E-mail Campaign Effectiveness Prediction

Midhun R
Data Science Trainee,
AlmaBetter, Bangalore

Abstract:

Email marketing is a form of direct marketing that uses electronic mail as a means of communicating commercial or fundraising messages to an audience. Sending emails with the purpose of acquiring new customers or convincing current customers to purchase something immediately.

Email campaign effectiveness measures how well an email campaign achieves its objectives. To measure email campaign effectiveness, marketers need to consider many factors.

In this project, I have attempted to analyze the data on e-mail marketing campaign and build a machine learning model to predict the mail that is ignored, read, or acknowledged by the reader. No personal information of recipient is provided in this dataset.

Keywords: EDA, Feature Engineering, Modelling, Classification, Decision Tree, Random Forest, XGBoost, KNN

01. Problem Statement

Most of the small to medium business owners are making effective use of Gmail-based e-mail 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.

A dataset containing 68353 records across 12 features has been given with information regarding the characteristics of e-mail marketing campaign. The main objective is to understand the existing data and identify the key factors that will affect the e-mail status, so that a predictive model can be built to improve campaigns.

02. Introduction

In this day and age, when data science has transformed into an invisible force that influences the decisions of the modern world, marketing sector has also become beneficiary of data science.

In its broadest sense, every email sent to a potential or current customer could be considered email marketing. However, the term is usually used to refer to: Sending emails with the purpose of enhancing the relationship of a merchant with its current or previous customers and to encourage customer loyalty and repeat business.

Email marketing is important because it allows organizations to reach a large audience at a low cost. It also allows organizations to track the results of their campaigns and make improvements based on feedback. Email marketing can be an extremely effective way to reach your target audience. When done correctly, it can result in increased sales, higher customer loyalty, and improved brand awareness.

The goal of this project is to inspect, clean and analyze the given dataset on e-mail marketing campaign to find out which factors affect the success of the campaign and how they will affect it. This information is used to build predictive models and compare them to find out the best model to predict the outcome of each campaign.

03. Approach

The sequence of steps taken to solve the task are as follows:

1. Understanding the business task.
2. Import relevant libraries and define useful functions.
3. Reading data from files given.
4. Data pre-processing, which involves inspection of both datasets and data cleaning.
5. Exploratory data analysis, to find which factors affect sales and how they affect it.
6. Feature engineering, to prepare data for modelling.
7. Modelling data and comparing the models to find out most suitable one for forecasting.
8. Conclusion.

04. Business Task

Build a machine learning model to predict the mail that is ignored, read, or acknowledged by the reader. This is undertaken as an individual project.

05. Import Libraries and Define Function

Some relevant libraries imported for aid are:

1. NumPy, for numerical operations
2. Pandas, for data manipulation
3. Matplotlib and Seaborn, for data visualization.
4. Statsmodels, for statistical data exploration.
5. Scikit Learn, for machine learning.
6. Imbalanced-learn, for dealing with classification with imbalanced classes.
7. XGBoost

In addition to this, few useful functions were defined to avoid repetition of codes.

06. Reading Data

After drive was mounted, data from csv files were read and stored in pandas dataframes.

07. Data Inspection

After loading the datasets and importing relevant libraries, the dataset has been explored thoroughly by looking into its head, tail, brief summary, number of records and features, etc.

The dataset contains 68353 records across 12 features with information regarding the characteristics of e-mail marketing campaign. The dataset doesn’t have any duplicate rows but four columns in store dataset have missing values. Three columns require conversion of datatypes.

The features in store dataset were identified as:

1. Email_ID: E-mail ID of recipients.
2. Email_Type: Differentiates between 2 different e-mail types: 1 and 2.
3. Subject_Hotness_Score: Measures the strength and effectiveness of mail subject.

4. Email_Source_Type: Differentiates between 2 different e-mail source types: 1 and 2.
5. Customer_Location: Differentiates between 7 different e-mail customer locations: A, B, C, D, E, F and G.
6. Email_Campaign_Type: Differentiates between 3 different e-mail campaign types: 1, 2 and 3.
7. Total_Past_Communications: Number of previous communications from the same source.
8. Time_Email_sent_Category: Differentiates between 3 different time of day category: 1, 2 and 3.
9. Word_Count: Number of words in the mail.
10. Total_Links: Number of links in the mail.
11. Total_Images: Number of images in the mail.
12. Email_Status: Differentiates between 3 different e-mail statuses: 1, 2 and 3, representing ignored, read & acknowledged respectively. This is our target variable.

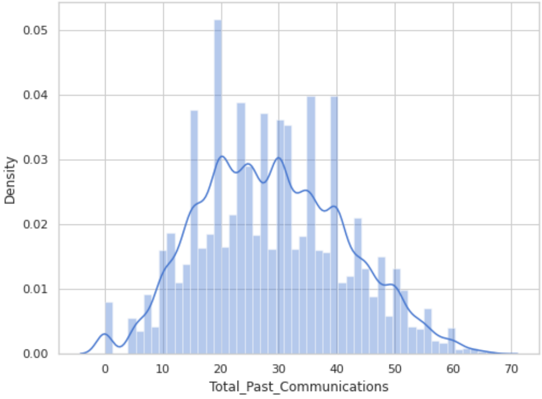
08. Data Cleaning

Data cleaning is done to ensure that the dataset is correct, consistent, and usable.
Four out of 10 columns have missing values in them.

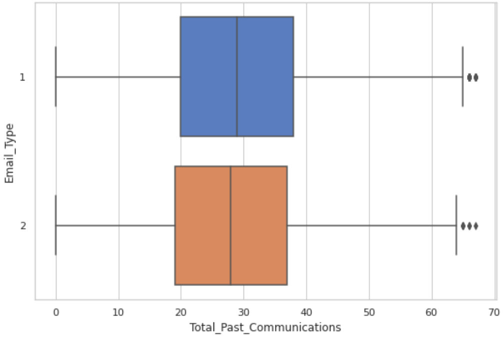
```
Customer_Location: 11595(16.96)%
Total_Past_Communications: 6825(9.98)%
Total_Links: 2201(3.22)%
Total_Images: 1677(2.45)%
```

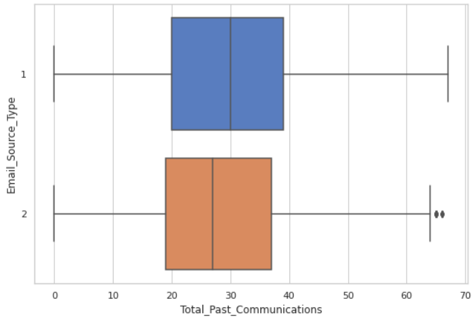
‘Customer_Location’ had the most number of missing values. ‘Customer_Location’ is a categorical feature but it cannot be blindly imputed with its mode since a large number of observations have missing values in it. We cannot remove this column or these observations as it will lead to a loss of information. None of the other features has any possibility of causing some effect on customer location or vice versa. So, it is difficult to find a value to impute in relation to other features.

‘Total_Past_Communications’ is a numerical feature. So, the distribution of its values was checked using a density plot.



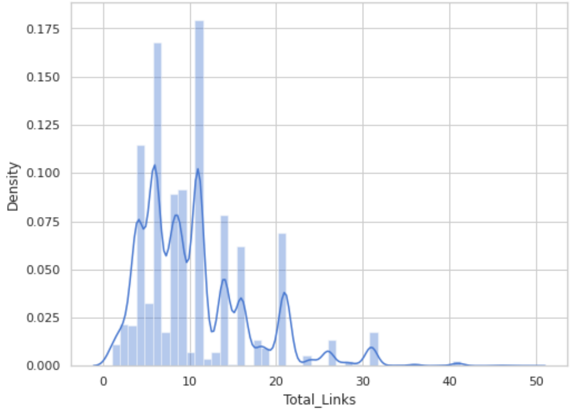
‘Total_Past_Communications’ has an almost normal distribution. So, mean or median can be used to impute missing values. Box plots were generated to show the distribution of ‘Total_Past_Communications’ in each category of ‘Email_Type’ and ‘Email_Source_Type’ to check if some other important features affect its values.





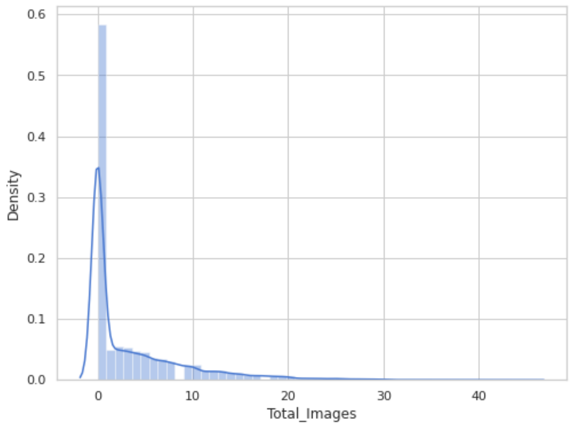
A box plot to show the distribution of values in 'Total_Past_Communications' to check for outliers. It has very less outliers. So, the missing values were imputed with its mean.

'Total_Links' is a numerical feature. So, the distribution of its values was checked using a density plot.



The distribution of 'Total_Links' is positively skewed. Since only a small number of observations have missing values in it, checking for the influence of other features is not necessary and it is safe to impute missing values with mode or median. Mean can be avoided as it has outliers. Since the PDF of the peak of the distribution is very low, it was imputed with median instead of mode.

'Total_Images' is a numerical feature. So, the distribution of its values was checked using a density plot.



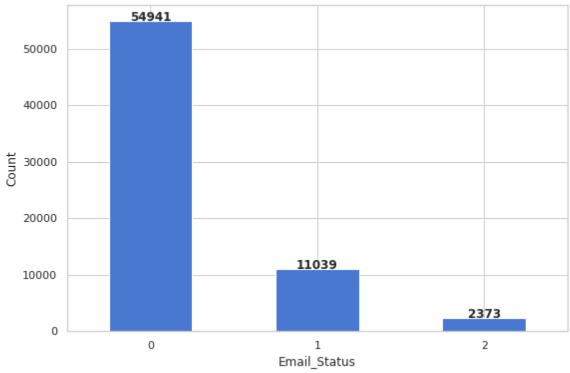
The distribution of 'Total_Images' is positively skewed. Since only a small number of observations have missing values in it, it is safe to impute missing values with mode or median. Since the peak of the distribution has high PDF, mode can be chosen over median.

Some columns were converted to appropriate datatypes to make analysis easier and more accurate. These columns were 'Total_Past_Communications', 'Total_Links' and 'Total_Images' (int).

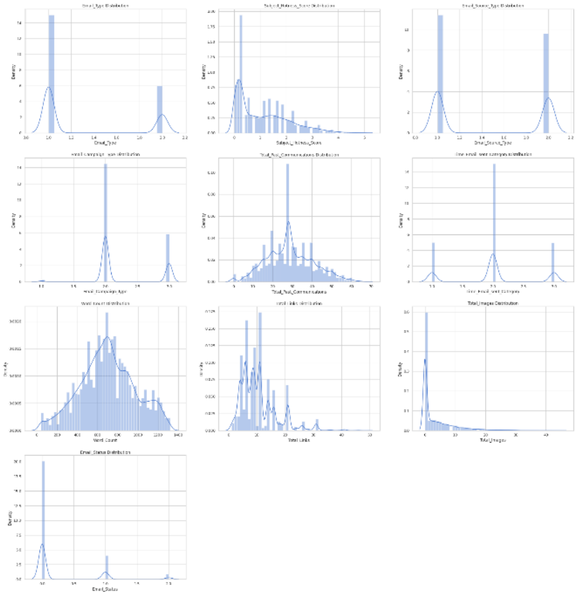
09. Exploratory Data Analysis

Dataset, after cleaning, is subjected to exploratory data analysis.

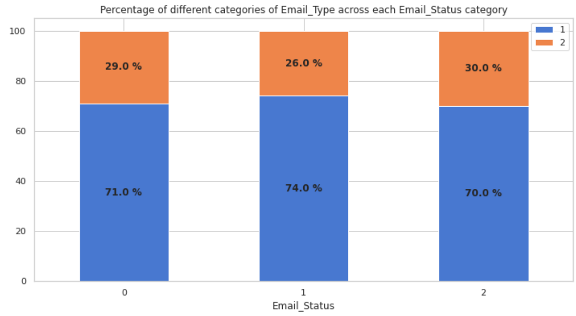
The following observations were made after EDA:

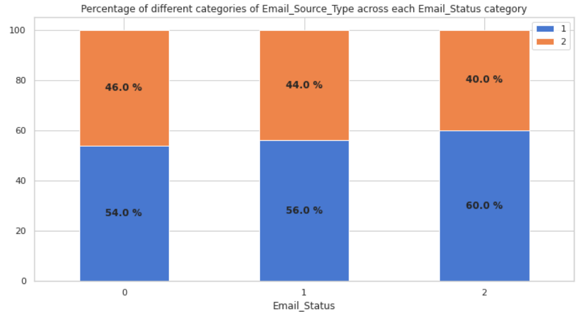


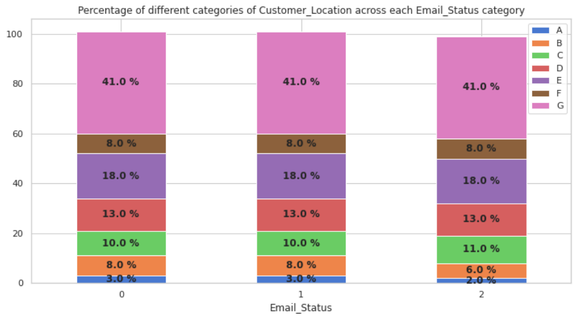
It is observed that 'Email_Status' 0 has more than 4 times the observations than rest of them combined. So, the dataset is highly imbalanced and ‘Email_Status’ 0 is the majority class and the rest of them are minority classes.

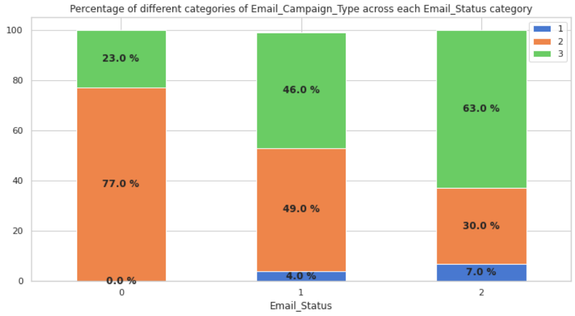


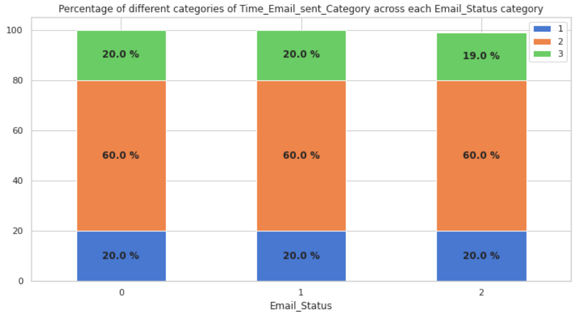
‘Subject_Hotness_Score’, ‘Total_Links’ and ‘Total_Images’ are positively skewed while ‘Total_Past_Communications’ and ‘Word_Count’ are almost normally distributed.



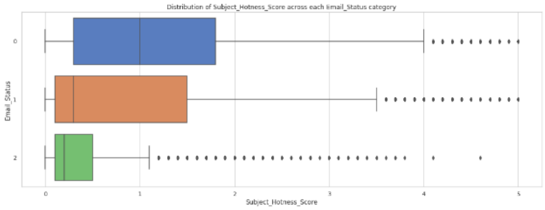


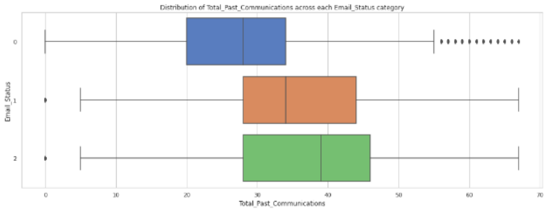


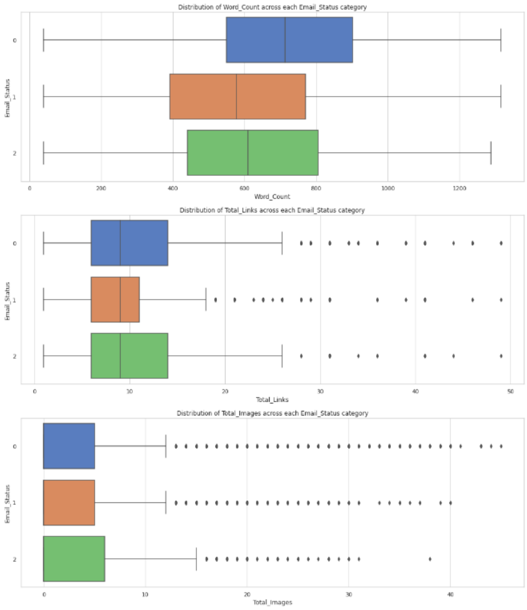




All categories of a feature have same distribution of e-mails across categories of ‘Email_Status'. ‘Email_Campaign_Type’ is the only feature which does not follow this trend. So it has the most impact on the target feature. If Email_Campaign_Type is 1, then the mail has 66% chance of getting read and 23% chance of getting acknowledged.







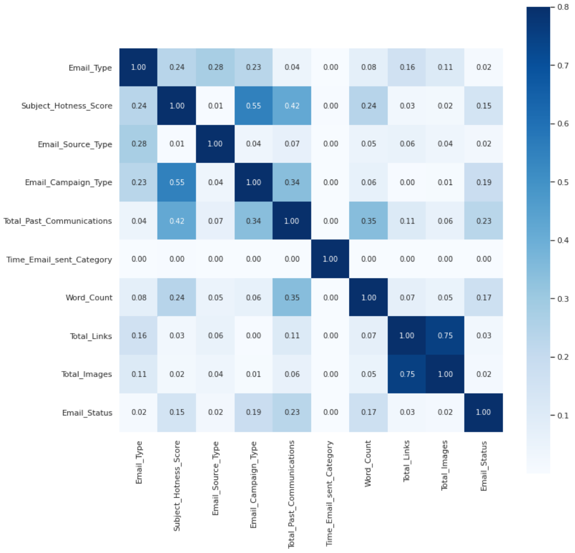
As the ‘Subject_Hotness_Score’ increases, probability of mails getting acknowledged decreases. There is a good chance of getting e-mails read or acknowledge if the ‘Subject_Hotness_Score’ is less than 5.

As the number of ‘Total_Past_Communications’ increases, probability of mails getting acknowledged or read also increases. There is a high chance of getting e-mails read or acknowledge if the number of ‘Total_Past_Communications’ is greater than 25.

As the ‘Word_Count’ increases, probability of mails getting ignored also increases. There is a high chance of getting e-mails read or acknowledge if the ‘Word_Count’ is between 400 and 600.

The ‘Total_Links’ have similar distribution across each ‘Email_Status’ category but read mails have slightly less variance than others.

The ‘Total_Images’ have similar distribution across each ‘Email_Status’ category but acknowledged mails have slightly more variance than others.



‘Time_Email_sent_Category’ has no correlation with ‘Email_Status’ or any of the other independent features.

Multicollinearity can be observed between ‘Subject_Hotness_Score’, ‘Email_Campaign_Type’ & ‘Total_Past_Communications’ and ‘Total_Links’ & ‘Total_Images’.

10. Feature Engineering

Feature engineering is the process of selecting, manipulating, and transforming raw data into features that can be used in machine learning. This process can significantly affect the performance of the model.

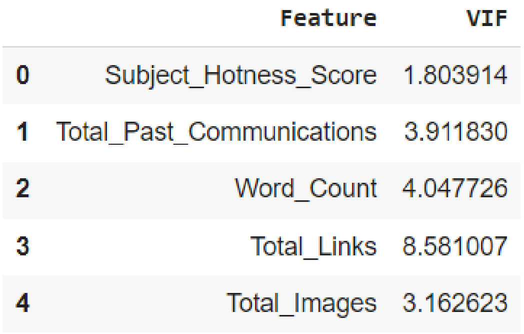
The following processes were involved in feature engineering:

1. **Feature Selection**: ‘Email_ID’ is dropped since it does not affect the status of e-mail.

   ‘Date Customer_Location’ is also dropped since it has a lot of missing values which cannot be easily imputed and also it does not have much impact on the target variable as all the locations have same probability of mails getting ignored, read and acknowledged.

‘Time_Email_sent_Category’ is also dropped as it is already established in EDA that it has no correlation to any features and therefore, it doesn't affect the mail status.

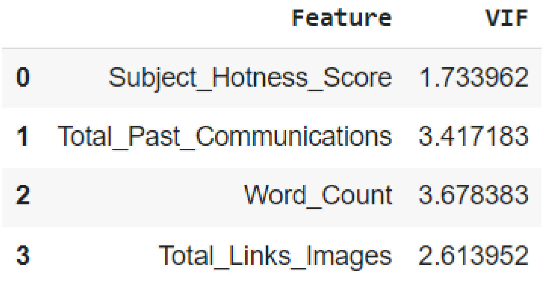
2. **Handling Multicollinearity**: The variance inflation factor (VIF) of all numerical features is calculated in order to remove highly correlated features. Features having VIF greater than 5 should be eliminated.



	Feature	VIF
0	Subject_Hotness_Score	1.803914
1	Total_Past_Communications	3.911830
2	Word_Count	4.047726
3	Total_Links	8.581007
4	Total_Images	3.162623

The only feature with VIF greater than 5 is ‘Total_Links’. But it cannot be removed from our dataset because it is an important feature. As observed in EDA, ‘Total_Links’ and ‘Total_Images’ have good a correlation. So, these two features were combined into a new feature ‘Total_Links_Images’.

Then VIF of all remaining numerical features were calculated.

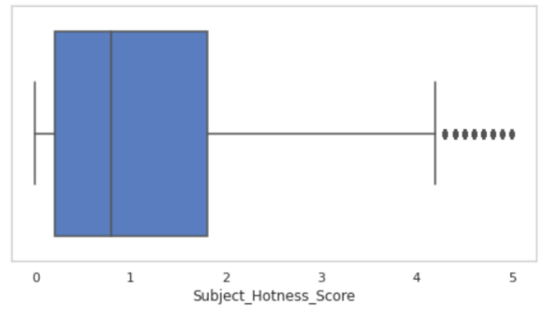


	Feature	VIF
0	Subject_Hotness_Score	1.733962
1	Total_Past_Communications	3.417183
2	Word_Count	3.678383
3	Total_Links_Images	2.613952

Now, all features have VIF less than 5.

3. **Handling Outliers**: Outliers in ‘Subject_Hotness_Score’ were checked with a box plot.

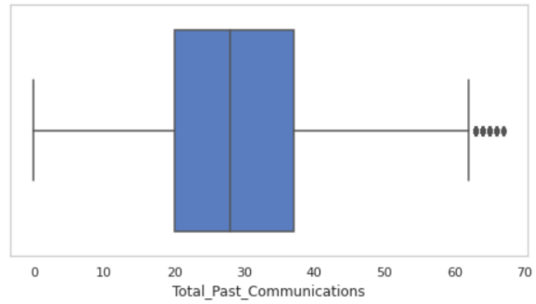
```
Outliers in Subject_Hotness_Score : 247 (0.36%)
```



Since this is a very small portion of the data. So, it was removed.

Outliers in ‘Total_Past_Communications’ were checked with a box plot.

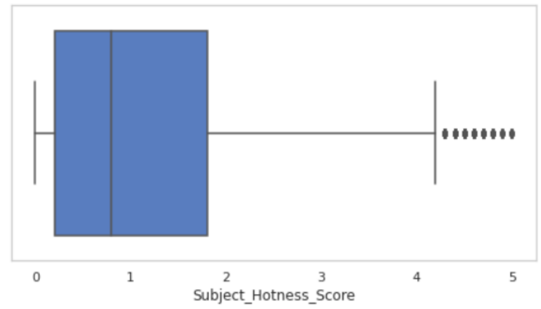
```
Outliers in Total_Past_Communications : 136 (0.2%)
```



Since this is a very small portion of the data. So, it was removed.

Outliers in ‘Word_Count’ were checked with a box plot.

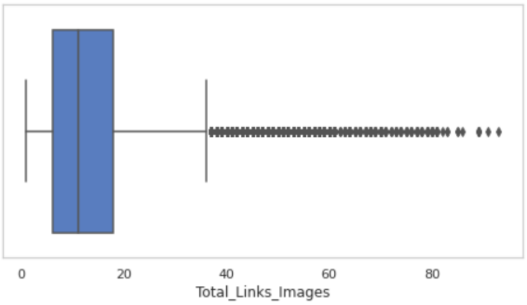
```
Outliers in Subject_Hotness_Score : 247 (0.36%)
```



There were no outliers in ‘Word_Count’.

Outliers in ‘Total_Links_Images’ were checked with a box plot.

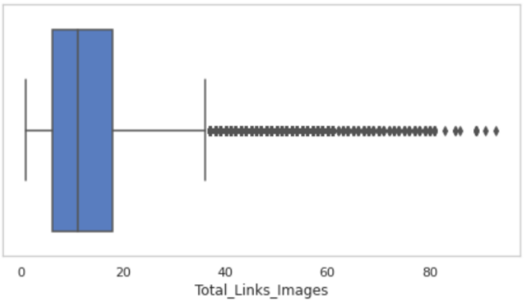
```
Outliers in Total_Links_Images : 3594 (5.31%)
```



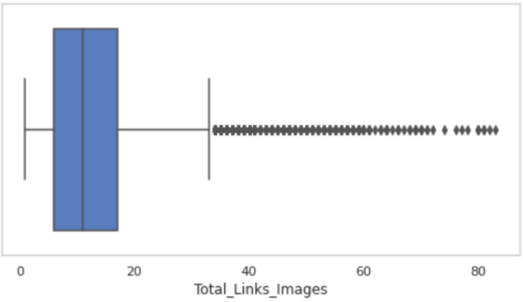
‘Total_Links_Images’ has a large number of outliers, which may cause loss of information. But it is acceptable if there are only less than 5% outliers in minority classes.

The percentage of outliers in majority and minority classes were found out using box plots.

```
Outliers in Total_Links_Images : 3043 (5.59%)
```

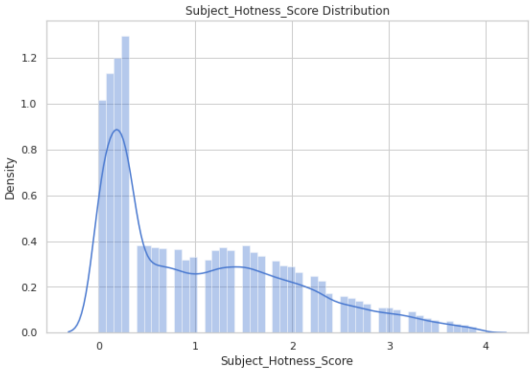


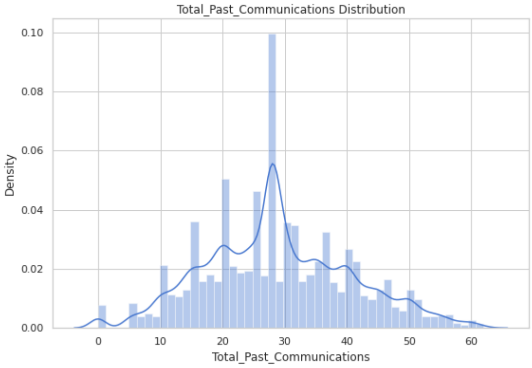
```
Outliers in Total_Links_Images : 721 (5.46%)
```

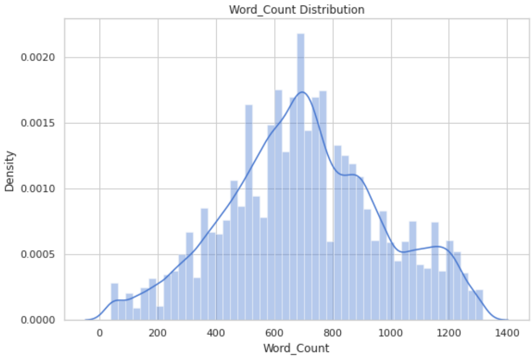


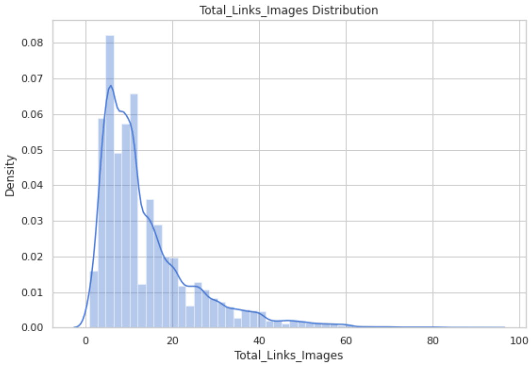
The percentage of outliers in minority classes is above 5. So, it is better to not remove them.

4. **Feature Transformation**: Transformation of numerical features is important because algorithms like linear regression, which assumes normal distribution, performs better when features are close to Normal Distribution.



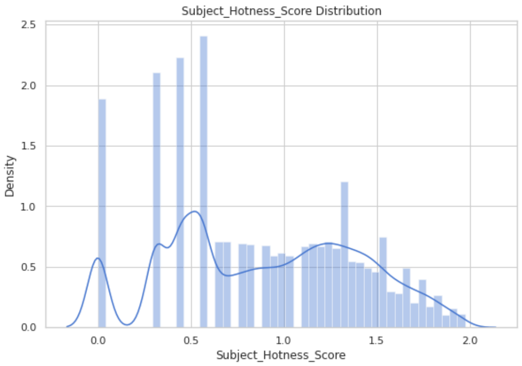


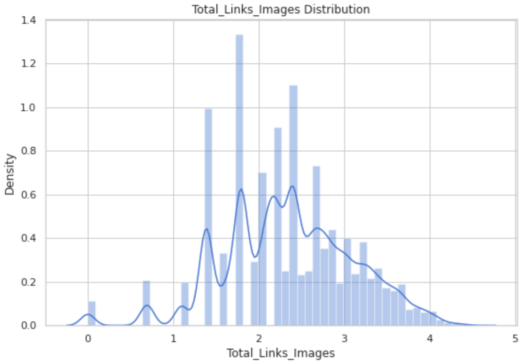




‘Subject_Hotness_Score’ and ‘Total_Links_Images’ are positively skewed. So, they must be transformed to normal distribution.

‘Subject_Hotness_Score’ has zero values while ‘Total_Links_Images’ has only positive values. So, ‘Subject_Hotness_Score’ is square root transformed and ‘Total_Links_Images’ is log transformed.





5. **Categorical Feature Encoding**: Most algorithms cannot handle the categorical variables unless they are converted into a numerical value. So, ‘Email_Type’, ‘Email_Source_Type’ and ‘Email_Campaign_Type’ were encoded using one hot encoder. To overcome dummy variable trap, one resultant feature from each encoded feature must be removed. There are only two unique values in ‘Email_Type’ and ‘Email_Source_Type’, so removal of any one encoded feature from each of them would be sufficient. But in the case of ‘Email_Campaign_Type’, since there are more than two unique values, correlation matrix was used to decide which feature to remove. ‘Email_Campaign_Type_1’ has the least correlation to Sales. So, it was removed along with ‘Email_Type_2’ and ‘Email_Source_Type_2’.

11. Modelling

Input and target data were separated, and both were split into training and test data with 25% test data. Training and test data of independent features were scaled using standardization.

The dataset is highly imbalanced. If the models are trained without fixing this problem, the model will be completely biased. So, 2 different techniques are used to balance the training data: Random Undersampling and SMOTE.

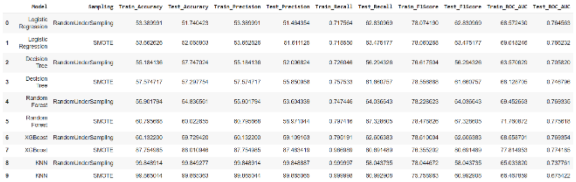
Each class in undersampled data has number of samples equal to the smallest original class whereas each class in oversampled data has number of samples equal to the largest original class.

Model training was done with these data using 5 different algorithms:

1. Logistic classification
2. Decision tree classification
3. Random forest classification
4. XGBoost classification
5. KNN classification

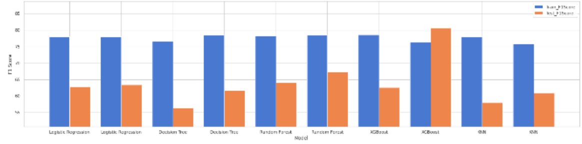
Models were trained with each algorithm twice, first with under-sampled data and second with over-sampled data.

Evaluation metrics like Accuracy, Precision, Recall, F1 Score and ROC-AUC were calculated for each model.



	Model	Sampling	Train_Accuracy	Test_Accuracy	Train_Precision	Test_Precision	Train_Recall	Test_Recall	Train_F1Score	Test_F1Score	Train_ROC_AUC	Test_ROC_AUC
0	Logistic Regression	RandomUnderSampling	53.389691	51.740423	53.389691	51.464354	0.717584	62.830069	78.074190	62.830069	68.572430	0.764563
1	Logistic Regression	SMOTE	53.662626	62.086903	53.662626	61.611126	0.718650	63.476177	78.060288	63.476177	69.016246	0.766232
2	Decision Tree	RandomUnderSampling	55.184136	57.747024	55.184136	52.096824	0.726046	56.204326	78.617504	56.204326	63.570626	0.705820
3	Decision Tree	SMOTE	57.574717	57.297754	57.574717	55.850858	0.757533	61.860757	78.566886	61.860757	68.126706	0.746796
4	Random Forest	RandomUnderSampling	55.901794	64.836561	55.901794	63.604338	0.747446	64.036543	78.228623	64.036543	69.452663	0.769936
5	Random Forest	SMOTE	60.785686	60.022855	60.796968	56.871044	0.787416	67.328605	78.478626	67.328605	71.780872	0.775618
6	XGBoost	RandomUnderSampling	60.132200	69.729420	60.132200	69.109163	0.795191	62.606383	78.610004	62.606383	68.658701	0.769954
7	XGBoost	SMOTE	87.754985	88.010946	87.754985	87.469419	0.986869	80.891489	78.355202	80.891489	77.814853	0.774185
8	KNN	RandomUnderSampling	99.848614	99.849277	99.848914	99.848867	0.999997	58.043795	78.044672	58.043795	65.039820	0.737761
9	KNN	SMOTE	99.865044	99.865363	99.865044	99.865065	0.999990	60.862909	75.750883	60.962905	68.467858	0.675422

Accuracy, precision, recall or ROC-AUC cannot be used to compare the performance of models since the data is imbalanced. So F1 score was used to compare different models and find out which one is better. Higher the F1 score, better the model.



The model built using XGBoost algorithm with SMOTE dataset has the highest F1 score and also all other models tends to overfit. So this model can be chosen.

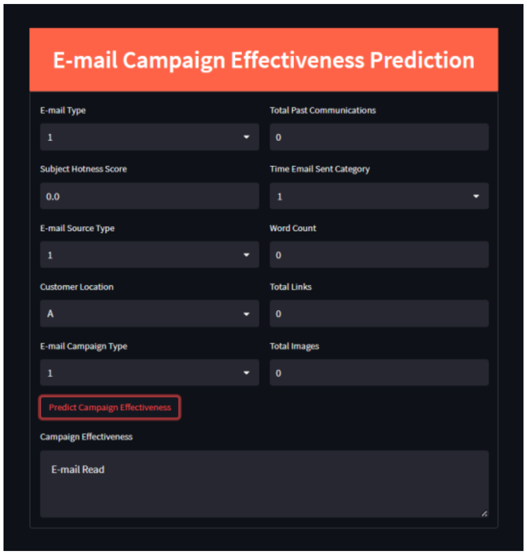
12. Challenges

Handling 68353 observations of data over 12 features of data were a bit difficult as a beginner. The inspection and cleaning of dataset was tedious and complicated. Visualization of data was properly carried out after providing a great amount of attention to each and every details.

Handling imbalanced data was a relatively newer task. Understanding what features most important and what features are to avoid was a difficult task. Selection of the right type of transformation took a lot of research. Training of models was a time-consuming process.

13. Model Deployment

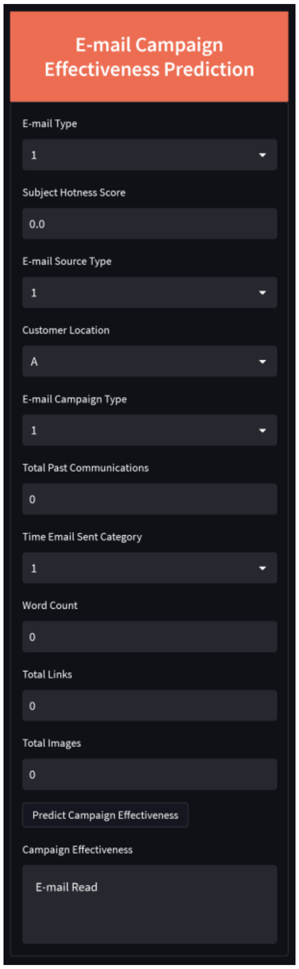
- A web application is built to demonstrate the working of the trained machine learning model using Streamlit.
- This web application is deployed with Streamlit Cloud Community, employing CI/CD pipeline.



E-mail Campaign Effectiveness Prediction

E-mail Type: 1
Total Past Communications: 0
Subject Hotness Score: 0.0
Time Email Sent Category: 1
E-mail Source Type: 1
Word Count: 0
Customer Location: A
Total Links: 0
E-mail Campaign Type: 1
Total Images: 0

Campaign Effectiveness: E-mail Read



E-mail Campaign Effectiveness Prediction

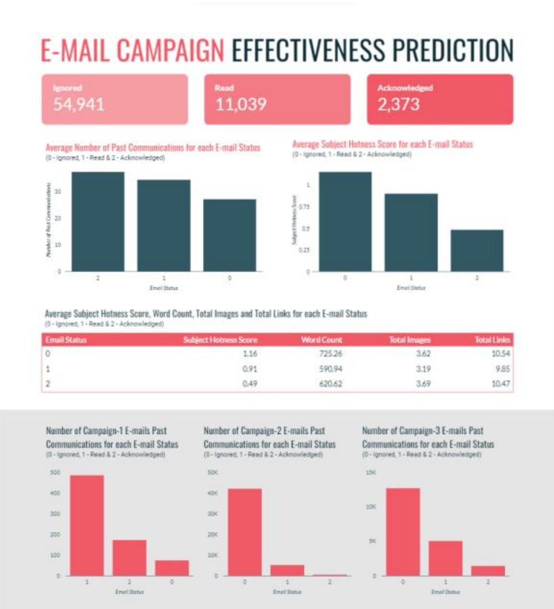
E-mail Type: 1
Subject Hotness Score: 0.0
E-mail Source Type: 1
Customer Location: A
E-mail Campaign Type: 1
Total Past Communications: 0
Time Email Sent Category: 1
Word Count: 0
Total Links: 0
Total Images: 0

Campaign Effectiveness: E-mail Read

Link to deployed model: https://connect-midhunr-email-campaign-effectiveness-predict-app-e58c8b.streamlit.app/

14. Data Visualization

An interactive dashboard was also created with Tableau to display charts associated with the analysis.



E-MAIL CAMPAIGN EFFECTIVENESS PREDICTION

Ignored 54,941

Read 11,039

Acknowledged 2,373

Average Number of Past Communications for each E-mail Status
(0 - Ignored, 1 - Read & 2 - Acknowledged)
Number of Past Communications
0
10
20
30
2
1
0
Email Status
Average Subject Hotness Score for each E-mail Status
(0 - Ignored, 1 - Read & 2 - Acknowledged)
Subject Hotness Score
0
0.25
0.5
0.75
1
0
1
2
Email Status

Average Subject Hotness Score, Word Count, Total Images and Total Links for each E-mail Status
(0 - Ignored, 1 - Read & 2 - Acknowledged)

Email Status	Subject Hotness Score	Word Count	Total Images	Total Links
0	1.16	725.26	3.62	10.54
1	0.91	590.94	3.19	9.85
2	0.49	620.62	3.69	10.47

Number of Campaign-1 E-mails Past Communications for each E-mail Status
(0 - Ignored, 1 - Read & 2 - Acknowledged)
0
100
200
300
400
500
1
2
0
Email Status
Number of Campaign-2 E-mails Past Communications for each E-mail Status
(0 - Ignored, 1 - Read & 2 - Acknowledged)
0
10K
20K
30K
40K
50K
0
1
2
Email Status
Number of Campaign-3 E-mails Past Communications for each E-mail Status
(0 - Ignored, 1 - Read & 2 - Acknowledged)
0
5K
10K
15K
0
1
2
Email Status

Click here to download the data visualization.

15. Conclusion

The following conclusions were drawn from EDA:

- No e-mails of campaign type 1 got ignored.
- If campaign type is 1, then the mail has 66% chance of getting read and 23% change of getting acknowledged.
- Customer location or time of day does not affect the status of e-mail.
- As the number of previous communication increases, the chances of the e-mail being read or acknowledged also increases.
- E-mails tend to get ignored when word count is greater than 800.

The following conclusions were drawn from Modelling:

- Oversampled data seems to be better than undersampled data. This can be due to the fact that undersampling causes loss of information.
- The model built using XGBoost algorithm with SMOTE dataset performed better than the other models. It should be preferred for predicting mail statuses.
- If model interpretability is more important than accuracy, model built using logistic regression algorithm and SMOTE dataset should be chosen over the one using XGBoost algorithm. It is the best performer among the white box models.

References-

1. Kaggle
2. GeeksforGeeks
3. Machine Learning Mastery
4. Analytics Vidhya
5. Towards Data Science