Max Sterman

How to run Can News Headlines Predict the VIX?

**Programs:**

**Python Packages needed: Pandas, Numpy, News API, Alpha Vantage (keys are given in the code for the previous two packages), TextBlob, and VaderSentiment, matplotlib, operator, statsmodels, pickle, ntlk, datetime, math, time, copy.**

**R Packages need: Tsereies, forecast, ggplot2, lmtest, rnn**

**Suggested Method for Running Python Scripts**

**GO TO “PA Term Project\_submit/Programs”->run\_analysis\_batch.py**

**I suggest running the python files with the following batch file: run\_analysis\_batch.py. Make sure to send to change rootDir to the path to “<root-path>/PA Term Project\_submit”. This will run the following files.**

**After that run, the following three R Scripts:**

**1) Granger\_Causality\_test.R**

**2) Correlation\_Matrix\_R.R**

**3) rnn\_project.R**

**REMEMBER TO SET ROOTPATH to the path of “PA Term Project\_submit”**

**Output Files (In Data/Results Folder):**

**1) ARIMAX\_first\_diff\_direction.csv (Directional Accuracy of ARIMAX)**

**2) RNN\_first\_diff\_direction.csv (Direction Accuracy of RNN)**

**3) ARIMAX\_rmses.csv (RMSEs of ARIMAX)**

**4) RNN\_RMSES.csv (RMSEs of RNN)**

**5) Granger\_Causality\_test.txt (List of summary outputs of Granger Causality tests)**

**6) Correlation\_Matrix\_All.csv (Pearson Correlation Matrix and Correlations of Sentiment on Future VIX Values)**

**7) all\_time\_series\_against\_VIX.png (PNG File of the time series of sentiment and VIX)**

**Input Files (In Data and subfolders of Data):**

**Varied, see the info on files below and the note below.**

**==================================================================**

Note: I write a suggested file to run because i did some of my data-preparation as new headlines came in. Therefore, it is highly suggested that you DO NOT RUN these programs or else it will pull in more data and change results.

**==================================================================**

**Info on files that are run in run\_analysis\_batch.py:**

1) creating\_half\_hour\_news\_sections.py: This will group all the headlines by half hour period.

**-Input File:**

a) All\_Metadata\_with\_n\_grams\_and\_lemmas.pickle (all the metadata as a pandas data frame)

**-Output Files:**

a) articles\_by\_date.pickle (metadata for all the articles in a pandas data frame per half hour),

b)topics\_by\_date.pickle (list of topics sorted by frequency), c)All\_Metadata\_with\_n\_grams\_lemmas\_and\_better\_time.pickle (all the metadata as a pandas data frame)

2) create\_top\_news\_cluster.py: Finds the “Top News” articles by half hour.

-**Input Files:**

a) articles\_by\_date.pickle (metadata for all the articles in a pandas data frame per half hour),

b)topics\_by\_date.pickle (list of topics sorted by frequency),

c)All\_Metadata\_with\_n\_grams\_lemmas\_and\_better\_time.pickle (all the metadata as a pandas data frame)

**-Output File:**

a) per\_half\_hour\_headlines.pickle (top headlines for the half hour)

3) makingSentimentScores.py: Create the sentiment scores for the half hour by averaging all the top news articles. merging these scores.

**-Input Files:**

a) per\_half\_hour\_headlines.pickle (top headlines for the half hour)

b)All\_Metadata\_with\_text\_blob.csv (all the metadata with text blob sentiment scores)

c)All\_Metadata\_with\_Vader\_Sentiment.csv (all the metadata with Vader sentiment scores)

**-Output Files:**

a)sentiment\_scores\_DBSCAN\_training\_and\_test.csv (average sentiment scores per half hour)

b)All\_Data\_Original\_DBSCAN.csv (merged sentiment scores and VIX values)

4) saving\_lags\_to\_csv.py: Creating the lagged values for all variables

**-Input Files:**

a)All\_Data\_Original\_DBSCAN.csv (merged sentiment scores and VIX values)

**-Output Files:**

a)All\_Data\_Original\_DBSCAN\_plus\_lags.csv (merged sentiment scores and VIX values with lags for each)

5) calculating\_ar\_test.py: Doing the ARIMAX evaluations

**-Input Files:** All\_Data\_Original\_DBSCAN.csv (merged sentiment scores and VIX values)

**-Output Files:**

a)ARIMAX\_RMSE.csv (RMSE of forecasts in test set)

c)ARIMAX\_direction.csv (direction of forecasts in test set)

d)Forecast\_first\_diff\_direction.csv (first difference of direction of forecasts in test set)

6) plotting\_time\_series\_of\_vars.py:

**-Input Files:** All\_Data\_Original\_DBSCAN.csv (merged sentiment scores and VIX values)

**-Output Files:** all\_time\_series\_against\_VIX.png (picture of the time series, used for table 1)

**INFO ON R-Scripts:**

1) Correlation\_Matrix\_R.R: Creates the correlation matrix and correlations vs. time steps of the VIX

-**Input File**: VIX\_and\_Sentiment.csv (This is the training set VIX and Sentiment Scores)

**-Output File:** Correlation\_Matrix\_All.csv (This is the output of scores)

2) Granger\_Causality\_test.R: Runs the Granger Causality Test

-**Input File**: VIX\_and\_Sentiment.csv (This is the training set VIX and Sentiment Scores)

**-Output File:** Granger\_Causality\_test.txt (This provides a text file of the summary of the granger tests)

3) rnn\_project.R: Runs the neural network for the project

**-Input Files:**

a)All\_Data\_Original\_DBSCAN\_plus\_lags.csv (merged sentiment scores and VIX values with lags for each)

**-Output Files:**

a)RNN\_RMSES.csv (accuracy of forecasts in test set)

b)RNN\_Forecast\_cor.csv (correlation of forecasts in test set)

c)RNN\_Forecast\_dir.csv (direction of forecasts in test set)

d)RNN\_Forecast\_dir\_first\_diff.csv (first difference of direction of forecasts in test set)

**ALL\_FILES INFO**

**newsapi\_pull\_only\_metadata.py:**

Gathering the News Data. Change the Dates at the top for the days you want to pull. Saves the values to text files.

**VIX\_pull.py:**

Gathering the VIX Data. Saves VIX Data to csv

**Data Preparation:**

**convert\_metadata\_to\_pandas.py:**

Puts the metadata into a pandas data frame. Saves to a csv.

**stop\_words\_test.py:**

Removes the Stop words.

**gettingVaderResults.py:**

Calculates the Vader Sentiment of the Headlines.

**gettingTextBlobResults.py:**

Calculates the Text Blob Sentiment of the Headlines.

**Data Preparation:**

**gettingNGrams.py:**

This creates the Ngrams for the document.

**lemmanize\_ngrams.py:**

This lemmanizes all the ngrams in the document

**creating\_half\_hour\_news\_sections.py:**

creates the sections of the news by the half hour.

**create\_top\_news\_cluster.py:**

This finds the top news articles for the half hour period given the sectioned clusters from creating\_half\_hour\_news\_sections.py. This does so with the DBSCAN inspired algorithm

**makingSentimentScores.py:**

Given the top articles, this makes the sentiment scores and averages them for the half hour.

**Evaluation:**

**calculating\_ar\_test\_Set.py:**

Calculates the test ARIMAX results. Outputs to csv. In this the RMSE’s and the

**saving\_lags\_to\_csv.py:**

Creates lags of the data (important for the RNN and ARIMAX).

**plotting\_time\_series\_of\_vars.py:**

Plotted the time series of the paper.

**granger\_causality\_test.R:**

Provides the granger causality test for the data.

**rnn\_project.R:**

Provides the RNN Values for the VIX.

**Correlation\_Matrix\_R.R:**

Creates the correlation matrix for both the VIX and the time steps ahead vs. the VIX