Determining the Sentiment of Financial News

The news database here will train the Naive Bayes, then RandomForrest For deploying I'd recommend using the NewsAPI code shared and tag the sentiment via the trained NB.

Constructing a Naive Bayes Classifier

- Load dataset
- Vectorize data
- Split data (80/20, train test, random_state=0 so as to allow reproducability)
- Initialize the NB classifer and fit
- Predict and measure accuracy

In [9]:

```
import numpy as np # linear algebra
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_c
from sklearn.feature_extraction.text import CountVectorizer

news_pd = pd.read_csv("./news_with_sentiment.csv")
news_pd = news_pd[:2000] # 28,000 rows will use more RAM than is av

cv = CountVectorizer() # Convert text data to a vector as that is r
X = cv.fit_transform(news_pd['text']).toarray()
y = news_pd['sentiment'] # y = the variable we are trying to predic
```

In [10]:

```
# Split train and test data (80/20)
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size

# Initialize the Gaussian Naive Bayes Classifier, then fit the data
from sklearn.naive_bayes import GaussianNB
classifier = GaussianNB()
classifier.fit(X_train, y_train)
```

Out[10]:

```
GaussianNB(priors=None)
```

In [11]:

```
# Predict sentiment of our test data
y_pred = classifier.predict(X_test)

from sklearn.metrics import accuracy_score
score = accuracy_score(y_test, y_pred)
```

And now we can view the accuracy:

In [12]:

```
print(score)
```

```
0.685
```

```
Roughly 68% accuracy. Not exactly stellar, if you reduce the
dataset further you end up with higher accuracy which is
interesting.
```

In [14]:

```
news_pd = pd.read_csv("./news_with_sentiment.csv")
news_pd = news_pd[:1000] # 28,000 rows will use more RAM than is ava

cv = CountVectorizer()
X = cv.fit_transform(news_pd['text']).toarray()
y = news_pd['sentiment']

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size

classifier = GaussianNB()
classifier.fit(X_train, y_train)

y_pred = classifier.predict(X_test)

score = accuracy_score(y_test, y_pred)

print(score)
```

```
0.775
```

77.5% accuracy on a 1000 row dataset with an 80/20 split.

After research, Naive Bayes appears to be better with smaller datasets but perhaps we can improve:

To improve on our Naive Bayes we can now try a Random Forest:

- Load dataset
- Remove stopwords, min_df=7 means the data is irrelevant if used in more than 7 documents, max_df of 0.8 means it also is irrelevant if used in more than 80% of documents
- Vectorize data (max_features is the max number of WORDS in Vector form that will influence the sentiment)
- Split data (80/20, train test, random_state=0 so as to allow reproducability)
- Initialize the Random Forest classifer and fit
- Predict and measure accuracy

In [15]:

```
# Read in 20,000 headlines
news_pd = pd.read_csv("./news_with_sentiment.csv")
news_pd = news_pd[:20000] # 28,000 rows will use more RAM than is a
y = news_pd['sentiment']
```

In [16]:

```
from nltk.corpus import stopwords
from sklearn.feature_extraction.text import TfidfVectorizer

# Remove stopwords and vectorize the dataset
#TfidVectorizer converts a collection of raw documents to a matrix 
vectorizer = TfidfVectorizer(max_features=2500, min_df=7, max_df=0.
processed_features = vectorizer.fit_transform(news_pd['text']).toar
```

In [17]:

```
# 80/20 data split
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(processed_featu

# Fit our model with split data, starting with 450 estimators (450
from sklearn.ensemble import RandomForestClassifier

text_classifier = RandomForestClassifier(n_estimators=450, random_s
text_classifier.fit(X_train, y_train)
```

```
/home/nbuser/anaconda3_420/lib/python3.5/site-packages/sklearn/ensembl
e/weight_boosting.py:29: DeprecationWarning: numpy.core.umath_tests is
an internal NumPy module and should not be imported. It will be remove
d in a future NumPy release.
  from numpy.core.umath_tests import inner1d
```

Out[17]:

```
RandomForestClassifier(bootstrap=True, class_weight=None, criterion='g
ini',
            max_depth=None, max_features='auto', max_leaf_nodes=None,
            min_impurity_decrease=0.0, min_impurity_split=None,
            min_samples_leaf=1, min_samples_split=2,
            min_weight_fraction_leaf=0.0, n_estimators=450, n_jobs=1,
            oob_score=False, random_state=0, verbose=0, warm_start=Fal
se)
```

In [18]:

```
# Predicting the sentiment of our test data
predictions = text_classifier.predict(X_test)


# Checking our accuracy
from sklearn.metrics import accuracy_score
print(accuracy_score(y_test, predictions))
```

```
0.93575
```

93.57% accuracy

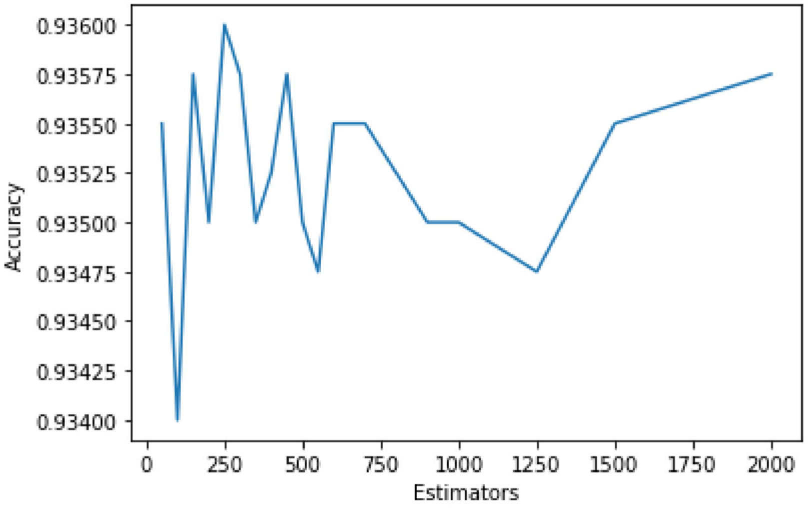
Hyperparameter Tuning:

- Choose a set of trees we want to test
- Train the model with n trees. store accuracy

- Loop above until complete
- Plot the resulting trees v accuracy

In [ ]:

```
from sklearn.ensemble import RandomForestRegressor

estimators = [50, 100, 150, 200, 250, 300, 350, 400, 450, 500, 550,
accuracy = []

for estimator_num in estimators:
    # Fit and predict
    text_classifier = RandomForestClassifier(n_estimators=estimator
    text_classifier.fit(X_train, y_train)
    predictions = text_classifier.predict(X_test)

    # Store accuracy
    from sklearn.metrics import accuracy_score
    accuracy.append(accuracy_score(y_test, predictions))


# Graph reported accuracy of various sets of estimators
import matplotlib.pyplot as plt

plt.plot(estimators, accuracy)
plt.ylabel('Accuracy')
plt.xlabel('Estimators')
plt.show()

print(estimators)
print(accuracy)
```



A strange curve?

As per: https://en.wikipedia.org/wiki/Talk%3ARandom_forest (https://en.wikipedia.org/wiki/Talk%3ARandom_forest)

"Random Forests does not overfit. The testing performance of Random Forests does not decrease (due to overfitting) as the number of trees increases. Hence after certain number of trees the performance tend to stay in a certain value."

However, we can also see that ~250 estimators/trees is the ideal parameter.

Naive Bayes v Random Forest v SVM:
https://www.researchgate.net/publication/336225950_Comparison_of_Naive_Bayes_Sup
(https://www.researchgate.net/publication/336225950_Comparison_of_Naive_Bayes_Sup

Pull Fresh News:

In [19]:

```
import requests
import time
import datetime


articleCount = 0

headers = {
    'User-Agent': 'Mozilla/5.0 (Windows NT 6.1) AppleWebKit/537.36
}

stocks = ['TSLA', 'AMZN', 'MMM', 'INTC', 'GOOGL', 'FB', 'MSFT', 'AA
list_of_headlines = []
for line in stocks:
    ticker = line

    try:

        #Query for the stock name, for refined news queries.
        resp = requests.get(
            url="https://www.alphavantage.co/query?function=SYMBOL_
                ticker), headers=headers)
        data = resp.json()
        companyName = data['bestMatches'][0]['2. name']
        print("Company Name: " + companyName)

        #Query for news
        resp = requests.get(
            url='https://newsapi.org/v2/everything?'
'q={}&'
'from=2020-01-05' # This is the OLDEST date an article can be from,
'sortBy=popularity&' #Filter by popularity (read the newsapi docs)
'apiKey=fe00115ceffe418988616191b03e1c74'.format(
                ticker + " " + companyName), headers=headers) #Add
        data = resp.json()

        for article in data['articles']:
            articleCount = articleCount + 1
            newsTitle = article['title']
            print(newsTitle)
            list_of_headlines.append(newsTitle)

        time.sleep(1)

    except Exception as e:
        print("Error: " + str(e))
        time.sleep(10)

# Create the pandas DataFrame and save to csv
df = pd.DataFrame({'headlines':list_of_headlines})
df.to_csv('fresh_news_month_tsla.csv', encoding='utf-8', mode='w',
```

```
Cramer Weighs In On Cracker Barrel, UPS And More
AWS Announces General Availability of Amazon Keyspaces (for Apache C
assandra)
Company Name: 3M Company
Dow Jones 378-Point Intraday Gain Fades, But 3M A Bright Spot; Netfl
ix, Tesla Weigh On Nasdaq - Investor's Business Daily
```

```
ix, Tesla Weigh On Nasdaq - Investor's Business Daily
Dow Jones, US Stocks Rise As Countries Begin To Reopen Economies - I
nvestor's Business Daily
3M’s stock surges on earnings beat, that was nearly 20 years in the
making
3M Co (MMM) Q1 2020 Earnings Call Transcript
Is 3M Oversold At $155?
3M Company (MMM) CEO Mike Roman on Q1 2020 Results - Earnings Call T
ranscript
Were Hedge Funds Right About 3M Company (MMM)?
Did You Acquire 3M (MMM) Before February 9, 2017? Johnson Fistel Con
tinues its Investigation of 3M; Should Management be Held Accountabl
e for Investors Losses?
3M (MMM) Gains But Lags Market: What You Should Know
3M Holds Good On Its Promise To Prioritize Dividend
```

In [20]:

```python
from nltk.corpus import stopwords
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.feature_extraction.text import CountVectorizer

# Read in fresh news
fresh_news = pd.read_csv('./fresh_news_month_tsla.csv')
fresh_news['headlines'].head(5)

# Vectorize new text data with a max of 40 words being predictors
vectorizer_new_data = CountVectorizer(max_features=40, min_df=9)
processed_features_new_data = vectorizer_new_data.fit_transform(fre

# Vectorize training text data with a max of 40 words being predict
vectorizer = CountVectorizer(max_features=40, min_df=9)
processed_features = vectorizer.fit_transform(news_pd['text']).toar

X_train, X_test, y_train, y_test = train_test_split(processed_featu

# Predict on new/fresh news after fitting on training data
text_classifier = RandomForestClassifier(n_estimators=650, random_s
text_classifier.fit(X_train, y_train)

predictions = text_classifier.predict(processed_features_new_data)

# Output our predictions
print(predictions)

for i in range(len(predictions)):
    if predictions[i] == 1:
        print("Positive: " + fresh_news['headlines'][i])
    if predictions[i] == -1:
        print("Negative: " + fresh_news['headlines'][i])

```

```
[ 0  0  0  0  1  1  0  0 -1  0  0  0  0  0  0  0  1  0  1  1  1 -1
0  0
  0  1  0 -1  0  0  0  0  0  0  0  0  0  0  0  0  0  1  0  0  0  0
0  1
  1  1  1  0  1  0  0  0 -1  0  0 -1  0  0  0  0  0 -1  0  0  0  0
0  0
  0  0  0 -1  0  0  0  0  1  0 -1  0  0  0  0 -1  1  0  1  0  0  0
0  1
  0  0  0  0  1  0  0  0  0  0  0  1  0  0  0  0  0  0  0  1 -1  0
0  0
  0  1  0  0 -1  0  0  0  0  0  0  0 -1  1  1  0  0  0  1  0  0  1
0  0
  0  0  1  0  0  0  0  0  1  0  0  0  0  0  1  0]
Positive: Tesla wants to reopen California factory, but local author
ities say not yet
Positive: Ford is first auto maker to warn of lower sales, but unlik
ely to be last
Negative: Market Extra: The S&P 500 just posted the most daily swing
s of 3% or greater in more than a decade—even as the stock market hi
ts a 5-week high
Positive: The force that’s propelled the stock market rally will exh
aust itself this week
Positive: Tesla Confirms Shanghai Gigafactory Shutdown, But Says I
```

```
t's All 'According To Plan'
Positive: Steve Grasso Says Tesla Has 'Defied All Laws Of Probabilit
y'
Positive: Amazon stock hits record high on hopes for a coronavirus-r
elated boom
Negative: AMC’s stock soars after report Amazon held merger talks
Positive: Netflix stock surges to record high as investors bet on st
reaming during coronavirus
Negative: Market Extra: The S&P 500 just posted the most daily swing
s of 3% or greater in more than a decade—even as the stock market hi
ts a 5-week high
Positive: Dow Jones, US Stocks Rise As Countries Begin To Reopen Eco
nomies - Investor's Business Daily
Positive: Did You Acquire 3M (MMM) Before February 9, 2017? Johnson
Fistel Continues its Investigation of 3M; Should Management be Held
Accountable for Investors Losses?
Positive: 3M (MMM) Gains But Lags Market: What You Should Know
Positive: 3M Holds Good On Its Promise To Prioritize Dividend
Positive: Stocks fights for gains as earnings season revs up - Fox B
usiness
Positive: The Importance Of Reading Footnotes – Uncovering Material
Items In Filings
Negative: Dow Blue-Chip 3M Co Surges on Coronavirus Demand, but Bond
King Warns of Danger
Negative: Why I Think You Should Buy This Defence Stock Before May
Negative: Call Traders Blast These 2 Chip Stocks
Negative: 5 Tech Stocks Poised to Beat Estimates This Earnings Seaso
n - Yahoo Finance
Positive: Making Most Of Lockdowns, Facebook Gaming Launches Earlier
Than Planned - Benzinga
Negative: Alphabet Announces First Quarter 2020 Results (Alphabet)
Negative: Market Extra: The S&P 500 just posted the most daily swing
s of 3% or greater in more than a decade—even as the stock market hi
ts a 5-week high
Positive: Big Data tech CEO on the federal government’s response to
coronavirus: ‘A total failure of leadership’
Positive: Dropbox’s first quarterly profit is a sign of the ever-cha
nging economy
Positive: Investors have $5.1 trillion hiding out in the shares of f
ive companies, which will be tested this week
Positive: Making Most Of Lockdowns, Facebook Gaming Launches Earlier
Than Planned - Benzinga
Positive: Dropbox’s first quarterly profit is a sign of the ever-cha
nging economy
Positive: Investors have $5.1 trillion hiding out in the shares of f
ive companies, which will be tested this week
Negative: FB to allow employees to work remotely until year end
Positive: How Large Option Traders Are Playing Microsoft As Cloud Bu
siness Booms - Yahoo Finance
Negative: Tech Stocks' Apr 29 Earnings Lineup: NOW, FB, MSFT, FICO,
GIB - Yahoo Finance
Negative: Hedge Funds’ #3 Stock Pick Debunked Naysayers
Positive: Optimism May Be Over Done In The Equity Markets
Positive: Jonathan Angrist’s Cognios Can’t Deliver Despite Apple, Am
azon, Microsoft Bets
Positive: Microsoft announces registered exchange offers
Positive: The Ratings Game: Why Apple investors should be worried by
```

```
AT&T’s earnings
Positive: Big Data tech CEO on the federal government’s response to
coronavirus: ‘A total failure of leadership’
Positive: Investors have $5.1 trillion hiding out in the shares of f
ive companies, which will be tested this week
Positive: Gene Munster Dismisses Goldman's Apple Downgrade, Says Cup
ertino Has Long-Term Earnings Power
```

Save our model to disk for production deployment to Sparkbot

In [ ]:

```python
from joblib import dump
dump(text_classifier,'sentimentclassified.joblib')
```

In [ ]:

```
 
```