Sentiment Analysis Using Twitter Data

Capstone 3 Project

Damilola T. Olaiya

Audience

The potential audience includes:

1. A combination of executive + technical professionals
2. Springboard
3. Potential future employers

Proposal

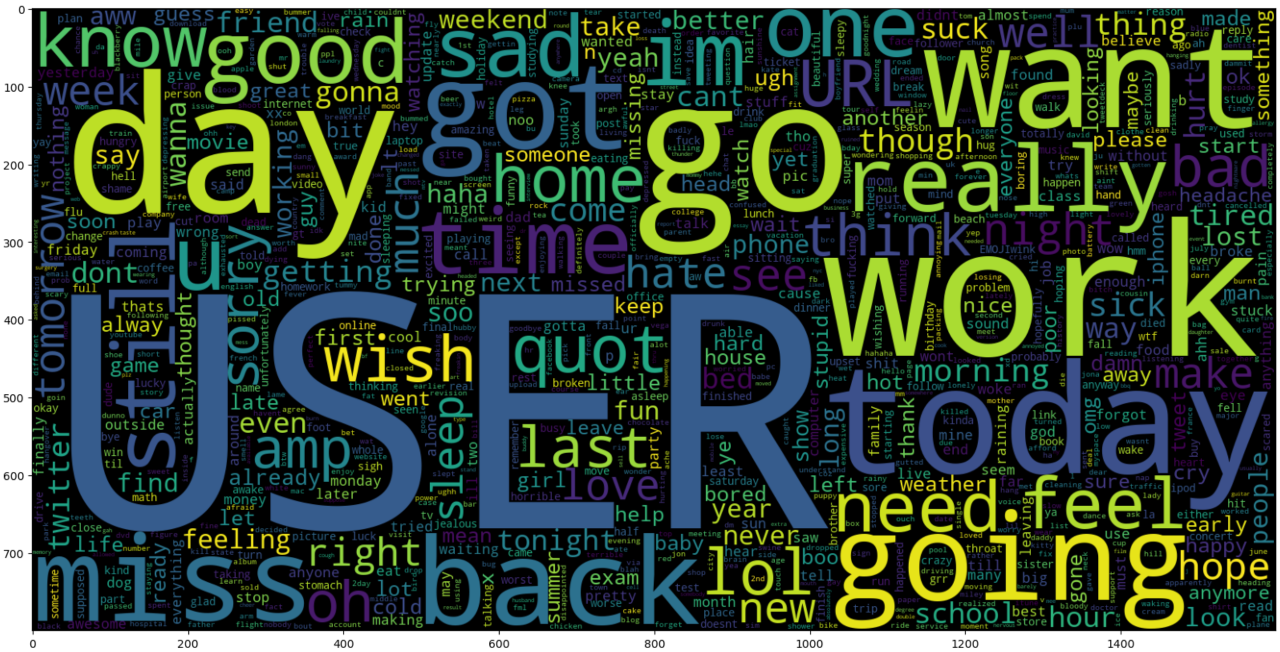
1. **Hypothesis** → How can a given company use tweets about itself and its products or employees to (i) gauge overall sentiment about its products, services, employees or overall branding to make decisions about demand and/or marketing?
2. **Criteria for success** → Creating a model that can accurately predict the [sentiment of tweets about a given entity scraped from Twitter's API.

Data Wrangling

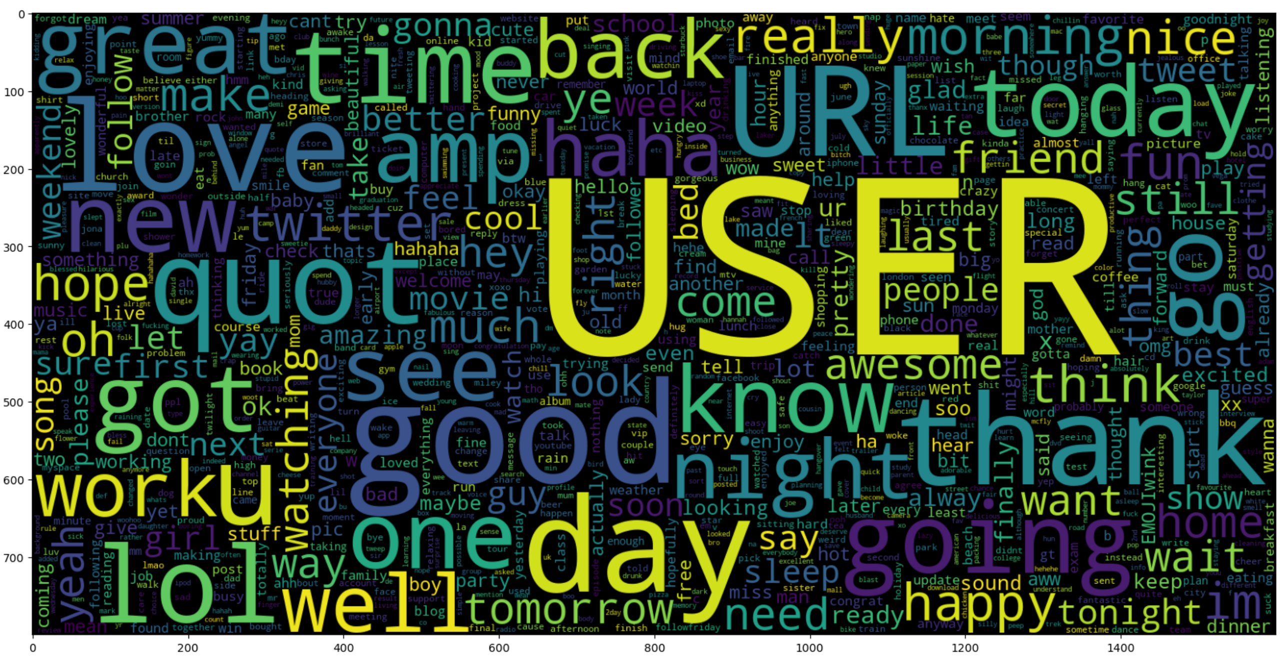
1. The dataset contains 1.6 million tweets and was obtained from here.
2. There were 6 features/columns which were **ids, date, flag, user, text, sentiment**
    a. The target feature is sentiment which was the polarity of the tweet → (0 = negative, 4 = positive)
3. There were no missing values. The data was clean.
4. Uniqueness
    a. The data was checked for uniqueness and no duplicate rows were found.
    b. This suggested that there were no tweets with exactly the same information from exactly the same users on exactly the same dates.
    c. The data was also perfectly split into 2 groups → 800k tweets with a positive sentiment and 800k with a negative sentiment. There was **no skew**.
5. To improve readability, the positive sentiment values were mapped from **4 to 1**.
6. Also, all contractions were "**de-contracted**" eg
    a. n't to not
    b. 'll to will
    c. 'd to would
    d. 's to is
    e. 're to are
    f. 've to have
    g. 'm to am
7. It should be noted that **stop words** were left in the corpus although there is the option of removing them.
    a. Model tuning suggested that accuracy was worsened by the removal of stop words.
    b. This is because stop words include words that indicate negation (e.g. can't, wouldn't, not, don't etc) which strongly affect sentiment.

Exploratory Data Analysis (EDA)

1. A word cloud was generated for tweets with a **negative sentiment**.



2. A word cloud was generated for tweets with a **positive sentiment**.



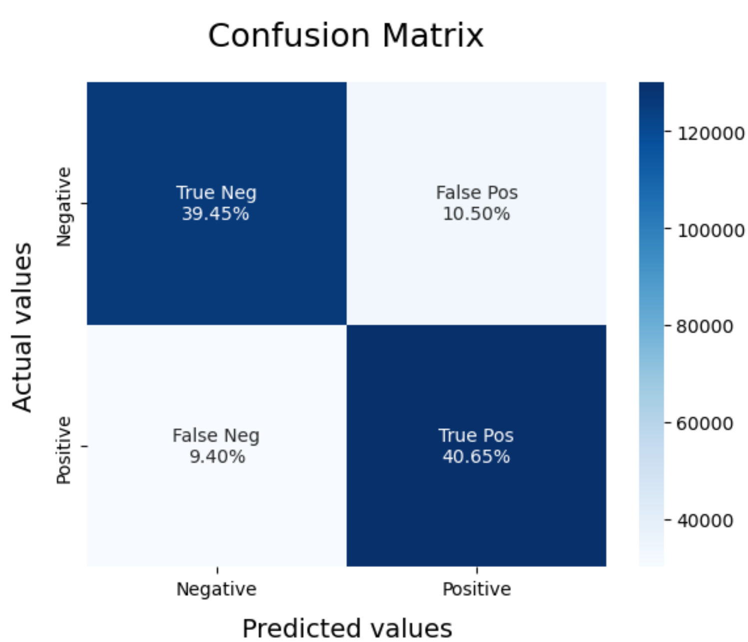
Pre-processing

1. All features besides the **text** and **sentiment** were dropped.
2. The data was split into X_train, X_test, y_train and y_test.
    a. It was then transformed using **Tfidf Vectorization**.
    b. The vectorizer included both **single words** and **bigrams**.
    c. The vectorizer was set to use only the **500K** most populous features/words.
    d. Note that the vectorizer was fit and transformed using X_train only then used to transform X_test. This was done so that the model would be completely unaffected by the testing data.

Modeling

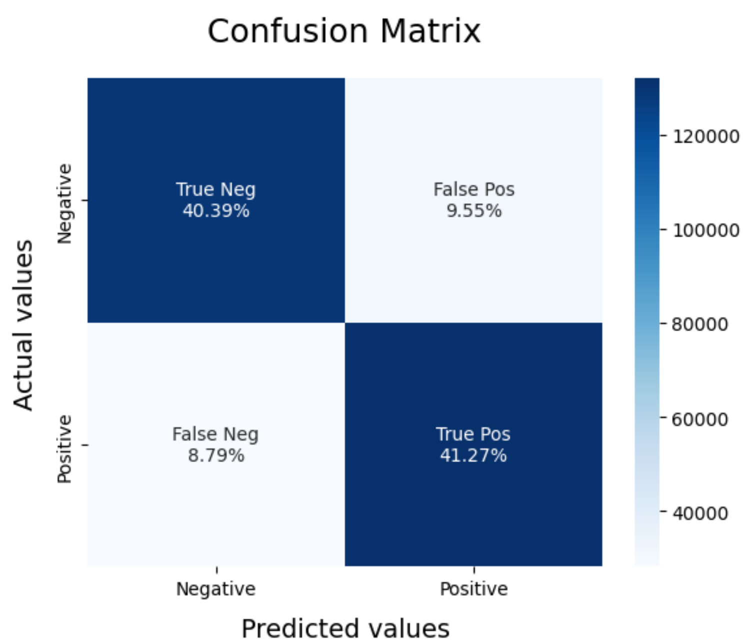
1. The following models were be used:
    a. Bernoulli Naive Bayes (BernoulliNB)
    b. Linear Support Vector Classification (LinearSVC)
    c. Logistic Regression (LR)
2. Since our data was not skewed, **accuracy** was chosen as the **evaluation metric**.
3. **Confusion matrices** and **Classification Reports** were used to get an understanding of how our models were performing in both classes.
4. Results
    a. Bernoulli Naive Bayes model results

```
              precision    recall  f1-score   support

           0       0.81      0.79      0.80    159815
           1       0.79      0.81      0.80    160185

    accuracy                           0.80    320000
   macro avg       0.80      0.80      0.80    320000
weighted avg       0.80      0.80      0.80    320000


 <<< ------------------------------------------------- >>>
```



b. LinearSVC model results

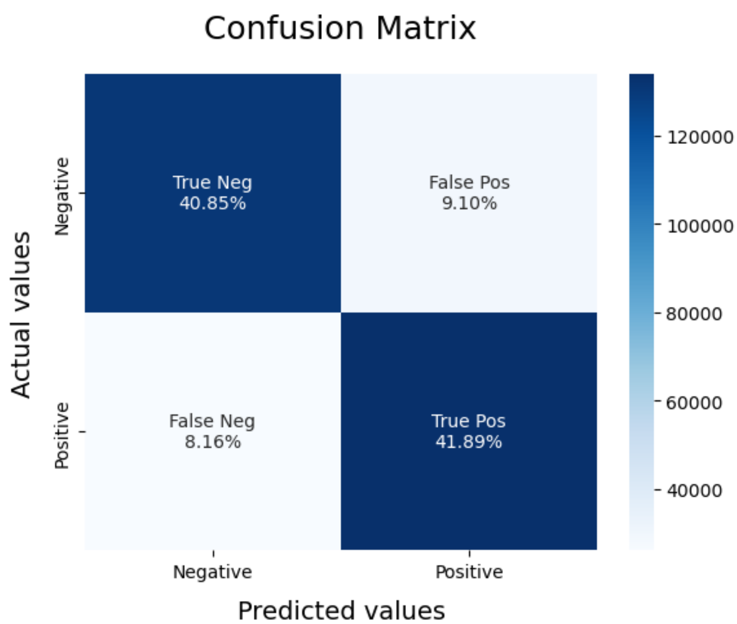
```
              precision    recall  f1-score   support

           0       0.82      0.81      0.81    159815
           1       0.81      0.82      0.82    160185

    accuracy                           0.82    320000
   macro avg       0.82      0.82      0.82    320000
weighted avg       0.82      0.82      0.82    320000


<<< ------------------------------------------------- >>>
```



c. Logistic Regression model results

	precision	recall	f1-score	support
0	0.83	0.82	0.83	159815
1	0.82	0.84	0.83	160185
accuracy			0.83	320000
macro avg	0.83	0.83	0.83	320000
weighted avg	0.83	0.83	0.83	320000

<<< --- >>>



Inference

1. The **Logistic Regression model** performed the best out of all the different models that were tried. It achieved **83% accuracy**.
2. This is followed by the **LinearSVC model** with **82% accuracy** and the **BernoulliNB model** with **80% accuracy**.

Conclusions

1. All three models performed adequately but **Logistic Regression** was the best performer and we will proceed with it.

Further steps to consider

1. Using a larger dataset
2. Using regularization
3. Using cross validation on model parameters
4. Using real time scrapping of Twitter's API and adjusting the model using batch machine learning
5. Using more models
6. Excluding stop words