LOAN APPROVAL PREDICTION

Simran Goswami

Department of Statistics, Faculty of Mathematical Sciences, University of Delhi
North Campus, Delhi - 110007

Author E-mail: simran.goswami6.stats@gmail.com

INTRODUCTION

Financial Institutions make various loans and advances to businesses, corporations, and individuals. The interest on these loans is their primary source of income. They grant a loan after a lengthy verification process. However, it does not guarantee that the applicant will be able to repay the loan without difficulty. Thus, the risks associated with making a loan approval decision are enormous. Through this research, the student researcher hopes to develop a model that can provide our client "Dream Housing Finance" with a fair idea about their customers' segments who are eligible for the home loan amounts. It would also allow them to automate the loan eligibility procedure in real-time based on the information provided by a consumer through an application form.

RESEARCH STATEMENT

To build a model that predicts the probability of getting approved for a loan given financial information, so as to classify the customers.

DATA

The secondary research method is used for the research purpose (Source: Datahack). The dataset is structured which means that the target is already defined. It contains home loan application details of **614** customers of "Dream Housing Finance" on several parameters which are considered important during the application for loans. The parameters included are:

S.No	VARIABLE	DESCRIPTION
1	Loan_ID	Unique Loan ID
2	Gender	Male/ Female
3	Married	Applicant's marital status (Y/N)
4	Dependents	Number of dependents
5	Education	Applicant Education (Graduate/ Undergraduate)
6	Self_Employed	Self-employed (Y/N)
7	ApplicantIncome	Applicant's income
8	CoapplicantIncome	Co-applicant's income
9	LoanAmount	Sum of money that is lent (in 1000s)
10	Loan_Amount_Term	Term of a loan in months
11	Credit_History	Credit history meets guidelines
12	Property_Area	Urban/ Semi-Urban/ Rural
13	Loan_Status	Loan approved (Y/N)

CONCEPTUAL KNOWLEDGE

1. **Binary Logistic Regression - The Supervised Classification Algorithm**

It is used for classification problems, i.e. to classify and predict in which category our data will fall inside pr what class an observation belongs to.

It is used when the response variable is categorical, i.e., it is a binary variable (**dichotomous in nature**) containing data coded as 1 (Y) or 0 (N). The idea of Logistic Regression is to find a relationship between features and the probability of a particular outcome. Based on the probability, a classification of 1 (positive class) or 0 (negative class) will be given.

In simple words, here, we have a dataset that has multiple features and a categorical target variable. The main goal is to create a model that predicts the probability of each class target variable based on the features.

Logistic Regression is a **generalized linear model** (GLM).

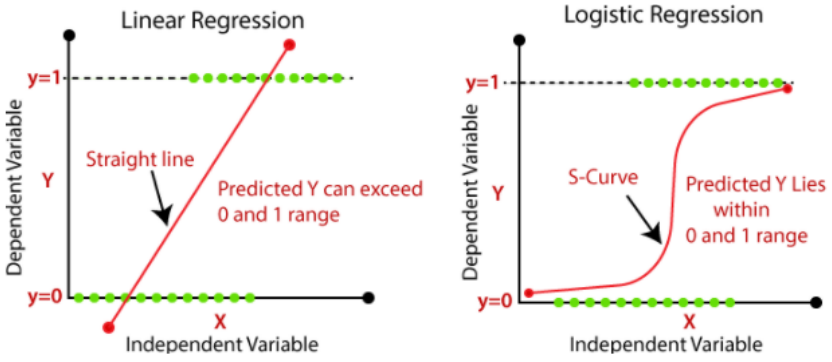
2. **How is it different from Linear Regression?**

In regression, we deal with a problem with a number as a result. While, in classification, we deal with problems with a discrete label as a result.

One of the assumptions of Linear Regression is that the target variable should follow Normal Distribution. In classification problems, the target variable is categorical that has 2 categories, and is following **Binomial Distribution**, hence violates the assumption of LR.

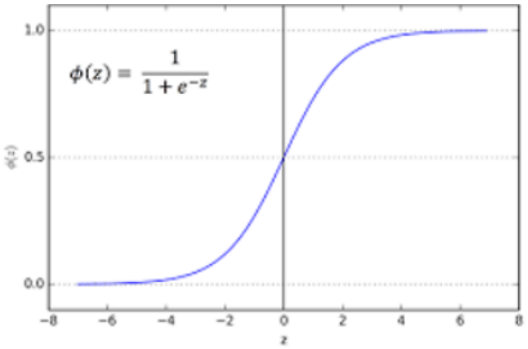
In addition to this, there would be some **relationship** between X and Y in classification problems, however, that wouldn't be linear according to the Linear Regression assumption. **[logit fn is linear with X]**

Moreover, the output from Linear Regression can have negative values and values greater than one as well. This will exceed the range of predicted values for Y from 0 and 1.



3. **Mathematical Model of Binary Logistic Regression**

Logistic Regression is a **Sigmoid** function, i.e., a mathematical function that takes on a characteristic "S" shaped curve. The Sigmoid function always gives values in the range of 0 to 1. Mathematically;



$$g(x) = \frac{1}{1+e^{-x}}$$

Step 1: Let us consider that we have a dependent categorical variable **"Y"** and an independent variable **"X"**. We want to **predict the probabilities for each class of the variable** Y using the independent variable X.

Step 2: Denoting the probability for a specific class as P(Y|X) or P(Y) which lies in between 0 and 1.

$$P(Y|X) = P(Y) = \beta_0 + \beta_1 X$$

Step 3: On passing this equation to the sigmoid function.

$$g(X) = \frac{e^{\beta_0+\beta_1 X}}{1+e^{\beta_0+\beta_1 X}}; \; 0 < g(x) < 1$$

Where;

- β_0 controls the location of the midpoint of the S-shaped curve (Intercept)
- β_1 controls the slope of the rise (Slope)

Step 4: The **method of maximum likelihood*** (In maximum likelihood estimation, we maximize likelihood function and accuracy by minimizing the error) to fit this model and try to find the estimates $\widehat{\beta_0}$ and $\widehat{\beta_1}$ as close as to the true values.

$$\text{odds} = \frac{p(X)}{1-p(X)} = e^{\beta_0+\beta_1 X}; \; 0 < \text{odds} < \infty$$

In simple terms, **odds** means the ratio of the probability that an event will occur i.e. p(x), and the probability that an event will not occur i.e. 1-p(x).

***likelihood function** refers to the joint pdf of sample observations.

Step 5: Taking logarithm on both sides of the above equation.

$$\text{Logit function} = \log(\text{odds}) = log\left(\frac{p(X)}{1-p(X)}\right) = \beta_0 + \beta_1 X \; ; -\infty < \log \text{odds} < \infty$$

This is the equation for linear regression in X with parameters β_0 and β_1.

Where;

- $log\left(\frac{p(X)}{1-p(X)}\right)$ is the **"logit function" or "log odds"**. Thus, here, the logit function is linear in X.
- It can be concluded that 1 unit in the increase in X results in β_1 units change in log odds.
- **Odds Ratio** - Represents the constant effect of X, on the likelihood that one outcome will occur.

Thus, Logistic regression is in reality an ordinary linear regression using the logit as the response variable. The logit transformation allows for a linear relationship between the response variable and the coefficients

Step 6: To make **predictions**, we put the test data in the above equation and get the value of log odds. On applying the exponent function on log-odds, we can transform it to obtain the value of **p(X)**.

- If the value of p(X) is too low **(or close to 0)**, we classify it as **"one class"**.
- If the value of p(X) is too high **(or close to 1)**, we classify it as **"another class"**.

We can also decide on a **threshold**. For example: If p(X) < 0.5 then it is termed as p(X) = 0 and if p(X) > 0.5 then it is termed as p(X) = 1. However, the threshold is decided on the basis of domain knowledge and type of dataset and is finalized using the ROC-AUC curves.

4. Assumptions of Binary Logistic Regression

- Observations should be independent and the dependent variable is binary or dichotomous.
- There should not be any multicollinearity present in the input data.
- There should not be any outliers present in the data.
- There should be a linear relationship between log(odds) or the logit function and the input variables.
- Logistic regression requires fairly large sample sizes.

5. Evaluation Metrics of Classification

Our primary aim is to build a model that increases the True Positive (TP) and True Negative (TN) while decreasing the False Negative (FN) and False Positive (FP). It is valuable to assess the efficacy of an algorithm. The performance metrics used to evaluate a classification algorithm are as follows:

- **Confusion Matrix**

A confusion matrix is a matrix that will convey your model's right and wrong predictions on data.

		Predicted	
		Positive	Negative
Actual	Positive	TP	FN
	Negative	FP	TN

Confusion Matrix

Where;

True Positive (TP): A true positive is an outcome where the model correctly predicts the positive class.
True Negative (TN): A true negative is an outcome where the model correctly predicts the negative class.
False Positive (FP): A false positive is an outcome where the model incorrectly predicts the positive class.
False Negative (FN): A false negative is an outcome where the model incorrectly predicts the negative class.

- **Model Accuracy**

Model accuracy in terms of classification models can be defined as the ratio of correctly classified samples to the total number of samples:

$$\text{Accuracy} = \frac{\text{Number of correct predictions}}{\text{Total number of predictions}}$$

Or for binary classification models, the accuracy can be defined as:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

Note: Accuracy may not be sufficient to ensure the model's performance as it completely ignores the FP and FN values. In the case of class-imbalanced data, we do not use accuracy because it gives deceptive results about the model's performance.

For Example: Consider a Binary Classification Model to Predict whether the patient has cancer or not. The training dataset of 100 cases has imbalance classes, such that **10** are labeled as **'Cancer'**, and **90** are labeled as **'Normal'**. We fit a model that does not take into consideration the whole dataset rather predicts a single class for each observation i.e. **'Normal'**.

Mathematically, the accuracy would come out to be: 90/[100] = **90% accuracy**

Inferentially, it means 90 out of 100 were correctly classified. This highly accurate model may not be useful, as it isn't able to predict the actual cancer patients as it does not focus on **FN** and **FP** values rather takes into consideration only corrected predicted values which may be high due to imbalanced data.

- **Precision (Predictive Value)**

It is the number of true positives (the number of items correctly labeled as belonging to the positive class) divided by the total number of elements labeled as belonging to the positive class (the sum of true positives and false positives, which are items incorrectly labeled as belonging to the class).

In simple words, precision says, out of total predicted yes values, how many were actually correct yes values.

Here, precision says, of the applicants classified as approved, how many were actually approved?

$$\text{Precision} = \frac{TP}{(TP+FP)}$$

High precision means that an algorithm returns substantially more relevant results.

For Example: In the above example only, the precision would come out to be:

$$\text{Precision} = \frac{1}{(1+1)} = 0.5$$

Therefore, through precision, we can say when the model predicts cancer, it's correct 50% of the time.

- **Recall (Sensitivity)/(TPR)**

It is the number of true positives divided by the total number of elements that actually belong to the positive class (i.e. the sum of true positives and false negatives, which are items that were not labeled as belonging to the positive class but should have been).

In simple words, precision says, out of total actual yes values, how many are predicted as yes values.

Here, recall says, of the applicants who are actually approved for loan, how many are classified approved?

$$\text{Recall} = \frac{TP}{(TP+FN)}$$

High recall means that an algorithm returned most of the relevant results.

For Example: In the above example only, the recall would come out to be:

$$\text{Recall} = \frac{1}{(1+8)} = 0.11$$

Thus, through recall, we can say the model correctly identifies only 11% of all cancer patients.

- **Precision vs Recall**

To fully evaluate the effectiveness of a model, it's necessary to examine both precision and recall. Unfortunately, precision and recall are often in conflict. That is, improving precision typically reduces recall and vice versa.

By understanding our problem, we could decide whether high precision or high recall is most desirable.

Use Case	High Precision	High Recall	Consequences
Identification of cancer	Desirable	Desirable	**Low Precision**- unnecessary medical treatment for FP cases **Low Recall**- undetected cancer patient's treatment get delayed
Identifying a good candidate for hiring	Desirable	Relaxed	**Low Precision**- Wrong candidate may be chosen **Low Recall**- Some good candidate may be left behind but the cost of hiring the wrong candidate is more in this case
Predicting Truck Driver Accidents	Relaxed	Desirable	**Low Precision**- Just extra cost of preventive training to low-risk driver **Low Recall**- Miss out the accident-prone driver and may end up in major accidents

- **F-1 score**

F1 score should be used when both precision and recall are important for the use case. It helps us get the best of both metrics (Precision & Recall). F1 score is the harmonic mean (to mitigate the impact of large outliers and aggravate the impact of small ones) of precision and recall. It lies between [0,1]. An F1 score punishes extreme values more.

$$F1\ Score = \frac{2 * Precision * Recall}{Precision + Recall}$$

F-1 score is derived from **F-Beta** Score,(for Beta = 1). **F-Beta** score is the weighted harmonic mean of precision and recall.

$$F_\beta\ Score = \frac{(1+\beta^2) * Precision * Recall}{(\beta^2 * Precision) + Recall}$$

- If both False Positives (FP) and False Negatives (FN) are important then **β = 1.**
- If False Positive (FP) is important then β lies between 0 and 1.
- If False Negative (FN) is important then **β > 1.**

- **ROC (Receiver Operating Characteristic Curve) & AUC (Area Under Curve)**

If you are implementing a logistic regression, and your model has predicted some probabilities. So, what is the threshold that you decide for the model to classify observations in one class or another? In this case, the use case plays a crucial role, such that in the case of medical research, you want a higher **TPR** and a lower FPR.

Let us first understand the 2 terms which plays a crucial role in deriving the ROC curve:

Sensitivity (TP): It tells us what percentage of observations with **positive** results were correctly identified. Thus, it measures how many observations of **positive class** are actually labelled correctly. It is known as recall.

Specificity (TN): It tells us what percentage of observations with **negative** results were correctly identified. Thus, it measures how many observations of **negative class** are actually labelled correctly.

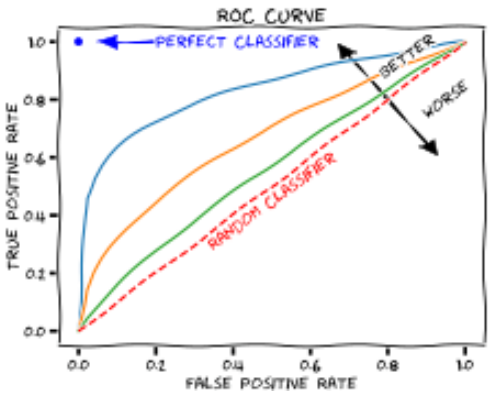
$$\text{Sensitivity} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}}$$

$$\text{Specificity} = \frac{\text{True Negatives}}{\text{True Negatives} + \text{False Positives}}$$

- AUC-ROC Curve is a performance metric **(ideal for Binary Classification)** that is used to measure the performance for the classification model at different **threshold values**. It summarizes the model's performance by evaluating the tradeoff between **TPR** and **FPR.**

- **ROC** curve is plotted b/w True Positive Rate (**TPR**) on **y-axis** and False Positive Rate (**FPR**) on **x-axis**.

- The higher the value of AUC, the better is our classifier in predicting the classes. As **FPR** increases, the **AUC** decreases, and model performance also decreases. Ideally, we want our model to have high TPR and low FPR i.e. **(TPR ~ 1 and FPR ~ 0)**



- The **higher** the value of **AUC**, the **better** is our classifier in predicting the classes.

For Example:

- An excellent classifier has an AUC value near 1 **[coordinate (0,1)]**, representing 100% sensitivity (no FN) and 100% specificity (no FP)

- A poor-performing classifier has an AUC value near 0.

Note: A classifier with an AUC score of 0.5 doesn't have any class separation capacity. When AUC is approximately 0.5, the model has no discrimination capacity to distinguish between positive class and negative class.

$$TPR = \frac{TP}{TP + FN}$$

True Positive rate

$$FPR = \frac{FP}{FP + TN}$$

False Positive Rate

ANALYSIS METHOD USED

In this research, a positive instance refers to YES (signifying the loan is approved as there will be no default in the payment of the loan), whereas the negative instance refers to NO (signifying the loan is approved as there will be default in the payment of the loan).

The proposed model focuses on predicting customers' loan repayment credibility by analyzing their behavior. The input to the model is the customer behavior collected. The classifier output can be used to decide whether to approve or reject the customer's home loan request.

Every new applicant's information entered on the application form serves as a test data set. Following the testing operation, the model predicts whether the new applicant is a fit case for loan approval or not based on the inferences it draws from the training data sets. To extract critical information and forecast whether or not a customer will be able to repay his loan. To achieve the most accurate results, the student researcher will go through several steps, including **data preprocessing** (imputing null values, duplicate values check, encoding categorical columns, and handling the imbalanced data), and **model selection**, to train a binary logistic regression model, make predictions, and measure its performance.

We have built a **Logistic Regression Model** on Python using an open-source library - scikit-learn (sklearn).

The major steps we employed in developing the machine learning tasks/algorithms are further discussed below:

1. **Collection & Preparation of input data:** Done by the owners of the dataset. (Source: Datahack)

2. **Analysis of input data:** To comprehend the interrelationships between various features A graph depicting the primary features as well as the whole dataset. The dataset is then divided into two parts; one for training and the other for testing the algorithms. Furthermore, in order to obtain a representative sample, each class in the full dataset is represented in about the right proportion in both the training and testing datasets.

3. **Train the algorithm:** The logistic regression algorithms are trained using a different set of data.

4. **Test the algorithm:** In evaluating the performance of the classification algorithms, It includes accuracy, precision, recall, specificity, and F-measure. These values are calculated using the Python scikit- learn tool with input values as the entities of the confusion matrix.

ANALYSIS

PART-I: EXPLORATORY DATA ANALYSIS

Exploratory Data Analysis refers to the critical process of performing initial investigations on data so as to discover patterns, spot anomalies, test hypotheses, and check assumptions with the help of summary statistics and graphical representations.

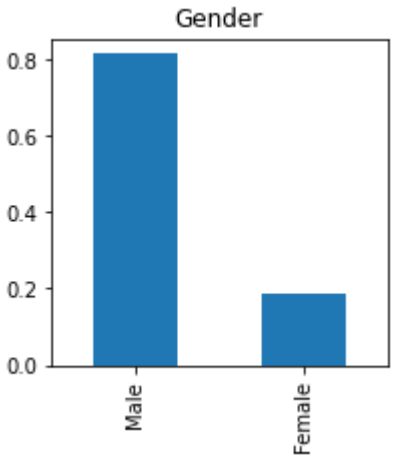
1.1. UNIVARIATE ANALYSIS

Feature 1: Loan_ID

- The Loan ID is typically used to uniquely identify an application, but it has no bearing on the loan status. As a result, we can ignore the Loan ID field when making a prediction.

Feature 2: Gender

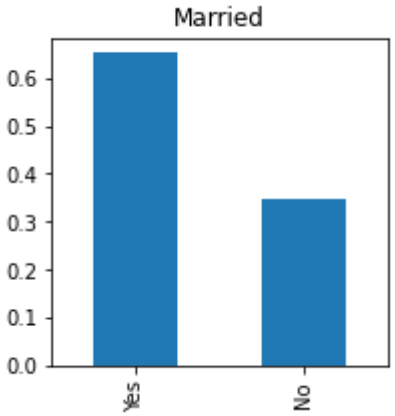
- Data Type: Nominal (Qualitative Data)
- The 2 unique values are Male and Female. Around 80% of applicants in the dataset are male.



- Missing Value Status: Yes (13)
- Missing Value Imputation: Mode (Highest Frequency) i.e. Males

Feature 3: Married

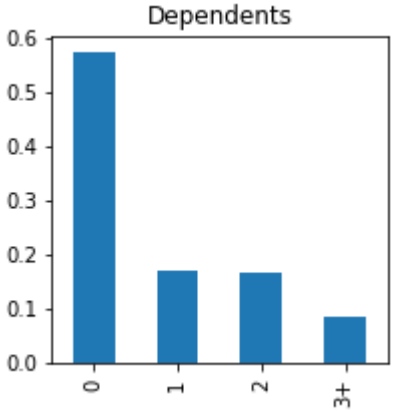
- Data Type: Nominal (Qualitative Data)
- The 2 unique values are Yes (signifying Married) and No (signifying Unmarried, separated, or divorced). Around 65% of the applicants in the dataset are married.



- Missing Value Status: Yes (3)
- Missing Value Imputation: Mode (Highest Frequency) i.e. Married

Feature 4: Dependents

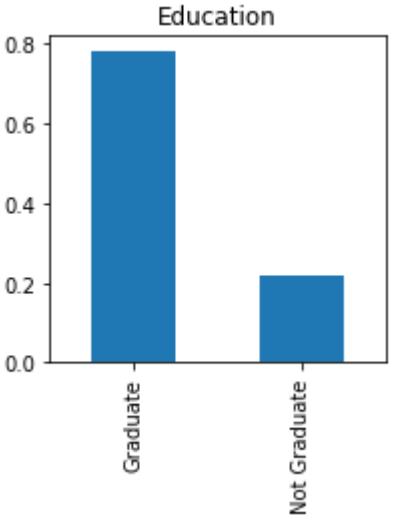
- Data Type: Ordinal (Qualitative Data)
- The 4 unique values are 0, 1, 2, and 3+. More than half of the applicants don't have any dependents.



- Missing Value Status: Yes (15)
- Missing Value Imputation: Mode (Highest Frequency) i.e. 0

Feature 5: Education

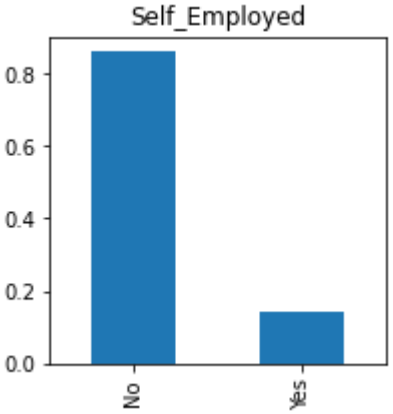
- Data Type: Nominal (Qualitative Data)
- The 2 unique values are Graduate and Not Graduate. Around 80% of the applicants are graduates.



- Missing Value Status: No

Feature 6: Self-Employed

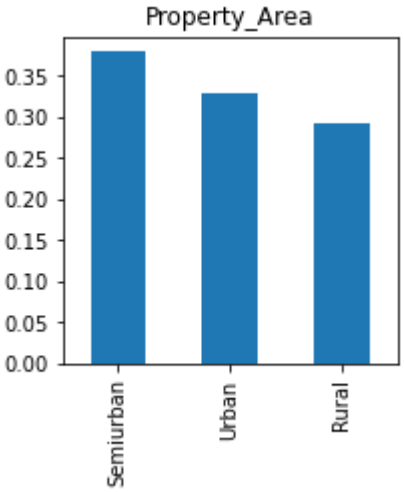
- Data Type: Nominal (Qualitative Data)
- The 2 unique values are Yes and No. Around 15% of applicants in the dataset are self-employed.



- Missing Value Status: Yes (32)
- Missing Value Imputation: Mode (Highest Frequency) i.e. No

Feature 8: Property Area

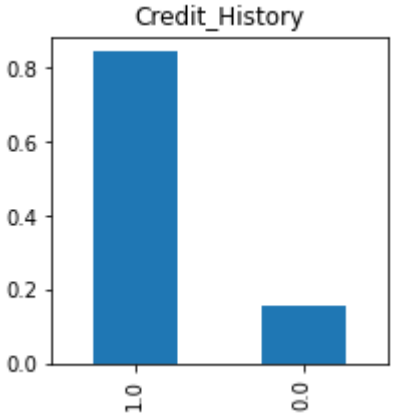
- Data Type: Ordinal (Qualitative Data)
- The unique values are Urban, Rural, and Semi-urban. Most of the applicants are from semi-urban areas.



- Missing Value Status: No

Feature 7: Credit History

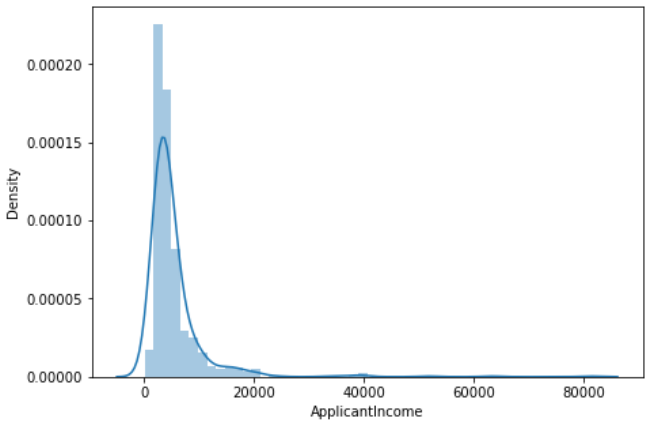
- Data Type: Float (Quantitative Data)
- The 2 unique values are 1 and 0. Around 85% of applicants have a credit history i.e. repaid their debts.

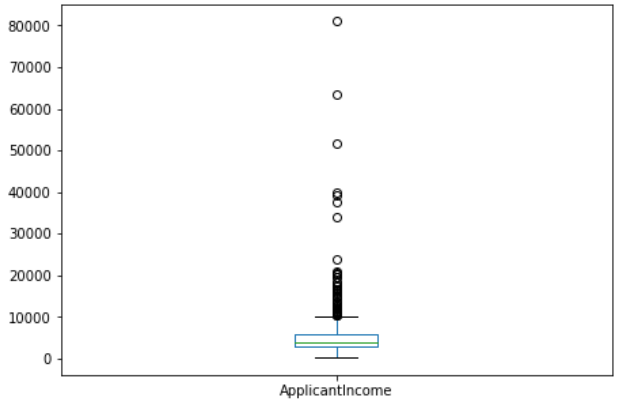


- Missing Value Status: Yes (50)
- Missing Value Imputation: Mode (Highest Frequency) i.e. 1

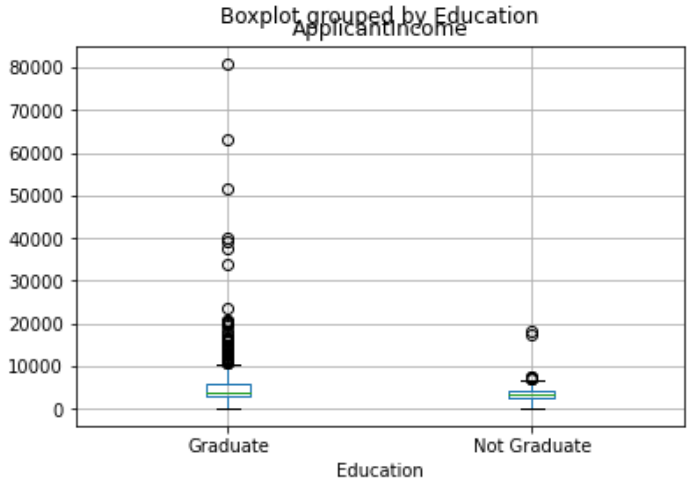
Feature 9: Applicant Income

- Data Type: Numerical (Quantitative Data)
- From the Distplot and Boxplot below, it can be concluded that the distribution of the feature "ApplicantIncome" is positively skewed, as well as, there is the presence of outliers. This can be attributed to the income disparity in society.





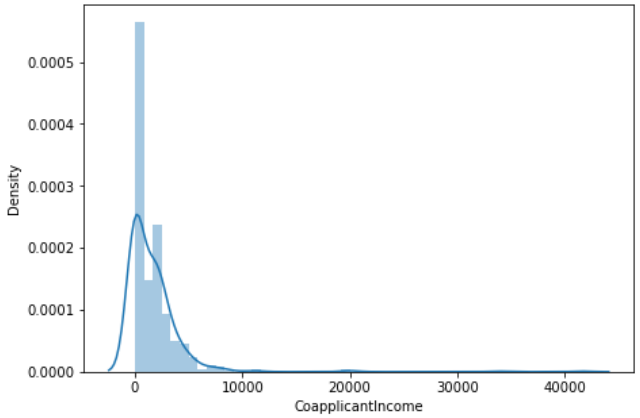
- Missing Value Status: No

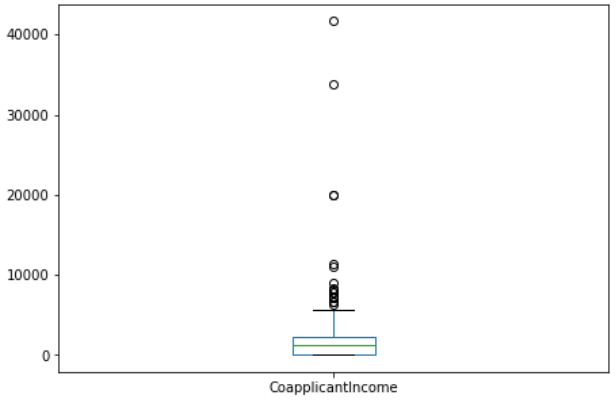


In the boxplot above in between ApplicantIncome and Education, it can be concluded that there is no substantial in between the median income of graduates and non-graduates. However, this confirms that people who are graduates have higher incomes.

Feature 10: Co-applicant Income

- Data Type: Numerical (Quantitative Data)
- Same as quoted above in the feature "Applicant Income".

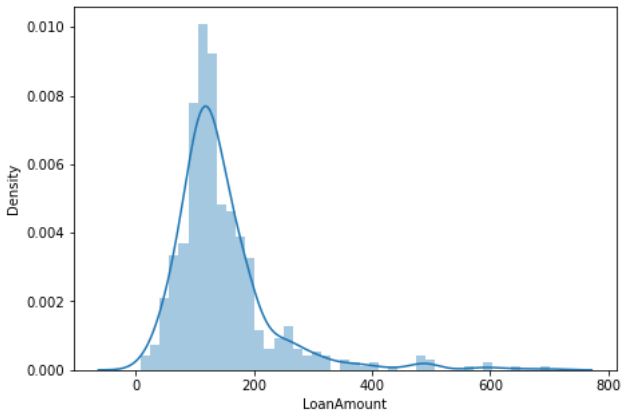


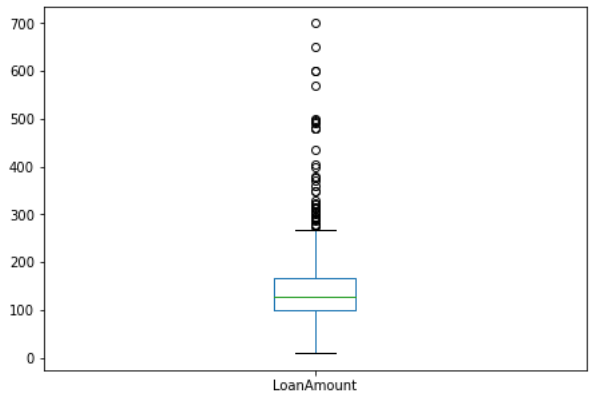


- Missing Value Status: No

Feature 11: Loan Amount

- Data Type: Numerical (Quantitative Data)
- From the Distplot and Boxplot below, it can be concluded that the distribution of the feature "ApplicantIncome" is positively skewed, as well as, there is the presence of outliers.

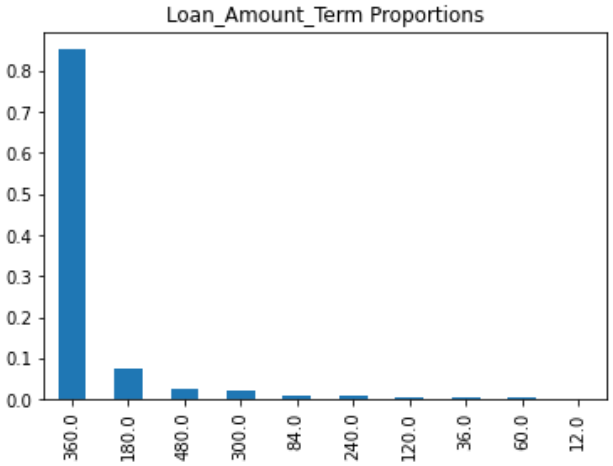




- Missing Value Status: Yes (22)
- Missing Value Imputation: Median (Presence of Outliers) i.e. 128.

Feature 12: Loan_Amount_Term

- Data Type: Discrete (Quantitative Data)
- Around 85% of the loans are 360 months term or 30 years period.



- Missing Value Status: Yes (14)
- Missing Value Imputation: Median (Presence of Extreme Values) i.e. 360.

Feature 13: Loan Status

- Data Type: Nominal (Qualitative Data)

- The 2 unique values are Yes and No. Out of total applications, 68.72% of loans were approved.
- Missing Value Status: No

1.2. BI-VARIATE ANALYSIS

Loan_Status	N	Y
Gender		
Female	33.035714	66.964286
Male	30.674847	69.325153

Loan_Status	N	Y
Married		
No	37.089202	62.910798
Yes	28.391960	71.608040

Loan_Status	N	Y
Education		
Graduate	29.166667	70.833333
Not Graduate	38.805970	61.194030

Loan_Status	N	Y
Self_Employed		
No	31.400000	68.600000
Yes	31.707317	68.292683

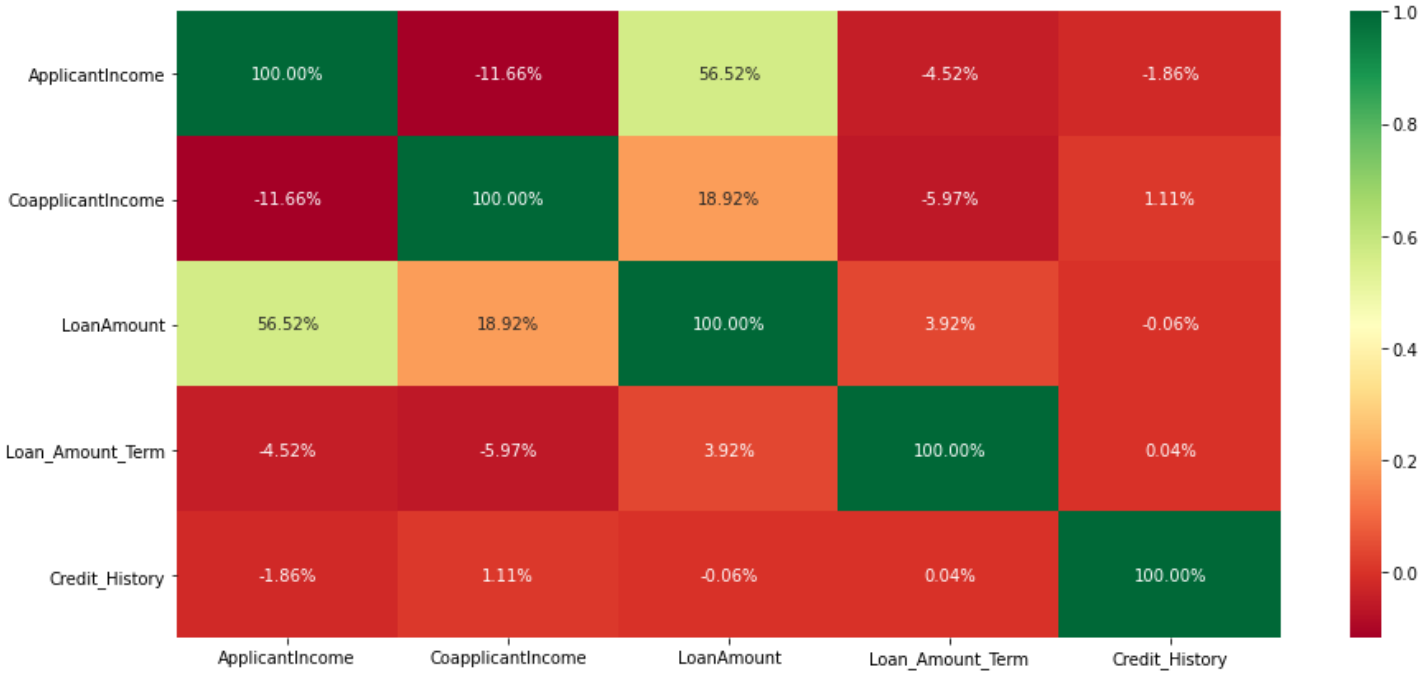
Loan_Status	N	Y
Property_Area		
Rural	38.547486	61.452514
Semiurban	23.175966	76.824034
Urban	34.158416	65.841584

Loan_Status	N	Y
Credit_History		
0.0	92.134831	7.865169
1.0	20.421053	79.578947

- The percentage of male and female applicants is approximately the same for both approved and unapproved loans. Thus, the **gender** of the applicant has **no** substantial difference in the status of a loan.
- The percentage of not-married applicants who got loan approval is higher than that of married applicants. Thus, the **marital status** of the applicant **may** have an effect on the status of a loan.
- The percentage of graduate applicants who got loan approval is higher than that of non-graduates. Thus, the **education** of the applicant **may** have an effect on the status of a loan.
- The percentage of self-employed and non-self-employed applicants is approximately the same for both approved and unapproved loans. Thus, being **self-employed** or not has **no** substantial effect on the status of a loan.
- The percentage of applicants settled in semi-urban areas who got loan approval is higher than that of others. Thus, the **property area** of the applicant **may** have an effect on the status of a loan.
- The percentage of applicants with credit history who got loan approval is higher than that of others. Thus, the **credit history** of the applicant **may** have an effect on the status of a loan.
- The percentage of applicants with high income will have more chances of loan approval. Thus, the Total Income (**Applicant's income + co-applicant income) may** have a substantial effect on the status of a loan.
- The percentage of applicants with approved loans is higher for Low and Average Loan Amounts as compared to that of High Loan Amounts. Thus, **Loan Amount may** have a substantial effect on the status of a loan.

1.3. CORRELATION MATRIX

- The correlation matrix is as follows:



It can be seen that there is an existence of a correlation between (ApplicantIncome and LoanAmount), (Credit_History and Loan_Status) and (CoapplicantIncome and LoanAmount).

PART-II: DATA PRE-PROCESSING

Data preprocessing is a data mining technique that entails converting raw data into a usable format. Real-world data is frequently incomplete, inconsistent, and/or lacking in certain behaviors or trends, and it is rife with errors. Data preprocessing is one method for dealing with such problems.

2.1. MISSING VALUE IMPUTATION

- We need to replace the missing values in the Test set using the mode/median/mean of the Training set, not from the Test set. Likewise, if you remove values above some threshold in the test case, make sure that the threshold is derived from the training and not the test set. Make sure to calculate the mean (or any other metrics) only on the train data to avoid data leakage to your test set.

2.2. DUPLICATE VALUE CHECK

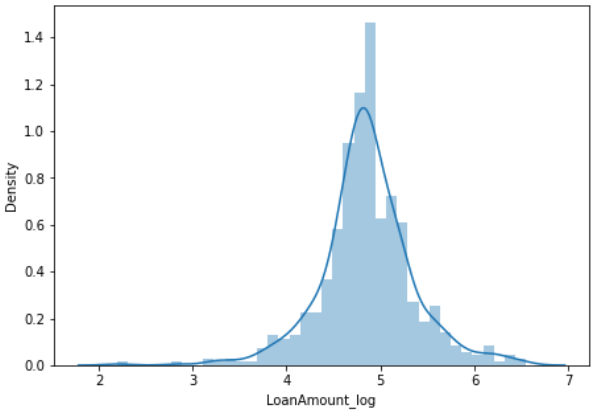
- No duplicate values were found.

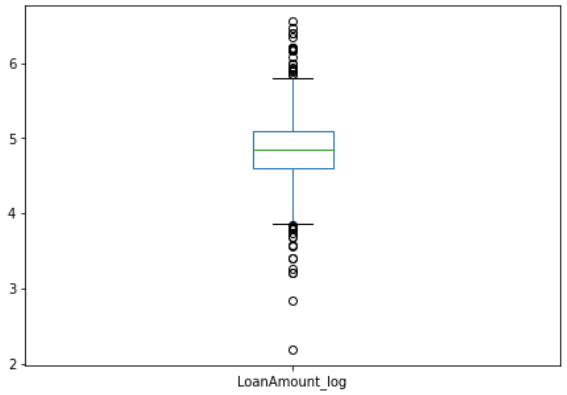
2.3. ENCODING

- Categorical variables were coded into dummy variables.

2.4 OUTLIER TREATMENT

- LoanAmount contains outliers, as we saw in univariate analysis, and we must treat them because the presence of outliers affects the distribution of the data. Outliers in a dataset frequently have a significant effect on the mean and standard deviation, affecting the distribution. We need to take steps to eliminate outliers from our data sets. The **log transformation** is one method for removing skewness. When we use the log transformation, it has little effect on the smaller values but greatly reduces the larger values. As a result, we obtain a distribution that is similar to the normal distribution.





PART-III: MODEL BUILDING

Library Used: scikit-learn (sklearn)

- Sklearn requires the **target variable** in a separate dataset. We drop the target variable from the train dataset to create a new train dataset **"x".** We would create another set with only the target variable **" y".**

- Now, we will train the model on the training dataset **(x and y)** and make predictions for the test dataset.

- For validation of predictions, we divide our train dataset **(x and y)** into 2 parts using the **train_test_split function** from sklearn:

  - Train **(x_train and y_train)**
  - Validation **(x_cv and y_cv)**.

  We can train the model on this train part and use that to make predictions for the validation part. In this way, we can validate our predictions as we have the true predictions (i.e. loan Status) for the validation part (which we do not have for the given test dataset).

- We will further import **LogisticRegression** from sklearn and fit the logistic regression model.

PART-IV: MODEL EVALUATION

We have calculated the confusion matrix, as well as accuracy of the model However, considering what our objective is we would further delve into finding the precision and recall of the model.

Here, our aim is to identify the applications who are genuinely eligible for loan approval. We want the right predictions so as to avoid the scenario of our client "Dream Housing Finance" bearing any loss due to non-repayment of loan.
Thus, let us understand if we are concerned with precision, recall or both.

Precision says, of all applicants who are approved for loan by the model, how many are actually eligible.
Recall says, of all applicants who are eligible for loan approval, how many actually got approved by the model.

Thus, we are interested in high precision as low precision would imply the wrong applicant may be chosen. Whereas, low precision would imply a genuine eligible applicant not getting loan approval. But the cost of granting a loan to a wrong applicant is more in this case.

Finally, the performance of our model seems encouraging, with accuracy of 83%, precision of 82% and recall of 99%.