

COS30018 - Option B - Task 7: Extension

After you have completed Task 6, our code base has moved to version **v0.6**. If you have executed Tasks 1-6 well then version **v0.6** should be as good as possible given the limitations of historical stock price prediction techniques. However, it is well known that predicting stock prices based only on historical data does not always work very well, especially as an investment tool. There are many ways people have considered to extend these prediction tools such as combining stock price data with other types of data (e.g., the number of times the company is mentioned on Twitter, etc.) or the approach taken in project (**P2**):

<https://github.com/jason887/Using-Deep-Learning-Neural-Networks-and-Candlestick-Chart-Representation-to-Predict-Stock-Market>

In this task we would like you to take the initiative to do some research and come up with an idea to extend our project. You will need to discuss with your project leader to obtain their approval before proceeding to implement your extension.

Your tasks:

1. Research potential approaches for predicting companies' stock prices/trends.
2. Choose one approach and get your project leader's approval before proceeding to implement your idea. Note that your selected approach/idea does not have to be a new one, it can be something that has been described in some online resources or reported in an academic article and you only need to implement it.
3. Upload your Task 7 Report (as a PDF file) to the project Wiki before the deadline and email your project leader to notify that it is ready for viewing and feedback.

Your Task 7 Report will contain the following details:

- Summary of the research you have done to explore potential approaches for predicting companies' stock prices/trends.
- Summary of the implementation you have done to realise your selected idea and the evaluation results you have obtained when assessing the performance of the selected idea/implementation.

Due date: 11:59pm Sunday 27 October 2024

Assessment Criteria:

Simple extension: If the extension you choose has been documented and implemented elsewhere (such as **P2**) then your task is simple as you only need to learn from an existing codebase and make appropriate modification/integration to **v0.6**. You can get up to 15 marks for successfully completing the **Simple Extension** for Task B.7.

COS30018 - Option B - Task 6: Machine Learning 3

After you have completed Task 5, our code base has moved to version **v0.5**. The basic idea behind ensemble methods is to use multiple learning algorithms to obtain better predictive performance than could be obtained from any of the single learning algorithms alone. In this task we aim to develop an ensemble approach for combining multiple machine learning and statistical analysis methods for improving prediction quality. You will take some idea from here:

<https://medium.com/analytics-vidhya/combining-time-series-analysis-with-artificial-intelligence-the-future-of-forecasting-5196f57db913>

Your tasks this week:

1. Develop an ensemble modeling approach consisting of at least two models ARIMA (or SARIMA) and our existing DL model (starting with the LSTM one).
2. Experiment with different ensemble models (e.g., ARIMA/SARIMA/Random Forrest/LSTM/RNN/GRU, etc.) and with different hyperparameter configurations.
3. Upload your Task 6 Report (as a PDF file) to the project Wiki before the deadline and email your project leader to notify that it is ready for viewing and feedback.

Your Task 6 Report will contain the following details:

- Summary of your effort to implement the ensemble models and explain the less straightforward lines of code, focusing especially on those lines that require you to do some research on the Internet (with proper references to the online resources you used).
- Summaries of the results of your experiments with different configurations of ensemble models and model training.

Due date: 11:59pm Sunday 6 October 2024

Assessment Criteria:

You can get up to 15 marks for successfully completing Task B.6.

COS30018 - Option B - Task 5: Machine Learning 2

After you have completed Task 4, our code base has moved to version **v0.4**. Our current program only uses one feature (e.g. the closing stock price of a company) to predict its closing price for a single day in the future. We will now try to solve more advanced prediction problems including multivariate prediction and multistep prediction. The multistep prediction problem applies to time series forecasting that requires a prediction of multiple time steps into the future (e.g. the closing stock prices of the company in multiple days into the future). The multivariate prediction problem on the other hand would take multiple time series as the input for the machine learning model to make the prediction. In its simplest form, the multivariate prediction problem for our project would just take the different features (such as opening price, highest price, lowest price, closing price, adjusted closing price, trading volume) as the input time series for making the prediction of the closing price of the company in the future. The more general multivariate prediction problem can also take other time series as the input to the model (for instance, the time series of the closing prices of the related companies in the same sector, or the time series of the market index).

Your tasks this week:

1. Implement a function that solve the multistep prediction problem to allow the prediction to be made for a sequence of closing prices of k days into the future.
2. Implement a function that solve the **simple** multivariate prediction problem to that takes into account the other features for the same company (including opening price, highest price, lowest price, closing price, adjusted closing price, trading volume) as the input for predicting the closing price of the company for a specified day in the future.
3. Combine the above two functions to solve the multivariate, multistep prediction problem.
4. Upload your Task 5 Report (as a PDF file) to the project Wiki before the deadline and email your project leader to notify that it is ready for viewing and feedback.

Your Task 5 Report will contain the following details:

- Summary of your effort to implement the functions specified above and explain the less straightforward lines of code, focusing especially on those lines that require you to do some research on the Internet (with proper references to the online resources you used).
- Summaries of the results of your experiments with solving the multivariate and multistep prediction problems.

Due date: 11:59pm Sunday 22 September 2024

Assessment Criteria:

You can get up to 15 marks for successfully completing Task 5.

COS30018 - Option B - Task 4: Machine Learning 1

After you have completed Tasks 2 and 3, our code base has moved to version **v0.3**. We will now explore better ways to construct our Deep Learning model. Instead of having to construct the Deep Learning (DL) network manually by explicitly adding the layers of the network, it is possible to write a function that takes as input the number of layers, the size of each layer, the layer name (e.g., LSTM, RNN, GRU, etc.). In this task, you are also required to explore different DL networks and experiment with different hyperparameter configurations.

Your tasks this week: mỗi DL network nên tạo 1 cell để cho cái ML số 3 là
lấy kết quả của cell đó rồi ensemble lại

1. Write a function that takes as input several parameters including the number of layers, the size of each layer, the layer name and return a Deep Learning model. Again, our reference project (**P1**) will give you an example of how this can be done. You can reuse this example, extend it, and most importantly, explain in detail all the code in your program.
2. Use the above function to experiment with different DL networks (e.g., LSTM, RNN, GRU, etc.) and with different hyperparameter configurations (e.g. different numbers of layers and layer sizes, number of epochs, batch sizes, etc.)
3. Upload your Task 4 Report (as a PDF file) to the project Wiki before the deadline and email your project leader to notify that it is ready for viewing and feedback.

Your Task 4 Report will contain the following details:

- Summary of your effort to implement the function to create DL models and explain the less straightforward lines of code, focusing especially on those lines that require you to do some research on the Internet (with proper references to the online resources you used).
- Summaries of the results of your experiments with different configurations of DL models and model training.

Due date: 11:59pm Sunday 8 September 2024

Assessment Criteria:

You can get up to 10 marks for successfully completing Task B.4.

Option B - Task 3: Data processing 2

After successfully completing Task 2, our project moves to a new version **v0.2**. In this version, our project does some simple visualization of the results using a library called `matplotlib`. Data visualization can add much value to a data science and machine learning project such as allowing human users to gain insights into their data and also enabling a number of AI techniques based on graphical input or image processing. We would like to implement a few more visualization techniques in our project. For instance, some people developed a prediction technique based on analysing the candlestick chart representation of the stock price in project (P2) <https://github.com/jason887/Using-Deep-Learning-Neural-Networks-and-Candlestick-Chart-Representation-to-Predict-Stock-Market/graphs/commit-activity>

Your tasks this week:

1. Write a function to display stock market financial data using candlestick chart. You can use the following tutorial: <https://coderzcolumn.com/tutorials/data-science/candlestick-chart-in-python-mplfinance-plotly-bokeh> but, again, you will need to explain in details all the code in your program (including the meanings of the arguments in a function call). Furthermore, you need to include an option in your function to allow each candle stick to express the data of n trading days ($n \geq 1$).
2. Write a function to display stock market financial data using boxplot chart. This is particularly useful when you are trying to display your data for a moving window of n consecutive trading days. Again, feel free to use online resources that teach you how to do this but you need to add comments to explain your codes and the parameters you use.
3. Upload your Task 3 Report (as a PDF file) to the project Wiki before the deadline and email your project leader to notify that it is ready for viewing and feedback.

Your Task 3 Report will contain the following details:

- Summary of your effort to explain the less straightforward lines of code, focusing especially on those lines that require you to do some research on the Internet (with proper references to the online resources you used). Outline the main challenges you faced when accomplishing this task.

Due date: 11:59pm Sunday 1 September 2024

Assessment Criteria:

You can get up to 10 marks for successfully completing Task B.3.

COS30018 - Option B - Task 2: Data processing 1

The current project code (**v0.1**) has many issues and limitations. Some of them are already discussed by the comments in the code. In particular, the data processing is not very good as the user must manually choose the start date and end date for the training data and then start date and end date for the test data. Furthermore, even though we can obtain a dataset with multiple features (e.g., Open, High, Low, Volume, AdjClose), the current version **v0.1** just ignores them all and use the feature Close only. There are several ways you can rectify these issues. In particular, you can learn better data processing methods from project (**P1**)

(**P1**) <https://github.com/x4nth055/pythoncode-tutorials/tree/master/machine-learning/stock-prediction>

Clearly, it is not good enough to just copy-and-paste the code from (**P1**) without understanding what it does.

Your tasks this week:

1. Write a function to load and process a dataset with multiple features with the following requirements:
 - a. This function will allow you to specify the start date and the end date for the whole dataset as inputs.
 - b. This function will allow you to deal with the NaN issue in the data.
 - c. This function will also allow you to use different methods to split the data into train/test data; e.g. you can split it according to some specified ratio of train/test and you can specify to split it by date or randomly.
 - d. This function will have the option to allow you to store the downloaded data on your local machine for future uses and to load the data locally to save time.
 - e. This function will also allow you to have an option to scale your feature columns and store the scalers in a data structure to allow future access to these scalers.
2. Most of the above requirements have already been fulfilled by the code in the project (**P1**). Feel free to learn from it. But you will have to explain what their code does using detailed comments (the same way we commented the code in **v0.1**)
3. Upload your Task 2 Report (as a PDF file) to the project Wiki before the deadline and email your project leader to notify that it is ready for viewing and feedback.

Your Task 2 Report will contain the following details:

- Summary of your effort to explain the less straightforward lines of code, focusing especially on those lines that require you to do some research on the Internet. Note that, if you choose not to explain a particular line of code or explain it with too few details, a teaching staff can ask you to explain it and you fail to explain properly, you'll receive a low mark for this task.

Due date: 11:59pm Friday 23 August 2024

Assessment Criteria:

You can get up to 10 marks for successfully completing Task B.2.

COS30018 - Option B - Task 1: Setup

You should be able to download the current project code base (**v0.1**) from Canvas. It is a single Python file: `stock-prediction.py`. The program is created based on a YouTube tutorial and has many issues. The project leader is aware of the issues and would like to improve this code. The project leader also identifies two interesting projects (**P1** and **P2**) on Github to allow you to learn from so that you can create some ideas to improve your project.

(**P1**) <https://github.com/x4nth055/pythoncode-tutorials/tree/master/machine-learning/stock-prediction>

Your tasks this week:

1. Watch the tutorial YouTube video to learn about **v0.1**:
<https://www.youtube.com/watch?v=PuZY9q-aKLw>
2. Download the code bases of **v0.1** and **P1** to your local machine to test them out.
3. Follow the instructions from the tutorial video and from (**P1**) to setup your environment on your local machine. We strongly recommend you set this up in a virtual environment. Note that both **v0.1** and **P1** can share the same virtual environment as they use the same libraries. Please supply a requirements file for this setup. More details about requirements files can be found here: https://pip.pypa.io/en/stable/user_guide/#requirements-files
4. Test both **v0.1** and **P1** and make sure that you can run them, train the models and get the results.
5. Setup the Github repository for your project and commit **v0.1** to your repo and setup the Wiki page of the project to contain all documentations for the project (including the Weekly reports to be submitted)
6. Upload your Task 1 Report (as a PDF file) to the project Wiki before the deadline and email your project leader to notify that it is ready for viewing and feedback.

Your Task 1 Report will contain the following details:

- Summaries of your attempt to setup your environment, including details of your requirements file.
- Summaries of your attempts to test the provided code bases (**v0.1** and **P1**) with screenshots.
- Summary of your understanding of the initial code base **v0.1**.

Due date: 11:59pm Friday 16 August 2024

Assessment Criteria:

You can get up to 10 marks for successfully completing Task B.1.

Bonus task: You can also try to explore project (**P2**):

(**P2**) <https://github.com/jason887/Using-Deep-Learning-Neural-Networks-and-Candlestick-Chart-Representation-to-Predict-Stock-Market>

Test it in a separate virtual environment from the one you use for **v0.1** and **P1** to avoid messing up the libraries and dependencies. You can also report your attempt to do this. **You can get up to 3 bonus marks for this effort.**