Airline Tweets Sentiment Analysis

General Assembly

Data Science

by Michael Lin

Sentiment Analysis

AGENDA

01

DATA SETS

What were the data collection process.
When, where, and how?

02

PRE-PROCESSING

Natural language data pre-processing steps

03

RULE-BASED CLASSIFICATIONS

Without using machine learning, can we use some basic rules to classify sentiments?

04

MACHINE LEARNING BINARY CLASSIFICATIONS

Using machine learning algorithms, we attempt to classify positive and negative sentiments

05

MACHINE LEARNING MULTI-CLASS CLASSIFICATIONS

Using machine learning to predict positive, neutral and negative sentiments and determine the best model for data with unknown sentiments

06

REASULTS

The summary and statistics of the collected airline tweets

07

CLOSING THOUGHTS

FEEDBACK | Q&A

DATA SETS

Where, when and how?

Sentiment Analysis

FOUR MAJOR AIRLINES

@AmericanAir

AA

#americanairlines

#americanair

@delta

DELTA

#deltaairlines

#deltaair

@southwestair

SOUTHWEST

#southwestairlines

#southwestair

@united

UNITED

#unitedairlines

#unitedair

Sentiment Analysis

SOURCES

80,121 Tweets

TWITTER API

Using Python and **twython** to retrieve tweets through Twitter's API during 7 days period.

14,640 Tweets

KAGGLE

Reformatted/cleaned tweets with graded sentiment of Major Airlines from Feb 2015

4,000 Tweets

Newsroom

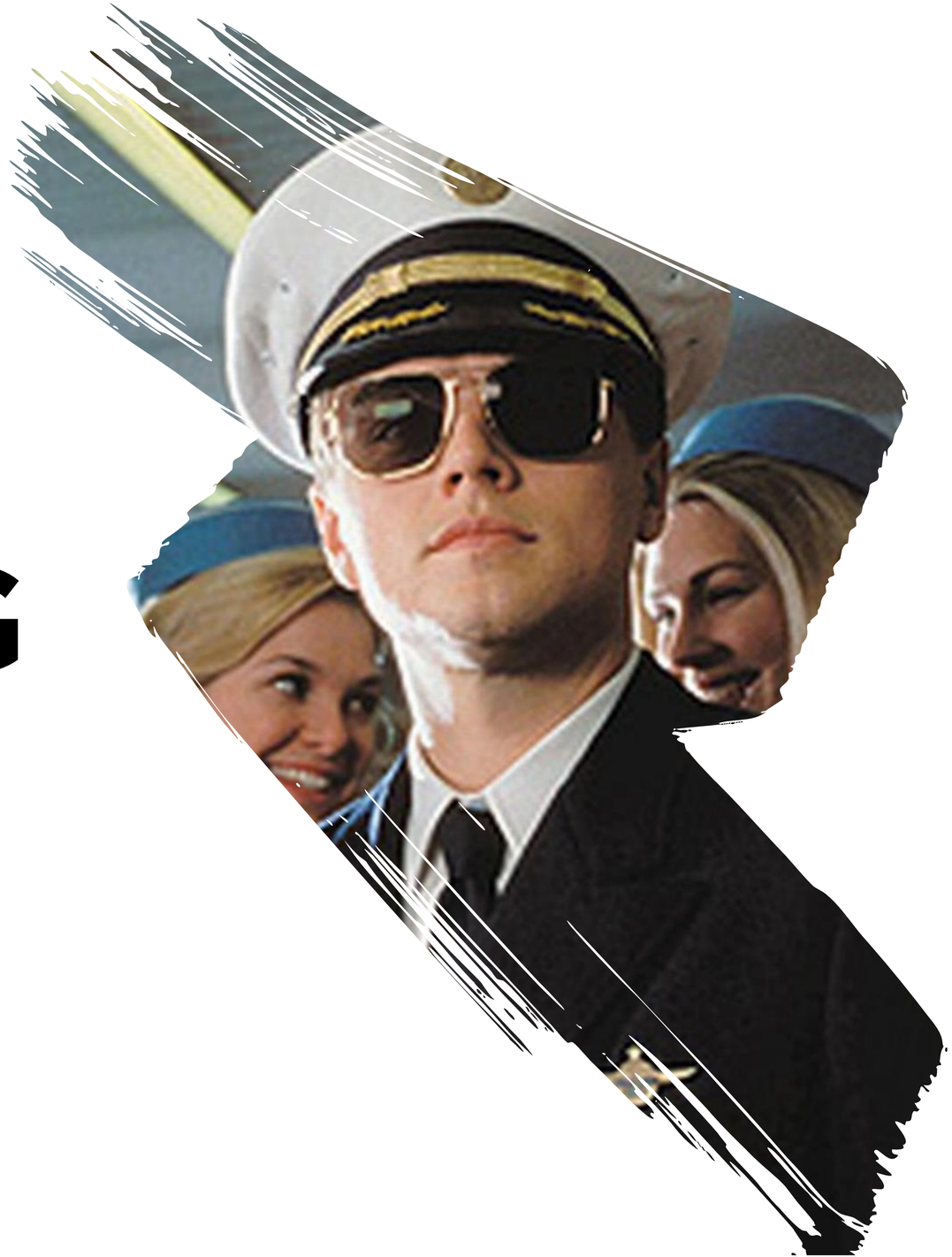
Commercial datasets provided by Newsroom with machine graded tweets

itsvatime.com

600 + 600 + 600 + 600

DATA PRE-PROCESSING

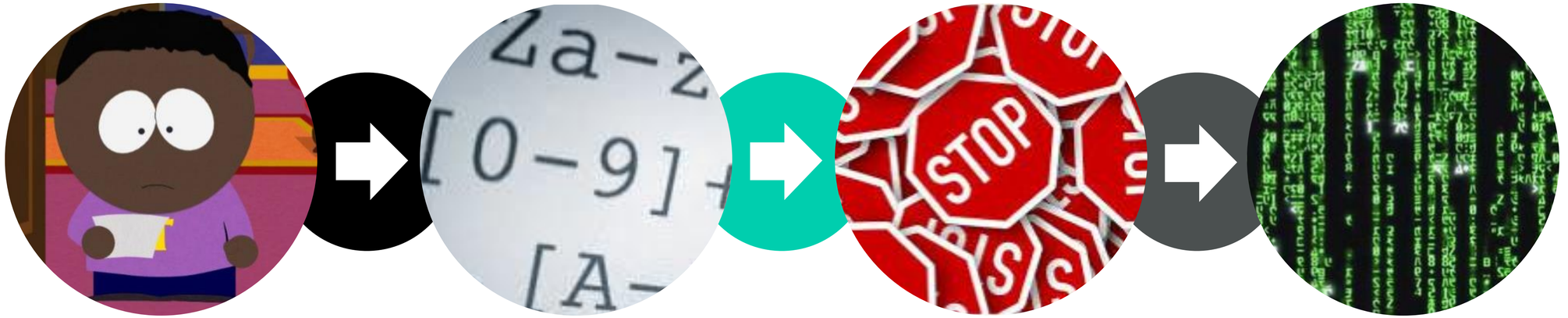
Natural language processes to prepare and transform the tweet content for various classification models and sentiment analysis



Sentiment Analysis

NATURAL LANGUGE PROCESSING

Major processing steps



1 Tokenization

Tokenize all tweet contents

2 RegEx

Perform regular expression

3 Stop Words

Remove all the English stop words

4 Vectorizer

For machine learning

Vector → Matrix… get it?

Sentiment Analysis

REGULAR EXPRESSION

tweet.lower()

Lower Cases

Turn all letters to lower cases. One option is to retain original upper cases

re.sub('((www\.[^\s]+)|(https?://[^\s]+))','URL',tweet)

Eliminate URLs

Turn all https://... and www… format to display just the word ‘URL’

re.sub(r'@([^\s]+)', r'\1', tweet)

Replace @username

Convert all @username to display just ‘username’ without the ‘@’ sign

re.sub(r'#([^\s]+)', r'\1', tweet)

Replace #hastags

Convert all #hashtagwords to just ‘hashtagwords’

re.sub(r'&', r' and', tweet)

Replace ‘&amp’ Signs

Replacing all &amp symbols with the actual words ‘and’ with a space

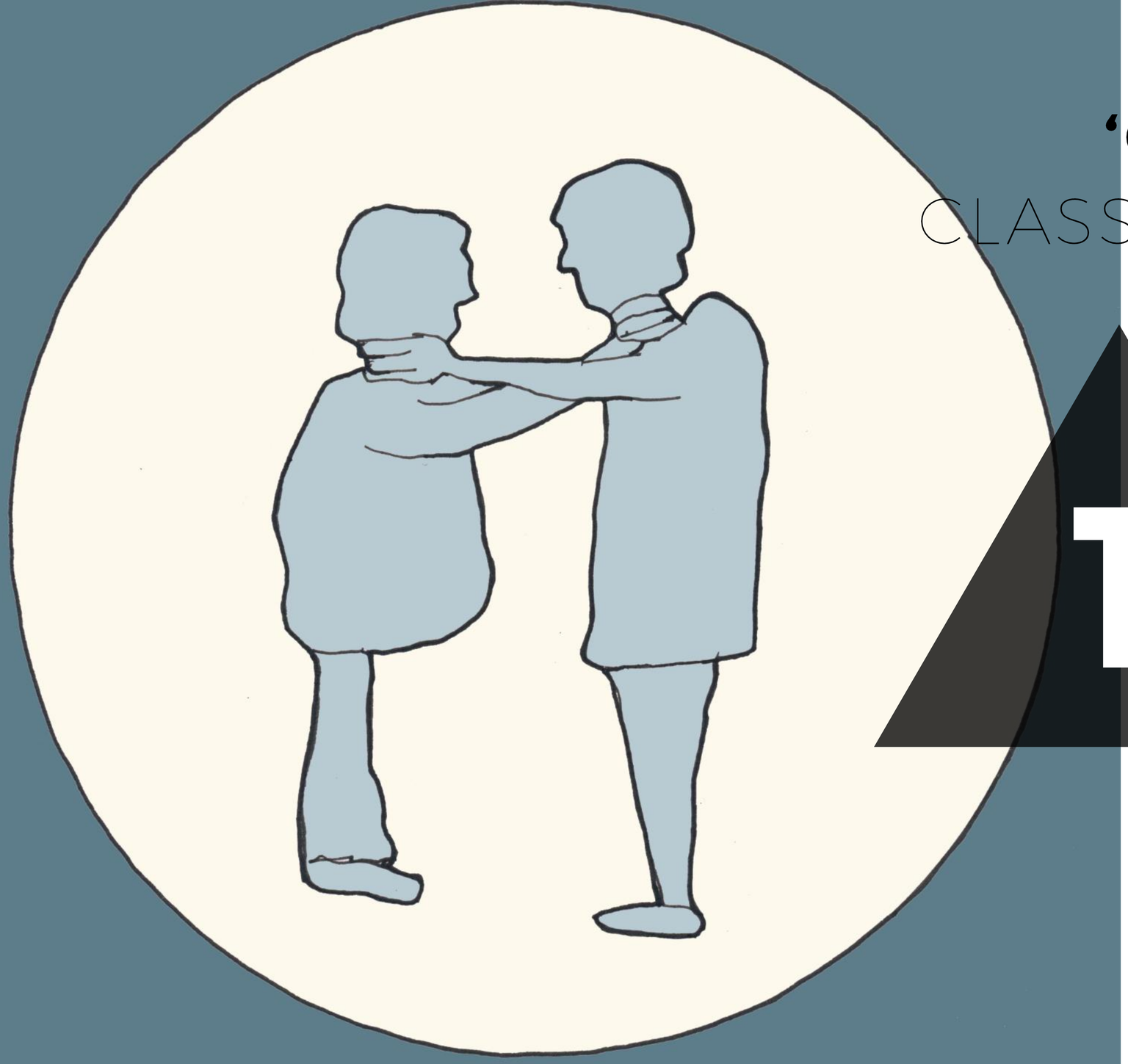
re.sub('[\s]+', ' ', tweet)

Remove Additional Spaces

Remove all annoying white spaces

RULE BASED
CLASSIFICATION

First we will use two simple rule-based techniques to conduct our sentiment analysis to see what happens



‘COMMON SENSE’

CLASSIFICATION RULES

1

- **(pos) x (pos) = (pos)**

This movie is very good

- **(pos) x (neg) = (neg)**

This movie is not good at all

- **(neg) x (neg) = (pos)**

This movie was not bad

Require a Pre-defined List of Positive & Negative Words ←

Accuracy

MANUAL LABEL (2,400)

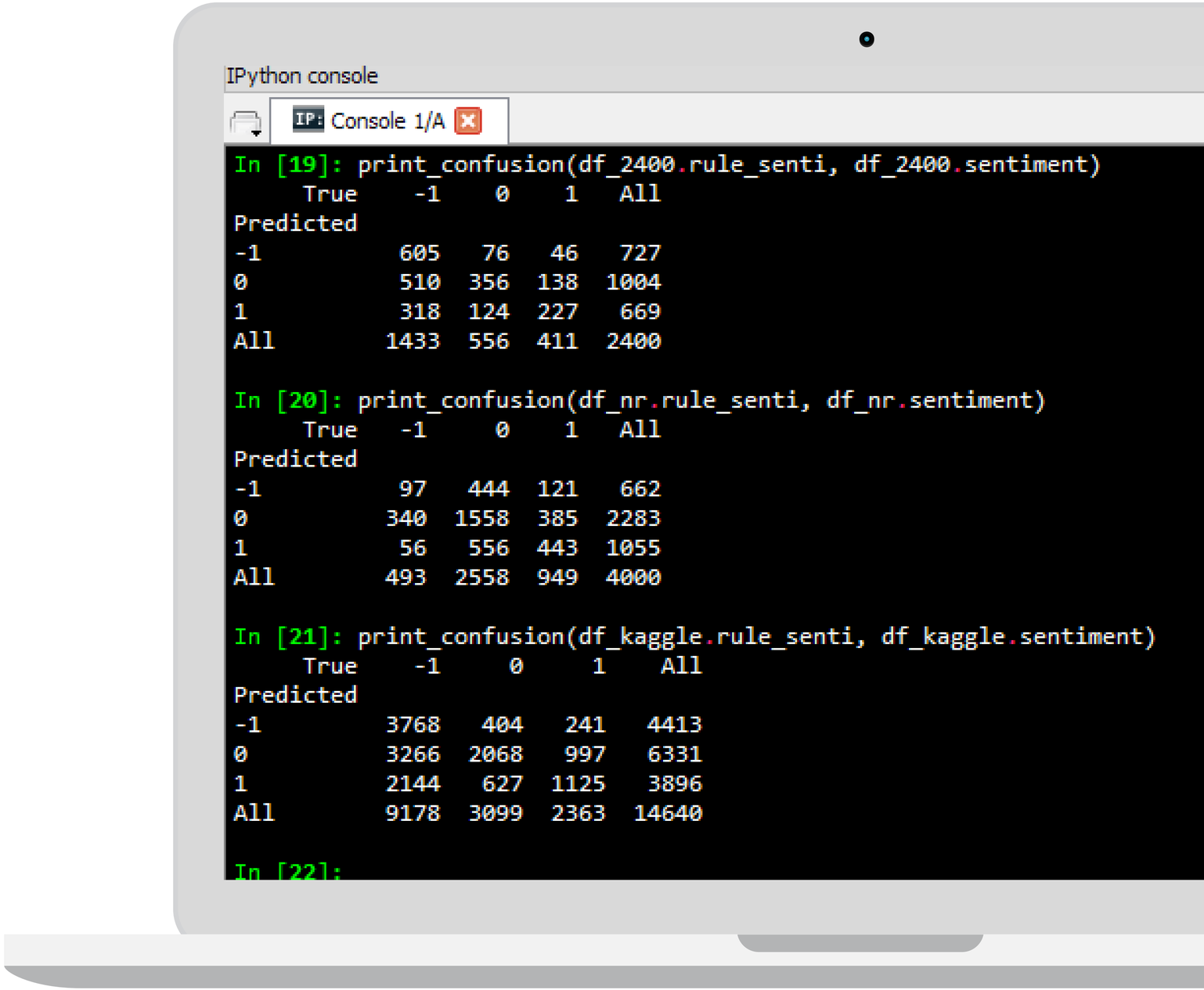
49.50 %

NEWSROOM (4,000)

52.45 %

KAGGLE (14,600)

47.53 %



V.A.D.E.R.
SENTIMENT SCORES

2

VADER is a lexicon and rule-based sentiment analysis tool that is especially attuned to sentiments expressed in social media.

→ Valence Aware Dictionary and sEntiment Reasoner

Examples

“VADER is smart, handsome and funny.”

{‘neg’: 0.0, ‘neu’: 0.254, ‘pos’: 0.746, **‘compound’: 0.8316**}

“VADER is very smart, handsome, and funny!”

{‘neg’: 0.0, ‘neu’: 0.293, ‘pos’: 0.707, **‘compound’: 0.8646**}

“VADER is VERY SMART, handsome, and FUNNY!!!”

{‘neg’: 0.0, ‘neu’: 0.233, ‘pos’: 0.767, **‘compound’: 0.9342**}

“Today SUX!”

{‘neg’: 0.779, ‘neu’: 0.221, ‘pos’: 0.0, **‘compound’: -0.5461**}

“Today kinda sux! But I’ll get by, lol :)”

{‘neg’: 0.154, ‘neu’: 0.42, ‘pos’: 0.426, **‘compound’: 0.5974**}

“At least it isn’t a horrible book.”

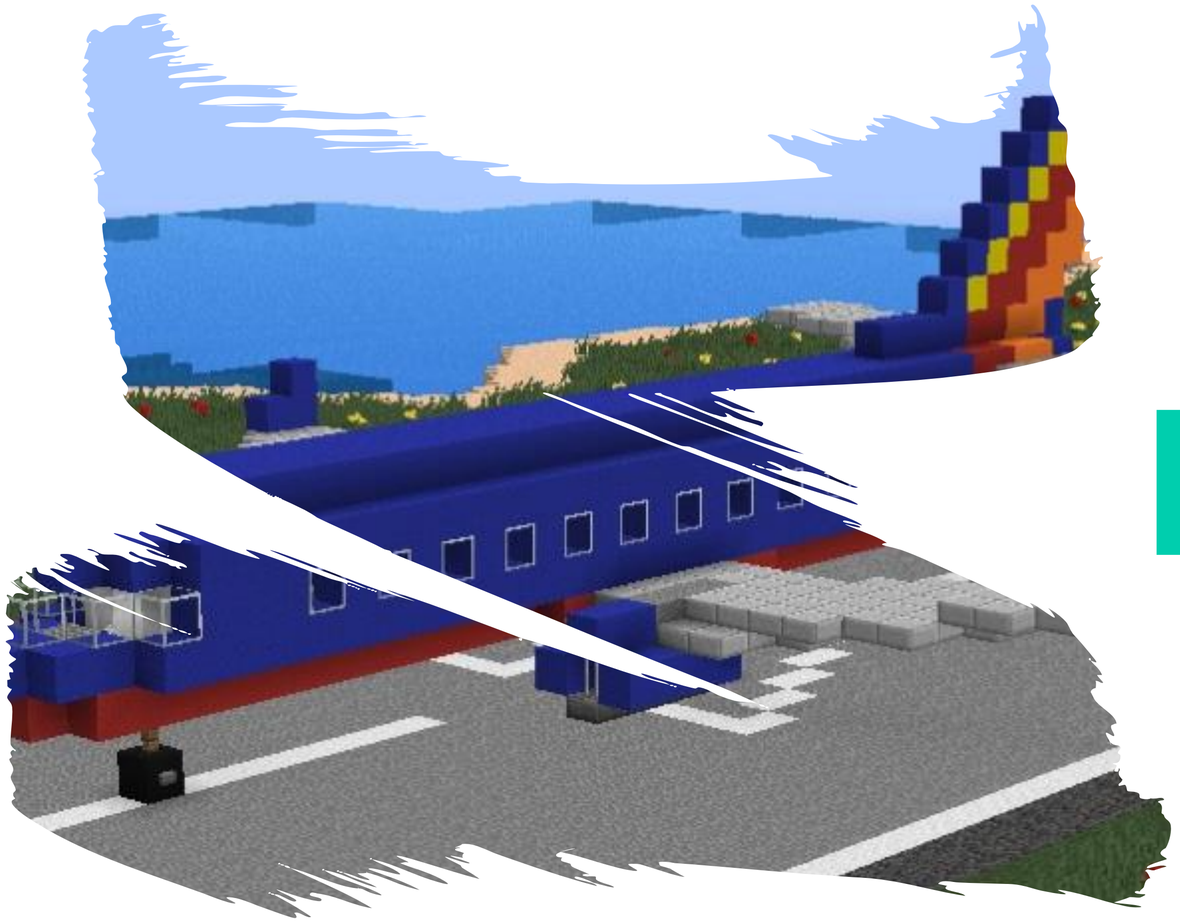
{‘neg’: 0.0, ‘neu’: 0.637, ‘pos’: 0.363, **‘compound’: 0.431**}

IPython console

Console 1/A

```
In [30]: print_confusion(df_2400.polarity_senti, df_2400.sentiment)
     True    -1    0    1   All
Predicted
-1           768   70   31   869
0            195  191   25   411
1            470  295  355  1120
All         1433  556  411  2400

In [31]: print_confusion(df_nr.polarity_senti, df_nr.sentiment)
     True   -1     0    1   All
Predicted
-1           144   628  117   889
0            271   674  130  1075
1             78  1256  702  2036
All          493  2558  949  4000

In [32]: print_confusion(df_kaggle.polarity_senti, df_kaggle.sentiment)
     True    -1     0     1    All
Predicted
-1          4890   477   197   5564
0           1221  1008   155   2384
1           3067  1614  2011   6692
All         9178  3099  2363  14640

In [32]:
```

Accuracy

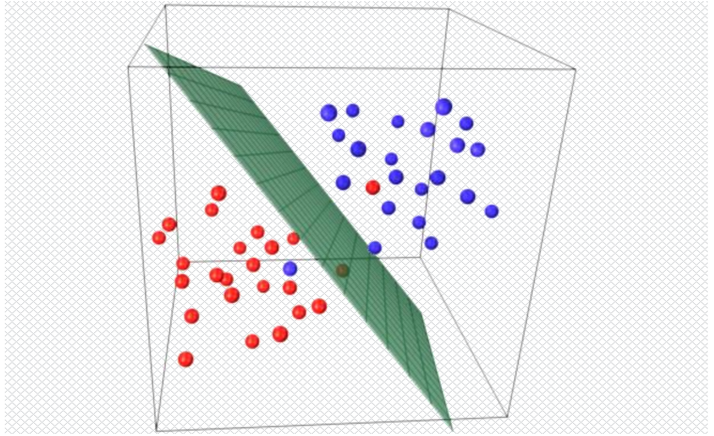


MACHINE LEARNING

Part I – **Binary** Classifications

Sentiment Analysis

MACHINE LEARNING I



1 Data Prep

Remove All Neutral Sentiment

2 Random Forest

Perform Random Forest Model

3 Linear SVM

Perform Support Vector Classifier

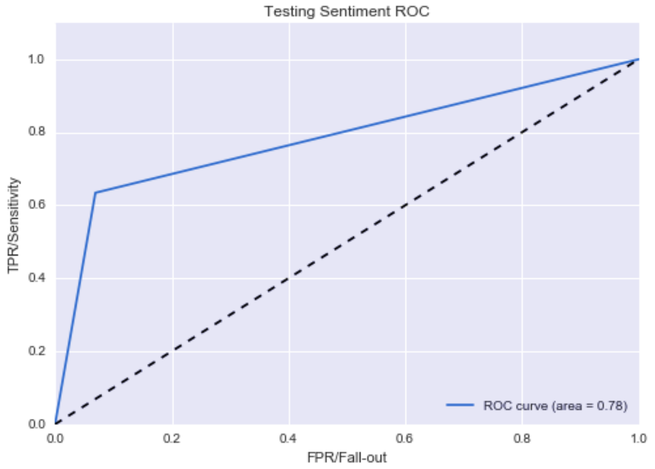
Hyperplane

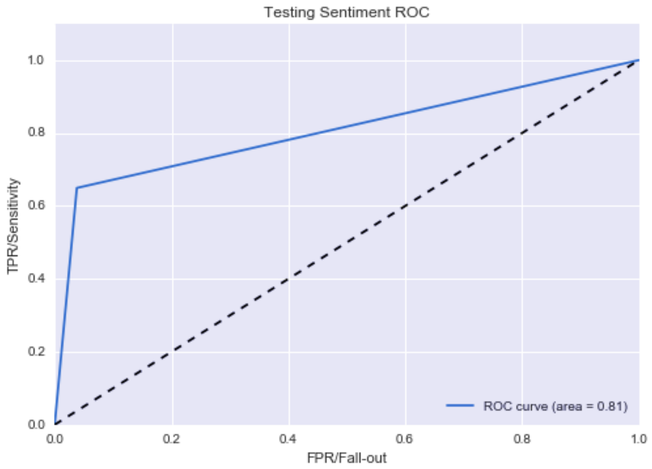
Sentiment Analysis

RANDOM FOREST

TRAINING: 9,266 TESTING: 2,275

Training Accuracy: 99.87%
Testing Accuracy: 89.58%





TRAINING: 1,124 TESTING: 290

Training Accuracy: 99.91%
Testing Accuracy: 84.48%

TRAINING: 11,541 TESTING: 1,843

Training Accuracy: 99.88%
Testing Accuracy: 86.49%

Sentiment Analysis

SVC

TRAINING: 1,124 TESTING: 290

Training Accuracy: 99.56%
Testing Accuracy: 84.14%

TRAINING: 9,266 TESTING: 2,275

Training Accuracy: 99.18%
Testing Accuracy: 90.95%

TRAINING: 11,541 TESTING: 1,843

Training Accuracy: 99.12%
Testing Accuracy: 86.38%

MACHINE LEARING

Part II – **Multi-Class** Classifications

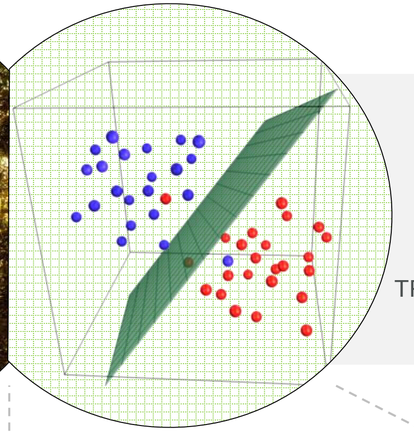
Sentiment Analysis

MACHINE LEARNING II

TRAINING: 2,040

TESTING: 360

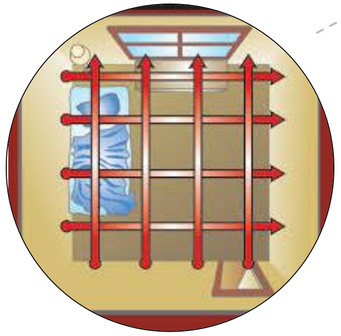
AdaBoost



SVC

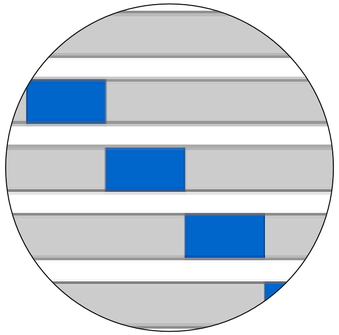
TRAINING: 14,640

TESTING: 2,400



Grid Search Cross-Validation

GridSearchCV ( )



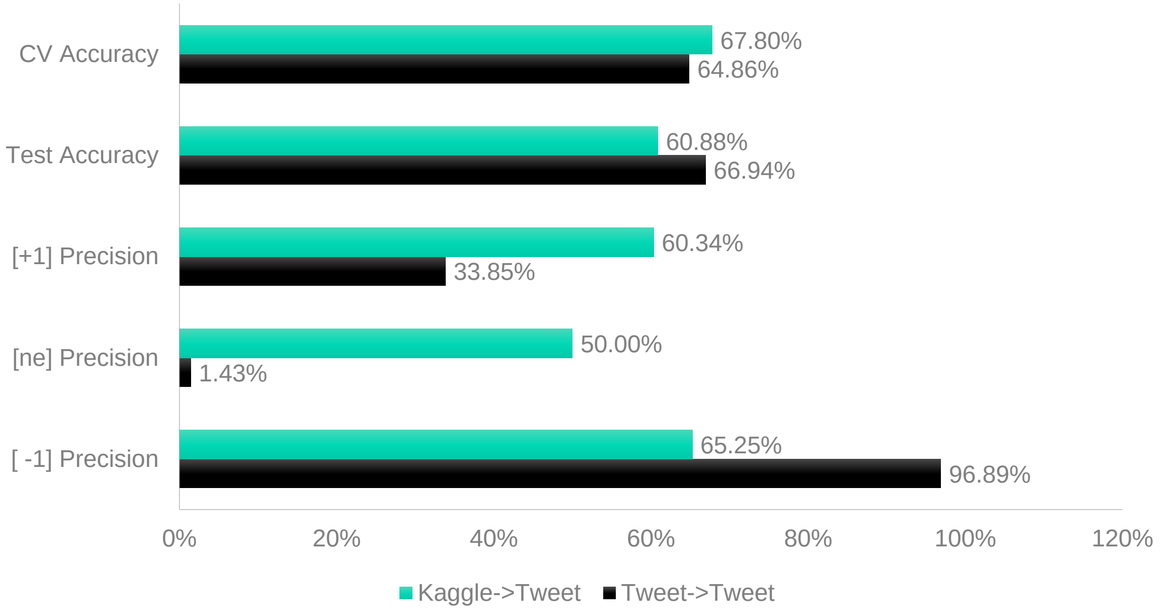
5-Fold Cross Validation

StratifiedKFold ( )

Apply Model on Testing

model.fit ( )

ADABOOST MODEL PERFORMANCE



While the test accuracy is slightly lower, using Kaggle data as training set proved to be a stronger predictor based on the cross-validation accuracy results. In addition, it is superior in positive and neutral precision, not to mention much higher training volume.

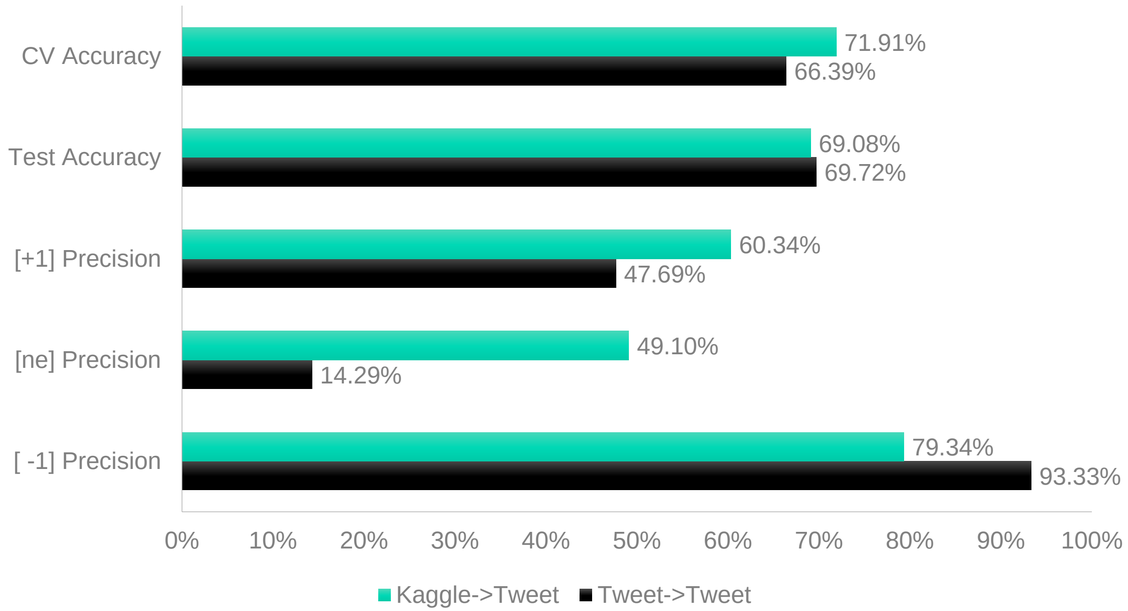
IPython console

Console 1/A

```
In [59]: print('The Average Score is {:2.4}%'.format(np.average(accu_cv2)*100))
    ...: multi_class_outcome(X_2400_test, y_2400_test, ada_model2)
    ...:
The Average Score is 64.86%
     True   -1   0   1  All
Predicted
-1        218  69  42  329
0           1   1   1    3
1           6   0  22   28
All       225  70  65  360
66.9444444444
Your Model Score is 66.94%

In [60]: print('The Average Score is {:2.4}%'.format(np.average(accu_cv)*100))
    ...: multi_class_outcome(X_comb_test, y_comb_test, ada_model)
    ...:
The Average Score is 70.88%
     True    -1    0    1   All
Predicted
-1         935  243   82  1260
0          377  278   81   736
1          121   35  248   404
All       1433  556  411  2400
60.875
Your Model Score is 60.88%

In [61]: '''
    ...: parameters = {'learning_rate':[0.1, 0.5, 1, 1.5, 2], 'algorithm': ['SAMME', 'SAMM
    ...: ada = AdaBoostClassifier(n_estimators=100, random_state=2016)
    ...: clf = grid_search.GridSearchCV(ada, parameters, cv=5, n_jobs=4)
    ...: clf.fit(X_comb_train, y_comb_train)
    ...: print(clf.best_params_)
    ...: #
    ...: '''
    ...:
    ...: ## Verifying result using Cross Validation on the entire Data Set: n_fold=5
    ...: skf = StratifiedKFold(y_comb_train, n_folds=5, random_state=2016)
    ...:
    ...: ada_model = AdaBoostClassifier(n_estimators=100, learning_rate=1.5, algorithm='SA
    ...: accu_cv = []
    ...: true_cv = []
    ...: pred_cv = []
    ...: for train_index, valid_index in skf:
    ...:     ada_model.fit(X_comb_train[train_index], y_comb_train[train_index])
```

```
IPython console
Console 1/A

In [47]: print('The Average Score is {:2.4}%'.format(np.average(accu_cv2)*100))
    ...: multi_class_outcome(X_2400_test, y_2400_test, svm_model2)
    ...:
The Average Score is 66.39%
     True   -1   0   1  All
Predicted
-1        210  57  32  299
0           6  10   2   18
1           9   3  31   43
All       225  70  65  360
69.7222222222
Your Model Score is 69.72%

In [48]: print('The Average Score is {:2.4}%'.format(np.average(accu_cv)*100))
    ...: multi_class_outcome(X_comb_test, y_comb_test, svm_model)
    ...:
The Average Score is 71.91%
     True    -1    0    1   All
Predicted
-1        1137  253   97  1487
0          217  273   66   556
1           79   30  248   357
All       1433  556  411  2400
69.0833333333
Your Model Score is 69.08%

In [49]: ## Here we Correlation between the truth and prediction
    ...: TrueLabel = list(itertools.chain(*true_cv))
    ...: PredictedLabel = list(itertools.chain(*pred_cv))
    ...: print ('Correlation between the actual and prediction is:', pearsonr(TrueLabel
    ...:        'with p-value',  ("%2.2f" % pearsonr(TrueLabel, PredictedLabel)[1]))
    ...:
Correlation between the actual and prediction is: 0.576245965274 with p-value 0.00

In [50]: ## Here we plot out the confusion matrix of the Cross-Validation Results
    ...: cm = confusion_matrix(PredictedLabel, TrueLabel)
    ...: fig, ax = plt.subplots()
    ...: im = ax.matshow(cm)
    ...: for (i, j), z in np.ndenumerate(cm):
    ...:     ax.text(j, i, '{:0.1f}'.format(z), ha='center', va='center',
    ...:             bbox=dict(boxstyle='round', facecolor='white', edgecolor='0.3'))
    ...: plt.title('Confusion matrix')
```

SVC MODEL PERFORMANCE



Again, while we see the precision of the twitter data is superior, the Kaggle data yields a more balanced results for all three sentiments. Because the overall test accuracy is about the same, we chose to use Kaggle data as training due to its superior CV accuracy as well as its large sum of data points (~8X).

CONCLUSION

Support Vector Classifier

Model Parameters: decision_fun='ovo', kernel='linear', cost=0.4, gamma=0

Use Entire Kaggle Dataset as Training Set (14,640 data points)

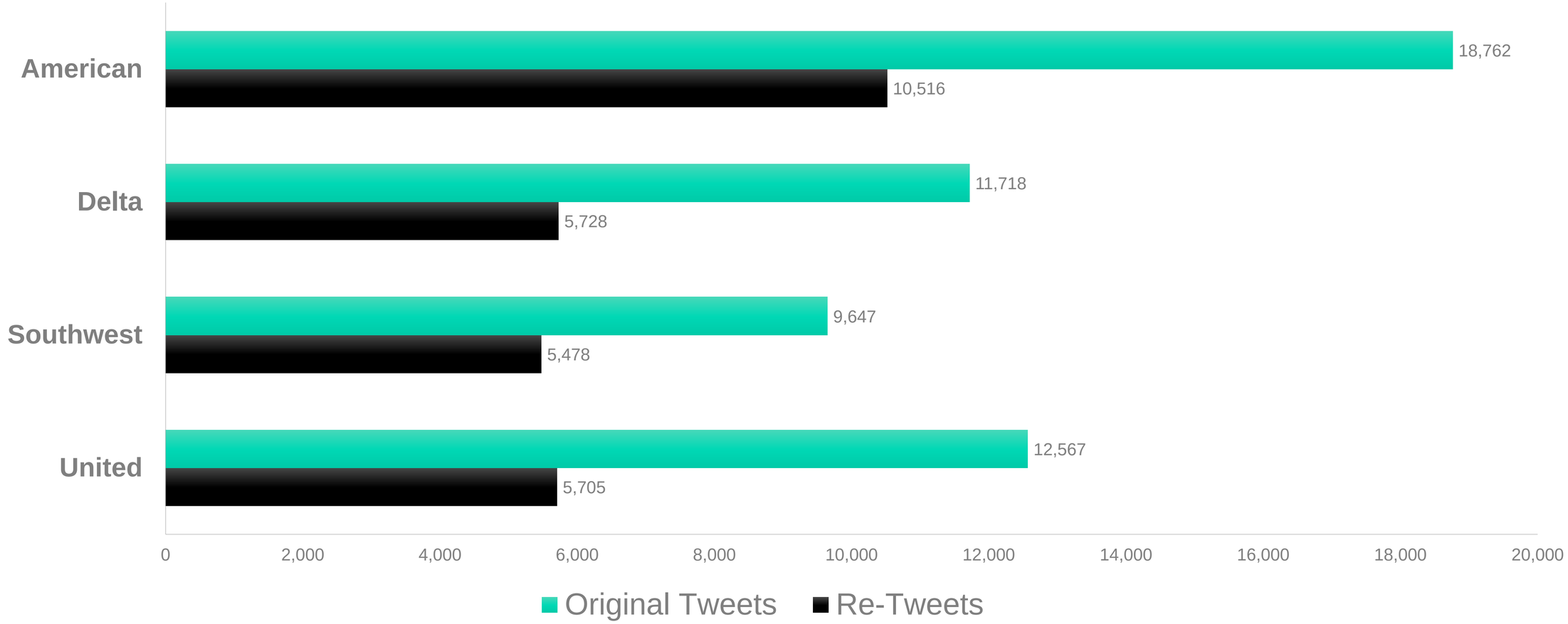
Apply Model to Tweets of Unknown Sentiment

STATISTICS

Now we have the model, let’s take a look at the results of collected tweets!

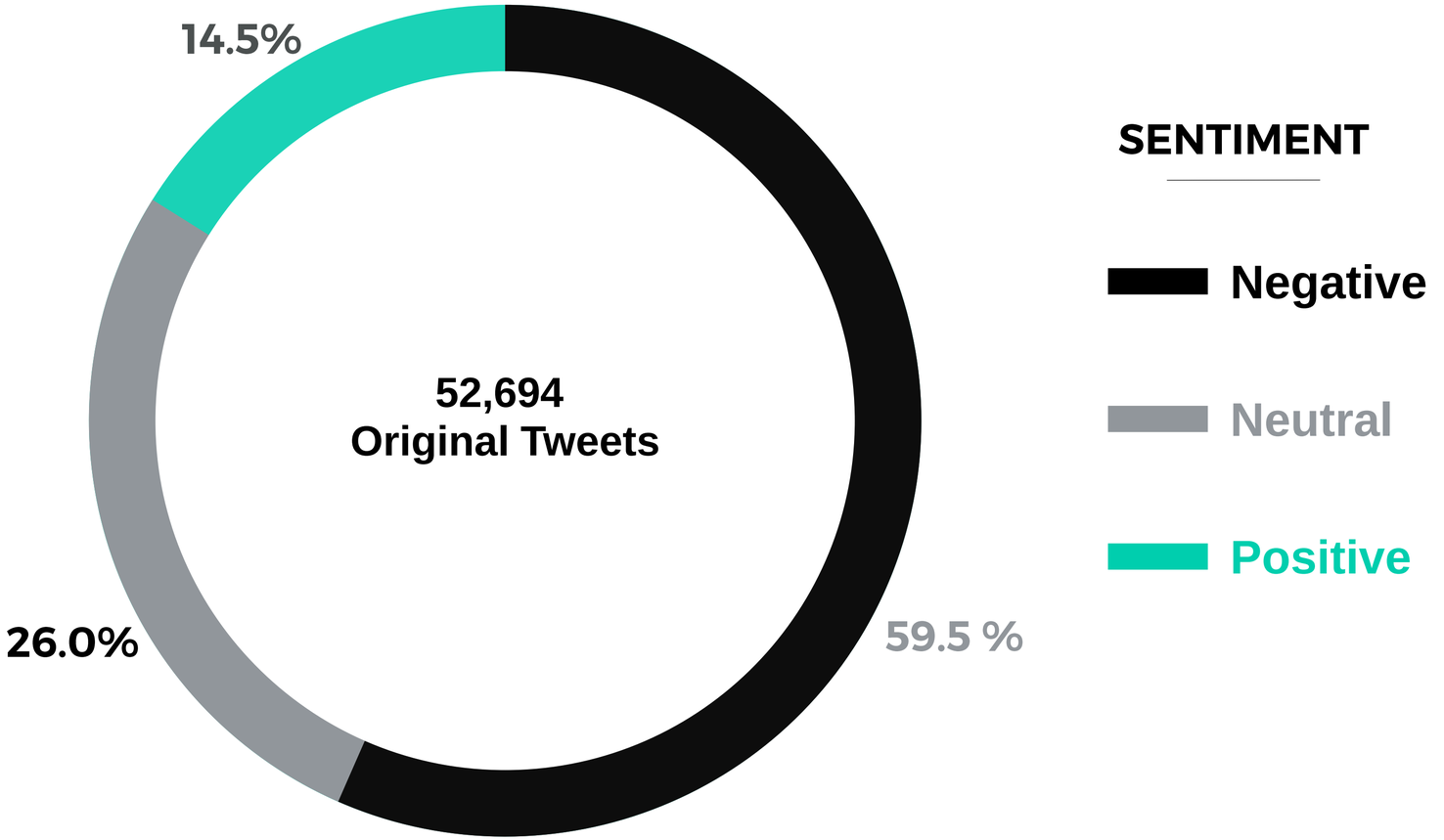
Sentiment Analysis

NUMBER OF TWEETS

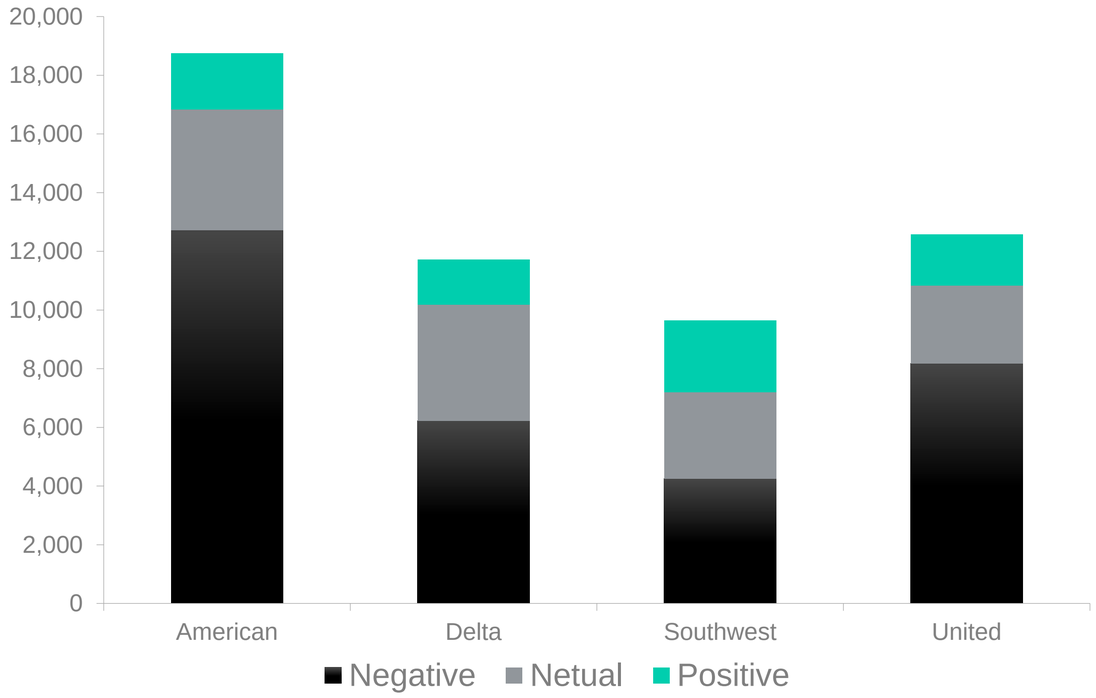


Sentiment Analysis

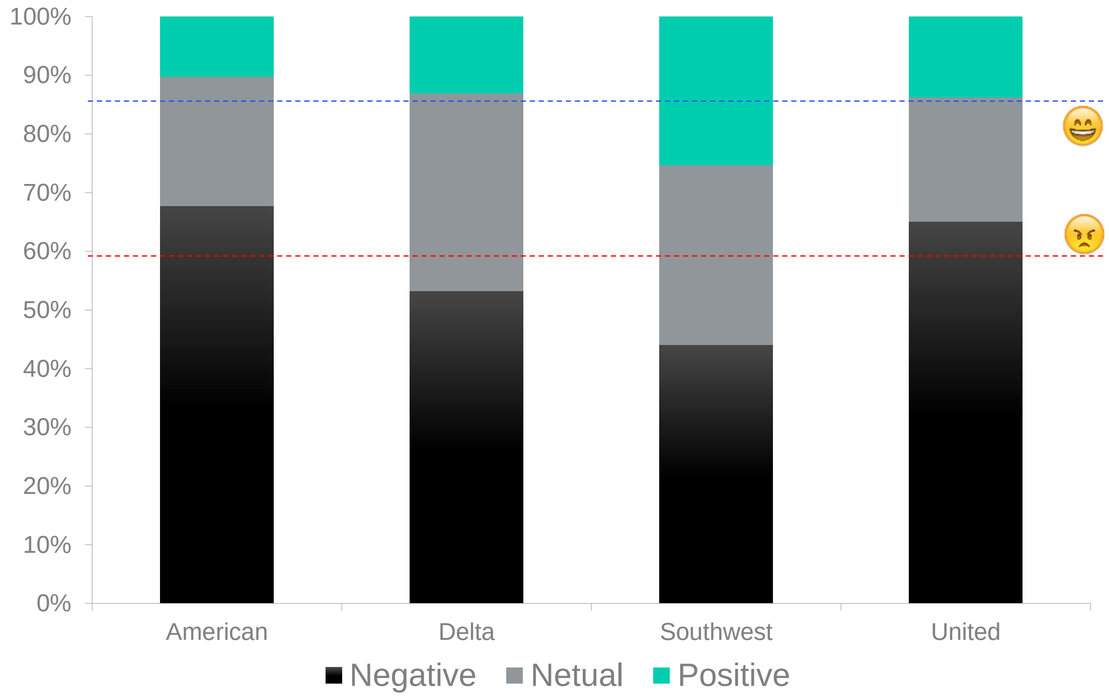
OVERALL SENTIMENT



SENTIMENT PER AIRLINE



Number of Original Tweets



Percentage of Original Tweets

AMERICAN AIRLINE

Negative Sentiment

“Cenk Uygur”

Positive Sentiment

-

Sentiment Analysis

DELTA AIRLINE

Negative Sentiment

“Delta Assist”

Positive Sentiment

-

SOUTHWEST AIRLINE

Negative Sentiment

“Dalton Rapattoni”

Positive Sentiment

“Dalton Rapattoni”

Sentiment Analysis

UNITED AIRLINE

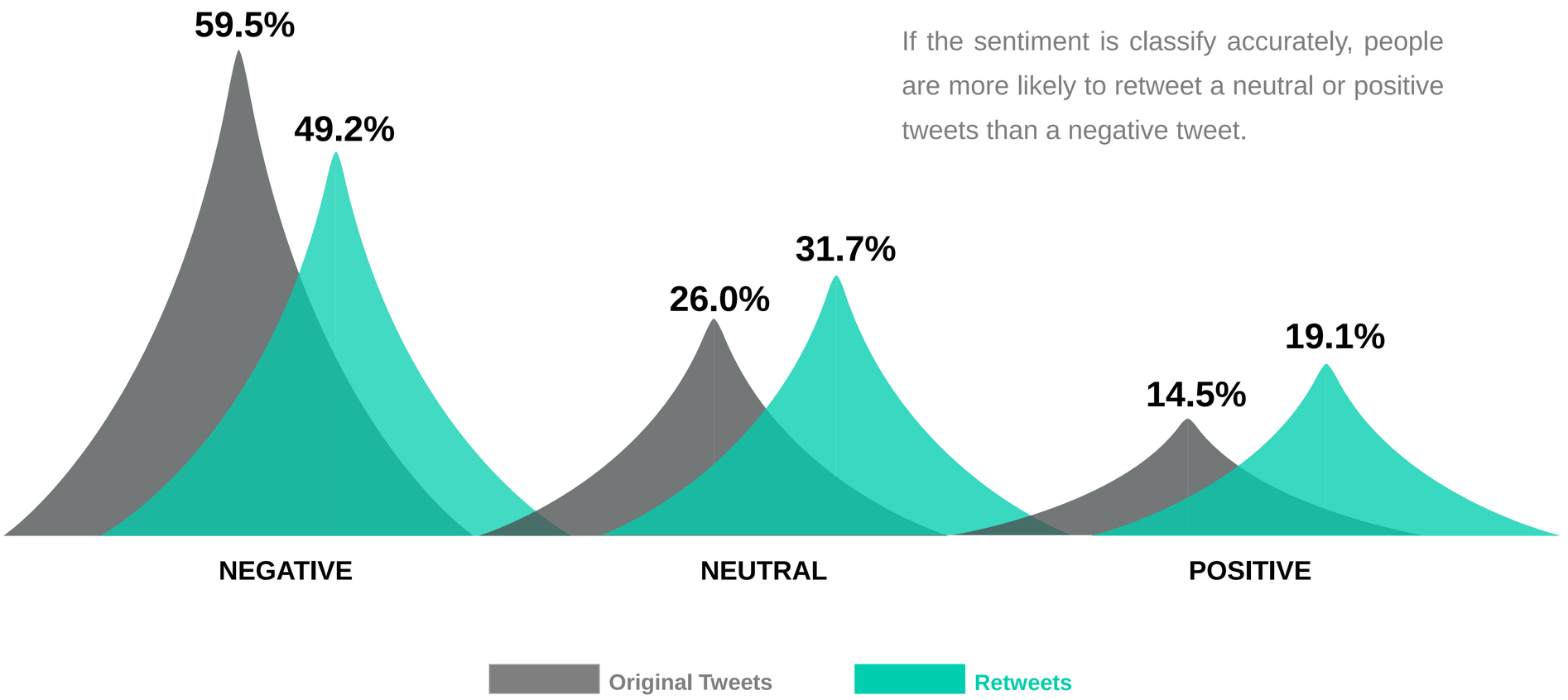
Negative Sentiment

“Muslim”, “Shame”

Positive Sentiment

“Happy Birthday”, “90th"

RETWEET PER SENTIMENT



If the sentiment is classify accurately, people are more likely to retweet a neutral or positive tweets than a negative tweet.

MOST RETWEETED TWEETS

10,260 Retweets…

RT @JensenAckles: @AmericanAir I know it's crazy #DFWsnow I'm just excited 2see my baby girl! Man freaking out next 2 me isn't helping

17,093 Retweets…

RT @Mets: We're partnering with @Delta to send a lucky fan to games 1 & 2! RT for your chance to win #DeltaMetsSweeps https://t.co/CDELSe84

3,362 Retweets…

RT @SouthwestAir: Yo #Kimye, we're really happy for you, we'll let you finish, but South is one of the best names of all time.

3,329 Retweets…

RT @antoniodelotero: this skank bitch was SO RUDE to me on my flight!!!! @united I asked for her name and she goes, "it's Britney bitch."

3,795 Retweets…

RT @UMG: The perfect trio @theweeknd @ddlovato and Lucian Grainge #MusicIsUniversal #UMGShowcase w/ @AmericanAir & @Citi https://t.co/J

3,535 Retweets…

RT @JackAndJackReal: Thanks @Delta for adding our movie to all your flights! If any of you are traveling and bored make sure u peep it

1,952 Retweets…

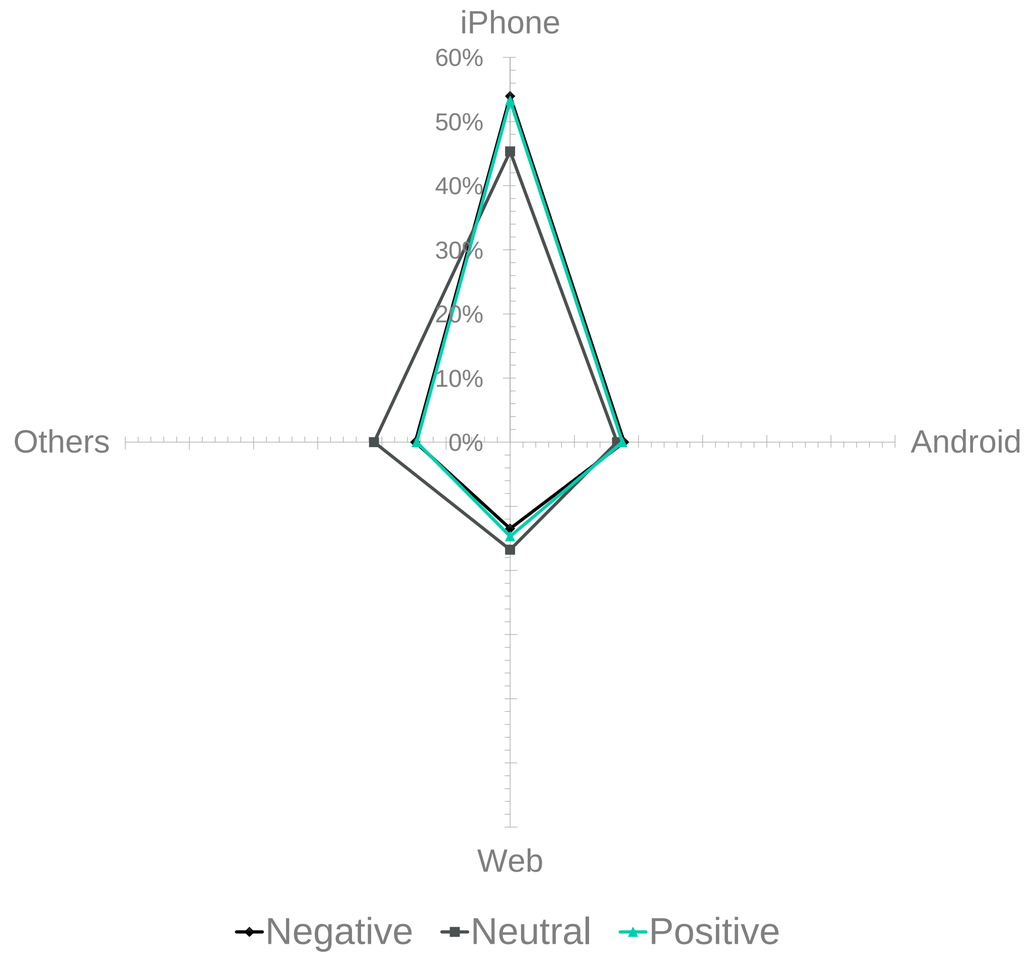
RT @DaltonRapattoni: Been home in Dallas for hours now but still in the airport. @SouthwestAir PLEASE find my bag. Please RT. #SWAFindDalto

352 Retweets…

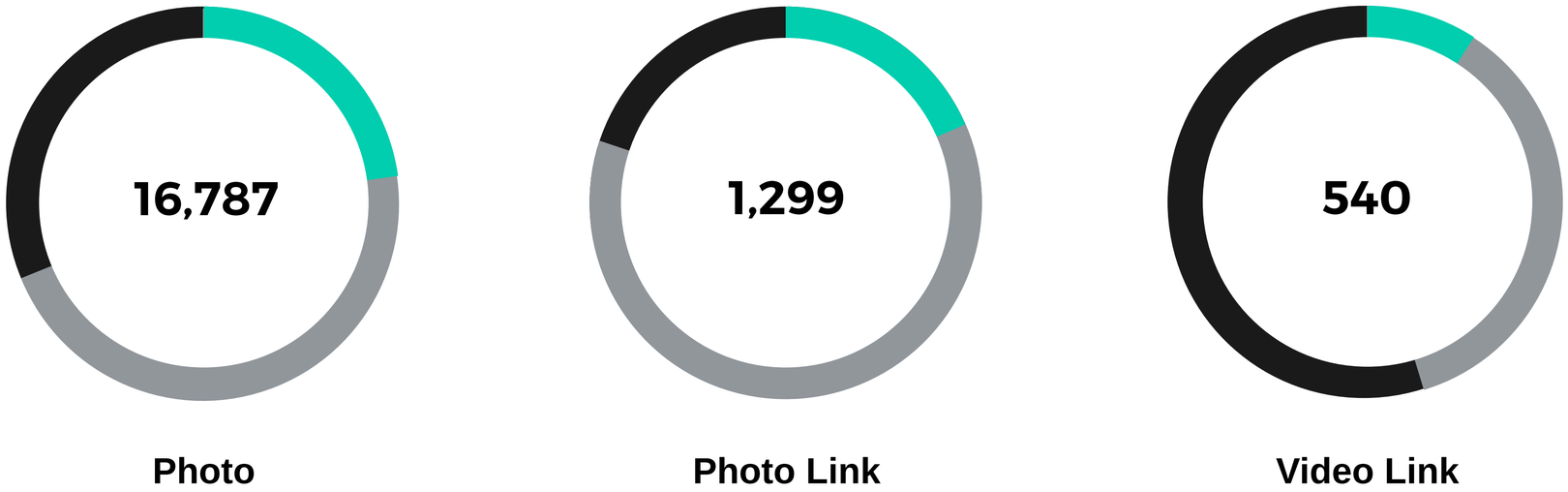
RT @ggreenwald: Arab-American family of 5 going on vacation - 3 young kids in tow - removed from @United plane for "safety reasons" https:/…

TWEET SOURCES

The majority of the tweets are from iPhone. There is also almost 20% of tweets from Android devices. The rest from the Web client and other applications. There is little difference between the source of negative and positive tweets; but there seems to be more alternative sources for the neutral sentiment.



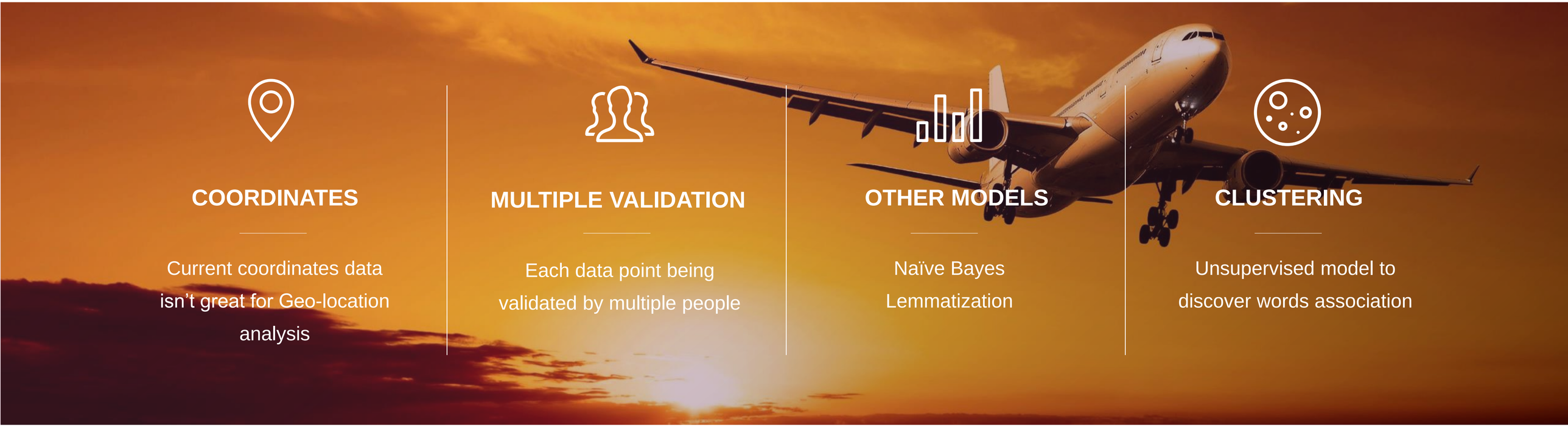
SENTIMENT PER MEDIA TYPE



ENDING THOUGHTS

Discussing final thoughts and some future steps

THOUGHTS & FUTURE STEPS



COORDINATES

Current coordinates data isn’t great for Geo-location analysis

MULTIPLE VALIDATION

Each data point being validated by multiple people

OTHER MODELS

Naïve Bayes
Lemmatization

CLUSTERING

Unsupervised model to discover words association

https://github.com/**michaelucsb**/ga-capstone-project-airlines


THANK
YOU
Q&A