Stock Price Prediction Project

This project involves using deep learning and traditional machine learning techniques to predict the closing stock price of S&P 500 companies, using both time-series data and various features about each company. The main objective is to predict the "Close" price based on past data of stock exchange prices, volume, and relevant news events.

The given files are “company_info.csv” which contains details about the companies listed and “company_stock_details.csv” which contains details about stock close price for 495 companies across the span of 2 years with respective volume of trading and some news for each respective company of daily basis.

First, the code starts by importing necessary libraries to achieve the task, feature selection, hyperparameter tuning, model training and predicting the close price. There are a few models that need to be downloaded first, then user can run this code “!pip install {module_name}” upon seeing this error message: “ModuleNotFoundError: No module named {module_name}.”

Next section is preprocessing the dataset and cleaning it. The code starts by combining both dataset into once using “left_join” based on “Symbol.” The number of missing rows will be printed for each column, and the two rows with missing Symbol, date, and companies value. Then, for missing companies, date, and close, we dropped the row. Then, for each missing value of news, we replace it by 0 because it represents volumes. Continuing, the code will print two datasets, “train_data_lstm.csv” and “test_data_lstm.csv” that will be used as input to the sequential model. Lastly, we reformatted it such that each row will comprises of that day feature (close price, volume, news) and features from past five days, differentiated by the ‘lag’ names. This data is stored as “train_data.csv” and “test_data.csv” for both train and test data respectively. To successfully run this part, make sure the two datasets provided are in the working directory and all the libraries have been successfully imported.

The next section is Feature Selection. We performed feature selection using 5 main methods, the first cell does 4 methods, Lasso, SVR, RandomForestRegressor, Affinity propagation. The results for each method will be stored in “lasso_top_features”, “svr_top_feature”, “rf_top_features”, and “mffs_top_feature” respectively. The second cell in feature selection will perform the multi filter neural network and the result will be stored in “selected_feature.” Printing any of the above variables will give the selected features for each feature selection method. Running this part requires a lot of time, especially the SVR feature selection. For that reason, user can just use part of the train data.

Next part is hyperparameter tuning. In general, we performed two methods for each model, that is Random Grid Search and Optuna. We did hyperparameter tuning for ANN, GRU, RNN, LSTM, CatBoost, XGBoost, Random Forest, and Decision Tree. Running each cell will print the selected hyperparameter based on the chosen method. The combination of these hyperparameters then will be supplied for train and testing the model. Hyperparameter tuning will also takes quite a while.

Last part is train and test each model using the 7 selected features (5 from the feature selection, one is using whole feature and using only close price). We defined the dataset to train and test the model, the train consists of whole companies, some companies, and AAPL only. Then, we

test to predict the close price for test data only for AAPL symbol. We use expanding window cross validation for fitting the model to the train data. Then, we output the result of each testing iteration of each model to csv file, in this case we save the R^2, RMSE for train data and also RMSE for test data.