Stock Price Prediction Challenge

Exploratory Data Analysis (EDA)

1. Data Preprocessing.

- **Handling Missing Values** – The Close column had some missing values, which were filled using a **5-day rolling mean** to keep the trend consistent. A shorter window (like 3 days) caused the predictions to become unstable, while a longer window (like 10 days) removed important details. The 5-day window was chosen because it matched the natural changes in the Close price and worked well with the goal of predicting the next 5 days.

- **Feature Scaling** – The data was scaled using MinMaxScaler to keep the values between 0 and 1. MinMaxScaler was chosen over other scalers like StandardScaler and RobustScaler because it keeps the original shape of the data while making sure all values fit into a fixed range. Since LSTM models are sensitive to the scale of input data, MinMaxScaler helped the model train better and faster.

- **Feature Engineering** – A new feature was created based on the difference between High and Low prices to capture price volatility. However, it didn't improve prediction accuracy and added noise, so it was removed. Using only the Close price gave the best results.

2. Visualizations and Key Patterns

Visualizing the data needed to better understand the patterns and relationships between the different features.

- **Close vs. Adjusted Close Price Over Time**
  - Close and Adj Close prices followed a similar pattern, meaning they reflect similar market behavior.
  - Overall, the trend was upward with occasional short-term drops
- **Volume Over Time**
  - Large spikes in trading volume were seen before big price changes, but the correlation between Volume and Close was low (around 0.30).
  - Adding Volume to the model made it more complicated without improving accuracy, so it was excluded.

- **Open vs. Close Price Over Time**
    - Open and Close prices followed similar patterns, but price gaps (differences between open and close) happened more often during market instability.
    - Including both Open and Close prices made the model more complex without improving accuracy, so Open was excluded.

- **High, Low, and Close Price Over Time**
    - High and Low prices closely tracked the Close price, showing that they reflect short-term changes rather than long-term trends.
    - The model couldn't find any useful patterns from High and Low prices.
    - Including them added noise to the model without improving predictions, so they were excluded.

- **Correlation Matrix**
    - Open, High, Low, and Close had strong positive correlations (between 0.95 and 0.99).
    - Volume had a low correlation (~0.30) with Close, explaining why it didn't improve predictions.
    - Since the price-based features were highly related to each other, using just Close price avoided redundancy and kept the model simple.

- **Pair Plot**
    - Strong correlation was seen between Close, High, and Low.
    - Volume followed a different pattern, supporting the decision to exclude it.
    - The patterns confirmed that Close price alone was enough for prediction.

3. Trend and Seasonality Analysis

- **Trend Analysis** – The overall trend showed that the stock price increased over time, with some drops.

- **Seasonality** – Stock prices followed regular patterns.

- **Anomalies** – Large spikes in trading volume and sudden price changes were seen during certain periods of market instability.

4. Justification for Feature Selection

Choosing the right features was important to avoid overfitting and improve prediction accuracy.

- **Close Price** – Selected because it gave the highest prediction accuracy.
- **Open, High, Low** – These had high correlation with Close, but adding them didn't improve predictions, so they were removed to avoid redundancy.
- **Volume** – Weak correlation with Close price (~0.30), so it was excluded.
- **Difference Between High and Low** – Tried to use this to capture price changes, but it added noise and didn't help predictions.
- **Engineered Features** – Several new features were tested, but none worked better than using only the Close price.

5. Justification for Moving Average Selection

Different moving averages were tested to find the best balance between noise and trend.

- **Below 60-day(30-days) moving average** – Best visual fit but poor predictive accuracy.
- **60-days moving average** – Gave the best prediction accuracy and balanced noise and trend strength.
- **Above 60-day(70-days) moving averages** – These made the model more complex without improving predictions.

A **60-days moving average** was selected because it gave the best balance between capturing trends and avoiding overfitting.

- Because of that, for the Sequence Length, 60-day window was used since it gave the best prediction accuracy.

Model Selection

1. Comparison of Different Modeling Approaches

Several different models were tested to find the best-performing approach for predicting the next 5-day closing prices. Each model was evaluated based on its prediction accuracy and how well it handled time-series patterns.

- **LSTM (Long Short-Term Memory)**
    - LSTM was chosen because it is designed to handle sequential data and time-series patterns.
    - Multiple variations of LSTM were tested:
        - **Single LSTM Layer** – Simple architecture but gave lower accuracy due to underfitting.
        - **Two LSTM Layers** – Improved predictive accuracy and captured more complex patterns.
        - **Bidirectional LSTM** – Attempted to capture both past and future dependencies in the sequence, but increased complexity without improving accuracy.
- **GRU (Gated Recurrent Unit)**
    - GRU is similar to LSTM but has fewer parameters, making it computationally faster.
    - Tested GRU models showed faster training times, but accuracy was lower than LSTM models.
    - GRU models struggled to capture long-term dependencies in the data.

- **Dense Neural Networks (Fully Connected)**
    - A simple dense neural network was tested using the same input features.
    - Accuracy was low because dense networks are not designed for sequential data.
    - Dense networks failed to capture the time-dependent nature of stock price movements.
    -

2. Explanation of Evaluation Metrics

Different evaluation metrics were used to compare model performance and select the final model.

- **Root Mean Squared Error (RMSE):**
    - Lower RMSE indicates that the model's predictions are closer to actual values.
    - RMSE was selected because it gives more weight to larger errors, which helps evaluate the model's ability to handle significant price changes.

- **Mean Absolute Error (MAE):**
    - MAE measures the average of the absolute differences between predicted and actual values.
    - It provides a direct measure of prediction accuracy but does not highlight large errors as much as RMSE.

- **Directional Accuracy:**

    - Measures how often the model correctly predicts the upward or downward movement of stock prices.
    - Useful for evaluating the model's effectiveness in predicting market trends rather than exact values.

- **Training Loss:**

    - Measures how well the model fits the training data.
    - Lower training loss indicates better model convergence and learning.

3. Justification for Final Model Choice

After testing various models and architectures, a two-layer LSTM model was selected as the final model based on its superior performance and predictive accuracy.

- **Model Architecture:**

    - First LSTM layer with 128 units and `return_sequences=True` to allow information flow to the next layer.
    - Second LSTM layer with 64 units and `return_sequences=False` to capture long-term dependencies.
    - Two dense layers were added:
        - First dense layer with 25 neurons to improve feature extraction.
        - Final dense layer with 1 neuron for output (predicted closing price).
- **Added Dropout and BatchNormalization:**

    - Dropout and batch normalization layers were added to improve generalization and avoid overfitting:
        - Dropout with 0.2 probability after each LSTM layer to prevent overfitting.
        - Batch normalization to reduce internal covariate shift and improve learning stability.

    - However, adding dropout and batch normalization did not improve prediction accuracy. The model became slower to train and less stable during testing.
- **Additional Dense Layer:**
    - An extra dense layer was added with 50 neurons to increase complexity and improve feature extraction.
    - However, this increased model complexity and training time without improving prediction accuracy.
    - Removing the extra dense layer improved stability and reduced overfitting.
- **Learning Rate and Optimization:**
    - Adam optimizer with a learning rate of 0.001 was used because it provided the best balance between training speed and prediction accuracy.
    - ReduceLROnPlateau and RMSprop were also tested, but both gave lower prediction accuracy on the test set.
    - ReduceLROnPlateau reduces the learning rate when the loss stops improving, but it caused slower convergence without improving the final result.
    - Adam optimizer adjusted the learning rate dynamically, which allowed the model to train faster and generalize better.
- **Sequence Length:**
    - A sequence length of 60 days was selected based on performance tests.
    - A 60-day sequence captured both short-term fluctuations and long-term trends, improving prediction accuracy.
- **Why LSTM Was Chosen:**
    - LSTM performed better than GRU and dense models because it is specifically designed for sequential data.
    - Bidirectional LSTM and deeper models overfitted the data without improving accuracy, making the two-layer LSTM the optimal balance of complexity and performance.

4. Analysis of Model Limitations and Potential Improvements

- **Limitations**
    - **Relatively Small Dataset:**
        - The dataset size was relatively small compared to the complexity of stock price fluctuations.
        - Stock prices can change based on hidden patterns that might require more data to identify accurately.

        - A larger dataset could help the model capture more complex price movements and reduce prediction errors.
    - **High Correlation Between Features:**
        - Open, High, Low, and Close prices were highly correlated, which introduced feature redundancy.
        - Removing redundant features improved performance, but more advanced techniques could be tested to extract additional patterns.

- **Potential Improvements**
    - **Technical Indicators:**
        - Additional technical indicators such as Relative Strength Index (RSI) and Bollinger Bands were tested to provide more context for market movements.
        - However, adding these features increased model complexity without improving prediction accuracy.
    - **More Data:**
        - Increasing the size of the dataset would likely improve generalization and reduce overfitting.
        - A larger dataset would allow the model to learn more complex patterns and adjust better to market volatility.
    - **Advanced Models with More Data:**
        - With a larger dataset, testing more complex models such as **transformer-based models** or **hybrid models** (combining LSTM with CNN) could help capture complex time-series patterns.
        - Transformer models could improve performance by focusing on important time steps, but they require large datasets to avoid overfitting.