# Section 1: Project Definition

## Project Overview

This project is based on an amalgamation of banking industry and the real-estate domain. The bank loan section has had a bitter sweet history regarding sanctioning the loans. They have already collected a huge number of demographic information regarding their previous clients. The data also entails the necessary information regarding their marital status and that all the other column details are given below.

## Problem Statement:

Based on data provided by consumers while completing an online application form, the firm aims to automate (in real time) the loan qualification process. The **creation of ML models** **that may assist the business in predicting loan acceptance** is anticipated to hasten the process of deciding whether or not an application is suitable for a loan.

## Metrics

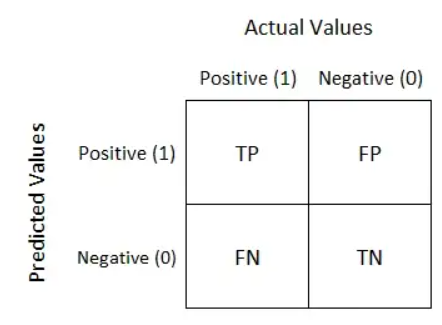
The metrics will help the data scientists to take the decision as to which machine learning works better on the given dataset and helps us make more accurate decisions. Following are the metrics used to evaluate the data models used later.

1. Accuracy.

This will help us determine how far are the predictions from the actual values of the target variable used as test records. Accuracy of a model is an important matrix as it is the baseline for evaluating any of the classification models used. Better the accuracy better is the model.

1. Confusion matrix.

A performance indicator for machine learning classification problems where the output might be two or more classes is the confusion matrix. There are four possible anticipated and actual value combinations in the table.



1. Recall

Recall quantifies the number of positive class predictions made out of all positive examples in the dataset

Recall = TP/(TP+FN)

1. Precision

Precision quantifies the number of positive class predictions that actually belong to the positive class.

Precision = TP / (TP+FP)

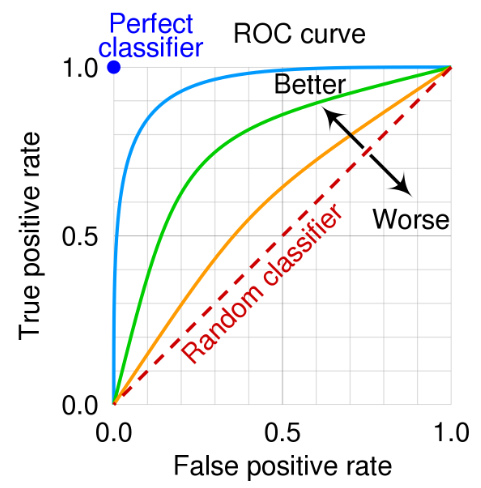
1. F-Score

Comparing two models that have a high recall but a poor accuracy is challenging. So, we utilize F-Score to compare them. F-score aids in measuring both recall and precision simultaneously. By penalizing the extreme values more harshly, it substitutes the harmonic mean for the arithmetic mean.

F-measure = (2\*Precision\*Recall) / (Recall + Precision)

1. ROC Curve

The receiver operating characteristic curve (ROC curve) is a graph that displays how well a classification model performs across all categorization levels. Two parameters are shown on this curve: % True Positive. Rate of False Positives.



# Section 2: Analysis

## Data Exploration and Data Visualization.

### Columns Given

|  |  |  |
| --- | --- | --- |
| **Variable Name** | **Description** | **Sample Data** |
| Loan ID | Loan reference number   (unique ID) | LP001002; LP001003; ... |
| Gender | Applicant gender   (Male or Female) | Male; Female |
| Married | Applicant marital status   (Married or not married) | Married; Not Married |
| Dependents | Number of family members | 0; 1; 2; 3+ |
| Education | Applicant education/qualification   (graduate or not graduate) | Graduate; Under Graduate |
| Self Employed | Applicant employment status   (yes for self-employed, no for employed/others) | Yes; No |
| Applicant Income | Applicant's monthly salary/income | 5849; 4583; ... |
| Co-applicant Income | Additional applicant's monthly salary/income | 1508; 2358; ... |
| Loan Amount | Loan amount | 128; 66; ... |
| Loan Amount Term | The loan's repayment period (in days) | 360; 120; ... |
| Credit History | Records of previous credit history   (0: bad credit history, 1: good credit history) | 0; 1 |
| Property Area | The location of property   (Rural/Semiurban/Urban) | Rural; Semiurban; Urban |
| Loan Status | Status of loan   (1: accepted, 0: not accepted) | 1; 0 |

The company has also done their own Exploratory data analytics but are not able to associate the right mix of the features that will help them to take a data driven decisions regarding loan approval. Especially based on these features

### Dataset Variable Distribution:

There are **13 variables** in total. Their statistical distribution is given below.

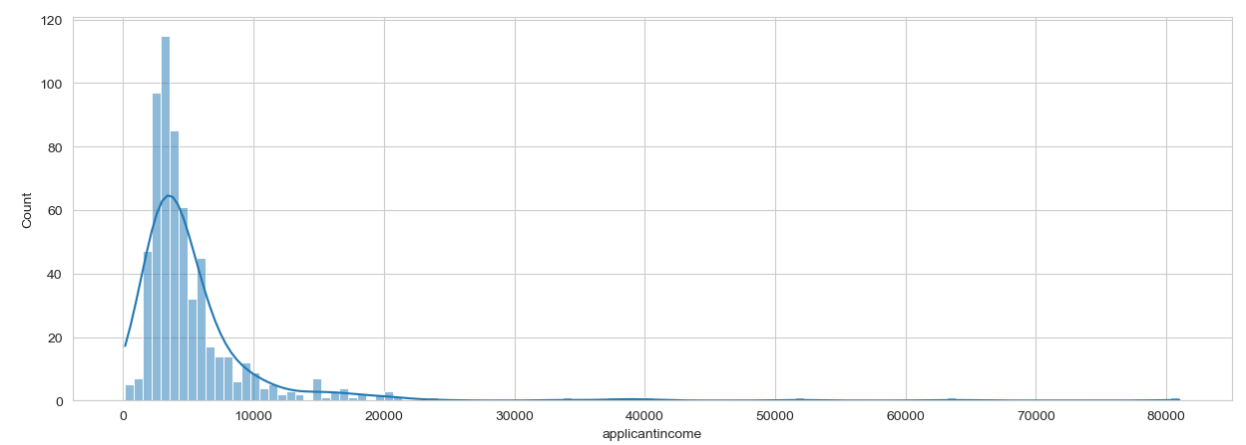
* 4 Numerical variables.
* 8 Categorical variables.
* 1 target variable.

### Numerical Variables Distribution

There are 4 Numerical columns totals and their statistical distribution is given below.

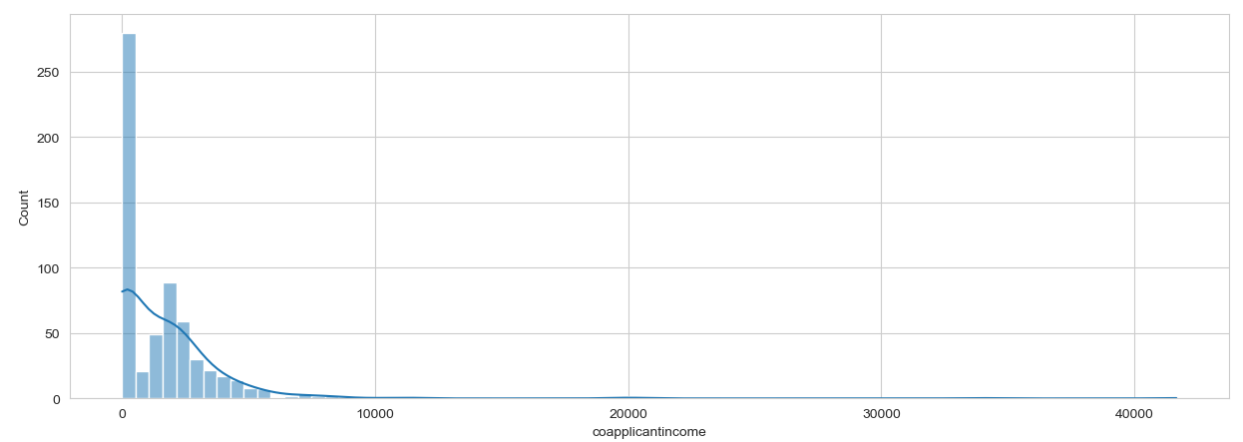
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **Applicant Income** | **Co-applicant Income** | **Loan Amount** | **Loan Amount Term** | **Credit History** |
| **count** | 614 | 614 | 592 | 600 | 564 |
| **mean** | 5403.459 | 1621.246 | 146.4122 | 342 | 0.842199 |
| **std** | 6109.042 | 2926.248 | 85.58733 | 65.12041 | 0.364878 |
| **min** | 150 | 0 | 9 | 12 | 0 |
| **25%** | 2877.5 | 0 | 100 | 360 | 1 |
| **50%** | 3812.5 | 1188.5 | 128 | 360 | 1 |
| **75%** | 5795 | 2297.25 | 168 | 360 | 1 |
| **max** | 81000 | 41667 | 700 | 480 | 1 |

1. **Applicant income Histogram Distribution**



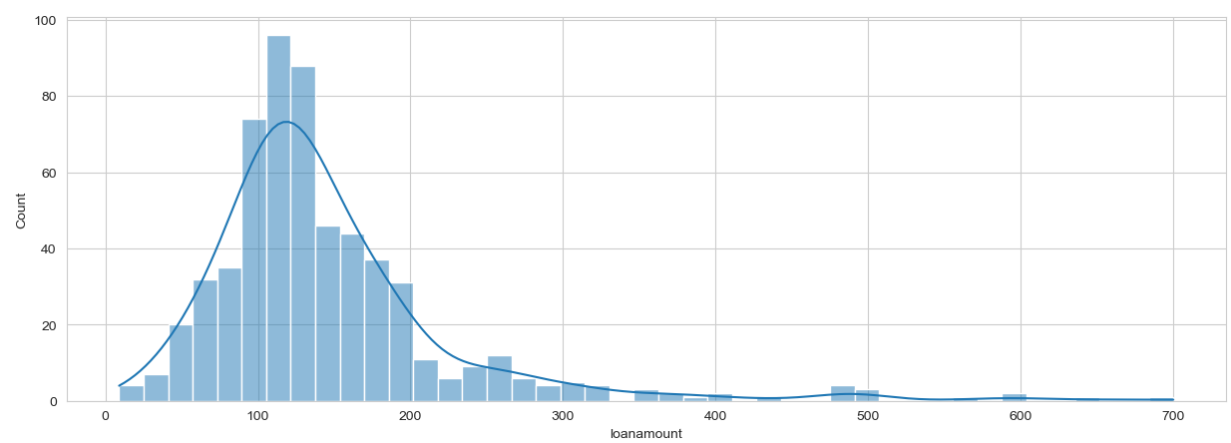
The histogram here shows that the applicant’s income is left skewed which means it distribution is positively skewed.

1. **Co- Applicant Income**



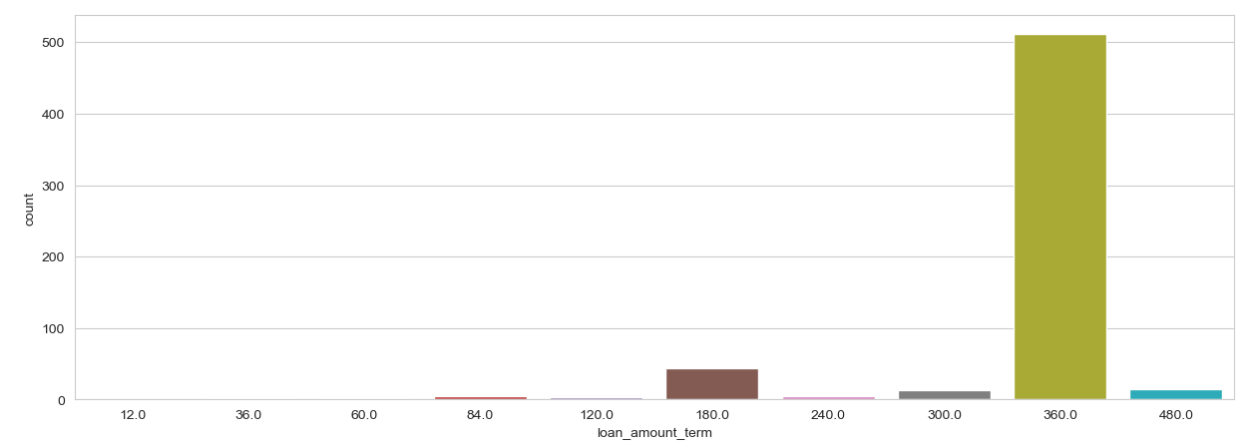
The histogram here shows that the co - applicant’s income is left skewed which means it distribution is positively skewed.

1. **Loan Amount**



The histogram here shows that the Loan Amount is left skewed which means it distribution is positively skewed. However, this plot is more normal distributed with respect to other plots.

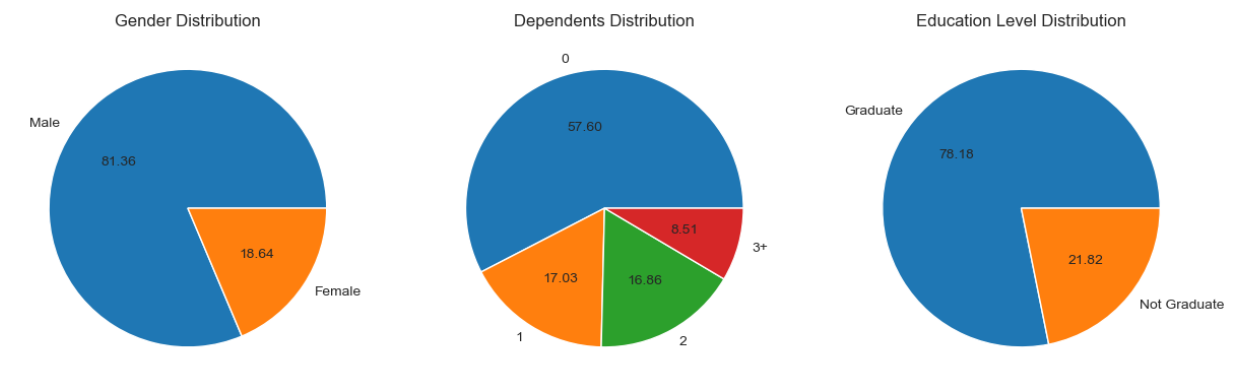
1. **Loan Term Count Plot**



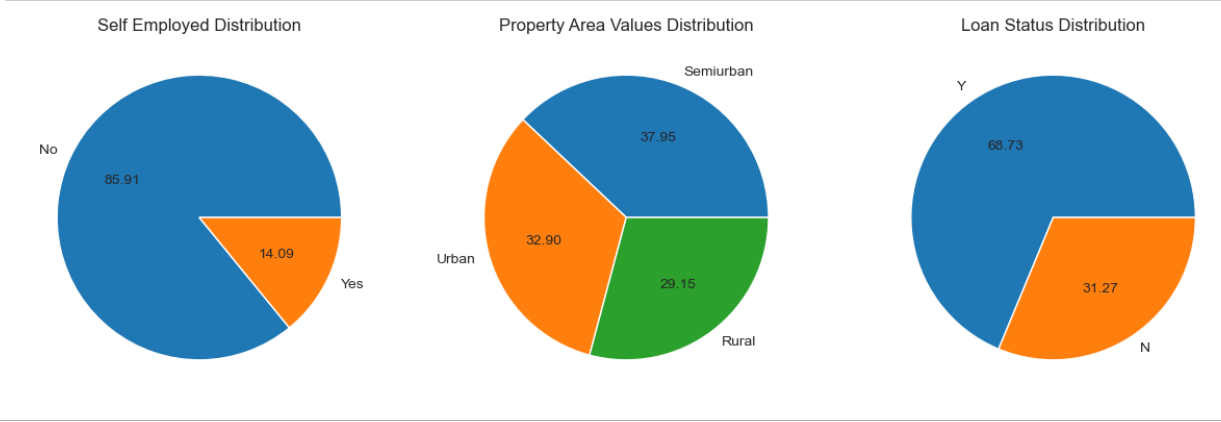
The count plot shows that there are more applicants who have applied for a 3 years loan so it is estimated that this one value will be influential in decision making.

### Categorical Distribution

1. Showing the Gender Distribution, Dependent Distribution, Education Distribution in a pie chart.



1. Showing the Self-Employed Distribution, Property Area Value Distribution, Loan Status Distribution in a pie chart.



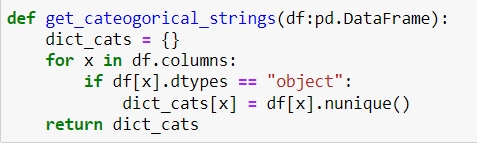
# Step 3 Methodology

## Data Pre-Processing and Implementation.

1. **Dealing With Categorical string values**

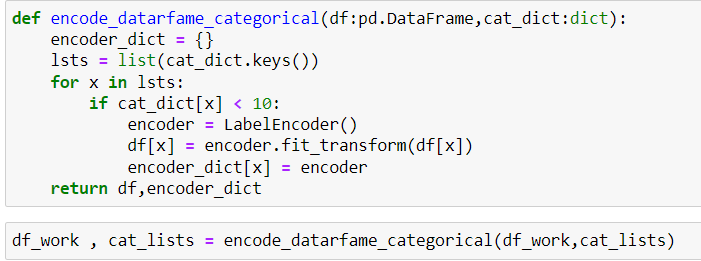
In order to implement any of the machine learning models, or computational algorithms, so we first need to convert the string categorical values to numerical values.

This will be done by using label encoders. So, we first had to target the categorical values in the dataset. The code below shows how to target the categorical values using python.



The targeted variables are as follows 'gender', 'married', 'dependents', 'education', 'self\_employed', 'property\_area', 'loan\_status'.

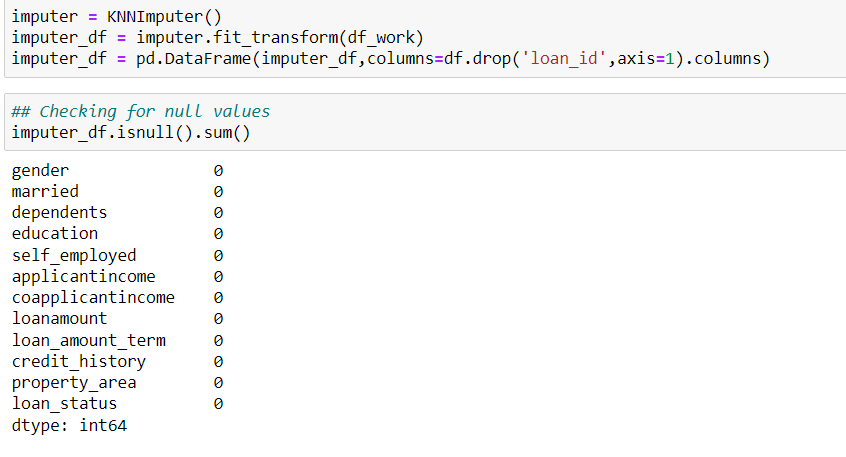
Then followed by encoding the variables accordingly



1. **Dealing With Null Values.**

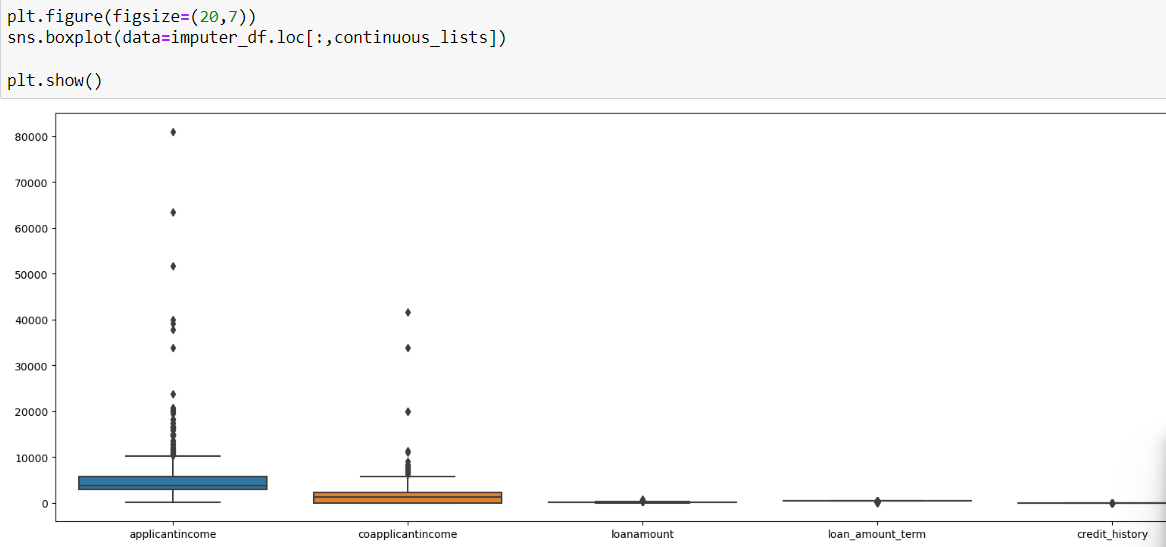
While going through the dataset, we found out that there are null values in a number of columns that are filled with null values. So, we need to deal with these values as soon as possible after the data is now converted into numerical type. This will help the machine learning model in order to not lose the data.

We will be using the KNN-Imputer in order to fill the columns with the relevant values which resemble their similar neighbours. This will help us to not go vaguely filling null values with mean, medians or mode values.



1. **Dealing with Outliers**

An outlier is a data point that deviates significantly from other findings. An observation in our data that is inaccurate or odd when compared to other observations can also be referred to as an outlier. Measurement uncertainty or experimental error are two possible causes of outliers. Outliers in the data can sabotage and mislead the machine learning model training process, leading to less accurate models and ultimately poor performance. Let's look at how we can identify outliers in our data now that we are aware.

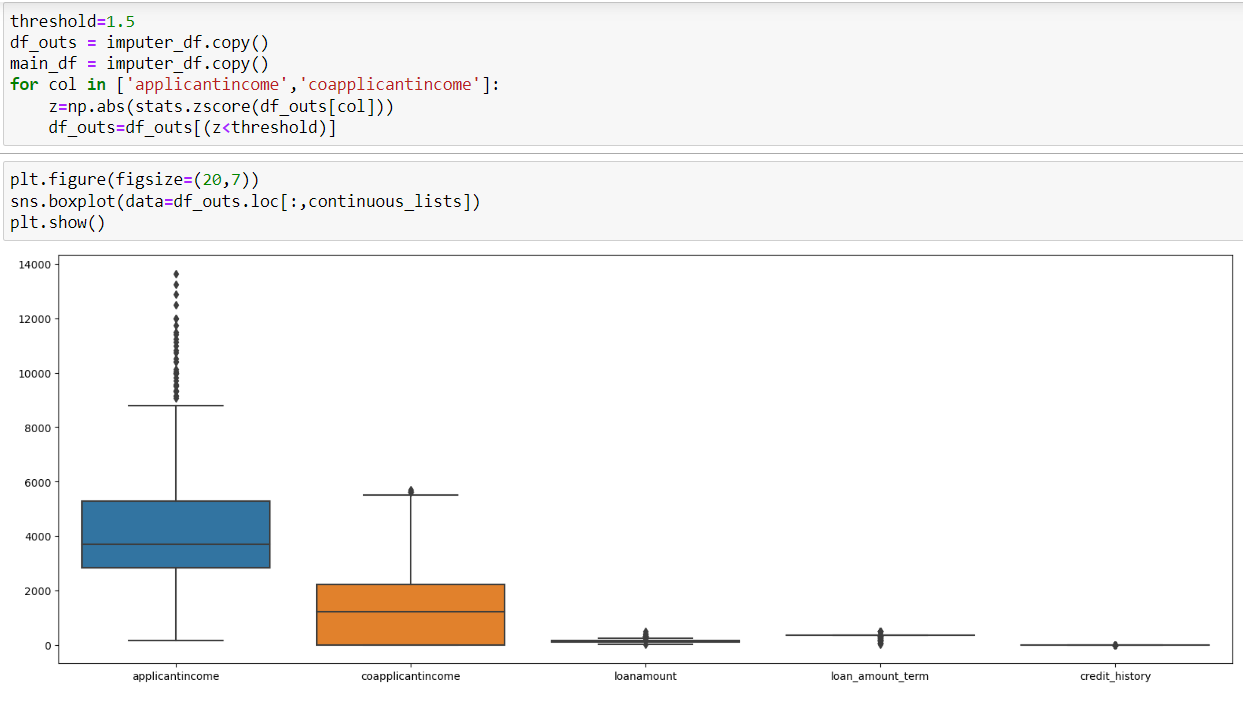


There are 2 ways of dealing with outliers.

* Z-Score
* IQR

**Z-Score**

Here all the values outside the threshold are given converted to their threshold values. Here the threshold is set to 1.5.



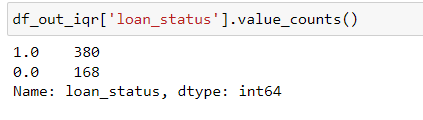
**Using IQR**

Using the IQR you calculate the Lower Quartile Range (LQR) and Upper Quartile Range (UQR) and the values below LQR are converted to LQR, and values above UQR are converted to UQR.



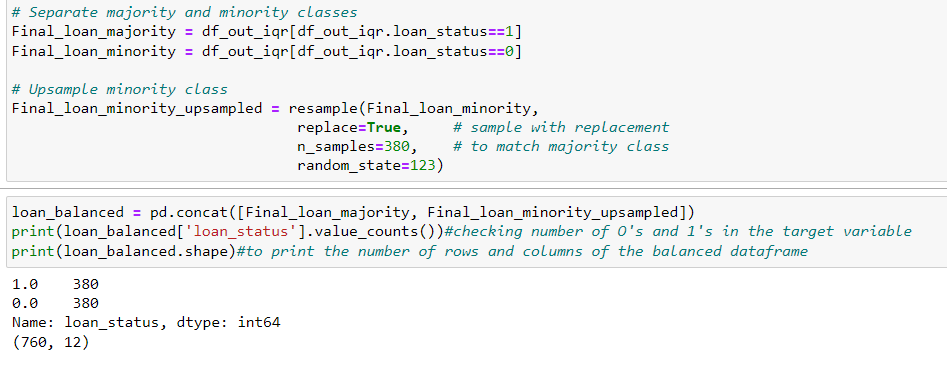
We will be choosing the IQR method.

1. **Dealing with Unbalanced Data**

****

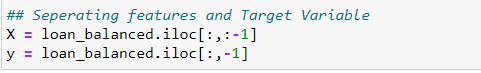
As shown in above figure, we can see that the data of the target variable ‘loan status’ is highly unbalanced. So, we need to balance the data so that the models are not biased towards the ‘1’s in the target variable.

We will be Oversampling the ‘0’ labelled rows and balance the difference between both.

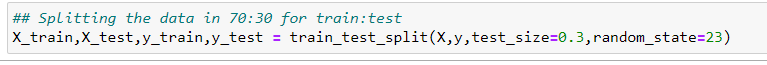


1. **Splitting the dataset into train and test sets**

Before going for splitting the dataset into train and test set, the dataset must be first separated to features and target variable.



*Splitting the dataset into 70:30 Train: Test Split*

**

NOTE: We have also Normalized the data so that the entire dataset is on the same scale (Using MinMaxScalar).

# Step 4: Results

## Models Used in this project

1. Random Forest

Parameter Configuration Used: n\_estimators =100

1. Stratified K-Fold Logistic Regression Model

Parameter Configuration Used: n\_splits=5, random\_state=23, shuffle=True

1. Naïve Bayes Classification
2. XGBoost Classifier

Parameter Configuration Used: base\_score=0.5, booster='gbtree', callbacks=None, colsample\_bylevel=1, colsample\_bynode=1, colsample\_bytree=1, early\_stopping\_rounds=None, enable\_categorical=False, eval\_metric=None, feature\_types=None, gamma=0, gpu\_id=-1, grow\_policy='depthwise', importance\_type=None,

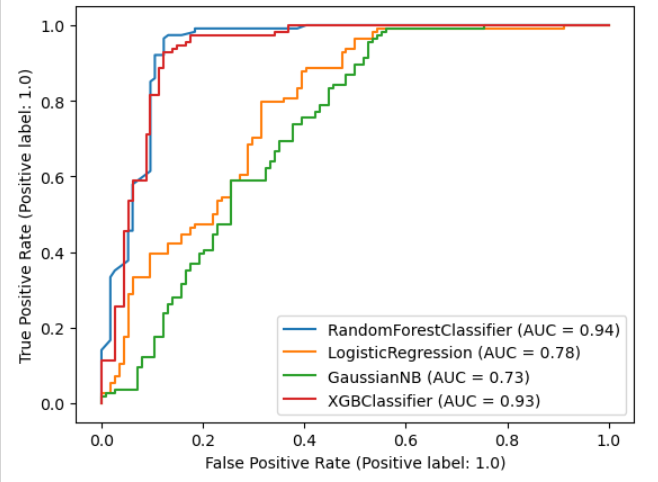
interaction\_constraints='', learning\_rate=0.300000012, max\_bin=256, max\_cat\_threshold=64, max\_cat\_to\_onehot=4, max\_delta\_step=0, max\_depth=6, max\_leaves=0, min\_child\_weight=1,

missing=nan, monotone\_constraints='()', n\_estimators=100,

## Evaluation and result Comparison

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Particulars | Random Forest Model | Stratified K-Fold | Naïve Bayes | XGBoost |
| Accuracy | 89.74% | 72.36% | 71.04% | 86.84% |
| Recall | 89.00% | 72.00% | 71.00% | 87.00% |
| Precision | 89.00% | 77.00% | 78.00% | 87.00% |
| F-Score | 89.00% | 72.00% | 71.00% | 87.00% |

#### **ROC Curve**



# Step 5: Conclusion:

As per the models trained on the dataset, we have boiled down the decision-making model, to Random Forest Classifier. It has shown the highest level of training accuracy according to the ROC curve which is 94%. Also the Accuracy score of this model surpasses all the other models that are used in the project.

## Reflections:

Throughout the entire project it was evident that the data stored in the company systems is very circumstantial and that this data has to go through a lot of pre-processing in order for them to be used for any data analysis or more specifically data science applications. It could also be seen that the tree models worked more better for this specific problem statement that other probabilistic, or boosted methods.

### Improvements:

It is advisable to take the support and add the random forest model in the decision-making mix of the loan assessment and sanctioning the loan however there must also be more relevant details available for doing the same. The company should store the data in encoded format in one of the servers and that it can still collect the data as raw as possible. There must be data validation while entering the data as it will help avoid null and irrelevant values.