Machine Learning Models to Estimate Stock Price Prediction Report

Nathan Diamond, Logan Armendariz, Eric Quezada

May 2025

Link: `https://github.com/adhfmz8/stock-price-prediction`

Abstract

This report explores the application of several machine learning models to estimate stock price predictions. Specifically, we implement and evaluate the following models: *an XGBRegressor with Randomized Search CV and Time Series Split, a Long Short-Term Memory (LSTM) model, and a Ridge Regression model.* These models were trained to forecast stock prices over a 7-day period using historical financial data collected over a five-year period from Yahoo Finance. The stocks analyzed include the following tickers: APPL, MSFT, NVDA, JNJ, PFE, JPM, GS, XOM, CVX, TSLA, AMZN, NEE, CAT, and BA. The dataset for each stock consists of daily price data from January 1, 2020, to January 1, 2025, as well as associated financial statements including balance sheets, cash flow statements, and income statements.

1 Introduction

Predicting stock prices is a challenging task due to the complex and volatile nature of financial markets. However, advancements in machine learning have made it possible to model historical patterns and trends to generate short-term forecasts. In this project, we investigate the performance of three machine learning models—XGBRegressor, Long Short-Term Memory (LSTM), and Ridge Regression—for predicting stock prices over a 7-day period. These models are applied to a selection of major companies across various industries, using historical price data and financial statements collected from January 1, 2020, to January 1, 2025. Our objective is to evaluate how effectively each model captures market behavior and produces accurate short-term price predictions.

The project is organized into four main directories:

- **data** – Contains a collection of CSV files representing the dataset used throughout the project.
- **models** – Optionally used to store and reuse trained machine learning models.
- **notebooks** – Contains four Jupyter notebooks, each responsible for a distinct phase of the data analysis and modeling pipeline.
- **scripts** – Includes `seed_data.py`, a script used to extract historical stock and financial data using the Yahoo Finance API.

Below is a brief description of each notebook:

1. ***01_data_exploration.ipynb*** – Performs initial exploration of the stock price dataset, including visualization of trends and statistical summaries.
2. ***02_model_testing.ipynb*** – Implements and evaluates an XGBRegressor model using `RandomizedSearchCV` with `TimeSeriesSplit`. It covers hyperparameter tuning, model training, and visualization of prediction results.

3. ***02_model_test_lstm.ipynb*** – Develops an LSTM model for time series forecasting. It prepares sequential data, constructs the model using Keras, and visualizes the predicted vs. actual results.
4. ***02_ridge_model.ipynb***– Applies Ridge Regression as a baseline linear model. The notebook includes data normalization, training, and prediction visualization.

2 Methods

The implementation of this project is organized into a modular, object-oriented framework, with each component encapsulating a specific part of the stock price prediction pipeline. The system is divided into four key stages: data collection, exploratory data analysis, model training, and evaluation.

- **Data Collection:** Historical stock data, including daily closing prices and financial statements, were collected from Yahoo Finance using the `yfinance` API. The script `seed_data.py` automates this process and saves the datasets as CSV files for reuse.
- **Exploratory Data Analysis:** The notebook `01_data_exploration.ipynb` visualizes trends in closing prices, analyzes volatility, checks for missing values, and summarizes key statistics. These insights helped guide feature engineering and model selection.
- **Data Preprocessing:** Prior to training, several preprocessing steps were applied:
  - **Normalization:** For the Ridge model, features were normalized using Min-Max scaling:

    $$x_{\text{scaled}} = \frac{x - x_{\text{min}}}{x_{\text{max}} - x_{\text{min}}}$$

    This ensured all input features were scaled to the range $[0, 1]$, improving model convergence.
  - **Sequence Windowing:** For the LSTM model, a rolling window approach was used to transform time series data into supervised learning format:

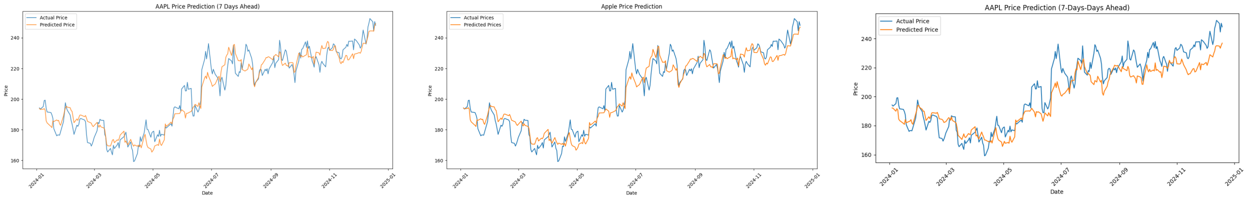
    ```
    def create_sequences(data, window_size):
        X, y = [], []
        for i in range(len(data) - window_size):
            X.append(data[i:i + window_size])
            y.append(data[i + window_size])
        return np.array(X), np.array(y)
    ```

    This allowed the model to learn temporal dependencies by looking at fixed-length sequences of past data to predict future values.
- **Model Training and Validation:** Three machine learning models were developed and evaluated:
  - `02_model_testing.ipynb` implements an XGBRegressor model, utilizing `RandomizedSearchCV` with `TimeSeriesSplit` for hyperparameter tuning. This allowed the model to optimize parameters like learning rate, depth, and number of estimators without data leakage from future values.
  - `02_model_test_lstm.ipynb` constructs a Long Short-Term Memory (LSTM) network using Keras. The architecture includes layers such as `LSTM`, `Dropout`, and `Dense`, and was trained using the Adam optimizer and MSE loss.
  - `ridge_model.ipynb` applies Ridge Regression as a linear baseline, incorporating normalized features and 7-day rolling targets. This model serves as a benchmark for comparing non-linear models.
- **Evaluation:** Each model's performance was assessed using Mean Squared Error (MSE) as the loss metric. Visualizations included line plots of true vs. predicted prices over time to qualitatively assess model behavior.

3 Results

This section presents the 7-day stock price predictions for three companies—Apple, Chevron, and Johnson & Johnson—using the XGBRegressor, Long Short-Term Memory (LSTM), and Ridge Regressor models. Each company's results are displayed with side-by-side model comparisons to facilitate evaluation.

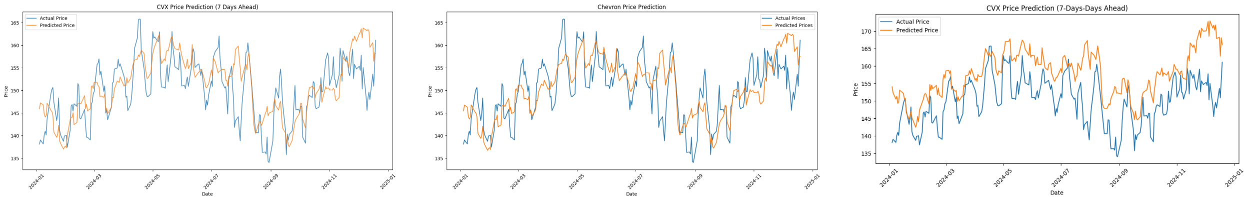
3.1 Apple (AAPL)



(a) XGBRegressor (b) LSTM (c) Ridge Regressor

Figure 1: Stock price predictions for Apple (AAPL) using three models.

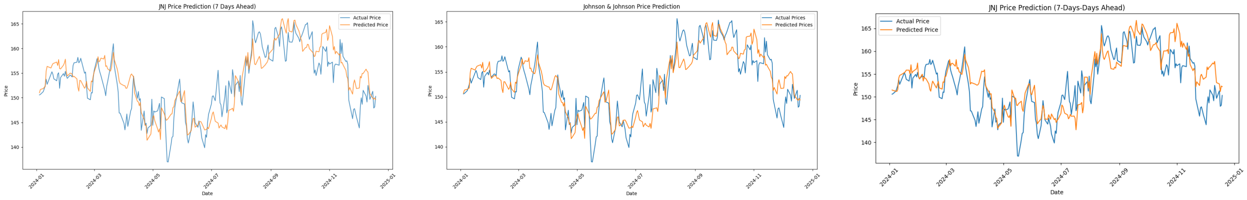
3.2 Chevron (CVX)



(a) XGBRegressor (b) LSTM (c) Ridge Regressor

Figure 2: Stock price predictions for Chevron (CVX) using three models.

3.3 Johnson & Johnson (JNJ)



(a) XGBRegressor (b) LSTM (c) Ridge Regressor

Figure 3: Stock price predictions for Johnson & Johnson (JNJ) using three models.

4 Discussion

In this study, several machine learning models were explored to evaluate their performance in learning. The performance of the three implemented models—XGBRegressor, LSTM, and Ridge Regression—varied significantly in their ability to predict short-term stock prices. Each model demonstrated distinct strengths and limitations, particularly when evaluated across diverse stock tickers from different market sectors. Each of these approaches brought unique advantages and challenges, which we will explore in detail below.

4.1 XGBRegressor Model

The XGBRegressor consistently delivered strong predictive accuracy across the majority of the selected stocks. Its robustness can be attributed to its ensemble learning structure and ability to handle non-linear relationships in the data. Moreover, the use of `RandomizedSearchCV` with a `TimeSeriesSplit` cross-validation strategy helped mitigate overfitting and ensured that temporal dependencies in the data were respected during training. Stocks with relatively stable trends saw the most reliable predictions, while more volatile stocks exhibited larger deviations between actual and predicted values.

4.2 Long-Short Term Memory (LSTM) Model

The LSTM model, although theoretically well-suited for time series forecasting due to its memory retention capabilities, showed mixed results. It performed reasonably well on stocks with smoother price trajectories but struggled with highly volatile or abrupt trend shifts. This may be due to the model's sensitivity to hyperparameter tuning and the limited size of training sequences. Additionally, the LSTM's performance was constrained by the need for normalized and sequenced data, which may have contributed to reduced flexibility across different ticker behaviors.

4.3 Ridge Regression Model

In contrast, the Ridge Regression model served as a simple linear baseline. As expected, it underperformed in comparison to the more complex models, particularly in capturing non-linear dynamics and sharp fluctuations. However, it provided a valuable reference point for evaluating the effectiveness of more advanced techniques. Its relatively faster training time and lower computational cost could make it a suitable choice for scenarios where model interpretability or resource constraints are prioritized.

Overall, the XGBRegressor emerged as the most balanced and reliable model for 7-day stock price forecasting in this study. The results also highlight the importance of aligning model complexity with data characteristics and forecasting objectives. Future work may explore hybrid models or incorporate additional financial indicators to further enhance prediction accuracy.

4.4 Comparison of Algorithms

Table 1 summarizes the strengths and weaknesses of each model:

Learning Model	**Strengths**	**Weaknesses**
XGBRegressor	Handles non-linear patterns well; robust performance across most stocks; effective hyperparameter tuning via RandomizedSearchCV	Can be computationally intensive; requires careful tuning to avoid overfitting
Long-Short Term Memory	Designed for sequential data; retains temporal dependencies; good for smooth time series	Sensitive to hyperparameters; less effective with highly volatile data; longer training time
Ridge Regressor	Simple and fast to train; interpretable coefficients; good baseline model	Limited to linear relationships; poor performance with complex or volatile patterns

Table 1: Comparison of Machine Learning Models for Stock Price Prediction

This project explored the effectiveness of three machine learning models—XGBRegressor, Long Short-Term Memory (LSTM), and Ridge Regression—in forecasting short-term stock prices across a diverse set of companies. Using historical stock price data and financial indicators from January 1st, 2020 - January 1st, 2025, we trained and evaluated each model on its ability to generate 7-day forward predictions.

Among the models tested, the XGBRegressor demonstrated the highest predictive accuracy and generalizability, particularly for stocks with moderate volatility. The LSTM model showed promise but proved more sensitive to data preparation and hyperparameter tuning, leading to inconsistent results. The Ridge Regression model, while less accurate, served as a useful baseline and demonstrated the trade-off between simplicity and predictive performance.

These findings underscore the potential of machine learning in financial forecasting, while also highlighting the importance of model selection and data preprocessing. Future work could involve extending the prediction horizon, integrating additional economic indicators, or employing ensemble and hybrid modeling approaches to further improve robustness and accuracy.