FT5005: Assignment 3 (10 Marks in total)

**Due: 11:59pm of April 21st Sunday Midnight 2024**

**Submission Formatting Requirements**

1. Please submit one \*.ipynb for the codes for Questions #1-3. Zip all your files into one zipped file for submission on Canvas.
2. You can add comments to help TA understand your codes. Avoid including unnecessary codes or comments.
3. Remember to set random seeds so TA can verify your results when needed.
4. TA can deduct your points up to 2 out of 10 if you included unnecessary codes and comments.

**Dataset:** I provided two small datasets for you to practice. The duration of the dataset is one year and is too short in practice but it can save your time testing and training for this assignment’s purpose. There are two files: stock\_tweets.csv and stock\_yfinance\_data.csv. There are only two stocks: one is Apple and the other is Microsoft. The Yahoo finance dataset is a typical dataset. The stock tweets include Twitter Tweets for those two firms.

**Q1 (3 marks) BERT and Clustering**

1. Please use “cardiffnlp/twitter-roberta-base-sentiment” and Huggingface to convert each Tweet to an embedding vector. This is the second piece of the new sample code that I just uploaded. Interestingly, standard BERT sentiment analysis and FinBERT sentiment analysis both do not work well on this dataset. Only the models trained on Twitter data can work well. So we use ROBERTA, not BERT or FINBERT in A3.
2. Please use Kmeans clustering with K=2 to cluster tweets into two clusters.
3. In this case, we know the label of each tweet and the label is the company name: Apple or Microsoft. Please print out the confusion matrix that compares your clustering result with the actual label (company name). You are required to choose the clustering results matching that leads higher accuracy. I mean by clustering, you only know cluster A and cluster B. You do not know which one is Apple. You just use the matching (cluster A is Apple versus cluster A is Microsoft) that leads to higher accuracy.

**Q2 (7 marks) Time Series Forecasting**

* 1. (1 mark) The main input feature is the Huggingface Transformer sentiment analysis.
     + We use the sentiment score created by ROBERTA in the sample code. When the sentiment is positive the value is +SentimentScore, when the sentiment is neutral, set the score =0, when the sentiment is negative, then the score is -SentimentScore.
     + This sentiment score is computed at the tweet-level, aggregate the positive or negative sentiment score of each stock to the daily level by median of all tweets on that date. If there is no tweet on a particular date, set the score at 0. So there is one new input feature by sentiment score.
     + To predict Y at time *t*, you need to use sentiment on time *t*-1.
     + Provide the summary statistics of this new input feature for TA to verify your works in this part.
  2. We will try two outcome variables for prediction. The return is defined as
     + Return is “(ClosePrice\_(t+1)/ClosePrice\_t) - 1”
  3. **Y variable**. The two outcome variables are
     1. Binary Yt: 1 means Apple’s daily return >= MSFT’s return. 0 means Apple’s daily return < MSFT’s return.
     2. Weekly volatility of returns: for the date t’s outcome variable, use the standard deviation of daily return on trading date t, t+1, t+2, t+3, and t+4.
     3. I created a new EXCEL file that shows how these variables are created.
  4. Prediction performance metric for binary classification is **accuracy** and for predicting volatility is **“R-square.”** We use R-square because this is more interpretable than RMSE and you can learn how well you can predict the outcome variables.

A white background with black text

Description automatically generated

* 1. You need to try 2 methods: LSTM and LightGBM.
  2. For both methods, the input features are column L to column W in the new EXCEL file.
     + You add lagged t-1, t-2, t-3, t-4, t-5 about which stock is the winner. In LSTM, this implies the window size is 5. In LightGBM, you need to add 5 lagged variables as the 5 additional input features.
     + How many times Apple is the winner in the last 5 trading days.
     + Volatility of daily return of the focal firm on (t-1, t-2, t-3, t-4, t-5). This is only one column! This does not mean lagged Y. For LightGBM, this means there is only one additional input feature. ~~For LSTM, you need to be careful so for predicting volatility from t to t+4, you only use information before and including t-1.~~
  3. To make things easier (with LSTM), we do the following for Train and Validation. There is no test set due to our shorter duration of the dataset.
     + Validation set is the last 2 months in the dataset.
     + Training set is all other rows of data.
     + If you do not have enough variable for calculating lagged variable or future volatility, then just do not include those rows of data in your training or validation.
     + Ideally, we should do time series split that I covered earlier this semester. But due to shorter deadline, we go with the easier solution without time series split.
  4. There are 3 cases of prediction results:
     + Cases 1 and 2: Classification of the winner by LSTM and LightGBM. One key point of this assignment is how to add lagged Y into your model. Be careful in this step.
       - The prediction performance may be around 50%, don’t worry about this.
     + Case 3: LightGBM for volatility. For the volatility case, the current design does not allow lagged Y and so we will not use LSTM.
     + You are allowed to tune your algorithm by any method or if you run out of time, you do not need to tune. But your grading will also depend on your prediction performance metric. For each case, we -1 for those cases with about bottom 10% prediction performance when compared with submissions from your classmates. We may be stricter for case 3, which is more predictable by sentiment scores. Like other assignments, you do not need to tune thoroughly, you only need to make sure your prediction performance is reasonable. “10%” is not an absolute threshold. We will only penalize cases with performances that are significantly worse than typical prediction performances.