Task 2 Report

Naca Hitchman - 103072170

# Function Setup

I began by creating the declaration for my function, as well as providing descriptions for each parameter input.

A computer screen shot of a program

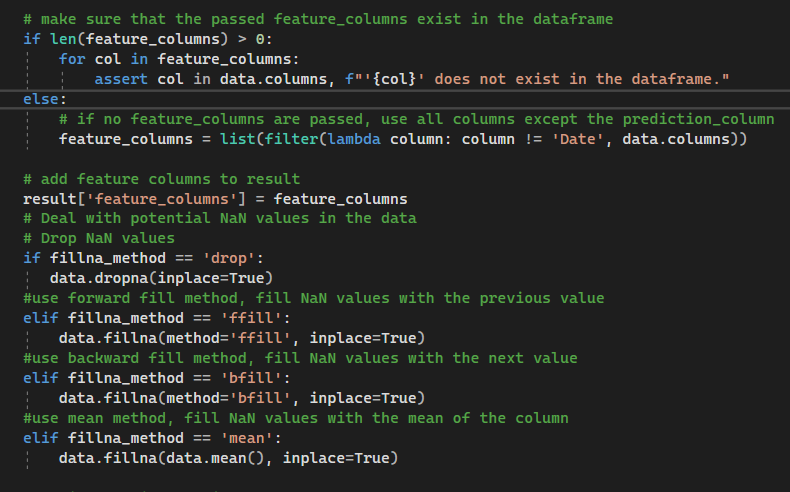
Description automatically generated

Then, I set up the system to check if the dataset file already exists, or if the ticker passed was an existing data frame. If the file doesn’t exist, the data is downloaded from yahoo finance and saved to a file using the .to\_csv() method. If the file does exist, it is read to a data frame. A result dictionary is also created, which will store all the relevant data to be returned.

A computer screen shot of a program code

Description automatically generated

Next, we check if the feature columns passed are in the data, then either drop or fill the nans based on the method passed as a parameter. Dropna is the default option, removing nan rows altogether from the dataset. The various fills use different methods to replace the nan values instead.



The dataset is then split into train and test, either by date if the split method is date, or randomly with a ratio. As data is a dataframe, the .loc() method can be used to compare the value of the ‘Date’ column, and compare it to the passed split date.

Both split date and split ratio are passed as parameters. Next, the datasets indexes and sorting are reset to ensure they are in the correct order.

A computer screen shot of a program code

Description automatically generated

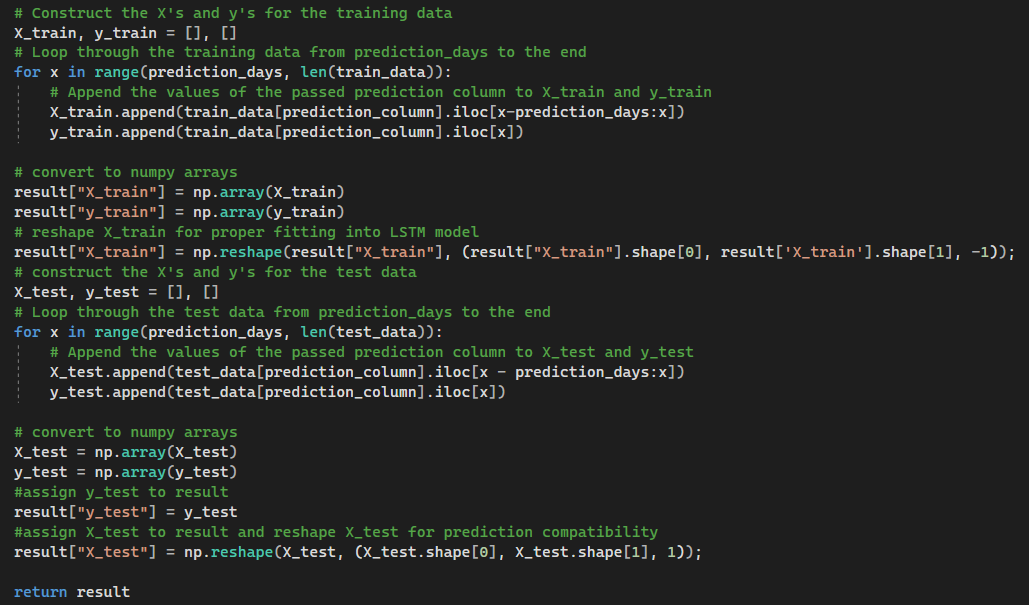
Next, if the specified, the data will be scaled. The min and max scale properties can be passed to the function. I’ve looped through each feature column passed, where a new min max scaler is created with the passed min and max values. Scaler.fit\_transform is used for the training data, to fit the scaler, and standard transform is used for the test data, to ensure the test data is scaled based on the scaling on the train data. Further, the values passed to the scaler for each column and reshaped to ensure it will be compatible.

The scalers are then saved to a text file if specified using pickle, then the scaled data is converted back into data frames and saved to the result dictionary.

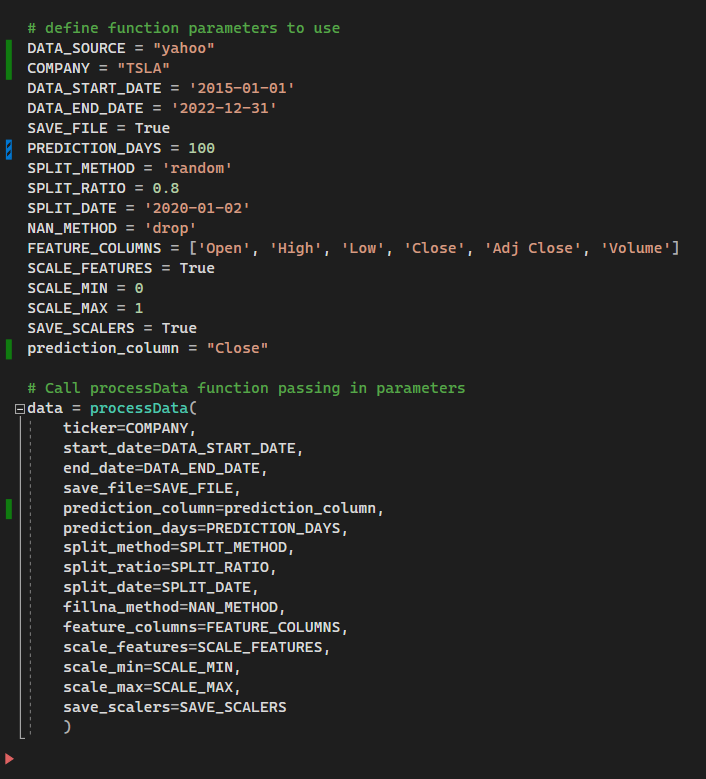
A computer screen shot of a program code

Description automatically generated

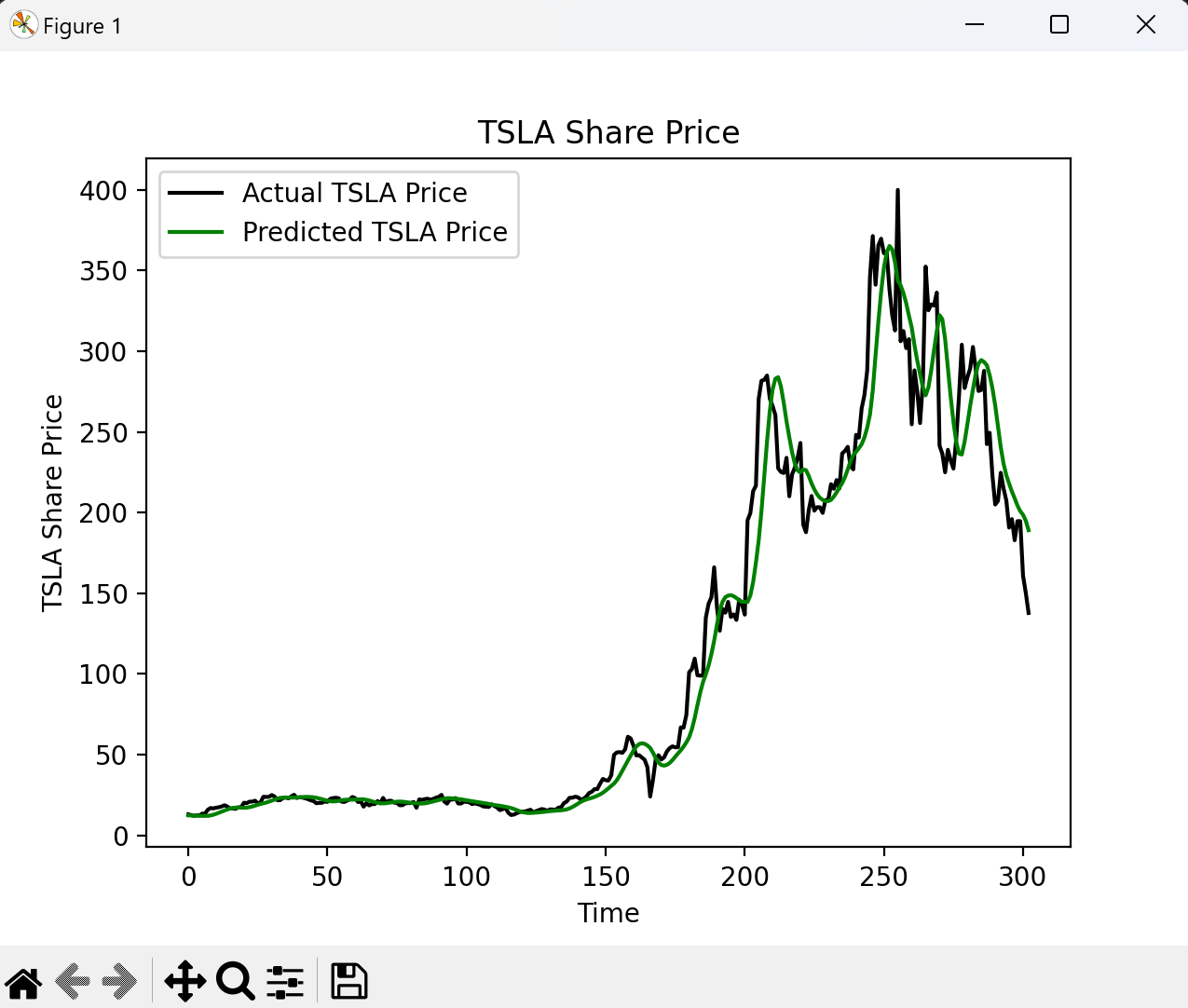
Lastly, the scaled data is split into their respective X and y arrays. This is also based on the prediction days as the original base is, and then saved into the results dictionary, before the result is returned.



Within the main method now, the parameters to be passed are defined, and sent to the function. The results are saved into ‘data’ and used to train the model and predict the prices as the original base did.



We can see this provides valid prediction results, with the next day being predicted.

 A black and white image of a number

Description automatically generated