Capstone Project

Email Campaign Effectiveness Prediction

By

Suneel Nelaturi

AGENDA

Problem Statement

Data Summary

03 Exploratory Data Analysis

04 Feature Engineering

05 Model Selection & Training

06 Model Evaluation & Cross – Validation

Problem Statement

Objective of Project:

- 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.

- The main objective is to create a machine learning model to characterize the mail and track the mail that is ignored; read; acknowledged by the reader.

- Develop the perceiving of various parameters involved in email campaigns.

Descriptive Statistical Analysis

Subject_Hotness_Score	Total_Past_Communications	Total_Links	Total_Images	Word_Count
2.2	33.0	8.0	0.0	440
2.1	15.0	5.0	0.0	504
0.1	36.0	5.0	0.0	962
3.0	25.0	16.0	0.0	610
0.0	18.0	4.0	0.0	947

Descriptive Statistical Analysis (Numerical Features)

- The average subject hotness score for the given data set is around 1.10
- The average total past communications are around 29. maximum total past communications are around 67 and the minimum is 0.
- The average word count is around 700 words. An email was sent with maximum words of around 1316 words.

Exploratory Data Analysis On

Numerical Features

Categerocal Features

Analysis of Categorical Features

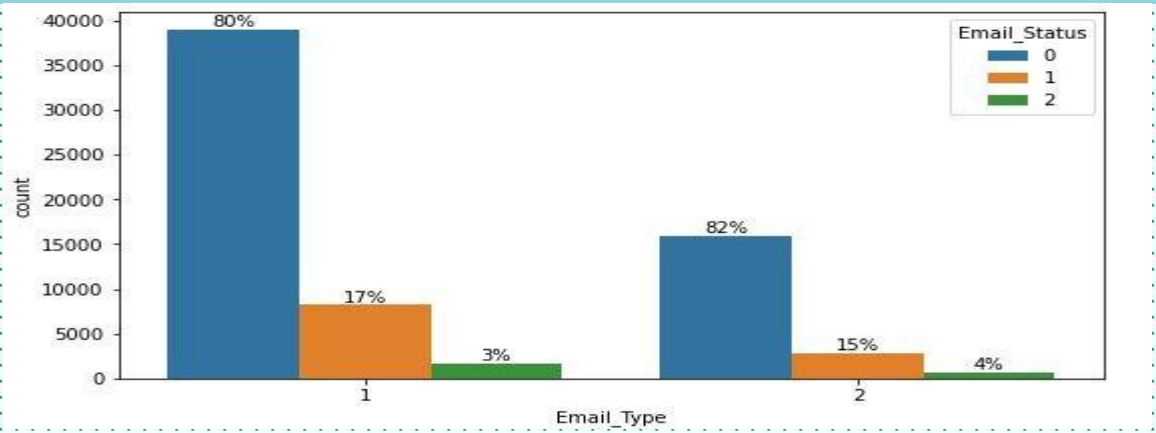


Fig 2. Email Type Vs Email Status (First)

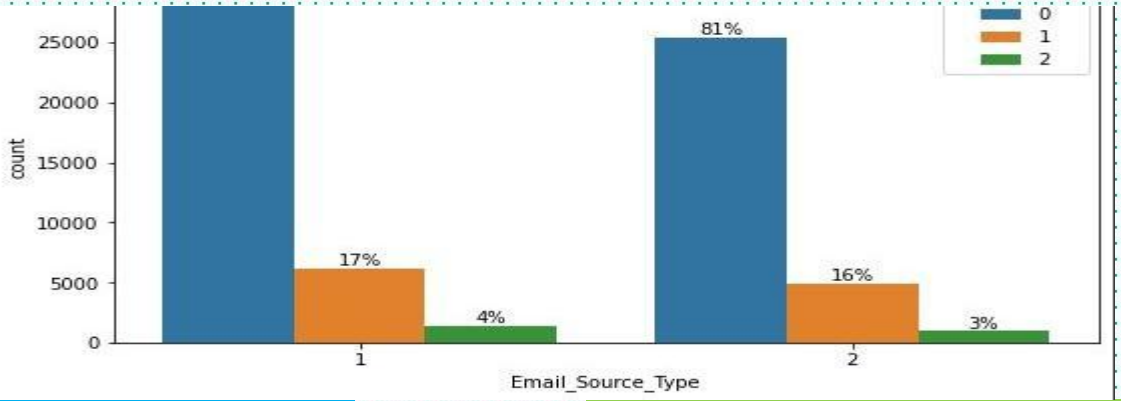
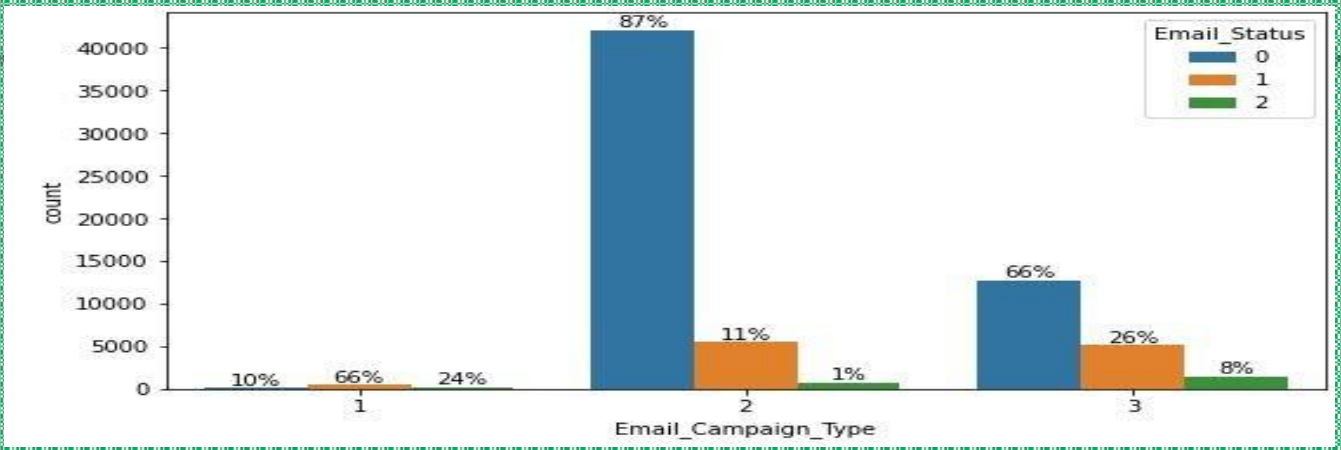


Fig 3. Email Source Type Vs Email Status (Second)

Analysis of Categorical Features (contd)



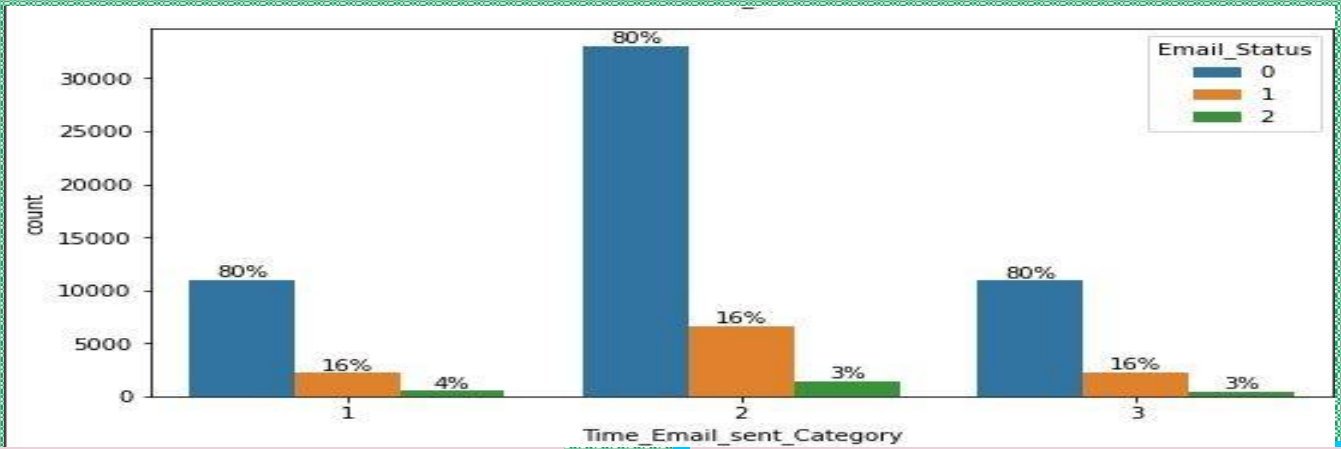
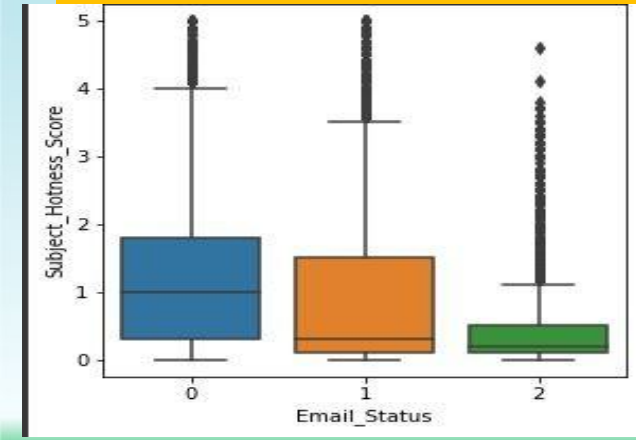
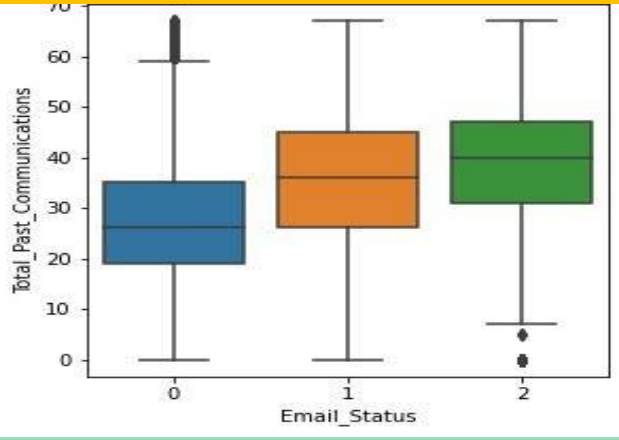


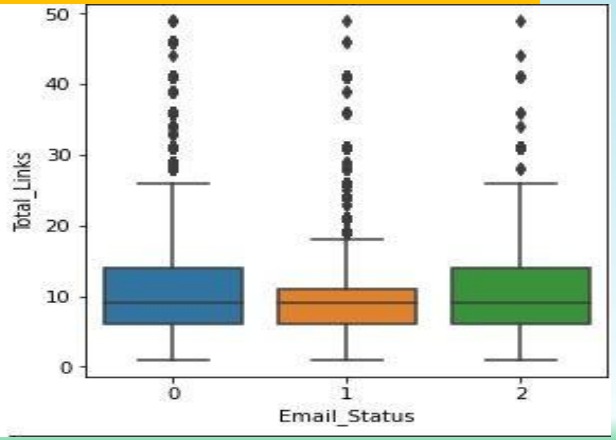
Fig 4: Email Campaign Vs Email Status (Above)

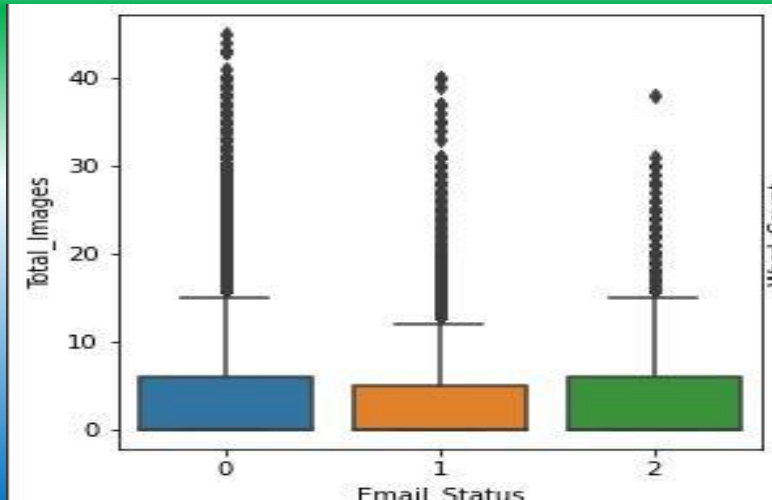
Fig 5: Time Email Sent Category Vs Email Status (Below)

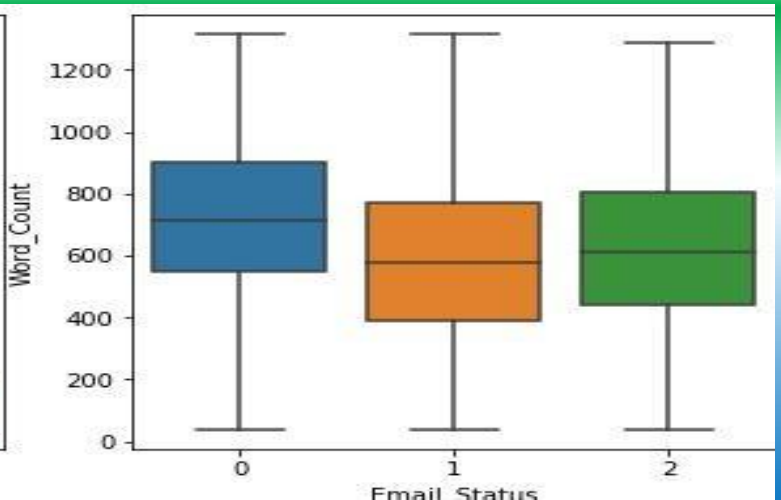
Analysis of Numerical Features



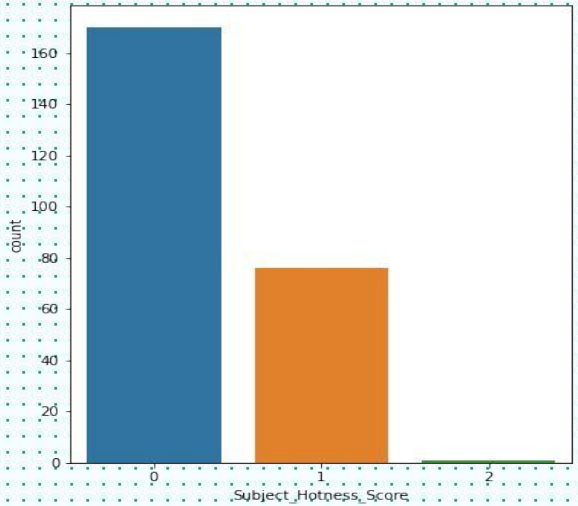


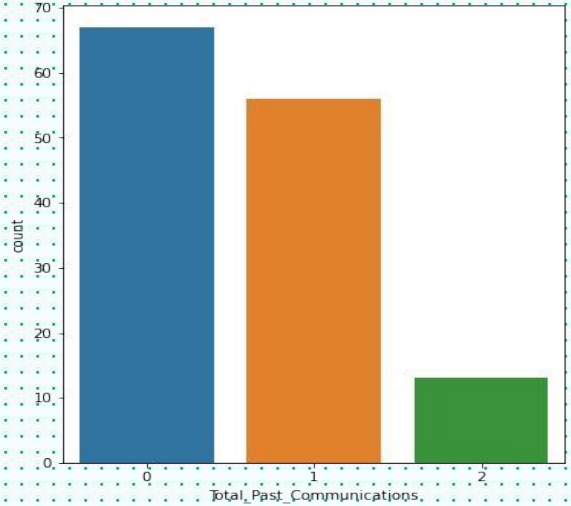


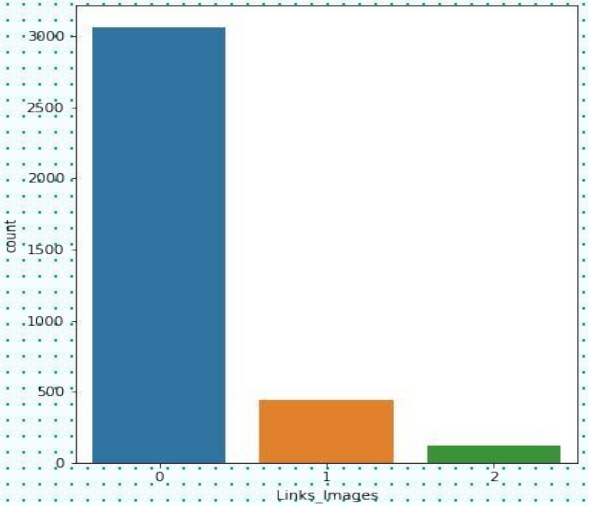




Outlier Treatment

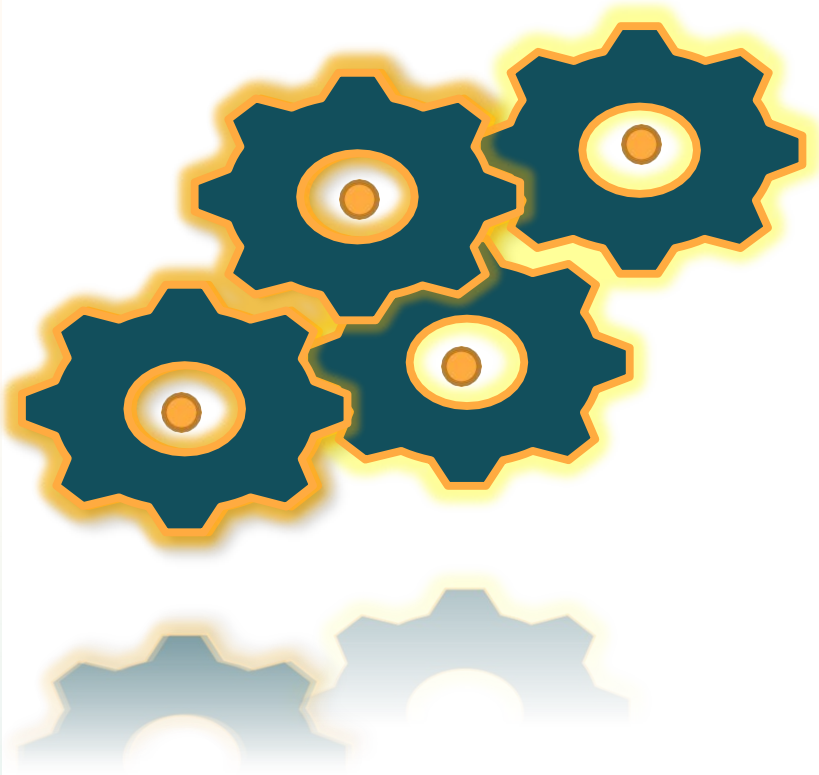







```
Percentage of majority class having outliers =  6.0
Percentage of minority class having outliers =  5.26
```

Feature Engineering



1. Combining Total Images and Total Links:

High positive correlation observed and hence Links Images = Total Images + Total Links

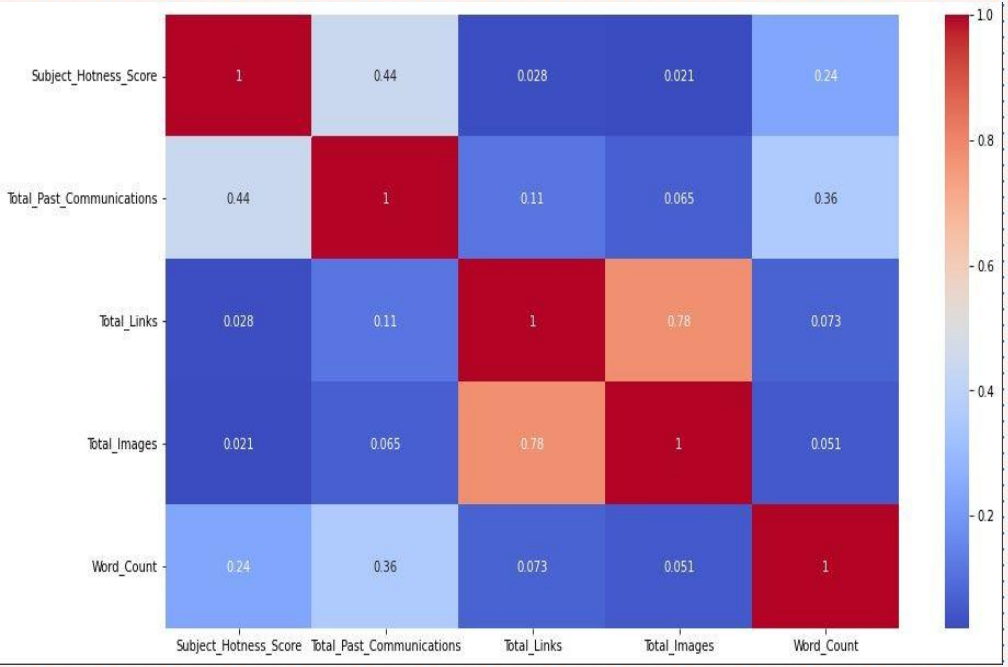


Fig 8: Heatmap Correlation

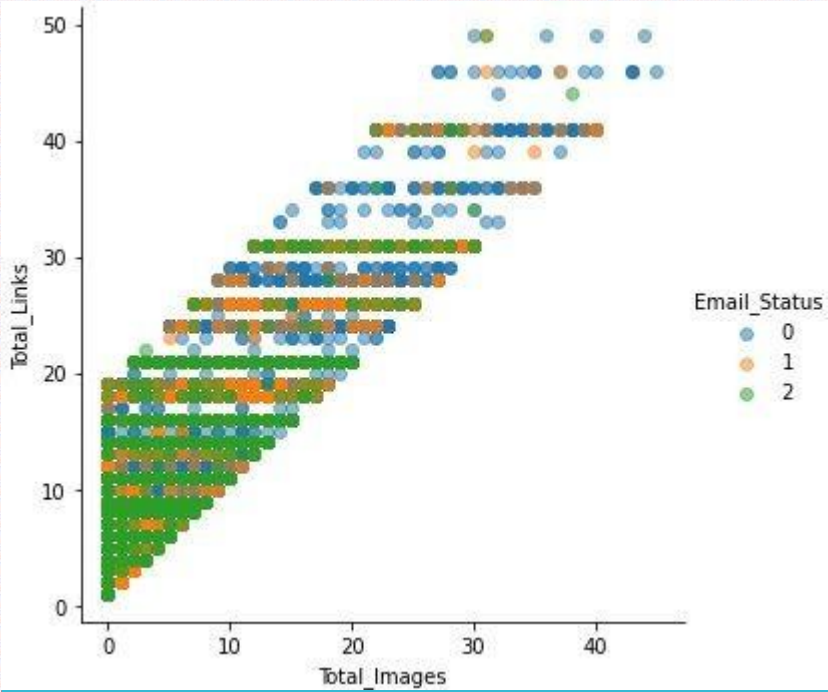


Fig 9: Linear Relationship Between Total Images & Total Links With Target Variable

2. Multicollinearity checking using VIF Factor:

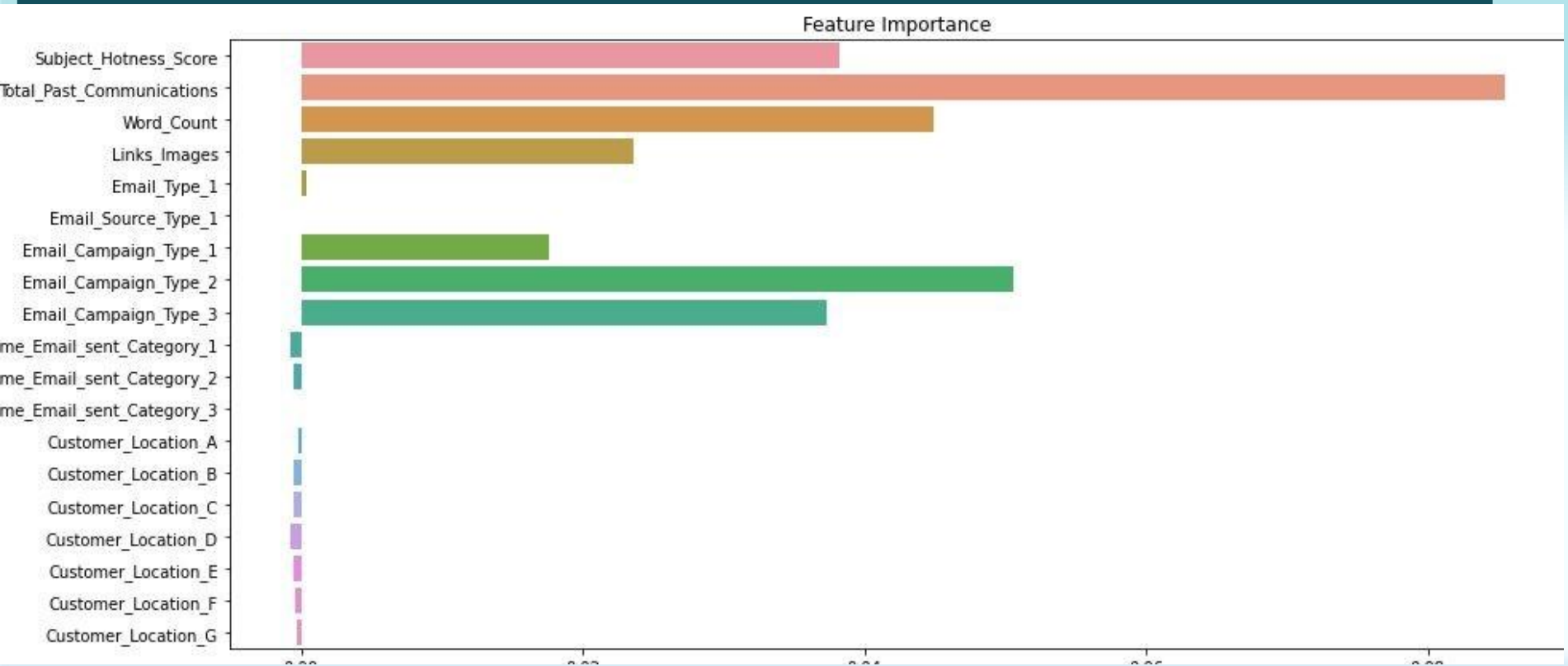
- Variables with high multicollinearity can adversely affect the model and removing highly correlated independent variables can help in reducing the curse of dimensionality as well

- We can observe that all numerical variables are within the threshold (i.e., 5)

	variables	VIF
0	Subject_Hotness_Score	2.062931
1	Total_Past_Communications	5.423955
2	Time_Email_sent_Category	8.815890
3	Word_Count	5.192721
4	Email_Status	1.300847
5	Links_Images	2.632242

Fig 10: Variance Inflation Factor

3. Feature Importance



Feature Importance

3. Understanding Feature Importance

AI

The concept used to understand feature importance is Information Gain.

- It explains which feature has maximum impact in classification, based on the notion of Entropy.
- It works well for numeric as well as categorical data.
- From the graph, we understand that Total_Past_Comunication and Email_Campaign_Type have high importance.
- Time_Email_Sent_Category and Customer_Location are not important and hence we decide to drop the feature.

4. Feature Scaling & OneHotEncoding

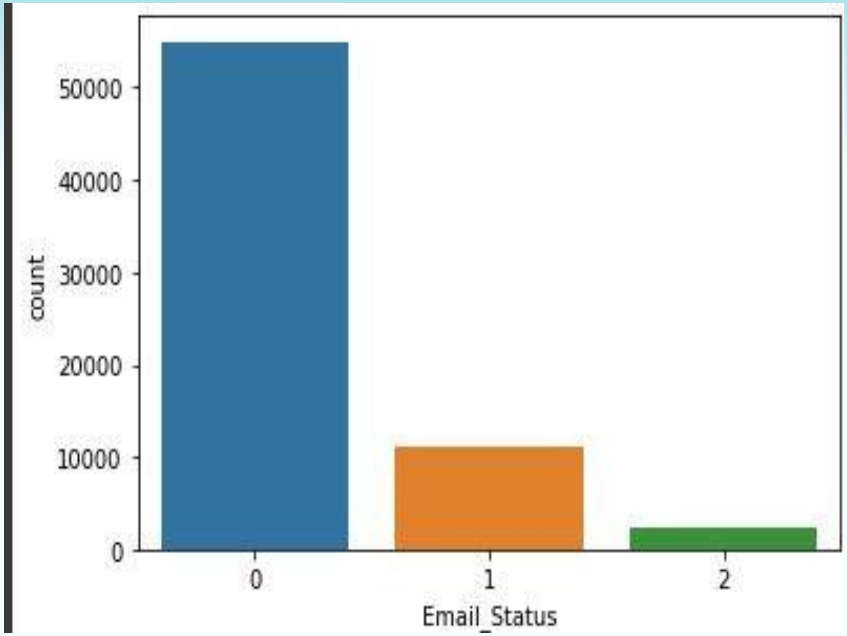
Numerical variables were scaled using **MinMaxScaler**.

The numerical features of the dataset do not have a certain range and they differ from each other.

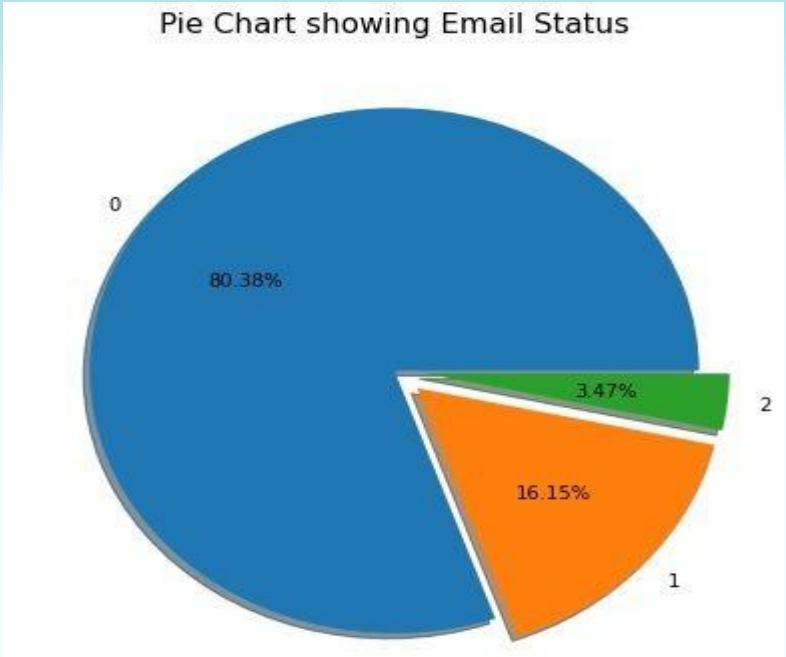
Categorical variables were encoded using **One-Hot Encoding.**

This method changes categorical data to a numerical format and enables you to group your categorical data without losing any information

Target Variable



Email Status (Target variable) Count



Email Status Percentage Count

Understanding Target Variable

The target variable consists of 3 classes:

- 0 - ignored - 54941
- 1 - read - 11039
- 2 - acknowledged - 2373
- The target Variable was highly imbalanced.

What is Imblnce in the Model?

Class imbalance is when the number of samples is different for the different classes in the data. Most models trained on imbalanced data will have a bias towards predicting the larger class(es) and, in many cases, may ignore the smaller class(es) altogether.

Techniques for preventing the Imbalane in the Model

There are techniques by which this problem can be prevented

1. Under Sampling
2. Over Sampling

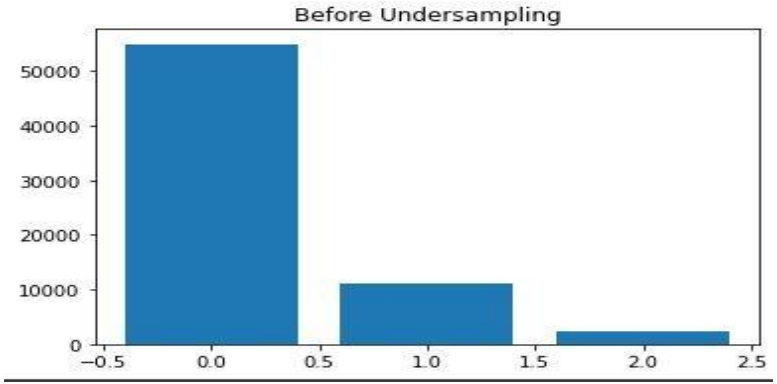
Handling Imbalanced data

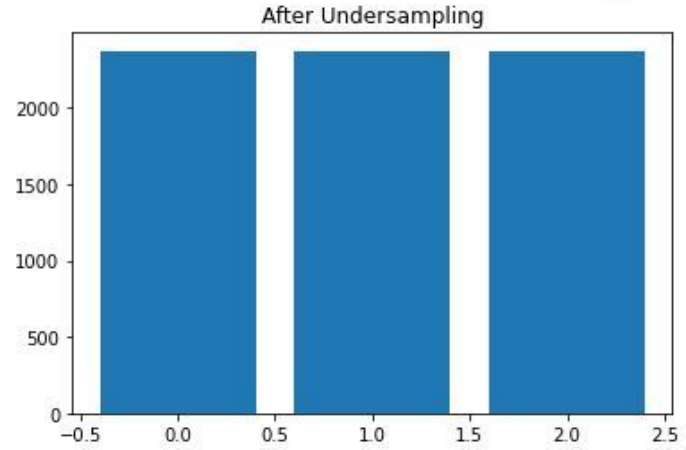
1. **Under-sampling Technique:**

- The technique used was Random Under Sampler.
- Created balanced data with 2373 records for each class.
- We observe the data before and after Under-Sampling from the following figure.

Why didn't it work?

Created baseline models with under sampled data and it was observed that they underperformed primarily due to loss of information.





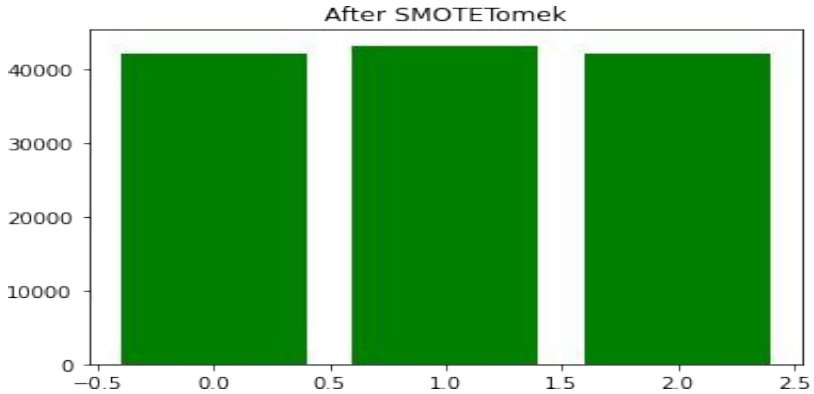
Handling Imbalanced data

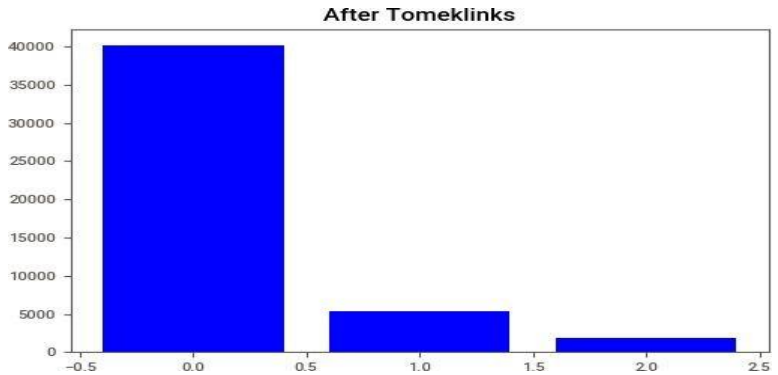
2. Oversampling Technique:

The technique used was **SMOTETomek & Tomeklink**

- Frequency of unique values of the Email Status: [[0, 1, 2] [42018, 42147, 43166]] using **SMOTETomek**

- Frequency of unique values of the Email Status: [[0 1 2] [40204, 5321, 1898]] using **Tomeklink**



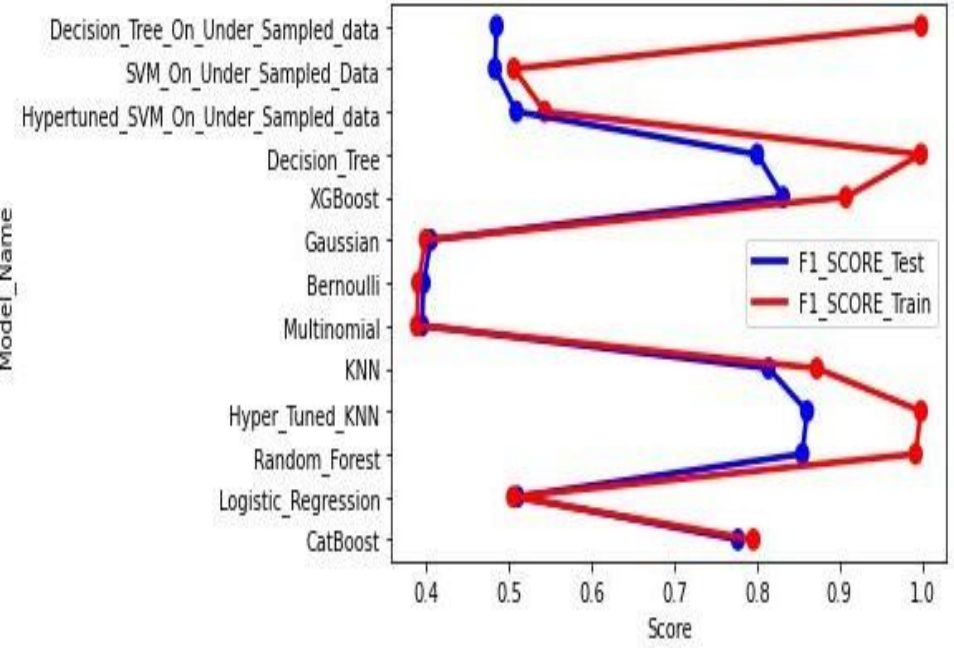


Model Training and Evaluation

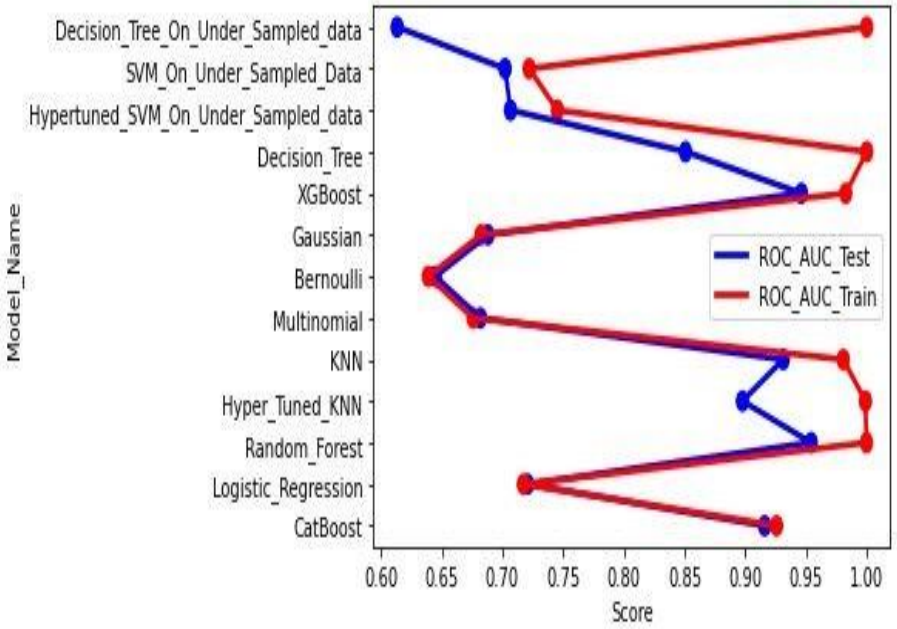
Applying Model (baseline model)

	MODEL_NAME	F1_SCORE_Train	PRECISION_Train	RECALL__Train	ACCURACY_Train	ROC_AUC_Train	F1_SCORE_Test	PRECISION_Test	RECALL__Test	ACCURACY_Test	ROC_AUC_Test
0	Decision_Tree_On_Under_Sampled_data	0.999122	0.999123	0.999122	0.999122	0.999998	0.484969	0.485657	0.484551	0.484551	0.613363
1	SVM_On_Under_Sampled_Data	0.505861	0.523580	0.536435	0.536435	0.722145	0.482963	0.505824	0.511938	0.511938	0.702254
2	Hypertuned_SVM_On_Under_Sampled_data	0.543274	0.554801	0.564355	0.564355	0.745338	0.508849	0.517580	0.527388	0.527388	0.706699
3	Decision_Tree	0.998658	0.998661	0.998658	0.998658	0.999998	0.800345	0.800267	0.800546	0.800546	0.850858
4	XGBoost	0.907731	0.911376	0.909258	0.909258	0.982778	0.831709	0.834949	0.835280	0.835280	0.946248
5	Gaussian	0.399132	0.460576	0.494752	0.494752	0.682553	0.405315	0.468087	0.502199	0.502199	0.688096
6	Bernoulli	0.390192	0.532041	0.486667	0.486667	0.639020	0.396654	0.530756	0.494373	0.494373	0.644810
7	Multinomial	0.388675	0.324004	0.486121	0.486121	0.675953	0.394757	0.329046	0.493736	0.493736	0.681925
8	KNN	0.872618	0.883761	0.875366	0.875366	0.980696	0.814249	0.827818	0.818868	0.818868	0.931304
9	Hyper_Tuned_KNN	0.998627	0.998627	0.998627	0.998627	0.998970	0.860853	0.868487	0.863613	0.863613	0.897708
10	Random_Forest	0.992279	0.992303	0.992280	0.992280	0.999827	0.855030	0.855383	0.855574	0.855574	0.954264
11	Logistic_Regression	0.505009	0.515434	0.530988	0.530988	0.717340	0.510220	0.520776	0.536963	0.536963	0.720834
12	CatBoost	0.795830	0.800387	0.802500	0.802500	0.925821	0.777185	0.781551	0.784741	0.784741	0.916035

Test F1 Score & Test Roc - AUC Score Comparison



F1 Score



ROC-AUC

Best Performing Model

XGBoost

- Robust to outliers.
- Supports regularization.
- Works well on small to the mediu dataset.
- F1 score for train & test set were 90% & 77% respectively

Conclusion

- In EDA, we observed that Email_Campaign_Type was the most important feature. If your Email_Campaign_Type was 1, there is a 90% likelihood of your Email to be read/acknowledged.
- 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 location or the time frame the emails were sent on.
- As the word_count increases beyond the 600 mark, we see that there is a high possibility of that email being ignored. The ideal mark is 400-600. No one is interested in reading long emails!
- For modeling, it was observed that for imbalance handling Oversampling i.e., SMOTELink & SMOTETomek worked way better than under-sampling as the latter resulted in a lot of loss of information.
- Based on the metrics, XGBoost Classifier worked the best giving a train score of 92% and a test score of 77% for F1 score.

Challenges

AI

1. Choosing the appropriate technique to handle the imbalance in data was quite challenging as it was a tradeoff between information loss vs the risk of overfitting.

2. Overfitting was another major challenge during the modeling process.

3. Understanding what features are most important and what features to avoid was a difficult task.

4. Decision-making on missing value imputations and outlier treatment was quite challenging as well.


THANK
YOU