DATA 598 / Spring 2020 / CAPSTONE PROJECT

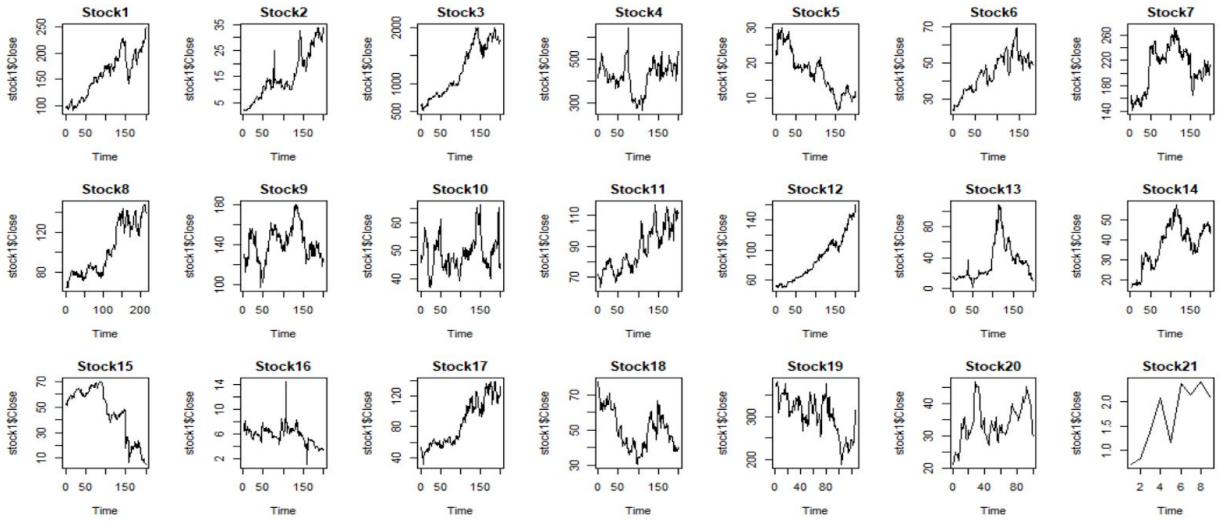
Group 3: Harsha Koonaparaju - Hrishi Kulkarni - Jeffrey Lai - Mabel Li - Liem Luong - Shuya Ma

Exploratory Data Analysis (EDA)

For exploratory data analysis, we generated summary statistics on 21 stocks. Then for each stock we generated a time series plot. The summary statistic table and the plots of 21 stocks are shown below. We can see from the plots that the time range of stocks is not the same. Most of the stocks were from January 1, 2016 to October 18, 2019, whereas some stocks such as number 20 had observations from December 1, 2017 to October 25, 2019 and number 21 only had 9 observations from December 6, 2019 to October 18, 2019.

```
    stock_id          Date          Open              High              Low               Close             Volume        
Min.   : 1.00   10/11/2019:  20   Min.   :   0.70   Min.   :   0.83   Min.   :   0.20   Min.   :   0.71   Min.   :     2500  
1st Qu.: 5.00   10/18/2019:  20   1st Qu.:  34.50   1st Qu.:  35.71   1st Qu.:  33.07   1st Qu.:  34.44   1st Qu.:  6179325  
Median :10.00   10/25/2019:  20   Median :  63.07   Median :  65.00   Median :  61.45   Median :  63.40   Median : 13463350  
Mean   :10.13   10/4/2019 :  20   Mean   : 159.41   Mean   : 164.07   Mean   : 154.57   Mean   : 159.80   Mean   : 45799991  
3rd Qu.:15.00   9/13/2019 :  20   3rd Qu.: 143.94   3rd Qu.: 147.39   3rd Qu.: 140.21   3rd Qu.: 143.97   3rd Qu.: 33603450  
Max.   :21.00   9/20/2019 :  20   Max.   :2008.27   Max.   :2050.50   Max.   :1958.26   Max.   :2012.98   Max.   :951988300  
                (Other)   :3710                                                                                                
```

Summary Statistic



Time Series of 21 stocks

Outline/justification of proposed methodology

- First, we split the training data into a training (90%) and a test set (10%) for each stock
- Next, we trained two different models for each stock id on the training dataset:
    - ARIMA - through ‘auto.arima’
    - ETS
    - Mixed model - take all the models, weight together their output per stock id by their share of the sum of CRPS scores for the stock id.

- Following this, we evaluated each model using the CRPS score on the test dataset:
    - We found that auto.arima with Box-Cox lambda parameter of 0 has the lowest sCRPS score on the test dataset (see figure below)

- The chosen model was used to generate 500 sample paths for the closing price for each stock from November 11, 2019 to January 31, 2020.

```
                                                auto_arima          auto_arima_lambda0
                                                 0.8016235                   0.7377000
auto_arima_lambda0_openlag1          auto_arima_lambda3                         ets
                  0.8630389                   0.9798977                   0.7897950
                mixed_model
                  0.9079231
```

The sCRPS score on test dataset of ARIMA and ETS.

Provide enough implementation details so that an engineer could deploy your method in production

- An engineer should take the parameters from the models and store them in a database with the key being the stock id (e.g., maybe the engineer stores a pkl file with all the models and parameters if small enough).
- At startup the machine(s) which will handle inferencing load the models into memory.
- The inferencing machines respond to API requests given a
    - stock id
    - date to predict the sample paths up to
    - Number of sample paths to return
- The API then returns the given number of sample paths per stock id up to the date provided in the API call.

How we would scale up this model for more stocks

Scaling up this model to more stocks (assuming the model fits in memory as before) should be a matter of:

1. Provisioning more compute instances to handle greater traffic.
2. Storing the parameters for each model in some low-latency data store (e.g., Azure Cosmos DB, Amazon Dynamo DB, etc.) and gathering them at runtime for inferencing.

Since our model is lightweight, the memory required to fit the parameters is low, so the engineering team may be able to fit the parameters in a local persistent memory store as well.

Reference:

Evaluating Probabilistic Forecasts with scoringRules
https://arxiv.org/pdf/1709.04743.pdf