Smart Email Filter: Predicting Which Emails You'll Read

Background

We receive a big number of emails every day and reading all of them is a time-consuming task. Some people (like me) are rapidly going over the emails by looking on the subject line and the sender email and open just the most relevant emails.
Some of us have got a very long history of receiving emails. My mailbox has been active since 2004, and has over 114K emails, while only a subset of them were opened.

Problem

I wanted to write a script that analyzes my historical emails and predict according to the words at the subject line and the sender email, if a new email is going to be read.

Interest

This method could be used as a filter for marking important mail, or for filtering spam emails from the other end.

Data sources

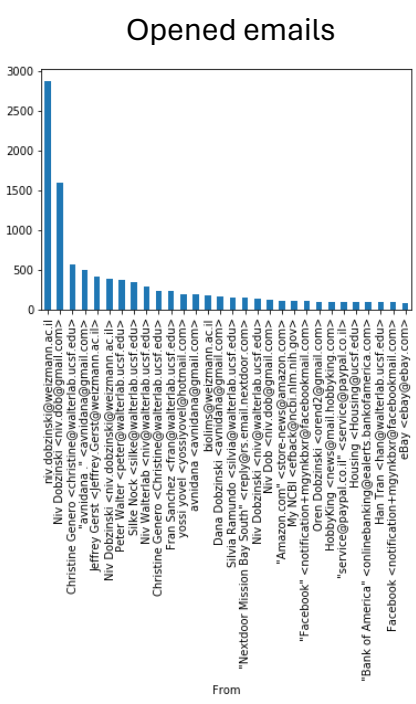
The full email box was retrieved using google services at https://takeout.google.com/

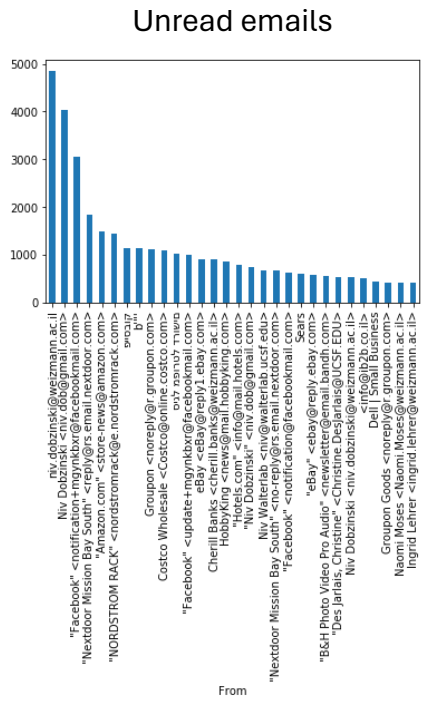
Methodology + results

The Gmail mailbox in the mbox format was parsed and the relevant fields of every email were retrieved and saved into a panda dataframe. The dataframe was saved in a csv format for future use.
The field of the dataframe was organized and relevant columns were converted to binary numbers.
Basic exploratory statistics were done on the mailbox data.

Emails by sender address

The emails were grouped by the sender address (From), sorted, and presented as bar graphs to see the distribution.





Subject line word cloud

All the words from the subject lines of unread emails and from opened emails were extracted and saved as two separate strings. A world cloud was created from each string.

Opened emails

Unread emails

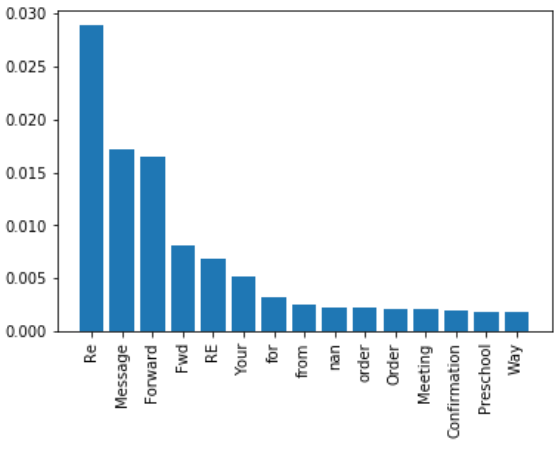
Train-test-split

The emails were split with a ratio of 80/20 for later machine learning prediction.

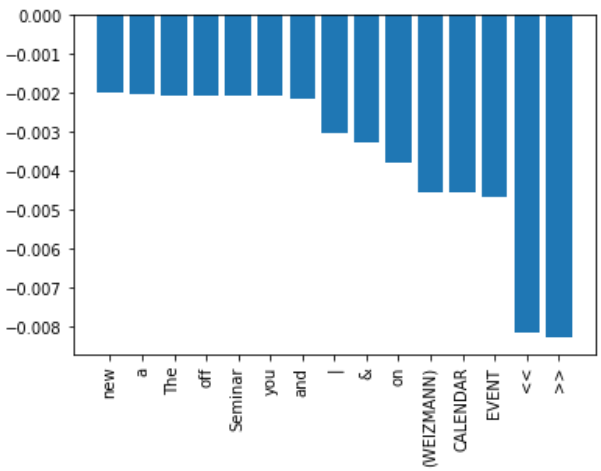
Subject line words scoring

The frequency of each word was calculated and saved in a dictionary structure, in which each key is a word, and its key is the frequency. The frequencies of the words were calculated to the opened emails and separately to the unread emails.
Since some words are abundant in both opened and unread emails, a score was calculated for each word according to the following calculation: (Opened frequency) – (unread frequency).
Words that are abundant in both groups, or not abundant in both – will have scores around zero.
Words that are more abundant in opened emails will have higher scores, and in unread emails – will have lower negative scores.

Opened emails

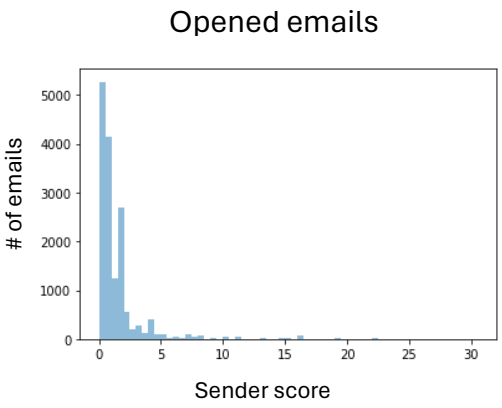


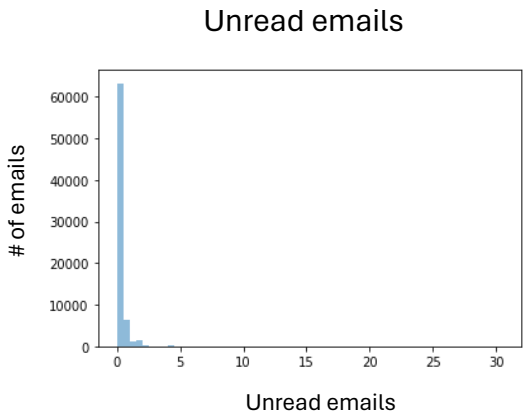
Unread emails



Sender email scoring

The number of times that an email was opened from a specific sender (From) was counted, as well as the number of times an email was left unread from the same sender. These two values were saved for each email in the dataframe.
The sender score was calculated as follows: (number of opened emails from a sender)/(number of unread emails from the sender).
The higher the score – the more times a email from the specific sender was opened versus unread.
In case of division in 0, the infinity value was replaced with the maximum value in the dataset that is not infinity.





As expected, the sender score was higher in the opened emails than in the unread emails.

Normalizing the scores

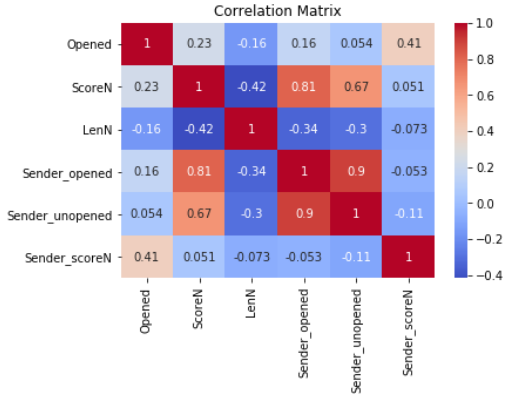
Since the scoring method results in different range of values, both scores were normalized to be at the same scope.
The normalization method used is min/max normalization:

(sender score) – (min sender score) / (max sender score)- (min sender score)

(subject line score) – (min subject line score) / (max subject line score)- (min subject line score)

Correlation overview

The correlation between the different parameters was calculated and visualized. The correlation between opened (what we want to predict) and Sender score is the highest. Following is the correlation to ScoreN – the subject line score.



Machine learning prediction:

Logistic Regression

To predict if an email is going to be opened, we first used logistic regression model.

First, only the sender score was used for prediction since it has the highest correlation to the opened email.

Sender score only:
Overall Accuracy: 86.34%
Precision: 0.85
Recall: 0.38
F1-Score for Class 0 (Negative): 0.92
F1-Score for Class 1 (Positive): 0.53

Sender score + subject line score:
Overall Accuracy: 85.89%
Precision: 0.86
Recall: 0.35
F1-Score for Class 0 (Negative): 0.92
F1-Score for Class 1 (Positive): 0.50

The prediction is slightly better without the subject line score.
The model shows good precision but lower recall for positive cases, suggesting it misses some actual positives.

K-nearest neighbors

I have tried to use K-nearest neighbors to predict the two classes.
The first step was to find the optimal k value for the prediction.
The optimal number for K in this prediction was found to be 18.

Sender score only:
Overall Accuracy: 86.94%
Precision: 0.75
Recall: 0.52
F1-Score for Class 0 (Negative): 0.92
F1-Score for Class 1 (Positive): 0.61

Sender score + subject line score:
Overall Accuracy: 86.7%
Precision: 0.75
Recall: 0.50
F1-Score for Class 0 (Negative): 0.92
F1-Score for Class 1 (Positive): 0.60

Conclusion

We have successfully identified the words that are more frequent in opened emails versus unopened emails and gave the correspondence score for each subject line.
We have also given a score for each email according to the frequency that an email from the specific sender was opened in the past.
We found that the more relevant information needed for predicting if an email is going to be opened or not is the sender email address and not the subject line words.

That means that when I decided to open an email or not to open an email, I made this decision based on the sender email and not based on what written on the subject line as I initially assumed.

The machine learning prediction model that I used, were giving a very good prediction to the negative class – the unread emails, however it was less effective toward identifying the positive class – opened email.
The K-nearest neighbors model with sender score only gave the more accurate prediction and identified better if an email is going to be opened.