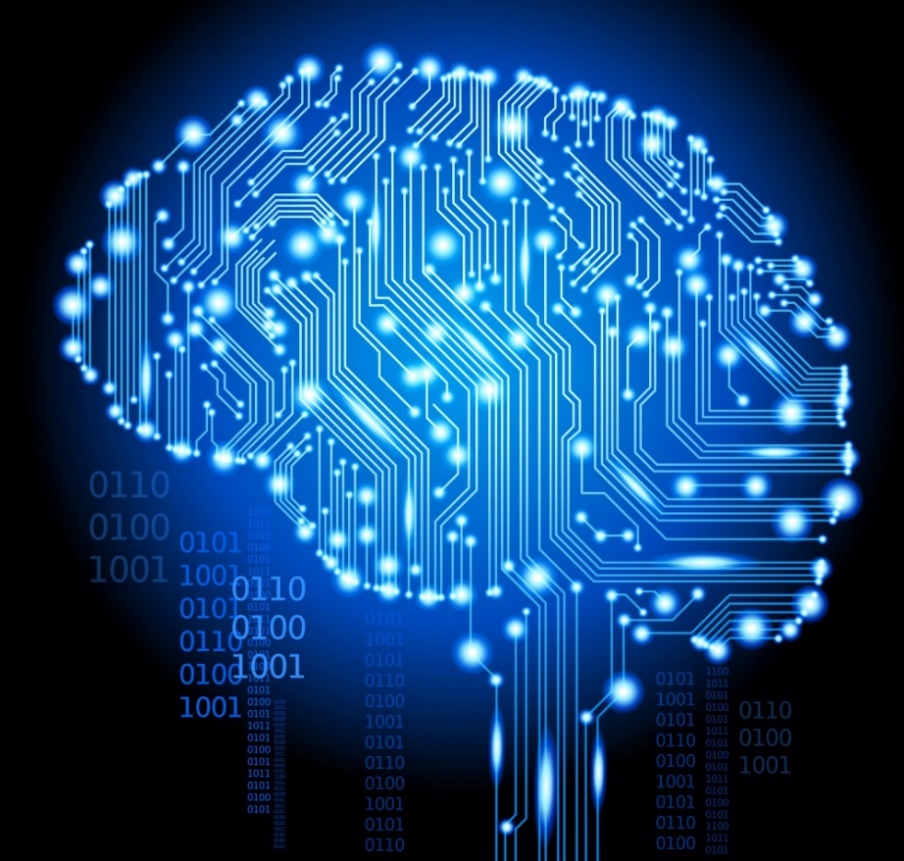


COS30018 – Intelligent Systems

**REPORT**

PROJECT ASSIGNMENT – OPTION B



**TABLE OF CONTENTS**

*Table of contents…………………………………………………………………………………………………………………2*

*I. Introduction……………………………………………………………………………………………………………………..3*

*II. Overall system architecture……………………………………………………………………………………………..3*

*III. Implemented data processing techniques………………………………………………………………………4*

*IV. Experimented machine learning…………………………………………………………………………………….6*

*V. Extension………………………………………………………………………………………………………………………..8*

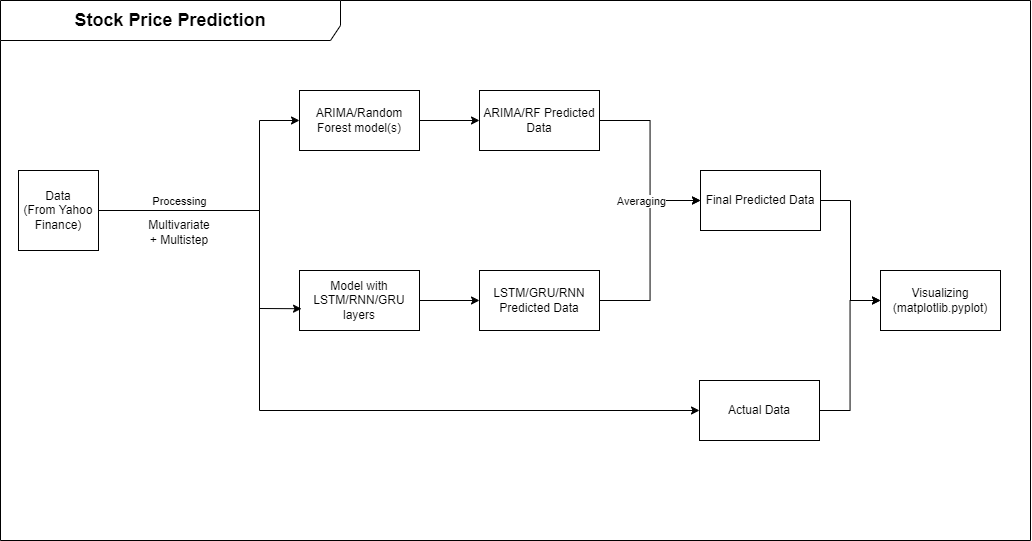
*VI. Demonstration………………………………………………………………………………………………………………8*

*VII. Summary……………………………………………………………………………………………………………………10*

**INTRODUCTION**

*My project is Stock Price Prediction using Python. The main idea is based on the use models of LSTM/RNN/GRU, ensembling with ARIMA and Random Forest technique, providing a solution to predict future stock price, including the multivariate and multistep prediction. This project is only developed for learning and research purposes, it does not meet the standards and reliability for practical purposes.*

**OVERALL SYSTEM ARCHITECTURE**



*Fig 1. My Stock price prediction project architecture design*

The system architecture for my project is provided in the above illustration. The instruction to run my project will be present in the “DEMONSTRATION” part.

**IMPLEMENTED DATA PROCESSING TECHNIQUES**

In my project, I created a file named “dataProcessing.py”, consisting of “processData()” method, which is responsible for downloading the data from Yahoo Finance (if the data has not been saved, if yes, load it), scaling and splitting the data, and returning the results.

Initially, the dictionary of “processedData” is created, containing all the outputs of the data processing stage, which would finally be the output for the method

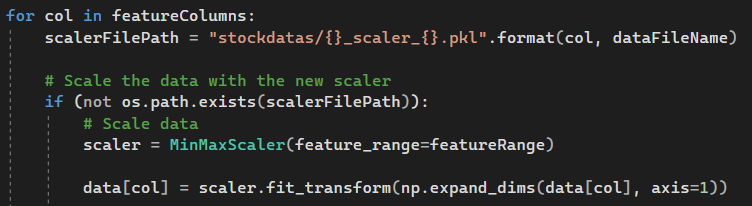
The data is retrieved from Yahoo Finance, and in a Pandas Dataframe format



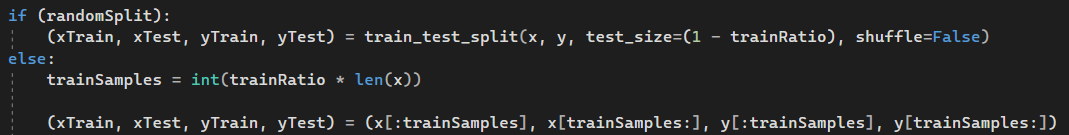
Or loaded if it has been dowloaded before:



Next, the data will be scaled using MinMaxScaler:

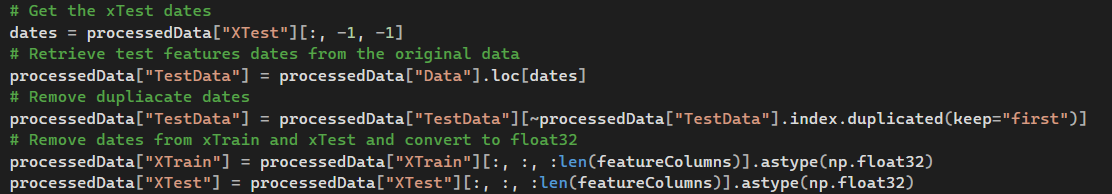


After that, sequence and target data (“x” and “y”) are created, before being splitted to xTrain, yTrain, xTest and yTest:



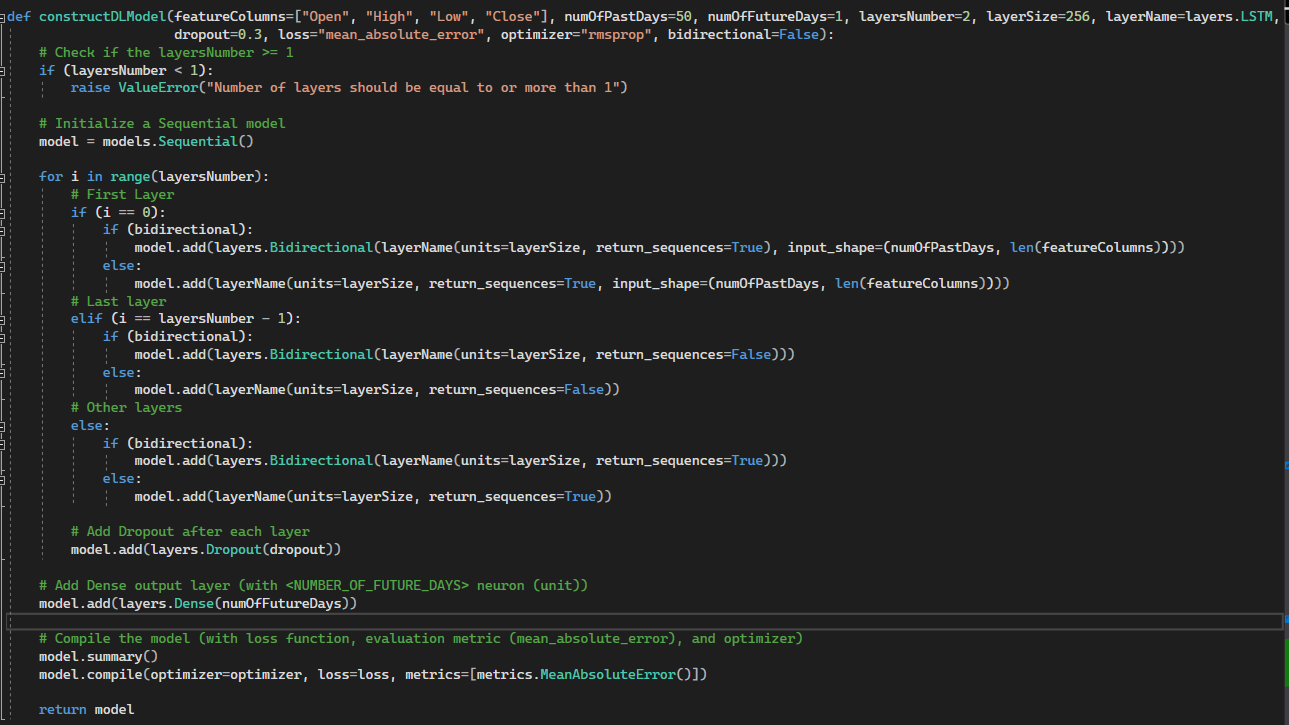
The “processedData” dictionary stored:

* The initial pandas dataframe: 
* The scalers used for scaling step: 
* The splitted train and test data:



**EXPERIMENTED MACHINE LEARNING**

In task 4, I created a method of “constructDLModel()” to create a Deep Learning model using LSTM/RNN/GRU layers:



<https://github.com/trungkiennguyen22082004/009_Python_Stock_Price_Prediction/wiki/Task-B4-%E2%80%90-Machine-Learning-1>

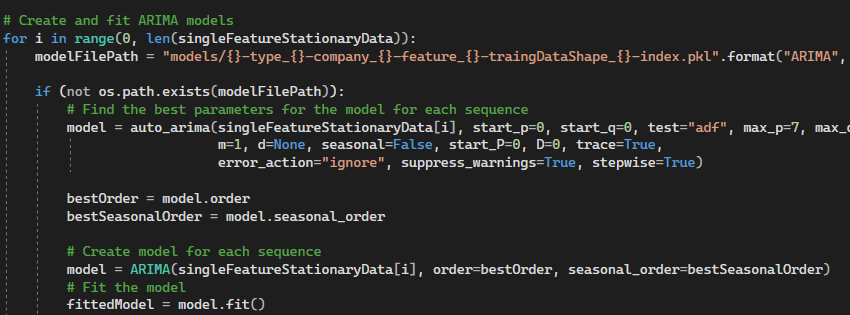
The multivariate and multistep problems were quite easy to solve.

* For multivariate, in the processing data, I simply splitted the “x” with all used feature columns, then when training and testing the model, use that splitted train and test data.
* For multistep, in the processing data, I reshape the targets (“y”) into 2D array, instead of a 1D list of a feature column from the data (each element of the reshaped array is a 1D array of k values, where k = number of want-to-predict days), then fit the model normally.

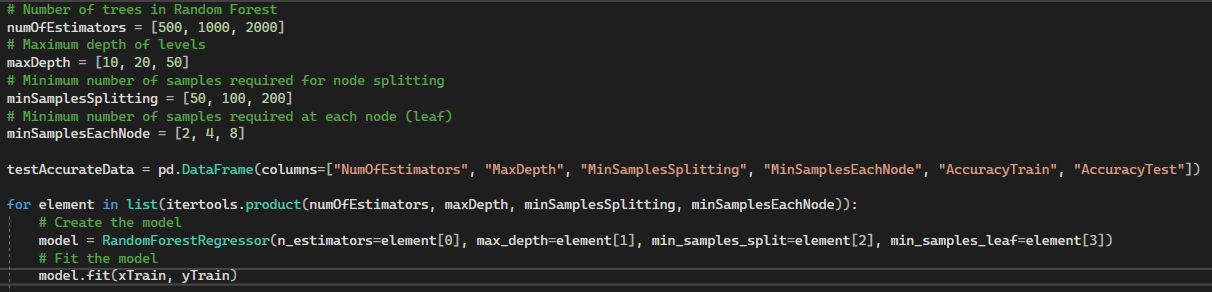
<https://github.com/trungkiennguyen22082004/009_Python_Stock_Price_Prediction/wiki/Task-B5-%E2%80%90-Machine-Learning-2>

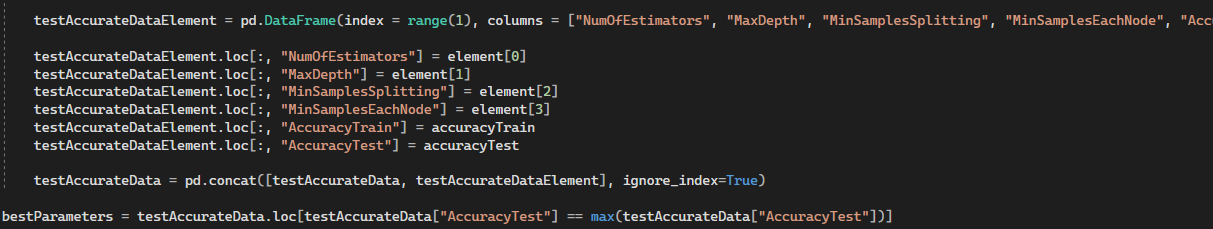
For the task B6, I chose 2 types of models to ensamble with the existing LSTM/RNN/GRU models. After create, fit and do prediction with the new models, I simply get the average values of the predicted data as the final output

* ARIMA: For this model, I used a lot of different ARIMA models, each corresponds to a future day in the prediction period.



* RandomForest: I used the RandomForestRegressor model from “sklearn.ensemble” module. The parameters for this model were decided using hyperparameters technique



<https://github.com/trungkiennguyen22082004/009_Python_Stock_Price_Prediction/wiki/Task-B6-%E2%80%90-Machine-Learning-3>

**EXTENSION**

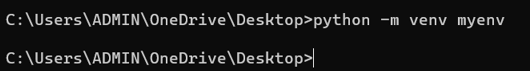
**DEMONSTATION**

My project use the following libraries/packages:

* Numpy (1.24.3)
* Pandas (2.1.0)
* Yfinance (0.2.28)
* Scikit-learn (1.3.0)
* Matplotlib (3.7.2)
* Mplfinance (0.12.10b0)
* Keras (2.13.1)
* Pmdarima (2.0.3)
* Statsmodels (0.14.0)
* Tensorflow (2.13.0)

Instruction to run my project:

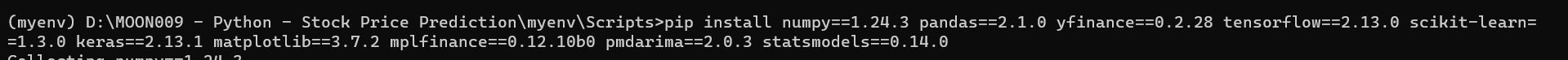
* Create a virtual environment:
  + python -m venv myenv



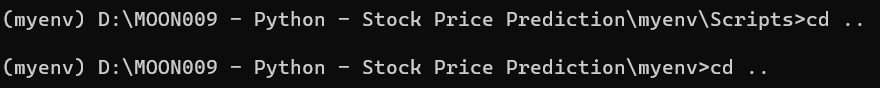
* Activate the virtual environment: change directory to myenv\Scripts and run the “activate.bat”
  + cd …\myenv\Scripts
  + activate.bat



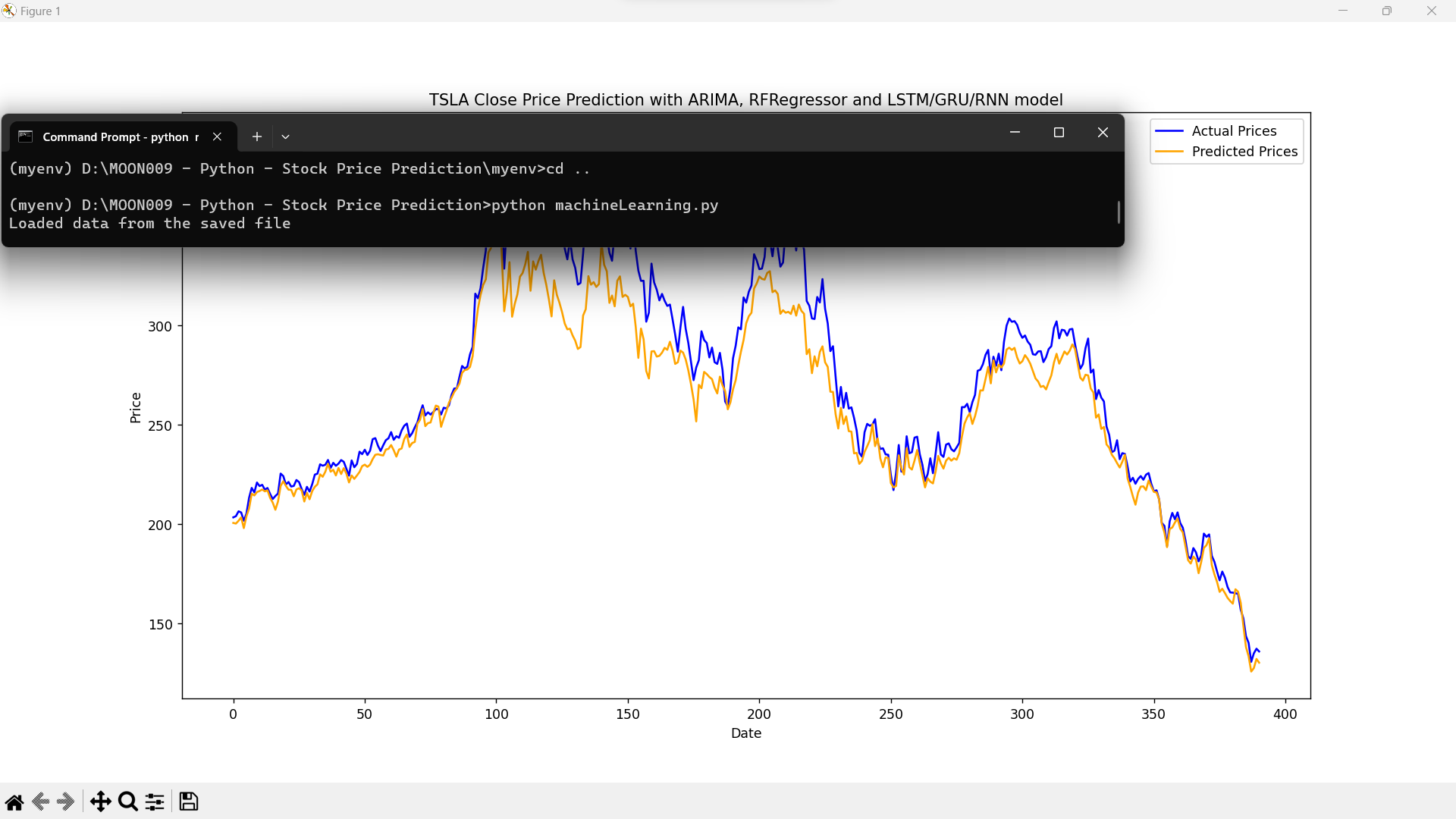
* Install the necessary libraries/packages:
  + pip install numpy==1.24.3 pandas==2.1.0 yfinance==0.2.28 tensorflow==2.13.0 scikit-learn==1.3.0 keras==2.13.1 matplotlib==3.7.2 mplfinance==0.12.10b0 pmdarima==2.0.3 statsmodels==0.14.0

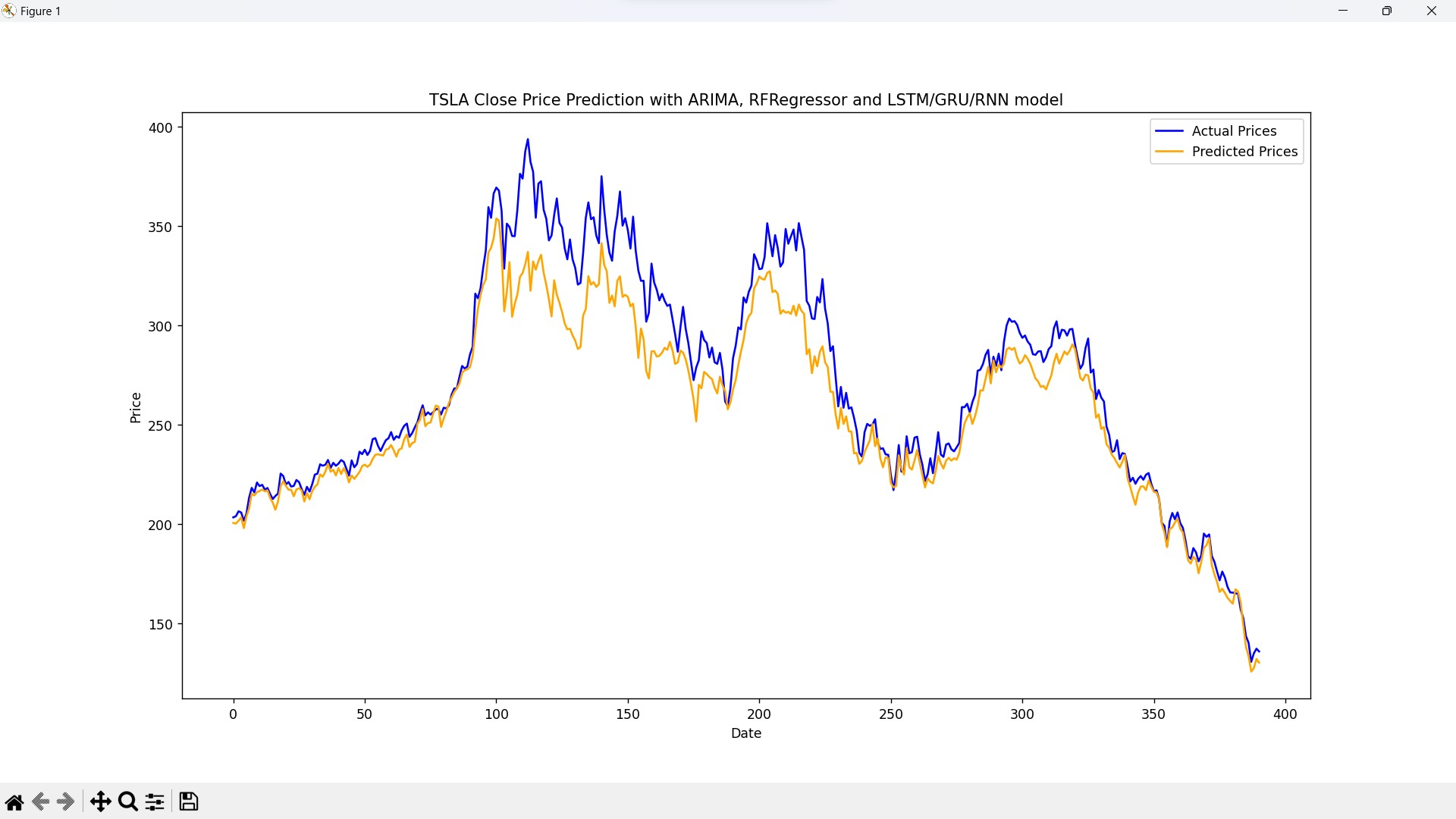


* Return to the project root directory:



* Run the “machineLearning.py” file:





*Fig 2. Final output*

**SUMMARY**

Throughout this semester, I believe that my work on the project B aligns well with the specified ULOs:

* My project explores and implements a range of techniques used in intelligent systems, including the use of LSTM/RNN/GRU models, ARIMA, and Random Forest ensembling models for stock price prediction, along with the detailed explanations in the weekly reports. I believed this demonstrates my practical understanding of “the AI/IA techniques”.
* While working with the project in the teaching period, I have applied some algorithms and techniques learned in the weekly lectures and tutorials, as well as gained by further researches. This is why, I am confident now in the ability to “use different AI/IA algorithms to address real-world challenges”