Phase-2

Student Name: JAYANTH R

Register Number: 410723104028

Institution: Dhanalakshmi College of Engineering

Department: Computer Science and Engineering

Date of Submission: 07/05/2025

Github Repository Link: https://github.com/jai-07-git/Nm_jayanth_DS

---

1. Problem Statement:

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

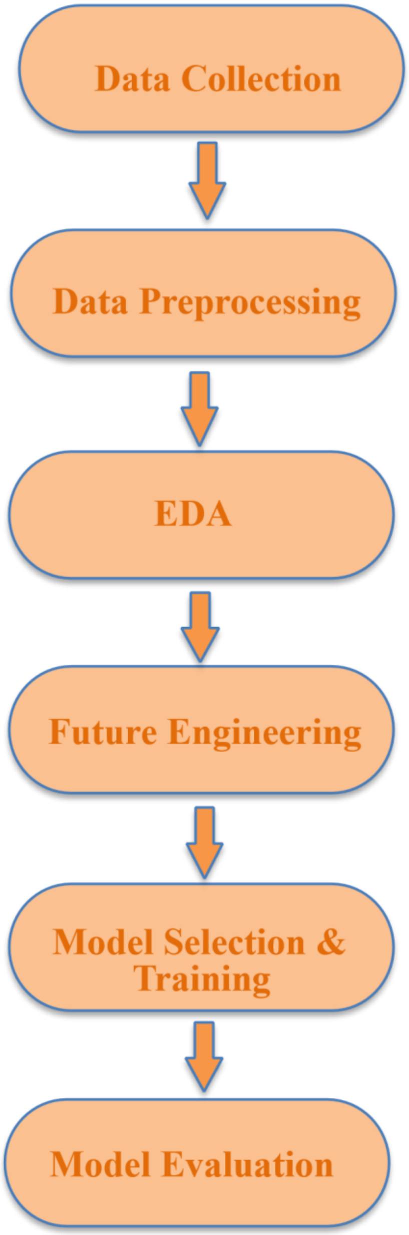
Stock market prices are highly volatile and influenced by numerous factors, making accurate prediction a complex task. With the rise of AI, especially deep learning models, there is an opportunity to analyze historical stock data and uncover patterns that can guide better investment decisions. This project aims to develop a predictive model using time series analysis to forecast stock prices effectively.

2. Project Objectives

1. Predict future stock prices using historical time series data.
2. Compare performance across traditional and deep learning models (e.g., ARIMA, LSTM).
3. Evaluate models based on accuracy and forecasting ability.
4. Provide visual insights for better understanding of stock trends.

-

3. Flow chart



4. Data Description

[Amazon Stock Market Data (2015-2024) 📈]

1. Source: e.g., Yahoo Finance, Kaggle stock datasets
2. Example Dataset: Apple Inc. (AAPL) historical prices
3. Type: Time series (Date, Open, High, Low, Close, Volume)
4. Target Variable: Close price or future Close price

5. Data Preprocessing

```
ticker = 'AAPL' start_date
= '2015-01-01'
end_date = '2024-12-31'

df = yf.download(ticker, start=start_date, end=end_date)
df.reset_index(inplace=True)  # Make 'Date' a column instead of index

print("Missing values:\n", df.isnull().sum())

df.dropna(inplace=True)

df['MA_5'] = df['Close'].rolling(window=5).mean()
df['MA_20'] = df['Close'].rolling(window=20).mean()

df['Lag_1'] = df['Close'].shift(1) df['Lag_2']
= df['Close'].shift(2)

df['Daily_Return'] = df['Close'].pct_change()

df['Volatility_5'] = df['Close'].rolling(window=5).std()

df.dropna(inplace=True)
```

6. Exploratory Data Analysis (EDA)

1. Plot trends, moving averages, and volatility
2. Autocorrelation & Partial Autocorrelation analysis
3. Seasonal decomposition if applicable

7. Feature Engineering

Lag features (Close t-1, t-2, ...)

Rolling window features

- Technical indicators (MACD, Bollinger Bands)

8. Model Building

Models Used:

- ARIMA: For baseline statistical prediction
- LSTM (Long Short-Term Memory): Deep learning model tailored for sequences
- Prophet (optional): Facebook’s library for time series forecasting

Evaluation Metrics:

- Mean Squared Error (MSE), RMSE, MAE
- Directional accuracy (did the model predict the trend correctly?)

9. Visualization of Results & Model Insights

1. Predicted vs Actual plot
2. Residual analysis
3. Highlight patterns captured well (e.g., trend reversal, volatility)

10. Tools and Technologies Used

1. Python Libraries: pandas, numpy, matplotlib, seaborn, scikit-learn, TensorFlow/Keras, statsmodels, yfinance, fbprophet
2. IDE: Google Colab, Jupyter Notebook
3. Version Control: GitHub

11. Team Members and Contributions

S.NO	NAMES	ROLES	RESPONSIBILITY
1	Mathesh S	Leader	Data Collection
2	Dhanajayan S	Member	Data Cleaning and Feature Engineering
3	Manoj C	Member	Visualization and Interpretation
4	Emaya Bharath	Member	Exploratory Data Analysis
5	Jayanth R	Member	Model Building and Model Evaluation