Phase-2

Student Name: Sasivardhan.S.K

Register Number: 410723106029

Institution: Dhanalakshmi College of Engineering

Department: Electronics and Communication Engineering

Date of Submission: 08/05/2025

Github Repository Link:

https://github.com/SKSasivardhan/NM_sasivardhan--DS

---

1. Problem Statement

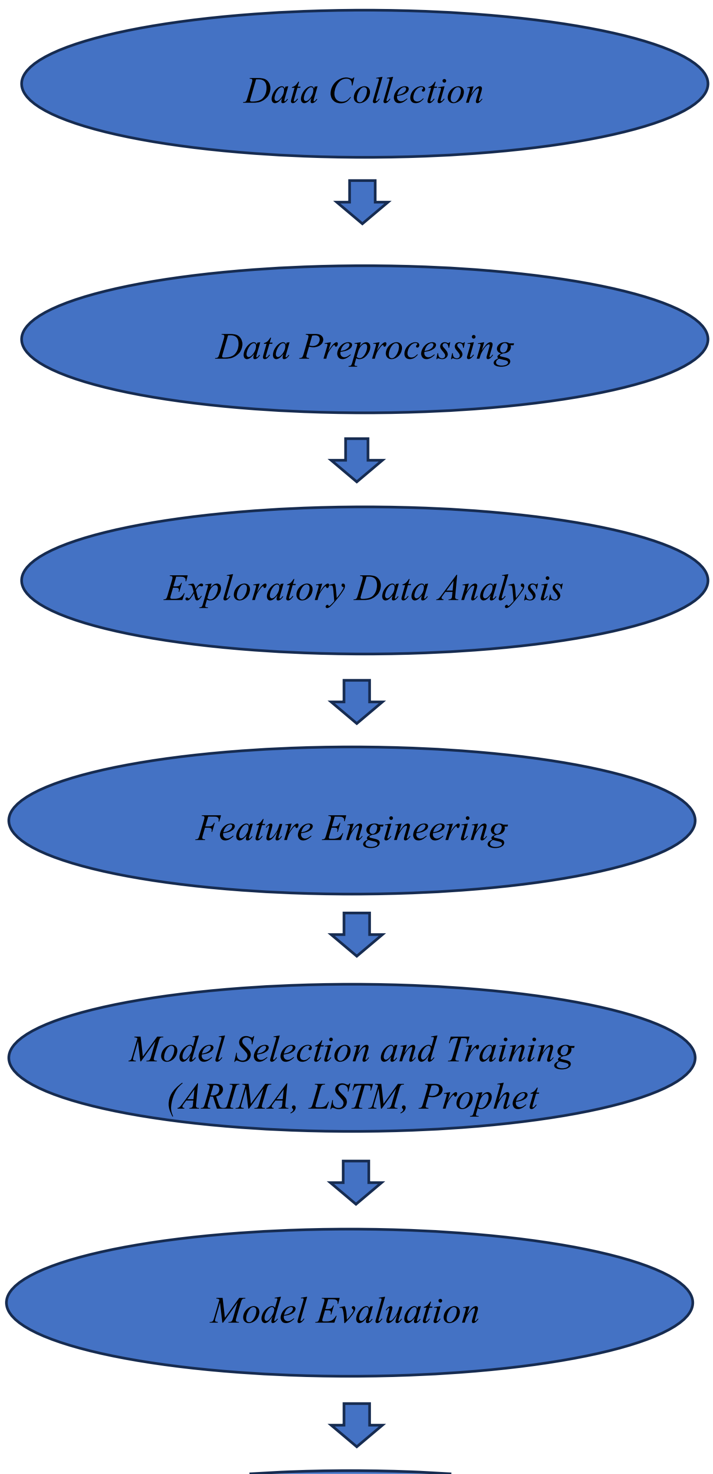
Cracking the Market Code with AI-Driven Stock Price Prediction Using Time Series Analysis

Stock market prices are inherently volatile and influenced by numerous factors, making accurate forecasting a complex task. Investors and financial analysts need reliable predictions to make informed trading decisions. This project tackles the regression problem of forecasting future stock prices using AI-driven time series models. With the availability of historical stock data, we aim to apply machine learning techniques to uncover temporal patterns and provide short-term predictive insights. This can significantly assist in algorithmic trading, risk assessment, and financial planning.

2. Project Objectives

- Predict future stock prices using historical market data.
- Apply time series forecasting models such as ARIMA, LSTM, and Facebook Prophet.
- Evaluate and compare model performance using MAE, RMSE, and R^2 Score.
- Handle seasonality, trends, and noise within the time series data.
- Visualize prediction accuracy to interpret model behavior and support decision-making.
- Build a model suitable for real-world application in trading or investment strategies.

3. Flowchart of the Project Workflow



4. Data Description

https://www.kaggle.com/datasets/mrsimple07/stock-pricepredictio

- Dataset Name: Stock Price History (e.g., Apple Inc. from Yahoo Finance)
- Source: Yahoo Finance API or Kaggle stock datasets.
- Type: Time-series (structured)
- Size: ~2,000 records (daily data over several years)
- Features: Date, Open, High, Low, Close, Volume
- Target Variable: 'Close' price (to predict future closing price)
- Nature: Static dataset for this project, but can be dynamically updated.

5. Data Preprocessing

Preprocessing:

- Handled missing values via forward fill (common in time series).
- Converted 'Date' column to datetime format and set as index.

- Removed outliers in volume or price spikes using rolling statistics.

  EDA:
- Plotted time series trends (closing prices over time).
- Conducted decomposition to observe trend, seasonality, and residual.
- Used ACF/PACF plots to check autocorrelation for ARIMA suitability.

  Feature Engineering:
- Created lag features (e.g., Close_t-1, Close_t-2).
- Created rolling mean and exponential moving averages.
- Split date features into Year, Month, and Weekday for seasonality patterns (for Prophet).

- Normalize or standardize features where required.
- Document and explain each transformation step clearly in code and markdown.]

6. Exploratory Data Analysis (EDA)

Preprocessing:

- Handled missing values via forward fill (common in time series).
- Converted 'Date' column to datetime format and set as index.
- Removed outliers in volume or price spikes using rolling statistics.

EDA:

- Plotted time series trends (closing prices over time).
- Conducted decomposition to observe trend, seasonality, and residual.
- Used ACF/PACF plots to check autocorrelation for ARIMA suitability.

7. Feature Engineering

- Created lag features (e.g., Close_t-1, Close_t-2).
- Created rolling mean and exponential moving averages.
- Split date features into Year, Month, and Weekday for seasonality patterns (for Prophet).

8. Model Building

- ARIMA: Classical statistical model for linear trends.
- LSTM (Long Short-Term Memory): Recurrent Neural Network for capturing long-term dependencies.
- Prophet: Facebook's tool for easy trend and seasonality modeling.

Evaluation Metrics:

- MAE (Mean Absolute Error)
- RMSE (Root Mean Squared Error)
- R^2 Score

LSTM typically showed lower RMSE on validation data, indicating better performance in capturing nonlinear patterns.

9. Visualization of Results & Model Insights •

Line plots: Actual vs. Predicted closing prices.

- ACF/PACF: Autocorrelation diagnostics.
- Residual plots: Model fit quality.
- Prophet components: Trend, weekly and yearly seasonality.
- LSTM loss curves: Training vs. validation loss over epochs.

10. Tools and Technologies Used

- Programming Language: Python
- IDE: Jupyter Notebook / Google
- Colab Libraries:
    - Data Handling: pandas, numpy
    - Visualization: matplotlib, seaborn, plotly
    - Modeling: statsmodels, tensorflow/keras, fbprophet, scikit-learn
- Version Control: Git & GitHub

11. Team Members and Contributions

S.No	NAME	ROLES	RESPONSIBILITY

1	**Sasivardhan S K**	**Leader**	**Data Collection, Data Cleaning**
2	**Joshua Prince S**	**Member**	**Visualization & Interpretation**
3	**Rubesh kumar S**	**Member**	**Exploratory Data Analysis (EDA), Feature Engineering**
4	**Mohit Sai Reddy**	**Member**	**Model Building, Model Evaluation**