

Name: Khandil Karimi
Course: CS Final Project
Due Date: May 1, 2025

For my final project, I chose to work on stock price prediction using machine learning. Out of the three project options (RAG, KRAG, and Stock Prediction), I selected the Stock Prediction project because it felt the most understandable and similar to the bike data project we did earlier in the semester. I also felt like it would give me a chance to practice using real-world data and tools that we've been learning in class. The main goal of this project was to create a program that could predict future stock prices using past data with a type of model called LSTM, which stands for Long Short-Term Memory. LSTM is useful when working with time-based data like stock prices because it can learn from patterns over time.

This project took me about two weeks to complete. I started by downloading historical Apple stock price data (AAPL) using the yfinance library. I used data from 2010 through late 2023 to give the model a long history of patterns to learn from. After collecting the data, I focused on the "Close" price and scaled the data using MinMaxScaler from scikit-learn. This step is important because it helps the model train more effectively by putting the values in a smaller, consistent range.

Once the data was cleaned and scaled, I created sequences of 60 days. This means I trained the model to look at the last 60 days of prices and try to predict what the next day would be. I split the dataset into 80% training and 20% testing so I could evaluate how well the model was performing. I reshaped the data into the format that LSTM models expect and then built the model using TensorFlow and Keras. The model had two LSTM layers followed by two Dense layers. I trained it over five epochs using a batch size of 32. After training, I used the model to make predictions on the test data and compared the results to the real prices. To measure how well the model did, I used RMSE, which stands for Root Mean Square Error. I ended up with an RMSE of about 5.74. That means, on average, the predictions were around \$5.74 off from the actual price. For a basic model like this, especially on stock market data, I think that's a pretty solid result. The entire project was done in JupyterLab, which helped me stay organized and test everything step-by-step. Once I had everything working, I created a GitHub repository and uploaded all my files, including my Jupyter notebook, this written report, and my PowerPoint slides. I made the repository public so it's easy for the professor to access and grade. That part didn't take long, but it's helpful to know how to submit code and reports professionally.

Overall, I learned a lot during this project. I became more confident working with Python and JupyterLab, and I finally got comfortable using tools like TensorFlow and Keras. I also learned how to clean data, create sequences, and train a deep learning model. On top of that, I practiced using GitHub for submitting my work and writing a full report and presentation, which made this feel more like a real-world project. Even though I had some bugs and had to look things up along the way, I'm really glad I chose this option. Spending two weeks on it gave me time to fully understand what was going on and get everything working properly. I feel like this project helped me connect everything we learned in class into one complete experience, and I'm looking forward to presenting it.