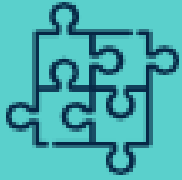


A circular word cloud featuring a variety of financial and business-related terms. The words are arranged in a circular pattern, with some words being significantly larger than others, indicating their frequency or importance. The colors of the words vary, including shades of blue, green, yellow, and red. The background is a solid light blue.

Key words visible in the word cloud include:

- Predict** (large, red)
- Machine Learning** (large, green)
- Risk** (large, blue)
- Loan** (large, blue)
- Bank** (large, blue)
- Investments** (large, blue)
- Offer** (large, blue)
- Training** (large, green)
- Applicant** (large, blue)
- Lending** (large, blue)
- Interest Rate** (large, green)
- Firms** (large, blue)
- Funds** (large, blue)
- Significant** (large, blue)
- Industry** (large, green)
- Benefits** (large, blue)
- Algorithm** (large, green)
- Project** (large, blue)
- Features** (large, green)
- Market** (large, blue)
- Customers** (large, blue)
- Accounts** (large, blue)
- Worthiness** (large, blue)
- AI** (large, blue)
- Category** (large, blue)
- Credit** (large, blue)
- Companies** (large, blue)
- Test** (large, blue)
- Machine Learning** (large, green)
- Algorithm** (large, green)
- Project** (large, blue)
- Features** (large, green)
- Market** (large, blue)
- Customers** (large, blue)
- Accounts** (large, blue)
- Worthiness** (large, blue)
- AI** (large, blue)
- Category** (large, blue)
- Credit** (large, blue)
- Companies** (large, blue)
- Test** (large, blue)
- Machine Learning** (large, green)
- Algorithm** (large, green)
- Project** (large, blue)
- Features** (large, green)
- Market** (large, blue)
- Customers** (large, blue)
- Accounts** (large, blue)
- Worthiness** (large, blue)
- AI** (large, blue)
- Category** (large, blue)
- Credit** (large, blue)
- Companies** (large, blue)
- Test** (large, blue)



## Problem



To predict the interest rate category (1 / 2 / 3) that will be assigned to each loan of a customer based on their past data.



## Target

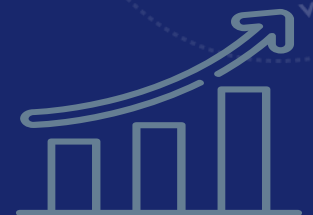


Interest\_Rate Predictions for different customers.

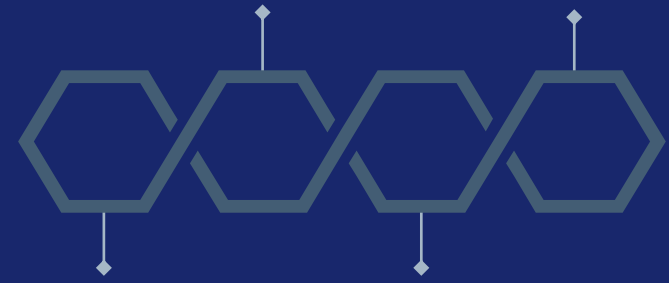


## Technology used.

Supervised Learning - Classification



# Our Process



1

- EDA
- Visualisations
- Missing value imputation
- outlier treatment

2

- Feature Engineering
- Statistical analysis

3

- Feature selection
- Balancing Data

4

- Base model - evaluation
- Testing various models and evaluating

5

- Hyper Parameter Tuning of the best Model



# Dataset Information

This dataset was provided by Analytics Vidhya for a banking sector problem. The dataset had 164309 Rows and 14 columns in csv format. The data comprises of different features pertaining to various factors of every customer applying for loan.

Variables
Loan_ID
Loan_Amount_Requested
Length_Employed
Home_Owner
Annual_Income
Income_Verified
Purpose_Of_Loan
Debt_To_Income
Inquiries_Last_6Mo
Months_Since_Delinquency
Number_Open_Accounts
Total_Accounts
Gender
Interest_Rate

Numerical
Loan_Amount_Requested
Length_Employed
Annual_Income
Debt_To_Income
Inquiries_Last_6Mo
Months_Since_Delinquency
Number_Open_Accounts
Total_Accounts
Total 8 Features

Categorical
Home_Owner
Income_Verified
Purpose_Of_Loan
Gender
Interest_Rate
Total 5 Features

Missing Values	
Length_Employed	7371
Home_Owner	25349
Annual_Income	25102
Months_Since_Delinquency	88379
8% of total Data values	

# Data Cleaning

# Feature Engineering

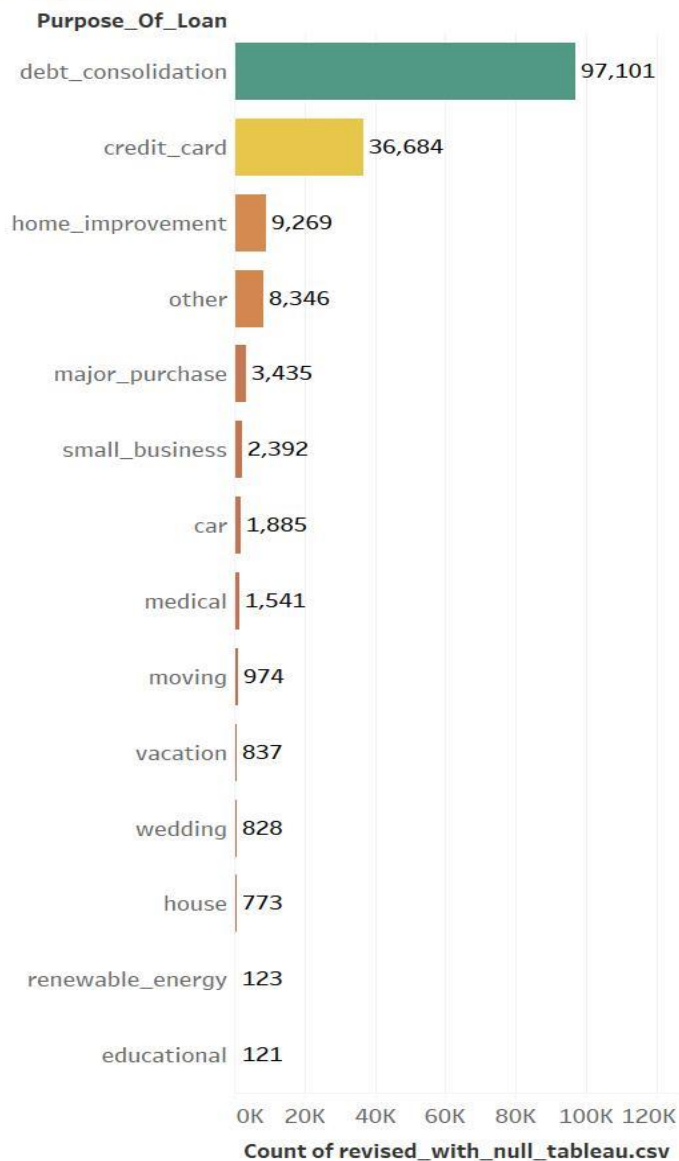
- **Loan\_ID** : Removed as all values were unique.
- **Loan\_Amount\_Requested** : Converted into numeric as was in string because of “,” in between.
- **Length\_Employed** :Converted into numeric from strings.

- **Closed\_to\_total\_ratio** : Higher the percentage better the customer.
- **Assets or Liabilities** : Categorising Purpose\_of\_loan column into Asset , Liability & Others.
- **Financial Growth score** : It explains the customer growth compared to others. [Conceptual only]

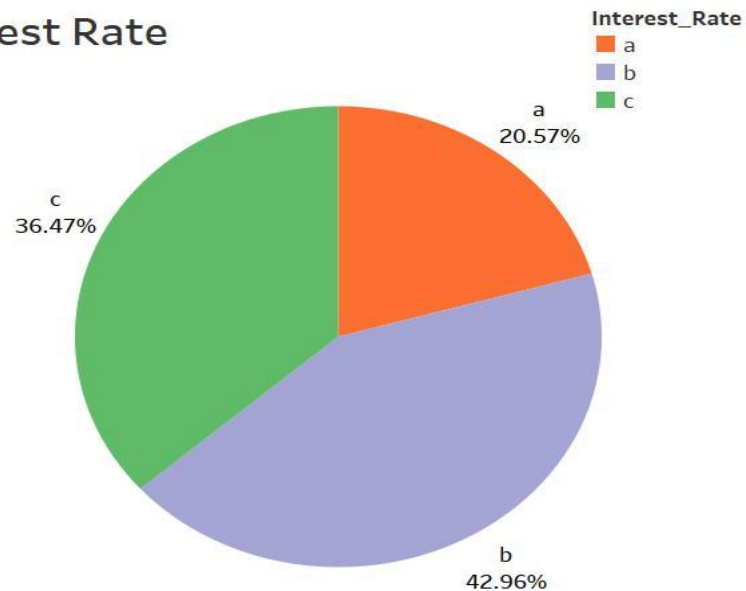


# Visualisation (A)

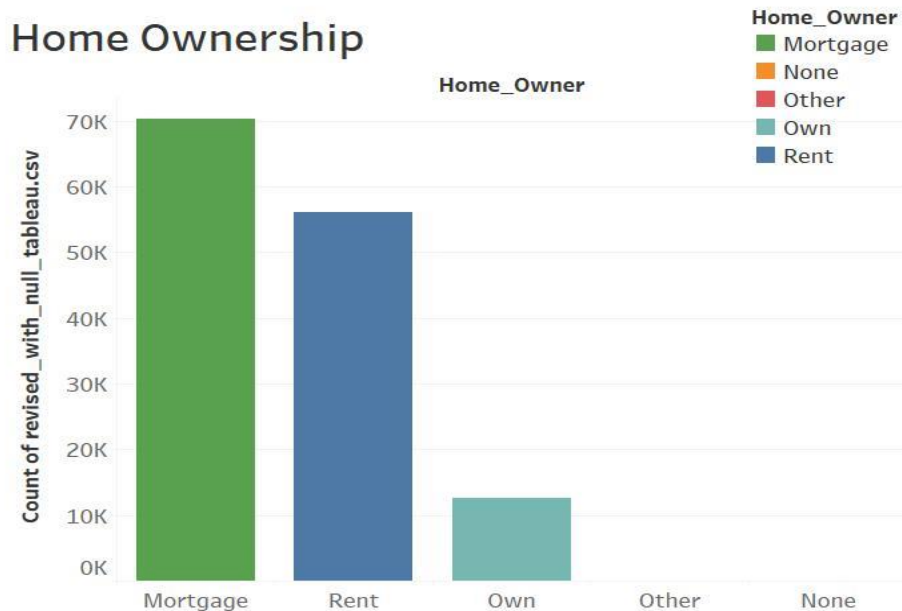
## Purpose



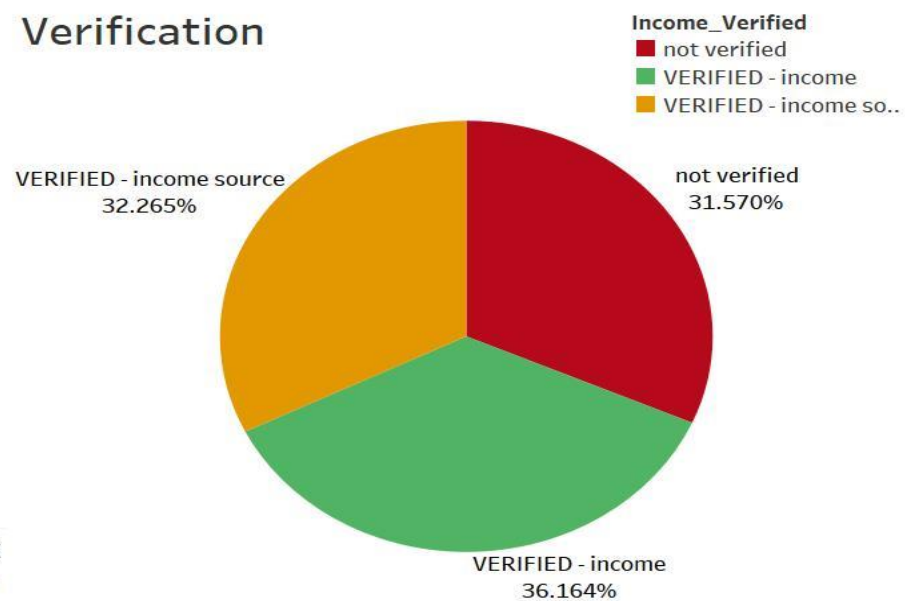
## Interest Rate



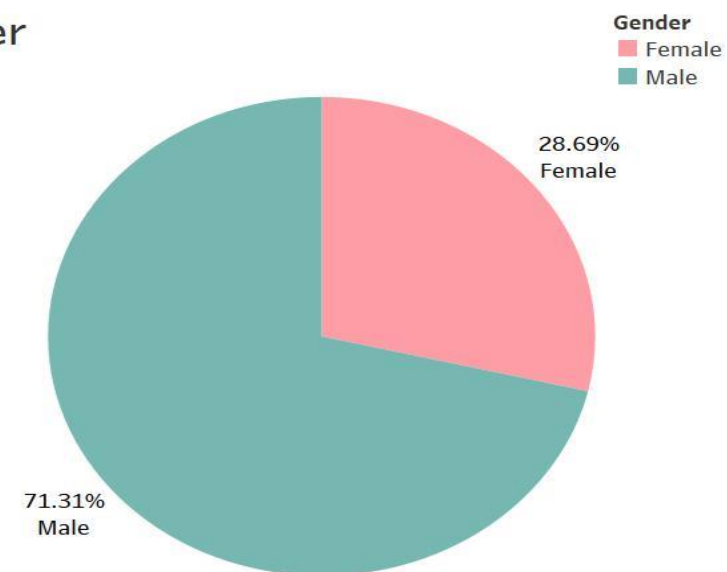
## Home Ownership



## Verification



## Gender

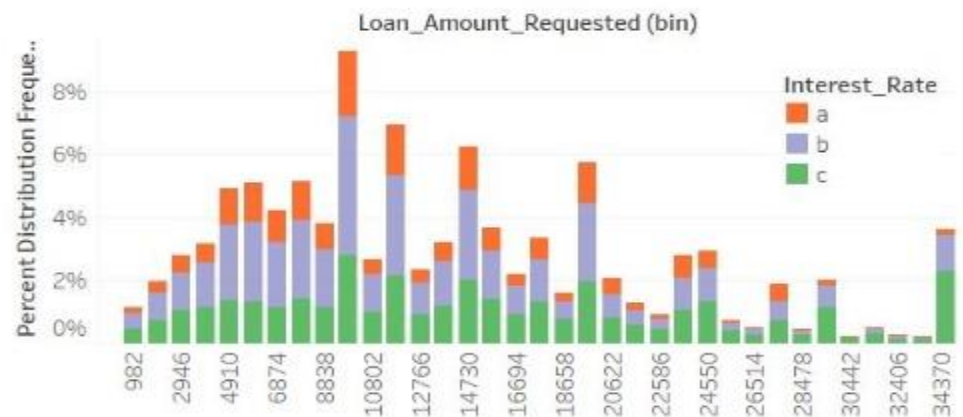




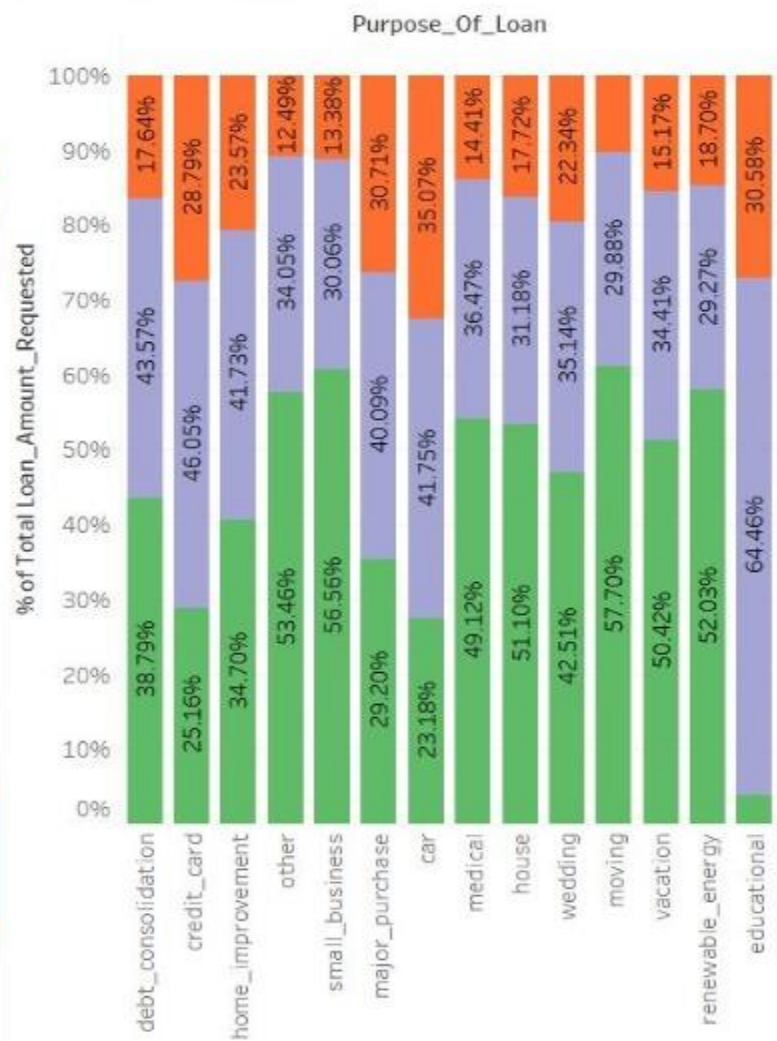
# Visualisation (B)



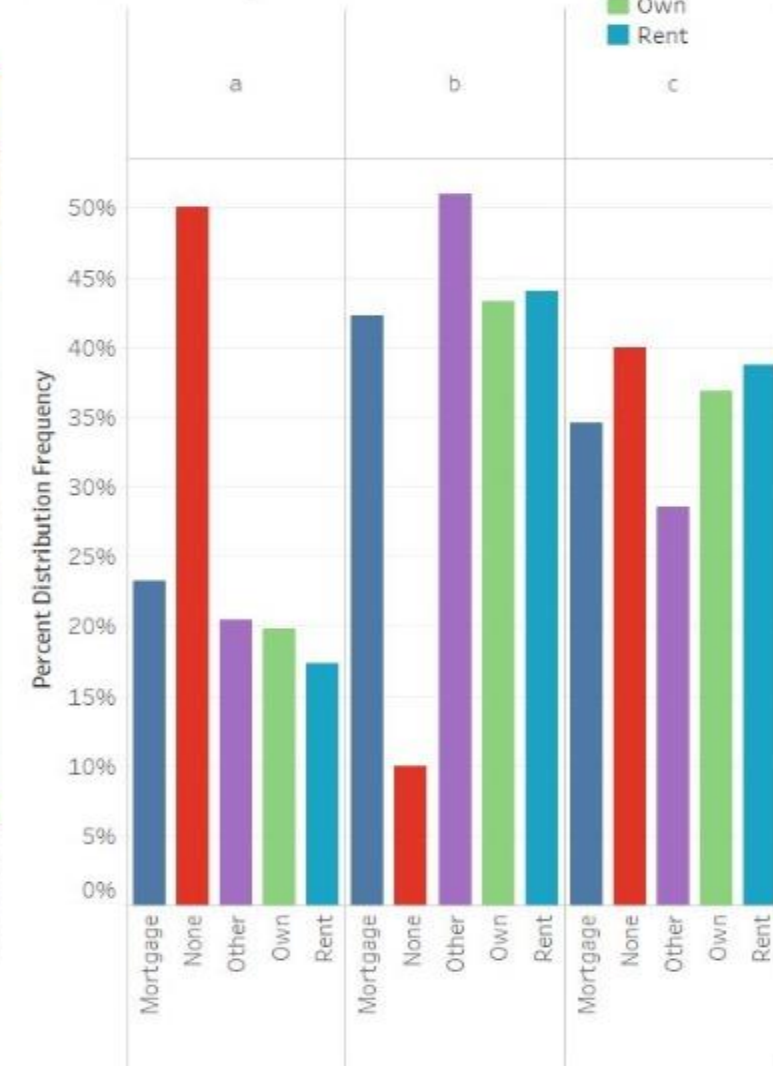
## Loan Amount vs Interest Rate



## Loan Purpose and Interest Rate Distribution



## Interest Rate vs Home ownership



## Interest Rate + Verification vs Annual Income



# Missing Values

The dataset has a total of 8 % missing values.

1. **Home ownership** : null values replaced with 'none'.
2. **Employment length** : null values imputed with '<1 years'.
3. **Annual income** : null values filled with median 63000.
4. **Months since Delinquency** : Clients who have never missed a debt repayment belong to the null values, hence imputed with max+ range =  $180+180 = 360$ .

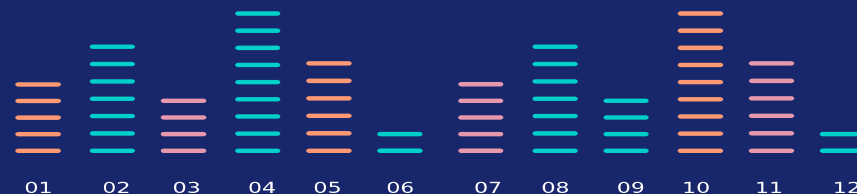
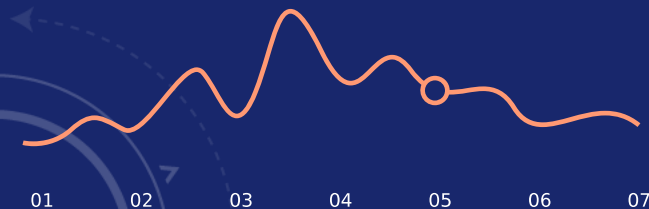
# Encoding

## One Hot Encoding :

- Home\_Owner
- Purpose\_Of\_Loan
- Gender

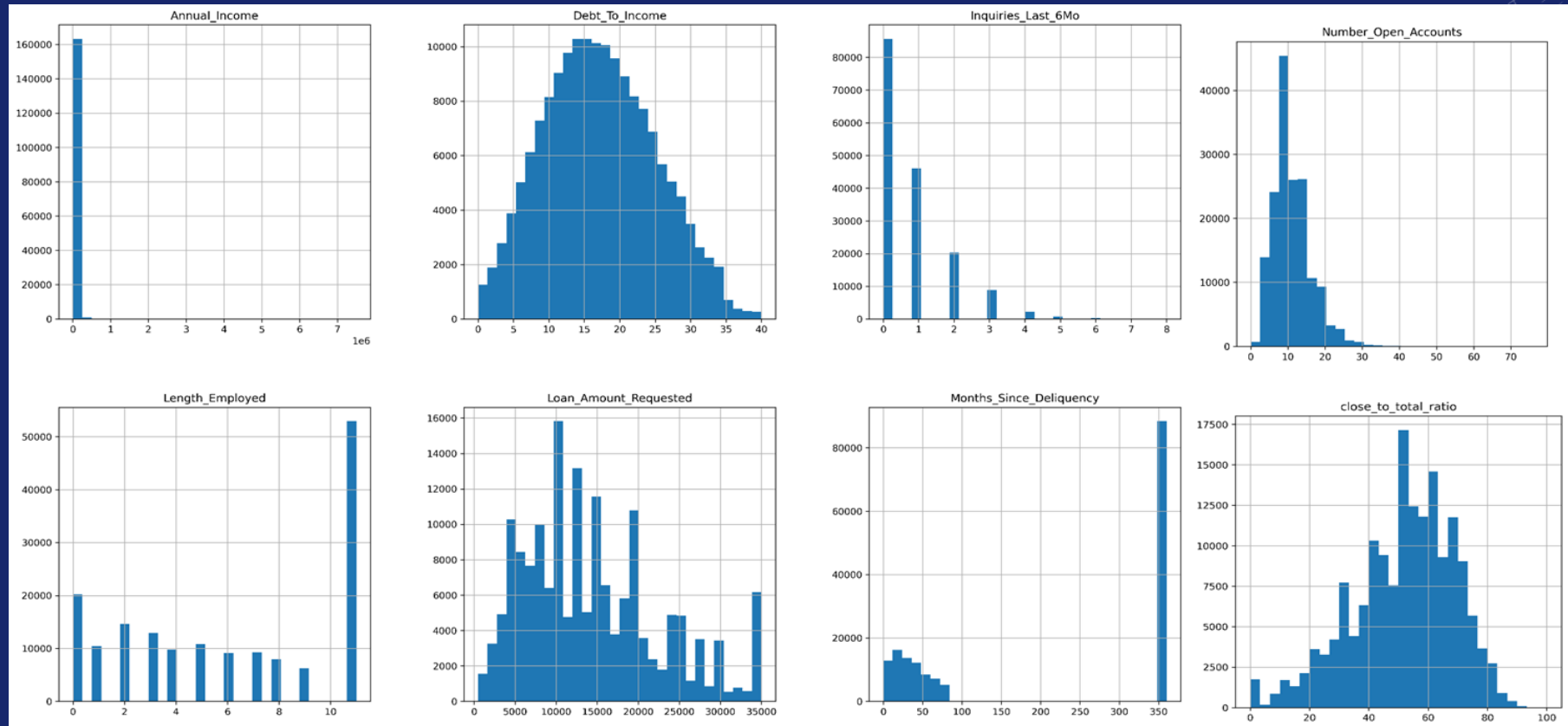
## Label Encoding :

- Assets\_liability
- Income verified

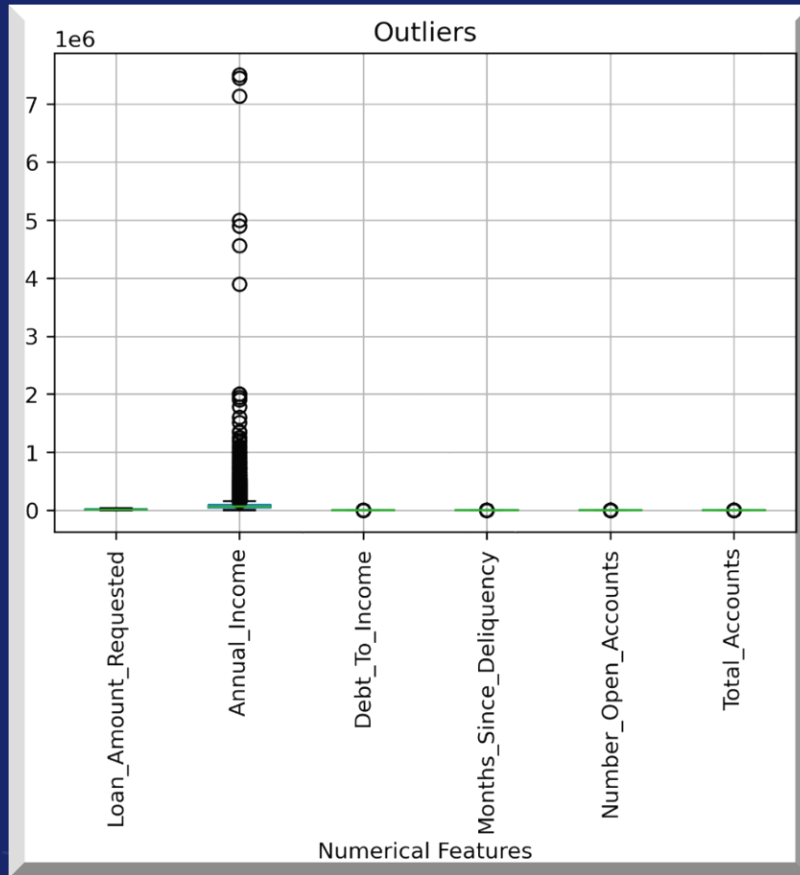


# Data Distribution

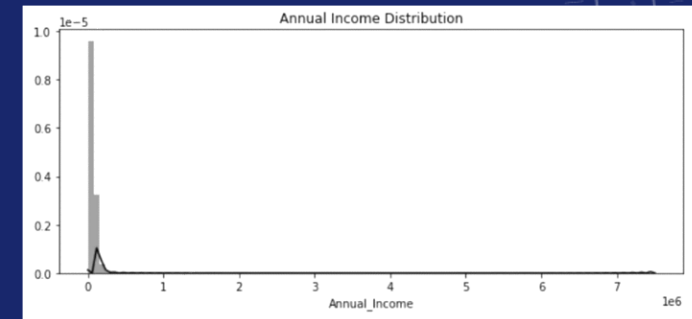
Using Data.describe() to create a 5 point summary of the data to get a better understanding of the numerical features in the dataset .



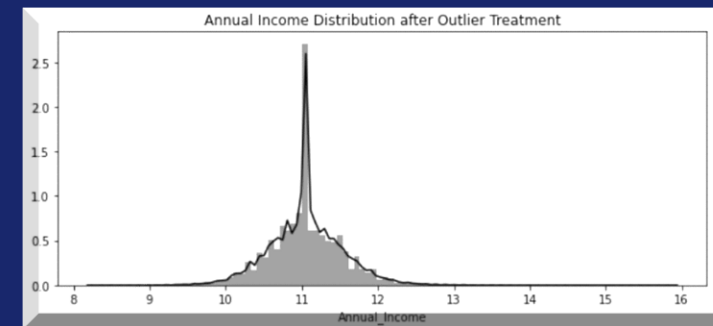
# Outliers and Treatment



`np.log(Annual_Income)`



Before



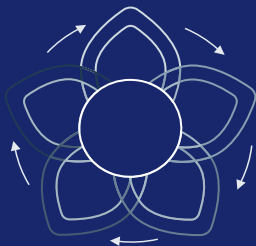
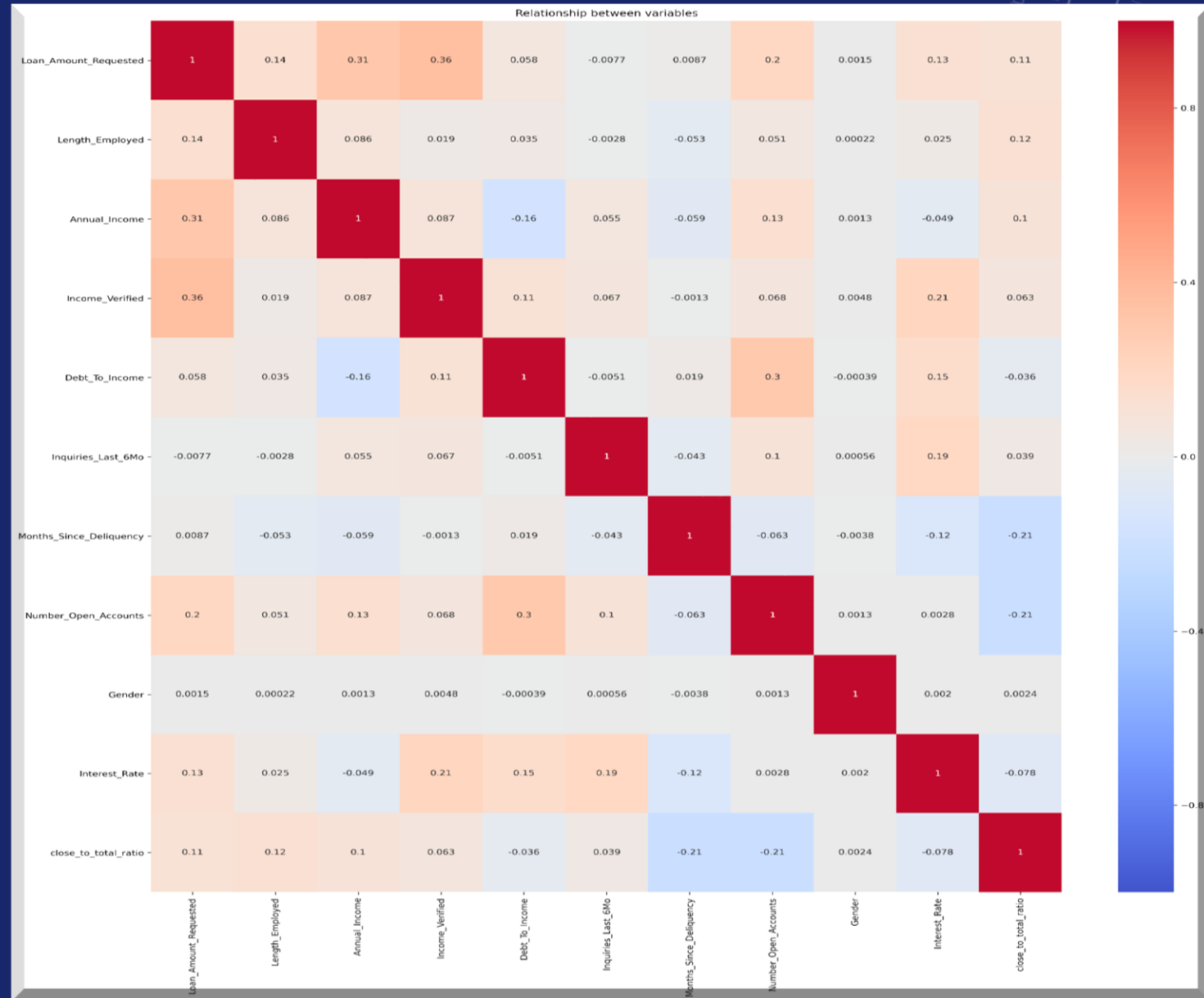
After

Apparently the 'Annual Income' feature is having many outliers.

# Feature Correlation Heat Map



We use Pearson correlation to find correlation among features and plot them on a heatmap in Seaborn.



# Statistics

Dep. Variable:	Interest_Rate	No. Observations:	164309
Model:	Logit	Df Residuals:	164292
Method:	MLE	Df Model:	16
Date:	Sun, 26 Jul 2020	Pseudo R-squ.:	0.08977
Time:	15:53:37	Log-Likelihood:	-1.0031e+05
converged:	True	LL-Null:	-1.1020e+05
Covariance Type:	nonrobust	LLR p-value:	0.000

	coef	std err	z	P> z	[0.025	0.975]
const	0.4247	0.036	11.721	0.000	0.354	0.496
Loan_Amount_Requested	3.323e-05	7.88e-07	42.158	0.000	3.17e-05	3.48e-05
Length_Employed	0.0140	0.001	10.711	0.000	0.011	0.017
Annual_Income	-3.256e-06	1.48e-07	-21.986	0.000	-3.55e-06	-2.97e-06
Income_Verified	0.0623	0.001	45.262	0.000	0.060	0.065
Debt_To_Income	0.0321	0.001	42.563	0.000	0.031	0.034
Inquiries_Last_6Mo	0.3367	0.006	60.151	0.000	0.326	0.348
Months_Since_Delinquency	-0.0016	3.36e-05	-48.055	0.000	-0.002	-0.002
Number_Open_Accounts	-0.0402	0.001	-33.040	0.000	-0.043	-0.038
Gender	0.0027	0.012	0.236	0.813	-0.020	0.025
close_to_total_ratio	-0.0143	0.000	-42.145	0.000	-0.015	-0.014
Home_Owner_None	0.1251	0.016	7.988	0.000	0.094	0.156
Home_Owner_Other	0.1480	0.307	0.482	0.630	-0.454	0.749
Home_Owner_Own	0.1154	0.021	5.562	0.000	0.075	0.156
Home_Owner_Rent	0.2859	0.013	22.355	0.000	0.261	0.311
Purpose_Of_Loan_Liability	-0.1911	0.020	-9.376	0.000	-0.231	-0.151
Purpose_Of_Loan_Others	0.2778	0.028	9.918	0.000	0.223	0.333

H0 -> Coef\_x = 0

H1 -> Coef\_x != 0

Two conditions:

> if P\_value > 0.05 Fail to reject H0

> if P\_value < 0.05 Reject H0

“Gender & Home\_Owner\_Other”

Have failed to reject the null hypothesis because p\_value is greater than 0.05 .



# Feature selection

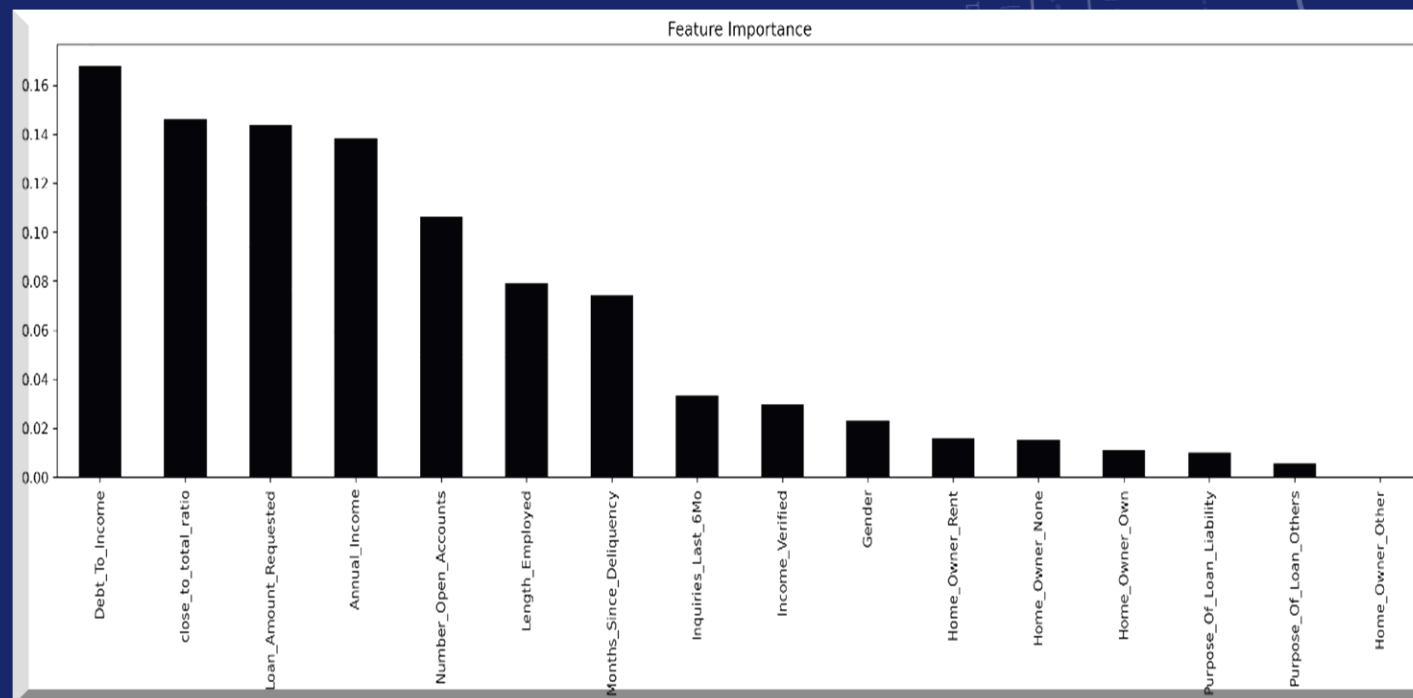
## Variance Inflation Factor

	vif
Loan_Amount_Requested	1.381524
Length_Employed	1.058955
Annual_Income	1.193171
Income_Verified	1.176391
Debt_To_Income	1.193632
Inquiries_Last_6Mo	1.030739
Months_Since_Delinquency	1.071113
Number_Open_Accounts	1.307091
Gender	1.000056
close_to_total_ratio	1.194204
Home_Owner_None	1.168579
Home_Owner_Other	1.000713
Home_Owner_Own	1.104473
Home_Owner_Rent	1.328451
Purpose_Of_Loan_Liability	1.874904
Purpose_Of_Loan_Others	1.846008



- $0 < Vif < 2$  Very less multicollinearity
- $2 < Vif < 5$  Moderate multicollinearity
- $5 < Vif < 10+$  High multicollinearity

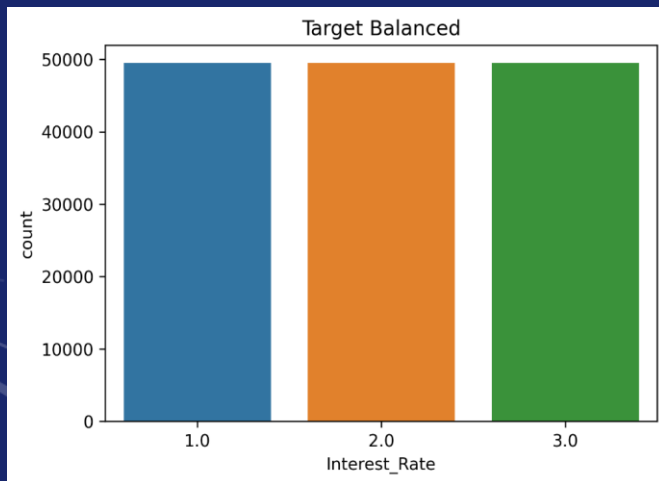
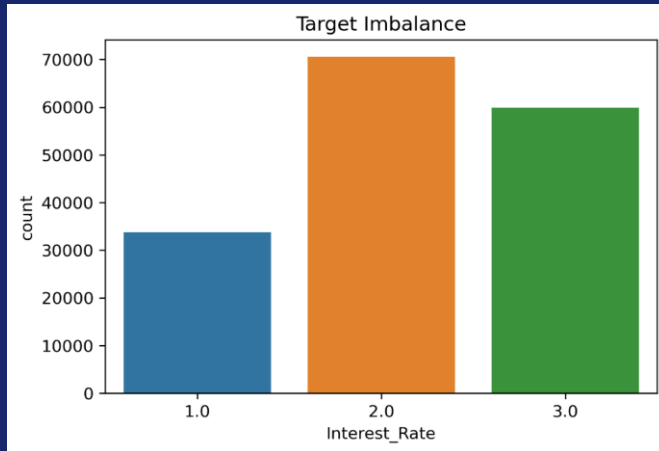
## Embedded Method



We used RandomForest to select features based on node impurities in each decision tree

“All features other than Home\_Owner\_other “

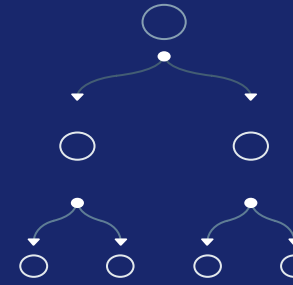
# Balance data



S  
M  
O  
T  
E

# Train Test Split

70 : 30



T\_Test 2 sample Independent

H0 -> Sample Mean = Population Mean

Ha -> Sample Mean != Population Mean

P\_values

Xtrain : [0.64, 0.81, 0.95, 0.76, 0.74, 0.57, 0.74, 0.83, 0.58, 0.91, 0.85, 1., 0.77, 0.96, 0.82]

Xtest : [0.41, 0.68, 0.92, 0.59, 0.56, 0.32, 0.56, 0.72, 0.33, 0.84, 0.75, 0.99, 0.6, 0.94, 0.7]

Ytest : 0.28

Ytrain : 0.54

Since all p\_values > 0.05 therefore we have failed to reject the null hypothesis.

# Base model

Here we will use Logistic Regression algorithm with 'multinomial' argument under the multiclass parameter as we have more than two classes in the target.

The report is as follows:

Target Class	precision	recall	f1-score	support
1	0.29	0.55	0.38	10077
2	0.48	0.33	0.39	21102
3	0.49	0.44	0.46	18114
accuracy			0.41	4923
macro avg	0.42	0.44	0.41	4923
weighted avg	0.45	0.41	0.42	4923

Our model gave an overall accuracy of **41%**.

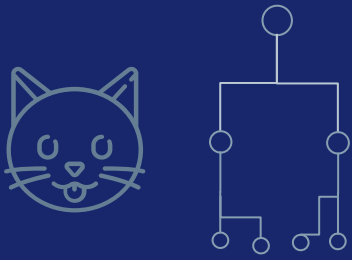
# PyCarat report



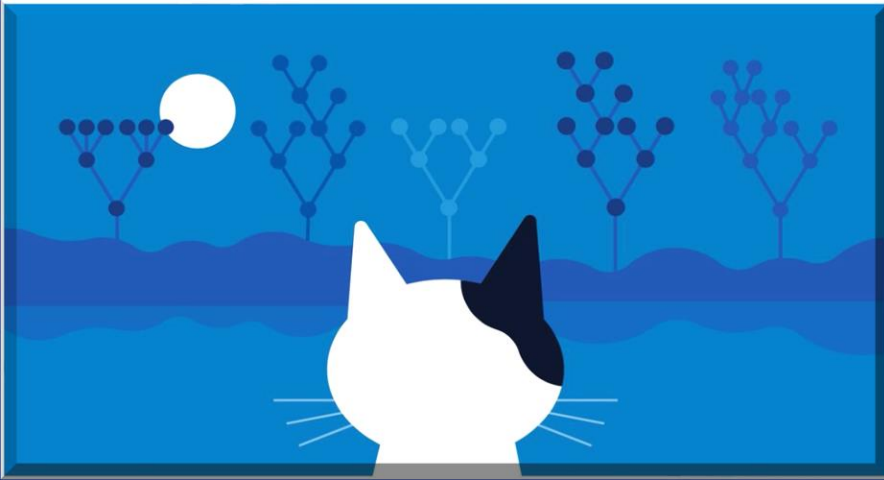
Model Comparison Table

	Model	Accuracy	AUC	Recall	Prec.	F1	Kappa
0	CatBoost Classifier	0.528800	0.000000	0.481500	0.531200	0.516600	0.234500
1	Gradient Boosting Classifier	0.527200	0.000000	0.468700	0.534900	0.506700	0.223300
2	Extreme Gradient Boosting	0.526100	0.000000	0.466300	0.534800	0.504300	0.220500
3	Light Gradient Boosting Machine	0.525900	0.000000	0.476500	0.527500	0.512200	0.228700
4	Ada Boost Classifier	0.519700	0.000000	0.462900	0.523800	0.500000	0.212800
5	Linear Discriminant Analysis	0.514500	0.000000	0.450400	0.523000	0.487500	0.198100
6	Ridge Classifier	0.508700	0.000000	0.432400	0.516900	0.465100	0.180500
7	Random Forest Classifier	0.480500	0.000000	0.437400	0.473200	0.467200	0.162700
8	Extra Trees Classifier	0.476000	0.000000	0.434600	0.468500	0.466000	0.156400
9	Logistic Regression	0.475500	0.000000	0.400700	0.464800	0.432100	0.122200
0	Naive Bayes	0.472700	0.000000	0.396100	0.461900	0.424600	0.113900
1	K Neighbors Classifier	0.424200	0.000000	0.379800	0.411100	0.409400	0.072100
2	Decision Tree Classifier	0.420400	0.000000	0.386400	0.417800	0.392100	0.081400
3	SVM - Linear Kernel	0.373600	0.000000	0.344400	0.219100	0.233800	0.017400
4	Quadratic Discriminant Analysis	0.216500	0.000000	0.341900	0.453300	0.094400	0.009300

We found that the top three models are CatBoost Classifier, Gradient Boost Classifier and XGB Classifiers were giving the best accuracies.



# CatBoost Classifier



CatBoost is based on gradient boosting.

## Procedure of CatBoost Classifier

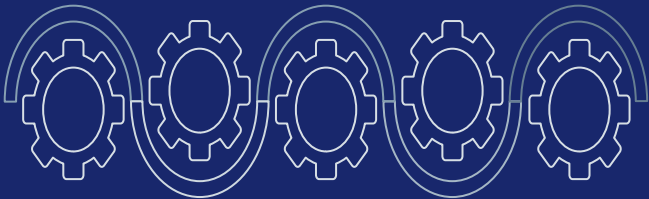
**Step 1:** Calculate residuals for each data point using a model that has been trained on all the other data points at that time. Hence we train different models to calculate residuals for different data points. In the end, we are calculating residuals for each data point that the corresponding model has never seen before.

**Step 2:** train the model by using the residuals of each data point as class labels.

**Step 3:** Repeat Step 1 & Step 2 (for n iterations).

## Limitations

- CatBoost does not support sparse matrices.
- When the dataset has many numerical features, CatBoost takes more time to train than Light GBM.



# Hyper parameter tuning using Grid Search CV

After using Randomised SearchCV we found that it was actually giving parameters which were not having any increase in the model performance, we use GridSearchCV which increased the accuracy and the f1 score to 0.5330, 0.5193 respectively.

## gridsearchCV

```
from sklearn.model_selection import GridSearchCV
from catboost import CatBoostClassifier

model = CatBoostClassifier(loss_function='MultiClass')

parameters = {'depth': [6,8,10], 'learning_rate': [0.01,0.05,0.10], 'iterations': [300,500]}
grid = GridSearchCV(estimator=model, param_grid = parameters, cv = 3)
grid.fit(X_train, y_train)

# Results from Grid Search

print("\n The best score across ALL searched params:\n",grid.best_score_)
print("\n The best parameters across ALL searched params:\n",grid.best_params_)

486:   learn: 0.9040090   total: 24.1s   remaining: 644ms
487:   learn: 0.9039205   total: 24.2s   remaining: 594ms
488:   learn: 0.9038592   total: 24.2s   remaining: 544ms
489:   learn: 0.9037678   total: 24.2s   remaining: 495ms
490:   learn: 0.9037149   total: 24.3s   remaining: 445ms
491:   learn: 0.9036607   total: 24.3s   remaining: 396ms
492:   learn: 0.9035798   total: 24.4s   remaining: 346ms
493:   learn: 0.9035377   total: 24.4s   remaining: 297ms
494:   learn: 0.9034591   total: 24.5s   remaining: 247ms
495:   learn: 0.9034319   total: 24.5s   remaining: 198ms
496:   learn: 0.9033642   total: 24.6s   remaining: 148ms
497:   learn: 0.9033153   total: 24.6s   remaining: 98.9ms
498:   learn: 0.9032279   total: 24.7s   remaining: 49.4ms
499:   learn: 0.9031683   total: 24.7s   remaining: 0us

The best score across ALL searched params:
0.5308044349986974

The best parameters across ALL searched params:
{'depth': 6, 'iterations': 500, 'learning_rate': 0.1}
```

## Final Model performance

Accuracy : **0.5330**



f1.Score : **0.5193**

Auc score ( averaged after one vs rest ) : 0.61271

The best accuracy achieved in the competition was 0.5399



# Business Application



# THANKS



## MR. JAYVEER NANDA

LEAD DATA SCIENTIST | DATA SCIENCE & BUSINESS  
ANALYTICS MENTOR | SUBJECT MATTER EXPERT |  
CONSULTANT | DS & AI ML TRAINER

Submitted By :-



Aakash Phadtare



Kshitij Saxena



Tejas Shrinivas Kulkarni



Pratik Waghmare