Project Report: Stock Price Prediction using Machine Learning in Python

# Project Overview

This project focuses on predicting stock prices using historical stock data and machine learning models. By leveraging time series data, it aims to model and forecast stock trends to aid in decision-making for investors and analysts.

# Objectives

• Analyze and visualize historical stock data.

• Build predictive models to forecast future stock prices.

• Evaluate the performance of these models.

• Provide actionable insights from predictions.

# Data Collection

• Data Source: Stock data is collected using the yfinance API.

• Stock Ticker Used: e.g., 'AAPL', 'GOOG', etc. (can be configured).

• Features Collected:

- Open, High, Low, Close prices

- Adjusted Close

- Volume

* Date Range: Typically, over several years for modeling.

# Exploratory Data Analysis (EDA)

• Displayed descriptive statistics of stock prices.

• Visualized trends using line plots:

- Closing price over time

- Moving averages (100-day and 200-day)

* Moving averages helped smooth the price curves for better trend visualization.

# Data Preprocessing

• Selected only the 'Close' price for modeling.

• Normalized the data using MinMaxScaler to scale between 0 and 1.

• Created sequences for supervised learning:

- Used 100 previous days as input to predict the 101st day.

# Model Building

• Used Long Short-Term Memory (LSTM) neural networks, suitable for time series data.

• Model architecture:

- LSTM layers

- Dropout layers for regularization

- Dense layer for output prediction

* Compiled with:

- Loss: 'mean\_squared\_error'

- Optimizer: 'adam'

# Model Training

• Data was split into training and testing sets.

• Model was trained over multiple epochs (e.g., 100).

• Batch size was chosen to optimize training speed and accuracy.

# Prediction and Evaluation

• Model predicted the next-day prices based on past 100 days.

• Evaluation metrics:

- RMSE (Root Mean Squared Error) was used to quantify prediction accuracy.

• Visual comparison:

- Predicted vs Actual closing prices were plotted to assess performance.

# Results & Observations

• The model captured overall trends but might lag in sharp changes due to the smoothing nature of LSTM and use of moving averages.

• Performance can be further improved by:

- Tuning hyperparameters

- Incorporating additional features (technical indicators)

- Using ensemble models

# Conclusion

• The project demonstrates the effectiveness of using LSTM for stock price forecasting.

• While results are promising, stock markets are influenced by many non-quantifiable factors.

• Model should be used as a supporting tool alongside other financial analysis methods.

# Future Work

• Include other stocks and compare model generalization.

• Add sentiment analysis (e.g., from financial news or social media).

• Deploy model via Streamlit or Flask for real-time use.

• Integrate technical indicators (RSI, MACD, Bollinger Bands).