Stock Prediction – Option C Task 1 Report

Student Name: Tommy
Student ID: 9946
Course: COS30018 – Technology Project
Date: 28 August 2025

---

1. Environment Setup

To run the provided code base (v0.1), I created a **Python virtual environment** to keep dependencies isolated.
I installed the following libraries required by the script:

requirements.txt

- numpy
- matplotlib
- pandas
- tensorflow
- scikit-learn
- pandas-datareader
- yfinance

I activated the virtual environment and installed everything with:

- pip install -r requirements.txt

The environment was tested successfully — Python and TensorFlow were both running.

---

2. Testing v0.1 (stock_prediction.py)

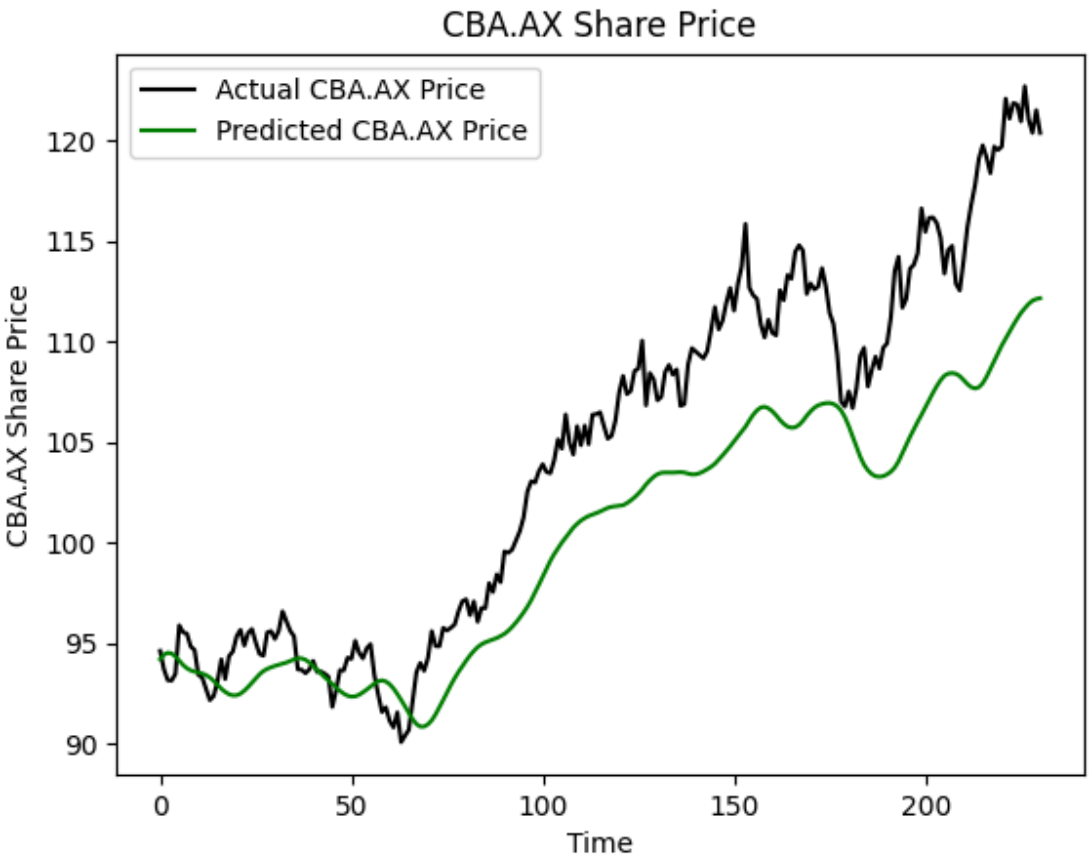
I executed the file using:

- python Code/stock_prediction.py

The script:

- Downloaded **CBA.AX** stock data using yfinance
- Scaled the **Close price** with MinMaxScaler
- Trained a **stacked LSTM model** (3 layers, dropout, dense output)
- Ran training for 25 epochs
- Compared **predicted stock prices vs actual test data**
- Plotted the results

Screenshot of Results



---

3. Understanding of Code Base v0.1

The v0.1 script follows a **basic machine learning workflow** for stock price prediction:

1. **Data Loading**
    - Uses yfinance to download stock price data (CBA.AX).
    - Training data: Jan 2020 – Aug 2023
    - Testing data: Aug 2023 – Jul 2024
2. **Preprocessing**
    - Uses MinMaxScaler to normalize close prices between 0–1.
    - Creates training sequences of length 60 days.
3. **Model**
    - Stacked **LSTM network** with dropout layers to prevent overfitting.
    - Optimizer: adam, Loss: mean_squared_error.
4. **Training**
    - Trains the model on historical data (epochs=25, batch_size=32).
5. **Testing & Evaluation**
    - Predicts stock prices on test data.
    - Inverse transforms predictions to original price range.
    - Plots **Actual vs Predicted prices**.
6. **Prediction**
    - Outputs a **next-day prediction**, though results are inaccurate (off by ~10–13%).

---

4. Observations & Limitations

- Predictions **lag behind actual prices**, showing poor short-term accuracy.
- The model only uses **Close prices**, no other features like Volume, High/Low, or market indicators.
- The scaler is fit only on training data, which may distort test scaling (Issue #2 mentioned in code).
- No saved model or reproducible evaluation metric (e.g., RMSE).

Conclusion: v0.1 runs successfully and produces a working stock price prediction with an LSTM model. However, predictions are not very accurate. The code provides a foundation but requires improvements (better data handling, evaluation metrics, more features) in future iterations.