Phase-4-Project

1. Overview

For this project, I used NLP algorithms, vectorizers, along with pipelines, cross validation and grid searches to create the most effective model to predict a binary class (positive emotion and negative emotion) pertaining to Apple and Google product sentiment via Twitter data.

- Link to Technical Notebook: https://github.com/AliRampur/Phase-4-Project/blob/main/Phase%204%20Project%20Notebook.ipynb
- Link to final presentation: https://github.com/AliRampur/Phase-4-Project/blob/main/presentation.pdf
- Link to original data sources: https://github.com/AliRampur/Phase-4-Project/tree/main/data

2. Business Problem

Legendary Preds is a consulting firm that works with tech companies all over the world. We have been hired by Google to help analyze customer and non-customer (i.e. Apple customers) sentiment of android and google products vs that of Apple products.

They would like us to help build various predictive models to assess positive and negative emotions against Google's own products (Android) and Apple products. Google will then use the results of our models on a weekly or bi-weekly basis to advertise or offer promotional events at those events or urban centers that have either more significant negative emotions toward Google products or negative emotions toward Apple products, in the hopes of retaining current customers and obtaining new customers from Apple.

3. Exploratory Data Analysis

In this first step, I analyzed tweets and the associated products and company (Apple or Google). Some of the pre-processing steps include:

- Removing null values within the 'tweet' column
- Renaming target column and other columns (features)
- Remove tweets with the indicator of 'i can't tell' within the target column
- Making all tweets lowercase to be able to prepare visuals and remove certain characters and terms
- Creating a binary feature (0 = Negative Emotion and 1 = Positive Emotion) for initial model
- Creating a "company" column to separate products into Google and Apple (for EDA, inferential analysis and potentially, multi class modeling)
- Creating a binary dataframe by dropping tweets with no emotion directed at the brand/product
- Removing hastags and mentions

Target Feature ('y') - Binary (positive or negative)

The target feature or column is the emotion column.

  - Negative Emotion = 570
  - Positive Emotion = 2,978
  - No emotion identified = 5,388

Target Engineering:

After analyzing the various features and data types, I created the target column of company_emotion_class to identify negative emotion toward Apple, negative emotion toward Google, and all others, respectively.

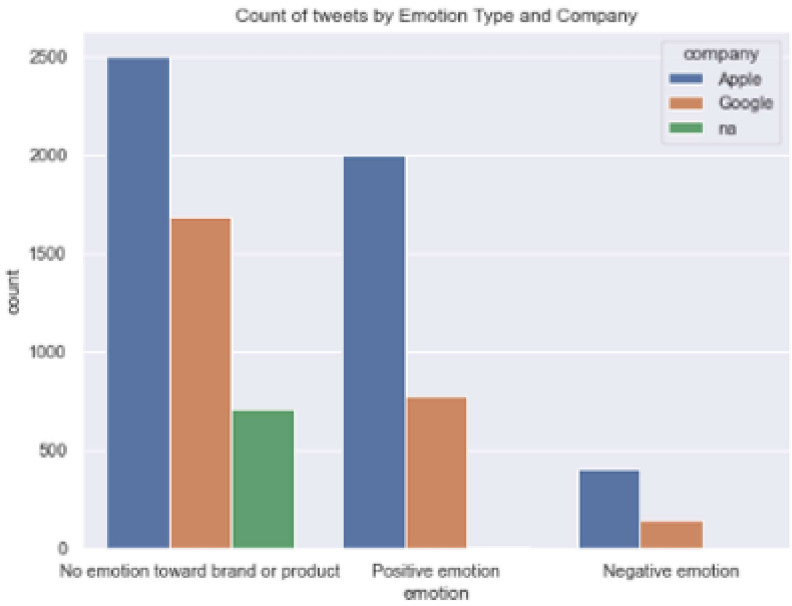
Token Preparation:

Next, I prepared the tweets for tokenization with the following characteristics:

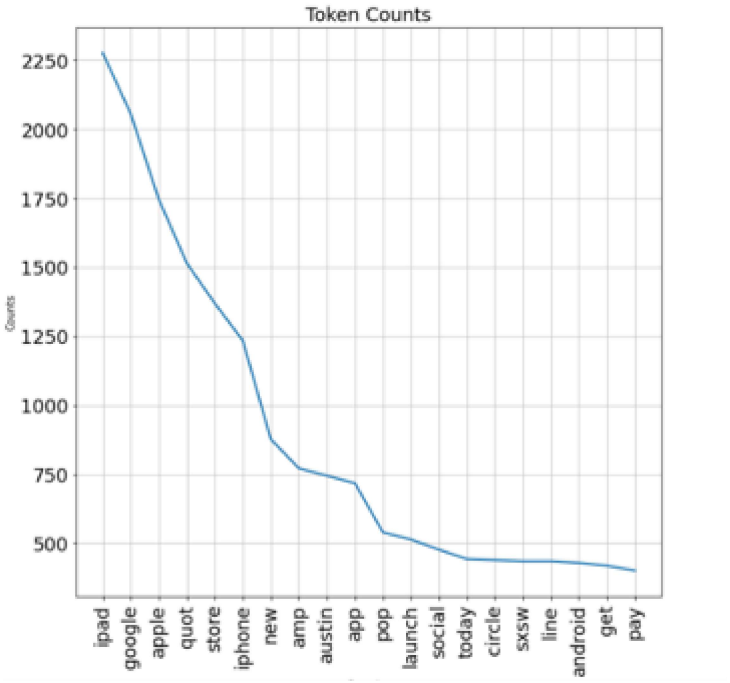
- Apply regex tokenization (r"([a-zA-Z]+(?:'[a-z]+)?)")
- Remove stop words
- Apply lemmatization
- Removing words less than 3 characters

Visualization:

Here is a bar graph of the emotion types by company (Google, Apple, na)



Here is a line graph of the most common tokens:



4. Initial NLP Model - Binary

As part of the modeling process, I first setup the train/test/split using the Tweet column and y feature as positive (1) or negative (0) emotion. I then further broke down the x_train and y_train set into training and validation sets.

I setup a class ModelWithCV() (taken from Flatiron Lecture #51) to help streamline the process of applying cross validation and extracting the results on each pipeline containing a vectorizer (Count to TFIDF) and the given model.

I instantiated a pipeline for each of the following models, beginning with the Dummy Classifier. The other model types I considered in my analysis include:

- Random Forest Classifier
- Decision Tree
- K Nearest Neighbors
- MultiNomialBayes

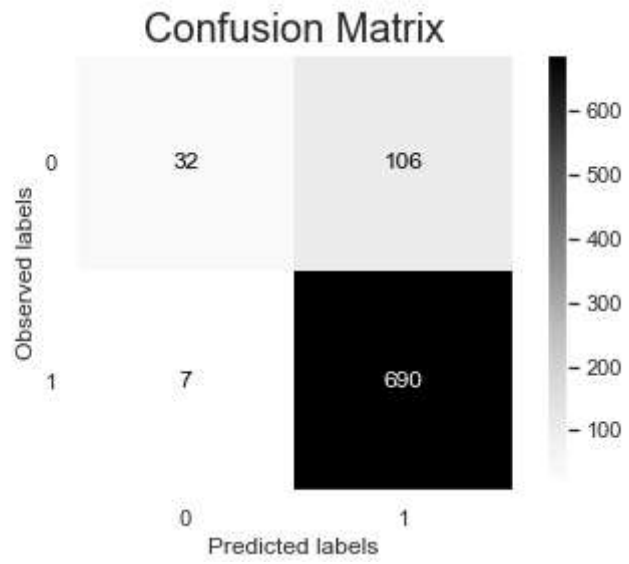
Based on the cross validation and pipeline, the most successful vectorizer / model combos for this particular binary prediction are:

1. tfidf_rf_model
2. cv_rf_model
3. cv_mnb_model

After further consideration of these 3 combinations, I applied a grid search and the best model seems to be cv_mnb_model:

```
- Y - Cross Validation Accuracy Score: .867
- Y - Test Set Accuracy Score: .865
```

Here is the confusion matrix on the test data:



5. NLP Model - Multi Class (Positive, Negative, No Emotion)

I then included the "no emotion" class into the model process, using the same class ModelWithCV and pipelines described above.

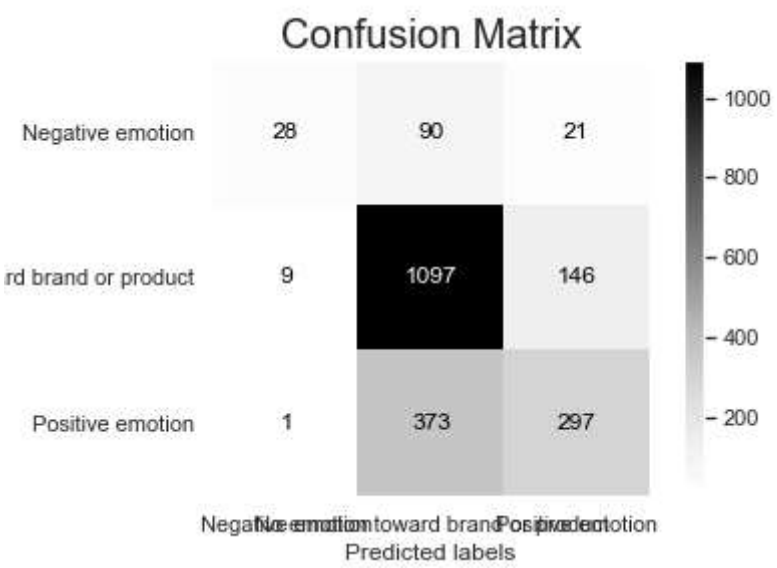
Based on the cross validation and pipeline, the most successful vectorizer / model combos for this particular multi-class prediction were the same:

```
1. tfidf_rf_model
2. cv_rf_model
3. cv_mnb_model
```

After further consideration of these 3 combinations, I applied a grid search and the best model seems to be tfidf_rf model:

```
- Y - Cross Validation Accuracy Score: .666
- Y - Test Set Accuracy Score: .69
```

Here is the confusion matrix on the test data:



6. NLP Model - Multi Class (6 classes: Positive, Negative, No Emotion for Apple and Google, respectively)

Next, I created a new target class into the model process by combining the multi class emotion with the specified company (Google or Apple), and used the same class ModelWithCV and pipelines described above. The target classes include: 1. Negative Emotion Apple 2. Negative Emotion Google 3. All others (positive, no emotion, not Google nor Apple)

Based on the cross validation and pipeline, the most successful vectorizer / model combos for this particular multi-class prediction were the same:

1. tfidf_rf_model
2. cv_rf_model
3. cv_mnb_model

After further consideration of these 3 combinations, I applied a grid search and the best model seems to be cv_rf_model:

- Y - Cross Validation Accuracy Score: .938
- Y - Test Set Accuracy Score: .945

7. Recommendations / Next Steps

Based on the results of the final model, here are my recommendations:

1. At the end of each weekly or two-week period, feed NLP model with Tweet data from multiple regions or events (e.g. #sxsw = South by Southwest)

2. Focus on quantity of negative emotion to Apple or Google products that are identified by the model

3. Setup temporary ‘Google’ branded (Android, etc.) advertisements, ‘give aways’ or other campaigns at the event

Next Steps:

If given additional time...

- Further tune the model to increase the overall accuracy score

- Continue to add tweet data each week and update model

- Consider more advanced NLP models (word embeddings, context, etc.)