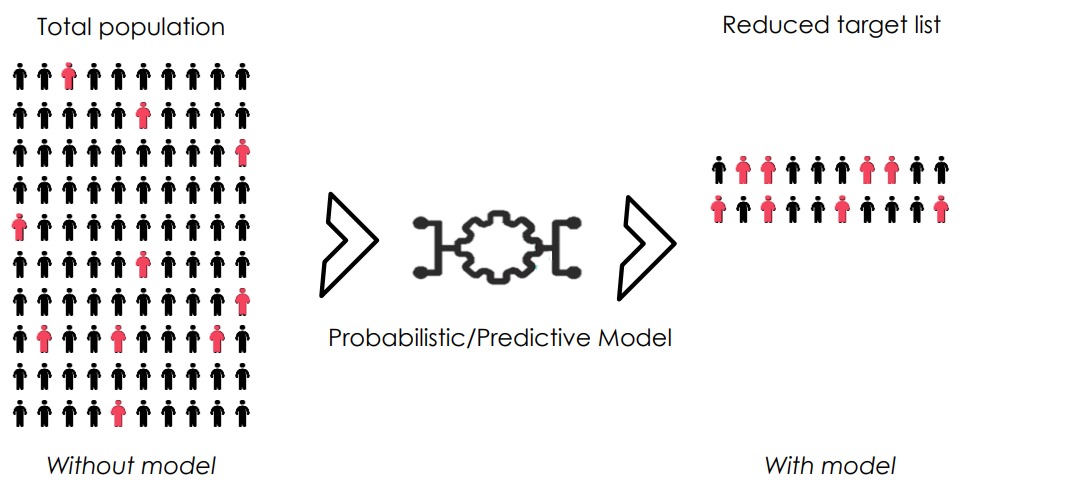
**BLOG ON**

**LOAN OFFER OPTIMIZATION**

**Challenge**

Development of a predictive model with good accuracy to predict which customers are likely to accept personal loan offers.



**Data Description**

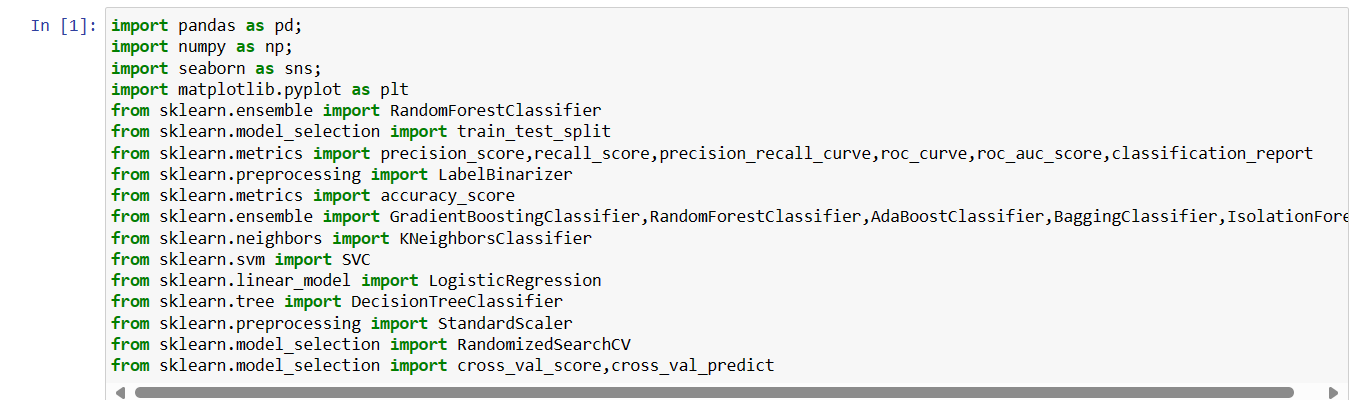
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Customer ID** |  |  |  |  |  |
| **Pin-code** | **Home Area pin-code** |  |  |  |  |  |
| **Age** | **Customer's age** |  |  |  |  |  |
| **Fam member** | **Total Family Members** |  |  |  |  |  |
| **Education** | **Customer's Education** |  |  |  |  |  |
| **T-Experience** | **years of professional experience** |  |  |  |  |  |
| **Income** | **Annual income of the customer** |  |  |  |  |  |
| **Mortgage** | **Value of house mortgage if any** |  |  |  |  |  |
| **Fixed Deposit** | **Does the customer have a certificate of deposit (CD) account with the bank?** |  |  |  |  |  |
| **De-mat** | **Does the customer have a De-mat account with the bank?** |  |  |  |  |  |
| **Net Banking** | **Does the customer use internet banking facilities?** |  |  |  |  |  |
| **Loan** | **Did this customer accept the personal loan offered in the last campaign?** |  |  |  |  |  |

**Question/Problem Description**

The competition is simple: Use the Bank Customer data (age, Education Income, etc.) to try to predict who will agree to take loan. This is a **binary classification.**

**Binary classification** is the task of [classifying](https://en.wikipedia.org/wiki/Statistical_classification) the elements of a [set](https://en.wikipedia.org/wiki/Set_(mathematics)) into two groups on the basis of a [classification rule](https://en.wikipedia.org/wiki/Classification_rule)

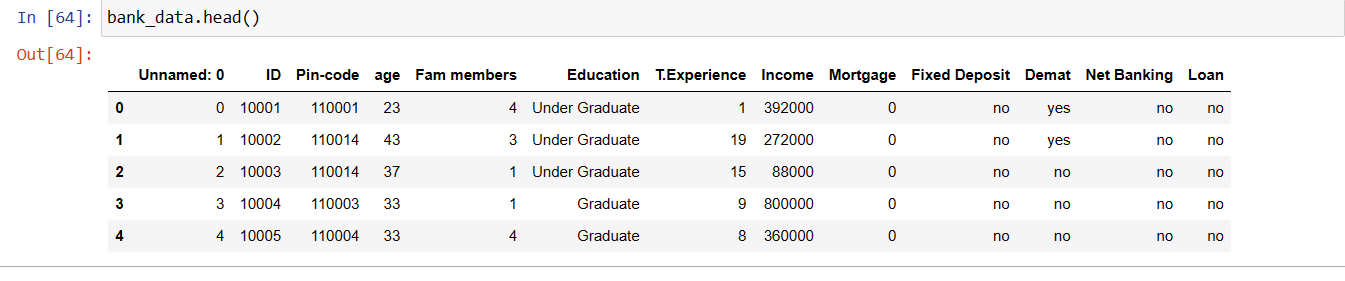
**Step 1:** Importing Dependencies



**Step 2:** Loading data:



**Printing first 5 rows of dataset:**

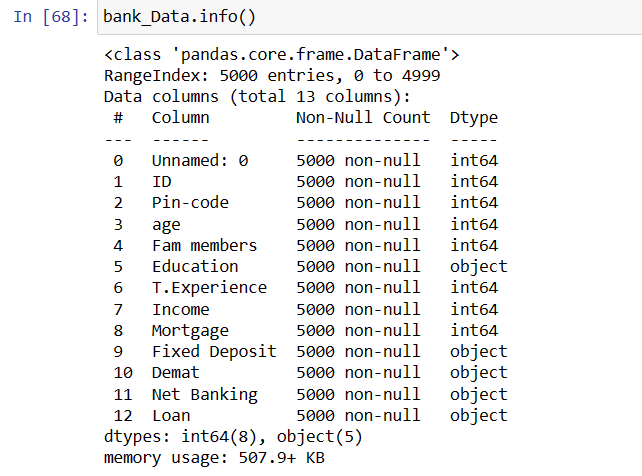


As we can see that the data is not properly labelled we need to change Yes to “1” and No to “0”

We will handle this in our **DATA FEATURE ENGINEERING** part

**Step 3:** Reading Data

The main goal of data understanding is to gain general insights about the data, which covers the number of rows and columns, values in the data, datatypes, and Missing values in the dataset.



**data.info()** shows:

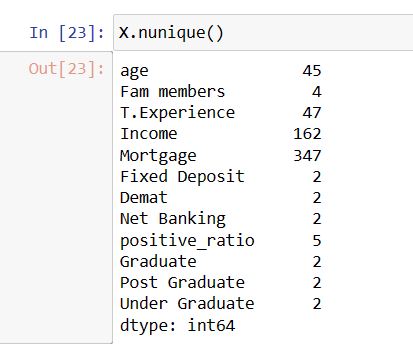
The variables ID, Pin code, Age, Fam members, Education, T. Experience, Income, Mortgage, Fixed Deposit, DE mat, Loan

Data types: float64, int64, and object

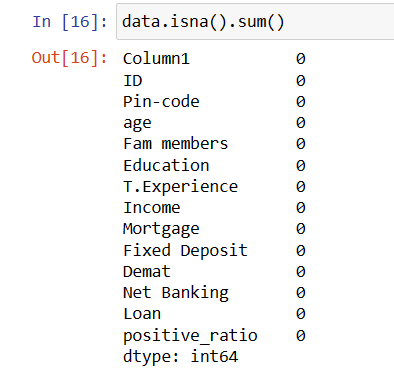
Shape: 13 columns and 5000 rows

**Check for duplicates:**

**nunique():** Based on several unique values in each column and the data description, we can identify the continuous and categorical columns in the data. Duplicated data can be handled (encoded) or removed based on further analysis



**Missing Values Calculation:**

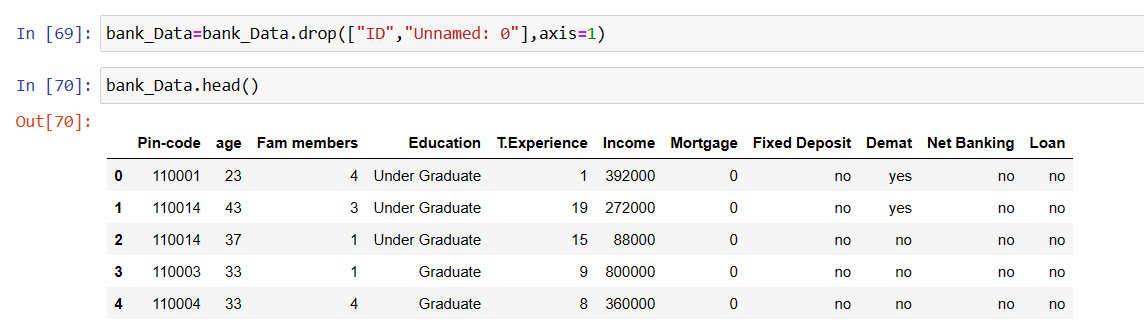


We can see that our data does no contains any null values

### **Step 4:** Data Reduction

Some columns or variables can be dropped if they do not add value to our analysis.

In our dataset, the column [Column1, ID], can be dropped assuming they don’t have any predictive power to predict the dependent variable.

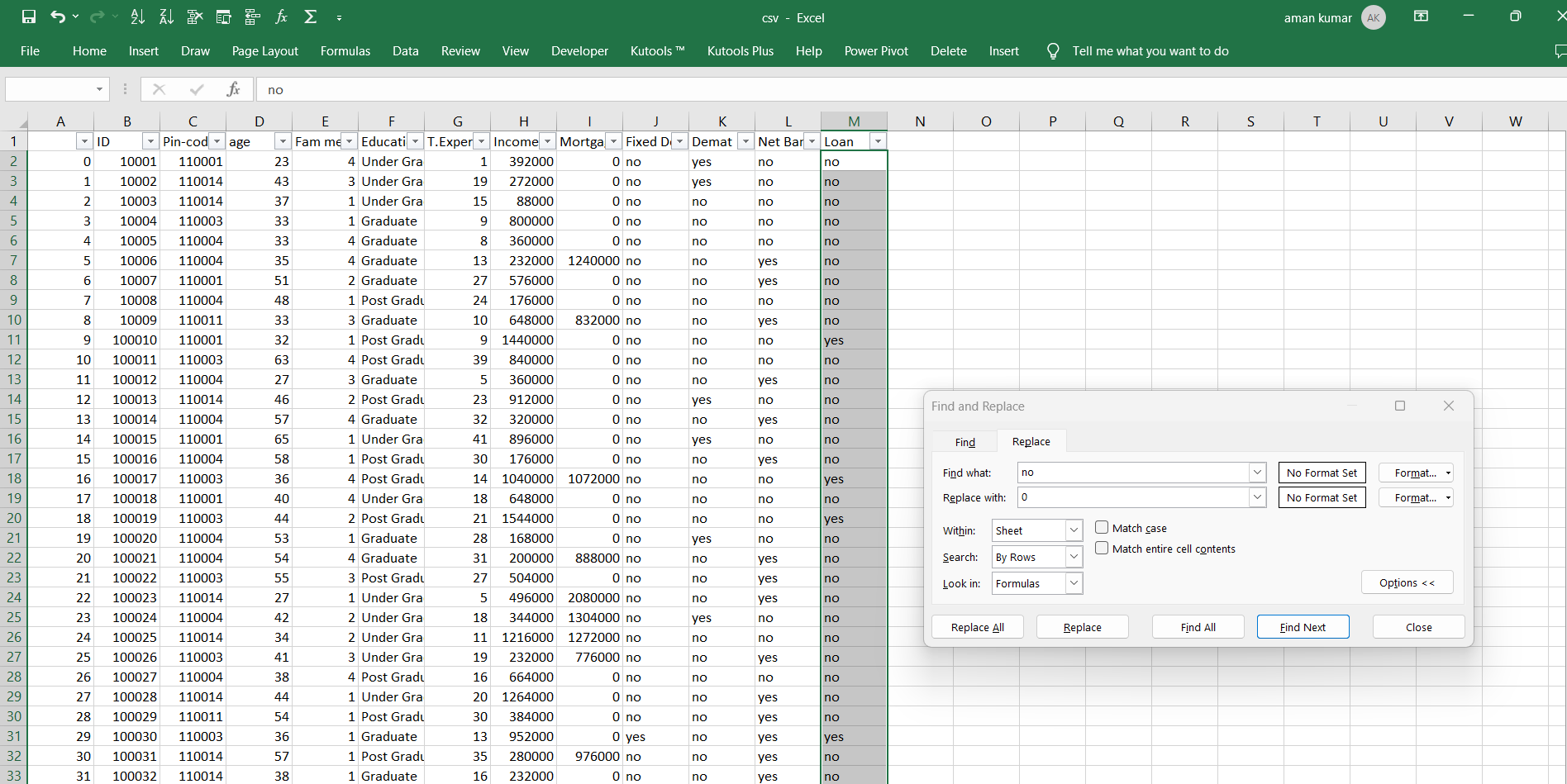


### **Step 5:** Feature Engineering

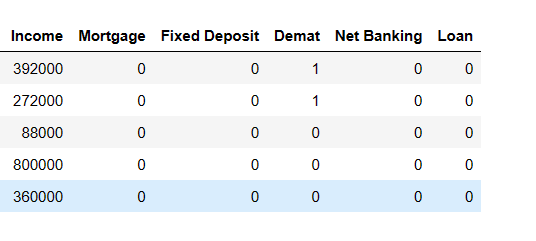
Feature engineering refers to the process of using domain knowledge to select and transform the most relevant variables from raw data when creating a predictive model using machine learning or statistical modelling. The main goal of Feature engineering is to create meaningful data from raw data.

As we saw above that few columns contains “yes” and “no” we can change them in 0 and 1

This could be done easily by excel “find and replace” method

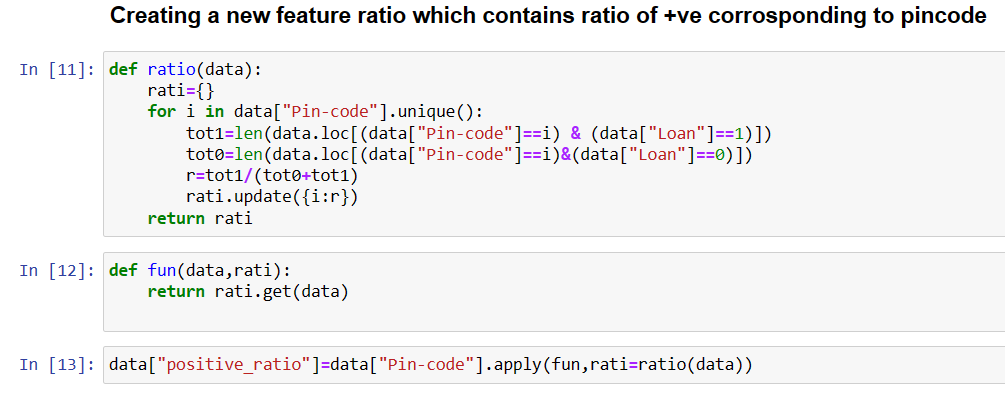


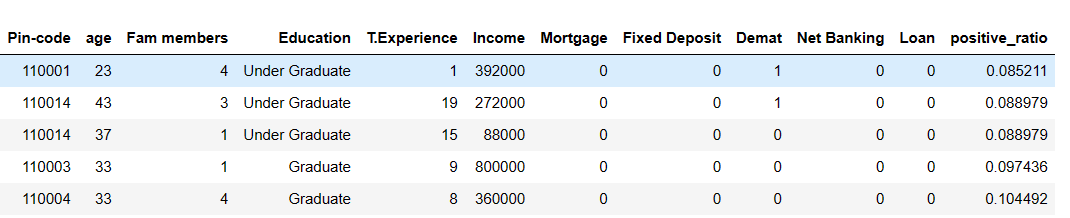
**Result**



### **Step 6:** Creating Features

We see that Pin-code alone does not make any sense in the data. Using Pin-code we could create a new column containing positive response ratio corresponding to that Pin-code

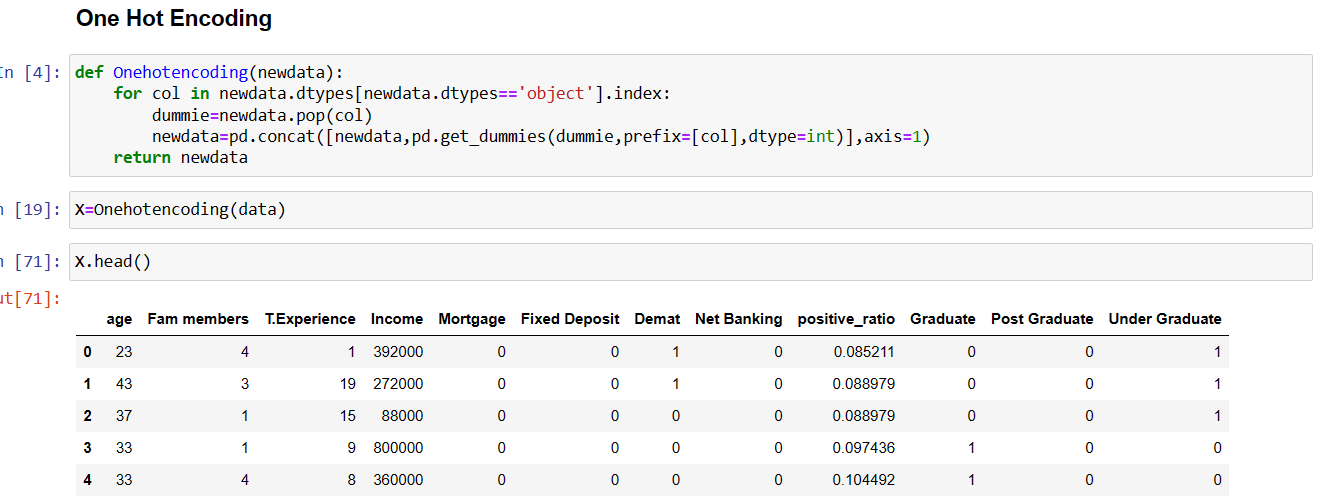




Now we will drop pin-code column.

We also see that Education column could not be feeded to our model as it is an object datatype object so we could do One-Hot Encoding

One-hot encoding is a technique used in machine learning to represent categorical data as numerical features suitable for machine learning algorithm.



Now we can see that three new columns had been created Graduate, Post Graduate, Under Graduate

### **Step 7:**EDA Exploratory Data Analysis

Exploratory Data Analysis refers to the crucial process of performing initial investigations on data to discover patterns to check assumptions with the help of summary statistics and graphical representations.

* EDA can be leveraged to check for outliers, patterns, and trends in the given data.
* EDA helps to find meaningful patterns in data.
* EDA provides in-depth insights into the data sets to solve our business problems.
* EDA gives a clue to impute missing values in the dataset

**Checking Correlation between data:**

We will be using heat map to represent the correlation.

A heat-map is a data visualization technique that uses a colour gradient to represent the magnitude of values within a two-dimensional dashboard. It’s a powerful tool for visually exploring and identifying patterns or trends in data.

Enhance clarity and enable easy exploration of data point and its correlation.

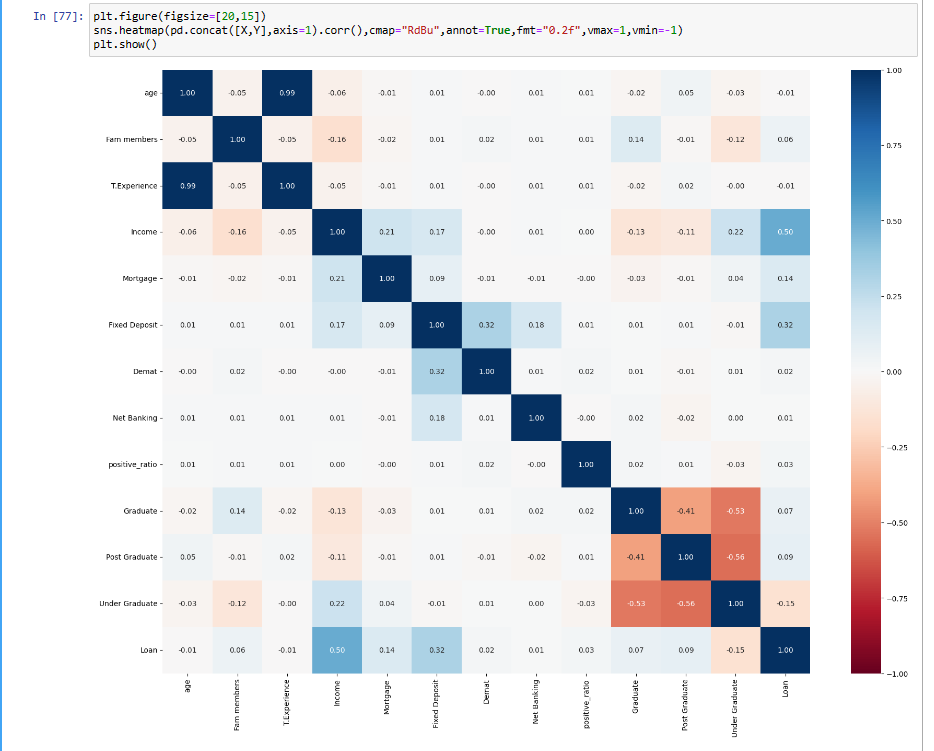


Figure 1

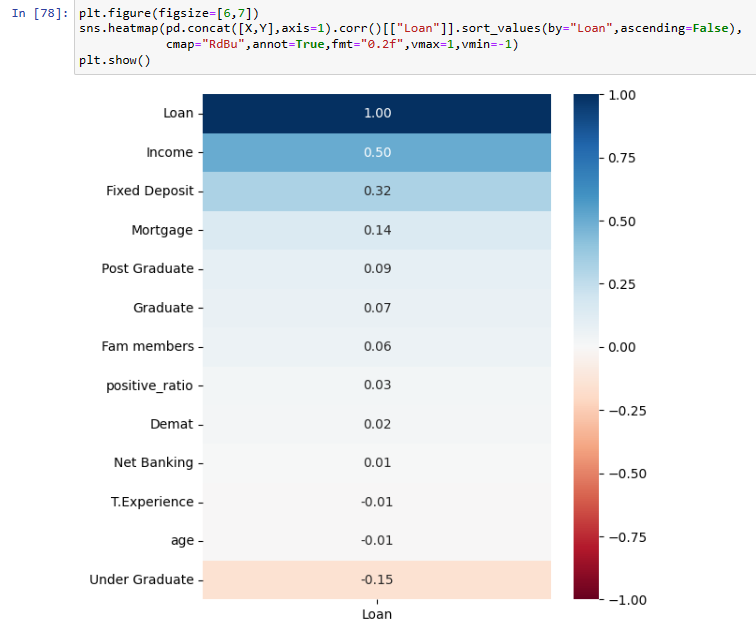


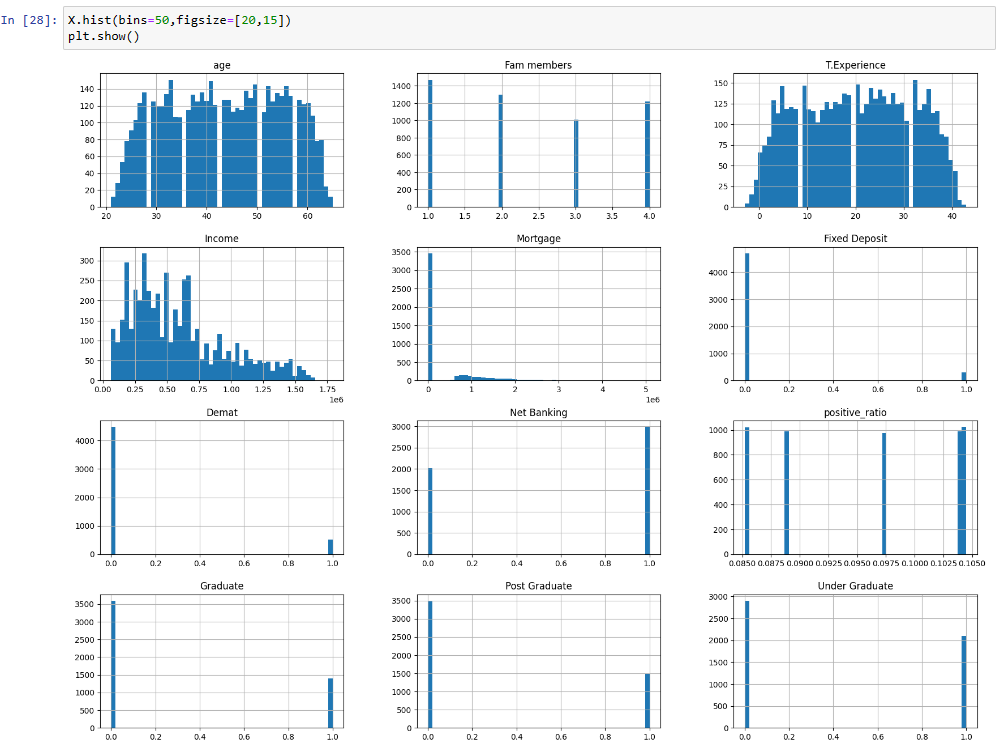
Figure 2

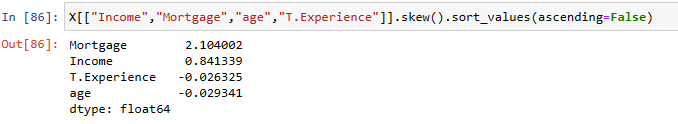
**Figure 1->** shows correlation between each and every features.

**Figure 2->** it shows numerical value of correlation for with respect to Loan i.e. how much each feature is correlated with our label “Loan” in descending order

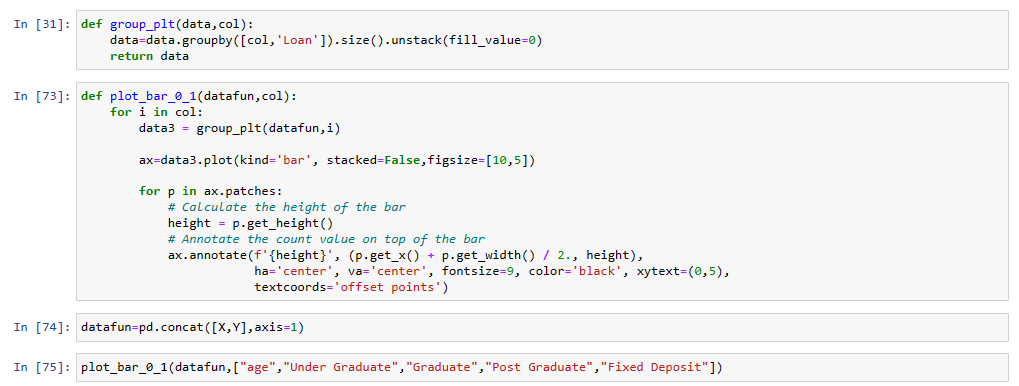
Observation: using heat map we can conclude that income (0.50) is the most important feature for our prediction followed by Fixed Deposit (0.32), Mortagage (0.14) and Under graduation (-0.15)

**Checking data distribution:**

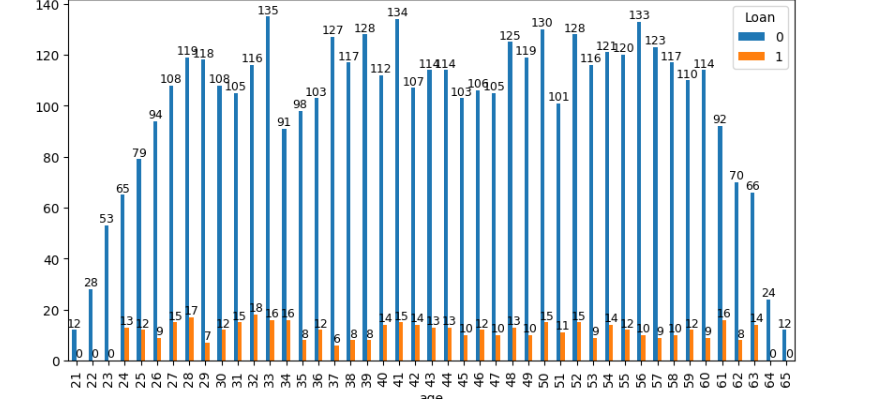




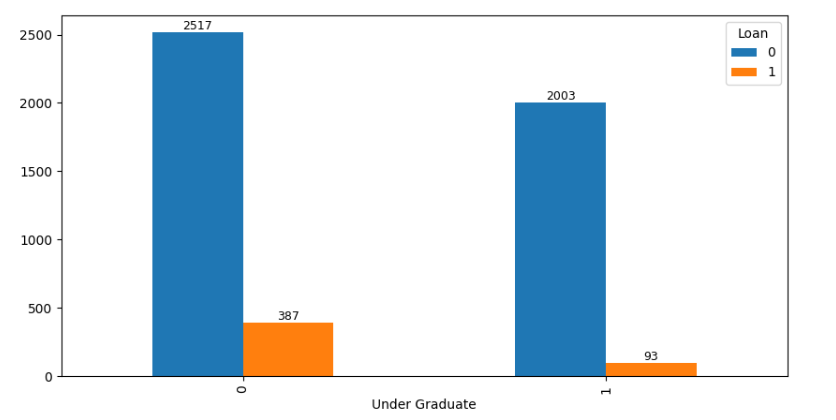
**Plotting count of positive response (1) and negative response (0) with respect to different features**



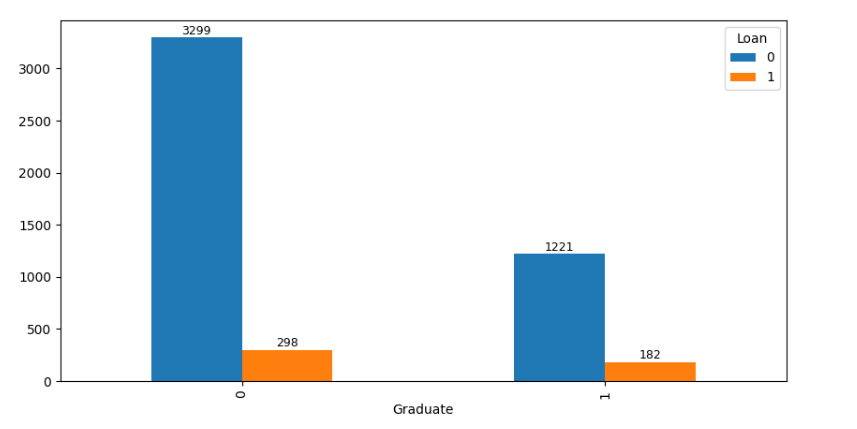
**With respect to age:**



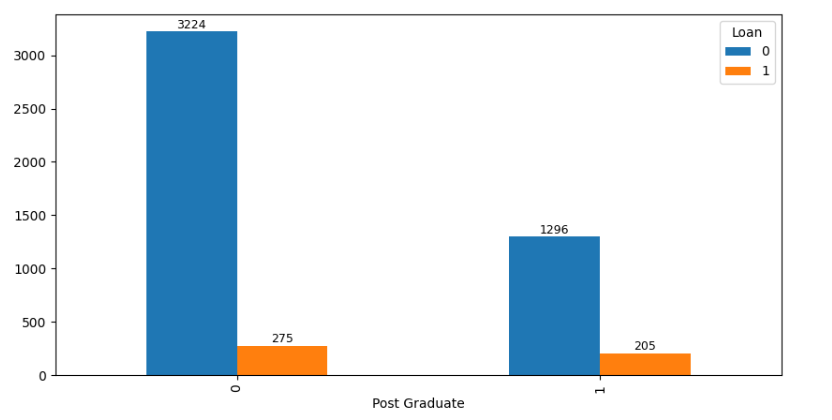
**With respect to under graduate**



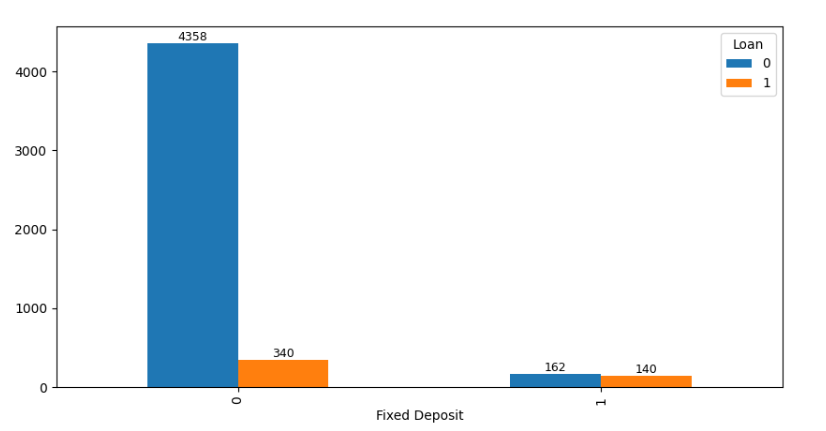
**With respect to graduate:**



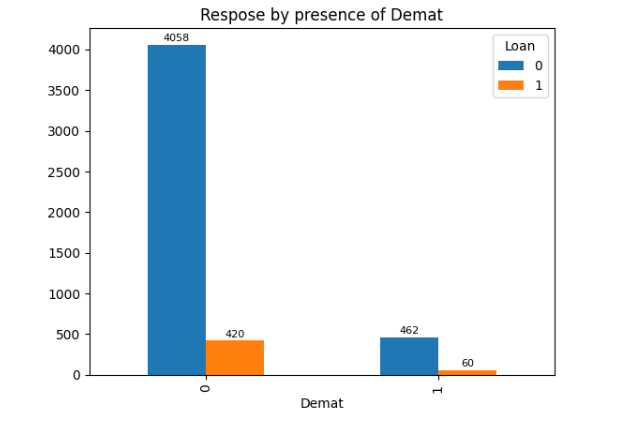
**With respect to post graduate:**



**With respect to fixed deposit:**



**We can see that people having fixed deposit are more likely to take loan**



**Step 8:** Scaling data

Data scaling is a crucial pre-processing step in machine learning, particularly for algorithms sensitive to feature scales. Here's why it matters:

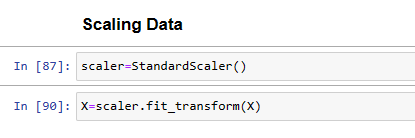
• Fair Play for Features: Ensures all features contribute equally to the model.

• Faster Learning: Helps algorithms converge faster by making distances between data points more meaningful.

• Improved Performance: Normalizes data distribution for better model performance and avoids local minima traps.

• In essence, scaling creates a level playing field for features, leading to better machine learning models.

In essence, scaling creates a level playing field for features, leading to better machine learning models.



**Step 9:** Splitting data into test and train set

Train-Test Split Importance:

• **Prevents Overfitting:** Avoids memorizing training data, ensuring the model generalizes well to new data.

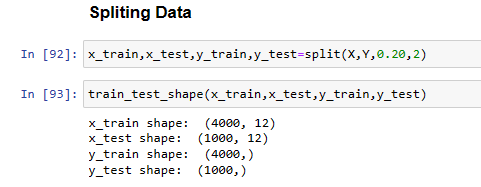
• **Realistic Performance:** Provides an unbiased assessment of the model's real-world performance.

• **Model Selection:** Helps compare different models on the same test set for best generalization.

• **Hyper-parameter Tuning:** Enables tuning on training data and evaluating on test data to prevent overfitting.

• **Generalizability:** Improves the model's ability to perform well on unseen data.

**We will be using train\_test\_split() from sklearn library to split our data set.**



**Step 10:** Training model

**Identifying best model:**

Before training any model we need to choose best model which could perform well for our classification task.

For this we will be make a list of classifiers and the using loop we will be training every model in the list. Once the model is trained we will append its scores in a dictionary.

We will be using **RandomizeSearchCV** to carry out hyper-parameter tuning.

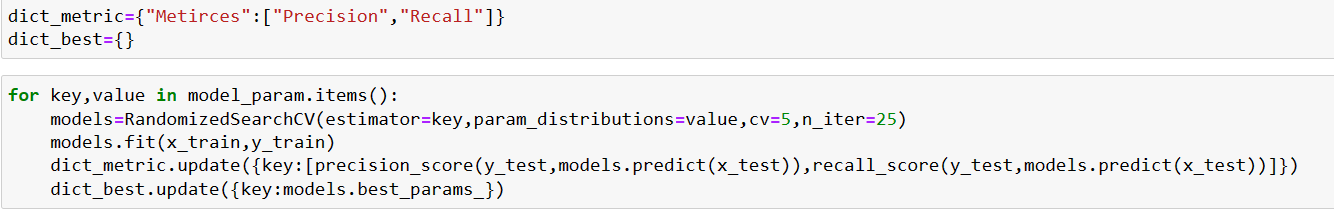
Each model will be trained with different hyper-parameters

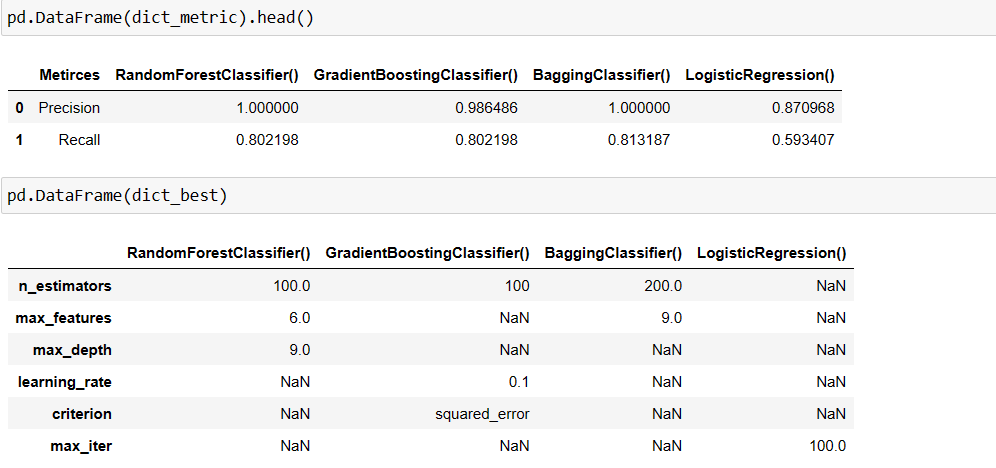
After every model is trained and its score is added to dictionary we will pick the model with best score



In the above code “model\_param” dictionary contains model as key with its trainable parameters as value

**Training models**





We can see score of every model and best parameter for every model in above two data frame

**Reason:** BaggingClassifier() has the best recall value so we will be choosing BaggingClassifier() to predict whether a person will take loan or not.



**Why we should give more preference to Recall score?**

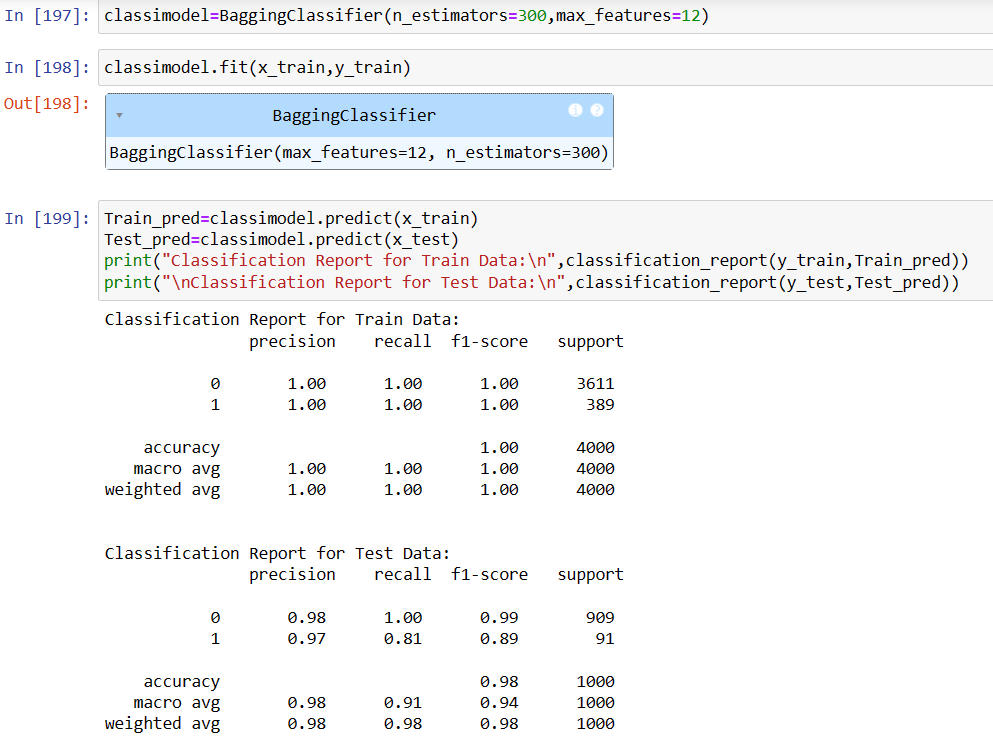
We should prioritize the recall score because we want to identify as many potential loan applicants as possible. A high recall score indicates that the model captures a significant portion of those who are likely to accept a loan, reducing the chances of overlooking qualified candidates. If our recall score is low, the model risks missing a significant number of potential loan applicants, leading to lost opportunities and less effective targeting in marketing campaigns. This could ultimately impact the bank's revenue and customer satisfaction. Improving recall helps ensure a broader and more inclusive pool of loan candidates, which aligns with our goal to maximize loan acceptance rates.

**Why should not rely much on Accuracy score?**

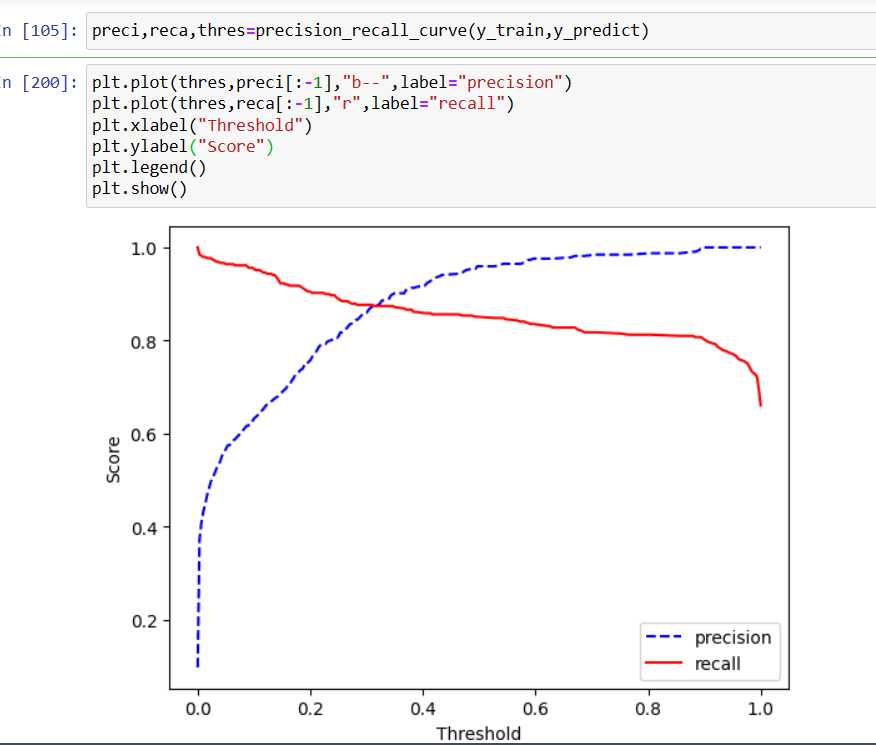
Relying solely on accuracy metrics in highly skewed datasets can be misleading. Accuracy doesn't distinguish between classes and may mask poor performance on minority classes. It's crucial to use metrics like precision, recall, or F1-score, which provide insights into class-specific performance. Ignoring this can lead to biased models and incorrect assessments of real-world impact, especially when misclassification costs vary across classes.

**Example :-** Consider a scenario where only a few cases are positive among many negatives. If a model predicts all cases as negative, it might achieve high accuracy due to the abundance of negative instances. However, this doesn't signify a reliable model; it simply exploits the skewed distribution. In reality, such a model fails to identify the crucial positive cases. Thus, while accuracy appears high, the model's actual performance is inadequate for practical use.

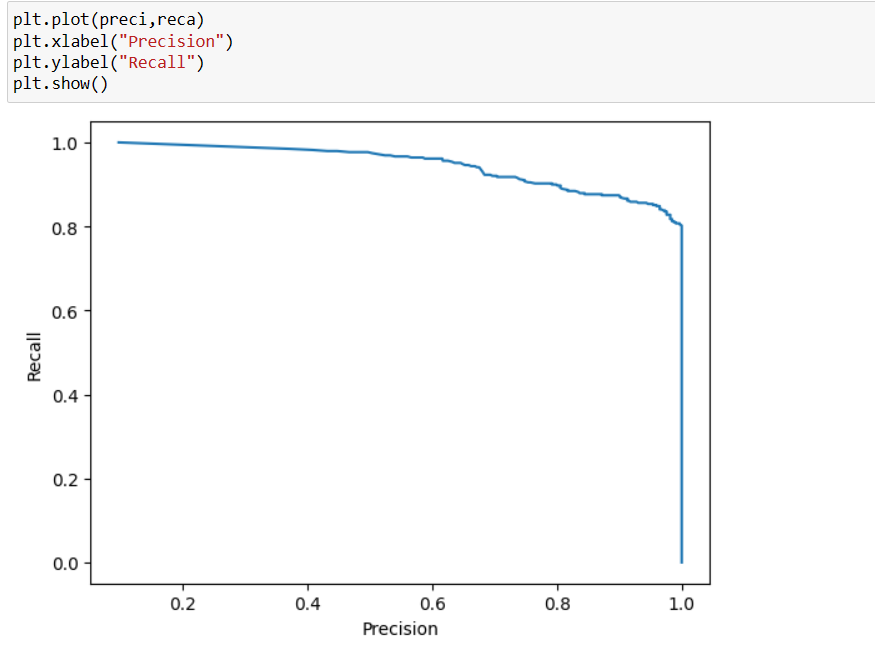
**Prediction**



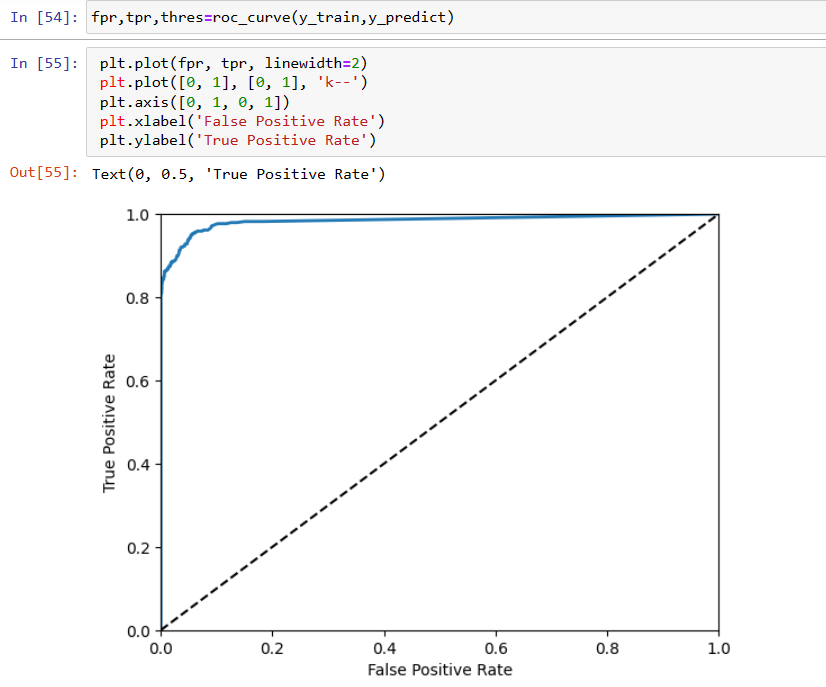
**Plotting PR curve**



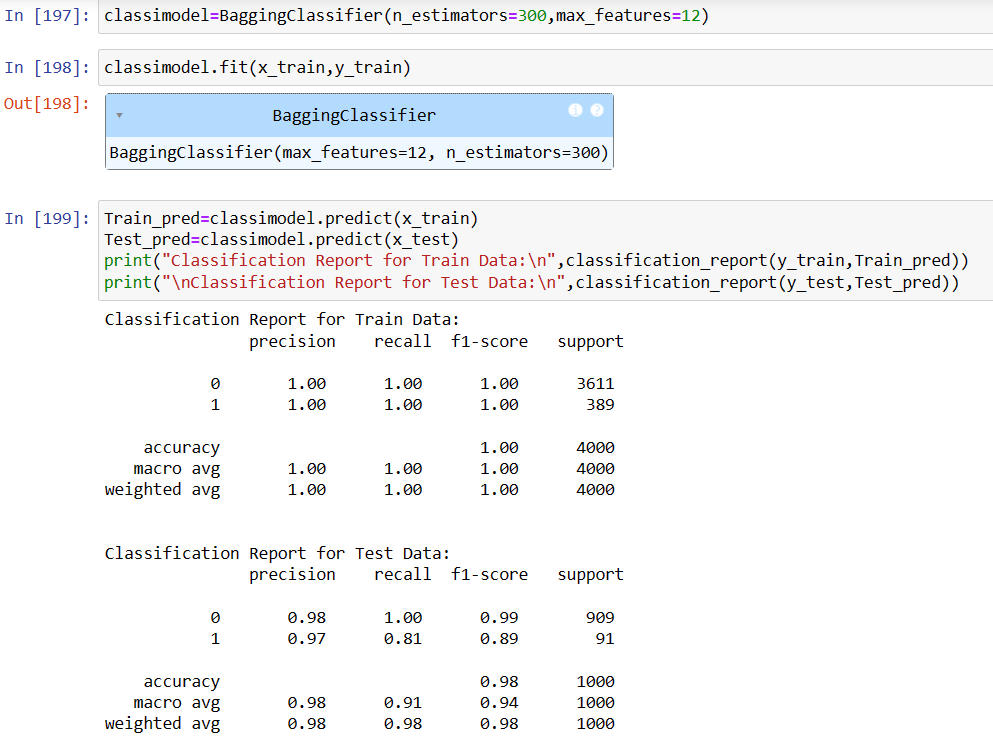
**Plotting PR graph**



**Plotting ROC Curve**



**CONCLUSION**



**So our final model has**

|  |  |
| --- | --- |
| **METRICES** | **SCORE** |
| Precision | 97% |
| Recall | 81% |
| F1-Score | 89% |
| Accuracy | 98% |

**Making prediction on Random Values:**

