CAPSTONE PROJECT

AUTOMOBILE LOAN DEFAULT PREDICTIONS

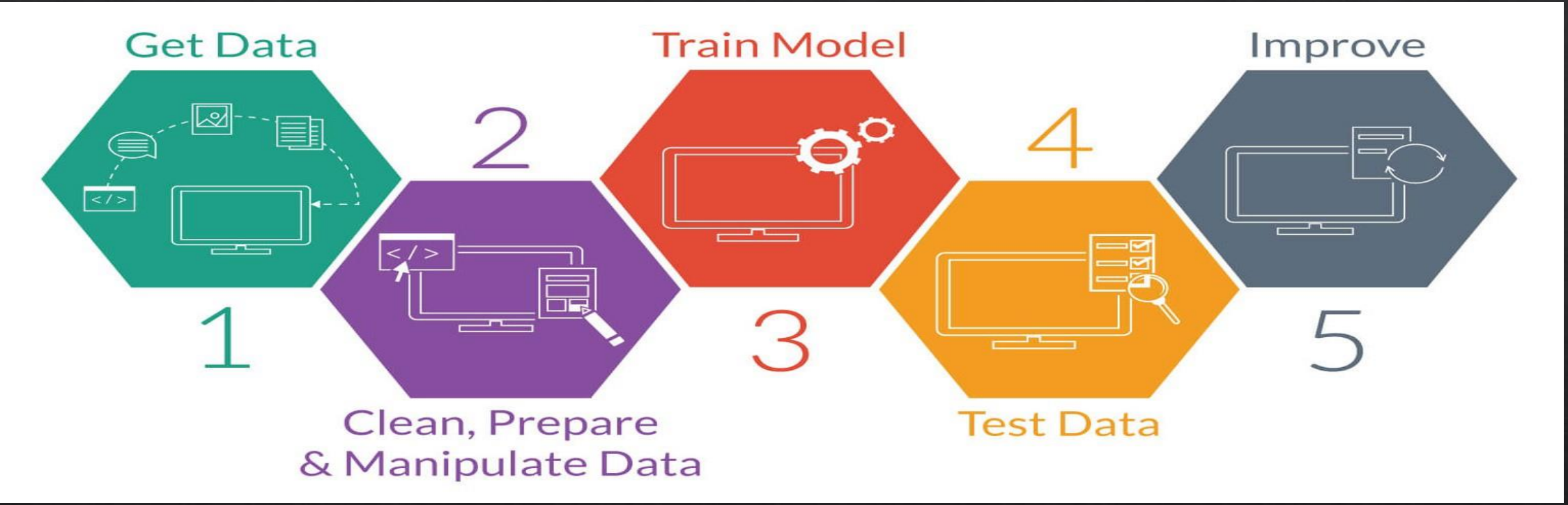
PROBLEM STATEMENT

- A non-banking financial institution (NBFI) or non-bank financial company (NBFC) is a Financial Institution that does not have a full banking license or is not supervised by a national or international banking regulatory agency. NBFC facilitates bank-related financial services, such as investment, risk pooling, contractual savings, and market brokering.
- An NBFI is struggling to mark profits due to an increase in defaults in the vehicle loan category. The company aims to determine the client's loan repayment abilities and understand the relative importance of each parameter contributing to a borrower's ability to repay the loan
- **The goal of the problem is to predict whether a client will default on the vehicle loan payment or not.**

SCOPE OF PROJECT

Developing a machine learning model that will automatically predict car loan defaults which would help banks take the call on whether to grant car loans to a person or not. With the data-driven regular assessment of customer credibility, credit risk monitoring is being performed proactively by observing clients' behavioral aspects in order to avoid loss and predict the defaulter status beforehand

DATA ANALYSIS PROCESS



The data set contains 121856 rows and 40 columns.

Dtypes: Float (15)

Int (5)

Object(20)

The target variable (Default) is highly imbalanced.

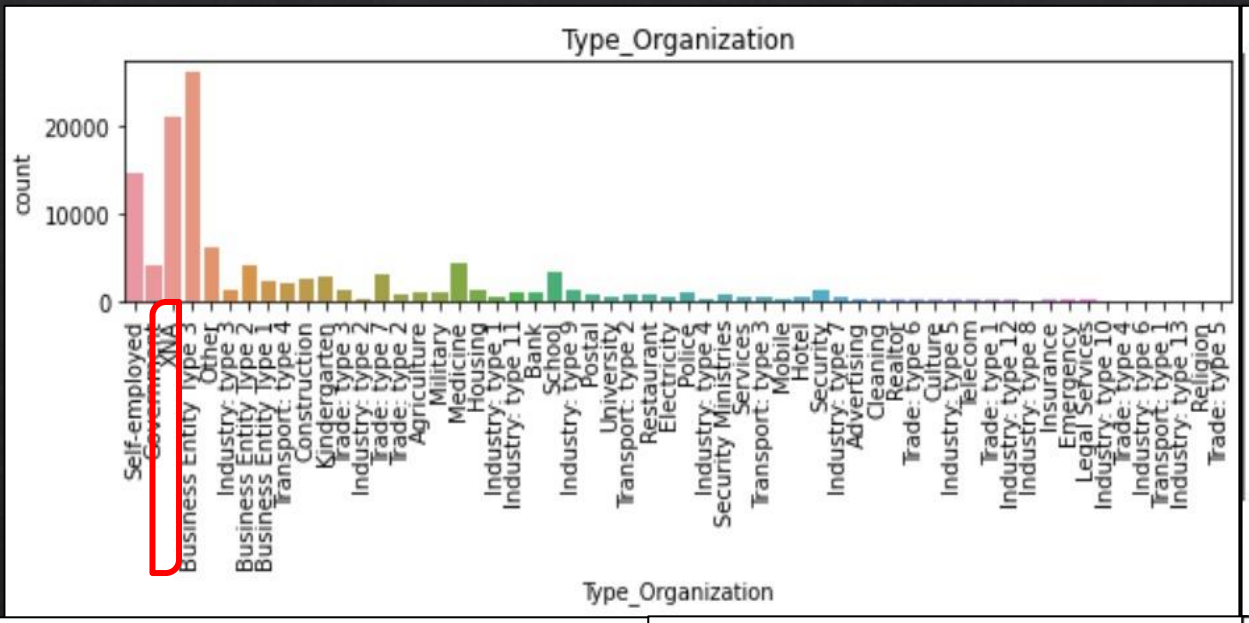
DATA CLEANING

Sno	Column name	Null Values before cleaning	Null Values after cleaning	Difference
1	Client_Income	3607	3622	15
2	Credit_Amount	3632	3637	5
3	Loan_Annuity	4812	4826	14
4	Population_Region_Relative	4857	4868	11
5	Age_Days	3600	3617	17
6	Employed_Days	3649	3666	17
7	Registration_Days	3614	3631	17
8	ID_Days	5968	5985	17
9	Score_Source_3	26921	26922	1

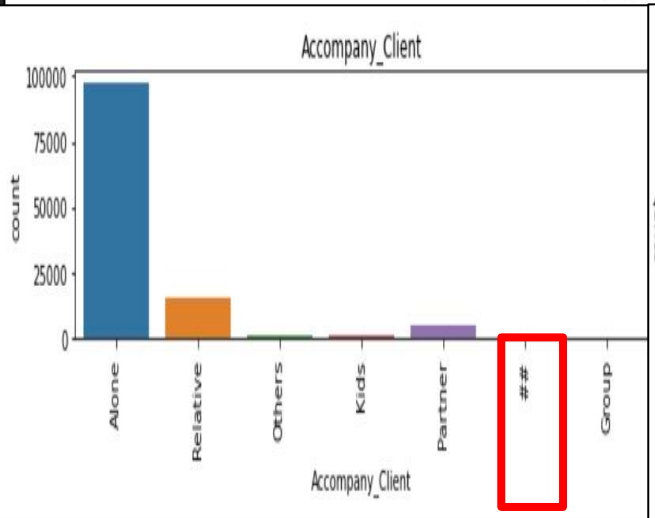
There are lots of nonstandard missing values present in the data set like - $, #value, @, #, x, &. Further, these are converted into standard missing values.

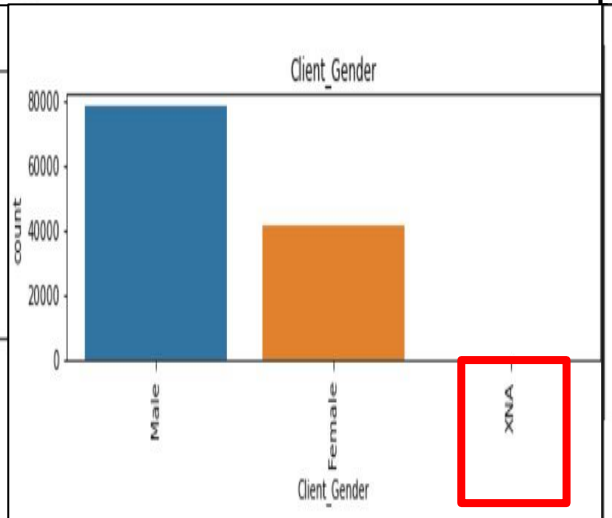
DATA CLEANING

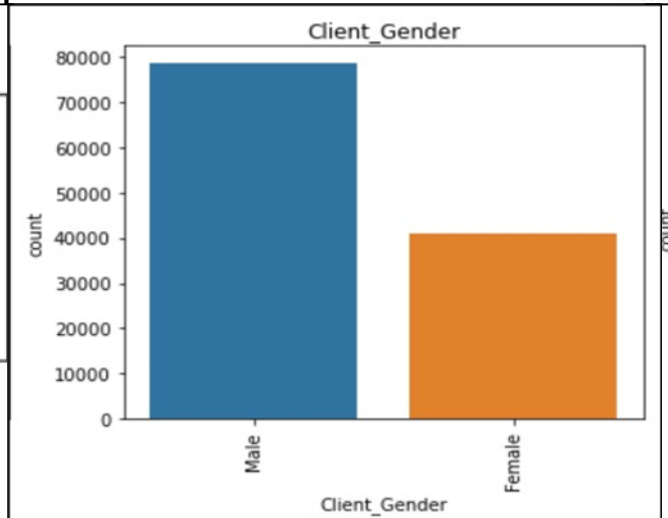
BEFORE

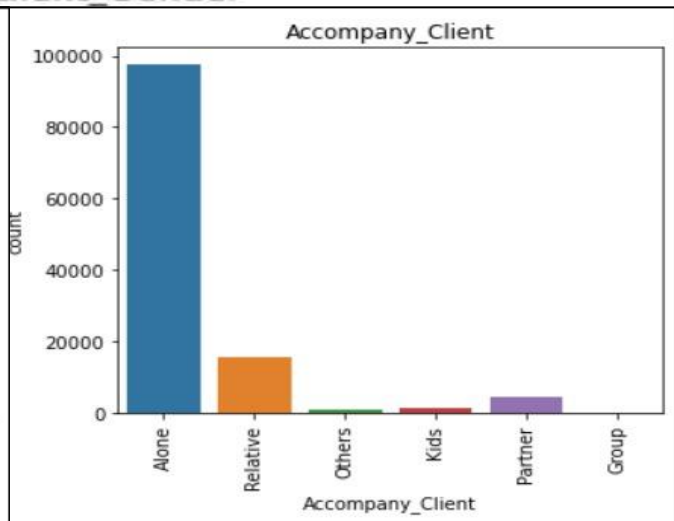


AFTER

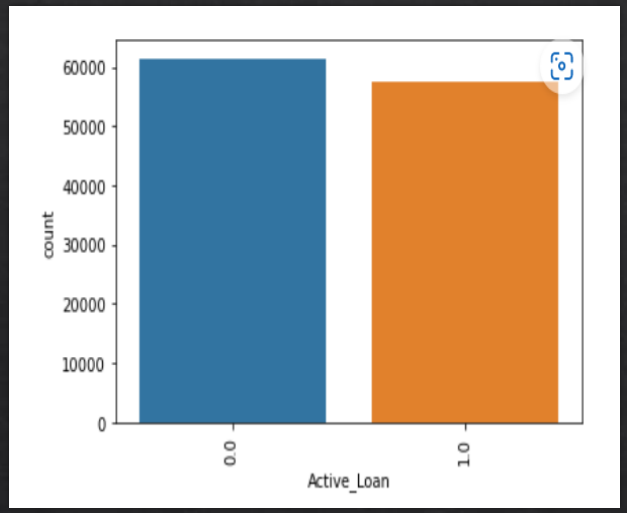


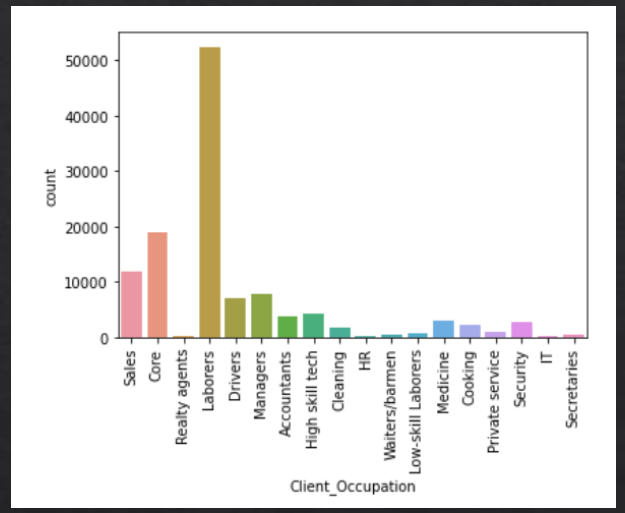


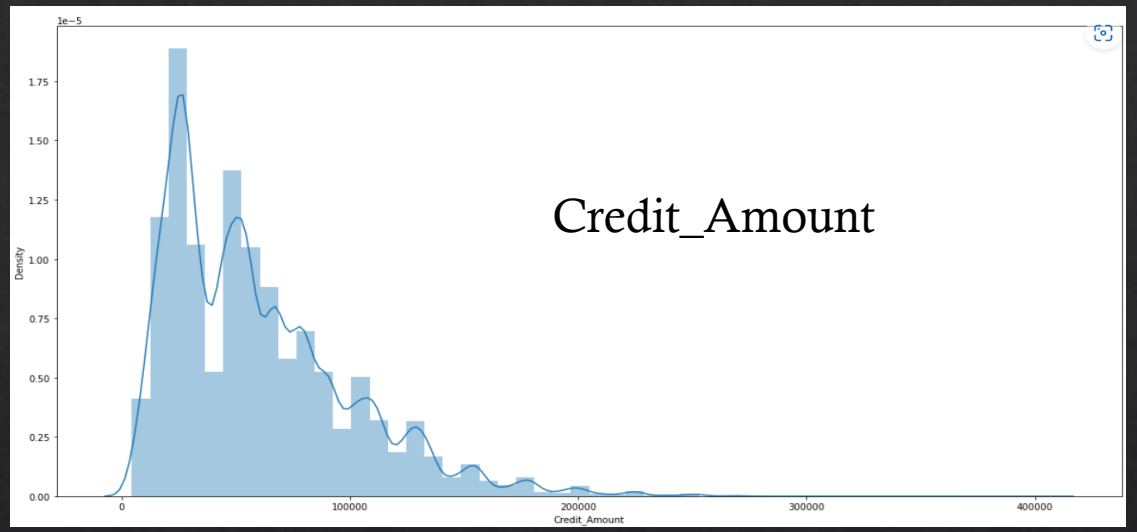


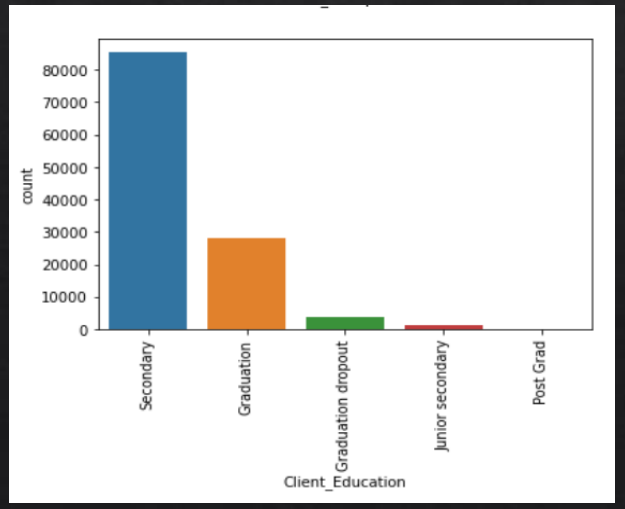


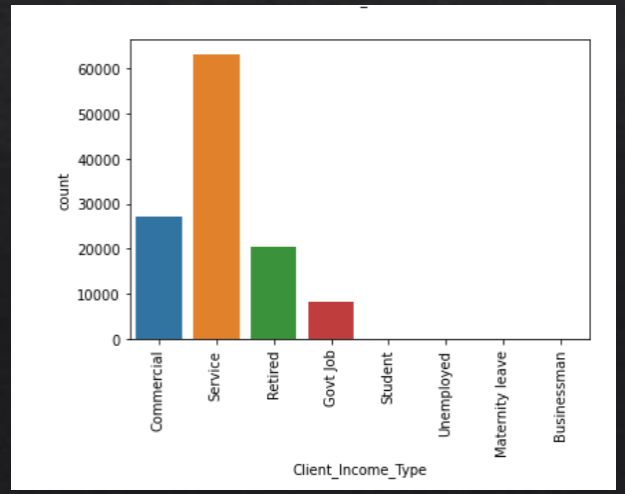
DATA EXPLORATION - UNIVARIATE ANALYSIS

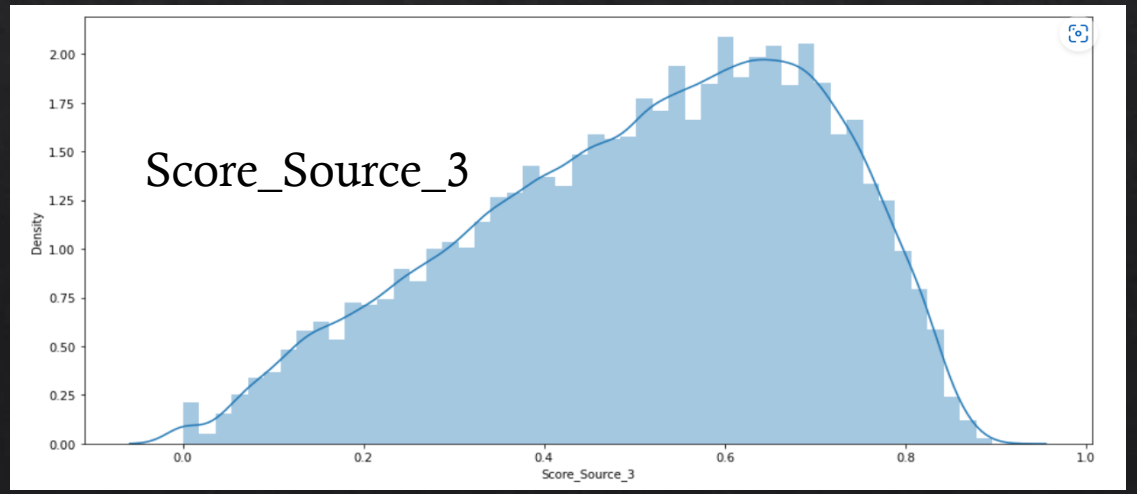








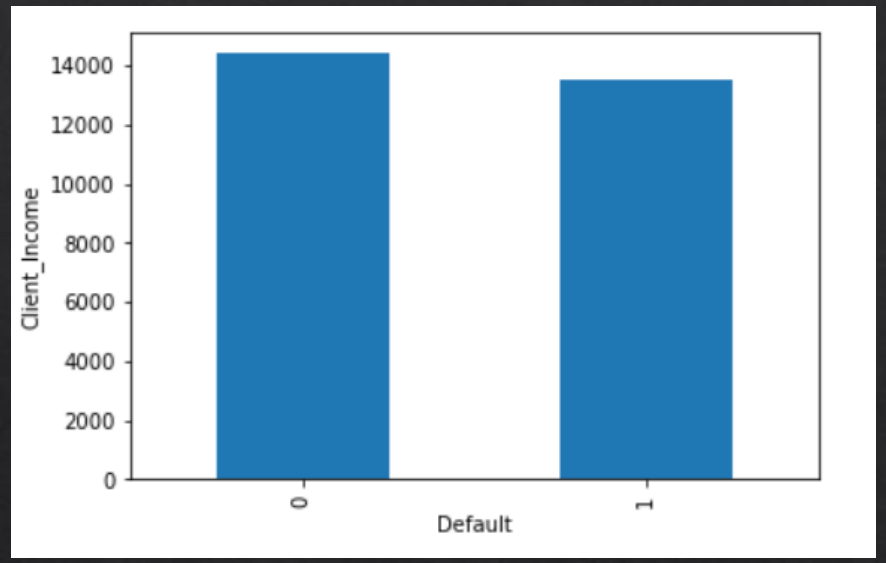


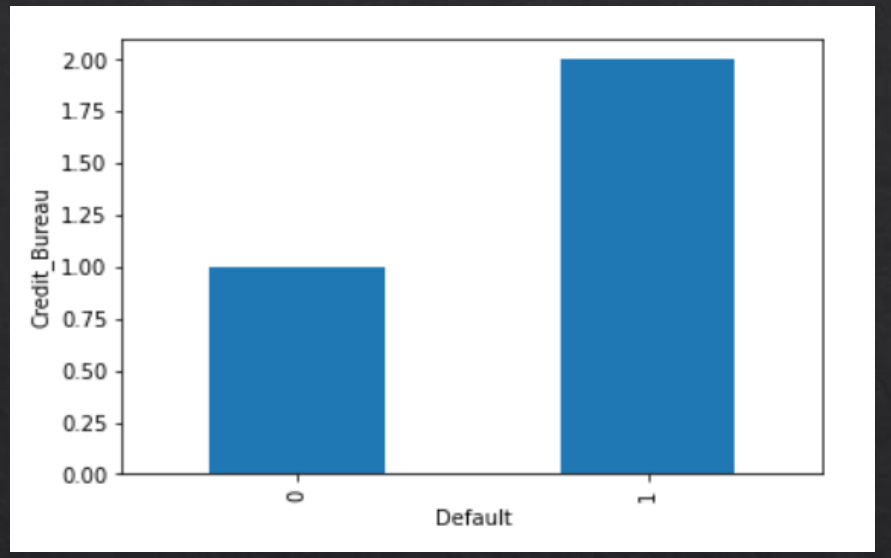


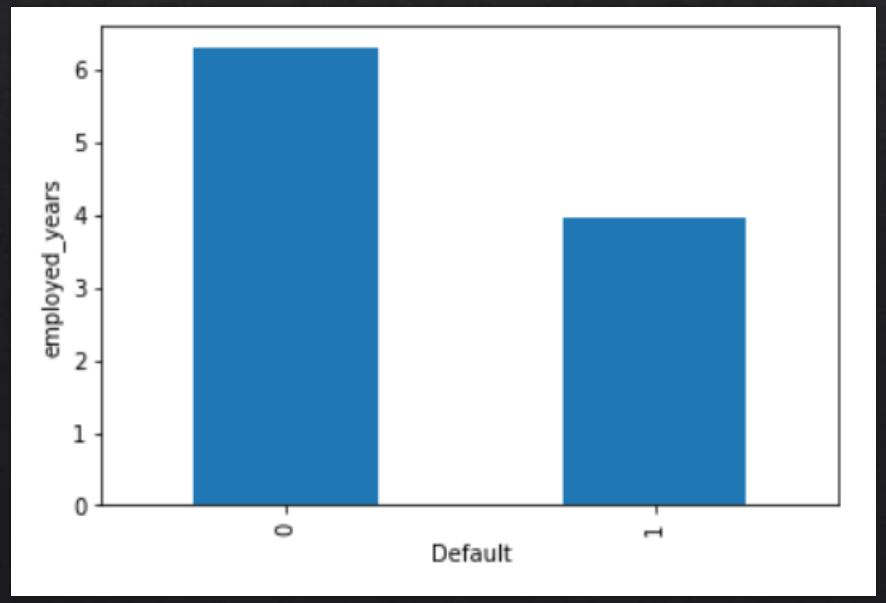
INFERENCE BY UNIVARIATE ANALYSIS

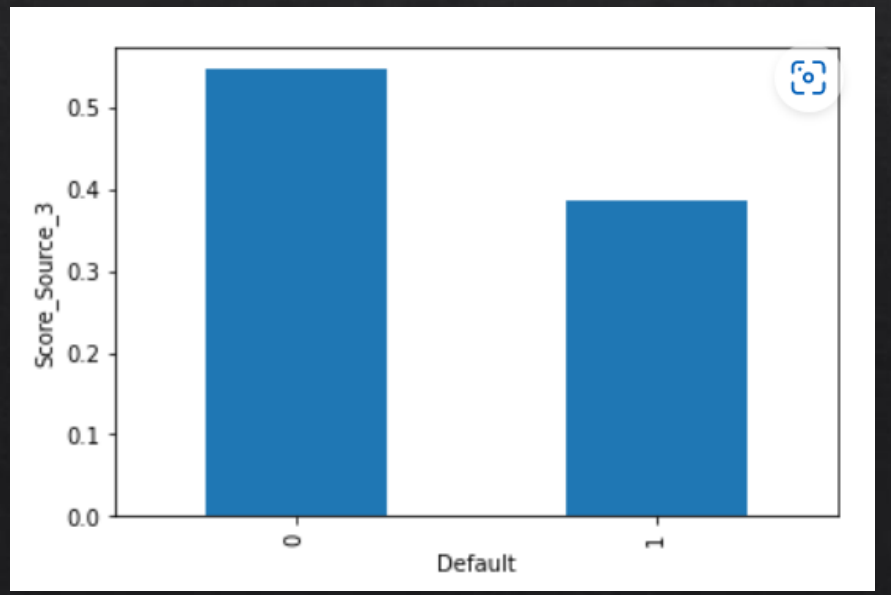
- Huge outliers present in the client income (that's a possible case).
- The loan amount is also skewed but most of the customers take loans between 20000$ and 80000$
- Loan annuity of most of the customers lies between 1000$ to 4000$
- Huge outlier present in population_region_relative – needs some treatment
- Most of the customers belong to the age group of 25-65
- Huge outliers present in employed years needed to be treated
- A high percentage of customers changed their registration in the last 10 years
- Most of the customers change their identity document within 12 years

DATA EXPLORATION - BIVARIATE ANALYSIS









INFERENCE BY BIVARIATE ANALYSIS

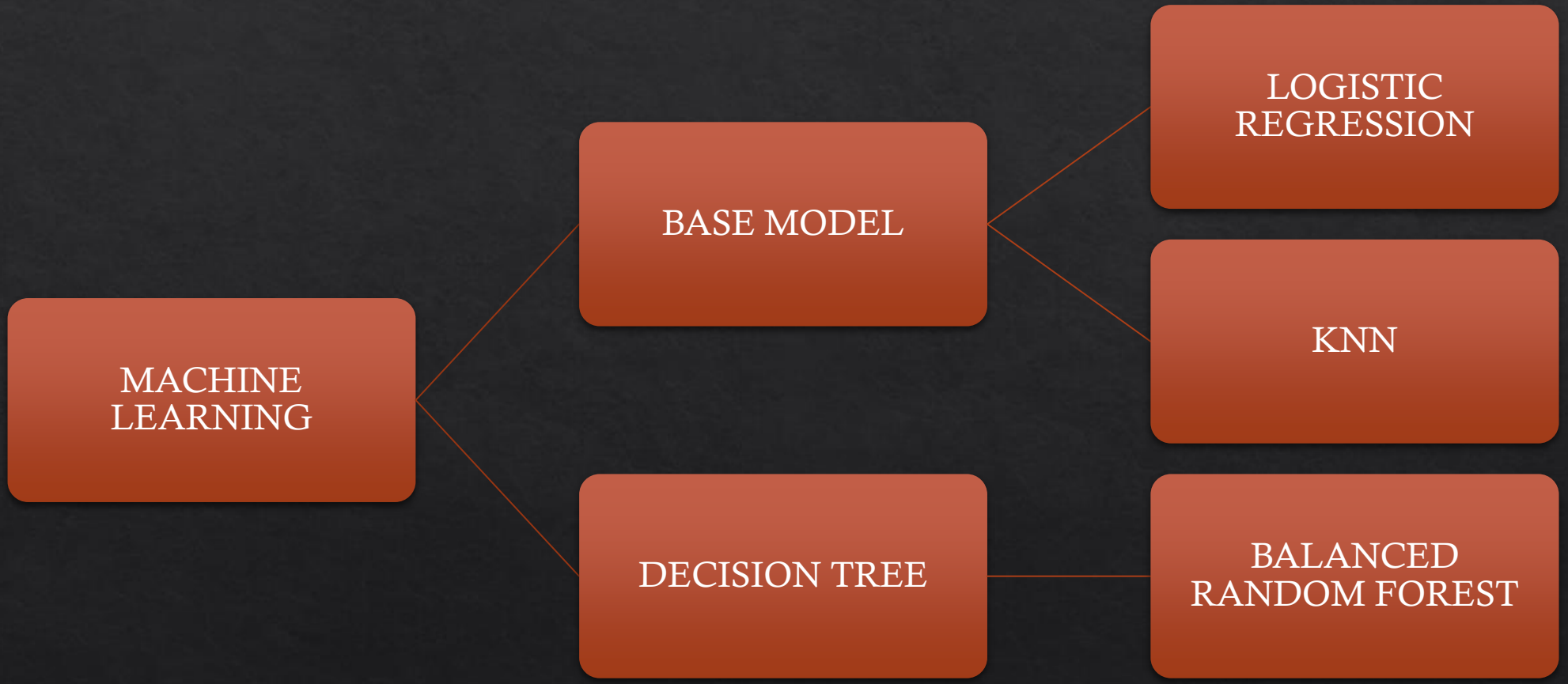
- Defaulter's income is fewer than the non-defaulters.
- Loan amt taken by both of them are merely equal.
- Loan annuity amounts paid by both defaulters and non-defaulters are merely equal.
- Population_region_relative is almost equal for both of them.
- Defaulters have less age as compared to non-defaulters.
- Defaulters have less working experience (about 4 years).
- Defaulters have changed their registration more recently as compared to non-defaulters.
- Defaulters change their id in less than 8 years.

Missing value

Client_Income	2.972361
Car_Owned	2.938715
Bike_Owned	2.974002
Active_Loan	2.983029
House_Own	3.004366
Child_Count	2.985491
Credit_Amount	2.984670
Loan_Annuity	3.960412
Accompany_Client	1.442686
Client_Income_Type	3.037191
Client_Education	2.991236
Client_Marital_Status	2.850085
Client_Gender	1.982668
Loan_Contract_Type	2.996159
Client_Housing_Type	3.025702
Population_Region_Relative	3.994879
Age_Days	2.968258
Employed_Days	3.008469
Registration_Days	2.979747
ID_Days	4.911535
Own_House_Age	65.729221
Mobile_Tag	0.000000
Homephone_Tag	0.000000
Workphone_Working	0.000000
Client_Occupation	34.003250
Client_Family_Members	1.977744
Cleint_City_Rating	1.976924
Application_Process_Day	1.992516
Application_Process_Hour	3.006007
Client_Permanent_Match_Tag	0.000000
Client_Contact_Work_Tag	0.000000
Type_Organization	20.264903
Score_Source_1	56.488806
Score_Source_2	4.666163
Score_Source_3	22.093290
Social_Circle_Default	50.820641
Phone_Change	3.006828
Credit_Bureau	15.214680
Default	0.000000

- Most of the columns have missing values in the range of 2-3%
- Own_House_Age, Score_Source_1, and Social_Circle_Default have null values greater than 50%.
- Whereas columns- Client_Occupation,Type_Organization,Score_Source_3 and Credit Bureau have null values 34%,20.2%,22% and 15.2% respectively.
- Missing values are filled with their respective median and mode values.

Model Building



ENCODING

- There are some columns in our data in which there are a lot of categories that the reason due to which we **frequency encoding technique**.
- And for the binary data we use **dummy encoding**.

BEFORE FREQUENCY ENCODING

Accompany_Client	Client_Income_Type	Client_Education
Alone	Commercial	Secondary
Alone	Service	Graduation
Alone	Service	Graduation dropout
Alone	Retired	Secondary
Alone	Commercial	Secondary

AFTER FREQUENCY ENCODING

Accompany_Client	Client_Income_Type	Client_Education
0.813717	0.227772	0.718852
0.813717	0.531804	0.236305
0.813717	0.531804	0.032443
0.813717	0.172336	0.718852
0.813717	0.227772	0.718852

TRAIN /TEST DATA SPLIT

We split two variables as below:

1. x= Independent variables
2. y=Dependent variable

```
X=df2.drop('Default',axis=1)
y=df2.Default
```

We split the dataset into training and testing with the 80: 20 ratio.

```
x_train,x_test,y_train,y_test=train_test_split(X,y,stratify=y,test_size=0.2,random_state=0)
```

SMOTE

Our y_train is highly imbalanced, thus we have used smote. To balance our dataset during the learning phase.

Before smote

```
1  y_train.value_counts()
```

```
0    87450
1     7708
Name: Default, dtype: int64
```

After smote

```
1  y_train_s.value_counts()
```

```
0    87450
1    87450
Name: Default, dtype: int64
```

LOGISTIC REGRESSION(BASE MODEL)

- First we tried a linear model like logistic regression but due to the poor performance, we moved toward other models like – Distance based (KNN), Tree based model (Decision Tree), and Boosting models (Ada boost).

Before smote

```
confusion_matrix
 [[21859     4]
 [ 1919     8]]
cohen_kappa_score 0.007256367386865303
accuracy 0.9191677175283732
recall 0.004151530877010 8976
auc_score 0.5019842866844461
classification_report
              precision    recall  f1-score   support

           0       0.92      1.00      0.96     21863
           1       0.67      0.00      0.01      1927

    accuracy                           0.92     23790
   macro avg       0.79      0.50      0.48     23790
weighted avg       0.90      0.92      0.88     23790
```

After smote

```
confusion_matrix
 [[15894  5969]
 [  890  1037]]
cohen_kappa_score 0.1204179850824526
accuracy 0.7116855821773854
recall 0.5381421899325376
auc_score 0.6325619242211744
classification_report
              precision    recall  f1-score   support

           0       0.95      0.73      0.82     21863
           1       0.15      0.54      0.23      1927

    accuracy                           0.71     23790
   macro avg       0.55      0.63      0.53     23790
weighted avg       0.88      0.71      0.77     23790
```

MODEL COMPARISION

	Performance Measure			
Model	**Precison**	**Recall**	**F1 score**	**AUC Score**
Logistic Regression	0.13	0.50	0.20	0.64
Decision Tree	0.18	0.27	0.22	0.58
KNN classifier	0.14	0.60	0.22	0.63
Ada Boost	0.18	0.28	0.22	0.58
BalancedRandomForest	0.17	0.70	0.28	0.70

Our best performance model is balanced random forest on the basis of Recall, and Auc-score.

THANK YOU