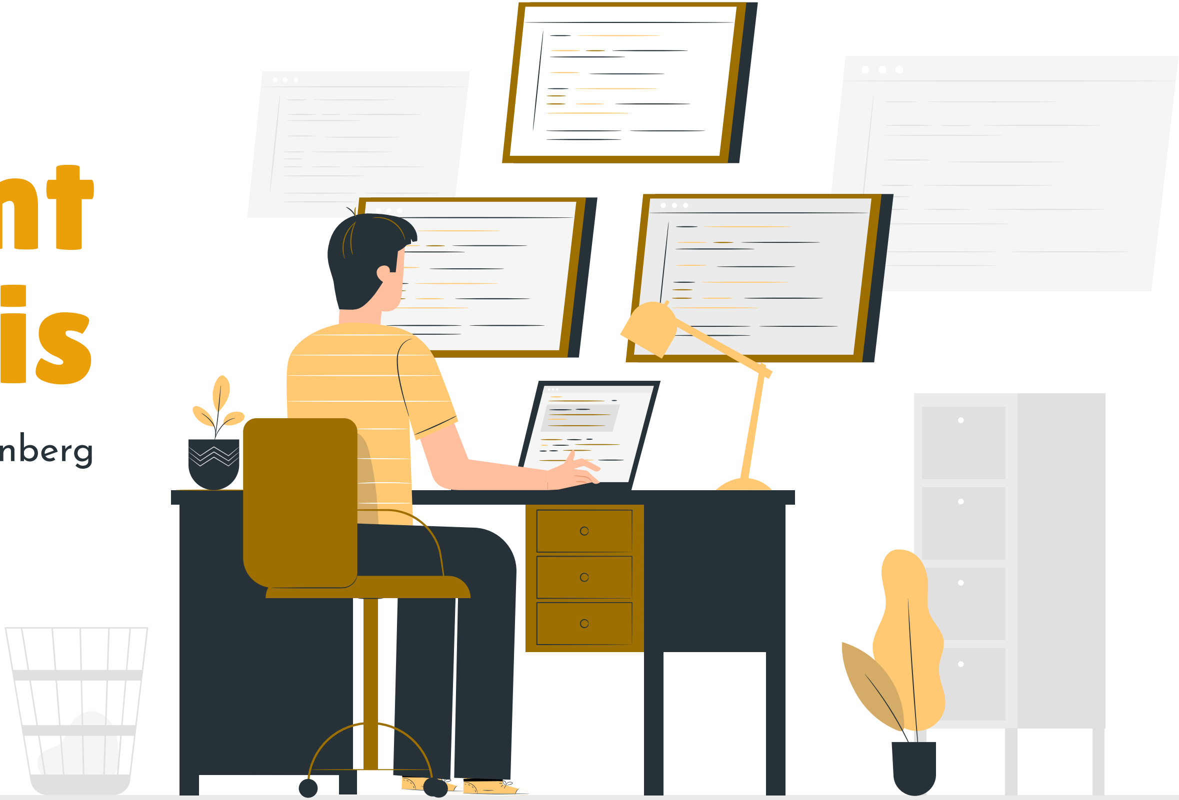
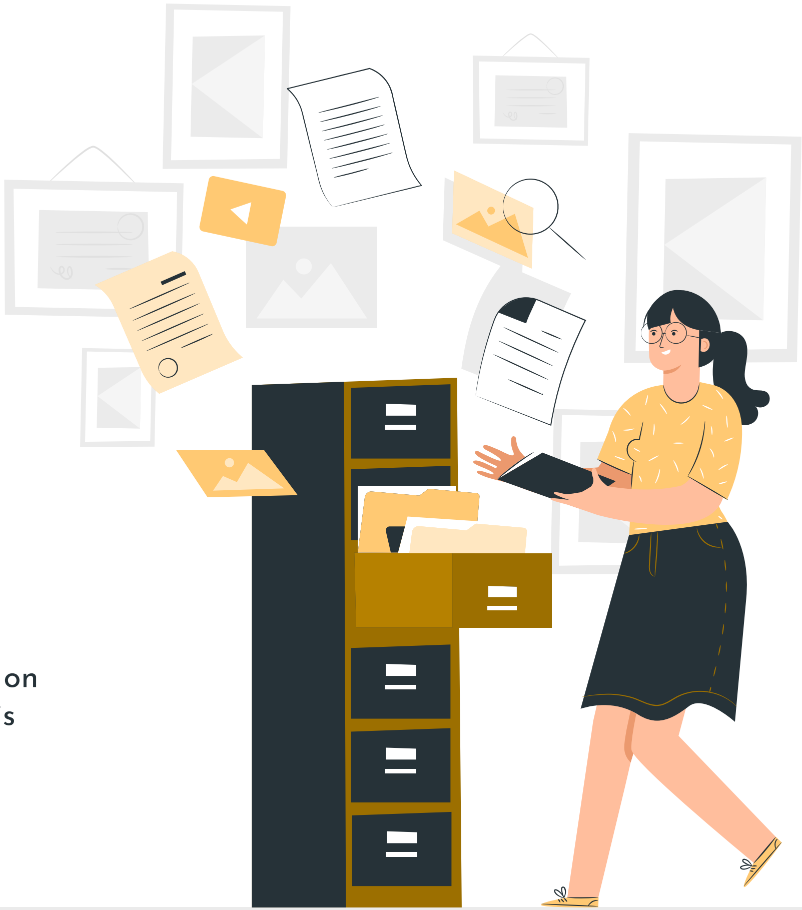
Sentiment Analysis

By Sam Stoltenberg



ABOUT THE DATA

The data used was from data.world, created by crowdflower. It is a crowdsourced dataset based on the sentiment of a tweet, and what the emotion is directed at.



Questions:

01 Which model will perform best?

02 Which aspects of a phone should the company improve

03 What can androids or iphones improve in their products to reduce negative feedback

OSEMN PROCESS

1. Obtain

Obtain the data by either scraping it, pulling from an API or downloading.

2. Scrub

Scrub the data, and transform it to a usable format.

3.Explore

Search for null values, and ways you can interpret the data.

4.Model

Build a model that can predict and interpret unseen data.

5. Interpret

Put the model to use on new data.

Scrubbing the Data:

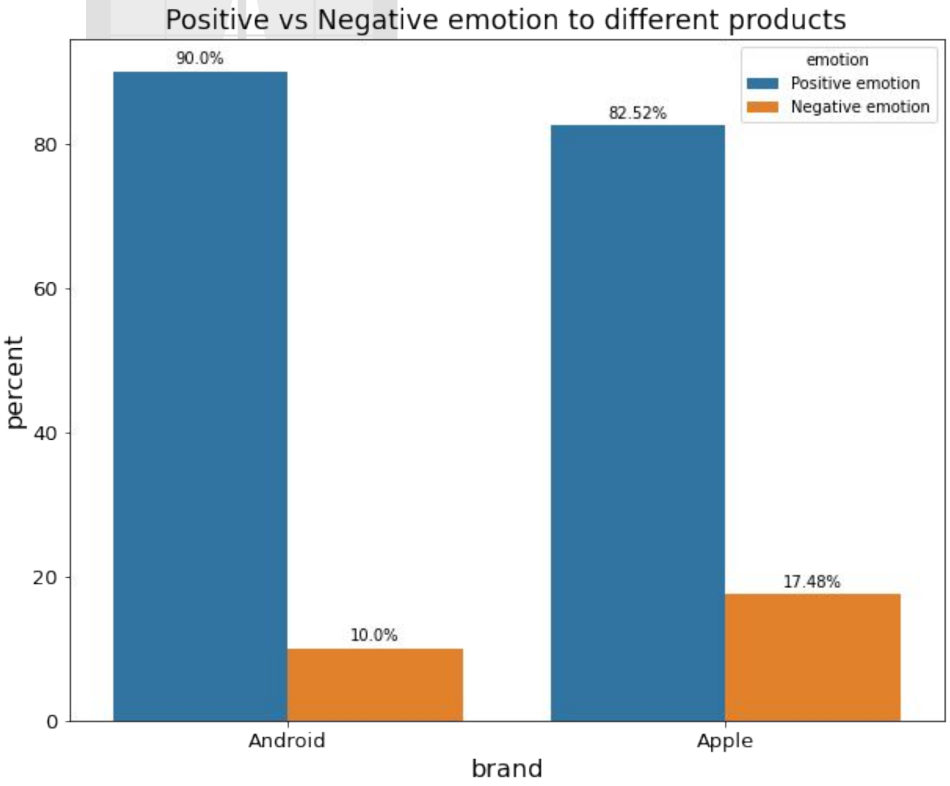
- There were some null values.
- Some yes/nos needed to be encoded to 0/1
- Imbalanced classes needed to be fixed as only 16% of sentiment was negative.
- The text needed to be vectorized i.e

[text, needed, vectorized, i.e]

[4, 18, 32, 13]

Explore

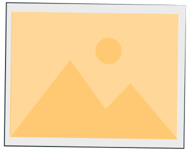
Overall Apple has more negative emotion, although this is skewed by there being 1600+ tweets directed at apple, and only 80 directed at android.



Positive and Negative Word Clouds

Positive

Negative



From this we have deciphered words that mostly show up in the negative or positive clouds.

'fun' is 96% positive
'music' is 97% positive
'autocorrect' is 100% negative

The phone companies should focus on music quality, the quality of their autocorrect, and making the phones more fun.

Models Used

- True negatives are how often the model successfully predicts negative sentiment.
- True positives are how often the model successfully predicts positive sentiment.
- Accuracy is how well a model performed on test data.

01 Support Vector Machine

- 66% True Negatives
- 86% True Positives
- 83% Accuracy

02 Stacking Classifier

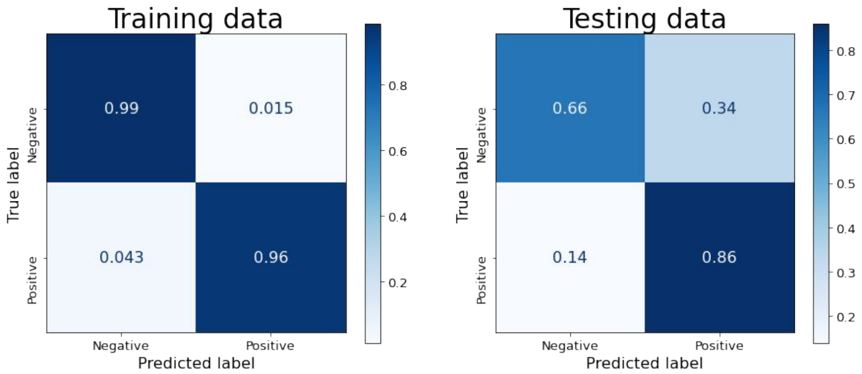
- 55% True Negatives
- 91% True Positives
- 85% Accuracy

03 Voting Classifier

- 40% True Negatives
- 99% True Positives
- 90% Accuracy

We will be focusing on the Support Vector Machine as it has the most balanced TN/TP. The Voting Classifier does better overall, but is only predicting 40% of the true negatives.

Support Vector Machine



Similar Words

```wv.most_similar()```

Below are some selected words from running equations on words i.e (1+2=3)

Iphone minus Android:

- interface, easier

It seems that Androids should focus on their interface, and keeping things easier.

Android minus Iphone:

- contest, events

It seems that Iphones may want to focus on events and contests.

Conclusion

Best Performing Model

Support Vector Machine for its' more balanced recall.

How Accurate?

The phone companies should focus on music quality, the quality of their autocorrect, and making the phones more fun.

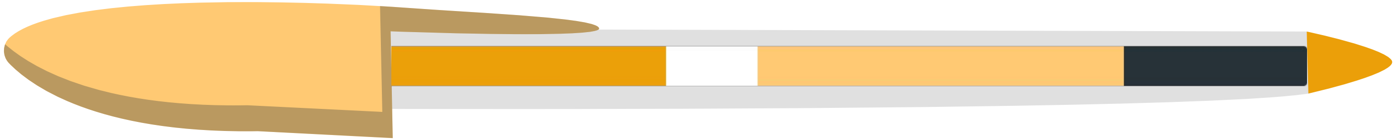
Company Focus?

Iphone

- Contests
- Events

Android

- Interface
- Ease of use



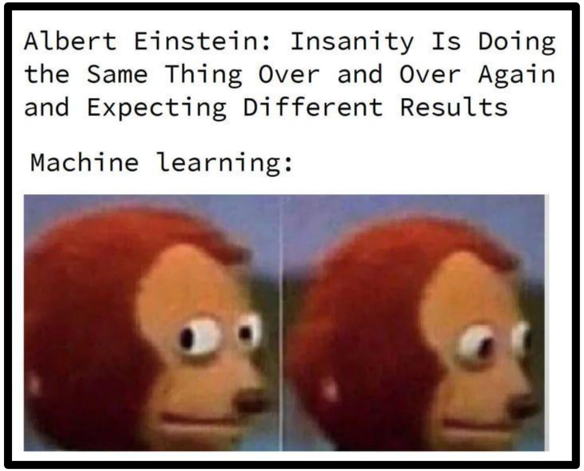
Future Work

- Build an sklearn pipeline and grid search with tokenizer and vectorizer parameters along with classifier parameters.
- Build neural networks for the data, and use the Oscar API for parameter tuning.
- More tuning on the noted models.

THANKS

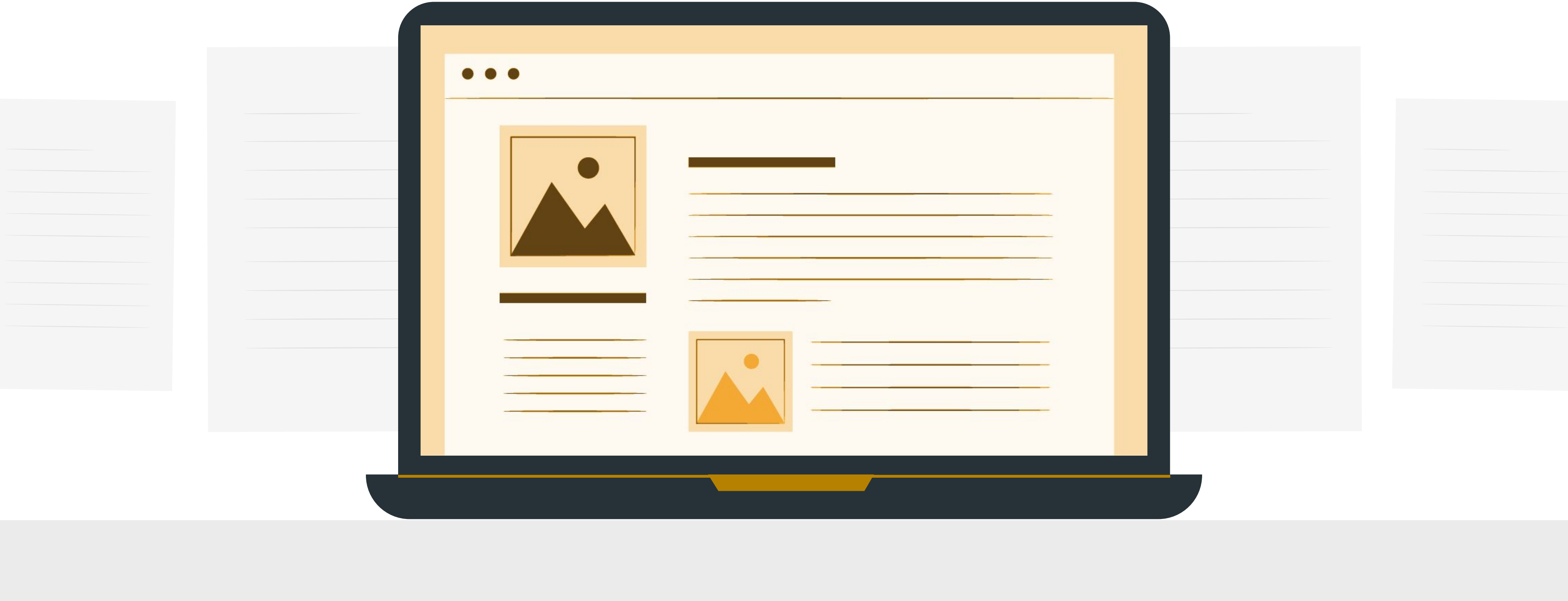
Do you have any questions?

sam@skelouse.com
214-724-6699
skelouse.com



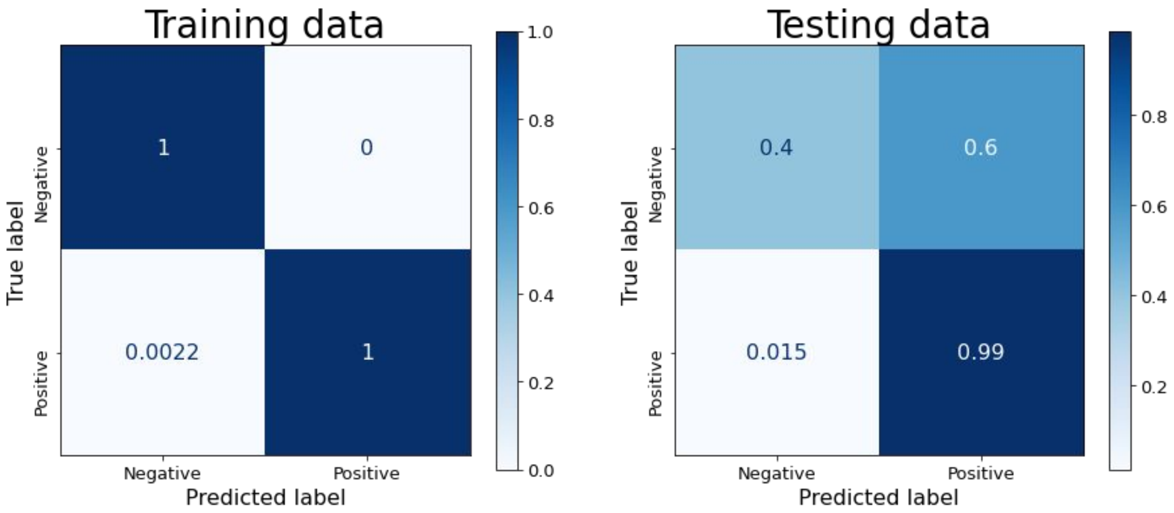
Appendix

Insert your multimedia content here. You can replace the image in the screen with your own work. Just delete this one and add yours



Voting Classifier Confusion Matrices

The Voting classifier has an overall Testing accuracy of 90%, but as you can see on the right matrix 40% true negatives are predicted.



Stacking Classifier Confusion Matrices

The Stacking classifier has an overall Testing accuracy of 85%, but as you can see on the right matrix 55% true negatives are predicted.

Word Analysis

(Iphone – Android)

```
1              0.309063
stock          0.265474
able           0.256434
winner         0.252086
barry          0.238984
going          0.236802
content        0.235509
cbatsxsw       0.231650
entire         0.230150
interface      0.230088
miss           0.226862
make           0.224043
hotel          0.218620
must           0.217708
etchasketch    0.214270
original       0.210932
heck           0.210874
almost         0.207319
easier         0.207113
someone        0.206420
dtype: float64
```

(Android – Iphone)

```
xoom           0.300534
working        0.258659
done           0.255140
hootsuite      0.246845
rocks          0.242207
events         0.230156
room           0.227836
woot           0.224581
contest        0.221765
part           0.217028
blocks         0.212759
choice         0.211492
fwd            0.211405
end            0.210556
featured       0.200980
ps             0.198907
walked         0.198528
call           0.197890
blogger        0.195487
excited        0.192184
dtype: float64
```