CIS 545 Project: Twitter Airline Sentiment Analysis

Xi He[a], **Yuchen Ding**[a], **Yibo Yang**[b]

[a] ***Department of Data Science,***
[b] ***Department of Mechanical Engineering and Applied Mechanics,***
University of Pennsylvania, 3401 Walnut,
ycding@seas.upenn.edu, xhe93@seas.upenn.edu, ybyang@seas.upenn.edu

Abstract

Every day, people write their comments, reviews, and blogs on Twitter to express their feelings and opinions toward some news and events. Because of globalization, there are tens of thousands of air flights among various countries and cities. Understanding passengers' feelings and comments about their flights will help various airlines to provide better services for passengers. Machine Learning will help airlines to objectively and efficiently classify the sentiment of tweets. Therefore, in this project, we want to use some machine learning methods to classify the sentiment (negative/positive/neutral) of tweets.

Keywords: Machine Learning, Natural Language Processing, Classification, Sentiment Analysis, Supervised Learning

1. Introduction

1.1. Intend of the project

In this project, we would like to use the knowledge we have learnt in class to do natural language processing analysis on the airline data we found in the previous link. Such supervised learning approaches would be very useful to predict the user's opinion based on their review.

1.2. Ultimate objective

We would like to randomly split the data set into three parts: training, testing and validation. The ultimate goal for the proposed project is to accurately predict the users' opinion of the airline they take based on their comments in the review panel. We would like to use the prediction accuracy as the quantity to justify our model's performance.

1.3. Models we use

We plan to implement multiclass classification models for the prediction of sentiments: positive, negative, and neutral. We are considering Decision Tree, SVC [1], AdaBoost [2] and Random Forest methods [3].

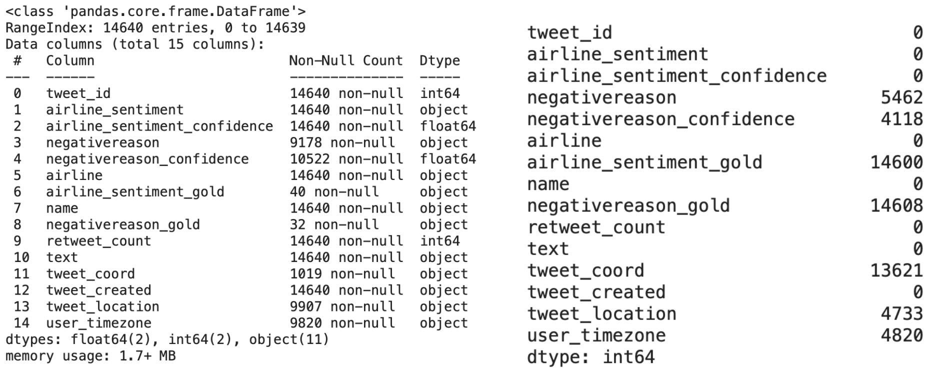
2. Data parsing and data preparation

In this section, we would describe: (1) resource and summary of the data set, (2) some representative samples of the data, (3) basic analysis to show some distribution to reflect the property of the data set.

2.1. Description of the data set

The original data could be found on the website:
https://www.kaggle.com/crowdflower/twitter-airline-sentiment
The summary of data information together with missing information could be found in figure 1. We would see there are 15 columns and 14640 rows. The data set includes the following columns: tweet_id, airline_sentiment, airline_sentiment_confidence negative reason, negativereason_confidence, airline, airline_sentiment_gold, name, negative reason_gold, retweet_count, text, tweet_coord, tweet_created, tweet_location, user_timezone. We are more interested in using the text information to run classification models to predict the customers' opinions on their flights.

Among the massive data we found, we would like to show some examples on the text information we would like to focus on as following:
Positive comments: "awesome. I flew yall Sat morning. Any way we can correct my bill?"
Neutral comments: "first time flying you all. do you have a different rate/policy for media Bags? Thanks"
Negative comments: "Hey, first time flyer next week - excited! But I'm having a hard time getting my flig ..."



```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 14640 entries, 0 to 14639
Data columns (total 15 columns):
 #   Column                        Non-Null Count  Dtype
---  ------                        --------------  -----
 0   tweet_id                      14640 non-null  int64
 1   airline_sentiment             14640 non-null  object
 2   airline_sentiment_confidence  14640 non-null  float64
 3   negativereason                9178 non-null   object
 4   negativereason_confidence     10522 non-null  float64
 5   airline                       14640 non-null  object
 6   airline_sentiment_gold        40 non-null     object
 7   name                          14640 non-null  object
 8   negativereason_gold           32 non-null     object
 9   retweet_count                 14640 non-null  int64
 10  text                          14640 non-null  object
 11  tweet_coord                   1019 non-null   object
 12  tweet_created                 14640 non-null  object
 13  tweet_location                9907 non-null   object
 14  user_timezone                 9820 non-null   object
dtypes: float64(2), int64(2), object(11)
memory usage: 1.7+ MB
```

```
tweet_id                            0
airline_sentiment                   0
airline_sentiment_confidence        0
negativereason                   5462
negativereason_confidence        4118
airline                             0
airline_sentiment_gold          14600
name                                0
negativereason_gold             14608
retweet_count                       0
text                                0
tweet_coord                     13621
tweet_created                       0
tweet_location                   4733
user_timezone                    4820
dtype: int64
```

Figure 1: *Data information:* (a) Summary of information on the data set. (b) Summary of missing data.

2.2. Basic data analysis on the distribution of label

We split the data processing into three parts. We first analyze the class distribution over the three classes and find out the data we are using is unbalanced. Such phenomenon would cast some difficulty in the training process. Several ways could possibly use to mitigate this issue is to split the data into small subsets and use boosting algorithms or implement a weighted loss function during the training process. For more details of the distribution of labels, we include the distribution of each airlines in figure 2. With the unbalanced data in place where the negative feedback seem dominate the reviews, we would like to use more information to understand the reason of negative feedback in figure 3.

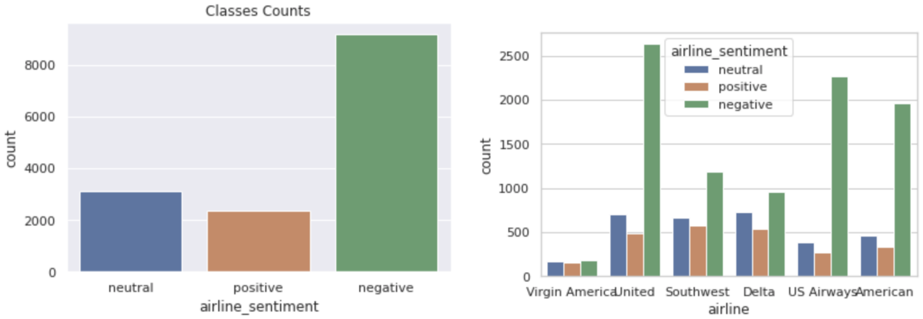


Figure 2: *Label distribution:* (a) Unbalanced label distribution of the data set where 0, 1, 2 are corresponding to the negative, neutral and positive feedback. (b) Label distribution across different airline company.

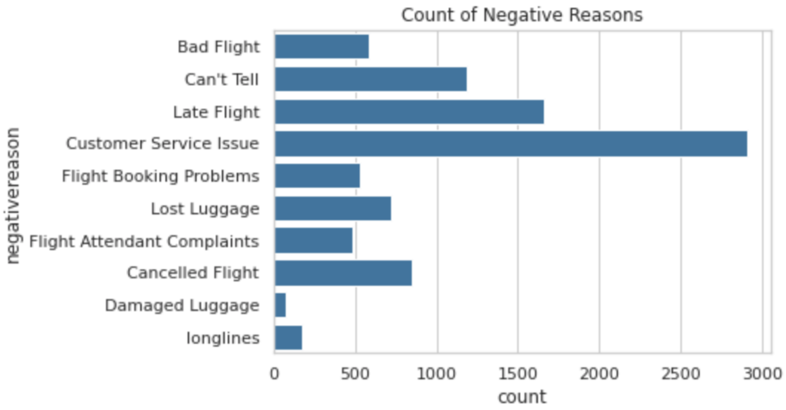


Figure 3: *Count of negative reason:* the counts of negative reasons.

2.3. Data cleaning

Before we move forward to word cloud analysis and machine learning models, we would like to spend a few time to further parse the data to make it clean enough for the later tasks. The work flow is briefly summarized as following:
(1) Remove mentions.
(2) Splitting.
(3) Convert the words to lower cases.
(4) Remove the stopping words in sentences.
(5) Remove all punctuation.
After such processes, we would be able to move forward to move extensive analysis and implement the machine learning models on the data.

2.4. Word cloud analysis

We would like to further take a look at the frequency of the words in the sentiment to see the most representative words that appears in each class. The word cloud that includes the summary of different class is shown in figure 4 from which we could easily summarize some words that seem to be critical in expressing the customers' mood. For instance:
In the word cloud of positive feedback, we would find words that contains positive information such as: thank, help, know, etc.
In the word cloud of negative feedback, we would find words that contains negative information such as: delay, cancelled flight, know, late flight, flight cancelled, etc.

While from the word cloud of neutral feedback, we would find more mild words such as: please, change, check, make, see, etc.
We can tell that from the word cloud for the negative reviews, the representative words seem to be very similar with what we observed in the negative reasons figure shown in figure 3.
We want to further propose the way that we would like to analyze this data set. We would like to split the data set into two groups: (1) negative reviews (2) positive and neutral reviews.

There are at least three reasons for doing this: (1) the airline company who is interested in this analysis would put more attention on the negative reviews and would like to treat the others as one group, (2) the word cloud tells us the positive review and neutral review do not have significant difference, (3) merging the positive and neutral together would further help us balance the data to make sure each group has roughly same number of data.

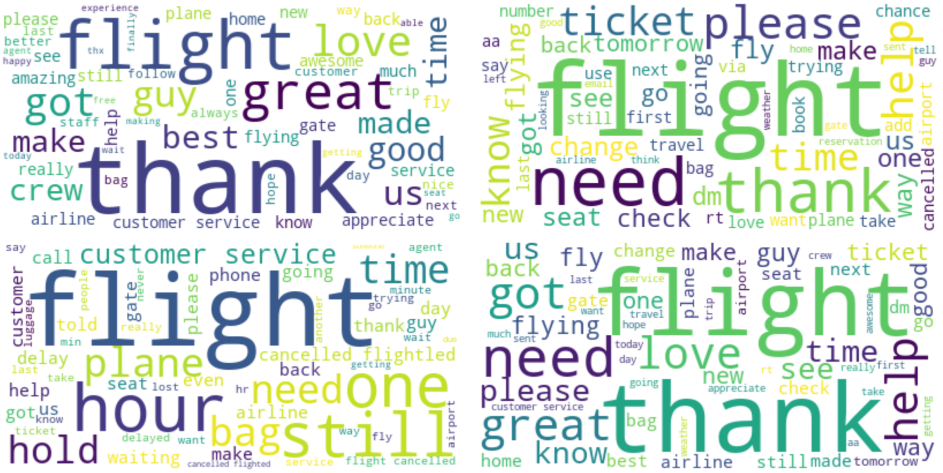


Figure 4: *Word cloud of different label:* (a) Upper left: word cloud for positive feedback. (b) Upper right: word cloud for neutral feedback. (c) Lower left: word cloud for negative feedback. (d) Lower right: word cloud for the combination of positive and neutral feedback.

3. Machine learning methods

In this section, we brief review the machine learning methods we would like to use in this project. We will start from the method that transform the data into vectors, then, briefly discuss the machine learning models for classifications.

3.1. Data transformation and embedding

We consider to embed the data into a long vector so that we can run machine learning methods on that. Specially, vectorize the sentence according to the words frequency (count) is a simple way to do so. One benefit would be: all sentences are having the same length and could be a good suit for the input format of machine learning models we will later use.

3.2. Logistic regression

Logistic regression is widely used in binary classification problem. It is a special class of generalized linear model [4]. This method is simple and could run efficiently on large data set and has the property to prone against overfitting. We consider use this method as a baseline for this project.

3.3. Decision tree

Decision tree [5] is one of the most popular tools we have learnt in our class. It is well know for the information gain on the training process and provide accurate prediction. While the depth of the decision tree needs fine tuning and there is a high likelihood that such method is tend to overfit and introduce high bias.

3.4. Random forest

Random forest [6] is a bagging method that combines a set of decision tree to make a final prediction using vote. Such method is a more advanced decision tree model and could help a lot mitigating the bias introduced by a singe decision tree. We would expect this model to perform better than decision tree.

3.5. Support vector machine

Support vector machine [7] is a popular machine learning method well known for its application if classification problems. Several variants of support vector machine are proposed to address the non-perfect linear separatable data [8], data transformation using kernel methods [9], etc.

3.6. Adaboost

Adaboost [2] is a very popular boosting method that achieves state-of-the-art robustness in the classification problems and is used for facial detection problems, feature selections, etc. Adaboost is sensitive to noisy data and outliers, we would expect it to help with determing the important words appear in each reviews from our word bag. Ideally, the accuracy of Adaboost should be higher then the previous mentioned methods.

3.7. Summary of performance for each machine learning methods

We summarize the performance of each method in table 1 from which we can see the logistic regression is performing unexpectedly well. We think this is because of its property of preventing over-fitting and also the reality that the embedding we are using is not leveraging the global information. A good observation is random forest model compared with decision tree model is safe-guard from over-fitting while also provide improvement in terms of the accuracy. Comparing the logistic regression with SVM, we can see although SVM is giving good testing accuracy, the model is prone to over-fitting by providing very large training-testing accuracy discrepancy. This also justifies that logistic regression is a simple but powerful model. Throughout all the methods we are using, we can see the Adaboost is exhibiting the least training-testing discrepancy and is better than random forest. One can say, Adaboost is potentially a better tool here as it at least requires less hyper-parameter tuning. To further illustrate the comparison of different methods in terms of training and testing accuracy together with their discrepancy, we show the accuracy in figure 5 left panel. More interestingly, random forest enables us to plot the importance of each feature used in the training. We are also reporting these features in figure 5 right panel. By looking at this figure, we can conclude that the marked out features are really expressing the emotion of the feedback such as ”afraid”, ”embarrassment”, ”occupied”, ”preventing”, etc.

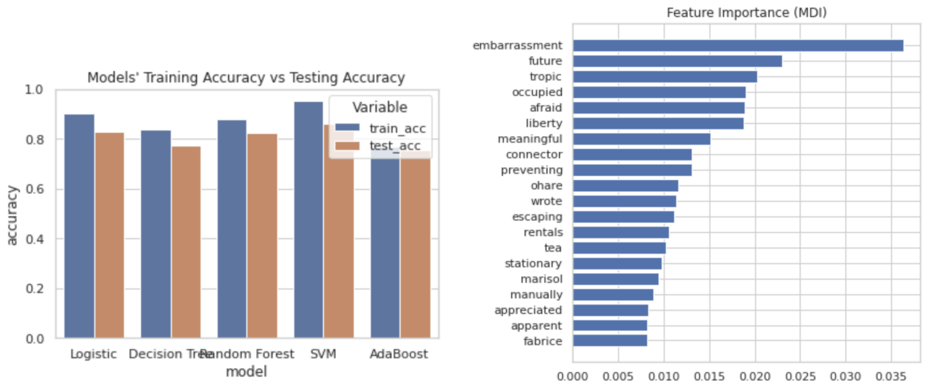


Figure 5: *Accuracy of different machine learning methods:* (a) Accuracy of different machine learning methods in terms of training and testing accuracy together with their discrepancy. (b) The important features extracted from random forest method.

Accuracy Methods	Training accuracy	Testing accuracy
Logistic regression	90.2%	82.8%
Decision tree	83.9%	77.4%
Random forest	88.2%	82.5%
Support vector machine	95.3%	86.0%
Adaboost	76.8%	75.8%

Table 1: *Accuracy summary of the machine learning methods:* we are reporting the training and testing accuracy of the machine learning methods we use: Logistic regression, Decision tree, Random forest, support vector machine and Adaboost.

4. What challenges and obstacles might you anticipate with this project?

The challenges could arise from several directions. First, the words in each review could include typos and have grammatical errors which cast great difficulty in analysis using classical semantic analytic methods. Specially, mature word embedding techniques have already established good word to vector dictionaries. However, this has nothing to do with words that have never appeared before.

5. Challenges

In this section, we would like to briefly discuss about the potential challenges for this research direction in the future.

5.1. Dealing with unbalanced data

We can see that the number of data in the summary of figure 2 is not balanced. Specially, the number of negative feedback is more 9,000 while the number of feedback for positive and neutral are only 2,000 and 3,000. As discussed in section 2.4, we showed our way of handling the unbalance in the data by merging the neutral class into positive class and do a binary classification on that. However, this still seems not sufficient enough because the total number of data for positive and neutral are less than 6,000. In this case, we need to consider more tricks for balancing the data. There are at least four ways we think could approach this problem and we have chosen to simply merge the groups to achieve the balanced data set:
(1) Data augmentation via bootstrap: we can upsample the data in the combination of positive and neutral data using bootstrap [10] to increase the number of data so that the upsampled data could match the negative reviews.
(2) Down sampling: we can down sample [11] the negative review data into subsets to match the other class. As the combination of positive and neutral are around 5,000 data, we do not loss too much and could still capture the original data distribution information.
(3) Use boosting methods. We can split each class into small group of data and use boosting [2] to adaptively tune the weights of each group. Such approach would take advantage all the data information for our inference task.
(4) Use weighted loss. We can assign the weights to the loss function [12] of each class according to the number of data contains in each. In this case, the gradient information could be balanced by the number of data to further achieve robust training behavior.

5.2. Embedding the reviews

As is well known, the embedding of words is playing a critical role in the sentiment analysis. A good embedding would enable the researchers to capture the local and global information of language. There are several ways to tackle this task and we have chosen the first one:

(1) If we discard the local information of a sentence, we can use the frequency information [13] of the words that appear in each sentence to embed a whole review.
(2) In language modeling, there exists a popular embedding dictionary word2vec [14] that embeds each word into a vector. Such approach would capture more local information between words.
(3) Glove embedding is a new state-of-the-art methods in word embedding [15] that leverage both global and local information of languages for word embedding that achieved the best performance in language modeling.

6. Future directions

We think there are at least two future directions we can consider in addition to what we have done in this report:
(1) Besides the proposed methods we will use, we can also try some more advanced methods for the classification such as: 1D convolutional neural network [16], XGboost [17], etc.
(2) Besides the accuracy of the prediction we have mentioned, we can further do extensive analysis using the full information of the confusion matrix. Information such as F1 score to study the classification performance on the unbalanced data.

Acknowledgements

We would like to thank professor Ives and TA Hoyt Gong of this class for stimulating discussions.

References

[1] J. A. Suykens, J. Vandewalle, Least squares support vector machine classifiers, Neural processing letters 9 (1999) 293–300.

[2] Y. Freund, R. E. Schapire, et al., Experiments with a new boosting algorithm, in: icml, volume 96, Citeseer, pp. 148–156.

[3] J. Friedman, T. Hastie, R. Tibshirani, The elements of statistical learning, volume 1, Springer series in statistics New York, 2001.

[4] P. McCullagh, Generalized linear models, Routledge, 2018.

[5] S. R. Safavian, D. Landgrebe, A survey of decision tree classifier methodology, IEEE transactions on systems, man, and cybernetics 21 (1991) 660–674.

[6] V. Svetnik, A. Liaw, C. Tong, J. C. Culberson, R. P. Sheridan, B. P. Feuston, Random forest: a classification and regression tool for compound classification and qsar modeling, Journal of chemical information and computer sciences 43 (2003) 1947–1958.

[7] W. S. Noble, What is a support vector machine?, Nature biotechnology 24 (2006) 1565–1567.

[8] D.-R. Chen, Q. Wu, Y. Ying, D.-X. Zhou, Support vector machine soft margin classifiers: error analysis, Journal of Machine Learning Research 5 (2004) 1143–1175.

[9] C. Leslie, E. Eskin, W. S. Noble, The spectrum kernel: A string kernel for svm protein classification, in: Biocomputing 2002, World Scientific, 2001, pp. 564–575.

[10] B. Efron, R. J. Tibshirani, An introduction to the bootstrap, CRC press, 1994.

[11] D. N. Politis, J. P. Romano, M. Wolf, Subsampling, Springer Science & Business Media, 1999.

[12] A. Sellami, H. Hwang, A robust deep convolutional neural network with batch-weighted loss for heartbeat classification, Expert Systems with Applications 122 (2019) 75–84.

[13] C. D. Manning, H. Schütze, P. Raghavan, Introduction to information retrieval, Cambridge university press, 2008.

[14] I. Yamada, A. Asai, J. Sakuma, H. Shindo, H. Takeda, Y. Takefuji, Y. Matsumoto, Wikipedia2vec: An efficient toolkit for learning and visualizing the embeddings of words and entities from wikipedia, in: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, pp. 23–30.

[15] J. Pennington, R. Socher, C. D. Manning, Glove: Global vectors for word representation, in: Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP), pp. 1532–1543.

[16] N. Kalchbrenner, E. Grefenstette, P. Blunsom, A convolutional neural network for modelling sentences, arXiv preprint arXiv:1404.2188 (2014).

[17] T. Chen, C. Guestrin, Xgboost: A scalable tree boosting system, in: Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining, pp. 785–794.