# Sentiment analysis for products launched using Twitter data

The aim is to use Twitter data to perform sentiment analysis on products that are being launched globally. This would help companies get the initial audience reception and also an insight into the features that are most talked about.

**Acquiring and cleaning data:**

- The source of the data is twitter, specifically the tweets with hashtags #iPhone11 and # GalaxyS10.
- Features contain the time of the tweet, country of origin, tweet text, and retweet count.
- Tweepy library was used to collect the tweets of interest after signing up for a dev account on Twitter. □ The cleaning process involved:

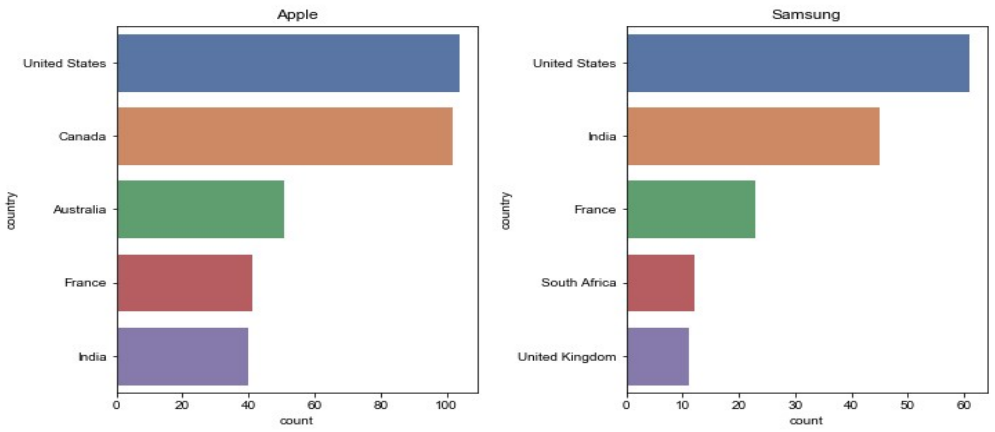
  * Removing special characters from tweet text by using python's regex.

  * With demoji library all emoji that are present in the text were cleaned up.

  * As for country of origin which the location of the tweet, the empty values and arbitrary values like 'earth, somewhere in this planet were replaced' with the value 'unknown'. To clean up other values where only city has been mentioned or location is in native language of that county we use Google Maps API to replace it with the respective country name sin English.

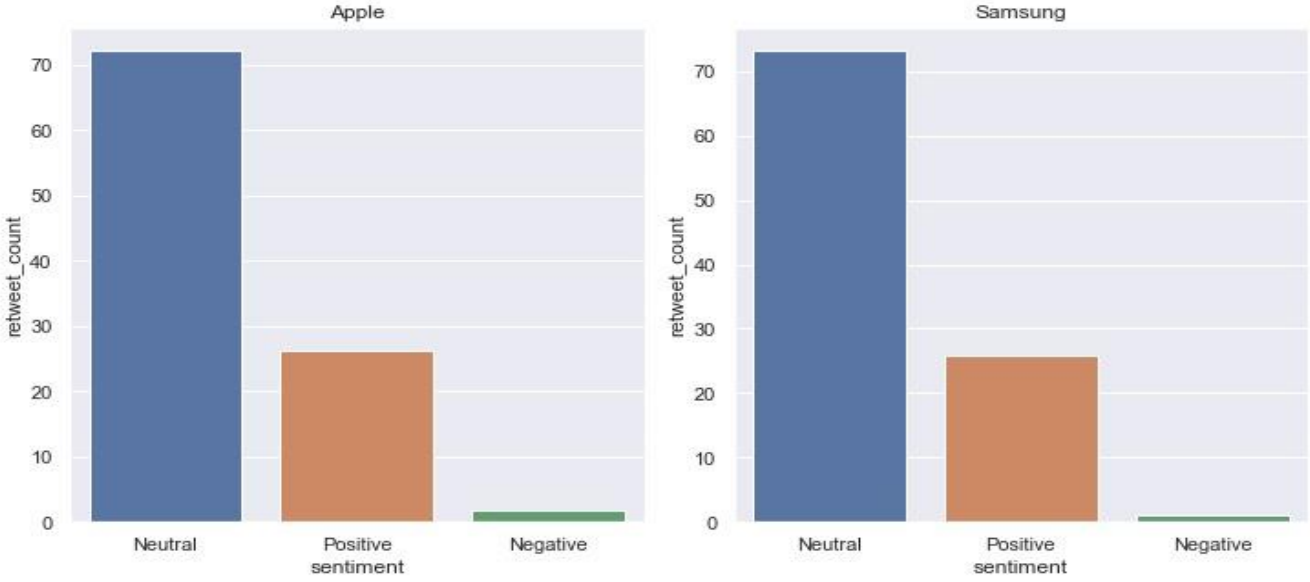
**Analysis and findings from the data:**

- Initial EDA shows the top 5 countries of tweet origin. Interestingly India figures for iPhone as well, a trend that might be attributed to the massive campaign by resellers, mobile carries, banks and Apple themselves before the launch.




- To get an idea about the most talked feature, likes/dislikes about the products we use worldcloud library and nltk library to get a list of stop words. Then, the entire tweet content of apple and Samsung are passed separately. This results in two separate wordclouds as below one each for iPhone11 and GalaxyS10.


- The above iPhone wordcloud shows the often talked points about iPhone11. These include the Pro/Max, Battery, Apple Pay, cases released.
- As for GalaxyS10 the frequent mentions include home screen, gradient body design, One UI, Android OS.
- To perform a basic sentiment analysis Textblob library is used to categorize a tweet as either Positive, Neutral or Negative. Any tweet with polarity value more than +0.3 is tagged as positive, anything less as -0.3 is negative and the rest is termed as neutral, for the threshold that we have set.
- A simple countplot using Seaborn shows that the initial wave of impressions are mostly positive (excluding the neutral)



Building a model required a different approach as the dataset in itself was small. So, we generalize the concept and using Genism's Doc2Vec model.

**Dataset:**

- Sentiment140 dataset from Kaggle is used for this purpose.
- 2 columns – Polarity with values 0 – Negative, 4 – Positive and tweet column containing the text of the tweet.

**Data Wrangling:**

- The same steps as mentioned earlier were followed, using demoji and re libraries to clean the texts.

**Model Building:**

- Genism's Doc2Vec is used to build the model.
- This involves 1st transforming the data into Tagged Documents format that is used in Doc2Vec. We use an inbuilt method for this.
- The entire corpus is split into train and test sets.
- Doc2Vec is initialized and the training set is used, post that we use build_vocab to build a vocabulary from the training set.
- We then convert the training and test data into vectors i.e texts to a matrix of numerical values. Using .infer_vector methodsover the model.
- LogisticRegrssion model from sklearn learn is used to train a model and then using it to predict.

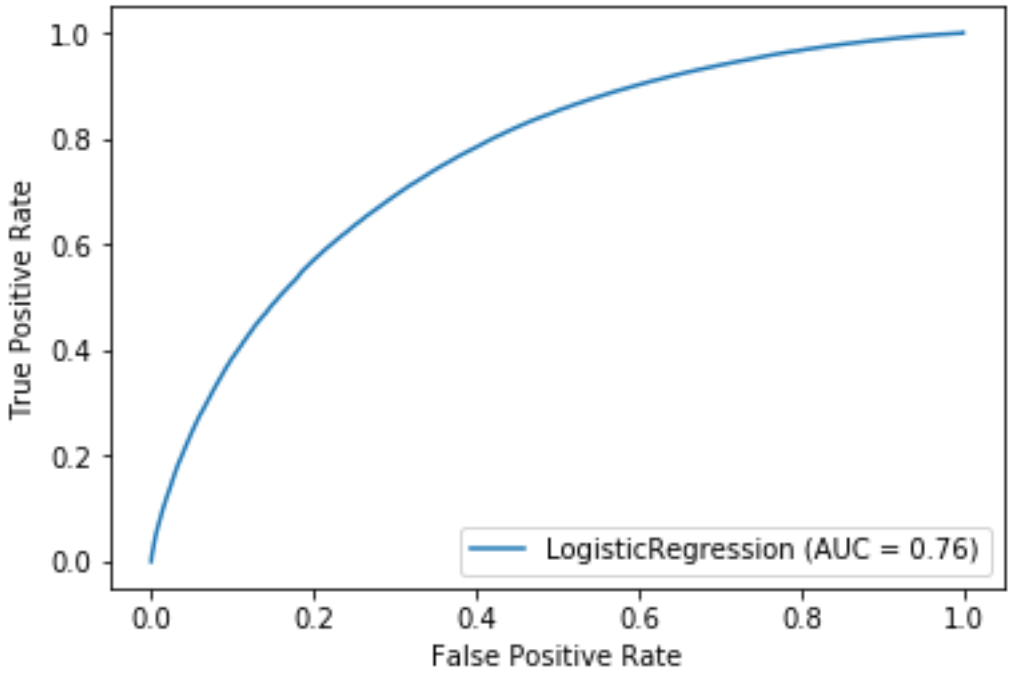
**Results:**

```
LogReg model train accuracy is : 0.741465625
LogReg model test accuracy is : 0.696284375
********************Classification Report**************
****
              precision    recall  f1-score   support

    Negative       0.71      0.67      0.69    160036
    Positive       0.69      0.72      0.70    159964

    accuracy                           0.70    320000
   macro avg       0.70      0.70      0.70    320000
weighted avg       0.70      0.70      0.70    320000
```

**AUC:**




- Using lemmatization actually lead to reduced performance of the model.
- Random forest and Gradient boosting were also tried which either produced similar results or performed less than LogReg.
- With sustained effort and essentially trying other avenues will lead to increased performance.

**Note:**

- Twitter free developer account has limits for the number tweets collected over an interval.
- Sign up to Google APIs premium account (free for 1 year) will be required. Although this contains a monthly usage limit.