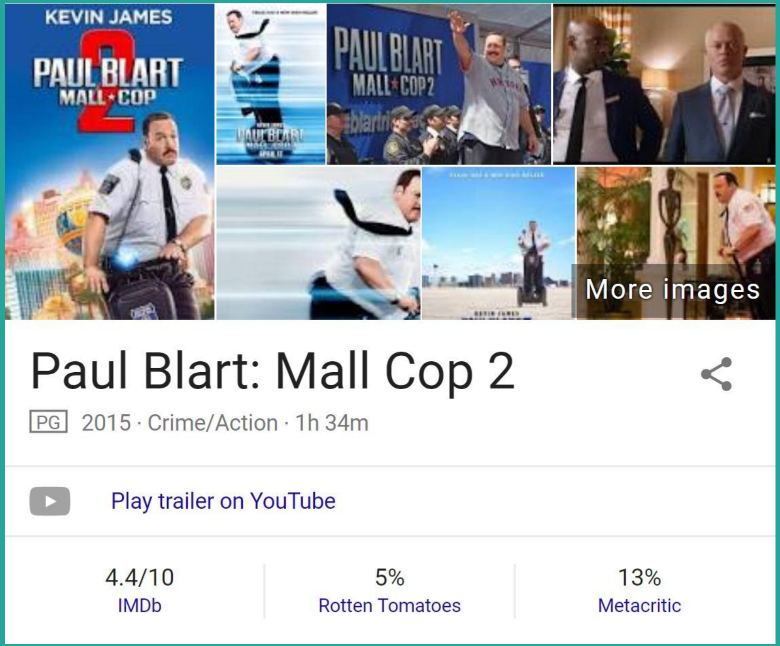
NLP: Multi-Class Emoji Sentiment Analysis

Hongyi Zhang

November 27th 2018

A look back on PA1 : Naive Bayes & movies!



Multi-Class Analysis to find determine emojis

Emotions/Classes

Confusion all around!
I hate you!! xD
I love history classes.

- Joy
- Fear
- Anger
- Sadness
- Annoyance/disgust
- Disappointment/shame
- Guilt

Data Set from DeepMoji

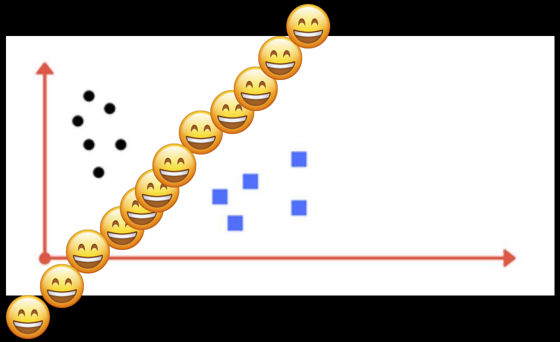
1. Tweets with emotions attached to them

2. Preprocessed with ngrams to find features!

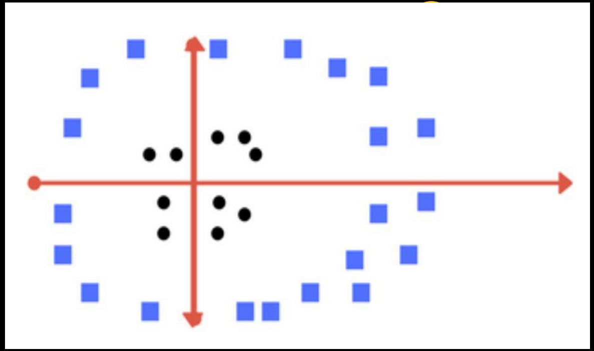
3. Unigrams, Bigrams, Trigrams

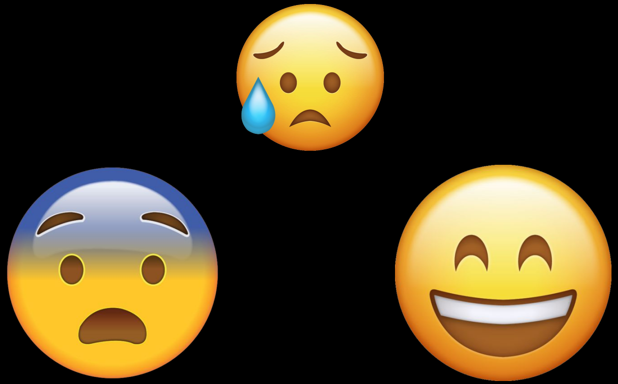

```
[ 0.  0.  0.  0.  0.  1.  0.] When I did not speak the truth.
[ 0.  0.  0.  0.  0.  0.  1.] When I caused problems for somebody because he could not keep the appointed time and this led to various consequences.
[ 1.  0.  0.  0.  0.  0.  0.] When I got a letter offering me the Summer job that I had applied for.
[ 0.  1.  0.  0.  0.  0.  0.] When I was going home alone one night in Paris and a man came up behind me and asked me if I was not afraid to be out alone so late at night.
[ 0.  0.  1.  0.  0.  0.  0.] When I was talking to HIM at a party for the first time in a long while and a friend came and interrupted us and HE left.
[ 0.  0.  0.  1.  0.  0.  0.] When my friends did not ask me to go to a New Year's party with them.
```

SVM- Support Vector Machines

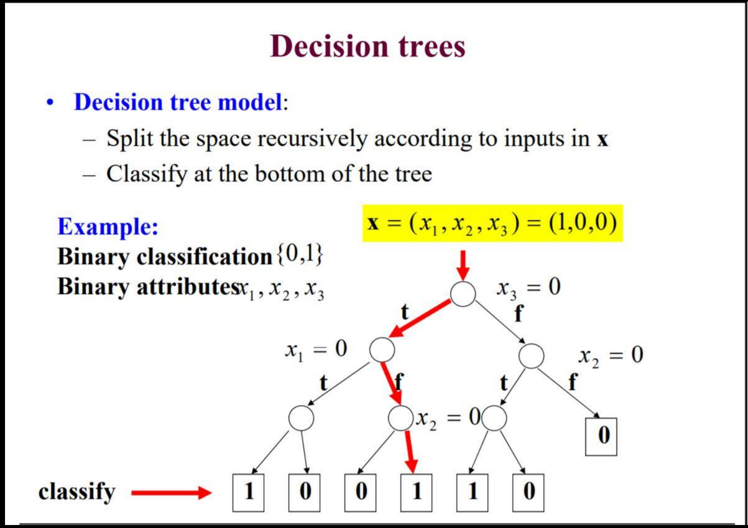


Linear SVM's





Decision Trees



Decision trees

- **Decision tree model**:
  - Split the space recursively according to inputs in **x**
  - Classify at the bottom of the tree

Example:
Binary classification $\{0,1\}$
Binary attributes x_1, x_2, x_3

With this data...

```
Tweet data example: ['0. 1. 0. 0. 0. 0. 0.', 'When I was involved in a
 traffic accident.']


Features with 1 ngrams using Counter:
Counter({'when': 1, 'i': 1, 'was': 1, 'involved': 1, 'in': 1, 'a': 1,
'traffic': 1, 'accident': 1, '.': 1})


Emotion Corresponding to the feature : fear
```

We get this...

Unigram

Classifier	Training Accuracy	Test Accuracy
SVC	0.1540404	0.1377005
LinearSVC	0.9872252	0.5561497
DecisionTreeClassifier	0.9988116	0.4385027

Bigram

Classifier	Training Accuracy	Test Accuracy
SVC	0.1460190	0.1350267
LinearSVC	0.9988116	0.5855615
DecisionTreeClassifier	0.9988116	0.4465241

We get this...

Bigram

Classifier	Training Accuracy	Test Accuracy
SVC	0.1460190	0.1350267
LinearSVC	0.9988116	0.5855615
DecisionTreeClassifier	0.9988116	0.4465241

Trigram

Classifier	Training Accuracy	Test Accuracy
SVC	0.1460190	0.1350267
LinearSVC	0.9988116	0.5842246
DecisionTreeClassifier	0.9988116	0.4518717

Conclusion

Trigram Implementation improves accuracy.

Reason for such a big difference in Test and Train? (Overfitting)

Linear SVM is very 😄 for multi-class classification

Need a bigger dataset!! 😥

EMOCONTEXT: HUMANIZING ARTIFICIAL INTELLIGENCE

BY: HAARIS PADELA

DATASET FORMAT

id	turn1	turn2	turn3	label
156	You are funny	LOL I know that. :p	😊	happy
187	Yeah exactly	Like you said, like brother like sister ;)	Not in the least	others

STEP 1: PROCESS DATA

```
Test Dataset
['then dont ask me <eos> youre a guy not as if you would understand <eos> im not a guy fuck off', 'mixed things  such as
 ?  <eos> the things you do .  <eos> have you seen minions ?', "today i'm very happy <eos> and i'm happy for you ☐ <eos>
 i will be marry", 'woah bring me some <eos> left it there oops <eos> brb', 'it is thooooo <eos> i said soon master .  <
eos> he is pressuring me', 'wont u ask my age ?  <eos> hey at least i age well !  <eos> can u tell me how can we get clo
ser ?', "i said yes <eos> what if i told you i'm not ?  <eos> go to hell", 'where i ll check <eos> why tomorrow ?  <eos>
 no i want now', "shall we meet <eos> you say- you're leaving soon . anywhere you wanna go before you head ?  <eos>  ?",
 "let's change the subject <eos> i just did it  . l .  <eos> you're broken", 'your  pic  pz <eos> thank you x☐d <eos> wo
']
Train Dataset
{'happy': {}, 'sad': {}, 'angry': {'1': 'when did i ?  <eos> saw many times i think -_- <eos> no .  i never saw you', '3
': 'u r ridiculous <eos> i might be ridiculous but i am telling the truth .  <eos> u little disgusting whore', '10': 'i
hate my boyfriend <eos> you got a boyfriend ?  <eos> yes'}, 'others': {'0': "don't worry  i'm girl <eos> hmm how do i kn
ow if you are <eos> what's ur name ? ", '2': 'by <eos> by google chrome <eos> where you live', '4': 'just for time pass
<eos> wt do u do 4 a living then <eos> maybe', '5': "i'm a dog person <eos> youre so rude <eos> whaaaat why", '6': 'so w
hatsup <eos> nothing much .  sitting sipping and watching tv .  how abt u ?  <eos> what are you watching on tv ? ', '7':
 'ok <eos> ok im back !  <eos> so ,  how are u', '8': 'really ?  <eos> really really really really really <eos> y saying
 so many times . i can hear you', '9': 'bay <eos> in the bay <eos> ðÿ˜˜ love you'}}
```

STEP 2: TOKENIZE

```
Train Matrix
[[ 0  0  0 ...  0  0  1]
 [ 0  0  0 ...  8  9 10]
 [ 0  0  0 ... 11  3  3]
 ...
 [ 0  0  0 ...  0 33 34]
 [ 0  0  0 ...  0  1 35]
 [ 0  0  0 ...  6 36 37]]

Label Matrix
[[0. 0. 1. 0.]
 [0. 0. 1. 0.]
 [0. 0. 1. 0.]
 [0. 0. 0. 1.]
 [0. 0. 0. 1.]
 [0. 0. 0. 1.]
 [0. 0. 0. 1.]
 [0. 0. 0. 1.]
 [0. 0. 0. 1.]
 [0. 0. 0. 1.]
 [0. 0. 0. 1.]]
Unique words: 37
Conversations: 11
```

TESTING: 2-FOLDS 2-EPOCHS

```
id      turn1   turn2   turn3   label
0       Then dont ask me        YOURE A GUY NOT AS IF YOU WOULD UNDERSTAND      IM NOT A GUY FUCK OFF   others
1       Mixed things  such as?? the things you do.      Have you seen minions?? others
2       Today I'm very happy    and I'm happy for you ♥ I will be marry others
3       Woah bring me some      left it there oops      Brb     others
4       it is thooooo   I said soon master.     he is pressuring me     others
5       Wont u ask my age??     hey at least I age well!        Can u tell me how can we get closer??   others
6       I said yes      What if I told you I'm not?     Go to hell      others
7       Where I ll check        why tomorrow?   No I want now   others
8       Shall we meet   you say- you're leaving soon...anywhere you wanna go before you head?   ?       others
9       Let's change the subject        I just did it .l.       You're broken   others
10      Your  pic  pz   thank you X-D   wc      others
```

METRICS: 2-FOLD 2-EPOCHS

```
Fold 2/2
Train on 6 samples, validate on 5 samples
Epoch 1/2
6/6 [==============================] - 1s 168ms/step - loss: 1.3873 - acc: 0.1667 - val_loss: 1
.0942 - val_acc: 0.8000
Epoch 2/2
6/6 [==============================] - 0s 12ms/step - loss: 0.9017 - acc: 0.6667 - val_loss: 0.
9003 - val_acc: 0.8000
Class happy : Precision : 0.000, Recall : nan, F1 : 0.000
Class sad : Precision : 0.000, Recall : nan, F1 : 0.000
Class angry : Precision : 0.200, Recall : 1.000, F1 : 0.333
Macro Precision : 0.0667, Macro Recall : nan, Macro F1 : 0.0000
Accuracy : 0.8000, Micro Precision : 0.0667, Micro Recall : 1.0000, Micro F1 : 0.1250

============= Metrics =================
Average Cross-Validation Accuracy : 0.7000
Average Cross-Validation Micro Precision : 0.1000
Average Cross-Validation Micro Recall : 1.0000
Average Cross-Validation Micro F1 : 0.1801
```

RESULTS

```
Fold 10/10
[Note:] Trainer ctor: 1 of the model parameters are not covered by any of the specified Learner
s; these parameters will not be learned
Train on 27144 samples, validate on 3016 samples
Epoch 1/10
27144/27144 [==============================] - 332s 12ms/step - loss: 0.8461 - acc: 0.6882 - va
l_loss: 0.7239 - val_acc: 0.7347
Epoch 2/10
27144/27144 [==============================] - 334s 12ms/step - loss: 0.6944 - acc: 0.7478 - va
l_loss: 0.6787 - val_acc: 0.7583
Epoch 3/10
27144/27144 [==============================] - 336s 12ms/step - loss: 0.6477 - acc: 0.7692 - va
l_loss: 0.6506 - val_acc: 0.7739
Epoch 4/10
27144/27144 [==============================] - 343s 13ms/step - loss: 0.6169 - acc: 0.7822 - va
l_loss: 0.6423 - val_acc: 0.7749
Epoch 5/10
27144/27144 [==============================] - 345s 13ms/step - loss: 0.5960 - acc: 0.7895 - va
l_loss: 0.6299 - val_acc: 0.7798
Epoch 6/10
27144/27144 [==============================] - 333s 12ms/step - loss: 0.5743 - acc: 0.7975 - va
l_loss: 0.6303 - val_acc: 0.7822
Epoch 7/10
27144/27144 [==============================] - 355s 13ms/step - loss: 0.5562 - acc: 0.8056 - va
l_loss: 0.6447 - val_acc: 0.7842
Epoch 8/10
27144/27144 [==============================] - 355s 13ms/step - loss: 0.5365 - acc: 0.8112 - va
l_loss: 0.6416 - val_acc: 0.7842
Epoch 9/10
27144/27144 [==============================] - 335s 12ms/step - loss: 0.5343 - acc: 0.8130 - va
l_loss: 0.6613 - val_acc: 0.7835
Epoch 10/10
27144/27144 [==============================] - 342s 13ms/step - loss: 0.5026 - acc: 0.8224 - va
l_loss: 0.6757 - val_acc: 0.7802
Class happy : Precision : 0.706, Recall : 0.712, F1 : 0.709
Class sad : Precision : 0.763, Recall : 0.655, F1 : 0.705
Class angry : Precision : 0.857, Recall : 0.744, F1 : 0.796
Macro Precision : 0.7755, Macro Recall : 0.7034, Macro F1 : 0.7377
Accuracy : 0.7802, Micro Precision : 0.7779, Micro Recall : 0.7017, Micro F1 : 0.7378
```

RESULT METRICS

- Recall: Fraction of docs I classified correctly
- Precision: Fraction of docs assigned class I that are class I
- Accuracy: Fraction of docs classified correctly

```
============= Metrics =================
Average Cross-Validation Accuracy : 0.7787
Average Cross-Validation Micro Precision : 0.7643
Average Cross-Validation Micro Recall : 0.7003
Average Cross-Validation Micro F1 : 0.7305
```

SOLUTION FILE

id	turn1	turn2	turn3	label
0	Then dont ask me	YOURE A GUY NOT AS IF YOU WOULD UNDERSTAND	IM NOT A GUY FUCK OFF	angry
1	Mixed things such as??	the things you do.	Have you seen minions??	others
2	Today I'm very happy	and I'm happy for you ♥	I will be marry	happy
3	Woah bring me some	left it there oops	Brb	others
4	it is thoooooo	I said soon master.	he is pressuring me	others
5	Wont u ask my age??	hey at least I age well!	Can u tell me how can we get closer??	others
6	I said yes	What if I told you I'm not?	Go to hell	angry
7	Where I ll check	why tomorrow?	No I want now	others
8	Shall we meet	you say- you're leaving soon...anywhere you wanna go before you head?	?	others
9	Let's change the subject	I just did it .l.	You're broken	sad
10	Your pic pz	thank you X□D	wc	others
11	not mine	done for the day ?	can my meet to sexy girl	others
12	I want to play the game if you just finished the game...	then you haven't finished the game......	#Emojisong	others
13	Iam sory	why sorry ! 😿	I insult you	others
14	How much	depends on how long your internet has been out!!!	U have bf	others
15	Ok	Thank you. xD	What about cortanan	others
16	So the story?	yeah indeed 😹	Tomorrow probably	others
17	May be	yeaa i hope soo!!	Can you do complex calculations	others
18	So come on na.. Want u so badly.	now you're tempting me to.	So why are you still away from my body??	sad

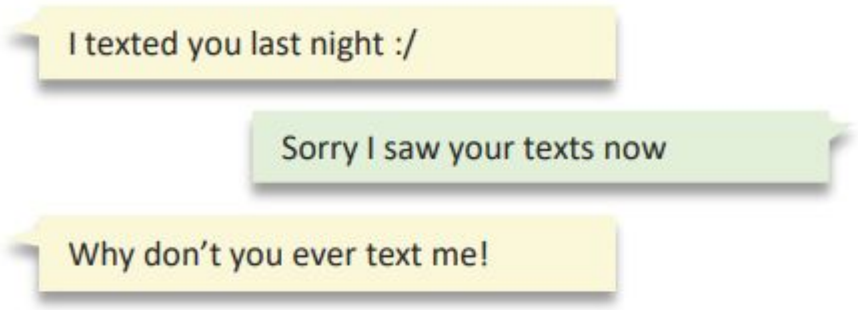
Questions?

Contextual Emotion Detection in Text Using Message Embeddings

Justin Lovelace

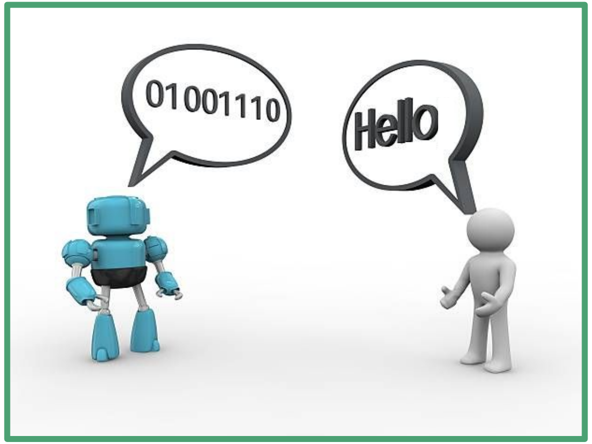
Task Description

- Given three turns of textual dialogue, classify the sentiment of the final message
- Classify as either happy, sad, angry, or other



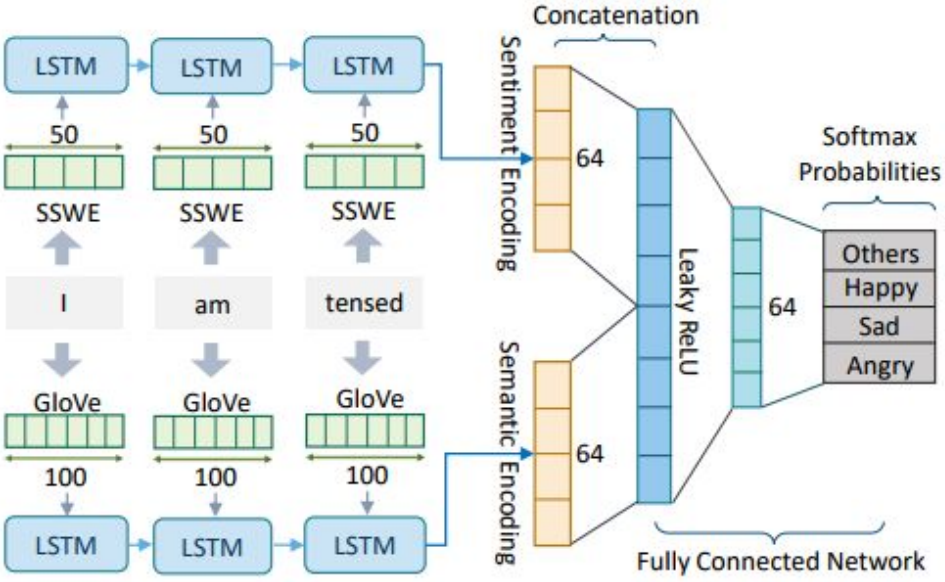
Dataset

- Conversations between a human and a bot collected from Twitter
- Training set: 456k utterances in the Others category, 28k in the Happy category, 34k in the Sad category , and 36k in the Angry category
- Happy, Sad, and Angry messages are more sparse in the test set



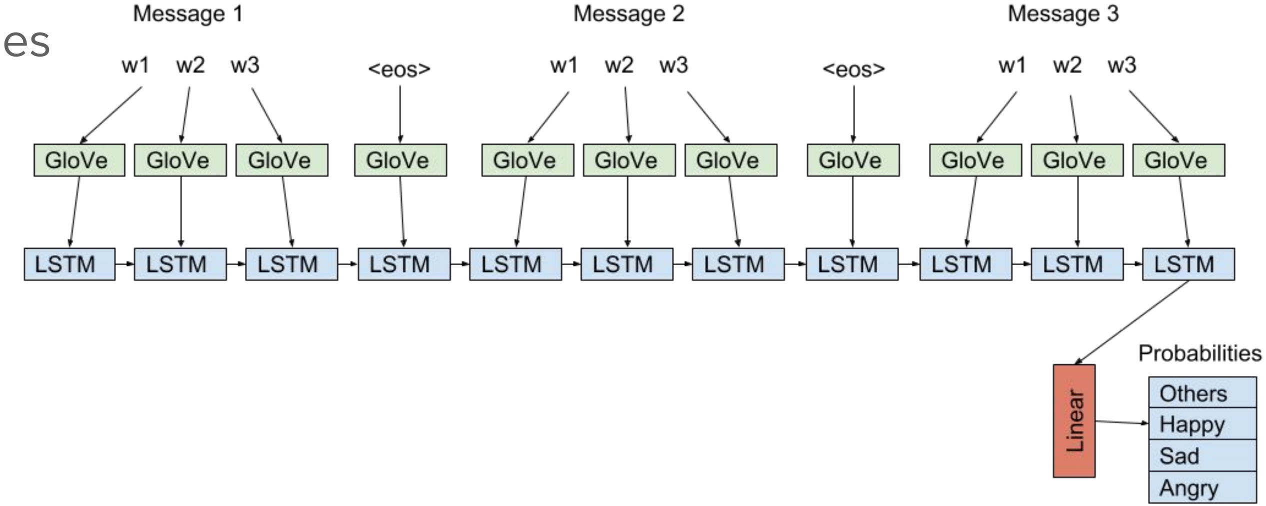
Prior Work

- Organizers have performed some preliminary work
- Developed and tested a number of baselines
- Found that deep learning models outperformed traditional models such as Naive Bayes, SVM, etc



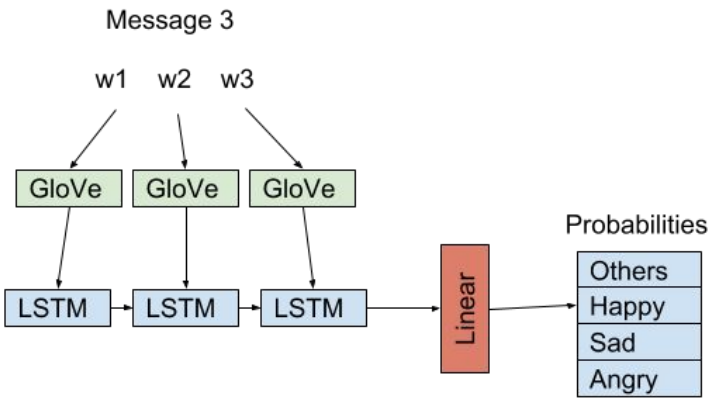
My Baseline

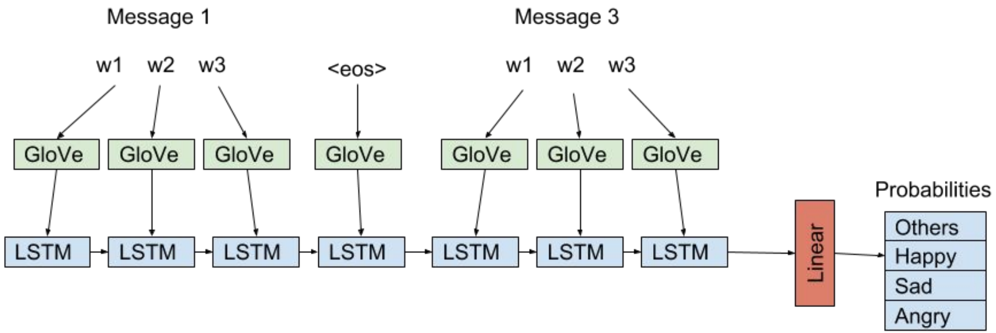
- Decided to start from the strongest baseline developed by organizers
  - Sequential LSTM using GloVe embeddings
- Perform preprocessing similar to that of organizers
  - Normalize punctuation (“!!!” becomes “ ! “) and emojis (“😊😊😊” becomes “ 😊 ”)
  - Lowercase the dialogue
- Concatenate messages with <eos> tag and feed to LSTMs



Evaluating Importance of Prior Messages

- Wanted to evaluate how important the context was for performance
- Tested three different baselines using different contexts
    - Concatenating all three messages with <eos> tag as described previously
    - Using only the final message for classification
    - Using only the human messages
        - Concatenating the first and third messages using an <eos> tag





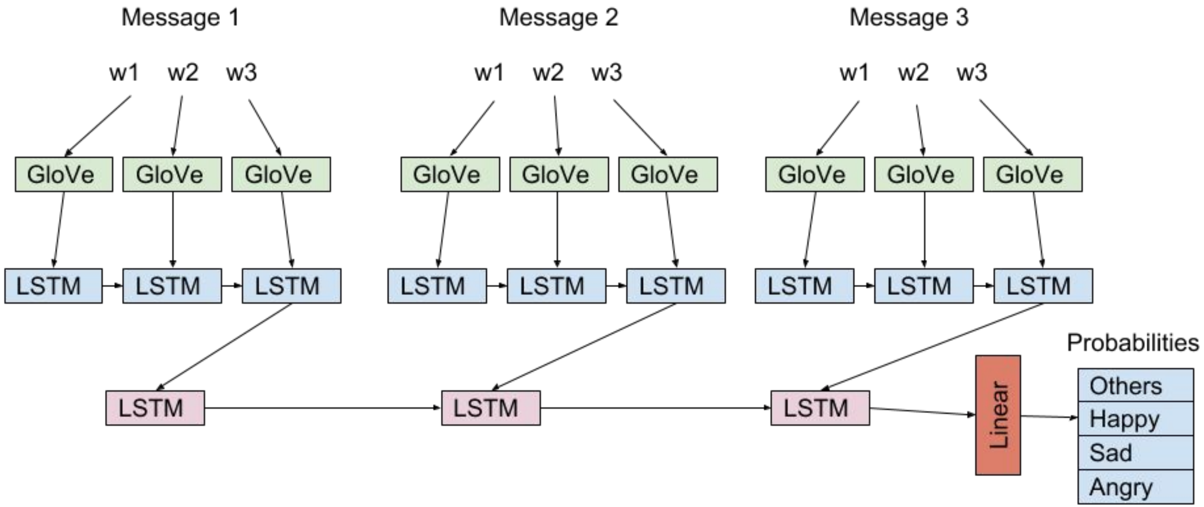
Baseline Results

- As expected context is important for accuracy
  - Although removing the second turn only leads to a minor loss of accuracy
  - Decide to include all turns for future models

	Happy			**Sad**			**Angry**			
	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1	Micro-Averaged F1
Full Context	48.65	63.38	**55.05**	47.92	73.60	58.04	50.86	78.67	61.78	**58.48**
No Context	34.26	69.72	45.94	46.86	65.60	54.67	45.87	74.00	56.63	52.00
Human Context	43.32	66.20	52.37	52.17	67.20	**58.74**	53.92	78.00	**63.76**	58.30

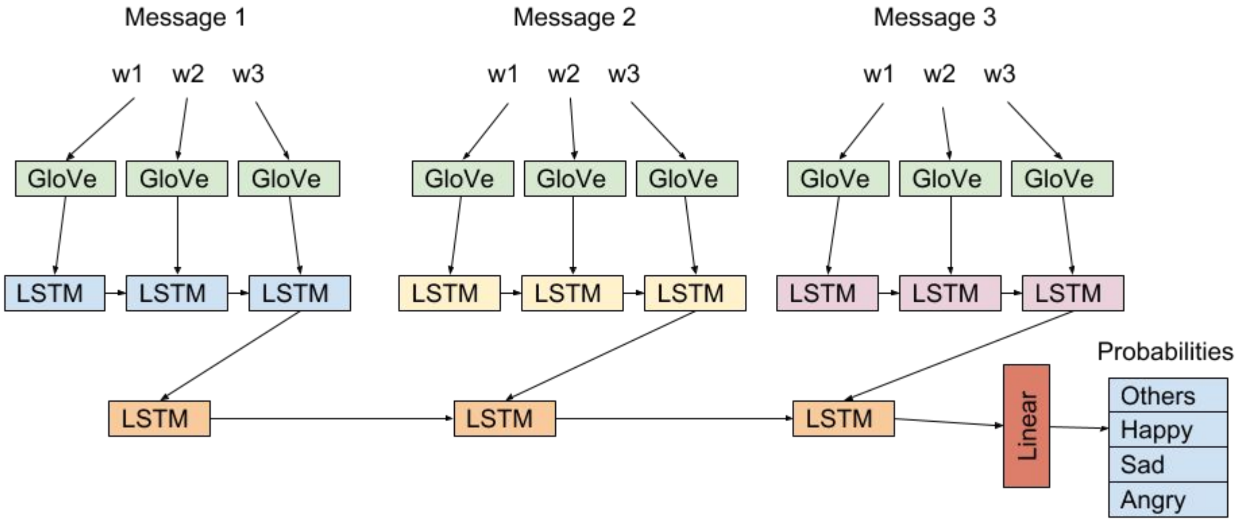
Proposed Improvement

- Develop a model that more accurately represents the three distinct turns of conversation
- Pass each message through an LSTM to construct a message embedding
- Pass message embeddings through a second LSTM layer to make final prediction



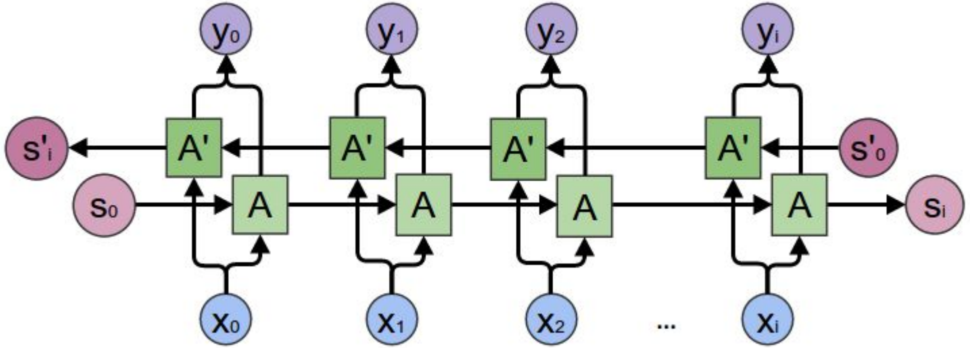
Proposed Improvement

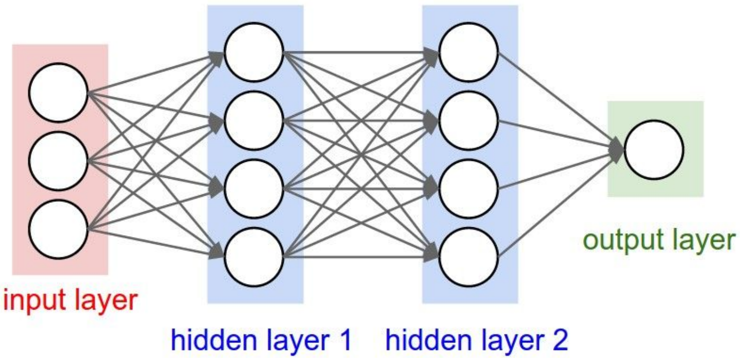
- Extend the previous model to differentiate between each turn
- Pass each message to a different LSTM to generate message embedding
- Learn turn-specific relationships within the data



Further Experimentation

- Experimented with augmenting the two previous models
- Tried using bidirectional LSTMs and adding dense layers
- All configurations tested degraded performance slightly
    - Although they still outperformed the baseline
- Decided to stick with original configurations





Final Results

- Both of my proposed models outperformed baseline on F1 measure for all classes
- First and second model improved upon the micro-averaged F1 score by 4.07 and 4.90 respectively

	Happy			**Sad**			**Angry**			
	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1	Micro-Averaged F1
Baseline	48.65	63.38	55.05	47.92	73.60	58.04	50.86	78.67	61.78	58.48
Model 1	45.79	69.01	55.06	60.54	71.20	**65.44**	58.10	81.33	67.78	62.55
Model 2	46.86	68.31	**55.59**	54.27	71.20	61.59	65.78	82.00	**73.00**	**63.38**

Manual Review

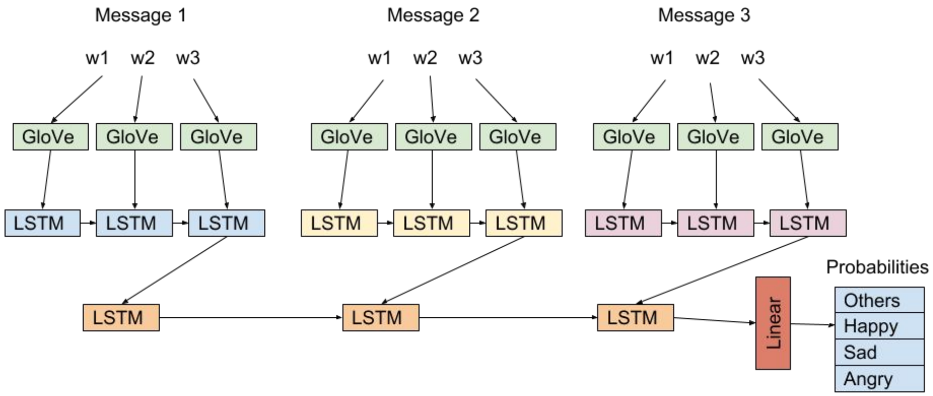
Message Number	Turn 1	Turn 2	Turn 3	Predicted Label
1	I was waiting you, how are you ?	I'm good! How are you?	nice thks	happy
2	yes, you broke my heart	yes you lost me :-) be happy	💔💔💔💔💔💔💔💔💔💔💔💔💔💔💔💔💔💔💔💔💔💔💖💖💖💖💔💔💔💔💔💔💔💔💔💔💔💔💔💔💔💔💔💔💔💔	sad
3	U r my lifee	I'm not eww you're confusing me with you	U r stupid	angry
4	I think what about Zombie Apocalypse	u are one of the zombie	I m not a zonbie.. But actually a dracula	others

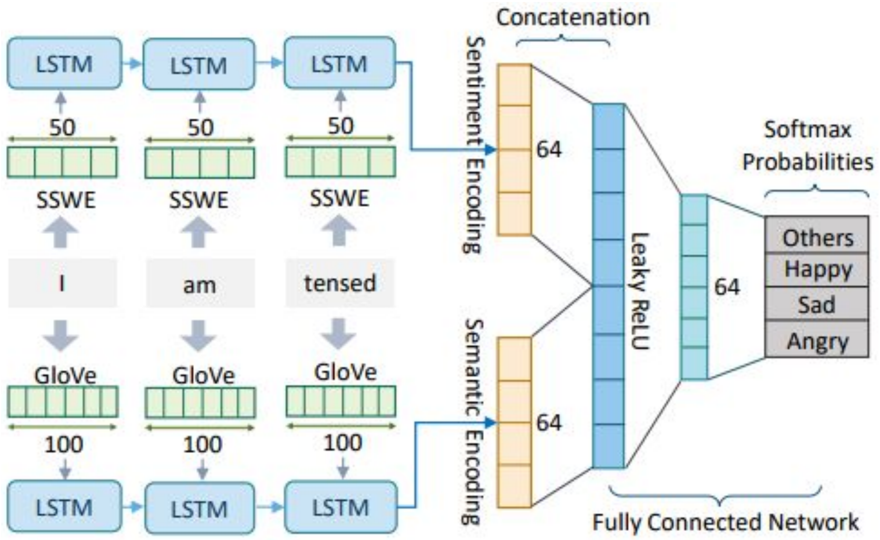
Conclusions

- Context of earlier messages is useful for classifying the sentiment of the third message
  - Demonstrated by work with baseline
- Developing message embeddings for each individual message further improves the classification performance
- There appear to be turn specific trends that can be leveraged to improve classification

Possible Future Work

- More sophisticated preprocessing
  - Automatic spelling correction
- Use both GloVe and SSWE embeddings as the organizers did





Questions?

Contextual Emotion Detection in Text

Christopher Rech

Background

- Sem-Eval 2019 Task 3, hosted on CodaLab
- Training and Dev data sets provided
- Participants ranked by accuracy of predictions

Problem Statement

- Determine emotional sentiment of a conversation
  - 4 different emotions: “happy”, “Sad”, “Angry”, “others”
- Each sample consists of three turns of a conversation
  - Each turn is one line from a conversation
- Accuracy is measured by testing output against actual result
  - CodaLab system automatically scores submission

Data Preparation

- Data provided as CSV, converted to dataframe
  - Split into Train (80%) and Test (20%) sets
- Turns are tokenized into individual words
  - Non-alphabetical characters treated as individual words
- Lookup data structure keeps track of words
  - Each word has a probability for each category

Strategy

- Naïve Bayes
  - Score each sample for all categories based on individual words
- Category Matching
  - Calculate probability for each category, sort by highest value
  - Assign samples to categories until quota is reached
- Turn Weighting
  - Weigh turns based on optimizing simulated results

Observations

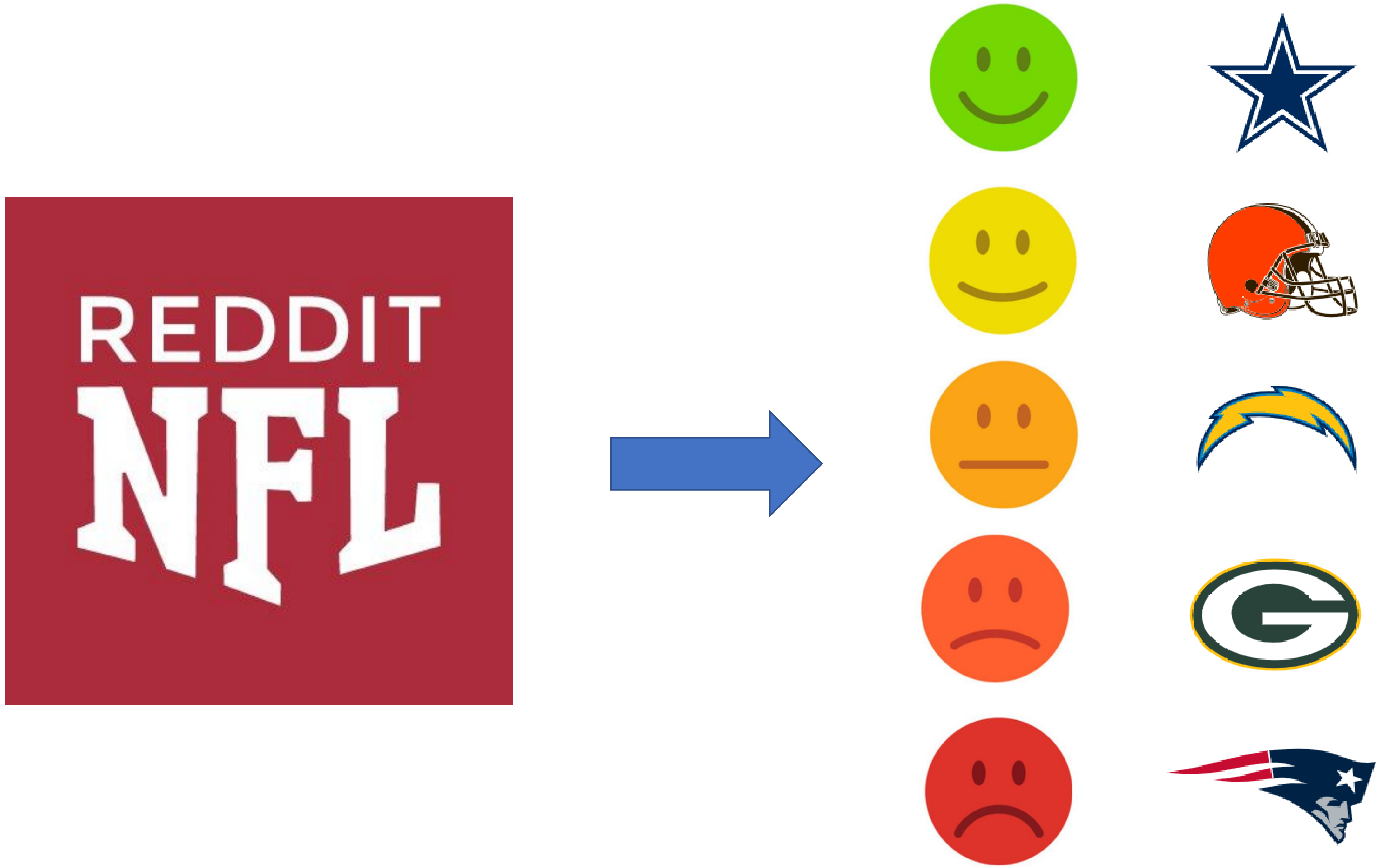
- Importance of non-alphabetical characters
  - Certain emojis were strongly associated with specific emotions
  - 21% of highly significant words were emojis
- Varied significance of turn
  - Final turn of conversations was most important, then first
  - Turn weighting produced significant accuracy improvements

Results

- Evaluation is a numeric score between 0 and 1
  - Considers true positives, false positives, false negatives
  - Harmonic mean of overall accuracy and precision
- Current placement: 54/171
  - Top third of participants with submission to CodaLab

NFL Team Sentiments

Bailey Guthrie


REDDIT
NFL

Example Reddit Layout

(+4) Thread 1

- (+7) Comment 1
- (+3) Comment 2
- (-4) Comment 3

(+2) Thread 2

- (+4) Comment 1
- (+1) Comment 2

Example Comment

The browns are really stepping it up this year while the jags have taken a big step back and look much worse than they ever did last year.

Issues:

- Refers to the Jacksonville Jaguars as the “jags”.
- Talks about two different two in different tones.

Team Alias Table

Jacksonville Jaguars	jacksonville	jaguars	jags	
Cleveland Browns	cleveland	browns		
Pittsburg Steelers	pittsburg	Steelers	pitt	
. . .	. . .	. . .	. . .	. . .
Tampa Bay Buccaneers	tampa bay	buccaneers	bucs	buccs

Comment Sentiment Segmenting

The [browns are really stepping it up this year while the] [jags have taken a big step back and look much worse than they ever did last year.]

From week to week the [saints keep improving and looking more like the best team in the league.]

Browns: +0.5

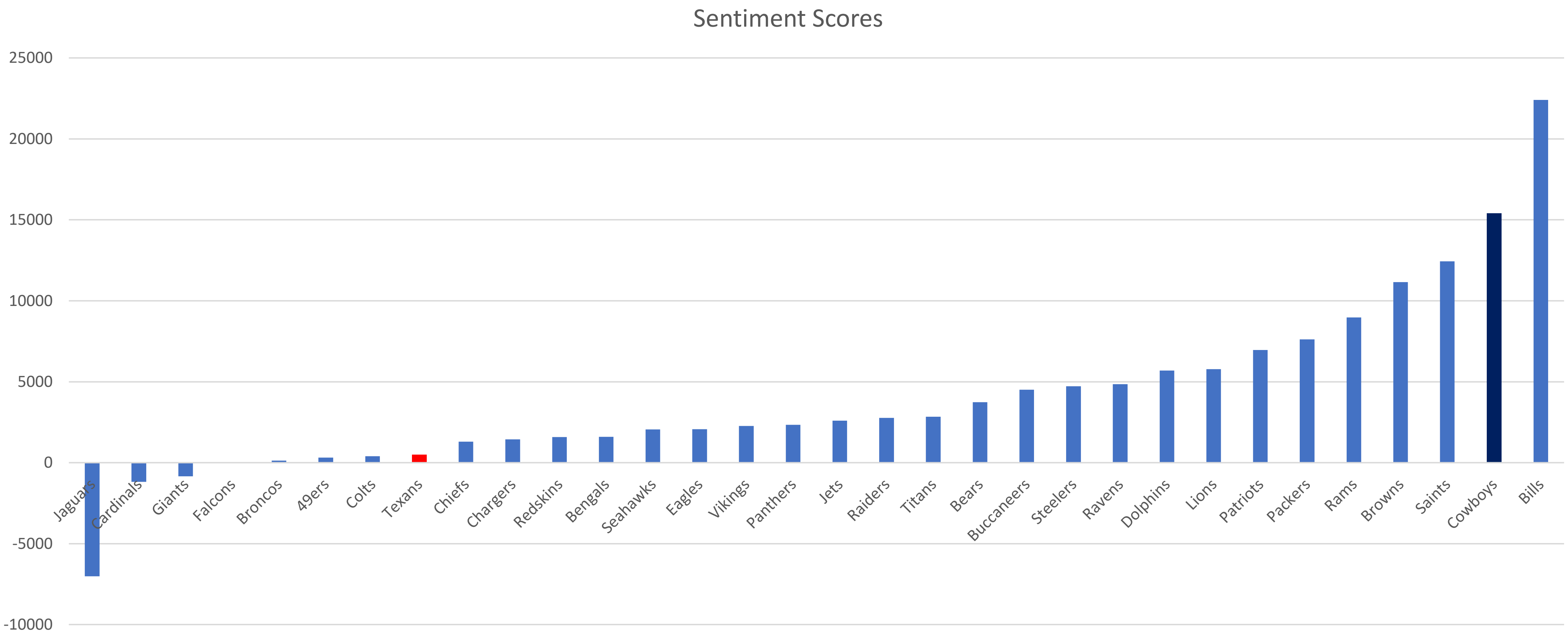
Jaguars: -0.7

Saints: +0.8

Results

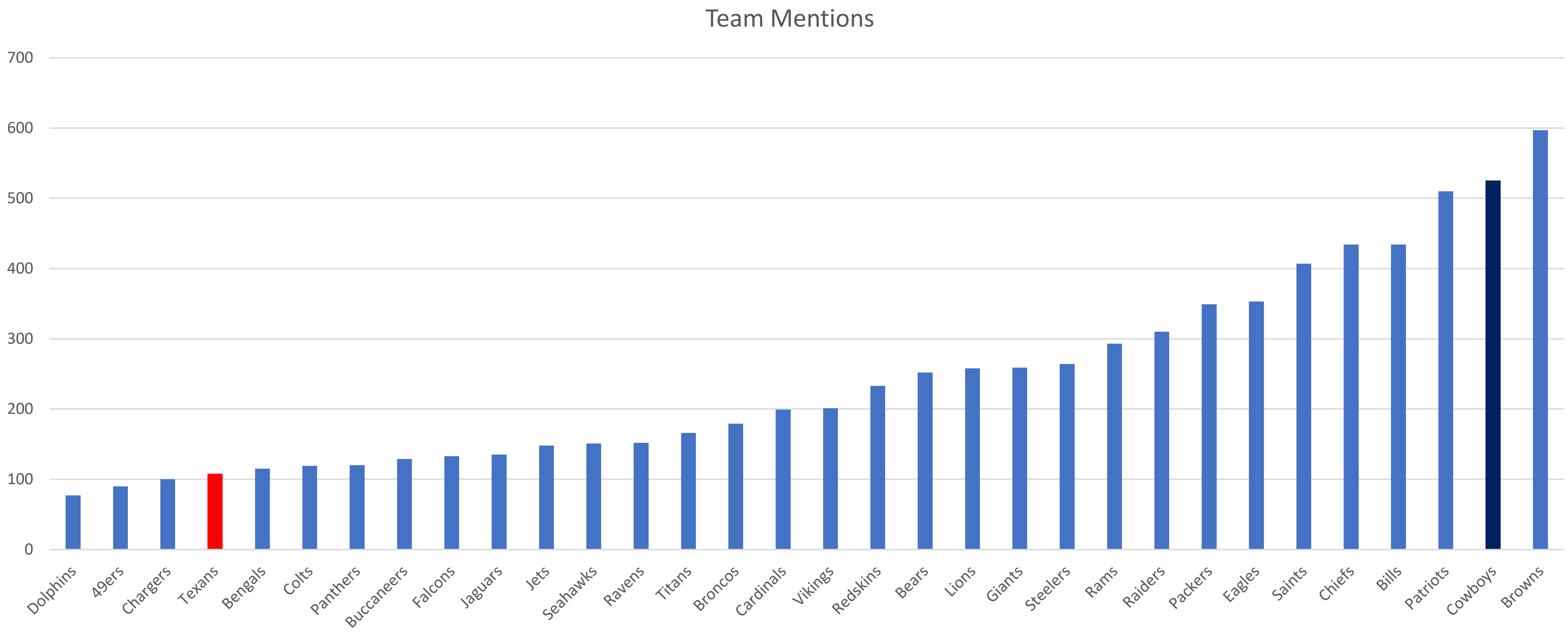
Team	Score	Mentions	Score / Mentions	Win Percentage
Jaguars	-7008.64	135	-51.92	0.273
Cardinals	-1185.8	199	-5.96	0.182
Giants	-833.48	259	-3.22	0.273
Falcons	14.05	133	0.11	0.364
Broncos	137.49	179	0.77	0.4
49ers	309.8	90	3.44	0.182
Colts	405.44	119	3.41	0.5
Texans	500.24	108	4.63	0.7
Chiefs	1300.28	434	3.00	0.818
Chargers	1440.68	100	14.41	0.727
Redskins	1579.74	233	6.78	0.545
Bengals	1596.65	115	13.88	0.455
Seahawks	2056.45	151	13.62	0.545
Eagles	2068.31	353	5.86	0.455
Vikings	2270	201	11.29	0.55
Panthers	2340.59	120	19.50	0.545
Jets	2593.93	148	17.53	0.273
Raiders	2769.66	310	8.93	0.182
Titans	2837.14	166	17.09	0.5
Bears	3742.7	252	14.85	0.727
Buccaneers	4505.08	129	34.92	0.364
Steelers	4728.41	264	17.91	0.75
Ravens	4848.43	152	31.90	0.545
Dolphins	5690.37	77	73.90	0.5
Lions	5782.83	258	22.41	0.364
Patriots	6965.77	510	13.66	0.727
Packers	7621.95	349	21.84	0.45
Rams	8967.73	293	30.61	0.909
Browns	11154.22	597	18.68	0.409
Saints	12438.02	407	30.56	0.909
Cowboys	15399.03	525	29.33	0.545
Bills	22410.52	434	51.64	0.364

Results

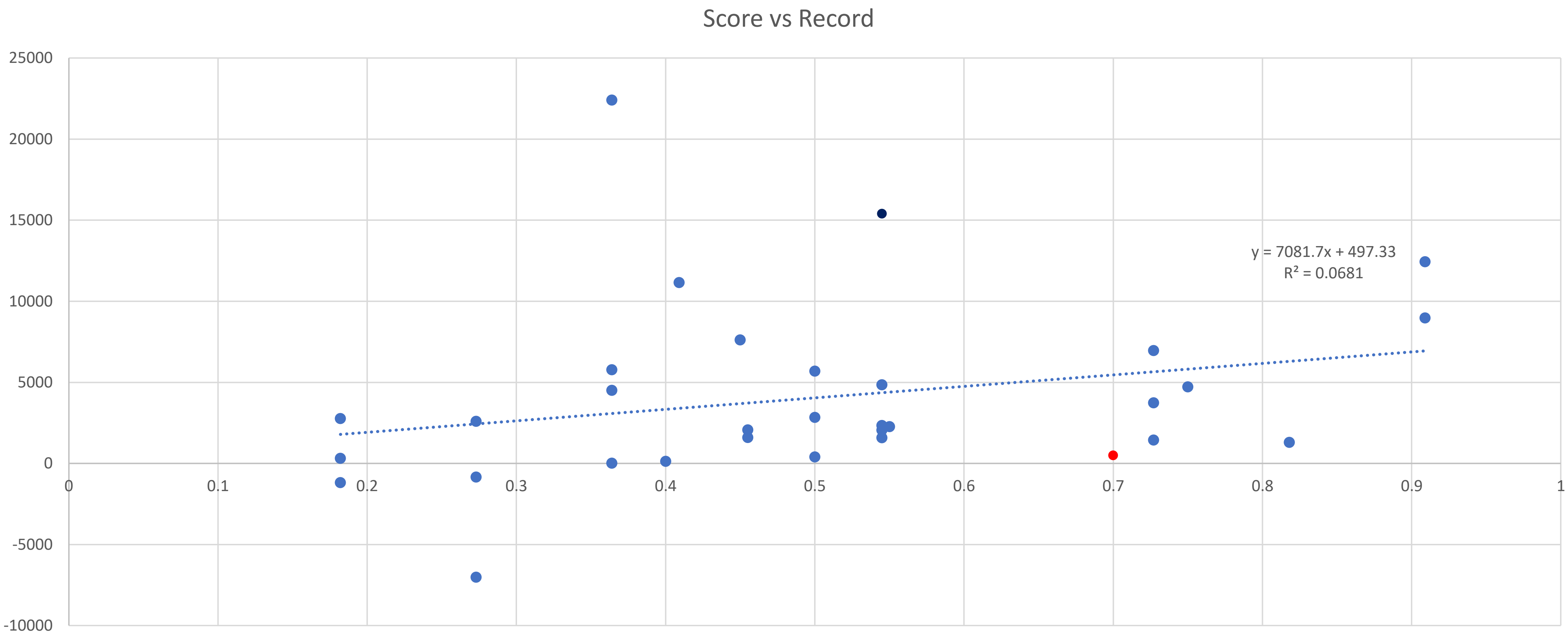


Sentiment Scores
25000
20000
15000
10000
5000
0
-5000
-10000
Jaguars
Cardinals
Giants
Falcons
Broncos
49ers
Colts
Texans
Chiefs
Chargers
Redskins
Bengals
Seahawks
Eagles
Vikings
Panthers
Jets
Raiders
Titans
Bears
Buccaneers
Steelers
Ravens
Dolphins
Lions
Patriots
Packers
Rams
Browns
Saints
Cowboys
Bills

Results

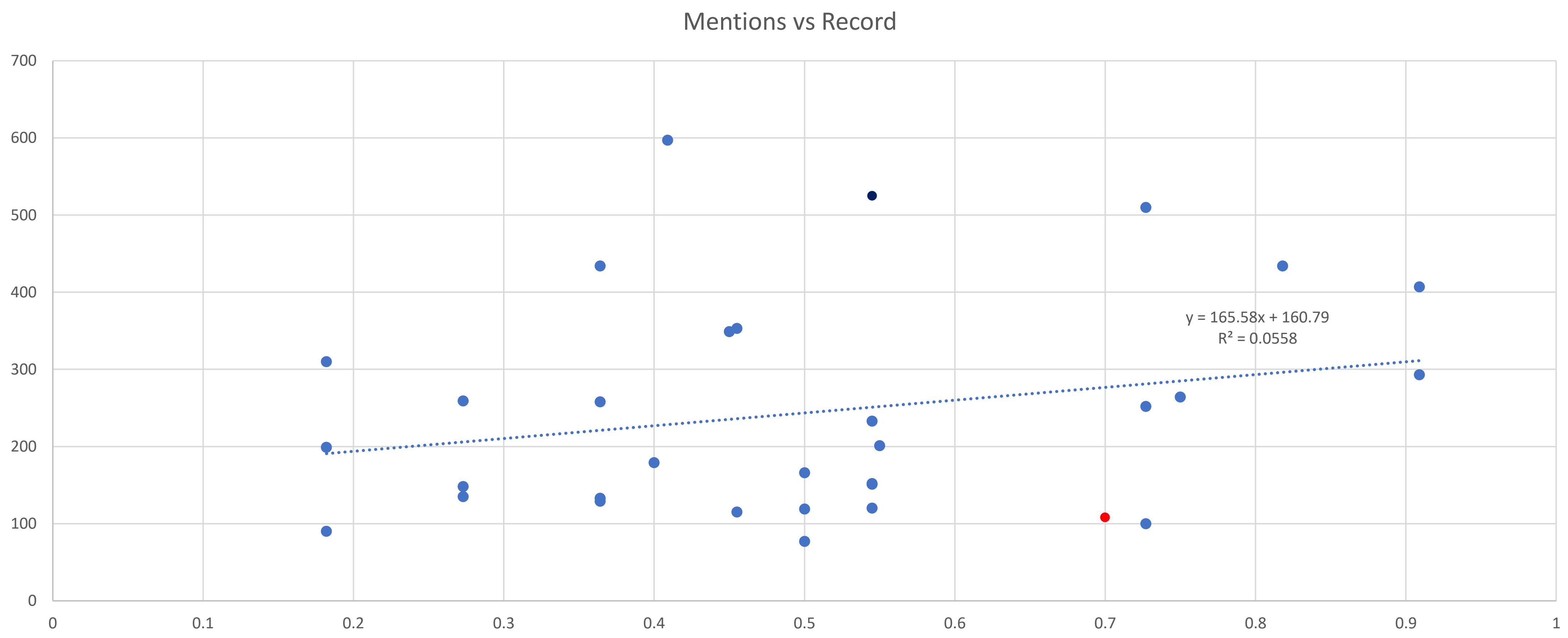

Team Mentions
700
600
500
400
300
200
100
0
Dolphins
49ers
Chargers
Texans
Bengals
Colts
Panthers
Buccaneers
Falcons
Jaguars
Jets
Seahawks
Ravens
Titans
Broncos
Cardinals
Vikings
Redskins
Bears
Lions
Giants
Steelers
Rams
Raiders
Packers
Eagles
Saints
Chiefs
Bills
Patriots
Cowboys
Browns

Results



Score vs Record
25000
20000
15000
10000
5000
0
-5000
-10000
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
y = 7081.7x + 497.33
R² = 0.0681

Results


Mentions vs Record
700
600
500
400
300
200
100
0
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
y = 165.58x + 160.79
R² = 0.0558

WORLD CUP TWEETS SENTIMENT ANALYSIS

Eubert Almenar and Akintunde Adegboye

BACKGROUND AND TASK

- The 2018 World Cup occurred this past summer and Twitter was blowing up with tweets about anything and everything related to football, evoking different emotions depending on the event
- Task – Sentiment Analysis
  - As we know, sentiment analysis is the process of reading, identifying and classifying data (tweets in this scenario) into different polarities. Since Naïve-Bayes was used in the first assignment, another method, namely SVMs, could be used for this task. The goal was to see how the different SVM kernels compare to each other
  - Accuracy Score and Report with precision, recall, f1-score, and support are to be displayed
  - Naïve-Bayes was also implemented as a comparison to the SVMs

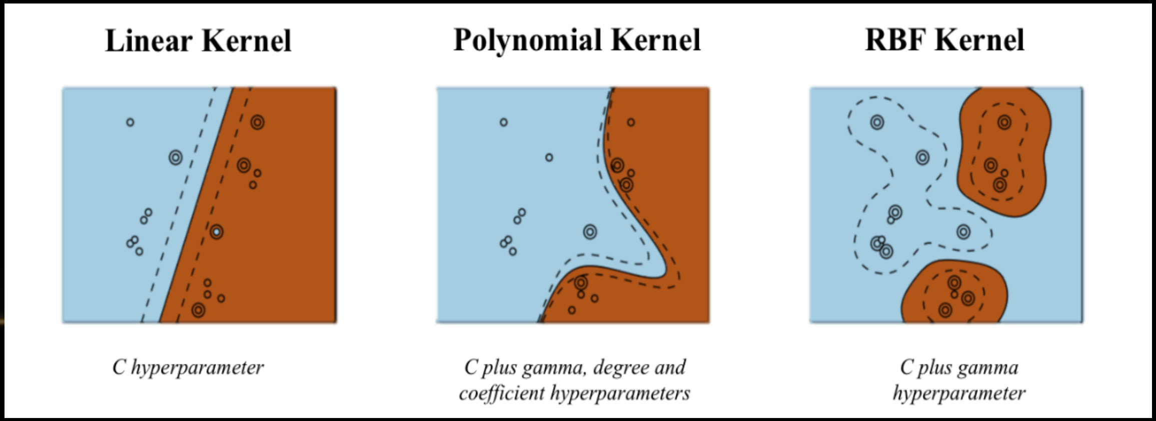
PROCESS

- Gather tweets with positive and negative polarity relating to the World Cup
- Label them as positive and negative and use as train data
- Gather a smaller set of tweets for testing data
- Label them as positive and negative and use as test data
- Make a prediction on a given tweet as to whether or not it is positive or negative using the training data
- Print out the accuracy score for each of the SVM Kernels and Naïve-Bayes based on the test data labels and the prediction from the training data
- Print out the reports with precision, recall, f1-score, and support for each of the SVM Kernels and Naïve-Bayes based on the test data labels and the prediction from the training data

SUPPORT VECTOR MACHINES

- SVMs or Support Vector Machines are usually used to classify a set of elements into two groups
  - Positive and Negative polarity makes sense for the base purpose of SVMs
- SVMs can have different kernels associated with them, rather than just a soft-margin/hard-margin (linear), these could be applied:
  - RBF -> Radial Basis Function
  - Polynomial
  - Sigmoid
  - Etc…

$$K(x_i, x_j) = \langle \phi(x_i), \phi(x_j) \rangle$$



LINEAR SVM

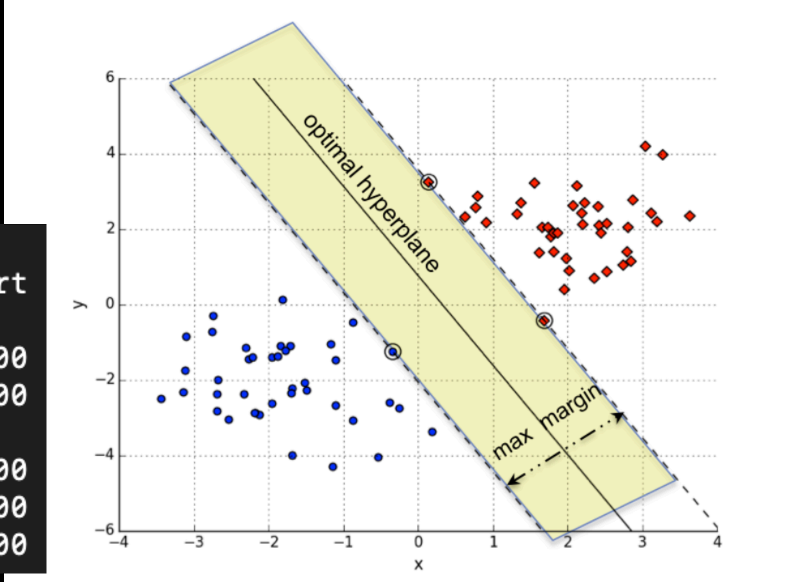
- A linear SVM comprises of a plane that splits the data at some hyperplane
  - Optimally, this hyperplane should have the max-margin distance to the closest point in both classes of data

- The problem arises when some of one class' members may be closer than expected to the other class' members
  - In the given image, if a blue point was somehow classified wrong and put closer to the red point, a hard-margin linear SVM would calculate the hyperplane with that outlier in consideration

$$k(\mathbf{x}_1, \mathbf{x}_2) = \mathbf{x}_1 \cdot \mathbf{x}_2$$

SVM Linear Kernel Accuracy: 0.945

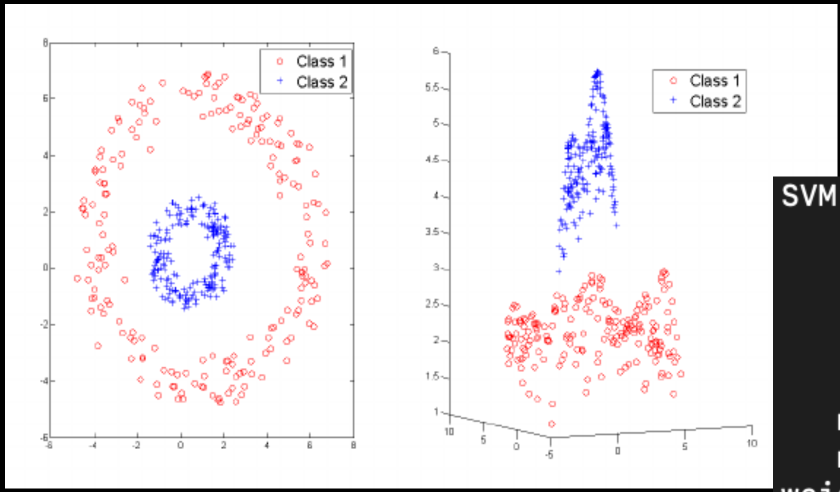
	precision	recall	f1-score	support
negative	0.95	0.94	0.94	100
positive	0.94	0.95	0.95	100
micro avg	0.94	0.94	0.94	200
macro avg	0.95	0.94	0.94	200
weighted avg	0.95	0.94	0.94	200



SVM WITH RBF KERNEL

- RBF, or Radial Basis Function Kernel changes the basis to where the data can be linearly separated when originally it was a non-linear set of data
  - In the image, it is seen how the data could be looked at from a different perspective when this kernel is applied, basically a higher dimensional space is bound to occur

- This should improve the classifier, especially when more data is being trained and tested, outliers may not have as drastic of an effect



$$K(x_i, x_j) = \exp(-\gamma \|x_i - x_j\|^2)$$

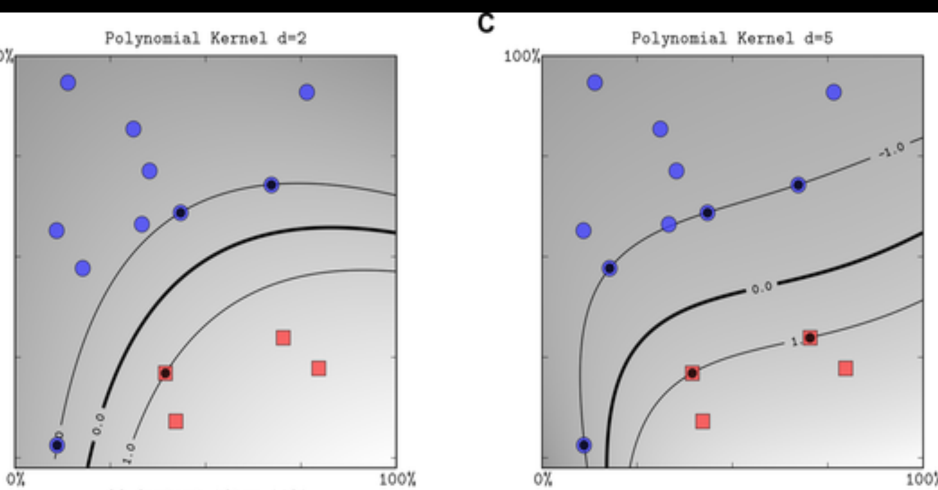
```
SVM RBF Kernel Accuracy: 0.975
              precision    recall  f1-score   support

    negative       0.96      0.99      0.98       100
    positive       0.99      0.96      0.97       100

   micro avg       0.97      0.97      0.97       200
   macro avg       0.98      0.97      0.97       200
weighted avg       0.98      0.97      0.97       200
```

SVM WITH POLYNOMIAL KERNEL

- Using the polynomial kernel requires the addition of degree d, as seen in the kernel function below
  - For example, a change to degree 2 would result in a quadratic kernel
  - Similar to the linear SVM kernel, but a constant c could be added if it's inhomogeneous, c = 0 means it's homogeneous
  - The previously linear hard-margin can now be looked at as a non-linear task, solely by applying a degree to the kernel



$$K(x_i, x_j) = (\gamma x_i^T x_j + c)^d$$

SVM Polynomial Kernel With Degree 2 Accuracy: 0.94

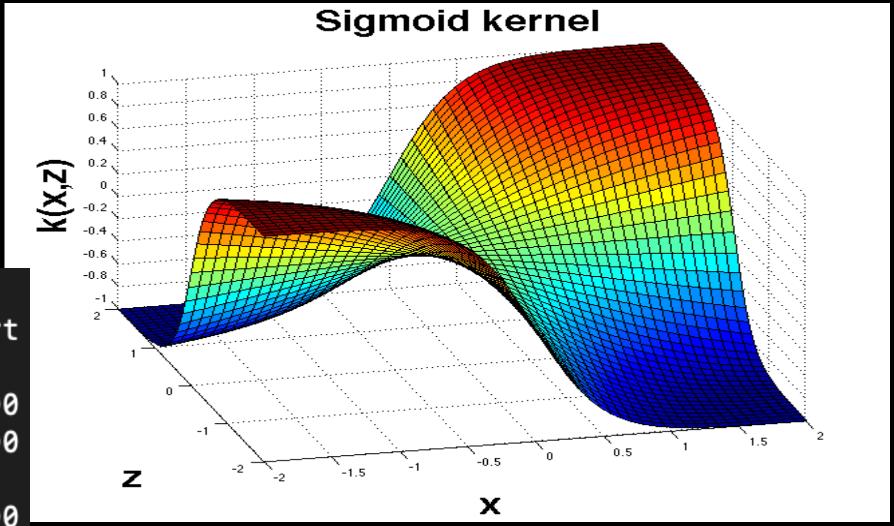
	precision	recall	f1-score	support
negative	0.95	0.93	0.94	100
positive	0.93	0.95	0.94	100
micro avg	0.94	0.94	0.94	200
macro avg	0.94	0.94	0.94	200
weighted avg	0.94	0.94	0.94	200

SVM Polynomial Kernel With Degree 5 Accuracy: 0.96

	precision	recall	f1-score	support
negative	0.95	0.97	0.96	100
positive	0.97	0.95	0.96	100
micro avg	0.96	0.96	0.96	200
macro avg	0.96	0.96	0.96	200
weighted avg	0.96	0.96	0.96	200

SVM WITH SIGMOID KERNEL

- A sigmoid function has a clear S-looking curve or sigmoid curve
  - There are many sigmoid functions, but the one used a lot in SVM kernels is the hyperbolic tangent function as detailed below
  - The constant c is adjustable as well as the slope, which is a coefficient similarly used in the RBF and polynomial kernels, and like RBF, works well with non-linear classification
  - The sigmoid kernel is similar to the sigmoid function in logistic regression



$$K(x_i, x_j) = \tanh(\gamma x_i^T x_j + c)$$

```
SVM Sigmoid Kernel Accuracy: 0.945
              precision    recall  f1-score   support

    negative       0.95      0.94      0.94       100
    positive       0.94      0.95      0.95       100

   micro avg       0.94      0.94      0.94       200
   macro avg       0.95      0.94      0.94       200
weighted avg       0.95      0.94      0.94       200
```

THINGS THAT COULD BE IMPROVED/ ALTERED

- Multi-Class SVM
  - Inherently, SVMs are binary classifiers, so to accomplish this we could use One vs. rest (training one classifier per class, and that class' samples are positives, while the others are negative) or One vs. One (training a separate classifier for each different pair of labels)
  - Neutrality -> Positive, Negative, Neutral
  - Emotions -> Happy, Sad, Angry, Concerned, etc.
    - Because Twitter is a social media platform that is used to express emotion about a large number of topics, emotions could be more useful than just Positive or Negative
- Add More Documents (Tweets)
  - Since tweets are short forms of text, the accuracy scores may be higher since each document is so small and it's easier to predict the polarity of a tweet than something such as a long movie or product review which would have a lot more variability in words

CONCLUSION

- SVMs are an effective method for classification of these tweets, or for any document/data
- RBF Kernel worked best for us and is normally associated with SVMs the most out of the other kernels
- As obvious as it is, the different kernel functions used as the parameters can greatly affect how accurate the classification is, and the more data we use, the better the kernels will more than likely perform as opposed to a Linear (Hard-Margin) SVM or the Naïve-Bayes Classifier
- SVMs are versatile because of these kernels, and could be in more real world scenarios than just sentiment analysis on tweets, and obviously they do have a place in machine learning

Sentiment Analysis of Bitcoin Tweets

Nandan Gade
CSCE 489 - 500

Abstract

- Prediction of future trend or forecasting
  - Make future investments
  - Analysis as a portfolio manager
- Bitcoin markets
- Real time Twitter data using APIs
- Ultimate goal: prove whether classification using Sentiment analysis can be advantageous to crypto coin trading strategies

Introduction

- Cryptocurrency: alternative medium of exchange consisting of numerous decentralized crypto coin types
- Bitcoin
  - Became a digital commodity of interest
  - Exchange rates of cryptocurrencies being volatile and rapidly changing
- Advantages of using Twitter data:
  - Easy access to most recent data
  - Can easily be obtained through a crawler or already existing datasets(Kaggle etc.)
  - Information regarding user, time stamp, hashtags etc. along with actual tweet

Introduction cont.

- Analysis:
  - Natural Language Processing in specific → sentiment analysis
  - Analysis on positive, negative, and neutral tweets
  - Frequency charts showing most occurring hashtag and distribution of sentiments
- Classifier:
  - Multinomial Naïve Bayes through NLTK
  - Logistic Regression through NLTK

Data

- API to crawl Twitter to get tweets → too much effort so an already existing dataset was used
- Cleaning hashtags:
  - Regex to remove unnecessary characters such as [,],# etc.
  - Convert to lowercase to make sure ‘Bitcoin’ equals ‘bitcoin’
  - English stopwords were removed
  - Tokenizer

Date	Tweet	User	Retweets		Hashtags		Sentiment
Fri Mar 23	RT @ALX	myresumer	16522	0	[]	<a href="h	['neutral']
Fri Mar 23	@lopp @_	BitMocro	1295	0	[u'Bitcoin']	<a href="h	['neutral']
Fri Mar 23	RT @tippe	hojachotop	6090	0	[u'blockch	<a href="h	['positive']
Fri Mar 23	free coins	denies_dis	2626	0	[]	<a href="h	['positive']

Method

- Text classification using sentiment analysis using Naïve Bayes
- Maybe even predict price of Bitcoin over a set time frame?
- More about the Naïve Bayes:

$$\underset{y_j}{\operatorname{argmax}}\; P(Y = y_j) \prod_{i=1}^{m} P(x_i|Y = y_j)$$

  - Generative learning algorithm (using probability)
  - Modeled Multinomial distribution rather than by Bernoulli distribution(binary values)
  - y_j represents the classification of whether the Bitcoin price is increasing or decreasing over a predetermined time interval
  - variable x_i is the feature vector for tweet i where a total of m tweets are collected

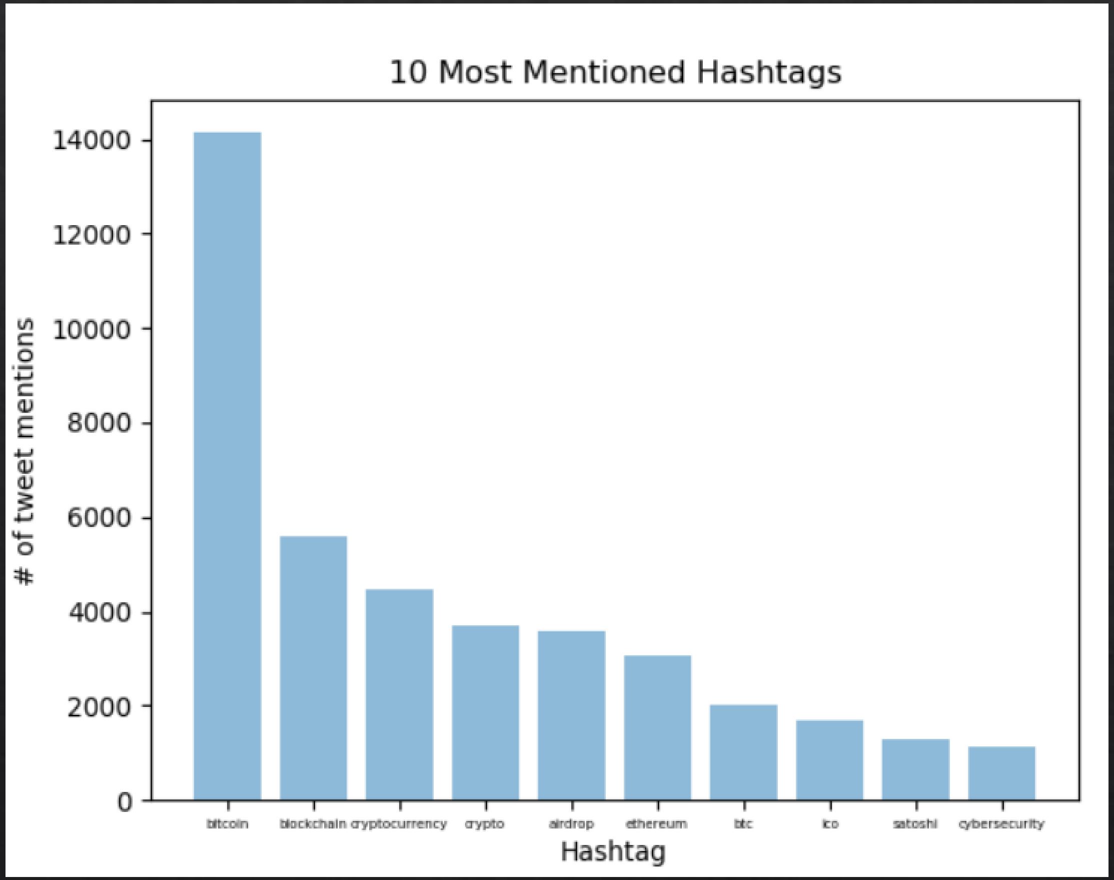
Method cont.

- More about Logistic Regression:
  - Discriminative learning algorithm
  - Examines two classes in the training set and determines the best separation
- Training and testing:
  - Partitioned data into normal 70-30 spliti (training and testing)
  - Equal representation of tweets in both
  - Record rate at which samples are classified correctly with respect to test set
  - Accuracy of classifier is used as a metric to measure performance

```
['positive']    22937
['neutral']     21939
['negative']     5983
```

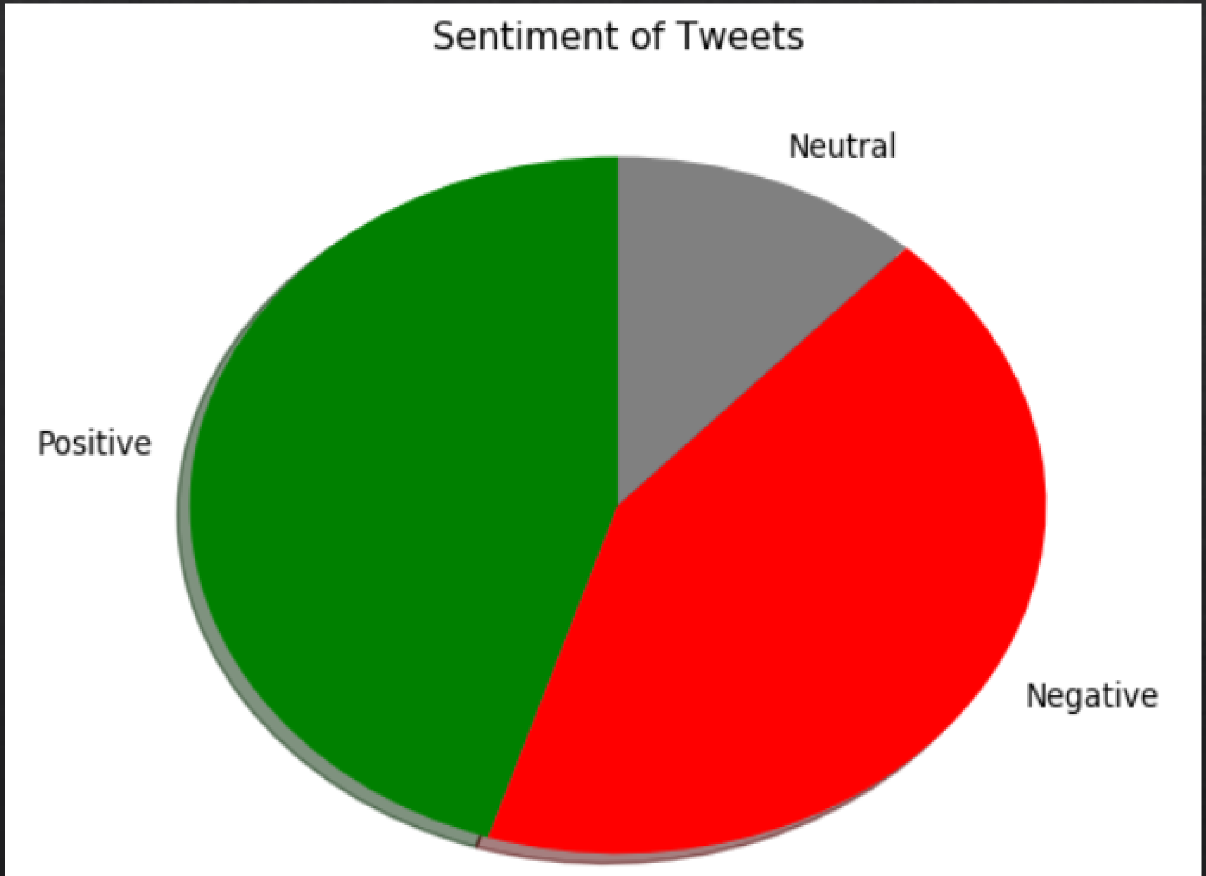
Sentiment count

Results



Most occurring hashtag

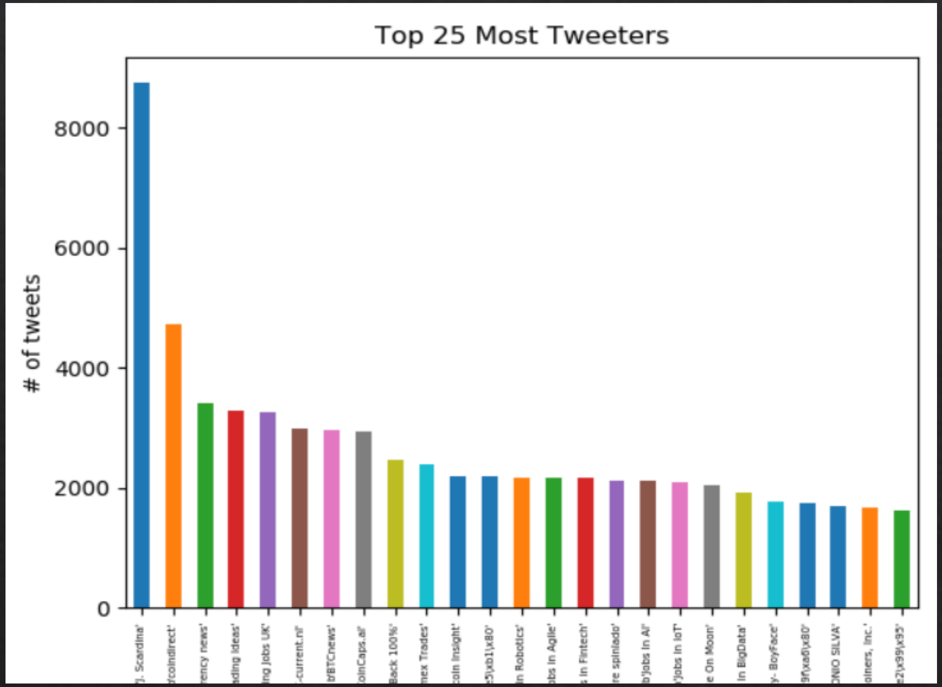
shows which topic is trending under cryptocurrency i.e shows current areas of interest



Sentiment distribution

for developer to see if there is equal distribution of tweets

Results cont.



Most tweeting user

shows top 25 most active user i.e. an individual with either knowledge or interest on Bitcoin and respective platforms

```
Logistic accuracy: 0.9458644645431905
```

Logistic Regression accuracy

expected to have better performance than NB because it covers the case of binary dependent variables (positive and negative)

```
MultinomialNB Accuracy: 0.8927775593131472
```

Multinomial Naive Bayes accuracy

performs with less accuracy but provides basic understanding of features selection in the early stage

Error Analysis

- First dataset was used for both models
- Second dataset was introduced later on to find most frequent tweeters
- Does combining both datasets create better accuracy because of bigger corpus?

Future Work

- Additional research can be performed in the error analysis area or finding more suitable algorithms instead of just sentiment analysis
  - K-means clustering to cluster into groups and select optimal bitcoin platform (bull market)
  - N-grams to see trends Ex: 'Ethereum rose in price'
  - Information extraction of tweet to see what is actually being mentioned

References

- https://machinelearningmastery.com/logistic-regression-for-machine-learning/
- https://www.twilio.com/blog/2017/12/sentiment-analysis-scikit-learn.html
- https://www.datacamp.com/community/tutorials/text-analytics-beginners-nltk
- https://nycdatascience.com/blog/student-works/cluster-analysis-twitter/
- https://towardsdatascience.com/sentiment-analysis-with-python-part-1-5ce197074184
- https://www.analyticsvidhya.com/blog/2018/07/hands-on-sentiment-analysis-dataset-python/
- https://cseweb.ucsd.edu/classes/wi17/cse258-a/reports/a080.pdf
- http://cs229.stanford.edu/proj2015/029_report.pdf
- https://www.researchgate.net/publication/314667612_Using_logistic_regression_method_to_classify_tweets_into_the_selected_topics
- And a lot of Stackoverflow and Google