CAPSTONE PROJECT - 3

Email Campaign Effectiveness Prediction

By

Bhaskar Mendhe
Karan Rawat

Points to Discuss:

- Agenda
- Data summary
- Data Reading
- Exploratory Data Analysis
- Data Preparation before Modelling
- Handling Class Imbalance
- Metrics Selection
- Model Explainability and Feature Importance
- Modelling
- Feature Engineering and Modelling
- Final Selected Model
- Summary
- Conclusion
- Challenges Faced

Agenda

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 that is ignored ; read ; acknowledged by the reader.

Data Summary

The dataset contains information related to emails like total number of past communications between business owners and their customers, total links attached etc.

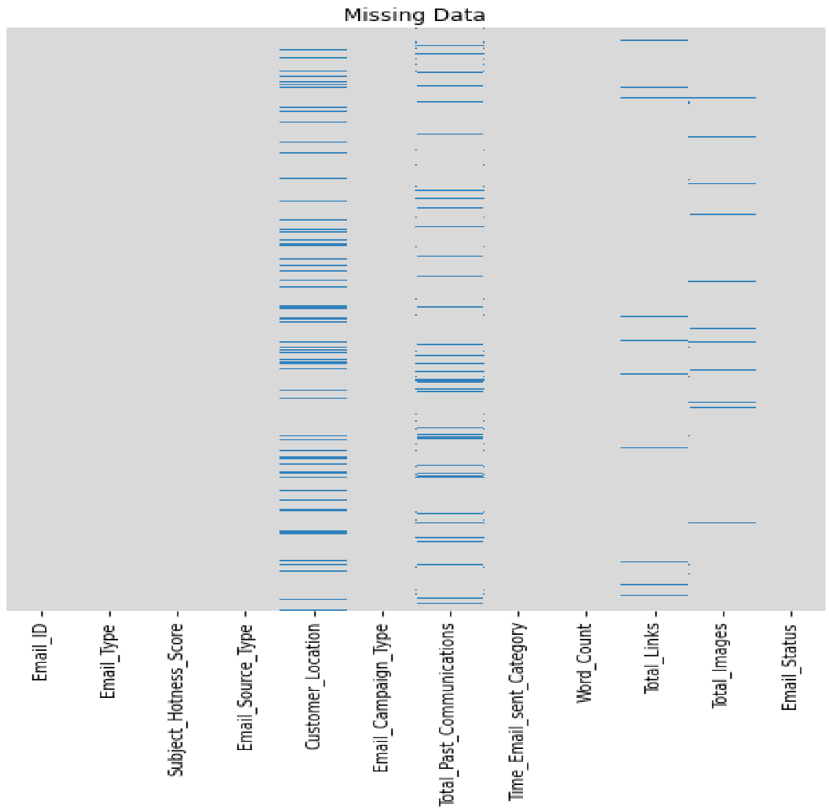
Attribute Information:

- **Email_ID** - Email ids of the customers.
- **Email_type** - There are 2 categories 1 and 2. We can think of them as marketing email or important updates,notices like emails regarding the business.
- **Subject_Hotness_Score** - Emails subject scores on the basis of how good and effective the content is.
- **Email_Source** - Represents the source of the email like sales,marketing or product type email.
- **Email_Campaign_Type** - The campaign type of the emails.
- **Total_Past_Communications** - The number of previous mails from the same source, the number of communications had.
- **Customer_Location** - Contains demographical data of the customers, the location where the customer resides.
- **Time_Email_sent_Category** - It has 3 categories: 1,2 and 3 which are considered as morning,evening and night time slots.
- **Word_Count** - The number of words contained in the mail.
- **Total_Links** - The number of links in the email.
- **Total_Images** - The number of images in the email.
- **Email_Status** - The target variable which contains whether the mail was is ignored; read; acknowledged by the reader.

Data Reading

- There are total 12 features in the dataset out of which the feature "Email_Status " is a response variable and rest are predictor variables and total 68353 entries.
- the features "Email_Type", "Email_Source_Type", "Email_Campaign_Type" and "Time_Email_sent_Category" contains categorical values but these features mapped to wrong data type, so surely we have to look upon this.
- The "Email_ID" feature contains identity information so we can drop this feature for further procedures.

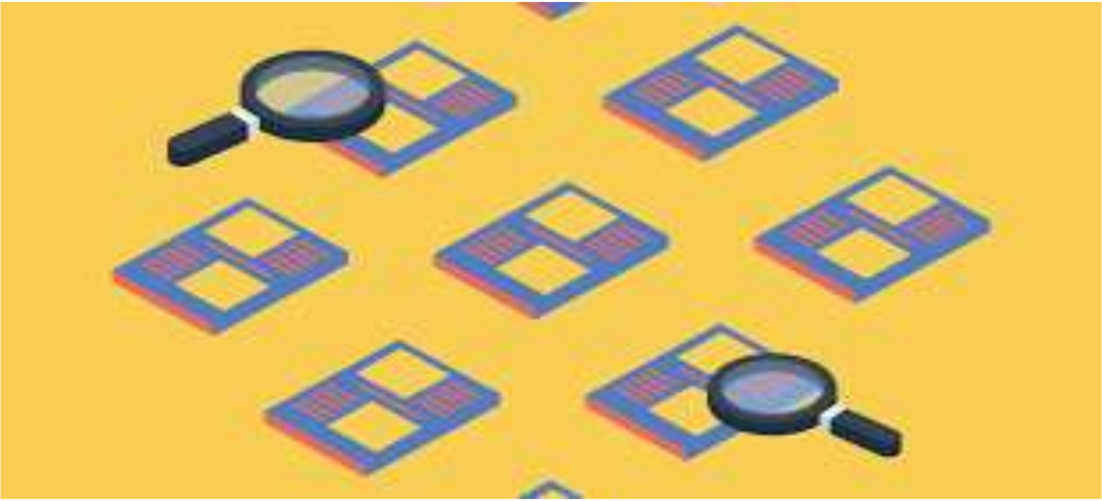
● Checking Missing Values



The features "Customer_Location", "Total_Past_Communications", "Total_Links" and "Total_Images" have missing values. We need to take care of these missing values.

- **Checking Duplicate Rows**

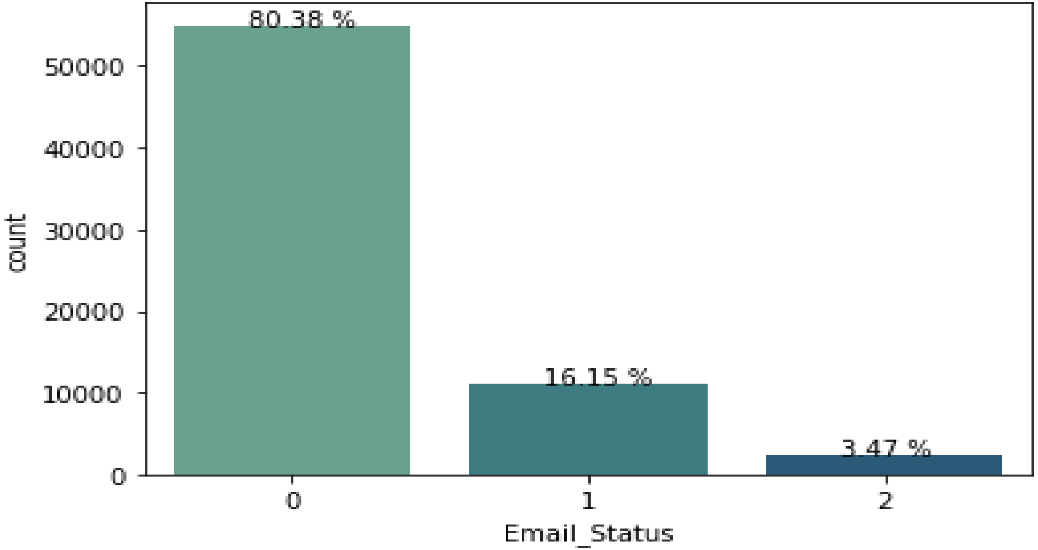
There are no duplicate rows in the dataset.



Exploratory Data Analysis

- **Response Variable**

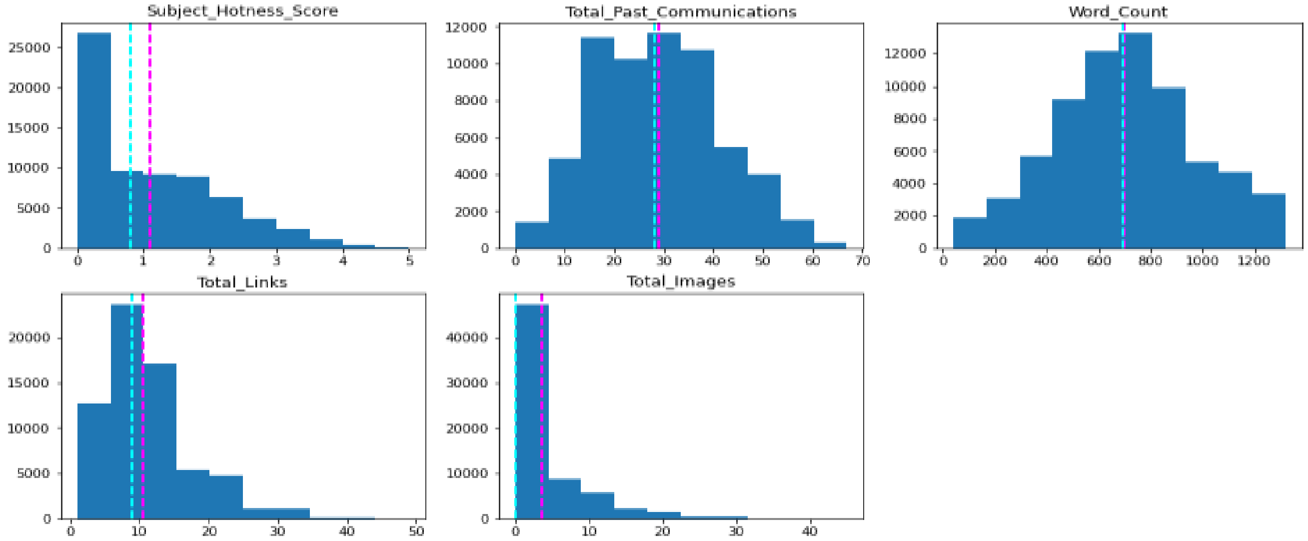
  There is high imbalance in class distribution of response variable. The majority of the data, 54941 data points which is 80.38 % belongs to "class 0", 11039 points which is 16.15 % belongs to class "1" and very small amount of data, 2373 data points which is 3.47% belongs to "class 2". We need to take care of this class imbalance.



- **Predictor Variables**
    - The features "Email_Type", "Email_Source_Type", "Email_Campaign_Type" and "Time_Email_sent_Category" contains categorical information , so we will change the datatype according to that.
    - There are total five numerical and categorical features each
    - Numerical Features – 'Subject_Hotness_Score', 'Total_Past_Communications', 'Word_Count', 'Total_Links', 'Total_Images'
    - Categorical Features – 'Email_Type', 'Email_Source_Type', 'Customer_Location', 'Email_Campaign_Type', 'Time_Email_sent_Category'

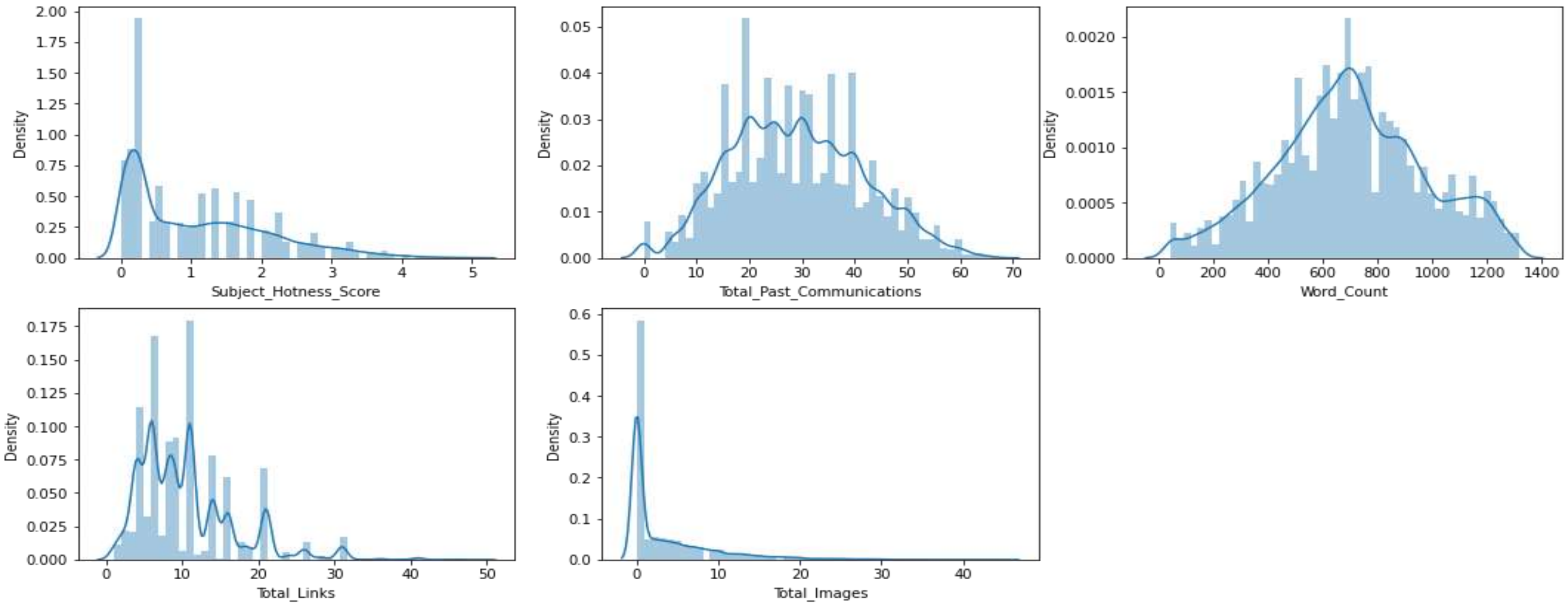
- **Numerical Features**

  - **Visualising using histogram plot.**



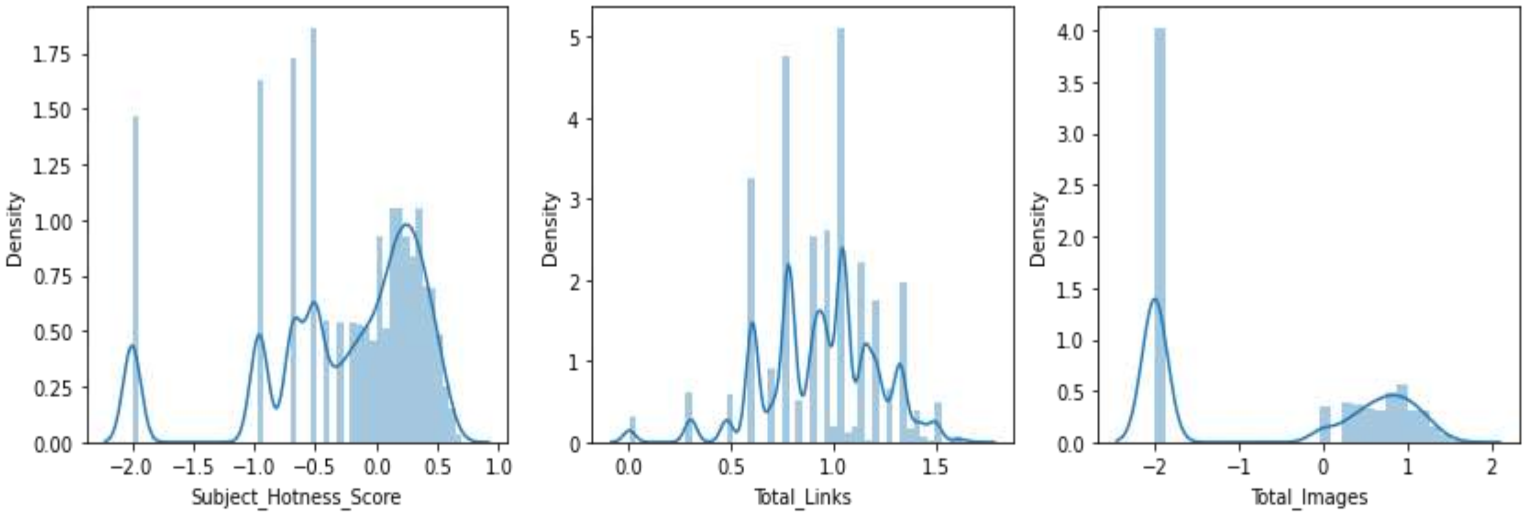
The mean and median for all the features except 'Word_Count' differs so surely these features contains outliers.The features 'Total_Past_Communications' and 'Word_Count' are distributed with good variance.

- # Visualising using distribution plots

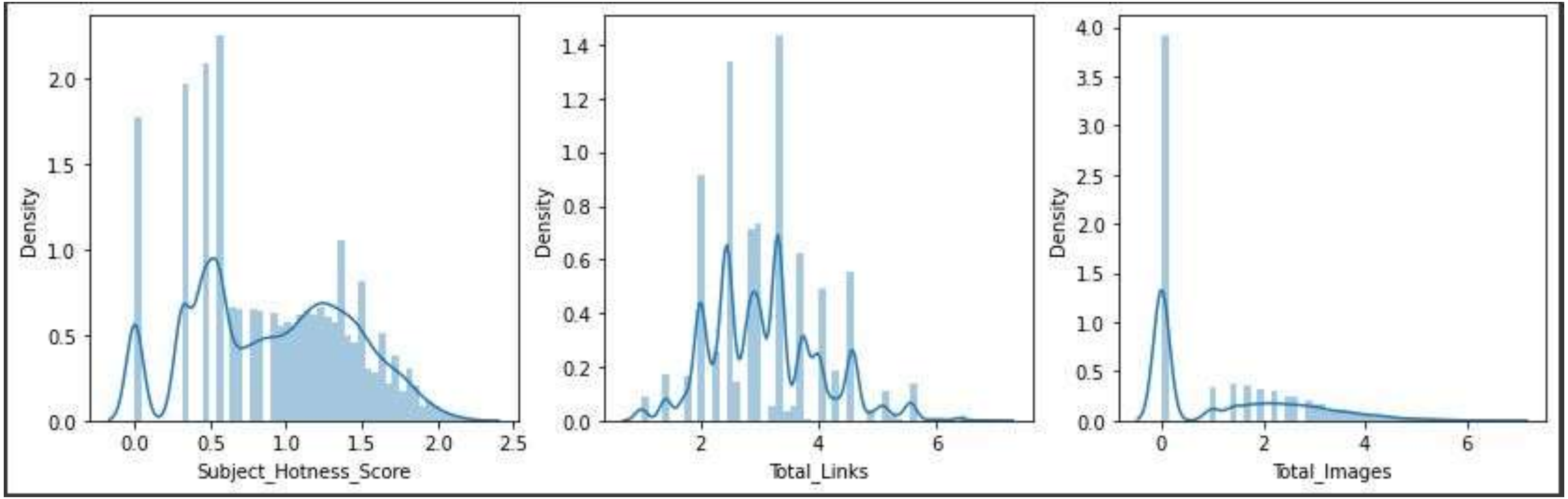


The features 'Subject_Hotness_Score','Total_Links' and 'Total_Images' are positively skewed..

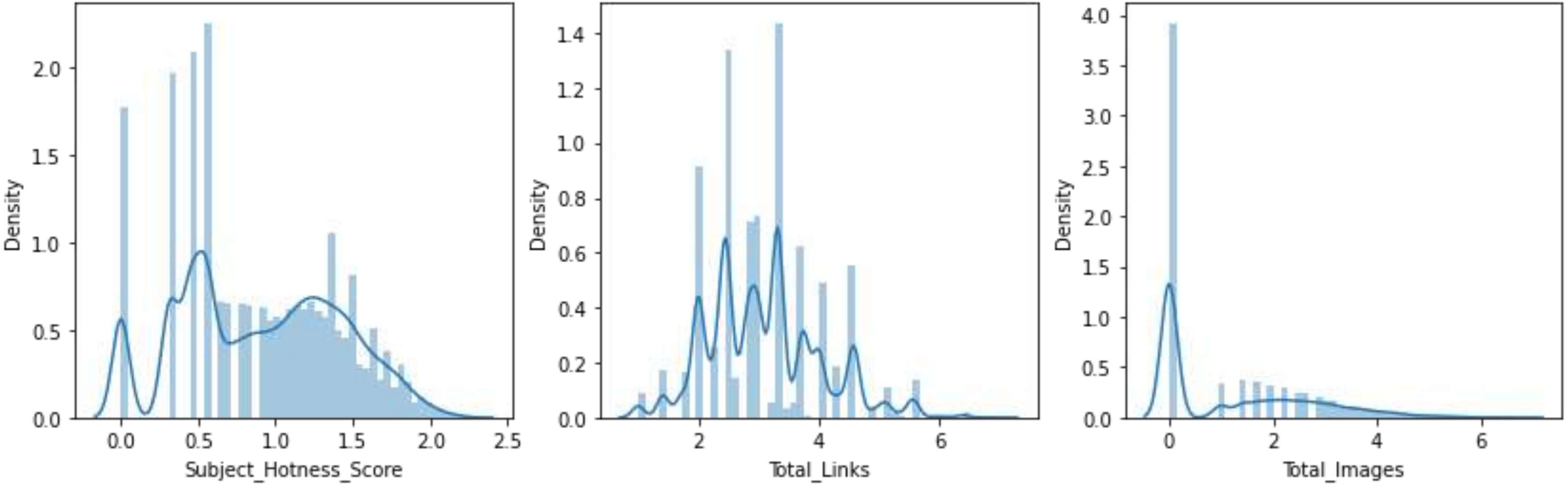
- **Let's try to find the solution to fix the skewed features.**
  - **Log Transformation**



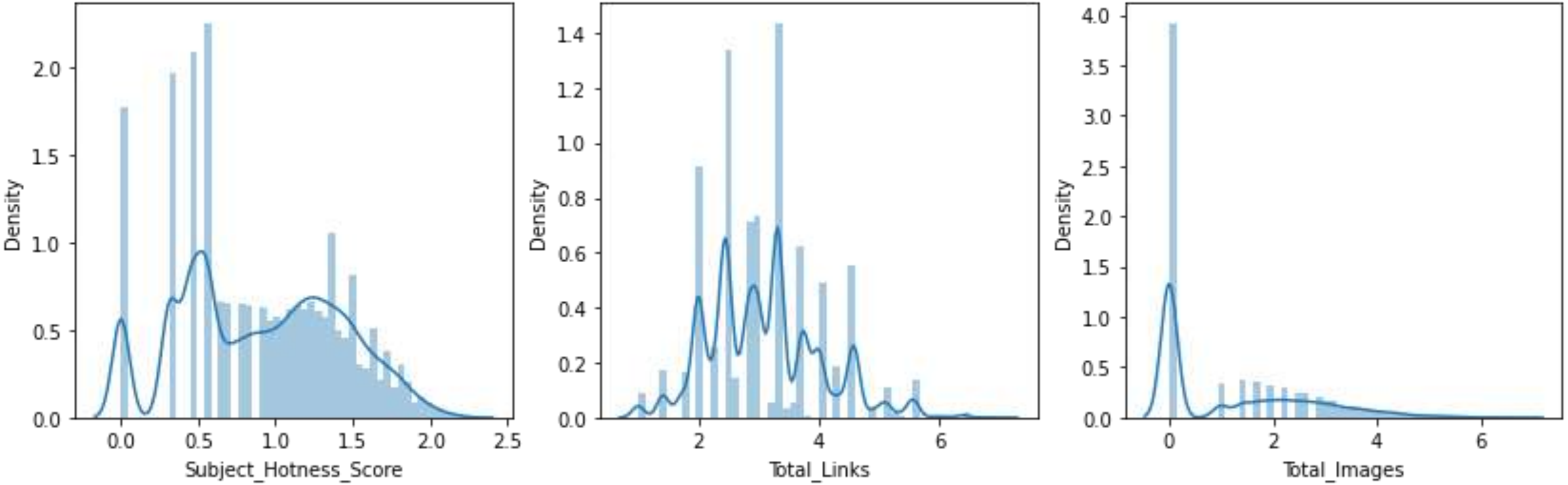
- ## Square Root Transformation



- **Cube Root Transformation**



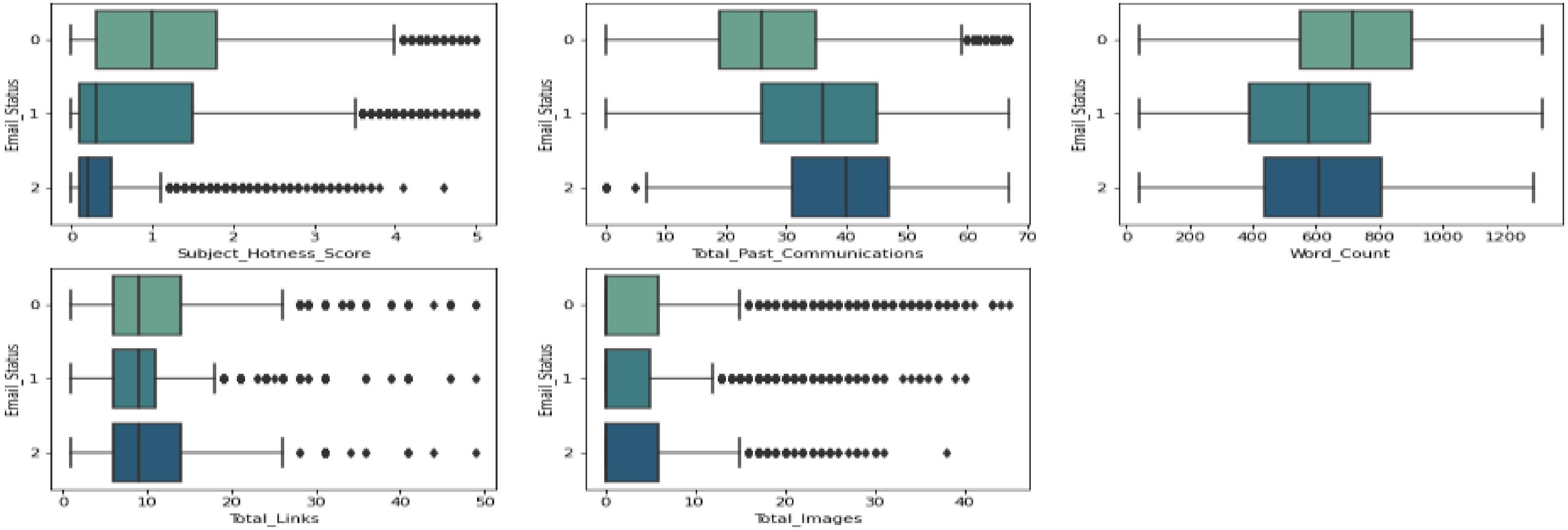
Power Transformation



The Log, Square Root and Cube Root transformations used are able to remove the skewness , but Power transformation removed skewness outstandingly and also it standardize the data as well. Hence, we will use power transformation for the all numerical features which are skewed and non-skewed as it removes the skewness and standardize the features as well.

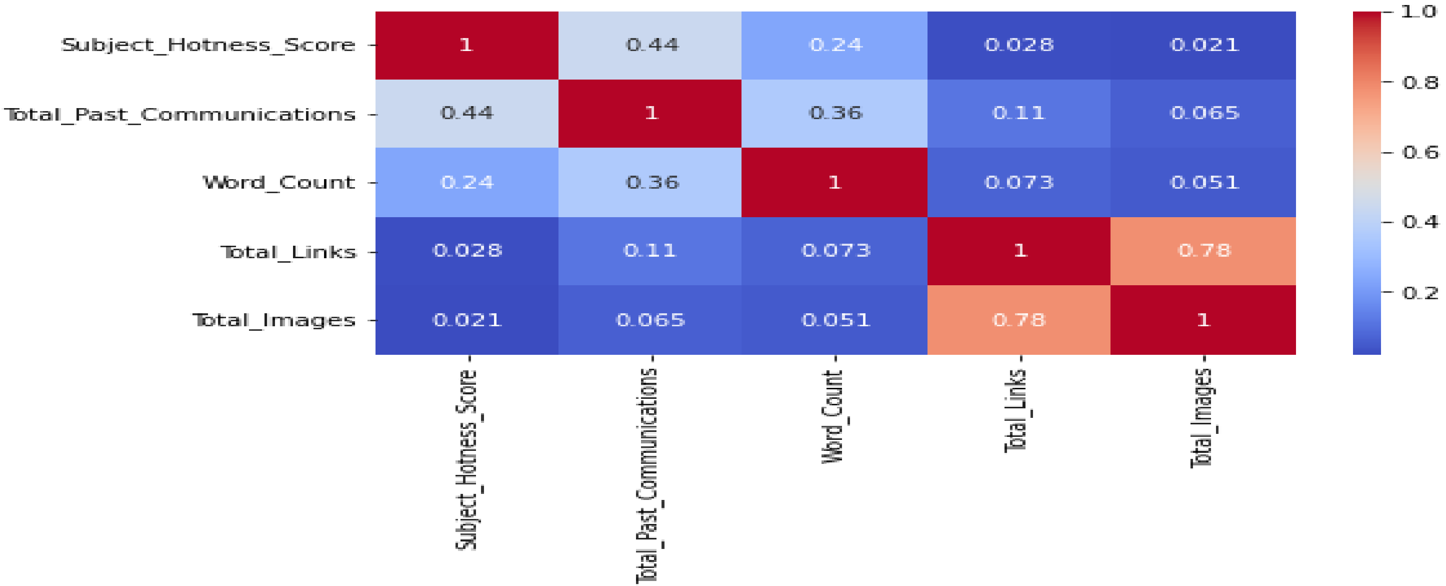
- ## Check for Outliers

The outliers being checked with respect to response variable.



All the numerical features except 'Word_Count' contains outliers. So we need to take care of these outliers.

- **Check for Correlation among features**



The features "Total_Images" and "Total_Links" are highly correlated with correlation value as 0.78.

- **Check for VIF values**

Variables	VIF
Subject_Hotness_Score	1.812747
Total_Past_Communications	3.764936
Word_Count	3.992614
Total_Links	9.525929
Total_Images	3.574892

The "Total_Links" feature have high VIF value.

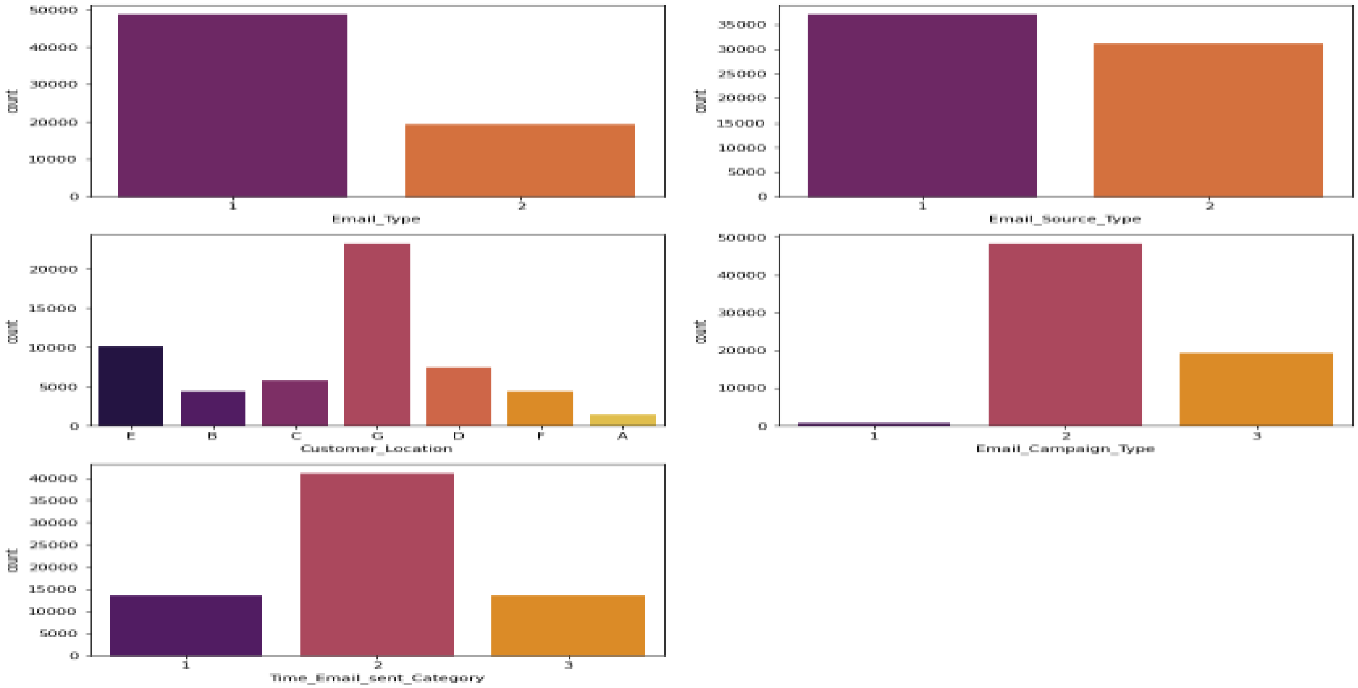
- Let's calculate VIF values without "Total_Links" feature.

Variables	VIF
Subject_Hotness_Score	1.694566
Total_Past_Communications	2.958079
Word_Count	3.347659
Total_Images	1.421444

The VIF values get reduced. So we should either drop "Total_Links" feature or use a combination of the features "Total_Links" and "Total_Images".

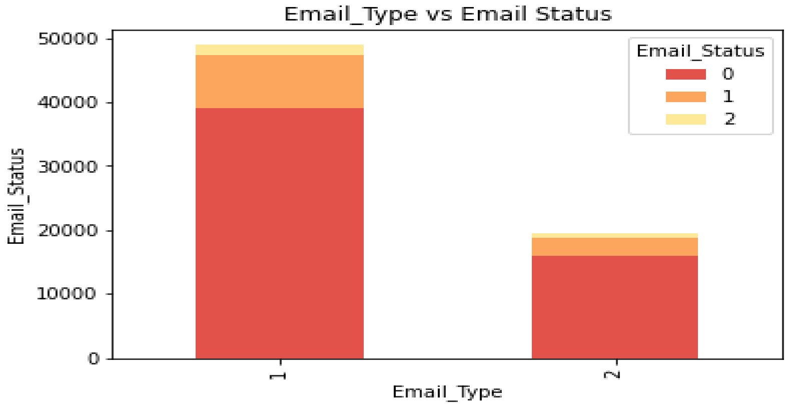
Categorical Features

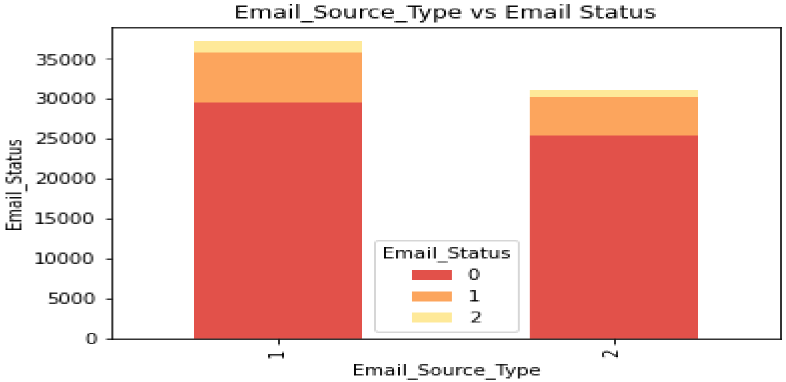
Visualising using bar plot

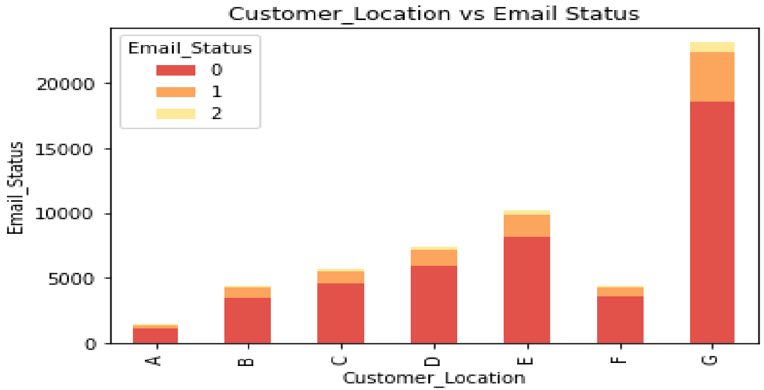


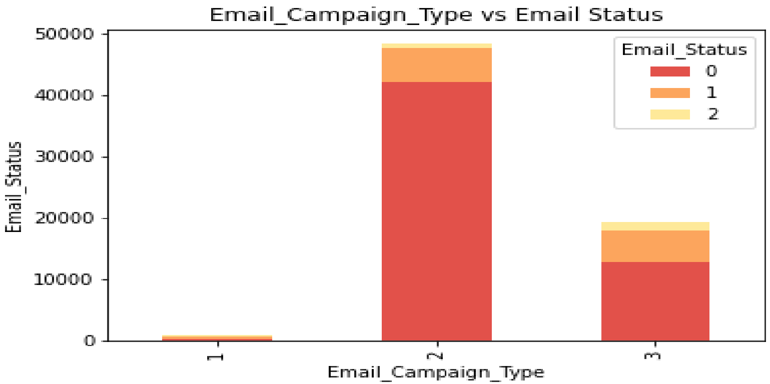
- For the feature "Email_Type" ,there are more number of points of Type 1 rather than Type 2.
- For the feature "Email_Source_Type",there is slight difference in distribution of points of Type 1 and Type 2.
- For the feature "Customer_Location", majority of the points belongs to category G.
- For the feature "Email_Campaign_Type" , majority of the points belongs to Type 2.
- For the feature "Time_Email_Sent_Category" also, majority of the points belongs to Type 2.

- **Checking categorical variables distribution with respect to response variable**

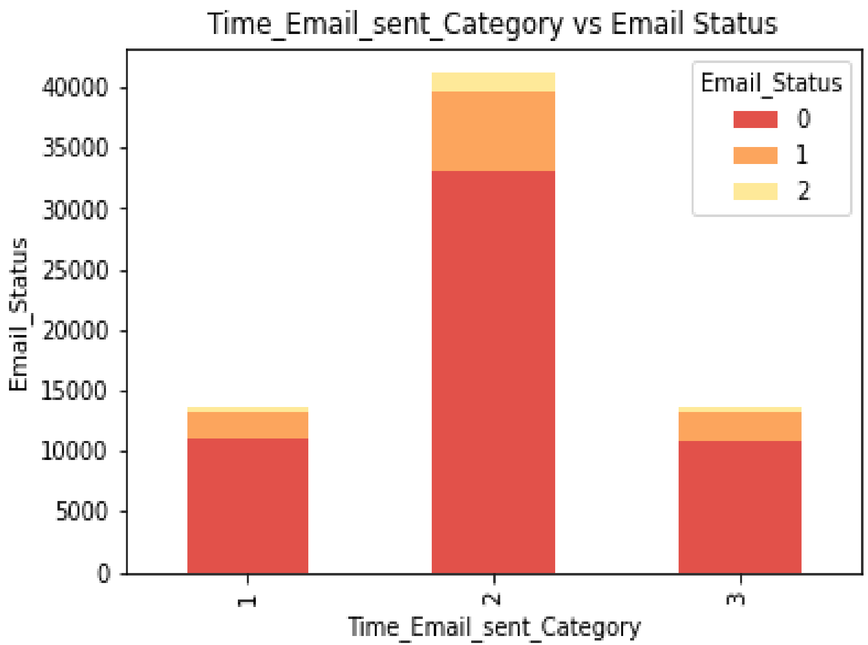








- The distribution of Email_Status is similar in all categorical features except "Email_campaign_Type".There are more number of points related to majority class in each feature.For "Email_campaign_Type" as Type 1 ,the distribution of points w.r.t to classes can be seen similar, as there are very less number of points for the same.



Data Preparation before Modelling

- The "Email_Id" feature contains identity information. So we have dropped this feature.
- As there is high class imbalance and we have very few points of class "2" and class "1" , so we will remove only those outliers which belongs to class "0".
- We have used Iterative Imputer to fill missing values of numerical features and treating missing values of categorical feature as separate category using Simple Imputer.
- To remove collinearity combination of collinear features as a single feature have been taken.
- The dataset is split into 80% train and 20% test.
- The Power Transformer is used to transform and standardize the numerical features.
- One Hot Encoder is used to encode the categorical features as these features are nominal in nature.
- The final train set has 50075 rows and 18 columns ,and final test set has 12519 rows and 18 columns.

Handling Class Imbalance

- There is high class imbalance in the dataset. To solve this we will provide different weights to both the majority and minority classes. The difference in weights will influence the classification of the classes during the training phase. The whole purpose is to penalize the misclassification made by the minority class by setting a higher class weight and at the same time reducing weight for the majority class.
- The weights can be assigned according to classes simply by using parameter "class_weight" as "balanced" while defining the machine learning models.

Metrics Selection

There is high imbalance of classes in the dataset, and also our objective is to classify the mails as ignored ; read and acknowledged as correctly as possible to corresponding classes.So for this task we will look upon weighted Precision,Recall and F1 score, as Precision and Recall account for true positives which is nothing but correctly classified points belonging to respective classes and F1 score is just harmonic mean of these two which is a combined single metric to look for. We will also look for ROC_AUC score. In final we look upon below metics.

- Precision Score
- Recall Score
- F1 Score
- ROC AUC Score

- **Precision Score** : Precision is defined as the ratio of correctly classified positive samples (True Positive) to a total number of classified positive samples
- **Recall Score** : The recall is calculated as the ratio between the numbers of Positive samples correctly classified as Positive to the total number of Positive samples.
- **F1 Score** : The F1-score combines the precision and recall of a classifier into a single metric by taking their harmonic mean.
- **ROC_AUC_Score :** ROC is a probability curve and AUC represents the degree or measure of separability. It tells how much the model is capable of distinguishing between classes.

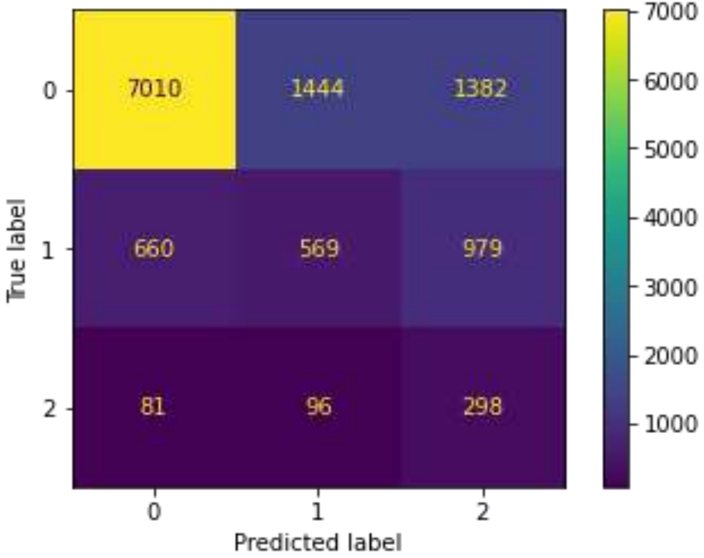
Model Explainability and Feature Importance

- The objective is to classify the mails as ignored ; read and acknowledged. For this , we need to find the reasons why the mails being ignored , how many being read and finally how many being acknowledged.So we need a model which can explain the reasons for classifications , so that we can improve the content for the mails such that mails could get read and acknowledged in the future which helps the owners stay connected with their prospective customers.
- The feature importance is also very important in this case as we need to know which are the most important features for classification, so we can focus on those to improve the content for mails.

Modelling

- **Logistic Regression**

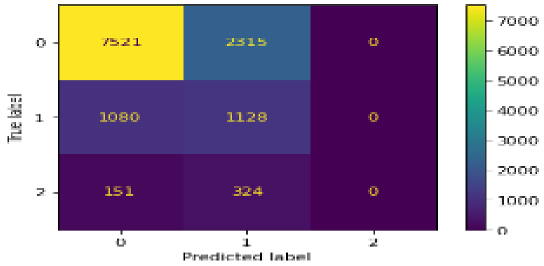
- The training and test scores of tuned Logistic Regression don't differ much which is a good sign ,shows that model is not overfitting or underfitting..
- It gives score of precision as 0.762 and roc_auc as 0.779 but recall score is low which is 0.629 and thus f1 score is also low which is 0.68.
- It is able to correctly classify 7010 points out of 9836 of class 1.
- It is able to correctly classify 298 points out of 475 of class 2, but it still classified only 569 points out of 2208.



Naive Bayes Classifier

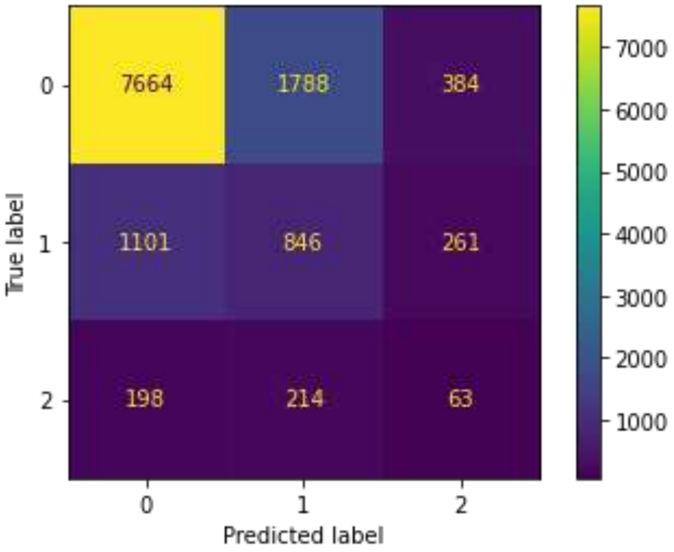
* To handle class imbalance in NB we will simply ignore the class prior probabilities , and to that we will use ComplementNB instead of Gaussian NB.
* The Complement NB assumes that numerical features comes from Multinomial Distribution that means these features can't contain negative values, so to keep this in mind we will normalize the data using MinMaxScaler instead of using standardization (using Power Transformer).
* To remove skewness from numerical features we will use cube root transformation before normalization , as we have seen in EDA above that cube root transformation remove skewness well.

- The training and test scores for tuned Naive Bayes don't differ much so the model is not overfitting or underfitting.
- The Naive Bayes gives test score of precision as 0.727 and roc_auc as 0.714 ,recall score as 0.69, and f1 score as 0.71.
- It is able to correctly classify 7521 points out of 9836 of class 0.
- It is able to correctly classify 1128 points out of 2208 of class 1 which is a great job done , but no points of class 2 have been classified which is not a good sign.



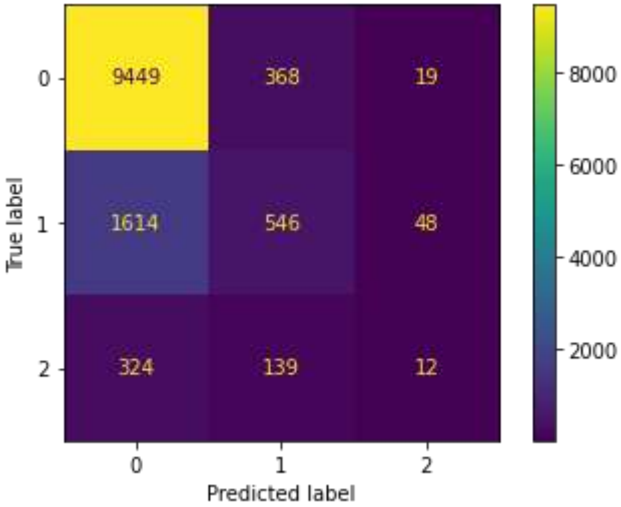
- **Decision Tree Classifier**

- The difference between training and test scores of tuned Decision Tree are still very large . The model gives very high scores for train set but not for test set , so it is clearly overfitting.
- The Decision Tree gives test score of precision as 0.727 and roc_auc as 0.646 ,recall score as 0.684, and f1 score as 0.703.
- It is able to correctly classify 7664 points out of 9836 of class 0.
- It is able to correctly classify 846 points out of 2208 of class 1 and only 63 points out of 475 of class 2.



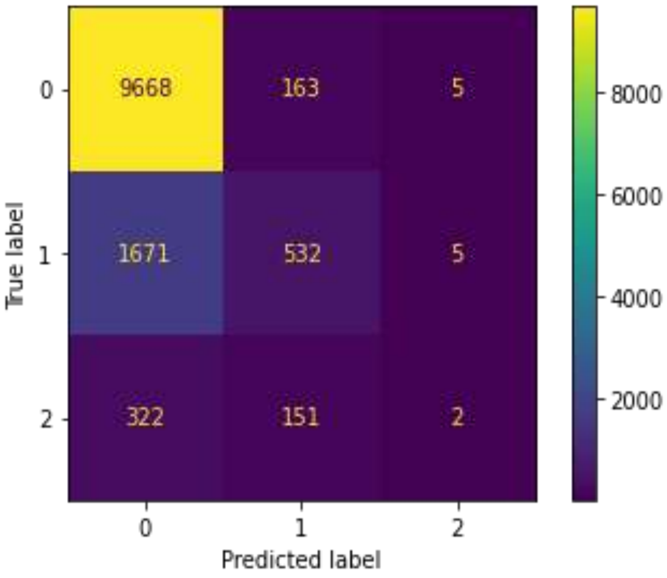
- # Random Forest Classifier

- The difference between training and test scores of tuned Random Forest are very large. The model gives very high scores for train set but not for test set , so it is clearly overfitting.
- The Random Forest gives test score of precision as 0.749 and roc_auc as 0.782 ,recall score as 0.799, and f1 score as 0.760.
- It is able to correctly classify 9449 points out of 9836 of class 0 which is great job done.
- It is able to correctly classify 546 points out of 2208 of class 1 and only 12 points out of 475 of class 2.



XGBoost Classifier

- The training and test scores for tuned XGBoost don't differ much so the model is not overfitting or underfitting.
- The XGBoost gives test score of precision as 0.768 and roc_auc as 0.819 ,recall score as 0.814, and f1 score as 0.768.
- It is able to correctly classify 9668 points out of 9836 of class 0 which is a great job done.
- It is able to correctly classify 532 points out of 2208 of class 1, but only 2 points of class 2 have been classified which is not a good sign.



Model Performance Comparison

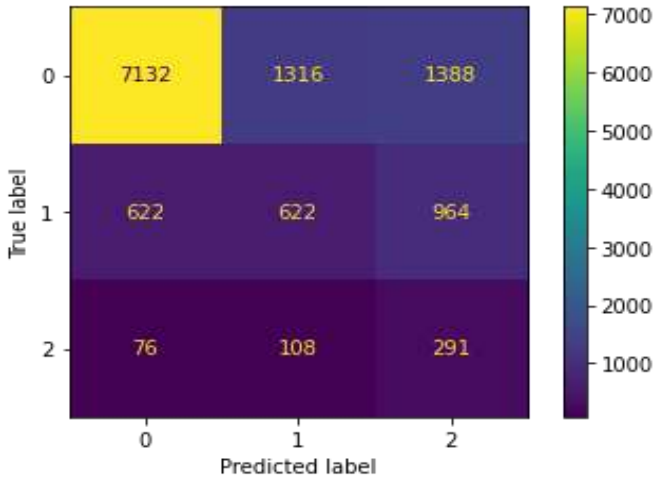
Model (tuned)	Train Precision	Test Precision	Train Recall	Test Recall	Train F1	Test F1	Train ROC_AUC	Test ROC_AUC
Logistic Regression	0.763	0.762	0.628	0.628	0.679	0.679	0.775	0.779
Naive Bayes Classifier	0.728	0.727	0.693	0.690	0.704	0.702	0.718	0.714
Decision Tree Classifier	0.943	0.727	0.923	0.684	0.928	0.703	0.989	0.646
Random Forest Classifier	0.999	0.749	0.999	0.799	0.999	0.760	0.999	0.782
XGBoost Classifier	0.797	0.768	0.819	0.814	0.775	0.768	0.828	0.819

Feature Engineering and Modelling

- Transforming numerical features into combination of polynomials of degree 5 and then training models.
- The final train set has 50075 rows and 144 columns ,and final test set has 12519 rows and 144 columns.

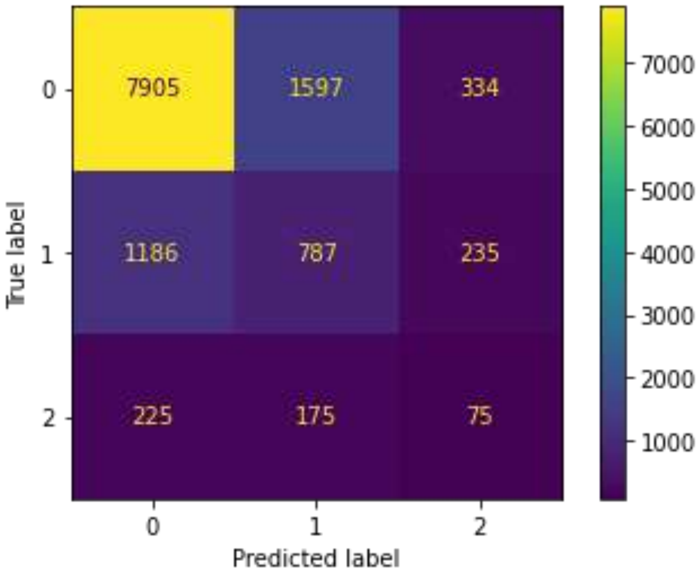
- **Logistic Regression**

- The training and test scores of tuned Logistic Regression don't differ much which is a good sign.
- It gives score of precision as 0.773 and roc_auc as 0.791 but recall score is low which is 0.642 and thus f1 score is also low which is 0.693.
- It is able to correctly classify 7132 points out of 9836 of class 1.
- It is able to correctly classify 291 points out of 475 of class 2, but it still classified only 622 points out of 2208.



Decision Tree Classifier

- The difference between training and test scores of tuned Decision Tree are still very large . The model gives very high scores for train set but not for test set , so it is clearly overfitting.
- The Decision Tree gives test score of precision as 0.725 and roc_auc as 0.632 ,recall score as 0.70, and f1 score as 0.711.
- It is able to correctly classify 7905 points out of 9836 of class 0.
- It is able to correctly classify 787 points out of 2208 of class 1 and only 75 points out of 475 of class 2.

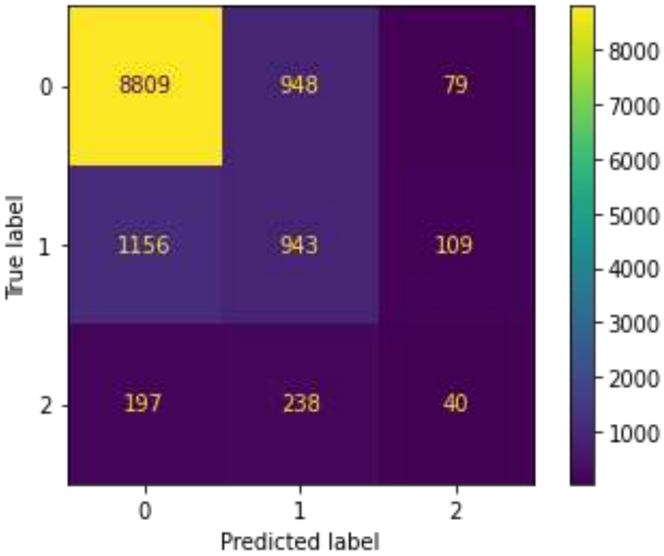


Note :

- Due to increase in dimensionality, the Decision Trees training time complexity increase and thus ,the Ensemble Models training time complexity increases so for these models we will only take interacted features.
- So for this we have reduced the number of combinations of features by passing “interaction_only” parameter as “True” in PolynomialFeatures.
- The final train set has 50075 rows and 34 columns ,and final test set has 12519 rows and 34 columns.

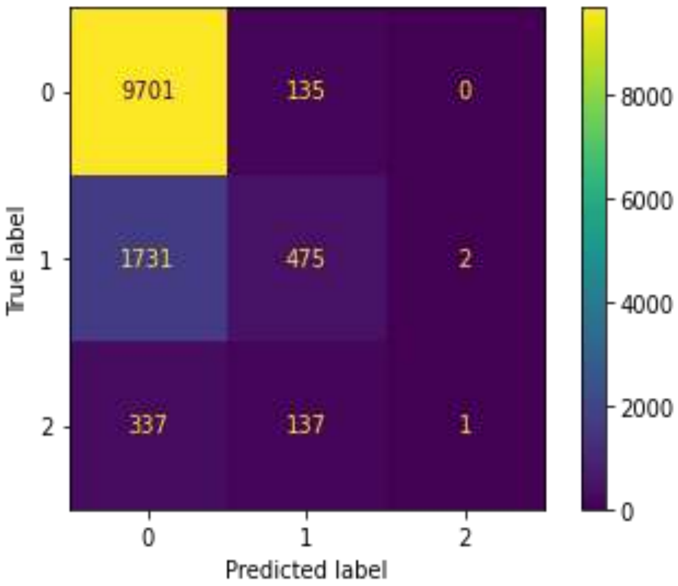
Random Forest Classifier

- The difference between training and test scores of tuned Random Forest are very large. The model gives very high scores for train set but not for test set , so it is clearly overfitting.
- The Random Forest gives test score of precision as 0.765 and roc_auc as 0.801 ,recall score as 0.782, and f1 score as 0.773.
- It is able to correctly classify 8809 points out of 9836 of class 0 which is great job done.
- It is able to correctly classify 943 points out of 2208 of class 1 and only 40 points out of 475 of class 2.



XGBoost Classifier

- The training and test scores for tuned XGBoost don't differ much so the model is not overfitting or underfitting.
- The XGBoost gives test score of precision as 0.772 and roc_auc as 0.816 ,recall score as 0.812, and f1 score as 0.762.
- It is able to correctly classify 9701 points out of 9836 of class 0 which is a great job done.
- It is able to correctly classify 475 points out of 2208 of class 1, but only 1 point of class 2 have been classified which is not a good sign.



Model Performance Comparison after Feature Engineering

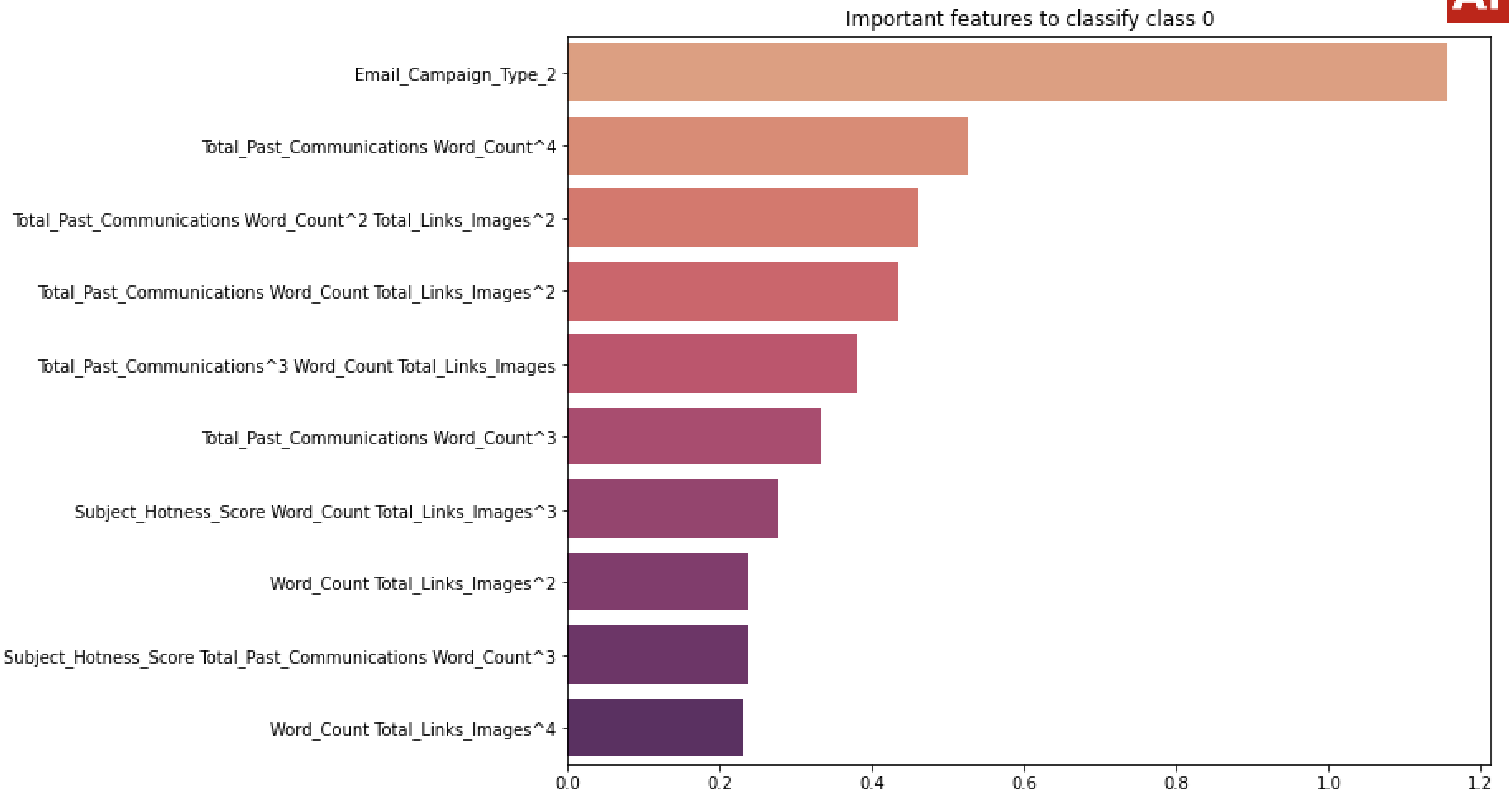
Model (tuned)	Train Precision	Test Precision	Train Recall	Test Recall	Train F1	Test F1	Train ROC_AUC	Test ROC_AUC
Logistic Regression	0.770	0.773	0.642	0.642	0.690	0.693	0.785	0.791
Decision Tree Classifier	0.968	0.725	0.963	0.700	0.964	0.711	0.997	0.632
Random Forest Classifier	0.961	0.765	0.957	0.782	0.958	0.773	0.986	0.801
XGBoost Classifier	0.797	0.772	0.815	0.812	0.766	0.762	0.819	0.816

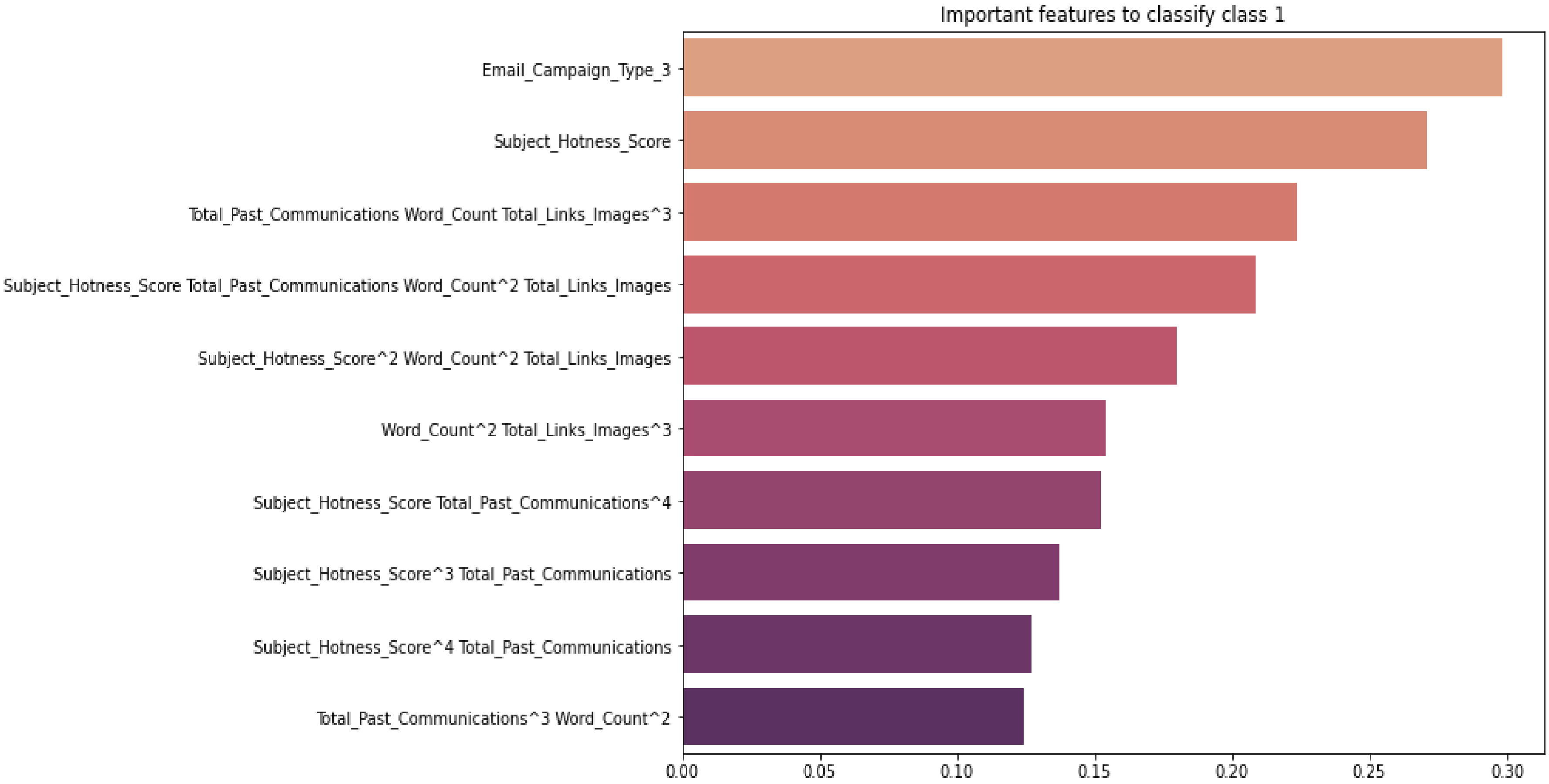
Final Selected Model

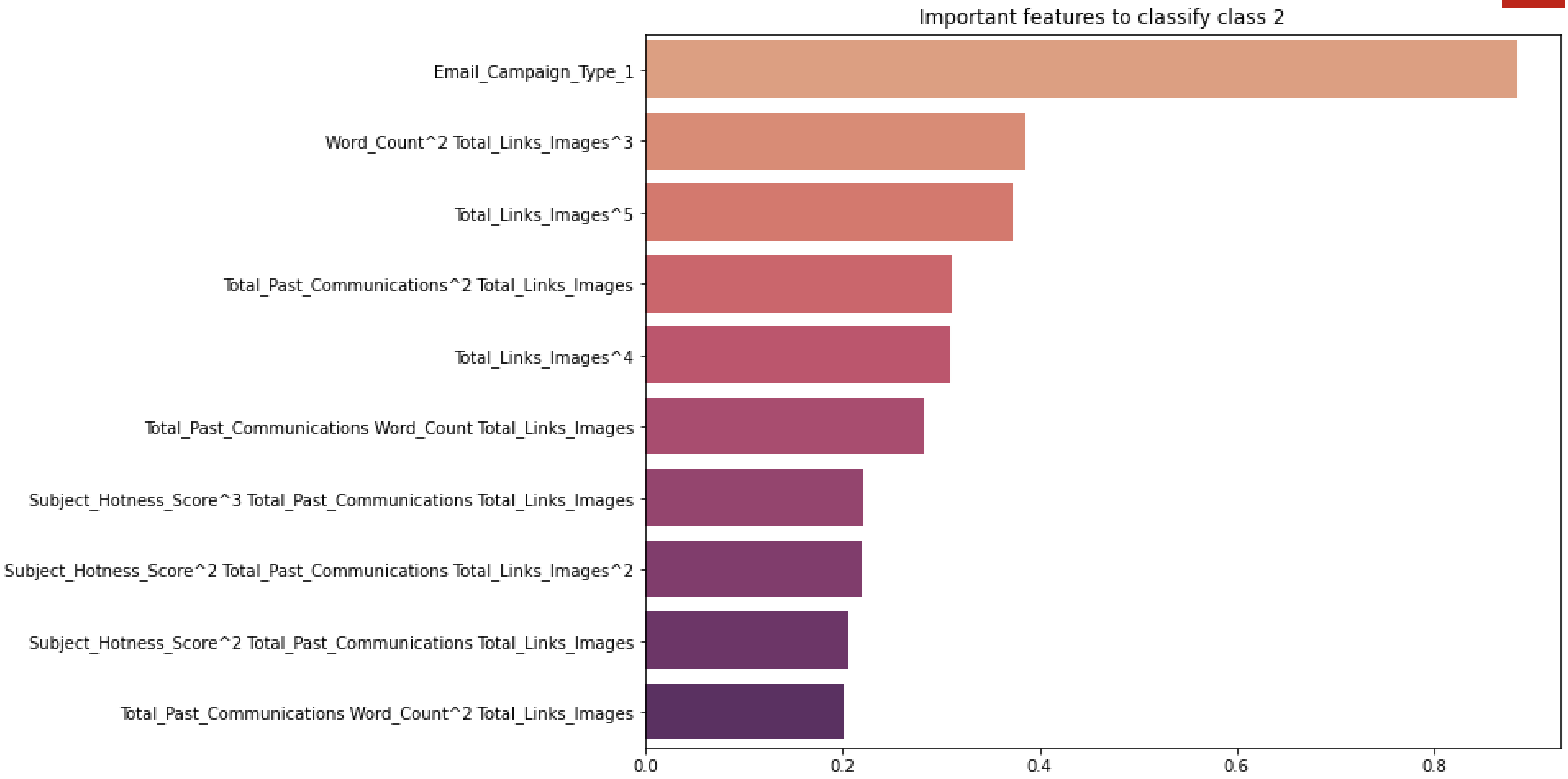
Taking Scores into consideration, XGBoost outperforms all models , but it hardly classify 1 or 2 points correctly from the minority class. At other hand Logistic Regression being simplest model able to correctly classify most number of points from minority class other than any model. Also Logistic Regression is very easy to interpret , as it fits a hyperplane for a classification.So the final model selected is Logistic Regression

Feature Importance

- We can derive feature importance from the coefficient of Logistic Regression.
- Note that coefficient will be of shape - (number of classes,total_features).
- To get the features which are important to classify points of class 0, we will look at the first array of coefficient. The more larger values of coefficients corresponding to features , then more the feature is important.
- Similarly we will fetch the features which are important for classification of points belong to class 1 and class 2.
- We will take top 10 most important features for visualization.


Important features to classify class 0
Email_Campaign_Type_2
Total_Past_Communications Word_Count^4
Total_Past_Communications Word_Count^2 Total_Links_Images^2
Total_Past_Communications Word_Count Total_Links_Images^2
Total_Past_Communications^3 Word_Count Total_Links_Images
Total_Past_Communications Word_Count^3
Subject_Hotness_Score Word_Count Total_Links_Images^3
Word_Count Total_Links_Images^2
Subject_Hotness_Score Total_Past_Communications Word_Count^3
Word_Count Total_Links_Images^4
0.0
0.2
0.4
0.6
0.8
1.0
1.2


Important features to classify class 1
Email_Campaign_Type_3
Subject_Hotness_Score
Total_Past_Communications Word_Count Total_Links_Images^3
Subject_Hotness_Score Total_Past_Communications Word_Count^2 Total_Links_Images
Subject_Hotness_Score^2 Word_Count^2 Total_Links_Images
Word_Count^2 Total_Links_Images^3
Subject_Hotness_Score Total_Past_Communications^4
Subject_Hotness_Score^3 Total_Past_Communications
Subject_Hotness_Score^4 Total_Past_Communications
Total_Past_Communications^3 Word_Count^2
0.00
0.05
0.10
0.15
0.20
0.25
0.30


Important features to classify class 2
Email_Campaign_Type_1
Word_Count^2 Total_Links_Images^3
Total_Links_Images^5
Total_Past_Communications^2 Total_Links_Images
Total_Links_Images^4
Total_Past_Communications Word_Count Total_Links_Images
Subject_Hotness_Score^3 Total_Past_Communications Total_Links_Images
Subject_Hotness_Score^2 Total_Past_Communications Total_Links_Images^2
Subject_Hotness_Score^2 Total_Past_Communications Total_Links_Images
Total_Past_Communications Word_Count^2 Total_Links_Images
0.0
0.2
0.4
0.6
0.8

- The "Email_Campaign" as "Type_2" is the most individual important feature for classification of class 0 points and rest important features are the combination of the polynomial of features.
- The "Email_Campaign" as "Type_3" and "Subject_Hotness_Score" are the most important individual features for classification of class 1 points and rest important features are the combination of the polynomial of features.
- The "Email_Campaign" as "Type_1" and "Total_Link_Images" are the most important individual features for classification of class 2 points and rest important features are the combination of the polynomial of features.
- The "Email_Campaign" as a whole feature is the most important individual feature for classification. The other features "Total_Past_Communications","Word_Count","Subject_Hotness_Score","Total_Link_Images" which is combination of "Total_Links" and 'Total_Images", are next important features which are used in combinations of polynomials for classifications. So we can look upon these features to improve the business work.

Summary

- The objective is to create a machine learning model to characterize the mail that is ignored ; read ; acknowledged by the reader.
- There are total 12 features out of which the feature "Email_Status " is a response variable and rest are predictor variables.
- The features "Customer_Location", "Total_Past_Communications", "Total_Links" and "Total_Images" have missing values. We need to take care of these missing values.
- There are no duplicate rows in the dataset.
- There is high imbalance in class distribution of response variable. The majority of the data, 54941 data points which is 80.38 % belongs to "class 0", 11039 points which is 16.15 % belongs to class "1" and very small amount of data, 2373 data points which is 3.47% belongs to "class 2".
- The features "Email_Type", "Email_Source_Type", "Email_Campaign_Type" and "Time_Email_sent_Category" contains categorical information , so we have changed the datatype according to that.
- There were five categorical and numerical features each afterwards.
- The numerical features 'Subject_Hotness_Score','Total_Links' and 'Total_Images' were positively skewed.
- The Log, Square Root and Cube Root transformations used for removing skewness and these were able to remove the skewness , but Power transformation removed skewness outstandingly and also it standardize the data as well.
- All the numerical features except 'Word_Count' had outliers.
- The numerical features "Total_Images" and "Total_Links" were highly correlated with correlation value as 0.78.

- There are two subcategories under "Email_Type" and "Email_Source_Type" features, three subcategories under "Email_Campaign_Type" and "Time_Email_Sent_Category" features, and seven subcategories under "Customer_Location" feature.
- For the feature "Email_Type" ,there are more number of points of Type 1 rather than Type 2 .For "Email_Source_Type",there is slight difference in distribution of points of Type 1 and Type 2. For "Customer_Location", majority of the points belongs to category G. For "Email_Campaign_Type" , majority of the points belongs to Type 2. For the "Time_Email_Sent_Category" also, majority of the points belongs to Type 2.
- The distribution of Email_Status is similar in all categorical features except "Email_campaign_Type".There are more number of points related to majority class in each feature.For "Email_campaign_Type" as Type 1 ,the distribution of points w.r.t to classes can be seen similar, as there are very less number of points for the same.
- The "Email_Id" feature contains identity information. So dropping this feature.
- As there is high class imbalance and we have very few points of class "2" and class "1" , so we have removed only those outliers which belongs to class "1".
- We have used Iterative Imputer to fill missing values of numerical features and treating missing values of categorical feature as separate category using Simple Imputer.
- To remove collinearity combination of collinear features as a single feature have been taken.
- We have used Power Transformer for transforming numerical features as it helps in removing the skewness and standardizing as well and One hot Encoder for Categorical Features.
- The shape of Train Set becomes - (50075, 22) and Test Set Shape becomes =(12519, 22)
- There is high class imbalance in the dataset. To solve this we will provide different weights to both the majority and minority classes. The difference in weights will influence the classification of the classes during the training phase. The whole purpose is to penalize the misclassification made by the minority class by setting a higher class weight and at the same time reducing weight for the majority class. The weights can be assigned according to classes simply by using parameter "class_weight" as "balanced" while defining the machine learning models.

- There is high imbalance of classes in the dataset, and also our objective is to classify the mails as ignored ; read and acknowledged as correctly as possible to corresponding classes.So fo this task we will look upon weighted Precision,Recall and F1 score, as Precision and Recall account for true positives which is nothing but correctly classified points belonging to respective classes and F1 score is just harmonic mean of these two which is a combined single metric to look for. We will also look for ROC_AUC score.
- The objective is to classify the mails as ignored ; read and acknowledged. For this , we need to find the reasons why the mails being ignored , how many being read and finally how many being acknowledged.So wee need a model which can explain the reasons for classifications , so that we can improve the content for the mails such that mails could get read and acknowledged in the future which helps the owners stay connected with their prospective customers. The feature importance is also very important in this case as we need to know which are the most important features for classification, so we can focus on those to improve the content for mails.
- The training and test scores don't differ much for tuned Logistic Regression which is a good sign. It gives score of precision as 0.762 and roc_auc as 0.779 but recall score is low which is 0.629 and thus f1 score is also low which is 0.68. It is able to correctly classify 7010 points out of 9836 of class 1. It is able to correctly classify 298 points out of 475 of class 2, but it still classified only 569 points out of 2208.
- The training and test scores for tuned Naive Bayes don't differ much so the model is not overfitting or underfitting. The Naive Bayes gives test score of precision as 0.727 and roc_auc as 0.714 ,recall score as 0.69, and f1 score as 0.71. It is able to correctly classify 7521 points out of 9836 of class 0. It is able to correctly classify 1128 points out of 2208 of class 1 which is a great job done , but no points of class 2 have been classified which is not a good sign.
- The difference between training and test scores of tuned Decision Tree are very large . The model gives very high scores for train set but not for test set , so it is clearly overfitting. The Decision Tree gives test score of precision as 0.727 and roc_auc as 0.646 ,recall score as 0.684, and f1 score as 0.703. It is able to correctly classify 7664 points out of 9836 of class 0. It is able to correctly classify 846 points out of 2208 of class 1 and only 63 points out of 475 of class 2.
- The difference between training and test scores of tuned Random Forest are very large. The model gives very high scores for train set but not for test set , so it is clearly overfitting. The Random Forest gives test score of precision as 0.747 and roc_auc as 0.777 ,recall score as 0.798, and f1 score as 0.758. It is able to correctly classify 9450 points out of 9836 of class 0 which is great job done. It is able to correctly classify 531 points out of 2208 of class 1 and only 13 points out of 475 of class 2.

- The training and test scores for XGBoost don't differ much so the model is not overfitting or underfitting. The XGBoost gives test score of precision as 0.768 and roc_auc as 0.819 ,recall score as 0.814, and f1 score as 0.768. It is able to correctly classify 9668 points out of 9836 of class 0 which is a great job done. It is able to correctly classify 532 points out of 2208 of class 1, but only 2 points of class 2 have been classified which is not a good sign.
- Then we have done Feature engineering by transforming numerical features into polynomials of degree 5.
- The training and test scores for tuned Logistic Regression don't differ much which is a good sign. It gives score of precision as 0.773 and roc_auc as 0.791 but recall score is low which is 0.642 and thus f1 score is also low which is 0.693. It is able to correctly classify 7132 points out of 9836 of class 1. It is able to correctly classify 291 points out of 475 of class 2, but it still classified only 622 points out of 2208.
- The difference between training and test scores of Decision Tree are very large . The model gives very high scores for train set but not for test set , so it is clearly overfitting. The Decision Tree gives test score of precision as 0.725 and roc_auc as 0.632 ,recall score as 0.70, and f1 score as 0.711. It is able to correctly classify 7905 points out of 9836 of class 0. It is able to correctly classify 787 points out of 2208 of class 1 and only 75 points out of 475 of class 2.
- The difference between training and test scores of Random Forest are very large. The model gives very high scores for train set but not for test set , so it is clearly overfitting. The Random Forest gives test score of precision as 0.765 and roc_auc as 0.801 ,recall score as 0.782, and f1 score as 0.773. It is able to correctly classify 8809 points out of 9836 of class 0 which is great job done. It is able to correctly classify 943 points out of 2208 of class 1 and only 40 points out of 475 of class 2.
- The training and test scores for XGBoost don't differ much so the model is not overfitting or underfitting. The XGBoost gives test score of precision as 0.772 and roc_auc as 0.816 ,recall score as 0.812, and f1 score as 0.762. It is able to correctly classify 9701 points out of 9836 of class 0 which is a great job done. It is able to correctly classify 475 points out of 2208 of class 1, but only 1 point of class 2 have been classified which is not a good sign.

Conclusions

- The final model selected is Logistic Regression.
- Taking Scores into consideration, XGBoost outperforms all models , but it hardly classify 1 or 2 points correctly from the minority class. At other hand Logistic Regression being simplest model able to correctly classify most number of points from minority class other than any model. Also Logistic Regression is very easy to interpret , as it fits a hyperplane for a classification.
- Also We can derive feature importance from the coefficient of Logistic Regression very easily. To get the features which are important to classify points of class 0, we will look at the first array of coefficient. The more larger values of coefficients corresponding to features , then more the feature is important. Similarly we fetch the features which are important for classification of points belong to class 1 and class 2.
- The "Email_Campaign" as a whole feature is the most important individual feature for classification. The other features "Total_Past_Communications","Word_Count","Subject_Hotness_Score","Total_Link_Images" which is combination of "Total_Links" and 'Total_Images", are next important features which are used in combinations of polynomials for classifications. So we can look upon these features to improve the business work.

Challenges Faced

- The dataset contains missing values.
- The dataset contains outliers.
- Some of the features mapped to wrong datatype.
- The dataset contains correlated features also.

Thank You