Email Campaign Effectiveness Prediction

Shubhan Deshmukh Email –Id: - Shubhamdeshmukh278@gmail.com

Abstract:*Machine Learning (ML) is developing under the great promise that marketing can now be both more efficient and human. Cognitive systems, embedded or not into marketing software, are powering every single functional area of marketing and each step of the consumer journey. In order to help the business grow with the Email Marketing Strategies, we are trying to find all the features that are important for an Email to not get ignored. Many of the times we do not tend to read an Email due to a number of reasons. Some of it can be no proper structure of the email, too many direct links and images in a single email and may be too long emails. We are basically trying different machine learning algorithms like Decision Tree, KNN, Random Forest, XGBoost, etc.We will be using all models for prediction of effectiveness of email campaigns and compute the results and compare them for best accuracy and performance.*

Keywords: Decision Tree, KNN, Random Forest, XGBoost Model, SHAP

I. Problem Statement

Most of the small to medium business owners are making effective use of Gmail-based Email marketing Strategies for offline targeting of converting their prospective customers into leads so that they stay with them in Business.

II. Data Description

Email_ID - This column contains the email ids of individuals.

Email_type - Email type contains 2 categories 1 and 2. We can assume that the types are like promotional email or important email.

Subject_Hotness_Score - It is the email effectiveness score.

Email_Source - It represents the source of the email like sales or marketing or product type email.

Email_Campaign_Type - Campaign type

Total_Past_Communications - This column contains the previous mails from the same source.

Customer_Location - Categorical data which explains the different demographics of the customers.

Time_Email_sent_Category - It has 3 categories 1, 2 and 3 which may give us morning, evening and night time slots.

Word_Count - It contains the number of words contained in the mail.

Total_Links - Total links from the mail.

Total_Images - The banner images from the promotional email.

Email_Status - It is the target variable which contains the characterization of the mail that is ignored; read; acknowledged by the reader.

III. Introduction

Nowadays, digital advertising strategies aim to engage customers over multiple touch points where highly personalised content and messages are delivered. Direct email marketing represents a crucial moment along the customer journey where a sequence of optimised messages could influence customer decisions. Not only this optimisation process may lead to revenue growth but it also promotes a fruitful customer engagement and satisfaction over a long time period. In this paper, we propose a machine learning based approach to characterize the mail and track the mail that is ignored; read; acknowledged by the reader.

IV. Exploratory Data Analysis

A) Data Cleaning :-

First, we will rename the columns of each file. Because the name of the column contains space, and uppercase letters so we will correct it to make it easy to use. Here we simplified the column names.

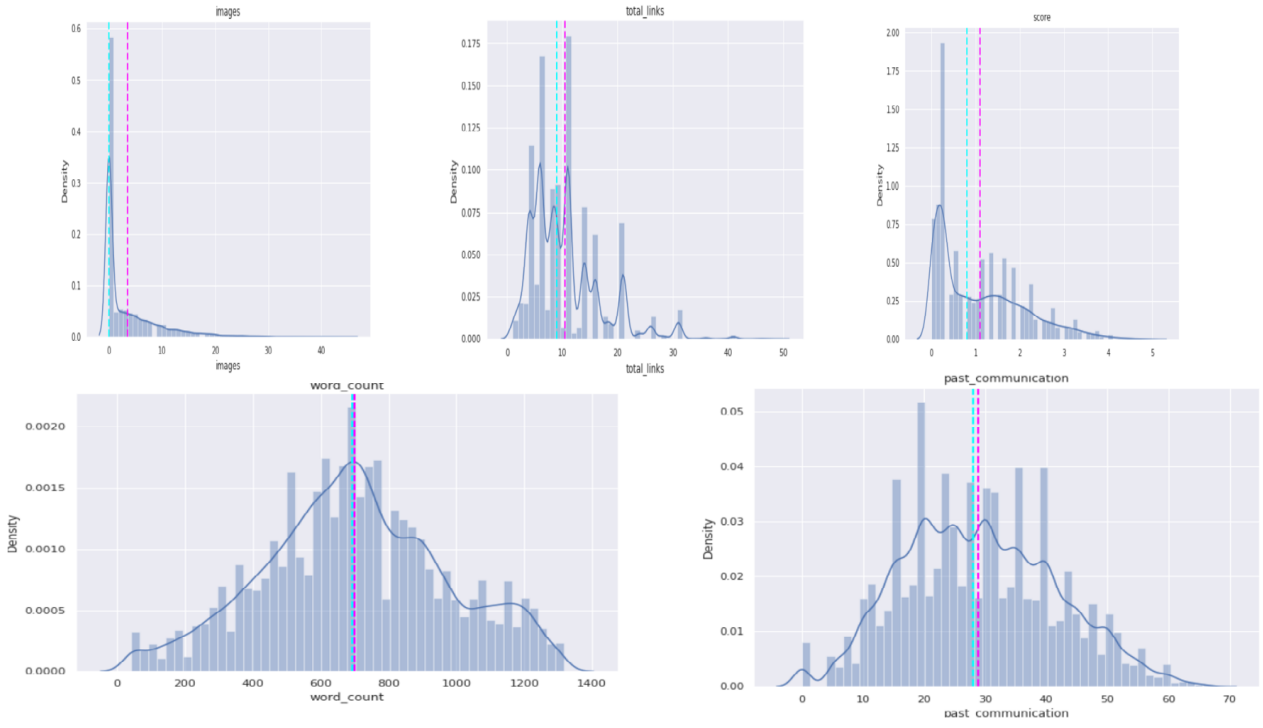
B) Duplication :-

In our further analysis we found that the dataset has no duplicate entries.

C) Data Visualization : -

a) Handling numerical Features

I. Univariate Analysis – In the dataset some feature's histogram plots with skewed distribution and some are symmetrically distributed.



The conclusion from above histograms

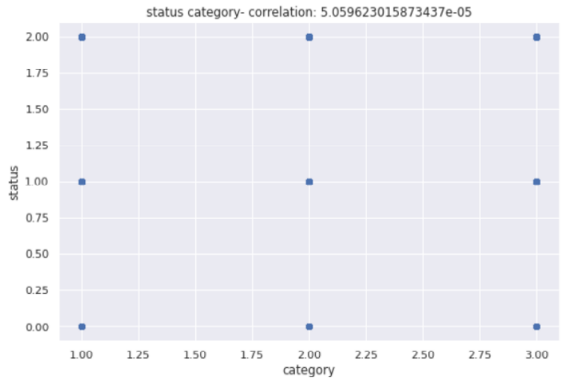
1. The features for distribution is symmetric -

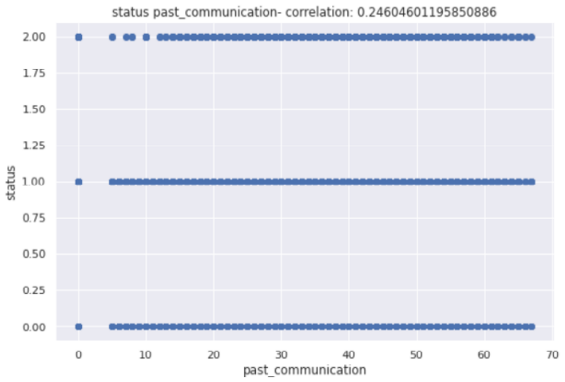
1) Word Count
2) Past Communication

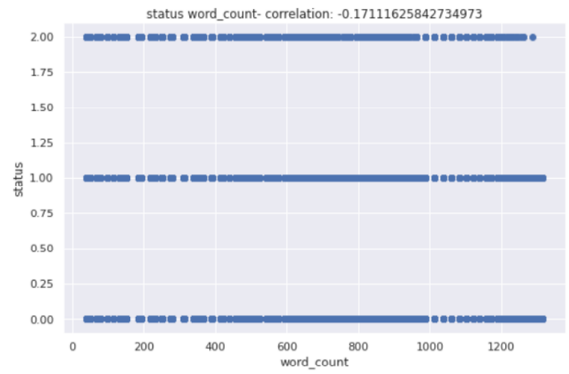
2. The features for distribution is skewed -

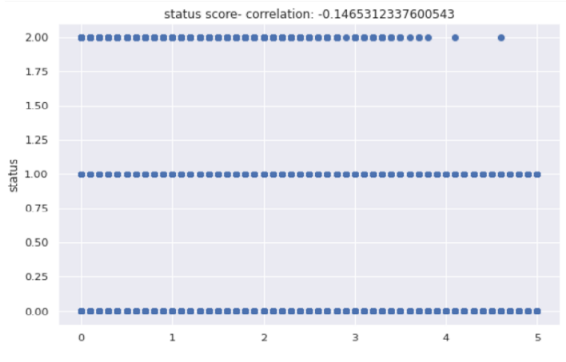
1) Images
2) Total links
3) Score

a) Bivariate Analysis – In bivariate analysis we consider status features as dependent variables and other features as independent features. Below is some scatter plot which showed high positive and negative correlation.









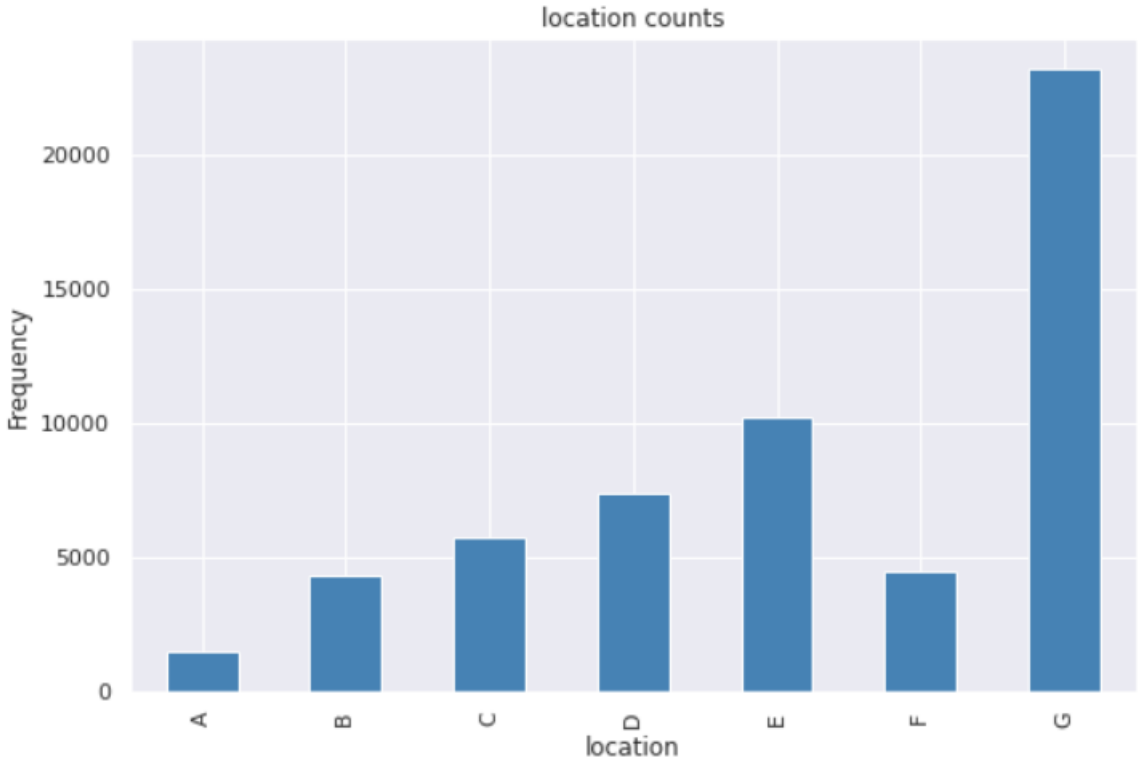
Conclusion from bivariate analysis

1) Status feature is positively correlated with features past communication and category.

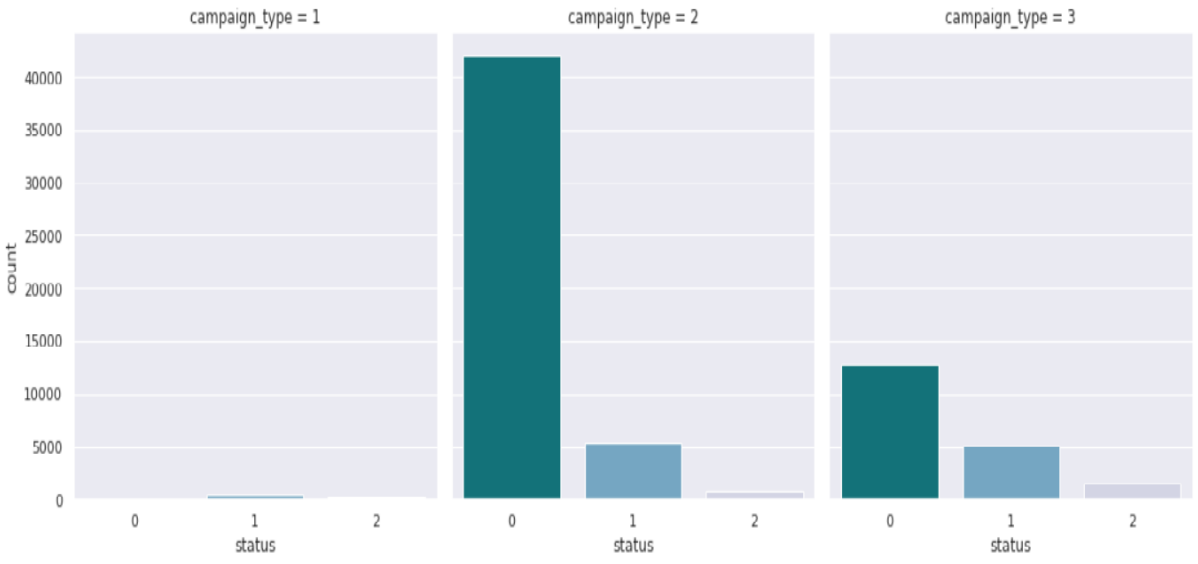
2) Status feature is negatively correlated with features word count and score.

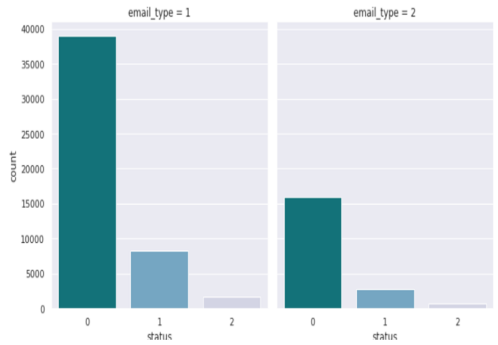
B) Handling Categorical Features –

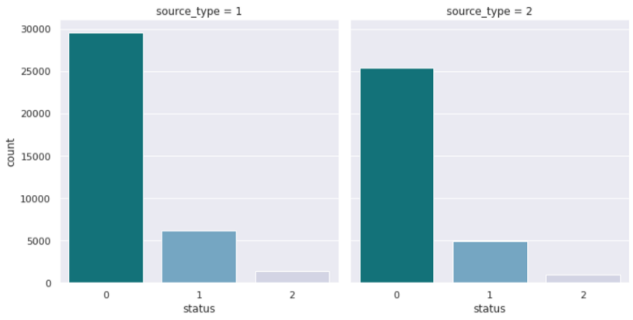
I. Univariate Analysis – We have two categorical features: Id and location. Below plot gives information about the location of users. From this plot we conclude that the location G is the highest number.

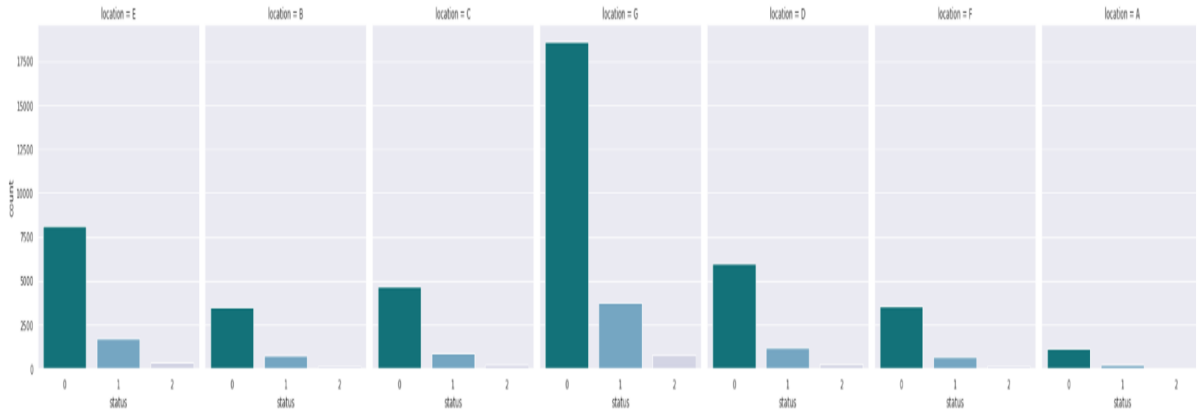


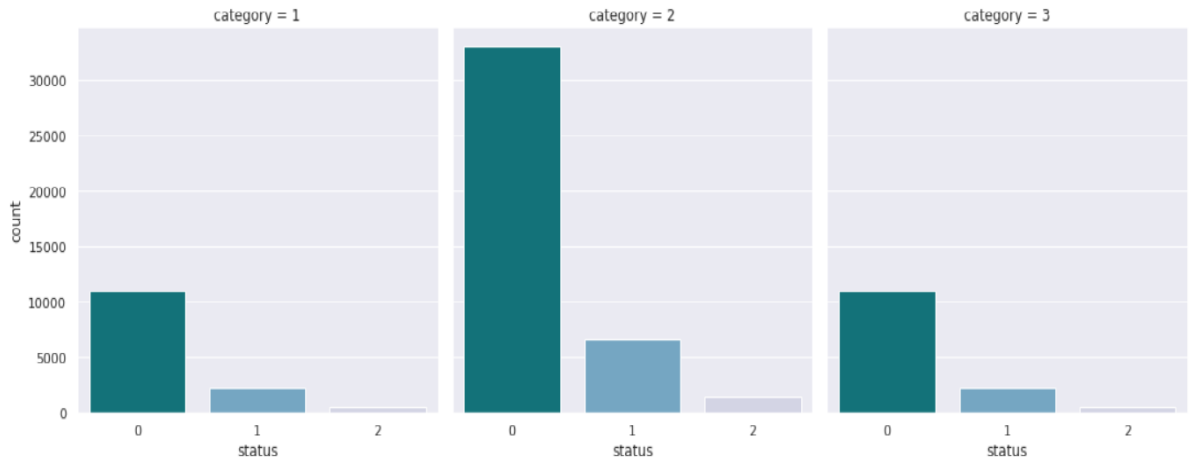
C) Relationship between independent variable and dependent variable.











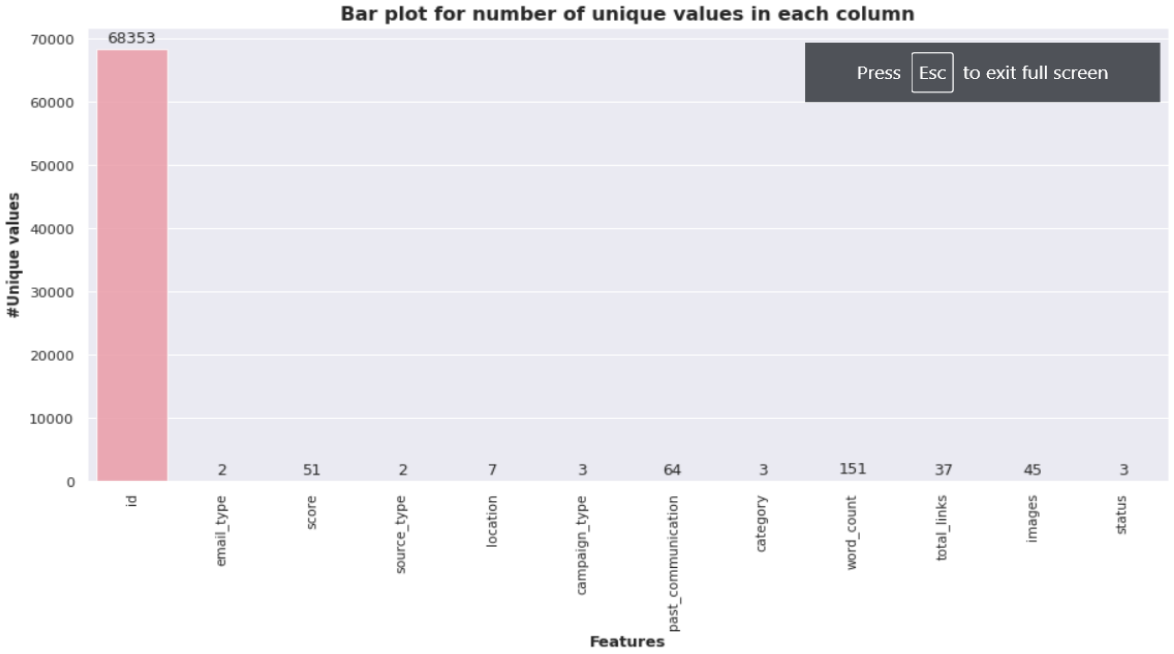
Conclusion from above graphs.1) Campaign type 2 is highest in number and status 0 high in each category.

2) In our dataset we have two types of emails in datasets where email type 1 is highest in number.

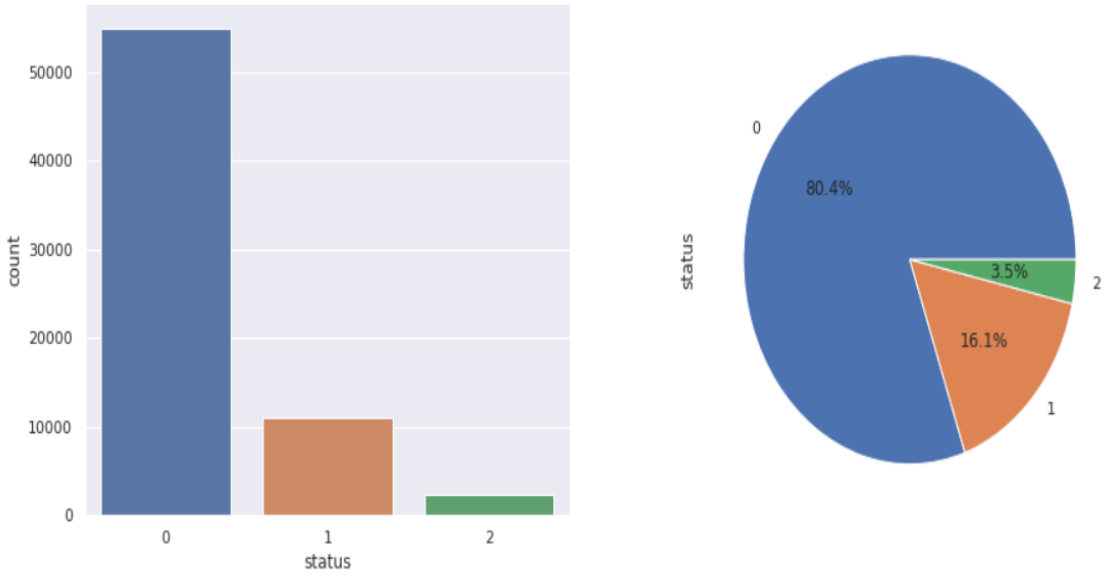
3) The location G has the highest number of users and in each location status 0 is high in number.

4)Category 1 and category 2 have approximately similar counts and category 2 has more counts as compared to both of them along with status 0 is high in number.

E) Unique values in each feature From the plot below we can see unique features in each column.



F) Dependent variable Analysis



We have multiclass dependent variable and from pie chart we can see status has 80.4% data so we can conclude our data is imbalanced.

G) Missing Data –

Once you have raw data, you must deal with issues such as missing data and ensure that the data is prepared for forecasting models in such a way that it is amenable to them.

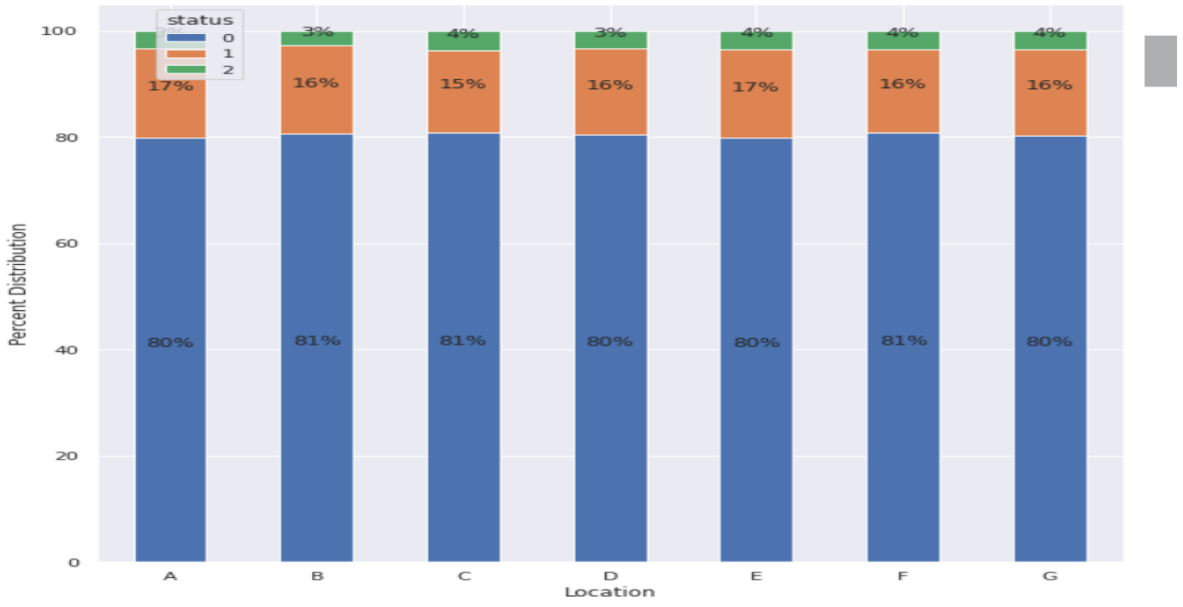
There were missing values in some columns in our dataset. The following code snippet gave us the idea about missing values in different columns.

	No. of Missing Data	% Missing Data
id	0.0	0.000000
email_type	0.0	0.000000
score	0.0	0.000000
source_type	0.0	0.000000
location	11595.0	16.963411
campaign_type	0.0	0.000000
past_communication	6825.0	9.984931
category	0.0	0.000000
word_count	0.0	0.000000
total_links	2201.0	3.220049
images	1677.0	2.453440
status	0.0	0.000000

The feature location, total links, images, past communication features have missing values.

Handling the missing data –

We can see in the above figure we have 11595 missing values in the location feature so let's visualize the location feature.



The frequency graph of different values of Customer location with respect to the Email status category we found that the percentage ratio of Email being Ignored, Read or Acknowledged is the same irrespective of the location .So we will drop the location column.

We saw in Univariate analysis the past communication feature is distributed symmetrical so we will substitute the null values with mean value.

```
#replacing the missing values in past communication feature
print('Number of missing values before imputing is = ',df['past_communication'].isnull().sum())
df['past_communication'].fillna(df['past_communication'].mean(),inplace=True)
print('Number of missing values after imputing is = ',df['past_communication'].isnull().sum())
```

```
Number of missing values before imputing is =  6825
Number of missing values after imputing is =  0
```

In Univariate analysis the total images and links have a right skewed distribution so we will substitute the null values with mode value.

```
#Filling missing values of Total_Links column
print('Number of missing values before imputing is = ',df['total_links'].isnull().sum())
df['total_links'].fillna(df['total_links'].mode()[0],inplace=True)
print('Number of missing values after imputing is = ',df['total_links'].isnull().sum())
#Filling missing values of images column
print('Number of missing values before imputing is = ',df['images'].isnull().sum())
df['images'].fillna(df['images'].mode()[0],inplace=True)
print('Number of missing values after imputing is = ',df['images'].isnull().sum())
```

```
Number of missing values before imputing is =  2201
Number of missing values after imputing is =  0
Number of missing values before imputing is =  1677
Number of missing values after imputing is =  0
```

H) Outliers

An outlier is an observation that lies an abnormal distance from other values in a random sample from a population. In a sense, this definition leaves it up to the analyst (or a consensus process) to decide what will be considered abnormal. Before abnormal observations can be singled out, it is necessary to characterize normal observations. A point beyond an inner fence on either side is considered a mild outlier. A point beyond an outer fence is considered an extreme outlier.

We separated the continuous variable and discrete variables.

Discrete Variables

Email type values: [1 2]

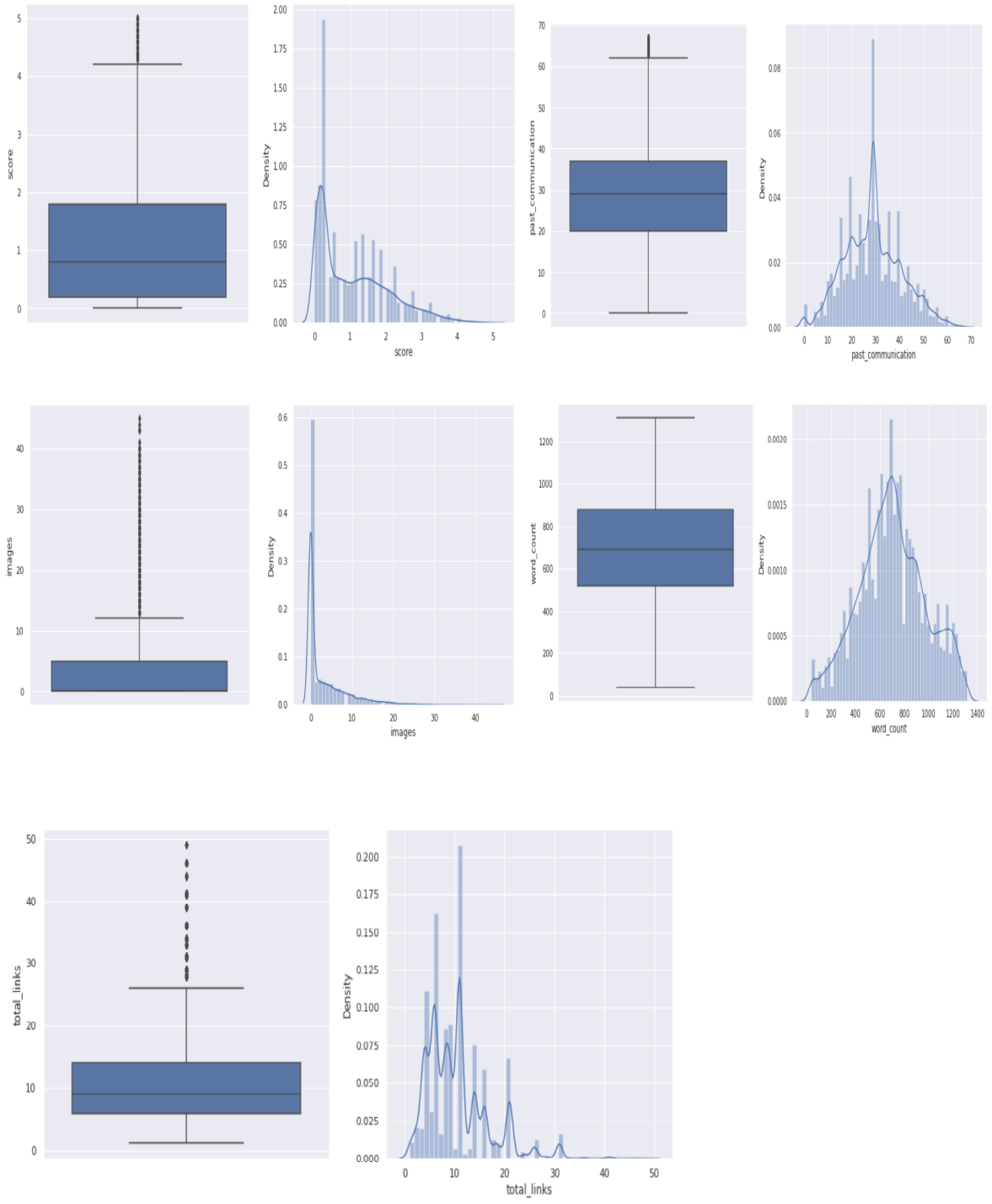
Source type values: [2 1]

Campaign type values: [2 3 1]

Category values: [1 2 3]

Status values: [0 1 2]

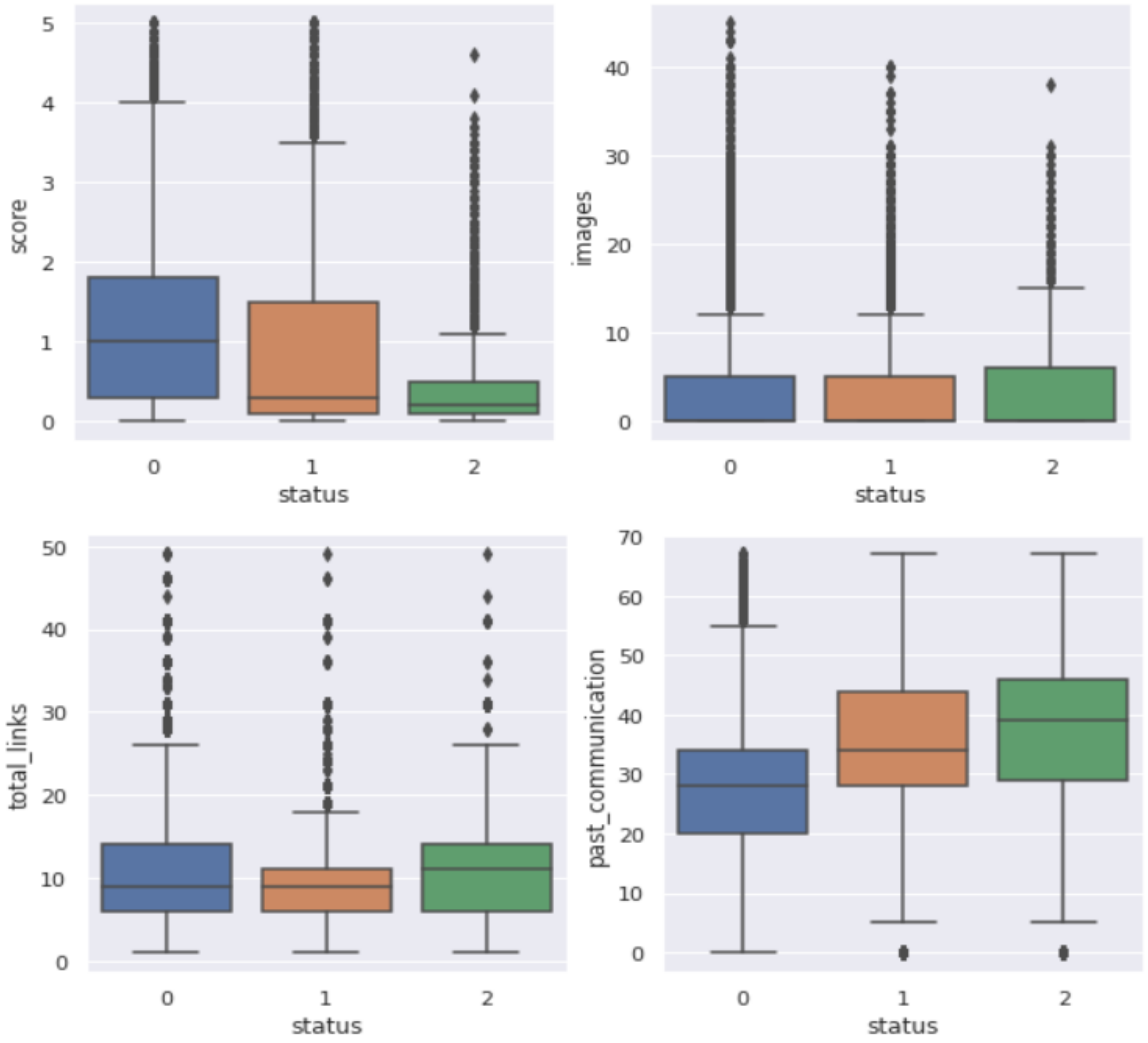
Continuous variables and Outliers



From above boxplot we can see features with null values are as follows:-

1. Score
2. Past communication
3. total links
4. total images
5. Word count

Deep dive into these feature's outliers.



As we know our dataset is imbalanced so removing outliers will affect the minority class so we will keep the outliers and we will use machine learning models which are robust to outliers.

I) Categorical Encoding and Standardization –

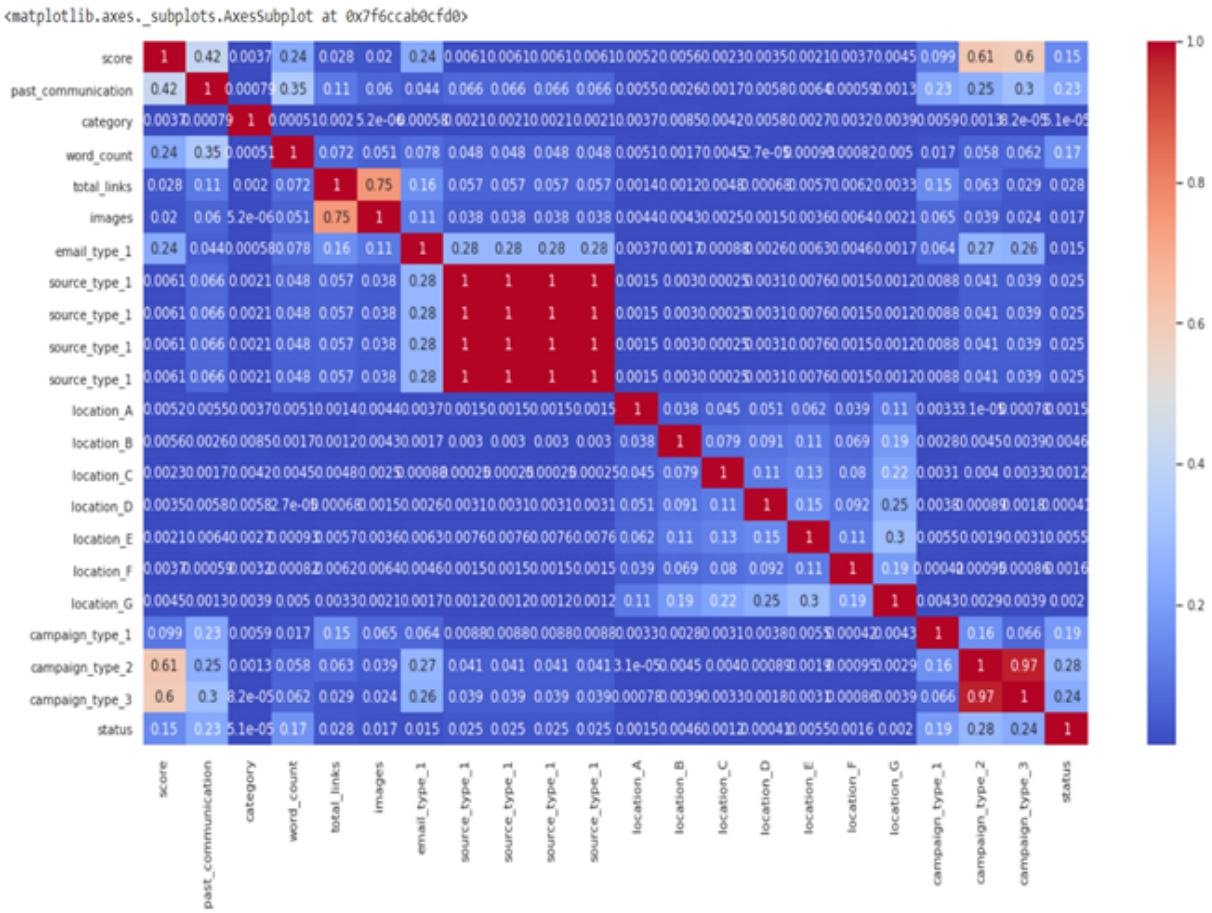
Standardization is another scaling technique where the values are centred on the mean with a unit standard deviation. This means that the mean of the attribute becomes zero and the resultant distribution has a unit standard deviation.

The standard score of a sample x is calculated as:

$z = (x - u) / s$

If scaling is not done then the algorithm only takes magnitude into account and not units hence incorrect modelling. To solve this issue, we have to do scaling to bring all the variables to the same level of magnitude.

Correlation Understanding:



VIF Factor:

Variance inflation factor (*VIF*) quantifies how much the variance is inflated. A VIF of 1 means that there is no correlation among the jth predictor and the remaining predictor variables, and hence the variance of bj is not inflated at all. The general rule of thumb is that VIFs exceeding 4 warrant further investigation, while VIFs exceeding 10 are signs of serious multicollinearity requiring correction. Below, we can observe that all the VIF values are within appreciable limits. Hence we can utilize these variables for the modelling.

	variables	VIF
0	score	2.411241
1	past_communication	1.939716
2	word_count	1.526551
3	email_type_1	1.121603
4	location_A	1.101572
5	location_B	1.287216
6	location_C	1.370743
7	location_D	1.461264
8	location_E	1.599116
9	location_F	1.292755
10	location_G	1.982246
11	campaign_type_1	1.261807
12	campaign_type_2	6.949802
13	campaign_type_3	3.346306

J) Feature Selection –

Tree-based: SelectFromModel

SelectFromModel is an embedded method. Embedded methods use algorithms that have built-in feature selection methods.

Here, we have used Random Forest to select features based on feature importance. We calculate feature importance using node impurities in each decision tree. In Random forest, the final feature importance is the average of all decision tree feature importance.

```
from sklearn.feature_selection import SelectFromModel
from sklearn.ensemble import RandomForestClassifier
# define SelectFromModel feature selection method
embeded_rf_selector = SelectFromModel(RandomForestClassifier(n_estimators=40), max_features=18,threshold=0.04)
embeded_rf_selector.fit(X, y)

embeded_rf_support = embeded_rf_selector.get_support()
embeded_rf_feature = X.loc[:,embeded_rf_support].columns.tolist()
print(str(len(embeded_rf_feature)), 'selected features')
```

```
6 selected features
```

```
#Important features list
embeded_rf_feature
```

```
['score',
 'past_communication',
 'category',
 'word_count',
 'total_links',
 'images']
```

Using these six features we will create a new dataframe for modelling purposes.

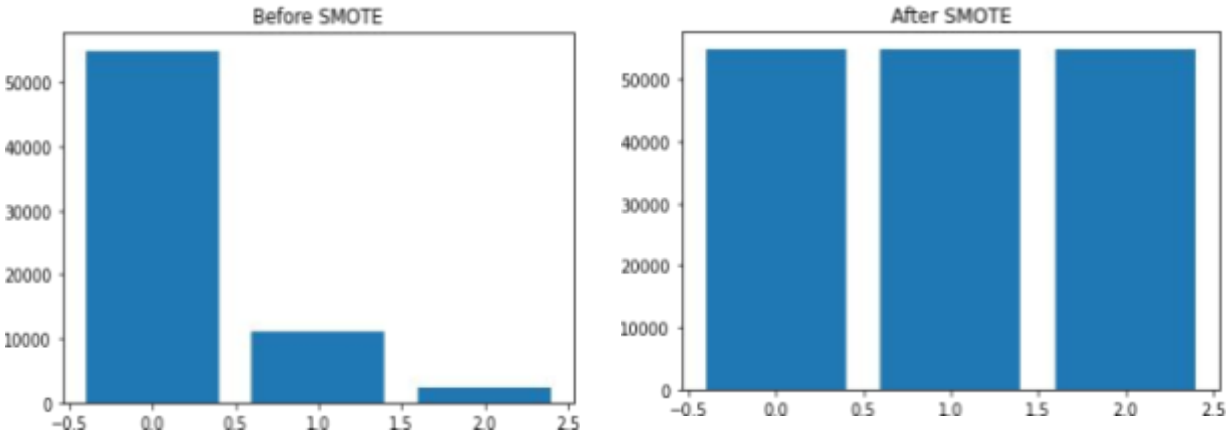
K) Handling imbalanced data

One of the major issues when dealing with unbalanced datasets relates to the metrics used to evaluate their model. Using simpler metrics like accuracy score can be misleading. In a dataset with highly unbalanced classes, the classifier will always “predict” the most common class without performing any analysis of the features and it will have a high accuracy rate, obviously not the correct one**.**

Synthetic Minority Oversampling Technique (SMOTE)

This technique generates synthetic data for the minority class.

SMOTE (Synthetic Minority Oversampling Technique) works by randomly picking a point from the minority class and computing the k-nearest neighbours for this point. The synthetic points are added between the chosen point and its neighbours**.**



Created balanced data with 54941 records for each class.

Splitting dataset into train-test

```
[ ] from sklearn.model_selection import train_test_split
    #Split data into train and test
    X_train, X_test, y_train, y_test = train_test_split(x_smote, y_smote, test_size = 0.2, random_state = 3,stratify= y_smote)
```

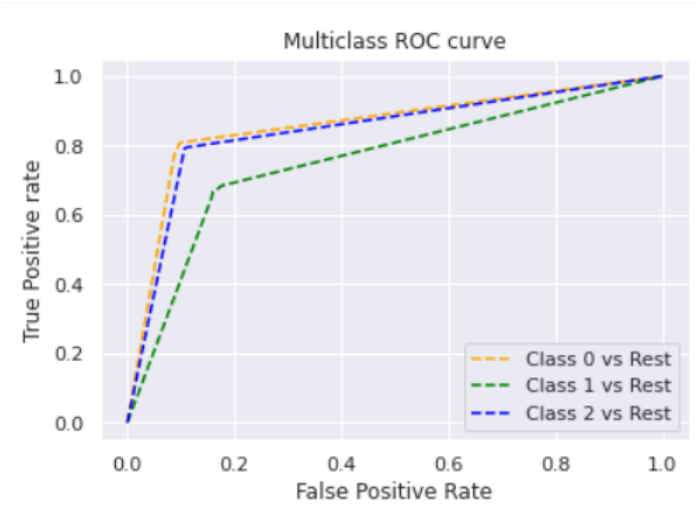
Here we splitted our data into train-test in 80:20 ratio respectively.

V. Modelling

A) Decision Tree Classifier –

A decision tree is a graphical representation of all possible solutions to a decision based on certain conditions. On each step or node of a decision tree, used for classification, we try to form a condition on the features to separate all the labels or classes contained in the dataset to the fullest purity.

F1_Score- Train Set: 0.99, Test_Set:0.76

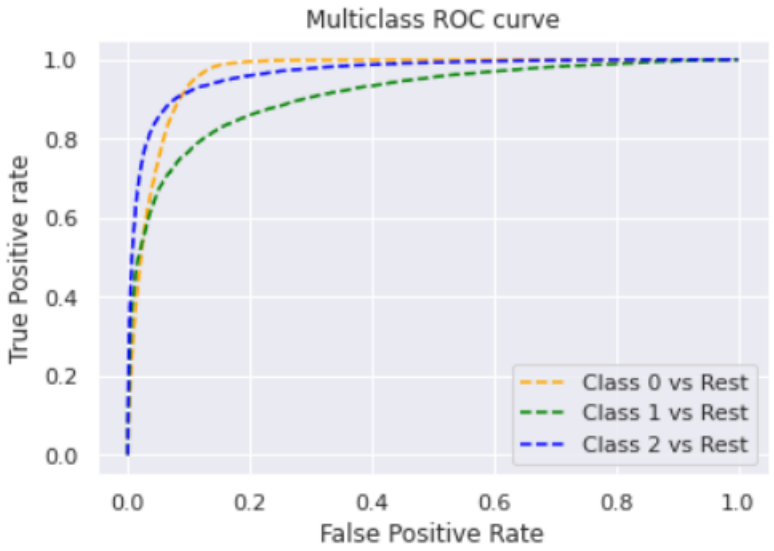


B) XGBoost Model –

XGBoost or extreme gradient boosting is one of the well-known gradient boosting techniques (ensemble) having enhanced performance and speed in tree-based (sequential decision trees) machine learning algorithms.

Parameters used - (n_estimators=100, max_depth=25, min_samples_leaf=20, min_samples_split=30)

F1_Score- Train Set: 0.98, Test_Set:0.84

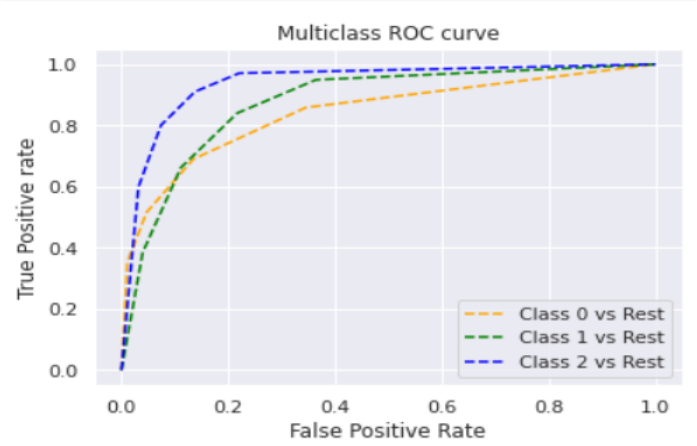


C) KNN -:

The k-nearest neighbors (KNN) algorithm is a simple, supervised machine learning algorithm that can be used to solve both classification and regression problems. It's easy to implement and understand, but has a major drawback of becoming significantly slower as the size of that data in use grows.

Parameters Used - (n_neighbors = 4, metric = 'minkowski', p = 2)

F1_Score- Train_Set:0.85, Test_Set:0.75

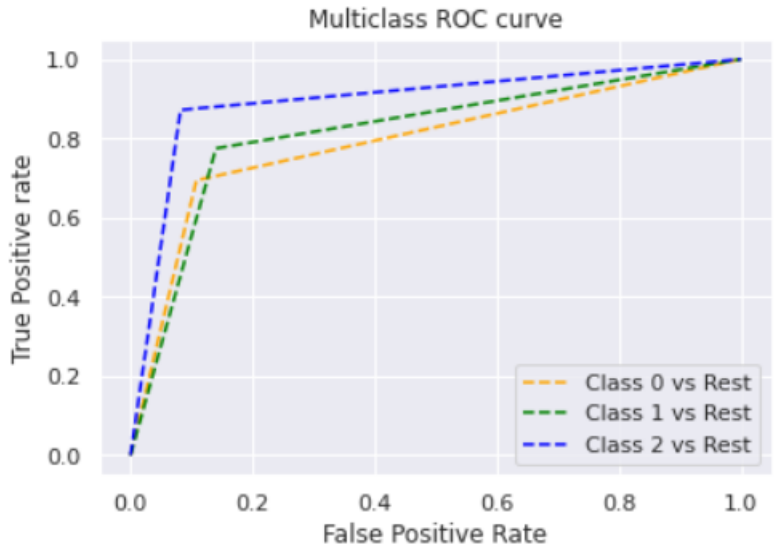


D) RandomizedsearchCV KNN –

Random search is a technique where random combinations of the hyper parameters are used to find the best solution for the built model. It is similar to grid search, and yet it has proven to yield better results comparatively.

Parameters used – 'n_neighbors':np.arange (1, 5)

F1_Score- Train_Set:0.98, Test_Set:0.78

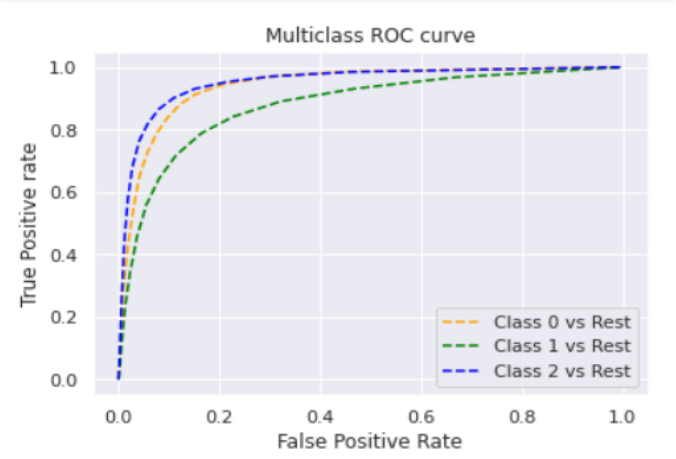


E) Random Forest –

Random forest is a supervised learning algorithm. The "forest" it builds is an ensemble of decision trees, usually trained with the "bagging" method. The general idea of the bagging method is that a combination of learning models increases the overall result.

Parameters Used - (n_estimators = 12, criterion = 'entropy', random_state = 42)

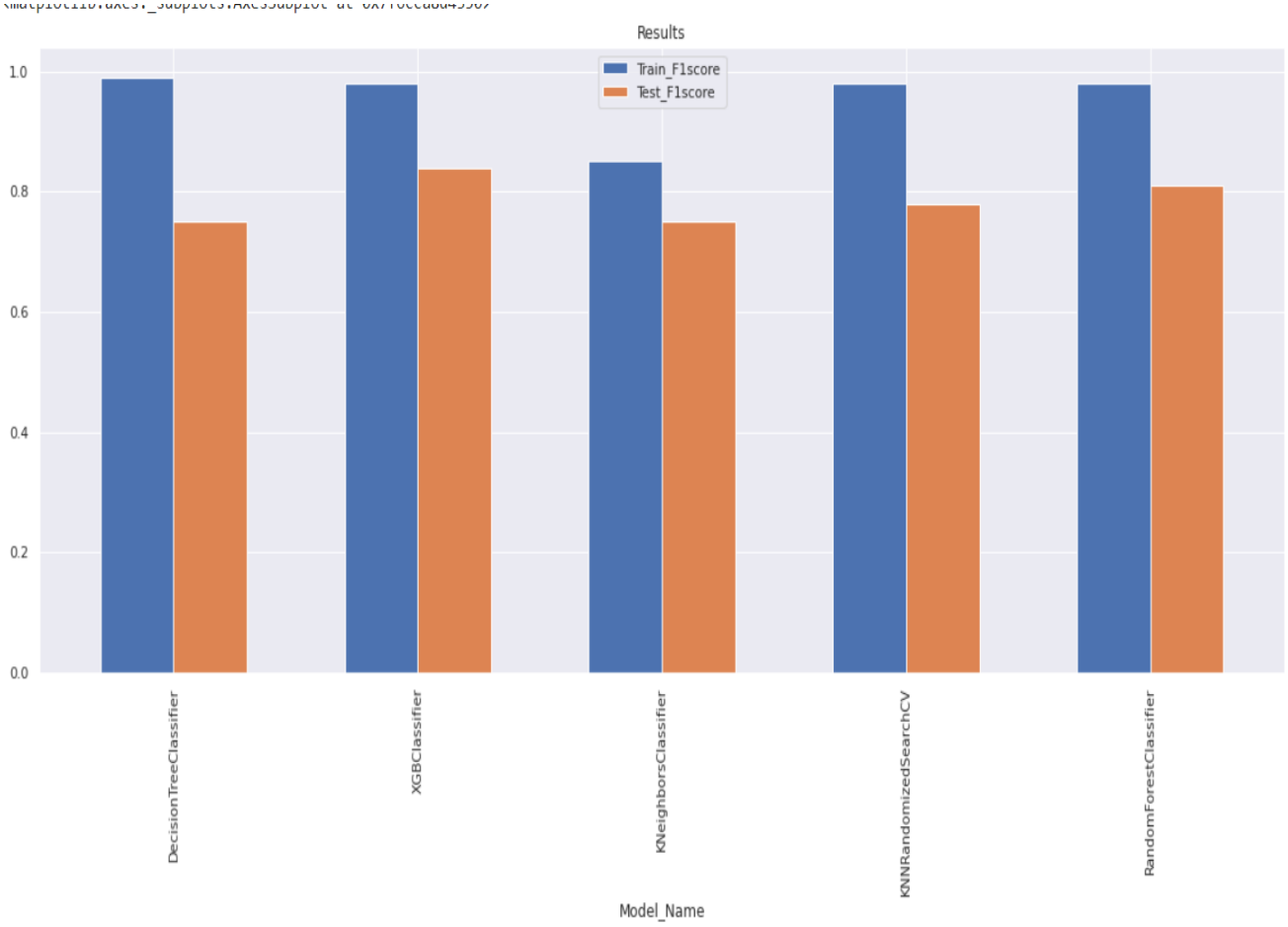
F1_Score- Train_Set:0.98, Test_Set:0.81



VI. Evaluating all the models

After modelling we will compare the results of each model.

	Model_Name	Recall_Train	Recall_Test	Precision_Train	Precision_Test	Train_F1score	Test_F1score	Train_AUC	Test_Auc	Train_accuracy	Test_accuracy
0	DecisionTreeClassifier	0.99	0.75	0.99	0.75	0.99	0.75	1.00	0.81	0.99	0.75
1	XGBClassifier	0.98	0.84	0.98	0.84	0.98	0.84	1.00	0.95	0.98	0.84
2	KNeighborsClassifier	0.85	0.75	0.85	0.75	0.85	0.75	0.97	0.89	0.85	0.75
3	KNNRandomizedSearchCV	0.98	0.78	0.98	0.78	0.98	0.78	0.99	0.84	0.98	0.78
4	RandomForestClassifier	0.98	0.81	0.98	0.81	0.98	0.81	1.00	0.92	0.98	0.81



From the above plots we can observe that XG-Boost performed well as compared to all the other models both for train as well as test.

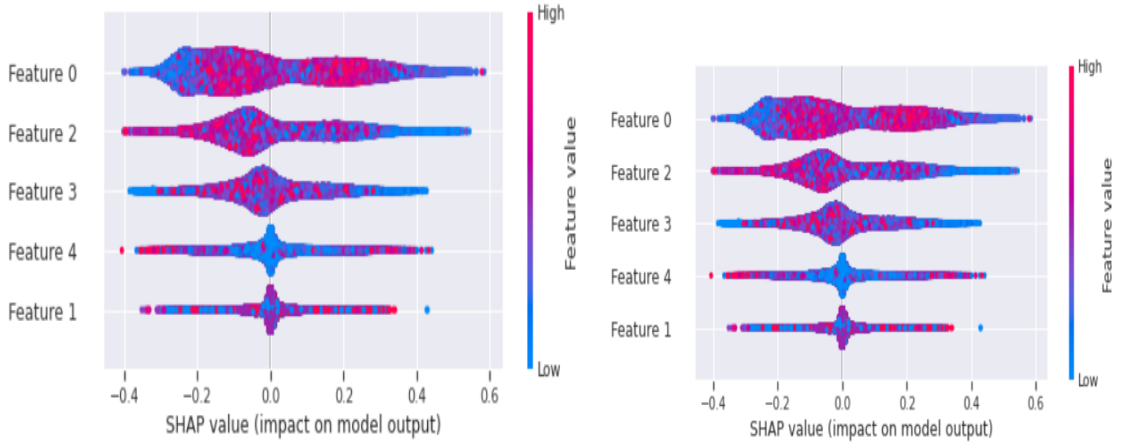
VII.SHAPE

SHAP (Shapley Additive Explanations) by Lundberg and Lee (2016) is a method to explain individual predictions, based on the game theoretically optimal Shapley values. Shapley values are a widely used approach from cooperative game theory that come with desirable properties. The feature values of a data instance act as players in a coalition. The Shapley value is the average marginal contribution of a feature value across all possible coalitions.

SHAP (SHapley Additive exPlanations) is a game theoretic approach to explain the output of any machine learning model. It connects optimal credit allocation with local explanations using the classic Shapley values from game theory and their related extensions (see papers for details and citations).

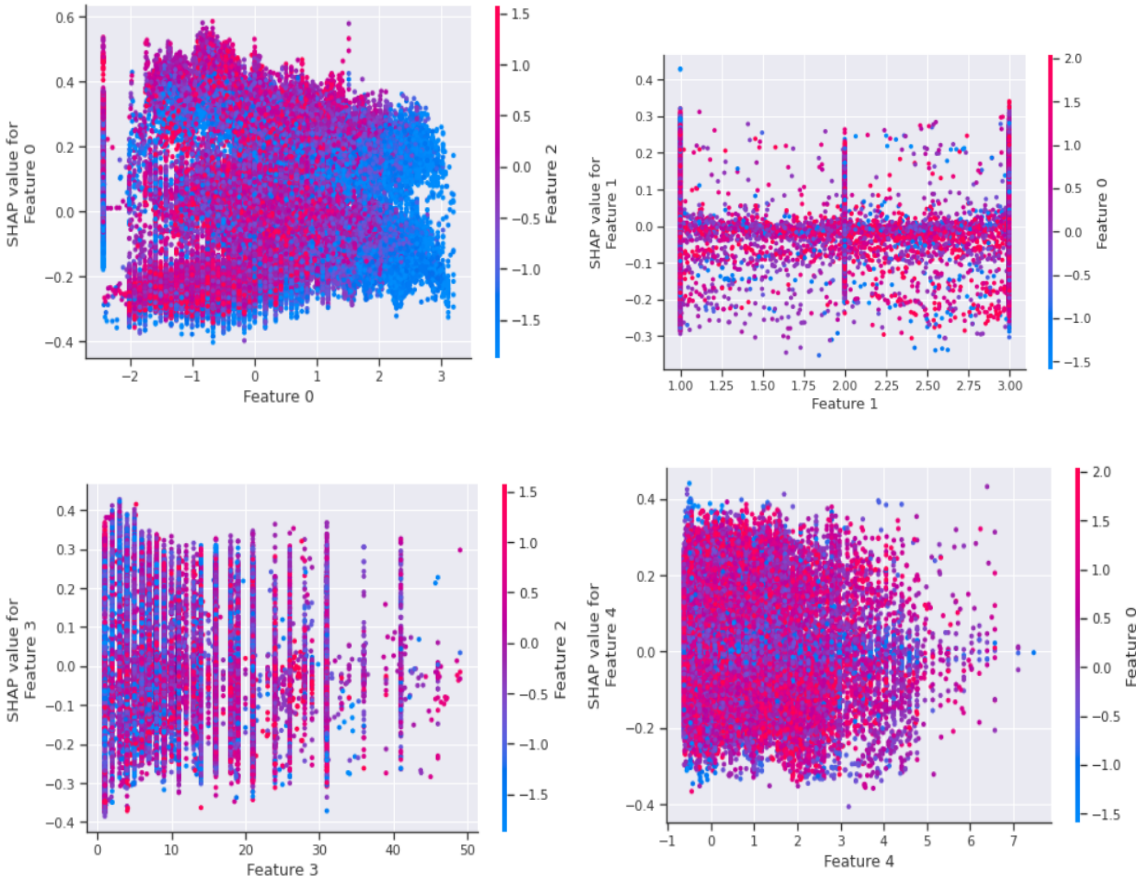
In Our case

- Feature 0 - Score
- Feature 1- Past communication
- Feature 2- Category
- Feature 3- Word Count
- Feature 4- images



Above plot shows a global summary of the distribution of SHAP values over all features. For each feature (horizontal rows), you can see the distribution of feature importance. From the diagram we can see that score and category have large effects on the prediction over the entire dataset (high SHAP value shown on bottom axis). High category values affect the prediction negatively (red values on the left hand side), while score category values affect the prediction positively (red values on the right hand side), similarly in the opposite direction for both variables.

Interpreting the interpretation plots –



Interpretation from interpretation plot

- 1) The 1 st graph - (Score vs Category) The high score and low value of category decreases the probability of target class.
- 2) The 2 nd graph - (Score vs Past communication) the low score and high past communication decreased the probability of target class.
- 3)The 3 rd graph - (Category vs Word count) The low word count and high category value increased the probability of target class
- 4) The 4 th graph - (Total links vs score) Low value of links and high score increased the probability of target class.

IX. Challenges

- Overfitting was another major challenge during the modelling process.
- Decision making on missing value imputations and outlier treatment was quite challenging as well.
- Interpreting the SHAPE model was challenging.

X. Conclusion

- In EDA, we observed that the category 2, email-type 1 & campaign type-2 was high in number. Univariate analysis and bivariate analysis gave us features of information distribution.
- It was observed that both Time_Email_Sent and Customer_Location were insignificant in determining the Email_status. The ratio of the Email_Status was the same irrespective of the demographic or the time frame the emails were sent on.
- We have imbalanced data; the status with value 0 comprises 80% of data. So to deal with it we used the SMOTE method.
- Based on the metrics, XGBoost Classifier worked the best, giving a train score of 98% and test score of 84% for F1 score.
- From SHAP we observed that the score and Category are important features.