EMAIL SPAM DETECTION

A Machine Learning Approach

Instructor: Rong Liu

BIA 660 – C – 19S

Author: Zixuan Wang, Shangjun Jiang,
Haodong Zhao, Rumeng Zuo

Table of Contents

I. Project background

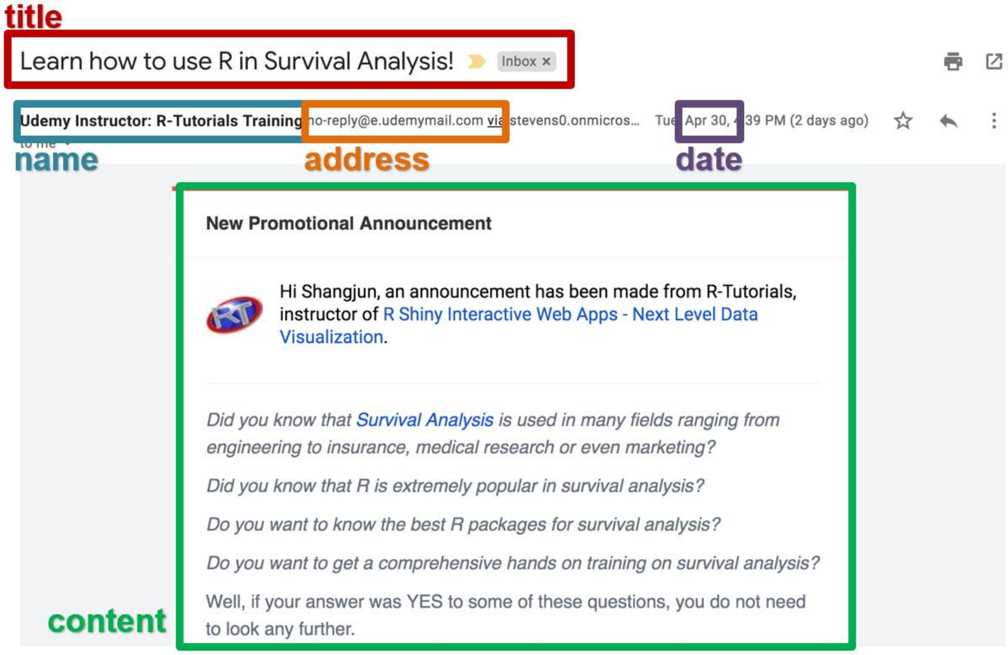
Email is the primary source of communication in the business world regardless of schoolwork or business event. People receive decades of emails every day but some of them called ham are useful but some we called spam is meaningless. Unfortunately, sometimes ham with important information would be classified into spam incorrectly. So, in order to avoid this situation and try to understand the classifier better, we would like to apply text mining and machine learning model to figure it out. Also, this understanding will promote business events which are related to outreach by emails like business development.

II. Data Collection & Processing

Before analytics, we have to collect text data from real-life emails and process them to make sure our model can perform well.

i. Scraping data from email

Firstly, we observed our emails and selected those useful parts shown in Fig.1.



Learn how to use R in Survival Analysis!

Udemy Instructor: R-Tutorials Training no-reply@e.udemymail.com via stevens0.onmicros... Tue Apr 30, 4:39 PM (2 days ago)

New Promotional Announcement

Hi Shangjun, an announcement has been made from R-Tutorials, instructor of R Shiny Interactive Web Apps - Next Level Data Visualization.

Did you know that Survival Analysis is used in many fields ranging from engineering to insurance, medical research or even marketing?

Did you know that R is extremely popular in survival analysis?

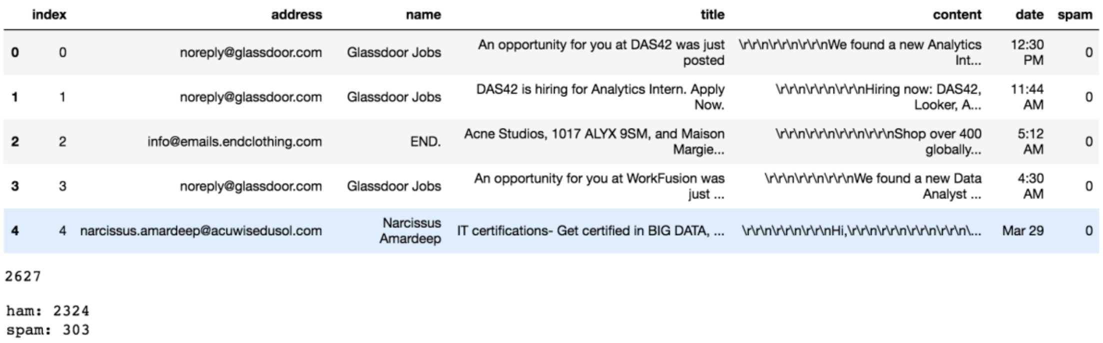
Do you want to know the best R packages for survival analysis?

Do you want to get a comprehensive hands on training on survival analysis?

Well, if your answer was YES to some of these questions, you do not need to look any further.

Fig.1 5 parts, title, name, address, date, content of each email

For this part, the main technology is using packages like Beatifulsoup and Selenium in Python to program a dynamic web crawler to scrape data from the different mailbox. We collected emails from ham and spam separately using two different mailboxes which belongs to 2 different group members and to merge the ham data set and spam data set. At last, we got two data set. One is an 1137 rows * 6 columns matrix and the other one is 454 rows * 6 columns. Except for those 5 parts, address, sender name, title, content, and date, the last column is "spam" which is the label to recognize if this email is spam or ham. Fig.2 show our data set format.



	index	address	name	title	content	date	spam
0	0	noreply@glassdoor.com	Glassdoor Jobs	An opportunity for you at DAS42 was just posted	\r\r\n\r\r\n\r\r\nWe found a new Analytics Int...	12:30 PM	0
1	1	noreply@glassdoor.com	Glassdoor Jobs	DAS42 is hiring for Analytics Intern. Apply Now.	\r\r\n\r\r\n\r\r\nHiring now: DAS42, Looker, A...	11:44 AM	0
2	2	info@emails.endclothing.com	END.	Acne Studios, 1017 ALYX 9SM, and Maison Margie...	\r\r\n\r\r\n\r\r\n\r\r\nShop over 400 globally...	5:12 AM	0
3	3	noreply@glassdoor.com	Glassdoor Jobs	An opportunity for you at WorkFusion was just ...	\r\r\n\r\r\n\r\r\nWe found a new Data Analyst ...	4:30 AM	0
4	4	narcissus.amardeep@acuwisedusol.com	Narcissus Amardeep	IT certifications- Get certified in BIG DATA, ...	\r\r\n\r\r\n\r\r\nHi,\r\r\n\r\r\n\r\r\n\r\r\n\...	Mar 29	0

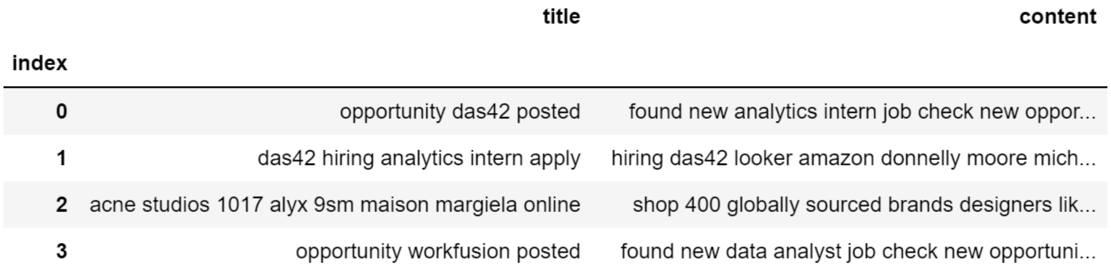
```
2627

ham: 2324
spam: 303
```

Fig.2 Data scraping from eamils

ii. Preprocess text data

Firstly, we find and remove those empty emails whose content should be picture or graph which we cannot deal with.

At this part, we would use functions in nltk and regular regression to process our data set. After removing those basic stop words or special signal like Emoji, we transform the text into a numeric vector which will be used in later analysis. The Fig.3 shows us the cleaning text data.



	title	content
index		
0	opportunity das42 posted	found new analytics intern job check new oppor...
1	das42 hiring analytics intern apply	hiring das42 looker amazon donnelly moore mich...
2	acne studios 1017 alyx 9sm maison margiela online	shop 400 globally sourced brands designers lik...
3	opportunity workfusion posted	found new data analyst job check new opportuni...

Fig.3 Cleaning text data including email title and content

III. Exploratory Data Analysis (EDA)

In order to better understand the dataset and obtain more insight from it, as well as prepared our data for modelling in the future, we decided to perform EDA towards it. In this phase, there are four parts in total.

i. Word Cloud

First, we would like to generate a word cloud for our ‘title’ and ‘content’ column and get the knowledge about which words are most frequently appeared in our email. Therefore, we combine these two columns together and make it as a new column ‘combine’. Then we use it to generate word cloud for emails from two types of label by each teammate. Here below shows one teammate’s EDA report as a demo. The word cloud shows as figure 4.

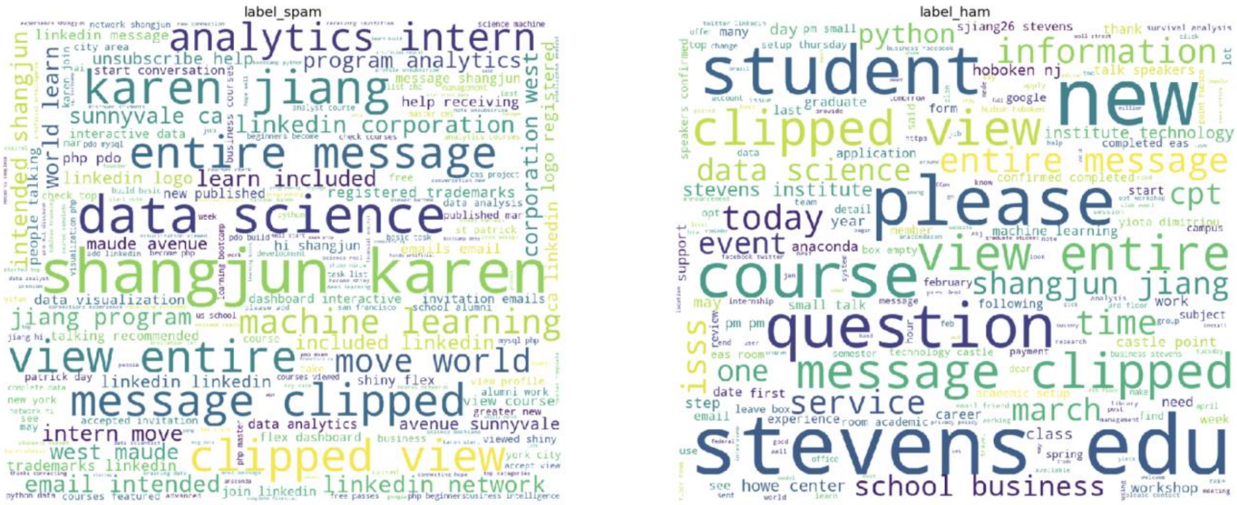


Fig.4 Word Cloud for two labels of email

From Fig 4, we can see that in the spam label of email word cloud, the mailbox owner’s name has large proportion. Also, there are same words show in both labels word cloud, therefore, we need to remove the words that have no function on helping to distinguish whether the email is spam or not so to generate the word cloud again to see the difference.

ii. Bar Chart for Words by Frequency

In order to take a deep look at the top frequently appeared words, we generate a bar chart for top 30 most frequently appeared words for both label of emails. The bar chart show as figure 5.

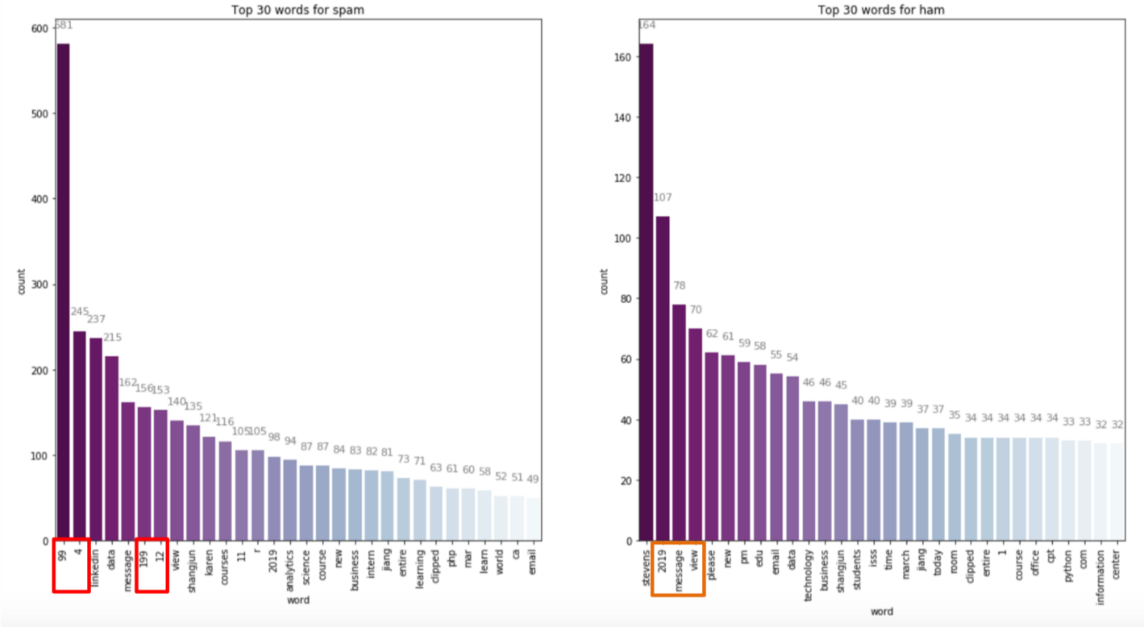
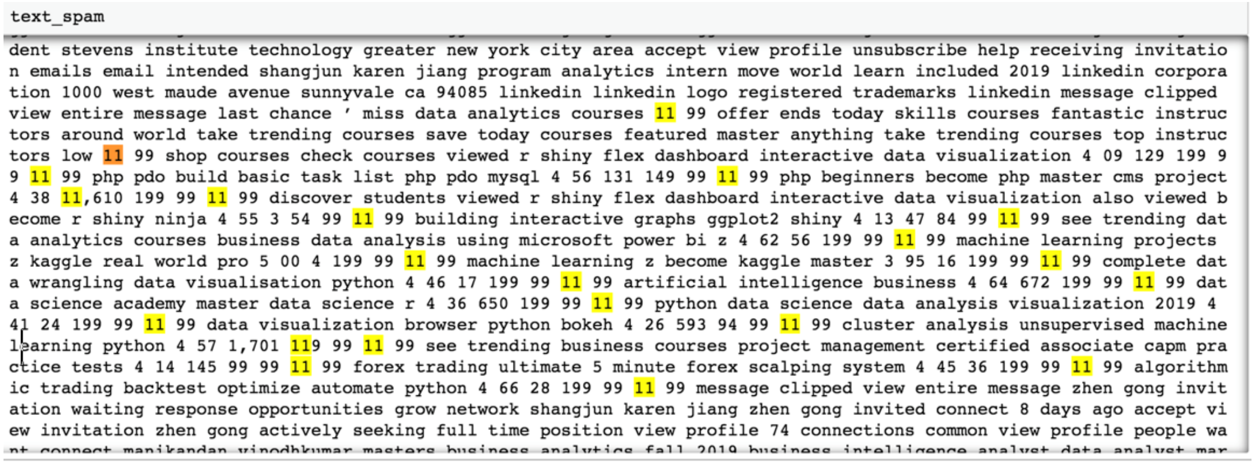


Fig.5 Bar chart for top 30 most frequently appeared words of two labels of email

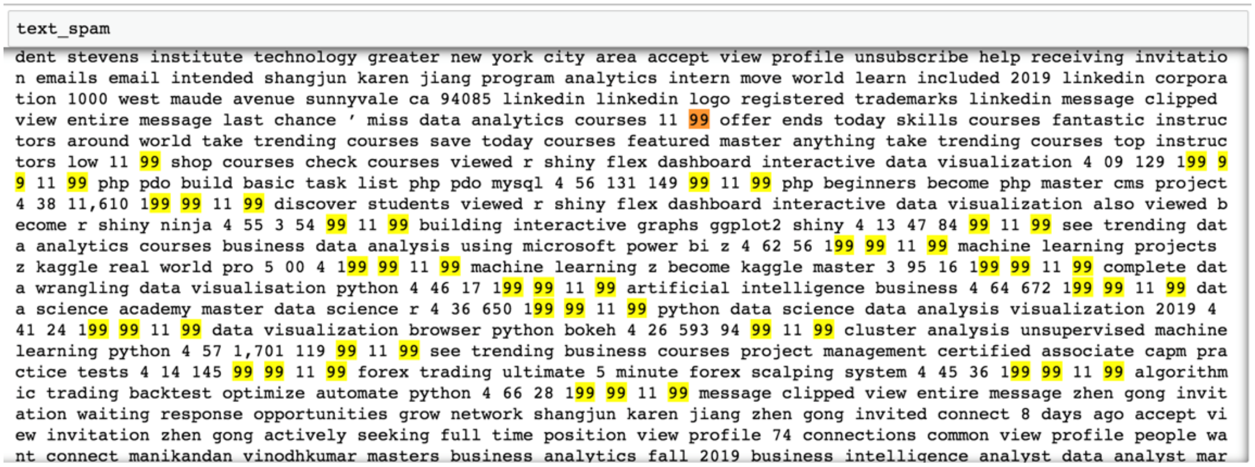
As showed in Fig 5 red box, there are some meaningless number showed in spam label of email. Also, as show in Fig 5 orange box, there are some same words that appeared in both labels of emails. In order to get a more distinguished EDA, we decide to delete these words.

iii. Data Cleanup and 2nd round EDA

To clean up the words that we mentioned above, we use regular expression to do this work. In order to figure the meaningless words out, we first go back to the text data and see, then we find out that all the numbers, such as 11, 99, 199, are from Udemy courses promotion email as showed in Fig 6.



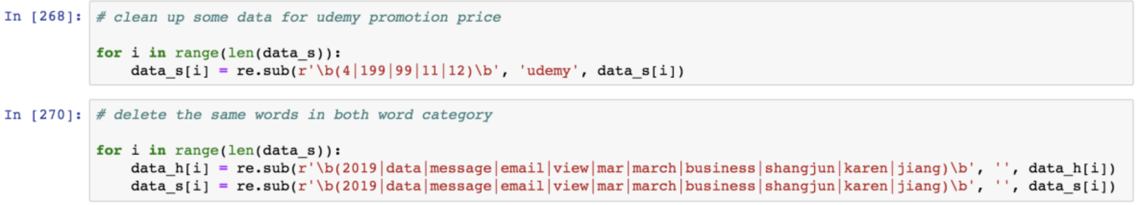
```
text_spam
dent stevens institute technology greater new york city area accept view profile unsubscribe help receiving invitatio
n emails email intended shangjun karen jiang program analytics intern move world learn included 2019 linkedin corpora
tion 1000 west maude avenue sunnyvale ca 94085 linkedin linkedin logo registered trademarks linkedin message clipped
view entire message last chance ’ miss data analytics courses 11 99 offer ends today skills courses fantastic instruc
tors around world take trending courses save today courses featured master anything take trending courses top instruc
tors low 11 99 shop courses check courses viewed r shiny flex dashboard interactive data visualization 4 09 129 199 9
9 11 99 php pdo build basic task list php pdo mysql 4 56 131 149 99 11 99 php beginners become php master cms project
4 38 11,610 199 99 11 99 discover students viewed r shiny flex dashboard interactive data visualization also viewed b
ecome r shiny ninja 4 55 3 54 99 11 99 building interactive graphs ggplot2 shiny 4 13 47 84 99 11 99 see trending dat
a analytics courses business data analysis using microsoft power bi z 4 62 56 199 99 11 99 machine learning projects
z kaggle real world pro 5 00 4 199 99 11 99 machine learning z become kaggle master 3 95 16 199 99 11 99 complete dat
a wrangling data visualisation python 4 46 17 199 99 11 99 artificial intelligence business 4 64 672 199 99 11 99 dat
a science academy master data science r 4 36 650 199 99 11 99 python data science data analysis visualization 2019 4
41 24 199 99 11 99 data visualization browser python bokeh 4 26 593 94 99 11 99 cluster analysis unsupervised machine
learning python 4 57 1,701 119 99 11 99 see trending business courses project management certified associate capm pra
ctice tests 4 14 145 99 99 11 99 forex trading ultimate 5 minute forex scalping system 4 45 36 199 99 11 99 algorithm
ic trading backtest optimize automate python 4 66 28 199 99 11 99 message clipped view entire message zhen gong invit
ation waiting response opportunities grow network shangjun karen jiang zhen gong invited connect 8 days ago accept vi
ew invitation zhen gong actively seeking full time position view profile 74 connections common view profile people wa
nt connect manikandan vinodhkumar masters business analytics fall 2019 business intelligence analyst data analyst mar
```



```
text_spam
dent stevens institute technology greater new york city area accept view profile unsubscribe help receiving invitatio
n emails email intended shangjun karen jiang program analytics intern move world learn included 2019 linkedin corpora
tion 1000 west maude avenue sunnyvale ca 94085 linkedin linkedin logo registered trademarks linkedin message clipped
view entire message last chance ’ miss data analytics courses 11 99 offer ends today skills courses fantastic instruc
tors around world take trending courses save today courses featured master anything take trending courses top instruc
tors low 11 99 shop courses check courses viewed r shiny flex dashboard interactive data visualization 4 09 129 199 9
9 11 99 php pdo build basic task list php pdo mysql 4 56 131 149 99 11 99 php beginners become php master cms project
4 38 11,610 199 99 11 99 discover students viewed r shiny flex dashboard interactive data visualization also viewed b
ecome r shiny ninja 4 55 3 54 99 11 99 building interactive graphs ggplot2 shiny 4 13 47 84 99 11 99 see trending dat
a analytics courses business data analysis using microsoft power bi z 4 62 56 199 99 11 99 machine learning projects
z kaggle real world pro 5 00 4 199 99 11 99 machine learning z become kaggle master 3 95 16 199 99 11 99 complete dat
a wrangling data visualisation python 4 46 17 199 99 11 99 artificial intelligence business 4 64 672 199 99 11 99 dat
a science academy master data science r 4 36 650 199 99 11 99 python data science data analysis visualization 2019 4
41 24 199 99 11 99 data visualization browser python bokeh 4 26 593 94 99 11 99 cluster analysis unsupervised machine
learning python 4 57 1,701 119 99 11 99 see trending business courses project management certified associate capm pra
ctice tests 4 14 145 99 99 11 99 forex trading ultimate 5 minute forex scalping system 4 45 36 199 99 11 99 algorithm
ic trading backtest optimize automate python 4 66 28 199 99 11 99 message clipped view entire message zhen gong invit
ation waiting response opportunities grow network shangjun karen jiang zhen gong invited connect 8 days ago accept vi
ew invitation zhen gong actively seeking full time position view profile 74 connections common view profile people wa
nt connect manikandan vinodhkumar masters business analytics fall 2019 business intelligence analyst data analyst mar
```

Fig.6 Text from ‘title’ and ‘content’ of spam email

Therefore, we use regular expression to change all the price number words into ‘udemy’ for helping us better understand the source of the words and process data. Other than process the meaningless data, we also delete all the identical words in the top 30 most frequently appeared word list for both labels of email. Python code show as Fig 7.



```
In [268]: # clean up some data for udemy promotion price

          for i in range(len(data_s)):
              data_s[i] = re.sub(r'\b(4|199|99|11|12)\b', 'udemy', data_s[i])

In [270]: # delete the same words in both word category

          for i in range(len(data_s)):
              data_h[i] = re.sub(r'\b(2019|data|message|email|view|mar|march|business|shangjun|karen|jiang)\b', '', data_h[i])
              data_s[i] = re.sub(r'\b(2019|data|message|email|view|mar|march|business|shangjun|karen|jiang)\b', '', data_s[i])
```

Fig.7 Python code for data cleanup

After cleaning up the meaningless words, we use the data to generate the word cloud for text in both labels of email and plot the bar chart for the top 30 most frequently appeared words in both labels of emails again.

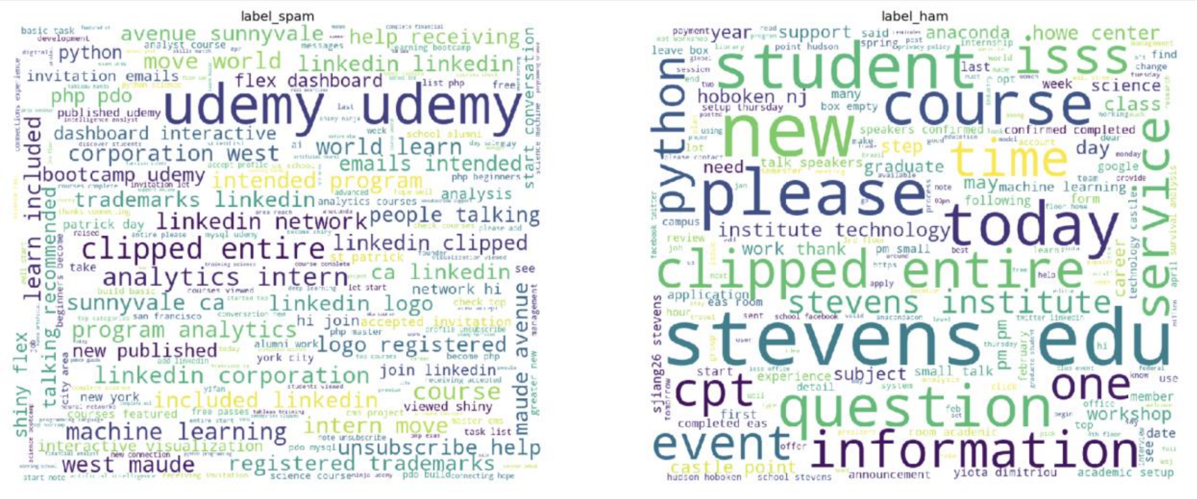


Fig.8 2nd Word Cloud for two labels of email

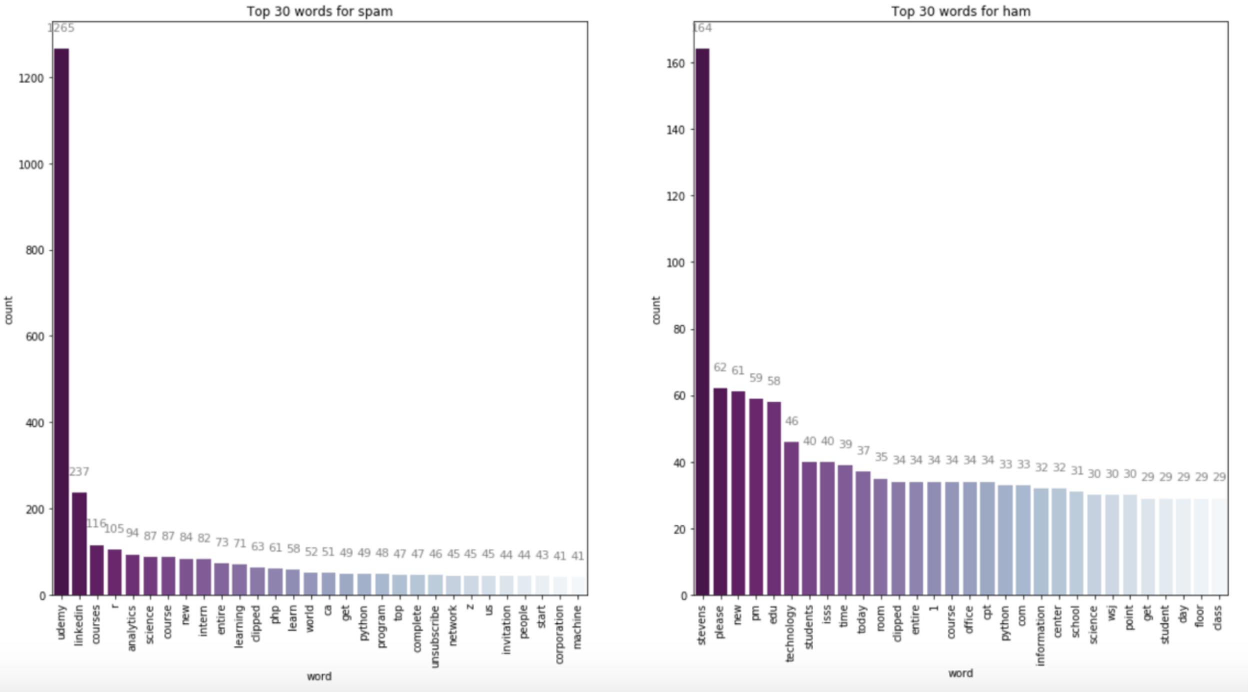


Fig.9 2nd Bar chart for top 30 most frequently appeared words of two labels of email

As showed in Fig 8 and Fig 9, after data cleanup, the most frequently appeared words in both are much distinguishable. In the spam mailbox, most of the emails are from Udemy and LinkedIn, and for the ham mailbox, most of the emails are from Stevens, which is make sense since we are scraping data from school's email.

iv. Pie chart for suffix of sender's email address

In order to better understand who sending the emails to us in both labels of emails, we extract the suffix of the email sender then generate a pie chart for it base on its frequency. As showed in Fig 10, in the mailbox of this team member, most of the spam emails are from LinkedIn.com, then is Udemy.com. And for the ham emails, almost half of them are from Stevens.edu.

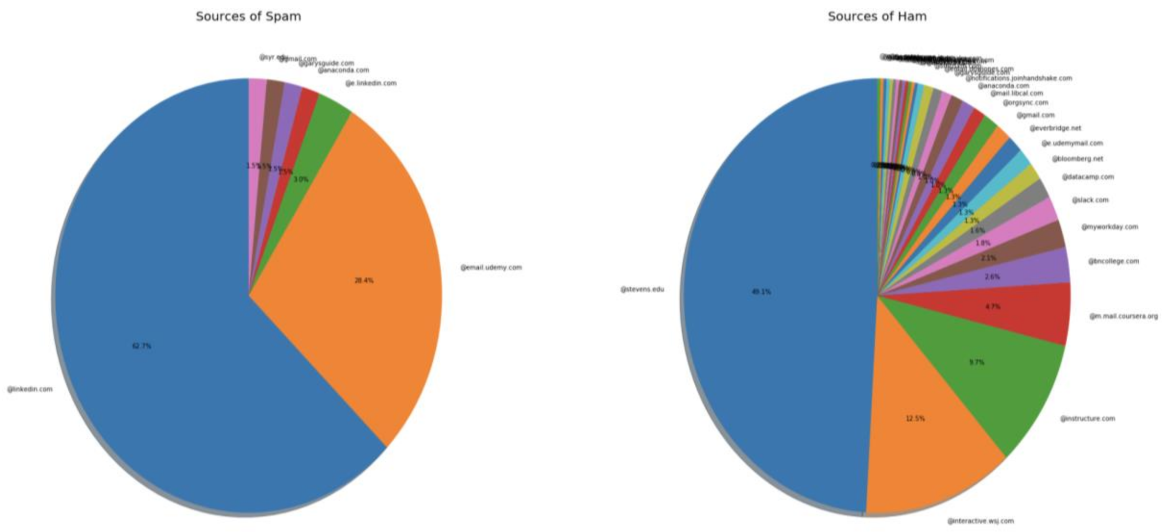
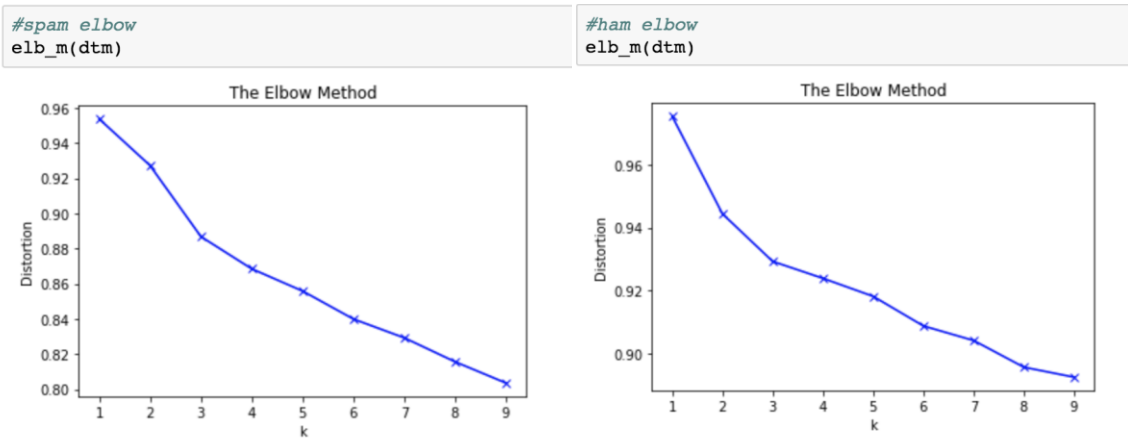


Fig.10 2nd Bar chart for top 30 most frequently appeared words of two labels of email

IV. Modelling: Clustering

After the EDA and perform data cleanup, here we come to the Modelling part. In order to find out whether the email topic has relationship with they will go the spam mailbox or not. We first use clustering algorithms to build model. Since emails do not come with a topic, to solve this problem, we need to manually separate the emails into several topic groups. Every team member performed the clustering modelling for their email separately and here below is a demo for one team member’s model.

First, we generate a scree plot to determine how many clusters would have the best performance of the model for both labels of emails. The scree plots for both emails show as Fig 11.

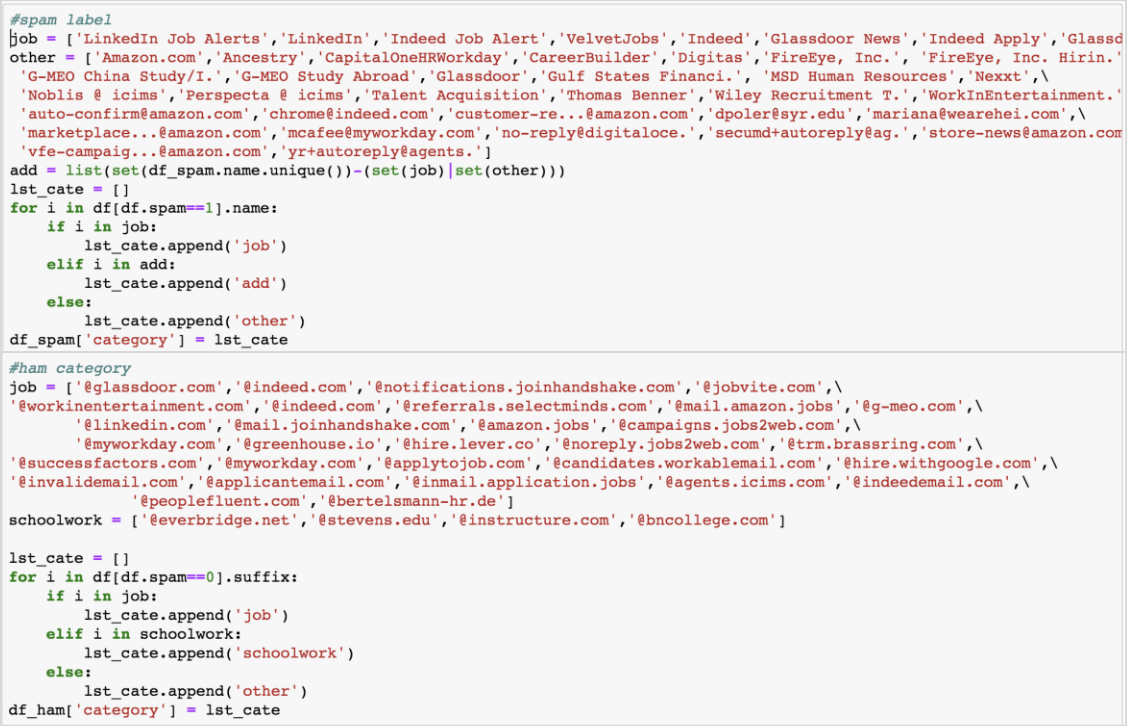


```
#spam elbow
elb_m(dtm)
```

```
#ham elbow
elb_m(dtm)
```

Fig.11 Scree plots for two labels of email

As it shows in Fig 11, for both spam and ham emails, the best number for clusters is 3. Therefore, we manually separate the emails into three topics, job, connection from LinkedIn and other, by using the suffix of sender's email. Fig 12 shows the python code about how we cluster emails.

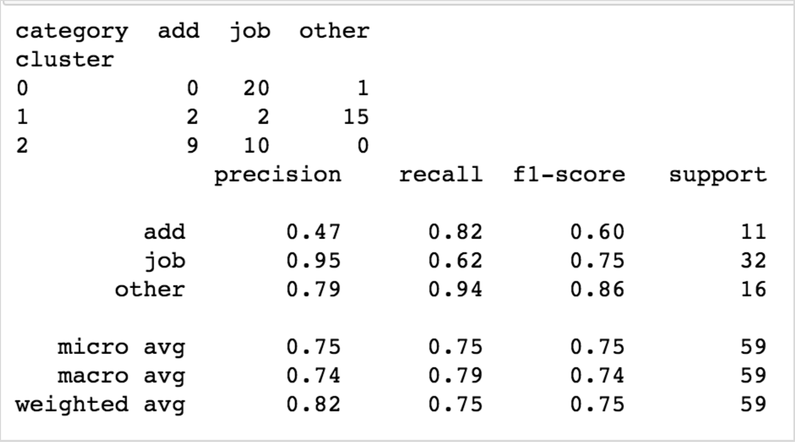


```
#spam label
job = ['LinkedIn Job Alerts','LinkedIn','Indeed Job Alert','VelvetJobs','Indeed','Glassdoor News','Indeed Apply','Glassd
other = ['Amazon.com','Ancestry','CapitalOneHRWorkday','CareerBuilder','Digitas','FireEye, Inc.', 'FireEye, Inc. Hirin.'
 'G-MEO China Study/I.','G-MEO Study Abroad','Glassdoor','Gulf States Financi.', 'MSD Human Resources','Nexxt',\
 'Noblis @ icims','Perspecta @ icims','Talent Acquisition','Thomas Benner','Wiley Recruitment T.','WorkInEntertainment.'
 'auto-confirm@amazon.com','chrome@indeed.com','customer-re...@amazon.com','dpoler@syr.edu','mariana@wearehei.com',\
 'marketplace...@amazon.com','mcafee@myworkday.com','no-reply@digitaloce.','secumd+autoreply@ag.','store-news@amazon.com
 'vfe-campaig...@amazon.com','yr+autoreply@agents.']
add = list(set(df_spam.name.unique())-(set(job)|set(other)))
lst_cate = []
for i in df[df.spam==1].name:
    if i in job:
        lst_cate.append('job')
    elif i in add:
        lst_cate.append('add')
    else:
        lst_cate.append('other')
df_spam['category'] = lst_cate
```

```
#ham category
job = ['@glassdoor.com','@indeed.com','@notifications.joinhandshake.com','@jobvite.com',\
'@workinentertainment.com','@indeed.com','@referrals.selectminds.com','@mail.amazon.jobs','@g-meo.com',\
       '@linkedin.com','@mail.joinhandshake.com','@amazon.jobs','@campaigns.jobs2web.com',\
       '@myworkday.com','@greenhouse.io','@hire.lever.co','@noreply.jobs2web.com','@trm.brassring.com',\
'@successfactors.com','@myworkday.com','@applytojob.com','@candidates.workablemail.com','@hire.withgoogle.com',\
'@invalidemail.com','@applicantemail.com','@inmail.application.jobs','@agents.icims.com','@indeedemail.com',\
             '@peoplefluent.com','@bertelsmann-hr.de']
schoolwork = ['@everbridge.net','@stevens.edu','@instructure.com','@bncollege.com']

lst_cate = []
for i in df[df.spam==0].suffix:
    if i in job:
        lst_cate.append('job')
    elif i in schoolwork:
        lst_cate.append('schoolwork')
    else:
        lst_cate.append('other')
df_ham['category'] = lst_cate
```

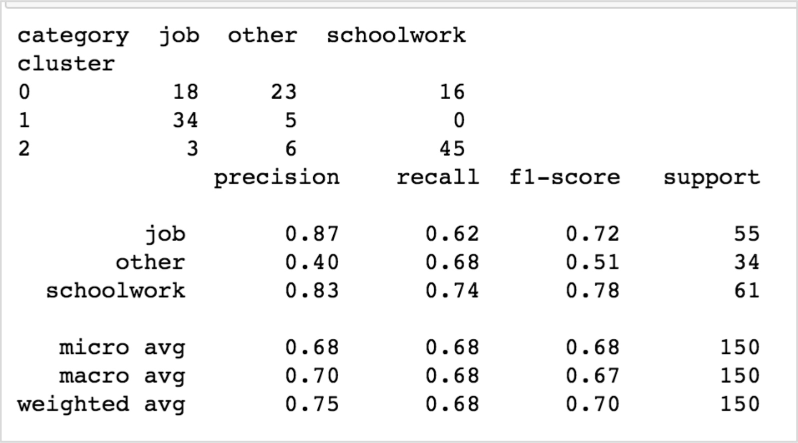
Fig.12 Python code for email cluster

After manually separate the clusters and add a column as 'topic' to the dataset, we build model by using topic model clustering and generate a classification report for it. The report for spam emails shows in Fig 13 and report for ham emails shows in Fig 14.



```
category  add  job  other
cluster
0           0   20      1
1           2    2     15
2           9   10      0
               precision    recall  f1-score   support

          add       0.47      0.82      0.60        11
          job       0.95      0.62      0.75        32
        other       0.79      0.94      0.86        16

    micro avg       0.75      0.75      0.75        59
    macro avg       0.74      0.79      0.74        59
 weighted avg       0.82      0.75      0.75        59
```

Fig.13 Classification report for spam emails

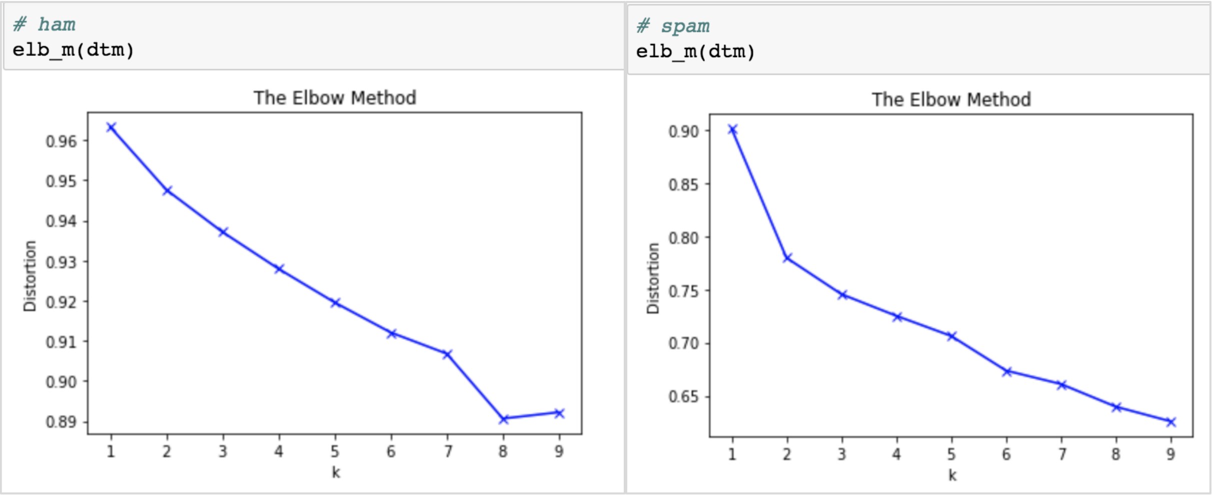


```
category  job  other  schoolwork
cluster
0          18     23          16
1          34      5           0
2           3      6          45
             precision    recall  f1-score   support

         job       0.87      0.62      0.72        55
       other       0.40      0.68      0.51        34
  schoolwork       0.83      0.74      0.78        61

   micro avg       0.68      0.68      0.68       150
   macro avg       0.70      0.68      0.67       150
weighted avg       0.75      0.68      0.70       150
```

Fig.14 Classification report for ham emails

As show from the above reports, we can get that the clustering performance is acceptable but not very good. Since for the cluster 2, the ‘add’ topic (connection application from LinkedIn) and the ‘job’ topic basically share the same proportion. And for ham emails, the clustering performance is a little bit poorer than the spam emails.

In order to evaluate if the clustering model is suitable for our data. We decided to take a look at another team member’s mailbox. First, we still use scree plot to determine the number of clusters.



```
# ham
elb_m(dtm)
```

```
# spam
elb_m(dtm)
```

Fig.15 Scree plots for two labels of email from another team member

As we can see from Fig 15, for ham emails, the number for clusters that have best modelling performance is 8, whereas for spam emails is 2. Obviously, the feasibility to perform clustering model towards their two types of emails vary from person to person. So, we decided change to use classification model.

V. Classification

i. Data balancing

By exploring the data, we found our data is an imbalance, which means the number of ham emails is different from the number of spam emails in our dataset. The imbalance data may affect the accuracy of the results. From the data distribution plot for the number of spam emails and ham emails, we can find the amount of ham emails is about 6 times the amount of spam emails. Which means if our classifiers recognize all emails are ham, there will still be high accuracy because most of our data is ham emails. Therefore, we believe that it is necessary to balance our data by setting ration of ham and spam as 1: 1. Compared with the re-sampling method we selected is under-sampled, randomly select the same amount of ham emails then append to all spam email. Fig.16 shows us result.

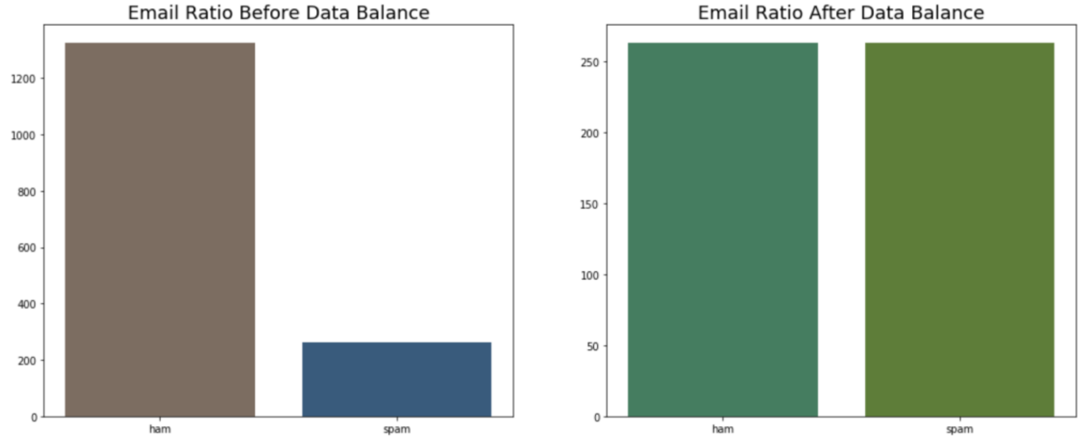


Fig.16 After and Before data balancing

ii. Data balancing

We successfully implemented machine learning algorithms of Linear Discriminant Analysis, Logistic Regression, Support Vector Machine, Gaussian Naïve Bayes and Multinomial Naïve Bayes. Accuracy can be computed by comparing actual test set values and predicted values (We use AUC score to judge whether the classification prediction results are good or bad). we could see how accurately the classifier or model can predict the success of emails. And we also implemented voting ensemble method to find the ensemble performance of the previous five classifiers. We tried to use the title and content of the emails to make classification predictions and test the performance of different classifiers. The method we used is to plot ROC curve. ROC curve is a graphical plot that illustrates the diagnostic ability of a binary classifier system as its discrimination threshold is varied. The ROC curve is created by plotting the true positive rate (TPR) against the false positive rate (FPR) at various threshold settings. The larger the area under the curve, the higher the accuracy rate the model has.

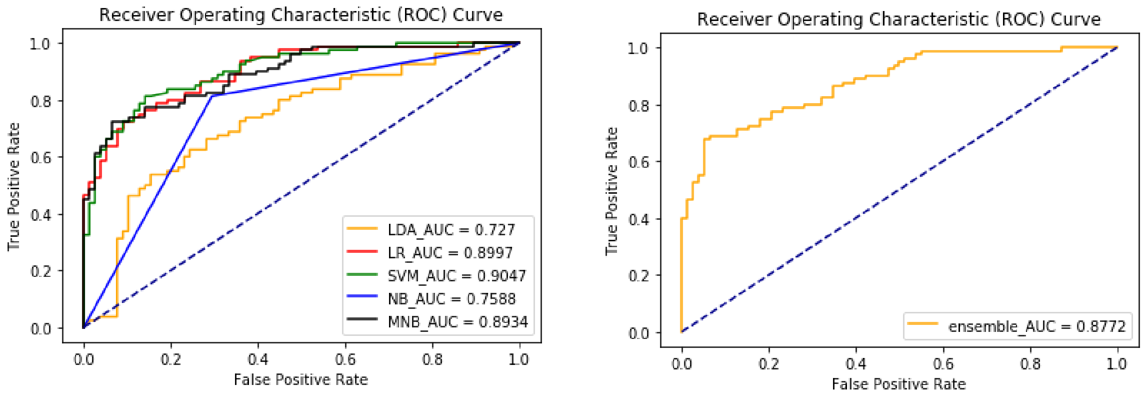


Fig.17 Roc curve of “Title” for different models

The ROC curve above is the classification prediction performance using the title of emails. We can easily find that the Logistic Regression, Support Vector Machine and Multinomial Naïve Bayes classifiers have better forecast performance than the other classifiers, they can achieve approximately 90% prediction accuracy. To make more accurate predictions, we used the voting ensemble modeling, which is a voting classifier to combine the predictions from multiple models to optimize the results. By using cross-validation, we got a classification rate of about 88%, considered as good accuracy.

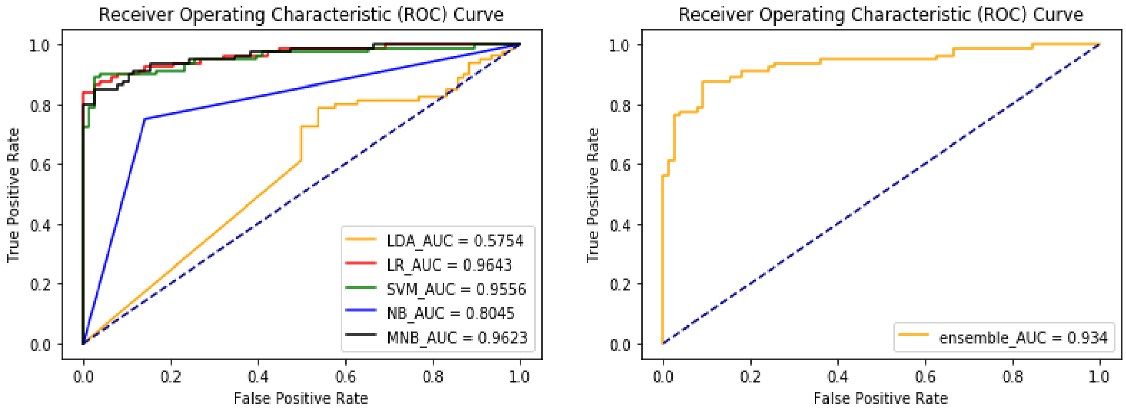


Fig.18 Roc curve of “Content” for different models

The above is the ROC curve for classification prediction with the content of emails. We can find among the five classifiers we selected, the three best performers are still Logistic Regression, Support Vector Machine and Multinomial Naïve Bayes. And their prediction accuracy has increased to around 96%. Then by assembling these five classifiers, we got a classification rate over 93%.

Since we also want to know what the classification result will be if we combine the title and content of emails together. We created a new column --- 'combine' in our dataset, which includes all the information (words) in the title and content of each email.

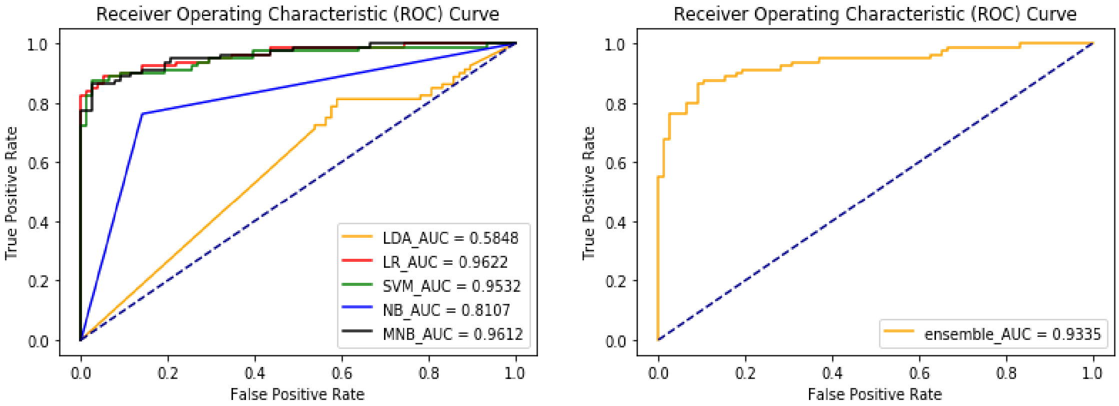
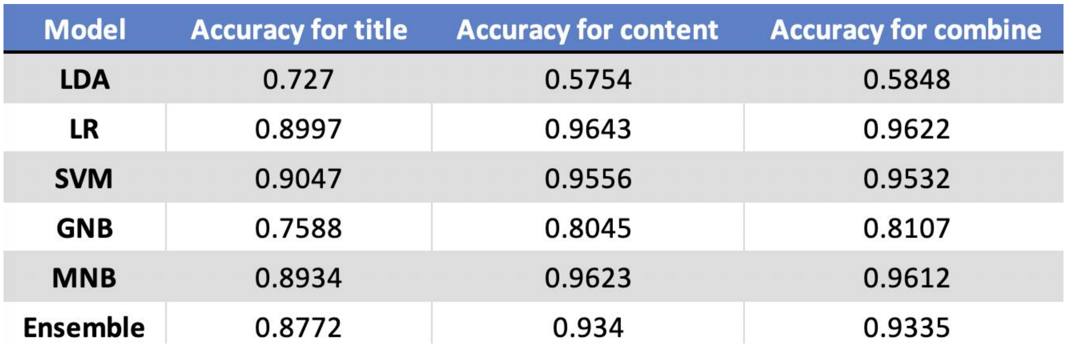


Fig.19 Roc curve of combination of "Content" and "Title" for different models

The ROC curves above are the classification prediction result for the combine of title and content of emails. It's not hard to find the result is pretty similar to the result of classification with content.

iii. Results comparison



Model	Accuracy for title	Accuracy for content	Accuracy for combine
LDA	0.727	0.5754	0.5848
LR	0.8997	0.9643	0.9622
SVM	0.9047	0.9556	0.9532
GNB	0.7588	0.8045	0.8107
MNB	0.8934	0.9623	0.9612
Ensemble	0.8772	0.934	0.9335

Fig.20 AUC of different models

From the classifier comparison table, we can find the best three performers are always Logistic Regression, Support Vector Machine and Multinomial Naïve Bayes. The Gaussian Naïve Bayes classifier also provides acceptable predictions, but Linear Discriminant Analysis classifier is not performing well in our project. Moreover, in the process of testing the model, we found that the running time of the Support Vector Machine is significantly larger than other classifiers. And because the shorter running Logistic Regression can provide similar or even better classification results, we think Logistic Regression is the best of our five selected classifiers.

Obviously, the results of classification prediction based on the content of emails are more accurate than the results of classification prediction by the title of emails. This is because the length of the email content is usually much larger than the length of the email title, there are more words in email content, meaning that email content contains more information. Therefore, by classifying the content of the emails, the prediction result will be more accurate.

Furthermore, the results of classification with combine are very similar to the results of classification with content. The reason is that the amount of information contained in content is usually much larger than the amount of information in the title. So, after combining title and content of emails, the title with a small amount of information is almost ignored, and the classification will still be based on the information in the content.

VI. Business Value and Future Work

i. Business Values to the Real World

Currently, based on what we have done in this stage, there are three main business values. First, the project can provide general knowledge about email's classification to users. In the clustering section, we introduce how to get the feature name of each cluster and determine the appropriate number of clusters for each user.

Second, our suggestion is that compared with “title”, people should pay more attention to the “content” and be careful of using the words frequently appeared in spam emails. Especially to companies, if they want to avoid their emails sorting into spam, they had better avoid these words, with high frequency in spam emails.

Third, depended on the classification model, we find out that when new emails come, users can use our model to classify these new emails into spam or ham relatively accurately. Another suggestion for companies is that they should refine email content to match user personal preference. For example, I am looking for an internship right now. I hope to get emails from LinkedIn about job recommendation. If LinkedIn always sends me emails about salary, or not relative job recommendation, I will add them into spam emails. After doing once, the probability of LinkedIn emails sorted into spam will increase a lot.

ii. Future Work

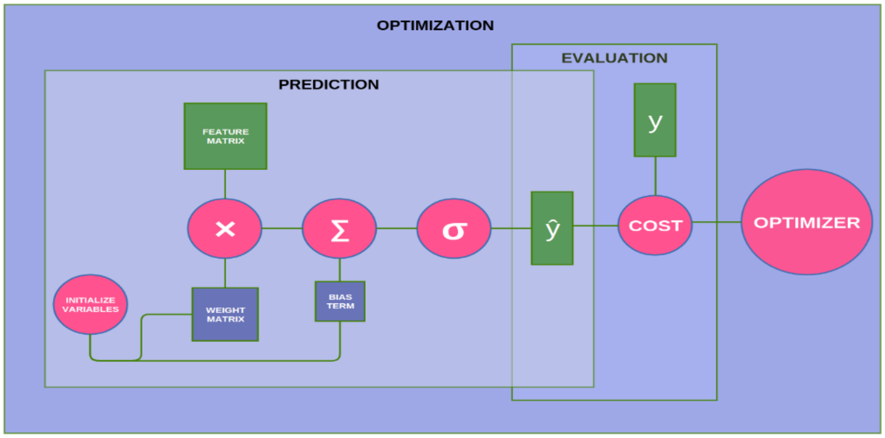
Until now, we still need more analysis form different aspects. A lot of future work needs to be done in the following.

First, more factors will be taken into consideration. Currently, we are focusing on single word and its tf-idf to fit the model. However, phrases and n-grams are also needed to be considered. Most times, phrases are more likely to stand for emails' sentiment. We will create new vectors and use classification models to get new scores. By comparing with the current scores, the number of grams will be evaluated whether there is significant impact on classification.

Second, optimizing classification models is the next thing we will do. Although we have perfect precision and curves in classification, we also need to set more parameters to get improvement.

Third, in the current content, we haven't added images in the content part. There is also probability to say that if too much unrelative images in an email, Gmail may sort it into spam. In next stage, after we scrape all information of emails, we will convert images into useful information and consider image's effect, to get more reasonable spam emails prediction. Not only for images, but also videos, or links may affect Gmail spam's classification.

Finally, the Flow of TensorFlow will be used next step. It's not enough for computer to remember high-frequency words appeared in spam emails just once. By using TensorFlow, after the first training phase, each email's predicted label will be presented. The actual label of emails will also be told to the program. The program will remember the difference and make new configuration to make better prediction next round. This program will be finished manually, or reasonable prediction presented. TensorFlow will improve the precision. General spam corpus can also be created by collection of all emails, regardless of personal difference on emails.



VII. Reference

Michael Capizzi. "*A TensorFlow Tutorial: Email Classification", 1 Feb 2016,* *http://jrmeyer.github.io/machinelearning/2016/02/01/TensorFlow-Tutorial.html* Accessed 10 May 2019

Bala Deshpande. *"Three challenges with Naive Bayes classifiers and how to overcome"*, 5 Jan 2016, http://www.simafore.com/blog/3-challenges-with-naive-bayes-classifiers-and-how-to-overcome, Accessed 10 May 2019

"The Logistic Regression Algorithm." 23 Apr 2018, https://machinelearning-blog.com/2018/04/23/logistic-regression-101/, Accessed 10 May 2019

"*Topic Modeling with Latent Dirichlet Allocation*" , https://pythonhosted.org/trustedanalytics/LdaNewPlugin_Summary.html, Accessed 10 May 2019

"SVM : Advantages Disadvantages and Applications", 9 Apr 2019, https://statinfer.com/204-6-8-svm-advantages-disadvantages-applications/, Accessed 10 May 2019