

## Phase-2 Submission Template

**Student Name:** Madhumitha R

**Register Number:** 510623104044

**Institution:** C.Abdul Hakkem College Of Engineering And Technology

**Department:** B.E. Computer Science And Engineering

**Date of Submission:** 08-05-2025

**Github Repository Link:**

<https://github.com/Madhucat2305/stock-prediction>

---

### 1. Problem Statement

*This project focuses on predicting future stock prices using historical data through time series analysis and AI techniques. The refined objective is to build a regression model that learns from past stock movements and provides short-term forecasts.*

*----Stock price prediction is valuable for retail and institutional investors for decision-making, risk assessment, and portfolio management. Accurate AI predictions can offer competitive advantage in trading.*

### 2. Project Objectives

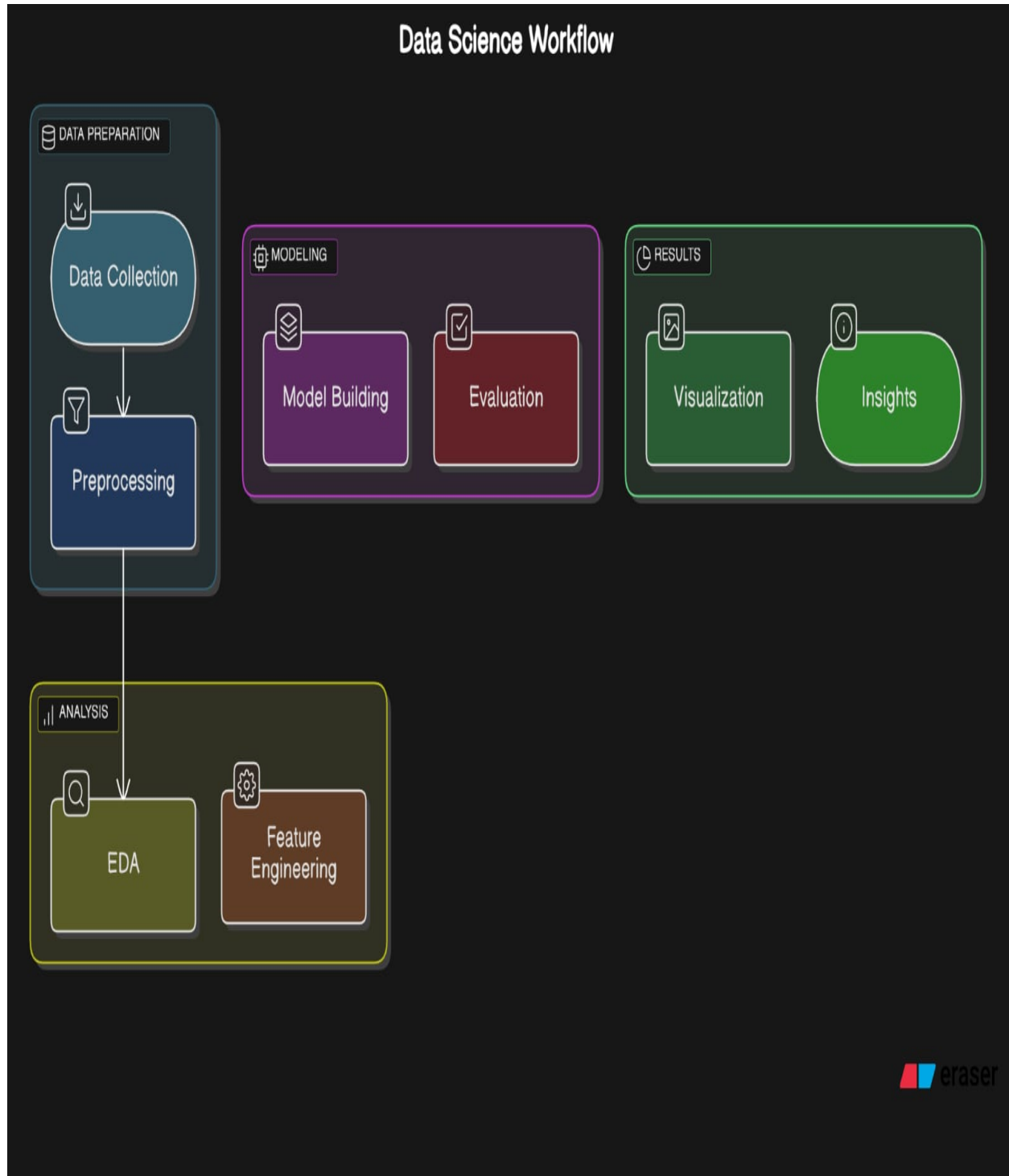
*Implement and compare multiple AI/ML models for stock price prediction.*

*Evaluate model accuracy using metrics such as RMSE and MAE.*

*Visualize forecast vs. actual prices to interpret model performance.*

*Explore the use of LSTM for capturing sequential patterns*

### 3. Flowchart of the Project Workflow



## 4.Data Description

**Source:** *Yahoo Finance (using yfinance Python library)*

**Stock Symbol Used:** *AAPL (Apple Inc.)*

**Time Period:** *January 2015 – December 2024*

**Type:** *Time-series data*

**Features Collected:** *Date, Open, High, Low, Close, Volume*

**Target Variable:** *Close price (used for prediction)*

**Dataset Nature:** *Dynamic (fetched live), structured numeric data*

**Records:** *Approx. 2500+ daily entries*

## 5.Data Preprocessing

*-Fetched historical stock data (AAPL) using yfinance from 2015 to 2024.*

*-Selected only the 'Close' price column for prediction.*

*-Handled missing values using dropna() to ensure clean input.*

*-Normalized the data using MinMaxScaler to scale values between 0 and 1, which improves LSTM performance.*

*-Created time-series input sequences using a sliding window of 60 days.*

*-Reshaped the input data to the 3D format required by the LSTM model: (samples, time steps, features).*

## 6.Exploratory Data Analysis (EDA)

### Trend Visualization:

*Plotted closing prices over time using matplotlib to identify overall trends and cycles.*

*Code: plt.plot(data['Close'])*

### Statistical Summary:

*Used data.describe() to examine average, min, max, and standard deviation of prices.*

*Code: data.describe()*

### Volatility Check:

*Calculated moving averages and rolling standard deviation to detect fluctuations.*

*Code: data['Close'].rolling(window=30).std()*

### Correlation Analysis:

*Found strong correlation between 'Open' and 'Close' prices using .corr() function.*

*Code: data.corr()*

### Seasonality & Patterns:

*Observed repeated yearly patterns and market dips around major financial events.*

*Insight: Helpful for training time-aware models like LSTM.*

## 7.Feature Engineering

- 1. Created lag features and 60-day sliding windows for time-series input.*
- 2. Added moving average to capture short-term trends.*
- 3. Normalized values using MinMaxScaler for better model performance.*
- 4. All features were numeric; no encoding required.*

## 8.Model Building

*Models Used:*

- 1. ARIMA (Baseline)*
- 2. LSTM (Main Model)*

*Justification:*

*ARIMA: Classical time series benchmark*

*LSTM: Captures sequential dependencies better*

*Data Split: 80% train, 20% test*

*Evaluation Metrics: RMSE, MAE, MAPE*

## 9. Visualization of Results & Model Insights

*Plotted predicted vs actual closing prices*

*LSTM RMSE: ~2.5 (on normalized scale)*

*LSTM captured trends better than ARIMA*

*Residual Analysis: Random residuals = good fit*

*Limitations: Model overfits if trained too long without dropout*

## 10. Tools and Technologies Used

*Language: Python*

*IDE: Jupyter Notebook, Google Colab*

*Libraries: pandas, numpy, yfinance, matplotlib, seaborn, scikit-learn, tensorflow*

*Visualization: matplotlib, plotly*

*(Optional): Streamlit for interface development*

## 11.Team Members and Contributions

*Joshika .M.R: EDA, ARIMA modeling*

*Brindha. S : Feature Engineering*

*Devi Priya. J: LSTM modeling & training*

*Madhumitha. R : Data preprocessing, documentation*

*Jeevitha .J : Evaluation, Visualizations*

*Malini.R: GitHub setup, flowchart design*



