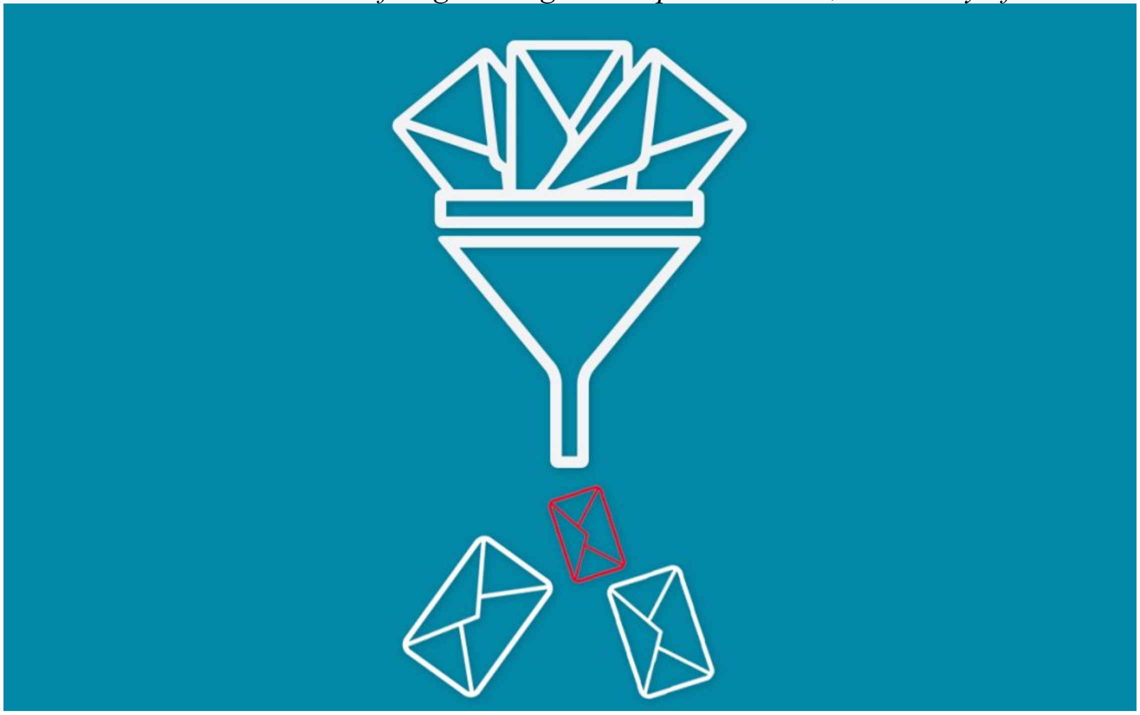
Personal vs. Promotional Email Prediction

Jingxiu Hu Xiao Lin

Daniel Felix Ritchie School of Engineering & Computer Science, University of Denver



Abstract

As one of the fastest and the cheapest tool to communicate with people who are far away, email plays an important role in our daily life. However, it also raises some problems. It is used as commercials by business and causes time and safe issues. On the other hand, a mistakenly labeled email might cause huge loss for large business. Email spam filter helps people to save time by filter promotional or junk email, but the existing spam filter has to read the full content and capture the "sensitive" words. In this paper, we interpreted the binary logistic regression and gradient boosting to classify the email into personal or promotional email without using full content of email that the Naïve Bayes usually used for traditional spam filter. The accuracy of the gradient boosting is 0.9990 which scores calculated by f1 scores on Kaggle competition.

1. Introduction

Email spam filter is used for many people, but it seems not very efficient now. The traditional spam filter has to scan each email contents and find the words that have more probability showing up in commercials. If the senders avoid those "sensitive" words, the promotional email will show up in the index.

To deal with this issue, we used two models and tried to find the most accurate one. At first, we used logistic regression model, but the better performance seems workable (based on Kaggle leaderboard), so we decided to try some other models. Then we noticed that compare to python, R and spark, the xgboost has the fastest running time for large input size n; hence, we used gradient boosting algorithm which is a feature in the xgboost model. The accuracy of those two models are both good with 0.96 for logistic regression and 0.99 for gradient boosting.

2. Literature Review

There are many papers worked on the development of spam filter. For the paper named “Proposed efficient algorithm to filter spam using machine learning techniques”, the author introduces the common methodologies includes multilayer perceptron which is a model that giving information “by activating input neurons containing values labelled on them”[1]; the C4.5 decision tree classifier which outputs structural data in form of binary tree; and the Naïve Bayesian classifier. “A Study on Email Spam Filtering Teckniques” indicates other models that workable for spaming such as K nearest neighbors and support vector machine[2].

However, for most of those kinds of papers, authors focus on the theory and there is no data tested the performance of each model. Also, these models require the original emails with full contents. For us, we focus on the model performances and try to find the most useful model for spam filter. It does not matter if the full contents of emails given or not as long as the useful data collected.

3. Models Used

This section introduces the basic concepts of the two models we used for classifying emails to personal or promotional email.

3.1 Logistic regression

Logistic regression is used for classifying a problem with two possible outcomes[3]. It is a predictive analysis which could interpret the relationship between a dependent binary variable and other independent variables. It uses logistic function to find the output of a linear equation that is between 0 and 1. The logistic function is:

$$\text{logistic}(\eta) = \frac{1}{1 + exp(-\eta)}$$

For classification, the probability is between 0 and 1.

$$P(y^{(i)} = 1) = \frac{1}{1 + exp(-(\beta_0 + \beta_1 x_1^{(i)} + \ldots + \beta_p x_p^{(i)}))}$$

For binary logistic regression, a single categorical variable is: $\pi = Pr\ (Y = 1|X = x)$ where Y is a binary response variable. Yi=1 if the trait is present in observation i, and Yi=0 if it is not present. Therefore,

π_i=Pr(Y_i=1|X_i=x_i)=exp(β_0+$\beta_1 x_i$)/(1+exp(β_0+β_1 x_i)).

[1] Ali Shafigh Aski, Navid Khalilzadeh Sourati, Pacific Science Review A: Natural Science and Engineering, Volume 18, Issue 2, July 2016, https://www.sciencedirect.com/science/article/pii/S2405882316300412

[2] Christina V, Karpagavalli S, Suganya G, A study on email spam filtering techniques, International Journal of Computer Applications, http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.206.4041&rep=rep1&type=pdf

[3] Interpretable Machine Learning, https://christophm.github.io/interpretable-ml-book/logistic.html#advantages-and-disadvantages

Or,

$$\begin{aligned}\operatorname{logit}(\pi_i) &= \log\left(\frac{\pi_i}{1-\pi_i}\right) \\ &= \beta_0 + \beta_1 x_i \\ &= \beta_0 + \beta_1 x_{i1} + \ldots + \beta_k x_{ik}\end{aligned}$$

[4]

3.2 Gradient Boosting

Gradient boosting is an algorithm for building predictive models by converting weak models into a single strong model. There is a loss function $L(y,F(x))$. The goal is to find:

$$\hat{F} = \arg\min_{F} \mathbb{E}_{x,y}[L(y, F(x))]$$

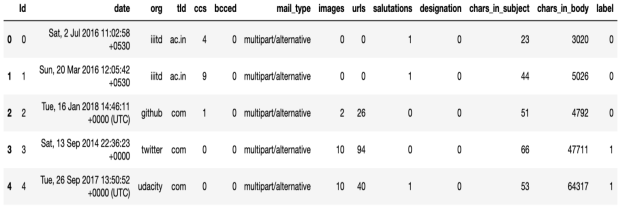
That is:

$$\hat{F}(x) = \sum_{i=1}^{M} \gamma_i h_i(x) + const$$

where hi(x) is the weak learner[5].

4. Data

The data set is from Kaggle competition which the meta data is extracted from emails. One train data file and one test data file are provided. Train data file contains 14064 observations and test file includes 6029 rows. There are 13 features included. Figure 1 is a short cut of the training data.



	Id	date	org	tld	ccs	bcced	mail_type	images	urls	salutations	designation	chars_in_subject	chars_in_body	label
0	0	Sat, 2 Jul 2016 11:02:58 +0530	iiitd	ac.in	4	0	multipart/alternative	0	0	1	0	23	3020	0
1	1	Sun, 20 Mar 2016 12:05:42 +0530	iiitd	ac.in	9	0	multipart/alternative	0	0	1	0	44	5026	0
2	2	Tue, 16 Jan 2018 14:46:11 +0000 (UTC)	github	com	1	0	multipart/alternative	2	26	0	0	51	4792	0
3	3	Sat, 13 Sep 2014 22:36:23 +0000	twitter	com	0	0	multipart/alternative	10	94	0	0	66	47711	1
4	4	Tue, 26 Sep 2017 13:50:52 +0000 (UTC)	udacity	com	0	0	multipart/alternative	10	40	1	0	53	64317	1

Fig 1. Training data screenshot.

Here are the definitions for each feature:

- Id - the id of the email
- date - date and time at which mail was received
- org - the organisation of the sender
- tld - top-level domain of the sender's organisation
- ccs - number of people cc'd in the email
- bcced - is the receiver bcc'd in the email
- mail_type - the type of the mail body
- images - number of images in the mail body
- urls - number of urls in the mail body
- salutations - is salutation used in the email?
- designation - is designation of the sender mentioned in the email?
- chars_in_subject - number of characters in the email's subjectchars_in_body - number of characters in the email's body
- label - the label of the email (1 -> Promotional, 0 -> personal)

5. Experiment

This section records our experiment steps and results.

5.1 Data cleaning

First of all, we checked if there was any null values in the data set. Only column “org” and “tld” contained null value. Since we did not know if they might affect our test results, we decided to keep those null values

[4] Analysis of Discrete Data, PennState Eberly College of Science, https://newonlinecourses.science.psu.edu/stat504/node/150/

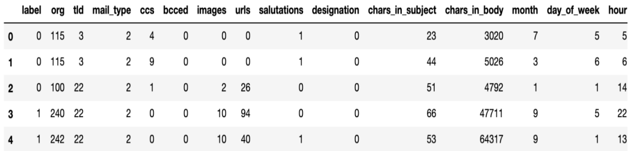
[5] Gradient Boosting Algorithm Theory, https://blog.csdn.net/wong2016/article/details/88851819

but filled “Not mentioned” in the empty cells both in train and test dataset.

Secondly, in our hypothesis, the time might play an importance role for classification, but the original data only contains one column named “date” that includes month, day of week and hours. Thus, we dropped this column, but added three new columns that the month, day of week and hours are extracted from “date” column.

Lastly, we checked the number of unique values in nominal variables. There are over 200 variables in “org”, over 100 variables in “tld” and 7 variables in “mail_type”. There are too many categories for columns “org” and “tld” which would increase the dimension of our model that we might end up with no enough data to accurately train the model, but for now, we decided to keep those many dimensions, but encode those columns to feed them to the model.

Fig 2 shows the final version of our dataset.



	label	org	tld	mail_type	ccs	bcced	images	urls	salutations	designation	chars_in_subject	chars_in_body	month	day_of_week	hour
0	0	115	3	2	4	0	0	0	1	0	23	3020	7	5	5
1	0	115	3	2	9	0	0	0	1	0	44	5026	3	6	6
2	0	100	22	2	1	0	2	26	0	0	51	4792	1	1	14
3	1	240	22	2	0	0	10	94	0	0	66	47711	9	5	22
4	1	242	22	2	0	0	10	40	1	0	53	64317	9	1	13

Fig 2. The training data set after cleaning up.

The columns “org”, “tld” and “mail_type” are converted to numbers; the column “date” is dropped and three new columns “month”, “day of week” and “hour” are added; no null values contained.

5.2 Exploratory Data Analysis

After cleaning up dataset, we did some basic feature analysis to check the importance of 13 features. When we selected this topic, the first factor we came up with is time since we barely receive personal emails at midnight or weekend. Hence, we firstly checked the relationship of hours, day of week and month. Fig 3, 4 and 5 represent the relationship between hours, day of week, month, and labels which indicate personal and promotional email.

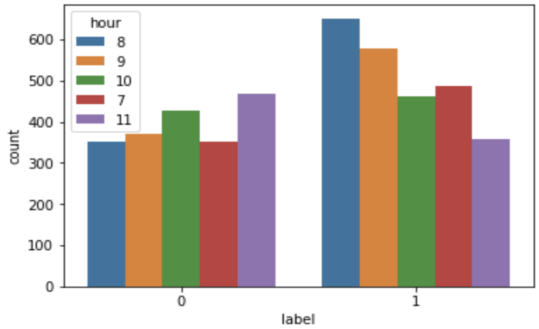


Fig 3. Top 5 hours receives most emails.

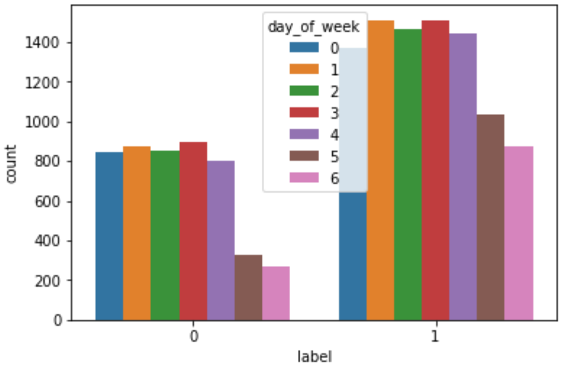


Fig 4. The number of personal and promotional emails received at different day of week.

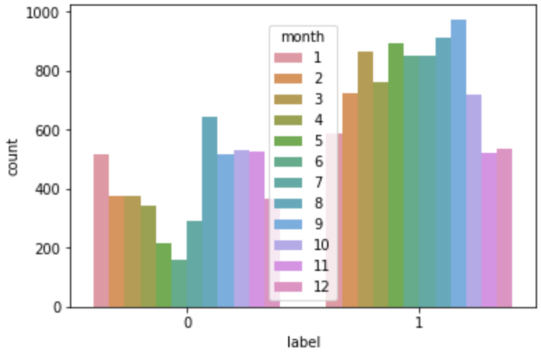


Fig 5. The number of personal and promotional emails received in different month.

Apparently, there is no significant differences between hours or day of week with labels. If an email received at 8 am, it has higher chance to be a promotional email, but it does not mean the email could not be a personal email. The distributions for personal and promotional emails received at different day of a week are very similar. From Monday to Friday, large amount emails received, but on the weekend, the amount decreases. The reason might be that employees usually do not work during the weekend unless emergency happens. As month, there are huge differences between number of personal and promotional emails received in May, June and July which indicates the high probability to receive promotional emails during these months, but it is not absolute to receive no personal emails. Therefore, time seems not play an important role in classifying emails, then how does other features perform?

Except date, we selected “org”, “tld” and “ccs” to draw relationship plots which fig 6, 7, and 8 show below.

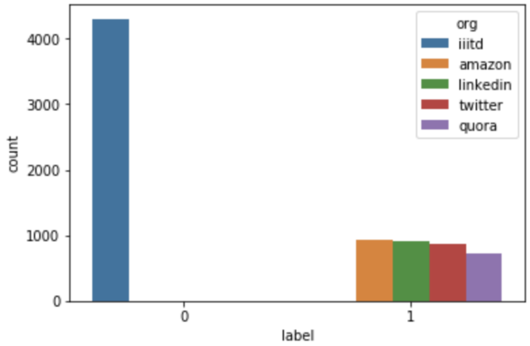


Fig 6. The number of personal and promotional emails received that are sends by different organizations.

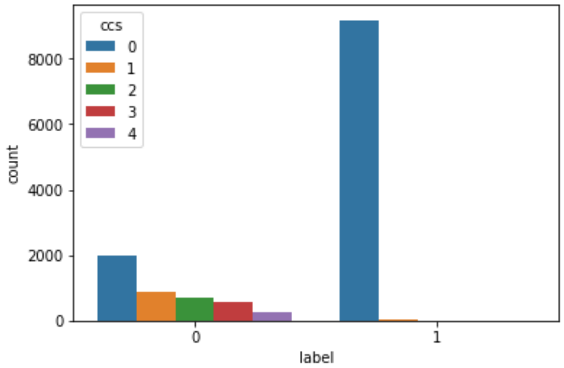


Fig 7. The number of personal and promotional emails received that how many people cc’d in the emails.

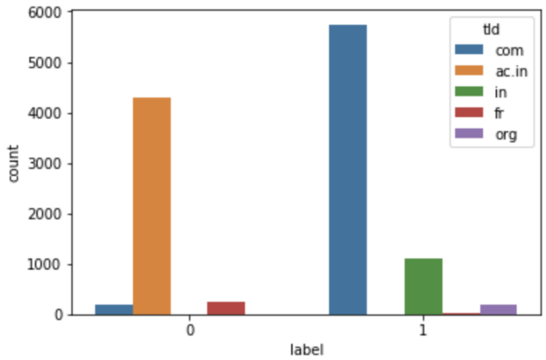


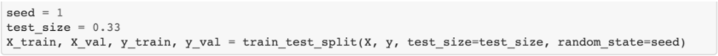
Fig 8. The number of personal and promotional emails received that addresses are ended with different top-level-domains.

Those three features seem play more important role to classify emails. As organization, all emails are sent from liiitd are personal. The more people cc’d in the email, the higher chance the email is personal. And, if an email address ends with ac.in or fr, it is considered as personal, but if it ends with in and org, it should be labeled as promotional. “com” has higher probability to be a promotional, but you should at least take a look.

5.3 Model Fitting

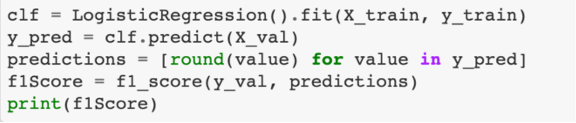
After basic data analysis, we thought logistic regression model might be a good start since it could be used to answer classification questions and returns probability.

We split train data file to train and validation data set to test.



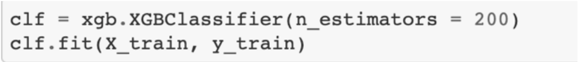
```
seed = 1
test_size = 0.33
X_train, X_val, y_train, y_val = train_test_split(X, y, test_size=test_size, random_state=seed)
```

Then fitting the train set and predicting the results by using validation data. Finally, returning the f1score which the Kaggle competition is used to evaluate test results.



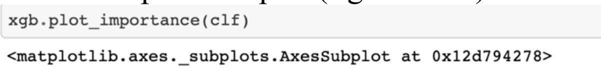
```
clf = LogisticRegression().fit(X_train, y_train)
y_pred = clf.predict(X_val)
predictions = [round(value) for value in y_pred]
f1Score = f1_score(y_val, predictions)
print(f1Score)
```

However, we noticed that xgboost has better running time for large input n, so we decided to try xgb classifier (gradient boosting) to pursue a better performance.



```
clf = xgb.XGBClassifier(n_estimators = 200)
clf.fit(X_train, y_train)
```

For now, we still did not know how features perform in our model; hence, we plot the features importance plot (fig 9 shows).



```
xgb.plot_importance(clf)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x12d794278>
```

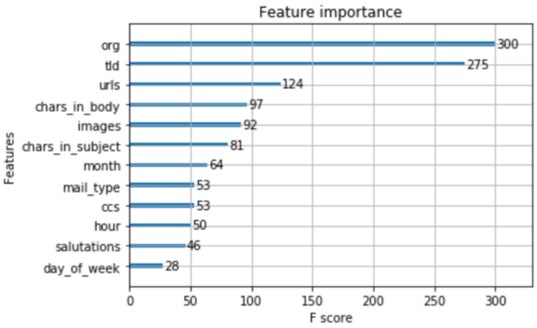


Fig 9. Feature importance

As we analyzed before, the feature "day of week" and "hour" do not indicate tight relationship between labels while "org", "tld" and "urls" represent high importance.

Even though it returns a not bad f1 score, is there any other stuff that might improve the model performance? Apparently, the model requires parameters, so we tried to find the best combination of "max_depth" and "n_estimaters" that returns best result.

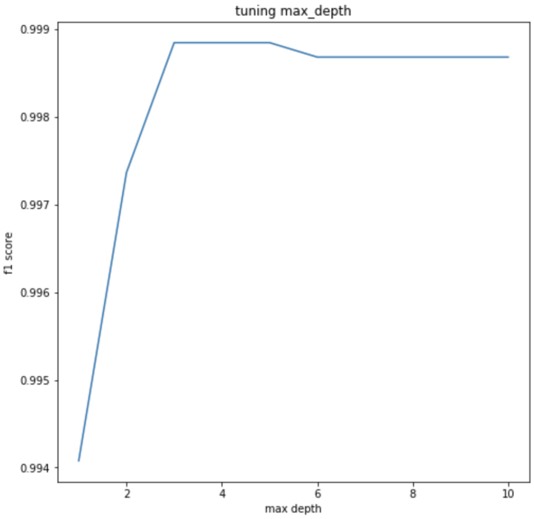


Fig 10. Relationship between max_depth and f1 score

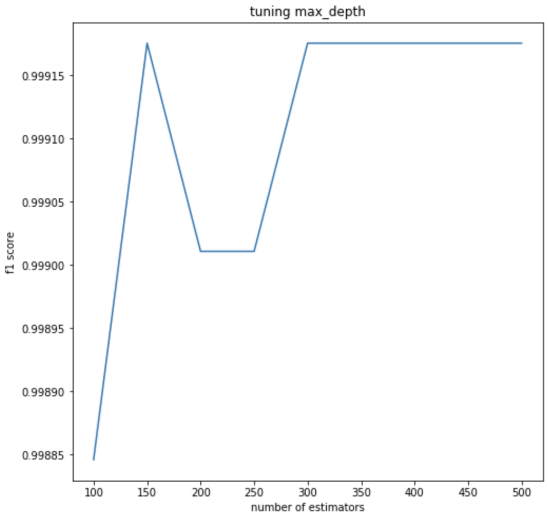
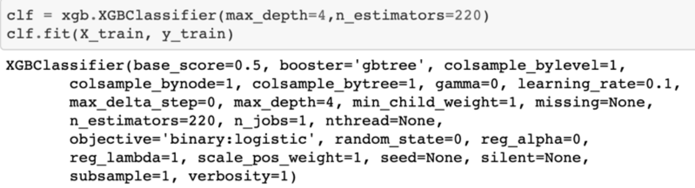


Fig 11. Relationship between number of estimators and f1 score.

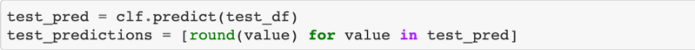
Fig 10 and 11 indicates that the best f1 scores given when "max_depth" is from 3 to 5 and when "number of estimators is 150 and higher than 300, but we have to test each combination of those two parameters. Based on Kaggle competition results, when "max_depth" is 4 and "n_estimators" equals to 200, the f1 score is the highest.

Finally, we realized that the gradient boosting algorithm performed better than the logistic regression model on validation dataset, so we trained the complete train data file and applied it on test data file.



```
clf = xgb.XGBClassifier(max_depth=4,n_estimators=220)
clf.fit(X_train, y_train)
```

```
XGBClassifier(base_score=0.5, booster='gbtree', colsample_bylevel=1,
       colsample_bynode=1, colsample_bytree=1, gamma=0, learning_rate=0.1,
       max_delta_step=0, max_depth=4, min_child_weight=1, missing=None,
       n_estimators=220, n_jobs=1, nthread=None,
       objective='binary:logistic', random_state=0, reg_alpha=0,
       reg_lambda=1, scale_pos_weight=1, seed=None, silent=None,
       subsample=1, verbosity=1)
```



```
test_pred = clf.predict(test_df)
test_predictions = [round(value) for value in test_pred]
```

5.4 Model Performance

We predicted the results, saved in csv file and submitted on Kaggle competition. The final performance is evaluated by F1 score which measures accuracy using the statistics precision p and recall r. The function is defined as:

$$F1 = 2\frac{p \cdot r}{p + r} \quad \text{where} \quad p = \frac{tp}{tp + fp}, \quad r = \frac{tp}{tp + fn}$$

The precision and recall are weighted equally, and the best score is 1.

The performance of logistic regression is 0.96 and gradient boosting earns a score around 0.99. Both of models perform not bad, but gradient boosting works better.

6. Error Analysis

For the logistic regression model, firstly, the logistic regression model prefers balanced dataset, but our data contains 65.4% promotional emails and 34.6% personal email. The not balanced data might influence the final test result. Secondly, adding too much features might result in overfitting that reduces the generalizability of the model[6].

For the gradient boosting algorithm, the max depth and number of estimators are not fixed. We tested several numbers and picked the number with highest f1 score, but we do not know if there is any other better selection with higher score than 0.99. Also, there are too many categories for variable "org" and "tld" which might increase the dimension of our model that might affect us to accurately train the model.

7. Conclusion

A promotional email mistakenly classified into personal does not hurt, but if a personal email is labelled as promotional, it might cause a huge loss. Experts are working hard to develop the spam filter to avoid loss. There are varies of models could be used to develop spam filter. The logistic regression model or gradient boosting algorithm might not be the one with best performance, but they prove that it is possible to classify emails accurately with the metadata extracted from the original emails.

Even though we get high scores on the test result, we are not satisfied. Next step, we will try to use other deep learning methods such as Keras to improve performance.

References

Ali Shafigh Aski, Navid Khalilzadeh Sourati, Pacific Science Review A: Natural Science and Engineering, Volume 18, Issue 2, July 2016, https://www.sciencedirect.com/science/article/pii/S2405882316300412

Christina V, Karpagavalli S, Suganya G, A study on email spam filtering techniques, International Journal of Computer Applications, http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.206.4041&rep=rep1&type=pdf

Interpretable Machine Learning, https://christophm.github.io/interpretable-ml-book/logistic.html#advantages-and-disadvantages

Analysis of Discrete Data, PennState Eberly College of Science, https://newonlinecourses.science.psu.edu/stat504/node/150/

[6] Statistics Solutions, https://www.statisticssolutions.com/what-is-logistic-regression/

Gradient Boosting Algorithm Theory, https://blog.csdn.net/wong2016/article/details/88851819

Statistics Solutions, https://www.statisticssolutions.com/what-is-logistic-regression/