SENTIMENT ANALYSIS

DSC 680 -PROJECT2 MILESTONE 2

Abhijit Mandal

04/30/2022

Introduction

This project will focus on performing a sentiment analysis on some tweets in Twitter which will be chosen as the project progresses. The main goal of this analysis is to discover the underlying sentiment from a users tweet. The opinions that are mined will be classified into two categories positive and negative. An analysis will then be performed on the classified data to see what percentage of the population sample fall into each category.

Natural Language Processing (NLP) is a hotbed of research in data science these days and one of the most common applications of NLP is sentiment analysis. From opinion polls to creating entire marketing strategies, this domain has completely reshaped the way businesses work, which is why this is an area every data scientist must be familiar with.

Thousands of text documents can be processed for sentiment in seconds, compared to the hours it would take a team of people to manually complete the same task.

We will do so by following a sequence of steps needed to solve a general sentiment analysis problem. We will start with preprocessing and cleaning of the raw text of the tweets. Then we will explore the cleaned text and try to get some intuition about the context of the tweets. After that, we will extract numerical features from the data and finally use these feature sets to train models and identify the sentiments of the tweets.

Business Problem:

Sentiment analysis can be defined as a process that automates mining of attitudes, opinions, views and emotions from text, speech, tweets and database sources through Natural Language Processing (NLP). Sentiment analysis involves classifying opinions in text into categories like "positive" or "negative”. It's also referred as subjectivity analysis, opinion mining, and appraisal extraction.

The words opinion, sentiment, view and belief are used interchangeably but there are differences between them.

Opinion: A conclusion open to dispute (because different experts have different opinions)

View: subjective opinion

Belief: deliberate acceptance and intellectual assent

Sentiment: opinion representing one’s feelings

Sentiment analysis and Natural Language processing are very important area nowadays. There is a massive amount of information being uploaded to the internet daily on social media websites and blogs that computers cannot understand. Traditionally it was not possible to process such large amounts of data, but with computer performance following the projections of Moore’s law and the introduction of distributed computing like Hadoop or Apache Spark, large data sets can now be processed with relative ease. With further research and investment into this area, computers will soon be able to gain an understanding from text which will greatly improve data analytics and search engines.

A good use case is to identify a customer’s perception for a product, this is an extremely valuable data to some companies. From the knowledge gained from an analysis such as this a company can identify issues with their products, spot trends before their competitors, create improved communications with their target audience, and gain valuable insight into how effective their marketing campaigns were. Through this knowledge companies gain valuable feedback which allows them to further develop the next generation of their product.

Approach:

This section will highlight the technical approach that will be followed for this project and will include the system description.

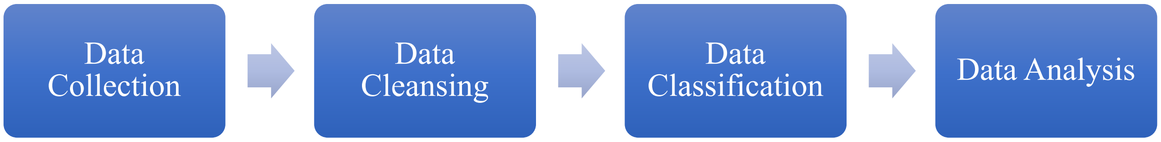
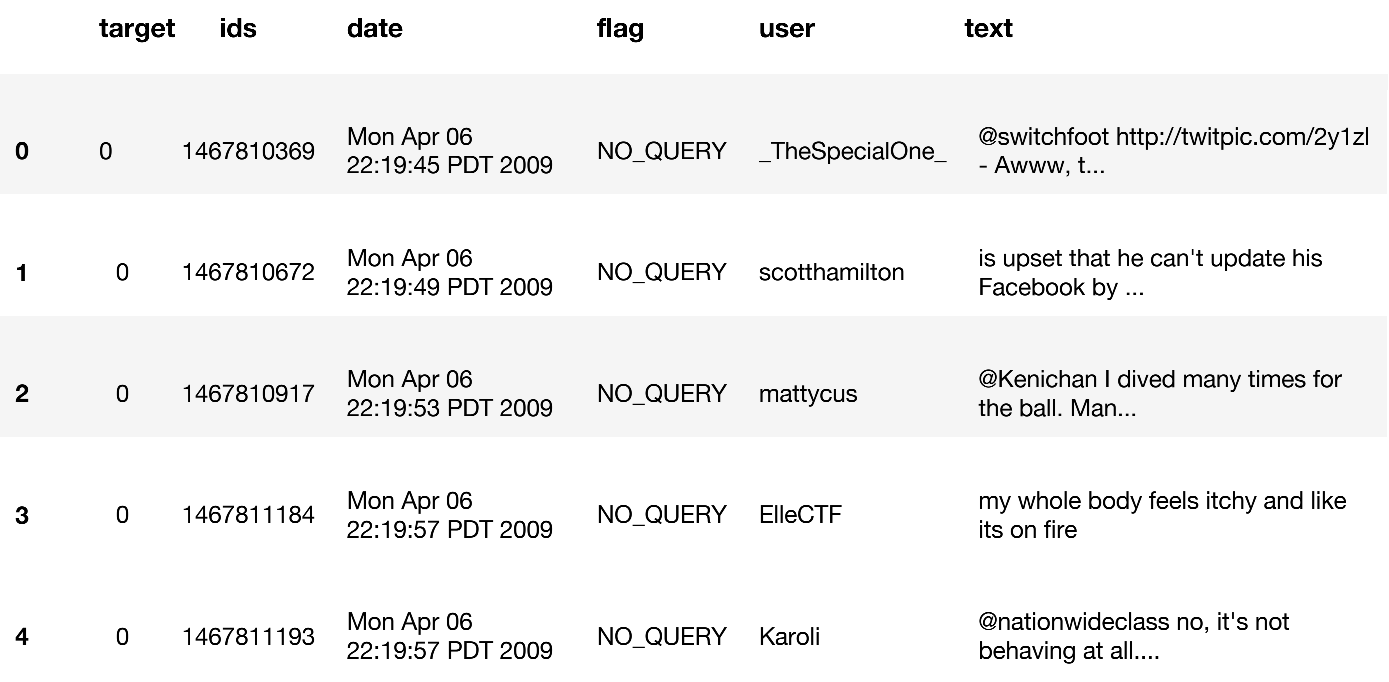


Figure-1 Approach for Sentiment Analysis

Data Collection

The dataset provided is the Sentiment140 Dataset which consists of **1,600,000** tweets that have been extracted using the Twitter API. The various columns present in the dataset are:

- **target:** the polarity of the tweet (positive or negative)
- **ids:** Unique id of the tweet
- **date:** the date of the tweet
- **flag:** It refers to the query. If no such query exists then it is NO QUERY.
- **user:** It refers to the name of the user that tweeted
- **text:** It refers to the text of the tweet



	target	ids	date	flag	user	text
0	0	1467810369	Mon Apr 06 22:19:45 PDT 2009	NO_QUERY	_TheSpecialOne_	@switchfoot http://twitpic.com/2y1zl - Awww, t...
1	0	1467810672	Mon Apr 06 22:19:49 PDT 2009	NO_QUERY	scotthamilton	is upset that he can't update his Facebook by ...
2	0	1467810917	Mon Apr 06 22:19:53 PDT 2009	NO_QUERY	mattycus	@Kenichan I dived many times for the ball. Man...
3	0	1467811184	Mon Apr 06 22:19:57 PDT 2009	NO_QUERY	ElleCTF	my whole body feels itchy and like its on fire
4	0	1467811193	Mon Apr 06 22:19:57 PDT 2009	NO_QUERY	Karoli	@nationwideclass no, it's not behaving at all....

Figure-2 Data Snapshot

Data Cleansing

A tweet contains a lot of opinions about the data which are expressed in different ways by different users. The twitter dataset used in this project work is already labeled into two classes viz. negative and positive polarity and thus the sentiment analysis of the data becomes easy to observe the effect of various features. The raw data having polarity is highly susceptible to inconsistency and redundancy. Preprocessing of tweet include following points,

- Remove all URLs (e.g. www.xyz.com), hash tags (e.g. #topic), targets (@username)
- Remove Stop words.
- Replace Repeated Characters.
- Remove all punctuations, symbols, numbers.

The second phase of the system will be to cleanse the data collected, this will involve removing any punctuations and making everything lower case. This will help in the next stage of the project especially in the “Bag of Words” approach. Removing lower case words will decrease the redundancy in the database that will be used to store the words.

Classifying the data

To reach the ultimate goal, there was a need to clean up the individual tweets. I used a concept known as "Tokenization" in NLP. It is a method of splitting a sentence into smaller units called "tokens" to remove unnecessary elements. Another technique worthy of mention is "Lemmatization". This is a process of returning words to their "base" form. A simple illustration is shown below.

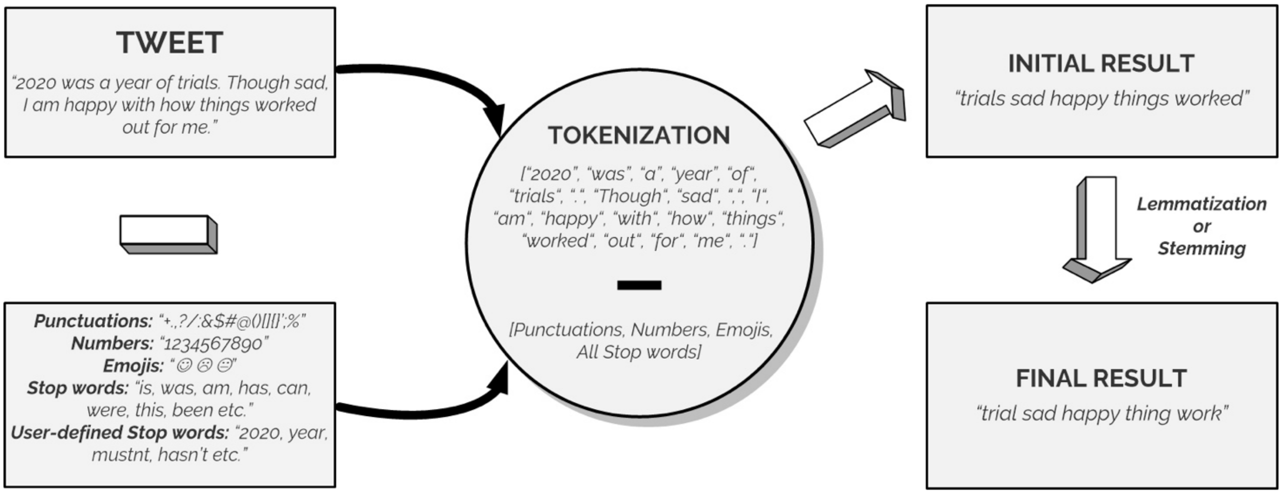

TWEET
"2020 was a year of trials. Though sad, I am happy with how things worked out for me."
Punctuations: "+.,?/:&$#@()[][]';%"
Numbers: "1234567890"
Emojis: "☺ ☹ 😑"
Stop words: "is, was, am, has, can, were, this, been etc."
User-defined Stop words: "2020, year, mustnt, hasn't etc."
TOKENIZATION
["2020", "was", "a", "year", "of", "trials", ".", "Though", "sad", ",", "I", "am", "happy", "with", "how", "things", "worked", "out", "for", "me", "."]
[Punctuations, Numbers, Emojis, All Stop words]
INITIAL RESULT
"trials sad happy things worked"
Lemmatization or Stemming
FINAL RESULT
"trial sad happy thing work"

Figure-3 – Tokenization and classification

Machine learning techniques require representing the key features of text or documents for processing. These key features are considered as feature vectors which are used for the classification task. Some examples feature that have been reported in literature are:

1. Words and Their Frequencies: Unigrams, bigrams and n-gram models with their frequency counts are considered as features.

2. Parts of Speech Tags: Parts of speech like adjectives, adverbs and some groups of verbs and nouns are good indicators of subjectivity and sentiment. We can generate syntactic dependency patterns by parsing or dependency trees.

3. Opinion Words and Phrases: Apart from specific words, some phrases and idioms which convey sentiments can be used as features. e.g. cost someone an arm and leg.

4. Position of Terms: The position of a term within a text can affect how much the term makes difference in overall sentiment of the text.

5. Negation: Negation is an important but difficult feature to interpret. The presence of a negation usually changes the polarity of the opinion.

6. Syntax: Syntactic patterns like collocations are used as features to learn subjectivity patterns by many of the researchers.

This is expected to be the most difficult part of the project; it will entail looking at individual words or groups of words in a tweet and attempting to assign a sentiment to them. This is no easy task as it is very difficult for a computer to “understand” slang words and sarcasm.

“Bag of Words” Model The bag of words approach will involve building databases of positive, negative, and neutral words. Each tweet will be broken up into individual words and then compared to the words in the databases. When there is a match, a counter will be incremented or decremented by a fixed amount depending on a weighting assigned. When this process is complete the counter will be used to classify the sentiment for example if the words in the tweet are largely positive the counter should be high.

To get the most common words used, I made use of the POS-tag (Parts of Speech tagging) module in the NLTK library. Using the WordCloud library, one can generate a Word Cloud based on word frequency and superimpose these words on any image. In this case, I used the Twitter logo and Matplotlib to display the image. The Word Cloud shows the words with higher frequency in bigger text size while the "not-so" common words are in smaller text sizes.

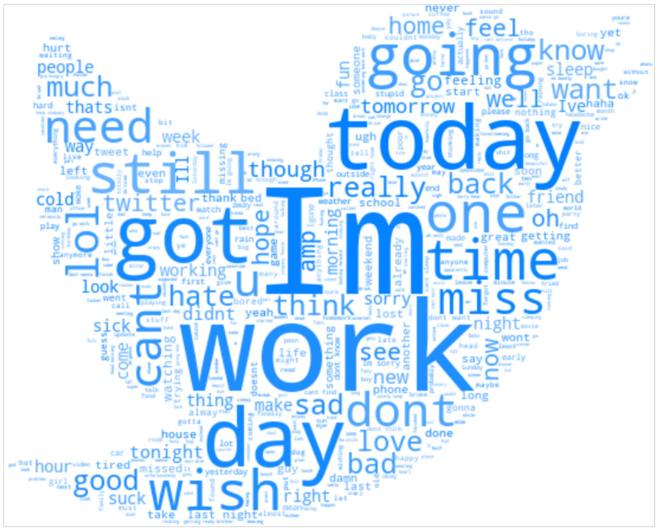


Figure-4 – Negative Sentiment Word Cloud

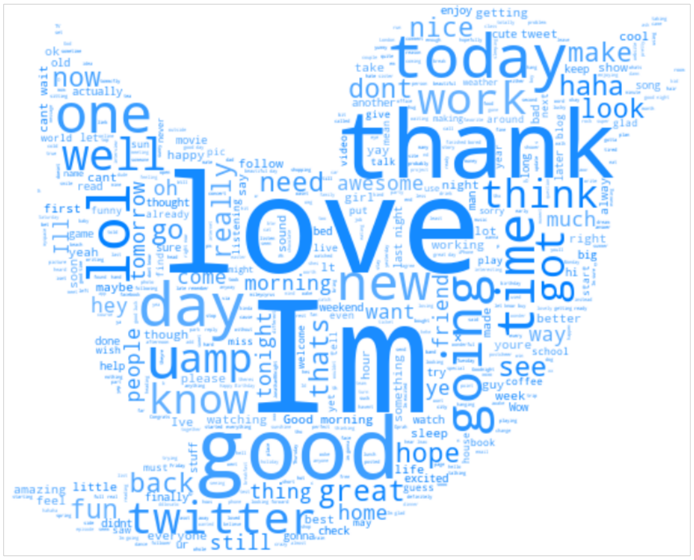


Figure-5 – Positive Sentiment Word Cloud

Data Analysis

When the data is classified, there will have to be analysis performed on it. This may include simple percentages of customer satisfaction, or a more complex analysis could be performed such as comparing the customer sentiment on two similar products with the aim of finding a correlation between good sentiment and high sales of those products.

Setting up the Classification Model

After training the model we then apply the evaluation measures to check how the model is performing. Accordingly, we use the following evaluation parameters to check the performance of the models respectively :

- Accuracy Score
- Confusion Matrix with Plot
- ROC-AUC Curve

MODEL1 : SVM (SUPPORT VECTOR MACHINE)

```
from sklearn.svm import SVC
clf=SVC()
clf.fit(X_train,y_train)
y_pred=clf.predict(X_test)
from sklearn.metrics import accuracy_score,confusion_matrix
test_acc=accuracy_score(y_test,y_pred)
print(test_acc)
cfm=confusion_matrix(y_test,y_pred)
print(cfm)
```

```
0.77375
[[1492  511]
 [ 394 1603]]
```

```
# Print the evaluation metrics for the dataset.
# Compute and plot the Confusion matrix

cf_matrix = confusion_matrix(y_test, y_pred)
categories = ['Negative','Positive']
group_names = ['True Neg','False Pos', 'False Neg','True Pos']
group_percentages = ['{0:.2%}'.format(value) for value in cf_matrix.flatten() / np.sum(cf_matrix)]
labels = [f'{v1}n{v2}' for v1, v2 in zip(group_names,group_percentages)]
labels = np.asarray(labels).reshape(2,2)
sns.heatmap(cf_matrix, annot = labels, cmap = 'coolwarm',fmt = '',
xticklabels = categories, yticklabels = categories)
plt.xlabel("Predicted values", fontdict = {'size':14}, labelpad = 10)
plt.ylabel("Actual values" , fontdict = {'size':14}, labelpad = 10)
plt.title ("Confusion Matrix", fontdict = {'size':18}, pad = 20)
```

Code Snippet for SVM

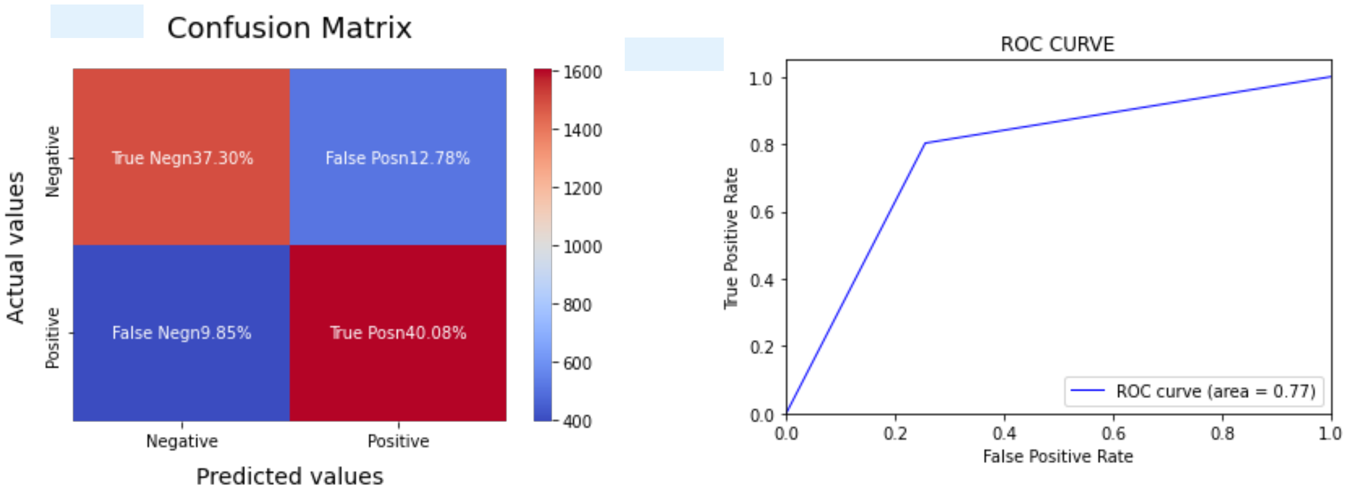


Figure-6 – Confusion Matrix for SVM

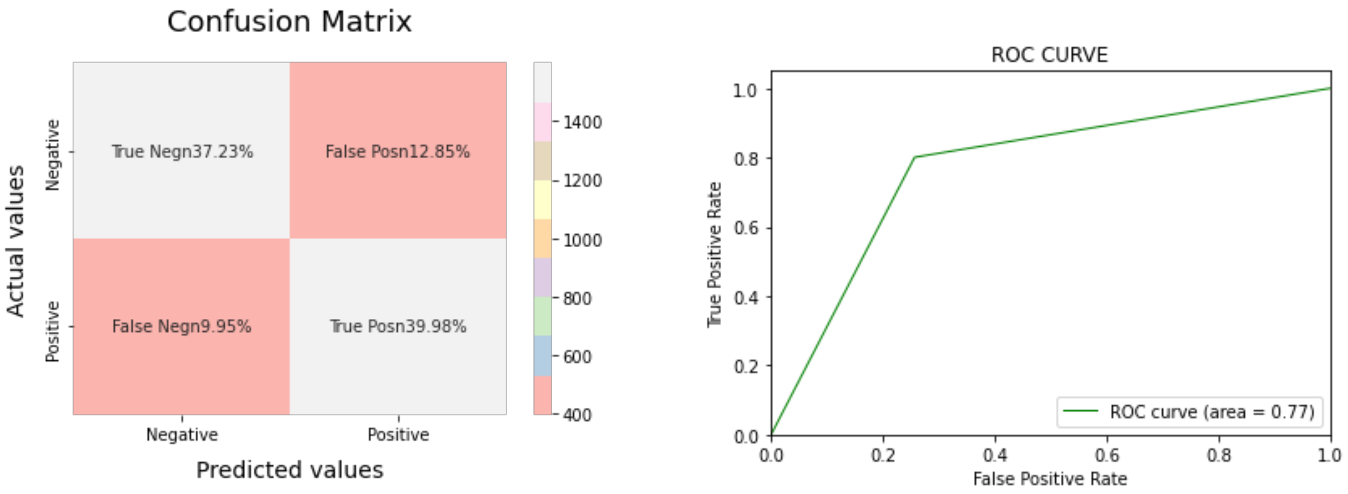


Figure-7 – Confusion Matrix for Logistic Regression

Conclusion

Overall, we found that Logistic Regression is the best model for analyzing Sentiments on the dataset.

Logistic Regression is following the principle of Occam's Razor which defines that for a particular problem statement if the data has no assumption, then the simplest model works the best. Since our dataset does not have any assumptions and Logistic Regression is a simple model, therefore the concept holds true for the above-mentioned dataset.

Questions:

1. What other algorithms can be applied to the model ?
2. What are the challenges in text mining from Twitter ?
3. What are the challenges in data cleansing ?
4. What kind of data needs to be filtered from the text ?
5. Are there any available data sources to mine data for twitter analysis?
6. What is the confidence level in predicting the sentiments ?
7. Which model provides the best prediction ?
8. What ethical considerations were taken care while analyzing the data ?
9. What are the future improvements that you can specify for this analysis?
10. Can this model be able to predict sentiment of any product which is tweeted ?

References

1. Alec Go, Lei Huang, Richa Bhayani, 2009. Twitter Sentiment analysis, s.l.: The Stanford Natural Language Processing Group.
2. https://developer.twitter.com/en/docs/tutorials/how-to-analyze-the-sentiment-of-your-own-tweets
3. https://textblob.readthedocs.io/en/dev/quickstart.html#sentiment-analysis
4. https://towardsdatascience.com/sentiment-analysis-of-tweets-167d040f0583
5. Alexander Pak, Patrick Paroubek, 2010. Twitter as a Corpus for Sentiment Analysis and Opinion Mining, s.l.: LREC.
6. Berry, N., 2010. DataGenetics. [Online]
Available at: http://www.datagenetics.com/blog/october52012/index.html
[Accessed 14 04 2014].

Appendix