Time-Series Forecasting of Tesla Stock Prices

Introduction:

Forecasting stock prices is always challenging due to their noisy and volatile nature. In this project, I implemented and compared a traditional statistical model (**ARIMA**) and a deep learning model (**LSTM**) to forecast Tesla's daily closing prices. I also applied a **rolling window evaluation** to assess the models more realistically, and finally deployed the trained LSTM model on Hugging Face for interactive use.

Data and Preprocessing:

- **Dataset:** I collected Tesla (TSLA) daily stock prices for the last 15 years using *Yahoo Finance*.

For ARIMA

- I used only the **closing price** for training and forecasting.
- The dataset was split into **80% training and 20% testing**.
- No feature scaling was required since ARIMA works directly on raw values.

For LSTM

- I used multiple features: **Close, Open, High, Low, Volume**.
- Missing values were dropped and the dataset was normalized using **MinMaxScaler** (fit only on the training data).

- The dataset was split into **70% training, 10% validation, and 20% testing**.

- I converted the time series into **supervised sequences** with a **60-day lookback window**, where the model predicts the next day's closing price from the past 60 days of features.

Models Implemented:

ARIMA

- I first tested for stationarity using the **ADF test** and found the series required first differencing (*d=1*).

- I used **AutoARIMA** to determine the optimal order and it selected (2,1,1) based on AIC.

- I implemented:

    - A **baseline ARIMA forecast**, trained once and applied on the test set.

    - A **rolling window ARIMA forecast**, where the model updates step-by-step with new data.

LSTM

- I built a stacked LSTM model with:

- 2 LSTM layers (64 units each)

- Dropout layers (0.2) to reduce overfitting

- Dense layers with ReLU and linear output for the final prediction
- The model was trained with the **Adam optimizer** (lr=0.001) using early stopping and a learning rate scheduler.
- I implemented:
  - A **baseline LSTM forecast**, predicting the full test horizon.
  - A **rolling window LSTM forecast**, predicting one step at a time in sequence.

Results and Comparison:

Model	RMSE	MAPE(%)
ARIMA baseline	76.99	29.71
ARIMA rolling	9.88	2.85
LSTM baseline	19.74	5.94
LSTM rolling	21.86	7.15

My observations:

- **ARIMA rolling** gave the best performance with the lowest error (RMSE ≈ 9.88, MAPE ≈ 2.85%).
- **LSTM baseline** was competitive but less effective in rolling forecasts, where errors accumulated.

Overall, ARIMA proved more robust for short-term sequential prediction, while LSTM showed potential but required more tuning and possibly more features.

Conclusion:

In my experiments, the **ARIMA rolling window model** generalized best. This could be due to the highly volatile nature of Tesla's stock prices, which made the deep learning model more susceptible to overfitting. ARIMA, despite being simpler, was able to capture short-term autocorrelation effectively.

Deployment:

I retrained the LSTM model on the full dataset and deployed it on Hugging Face. The deployed app allows users to input how many days ahead they want to forecast Tesla's stock price:
https://huggingface.co/spaces/foyez767/DataSynthis_ML_JobTask