**Loan Application Status Prediction**

Submitted By

Shubham Jain

**Table of Contents**

**Introduction**

**Project Flow**

**Conclusion**

**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 is able to repay the loan with no difficulties.

**Problem Statement:**

This dataset includes details of applicants who have applied for loan. The dataset includes details like credit history, loan amount, their income, dependents etc.

**Independent Variables:**

- Loan\_ID

- Gender

- Married

- Dependents

- Education

- Self\_Employed

- ApplicantIncome

- CoapplicantIncome

- Loan\_Amount

- Loan\_Amount\_Term

- Credit History

- Property\_Area

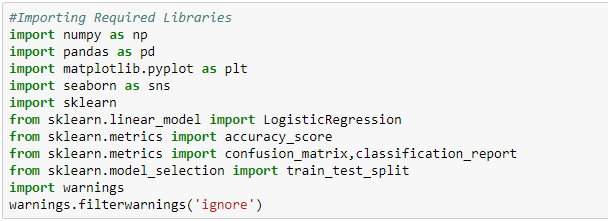
**Dependent Variable (Target Variable):**

- Loan\_Status

To build a model that can predict whether the loan of the applicant will be approved or not on the basis of the details provided in the dataset.

**Project Flow:**

**1. Importing Important Libraries:**



Numpy:

Numpy is core library for Numeric and Scientific Computing

Pandas:

Pandas is core library for Data manipulation and Data Analysis

Matplotlib:

Matplotlib is used for Data Visualisation

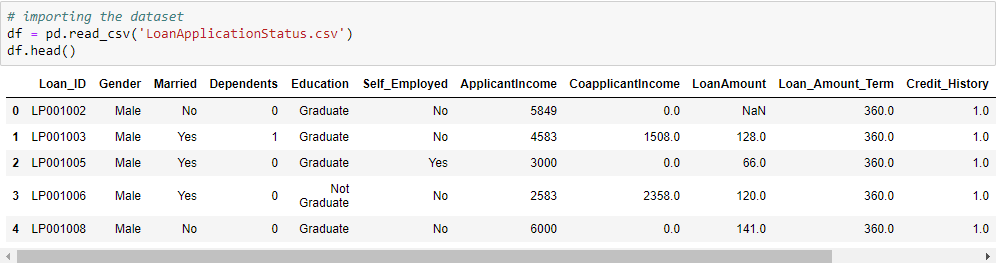
Seaborn:

Seaborn is a Python data visualization library based on matplotlib. It provides a high-level interface for drawing attractive and informative statistical graphics.

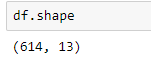
Sklearn:

Scikit-learn (Also known as sklearn) is a [software](https://en.wikipedia.org/wiki/Free_software) [machine learning](https://en.wikipedia.org/wiki/Machine_learning) [library](https://en.wikipedia.org/wiki/Library_(computing)) for the [Python](https://en.wikipedia.org/wiki/Python_(programming_language)) [programming language](https://en.wikipedia.org/wiki/Programming_language). It features various classification, regression and clustering algorithms.

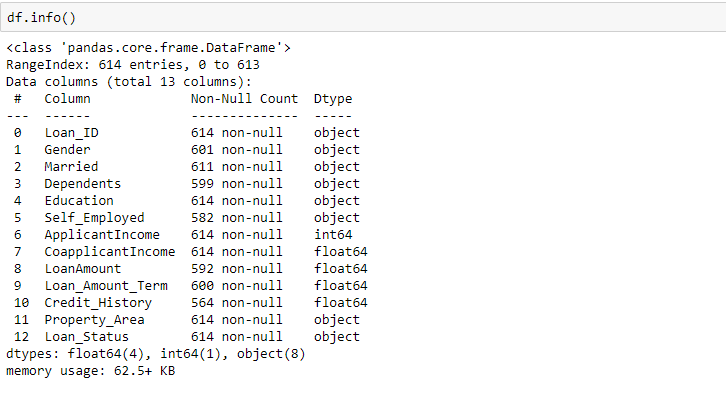
**2. Importing the Dataset** **:**



**3. Performing Basic Operations on data i.e. shape, info**



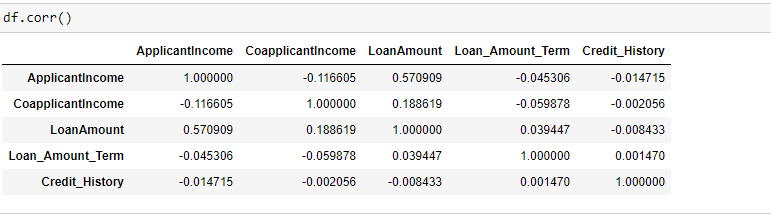
This gives shape of the dataset



We see that Categorical Data Is Loan\_ID, Gender, Married, Dependents, Education, Self\_Employed, Property\_Area, Loan\_Status

We see that Numerical Data is ApplicantIncome, CoapplicantIncome, LoanAmount, Loan\_Amount\_Term, Credit\_History

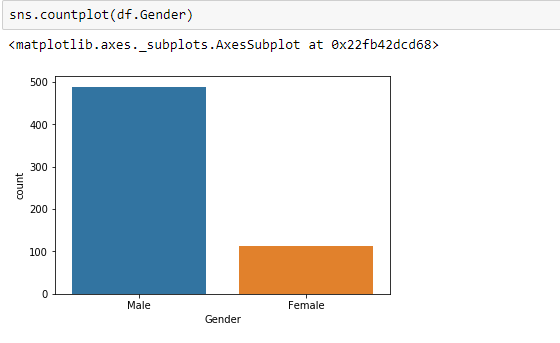
**4. Checking Correlation**



**5. Data Analysis**

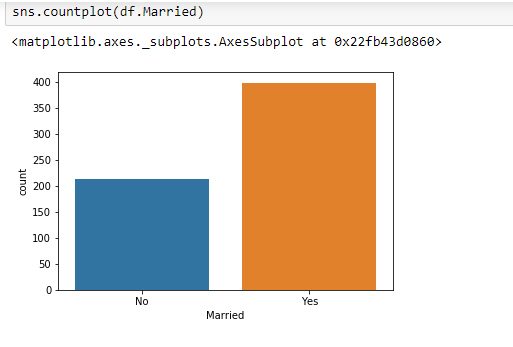
**Univariate Analysis**

**Gender**



Number of male applicant are more than females, which is kinda obvious.

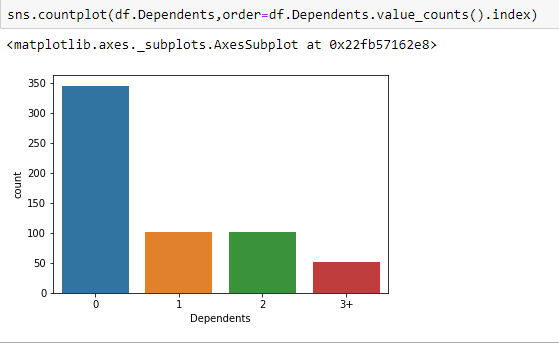
**Married:**



The number of married applicant are greater than Unmarried applicants.

The married guys are in more need of money, than the non-married ones.

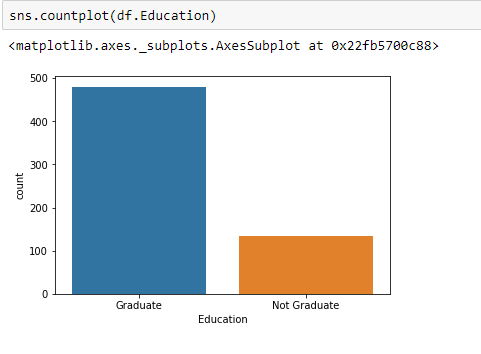
**Dependence**



Dependence is a categorical value

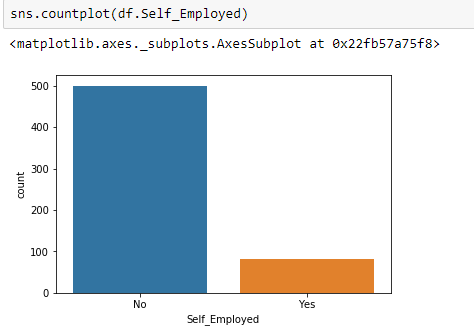
Most of the applicants have no dependencies.

**Education**



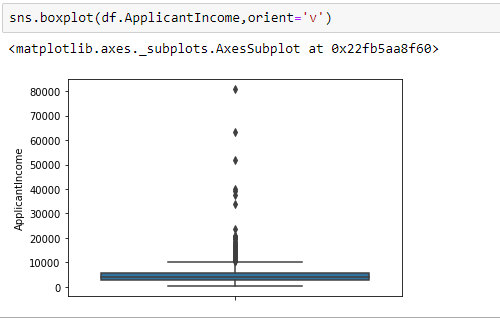
We see that majority of people who apply for loan are Graduates

**Self Employed**

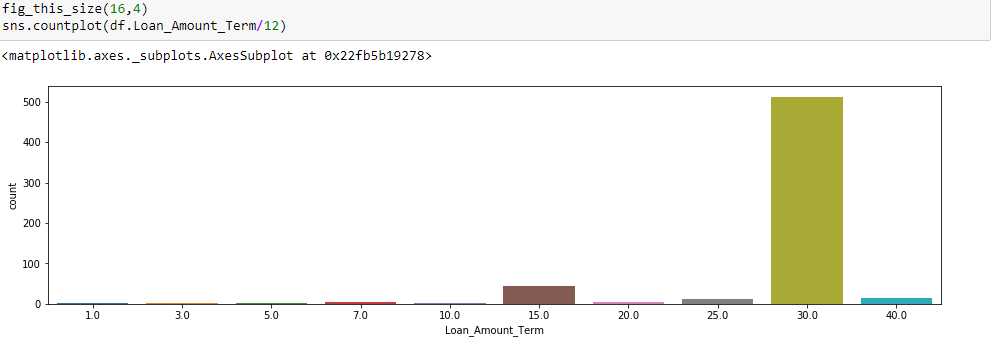


We see that maximum people who apply for loan are not self employed. Very few people are there who are self employed and apply for loan.

**Applicant Income**

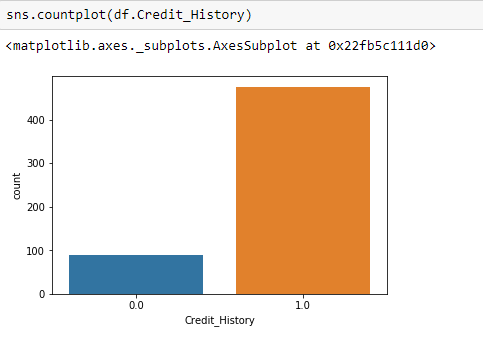
We see that Maximum of the Applicants have income less than 20k.

**Loan Amount Term (In Years)**



Most of the Applicants have applied for loan amount term of 30 Years

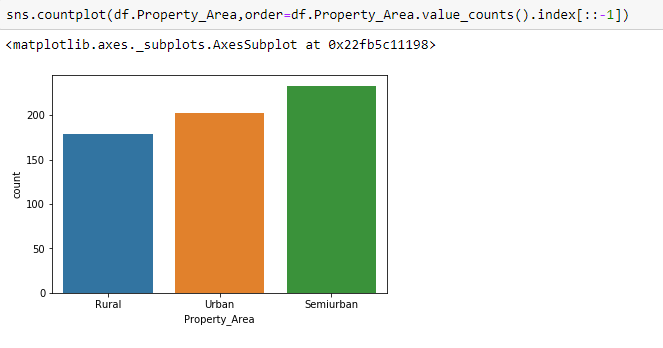
**Credit History**



1 for Positive Credit History which are in majority

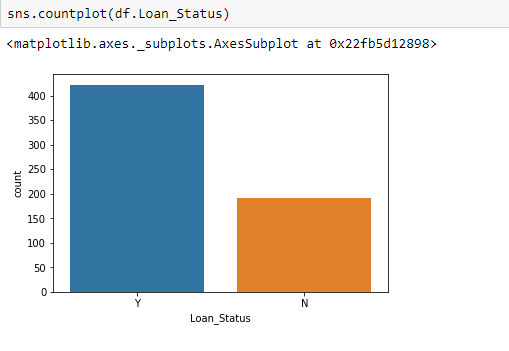
0 for Negative Credit History

**Property Area**



We see that Applicants with Property Area is highest for Semiurban, than comes Urban and then Rural.

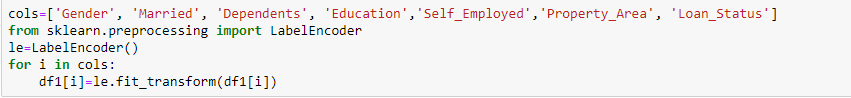
**Loan Status**



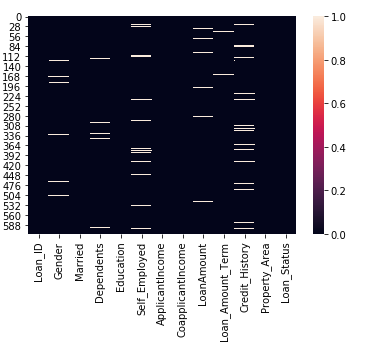
This is Target Variable. And we see that Loan is approved for majority of Applicants.

**6. Encoding The Data**

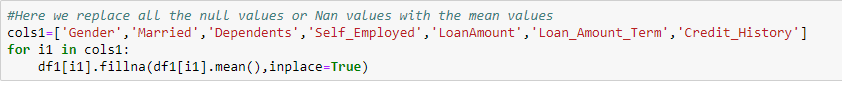
We see that many of the columns are of Object Datatype which needs to be encoded to perform training of the Model



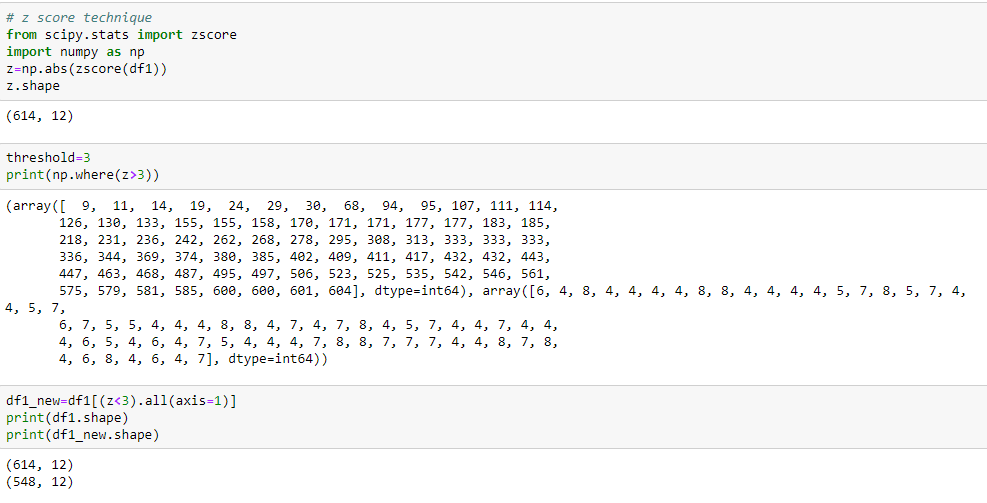
We see that there are missing values in the dataset



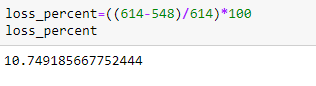
These missing values are replaced with mean values of that particular column.



**7. Checking for Outliers**



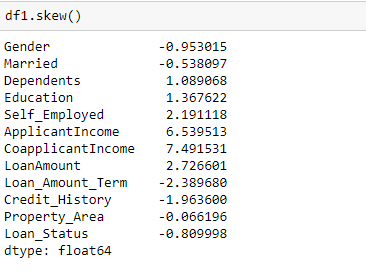
Used Z Score Technique for Outlier Removal. In this technique z score value greater than 3 will be removed from the dataset but we need to check what is total percentage data loss. If Data Loss is more than 7% , then we will not go for Outlier Removal.



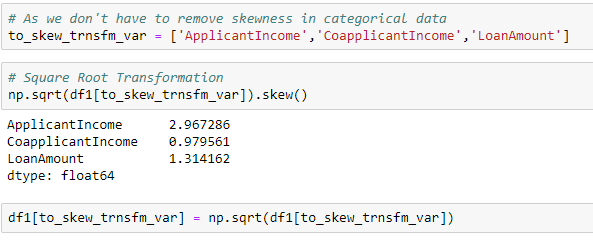
We see that loss of data is above 10%. So we will not go for Outlier removal using Z score Technique.

**8. Checking Skewness of Data**

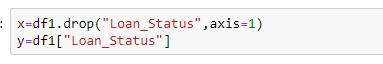
We check skewness in data. Skewness limit is +/- 0.65.



We see that there is high skewness in columns ApplicantIncome, CoapplicantIncome and LoanAmount



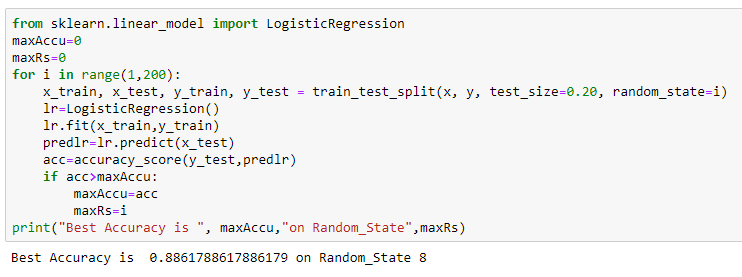
**9. Splitting the Features and Target Variable**



**10. Training the Model**

**Finding Best Random State**

To find best random state, we can use any one algorithm and find the Random\_State for which we get the highest accuracy

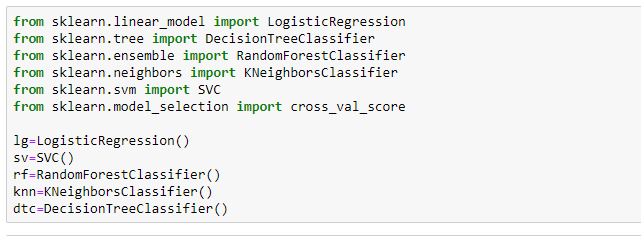


Best Accuracy is Obtained at Random\_State=8, so we will use that Random\_State to train our model.

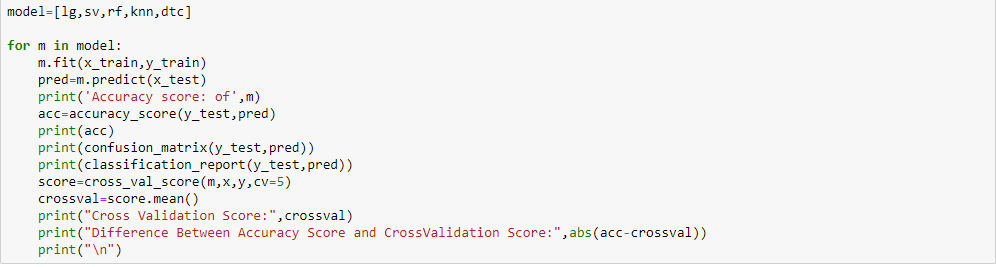
**Train and Test Split**



**Building all the Models**

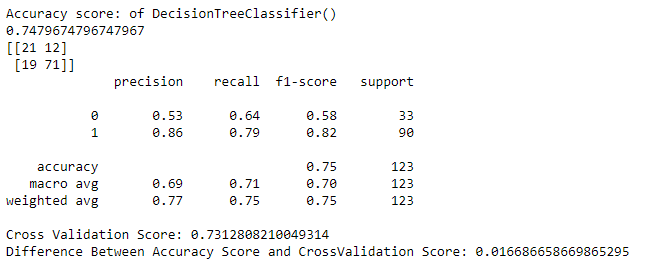


**Training the Model**



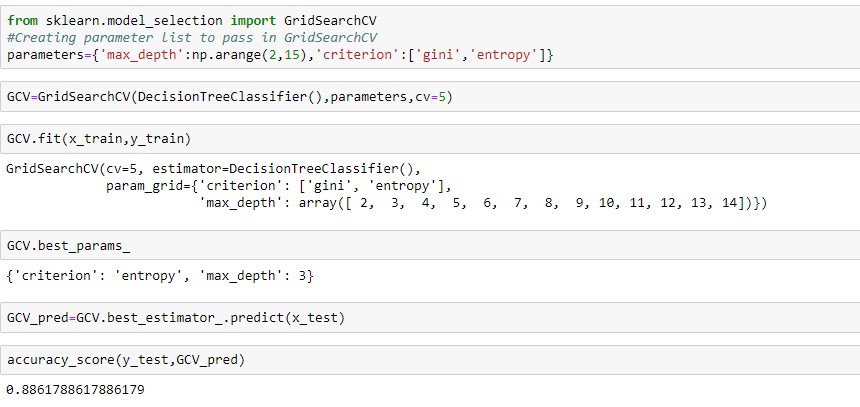
Here we get our Accuracy score, Confusion Matrix, Classification Report and Cross val Score. We will consider the model which is having least difference between Accuracy Score and Cross Validation Score

We see that difference between Accuracy Score and CrossValidation Score is Minimum for Decision Tree Classifier, So DecisionTreeClassifier will be used to train the model.



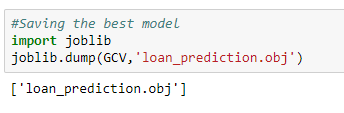
To improve the Accuracy Score of the Model, we have to use HyperParameter Tuning

**Hyper Parameter Tuning**



We see that efficiency of our model is increased to 88.61% from 74.7%

**11. Saving The Model**



**Conclusion**

* Of all the models Logistic Regression, Random Forest Classifier have the best accuracy score but the minimum difference between Accuracy Score and Cross Validation Score was obtained for Decision Tree Classifier Model.
* So we have used Decision Tree Classifier to Train our Model
* The Accuracy then was improved by using Hyper Parameter Tuning and selecting the best parameters for our model.
* Best Parameters obtained was criterion = 'entropy', max\_depth = 3
* Which when we use to train our model, the accuracy increases of our model to 88.6%
* Finally we train our model with the new parameterized Decision Tree Classifier Model and Save the Model.