Filtering Spam Emails with Supervised Learning Algorithms

Using the Enron Datasets

Final Project, DS-GA 1001: Introduction to Data Science

Tong Jin (tj1061), Jiejie Wang (jw6190), Zixuan Zhou (zz2478)

December 5, 2020

GitHub

Structure

Introduction

Have you ever checked your mailbox to get a letter that you were expecting but ended up holding a thick roll of advertising mails that you did not want? Those unavoidable junk messages, such as new insurance plans, local trivias, grocery sales, are consistently ruining people's mailbox-checking mood. When it comes to the online world, similar situations happen. If you are a modern electric-mail user, you have probably dealt with spam emails, including unsolicited marketing messages, phishings and scams, or even malware-infected pieces. Just like junk mails, spam emails are not only a waste of limited internet resources but also a direct attack on people's privacy and web security. According to a recent report published by F-Secure, a cybersecurity company, spam-based cyber attacks have caused millions of dollars loss in 2020. Led by COVID-19 related scams, spam emails have become one of the most popular channels to spread malware and to commit crimes. Among industries that are heavily disturbed with spam attacks, finance is the most frequently scammed one in phishing emails (Pilkey, 2020). Given the fact that more professionals are accommodating themselves to work remotely and are more relying on emails for correspondence, it is urgent for email service providers, as well as employers, to implement anti-spam strategies in order to prevent further privacy invasion and capital losses.

Business Understanding

Thanks to the soar of data science and artificial intelligence, people developed various solutions that apply data mining approaches and machine learning techniques to proactively detect spam emails and block senders. For instance, Google launched its email service, Gmail, in 2004. Since then, the company has been developing algorithm-based solutions to control the distribution of spam emails. One of the most effective strategies is to use supervised machine

learning models to automatically detect spam contents in an email message. These models can also improve by fitting with new data records on a daily basis. Through self-training, supervised learning models can continuously increase spam filtration performance without requiring heavy labor or capital investments. Additionally, supervised learning powered spam filters are easy to maintain and expand. They are also adaptive to fast-paced business environments. Effective, and sustainable, smart spam filter is the ultimate solution that a company needs to defend its employee's email-browsing mood.

At HamNoSpam, we believe that every customer deserves a spam-free email browsing experience. As an industry-leading company, we fully utilized the power of supervised learning and have designed a spam filtration system that is specifically tailored to satisfy customer's needs. We believe it is a billion-dollar opportunity based on the revenue Gmail, Outlook, and other major email companies are making. Deploying our system can effectively pre-filter spam messages, giving a clean environment to employees and clients. With more and more clients selecting the email platform with our system, we will have more cases to further train our models and, therefore, achieve a win-win situation: clients are satisfied with a clean inbox and we double our revenues through making deals on advertisement with user data.

This project serves as an introduction to our system. To establish the filter, we first built a content-based model. By training the model with a vast collection of predefined spam and non-spam emails, we taught the model to understand the characteristics of typical spam emails, namely, the occurrence and distributions of words. The model then continuously gained prediction power. Eventually, it was smart enough to classify new emails as spam and non-spam at a high accuracy.

Although we are an emerging company, we always pair our model design and implementation with state-of-the-art industry best practices. Currently, knowledge engineering and machine learning are two standard methods used for spam filtration. In the machine

learning field, decision trees, support vector machines, and neural networks are common classification methods for solving the problem. Among neural networks, multilayer perceptron and radial basis functions are the most popular. According to Emmanuel G Dada et al.(2019), Gmail adapts logistic regression and deep neural networks in its classification. It also applies optical character recognition (OCR) to shield Gmail users from image spam (Amjad, 2019). Gmail claims that its filtering system can block 99% of all spam messages. Yang Song et al. used a naïve Bayes classifier, and proposed a new term weight aggregation function to lower the false-positive rate (Yang Song, 2009). Bhagyashri U. Gaikwad et al. adopted ensemble learning as the technique, and random forest as the classifier, to build the filtering system (Bhagyashri, 2014).

Data Understanding

The data we used were derived from the Enron-Spam dataset (Metsis et. al., 2006). We captured all six preprocessed, malware-free datasets. There are 785,648 instances, along with an indicator showing if one is spam or not. Each instance is an email message written by one of the six employees in Enron. Among instances, 378,508 (48.18%) are marked as spam and 407,140 (51.82%) are marked as ham (non-spam). The distribution of the number of words in each email is highly right skewed, with the average of 60 words (Appendix I, *Figure 1*). Most of the emails contain 60 to 75 words.

Data Preparation

To prepare for data mining, after aggregation, we completed the following data processing steps:

1. **Remove punctuation and stop words**. We removed frequently used stop words and punctuations from features because they are non-informative. This increases model

efficiency. We also manually added additional stop words based on word frequency. After feature engineering, we visualized top 15 common words in emails (Appendix I, *Figure 2* and *Figure 5*). The top 3 words are “com”, “company” and “please”. We also checked the top 15 frequent adjectives and adverbs (Appendix I, *Figure 3* and *Figure 4*). The top 3 adjectives are “new”, “financial” and “free”. The top 3 adverbs are “also”, “forward”, and “well”. This matches the characteristics of spam emails: web-based, wide-spread fake financial offers.

2. **Drop missing values**. After moving punctuation and stop words, we removed blank emails that originally consisted entirely of stop words. These cases took up 5% of our data. Since we assumed that all stop words wouldn’t contribute to the classification, we dropped them to save computational time.
3. **Random Sampling and Tokenization**. We tokenized the original dataset into unique word tokens. After tokenization, the number of feature columns boosted from 143,176. Given the enormous feature size, training models with the original tokenization would consume too much computational resources. As a result, we decided to randomly select 5% (39,279 records) of records to train models.
4. **Vectorization**. We applied the TF-IDF approach to vectorize the tokenized data. After vectorization, each word was assigned a unique integer index in the output vector. We created a vector with 32,796 unique indexes and used them as feature variables.
5. **Train-test Split**. To precisely evaluate model performance, we applied train-test split to create a hold-out set for evaluation. The hold-out set contains 20% of the records in the original dataset.

Modeling & Evaluation

There are various data mining algorithms we can apply for this type of problem. In lieu of the logistic regression, naïve Bayes, neural network, and random forest approaches we mentioned in the literature above, Amjad et. al. used a Scatter Search and K-Nearest Neighbors algorithm (Amjad, Gharehchopogh, 2019). Wang et. al. adopted the Support Vector Machine algorithm (Wang, Yu, Liu, 2005). Shi et. al. took the decision trees ensemble (Wang, Ma, Weng, Qiao, 2012).

The advantage of black-box algorithms, such as random forest and neural networks, is their ability to efficiently predict based on the learning of big size data. However, these models are highly complex and are hard to interpret and translate. Logistic regression and Naive Bayes, are more intuitive. Especially Naive Bayes, it has been proved to be powerful yet simple. With the potential of being used to train data based on a per-user characteristics, Naive Bayes can also lower false positive rate effectively due to the probability estimation ability. The limitation of logistic regression models, however, lie in the assumption of linearity between the target variable and features. On the other hand, Naive Bayes classifier doesn't do well in word phrases or image detection in spam filtering. As a result, when selecting models, we decided to build several models using the following algorithms:

- **Random Forest** as the baseline model
- **Logistic Regression with Elastic Net**
- **Multinomial Naive Bayes**
- **Gradient Boosting Machine**
- **Multilayer Perceptron Neural Network**

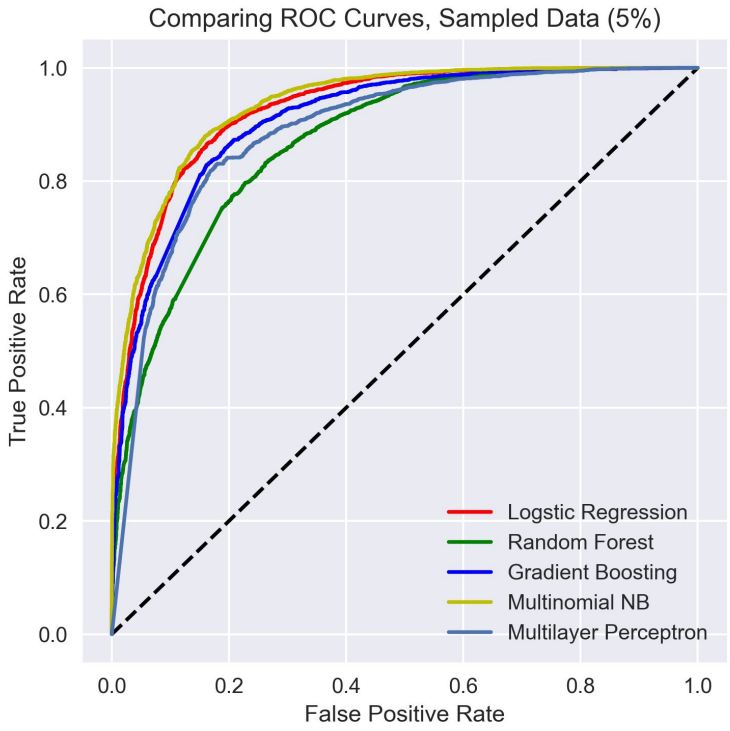
For each model, we applied grid search and randomized search to fine tune the hyperparameters. We also used 5-fold cross validation to increase model accuracy. The

evaluation metrics we used include: (1) Precision; (2) F-0.5 score; (3) Receiver Operating Characteristic (ROC); (4) Area Under the Curve (AUC). We measured the precision and the F-0.5 score because we want to lower the false-positive rate. As an email product, we do not want to mark non-spam emails as spam to cause our users to lose important information.

Since the spam ratio of the full data is 48:52, we expected that our baseline model should at least achieve an accuracy that is higher than 52%. Results favored our expectations. The model test AUC scores, precisions, F-0.5 scores, and accuracy scores are as follows:

Model	AUC Score	Precision	F-0.5 Score	Accuracy
Random Forest	0.8699	0.71	0.7367	0.7665
Logistic Regression with Elastic Net	0.9264	0.83	0.8388	0.8368
Multinomial Naive Bayes	**0.9354**	**0.87**	**0.8612**	**0.8533**
Gradient Boosting Machine	0.9083	0.79	0.8086	0.8309
Multilayer Perceptron	0.8879	0.83	0.8207	0.8196

The Multinomial Naive Bayes model is our optimal model because it has the best performance in all evaluation metrics we cared about. The graph below compares the ROC curve of each model. (All codes and documentations are updated on the Github repo as well)



Deployment

The Multinomial Naive Bayes model is the best algorithm with the highest AUC. However, a simple model can be easily baised. Therefore, we plan to use it, combined with Gradient Boosting Machine, as our main algorithms when building the spam filtering system for our clients. In order to improve its functions, there are three more things we could do in future deployment: first, explore more possible values of hyperparameters and use grid search to exhaust all combinations. Second, explore and incorporate other ensemble methods. Third, expand the system by including techniques that detect image scams.

We believe that we do not continuously need to train our model for its deployment. For saving resources, our model will simply be trained ad-hoc when it's in demand, and pushed to production until it deteriorates enough to need some fixing (Terence S, 2020). It is time to train the model again when we detect that the evaluation metrics seriously drop, or we receive a lot

of complaints. The model will be monitored on a daily basis. By providing users opportunities to tell us if we are categorizing spam emails right, we will know the “label” in real-time cases. We will calculate our evaluation metrics on all the data on one day, and evaluate if the fluctuation in metrics is normal.

In real-world practice, we should also monitor self-report scams from users and use these cases to evaluate our model and detect new patterns. Nowadays, Internet hackers can personalize scams to target a certain group of vulnerable people. Therefore, it’s important to keep updating our systems and capture new filtering opportunities for the purpose of social good instead of solely focusing on the general spam detection rate and overall accuracy.

In deployment of our email product, we should also inform clients to pay attention to three things: (1) Anti-virus protection. It helps eliminate suspicious emails before it even arrives at our filtering system. It also reduces risks on the client’s side. (2) Outbound mail scanning. It ensures that emails from our domain wouldn’t be blacklisted. (3) Greylisting. This technique blacklists certain senders who are involved with suspicious scams frequently.

There are three main ethical considerations in the deployment phase. And some of them are inducing controversial ideas.

(1) Influence on children. Kids, as a vulnerable email user group, are heavily targeted by inappropriate spams. It is necessary to increase our acceptance of false-positive rate while building the filtering system. We would rather do so to lower the false-negative rate so that kids would potentially be exposed to less scams. For adults who are not that vulnerable, we do not need to be this conservative.

(2) Privacy concern. John P. Buerck et al. (2006) mentioned that allowing spam filtering systems to read email content as an essential way to do classification would harm users’ privacy. We also want to add the potential risk that hackers can get private information just through the filtering system instead of hacking the whole email environment.

(3) Blacklist. Blacklist prevents users from receiving any emails from certain types of senders. Sometimes it is because the domain address your email organization is using is not commonly recognized, sometimes it's because too many people have intentionally marked your email as spam previously. Procedure one can take to file a complaint and get it corrected could be of too much trouble. These kinds of false positive cases can drastically cause time cost and inconvenience to email users in general.

Risks associated with our system include the lack of detection of image scam. Due to constraints on resources and time, we did not incorporate more models or techniques into our algorithms. All image scams will not be filtered out if we deploy this system in the producing market, which would expose our users to potential risks. With one click, users might lose money or important personal information. Given more time and resources, we can apply optical character recognition to detect image scam in our system to mitigate this kind of risk.

Appendix I: Exploratory Data Analysis

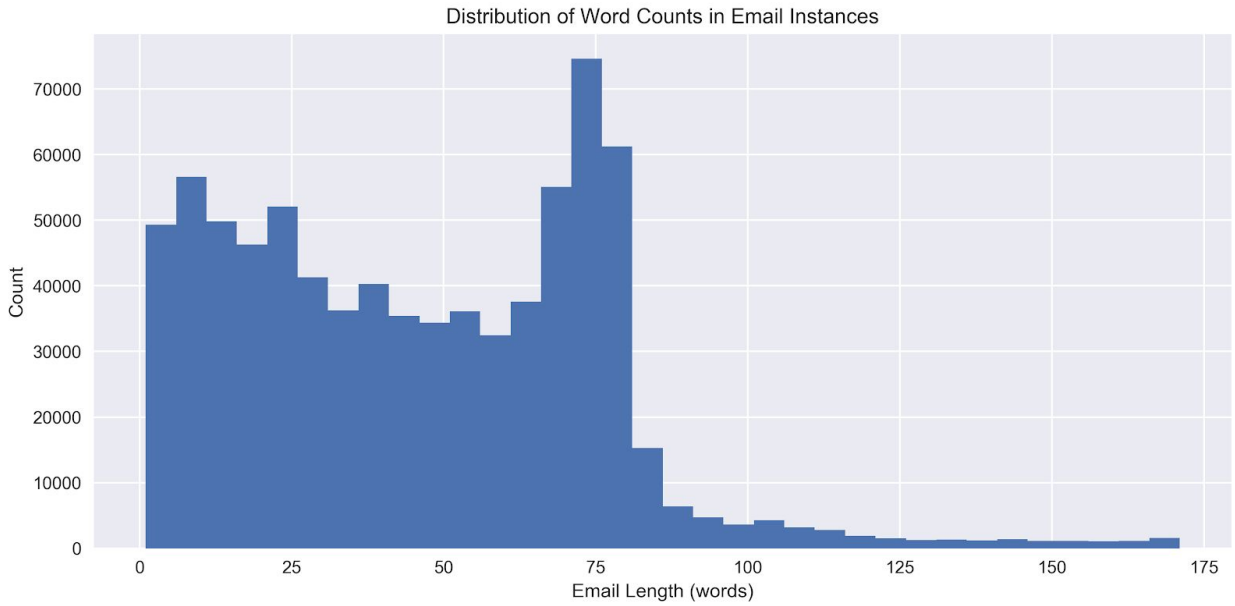


Figure 1. Distribution of Word Counts in Email Instances

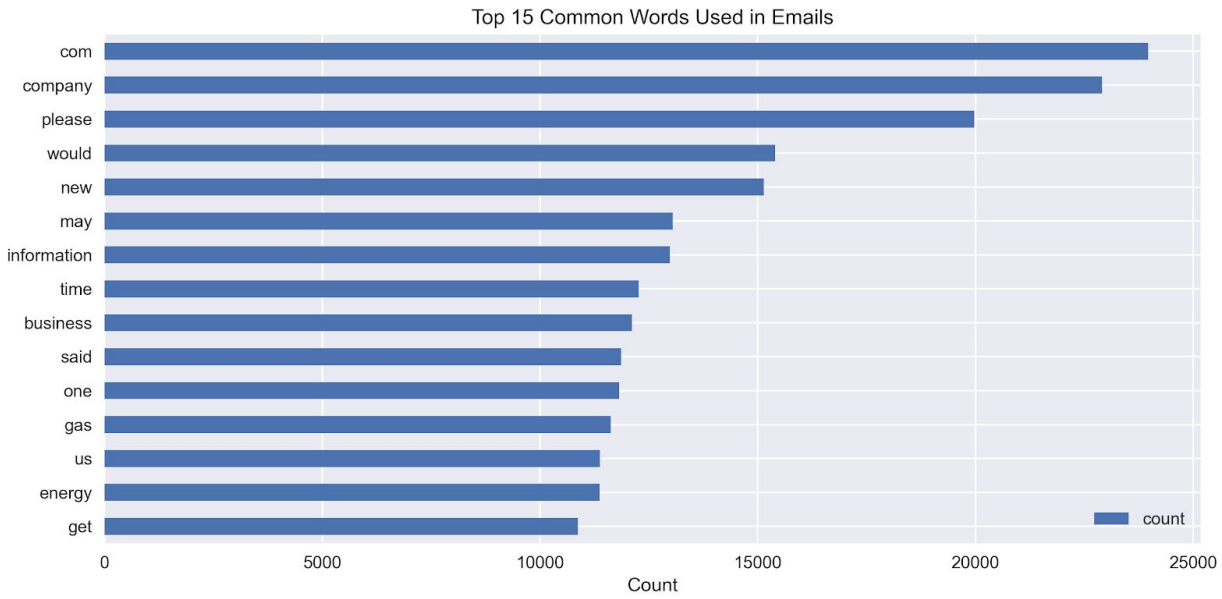


Figure 2. Top 15 Common Words Used in Emails

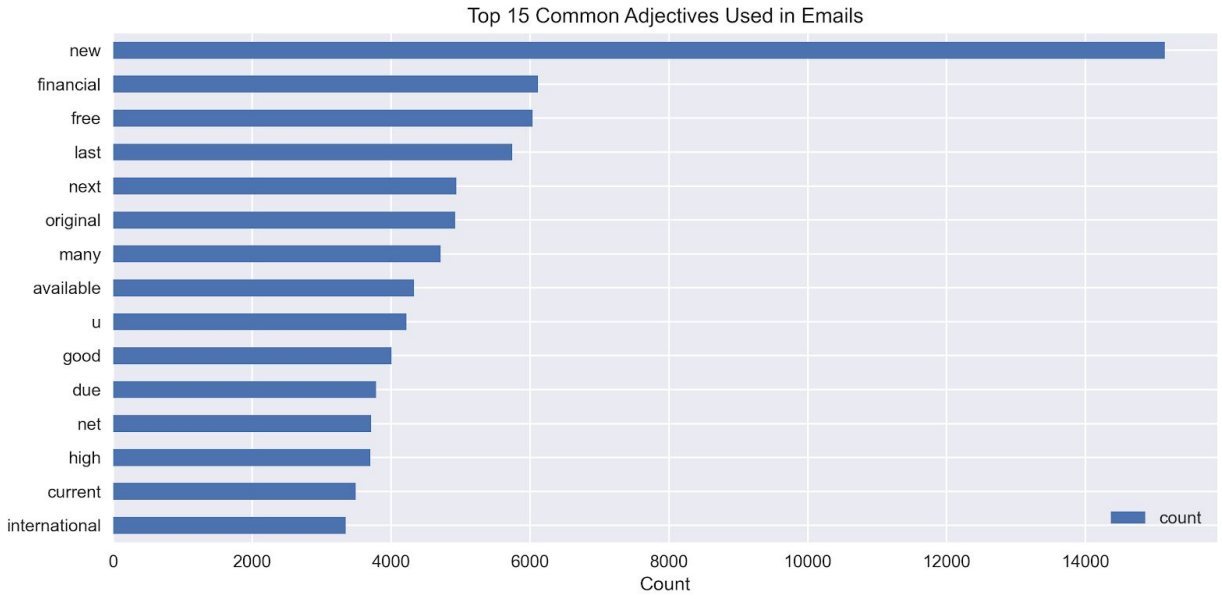


Figure 3. Top 15 Common Adjectives Used in Emails

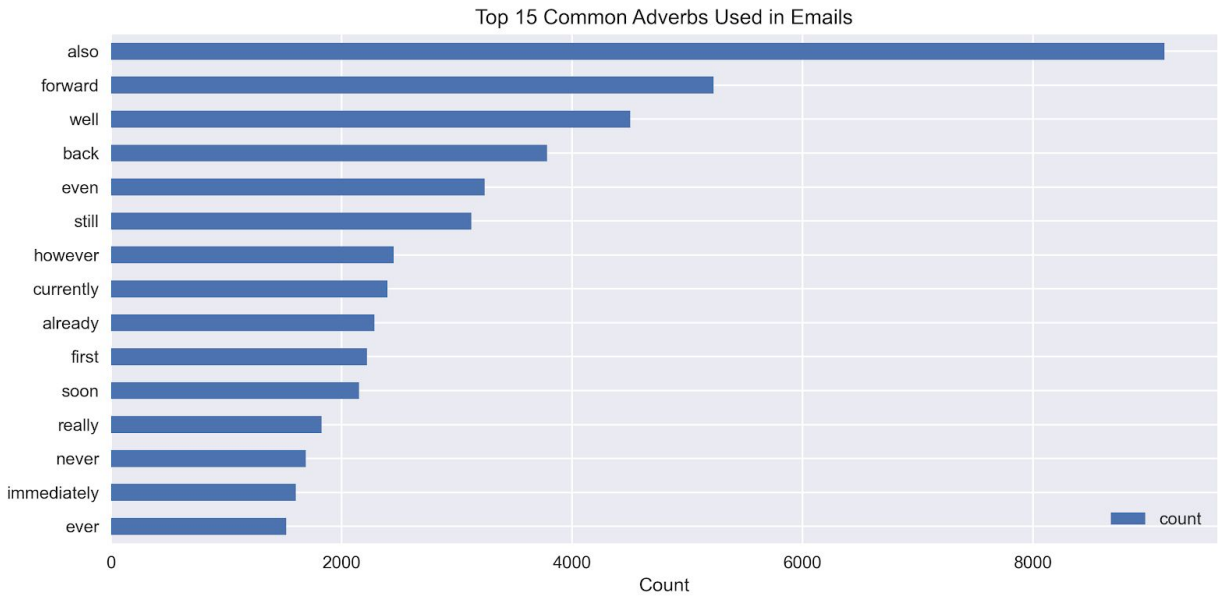


Figure 4. Top 15 Common Adverbs Used in Emails

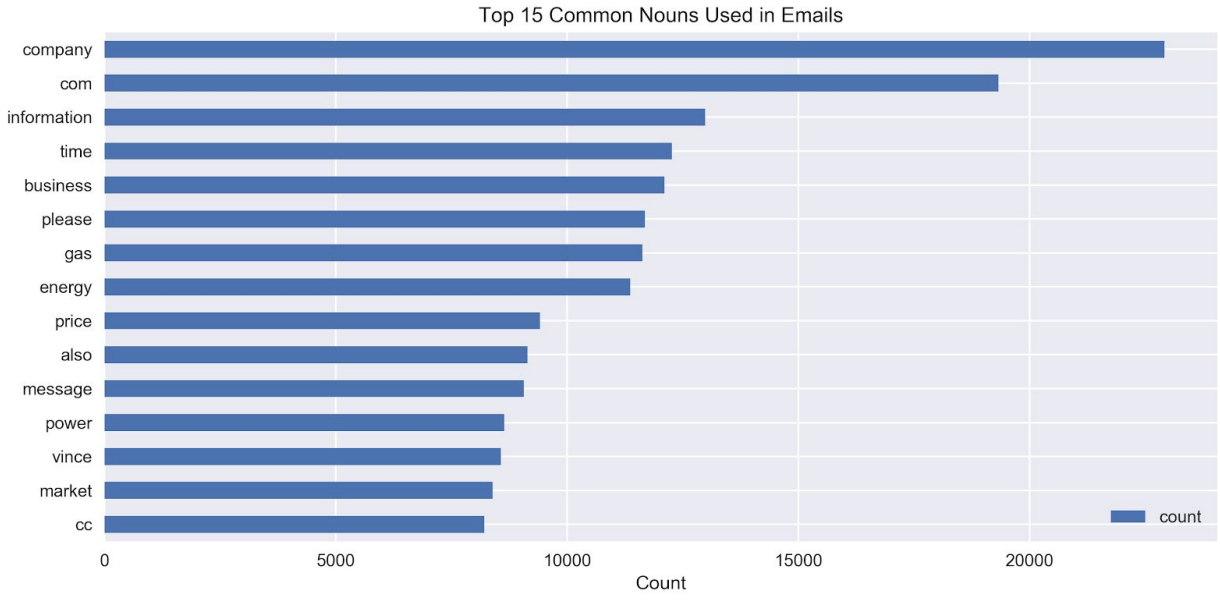


Figure 5. Top 15 Common Nouns Used in Emails

Figure 6. Visualizing Most Common Used Words in Emails

Appendix II: Model Results

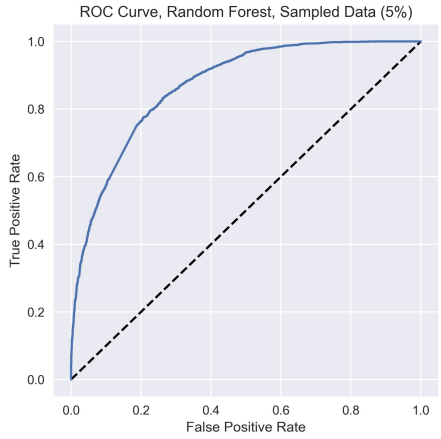


Figure 7. ROC Curve, Random Forest

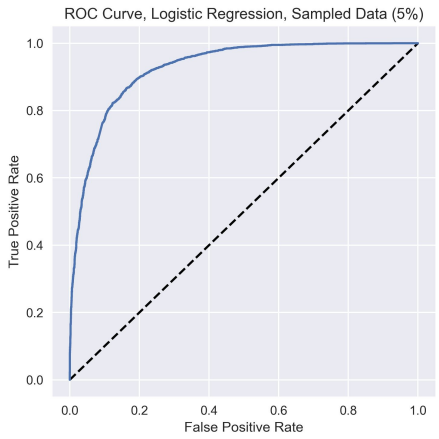


Figure 8. ROC Curve, Logistic Regression

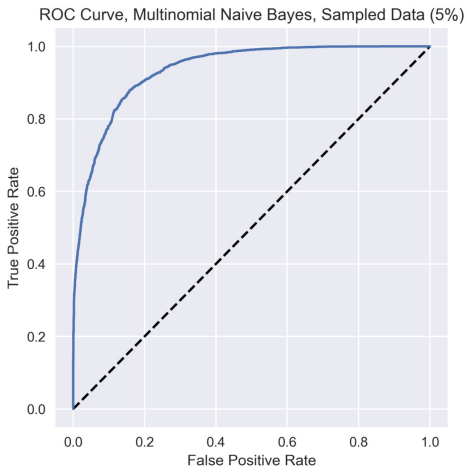


Figure 9. ROC Curve, Naive Bayes

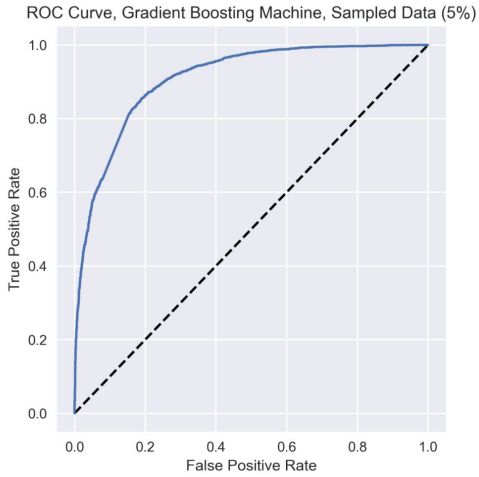


Figure 10. ROC Curve, Gradient Boosting Machine

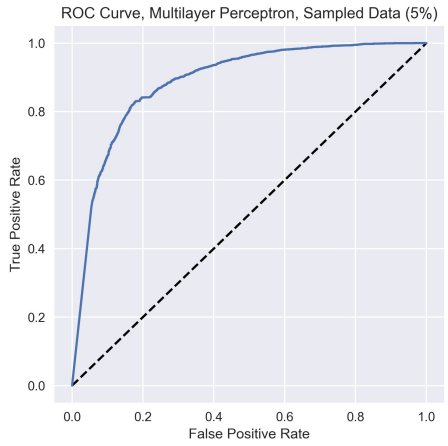


Figure 11. ROC Curve, Multilayer Perceptron

Appendix III: Team Contribution

Tong Jin: major contributor of all coding in Python + final report editing

Jiejie Wang: major contributor of final report draft + supplementary coding for data process and exploratory data analysis

Zixuan Zhou: supplementary coding for exploratory data analysis and models

References

1. Amjad, S., & Soleimanian Gharehchopogh, F. (2019, August 01). A Novel Hybrid Approach for Email Spam Detection based on Scatter Search Algorithm and K-Nearest Neighbors. Retrieved from http://journals.srbiau.ac.ir/article_14397.html
2. Attack landscape update: Facebook phishing, COVID-19 spam, and more - F-Secure Blog. (2020, September 17). Retrieved from https://blog.f-secure.com/attack-landscape-h1-2020/
3. Dada, E. G., Bassi, J. S., Chiroma, H., Abdulhamid, S. M., Adetunmbi, A. O., & Ajibuwa, O. E. (2019, June 10). Machine learning for email spam filtering: Review, approaches and open research problems. Retrieved from https://www.sciencedirect.com/science/article/pii/S2405844018353404
4. Ethical dimensions of spam. (n.d.). Retrieved from https://www.inderscienceonline.com/doi/abs/10.1504/IJEB.2011.043255
5. Gaikwad, B. U., Halkarnikar, P., & Student, M. T. (1970, January 01). Random Forest Technique for E-mail Classification: Semantic Scholar. Retrieved from https://www.semanticscholar.org/paper/Random-Forest-Technique-for-E-mail-Classification-Gaikwad-Halkarnikar/e0c37ec1359268e4431e49ee3729227489bd7ce4
6. Spam Email Classification using Decision Tree Ensemble Retrieved from http://jof-cis.com/article/spam-email-classification-using-decision-tree-ensemble/
7. Metsis, Vangelis, Androutsopoulos, Ion, Paliouras Georgios (2006). Spam Filtering with Naive Bayes -- Which Naive Bayes?. Retrieved from http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.61.5542
8. Wang, H., Yu, Y., & Liu, Z. (2005, December 06). SVM Classifier Incorporating Feature

Selection Using GA for Spam Detection. Retrieved from https://link.springer.com/chapter/10.1007/11596356_113

9. Yang Song Department of Computer Science and Engineering. (2009, August 01). Better Naive Bayes classification for high-precision spam detection. Retrieved from https://dl.acm.org/doi/10.5555/1568514.1568517
10. S, T. (2020, May 13). What Does it Mean to Deploy A Machine Learning Model? Retrieved December 05, 2020, from https://towardsdatascience.com/what-does-it-mean-to-deploy-a-machine-learning-model-dddb983ac416