Stock Market Prediction Using Machine Learning

1. Introduction:

The stock market is highly volatile, and predicting stock prices is a challenging task. This project aims to develop a machine learning model that fetches real-time stock market data, processes it, and predicts stock prices using Linear Regression. The project follows a structured approach, integrating data fetching, processing, training, evaluation, and visualization. The model mainly analyzes IBM stock prices, but it can be modified to predict other stocks.

2. Objective : The main goals of this project are:

- Retrieve real-time stock data from the Alpha Vantage API.
- Process and clean the stock data for analysis.
- Train a machine learning model to predict stock prices.
- Evaluate the model's performance using appropriate metrics.
- Visualize the predictions and generate a comprehensive report.

3. Tools & Technologies Used:

- Python: The primary programming language used for implementation.
- Alpha Vantage API: Used to fetch real-time stock market data.
- Pandas: For data processing and cleaning.
- Scikit-learn: Used for implementing the Linear Regression model.
- Matplotlib: For visualizing actual and predicted stock prices.
- JSON: For storing and generating reports.

4. Project Overview:

This project consists of multiple modules, each responsible for a specific function. Below is a detailed breakdown of each module:

4.1 Data Fetching Module

- Function: Fetches real-time stock data using the Alpha Vantage API.
- Implementation:
    - Sends an API request to retrieve stock market data at 5-minute intervals.

  - Converts the JSON response into a Python dictionary.
  - Handles errors if the API request fails.

4.2 Data Processing Module

- Function: Cleans and structures the fetched data.
- Implementation:
  - Converts raw JSON data into a pandas DataFrame.
  - Renames columns for clarity.
  - Ensures all values are numeric and removes any missing or invalid data.
  - Sorts data based on the timestamp to maintain chronological order.

4.3 Model Training Module

- Function: Trains a machine learning model using historical stock data.
- Implementation:
  - Uses the Linear Regression algorithm.
  - Splits data into training and testing sets.
  - Fits the model to the training data.

4.4 Model Evaluation Module

- Function: Evaluates the model's performance.
- Implementation:
  - Uses Mean Squared Error (MSE) to measure the accuracy of predictions.
  - Compares actual and predicted stock prices.

4.5 Prediction Module

- Function: Uses the trained model to make stock price predictions.
- Implementation:
  - Takes recent stock data as input.

- ○ Predicts the closing price of a stock.

4.6 Visualization Module

- Function: Displays the actual vs. predicted stock prices in a graphical format.
- Implementation:
    - Uses Matplotlib to generate a line chart.
    - Differentiates actual prices (blue) and predicted prices (red).

4.7 Report Generation Module

- Function: Stores model performance metrics and predictions.
- Implementation:
    - Saves Mean Squared Error and predicted values in a JSON file.

4.8 Live Prediction Module

- Function: Predicts stock prices based on the latest market data.
- Implementation:
    - Retrieves the most recent stock price data.
    - Uses the trained model to generate a prediction.
    - Prints the predicted closing price.

5. Results

- The model was able to predict stock prices based on historical data.
- Visualizing the actual vs. predicted prices helped in understanding market trends.
- A JSON report was generated containing key insights.

6. Challenges Faced

- The model does not account for real-world influences like news, global events, or economic shifts.
- Limited feature selection affects prediction accuracy.

- API rate limits sometimes interrupted data retrieval.

7. Conclusion

This project successfully demonstrates stock price prediction using machine learning. While it has limitations, it provides a solid foundation for future improvements.