Aalto University
School of Science

Machine Learning in Email Marketing

Muhammad Kamal Memon
February 2019

CS-E4870 - Research Project in Machine Learning and Data Science
Neda Barzegar Marvasti

Table of Contents

Introduction

Email marketing is an important tool used by almost all established companies today as a communication mechanism to target their customers with specific content. Sending emails is fast, cheap and highly targeted and enables companies to push the content they want to their customer base. One big advantage in this approach is that related statistics can be gathered directly concerning user responses to these emails such as how many opened and followed through to a link, which users remained idle and who opted to quit the mailing list. These statistics are important as they reveal quantifiable metrics such as conversion rate and unsubscription ratios. Email marketing is prevalent across all industries and is not confined to any particular market hence there has always been a need to improve its efficiency.

The goal of this research project is to focus on various Machine learning methods which can help improve the efficacy of email marketing. We strive to find an optimal solution using the gathered data of user interaction or lack of it to classify an email response in order to increase the conversion rate.

In this report we have worked on an email marketing campaign dataset for a small and medium sized enterprise (SME). We approached the problem by first examining the data, performing Exploratory data analysis to find the relations and pattern in between the features. We then moved on to do the processing and modeling classifiers around it. Lastly we optimized and evaluated the results with a comparison of various models we employed for classification purposes.

Literature Review

Any business entity or a company has objective functions, such as maximising revenue, customer satisfaction and/or customer loyalty. These depends primarily on the sequence of interactions between company and customers. A key aspect of this setting is that interactions with different customers which occur in parallel. Using a variant of temporal-difference learning algorithm to learn from online partial interaction sequences, so that information acquired from one customer is efficiently assimilated and applied in subsequent interactions with other customers is one way to accommodate the information for better use. The difference in this approach from traditional reinforcement learning is that the agent interacts with many customers concurrently. Using a simulator to compare this method with Monte Carlo, traditional TD and Contextual Bandit algorithms demonstrates that it works better (Silver, David et al 2014).

To model consumer responses to direct marketing, another interesting proposal is to use Bayesian networks learned by evolutionary programming. By using a dataset of a company that sells multiple product lines of general merchandise It has been shown that this approach of predictive modeling in direct marketing works better than other benchmark methods, including neural networks, classification and regression tree (CART), and latent class regression. The company sends regular mailings to its list of customers, and this particular data set contains the records of 106,284 consumers. Each customer record contains 361 variables, including purchase data from recent promotions and the customer's

purchase history over a 12-year period. A recent promotion, achieved a 5.4% response rate, which represents 5,740 customers who made purchases from emailed catalog (Cui, Geng et al 2006).

Reinforcement learning approaches are also good candidates for the problem at hand of sequential targeted marketing. There are "batch" and "simulation based" RL methods. Direct or batch reinforcement learning attempts to estimate the value function $Q(s, a)$ by reformulating value iteration as a supervised learning problem. On the other hand, Indirect or simulation-based methods of reinforcement learning first build a model of Markov Decision Process by estimating the transition probabilities and expected immediate rewards, and then learn from data generated using the model and a policy that is updated based on the current estimate of the value function. Naoki, Abe et al found that where modeling is possible indirect methods works better whereas in realistic situations the performance degrade. It is also shown that semi-direct methods are effective in reducing the amount of computation necessary to attain a given level of performance, and often result in more profitable policies (Naoki, Abe et al. 2002).

Dataset Review

Email Marketing Stats (Infographic): This is a blog which has summary statistics showcasing the impact of email marketing by concerte numbers. However since it's an infographic I don't think we can extract enough data to work on the problem at hand (Mohsin, Maryam. 2018).

Enron Email Dataset: The Enron email dataset contains text of the emails sent by employees of the Enron Corporation. It was obtained during investigation of systemic financial fraud in Enron. I think this dataset is also not suitable for us as it does not pertain to any kind of marketing but would rather be appropriate for some NLP problem (Cukierski, William. 2016).

Crossover Analysis between Archivist and Research Data Management: This data set contains analysis based on the the connections between archives professionals and research data management. These connections are established on the email correspondences. The dataset demonstrates how frequently archivists and records professionals discuss research data on the Archives-NRA list, the topics which are discussed, and an increase in these discussions over time. Anyhow as with earlier dataset this also doesn't present a problem which directly relates to our topic of research on email marketing (Grant, Rebecca, 2018).

Email Campaign Management for SME: This is the most interesting dataset as it has data from Small and medium-sized enterprises (SME) who use email marketing to target their customers. The dataset includes different aspects of emails to characterize it and also tracks if the mail is ignored, read or acknowledged by the recipient. This is of particular interest as we are striving to work on a similar solution (Gokagglers, 2018).

Of the above, we settled upon the Email Campaign Management for SME dataset. There is potential to apply supervised machine learning approaches such as multi-class classification and/or regression analysis to establish relations between email attributes and the recipient's response.

Exploratory Data Analysis

The data is aggregated for a single user and doesn't have any time series. However there is 'Time Email Sent' feature. We have ~68k unique users with 11 features in this dataset.

Summary statistics

We have 11 features with following descriptive statistics summarizing the central tendency, dispersion and shape of their distribution:

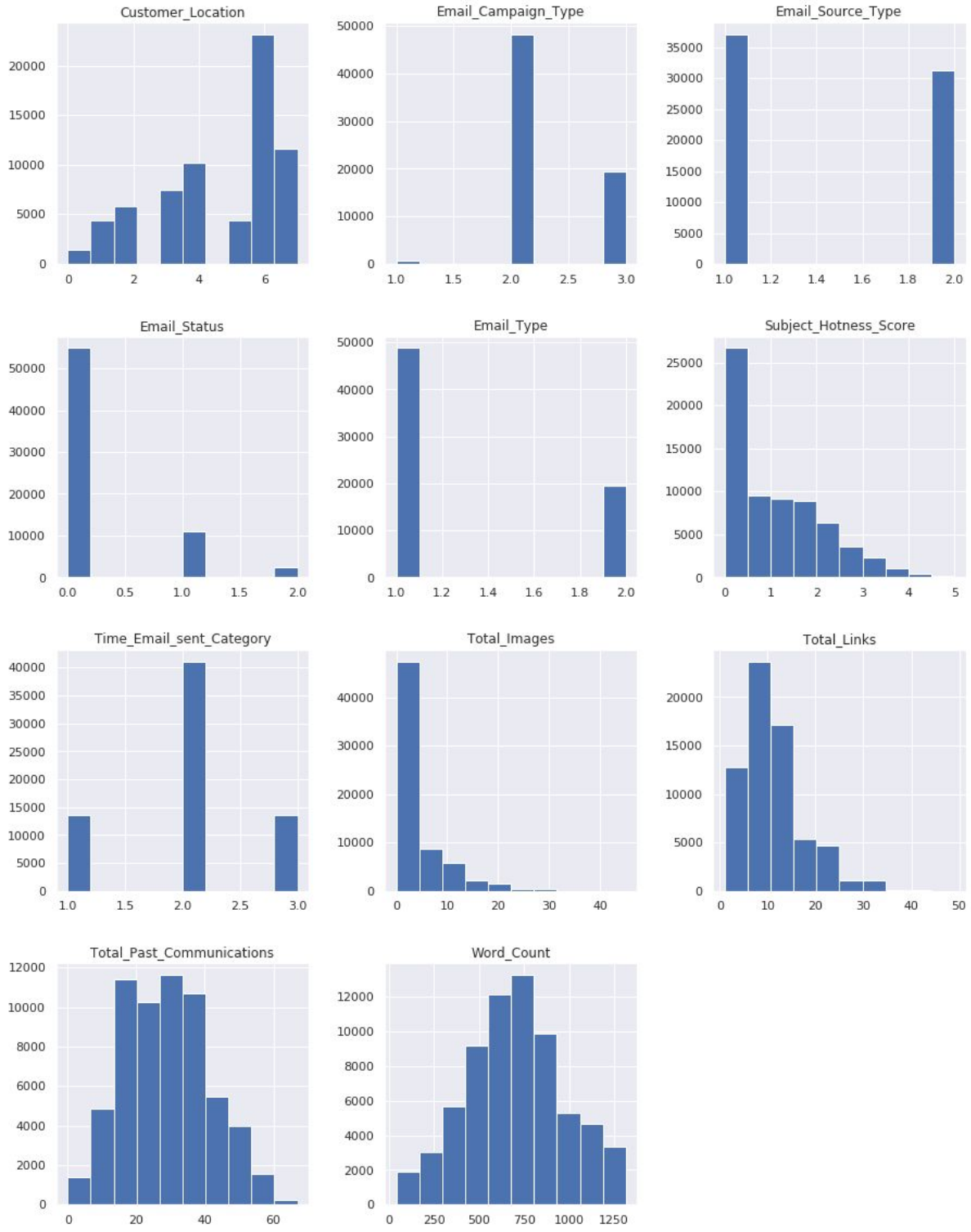
	count	unique	top	freq	mean	std	min	25%	50%	75%	max
Email_Type	68,353.00	NaN	NaN	NaN	1.29	0.45	1.00	1.00	1.00	2.00	2.00
Subject_Hotness_Score	68,353.00	NaN	NaN	NaN	1.10	1.00	0.00	0.20	0.80	1.80	5.00
Email_Source_Type	68,353.00	NaN	NaN	NaN	1.46	0.50	1.00	1.00	1.00	2.00	2.00
Customer_Location	56758	7	G	23173	NaN	NaN	NaN	NaN	NaN	NaN	NaN
Email_Campaign_Type	68,353.00	NaN	NaN	NaN	2.27	0.47	1.00	2.00	2.00	3.00	3.00
Total_Past_Communications	61,528.00	NaN	NaN	NaN	28.93	12.54	0.00	20.00	28.00	38.00	67.00
Time_Email_sent_Category	68,353.00	NaN	NaN	NaN	2.00	0.63	1.00	2.00	2.00	2.00	3.00
Word_Count	68,353.00	NaN	NaN	NaN	699.93	271.72	40.00	521.00	694.00	880.00	1,316.00
Total_Links	66,152.00	NaN	NaN	NaN	10.43	6.38	1.00	6.00	9.00	14.00	49.00
Total_Images	66,676.00	NaN	NaN	NaN	3.55	5.60	0.00	0.00	0.00	5.00	45.00
Email_Status	68,353.00	NaN	NaN	NaN	0.23	0.50	0.00	0.00	0.00	0.00	2.00

```
Email_Type                  68353 non-null int64
Subject_Hotness_Score       68353 non-null float64
Email_Source_Type           68353 non-null int64
Customer_Location           56758 non-null object
Email_Campaign_Type         68353 non-null int64
Total_Past_Communications   61528 non-null float64
Time_Email_sent_Category    68353 non-null int64
Word_Count                  68353 non-null int64
Total_Links                 66152 non-null float64
Total_Images                66676 non-null float64
Email_Status                68353 non-null int64
dtypes: float64(4), int64(6), object(1)
```

We can see that there are missing values for *Customer_Location*, *Total_Past_Communications*, *Total_Links* and *Total_Images*.

Distributions

The features in the dataset are distributed as such:



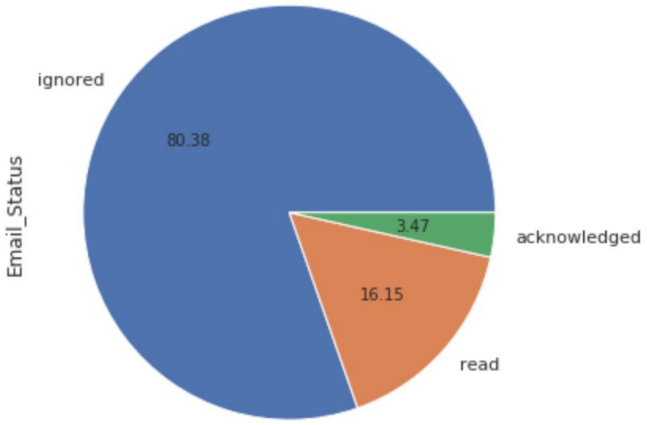
The above histograms tells us that about 6 features are categorical and '*Total Past Communications*' and '*Word Count*' are almost normally distributed.

Email Status

This is the feature of interest as the dataset considers its the label feature. It has three possible values describing the outcome of the email sent to a particular user:

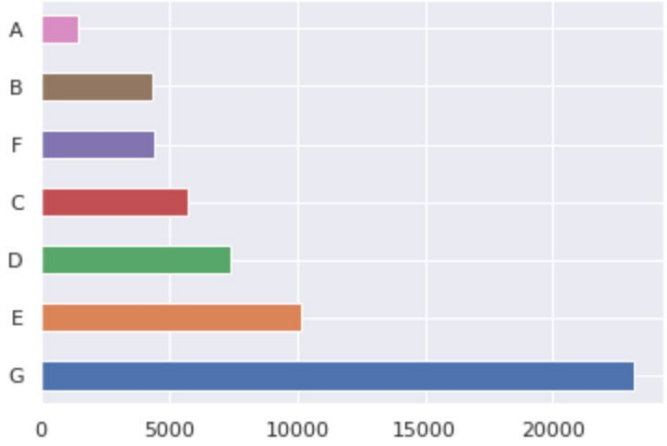
- 0 - Ignored
- 1 - Read
- 2 - Acknowledged

Following are the proportions of these values in the dataset:



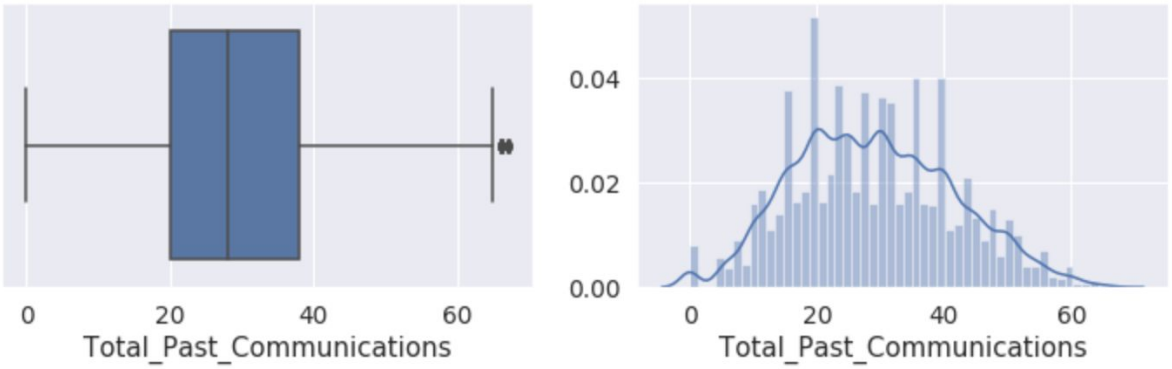
As the pie chart shows we have a very high ratio of ignored emails.

Customer Location



The value count in the plot above shows that majority of the users are from the same G location but the other half is relatively evenly divided in other locations. The *Customer_Location* feature is categorical and we will hot-encode it the processing phase.

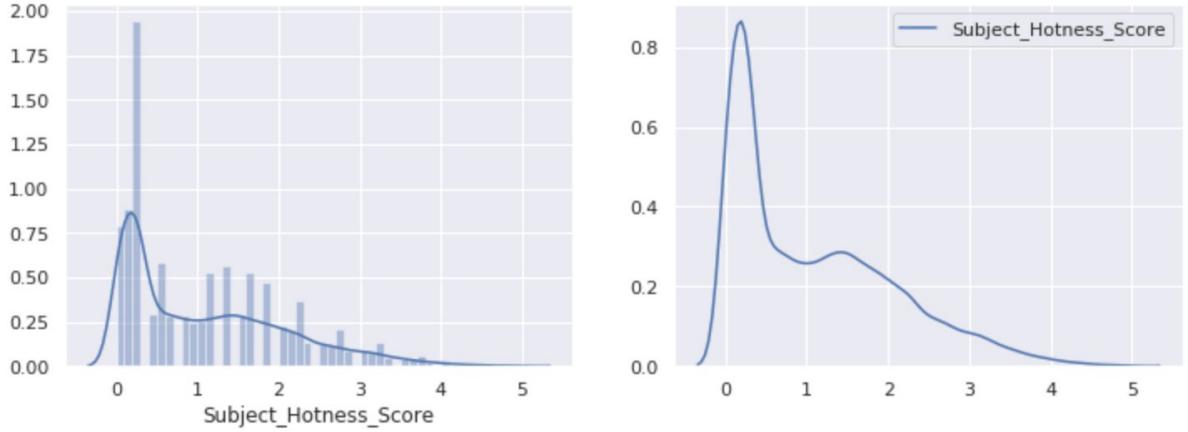
Total Past Communications



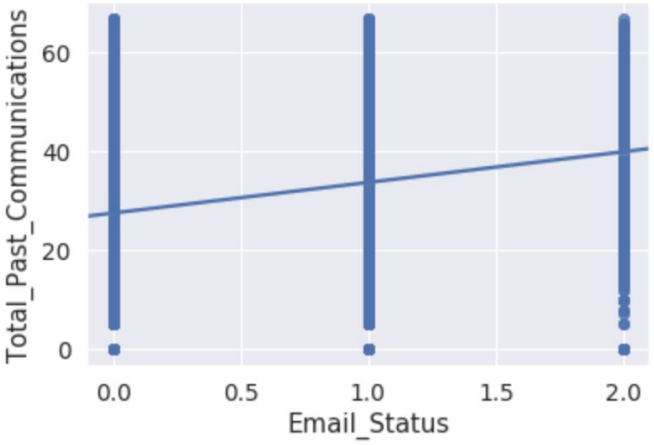
We can observe that the distribution is mostly centered around mean and there are very few outliers.

Subject Hotness Score

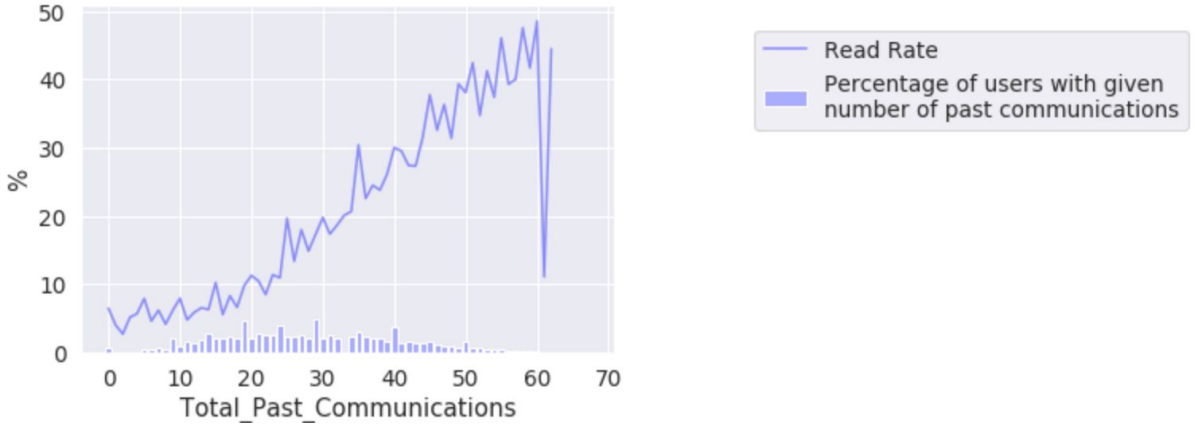
Another feature of interest in the dataset is about the emails subject hotness score. When plotted its distribution represent a sharp hike around 0.3



Email Status vs Past Communications



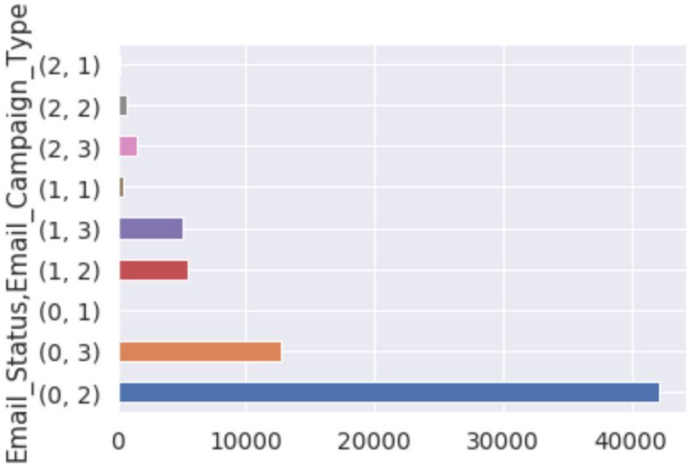
We try to visualize the relation between past communications and the email status. The plot above show a steady ascent when the status is 2 means when the email is read. Hence a simple hypothesis is that the more users have communicated the more the might tend to read the email. Lets explore this more by comparing Past communications with only the values when the Email was actually read:



The plot above depicts that the relation is not exactly proportional as we thought but still has significant impact.

Email Status VS Email Campaign Type

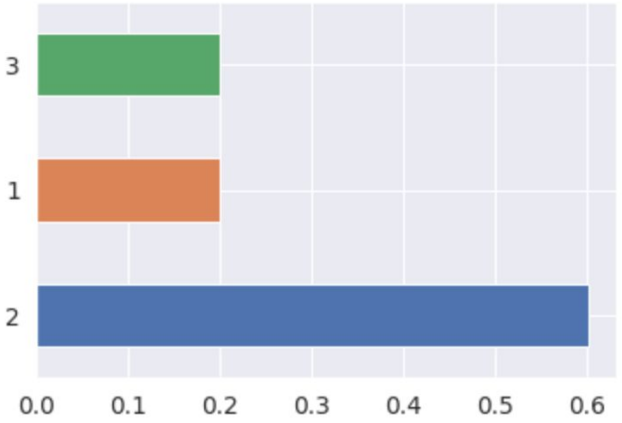
There are three campaign categories and we would like to learn which one has impacted the conversions most. We would group the types together and plot to visualize the results:



Here we can observe that Campaign 3 has resulted in most READ and ACKNOWLEDGED emails.

Email Status vs Time Email Sent Category

Let's explore how many time categories we have and how they are distributed.



We can see that 1 and 3 are each 20% wile category 2 covers 60% of the records. Let's check what relation they have with email response (*Email_status*).

The plot actually follows the time email sent category distribution. The more email sent at a time the more the responses were hence there isn't any anomaly to notice here.

Email Status: Time Sent vs Past Communications

We will now consider how past communications and the time the email was sent effect email response in the data:

The plot shows that the most favourable responses (read or acknowledged) are with the time category 2 with a high number of past communications.

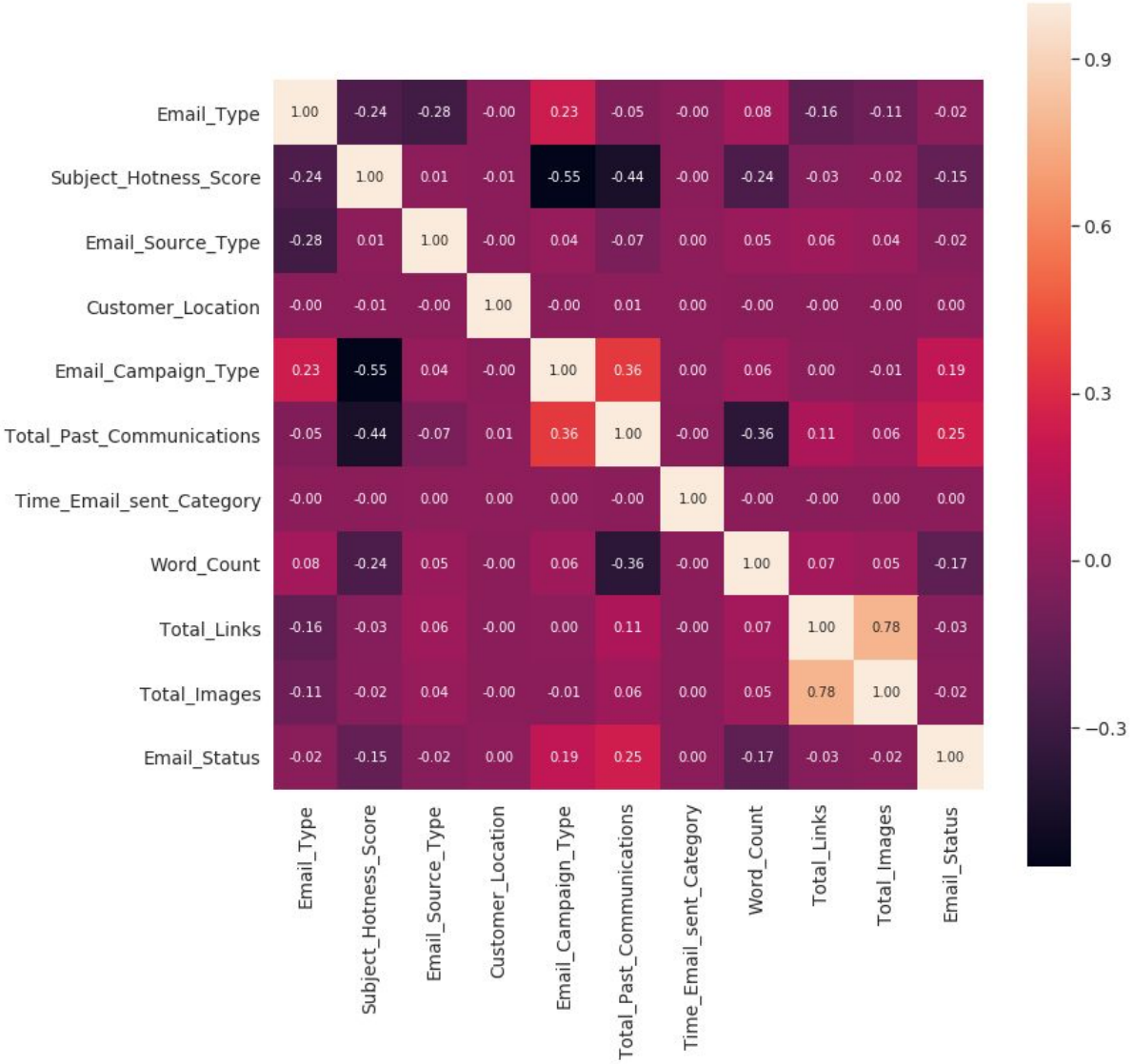
Email Status: Subject Hotness Score vs Past Communications

We will do the same factor impact analysis of Subject hotness score feature along with past communication on favorable email responses.

We can see that with lower subject hotness score and higher past communications, the email's response of Read and Acknowledge is high.

Pearson Correlation

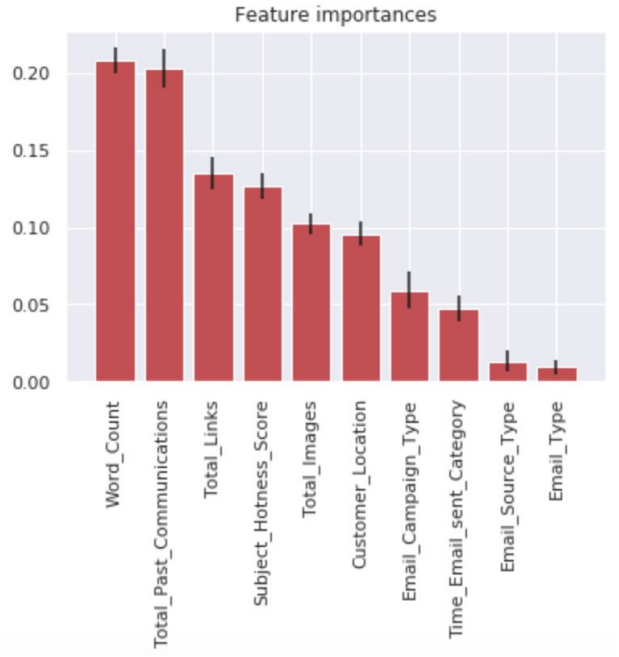
We now take all features and compute a Pearson correlation heatmap. A Pearson correlation coefficient as present in the matrix below is a number between -1 and 1 that indicates the extent to which two variables are linearly related.



We can observe that Email_Status (which is our target label) has a moderate uphill (positive) linear relationship with past communications and campaign features. Apart from that other features are very weakly correlated.

Random forest feature importance

Random forests are among the most popular machine learning methods thanks to their relatively good accuracy, robustness and ease of use. For classification, they also provide a straightforward methods for feature selection based on Gini impurity. Gini impurity is a measure of how often a randomly chosen element from the set would be incorrectly labeled if it was randomly labeled according to the distribution of labels in the subset. We measured the same for our datasets which resulted into following:



As we have observed earlier in out exploratory analysis, past communications and word count has relatively higher impact on the outcome of email response.

Analysis Conclusion

By exploring and analysing the features in the data it can be deduced that we can model a multi-class classification solution around the Email_Status variable treating it as target label and taking into account all other features. We settle on not removing any feature from the set for now as training, optimizing and comparing various models will yield the best possible solution.

Implementation

Methodology

For classification, optimization and training the models we are using Python Scikit-learn library. For cleaning and building the dataset we have use Pandas and for visualizations and exploratory analysis we used matplotlib and seaborn.

For comparison of different classifiers we have implemented and evaluated key methods on two metric, namely Accuray and F1 Score. Since we have three labelled classes we have used weighed F1 scores. The accuracy here represents the percentage of set of labels predicted for a sample matched the corresponding set of actual labels whereas the F1 Score is the weighted average of the precision and recall of classification.

Our method to find the best fit classifier for our dataset involve experimenting with the four different algorithms and comparing them. We start with preprocessing the data by normalizing the continuous feature space and categorical features are hot-encoded to numeric ones. In our dataset we face a huge class imbalance due to the nature of email responses hence in order to overcome that we employed oversampling techniques to have a viable distribution of labels for classification. Finally to optimize these models we have employed Bayesian optimization techniques using Hyperopt which is a Python library for serial and parallel optimization. The models are trained on a 70/30 split training and testing dataset and we used 3 fold cross-validation.

SMOTE Oversampling

The biggest challenge at hand is the distribution of our class that is Email Status variable. In the dataset the three classes (ignored, read and acknowledged) are distributed as such:

```
0                   0.80
1                   0.16
2                   0.03
Name: Email_Status, dtype: float64
```

We can clearly see there is a huge class imbalance there and regardless of the type of classifying model we use it won't be able to fit properly. Hence we proceed to tackle this problem by oversampling the data from the same distribution from where it comes. Oversampling is a well-known way to potentially improve models trained on imbalanced data.

To proceed with oversampling we used **SMOTE - Synthetic Minority Over-sampling Technique** as presented in [1]. At a high level, SMOTE creates synthetic observations of the minority class(es) by:

- Finding the k-nearest-neighbors for minority class observations (finding similar observations)
- Randomly choosing one of the k-nearest-neighbors and using it to create a similar, but randomly tweaked, new observation.

After oversampling our class balance is much better:

```
0                   0.71
2                   0.14
1                   0.14
dtype: float64
```

Classifiers

Following are the supervised machine learning methods we have used in this study:

Naive Bayes Classifier

Naive Bayes methods are a set of supervised learning algorithms based on applying Bayes' theorem with the "naive" assumption of conditional independence between every pair of features given the value of the class variable. In spite of their apparently over-simplified assumptions, naive Bayes classifiers have worked quite well in many real-world situations.

Support Vector Classification (SVC)

Support vector machines are a set of supervised learning methods. SVM differs from the other classification algorithms in the way that it chooses the decision boundary that maximizes the distance from the nearest data points of all the classes. An SVM strives to find the most optimal decision boundary. The most optimal decision boundary is the one which has maximum margin from the nearest points of all the classes. The nearest points from the decision boundary that maximize the distance between the decision boundary and the points are called support vectors.

Random Forest Classifier

Random forest method is based on an ensemble of decision trees. It works as a meta estimator that fits a number of decision tree classifiers on various sub-samples of the dataset and uses averaging to improve the predictive accuracy and control over-fitting.

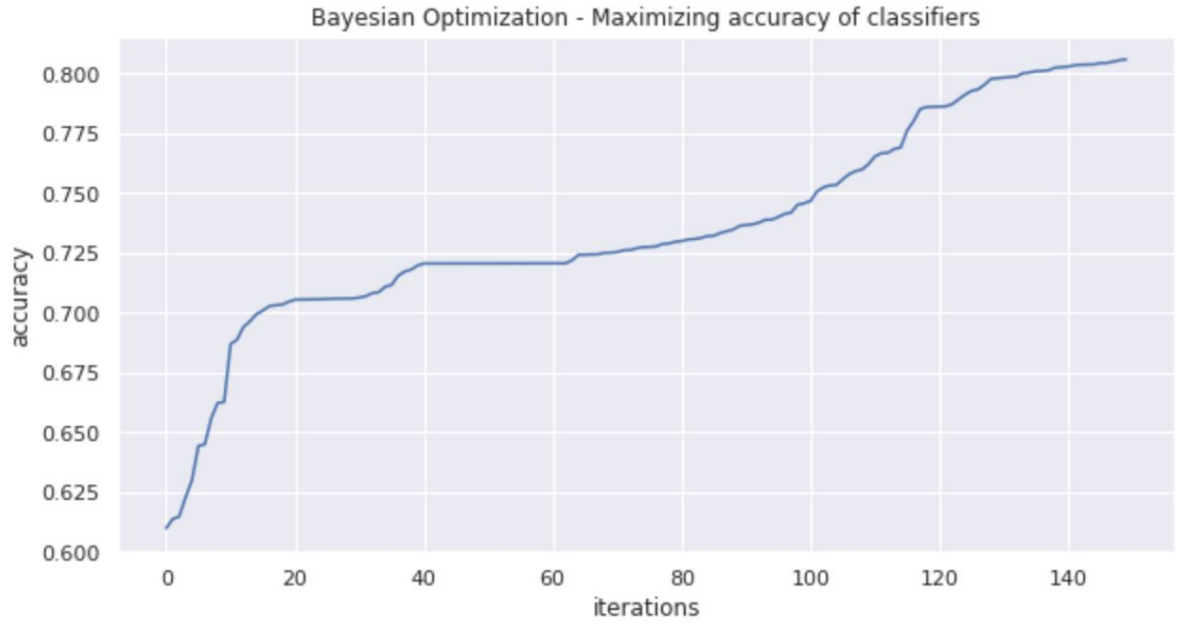
KNeighborsClassifier

This classifier implements the k-nearest neighbors vote mechanism. In the generic k-NN model, each time a prediction is to be made for a data point, first this data point's distance from all other points is to be calculated and then only nearest k-points can be discovered for voting.

Bayesian Optimization

Bayesian optimization is a probabilistic model based approach for finding the minimum of any function that returns a real-value metric. Recent results (http://proceedings.mlr.press/v28/bergstra13.pdf) suggest Bayesian hyperparameter optimization of machine learning models is more efficient than manual, random, or grid search with better overall performance on the test set and less time required for optimization.

In our implementation, to optimize the classification models we used the Hyperopt python library with accuracy from 5 fold cross validation as evaluation metric. The optimization run with a choice of hyperparameters for each model in a well defined space, we track history of the accuracy achieved with each random combination of the hyperparameters and save the best. Once the optimization process completed we use the best found classifier and hyperparameters to train the model.



Results

For each of the four classifiers in consideration we found the most optimal hyperparameters by optimization which gives us the best accuracy. Here below are our findings:

Classifier	**Best parameters**	**Highest accuracy**	**Weighted F1 Score**
KNN	*n_neighbors*: 27	0.7522	0.6809
SVC	C = 9.8679, gamma = 19.9878, kernel = rbf	0.7850	0.6323
Random Forest	'max_depth': 19, 'max_features': 4, 'n_estimators': 19, 'criterion': gini, 'class_weight': 'balanced'	0.8058	0.8007
Naive Bayes	*alpha:* 1.9881	0.7204	0.6137

Following are the classification reports for each method displaying the main classification metrics:

KNN

```
Accuracy:  0.739
Classification Report:
                precision    recall  f1-score   support

           0       0.78      0.94      0.85     11632
           1       0.43      0.03      0.06      2352
           2       0.52      0.45      0.48      2353

   micro avg       0.74      0.74      0.74     16337
   macro avg       0.57      0.47      0.46     16337
weighted avg       0.69      0.74      0.68     16337
```

SVC

```
Accuracy:  0.7297
Classification Report:
                precision    recall  f1-score   support

           0       0.73      1.00      0.84     11632
           1       0.69      0.00      0.01      2352
           2       0.98      0.12      0.22      2353

   micro avg       0.73      0.73      0.73     16337
   macro avg       0.80      0.37      0.35     16337
weighted avg       0.76      0.73      0.63     16337
```

Naive Bayes

```
Accuracy:  0.7206
Classification Report:
                precision    recall  f1-score   support

           0       0.72      1.00      0.84     11632
           1       0.64      0.01      0.02      2352
           2       0.63      0.06      0.10      2353

   micro avg       0.72      0.72      0.72     16337
   macro avg       0.66      0.36      0.32     16337
weighted avg       0.70      0.72      0.61     16337
```

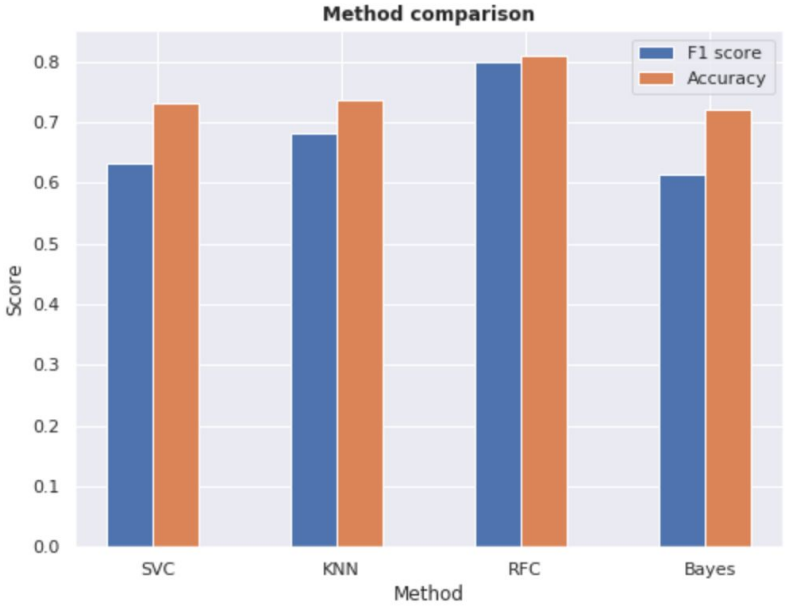
Random Forest

```
Accuracy:  0.8062
Classification Report:
                precision    recall  f1-score   support

           0       0.85      0.91      0.88     11632
           1       0.38      0.31      0.34      2352
           2       0.92      0.78      0.84      2353

   micro avg       0.81      0.81      0.81     16337
   macro avg       0.72      0.67      0.69     16337
weighted avg       0.79      0.81      0.80     16337
```

Comparison

Based on the results of our experimentation we can assert that the Random Forest method has performed better than other approaches in both Accuracy and F1 Score metrics. In

general we can observe that methods other than Random Forest are on a similar level with each other and KNN has the lowest contrast in F1 score and accuracy. This comparison however is on two metrics and doesn't fully reflect the nuances of the multi-class classification problem at hand but still however provide and empirical baseline for evaluation of methods. The comparison is shown in the figure below:



Code

The code of this implementation can be found here:
https://github.com/kamalmemon/MLResearch/blob/master/src

Conclusion

Through this work, we employed learning models for predicting the response of targeted marketing emails in terms of Ignored, Read or Acknowledged response. The dataset we used is based on the features extracted from the emails, email recipients' profiles and the engagement characteristics of the recipients. Extensive exploratory analysis and visualizations is done in order to gauge the relationships of features among themselves and with the target label. After getting a good grasp on the dataset, four classification methods are applied and compared for this multi-class classification problem. The results show that Random Forest Classifier performs better among all, however more experimentation with different evaluation metrics and methods is left for future work.

Future Work

Based on the literature review, there can be various approaches to solve the problem of the efficiency of marketing emails through various machine learning techniques. We have attempted one in this study however the approaches of online Reinforcement Learning and

Bayesian Networks also holds immense potential. In the future, an extensive study can be done using different datasets with various approaches to determine which method work better under which circumstances. On the other hand more features can be extracted for the dataset from the emails and user profiles.

References