Predicting Loan Status

OBJECTIVE : To predict weather loan will be approved or not based on applicant's informations

Data Source : Analytics Vidhya Data Science Hacakthon pltform

Tools Used: Numpy, Pandas, Matplotlib, Scikit Learn libraries of python

Getting to know data

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 614 entries, 0 to 613
Data columns (total 13 columns):
 #   Column             Non-Null Count  Dtype  
---  ------             --------------  -----  
 0   Loan_ID            614 non-null    object 
 1   Gender             601 non-null    object 
 2   Married            611 non-null    object 
 3   Dependents         599 non-null    object 
 4   Education          614 non-null    object 
 5   Self_Employed      582 non-null    object 
 6   ApplicantIncome    614 non-null    int64  
 7   CoapplicantIncome  614 non-null    float64
 8   LoanAmount         592 non-null    float64
 9   Loan_Amount_Term   600 non-null    float64
 10  Credit_History     564 non-null    float64
 11  Property_Area      614 non-null    object 
 12  Loan_Status        614 non-null    object 
dtypes: float64(4), int64(1), object(8)
memory usage: 62.5+ KB
```

Independent Variable : Loan_Status

Notes -: Categorical/binary variables : Loan_ID, Gender, Married, Dependents, Education, Self_Employed, Credit_History, Property_Area, Loan_Status. Continous Variables : ApplicantIncome, CoapplicantIncome, LoanAmount Descrete Numerical variable : Loan_Amount_Term

Data Exploration

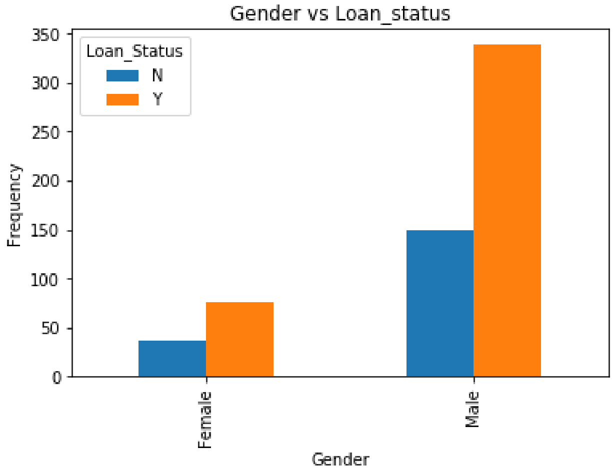
Creating frequency table for each variable to explore dependecy of Loan_Status variable on every other variable

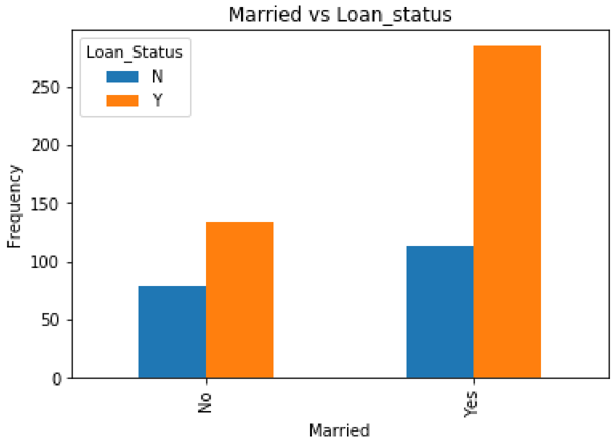
```
Frequency Table for variable 'Gender' :
 Loan_Status     N     Y   All
Gender                        
Female       0.06  0.12  0.19
Male         0.25  0.56  0.81
All          0.31  0.69  1.00

Frequency Table for variable 'Married' :
 Loan_Status     N     Y   All
Married                       
No           0.13  0.22  0.35
Yes          0.18  0.47  0.65
All          0.31  0.69  1.00

Frequency Table for variable 'Dependents' :
 Loan_Status     N     Y   All
Dependents                    
0            0.18  0.40  0.58
1            0.06  0.11  0.17
2            0.04  0.13  0.17
3+           0.03  0.06  0.09
All          0.31  0.69  1.00


Frequency Table for variable 'Education' :
 Loan_Status      N     Y   All
Education                      
Graduate      0.23  0.55  0.78
Not Graduate  0.08  0.13  0.22
All           0.31  0.69  1.00

Frequency Table for variable 'Self_Employed' :
 Loan_Status       N     Y   All
Self_Employed                   
No              0.27  0.59  0.86
Yes             0.04  0.10  0.14
All             0.31  0.69  1.00
```

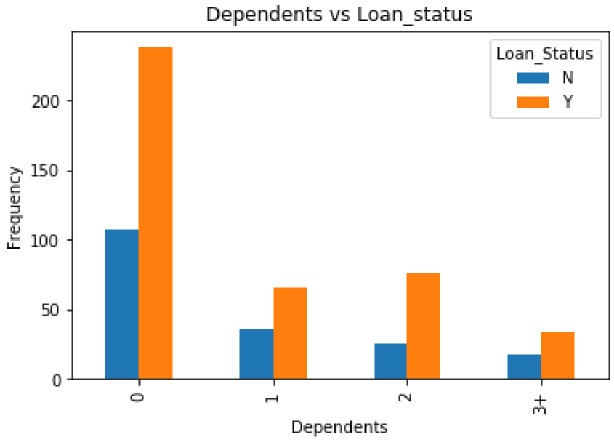
```
Frequency Table for variable 'Credit_History' :
 Loan_Status        N     Y    All
Credit_History
0.0             0.15  0.01  0.16
1.0             0.17  0.67  0.84
All             0.32  0.68  1.00


Frequency Table for variable 'Property_Area' :
 Loan_Status       N     Y    All
Property_Area
Rural          0.11  0.18  0.29
Semiurban      0.09  0.29  0.38
Urban          0.11  0.22  0.33
All            0.31  0.69  1.00
```

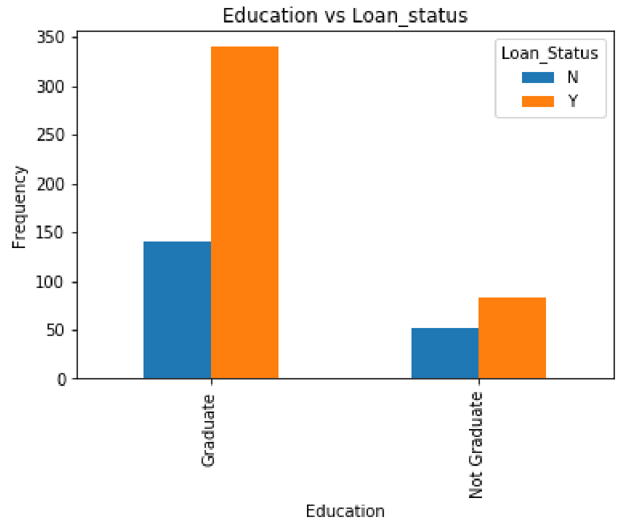
Visualizing data

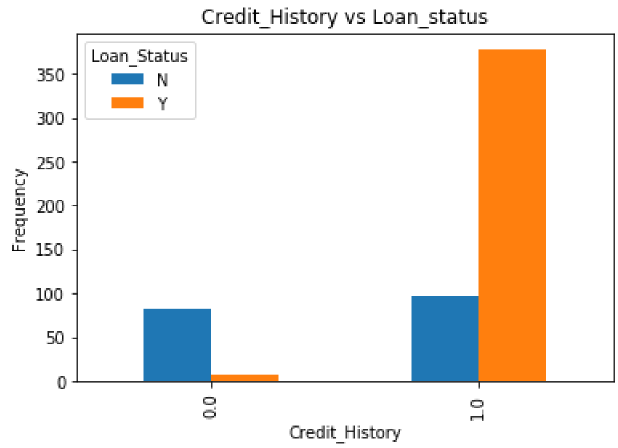
Visually exploring data : Bar plots for categorical independent variables. Box Plots for continous variables





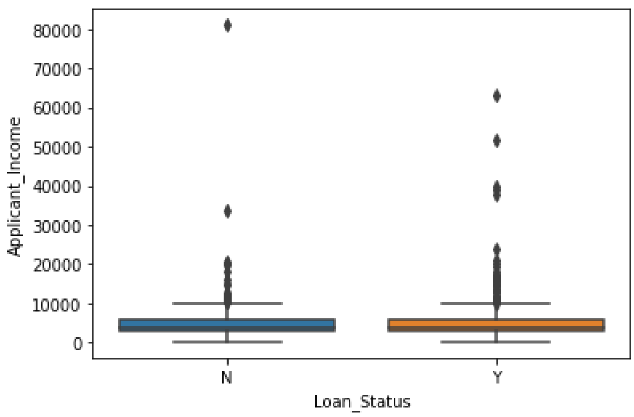






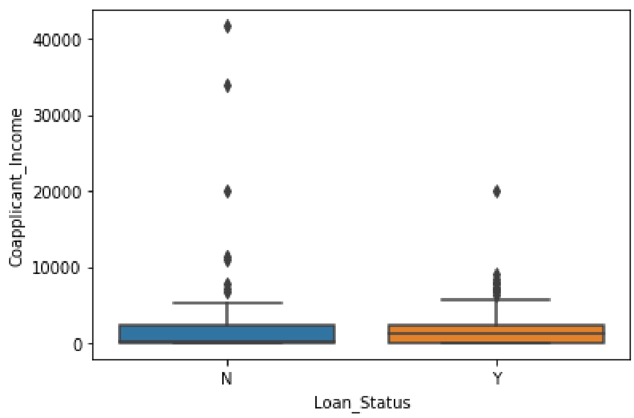

```
Applicant_Income Box Plot :
```

<matplotlib.axes._subplots.AxesSubplot at 0xe99ea42ac8>



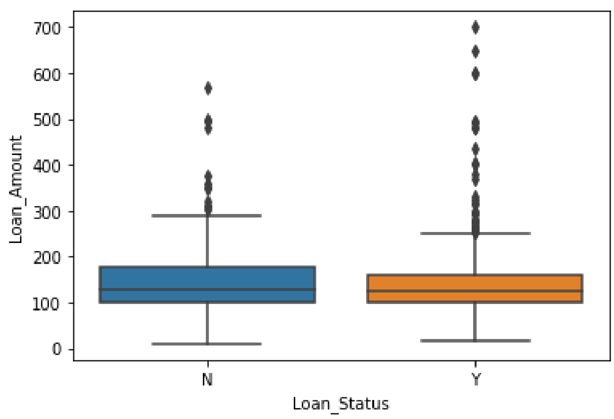
Coapplicant_Income Box Plot :

<matplotlib.axes._subplots.AxesSubplot at 0xe99ead1f08>



Loan_Amount Box Plot :

<matplotlib.axes._subplots.AxesSubplot at 0xe99eb19c88>



Implementing Logistic Regression Model

Intercept is [-2.41961967]

Coefficents are :

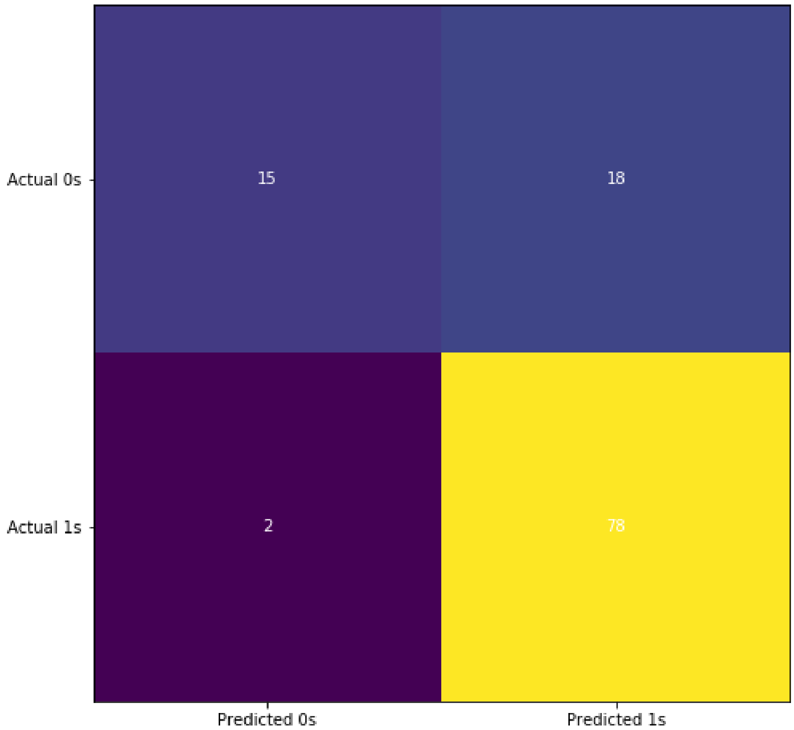
	Independent variable	coefficents
0	Applicant_Income	1.35e-05
1	Coapplicant_Income	-5.08e-05
2	Loan_Amount	-2.22e-03
3	Loan_Amount_Term	-2.30e-03
4	Credit_History	3.83e+00
5	Gender_Male	1.62e-01
6	Married_Yes	4.36e-01
7	Dependents_1	2.74e-01
8	Dependents_2	4.30e-01
9	Dependents_3+	4.46e-01
10	Education_Graduate	4.72e-01

```
11        Self_Employed_Yes    -1.33e-01
12  Property_Area_Semiurban     7.19e-01
13      Property_Area_Urban    -2.51e-01
```

Evaluating the Model

```
Accuracy Score of model is : 0.8230088495575221
```

Confusion Matrix :




```
Classification report is :
               precision    recall  f1-score   support

           0       0.88      0.45      0.60        33
           1       0.81      0.97      0.89        80

    accuracy                           0.82       113
   macro avg       0.85      0.71      0.74       113
weighted avg       0.83      0.82      0.80       113
```

Comment : Logistic regression model seems to be a good fit for this data

Test Data

```
Test data Information :

<class 'pandas.core.frame.DataFrame'>
Index: 367 entries, LP001015 to LP002989
Data columns (total 11 columns):
 #   Column             Non-Null Count  Dtype  
---  ------             --------------  -----  
 0   Gender             367 non-null    object 
 1   Married            367 non-null    object 
 2   Dependents         367 non-null    object 
 3   Education          367 non-null    object 
 4   Self_Employed      367 non-null    object 
 5   Applicant_Income   367 non-null    float64
 6   Coapplicant_Income  367 non-null    float64
 7   Loan_Amount        367 non-null    float64
 8   Loan_Amount_Term   367 non-null    float64
 9   Credit_History     367 non-null    uint8  
 10  Property_Area      367 non-null    object 
dtypes: float64(4), object(6), uint8(1)
memory usage: 31.9+ KB
```

Output

Select Loan ID from dropdown to check predicted Loan approval status

Select ID LP001015

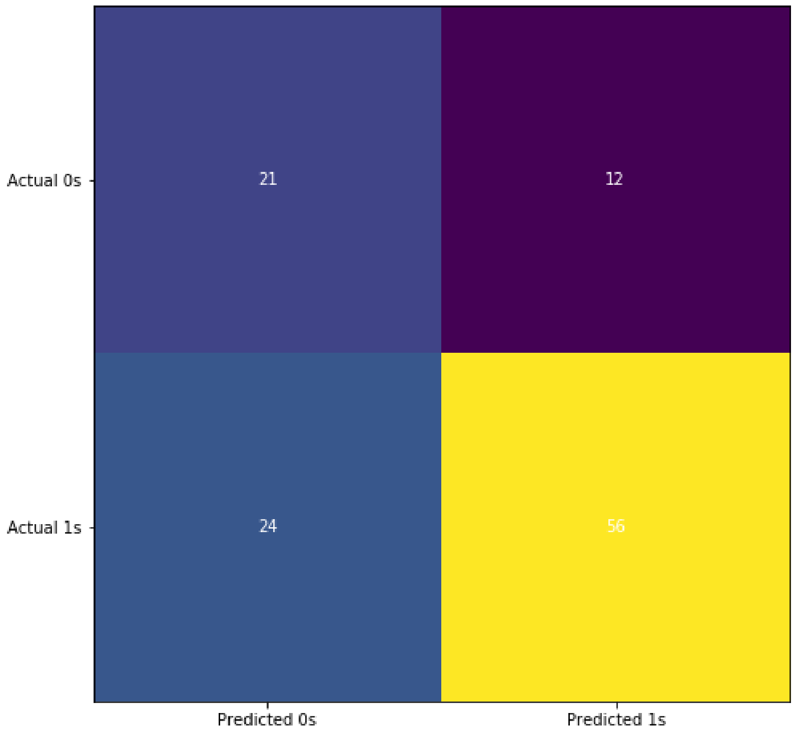
	Loan_ID	Loan_Status
1	LP001022	Y

```
<function __main__.select_func(l)>
```

To check if Decision Tree Classification model can be a better fit, data was also trained in decision tree classifier and model evaluated based n classification accuracy

Building Decision Tree Classification Model

```
Accuracy Score of model is : 0.6814159292035398

Classification report is :
               precision    recall  f1-score   support

           0       0.47      0.64      0.54        33
           1       0.82      0.70      0.76        80

    accuracy                           0.68       113
   macro avg       0.65      0.67      0.65       113
weighted avg       0.72      0.68      0.69       113
```



Comments : Decision Tree Model is rejected due to low accuracy