1. **Problem statement:**
    a. Using a datasets of stock data, we will predict individual stock price for the next day and next week. We can then use that data to classify the stock into different categories
2. **Data preprocessing**
    a. We are working with this dataset.
        i. https://www.kaggle.com/datasets/andrewmvd/sp-500-stocks/code
    b. We will also use the yahoo finance API to get individual stock information
        i. https://finance.yahoo.com/quote/%5EGSPC/history?p=%5EGSPC
    c. We didn't have to preprocess the data because all the data is needed and the data is in units that don't need conversion
    d. We will use this API (more like library) to help train, evaluate and test our model
        i. https://github.com/mjbhobe/dl-pytorch/blob/master/README.md
        ii.
3. **Machine Learning model:**
    a. As a group, we initially proposed utilising a Long Short-Term Memory (LSTM) Recurrent Neural Network (RNN) implemented using scikit-learn. However, after collective experimentation, we found that scikit-learn's implementation of RNNs is limited compared to other frameworks like TensorFlow or PyTorch. Therefore, we collectively decided to switch to PyTorch for its more comprehensive support for RNNs, along with dynamic computational graphing to help visualise our model. Additionally, we will use GRU's as it provides more simplicity, and with that, shorter processing time.
    b. We chose PyTorch as the framework for implementing the model due to its extensive capabilities for building and training neural networks. The architecture of the model includes GRU's layers followed by fully connected layers for prediction.
    c. For the training/validation/test splits, we followed a standard approach of splitting the data into 70% training, 15% validation, and 15% test sets. We used a variety of stocks from different industries so the model wouldn't be able to learn from trends solely from the tech industry (etc unintentionally learns about the tech boom). Overall testing was good
    d. To evaluate the model's performance and detect overfitting or underfitting, we monitored the training and validation loss during training. Additionally, we collectively analysed the model's performance on the test set to ensure generalisation to unseen data.
4. **Preliminary results**
    a. We had missed in the deliverable 1 where we would use a primary evaluation metric, but upon researching we should use mean squared error to see the difference between the actual result, and the model's prediction. Upon closer inspection to our model, we observed signs of overfitting in our model. The training loss was decreasing steadily, but

the validation loss began to plateau or even increase after a certain number of epochs. This indicated that while the model was learning well from the training data, it was struggling to generalise to unseen data, leading to overfitting. To address this issue, we need to reevaluate our model architecture and training approach.. We may consider reducing the complexity of the model by decreasing the number of LSTM layers or hidden units.

5. **. Next Steps**
    a. Our immediate priority is to address the overfitting issue by collaboratively fine-tuning these hyperparameters and regularisation techniques to strike a balance between fitting the training data well and generalising to unseen data.Throughout this process, we will maintain a collaborative and iterative approach, regularly evaluating the model's performance on both the training and validation datasets.
    b. Make sure the model doesn’t overfit, and makes sure the model doesn’t learn from significant market crashes and booms.