Predicting Sentiments from Tweets

Tweets related to 2020 US Election

Motivation & Project Overview

Motivation:

Genuine interest to learn how the public options before the election day align with the actual outcome of the election

Project Overview:

This project explores public opinions about 2020 US election from the tweets posted by the users of Twitter and seeks to fully/partially answer the following questions.

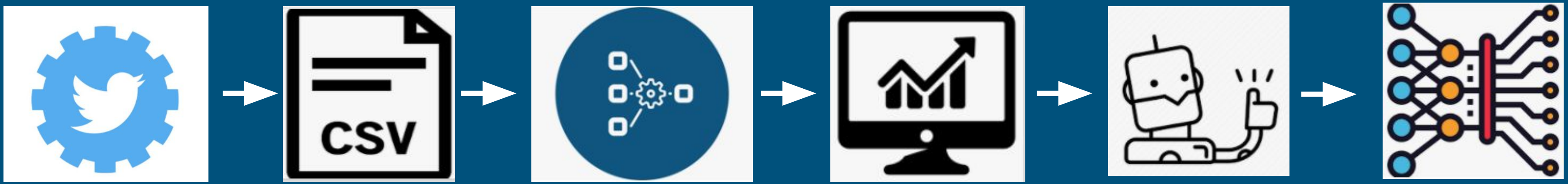
- What are the main topics of discussion
- Can we detect any possible October surprises
- Can we detect whether a tweet is negative or positive

Project Objective

The **main objective** of this project is going to be to set up a **data science pipeline** that facilitates:

1. Collection and creation of raw text data
2. Preprocessing and categorizing unlabeled text data
3. Training and evaluating deep learning models in Keras.

Approach



Twitter Streaming → Store data in CSV → Data wrangling → Visualize → Categorize unlabeled text data w/ Unsupervised learning algorithm → Prediction w/ Deep learning model

Data

32 columns

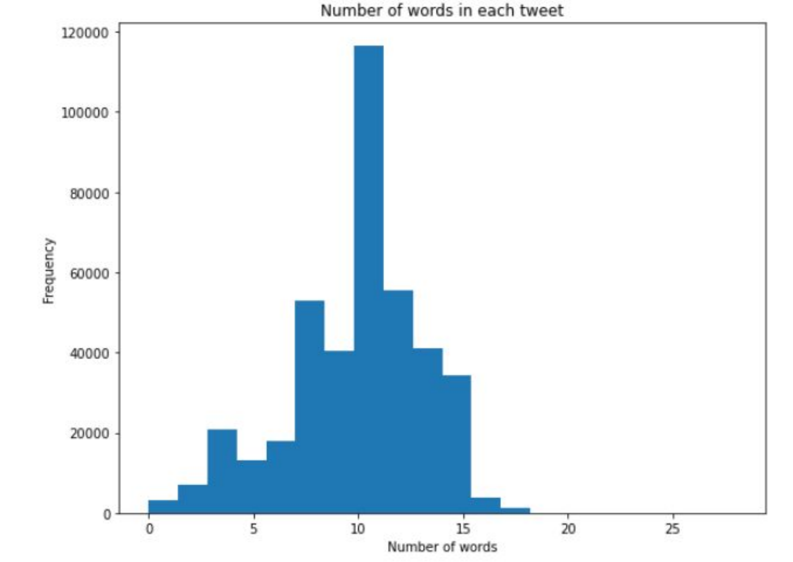
	created_at	id	id_str	text	source	truncated	in_reply_to_status_id	in_re
0	Fri Oct 16 05:01:51 +0000 2020	1316967569660776450	131696756966077645	RT @RudyGiuliani: The competing Town Halls wer...	a href="https://mobile.twitter.com" el="nofo...	False	NaN	NaN
1	Fri Oct 16 05:01:51 +0000 2020	1316967569648222211	1316967569648222211	RT @rachelv12: Trump and machismo https://t.c...	a href="https://mobile.twitter.com" el="nofo...	False	NaN	NaN
2	Fri Oct 16 05:01:51 +0000 2020	1316967569652371456	131696756965237145	RT @briantylercohen: Biden is like an encyclop...	a ref="http://twitter.com/download/iphone" ...	False	NaN	NaN
3	Fri Oct 16 05:01:51 +0000 2020	1316967569652371458	131696756965237145	RT @BradleyWhitford: Yo Semites!!! QAnon doesn...	a ref="http://twitter.com/download/iphone" ...	False	NaN	NaN
4	Fri Oct 16 05:01:51 +0000 2020	1316967569794977792	131696756979497779	RT @ACTBrigitte: Retweet if President Trump wo...	a ref="http://twitter.com/download/iphone" ...	False	NaN	NaN

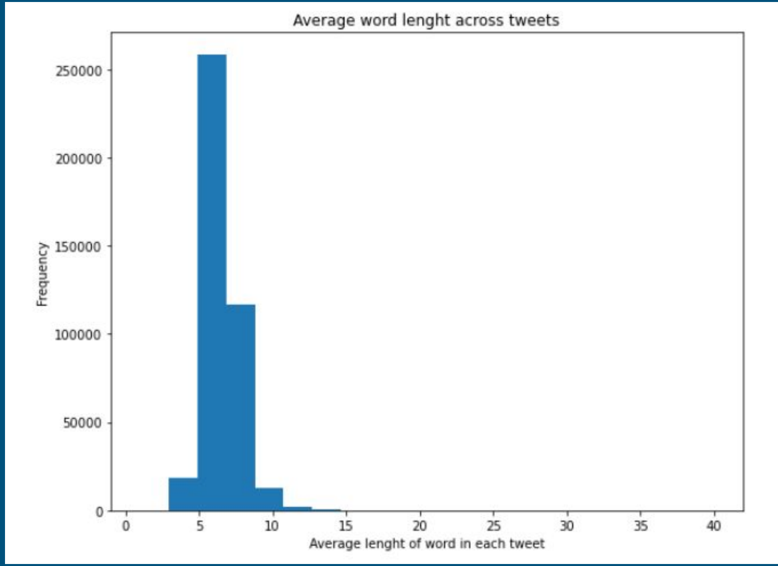
440,000 rows

Data Wrangling

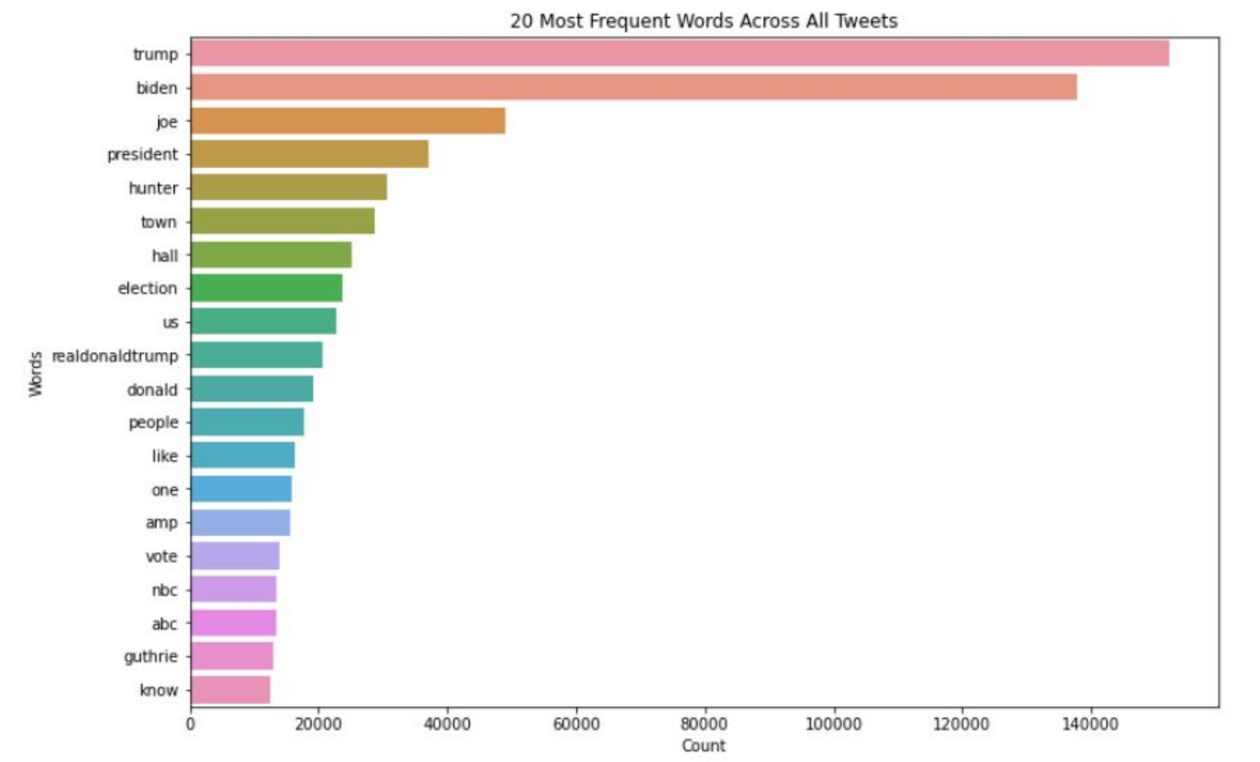
- Select only “English” tweets
- Drop duplicates
- Drop missing values
- Lowercase
- Remove punctuations
- Remove URLs, other acronyms
- Tokenize
- Remove stopwords

Exploratory Data Analysis





Unigrams - Top 20

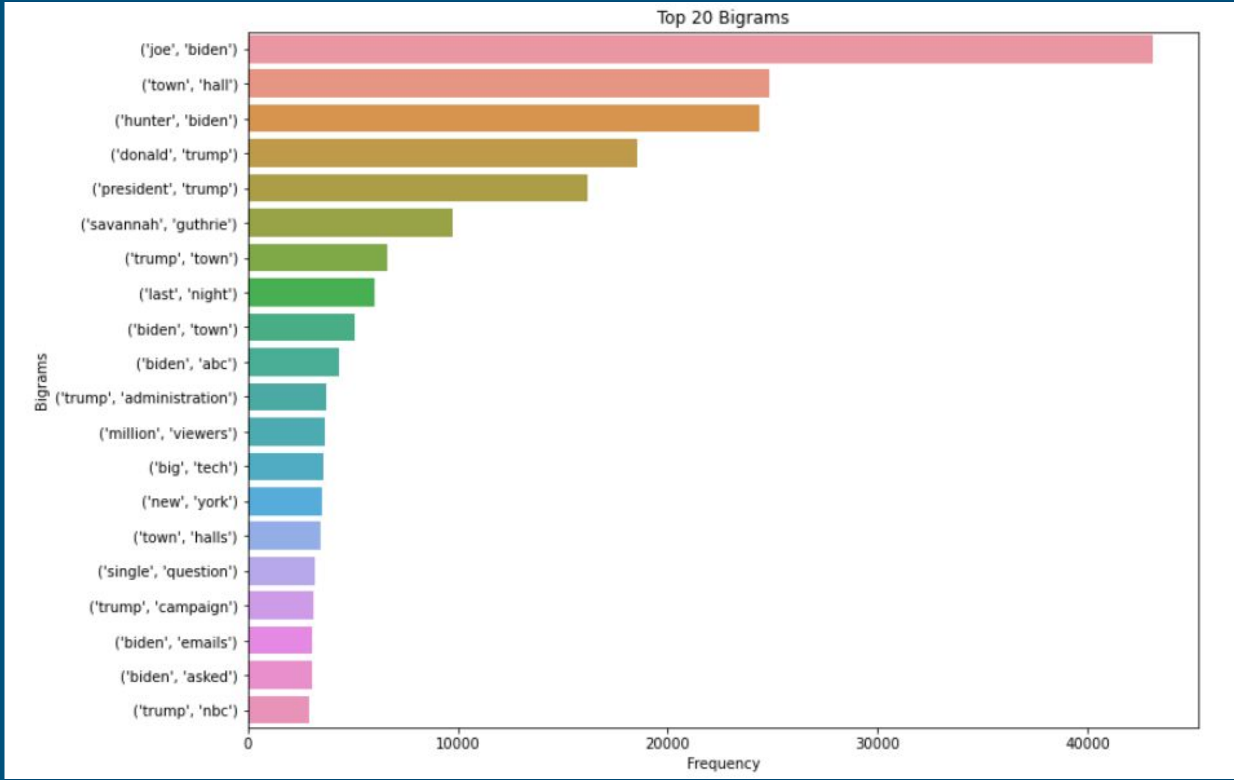


gensim
topic modelling for humans

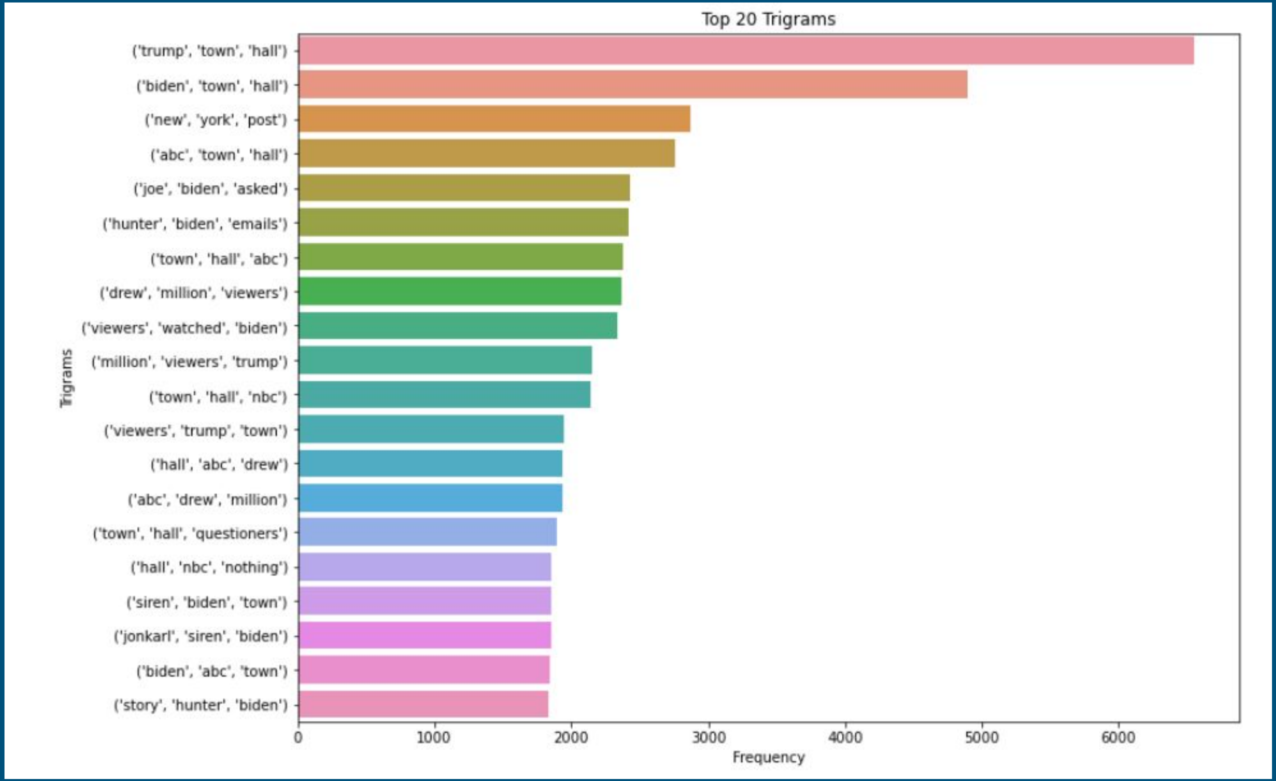
Seaborn

python
NLTK

Bigrams - Top 20



Trigrams - Top 20

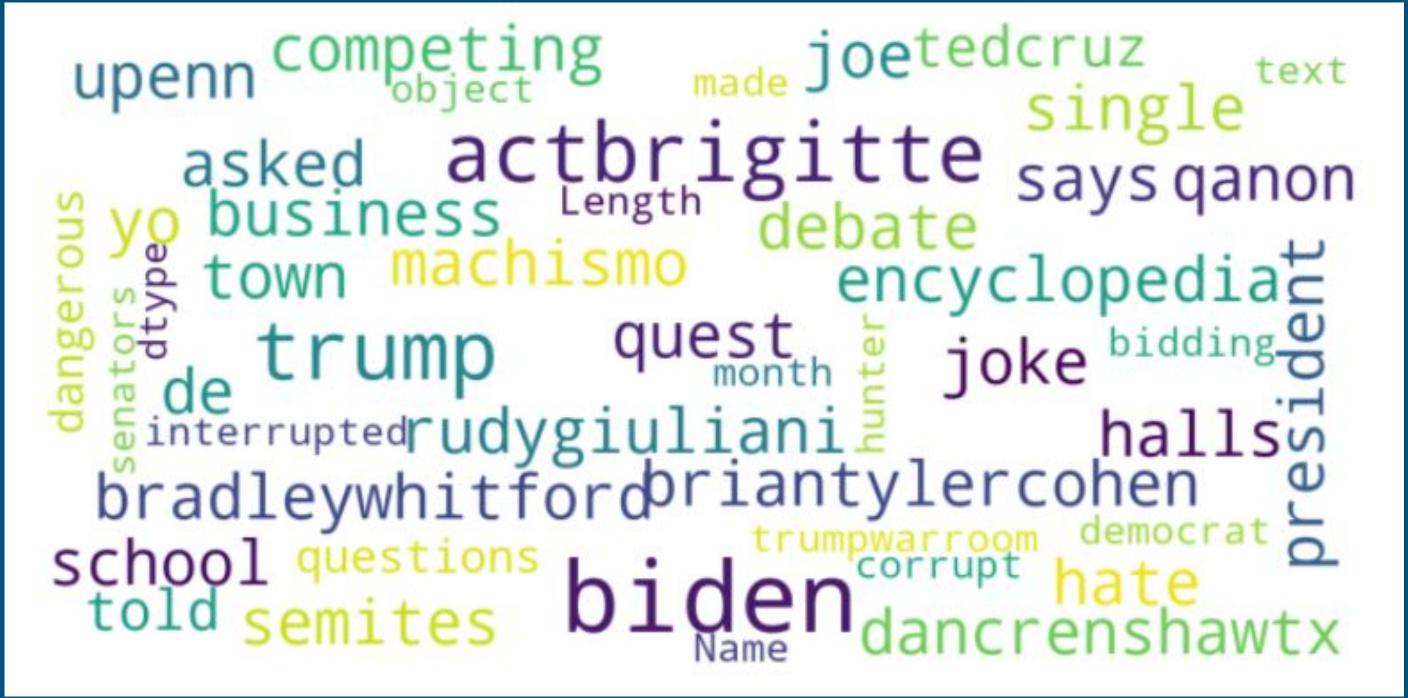


gensim
topic modelling for humans

Seaborn

python
NLTK

Wordcloud


upenn
competing
object
made
joe
tedcruz
text
single
asked
actbrigitte
says
qanon
Length
yo
business
debate
dangerous
dtype
town
machismo
encyclopedia
senators
trump
quest
hunter
joke
bidding
president
de
month
interrupted
rudygiuliani
halls
bradleywhitford
briantylercohen
democrat
trumpwarroom
school
questions
corrupt
hate
told
semites
biden
dancrenshawtx
Name


Python
type
function
use
expression
statement
language
object
used
also
version
code

Categorizing the unlabeled data

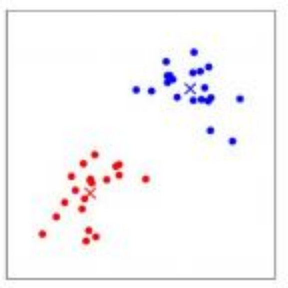
Text

Vader
Python
Package

	neg	neu	pos	compound
0	0.000	0.722	0.278	0.5000
1	0.000	1.000	0.000	0.0000
2	0.000	0.667	0.333	0.5423
3	0.552	0.448	0.000	-0.9022
4	0.000	1.000	0.000	0.0000

Sentiment metric

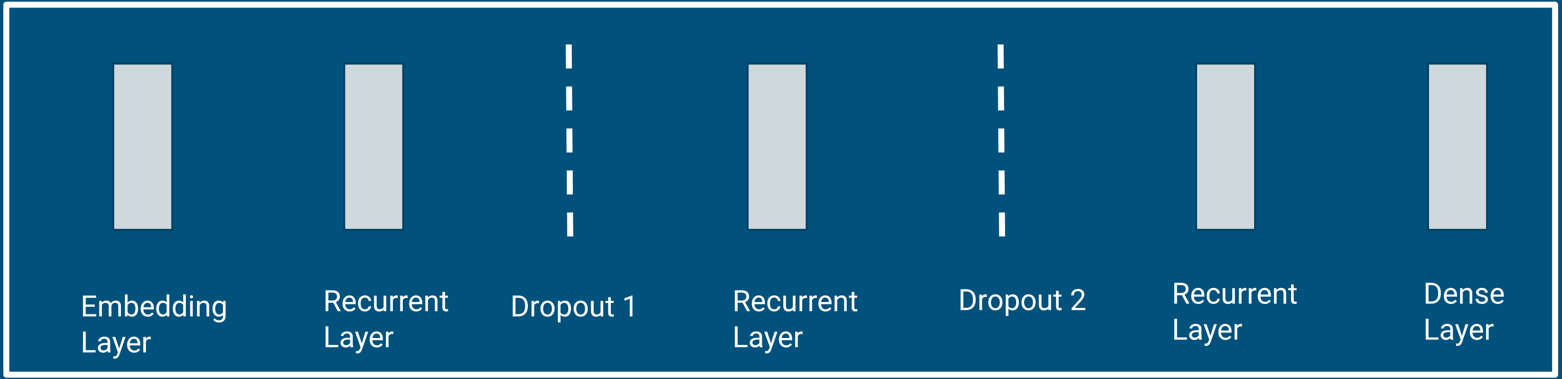
K-means clustering



2 classes

Deep learning architectures

SimpleRNN



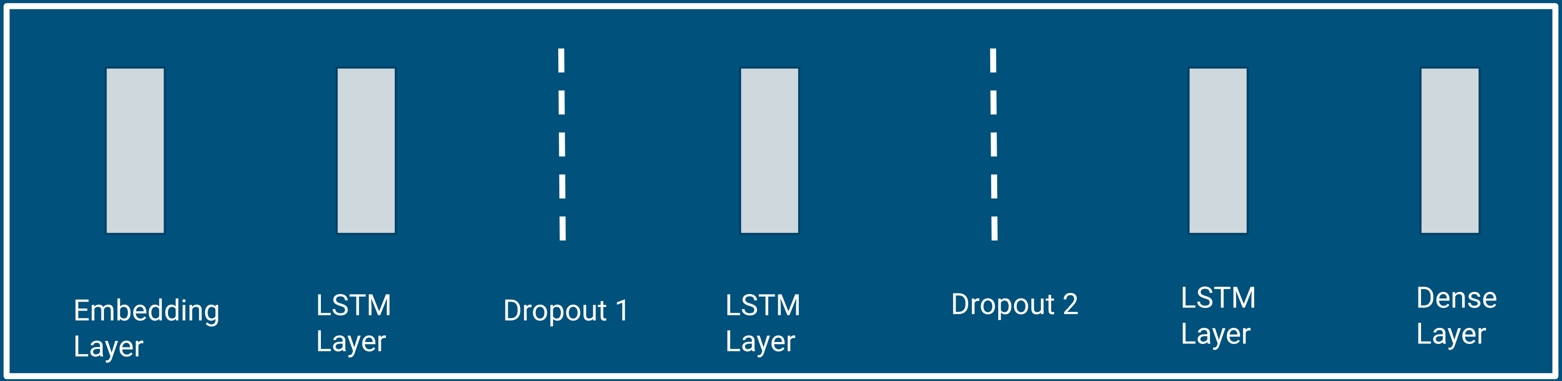
Activation function: tanh
Optimizer: Adam
Loss function: binary_crossentropy
Metric: accuracy
Train/Test ratio: 70% / 30%

Keras & TensorFlow

Amazon SageMaker

Deep learning architectures

LSTM



Activation function: tanh
Optimizer: Adam
Loss function: binary_crossentropy
Metric: accuracy
Train/Test ratio: 70% / 30%

Modeling - SimpleRNN

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
embedding_2 (Embedding)      (None, 14, 32)            2084032
_________________________________________________________________
simple_rnn_4 (SimpleRNN)     (None, 14, 32)            2080
_________________________________________________________________
dropout_3 (Dropout)          (None, 14, 32)            0
_________________________________________________________________
simple_rnn_5 (SimpleRNN)     (None, 14, 32)            2080
_________________________________________________________________
dropout_4 (Dropout)          (None, 14, 32)            0
_________________________________________________________________
simple_rnn_6 (SimpleRNN)     (None, 32)                2080
_________________________________________________________________
dense_2 (Dense)              (None, 2)                 66
=================================================================
Total params: 2,090,338
Trainable params: 2,090,338
Non-trainable params: 0
```

```
Train on 154526 samples, validate on 66226 samples
Epoch 1/10
154526/154526 [==============================] - 202s 1ms/step - loss: 0.2016 - acc: 0.9202 - val_loss: 0.1544 - val_acc: 0.9487
Epoch 2/10
154526/154526 [==============================] - 195s 1ms/step - loss: 0.1370 - acc: 0.9521 - val_loss: 0.1483 - val_acc: 0.9488
Epoch 3/10
154526/154526 [==============================] - 194s 1ms/step - loss: 0.1350 - acc: 0.9529 - val_loss: 0.1404 - val_acc: 0.9519
Epoch 4/10
154526/154526 [==============================] - 194s 1ms/step - loss: 0.1345 - acc: 0.9531 - val_loss: 0.1632 - val_acc: 0.9440
Epoch 5/10
154526/154526 [==============================] - 194s 1ms/step - loss: 0.1350 - acc: 0.9534 - val_loss: 0.1486 - val_acc: 0.9481
Epoch 6/10
154526/154526 [==============================] - 194s 1ms/step - loss: 0.1338 - acc: 0.9537 - val_loss: 0.1421 - val_acc: 0.9520
Epoch 7/10
154526/154526 [==============================] - 194s 1ms/step - loss: 0.1327 - acc: 0.9539 - val_loss: 0.1535 - val_acc: 0.9464
Epoch 8/10
154526/154526 [==============================] - 194s 1ms/step - loss: 0.1334 - acc: 0.9539 - val_loss: 0.1457 - val_acc: 0.9504
Epoch 9/10
154526/154526 [==============================] - 194s 1ms/step - loss: 0.1323 - acc: 0.9537 - val_loss: 0.1395 - val_acc: 0.9510
Epoch 10/10
154526/154526 [==============================] - 194s 1ms/step - loss: 0.1324 - acc: 0.9535 - val_loss: 0.1372 - val_acc: 0.9560
Accuracy: 95.55%
Training duration(minutes): 33.20819400548935
```

Keras & TensorFlow

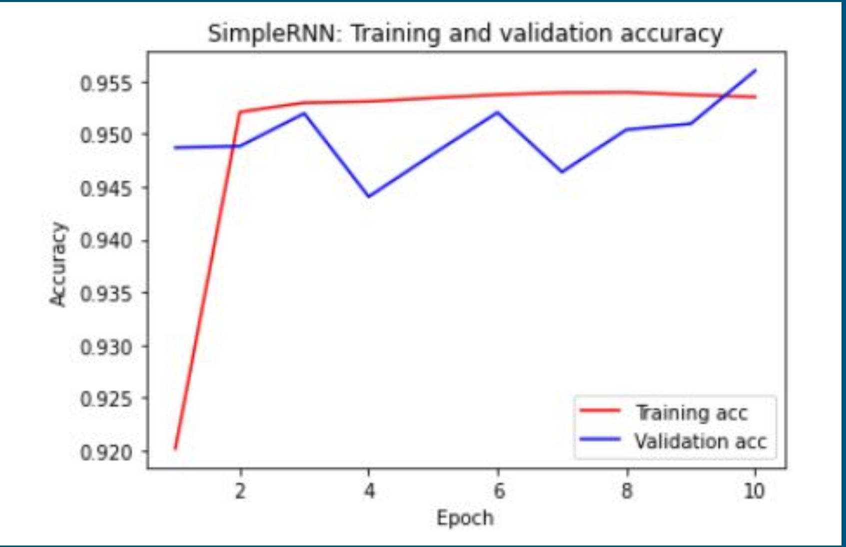
Amazon SageMaker

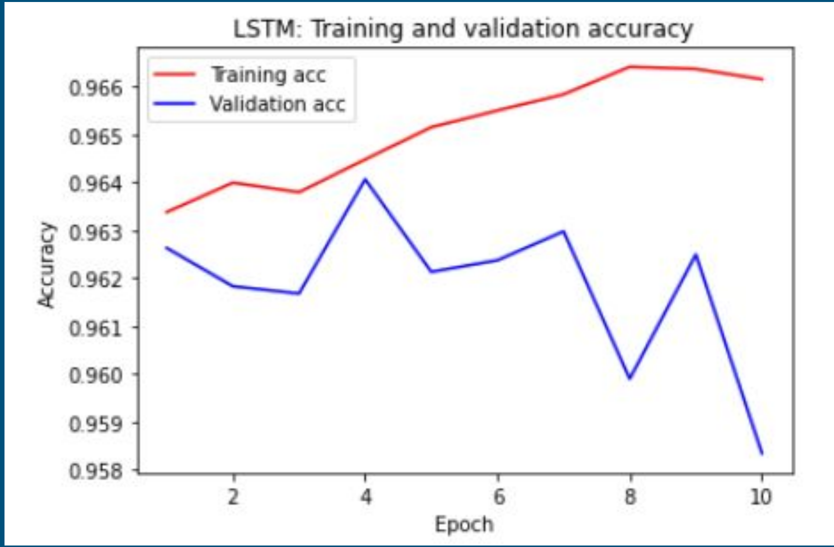
Modeling - LSTM

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
embedding_3 (Embedding)      (None, 14, 32)            2084032
_________________________________________________________________
lstm_7 (LSTM)                (None, 14, 32)            8320
_________________________________________________________________
dropout_5 (Dropout)          (None, 14, 32)            0
_________________________________________________________________
lstm_8 (LSTM)                (None, 14, 32)            8320
_________________________________________________________________
dropout_6 (Dropout)          (None, 14, 32)            0
_________________________________________________________________
lstm_9 (LSTM)                (None, 32)                8320
_________________________________________________________________
dense_3 (Dense)              (None, 2)                 66
=================================================================
Total params: 2,109,058
Trainable params: 2,109,058
Non-trainable params: 0
```

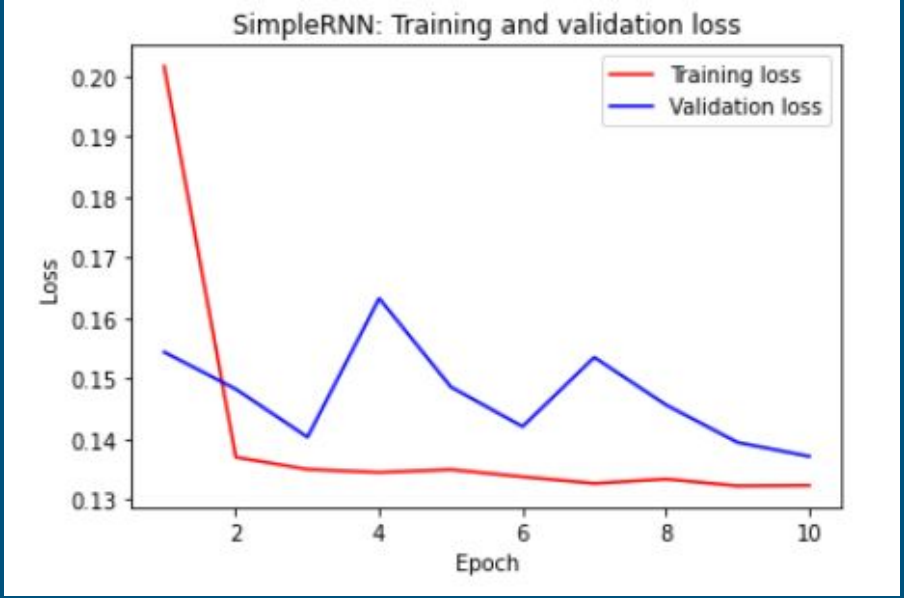
```
Train on 154526 samples, validate on 66226 samples
Epoch 1/10
154526/154526 [==============================] - 418s 3ms/step - loss: 0.1038 - acc: 0.9634 - val_loss: 0.1088 - val_acc: 0.9626
Epoch 2/10
154526/154526 [==============================] - 418s 3ms/step - loss: 0.1030 - acc: 0.9640 - val_loss: 0.1112 - val_acc: 0.9618
Epoch 3/10
154526/154526 [==============================] - 419s 3ms/step - loss: 0.1031 - acc: 0.9638 - val_loss: 0.1114 - val_acc: 0.9617
Epoch 4/10
154526/154526 [==============================] - 418s 3ms/step - loss: 0.1014 - acc: 0.9645 - val_loss: 0.1060 - val_acc: 0.9641
Epoch 5/10
154526/154526 [==============================] - 418s 3ms/step - loss: 0.0993 - acc: 0.9652 - val_loss: 0.1094 - val_acc: 0.9621
Epoch 6/10
154526/154526 [==============================] - 418s 3ms/step - loss: 0.0993 - acc: 0.9655 - val_loss: 0.1114 - val_acc: 0.9624
Epoch 7/10
154526/154526 [==============================] - 418s 3ms/step - loss: 0.0983 - acc: 0.9658 - val_loss: 0.1065 - val_acc: 0.9630
Epoch 8/10
154526/154526 [==============================] - 419s 3ms/step - loss: 0.0966 - acc: 0.9664 - val_loss: 0.1083 - val_acc: 0.9599
Epoch 9/10
154526/154526 [==============================] - 422s 3ms/step - loss: 0.0961 - acc: 0.9664 - val_loss: 0.1095 - val_acc: 0.9625
Epoch 10/10
154526/154526 [==============================] - 425s 3ms/step - loss: 0.0962 - acc: 0.9662 - val_loss: 0.1184 - val_acc: 0.9583
Accuracy: 96.07%
Training duration(minutes): 71.24150105714799
```

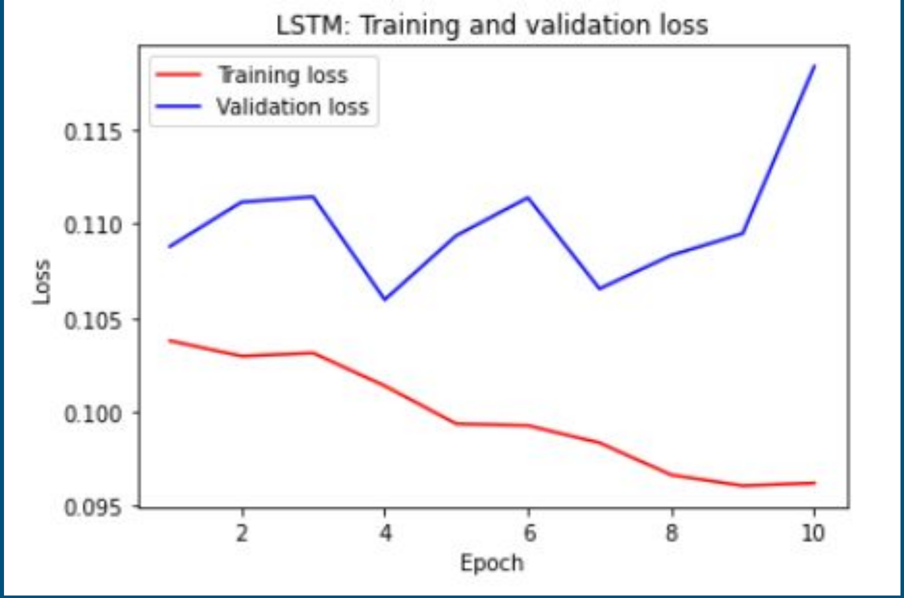
Model evaluation - accuracy





Model evaluation - loss





Conclusion

The accuracies from both SimpleRNN and LSTM appear to be overfitting and the validation accuracies have high variance for the same reason. While the validation accuracy in SimpleRNN starts to increase, it follows a trend in the opposite direction to drop down to lower levels in the LSTM model.

To improve the model performance, the following recommendations are suggested for the next step:

- Find another approach to categorize the unlabeled text data
- Try different RNN architectures
- Perform more advanced hyperparameter tuning of the RNNs
- Perform cross-validation
- Make the data multi-class
- Perform topic modeling with Latent Dirichlet Allocation

Skills Practiced During This Project

- How to efficiently collect large amount of data from Twitter via Tweepy and Twitter API
- How to efficiently work with large dataset
- How to build deep learning architectures, compile and fit in Keras
- How to apply basic NLP concepts and techniques to a text data

Jupyter notebooks can be found here