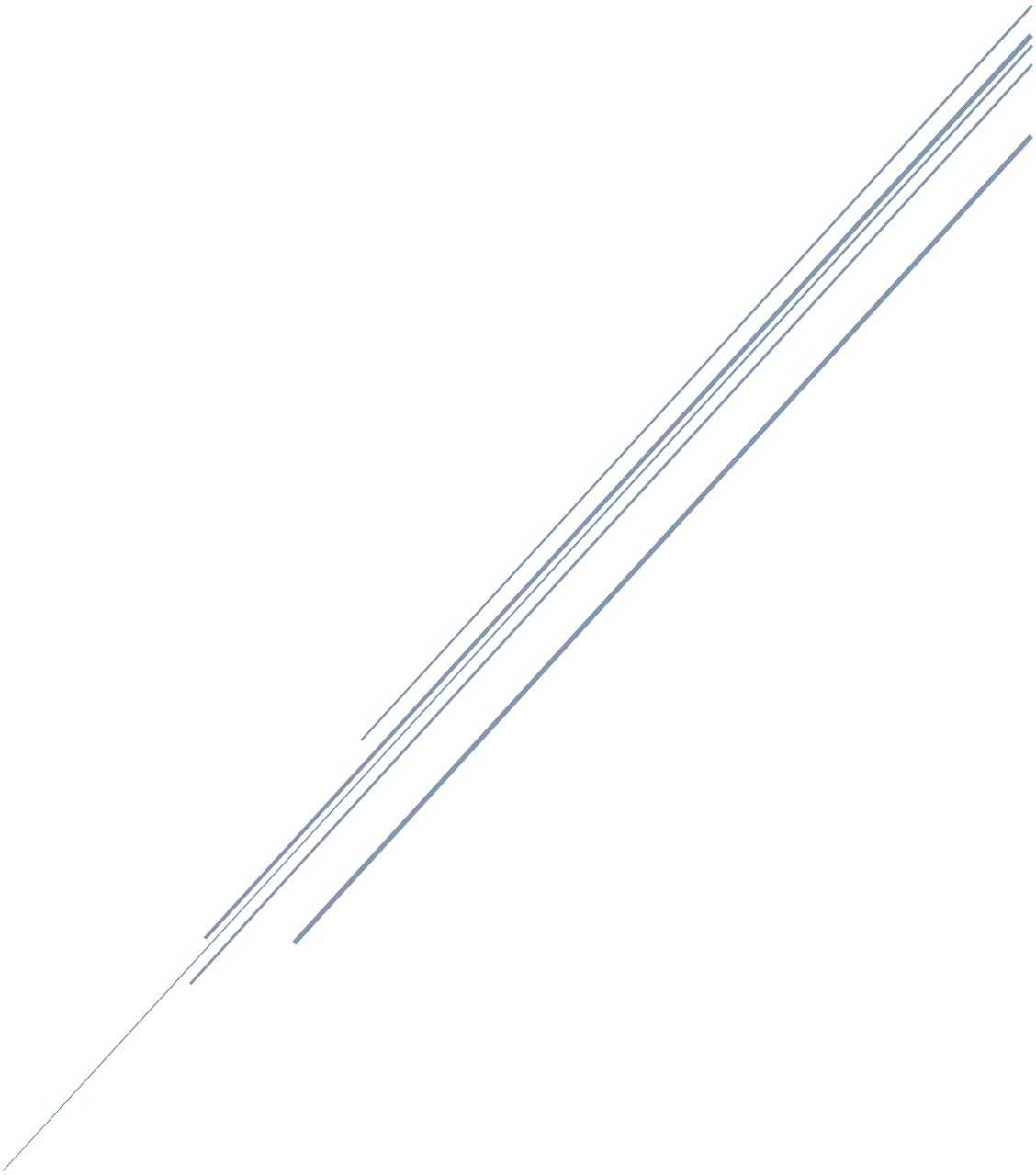
PROJECT SUMMARY REPORT

Naca Hitchman – 103072170 | Tutor – Mukesh Malani



COS30018 – Intelligent Systems

Contents

Introduction

For this unit I have worked on Project B – Stock Price Prediction System. This python project uses machine learning models to predict prices on a given stock over a given timeframe.

What's Included

Within this portfolio there is the following:

- Project Summary Report
- Weekly task reports
- Final code
- requirements.txt file

Outline of Capabilities

I have implemented the following capabilities after completing the weekly tasks:

1. Loading different datasets with adjustable timeframes, with data saving
2. Displaying box plot charts and candlestick charts
3. LTSM Model hyper configuration adjustability with alternative layer types
4. Multivariate and multistep predictions
5. Ensemble predictions with ARIMA model

How to run

The following steps should allow you to run the program:

1. Install python 3 or newer.
2. Open a command terminal or PowerShell window in base folder.
3. Install virtual environment package using: pip install virtualenv
4. Create a python virtual environment using: python -m venv [venv name]
5. Activate the virtual environment using: .\[venv name]\Scripts\activate
6. Install packages using: pip install -r requirements.txt
7. Make any desired configuration adjustments within the python script.
8. Run the python program using: python .\stock_prediction.py

Overall System Architecture

The Python program for predicting stock prices uses several libraries for data processing, visualization, and machine learning. The program fetches stock prices from Yahoo finance using the yfinance library. The pandas and numpy libraries are used for data manipulation and mathematical operations. For data visualization, the matplotlib and mplfinance libraries are used.

The machine learning part of the system uses the tensorflow and sklearn libraries. The tensorflow library is used to create and train the LSTM (Long Short-Term Memory) model. The sklearn library, on the other hand, is used to preprocess the data and split it into training and testing sets.

Implemented Data Processing Techniques

The program uses various data processing techniques to prepare the data for machine learning.

Data Fetching: The program fetches stock prices using the yfinance library. The fetched data is saved as a csv file for future use.

Data Cleaning: The program handles missing values in the data using various methods such as dropping the rows with missing values or filling them with forward fill, backward fill, or mean of the column.

Feature Scaling: The program uses the MinMaxScaler from the sklearn library to scale the features between a specified range.

Data Splitting: The program splits the data into training and testing sets either randomly or based on a specified date.

Data Sequencing: The program prepares sequences of a specified length from the data. These sequences serve as the input to the LSTM model.

Data Reshaping: The data is reshaped to match the input shape required by the LSTM model.

Experimented Machine Learning Techniques

The program uses various machine learning techniques for stock price prediction.

LSTM Model: The program uses the LSTM model for prediction. The LSTM model is a type of recurrent neural network that is capable of learning long-term dependencies in data. The model is implemented using the tensorflow library. The model's architecture includes LSTM layers, dropout layers to prevent overfitting, and a dense layer for output.

ARIMA Model: The program also uses the ARIMA (AutoRegressive Integrated Moving Average) model for prediction. The ARIMA model is a type of time series model that uses past values and past prediction errors to predict future values.

Random Forest Model: The program further experiments with the Random Forest model for prediction. The Random Forest model is an ensemble learning method that operates by constructing multiple decision trees and outputting the mean prediction of the individual trees.

Ensemble Prediction: The program calculates ensemble predictions by averaging the predictions from the LSTM and ARIMA models, and the LSTM and Random Forest models.

Scenarios / Examples

In this section, I will detail some screenshots to help show basic usage and adjustment areas.

Parameter adjustments

The primary two areas for adjustments are for the data processing, and the LTSM model hyper configuration parameters.

```
# define function parameters to use
DATA_SOURCE = "yahoo"
COMPANY = "TSLA"
DATA_START_DATE = '2015-01-01'
DATA_END_DATE = '2022-12-31'
SAVE_FILE = True
PREDICTION_DAYS = 100
SPLIT_METHOD = 'random'
SPLIT_RATIO = 0.8
SPLIT_DATE = '2020-01-02'
NAN_METHOD = 'drop'
FEATURE_COLUMNS = ['Open', 'High', 'Low', 'Close', 'Adj Close', 'Volume']
SCALE_FEATURES = True
SCALE_MIN = 0
SCALE_MAX = 1
SAVE_SCALERS = True
prediction_column = "Close"
N_STEPS = 5;
```

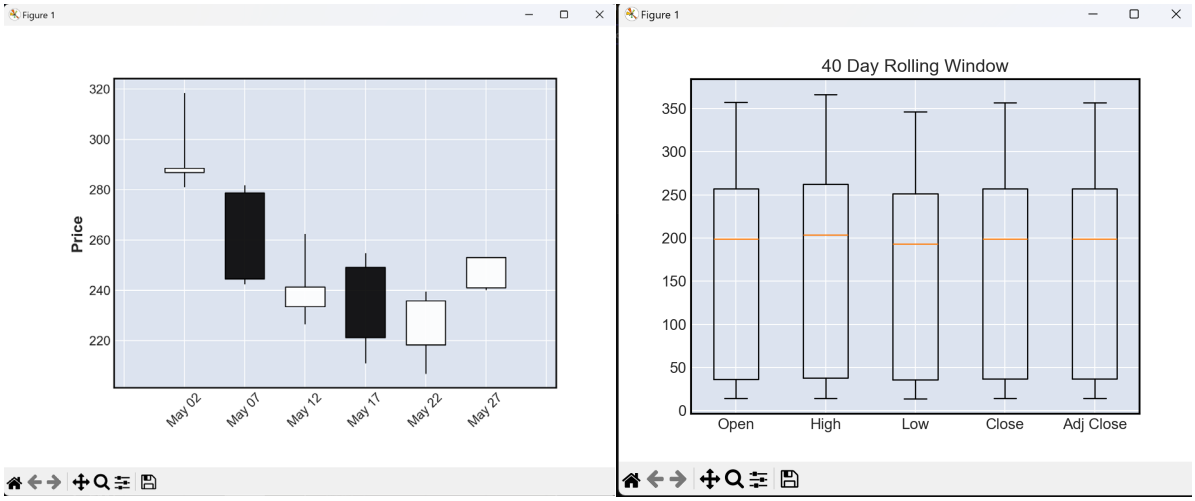
```
#set 1
units = [32, 16]
cells = ['LSTM','LSTM']
n_layers = 2
dropout = 0.3
loss = "mean_absolute_error"
optimizer = "rmsprop"
bidirectional = True

# Set the number of epochs and batch size
epochs = 10
batch_size = 32
```

The settings for the candlestick and boxplot charts can also be adjusted on these lines:

```
#task 4 candlestick
plot_candlestick(processNANs(downloadData(COMPANY, '2022-05-01', '2022-05-31', False), 'drop'), 5)
#task 4 bloxplot
plot_boxplot(downloadData(COMPANY, '2019-01-01', '2022-12-31', False), 40, ['Open', 'High', 'Low', 'Close', 'Adj Close'])
#-------------------------------------------------------------------------------
```

Charts

When the program is run, what will first appear are the candle stick and boxplot charts for the data. These must be closed before progressing to the next segments of the prediction process.



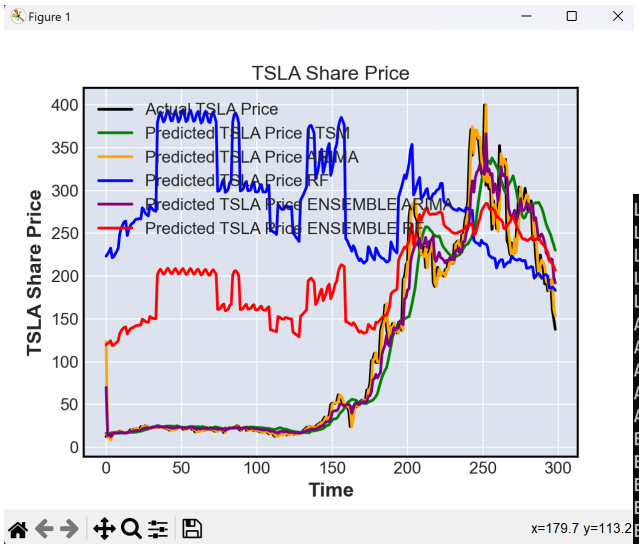
Predictions

After the first charts, there will be some time where the various models undergo their training and prediction processes. This can take a while depending on the provided data and configuration settings.

```
Epoch 1/10
48/48 [==============================] - 32s 385ms/step - loss: 0.0857
Epoch 2/10
48/48 [==============================] - 16s 341ms/step - loss: 0.0581
Epoch 3/10
48/48 [==============================] - 17s 364ms/step - loss: 0.0536
Epoch 4/10
39/48 [=======================>......] - ETA: 3s - loss: 0.0502
```

```
0.000000/299.000000, predicted=0.284363, expected=0.007023
1.000000/299.000000, predicted=0.004130, expected=0.009794
2.000000/299.000000, predicted=0.012890, expected=0.009834
3.000000/299.000000, predicted=-0.003187, expected=0.015766
4.000000/299.000000, predicted=0.006307, expected=0.018449
5.000000/299.000000, predicted=0.020658, expected=0.017926
6.000000/299.000000, predicted=0.018094, expected=0.018998
7.000000/299.000000, predicted=0.019267, expected=0.019716
8.000000/299.000000, predicted=0.019737, expected=0.020901
9.000000/299.000000, predicted=0.020920, expected=0.022799
```

The line chart with the predicted prices against the actual prices should then appear, and after closing this chart, the next n day predictions for the models will be printed to the console.




```
LTSM Prediction for day 1: 282.8350830078125
LTSM Prediction for day 2: 274.9612121582031
LTSM Prediction for day 3: 263.65716552734375
LTSM Prediction for day 4: 249.55677795410156
LTSM Prediction for day 5: 230.25540161132812
ARIMA Prediction for day 1: 123.43450947465706
ARIMA Prediction for day 2: 124.15168619563438
ARIMA Prediction for day 3: 124.25037227507583
ARIMA Prediction for day 4: 124.16050611888188
ARIMA Prediction for day 5: 124.18167944751954
Ensemble Prediction for day 1: 203.134796241234777
Ensemble Prediction for day 2: 199.55644917691876
Ensemble Prediction for day 3: 193.9537689012098
Ensemble Prediction for day 4: 186.85864203649172
Ensemble Prediction for day 5: 177.21854052942382
```

Critical Analysis

Following my progress in implementing this system, I have been able to analyse my work to determine some strengths and weaknesses:

Strengths:

1. Versatility of Models: The implementation leverages a variety of predictive models, including LSTM, ARIMA, and Random Forest. This approach allows the system to capture different types of patterns and dependencies in the data.
2. Data Preprocessing: The system employs comprehensive data preprocessing steps, including handling missing values and feature scaling. These steps can help improve the performance of the machine learning models.
3. Ensemble Predictions: The system incorporates multiple ensemble predictions, which average the predictions from multiple models. Ensemble methods can often yield better results by leveraging the strengths of each individual model (although they don't in my case).

Areas of Improvement:

1. Random Forrest Tuning: Although the system does include the Random Forest model that is trained upon the data, and displayed on the charts to the user, it is wildly inaccurate and certainly need to be fixed and adjusted to bring the results more in line to the LTSM and ARIMA models.
2. Extended Model Evaluation: While the system provides visual plots to compare the predictions of different models with the actual prices, the ARIMA and Random Forrest models lack quantitative metrics to evaluate the performance of the models. Metrics such as Mean Absolute Error (MAE), Root Mean Square Error (RMSE), or R-squared can provide a more objective measure of the models' performance.
3. Hyperparameter Tuning: The system does not include a process for hyperparameter tuning. Techniques such as grid search or random search could be used to find the optimal hyperparameters for the machine learning models, potentially improving their performance.
4. Outlier Handling: The system does not handle outliers in the data. Outliers can have a significant impact on the performance of machine learning models. Techniques such as the Z-score method or the IQR method could be used to detect and handle outliers.

Conclusion

To conclude, this unit was quite tough, forcing me to learn many new concepts and techniques to be able to complete the project. I was able to get a good introduction to neural network model creation in a relatively straight forward scenario of predicting stock prices. Overall, while there are many areas for potential improvements, I believe I have been quite successful in the unit with both my understanding of the content, and being able to implement a working system that follows the specifications.