Understanding User Sentiments Using KNN Method

Chintan Ajmera
cajmera@umich.edu
University of Michigan, Data Science

Nathan Martin
nathmart@umich.edu
University of Michigan, Data Science

1. INTRODUCTION

In a world that is as connected as today's, the importance of what we say, where we say them, and how we say them all become as important as ever. The internet, and specifically social media, has become a place for us to express ourselves, whether it be with opinion or facts. More often than not, it can be noted that the posts on such social media forums tend to be linked to the emotions and/or sentiments of the owner of the posts upon a particular subject matter.

Being able to detect the sentiment of such posts presents us with tremendous real-world value. It is a vastly studied area under the realm of Natural Language Processing as by being able to understand speech and text in more of a quantitative manner, we are able to actually learn more about the qualitative nuances of our own languages.

Similarly, the goal of this research paper is to understand the sentiment of social media posts of users, specifically Twitter tweets data of Airlines (discussed more in the Data section of this paper). While there are many types of sentiment analysis experiments that can be performed, this paper will look to conduct 'polarity analysis', which will help us detect whether the tweets of users are positive, negative, or neutral.

To do this, we will use two big Data Mining techniques that we have learned throughout the course: classification using K-Nearest-Neighbors and outlier/anomaly detection using the Isolation Forest algorithm. While outlier/anomaly detection was something that was covered in the first half of the course, classification using K-Nearest-Neighbors is topic that was discussed in the second half of the course (post midterm). Additionally, it is to be noted that neither of these techniques were used in projects, so they are brand new in terms of their usage for this project. Furthermore, we will also be some standard data cleaning and data processing steps to our tweets data, however we have not counted those steps as part of the main two techniques for this project.

Figure 1 depicts a diagram of the steps taken in this project (also further detailed in the Data Analysis section of the paper).

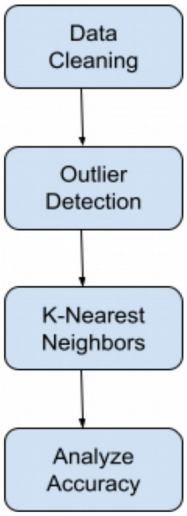

Figure 1: Project Pipeline

2. DATA

The data set that we are working with is titled "Twitter US Airline Sentiment" that is open-sourced by Kaggle on their website, but originally belongs to the company *Appen*. *Appen* is a "Data sets Resource Center" that creates and hosts data sets in various categories for different AI-applications. This dataset, while it was last updated two years ago, was the most complete and easy-to-use data set we could find for the purpose of this project.

The data set itself is made up of around 14,500 total individual tweets pertaining to six major US airlines. This is further divided into two sub-groups: training and testing data. The training set contains approximately 11,500 tweets whereas the testing data set contains around 3,000 that we will be using to test the accuracy of our different KNN models.

Moreover, the Twitter data set has fifteen different features ranging from the actual airline sentiment rating (positive, negative, or neutral) to the actual reason from the text why that particular tweet was labeled the way it was to the actual name of the airline as well. While we will not be making use of all such columns of the data set, it is still a valuable asset to have so many features for future experimentation.

However, as responsible data scientists, we noticed that there were also some features in the original data set that failed to provide anonymity of the collected data as it contained the user id of the tweet along with the tweet location and other sensitive information that not only would not help our classifier, but also violates some data collecting ethics. Hence, we have eliminated those columns in our data cleaning/processing stages.

The image below depicts some of the rows of the data set. (Note that only some of the features of the data set/attributes are depicted as all the dimensions will not fit. Please refer to the Kaggle dataset reference at the end of the paper to see the original source of the dataset in its entirety).

tweet_id	airline_sen...	negativere...	airline
570306133677760513	neutral		Virgin America
570301130888122368	positive		Virgin America
570301083672813571	neutral		Virgin America
570301031407624196	negative	Bad Flight	Virgin America
570300817074462722	negative	Can't Tell	Virgin America
570300767074181121	negative	Can't Tell	Virgin America

Figure 2: Selected Kaggle Dataset

3. DATA ANALYSIS

3.1 Question

What sort of outliers can we find in airline tweets and how can we perform polarity analysis on said tweets using KNN?

3.2 Data Required

The data required to answer the question above is listed in the introduction, a US Airline Sentiment dataset with nearly 14,500 tweets regarding US airlines. We used the entirety of the dataset.

3.3 Data Mining Techniques

We began our data analysis by preparing our dataset. First, these are tweets, so there are a large number of random punctuations, links, and other text entries that are irrelevant to NLP. We utilized Regex to remove things like hashtags, hyperlinks, special characters, and usernames, because these provide no insight into the sentiment of the text. Afterwards, we scanned through our train and test datasets to remove stopwords. Of course for more advanced sentiment analysis, you would want all these words because then you would be looking at sentences and order of words, but that's beyond the scope of this assignment.

Next, for our Word2Vec implementation, we turned each cleaned tweet into a vector representation using the Word2Vec library given by Google. This algorithm uses neural networks to learn word associations from a large text corpus and is used for sentence completion. This allows us to turn each tweet into a vector representation.

Then, we utilized sklearn's IsolationForest method to perform anomaly detection. This method of outlier detection was introduced in 2013 and serves to explicitly isolate anomalies through sub-sampling. We used 100 estimators and ended up removing around 1100 tweets from our dataset.

Finally, we performed KNN using the sklearn library with k=1, 3, 5, 7, and 10.

3.4 Experiment Setup

We set this up in Google Colab and ran with different estimators on Isolation Forest and different parameters on KNN using a grid-search method for finding good parameters.

3.5 Results

Our outlier detection methods removed close to 1100 tweets from our dataset of around 18,000 tweets, which is not an insignificant amount. As we can see from the below charts, it negatively impacted our KNN performance. Figures 3 and 4 below depict the accuracy scores as such.

Before outlier detection:

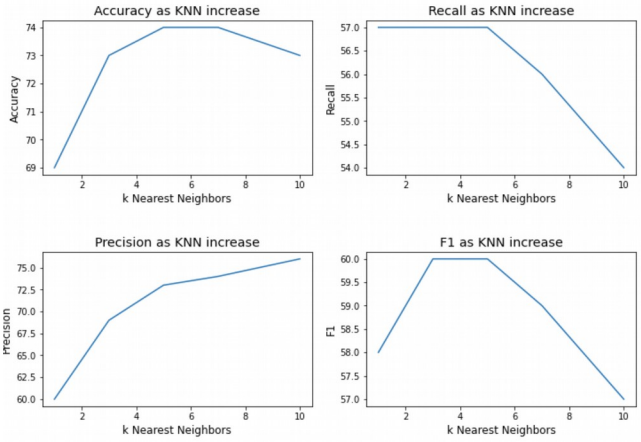

Figure 3: Pre-Outlier Detection Accuracy scores

After outlier detection:

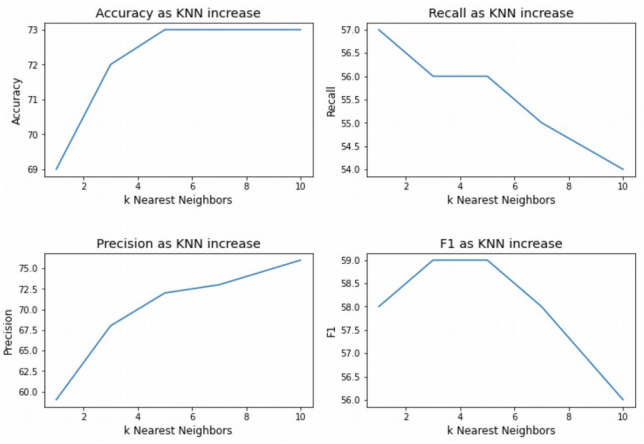

Figure 4: Post-Outlier Detection Accuracy scores

Notably, we do not see a massive change in performance, but especially around k=4-9 we see a drop in accuracy and recall. Our accuracy also flattens out at 10 instead of dropping, which is an interesting result.

It would be interesting to see how LOF would compare to Isolation Forest in outlier detection here. One trend we noticed with our method was that it mainly detected outliers in terms of tweets that had very little information in them after regex was performed, i.e. most of the tweet was a link and had been reduced greatly from before.

3.6 Challenges

The hardest challenge to begin with was deciding how to convert our text into features. An initial consideration was a bag of words model, but that proved to be quite inaccurate, as it did not take into consideration the placement of the words. However, Word2Vec considers the context of the word, which greatly increased the accuracy of our feature construction.

In addition, which classifier to use was also a bit of a challenge. We settled on KNN since it allowed us to utilize a topic which we covered in class but didn't use on a project.

4. CONCLUSIONS & DISCUSSION

The first most interesting result from our analysis is how our outlier removal decreased the accuracy of our classifier. Outlier removal is not necessary in every data analysis problem, sometimes outliers are not necessarily harmful and needn't be removed.

Second, our best accuracy achieved was 74.0% with a k value of 7. This tells us that we can predict the sentiment of tweets pertaining to airlines nearly three quarters of the time, which is quite impressive. A company could use this to gauge consumer approval with their product or a stock trader could gauge the same approval to know when to buy or sell shares in said company.

There were two big challenges that we faced in conducting this research project. The first being finding adequate data that would suffice our case study. During our planning stage for this project, we realized that finding data in both high quantity and high quality is tough ask. We were often able to find one, but not the other. Finally, once we narrowed in on conducting the sentiment analysis on tweets specifically, we came across the Kaggle data set.

The other challenge was conducting the outlier analysis on our data. LOF outlier analysis would work best on our data, but since a project on that was already done within the course, we had to find a different outlier analysis method that was outside the scope of the coursework. Finally, we decided on using the isolation forest algorithm as it can help us find points that can be considered anomalies for our KNN classification.

There was a great deal that we learned by doing this project. Foremost, we got to solidify our knowledge of the K-Nearest-Neighbor algorithm as well as improve upon our skills of doing outlier analysis on a dataset with the isolation forest algorithm. While we had originally implemented the LOF outlier detection technique to one of our previous projects, isolation forest was a very different technique for us. Similarly, while we had done an SVM classification problem for a previous project, implementing KNN was completely new as we had only learned about it on paper but application is a different task altogether. However we thoroughly enjoyed both of these new challenges and we are glad that we walked away learning some new skills.

What we liked the most, perhaps, is the knowledge and appreciation we gained in learning how to organize a data mining from the ground up. All the projects that we have done for the course have had a specification for us to follow and implement along with pre-processed data given to us. However, now that we had to conduct the project on our own, a lot more time was invested in the planning stage initially before we

could just start coding. While challenging, this was extremely rewarding and satisfying to see results from a project come alive from its research stage all the way to its write up stage.

However, there were also some limitations of this project that we are looking to address. While some of these limitations are due to time constraints, others are due to personal choices that we made during the course of the project that we feel can be improved upon for the next iteration of research. Foremost, while the data we used was the best we could find, it is still a bit outdated as it was last updated 2.5 years ago. Additionally, it was unclear from the Kaggle data description as to whether the labels of the tweets were from a computer program (hence having a probability of being mislabeled) or hand-labeled. Either way, if we had more time we would have loved to create a data set ourselves that would not only be more up-to-date and accurate, but also curated for just our specific research needs.

The other aspect that we feel we can embed into our research for the next time is of comparison of different methods. Specifically, during the later stages of the project, we discovered that there are actually more variations to the Word2Vec model that we used to develop the different word associations within our model. For instance, the BERT per-trained model by Google is a more recent and higher performing Natural Language Processing model as it takes into consideration the context and the ordering of the words that Word2Vec ignores. Furthermore, more recently the Stanford NLP Group developed the CoreNLP pipeline that derives linguistic annotations for text and considers specifics such as parts of speech, numeric and time values, and others. It would be really interesting to see a head-to-head comparison of all such NLP models to see which one delivers the most accurate classifications within the KNN infrastructure.

In detailing what each person contributed to the project, it was overall a complete team effort. Both of us contributed equally in every stage of the project: planning, coding, write-up, etc. Even times when we had questions about the project, we attended office hours as a team as well in order to ensure that we were both on the same page and not have any communication gaps down the line.

5. ACKNOWLEDGMENTS

We like to acknowledge first and foremost the Kaggle community for hosting this dataset on their website and open sourcing it so users like us can learn more about Data Mining techniques by using it to our advantage. Having the data in a nice and easy format with many interesting features allows for a very intuitive understanding of the data that truly is invaluable.

Additionally, we would also like to acknowledge the staff and coordinators of the course. It has been tough semester for many of us, but the continuous help over Piazza and at office has been great support in times of high stress. The projects and homework in this course, while not the easiest, have had the most real-life applications that we feel we have seen across the courses we have taken. We truly appreciate all the efforts put in this semester.

6. REFERENCES

[1] Appen. 2018. Twitter US Airline Sentiment. *Kaggle*. April 20, 2021. https://www.kaggle.com/crowdflower/twitter-airline-sentiment

[2] Liu, F., Ting. K., and Zhou, Z. 2013. *Isolation Forest*. Gippsland School of Information Technology and National Key Laboratory for Novel Software Technology. April 20, 2021. https://cs.nju.edu.cn/zhouzh/zhouzh.files/publication/icdm08b.pdf?q=isolation-forest

[3] Huq, M., Ali. A., and Rahman, A. 2017. *Sentiment Analysis on Twitter Data using KNN and SVM*. International Journal of Advanced Computer Science and Applications (IJACSA). April 20, 2021. https://thesai.org/Downloads/Volume8No6/Paper_3-Sentiment_Analysis_on_Twitter_Data_using_KNN_and_SVM.pdf

[4] Karani, D. 2018. *Introduction to Word Embeddings and Word2Vec*. Towards Data Science. April 20, 2021. https://towardsdatascience.com/introduction-to-word-embedding-and-word2vec-652d0c2060fa

[5] Mougan, C. 2020. *Isolation Forest from Scratch*. Towards Data Science. April 20, 2021. https://towardsdatascience.com/isolation-forest-from-scratch-e7e5978e6f4c

[6] TensorFlow Core Tutorials. *Word2Vec*. Google Tensorflow. April 20, 2021. https://www.tensorflow.org/tutorials/text/word2vec