Stock Price Prediction Using Machine Learning

Yoav Altman

August 24, 2025

1 Introduction

This project aims to compare a deep learning model, a long short-term memory (LSTM) network, to an autoregressive-type model and a benchmark linear regression model in the prediction of the price of the SPY ETF 30 days into the future. The main goal is to understand whether the added complexity of the LSTM translates to greater accuracy of predictions.

Working with financial data comes with a few challenges. One of the biggest is that many technical indicators do not carry useful predictive information. Moreover, stock prices are volatile with a complex autocorrelation structure.

2 Data

Daily SPY OHLCV (open, high, low, close, volume) data from 1998 to 2025 was downloaded using the `yfinance` library in Python. From this dataset, a large number of technical indicators were computed — including trend, momentum, volatility, volume, regime change, percentile characteristics, and lagged close prices — across different time windows.

Since many of these indicators require a lookback period (the number of past days that define a window for calculation), complete data was not available on the first m observations (where m corresponds to the largest lookback window among the indicators). We therefore removed the first m rows prior to analysis.

We did not encounter any missing values in our downloaded data provided by `yfinance`), but in order to allow predictions using other stocks or timeframes we implemented the forward-fill (`ffill`) and backward-fill (`bfill`) methods to safeguard against missing values by propagating the nearest valid observation forward or backward in time. We treated volumes recorded as zero as missing values.

The datetime index was reset to facilitate plotting. For the purpose of analysis, we also added time and time-squared columns, where time is measured in number of days past day m.

The dataset was then split into training, validation, and test sets. For the linear regression and autoregressive-type models, the validation set was combined with the training set for fitting the models. For the LSTM, the validation

set was used for hyperparameter tuning. We then combined the training and validation sets to refit the best model. The predictive performance of all three models was evaluated on the test set, which was not seen during training.

3 Methods

This project compares three predictive models: a linear regression model, an autoregressive-type model, and an LSTM neural network. The benchmark linear regression model used only time and time-squared as input features, while the other two models used time, time-squared, and the technical indicators.

We focus on prediction 30 days ahead. The linear regression and autoregressive-type models directly predict the price of the SPY ETF. In contrast, the LSTM predicts the 30-day price change of the SPY ETF.

3.1 Linear Regression Model

Linear regression is one of the most widely used methods in statistics and machine learning. It models the relationship between a dependent variable and one or more independent variables (features). We use this model as a benchmark in this project, a simple baseline for comparison against the more complex autoregressive-type model and the LSTM neural network.

Specifically, let Y_t be the observed SPY price at time t, $t = 1, \ldots, n$. We assume the following model:

$$Y_t = \beta_0 + \beta_1 t + \beta_2 t^2 + \epsilon_t,$$

where ϵ_t is an error term capturing the deviation between the observed and expected value at time t. In the standard linear regression model, we assume that these errors are independent and that $\epsilon_t \sim N(0, \sigma^2)$ for all t. In our time-series setting, the independence assumption is clearly violated; prices are highly autocorrelated. However, this violation is unimportant because we use this model only for point predictions, not inference.

Estimates of the coefficients, $\hat{\beta}_i$, $i = 0, 1, 2$, are obtained using the method of ordinary least squares (OLS). We then predict price at time t as

$$\hat{y}_t = \hat{\beta}_0 + \hat{\beta}_1 t + \hat{\beta} t^2$$

where $\hat{Y}_t$ is the predicted price at time t and $\hat{\beta}_i$ are the estimated coefficients.

3.2 Autoregressive-Type Model

We also consider an autoregressive-type model where the current price is specified in terms of functions of past prices. In particular, we assume the model

$$Y_t = \beta_0 + \beta_1 t + \beta_2 t^2 + \sum_{i=3}^{k} \beta_i x_{ti} + \epsilon_t,$$

where x_{ti} is the value of the i^{th} feature at time t, $i = 3, \ldots, k$, $t = 1, \ldots, n$. We assume that the errors are independent and that $\epsilon_t \sim N(0, \sigma^2)$ for all t. The features are derived from past prices and volume and include quantities such as moving averages and RSI. The complexity of this model makes checking the validity of its assumptions challenging. However, again, model violations are not necessarily concerning because we are not using the model for inference.

We estimate the β_i's using least squares and predict the price at time t as

$$\hat{y}_t = \hat{\beta}_0 + \hat{\beta}_1 t + \hat{\beta} t^2 + \sum_{i=3}^{k} \hat{\beta}_i x_{ti}.$$

3.3 LSTM Neural Network

LSTM networks are designed for sequential data such as time series. At each time point, the LSTM maintains two vectors: the hidden state, which produces the output, and the cell state, which serves as long-term memory. Three gates regulate the flow of information. The forget gate scales how much of the past information is retained, the input gate controls how much new information is added, and the output gate determines how much of the memory influences the current prediction. Together, these mechanisms allow the model to selectively preserve or discard information and capture dependencies that unfold over many time steps.

The LSTM differs from a standard neural network, which maps inputs directly to outputs with no memory. A standard neural network treats each input independently, so it cannot recognize patterns that depend on order or timing unless these patterns are engineered into the features. In contrast, an LSTM updates its internal states so each prediction reflects both the current input and the accumulated information from prior steps. The ability to handle autocorrelation in prices observed over time makes LSTMs better-suited than standard neural networks at dealing with financial data.

While LSTMs offer clear advantages in modelling sequential data, they also come with limitations. Their architecture introduces many parameters, which increases the risk of overfitting when training data is limited. They are also computationally expensive, making them slower to train compared to simpler models. Finally, while LSTMs can capture long-range dependencies, they are not guaranteed to do so effectively if the signal is weak or unstable, as is often the case in financial markets. These trade-offs mean that the added complexity of an LSTM is not guaranteed to result in better predictive performance than an autoregressive-type model or a linear model.

3.4 Evaluation Procedure

The performance of the models was evaluated using mean-squared error (MSE) and coefficient of determination (R^2). These metrics capture different aspects of performance, and interpreting them correctly is essential.

R^2 measures the proportion of observed variability in the target variable that is explained by the model. It is defined as

$$R^2 \;=\; 1 - \frac{\sum_t \left(y_t - \hat{y}_t\right)^2}{\sum_t \left(y_t - \bar{y}\right)^2},$$

where y_t is the observed value of the target variable (price in the case of the linear regression and autoregressive-type models and price change in the case of the LSTM model) at time t and $\bar{y}$ is the average of the observed values of the target variable. The summations are over all values in the test set.

4 Analysis and Results

In this section, we report on the relative predictive performance of the three models.

On the training and validation data, Figure 1 shows that the linear regression model provides a reasonable description only of the overall trend, while both the autoregressive-type model and LSTM model track the observed price quite closely.

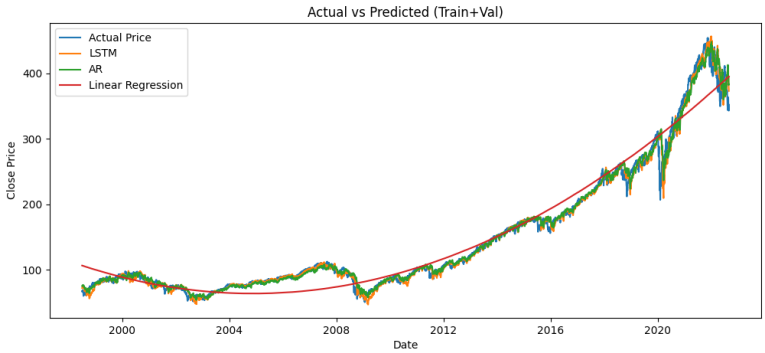

Figure 1: Observed and predicted close price vs. date (training and validation data)

On the test data, Figure 2 shows that the linear regression model no longer captures the overall trend well, although it does still capture the upward trend. On the other hand, while both the autoregressive-type and LSTM models' predictions are farther away from the actual prices than they are in the training and validation data, they track the actual price far better than do the linear regression model predictions. Table 1 summarizes the MSE and R^2 values for the three models using the test data; these values are consistent with the patterns observed in the plot. In contrast to the autoregressive-type and LSTM models, the linear regression model does not use information about autocorrelation in

the prices when predicting future price, which presumably explains its relatively poor performance.

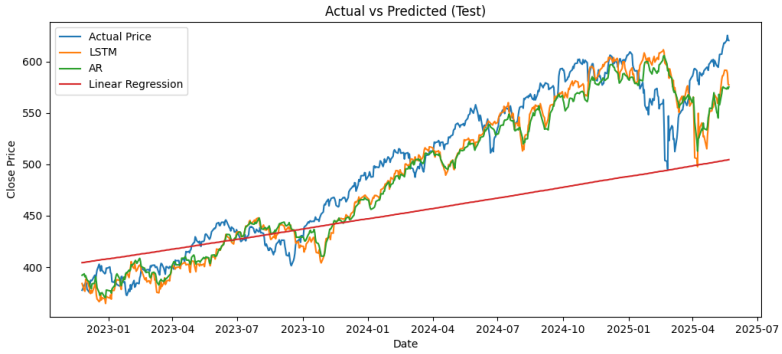

Figure 2: Observed and predicted close price vs. date (test data)

Model	**Dataset**	**MSE**	R^2 **Score**
LSTM	Test	616	0.886
AR	Test	592	0.904
Linear	Test	3815	0.381

Table 1: Performance of the linear regression, autoregressive-type, and LSTM models when predicting 30-day-ahead prices (test data)

Interestingly, the autoregressive-type and LSTM model predictions are very close, which may indicate that the two fitted models are quite similar. In addition, the plots show that predictions did not immediately reflect major price moves; rather changes in the predictions tended to lag behind changes in actual price by 30-60 days.

5 Conclusion

This project compared a benchmark linear regression model with an autoregressive-type model and an LSTM neural network for predicting the price of the SPY ETF 30 days ahead using time and technical indicators derived from historical prices and volume.

The autoregressive-type and LSTM models predicted price far more accurately than did the linear regressive model, using information in the autocorrelation of the prices to capture some of the volatility in the prices. They also responded to changes in the underlying trend of the prices — though with a 30-60 day delay.

Future work could explore incorporating features such as sentiment or economic indicators to improve the accuracy of the predictions further.

Overall, the results show the importance of model architecture when making predictions using financial time series data.