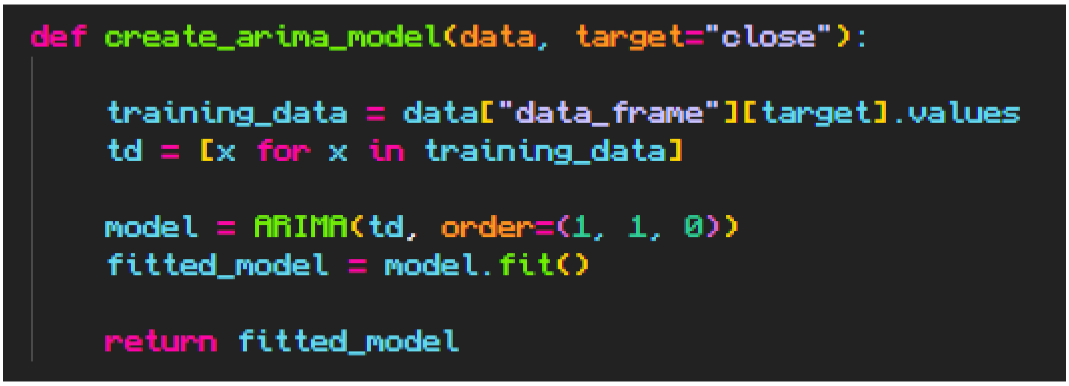
COS30018 Intelligent Systems – Task B6

Ryan Chessum 102564760

This week I had to use an ensemble approach to model prediction. A simple way of explaining an ensemble approach is that the average value of multiple machine learning models is used to make a prediction. The reason this is effective is that some models tend to guess higher than the actual price and others tend to predict lower. So if we find a value somewhere in the middle, we will be more likely to find the actual price for the next day.

Another useful model to use in the ensemble approach is the Arima model. Arima stands for “Auto Regressive Integrated Moving Average and as the name suggests is a regressive prediction algorithm that considers the moving average. This will be good for the ensemble approach as it will consider the trend of the price more heavily.

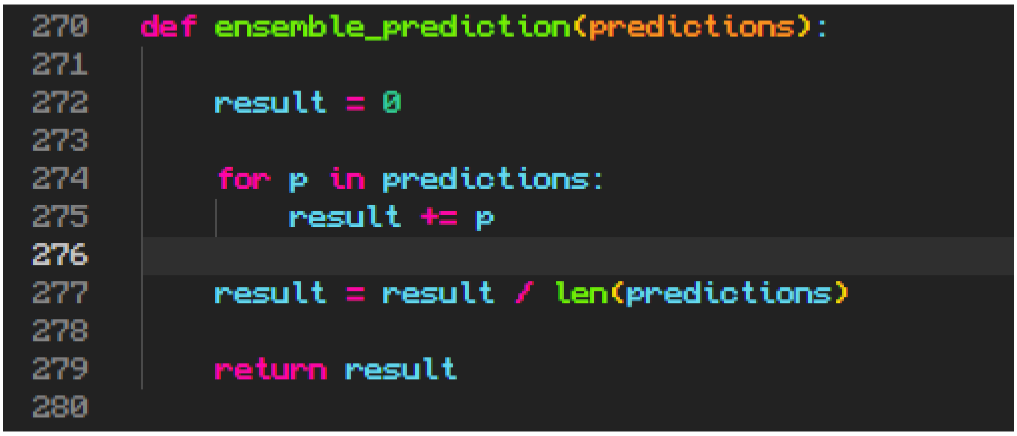
I made a simple function to build a simple arima model for us to use.



```
def create_arima_model(data, target="close"):

    training_data = data["data_frame"][target].values
    td = [x for x in training_data]

    model = ARIMA(td, order=(1, 1, 0))
    fitted_model = model.fit()

    return fitted_model
```

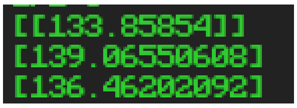
This does not use tensorflow to build the model so it works a bit differently.

I then made this function to make the ensemble prediction.



```
def ensemble_prediction(predictions):

    result = 0

    for p in predictions:
        result += p

    result = result / len(predictions)

    return result
```

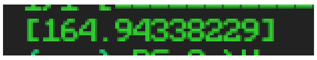
The predictions are made by each model then passed in through a list. The function will then find the average using the list.



```
[[133.85854]]
[139.06550608]
[136.46202092]
```

Here you can see the lstm prediction on top and the Arima predictions below that. The final prediction is the ensemble, which is the average of the two

After I got this working I added some more types of network to the ensemble. Here is the price with a gru network and rnn added.



```
 [164.943382229]
```