**Title:** Task B-5 Report

**Author:** Robin Findlay-Marks, s103603871

**Task information:**

Subtask 1:

For this subtask I had to make the program first of all ran the ARIMA prediction model, and then combined this in an ensemble with the original RNN type models. The first step of this was very easy. There were several tutorials showing me what I had to do so I followed them and made a working ARIMA prediction model in about 15 minutes. A big difference between ARIMA and the RNN models is that they often need to be remade after each prediction. Other than this the whole process of getting test and training data then making the model and predicting (or forecasting) is very straightforward and similar to the RNN models.

A screen shot of a computer program

Description automatically generated

The next step was significantly more tricky, despite the end solution being very simple. I quickly learnt that an ensemble ai model is a model consisting of two or more prediction models working together somehow. There are several ways of doing this including having an ensemble as a function with inputs of already trained models. I decided to try this method, because I already have trained models. I soon realised that this wouldn’t work because of the quirt of ARIMA prediction models where they often need to be remade after each prediction. I tried to find ARIMA prediction code where this did not happen but I couldn’t. After several hours of research and thinking I remembered that there is another type of ensemble prediction model that simply averages the outputs of the ensemble sub-model outputs. I quickly used this and made a working ensemble model.

In the end this was all the code needed to turn the LSTM and ARIMA prediction models into an ensemble.

A screen shot of a computer code

Description automatically generated

A graph showing the price of a stock market

Description automatically generated

Subtask 2:

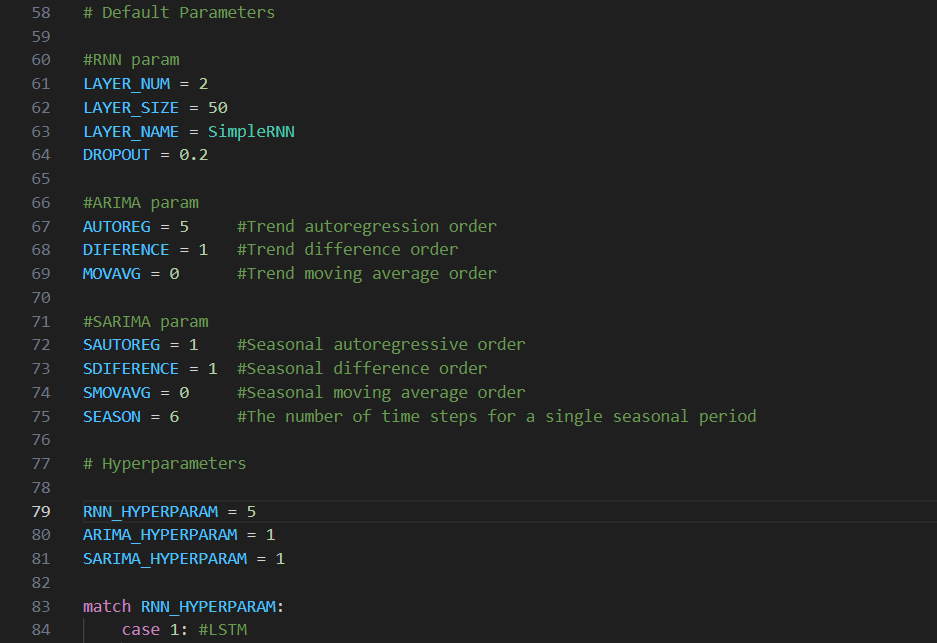
For this subtask first I modified the program to be able to use SARIMA instead of ARIMA. This was easy as SARIMA is really just ARIMA with extra parameters for a season component.

A screen shot of a computer

Description automatically generated   
  
Hyperparameters were added to the parameters.py file

I also renamed to old hyperparameters to RNN\_HYPERPARAMETERS

The new hyperparameters are ARIMA\_HYPERPARAMETERS for the arima parameters and SARIMA\_HYPERPARAMETERS for the seasonal arima parameters.



ARIMA

Hyperparameter 1

A black background with blue text

Description automatically generated

A graph showing the price of a stock market

Description automatically generated

Hyperparameter 2

A black background with blue text

Description automatically generated

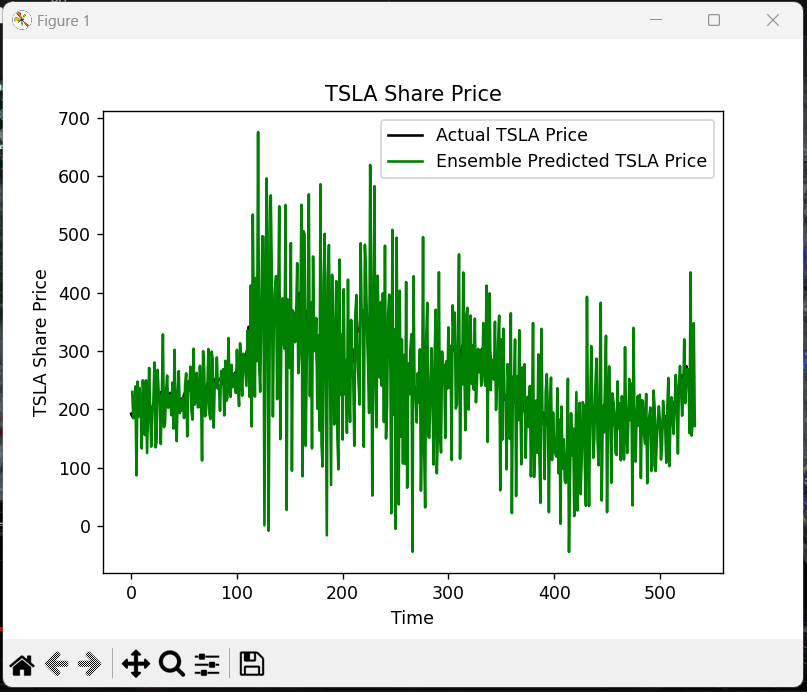
A graph showing the price of a stock market

Description automatically generated

Hyperparameter 3

A black background with blue text

Description automatically generated



Hyperparameter 4

A screen shot of a black screen

Description automatically generated

A graph showing the price of a stock market

Description automatically generated

Hyperparameter 5

A black background with blue text

Description automatically generated

A graph showing the price of a stock market

Description automatically generated

SARIMA

Hyperparameter 1:

A black background with blue text

Description automatically generated

A graph showing the price of a stock market

Description automatically generated

Other SARIMA hyperparameters

A screen shot of a computer

Description automatically generated

The second and third SARIMA hyperparameters were not run because they would take over 15 minutes to run

References:

**Brownlee, J 2020, *How to Create an ARIMA Model for Time Series Forecasting in Python*, relataly, viewed 4/10/2023, <https://machinelearningmastery.com/arima-for-time-series-forecasting-with-python/>.**

**Brownlee, J 2021, *Ensemble Machine Learning With Python (7-Day Mini-Course)*4/10/2023, <https://machinelearningmastery.com/ensemble-machine-learning-with-python-7-day-mini-course/>**