantee of interest-bearing repayment. That is, the lender only makes a return (interest) if the borrower repays the loan. However, whether he or she does not repay the loan, the lender loses money. Banks make loans to customers in exchange for the guarantee of repayment. Some would default on their debts, unable to repay them for several reasons. The bank retains insurance to minimize the possibility of failure in the case of a default. The insured sum can cover the whole loan amount or just a portion of it.

Banking processes use manual procedures to determine whether or not a borrower is suitable for a loan based on results. Manual procedures were most effective, but they were insufficient when there were a large number of loan applications. At that time, making a decision would take a long time.

As a result, the loan prediction machine learning model can be used to assess a customer's loan status and build strategies. This model extracts and introduces the essential features of a borrower that influence the customer's loan status. Finally, it produces the planned performance (loan status). These reports make a bank manager's job simpler and quicker.

This article discusses an approach toward predicting Loan Approval Status by the bank through utilizing basic elementary Data Science and Machine Learning techniques.

METHODOLOGY



ANALYSIS

➢ Data Collection:

Dream Housing Finance company deals in all home loans. They have a presence across all urban, semi-urban and rural areas. Customers first apply for a home loan after that company validates the customer's eligibility for a loan. The company wants to automate the loan eligibility process (real-time) based on customer detail provided while filling out the online application form. These details are Gender, Marital Status, Education, Number of Dependents, Income, Loan Amount, Credit History, and others.

To automate this process, they have given a problem to identify the customer segments, that are eligible for loan amounts so that they can specifically target these customers. Here they have provided a partial data set. We have downloaded the data set from Kaggle. (https://www.kaggle.com/datasets/altruistdelhite04/loan-prediction-problem-dataset)

➢ Data Exploration:

In this step, various Libraries and packages were imported which were required to explore the data. After that, some top rows were looked at a glance. Also, we checked if the dataset contains null values or not.

- **Importing Libraries:** The first and foremost step involves importing necessary libraries and packages.

```
#Loading Packages

from IPython.display import display
import pandas as pd
import numpy as np  #for mathematical calculation
import seaborn as sns   #for data visualization
import matplotlib.pyplot as plt # for plotting graphs
import missingno as msno
%matplotlib inline
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score
from scipy.stats import chi2_contingency
from scipy.stats import chi2
from scipy import stats
from statsmodels.stats import weightstats as stests
import warnings
warnings.filterwarnings("ignore")
```

- **Load Dataset:** After importing all the required libraries, we upload the dataset containing 980 rows and 13 columns.

```
1 #import data
2 data = pd.read_csv("/content/Loan Predicton Data.csv")
3 data
```

	Loan_ID	Gender	Married	Dependents	Education	Self_Employed	Applicant_Income	Coapplicant_Income	Loan_Amount	Loan_Amount_Term	Credit_History	Property_Area	Loan_Status
0	LP001002	Male	No	0.0	Graduate	No	5849	0.0	NaN	360.0	1.0	Urban	Y
1	LP001003	Male	Yes	1.0	Graduate	No	4583	1508.0	128.0	360.0	1.0	Rural	N
2	LP001005	Male	Yes	0.0	Graduate	Yes	3000	0.0	66.0	360.0	1.0	Urban	Y
3	LP001006	Male	Yes	0.0	Not Graduate	No	2583	2358.0	120.0	360.0	1.0	Urban	Y
4	LP001008	Male	No	0.0	Graduate	No	6000	0.0	141.0	360.0	1.0	Urban	Y
...	...	...	...	...	...	...	...	...	...	...	...	...	...
976	LP002971	Male	Yes	3.0	Not Graduate	Yes	4009	1777.0	113.0	360.0	1.0	Urban	Y
977	LP002975	Male	Yes	0.0	Graduate	No	4158	709.0	115.0	360.0	1.0	Urban	Y
978	LP002980	Male	No	0.0	Graduate	No	3250	1993.0	126.0	360.0	NaN	Semiurban	Y
979	LP002986	Male	Yes	0.0	Graduate	No	5000	2393.0	158.0	360.0	1.0	Rural	Y
980	LP002989	Male	No	0.0	Graduate	Yes	9200	0.0	98.0	180.0	1.0	Rural	Y

981 rows × 13 columns

Dataset

Loan_ID	Loan ID for the Applicant applying for a loan
Gender	Gender of the Applicant
Married	Applicant's marital status
Dependents	Number of dependents of the Applicant
Education	Applicant's education status (Graduate/Under Graduate)
Self_Employed	The applicant is self-employed or not
Applicant_Income	Applicant's Income
Coapplicant_Income	Co-applicant's Income
Loan_Amount	Loan Amount in thousands
Loan_Amount_Term	Term of the loan in months
Credit_History	Applicant's previous credit history meeting guidelines
Property_Area	Urban, Semi-Urban, or Rural Areas
Loan_Status	Loan Approval status (Target Variable)

We have 13 features in total out of which we have 12 independent variables and 1 dependent variable i.e., 'Loan_Status' in the dataset, and the 12 independent variables, the 'Loan_ID', 'Gender', 'Married', 'Dependents', 'Education', 'Self_Employed', 'Property_Area', 'Loan_Status' are all categorical.

This is a Binary Classification problem in which we need to predict our Target label which is "Loan Status".

Loan status can have two values: Yes or NO.

Y: if the loan is approved

N: if the loan is not approved

So, we divide the dataset into training (70% of total) and testing data (30% of total), and using the training dataset we will train our model and try to predict our target column which is "Loan Status" on the test dataset.

The machine learning model is trained using the training data set. Every new applicant detail-filled at the time of application form acts as a test data set. Based on the training data sets, the model will predict whether a loan would be approved or not. So, the train and test dataset would have the same columns except for the target column which is "Loan Status".

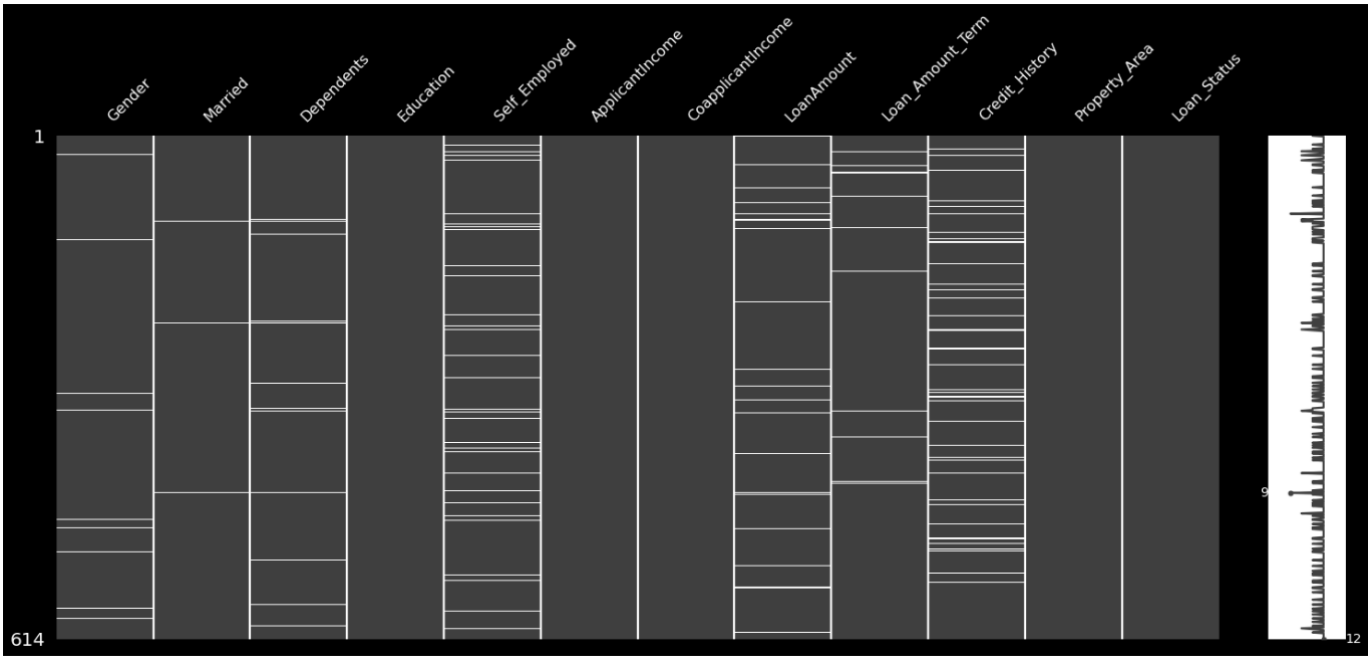
So, we have 614 rows and 13 columns in our training dataset. In the test data, we have 367 rows and 12 columns because the target column is not included in the test data. 'Loan_Status' which is a binary variable that takes the values Yes(Y) / No(N) serves as the Target Variable which needs to be predicted utilizing the test data.

➢ Data Cleaning:

- **Removing Missing Values:** Now, we have to check if there exist any missing values in the columns, we have to impute those with mean or median or mode by understanding the skewness of the data. i.e., if the data is normally distributed, we use mean instead of the missing values and if not, then we use median or mode.

```
1 train.isnull().sum()
```

```
Gender              13
Married              3
Dependents          15
Education            0
Self_Employed       32
Applicant_Income     0
Loan_Amount_Term    14
Credit_History      50
Property_Area        0
Loan_Status          0
Income_bin           0
dtype: int64
```



Missing Data Visualization

As, the data is not normally distributed we replace the missing values with the mode.

```
train['Gender'].fillna(train['Gender'].mode()[0], inplace=True)
train['Married'].fillna(train['Married'].mode()[0], inplace=True)
train['Dependents'].fillna(train['Dependents'].mode()[0], inplace=True)
train['Self_Employed'].fillna(train['Self_Employed'].mode()[0], inplace=True)
train['Credit_History'].fillna(train['Credit_History'].mode()[0], inplace=True)
train['Loan_Amount_Term'].fillna(train['Loan_Amount_Term'].mode()[0], inplace=True)
train['LoanAmount'].fillna(train['LoanAmount'].median(), inplace=True)
```

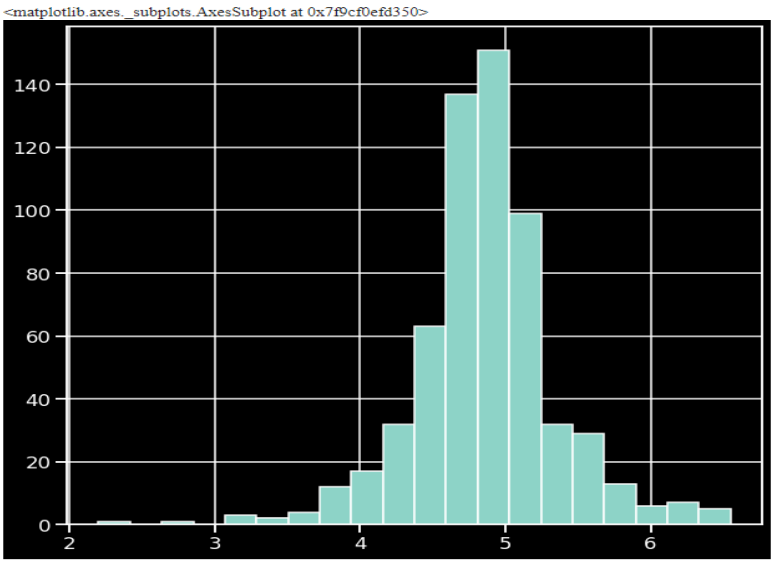
After replacing the missing values, we recheck and saw this:

```
train.isnull().sum()
```

```
Gender             0
Married            0
Dependents         0
Education          0
Self_Employed      0
ApplicantIncome    0
CoapplicantIncome  0
LoanAmount         0
Loan_Amount_Term   0
Credit_History     0
Property_Area      0
Loan_Status        0
dtype: int64
```

- **Outlier Treatment:**

Due to outliers in the Loan Amount. the data in the loan amount is skewed toward the right, which means the bulk of the data is towards the left. We remove this skewness by doing a log transformation. A log transformation doesn't affect the smaller values much but reduces the larger values. So, the distribution becomes normal.



- **Data Analysis:**

At first, we look into the type of our training dataset and analyze descriptively all the details of each column that contains the dataset.

```
1 #type of data
2 train.dtypes
```

```
Loan_ID              object
Gender               object
Married              object
Dependents           object
Education            object
Self_Employed        object
Applicant_Income      int64
Coapplicant_Income  float64
Loan_Amount         float64
Loan_Amount_Term    float64
Credit_History      float64
Property_Area        object
Loan_Status          object
dtype: object
```

	Loan_ID	Gender	Married	Dependents	Education	Self_Employed	Applicant_Income	Coapplicant_Income	Loan_Amount	Loan_Amount_Term	Credit_History	Property_Area	Loan_Status
count	614	601	611	599	614	582	614.000000	614.000000	592.000000	600.00000	564.000000	614	614
unique	614	2	2	4	2	2	NaN	NaN	NaN	NaN	NaN	3	2
top	LP001002	Male	Yes	0	Graduate	No	NaN	NaN	NaN	NaN	NaN	Semiurban	Y
freq	1	489	398	345	480	500	NaN	NaN	NaN	NaN	NaN	233	422
mean	NaN	NaN	NaN	NaN	NaN	NaN	5403.459283	1621.245798	146.412162	342.00000	0.842199	NaN	NaN
std	NaN	NaN	NaN	NaN	NaN	NaN	6109.041673	2926.248369	85.587325	65.12041	0.364878	NaN	NaN
min	NaN	NaN	NaN	NaN	NaN	NaN	150.000000	0.000000	9.000000	12.00000	0.000000	NaN	NaN
25%	NaN	NaN	NaN	NaN	NaN	NaN	2877.500000	0.000000	100.000000	360.00000	1.000000	NaN	NaN
50%	NaN	NaN	NaN	NaN	NaN	NaN	3812.500000	1188.500000	128.000000	360.00000	1.000000	NaN	NaN
75%	NaN	NaN	NaN	NaN	NaN	NaN	5795.000000	2297.250000	168.000000	360.00000	1.000000	NaN	NaN
max	NaN	NaN	NaN	NaN	NaN	NaN	81000.000000	41667.000000	700.000000	480.00000	1.000000	NaN	NaN

Descriptive Analysis

Now, we can perform Exploratory Data Analysis (EDA) on our training dataset.

- **Univariate Visual Analysis:**

We will start first with an independent variable which is our target variable as well. We will analyze this categorical variable using a bar chart as shown below.

Target Variable - Loan Status

```
1 plt.style.use('ggplot')
2 train['Loan_Status'].value_counts().plot.bar(title='Loan Status',rot=0)
3 display(train['Loan_Status'].value_counts())
```

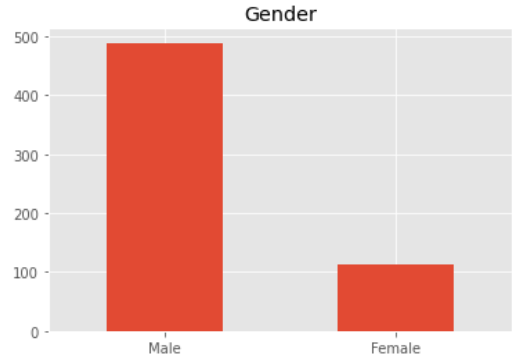
```
Y   422
N   192
Name: Loan_Status, dtype: int64
```

The bar chart shows that loan of 422 (around 69 %) people out of 614 was approved.

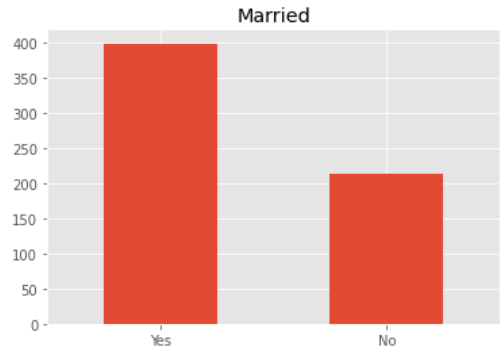
There are 3 types of Independent Variables: Categorical, Ordinal & Numerical.

Categorical Features are Gender, Marital Status, Employment Type, and Credit History.

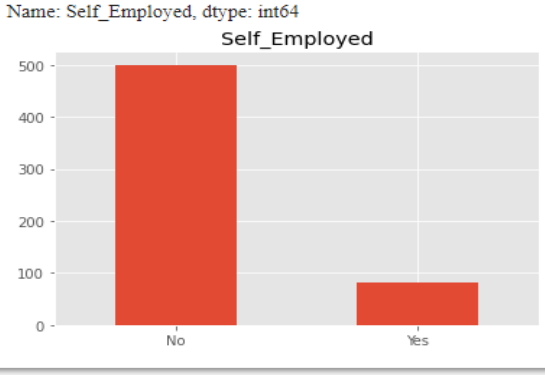
```
Male     489
Female   112
Name: Gender, dtype: int64
```



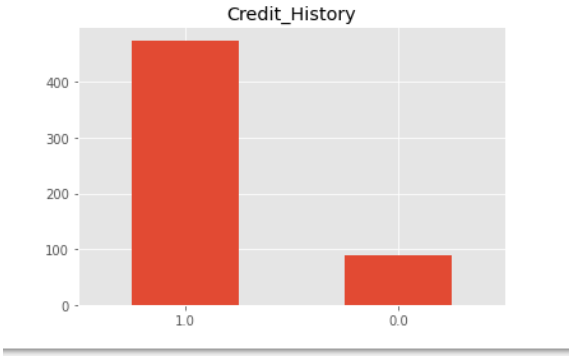

```
Yes   398
No    213
Name: Married, dtype: int64
```




```
No    500
Yes    82
Name: Self_Employed, dtype: int64
```



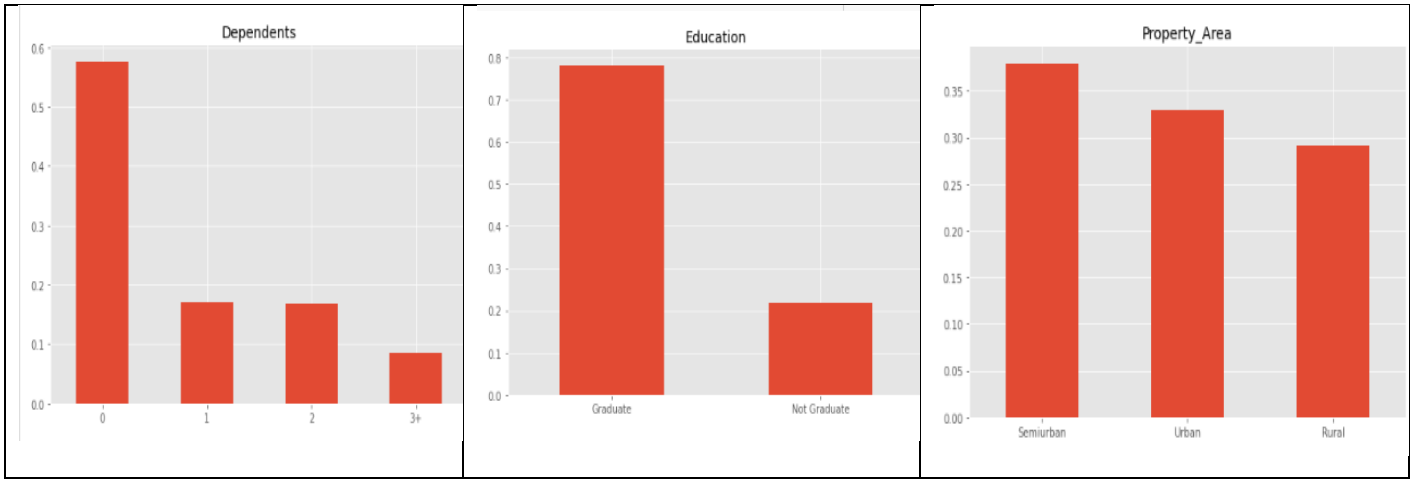

```
1.0   475
0.0    89
Name: Credit_History, dtype: int64
```



It can be inferred from the above bar plots that in our observed data:

- 80% of loan applicants are male in the training dataset.
- Nearly 70% are married.
- Nearly 85–90% of loan applicants are self-employed.
- The loan has been approved for more than 65% of applicants.

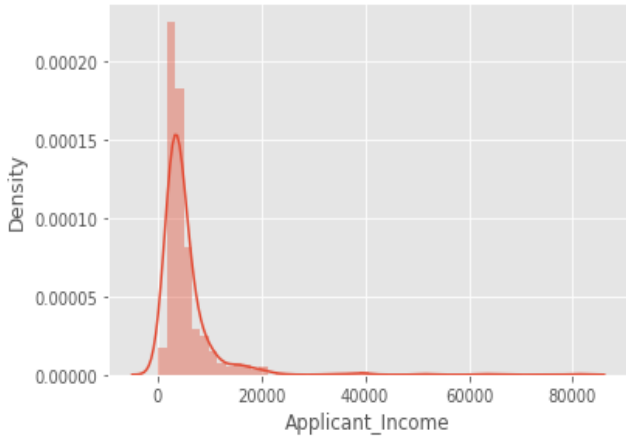
Ordinal Features are the Number of Dependents, Education Level, Property or Area Background.

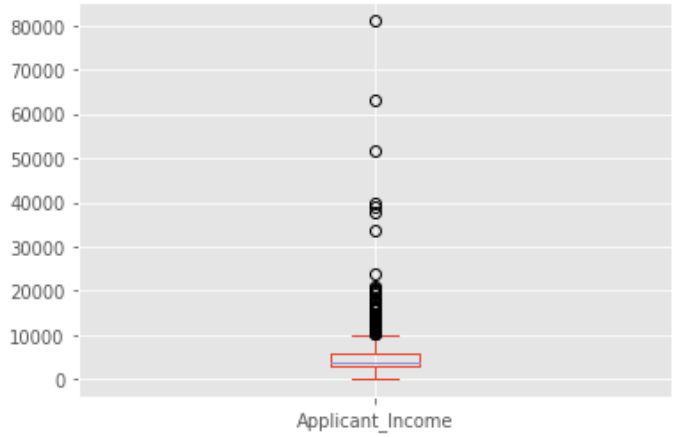


Our Visual Analysis above indicates that:

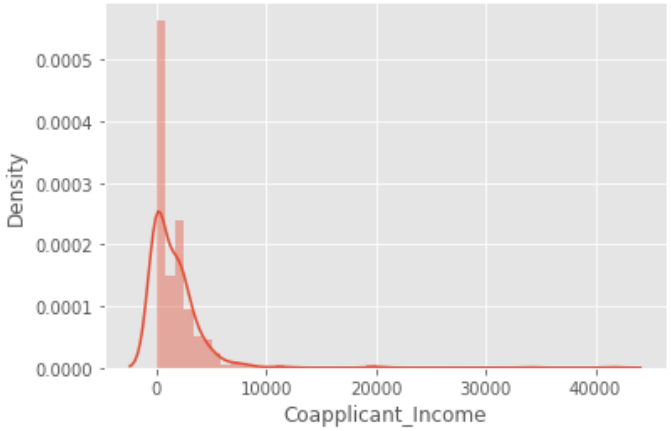
- Almost 58% of the applicants have no dependents.
- Highest number of applicants are from Semi-Urban areas, followed by urban areas.
- Around 80 % of the applicants are graduates.

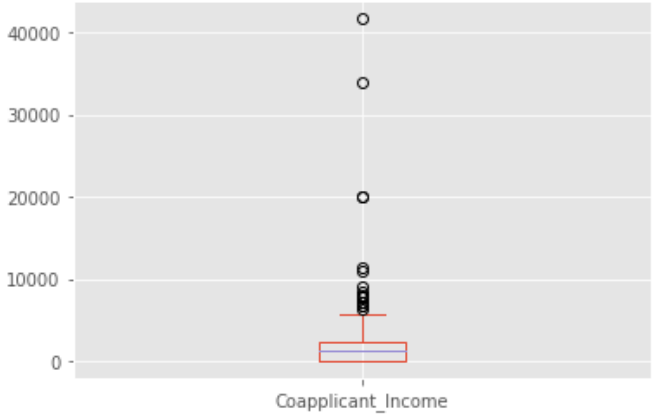
Numerical Features are: The Applicant's Income, The Co-Applicant's Income.





It can be inferred that most of the data in Applicant Income are towards the left which means it is not normally distributed. The boxplot confirms the presence of outliers. This can be attributed to income disparity in society.



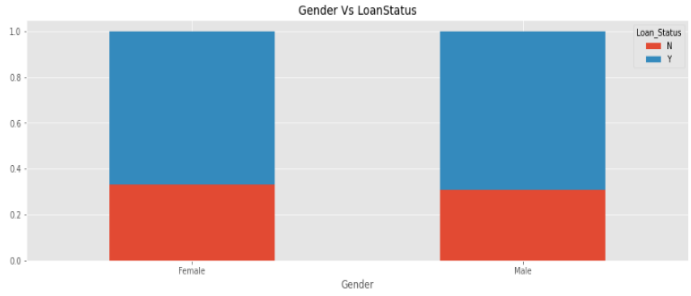


Co-applicant Income is lesser than applicant Income and is within the 5000–15000, again with some outliers.

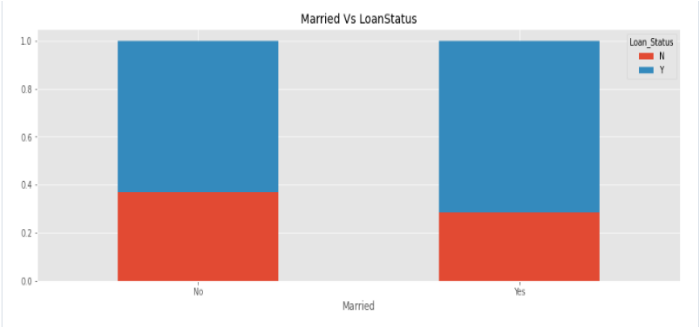
- **Bivariate Visual Analysis:**

Bivariate analysis is finding some kind of empirical relationship between two variables. Specifically, the dependent vs independent Variables.

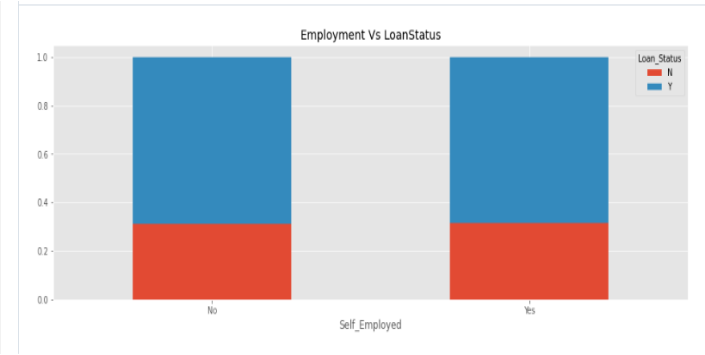
1. **Categorical Independents vs. Target:**



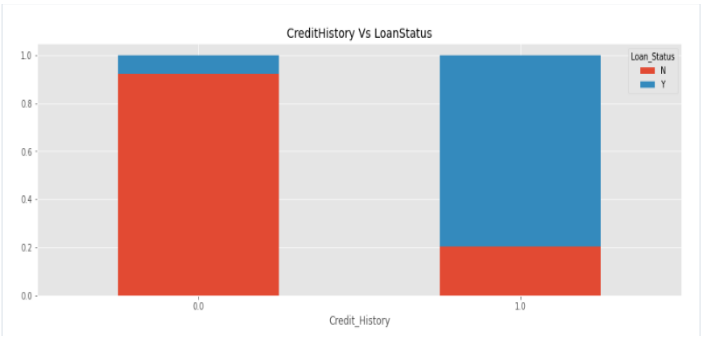
There is not a substantial difference between male and female approval rates.



Married applicants have a slightly higher chances of loan approval.

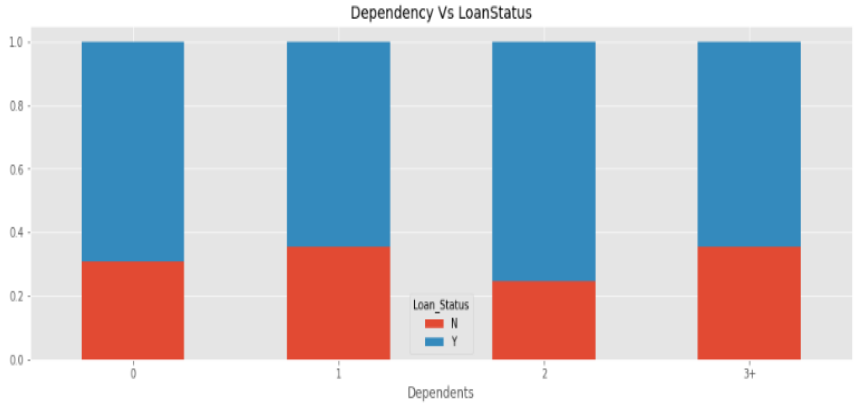


Self-Employed employees have slightly lower chances of loan approval but the situation is not that bad.

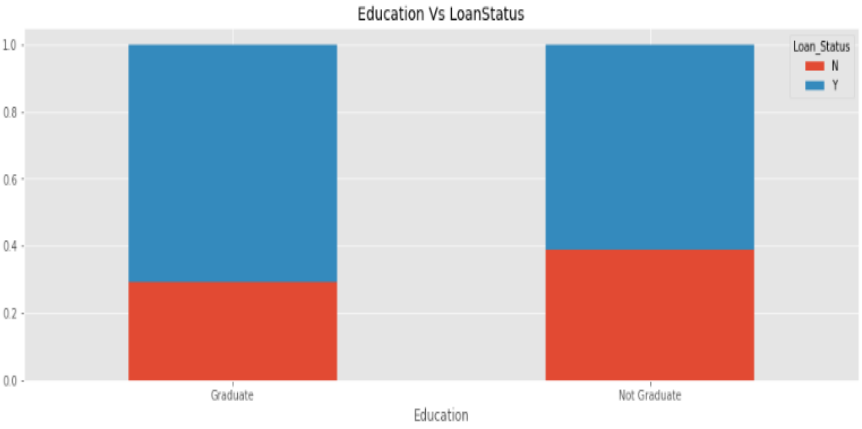


It seems people with credit history as 1 are more likely to get their loans approved

2. Ordinal Independent vs. Target:



Applicants with no dependents or 2 dependents have higher chances of approval. But this does not correlate well.

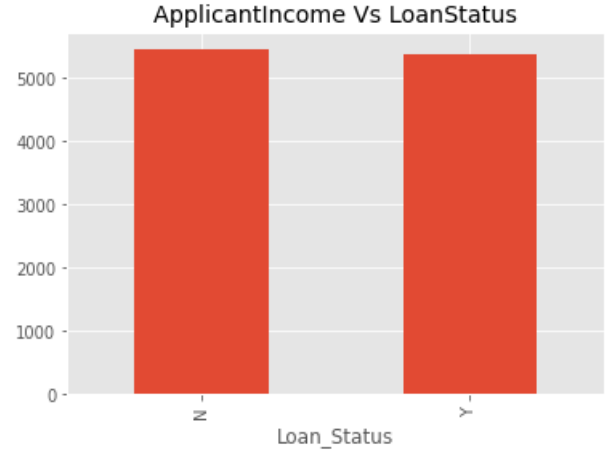


Graduates have higher chance of loan approval compared to non-graduates.

Proportion of loans getting approved in semiurban area is higher as compared to that in rural or urban areas.

3. Numerical Independent vs Target:

We tried to find the mean income of people for which the loan has been approved vs the mean income of people for which the loan has not been approved but we don't see any changes in the mean income.



So, we make bins for the applicant income variable based on the values like:

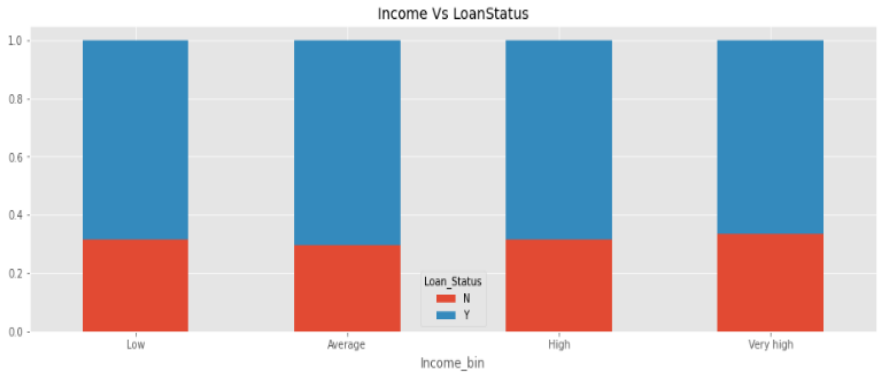
Low: 0-2500

Average: 2500-4000

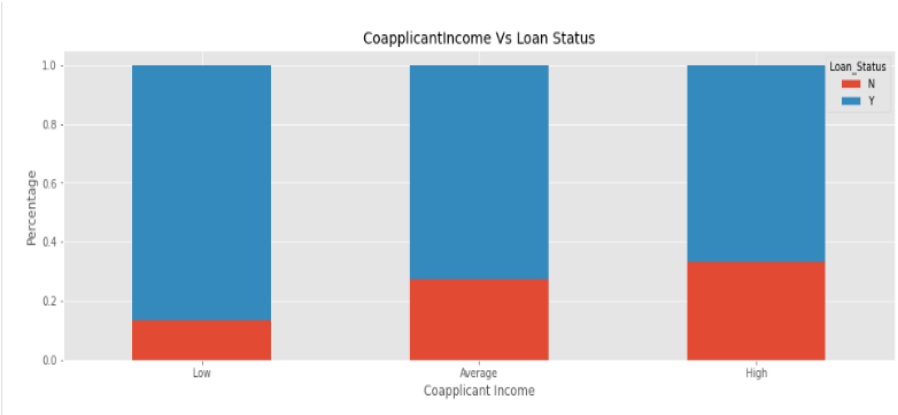
High: 4000-6000

Very High: 6000-81000

and analyze the corresponding loan status for each bin.



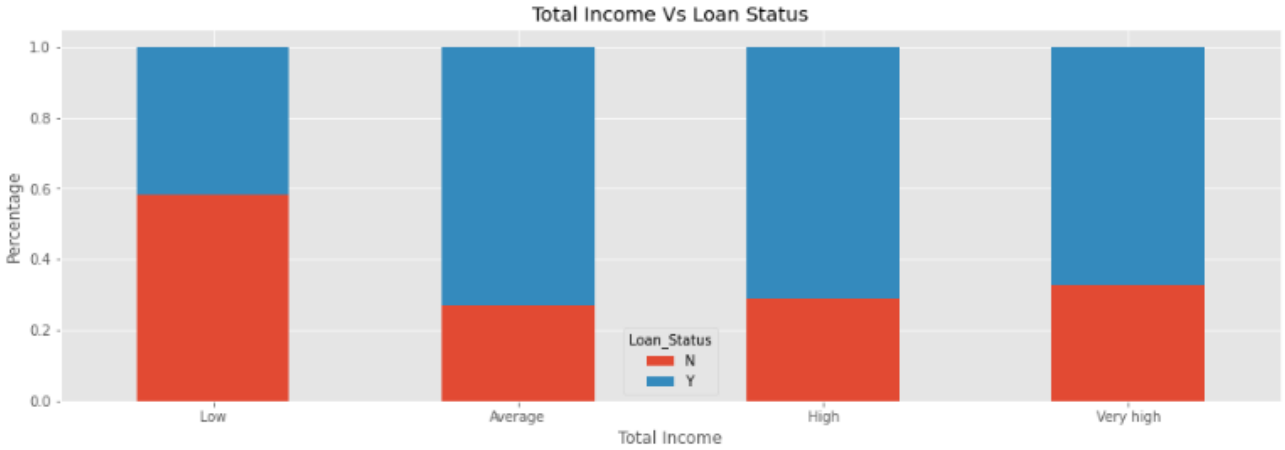
It can be inferred that Applicant income does not affect the chances of loan approval which contradicts our hypothesis in which we assumed that if the applicant income is high, the chances of loan approval will also be high.



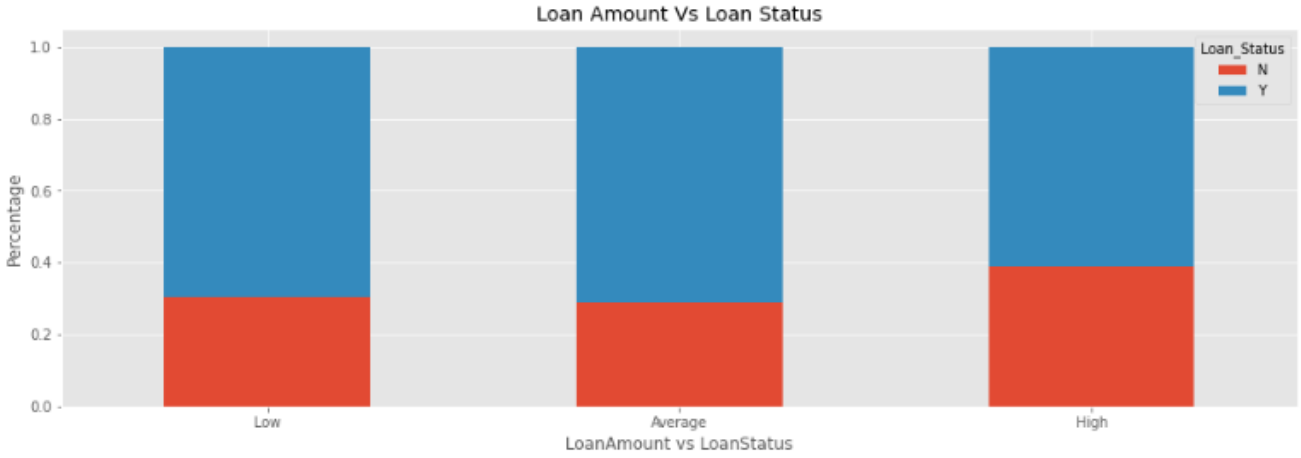
It shows that if the co-applicant's income is less the chances of loan approval are high.

The possible reason behind this may be that most of the applicants don't have any co-applicant so the co-applicant income for such applicants is 0 and hence the loan approval is not dependent on it.

So, we can make a new variable in which we will continue the applicant's and co-applicant's income to visualize the combined effect of income on loan approval.



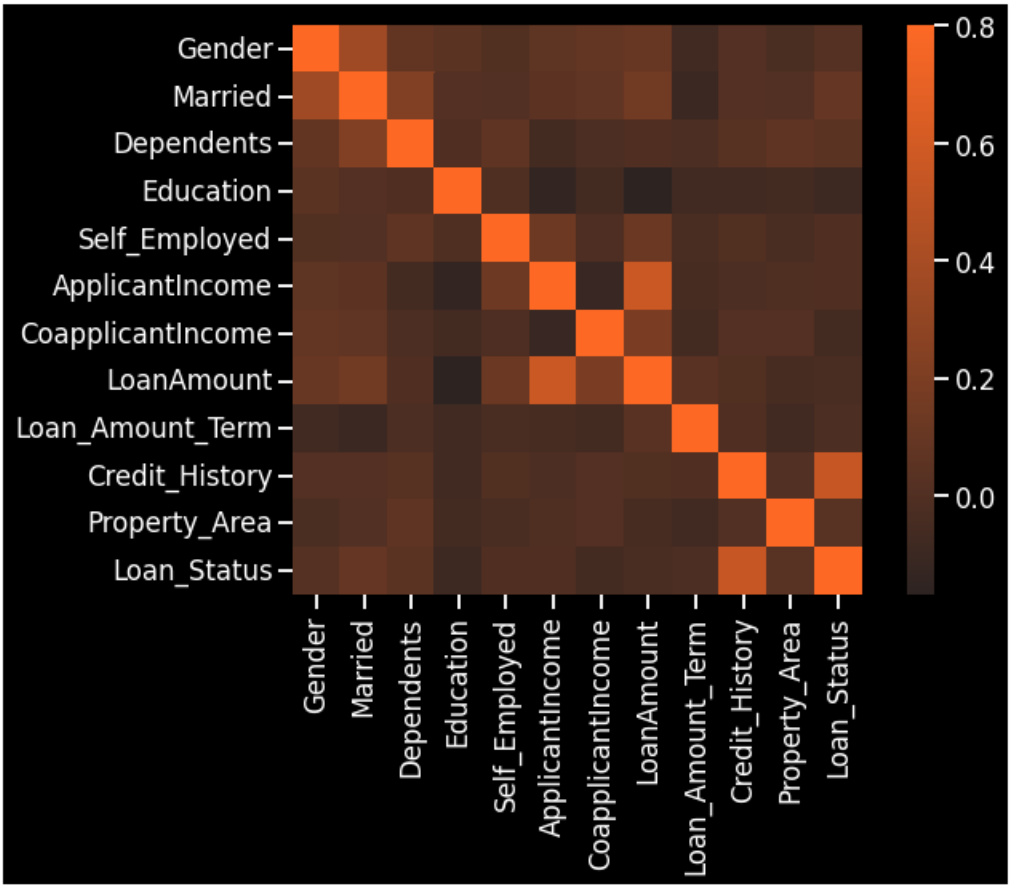
We can see that Proportion of loans getting approved for applicants having low Total_Income is very less as compared to that of applicants with the Average, High, and Very High Income.



It can be seen that the proportion of approved loans is higher for Low and Average Loan Amounts as compared to that of High Loan Amounts which supports our hypothesis in which we can be considered that the chances of loan approval will be high when the loan amount is less.

Visualize Correlation via Heatmap:

```
sns.set(style="ticks", context="talk")
plt.style.use("dark_background")
matrix= train.corr()
fig, ax = plt.subplots(figsize=(9,6))
cmap = sns.dark_palette("#fd6925", as_cmap=True)
fig.tight_layout()
HeatM = sns.heatmap(matrix,vmax=.8,square=True, cmap=cmap)
plt.rcParams['figure.figsize']=(10,10)
fig = HeatM.get_figure()
fig.savefig('output.png', transparent=True)
```



We see that the most correlated variables are (Applicant_Income – Loan_Amount) and (Credit_History - Loan_Status). Loan_Amount is also correlated with Coapplicant_Income.

Correlation between Quantitative Variables:

```
1 np.corrcoef(train['Loan_Amount'], train['Applicant_Income'])

array([[1.        , 0.56518052],
       [0.56518052, 1.        ]])
```

Correlation between loan amount and applicant income is 56%

Correlation between loan amount and co-applicant income is 19 %

```
1 np.corrcoef(train['Loan_Amount'], train['Coapplicant_Income'])

array([[1.        , 0.18921778],
       [0.18921778, 1.        ]])
```

- **Model Building:**

- **Logistic Regression:**

The logistic regression statistic modeling technique is used when we have a binary outcome variable. So, though we may have continuous or categorical independent variables, we can use the logistic regression modeling technique to predict the outcome when the outcome variable is binary.

Assumptions of Logistic regression:

a. Independent variables show a linear relationship with the log of output variables.
b. Non-Collinearity between independent variables. That is, independent variables are independent of each other.
c. Output variable is binary.

For categorical variables, we will have to create dummy variables. Creating dummy variables is nothing but assigning a numerical value to each category. And then transforming the rows with categories into multiple columns.

```
1 train_V1=pd.get_dummies(train_V1)
2 train=pd.get_dummies(train)
```

MODEL 1:

After creating a dummy variable, we performed Logistic Regression.

```
Optimization terminated successfully.
         Current function value: 0.472914
         Iterations 6
                           Logit Regression Results
==============================================================================
Dep. Variable:            Loan_Status   No. Observations:                  614
Model:                          Logit   Df Residuals:                      603
Method:                           MLE   Df Model:                           10
Date:                Sun, 05 Jun 2022   Pseudo R-squ.:                  0.2388
Time:                        06:49:18   Log-Likelihood:                -290.37
converged:                       True   LL-Null:                       -381.45
Covariance Type:            nonrobust   LLR p-value:                 8.370e-34
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
Gender        -0.2696      0.285     -0.944      0.345      -0.829       0.290
Married        0.5486      0.238      2.306      0.021       0.082       1.015
Dependents    -0.0324      0.126     -0.256      0.798      -0.280       0.215
Education     -0.5458      0.247     -2.210      0.027      -1.030      -0.062
Self_Employed -0.0391      0.305     -0.128      0.898      -0.637       0.559
ApplicantIncome  2.47e-06  2.3e-05      0.108      0.914   -4.25e-05    4.75e-05
CoapplicantIncome -6.188e-05  3.31e-05  -1.871      0.061      -0.000   2.94e-06
LoanAmount    -0.0021      0.002     -1.361      0.173      -0.005       0.001
Loan_Amount_Term  -0.0048      0.001     -4.211      0.000      -0.007      -0.003
Credit_History    3.4698      0.348      9.981      0.000       2.788       4.151
Property_Area     0.0027      0.130      0.021      0.983      -0.253       0.258
==============================================================================
```

	odds_ratio	variable
9	20.473323	Credit_History
1	1.737413	Married
0	1.116624	Gender
10	1.016208	Property_Area
5	1.000000	ApplicantIncome
6	0.999935	CoapplicantIncome
7	0.997866	LoanAmount
8	0.995325	Loan_Amount_Term
4	0.974260	Self_Employed
2	0.908414	Dependents
3	0.608918	Education

larger value of odds ratio indicates that the independent variable is a good predictor of target variable. In our case the association between the 'credit history' and 'loan status' is strongest.

The accuracy of the model:

```
1 predicted_1 = model_V1.predict(train_V1)
2 accuracy_score(y,predicted_1)
```

```
0.8094462540716613
```

As we can see p-value for only credit history is less than 0.05. Hence, we will remove all other independent variables and create a new model and check its accuracy.

MODEL 2:

Here, we are filtering the 'credit history' as it contains the highest Odds-ratio in our previous model.

By performing Logistic Regression after filtering got this:

```
Optimization terminated successfully.
         Current function value: 0.539390
         Iterations 5
                           Logit Regression Results
==============================================================================
Dep. Variable:         Loan_Status   No. Observations:                  614
Model:                      Logit   Df Residuals:                      613
Method:                     MLE     Df Model:                            0
Date:            Sun, 05 Jun 2022   Pseudo R-squ.:                  0.1318
Time:                    05:23:36   Log-Likelihood:                -331.19
converged:                  True    LL-Null:                       -381.45
Covariance Type:        nonrobust   LLR p-value:                       nan
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
Credit_History     1.3278      0.107     12.381      0.000       1.118       1.538
==============================================================================
```

Odds Ratio:

```
1 model_V2=LogisticRegression()
2 model_V2.fit(train_V2,y)
3
4 LogisticRegression(C=1.0, class_weight=None, dual=False, fit_intercept=True, intercept_scaling=1, max_iter=100,
5                    multi_class='ovr',n_jobs=1,penalty='12',random_state=1,solver='liblinear',tol=0.0001,verbose=0,warm_start=False)
6
7 df=pd.DataFrame({'odds_ratio':(np.exp(model_V2.coef_).T).tolist(),'variable':train_V2.columns.tolist()})
8 df['odds_ratio'] = df['odds_ratio'].str.get(0)
9
10 df=df.sort_values('odds_ratio', ascending=False)
11 df
```

	odds_ratio	variable
0	27.69363	Credit_History

Accuracy Score:

```
1 predicted_2 = model_V2.predict(train_V2)
2 accuracy_score(y,predicted_2)
```

```
0.8094462540716613
```

MODEL 3:

While finding correlation between independent variables, we found that applicant income and loan amount have good correlation so we will remove one of the variables from the data and then create a model.

```
train_V3 = train_V1.drop(['ApplicantIncome'], axis=1)
train_V3 =pd.get_dummies(train_V3)
```

```
model3=sm.Logit(y,train_V3 )

result3=model3.fit()

print(result3.summary())
```

```
Optimization terminated successfully.
         Current function value: 0.472923
         Iterations 6
                           Logit Regression Results
==============================================================================
Dep. Variable:            Loan_Status   No. Observations:                  614
Model:                          Logit   Df Residuals:                      604
Method:                           MLE   Df Model:                            9
Date:                Sun, 05 Jun 2022   Pseudo R-squ.:                  0.2388
Time:                        06:49:41   Log-Likelihood:                -290.37
converged:                       True   LL-Null:                       -381.45
Covariance Type:            nonrobust   LLR p-value:                 1.809e-34
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
Gender        -0.2681      0.285     -0.940      0.347      -0.827       0.291
Married        0.5479      0.238      2.304      0.021       0.082       1.014
Dependents    -0.0332      0.126     -0.263      0.792      -0.280       0.214
Education     -0.5469      0.247     -2.217      0.027      -1.030      -0.063
Self_Employed -0.0364      0.304     -0.120      0.905      -0.632       0.560
CoapplicantIncome -6.287e-05   3.18e-05     -1.977      0.048      -0.000   -5.42e-07
LoanAmount    -0.0020      0.001     -1.655      0.098      -0.004       0.000
Loan_Amount_Term   -0.0048      0.001     -4.217      0.000      -0.007      -0.003
Credit_History     3.4699      0.348      9.979      0.000       2.788       4.151
Property_Area      0.0035      0.130      0.027      0.978      -0.252       0.259
==============================================================================
```

	odds_ratio	variable
8	22.979782	Credit_History
1	1.521448	Married
2	1.019757	Dependents
5	0.999940	CoapplicantIncome
6	0.998169	LoanAmount
7	0.995876	Loan_Amount_Term
4	0.978631	Self_Employed
9	0.968565	Property_Area
0	0.854993	Gender
3	0.580621	Education

We can see the p-value for co-applicant income(nearly) and credit history is less than 0.05. Hence removing all other variables and creating another model.

Accuracy of the model:

```
predicted_3 = model_V3.predict(train_V3)
accuracy_score(y,predicted_3)
```

```
0.8110749185667753
```

MODEL 4:

Creating a fourth model with two columns: credit history and co-applicant income.

```
train_V4 = train.filter(['Credit_History', 'CoapplicantIncome'], axis=1)
train_V4=pd.get_dummies(train_V4)
```

```
model4=sm.Logit(y,train_V4 )

result4=model4.fit()

print(result4.summary())
```

```
Optimization terminated successfully.
         Current function value: 0.530149
         Iterations 5
                           Logit Regression Results
==============================================================================
Dep. Variable:            Loan_Status   No. Observations:                  614
Model:                          Logit   Df Residuals:                      612
Method:                           MLE   Df Model:                            1
Date:                Sun, 05 Jun 2022   Pseudo R-squ.:                  0.1466
Time:                        06:49:56   Log-Likelihood:                -325.51
converged:                       True   LL-Null:                       -381.45
Covariance Type:            nonrobust   LLR p-value:                 3.818e-26
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
Credit_History     1.5175      0.126     12.004      0.000       1.270       1.765
CoapplicantIncome -0.0001   3.68e-05     -2.916      0.004      -0.000   -3.51e-05
==============================================================================
```

	odds_ratio	variable
0	28.048905	Credit_History
1	0.999948	CoapplicantIncome

Accuracy:

```
predicted_4 = model_V4.predict(train_V4)
accuracy_score(y,predicted_4)
```

```
0.8127035830618893
```

As we can see our third model had maximum accuracy among all other models. Testing the model by splitting data into 70% for train and 30 % for test.

```
from sklearn.linear_model import LogisticRegression
from sklearn import metrics
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(train_V4, y, test_size=0.3, random_state=0)
logreg = LogisticRegression()
logreg.fit(X_train, y_train)
```

```
LogisticRegression()
```

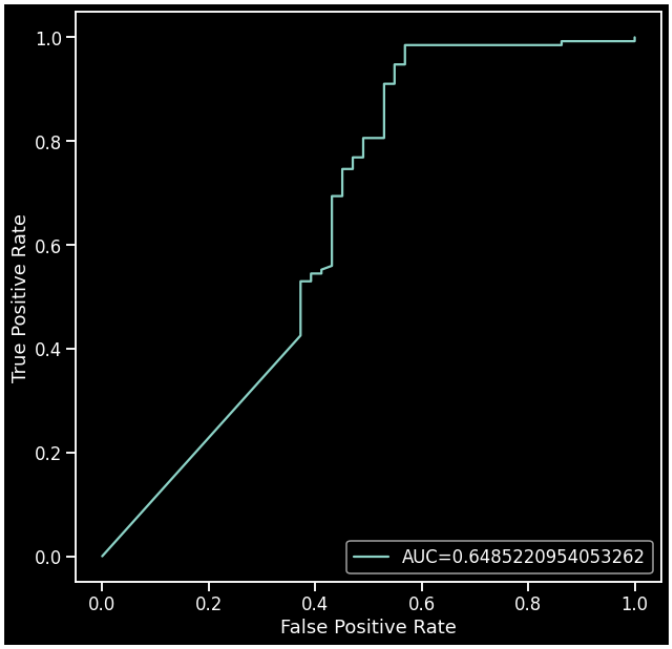
```
y_pred = logreg.predict(X_test)

print('Accuracy of logistic regression classifier on test set: {:.2f}'.format(logreg.score(X_test, y_test)))
```

```
Accuracy of logistic regression classifier on test set: 0.83
```

ROC Curve:

The Area Under the ROC curve (AUC) is an aggregated metric that evaluates how well a logistic regression model classifies positive and negative outcomes at all possible cutoffs. It can range from 0.5 to 1, and the larger it is the better. Here our AUC score is 0.6, so this is not so good score at all.



- **Random Forest:**

This is a supervised machine learning algorithm mostly used for classification problems. All features should be discretized in this model so that the population can be split into two or more homogeneous sets or subsets. This model uses a different algorithm to split a node into two or more sub-nodes. With the creation of more sub-nodes, the homogeneity and purity of the nodes increase with respect to the dependent variable.

```
rfc = RandomForestClassifier(random_state = 1, max_depth = 8,
                             n_estimators = 1200, min_samples_split = 15, min_samples_leaf = 2)
modelr = rfc.fit(X_train, y_train)
y_pred = modelr.predict(X_test)

from sklearn.metrics import accuracy_score
accuracy_score(y_test, y_pred)
```

```
0.827027027027027
```

The accuracy Score using Random Forest is: 83%

- **Decision Tree:**

This is a tree-based ensemble model which helps in improving the accuracy of the model. It combines a large number of Decision trees to build a powerful predicting model. It takes a random sample of rows and features of each individual tree to prepare a decision tree model. The final prediction class is either the mode of all the predictors or the mean of all the predictors.

```
#Decision Tree
from sklearn.tree import DecisionTreeClassifier
tree = DecisionTreeClassifier()
tree.fit(X_train, y_train)
ypred_tree = tree.predict(X_test)

accuracy = f1_score(y_test, ypred_tree)
accuracy
```

```
0.8375451263537905
```

The accuracy Score using the Decision Tree is: 84%

CONCLUSION

To predict whether the loan should be approved or not we collected the data from Kaggle, which contains 980 rows of 13 features. Then we divided the dataset into training and testing data and using the training dataset, we performed 3 models to predict our target column Loan Status. We removed missing values by replacing mode values and removing outliers by log transformation. Then we perform EDA (Univariate & Bivariate) on the covariates on the dataset and saw how each feature is distributed and associated with each other using correlation. Then we create dummy variables to construct the model and constructed models taking different variables into account and found through odd ratios that credit history creates the most impact on loans. Then we got a model with co-applicant income and credit history as independent variables with the highest accuracy. The predictive models based on Logistic Regression, Decision Tree, and Random Forest, give the accuracy as 83%, 83%, and 84%. This shows that for the given dataset, the accuracy of the model based on the decision tree is highest.

There may be many more algorithms that would have provided more efficient results. Ensemble Learning, Deep Learning, and Neural Networks can be the foundation of future work related to this work. I have limited the scope of the analysis to get a decent fair result. Machine Learning and Data Science are a huge ocean of opportunities to try and gamble out various techniques and processes.

REFERENCE

1. Dileep B. Desai, Dr. R.V.Kulkarni “A Review: Application of Data Mining Tools in CRM for Selected Banks”, (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 4 (2), 2013, 199 –201.M. Young, The Technical Writer’s Handbook. Mill Valley, CA: University Science, 1989.

2. J.H. Aboobyda, and M.A. Tarig, “Developing Prediction Model of Loan Risk in BanksUsing Data Mining”, Machine Learning and Applications: An International Journal (MLAIJ), vol. 3, no.1, pp. 1–9, 2016. K. Elissa, "Title of paper if known," unpublished.

3. A.B. Hussain, and F.K.E. Shorouq, “Credit risk assessment model for Jordanian commercial banks: Neuralscoring approach”, Review of Development Finance, Elsevier, vol. 4, pp. 20–28, 2014. JAC: A JOURNAL OF COMPOSITION THEORY Volume XIII, Issue V, MAY 2020ISSN: 0731-6755Page No: 324

4. T. Harris, “Quantitative credit risk assessment using support vector machines: Broad versus Narrow default definitions”, Expert Systems with Applications, vol. 40, pp. 4404–4413, 2013.

5. Kaggle