ANALYZING SENTIMENTS FROM SOCIAL MEDIA

Kartikeya Gupta, Anikait Sahota

ABSTRACT

Expressing sentiments on social media has become an important way for people to express their opinions. Understanding these can help set rules and protocols to protect people's opinions and maintain decorum. The dataset titled "Sentiment140 dataset with 1.6 million tweets" on Kaggle has about 1.6 million tweets which have been extracted using the twitter API. The tweets have been labelled as negative or positive and are used to train the machine learning model to detect sentiment of a tweet. We discuss the problem of classifying the positive and negative tweets and various ways to extract the information from tweets text, username and date of tweets. Various classification models like Logistic Regression, Decision Trees, XGBoost and Naïve Baye were tried and the best one was chosen based on evaluation metrics like F1 score and accuracy.

Git link - https://github.com/AnikaitSahota/ML-project

1. INTRODUCTION

With the increase in use of social media, people are able to access information much more easily. But with information, also comes opinions of other individuals, including the positive and negative perspective on various topics. With the differences in opinions and biases in viewpoints, these discussions can easily grow into bullying, hate comments, and harassment. It is important to maintain a positive environment so that its safe for everyone to communicate, discuss, debate and give opinions on social media, without being worried any backlash they might get. Social media is the ideal medium to observe and study the expression of emotions, and in turn, study the sentimental values each expression holds.

2. LITERATURE SURVEY

As discussed, there were three main procedures that were followed. The creation of template database, the processing of test image, and the detection algorithm.

2.1. Expressively Vulgar: The Socio-dynamics of Vulgarity

Study of vulgarity, also referred as profanity or use of swear/curse words has been of interest to linguists, psychologists and computer scientists. They have been studied in social media and online communities to determine their relation with expressing sentiments of a person, as well as their socio-economics impact.

This paper discusses how semantic vulgarity while expressing one's thought. It tries to answer question of how vulgar words and expression impact the perception of sentiments and does modeling vulgarity helps in analysis of the sentiments. The results show that vulgar words, does indeed play a role in sentiment. Majority of the time, vulgar words were used to intensity an already existing sentiment, and interacts with number of features like age, gender, education, income and political ideologies. These show to increase performance of sentiment analysis system.

2.1. DepecheMood++: Emotion lexicon Built

This paper discussed about how to obtain a high precision and high coverage lexica for sentiment and emotion recognition in English and Italian. It also discusses various traditional lexica using two distinct ways (manual and automatic creation). DepecheMood++'s way of obtaining lexicon (i.e. Using pearson correlation between data points file's word and target for training).

- Getting normalized frequency of words for every datapoint file.
- And building word-by-emotion matrix. Then finding correlation.

4. DATASET

In determining the role of sentiments and how to extract in written expressions, social media provides the perfect platform to obtain data. Among the various social media, Twitter with its particular function of allowing people to tweet, naturally provides a vast volume of text expressed by a number of people all over the world. Thus, the dataset and all further analysis, preprocessing, training and evaluation was done on tweets.

Dataset includes 1.6 million tweets extracted using twitter API, where each tweet has been annotated, 0 being negative, 2 being neutral and 4 being positive sentiment. It contains the following 6 fields –

- Target - polarity of the tweet (0 = negative, 2 = neutral, 4 = positive)

- Ids - id of the tweet {integer}
- Date - date of the tweets
- Flag - the query, if there is no query, then it is marked as NO_QUERY
- User - the user that tweeted the tweet {string}
- Text - the text in the tweet {string}

After plotting the target values for all tweets in the dataset, we noticed there were actually no neutral tweets. This reduced the target feature to a binary attribute with either value 0 (negative) or 4 (positive).

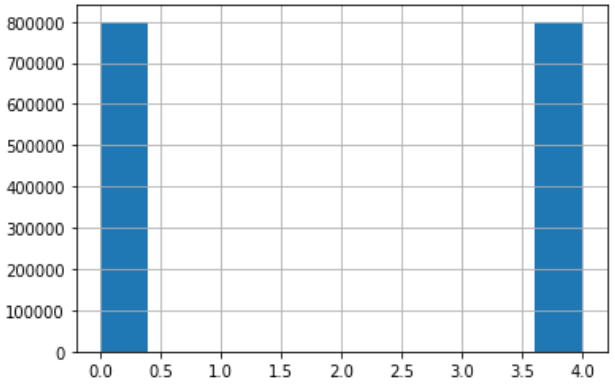

Figure 1 – Plotting of target values for all tweets in the dataset showing the distribution of tweets.

After plotting the dates for all tweets in the dataset and the count of positive and negative sentiment tweets for every date, we noticed there were mostly negative sentiments tweets between June and July.

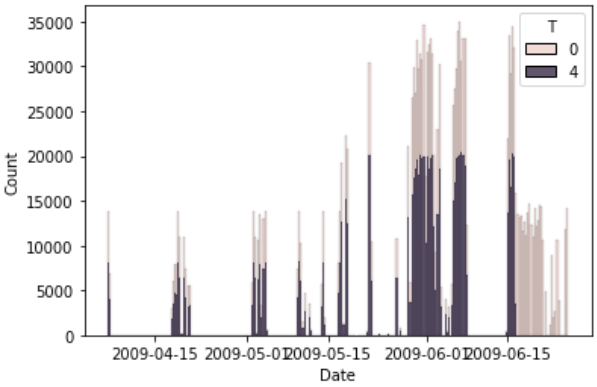

Figure 2 – Plotting of dates a tweet was tweeted, with respect to the frequency of a tweets occurring during a month.

After plotting the flag for all tweets in the dataset, we noticed there was only a single flag = 'NO_QUERY' (i.e. common between data points).

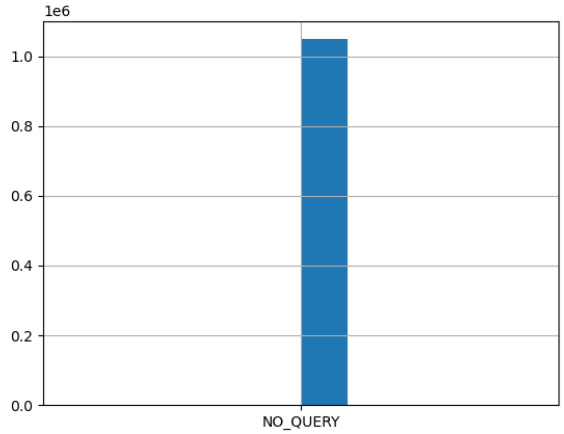

Figure 3 – Plotting of flag of tweets with respect to the frequency of tweets.

Since the most important part of a tweet that holds the most information in expressing the sentiment of a user is the text, we had to apply various preprocessing techniques to make the text be usable as features for our model.

We will be removing words or expressions that don't contribute in determining the sentiment. These include twitter handles (@name), short words (is, and, the) and punctuations. The cleaned tweets will be used for attribute selection then.

```
                                                Tweet
is upset that he can't update his Facebook by ...
@Kenichan I dived many times for the ball. Man...
  my whole body feels itchy and like its on fire
@nationwideclass no, it's not behaving at all....
                    @Kwesidei not the whole crew
```

```
                                           Tidy_Tweets
upset that update Facebook texting might resul...
   dived many times ball Managed save rest bounds
                 whole body feels itchy like fire
                 behaving here because over there
                                          whole crew
```

Figure 4 – Difference of before and after processing the text in the tweets.

After obtaining the tidy tweets, the bag-of-words extraction method was used. It is a method to extract features that can be directly used in machine learning models from text documents. After applying this technique, each tweet was converted into a 1000 feature vector where each feature denoted the term frequency of each word in the tweets. Date feature was broken in 5 more features, day, number date, month, year, time in minutes (13:01 = 13*60 + 1). For username, we used one-hot encoding to use them as features. This resulted in formation of 659775 features.

Total number of features were very high when combining date, username and tidy tweets, and were exceeding memory limit of py3. So, we used dimensionality reduction techniques SVD to reduce the feature numbers. The total features were reduced to 255.

4. METHODOLOGY

Preprocessing mainly involves text manipulation and cleaning, which is done by using strings and dictionaries in python. The tweets were cleaned using various python techniques like lambda function and regex expressions. To convert the tidy tweets to usable features bag-of-words feature extraction technique was used. This was applied by using the sklearn.feature_extraction.text library. These features along with dates and username features were then reduced in dimension using SGD feature decomposition technique, which was implemented using sklearn.decomposition library.

Since our problem is to classify tweets in positive, neutral or negative category, we will be using classification models to train the data. We have tried out various possible models based on the outlined details in the dataset and related to the problem statement of classification. Models used are –

1.1. Logistic Regression

Since we are having the classification problem, which is not a linear equation problem (reason for neglecting the linear regression), we use logistic regression. We got 67% accuracy and F1-score as {0: 69, 1: 66} using L2 loss.

1.2. Decision tree

We also applied the decision tree classifier model and found that we got an accuracy of 70%, with F1-score of {0: 70, 1: 70} using the best hyperparameters.

1.3. XGB classifier

Since the decision tree has been an important model to classify but it was taking too much time to compile, hence we also opted for XGB classifier, extending the logical boundary characteristics using it. We got 79% accuracy and F1-score as {0: 78, 1: 80} using learning rate 0.9.

1.4. Gaussian Naïve Bayes

Another model which is used for classification problem is gaussian naïve bayes. We got an accuracy of 64% using gaussian naïve bayes and F1-score as {0: 64, 1: 64}.

5. RESULT AND ANALYSIS

Various conclusions could be made after analysis of the dataset and training of the models -

- Since there are no neutral tweets present in the data, the problem reduces from multi-class classification to normal binary classification problem.
- UID and Flags were removed as features for the training dataset. Since username was not unique and multiple tweets were made by single user, username was kept as features for training.
- Date was divided into multiple features like month, date, hour and year to be used as a parameter. Username were used after one-hot encoding and features were extracted from the clean tweets using bag-of-words.
- XGboost classifier is outperforming all the other classifiers with an accuracy of 79 percent.

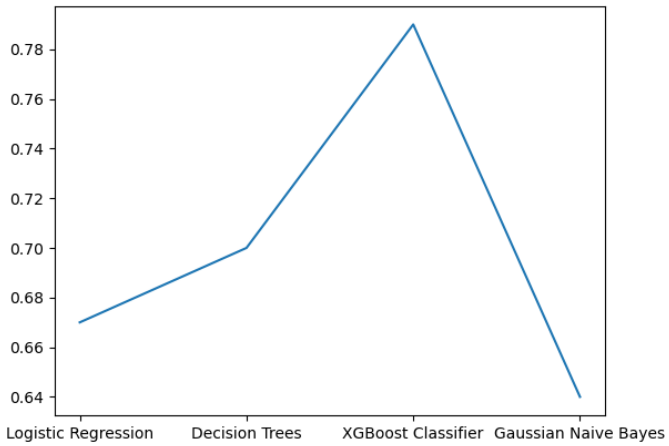

Figure 5 – Accuracies of various models

6. CONCLUSION

Since this was the first time, we handled dataset heavily based on text, we learned not only how to extract features from text-based dataset, but also how to analyze and clean the text to only use words that convey information about the data. This introduced us to regex expressions and how to use it to clean the texts. Since decision tree is taking too much time to train, we discovered a new model training technique called XGBoost that is an optimized decision tree classifier. Also, the project showed us how important dimensionality reduction can be after seeing the number of features we ended up with after apply one-hot encoding for usernames and bag-of-words on the tweets.

We also notice how XGBoost is performing comparatively much better than other models. This shows us how boosting can improve an already existing model like decision trees.

Both team members worked cooperatively to distribute the work. Kartikeya used bag of words to extract features from the tweets for data preprocessing, and Anikait used one hot encoding to convert usernames to usable features as well as divide date into 5 features to make them understandable as a feature. Feature decomposition technique was used after seeing the number of features were too many by Anikait. Models were equally distributed, with kartikeya trying Logistic Regression and naïve bayes classifier, and Anikait tried Decision tree and XGBoost classifier. We shared our findings with each other, as well as discussed codes to try various changes in hyperparameters. While the models were being tested, the best findings were being recorded by each member and final report was compiled together by Kartikeya.

6. REFERENCES

[1] Mozetič, I., Grčar, M. and Smailović, J., 2016. Multilingual Twitter sentiment classification: The role of human annotators. *PloS one*, *11*(5), p.e0155036.

[2] Poria, S., Hazarika, D., Majumder, N., Naik, G., Cambria, E. and Mihalcea, R., 2018. Meld: A multimodal multi-party dataset for emotion recognition in conversations. *arXiv preprint arXiv:1810.02508*.

[3]Cachola, I., Holgate, E., Preoţiuc-Pietro, D. and Li, J.J., 2018, August. Expressively vulgar: The socio-dynamics of vulgarity and its effects on sentiment analysis in social media. In *Proceedings of the 27th International Conference on Computational Linguistics* (pp. 2927-2938).