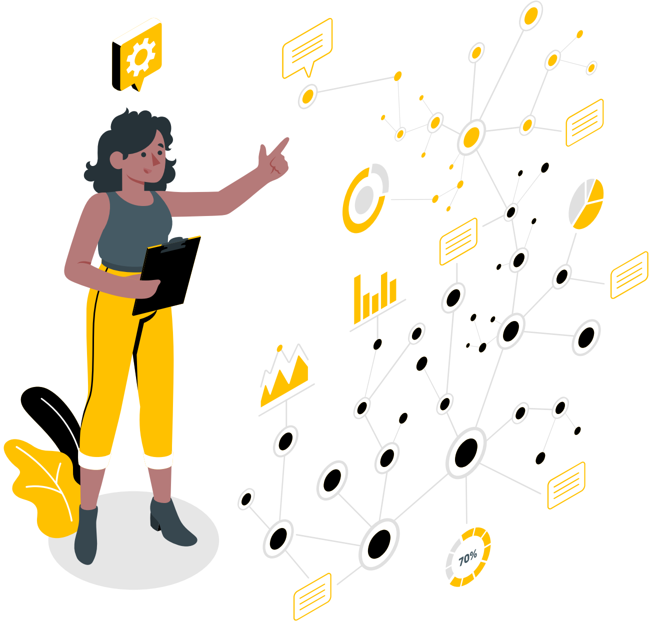
Table of Contents

Problem Statement

01



Context

Telemarketing campaigns for term payments

Type

Classification or Regression?

0 or 1

Class of Interest?

Dataset Description

02

	age	job	marital	education	default	balance	housing	loan	contact	day	month	duration	campaign	pdays	previous	poutcome	y
0	58	management	married	tertiary	no	2143	yes	no	unknown	5	may	261	1	-1	0	unknown	no
1	44	technician	single	secondary	no	29	yes	no	unknown	5	may	151	1	-1	0	unknown	no
2	33	entrepreneur	married	secondary	no	2	yes	yes	unknown	5	may	76	1	-1	0	unknown	no
3	47	blue-collar	married	unknown	no	1506	yes	no	unknown	5	may	92	1	-1	0	unknown	no
4	33	unknown	single	unknown	no	1	no	no	unknown	5	may	198	1	-1	0	unknown	no

ENTRIES

45,211

```
RangeIndex: 45211 entries, 0 to 45210
Data columns (total 17 columns):
 #   Column     Non-Null Count  Dtype
---  ------     --------------  -----
 0   age        45211 non-null  int64
 1   job        45211 non-null  object
 2   marital    45211 non-null  object
 3   education  45211 non-null  object
 4   default    45211 non-null  object
 5   balance    45211 non-null  int64
 6   housing    45211 non-null  object
 7   loan       45211 non-null  object
 8   contact    45211 non-null  object
 9   day        45211 non-null  int64
 10  month      45211 non-null  object
 11  duration   45211 non-null  int64
 12  campaign   45211 non-null  int64
 13  pdays      45211 non-null  int64
 14  previous   45211 non-null  int64
 15  poutcome   45211 non-null  object
 16  y          45211 non-null  object
dtypes: int64(7), object(10)
```

ATTRIBUTES

Categorical + Numerical

FEATURES

17

ENCODING DONE?

Yes, to handle categorical attributes

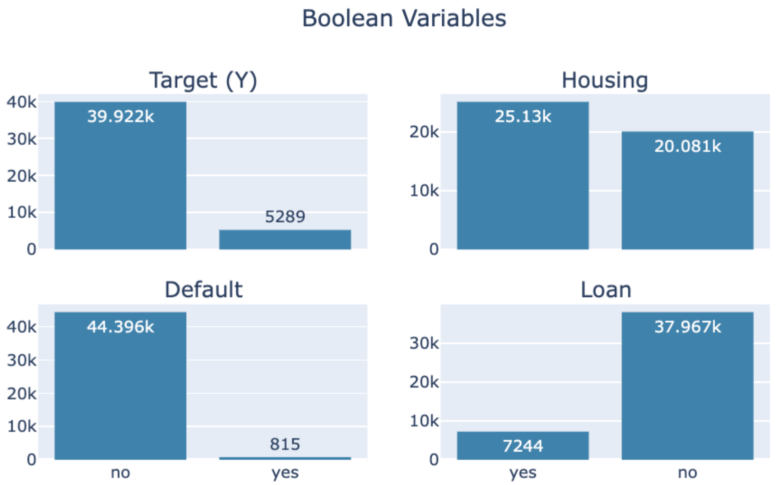
Pre-processing

- Null values
- Feature correlation
- Ranges
- Outcome variable
- Dummy variables
- Data distribution

Exploratory Data Analysis

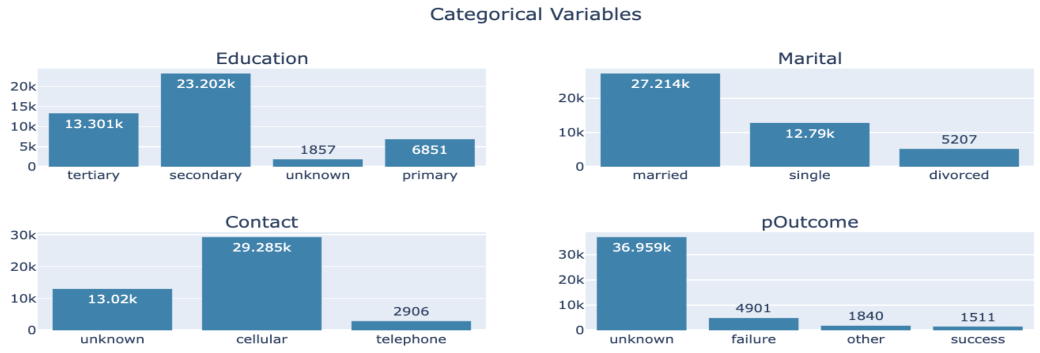
03

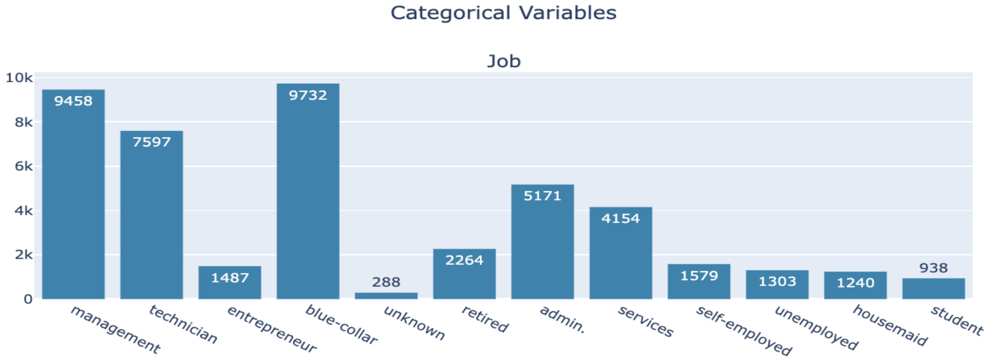
Distribution: Boolean Attributes



Count of categorical attributes

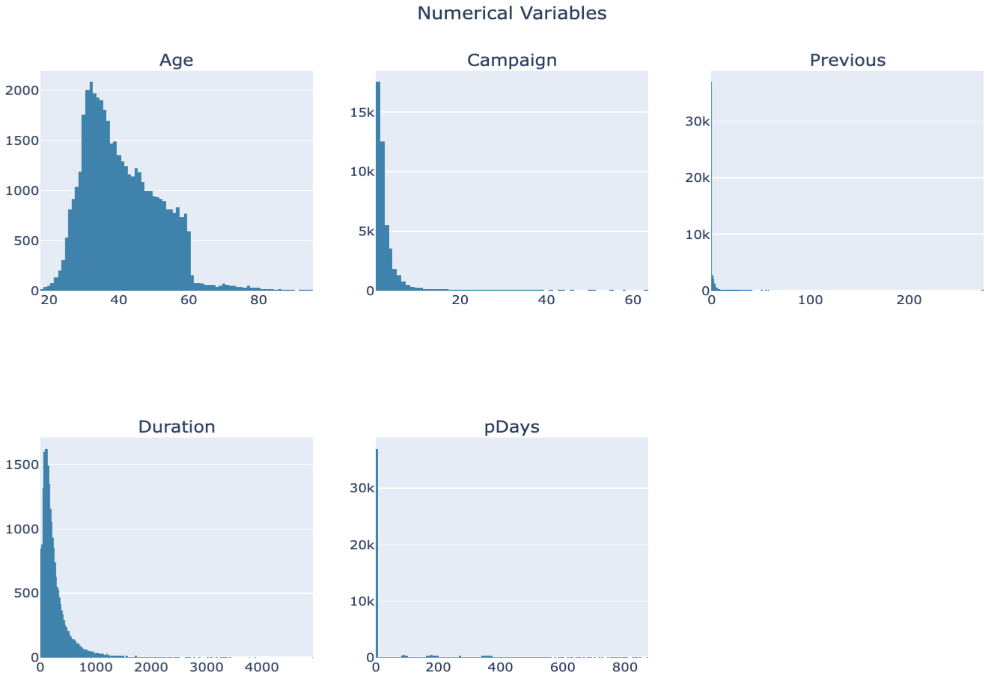
Distribution: Categorical Attributes





Categorical Variables

Distribution: Numerical Attributes



To check for skewness

Feature Engineering

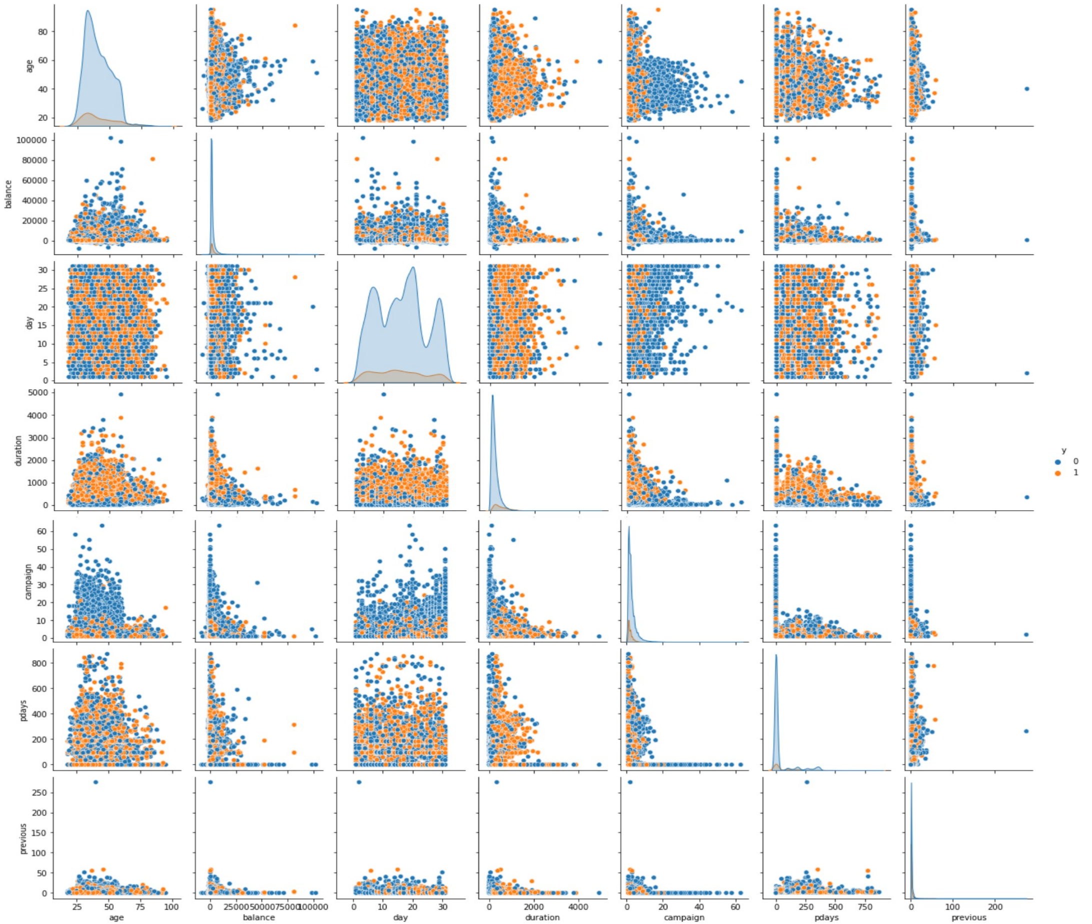
04

Heat Map:

To check for correlation of all the variables.

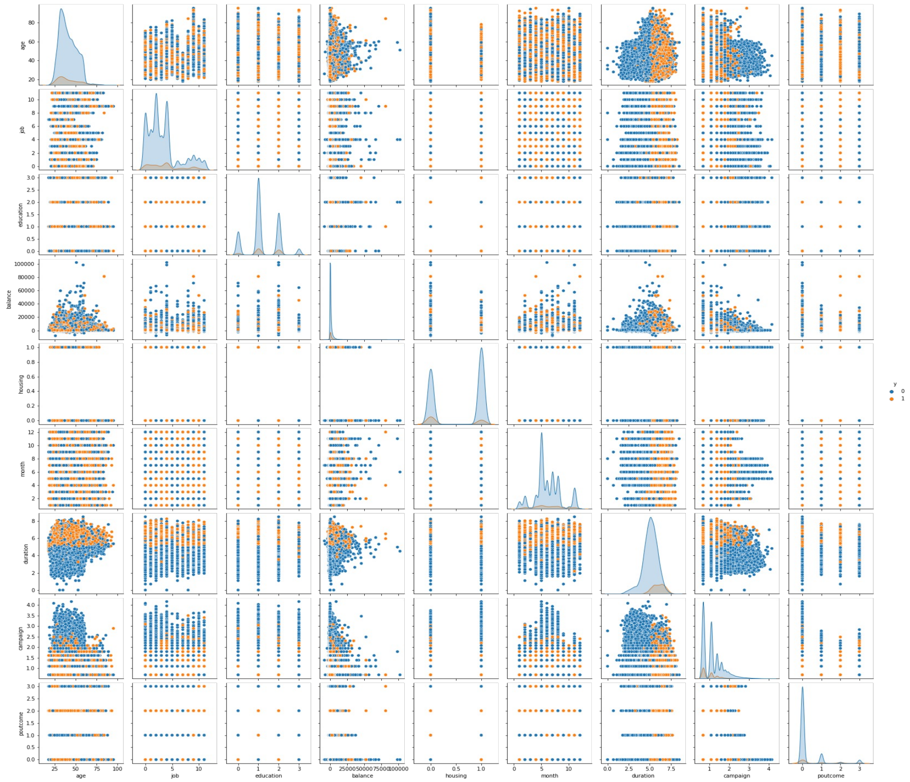
Pair-Plot:

Helps us to know that the data is non-linear. The image shows the columns before performing log transformation:



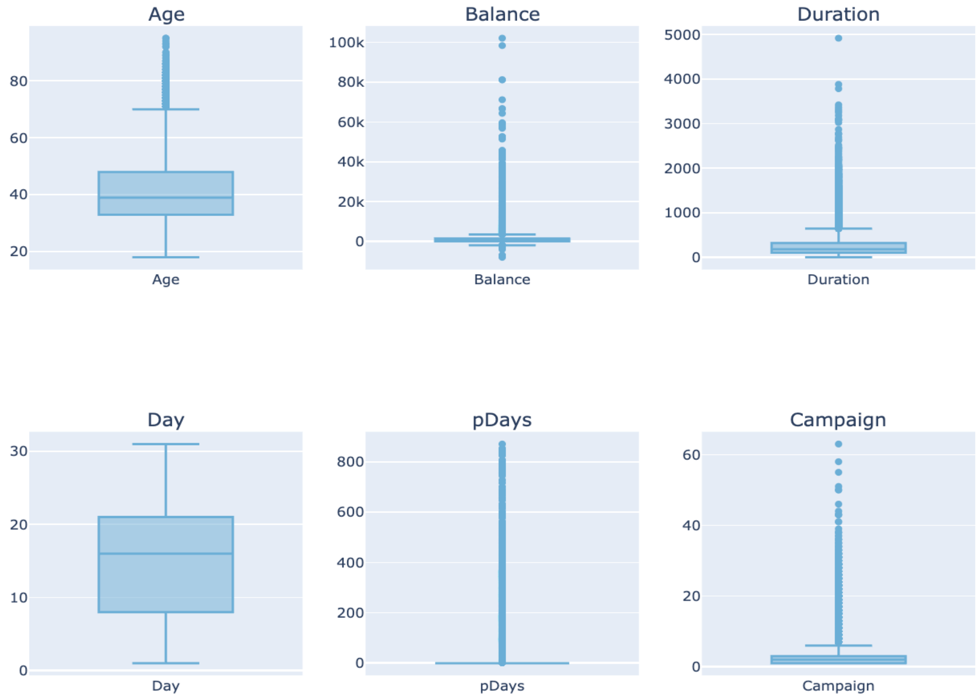
After Log Transformation

- The image besides depict the pair plot after log transformation:



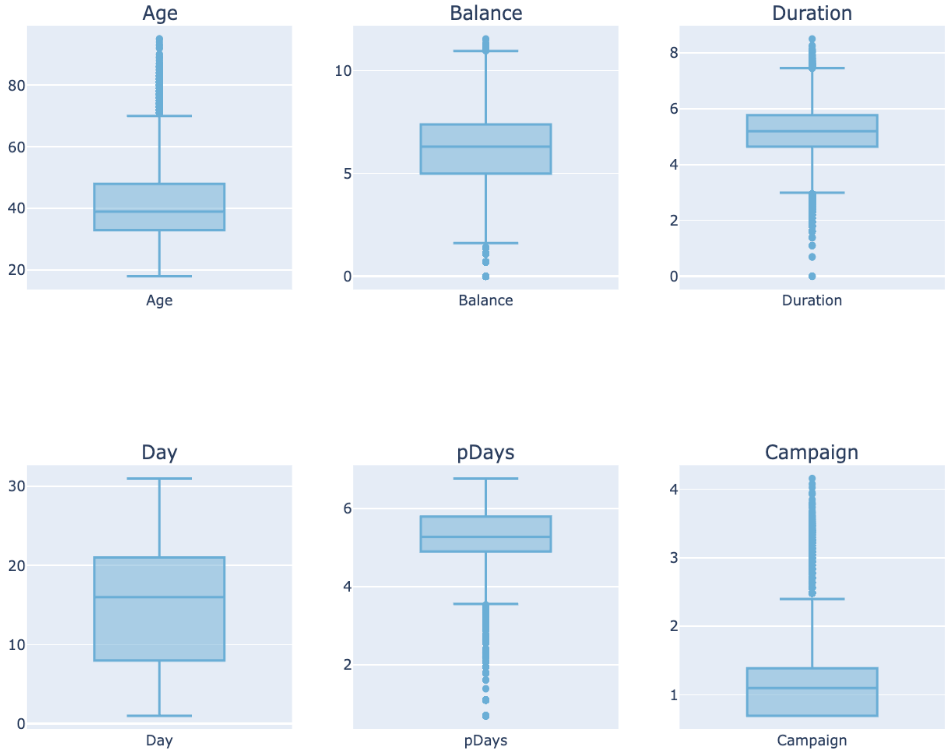
Before Outlier Distribution

Shows the outlier distribution of all numeric columns:



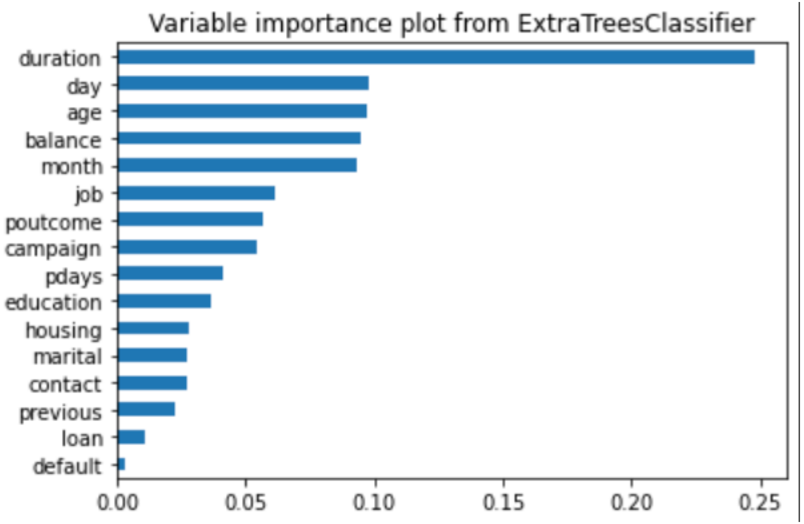
After Removing Outliers

Applied exponential function to reduce the skewness of our dataset.



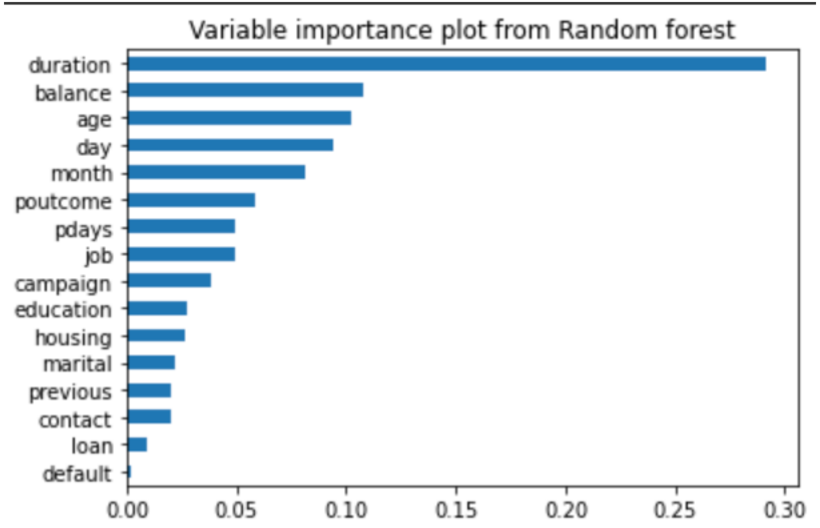
Feature Selection

Used ExtraTreesClassifier() to get variable importance.



Feature Selection

Used RandomForestClassifier() to get variable importance.



Standardization and Encoding

- Numerical columns were standardized and categorical variables were encoded.

Standardization:

$$z = \frac{x - \mu}{\sigma}$$

with mean:

$$\mu = \frac{1}{N} \sum_{i=1}^{N} (x_i)$$

and standard deviation

$$\sigma = \sqrt{\frac{1}{N} \sum_{i=1}^{N} (x_i - \mu)^2}$$

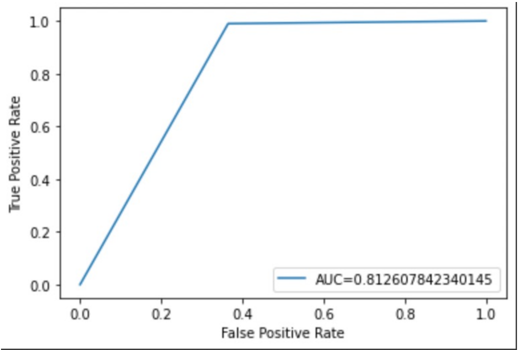
One-Hot Encoding

One-hot encoding was done for the following columns:

```
Index(['age', 'balance', 'duration', 'campaign', 'housing', 'job_blue-collar',
       'job_entrepreneur', 'job_housemaid', 'job_management', 'job_retired',
       'job_self-employed', 'job_services', 'job_student', 'job_technician',
       'job_unemployed', 'job_unknown', 'education_secondary',
       'education_tertiary', 'education_unknown', 'month_aug', 'month_dec',
       'month_feb', 'month_jan', 'month_jul', 'month_jun', 'month_mar',
       'month_may', 'month_nov', 'month_oct', 'month_sep', 'poutcome_other',
       'poutcome_success', 'poutcome_unknown', 'y'],
      dtype='object')
```

Model Implementation-1

Logistic Regression: Used this model as a benchmark against other classification models. We performed HyperParameter tuning on various learning rates and also did a K-fold cross-validation by taking k=5. The table displayed is an example of the average of the k-fold validation for one of the learning rates: 0.1 and the best AUC curve value has also been mentioned:



Average of K-fold

Data	Accuracy	Precision	Recall	f1_score	Misclassified
train	0.7016	0.4882	0.8984	0.6326	6664.8
test	0.6346	0.2140	0.9078	0.3136	3305.4

Model Implementation-2

Naïve Bayes: Implemented Gaussian Naïve Bayes on our dataset.

Performance Metrics:

1) Precision: 0.6512
2) Recall:0.7030
3) F1-score:0.5962

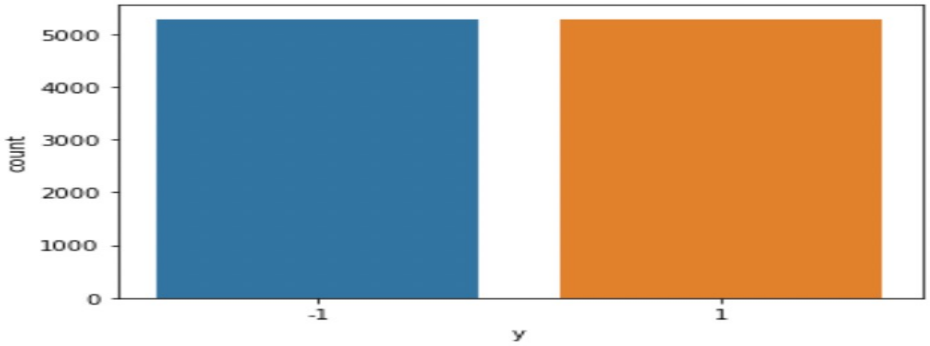
Prerequisite for SVM

Fixing Unbalanced Data:

Unbalanced Values:

```
df_temp = df["y"].unique()
df['y'].value_counts()
```

```
0    39922
1     5289
Name: y, dtype: int64
```



Model Implementation-3

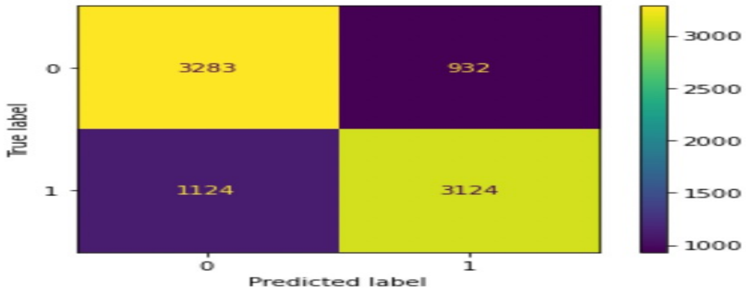
Soft SVM: Our dataset is Non-Linear, hence we decided to implement Soft SVM by making use of **Radial Basis Function** Kernel.

Note: Due to a large amount of dataset we have taken only 20% of our data for practical understanding.

Accuracy Score:
0.7570601441569184

Precision Score:
0.7702169625246549

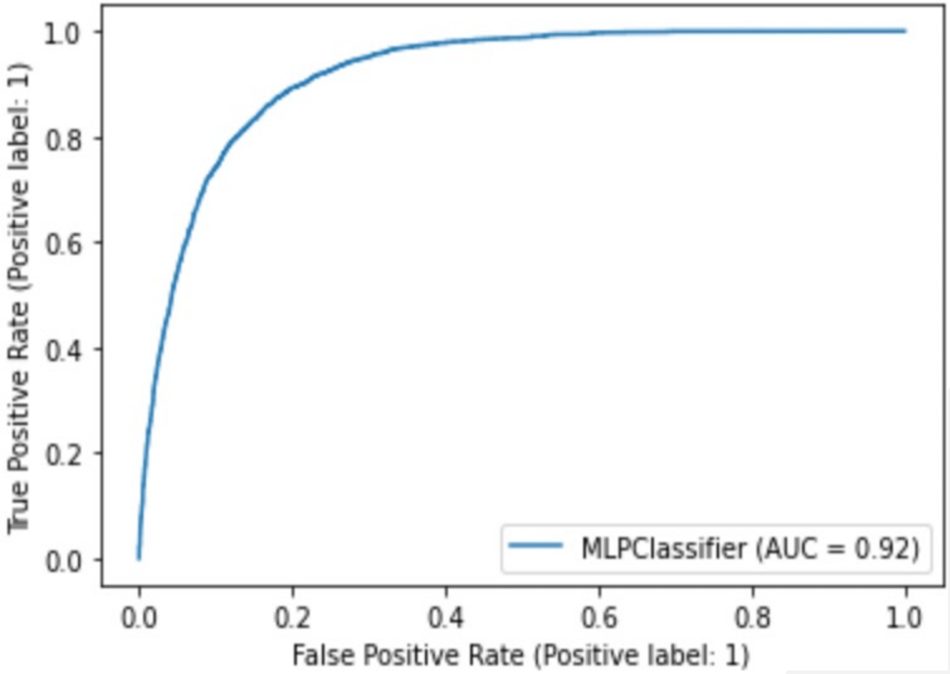
Confusion Matrix:



Model Implementation-4

Neural Network: Implemented Multi-Layer Perceptron on our dataset. It trains using Backpropagation. We oversampled our dataset and performed hyperparameter tuning on different learning rates and founc that the best recall value for our class of interest comes from the value->0.1.The AU curve for that has been mentioned:

AUC Curve



Model Comparison

In the next slide, we have given the best-performing parameters for all 4 models that we implemented for our dataset. We have considered f1-Score as our best performance metric

Conclusion

1. Logistic Regression: Learning Rate:0.0001,f1 score: 0.4166
2. Naïve Bayes:
3. SVM: f1-score: 0.757
4. Neural Network: Learning Rate:0.01,f1-score:0.79