phase_4_project

Description

To help Acme Online, an online electronics store, analyze customer tweets from their Twitter page about Apple and Google products. The result of this analysis will be used to find out which company's product has more favourable reviews and the reasons behind it - this will help Acme Online adjust their inventory accordingly.

Methodology

1. Analyze tweets to check what customers are talking about.
2. Analyze tweets to identify the most popular product.
3. For each product, we will look to see what customers like/dislike to identify opportunities for improvement, if applicable.
4. Since human intervention was used identify products based on tweets, we will attempt to build a model using NLP to automate this. We will use the f1-score for model evaluations since minimizing False Positive and False Negatives is desirable .

Dataset

Dataset sourced from CrowdFlower via data.world: https://data.world/crowdflower/brands-and-product-emotions

Analysis

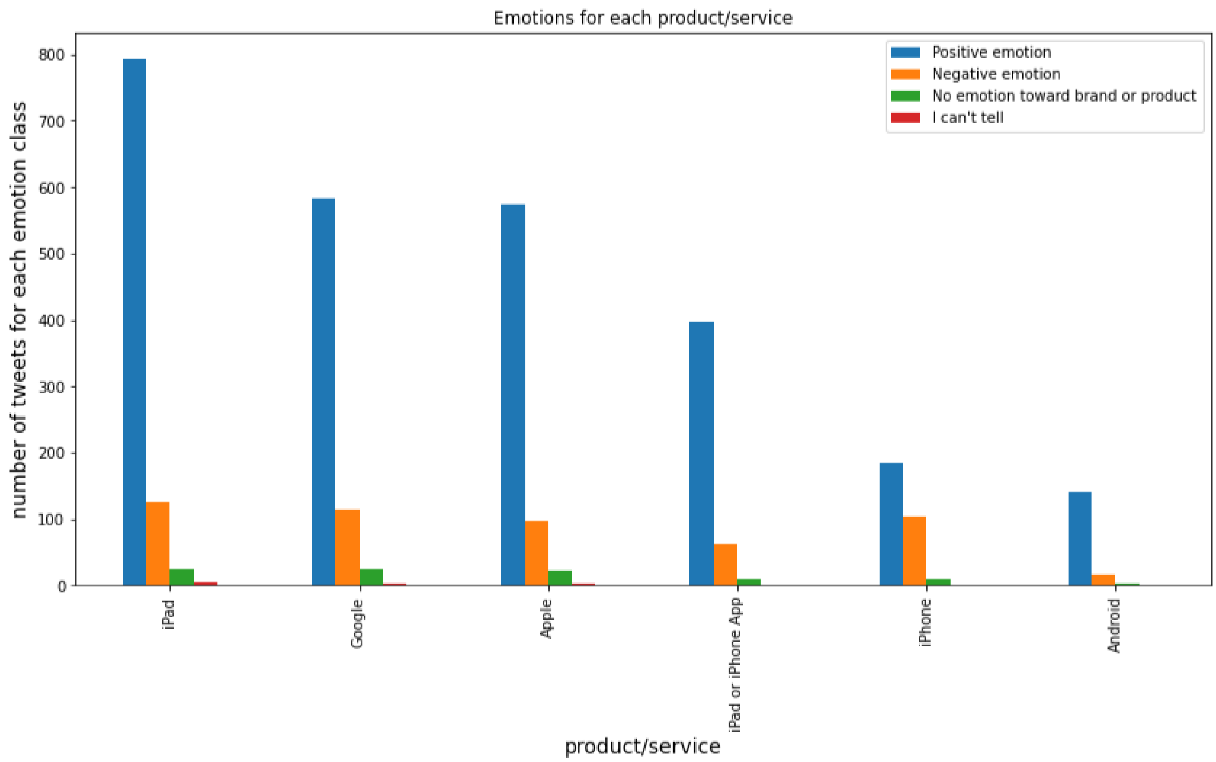
What words are tweeted the most (or) What is the trending topic on twitter?

By analyzing what words are tweeted the most, we get an idea about what customers are talking about. We can visualize this using *Wordcloud*:

We can see from the above that the words `SXSW,Google,iPad` are some of the most tweeted words. A google search of `SXSW` reveals it to be arts and music festival held in Austin,TX. Hence, we can reasonably conclude that tweets collected for the analysis was from the city of Austin,TX and also coincided when the festival was running. It is also quite possible that people were streaming it on their iPads with great success!

What is the most popular product in Acme Online's portfolio?

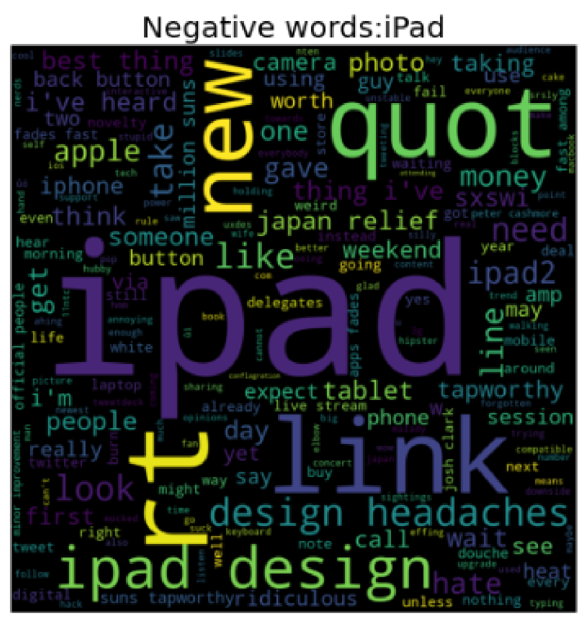
By identifying the most popular product, Acme Online can look for possible opportunites to boost sales and maximize profit:



We have a clear winner: *iPad!*

What do customers like/dislike a product?

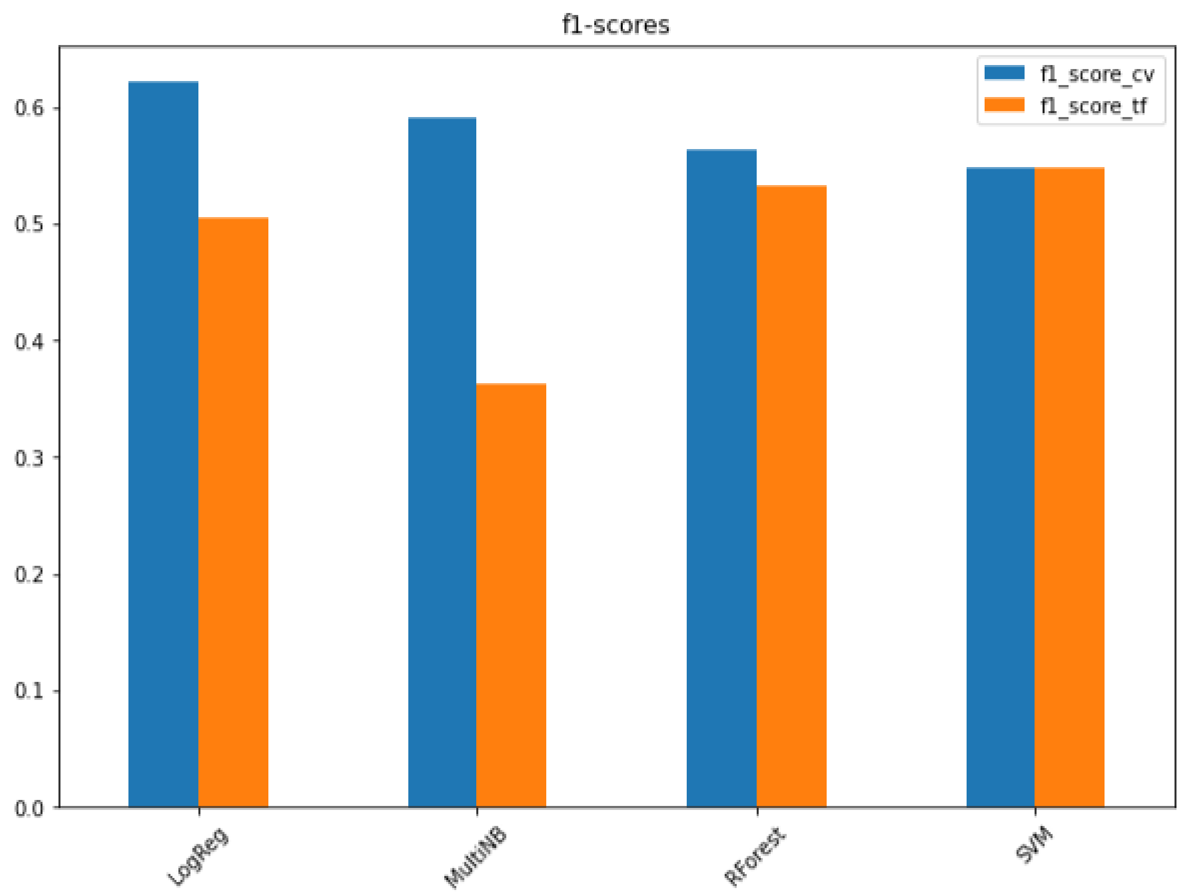
By answering this question, we can identify opportunites for improvement. We can again visualize this using *Wordcloud*. For brevity, only *Ipad* is shown here:



From the negative words, we can see `design headaches, iPad design,money,back button` are some of the words that feature prominently thus illustrating displeasure of the users regarding some of the features of the `iPad` . `iPad2` is also mentioned quite a lot.

Identifying company from tweets

Using NLP, baseline models with different models were built using *CountVectorization* and *Tfidf* vectorizers and f1-scores compared for each:



Since LogisticRegression with CountVectorizer has the highest score, we will try to optimize it for better results

min_df and max_df values

By elimnating words that occur rarely and too often, we can see is model performace improves. By iterating thru a range for each parameter, we can collect scores and check model performance. Ranges set: *min_df* = [1,5]; *max_df* = [1500,1505]

Clearly, we've made the model worse. Our initial f1-score was 0.62 but here we're maxed out at 0.574.

n-grams

The idea behind n-grams is that sometimes word pairings or short phrases are better. For eg: 'black sheep' is more informative than 'black' and 'sheep' seperately. We can set this using the n-gram parameter to (1,2)

	precision	recall	f1-score	support
Not Applicable	0.75	0.86	0.80	1449
Apple	0.67	0.54	0.60	593
Google	0.63	0.35	0.45	231
accuracy			0.73	2273
macro avg	0.68	0.59	0.62	2273
weighted avg	0.72	0.73	0.71	2273

f1-score remains unchanged from our earlier peak of 0.62

Stemming using PorterStemmer

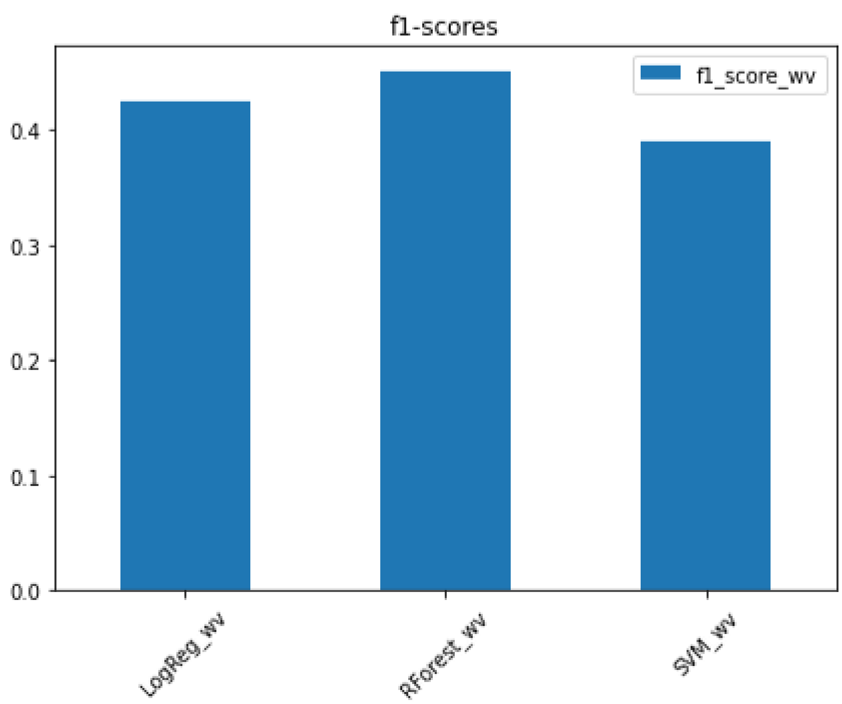
With stemming, we use the use root of the word. For eg: ran,runs,running all stem from the word run. This way we reduce the number of features and can improve accuracy of the model.

	precision	recall	f1-score	support
Not Applicable	0.75	0.85	0.80	1449
Apple	0.66	0.55	0.60	593
Google	0.61	0.37	0.46	231
accuracy			0.72	2273
macro avg	0.67	0.59	0.62	2273
weighted avg	0.71	0.72	0.71	2273

Again, model performmace stagnates at 0.62.

Word Embedding - Word2Vec

Word Embeddings are a type of vectorization strategy that computes word vectors from a text corpus by training a neural network, which results in a high-dimensional embedding space, where each word in the corpus is a unique vector in that space. Here, we will import the Word2vec vector from the open source *gensim* library and use the *skip gram* architecture for modelling.



Model performace has only worsened with this strategy

Conclusions

Recommendations

Since the iPad is the most popular product,Acme Online could look for opportunities to boost sales. Acme Online could also maybe expand their portfolio buy offering tablets from other manufacturers to see if they will bring in more revenue.

Modelling

1. Part-of-Speech tagging can be used to create more features.
2. Ensemble methods like XGBoost and Adaboost can also be trialed for modelling along with other word embedding techniques like fastText and Glove.

Limitations

More varied data is desirable. Current data is very imbalanced and localized to a time and place thus possibly skewing forecasts for the future. As next steps, that would be the starting point. We can then use it to improve model efficiency.

More Information

- Notebook
- Presentation

Repository Structure

```
├── README.md
├── notebook.pdf
├── presentation.pdf
├── project.ipynb
└── repo.pdf
```

Releases

No releases published
Create a new release

Packages

No packages published
Publish your first package

Languages

- **Jupyter Notebook** 100.0%