Data Analytics and Machine Learning | Prof. Lochstoer

==========

Problem Set 6

==========

Use the business_insider_text_data.csv and vix_data.csv datasets available on BruinLearn for this exercise. The Business Insider data is downloaded from www.kaggle.com (a really cool website for data and code, if you haven't seen it). You can and should look at the raw data using Excel before starting this exercise. The vix_data.csv file contains two columns where "label" indicated whether the closing value of the VIX is higher (1) or lower (0) than the closing value of the VIX for the previous trading day. The Business Insider data contains various headlines downloaded from www.businessinsider.com.

Creating a sentiment index from text data

1. Run the codes from the lecture to preprocess the data and create the text files corresponding to each date in the news headline data. That is, remove numbers, make all lower case, remove stopwords, stemming, etc. Use PlaintextCorpusReader to load the corpus. Now, each document corresponds to a different date in the dataset.

2. As in the lecture note, create a DocumentTermMatrix, call it dtm. Run the line "dtm.iloc[5:10, 201:210]." Notice that the matrix is quite *sparse* (a lot of zeros).

3. As in the lecture note, create a frequency matrix as the column sums of the DTM. Show in a bar plot the frequency of words that occur more than 25 times.

4. Create a wordcloud of the 20 most frequent words. Based on this (and 3.), how would you characterize the typical headline in terms of the news subject? Are there words that, intuitively, can matter for the stock market returns that day?

5. Create the endogenous variable "y_data = df_full['label']" and the exogenous matrix "x_data = dtm." You will try to construct an index based on the words in dtm that predicts the direction of changes in the VIX.

6. Split the data into a training dataset, based on data up to and including 2016-12-31. The remaining data should be used for actual out-of-sample testing.

7. We will first let the logistic regression create the word-based index. That is, try to fit a regular logistic regression using y_data and x_data and the training dataset. Explain why this doesn't work.

8. Next, run a logistic regression with a ridge constraint using cross-validation and the training dataset. Why does the regression routine work now (i.e., why does it give an answer (a coefficient vector; no meltdown))? Explain.

9. Using .C_ (the penalizing term chosen by the cross validation), what (if any) are the words chosen and their associated coefficients? Comment on your results.

10. Now, create instead a pre-defined sentiment word list:

    sent_words =
    ["trump","invest","growth","grow","high","strong","lead","good","risk","debt","oil","loss","war"
    ,"rate","hous","weak"]
    dtm_sentiment = dtm[sent_words]

    Run the ridge regression with "x_data_pre=dtm_sentiment" using cross-validation and the training sample. Create a bar plot with the words on the x-axis and the coefficients on the y-axis. Comment on differences and similarities to the case in 9. Again, get the coefficients using c_.

11. Create the ROC curves for the sentiment model in 10, using C_ to get coefficient vector. Is it better than random? You likely want to use the predict function to get the model predictions.

12. Now, using the ***test sample*** and the model in 10, what is the proportion of days the model would have made the right prediction in this new sample? Is it better than random (50/50)?