SchoolNova Computer Science 202

Homework 22

Due 3/27/2021 on Google Classroom

Task 1

Using PDR's .get_data_tiingo(), download the stock data for Microsoft from January 1st, 2010 to the present time.

Task 2

Convert the data to a numpy array. Generate the daily change in the stock price. Generate three lags for the change in the stock price. After that don't forget to delete the first three rows since we do not have all of the lags for those days.

Task 3

Using sklearn's train_test_split(), divide your data into a training set and a test set. Feel free to choose your own test_size value. Using the linear regression model, generate your predictions.

Task 4

In a perfect world, your predictions would match the y_test values. In reality, your predictions will be quite different. However, they may still be better than a random choice. Plot your predictions and the y_test data, similar to class work.

Task 5

Consider the following approach to evaluate the quality of your predictions: calculate for how many days the *direction* of your predictions is equal to the *direction* of the observations (y_test). In other words, if you predict the price of MSFT to go up and the price actually went up then you count this prediction as "correct". Similarly, if the prediction and observations both show a decrease in price then this is also "correct". However, if the prediction and observation show different directions, you should count this prediction as "incorrect". Calculate and display how many correct and incorrect predictions your model generates. Also, calculate and display the percentage of your correct predictions.

Task 6

Explore how changing the test_size value in train_test_split() affects the percentage of correct predictions in your model. I suggest running a for loop and exploring the following test_size values: {0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9}.

Task 7

Will the quality of your predictions change if you use more lags? What about four lags? What about five lags? Can you identify a number of lags AND a test_size value, which generate the highest percentage of the correct predictions?