Omar Sadek

## MAIS 202: Deliverable 2

i- The aim of this project is to train a machine learning model to predict how the S&P500 index will perform on any day, given that day's top news headlines.

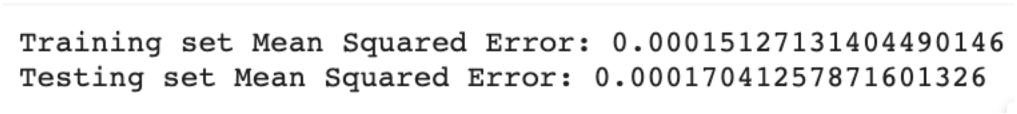
ii- As proposed in Deliverable 1, I will be using the two datasets I had listed there. One of which is from Kaggle and it has the top 25 news headlines for any day between August 8th, 2008 and July 1st, 2016. For that same time period, I will use the Yahoo Finance database to retrieve the daily opening and closing prices of the S&P500 index, and based off that, calculate its daily percentage change. The Kaggle dataset is originally designed to be a classification problem using the Dow Jones index; however, I have chosen to make my project a regression problem and have decided to use the S&P500 index instead.

After cleaning up the dataset, I was left with 1986 days with data, and each one of those days had 25 headlines, which gave me plenty of language to process and train my model on. The dataset was split up into 70% training and 30% test sets. I also made the decision to lemmatize all the headlines and remove stop words to prevent the model from focusing on commonly used filler words that do not have much meaning and would simply add unnecessary noise.

iii- I decided to try two different regression models for my project so far and they were RandomForestRegressor and SVM. Both yielded very small Mean Sum of Squared Errors for both the training and testing sets, which was impressive given that I had relied on countVectorizer rather than some of the more complex vectorizing methods available. Initially, I had planned on using a Facebook AI Lab developed library called FastText to create my X matrices; however, I was unable to figure it out in time and instead decided to use countVectorizer as it was easier to understand and work with.

I had many challenges when implementing my model and most were due to finding the right model that used NLP data in a regression context. Most NLP related datasets tend to be classification tasks; however, as I chose to do my project as a regression task, it was a lot more difficult to find the appropriate model. As for regularization techniques and setting hyper-parameters, I plan on focusing more on doing that for my next deliverable, especially after vectorizing my data differently.

iv- The model performed very well given the train and test sets, which was surprising to me. Given that the method used to vectorize the dataset can be considered somewhat primitive and lacks the ability to analyze context, I thought that the model would be a lot less accurate. You will find attached below a screenshot of the MSE calculations for the RandomForestRegressor model, which yielded MSE's that were slightly smaller than what was yielded from the SVM model:



```
Training set Mean Squared Error: 0.00015127131404490146
Testing set Mean Squared Error: 0.00017041257871601326
```

Given the good results, it seems that this project is feasible; however, I am still determined to see how the MSE's would change if I were to vectorize my data differently and hope to see improvement.

v- For my next steps I will definitely attempt to vectorize my data in a different way such as using FastText or Word2Vec. Both of these methods seem to have many pros as they do a good job at capturing context. I believe the model will also need a validation set to ensure that there is no overfitting and to properly set all hyperparameters. I am excited to see what more this model can do, when it is done and complete!