Twitter Sentiment Analysis

Abstract

Around two hundred and fifty million passengers travel using an aircraft in the United States on an average per year. This makes it a significantly important mode of travel and understanding the sentiments behind what people think of their journeys helps unfold the technicalities behind how airlines operate as a unit. We performed the techniques of encoding and decoding on our dataset to accurately fit the data into the Support Vector Machine (SVM) model and a Sequential Neural Network model. After evaluating the model performances, the SVM model arrived at a testing accuracy of 78% and the Sequential Neural Network model arrived at an accuracy of 96% using the Count Vectorizer and an accuracy of 86% using the TF-IDF Vectorizer. The results show that our models successfully trained on the given dataset and accurately determined the sentiments of the users.

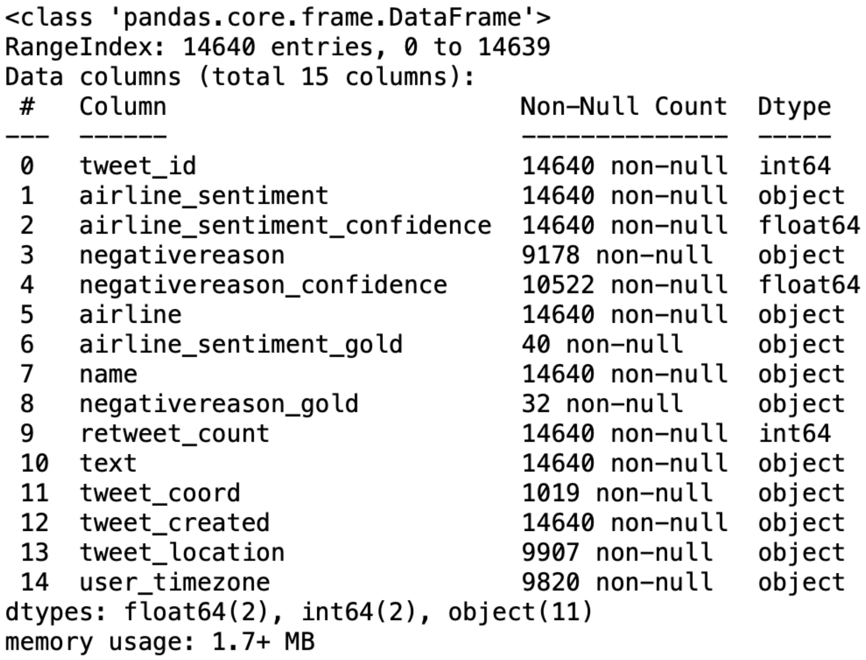
1. Introduction

Identifying the general sentiment of a given document is the goal of sentiment analysis, also known as opinion mining, which is a fundamental task of machine learning. We can extract the subjective information from a text and attempt to categorize it according to its polarity, such as positive, neutral, or negative, using machine learning techniques and natural language processing. We will employ airline data in our project, which contains all of the tweets and comments made by various individuals about certain airlines carriers . This analysis is particularly helpful since it allows us to infer the general viewpoint of airline passengers by reading through each user's remarks. Since the language is so complex (objectivity/subjectivity, negation, lexicon, grammar, etc.), sentiment analysis is actually still very much a work in progress, but that is also one of the reasons it is so fascinating to work on. By developing a model based on probabilities, we have decided to attempt to categorize messages from Twitter into "positive," "negative," and "neutral" emotion. Twitter is a microblogging website where users can post 140-character tweets to rapidly and spontaneously express their feelings. By include the target sign "@" or the hashtag "#" in your tweet, you can join in a topic or address someone directly in a tweet. Twitter is a wonderful source of information to ascertain the current general opinion about anything due to its popularity.

2. Method

2.1 Dataset and Data Preparation

The dataset was prepared by extracting data as of February 2015 from Twitter, the data was then classified into one of the following categories; Positive, Neutral, and Negative.



```
<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
```

Figure 1: Feature characteristics of the dataset

Figure 1 shows all the columns and their characteristics which exist in the dataset we have used for the sentiment analysis. Out of all the features, our study involves the following features more extensively; airline_sentiment, negativereason, airline, and text. Therefore, to process the dataset, we got rid of all the remaining columns which did not add particular value into our study. Below is a short description for each of these features:

- airline_sentiment: Categorized into either a positive, neutral, or negative review.
- negativereason: What is the reason behind a user writing a negative review.
- airline: The name of the airline.
- text: The twitter review that a passenger has written.

For better understanding the distribution of our dataset, we produced a few visualizations to help us with that. First we plotted a bar graph to look into the distribution of the review sentiments which would give an idea about the most dominant review, which turned out to be the negative review. We also looked into the number of flights associated with each tweet to determine the most popular flight choice amongst passengers and our results show that United Airlines is the most used airline, and Virgin America is the least popular airline.

We then performed text cleaning on the dataset to ensure our model trains on the most suitable data format for better accuracies. We used the following techniques to clean our dataset:

- Punctuation Removal: All the punctuations used in the text were removed using the string library. Also, any occurrences of any links were also eliminated from the texts.
- Lower Casing: All the words were converted into a lower case because case sensitivity would not make a difference for our study. Also by doing so, we reduce the size of the vocabulary.
- Lemmatization: Grouping together different inflected forms of a word so that it can be analyzed as a single root word. (i.e. eat and eating are both reduced to eat).

After cleaning and pre-processing the dataset, we then split the dataset to contain 10,000 reviews in the training set and the remaining 4640 reviews in the testing set. We then made more changes to the dataset in preparation for model fitting that will be discussed in the subsequent sections.

2.2 Word Embedding Techniques

Two frequency based word embedding techniques were implemented in this study, the CountVectorizer and the Term Frequency-Inverse Document Frequency (TF-IDF) Vectorizer. These techniques are essential to convert raw text with variable lengths into a numerical format, so that it can be used to train on a model. Let us look into both of these techniques more closely.

2.2.1 Count Vectorizer

The count vectorizer converts text into a matrix that consists of the word counts. Think of it as, the terms included in the documents are the columns and the documents are the rows. This technique is also referred to as the Bag-of-Words (BoW) technique. First we initialized CountVectorizer on the default parameters and trained that on the training dataset where it learned the vocabulary and transformed into a 10000*11477 sparse matrix for the training set and a 4640*11477 sparse matrix for the testing set.

2.2.2 TF-IDF Vectorizer

The acronym TF-IDF stands for term frequency - inverse document frequency, where TF simply refers to the frequency of the term in the document term matrix and IDF is equal to log [(1 + D) / (1 + df(d, t))] + 1, where D denotes the total number of documents and df(d,t) denotes the total number of documents in which the term t has appeared in the DTM. The TF-IDF Algorithm is widely used in document search and information retrieval systems. We initialized TfidfVectorizer by setting the following parameters; use_idf equals to True and lowercase equals to True. Then 66 we trained it on all the reviews of the entire dataset.

2.3 Encoding the Dataset

The dataset outcome columns, which are the tweet sentiments, exist in a string literal format, and comprises 3 categories, positive, neutral, and negative. These strings need to be converted into a numeral representation for more accurate model fitting. We made use of two encoders for this study, the LabelEncoder and the One Hot Encoder. Both of these are further explained in the following section.

2.3.1 LabelEncoder

The Label Encoder works by assigning a number to each of the different categories present in the column, if there are 3 categories like our dataset, the Label Encoder will assign 3 distinct numeral values to each category. The following is the shape of the outcomes after applying the fit_transform method to the outcome column:

- Training outcome (y_train): (11712,)
- Testing outcome (y_test): (2928,)

2.3.2 One Hot Encoder

The One Hot Encoder works by creating new features equal to the amount of distinct categories in the outcome column of the dataset. In this encoding technique, each category will be represented as a one hot vector, also known as a binary vector. In our scenario, we had 3 distinct sentiments therefore 3 vectors were added as features to represent the value of the outcome sentiments. The following is the shape of the outcomes after applying the fit_transform method to the outcome column:

- Training outcome (y_train): (11712, 3)
- Testing outcome (y_test): (2928, 3)

Before we proceeded ahead, we decided to not use the results obtained from the LabelEncoder. This was because the outcomes did not reflect to be multiclass friendly. Also, a LabelEncoder assigns numbers in such a way that it ends up treating the labels as a pattern, thus creating a slight bias which we want to avoid. Hence, only the results from One Hot Encoding were used in the study.

3. Model Implementation

This study was backed by the results obtained from 9 models. The reason behind using so many models was solely to compare the accuracies received from each one of them, this would allow us to see how consistent and well performing the data sentiment analysis is. We will now discuss in detail about each of the models used.

Sequential Neural Network

A sequential Neural Network model was first used, a reason behind this was that our data was sequential, meaning a sentence develops meaning as it is formed. Then each token from the words in the sentence is ingested, which makes it important to pick such a model for better fitting.

The model was built by adding 12 layers, 2 of them being assigned as the input and output layers, 5 dense layers, and 5 dropout layers. The activation function for the layers were initialized to relu and tanh. A kernel regularization of l2 was used across the layers. Overall, the model loss function was set to categorical_crossentropy, and the optimizer was set to RMSprop. The model was evaluated on the metrics; AUC score, Precision, Recall, and Accuracy.

Support Vector Machine

The Support Vector Machine was initialized by setting the probability parameter as true. The training dataset was fitted into the model which was later tested against the testing dataset. Below are the results obtained:

- Training Accuracy: 98.84 %
- Testing Accuracy: 90.81%
- Training AUC: 0.99
- Testing AUC: 0.95

4. Results

After implementing the model into practice, we obtained a 98.8% training accuracy and a 90.8% testing accuracy for SVC with TF-IDF. Additionally, we obtained a train AUC of 99.85 and a test accuracy of 95.6%. The KNN model had the best accuracy among them, coming in at 86.4% after we created the pipeline for our model and ran numerous models. Prior to performing our sentimental analysis, we cleaned the data and used count vectorization and TF-IDF vectorization, which improved the dataset and improved the performance of our model.

5. Discussion

Although the accuracy obtained from this study was high enough to conclude that the tweets were being correctly classified into one of the 3 categories, further improvements can always be made. One way this could be done is that the number of tweets taken for this study was limited to the date of February of 2015. If more tweets could be considered, the accuracy of categorizing tweets into a class of sentiment would have been much higher.

Apart from altering the data, making use of more complex vectorizers to transform the data into a numerical format would aid in better performing accuracies.

6. Conclusion

Throughout this study, we altered our dataset a few times to tune it in order to fit properly into a model. It was found that the neutral sentiment did not add much value into the study and was hence removed. From all the 9 models which were used to train the data, the Sequential Neural Network model has obtained the highest accuracy of 96.12% using the CountVectorizer technique to transform the data.

Another main decision made during this study was to exclude the results obtained from Label Encoding. This was to support the fitting of the data into the models as a Label Encoder did not contain any number of rows after the transformation.

7. Acknowledgment

This project was completed in the summer semester 2022 of CS6120 taught by Professor Uzair Ahmad.

8. Reference

[1] Apoorv Agarwal, Boyi Xie, Ilia Vovsha, Owen Rambow, Rebecca Passonneau. Sentiment Analysis of Twitter Data.

[2] Fuchun Peng. 2003, Augmenting Naive Bayes Classifiers with Statistical Language Models

[3] Multi Perspective Question Answering (MPQA) Online Lexicon
http://www.cs.pitt.edu/mpqa/subj_lexicon.html

[4] Tweet Stream: Simple Twitter Streaming API Access http://pypi.python.org/pypi/tweetstream

[5] Soo-Min Kim and Eduard Hovy. Determining the Sentiment of Opinions. In Proceedings of International Conference on Computational Linguistics (ICCL), 2004.

[6] Feldman, R. (2013). Techniques and applications for sentiment analysis. Communications of the ACM, 56(4), 82-89.

[7] Trinh, S., Nguyen, L., Vo, M., & Do, P. (2016). Lexicon-based sentiment analysis of Facebook comments in Vietnamese language. In Recent developments in intelligent information and database systems (pp. 263-276). Springer International Publishing.