Harrison Stein

Exam 3 Report

**Background/Overview:**

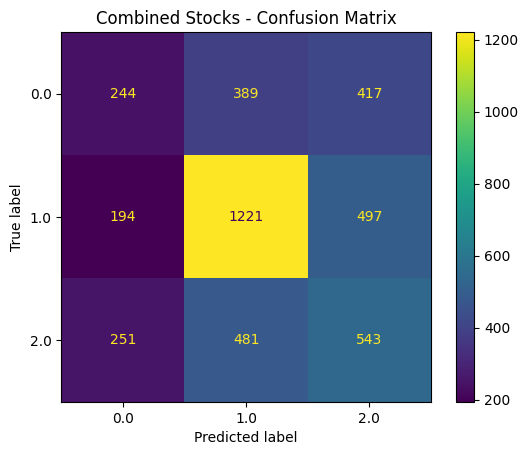
The premise of my exam is to experiment with stocks while using recurrent neural networks, stock price prediction, and hyperparameters along with the XGBoost model. Moreover, I used X and y train and test parameters in a time series split along with CNN, MLP, LSTM, and GRU neural networks. I played around with the activation functions, so I varied my usage of sigmoid, softmax, and tanh in terms of where they fit best with their respective neural networks. I also used categorical crossentropy for loss in all of the neural networks since we are dealing with combined categorical data with features (characteristics of the stocks) in the columns along with the target value (the three target classes). I used accuracy, RMSE, precision, recall, and f1-scores as values to calculate for determining how good the model is and the overall rate of precision in terms of prediction of the three classes (you can see this in the confusion matrix which I’ll explain later).

The performances of the neural networks are plotted with their trained and validation parameters to determine how well they perform over each epoch within 50 iterations. I have decided to combine each of the stocks I used (Apple, Dollar General, Blackberry, Amazon, and S&P 500 ETF) instead of doing each stock separately since this is more efficient, much less computationally expensive, and will provide a better and generalized view of how these stocks are able to be trained, performed, and modeled after especially when they vary so much while being in different industry sectors. I end the exam with a logarithmic graph that fully shows how the recurrent neural networks perform in terms of accuracy and RMSE, so it is visualized and clear with a realistic scale to go by instead of standard integers.

**The XGBoost Model/Confusion Matrix:**

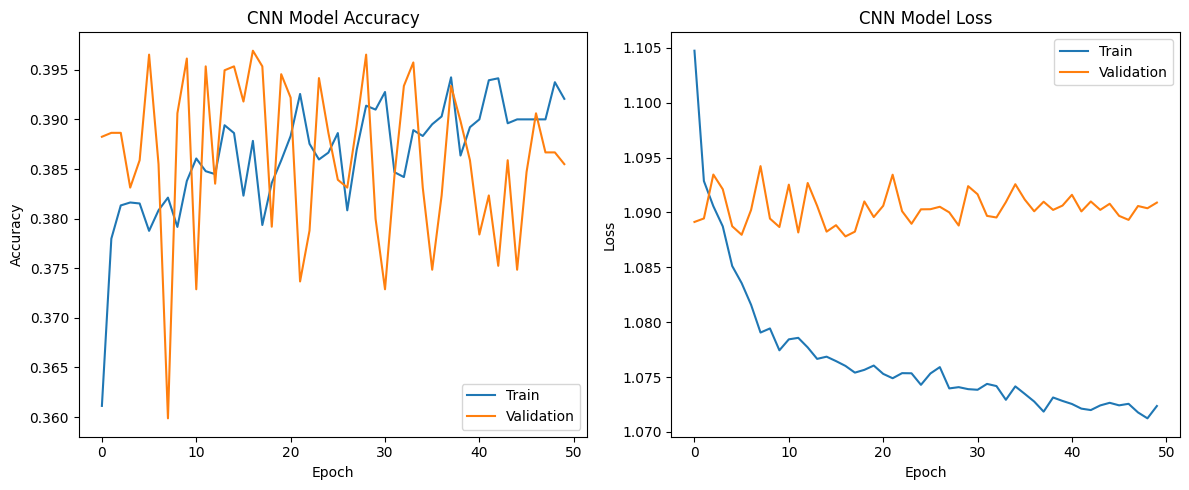
In the beginning of the assignment, I set up the stocks and features in an iterative loop, so it captures all of the stocks combined over the outlined dates with close, volume, and the additional features including the lag features that I included which give a better outlook of the stocks, descriptive indicators, and help to improve the scores of the models. Also, I made sure to drop any values that had NA within them to make sure we only deal with non-zero and values that exist. I added conditions and corresponding values to the set-up which shifts the lag features up by 1 and 2 periods while the return is shifted up 1 period when in relation to negative and positive 0.005. I also created three classes that are 0 indexed for a better depiction of the overall data. After, I apply the time split with the train and test values, so we go by periods in time for the stocks and then, I build the pipeline with XGBoost. I specifically chose XGBoost as it is a powerful model for classification and works great with stocks since it combines a series of weak decision trees to make informed predictions. I made sure to tune the values to the most effective within the pipeline including with the scale by StandardScaler and playing around with the parameters for XGBoost including mlogloss and multi:softprob which I found improved model performance steadily. The model ends up performing well for the size of the dataset and the combination of stocks, which is rather impossible to get to 50% or above as the stocks vary so much in their price skews over the dates.

The confusion matrix shows the three classes with the performance of the correct classification for each of the three classes from 0 to 2 (0 indexed) with the columns against the target data. It shows the correct predictions (highest values in the center and towards it) and misclassifications (lowest values outside of the center). I ended up making the confusion matrix here as it shows which features or classes are reliable and which are not in the model performance or inhibit good performance to occur. In terms of the performance, class 0 has the worst accuracy and correct predictions (high confusion), class 1 has the best accuracy and correct predictions (low confusion), and class 2 performs worse than class 1, but better than class 0 (high confusion).



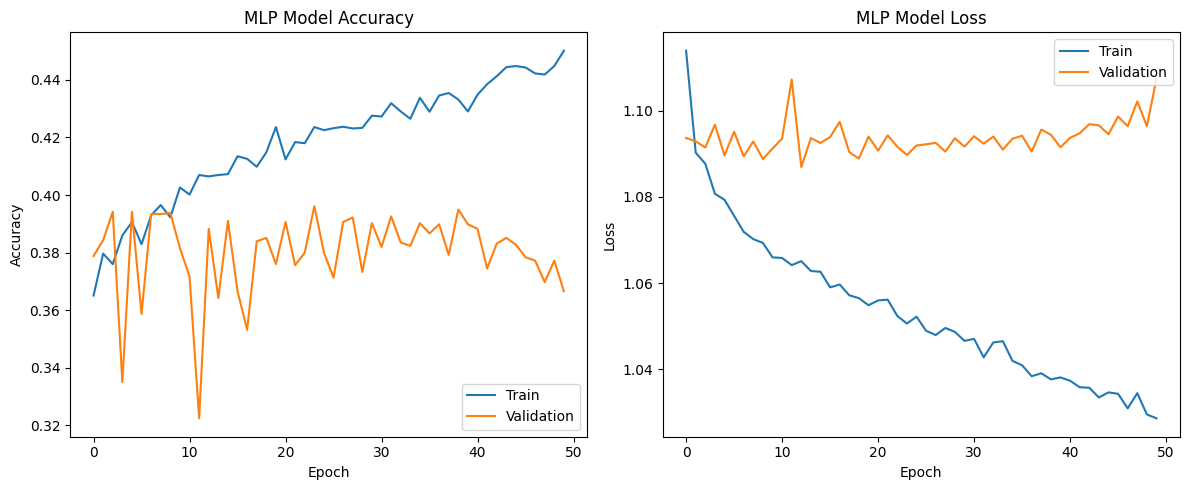
**The CNN Model:**

Before I begin with the CNN and for the rest of the neural networks, I form a method for creating the sequences for the train and test values for X and y with the timesteps to note the shift in periods within time for the stocks (this is how stocks are measured and recorded). I end up using 25 timesteps as it works best with the dataset of stocks here and then, I create the sequences for the training and testing data like I did before with the time split. I also one hot encode y since we are dealing primarily with categorical tabular values and categorical crossentropy for the loss, which is a necessity for this, so the class labels conform to the results, shape, and size of the dimensions of the overall matrix. I create the CNN with 1D layers including conversion, GeLU activation (better than ReLU and works great with stocks for better performance and handling), dropout, and both tanh and softmax activation in the dense layers. I plot both the accuracy and loss with the train and validation parameters amongst the 50 epochs. The model ends up performing well over the series of epochs but is rather messy with large variations in between the epochs. The accuracy is also better than the XGBoost model with 48% accuracy which shows the CNN models work well.



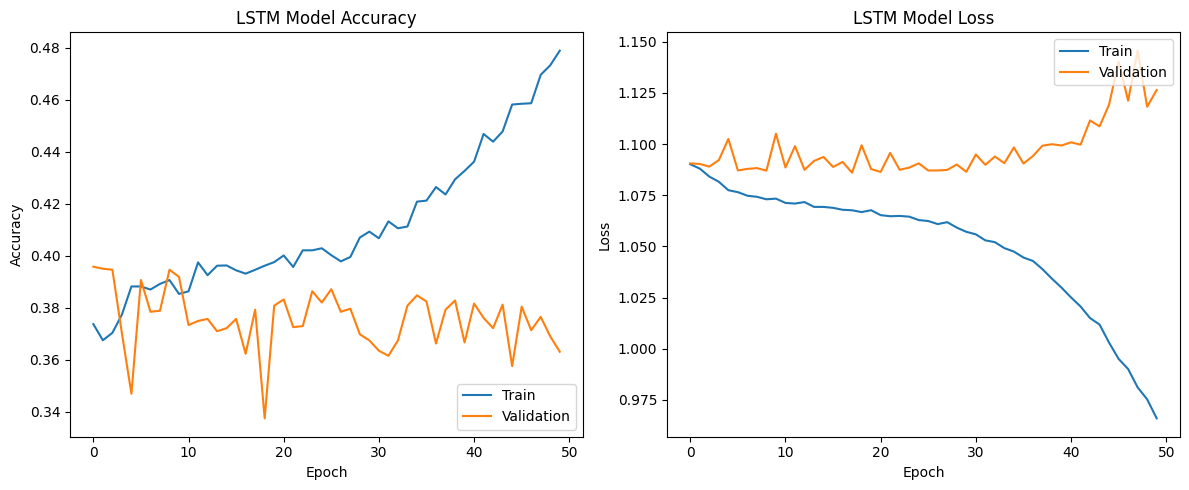
**The MLP Model:**

I ended up including a multilayer perceptron neural network since I have used this extensively in my Machine Learning class and it expands the scope of the exam to further examine other models with different performances especially for classification. I use both GeLU and sigmoid for the activation functions along with flattening the values for making sure the networks runs in 1D. The network ends up performing slightly worse in overall scores than the XGBoost model and the CNN with an accuracy of 47% and the validation parameter seems to run worse here as well compared to the train.



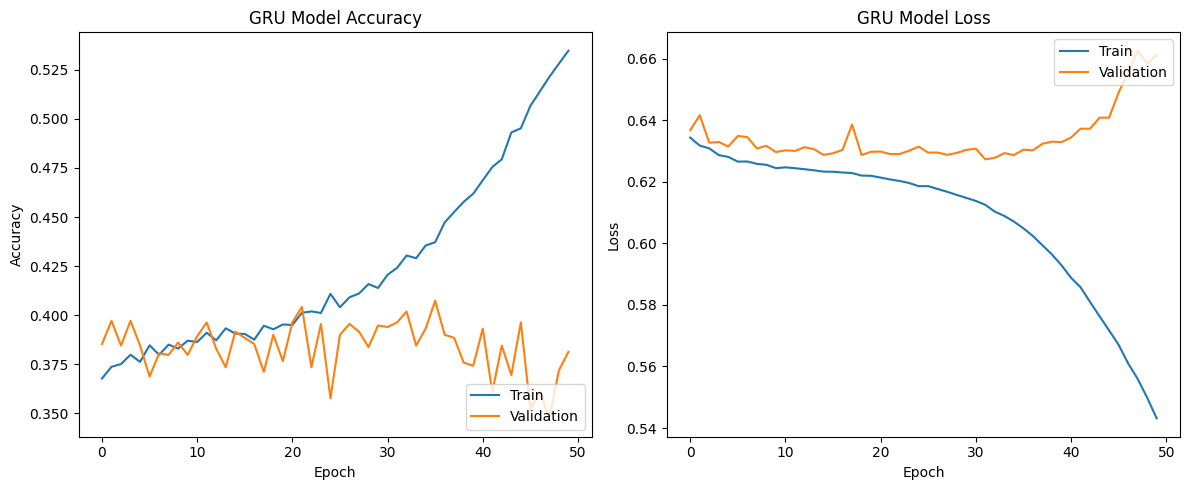
**The LSTM Model:**

The LSTM model was included in the .txt and is a useful network to use with stocks, so I adapted it to my dataset of combined stocks with GeLU and softmax activation. I also had to one hot encode the y\_test labels to class indices as the LSTM model requires this to run and process the values in shape and size of the overall input of data. Validation has high loss and low accuracy compared to the other models while train seems to majorly outperform the validation in both accuracy and loss. This is not a stable network, so it doesn’t look efficient or practical to use along with the MLP model which gave very similar results with the combination of stocks.



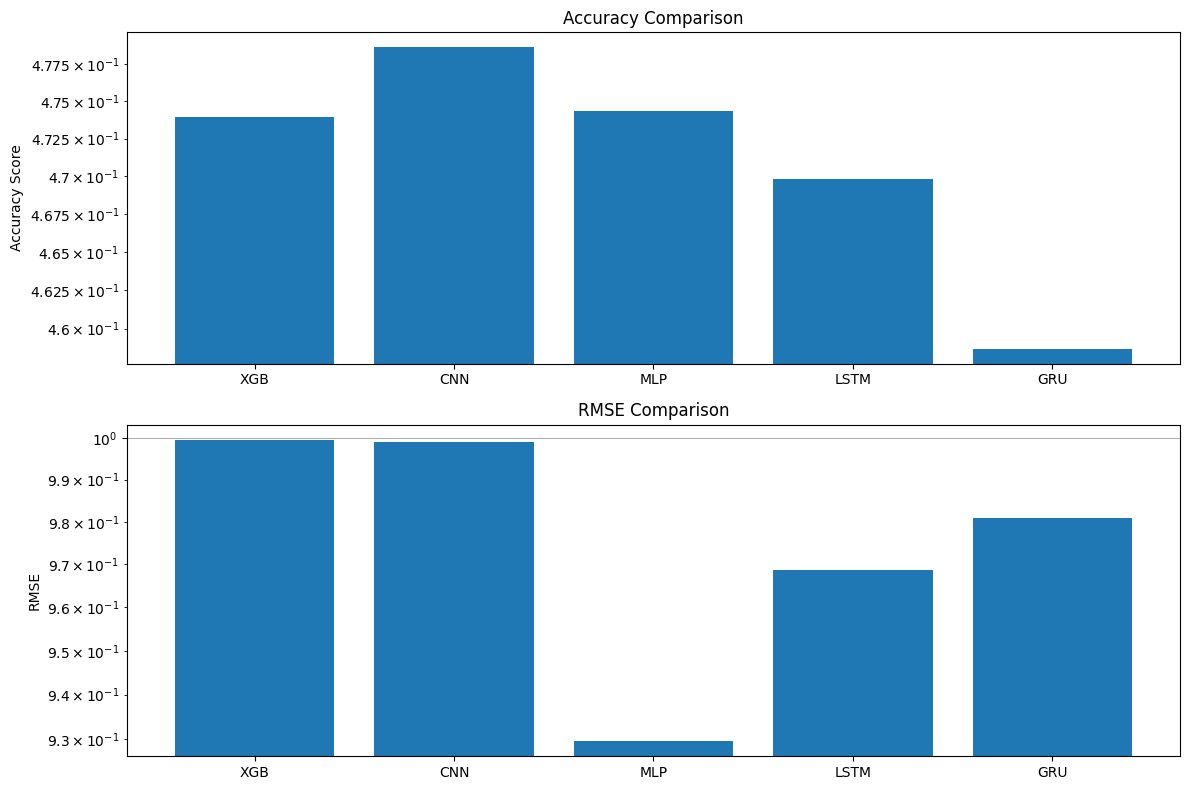
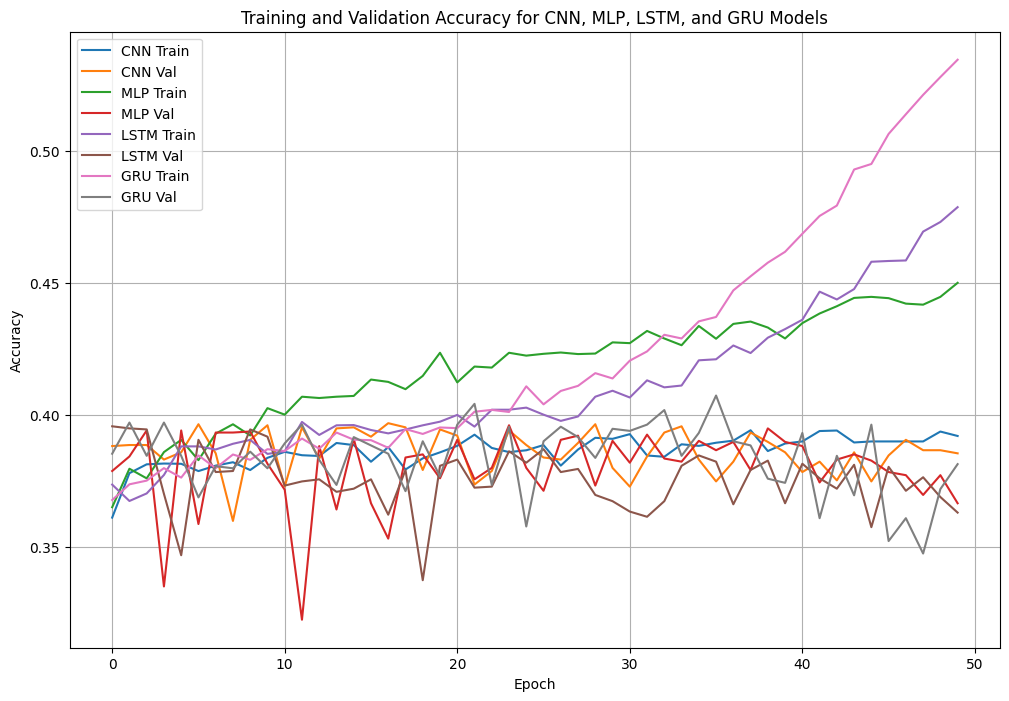
**The GRU Model:**

The GRU model was included in the .txt and is also a useful network to use with stocks, so I adapted it once again with GeLU and softmax along with two GRU layers within the network as required with the parameters. Morever, we must do the one hot encoding again as this model along with the LSTM only works with class indices for the y\_test value instead of class labels. This model ends up performing worse than the LSTM model and is even less stable as it has an accuracy of 46% which is the lowest yet and the validation data performs even worse than before. This is by far the worst model to use and is impractical for modeling and predicting stocks.



**Conclusion/Outlook:**

When plotting all the neural networks together with their training and validation accuracy, GRU train performs the best as a parameter, but the best model overall is the CNN as it has the highest accuracy which provides the best results for the stock predictions. This is further proven with the logarithmic graphs for the accuracy and RMSE comparison where CNN outperforms the other models in accuracy and has a high RMSE compared to the other models which is unfortunate, but the accuracy triumphs the other scores for analyzing stocks.

****