Reddit News Analysis For Stock Market Prediction

Eric Stevens

“Applying ML techniques to dig into large amounts of data can help discover patterns that were not immediately apparent.”

[1] Aurélien Géron, Hands On Machine Learning

Goal

Can we use general news stories to predict the movement of the stock market the following day?

The Data - News

Top 25 news headlines from Reddit on a given date:

```
News.head()
```

	Date	News
0	2016-07-01	A 117-year-old woman in Mexico City finally re...
1	2016-07-01	IMF chief backs Athens as permanent Olympic host
2	2016-07-01	The president of France says if Brexit won, so...
3	2016-07-01	British Man Who Must Give Police 24 Hours' Not...
4	2016-07-01	100+ Nobel laureates urge Greenpeace to stop o...

The Data - Market

Daily Dow Jones Industrial Average Statistics

```
Stock.head()
```

	Date	Open	High	Low	Close	Volume	Adj Close
0	2016-07-01	17924.240234	18002.380859	17916.910156	17949.369141	82160000	17949.369141
1	2016-06-30	17712.759766	17930.609375	17711.800781	17929.990234	133030000	17929.990234
2	2016-06-29	17456.019531	17704.509766	17456.019531	17694.679688	106380000	17694.679688
3	2016-06-28	17190.509766	17409.720703	17190.509766	17409.720703	112190000	17409.720703
4	2016-06-27	17355.210938	17355.210938	17063.080078	17140.240234	138740000	17140.240234

Data - Combining the datasets

The problem is reduced to a binary classification where label is 1 if the close of the current day is less than the close of the following day.

```
data_combined.head()
```

	Date	Label	Headline
0	2008-08-11	0.0	b'Why wont America and Nato help us? If they w...
1	2008-08-12	1.0	b'Remember that adorable 9-year-old who sang a...
2	2008-08-13	0.0	b' U.S. refuses Israel weapons to attack Iran:...
3	2008-08-14	0.0	b'All the experts admit that we should legalis...
4	2008-08-15	1.0	b"Mom of missing gay man: Too bad he's not a 2...

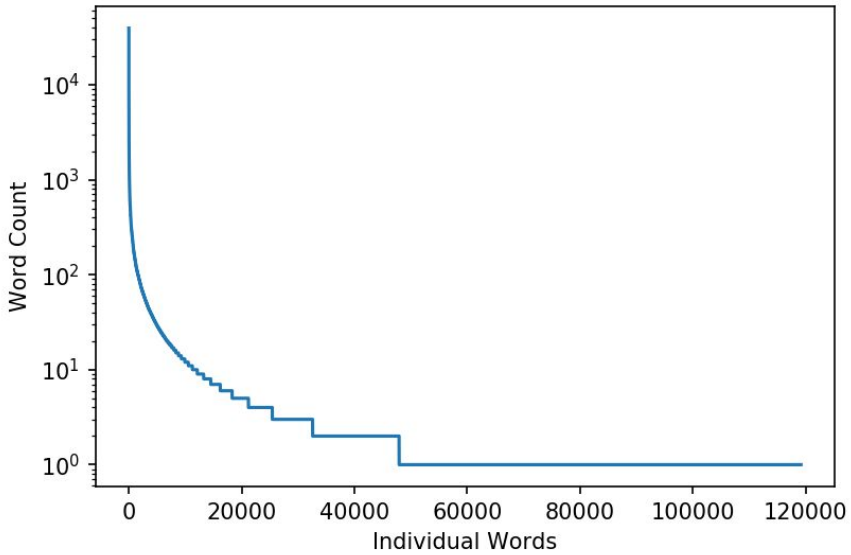
Data - The Label

An adjusted closing is a stock’s closing price on any given day of trading that has been amended to include any distributions and corporate actions that occurred at any time before the next days open.

The label of each observation is 1 if the adjusted closing price will be higher on the following data, 0 if it is lower.

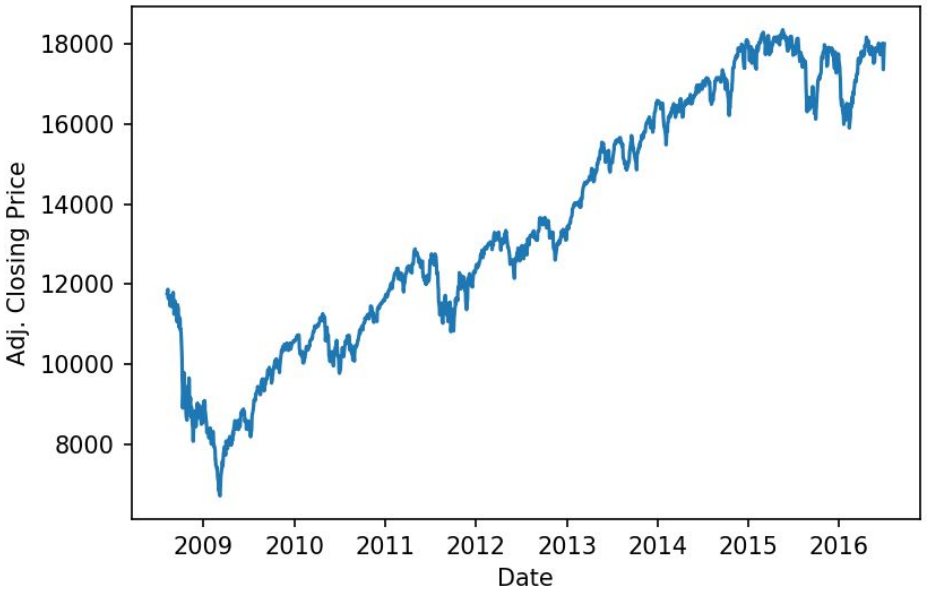
Data Analysis - word counts

- Roughly 120,000 unique words in all of news headlines.
- ~60% of them have only a single occurrence.
- Single occurrence words cannot be used for leaning.
- Preprocessing will have to reduce sparsity in the data.



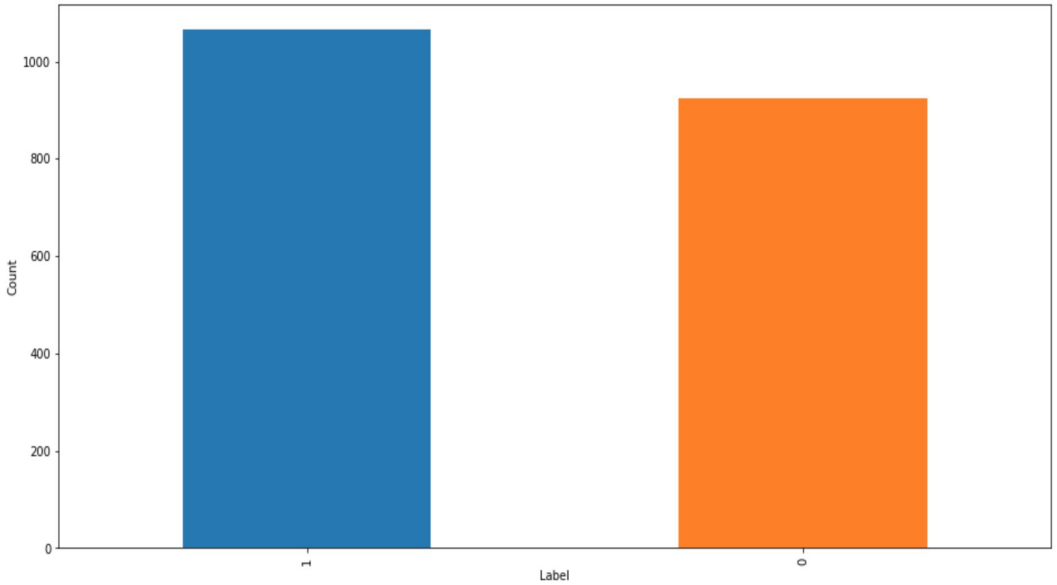
Data Analysis - stock movement

- Timespan of market data roughly covers Obama presidency, a period of steady growth.
- This could introduce a bias into our label data.
- Reducing the problem to a binary classification problem we may be able to reduce this effect.



Data Analysis - label counts

- Luckily there are almost an even number of labels.
- This difference will only slightly influence the resulting model.



Baseline - Totally Naive

- A completely naive model would just always guess ‘1’.
- If we randomly shuffle the data we get a range of accuracies ranging from about 50% to about 59%.
- The average accuracy ends up being around 54%.
- If there is information in the data we should be able to improve on this number.

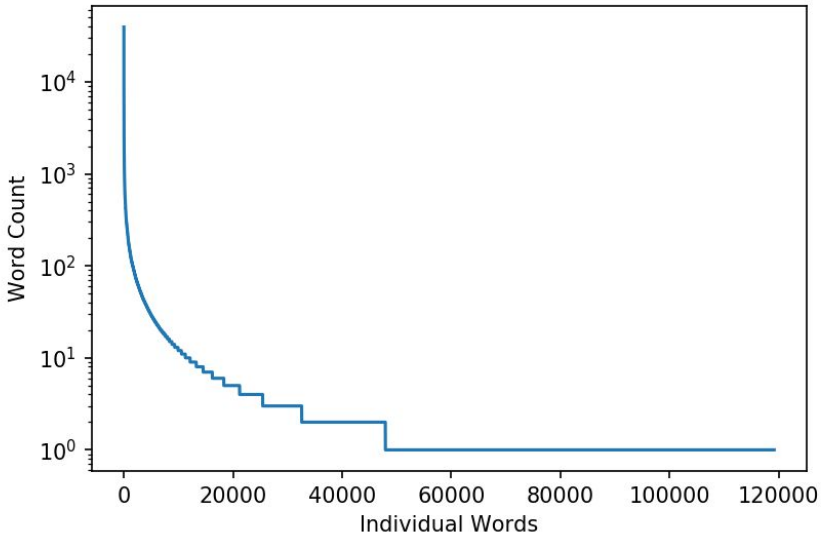
```
Run 0 accuracy: 0.5202020202020202
Run 1 accuracy: 0.494949494949495
Run 2 accuracy: 0.5606060606060606
Run 3 accuracy: 0.51010101010101
Run 4 accuracy: 0.5151515151515151
Run 5 accuracy: 0.5404040404040404
Run 6 accuracy: 0.5909090909090909
Run 7 accuracy: 0.5202020202020202
Run 8 accuracy: 0.5858585858585859
Run 9 accuracy: 0.5707070707070707

Average accuracy: 0.5409090909090908
```

Building Models - data preprocessing

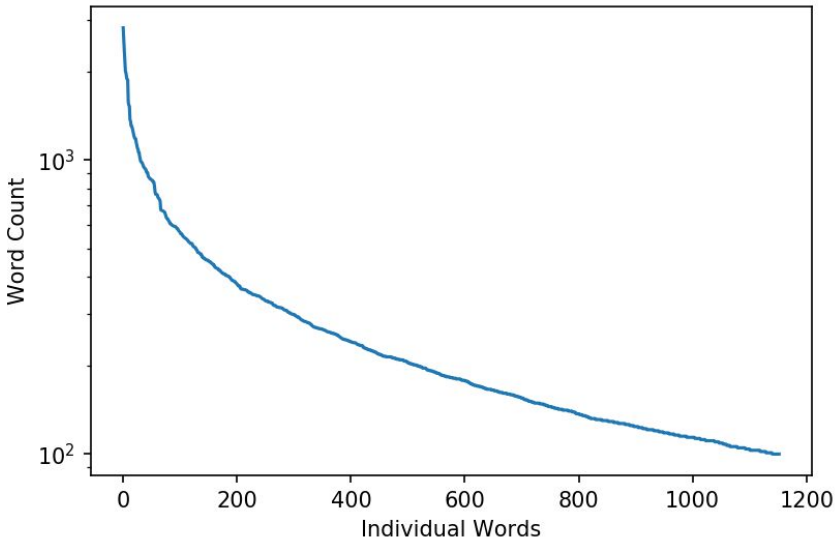
- Case independent
  - Make everything lower case.
- Tokenization
  - Using NLTK word_tokenize to split sentences up.
- Stop word removal
  - Remove all stop words using NLTK stopword list.
- Remove punctuation
  - Remove punctuation that has been tokenized.
  - Punctuation tokens will be removed but words containing punctuation such as ‘u.s.a.’ will not.
- Sparsity reduction
  - Remove words that appear less than 100 times in the dataset.

Data Preprocessing - counts

Before Preprocessing



After Preprocessing



Data Preprocessing - word profile

- Least common words do not appear to be too obscure.
- Most of the the most common words seem to contain some information about the world.
- Over seems that we will be able to extract some information about the state of the world with these words.

Least Common	MostCommon
('qatar', 100)	('country', 1207)
('countrys', 100)	('iran', 1229)
('push', 100)	('uk', 1257)
('meat', 100)	('first', 1285)
('drop', 100)	('killed', 1302)
('forest', 100)	('one', 1306)
('disease', 100)	('u.s.', 1352)
('early', 100)	('president', 1370)
('hands', 100)	('war', 1521)
('offers', 100)	('years', 1534)
('tens', 100)	('russia', 1572)
('daily', 100)	('people', 1871)
('towards', 100)	('israel', 1879)
('thailand', 101)	('police', 1905)
('hiv', 101)	('china', 1974)
('suggests', 101)	('government', 1999)
('conditions', 101)	('world', 2179)
('showing', 101)	('new', 2350)
('500', 101)	('says', 2561)
('reactor', 101)	('us', 2811)

Building Models - unigram probability model

- Mean of unigram probabilities in the headlines to determine label.
- Conditional probability read as percent of the appearances of word w_i associated with a label of 1.
- If P(S=1) >= .5, Label 1
- If P(S=1) < .5, Label 0

$$P(S=1) = \frac{\sum_i^N P(label{=}1|w_i)}{N}$$

$$P(label=1|w_i) = \frac{\#\ w_i\ appears\ in\ doc\ labeled\ 1}{\#\ w_i\ appears\ in\ doc}$$

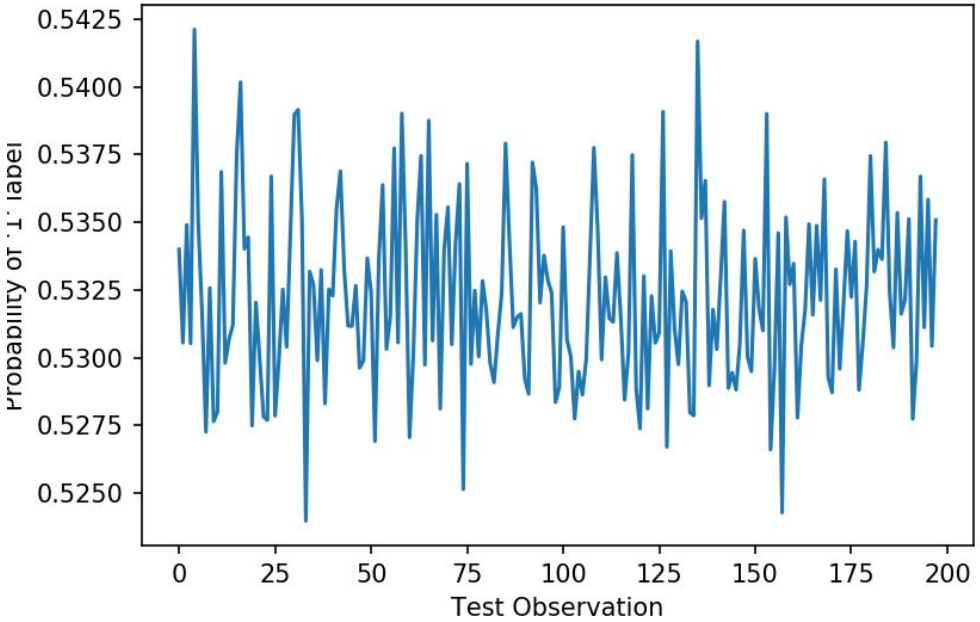
Unigram Model - performance

- Range of 50% to 60% performance.
- Average performance of 53%.
- Suspiciously close to totally naive solution.
- Even a little worse in this case.

```
Run 0 accuracy: 0.5151515151515151
Run 1 accuracy: 0.5454545454545454
Run 2 accuracy: 0.5757575757575758
Run 3 accuracy: 0.6060606060606061
Run 4 accuracy: 0.5303030303030303
Run 5 accuracy: 0.51010101010101
Run 6 accuracy: 0.5353535353535354
Run 7 accuracy: 0.494949494949495
Run 8 accuracy: 0.5353535353535354
Run 9 accuracy: 0.494949494949495

Average accuracy: 0.5343434343434343
```

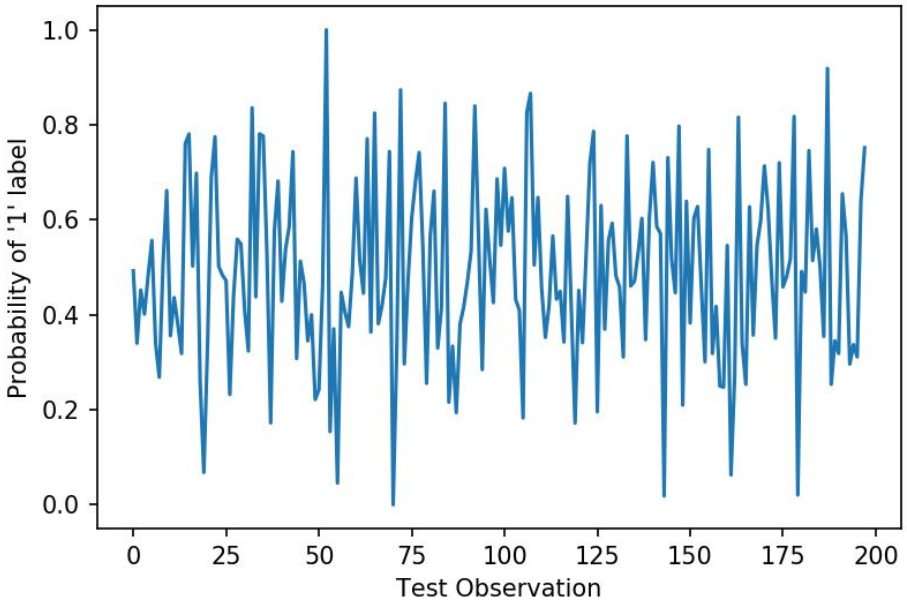
Unigram Model - the issue

- All probabilities for all collections of headlines are between the range of .525 and .5425.
- Highly regular but results in a guess of label ‘1’ every time.
- Equivalent to the totally naive method.



Unigram Model - normalize (last ditch effort)

```
Run 0 accuracy: 0.494949494949495
Run 1 accuracy: 0.4797979797979798
Run 2 accuracy: 0.5656565656565656
Run 3 accuracy: 0.4696969696969697
Run 4 accuracy: 0.5050505050505051
Run 5 accuracy: 0.46464646464646464
Run 6 accuracy: 0.5252525252525253
Run 7 accuracy: 0.4444444444444444
Run 8 accuracy: 0.4696969696969697
Run 9 accuracy: 0.48484848484848486

Average accuracy: 0.4904040404040405
```



Switch to data science packages

- Tf-idf vectorization
  - Similar to the count based method used above.
  - Will be used as the text representation for the package models.
  - Allows for n-gram vectorization
- Logistic regression
  - Discriminative model, more powerful than the unigram probability.
  - Great for classification problems.
- LSTM
  - Powerful model for time series data.
  - Better performance with order dependent relationships than n-grams.
  - Don’t have nearly enough data, but if all else fails.

Tf-Idf Vectorization

```
tfidf = TfidfVectorizer(sublinear_tf=True, min_df=100, norm='l2', encoding='latin-1',
                        ngram_range=(1, 4), stop_words='english')
```

- A minimum document frequency set to 100 is similar to the minimum occurrence value set to 100 in the unigram model.
- Stop words are still removed.
- N-grams for 1-4 are now all considered.

[2] Multi-Class Text Classification with Scikit-Learn, Susan Li

Logistic Regression

- Similar to original effort, relying on max likelihood type model.
- Because of the vectorization, will take into account n grams from 1 to 4.

Still nothing

```
Run 0 : 0.5105633802816901
Run 1 : 0.5080482897384306
Run 2 : 0.5187122736418511
Run 3 : 0.5099597585513078
Run 4 : 0.5142857142857142
Run 5 : 0.5172032193158954
Run 6 : 0.5127766599597585
Run 7 : 0.5259557344064386
Run 8 : 0.5214285714285715
Run 9 : 0.5045271629778671

Average Accuracy: 0.5143460764587525
```

LSTM (“hail-mary”)

```
Train on 1611 samples, validate on 378 samples
Epoch 1/10
1611/1611 [==============================] - 14s 9ms/step - loss: 0.6918 - acc: 0.5289 - val_loss: 0.6973 - val_acc: 0.5079
Epoch 2/10
1611/1611 [==============================] - 13s 8ms/step - loss: 0.6196 - acc: 0.6511 - val_loss: 0.7099 - val_acc: 0.5794
Epoch 3/10
1611/1611 [==============================] - 15s 9ms/step - loss: 0.2629 - acc: 0.8951 - val_loss: 0.9444 - val_acc: 0.5185
Epoch 4/10
1611/1611 [==============================] - 15s 9ms/step - loss: 0.0649 - acc: 0.9851 - val_loss: 1.3301 - val_acc: 0.5370
Epoch 5/10
1611/1611 [==============================] - 14s 9ms/step - loss: 0.0143 - acc: 0.9963 - val_loss: 1.4155 - val_acc: 0.5344
Epoch 6/10
1611/1611 [==============================] - 14s 9ms/step - loss: 0.0045 - acc: 1.0000 - val_loss: 1.6761 - val_acc: 0.5635
Epoch 7/10
1611/1611 [==============================] - 14s 9ms/step - loss: 0.0033 - acc: 0.9994 - val_loss: 1.8303 - val_acc: 0.5582
Epoch 8/10
1611/1611 [==============================] - 15s 9ms/step - loss: 0.0011 - acc: 1.0000 - val_loss: 1.8914 - val_acc: 0.5556
Epoch 9/10
1611/1611 [==============================] - 14s 9ms/step - loss: 0.0083 - acc: 0.9975 - val_loss: 1.5082 - val_acc: 0.5450
Epoch 10/10
1611/1611 [==============================] - 14s 9ms/step - loss: 0.0093 - acc: 0.9988 - val_loss: 1.9239 - val_acc: 0.5291
378/378 [==============================] - 1s 2ms/step
Test score: 1.9239365114736808
Test accuracy: 0.5291005287851606
Generating test predictions...
```

[3] Use News to Predict Stock Market, Kaggle:Iseiyjg

What Went Wrong?

- Bad Idea?
  - Reddit news headlines do not necessarily contain any actionable information about the stock market.
  - The stock market in notoriously unpredictable in general. Trying to derive a trading signal from random news data adds another layer of complexity.
- User error
  - I am not as skilled with these tools or this methodology as I would like to be.
- Bad dataset
  - The data was probably way too small to allow for the learning that would be needed to boost the signal strength.
  - Reddit news was a bad choice, stock market news wires would have drastically reduced the noise in the data.

Thank You

References:

1. Hands-On Machine Learning with Scikit-Learn & TensorFlow, Aurélien Géron, ISBN-10: 1491962291

2. Multi-Class Text Classification with Scikit-Learn, Susan Li, https://towardsdatascience.com/multi-class-text-classification-with-scikit-learn-12f1e60e0a9f

3. Use News to predict Stock Markets, Kaggle:Iseiyjg https://www.kaggle.com/Iseiyjg/use-news-to-predict-stock-markets