**import** pandas **as** pd

**import** numpy **as** np

**import** matplotlib.pyplot **as** plt

**import** os

**import** datetime

**import** yfinance **as** yf

In [2]:

*# Get all of the stock files to process*

files **=** os**.**listdir('data/')

*# Initialize new dataframe to hold stock data*

stocks **=** pd**.**DataFrame()

*# For each stock to process*

**for** x **in** range(len(files)):

*# Some tickers may be delisted, use try/except to avoid errors if ticker does not exist in yFinance*

**try**:

*# Assign current file name*

filename **=** files[x]

*# Read csv file for current ticker*

data **=** pd**.**read\_csv('data/'**+**filename**+**'/stock\_data\_sentiment.csv')

*# Convert string time to datetime*

data['Datetime'] **=** pd**.**to\_datetime(data['Datetime'])

*# Convert 0 sentiment to -1*

data**.**loc[data['Sentiment']**==**0, 'Sentiment'] **=** **-**1

*# Initialize Weights and Tweet numbers*

data['Tweets'] **=** 1

data['Weight'] **=** 1

*# Determine the mean and standard deviation of the number of followers a given user has for a given tweet*

*# This ensures tweets with less tweets per day are treated equally*

data['Followers\_Mean'] **=** data['Followers']**.**rolling(10000, min\_periods**=**1)**.**mean()

data['Followers\_Std'] **=** data['Followers']**.**rolling(10000, min\_periods**=**1)**.**std()

data['Followers\_Std'] **=** data['Followers\_Std']**.**fillna(data['Followers\_Std']**.**values[1])

*# weight tweets from users with higher follower counts more heavily. Calculate based on mean and standard deviation gates*

data**.**loc[ (data['Followers']**>=**data['Followers\_Mean']) **&** (data['Followers'] **<** (data['Followers\_Mean']**+**data['Followers\_Std'])), 'Weight'] **+=** 1

data**.**loc[ (data['Followers']**>=**(data['Followers\_Mean']**+**data['Followers\_Std'])) **&** (data['Followers'] **<** (data['Followers\_Mean']**+**data['Followers\_Std']**\***2)), 'Weight'] **+=** 2

data**.**loc[data['Followers']**>=**(data['Followers\_Mean']**+**data['Followers\_Std']**\***2), 'Weight'] **+=** 3

*# Determine the mean and standard deviation of the number of re-tweets a given user has for a given tweet*

*# This ensures tweets with less tweets per day are treated equally*

data['RTs\_Mean'] **=** data['RTs']**.**rolling(10000, min\_periods**=**1)**.**mean()

data['RTs\_Std'] **=** data['RTs']**.**rolling(10000, min\_periods**=**1)**.**std()

data['RTs\_Std'] **=** data['RTs\_Std']**.**fillna(data['RTs\_Std']**.**values[1])

*# Weight tweets with more re-tweests more heavily. Calculate based on mean and standard deviation gates*

data**.**loc[ (data['RTs']**>=**data['RTs\_Mean']) **&** (data['RTs'] **<** (data['RTs\_Mean']**+**data['RTs\_Std'])), 'Weight'] **+=** 1

data**.**loc[ (data['RTs']**>=**(data['RTs\_Mean']**+**data['RTs\_Std'])) **&** (data['RTs'] **<** (data['RTs\_Mean']**+**data['RTs\_Std']**\***2)), 'Weight'] **+=** 2

data**.**loc[data['RTs']**>=**(data['RTs\_Mean']**+**data['RTs\_Std']**\***2), 'Weight'] **+=** 3

*# multiply the sentiment score by the individual weigth*

data['Sentiment\_Weighted'] **=** data['Sentiment']**\***data['Weight']

*# Group the stock data by months and days*

data **=** data**.**groupby([data**.**Datetime**.**dt**.**month, data**.**Datetime**.**dt**.**day])**.**sum()

*# Reassign the ticker name that was lost after grouping*

data['Ticker'] **=** filename**.**split('\_')[0]

*# Reassign date based on index values of month and day*

data['Date'] **=** pd**.**to\_datetime([ str(x)**+**'/'**+**str(y)**+**'/2016' **for** (x,y) **in** data**.**index**.**values ])**+**datetime**.**timedelta(days**=**1)

*# Drop the index*

data **=** data**.**reset\_index(drop**=True**)

*# Divide the sentiment by the total number of tweets*

data['Sentiment\_Weighted'] **/=** data['Tweets']

*# Get the rolling average of the sentiment and tweet volume*

data['Sentiment\_MA'] **=** data['Sentiment\_Weighted']**.**rolling(3, min\_periods**=**1)**.**mean()

data['Tweets\_MA'] **=** data['Tweets']**.**rolling(3, min\_periods**=**1)**.**mean()

*# Get the starting date and ending date to extract stock prices*

start\_date **=** data['Date']**.**min()

end\_date **=** data['Date']**.**max()**+**datetime**.**timedelta(days**=**2)

*# Download stock price data for given range of tweets*

prices **=** yf**.**download(tickers**=**filename**.**split('\_')[0], start**=**start\_date, end**=**end\_date)**.**reset\_index()

*# Calculate percent change based on stock price changes*

prices['Percent\_Change'] **=** (prices['Adj Close']**.**pct\_change()**\***100)**.**shift(**-**1)

*# Bin percent changes by amount lost/gained by stock*

prices['Percent\_Change\_Bin'] **=** pd**.**cut(prices['Percent\_Change'], [**-**100, 0, 2, 100], labels**=**[0, 1, 2])

*# Combine the stock sentiment data and the pricing data*

data **=** data**.**merge(prices, on**=**'Date', how**=**'left')

*# Remove excess columns*

data **=** data[['Ticker', 'Date', 'Sentiment\_Weighted', 'Sentiment\_MA', 'Tweets', 'Tweets\_MA', 'Adj Close', 'Percent\_Change', 'Percent\_Change\_Bin']]

*# Drop missing values for days without pricing information*

data **=** data**.**dropna()**.**reset\_index(drop**=True**)

*# Save stock sentiment and pricing data to csv*

data**.**to\_csv('data/' **+** filename **+** '/stock\_data\_inputs.csv', index**=False**)

*# Add ticker data to the main data set*

stocks **=** pd**.**concat([stocks, data])

*# Print status of ticker*

print(filename**.**split('\_')[0], '- Completed')

*# If the ticker is not in yFinance*

**except** Exception **as** e:

*# Print error message*

print(filename**.**split('\_')[0], '-', e)

*# Save data for all stocks to csv*

stocks**.**to\_csv('combined\_stock\_inputs.csv', index**=False**)

[\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*100%\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*] 1 of 1 completed

aal - Completed

[\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*100%\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*] 1 of 1 completed

aapl - Completed

[\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*100%\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*] 1 of 1 completed

adbe - Completed

[\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*100%\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*] 1 of 1 completed

adp - Completed

[\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*100%\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*] 1 of 1 completed

adsk - Completed

[\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*100%\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*] 1 of 1 completed

akam - Completed

[\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*100%\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*] 1 of 1 completed

1 Failed download:

- ALXN: No data found, symbol may be delisted

alxn - Completed

[\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*100%\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*] 1 of 1 completed

amat - Completed

[\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*100%\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*] 1 of 1 completed

amgn - Completed

[\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*100%\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*] 1 of 1 completed

amzn - Completed

[\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*100%\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*] 1 of 1 completed

atvi - Completed

[\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*100%\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*] 1 of 1 completed

avgo - Completed

[\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*100%\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*] 1 of 1 completed

bbby - Completed

[\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*100%\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*] 1 of 1 completed

bidu - Completed

[\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*100%\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*] 1 of 1 completed

bmrn - Completed

[\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*100%\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*] 1 of 1 completed

ca - Completed

[\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*100%\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*] 1 of 1 completed

1 Failed download:

- CELG: No data found, symbol may be delisted

celg - Completed

[\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*100%\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*] 1 of 1 completed

cern - Completed

[\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*100%\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*] 1 of 1 completed

chkp - Completed

[\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*100%\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*] 1 of 1 completed

chtr - Completed

[\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*100%\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*] 1 of 1 completed

cmcsa - Completed

[\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*100%\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*] 1 of 1 completed

cost - Completed

[\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*100%\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*] 1 of 1 completed

csco - Completed

[\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*100%\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*] 1 of 1 completed

csx - Completed

[\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*100%\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*] 1 of 1 completed

1 Failed download:

- CTRP: No data found, symbol may be delisted

ctrp - Completed

[\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*100%\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*] 1 of 1 completed

ctsh - Completed

[\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*100%\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*] 1 of 1 completed

1 Failed download:

- DISCA: No data found, symbol may be delisted

disca - Completed

[\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*100%\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*] 1 of 1 completed

1 Failed download:

- DISCK: No data found, symbol may be delisted

disck - Completed

[\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*100%\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*] 1 of 1 completed

dish - Completed

[\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*100%\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*] 1 of 1 completed

dltr - Completed

[\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*100%\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*] 1 of 1 completed

ea - Completed

[\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*100%\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*] 1 of 1 completed

ebay - Completed

[\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*100%\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*] 1 of 1 completed

endp - Completed

[\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*100%\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*] 1 of 1 completed

esrx - Completed

[\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*100%\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*] 1 of 1 completed

expe - Completed

[\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*100%\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*] 1 of 1 completed

fast - Completed

[\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*100%\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*] 1 of 1 completed

fb - Completed

[\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*100%\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*] 1 of 1 completed

fisv - Completed

[\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*100%\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*] 1 of 1 completed

1 Failed download:

- FOXA: Data doesn't exist for startDate = 1459227600, endDate = 1466226000

foxa - Completed

[\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*100%\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*] 1 of 1 completed

1 Failed download:

- FOX: Data doesn't exist for startDate = 1459141200, endDate = 1466139600

fox - Completed

[\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*100%\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*] 1 of 1 completed

gild - Completed

[\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*100%\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*] 1 of 1 completed

googl - Completed

[\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*100%\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*] 1 of 1 completed

goog - Completed

[\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*100%\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*] 1 of 1 completed

hsic - Completed

[\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*100%\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*] 1 of 1 completed

ilmn - Completed

[\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*100%\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*] 1 of 1 completed

inct - Completed

[\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*100%\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*] 1 of 1 completed

incy - Completed

[\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*100%\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*] 1 of 1 completed

intu - Completed

[\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*100%\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*] 1 of 1 completed

isrg - Completed

[\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*100%\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*] 1 of 1 completed

jd - Completed

[\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*100%\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*] 1 of 1 completed

khc - Completed

[\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*100%\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*] 1 of 1 completed

lbtya - Completed

[\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*100%\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*] 1 of 1 completed

lbtyk - Completed

[\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*100%\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*] 1 of 1 completed

1 Failed download:

- LLTC: No data found for this date range, symbol may be delisted

lltc - Completed

[\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*100%\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*] 1 of 1 completed

1 Failed download:

- LMCA: No data found for this date range, symbol may be delisted

lmca - Completed

[\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*100%\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*] 1 of 1 completed

1 Failed download:

- LMCK: No data found, symbol may be delisted

lmck - Completed

[\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*100%\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*] 1 of 1 completed

lrcx - Completed

[\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*100%\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*] 1 of 1 completed

lrcx - Completed

[\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*100%\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*] 1 of 1 completed

1 Failed download:

- LVNTA: No data found for this date range, symbol may be delisted

lvnta - Completed

[\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*100%\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*] 1 of 1 completed

mar - Completed

[\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*100%\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*] 1 of 1 completed

mat - Completed

[\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*100%\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*] 1 of 1 completed

mdlz - Completed

[\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*100%\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*] 1 of 1 completed

mnst - Completed

[\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*100%\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*] 1 of 1 completed

msft - Completed

[\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*100%\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*] 1 of 1 completed

mu - Completed

[\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*100%\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*] 1 of 1 completed

1 Failed download:

- MXIM: No data found, symbol may be delisted

mxim - Completed

[\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*100%\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*] 1 of 1 completed

1 Failed download:

- MYL: No data found, symbol may be delisted

myl - Completed

[\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*100%\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*] 1 of 1 completed

nclh - Completed

[\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*100%\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*] 1 of 1 completed

nflx - Completed

[\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*100%\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*] 1 of 1 completed

ntap - Completed

[\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*100%\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*] 1 of 1 completed

ntes - Completed

[\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*100%\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*] 1 of 1 completed

nvda - Completed

[\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*100%\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*] 1 of 1 completed

nxpi - Completed

[\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*100%\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*] 1 of 1 completed

orly - Completed

[\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*100%\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*] 1 of 1 completed

payx - Completed

[\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*100%\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*] 1 of 1 completed

pcar - Completed

[\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*100%\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*] 1 of 1 completed

1 Failed download:

- PCLN: No data found for this date range, symbol may be delisted

pcln - Completed

[\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*100%\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*] 1 of 1 completed

pypl - Completed

[\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*100%\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*] 1 of 1 completed

qcom - Completed

[\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*100%\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*] 1 of 1 completed

1 Failed download:

- QVCA: No data found for this date range, symbol may be delisted

qvca - Completed

[\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*100%\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*] 1 of 1 completed

regn - Completed

[\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*100%\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*] 1 of 1 completed

rost - Completed

[\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*100%\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*] 1 of 1 completed

sbac - Completed

[\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*100%\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*] 1 of 1 completed

sbux - Completed

[\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*100%\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*] 1 of 1 completed

1 Failed download:

- SNDK: No data found for this date range, symbol may be delisted

sndk - Completed

[\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*100%\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*] 1 of 1 completed

srcl - Completed

[\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*100%\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*] 1 of 1 completed

stx - Completed

[\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*100%\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*] 1 of 1 completed

swks - Completed

[\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*100%\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*] 1 of 1 completed

1 Failed download:

- SYMC: No data found, symbol may be delisted

symc - Completed

[\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*100%\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*] 1 of 1 completed

tmus - Completed

[\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*100%\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*] 1 of 1 completed

trip - Completed

[\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*100%\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*] 1 of 1 completed

tsco - Completed

[\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*100%\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*] 1 of 1 completed

tsla - Completed

[\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*100%\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*] 1 of 1 completed

txn - Completed

[\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*100%\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*] 1 of 1 completed

ulta - Completed

[\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*100%\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*] 1 of 1 completed

1 Failed download:

- VIAB: No data found, symbol may be delisted

viab - Completed

[\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*100%\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*] 1 of 1 completed

vod - Completed

[\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*100%\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*] 1 of 1 completed

vrsk - Completed

[\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*100%\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*] 1 of 1 completed

vrtx - Completed

[\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*100%\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*] 1 of 1 completed

wba - Completed

[\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*100%\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*] 1 of 1 completed

wdc - Completed

[\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*100%\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*] 1 of 1 completed

1 Failed download:

- WFM: No data found for this date range, symbol may be delisted

wfm - Completed

[\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*100%\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*] 1 of 1 completed

1 Failed download:

- XLNX: No data found, symbol may be delisted

xlnx - Completed

[\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*100%\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*] 1 of 1 completed

1 Failed download:

- YHOO: No data found for this date range, symbol may be delisted

yhoo - Completed

In [5]:

**from** sklearn.ensemble **import** RandomForestClassifier

*# Sort values in stock sentiment/pricing data by date and ticker name*

stocks **=** stocks**.**sort\_values(['Date', 'Ticker'])

*# Use first 2 months of data as test data*

d\_train **=** stocks**.**loc[stocks['Date'] **<** datetime**.**datetime(year**=**2016, month**=**4, day**=**15)]**.**reset\_index(drop**=True**)

*# Use last 2-3 months as test data*

d\_test **=** stocks**.**loc[stocks['Date'] **>=** datetime**.**datetime(year**=**2016, month**=**4, day**=**15)]**.**reset\_index(drop**=True**)

*# Remove extra columns from train data*

X\_train **=** d\_train[['Sentiment\_Weighted', 'Sentiment\_MA', 'Tweets', 'Tweets\_MA']]**.**values

y\_train **=** d\_train['Percent\_Change\_Bin']**.**values

*# Remove extra solumns from test data*

X\_test **=** d\_test[['Sentiment\_Weighted', 'Sentiment\_MA', 'Tweets', 'Tweets\_MA']]**.**values

y\_test **=** d\_test['Percent\_Change\_Bin']**.**values

print(X\_train**.**shape, y\_train**.**shape)

print(X\_test**.**shape, y\_test**.**shape)

print('\n')

*# Define random forest classifier*

model **=** RandomForestClassifier(random\_state**=**1)

*# Train model with training data*

model**.**fit(X\_train, y\_train)

*# Predict the test data*

preds **=** model**.**predict(X\_test)

*# Print the percent of predictions that resulted in investing in a losing stock*

print(len(np**.**where( (preds**>**0) **&** (y\_test**==**0) )[0])**/**len(preds))

print('\n')

*# Add predictions to test dataset*

d\_test['Prediction'] **=** preds

*# Initialize starting capital to test model effectiveness*

bot\_capital **=** 10000

long\_capital **=** 10000

*# Add capital as first data points*

bots **=** [bot\_capital]

longs **=** [long\_capital]

*# Get unique dates*

dates **=** d\_test['Date']**.**sort\_values()**.**unique()

*# For each date*

**for** date **in** dates:

*# filter dataframe to date*

temp **=** d\_test[d\_test['Date']**==**date]

*# Calculate profit from investing equal parts to all tickers in timeframe*

long\_capital **=** long\_capital **+** ((long\_capital**/**len(temp)) **\*** (temp['Percent\_Change']**/**100))**.**sum()

*# Calculate profit from using the Twitter bot to determine what stocks to invest in*

temp **=** temp[temp['Prediction']**>**0]

bot\_capital **=** bot\_capital **+** ((bot\_capital**/**len(temp)) **\*** (temp['Percent\_Change']**/**100))**.**sum()

*# Keep track of the account totals ove time*

longs**.**append(long\_capital)

bots**.**append(bot\_capital)

*# Plot the accounts balances over time*

plt**.**figure(figsize**=**(15,7))

plt**.**plot(dates, longs[1:], c**=**'b', label**=**'Long Invest')

plt**.**plot(dates, bots[1:], c**=**'r', label**=**'Bot Invest')

plt**.**legend()

plt**.**show()

*# Show the return from each account over time*

print('Long-Term Investment:', round(longs[**-**1],2), '(', round((longs[**-**1]**-**longs[0])**/**longs[0]**\***100,2), '% )')

print('Bot Trading:', round(bots[**-**1],2), '(', round((bots[**-**1]**-**bots[0])**/**bots[0]**\***100,2), '% )')

(1121, 4) (1121,)

(3582, 4) (3582,)

0.20100502512562815

Long-Term Investment: 9795.35 ( -2.05 % )

Bot Trading: 10060.29 ( 0.6 % )