Email Campaign Effectiveness Prediction (Multiclass Classification)

Umesh, Ifraz, Shailendra, Akram
Data science trainees,
Alma Better, Bangalore.

Abstract:

Email advertising is the act of sending promotional emails to customers in mass quantities. It commonly is to generate income or leads and it can include advertising. Most importantly, email marketing allows businesses to build relationships with leads, new customers, and past customers. It's a way to communicate directly to the customers in their inbox, at a time that is convenient for them. With the right messaging tone and strategies, emails are one of the most important marketing channels.

Email campaign effectiveness is a way of analyzing the kind of email campaigns being run by businesses to carry out their marketing and promotional agendas and hence they need to know how well the campaign is working.

The work here characterizes and predicts the emails if they are going to be ignored; read; acknowledged based on the various features related to the emails in the dataset and makes recommendations to lower the number of ignored emails.

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. 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. Data columns are self-explanatory.

Introduction

The competition in the commercial world is so tough these days. The customers have so many options to go to, it is important to keep good relations with your customers to stay on top of the game. We see a lot of marketing strategies around us and almost half of our time we are engaging with different kinds of promotional schemes and advertising. One of the major digital marketing strategies of businesses involves the use of emails. Email Marketing can be summarized as a marketing technique in which businesses stay connected with their customers through emails, making them aware about their new products, updates,

and important notices related to the products they are using.

Talking from a business's point of view, they'd think that the emails they are creating are special, even the best and people are going to be excited about these promotions, but this is not necessarily the case. They are just one of the many emails they are getting every single day. We all subscribe to many kinds of businesses through emails simply because it's required to do so these days, sometimes to get digital receipts of the things we bought or to get digital information about the businesses to stay updated. But many times, we are not interested in reading those kinds of emails due to several reasons - to name a few would be- no proper structure, too many images, too many links inside the mail, complex vocabulary used or simply too long emails.

In this problem statement, we will be trying to create machine learning models that characterize and predict whether the mail is ignored, read, or acknowledged by the reader. In addition to this, we will be trying to analyze and find all the features that are important for an email to not get ignored and based on that some recommendations are made.

Approach:

The approach followed here is to first check the sanctity of the data and then understand the features involved. The events followed were in our approach:

- **Understanding the Data**

- **Data cleaning and preprocessing:** Finding null values and imputing them with appropriate values.

- **Exploratory data analysis:** of categorical and continuous variables against our target variable.

- **Data manipulation:**

feature selection and engineering, handling multicollinearity with the help of VIF scores, feature scaling and encoding.

- **Handling Class Imbalance:**

Our dataset was highly imbalance with 80% majority, strategy was to be splitting the stratified dataset and under sampling and oversampling with SMOTE and SMOTETomek on the train sets only so that our test set remains unknown to the models

- **Modeling:**

Worked on an evaluation code which was frequently used to evaluate the same models on under sampled and oversampled data in one go, logistic regression, decision trees, random forest, KNN and XGB were run to evaluate the results and then concluded based on model performance and some recommendations were made to improve the numbers of read and acknowledged emails.

1.Understanding the Data:

First step involved is understanding the data and getting answers to some basic questions like; What is the data about? How many rows or observations are there in it? How many features are there in it? What are the data types? Are there any missing values? And anything that could be relevant and useful to our investigation. Let's just understand the dataset first and the terms involved before proceeding further.

Our dataset consists of 68353 observations (i.e., rows) and 12 features (columns) about the emails. The data types were of integer, float, and object in nature.

Let's define the features involved:

● **Email Id:** It contains the email id's of the customers/individuals

● **Email Type:** There are two categories 1 and 2. We can think of them as marketing emails or important updates, notices like emails regarding the business.

● **Subject Hotness Score:** It is the email's subject's score based on how good and effective the content is.

● **Email Source:** It represents the source of the email like sales and marketing, or important admin mails related to the product.

● **Email Campaign Type:** The campaign type of the email.

● **Total Past Communications:** This column contains the total previous mails from the same source, the number of communications had.

● **Customer Location:** Contains demographical data of the customer, the location where the customer resides.

● **Time Email sent Category:** It has three categories 1,2 and 3; the time of the day when the email was sent, we can think of it as morning, evening, and nighttime slots.

● **Word Count** - The number of words contained in the email.

● **Total links:** Number of links in the email.

● **Total Images:** Number of images in the email.

● **Email Status:** Our target variable which contains whether the mail was ignored, read, acknowledged by the reader.

2. Reasons for Customer Ignoring Email:

- **Long and Winded Emails:**

  Sending long winding emails does not only hamper you, but it

also affects your recipient’s time. Imagine spending an hour or two on a very detailed email only to arrive at a “yes and no” response. You not only waste your time, but also waste your recipient’s time which he may have dedicated to other causes or work that is highly urgent.

- **Too Complicated to Understand:**

A complicated email generates more questions and leads to unnecessary back-and-forth feedback. This is usually the case for big organizations with huge hierarchies. Similarly, the more complicated your emails are, the more people will start ignoring them because they take a lot of time and energy to address.

3. Data Cleaning and Preprocessing:

Handling missing values is an important skill in the data analysis process. If there are very few missing values compared to the size of the dataset, we may choose to drop rows that have missing values. Otherwise, it is better to replace them with appropriate values.

The dataset had a lot of nulls in the following columns:

- Customer Location (11595)
- Total Past Communications (6825)
- Total Links (2201)
- Total Images (1677)

But customer locations had a lot of them. Since it is a categorical column, and it is difficult to just impute them with our understanding of where the customer's location is, it was important to see how much it affected our target variable, whether a particular location has anything to do with it or it is not correlated at all. If a particular location influences the target variables and aids in getting it ignored or otherwise, it should be filled on a condition (on Email Status) row wise.

For the rest of the continuous variables, the distribution was plotted to get an idea on how to impute these variables.

In Total_Past_Comunication feature we have imputed mean of that feature because the distribution of that feature is kind of normal

The distribution plot of both Total Links and Total Images are skewed to the right.

It seems like most of the values of the Total Links in the column are between 0-10 and the number of images in most of the emails seems to be 0 or fewer than 3-4. Consequently, the longer tail in an asymmetrical distribution pulls the mean away from the most common values. The mean is greater than the median. The mean overestimates the most common values in the distribution and hence mode (value with highest frequency) is used in these cases, it is more robust to outlier effect and the same is done here.

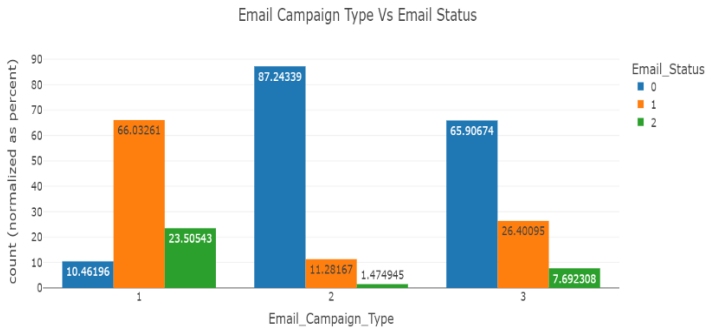
4. Exploratory Data Analysis:

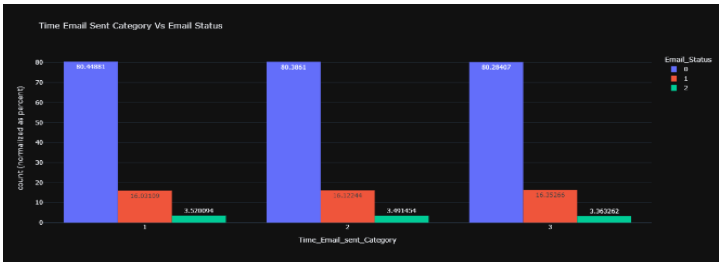
Exploratory data analysis is a crucial part of data analysis. It involves exploring and analyzing the dataset given to find out patterns, trends, and conclusions to make better decisions related to the data, often using statistical graphics and other data visualization tools to summarize the results. The visualization tools: Plotly, Matplotlib and seaborn. The goal here is to explore the relationships of different variables with "Email Status" to see what factors might be contributing to ignored emails and then be able to correctly characterize the three of them.

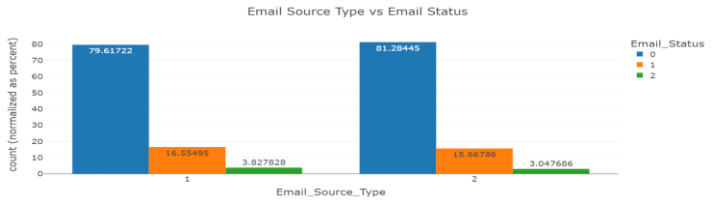
Approach:

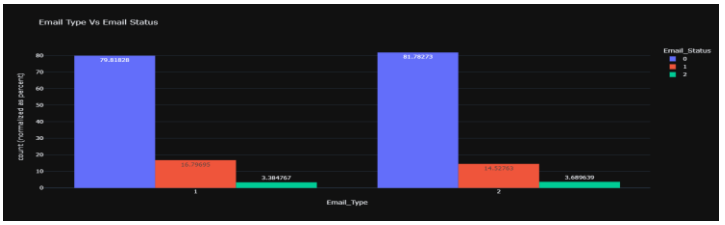
There are two kinds of features in the dataset: Categorical and Non-Categorical Variables. Categorical- A categorical variable is a variable that can take on one of a limited, and usually fixed, number of possible values putting a particular category to the observation. Non-Categorical- A non-categorical or continuous variable is a variable whose value is obtained by measuring, i.e., one which can take on an uncountable set of values. Both are analyzed separately. Categorical data is usually analyzed through count plots in accordance with the target variable and that is what is done here too. On the other hand, Numeric or Continuous variables were analyzed through distribution plots and box plots to get useful insights.

Categorical Variable Insights:









Observation:

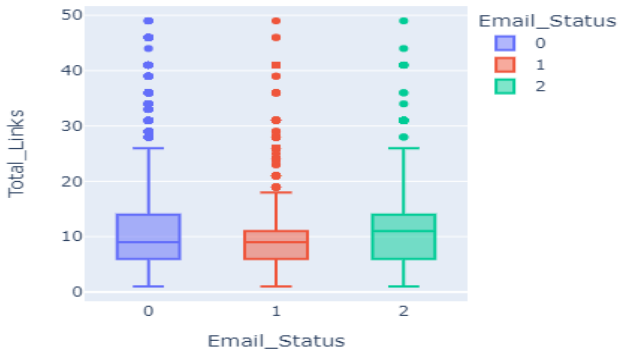
- The email type 1 which may be considered as promotional emails are
- sent more than email type 2 and hence are read and acknowledged more than the other type otherwise the proportion of ignored, read,

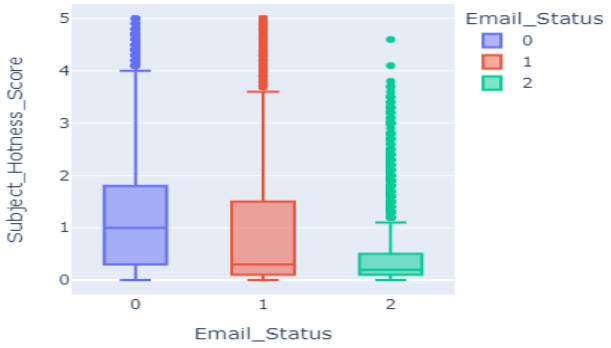
acknowledged emails are kind of same in both email types.

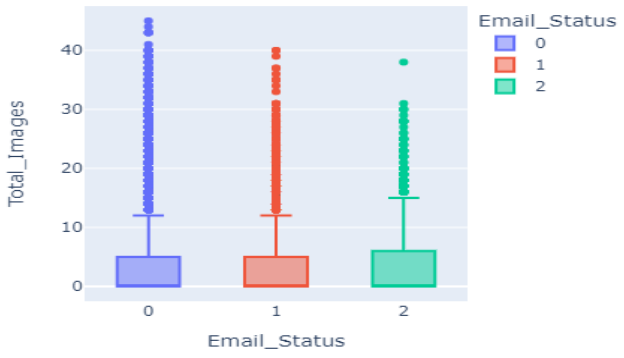
- Email source type shows kind of a similar pattern for both the categories.
- In the customer location feature we can find that irrespective of the location, the percentage ratio of emails being ignored, read, and acknowledged are kind of similar.

- It does not exclusively influence our target variable. It would be better to not consider location as a factor in people ignoring, reading, or acknowledging our emails. Other factors should be responsible for why people are ignoring the emails not location.
- In the Email Campaign Type feature, it seems like in campaign type 1 very few emails were sent but have a very high likelihood of getting read. Most emails were sent under email campaign type 2 and most ignored. Seems like campaign 3 was a success as even when a smaller number of emails were sent under campaign 3, more emails were read and acknowledged.
- If we consider 1 and 3 as morning and night category in time email sent feature, it is obvious to think 2 as middle of the day and as expected there were more emails sent under 2nd category than either of the others, sending emails in the middle of the day could lead to reading and opening the email as people are generally working at that time and they frequently check up their emails, but it cannot be considered as the major factor in leading to acknowledge emails.

Continuous Variable Insights:







Observation:

- In the subject hotness score, the median of ignored emails was around 1 with a few outliers. Acknowledged emails have the most outliers. It is observed that the Subject_Hotness_Score for read and acknowledged emails are much lower.

- Analyzing total past communications, we can see that the more the number of previous emails, the more it leads to read and acknowledged emails. This is just about making connections with your customers.

- The more the words in an email, the more it tends to get ignored. Too lengthy emails are getting ignored.

- The median is kind of similar in all the three cases in total links feature with several outliers.

- More images were there in ignored emails.

- There are a considerable number of outliers in Subject_Hotness_Score, Total Links and Total Images.

Correlation:

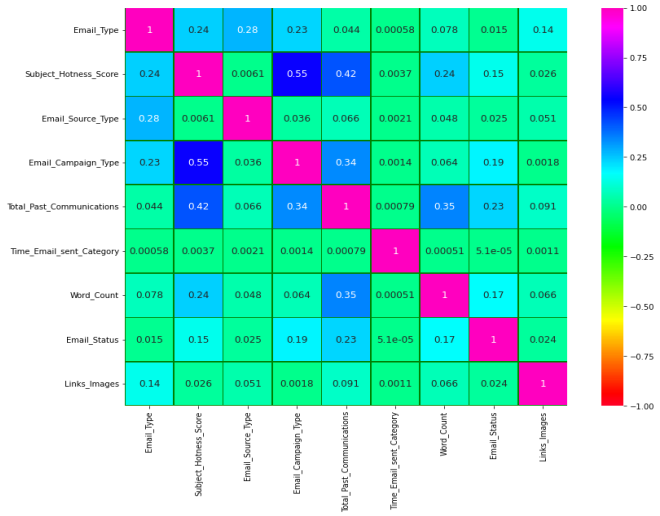


Fig: Heatmap Correlation Matrix

Observation:

- We can see multicollinearity involved in Email Campaign Type, Total past communication, and Total
- links, Total Images among others and we will have to deal with it.
- We can observe that there is a relationship between Total Images and Total Links, they have 75% correlation. To understand if this

relation holds true, we try and validate with a trendline plot b/w the two features.

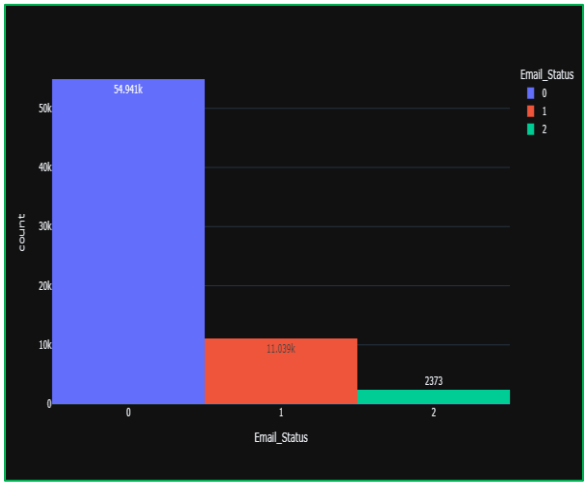
5. Data Manipulation and Feature Engineering:

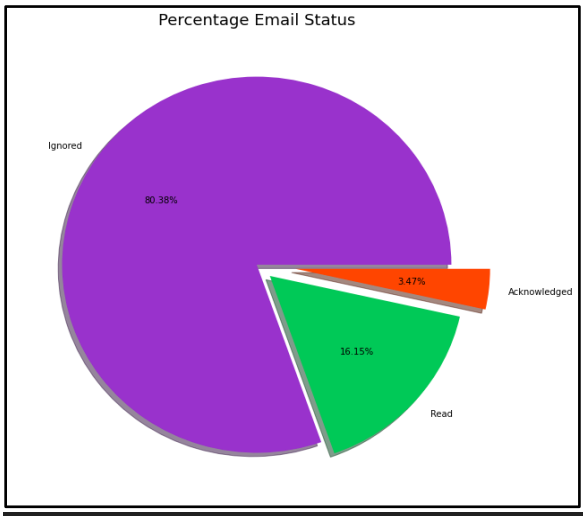
Data manipulation involves manipulating and changing our dataset before feeding it to various classification machine learning models. This involves keeping important

features handling multicollinearity in the dataset, outlier treatment and creating dummy variables if necessary.

We have outliers in our dataset as we saw it earlier in data exploration but as the classes are imbalanced and we cannot also risk overfitting, so we will be exploring how many outliers we have in each class and then decide whether we should keep them or get rid of them.

Target Variable:





Multicollinearity and Feature Selection:

Multicollinearity occurs when two or more independent continuous features in the dataset are highly correlated and can help predict each other and the dependent variable. This makes it difficult to individually analyze the effect of these individual independent variables on the target or dependent variable.

We can quantify multicollinearity using Variance Inflation Factors (VIF).

VIF = 1/(1-R^2) The more the value of R^2 is closer to 1 the more, VIF score tends to infinity. VIF starts with 1 and denotes that the variable has no correlation at all. VIF more than 5-10 can be considered as a serious case of multicollinearity and can affect prediction models.

	variables	VIF
0	Subject_Hotness_Score	1.734531
1	Total_Past_Communications	3.430879
2	Word_Count	3.687067
3	Links_Images	2.629047

Fig: VIF Matrix

Before combining Total Links and Total Images we can see that only Total Links is higher than 5. Earlier we saw that Total Images and Total Links are highly correlated to each other with a score of 0.75.

After combining Total Links and Total Images, now we have our multicollinearity in check.

Feature Scaling:

Feature Scaling is a technique to standardize the independent features present in the data in a fixed range. It is done to prevent biased nature of machine learning algorithms towards features with greater values and scale. The two techniques are:
Normalization: is a scaling technique in which values are shifted and rescaled so that they end up ranging between 0 and 1. It is also known as Min-Max scaling? [0,1].

$$x' = \frac{x - \min(x)}{\max(x) - \min(x)}$$

Normalization of the continuous variables was done here.

One hot encoding:

For categorical variables where no such ordinal relationship exists, the integer encoding is not enough. We have categorical data integers encoded with us, but assuming a natural order and allowing this data to the model may result in poor performance. If the categorical variable is an output variable, you may also want to convert predictions by the model back into a categorical form to present them or use them in some application.

Handling Class Imbalance:

In the exploratory data analysis, we clearly saw that the number of emails being ignored was a lot more than being read and acknowledged. This imbalance in the class can lead to biased classification towards ignored emails. We can handle it with Oversampling and Under sampling.

First, we will go with Random Under sampling and check the results for various models that we will be testing and then with SMOTE. This technique generates synthetic data for the minority class. and then with SMOTETomek which is a mixture of SMOTE and Tomek Links.

Lastly, we will analyze which method works best for our dataset.

Random under sampling involves randomly selecting examples from the

majority class to delete from the training dataset.

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

SMOTETomek is somewhere up sampling and down sampling. SMOTETomek is a hybrid method which is a mixture of the above two methods, it uses an under-sampling method (Tomek) with an oversampling method (SMOTE). This is present within the imblearn. combine module.

6. Modeling:

Logistic Regression:

Logistic Regression is a classification algorithm that predicts the probability of an outcome that can have only two values. Multinomial logistic regression is an extension of logistic regression that adds native support for multi-class classification problems. Instead, the multinomial logistic regression algorithm is a model that involves changing the loss function to cross-entropy loss and predicting probability distribution to a multinomial
probability distribution to natively support multi-class classification problems.

Decision Trees:

Decision tree algorithm falls under the category of supervised
learning. They can be used to solve both regression and classification problems. Decision trees use the tree representation to solve the problem in which each leaf node corresponds to a class label and attributes are represented on the internal node of the tree. Clearly Decision Tree models were overfitting. Both the datasets, whether under sampled or oversampled with SMOTE and SMOTETomek worked well on train data but not on test data.

Random Forest:

Random forests are an ensemble learning method for classification and regression that operates by constructing
a multitude of decision trees at training time. For classification tasks, the output of the random forest is the class selected by most trees. To prevent overfitting, a random forest model was built. Random forest builds multiple decision trees and merges them together to get a more accurate and stable prediction.

KNN Classification:

K-Nearest Neighbor is one of the simplest Machine Learning algorithms based on Supervised Learning technique.
K-NN algorithm assumes the similarity between the new case/data and available cases and put the new case into the category that is most like the available categories.
KNN algorithm stores all the available data and classifies a new data point based on the similarity. This means when new data appears then it can be easily classified into a well suite category by using KNN algorithm
KNN algorithm can be used for Regression as well as for Classification but mostly it is used for the Classification problems
. KNN is a non-parametric algorithm, which means it does not make any assumption on underlying data. It is also called a lazy learner algorithm because it does not learn from the training set immediately instead it stores the dataset and at the time of classification, it performs an action on the dataset.
KNN algorithm at the training phase just stores the dataset and when it gets new data, then it classifies that data into a category that is much like the new data.

XG Boost:

XGBoost is a decision-tree-based ensemble Machine Learning algorithm that uses a gradient boosting framework. The two reasons to use XGBoost are also the two goals of the project:

- Execution Speed.
- Model Performance.

Boosting is an ensemble technique where new models are added to correct the errors made by existing models. Models are added sequentially until no further improvements can be made. Gradient boosting is an approach where new models are created that predict the residuals or errors of prior models and then added together to make the final prediction. It is called gradient boosting because it uses a gradient descent algorithm to minimize the loss when adding new models. XGB SMOTETomek gave the best results till now, with good Test Recall, F1 score and AUC ROC.

7. Evaluation Metrics:

There are several model evaluation metrics to choose from but since our dataset was highly imbalanced, it is critical to understand which metric should be evaluated to understand the model performance.

Accuracy- Accuracy simply measures how often the classifier correctly predicts. We can define accuracy as the ratio of the number of correct predictions and the total number of predictions. Accuracy is useful when the target class is well balanced but is not a good choice for the unbalanced classes, because if the model poorly predicts every observation as of the majority class, we are going to get a high accuracy.

Confusion Matrix:

It is a performance measurement criterion for the machine learning classification problems where we
get a table with a combination of predicted and actual values.

Precision:
Precision for a label is defined as the number of true positives divided by the number of predicted positives.

Recall:
Recall for a label is defined as the number of true positives divided by the total number of actual positives. Recall explains how many of the actual positive cases we
were able to predict correctly with our model.

F1 Score: It's the harmonic mean of Precision and Recall. It is maximum when Precision is equal to Recall.

AUC ROC:
The Area Under the Curve (AUC) is the measure of the ability of a classifier to distinguish between classes. When AUC is 0.5, the classifier is not able to distinguish between the classes and when it's closer to 1, the better it becomes at distinguishing them.

	Model_Name	Train_Accuracy	Train_Recall	Train_Precision	Train_Fiscore	Train_AUC	Test_Accuracy	Test_Recall	Test_Precision	Test_Fiscore	Test_AUC
0	XGB SMOTETomek	0.924	0.924	0.926	0.923	0.987	0.786	0.786	0.751	0.764	0.765
1	XGB TomekLinks	0.929	0.929	0.930	0.921	0.980	0.802	0.802	0.746	0.759	0.767
2	KNN Tuned TomekLinks	0.874	0.874	0.849	0.848	0.912	0.801	0.801	0.746	0.759	0.715
3	KNN TomekLinks	0.883	0.883	0.864	0.860	0.930	0.798	0.798	0.742	0.758	0.701
4	RandomF Tuned TomekLinks	0.882	0.882	0.871	0.853	0.945	0.807	0.807	0.747	0.755	0.767
5	RandomF Tuned SMOTETomek	0.905	0.905	0.905	0.905	0.983	0.743	0.743	0.761	0.751	0.762
6	Random Forest TomekLinks	0.854	0.854	0.798	0.795	0.818	0.809	0.809	0.757	0.732	0.770
7	Decision Tree TomekLinks	0.996	0.996	0.996	0.996	1.000	0.731	0.731	0.731	0.731	0.601
8	Decision Tree SMOTETomek	0.999	0.999	0.999	0.999	1.000	0.698	0.698	0.731	0.713	0.697
9	Random Forest SMOTETomek	0.565	0.565	0.548	0.540	0.759	0.669	0.669	0.773	0.710	0.764
10	Random Forest RUS	0.564	0.564	0.559	0.541	0.752	0.629	0.629	0.774	0.684	0.763
11	LogisticReg TomekLinks	0.673	0.673	0.826	0.731	0.810	0.629	0.629	0.773	0.684	0.769
12	LogisticReg SMOTETomek	0.537	0.537	0.520	0.510	0.722	0.625	0.625	0.772	0.681	0.766
13	LogisticReg RUS	0.541	0.541	0.531	0.517	0.721	0.621	0.621	0.773	0.679	0.763
14	RandomF Tuned RUS	0.740	0.740	0.742	0.739	0.910	0.617	0.617	0.781	0.677	0.759
15	KNN Tuned RUS	0.609	0.609	0.606	0.607	0.802	0.598	0.598	0.772	0.661	0.728
16	KNN RUS	0.648	0.648	0.651	0.646	0.843	0.598	0.598	0.766	0.659	0.707
17	KNN SMOTETomek	0.891	0.891	0.900	0.888	0.987	0.591	0.591	0.752	0.650	0.680
18	KNN Tuned SMOTETomek	0.891	0.891	0.900	0.888	0.987	0.591	0.591	0.752	0.650	0.680
19	XGB RUS	0.975	0.975	0.975	0.975	0.999	0.570	0.570	0.768	0.641	0.732
20	Decision Tree RUS	0.999	0.999	0.999	0.999	1.000	0.492	0.492	0.742	0.573	0.604

Fig: Model Performance Comparison

XG-Boost is the best performing model for the given problem why?

1. Robust to outliers.
2. Supports regularization.
3. Works well on small to the medium data
4. F1 score for train & test set were 89% & 81% respectively.

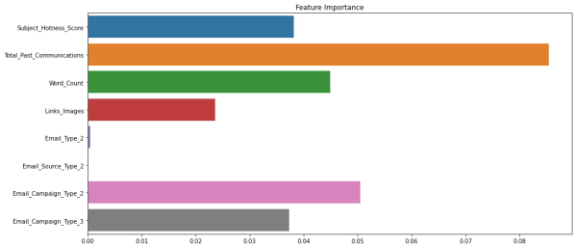


Fig: Feature Importance

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 was 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 modelling, it was observed that for imbalance handling Oversampling i.e., SMOTE 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 89% and test score of 81% for F1 score.

Challenges:

- Choosing the appropriate technique to handle the imbalance in data was quite challenging as it was a tradeoff b/w information loss vs risk of overfitting.
- Overfitting was another major challenge during the modelling process.
- Understanding what features most important and what features are to avoid was a difficult task.
- Decision making on missing value imputations and outlier treatment was quite challenging as well.

Future Work:

We can collect more relevant feature like behavior of people who ignore the emails.

References:

D-Tribe
SendinBlue.com
Researchgate.net