Stock Price Prediction

Submitted By

- Rineesh M S
-

1. Aim

To develop a Streamlit web application that predicts future stock prices using real-time stock data, technical analysis indicators, and a machine learning model (Linear Regression).

2. Problem Statement

Stock market forecasting is complex due to market volatility and external factors. This project aims to simplify stock prediction for educational and analytical purposes using a user-friendly tool that:
- Collects historical stock data
- Applies technical indicators
- Trains a prediction model
- Displays visual comparisons and prediction accuracy

3. Requirements

Software & Libraries:
- Python 3.7 or above
- pip (Python package manager)

Libraries to Install:

```
pip install yfinance pandas numpy streamlit matplotlib scikit-learn
```

Or use:

```
pip install -r requirements.txt
```

4. Procedure

Step 1: Launch Application

1. Download the file Stock_price.py
2. Run the app in terminal:

```
   streamlit run Stock_price.py
```

Step 2: Select Stock
- Choose from a list of popular stocks or enter a valid ticker symbol (e.g., AAPL, GOOGL)

Step 3: Choose Date Range
- Use preset (1 year, 3 years, etc.) or custom date range
- Adjust test size percentage (10%–40%)

Step 4: Start Prediction
- Click on “Start Prediction Process”
- The app fetches data, calculates indicators, and trains a Linear Regression model

Step 5: Analyze Output
- View historical stock prices and technical indicators

- See prediction vs actual prices on an interactive chart
- Check performance metrics (MSE, MAE, R^2, Accuracy %)
- View next-day predicted price

5. Technical Details

Technical Indicators Used:
- Moving Averages (5, 10, 20, 50 days)
- RSI (Relative Strength Index)
- MACD (Moving Average Convergence Divergence)
- Volatility (20-day rolling std. deviation)
- Daily Returns

Machine Learning Model:
- Algorithm: Linear Regression
- Features: 14 technical indicators
- Training: Based on historical stock data
- Validation: Time-series data split (to avoid data leakage)

6. Results Interpretation

Performance Metrics:
- MSE (Mean Squared Error): Lower is better
- MAE (Mean Absolute Error): Average difference
- R^2 Score: Closer to 1 means better model fit
- Accuracy %: Approximate match between predictions and real values

Charts:
- Blue line: Actual stock price
- Red line: Predicted price
- Hoverable tooltips for detailed values

7. Limitations

- Not suitable for making real financial decisions
- Doesn’t factor in news or sudden events
- Short-term focused (not reliable for long-term investment)
- Accuracy may decrease with high volatility

8. Troubleshooting

Common Issues:

No data found for ticker:
- Check the symbol

- Ensure stock is publicly traded
- Try different date range

Error fetching data:
- Check your internet
- Retry after some time

Poor model performance:
- Increase historical range
- Try a less volatile stock
- Adjust test/train data split

9. System Requirements

- Minimum RAM: 4 GB
- Active internet connection
- Browser: Chrome, Firefox, Safari, or Edge

10. Code Structure

```
Stock_price.py
├── Imports and Setup
├── Streamlit UI Configuration
├── Data Loading
├── Feature Engineering
│   ├── Technical Indicators
│   ├── Moving Averages
│   └── RSI/MACD Calculation
├── Model Training
├── Evaluation
└── Visualization
```

11. Conclusion

This application provides a simplified way to understand and visualize stock market predictions using technical indicators and machine learning. It is designed for educational and analytical purposes to explore the potential and limitations of predictive stock analysis.

12. Disclaimer

This tool is for educational purposes only. It does not offer financial advice. Always consult a financial expert and do your own research before making any investment decisions.