PROJECT IN DATA ANALYSIS & VISUALIZATION SPRING 2021
POINT PARK UNIVERSITY

FINAL CASE STUDY

HOUSING LOAN PREDICTION

PREPARED BY: SUCHARITHA DONEPUDI
SAMANTHA BROOKS
OZOYA OHILEBO
PROFESSOR: JEFFREY SEAMAN

Table of Contents

1. Abstract

Insurance is driven by risk and risk-taking. Risk is generally good, however, the threshold for risk must be calculated. Driven by the need to avoid bad debt, insurance companies tend to make rigorous research on individuals who apply for loans to establish their creditworthiness. This helps the insurance companies avoid bad debt as well as try to indemnify themselves to their best account.

2. Introduction

Loans are the core business of banks. The main profit comes directly from the loan's interest. The loan companies grant a loan after an intensive process of verification and validation. However, they still don't have assurance if the applicant can repay the loan with no difficulties. Based on these issues we have build a predictive model to predict if an applicant is eligible for the loan or not. We have prepared the data using Jupyter Notebook and use various models to predict the target variable. And also use Tableau and Power BI for effective visualization and storytelling.

2.1 Project Objective

In this project, we aim to predict if an applicant is eligible for the corresponding loan or not. The Loan prediction dataset has various data variables related to the applicant such as Gender, Income, Loan amount, Loan term, and credit history, etc. The project aims at utilizing Machine learning techniques to predict Loan eligibility based on the data collected from the Loan Predication dataset. Visualization of data is an important part of any project, as it helps to communicate the contents of the project more holistically.

Therefore to effectively understand the applicant data from the Loan Prediction dataset, we try to visualize the different data variables in the dataset using Tableau. Through the process of visualization, we can find relationships or trends within the variables in the dataset. These results can be a supplement to the Loan prediction process that we would be carrying out in the future.

2.2 Overview of Loan Prediction

Loans have been the backbone of the banking sector. Banks usually get their income from the interest they get from the Loans lent to the people. Banks have an outdated process to verify and validate Loan applications. This conventional process is time-consuming and not efficient in this technology-driven generation. The loan Prediction project helps to identify the most deserving applicant for a Loan. This provides a great advantage for the customers and the employees as the application will be scrutinized and validated swiftly in reduced time as compared to the conventional Loan processing methods.

Loan prediction requires a training dataset, which would be used as a reference for predicting the new customer's loan application. The Loan Prediction dataset retrieved from Kaggle containing the different variables of the Loan application would be a great training dataset for our Prediction process. New applicant information which is filled in an application form tends to act as a test dataset. Loan prediction works by creating a machine learning model that would predict the eligibility of the new applicant based on the inferences gathered from the training dataset. Machine learning models such as Random Forest, Logistic Regression, SVM (supporting vector machine), and KNN(K nearest neighbors) are applied to find which models provide higher accuracy. Depending on the results of the best model, we tend to predict the Loan applicant's eligibility in the future.

2.3 Different Types of Loans

Consumers avail credit or loan from the bank to pay for the items/services for which they would not be having money at that moment. Banks usually charge a nominal interest rate and have a standardized loan period as prescribed by the federal and state guidelines. Customers have a variety of loan and credit options to choose from Banks, depending on their requirements. Loans are classified as different types based on the repayment period and the interest rates. The different types of loans are listed below,

Personal Loans: Personal loans are versatile as they can be utilized for any reason. For example, if you want to renovate your house, meet your medical expenses, etc. personal loans can come in handy. Personal loan term generally varies from 1 to 5 years. The interest rate of the loan tends to vary 5 % to 35 % depending upon the lender and the amount borrowed. The minimum amount that can be loaned is generally $ 1000 and the maximum amount that can be borrowed is $ 100,000 (depends on the source of the loan). However, these personal loans may require some collateral and a good credit score.

Auto Loans: Loans that are linked to buying a vehicle (property) are termed to be Auto loans. These loans help the customers to afford to buy a car/vehicle. However, there is a condition for this type of loan that the customer may lose their vehicle or car if they miss repayments. The repayment period for this type of loan is 1 to 7 years and the annual interest rate ranges from 1 % to 14 %

Student Loans: Student loans are the most common type of loans that are centered around education. Student loans help the students & families to cover the cost of college/higher education. Student loans are offered to college students and their families to help cover the cost of higher education. Most institutional give student loans at low-interest rates and the repayment period are from 1 to 7 years. Student loans are provided by private and federal institutions and federal student loans tend to have the lowest interest rate (0 %) and better repayment conditions as compared to other lenders

Mortgages: Mortgages are a type of loan which can be utilized by consumers to buy a home or some real estate property. Mortgages are secured loans and they tend to have the lowest interest rates among other types of loans. There is a risk for foreclosure if the consumer falls behind in monthly payments. The repayment period ranges from 15 to 30 years and the annual interest rates range from 3 % to 5.5 %

Business loans: Entrepreneurs and people involved in small businesses tend to require money for their operation. Banks and government institutions provide business loans for such candidates at standardized rates and periods.

2.4 Basic Requirements and Eligibility of taking Loans

Loans applied by the consumers are evaluated based on various factors, to identify whether they qualify for the corresponding loan or not. Let us now look at the basic requirements for loans which are mentioned below,

Credit Score: Loan providers tend to look at applicant's credit scores as an important factor in the loan process. A credit score is a number from 300 to 850, which is calculated based on the applicant's level of debt, credit history, and history of repayment, etc. A higher credit score means the applicant is more trustworthy to lend money. Banks usually focus on applicants with high credit scores are 550 to 600.

Credit_History variable in the Loan-Predication dataset provides whether the applicant has a credit history or not. This factor can be incredibly useful in predicting the eligibility of loans

Applicant's employment: The information regarding how the applicant is employed is necessary to assess their creditworthiness of the applicant. Employment can be self, private, and government, etc. **The self_Employed** variable in the dataset provides a perspective of the employment status of the applicant.

Income: When banks/financial institutions lend money to people, they have to ensure that the applicant has the means to repay the loan. Therefore, the income of the applicants is usually analyzed before providing a loan to the customers. Income requirements can vary depending upon the loan sought, banks tend to give loans to people with income ranging from $ 30,000 per year (these figures tend to vary).

Applicant Income and **Co-applicant Income** variables in the Loan Prediction dataset provide the income values of the applicant and their dependents, which are important for the process of loan scrutinizing.

Collateral: Banks/financial lenders ask the applicant to provide a security/pledge for the loan that they are provided. It can be in the form of the property or the very thing for which the loan is applied for. Auto/Home loans have the collateral as the vehicle/home itself. The lenders tend to follow foreclosure the collateral if the repayments are not paid properly.

Demographic and personal details: Applicant needs to provide personal details such as age, gender, place where they come from, family members (dependents), and marital status, etc. This information is crucial for the banks and financial institutions to verify the authenticity/identity of the applicant. **Marriage**, **Dependents, and Gender** variables in the dataset tend to provide some information regarding the same.

3. Methodology

3.1 Data collection and analysis

To predict whether an individual is eligible for a housing loan, we utilized a data set from a data science community website called Kaggle. This particular data set contains 13 variables that are used to determine an applicant's eligibility. These variables are LoanID, Gender, Married, Dependents, Education, Self-employed, ApplicantIncome, CoapplicantIncome, LoanAmount, Loan_Amount_Term, Credit_History, Property_Area, and Loan_Status.

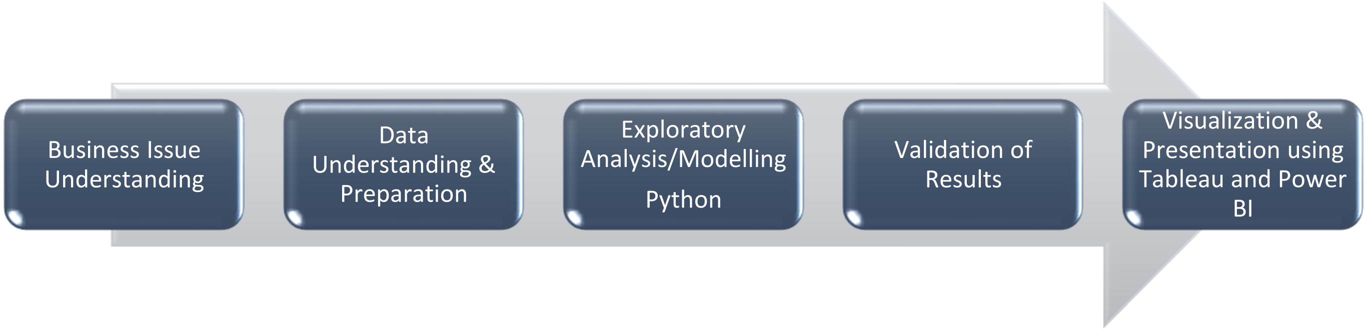
- ★ **LoanID** represents a unique identifier for each loan application.
- ★ **Gender** states whether the applicant is a male or female.
- ★ **Married** indicates if the applicant is married by stating "Yes" or "No".
- ★ **Dependents** display the number of individuals that are financially dependent on the applicant. If the applicant has three or more dependents, it is noted as 3+ in this data set.

- ★ The **Education** variable displays the highest level of education the applicant has received, which is noted as either "Graduate" or "Not Graduate".
- ★ **Self_Employed** shows whether the applicant is self-employed or works for a specific employer.
- ★ **ApplicantIncome** and **CoapplicantIncome** display how much the applicant and co-applicant earn annually, respectively.
- ★ **LoanAmount** illustrates the requested housing loan amount in thousands of dollars.
- ★ **Loan_Amount_Term** shows the agreed number of months it will take the applicant to completely pay off the loan.
- ★ **Credit_History** utilizes a 1 if the applicant has good credit and 0 if the applicant has bad credit.
- ★ **Property_Area** describes the area type of the house the applicant wishes to purchase. The author divides the data set into three property areas: urban, semiurban, and rural. Lastly,
- ★ **Loan_Status** states whether the loan was accepted or declined.

To analyze the total income of the loan application, we created a new variable called total income. This variable sums the ApplicantIncome and CoapplicantIncome figures of each loan.

3.2 Project Implementation and assessment plan

For this project, we will utilize three programs to predict loan eligibility and visualize the data. Python will be used to train the data and create a model to predict an individual's eligibility based on the variables provided in this data set. Furthermore, both Tableau and Power BI will visualize relationships such as gender, income, and marriage through bar charts and line graphs.


Business Issue Understanding
Data Understanding & Preparation
Exploratory Analysis/Modelling Python
Validation of Results
Visualization & Presentation using Tableau and Power BI

4. Technical Design

4.1 Python Script of detailed analysis

4.1a. Problem Statement

There is an organization named Housing Finance that bargains in every home credit. They have a presence across all metropolitan, semi-metropolitan, and rural regions. Applicants initially apply for a home credit after that organization approves the client qualification for the advance. Nonetheless, doing this manually takes a great deal of time. Hence it wants to automate the loan eligibility process (real-time) based on customer information.

So, the final thing is to identify the application segments that are eligible for taking the loan. How will the company benefit if we give the customer segments is the immediate question that arises? The solution is banks would give loans to only those applicants that are eligible so that they can be assured of getting the money back. Hence the more accurate we are in predicting the eligible applicants the more beneficial it would be for the Housing Finance Company.

4.1b. Types of Problems

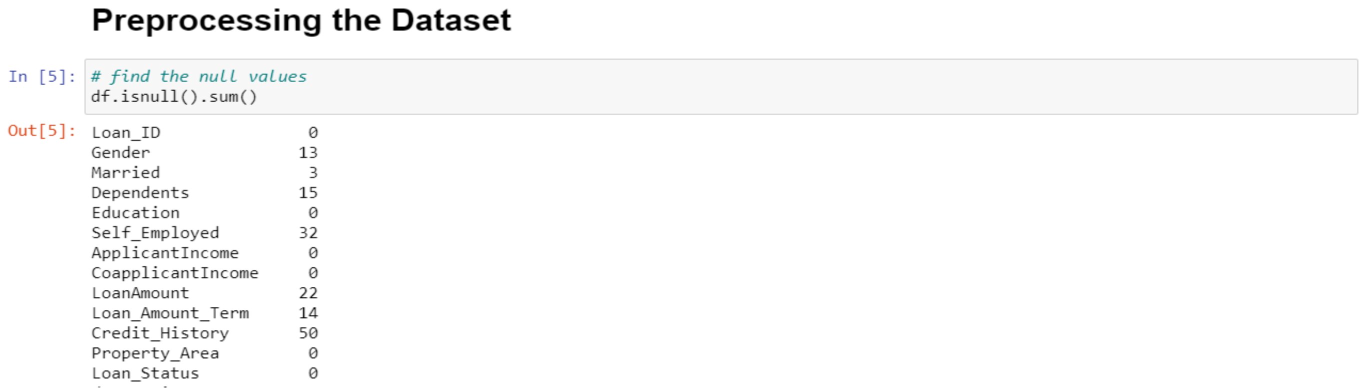
The above problem is a clear classification problem as we need to classify whether the applicant is eligible for the loan or not. So, this can be solved by any of the classification techniques like-

1) Logistic Regression.
2) Decision Tree Algorithm.
3) Random Forest Technique.

4.1c. Data Cleaning and Structuring

Before we go for modeling the data, we have to check whether the data is cleaned or not. And after cleaning, we have to structure the Data. For the cleaning part, First, we have to check whether there exist any missing values. For that, I am using the code snippet isnull().

The above code suggests that there are 13 missing values in Gender, 3 In Married, 15 in Dependents, 32

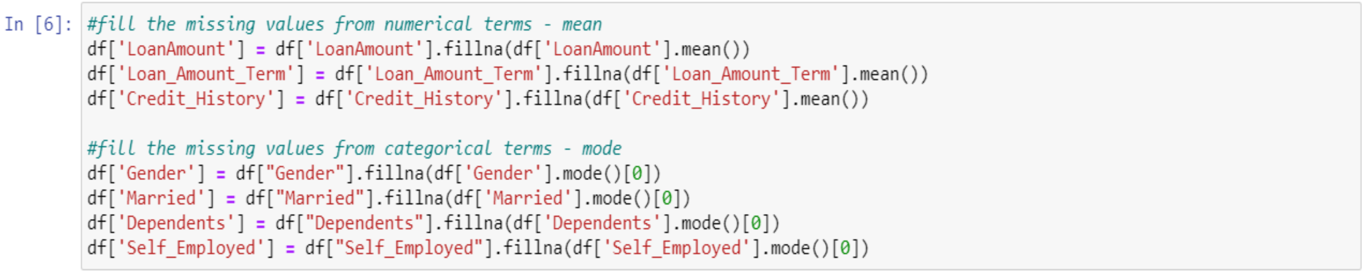


Preprocessing the Dataset

```
In [5]:  # find the null values
         df.isnull().sum()

Out[5]:  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
```

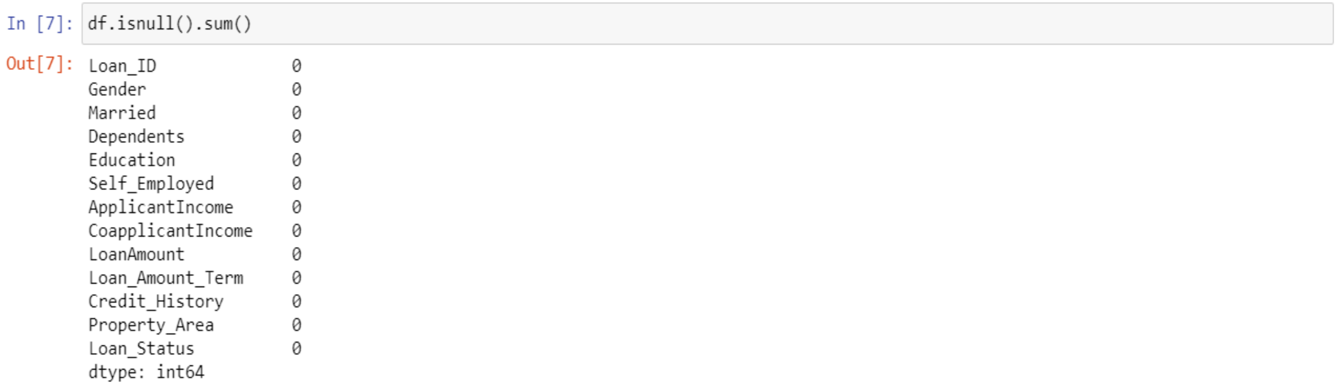
in Self_Employed, 22 in Loan Amount, 14 in Loan_Amount_Term, and 50 in Credit History.

Except for the Loan Amount and Loan_Amount_Term, everything else which is missing is of type categorical. Hence, we can replace the missing values with the mode of that particular column. Before getting into the code, I would like to say few things about the mean, median, and mode.



```
In [6]: #fill the missing values from numerical terms - mean
        df['LoanAmount'] = df['LoanAmount'].fillna(df['LoanAmount'].mean())
        df['Loan_Amount_Term'] = df['Loan_Amount_Term'].fillna(df['Loan_Amount_Term'].mean())
        df['Credit_History'] = df['Credit_History'].fillna(df['Credit_History'].mean())

        #fill the missing values from categorical terms - mode
        df['Gender'] = df["Gender"].fillna(df['Gender'].mode()[0])
        df['Married'] = df["Married"].fillna(df['Married'].mode()[0])
        df['Dependents'] = df["Dependents"].fillna(df['Dependents'].mode()[0])
        df['Self_Employed'] = df["Self_Employed"].fillna(df['Self_Employed'].mode()[0])
```

Mean is nothing but the average value whereas the median is nothing but the central value and mode the most occurring value. Replacing the categorical variable by mode makes some sense.



```
In [7]: df.isnull().sum()

Out[7]: Loan_ID              0
        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
```

Now we found that there are no missing values. However, we have to be very careful with the Loan_ID column too. As we know that Loan_ID should be unique. So, if there n number of rows, there should be n number of unique Loan_ID's.

4.1d. Exploratory Data Analysis

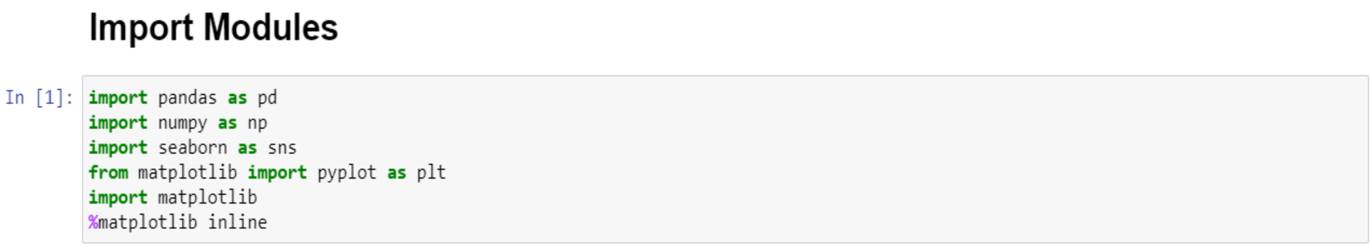
Before going to EDA analysis, we are going to create a new attribute that is going to help us to train the machine. We have two columns named applicant income and co-applicant income. It may be the case that total income might have a great impact on Loan Status.

Total Income = Applicant Income + Co-applicant Income

Now starts the Exploratory data analysis, we can make some assumptions through this EDA analysis. Like:

1. *The one whose salary is more can have a greater chance of loan approval.*
2. *The one who is a graduate has a better chance of loan approval.*
3. *Married people would have an upper hand over unmarried people for loan approval.*
4. *The applicant who has a smaller number of dependents has a high probability for loan approval.*
5. *The lesser the loan amount the higher the chance of getting the loan.*

Step 1: Import the required libraries

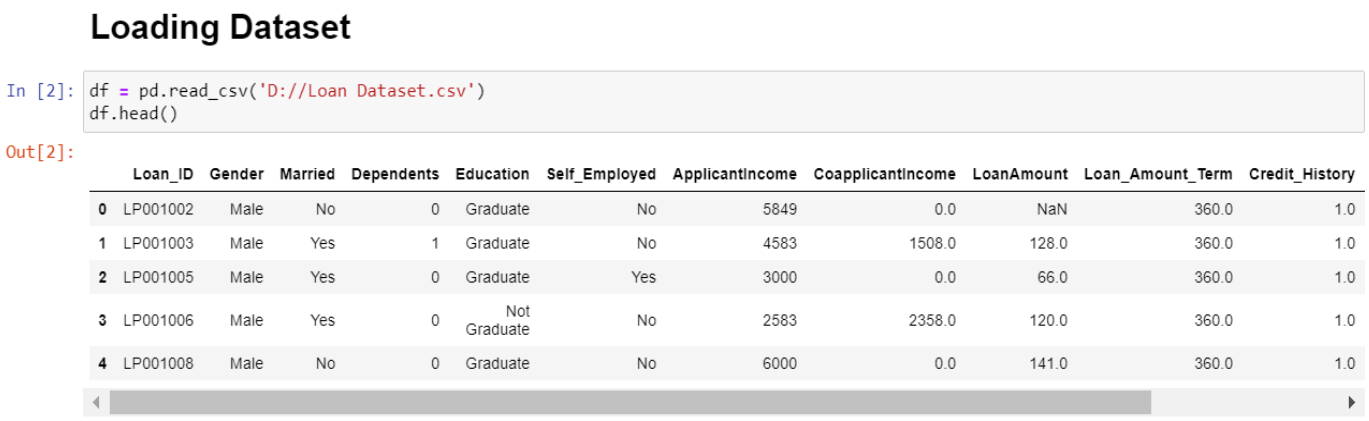


Import Modules

In [1]:

```
import pandas as pd
import numpy as np
import seaborn as sns
from matplotlib import pyplot as plt
import matplotlib
%matplotlib inline
```

Step 2: Loading the Data frame



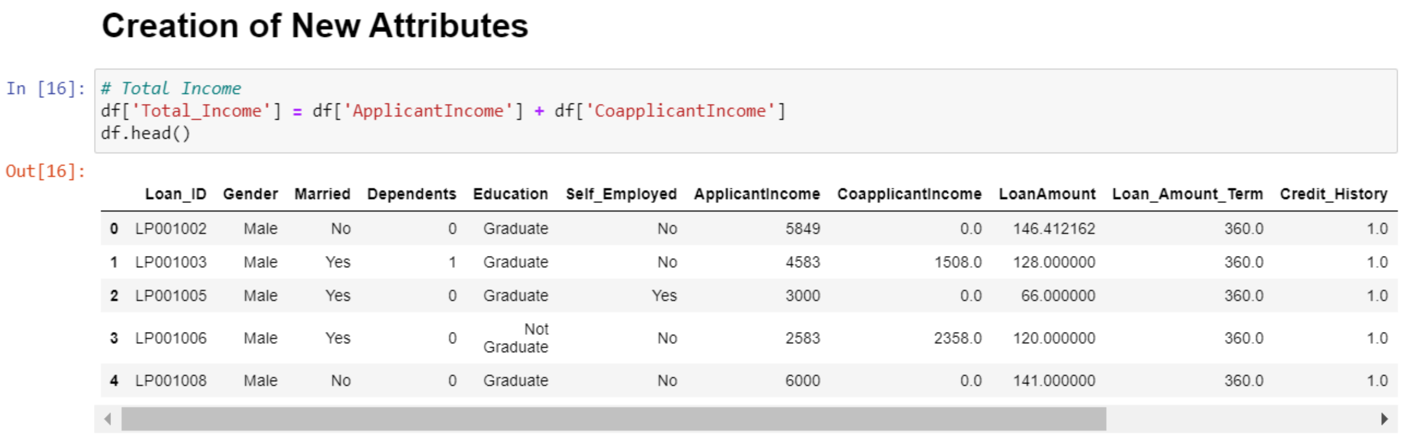
Loading Dataset

In [2]:

```
df = pd.read_csv('D://Loan Dataset.csv')
df.head()
```

Out[2]:

	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



Creation of New Attributes

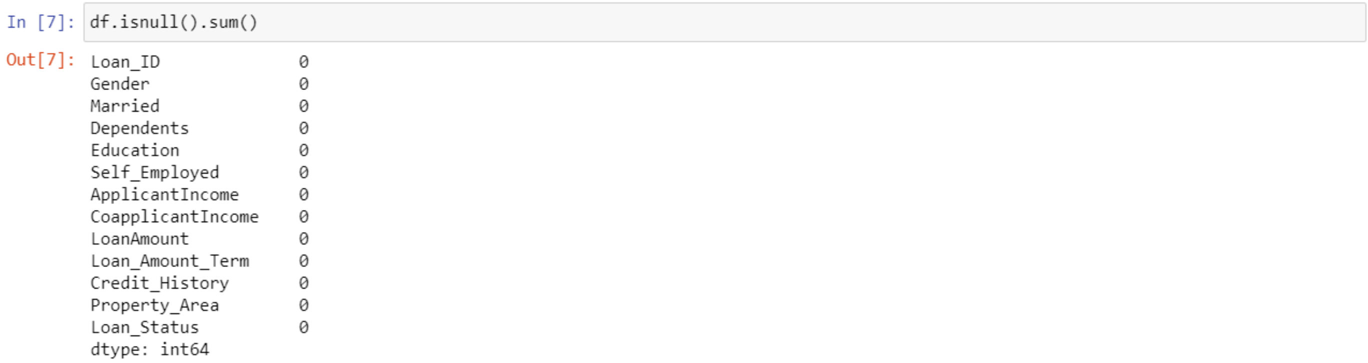
In [16]:

```
# Total Income
df['Total_Income'] = df['ApplicantIncome'] + df['CoapplicantIncome']
df.head()
```

Out[16]:

	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	146.412162	360.0	1.0
1	LP001003	Male	Yes	1	Graduate	No	4583	1508.0	128.000000	360.0	1.0
2	LP001005	Male	Yes	0	Graduate	Yes	3000	0.0	66.000000	360.0	1.0
3	LP001006	Male	Yes	0	Not Graduate	No	2583	2358.0	120.000000	360.0	1.0
4	LP001008	Male	No	0	Graduate	No	6000	0.0	141.000000	360.0	1.0

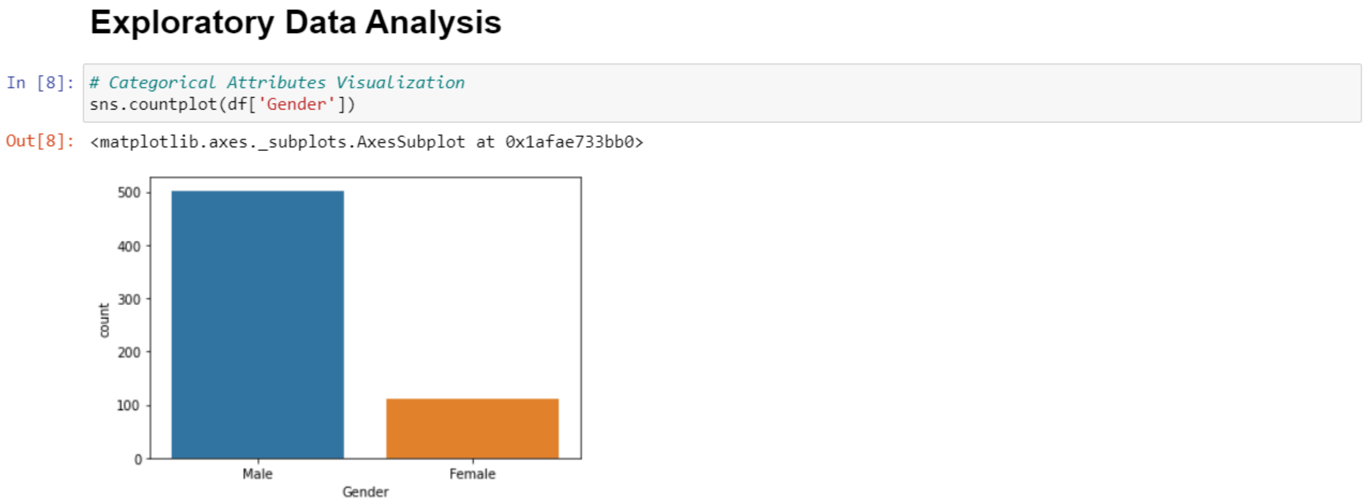
Step 3: Pre-processing of Dataset



```
In [7]: df.isnull().sum()
```

```
Out[7]: Loan_ID              0
        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
```

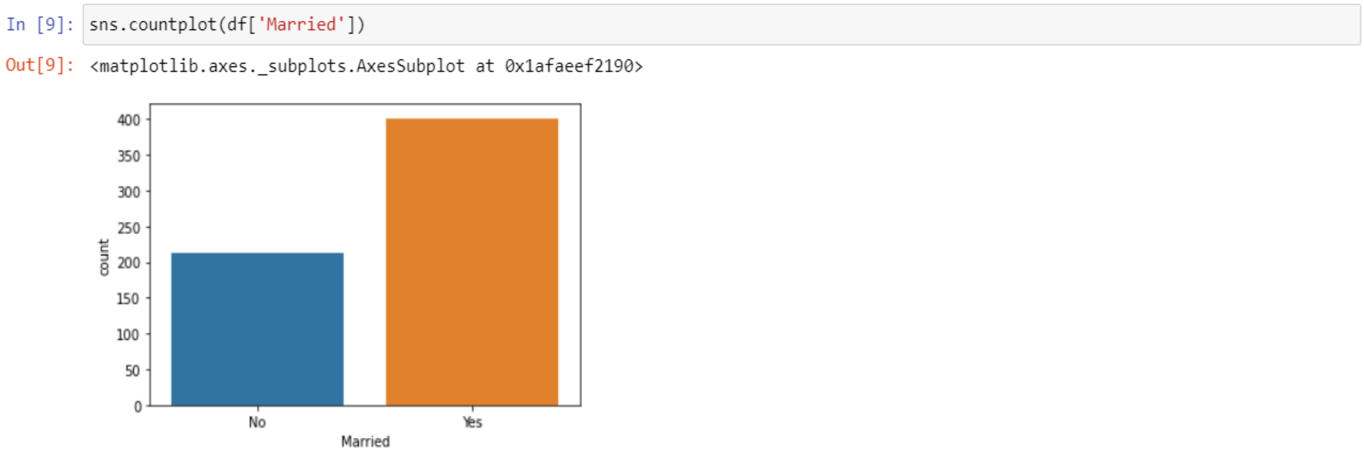
Step 4: Exploratory Data Analysis



Exploratory Data Analysis

```
In [8]: # Categorical Attributes Visualization
        sns.countplot(df['Gender'])
```

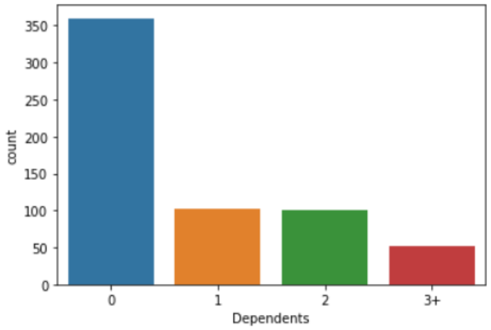
```
Out[8]: <matplotlib.axes._subplots.AxesSubplot at 0x1afae733bb0>
```



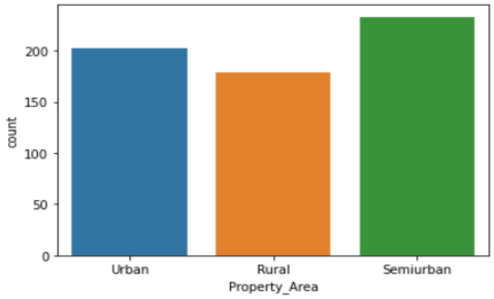
```
In [9]: sns.countplot(df['Married'])
```

```
Out[9]: <matplotlib.axes._subplots.AxesSubplot at 0x1afaeef2190>
```

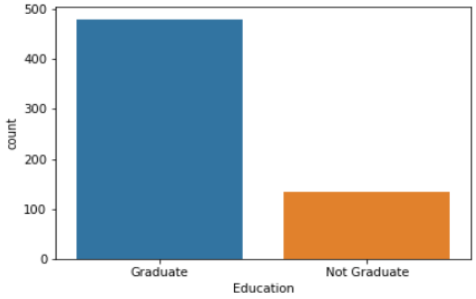
```
In [10]: sns.countplot(df['Dependents'])
Out[10]: <matplotlib.axes._subplots.AxesSubplot at 0x1afaef69310>
```



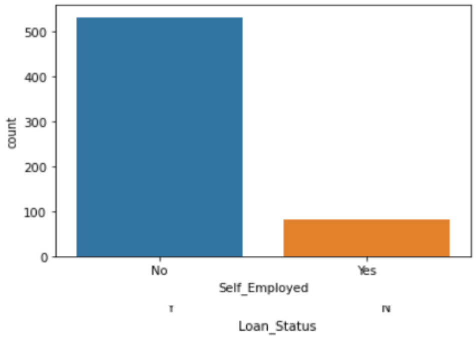
```
In [13]: sns.countplot(df['Property_Area'])
Out[13]: <matplotlib.axes._subplots.AxesSubplot at 0x1afaf052ac0>
```



```
In [11]: sns.countplot(df['Education'])
Out[11]: <matplotlib.axes._subplots.AxesSubplot at 0x1afaefbcb20>
```



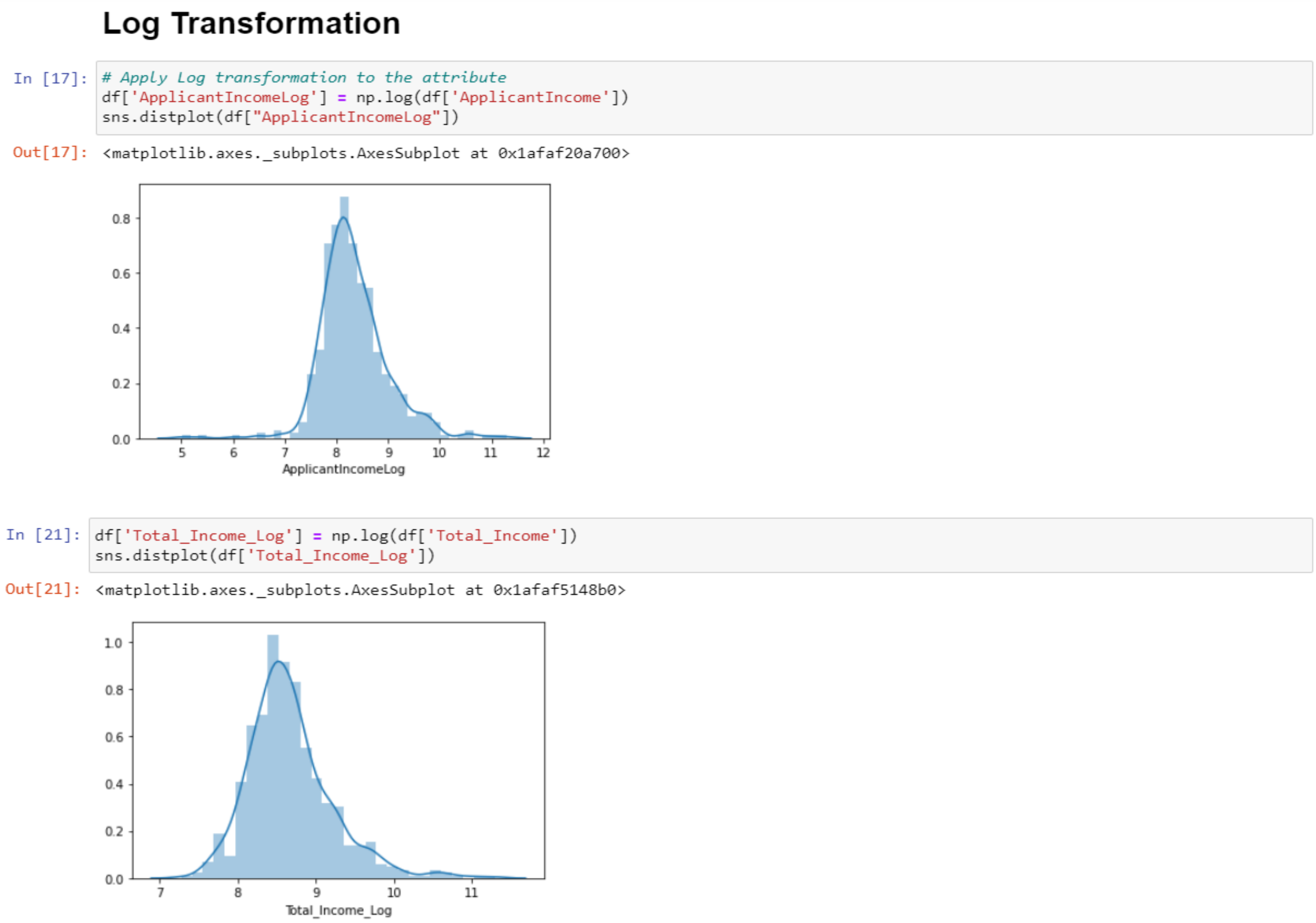
```
In [12]: sns.countplot(df['Self_Employed'])
Out[12]: <matplotlib.axes._subplots.AxesSubplot at 0x1afaf00f9a0>
```



Conclusions: (Through Single Variable Analysis)

1. *We can see that approximately 81% are Male and 19% are female.*
2. *The percentage of applicants with no dependents is higher.*
3. *There is more number of graduates than non-graduates.*
4. *Semi-Urban people are slightly higher than Urban people among the applicants.*
5. *A larger percentage of people have a good credit history.*
6. *The percentage of people that the loan has been approved has been higher rather than the percentage of the applicant for which the loan has been declined.*

Applying the log function will remove the skewness of data and will make it normal. As total income is skewed we have applied a log of that which makes it normal so that many machine learning algorithms can be applied smoothly.



Log Transformation

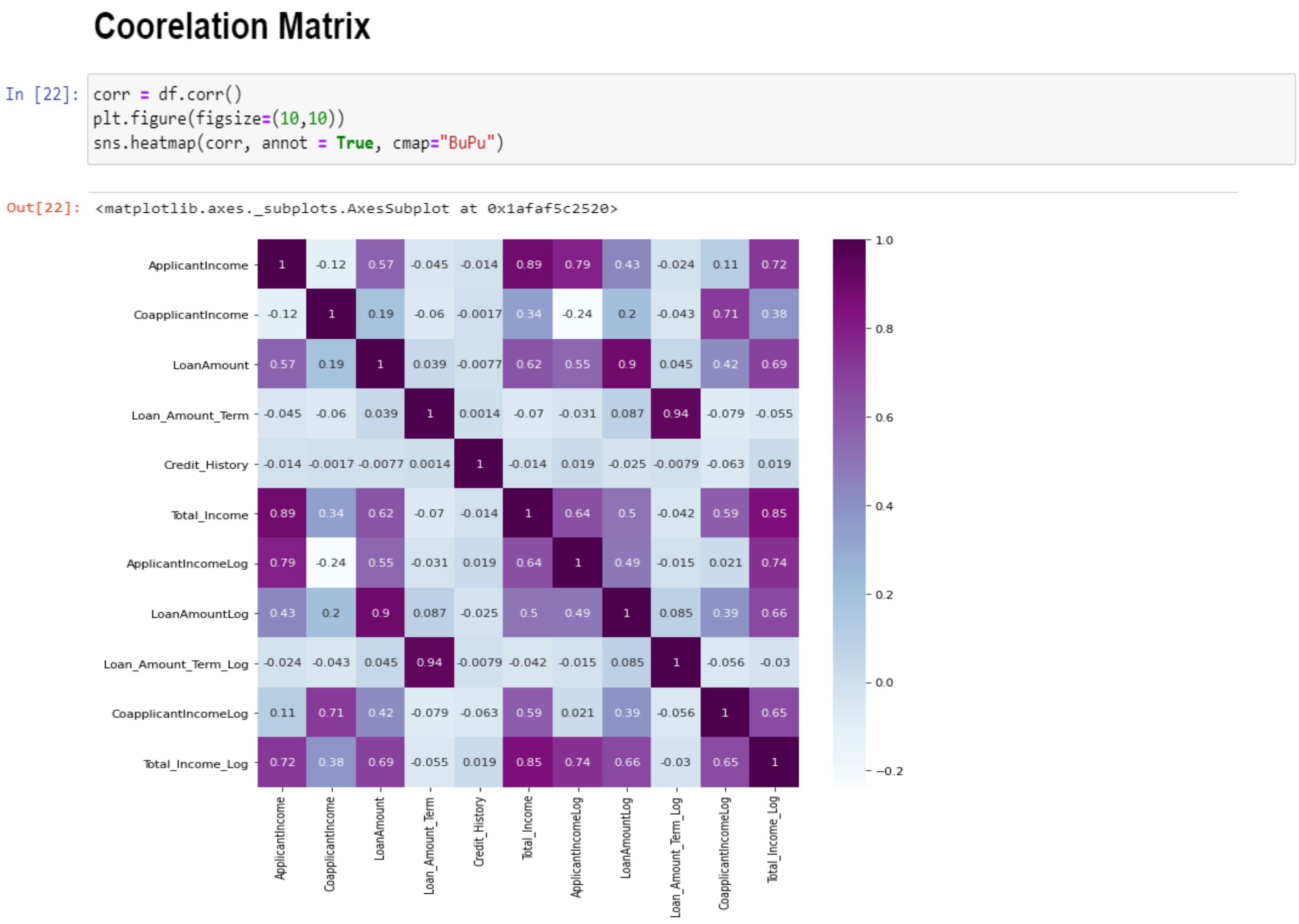
```
In [17]: # Apply Log transformation to the attribute
         df['ApplicantIncomeLog'] = np.log(df['ApplicantIncome'])
         sns.distplot(df["ApplicantIncomeLog"])

Out[17]: <matplotlib.axes._subplots.AxesSubplot at 0x1afaf20a700>
```

```
In [21]: df['Total_Income_Log'] = np.log(df['Total_Income'])
         sns.distplot(df['Total_Income_Log'])

Out[21]: <matplotlib.axes._subplots.AxesSubplot at 0x1afaf5148b0>
```

4.1e. Train-Test Split the data

Before starting to split the data for modeling, we need to check the relevant attributes by Correlation Matrix which will tell us the relation between pairs of variables in a dataset and the remaining attributes need to be in binary form to make it easy for the machine to train the data. This can be done by Label Encoding.

Step 6: Correlation Matrix



Coorelation Matrix

In [22]:
```
corr = df.corr()
plt.figure(figsize=(10,10))
sns.heatmap(corr, annot = True, cmap="BuPu")
```

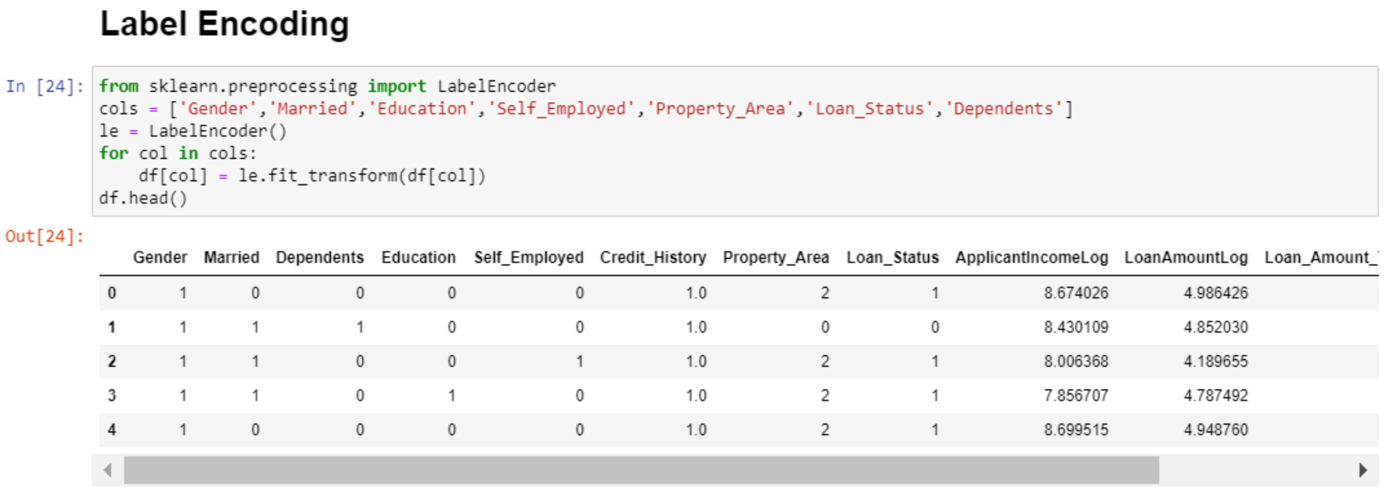
Out[22]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x1afaf5c2520>
```

	ApplicantIncome	CoapplicantIncome	LoanAmount	Loan_Amount_Term	Credit_History	Total_Income	ApplicantIncomeLog	LoanAmountLog	Loan_Amount_Term_Log	CoapplicantIncomeLog	Total_Income_Log
ApplicantIncome	1	-0.12	0.57	-0.045	-0.014	0.89	0.79	0.43	-0.024	0.11	0.72
CoapplicantIncome	-0.12	1	0.19	-0.06	-0.0017	0.34	-0.24	0.2	-0.043	0.71	0.38
LoanAmount	0.57	0.19	1	0.039	-0.0077	0.62	0.55	0.9	0.045	0.42	0.69
Loan_Amount_Term	-0.045	-0.06	0.039	1	0.0014	-0.07	-0.031	0.087	0.94	-0.079	-0.055
Credit_History	-0.014	-0.0017	-0.0077	0.0014	1	-0.014	0.019	-0.025	-0.0079	-0.063	0.019
Total_Income	0.89	0.34	0.62	-0.07	-0.014	1	0.64	0.5	-0.042	0.59	0.85
ApplicantIncomeLog	0.79	-0.24	0.55	-0.031	0.019	0.64	1	0.49	-0.015	0.021	0.74
LoanAmountLog	0.43	0.2	0.9	0.087	-0.025	0.5	0.49	1	0.085	0.39	0.66
Loan_Amount_Term_Log	-0.024	-0.043	0.045	0.94	-0.0079	-0.042	-0.015	0.085	1	-0.056	-0.03
CoapplicantIncomeLog	0.11	0.71	0.42	-0.079	-0.063	0.59	0.021	0.39	-0.056	1	0.65
Total_Income_Log	0.72	0.38	0.69	-0.055	0.019	0.85	0.74	0.66	-0.03	0.65	1

A correlation matrix is simply a table that displays the correlation. The measure is best used in variables that demonstrate a linear relationship between each other. By taking a view of this matrix we can remove some attributes which have a close match.

Step 7: Label Encoding

In label encoding in Python, we replace the categorical value with a numeric value between 0 and the number of classes minus 1. If the categorical variable value contains 5 distinct classes, we use (0, 1, 2, 3, and 4).



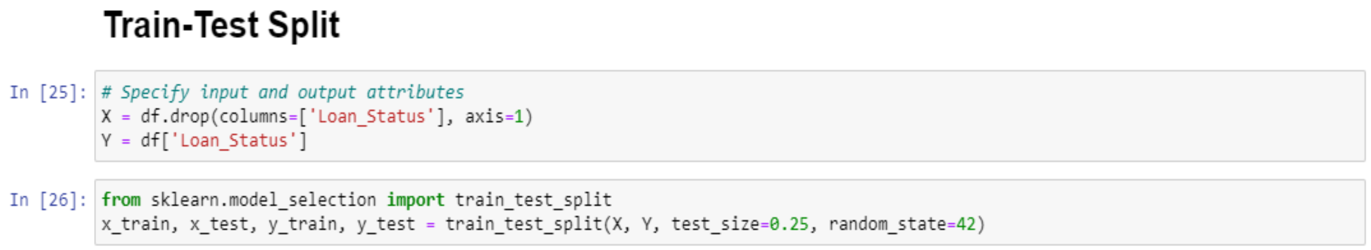
Label Encoding

```
In [24]: from sklearn.preprocessing import LabelEncoder
         cols = ['Gender','Married','Education','Self_Employed','Property_Area','Loan_Status','Dependents']
         le = LabelEncoder()
         for col in cols:
             df[col] = le.fit_transform(df[col])
         df.head()
```

Out[24]:

	Gender	Married	Dependents	Education	Self_Employed	Credit_History	Property_Area	Loan_Status	ApplicantIncomeLog	LoanAmountLog	Loan_Amount_
0	1	0	0	0	0	1.0	2	1	8.674026	4.986426	
1	1	1	1	0	0	1.0	0	0	8.430109	4.852030	
2	1	1	0	0	1	1.0	2	1	8.006368	4.189655	
3	1	1	0	1	0	1.0	2	1	7.856707	4.787492	
4	1	0	0	0	0	1.0	2	1	8.699515	4.948760	

Step 8: Train-Test Split

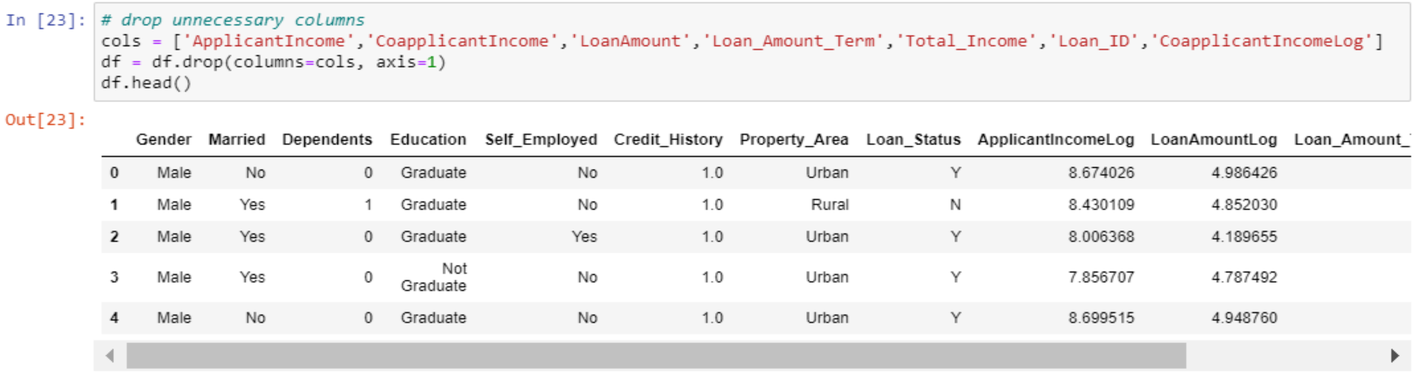


Train-Test Split

```
In [25]: # Specify input and output attributes
         X = df.drop(columns=['Loan_Status'], axis=1)
         Y = df['Loan_Status']
```

```
In [26]: from sklearn.model_selection import train_test_split
         x_train, x_test, y_train, y_test = train_test_split(X, Y, test_size=0.25, random_state=42)
```

We have used sklearn.model_selection for this project. train_test_split is a function in Sklearn model selection for splitting data arrays into two subsets: for training data and testing data. With this function, we don't need to divide the dataset manually. By default, Sklearn train_test_split will make random partitions for the two subsets.

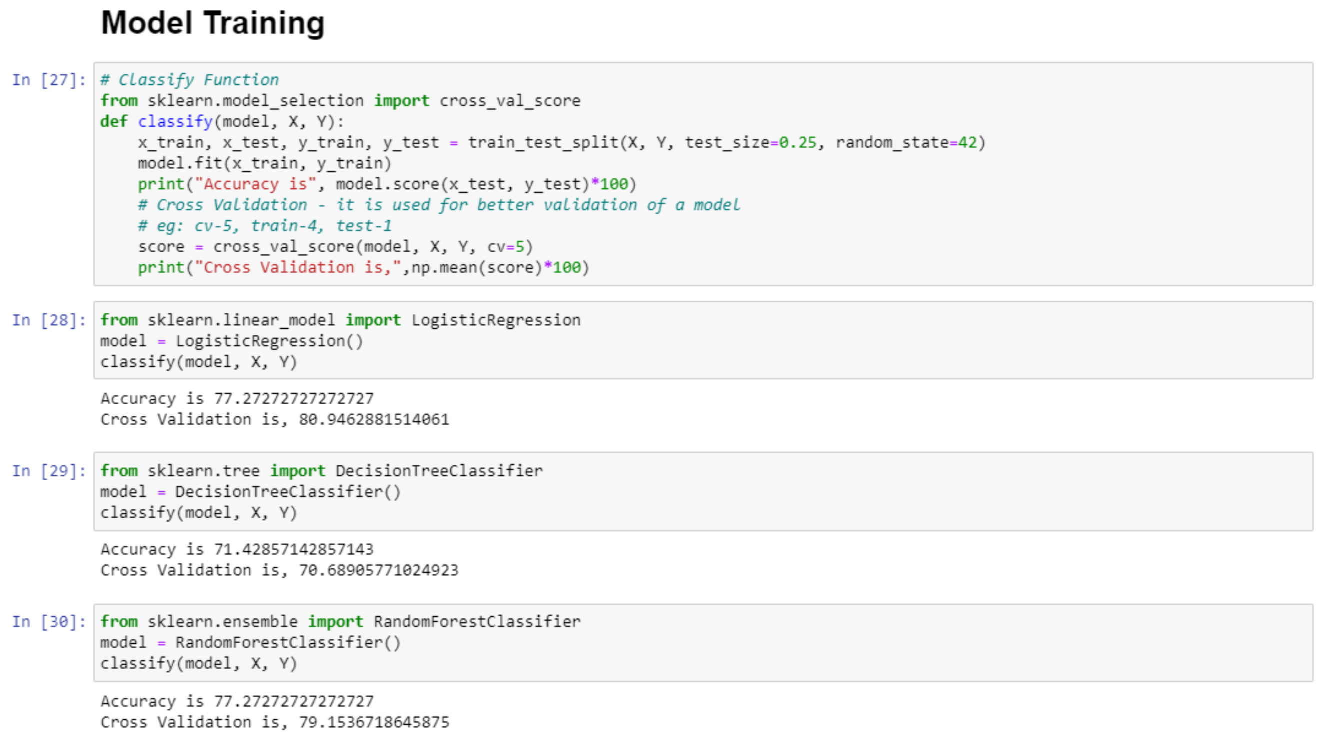


```
In [23]: # drop unnecessary columns
         cols = ['ApplicantIncome','CoapplicantIncome','LoanAmount','Loan_Amount_Term','Total_Income','Loan_ID','CoapplicantIncomeLog']
         df = df.drop(columns=cols, axis=1)
         df.head()
```

Out[23]:

	Gender	Married	Dependents	Education	Self_Employed	Credit_History	Property_Area	Loan_Status	ApplicantIncomeLog	LoanAmountLog	Loan_Amount_
0	Male	No	0	Graduate	No	1.0	Urban	Y	8.674026	4.986426	
1	Male	Yes	1	Graduate	No	1.0	Rural	N	8.430109	4.852030	
2	Male	Yes	0	Graduate	Yes	1.0	Urban	Y	8.006368	4.189655	
3	Male	Yes	0	Not Graduate	No	1.0	Urban	Y	7.856707	4.787492	
4	Male	No	0	Graduate	No	1.0	Urban	Y	8.699515	4.948760	

Step 9: Data Models and its Accuracy



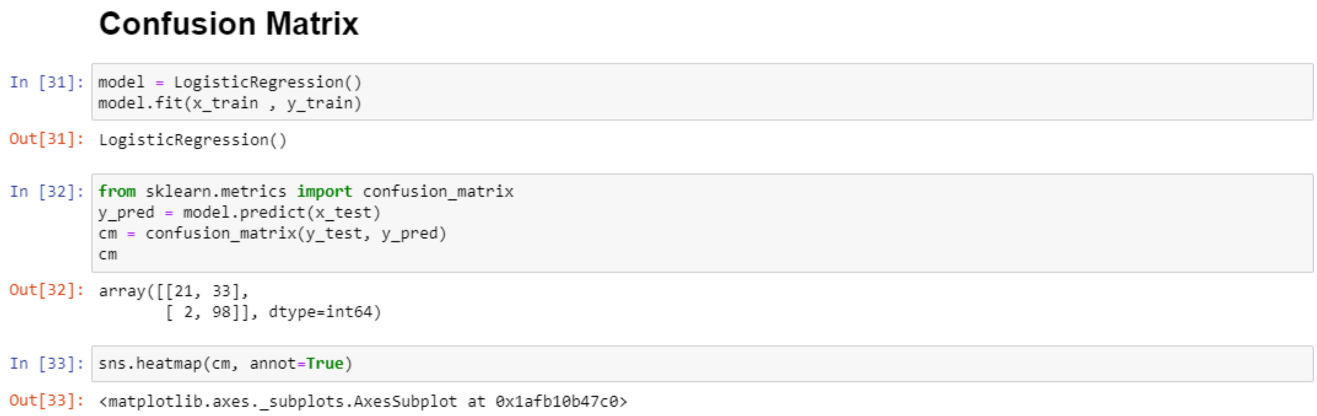
Model Training

```
In [27]: # Classify Function
         from sklearn.model_selection import cross_val_score
         def classify(model, X, Y):
             x_train, x_test, y_train, y_test = train_test_split(X, Y, test_size=0.25, random_state=42)
             model.fit(x_train, y_train)
             print("Accuracy is", model.score(x_test, y_test)*100)
             # Cross Validation - it is used for better validation of a model
             # eg: cv-5, train-4, test-1
             score = cross_val_score(model, X, Y, cv=5)
             print("Cross Validation is,",np.mean(score)*100)
```

```
In [28]: from sklearn.linear_model import LogisticRegression
         model = LogisticRegression()
         classify(model, X, Y)

         Accuracy is 77.27272727272727
         Cross Validation is, 80.9462881514061
```

```
In [29]: from sklearn.tree import DecisionTreeClassifier
         model = DecisionTreeClassifier()
         classify(model, X, Y)

         Accuracy is 71.42857142857143
         Cross Validation is, 70.68905771024923
```

```
In [30]: from sklearn.ensemble import RandomForestClassifier
         model = RandomForestClassifier()
         classify(model, X, Y)

         Accuracy is 77.27272727272727
         Cross Validation is, 79.1536718645875
```

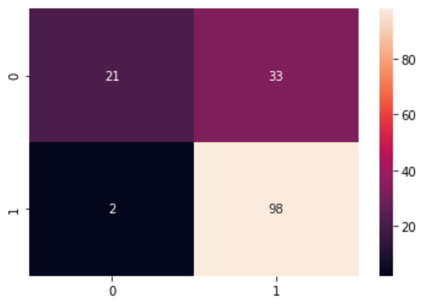


Confusion Matrix

```
 In [31]: model = LogisticRegression()
          model.fit(x_train , y_train)

Out[31]: LogisticRegression()
```

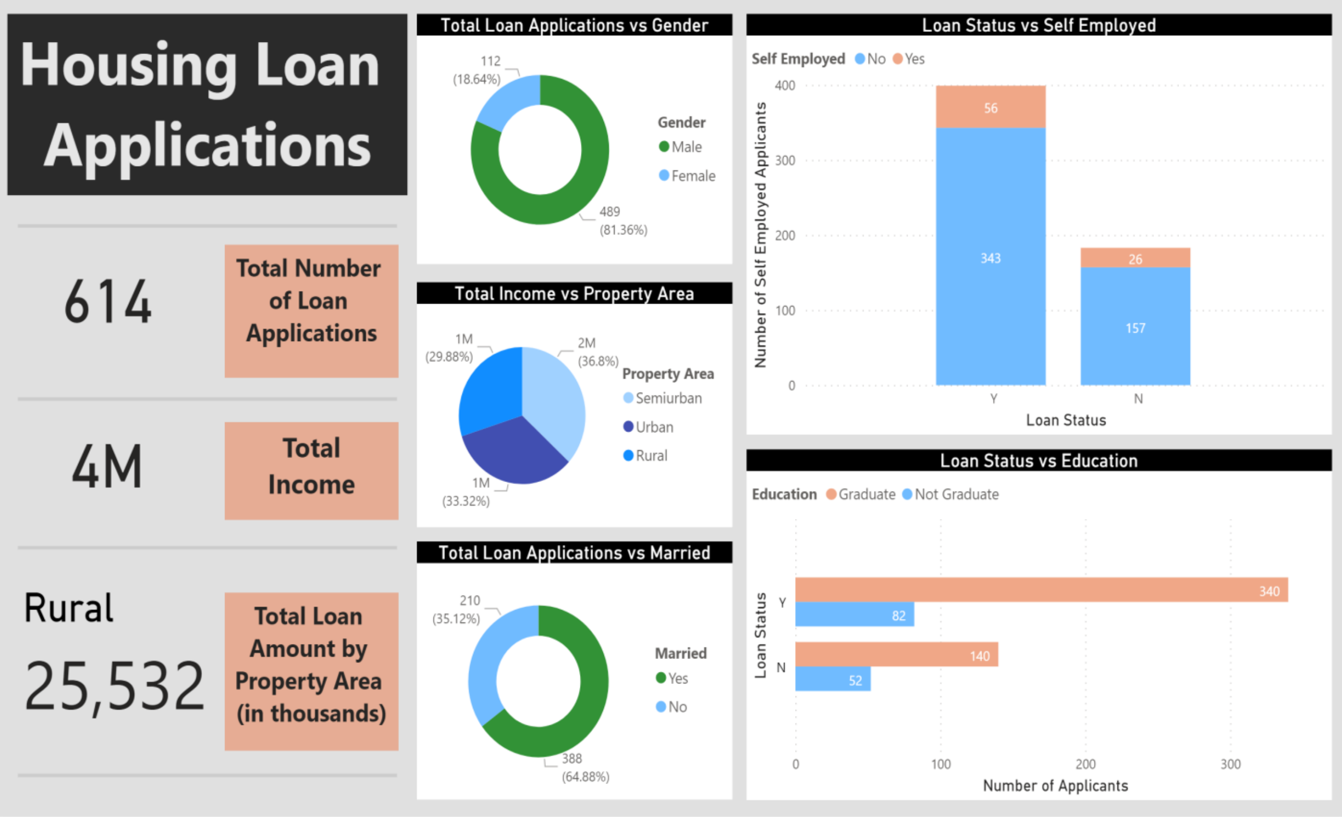
```
 In [32]: from sklearn.metrics import confusion_matrix
          y_pred = model.predict(x_test)
          cm = confusion_matrix(y_test, y_pred)
          cm

Out[32]: array([[21, 33],
                [ 2, 98]], dtype=int64)
```

```
 In [33]: sns.heatmap(cm, annot=True)

Out[33]: <matplotlib.axes._subplots.AxesSubplot at 0x1afb10b47c0>
```



4.2 Power BI Dashboard

As in Tableau, Power BI is utilized to display relationships between variables within the housing loan data set. To provide a summary, the Power BI Dashboard highlights three variables from the data set. These variables are the total number of loan applications, total income, and total loan amount by property area. Within the housing loan data set, 614 total applications have a total income of roughly four million. The total loan amount of applications from rural areas is 25,532,000, 32,251,000 from semi-urban areas, and 26,182,000 from urban areas.



The first relationship visualized with a donut chart is between the total number of loan applications and the gender of the applicants. The majority of loan applicants are male as are 81.36% of the applicants. There are a total of 112 applicants who identify as female and 489 applicants who identify as male. The following visualization illustrates the relationship between total income and property area. As previously stated, the total income variable was created to further analyze the total income of the application by combining the variables ApplicantIncome and CoapplicantIncome. The pie chart demonstrates that applicants from a semi-urban area have a higher total income than applicants from urban and rural areas. Surprisingly, there is not a significantly large difference in total income between each property area as

36.8% of the total income is from semi-urban areas, 33.32% is from urban areas, and 29.88% is from rural areas. The final donut chart analyzes the relationship between the number of applications and the marital status of the applicant, which states that roughly 65% of applicants are married and 35% are not married.

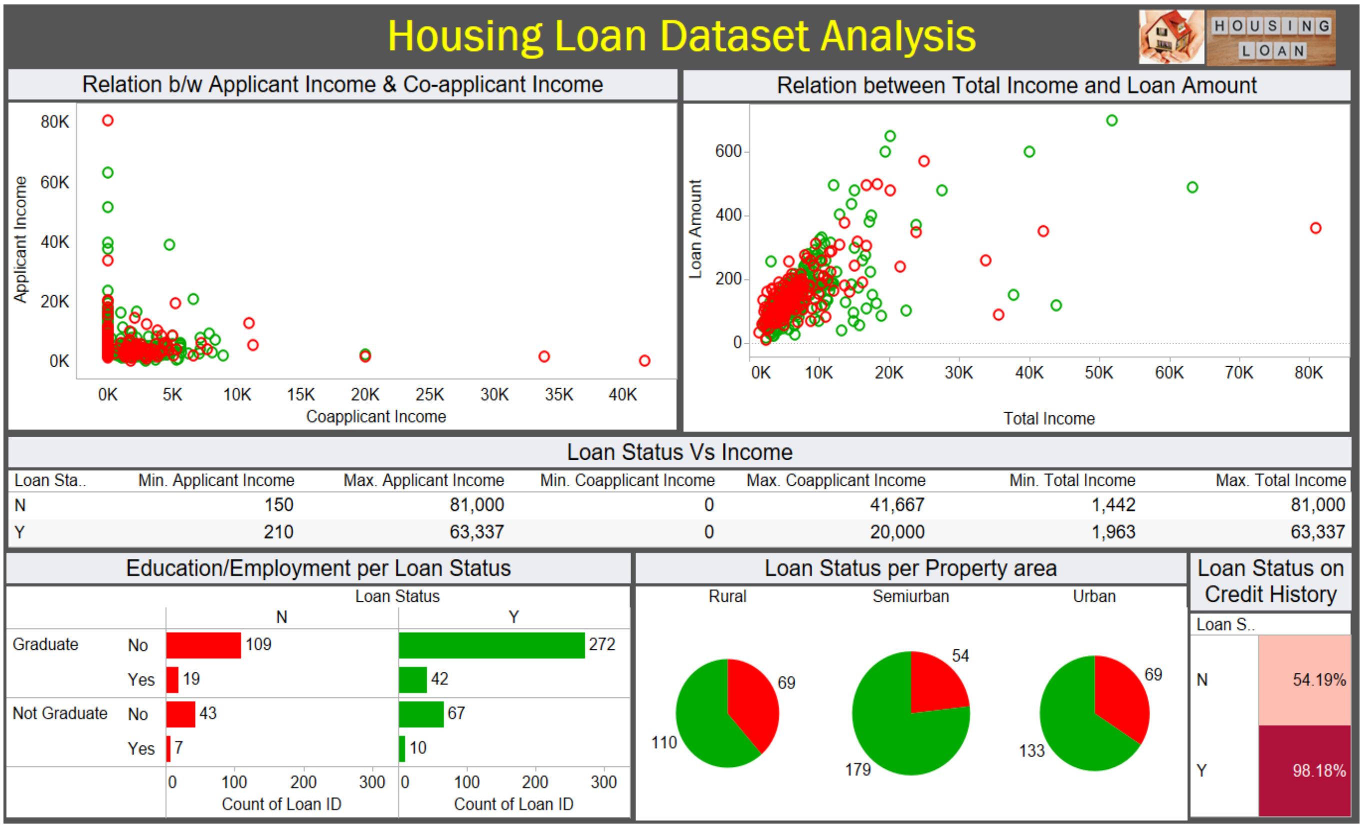
The Dashboard also contains two bar charts that display the relationships between the status of a loan, the applicant's level of education, and whether or not the applicant is self-employed. As stated in the first stacked bar chart, 56 self-employed applicants and 343 applicants who are not self-employed received a housing loan while 26 self-employed applicants and 157 applicants who are not self-employed were denied a housing loan. Therefore, more applicants who are not self-employed than those who are self-employed applied for a housing loan and were accepted. Lastly, the clustered bar chart shows the relationship between the applicant's loan status and their level of education, which is stated as "Graduate" or "Not Graduate". Out of 480 applicants who are graduates, 340 were granted a housing loan and 140 were denied. Out of 134 applicants who are graduates, 82 were granted a housing loan and 52 were denied. Hence, applicants who are considered graduates are more likely to be granted a housing loan than those who are not graduates.

4.3 Tableau Visualizations & Dashboard

Tableau is a data visualization tool first and foremost. Therefore, its technology is there to support complex computations, data blending, and dashboarding to create beautiful visualizations that deliver insights that cannot easily be derived from staring at a spreadsheet. It has climbed to the top of the data visualization heap because of its dedication to this purpose.

This is the reason we are using the Tableau tool for our visualization. In the below dashboard, there is a clear picture of the analysis made for Housing Loan from the attributes given in the dataset.

With this tableau analysis, we concluded that Loan approvals are not done based on a single variable. Loan Approvals are done by the combination of the data gathered about the applicant like credit history, Applicants income, Property Area, etc.



Loan Status Vs Income

Loan Sta..	Min. Applicant Income	Max. Applicant Income	Min. Coapplicant Income	Max. Coapplicant Income	Min. Total Income	Max. Total Income
N	150	81,000	0	41,667	1,442	81,000
Y	210	63,337	0	20,000	1,963	63,337

5. Conclusion

From a proper analysis of positive points and constraints on the project, it can be safely concluded that the project is highly efficient. This application is working properly. This component can be easily plugged into many other systems. There have been several cases of computer glitches, errors in content and the most important weight of features is fixed in an automated prediction system. In near future, this module of prediction can be integrated with the module of the automated processing system. We can also enhance this model accuracy by training the model on the historical loan data. A similar model can be transformed to use for other lines of business in the company like auto loans, refinance loans, etc. Integrate voice assistant to this model so the customers can determine their loan eligibility without having to visit a branch.

6. References

Amit Prajapati. Loan Prediction. Retrieved on March 28, 2021, from https://www.kaggle.com/ninzaami/loan-predication

7. Required Files

Loan Prediction -
Final Team Project .p

Tableau Dashboard:
https://public.tableau.com/profile/sucharitha3435#!/vizhome/LoanPrediction_FinalProject/Dashboard1?publish=yes

Power BI Dashboard:
https://app.powerbi.com/groups/me/reports/a7e28170-e83b-4121-9a06
8eb6d8fd3a58/ReportSection?ctid=c61c89fe-c3bc-436b-8c76-52493f91ff5a