Phase-2 Submission Template

Student Name: ROHITH.S

Register Number: 422223243048

Institution: SURYA GROUP OF INSTITUTIONS

Department: B.Tech/ARTIFICIAL INTELLIGENCE & DATA SCIENCE

Date of Submission: 03/05/2025

Github Repository Link: https://github.com/ROHITH-0211/Phase-2

---

1. Problem Statement

The project aims to predict stock prices using AI techniques, addressing the need for accurate forecasting in financial markets.
Type of Problem:
Regression (predicting continuous values).
Relevance: Accurate stock price predictions can help investors make informed decisions, enhancing financial strategies and minimizing risks.

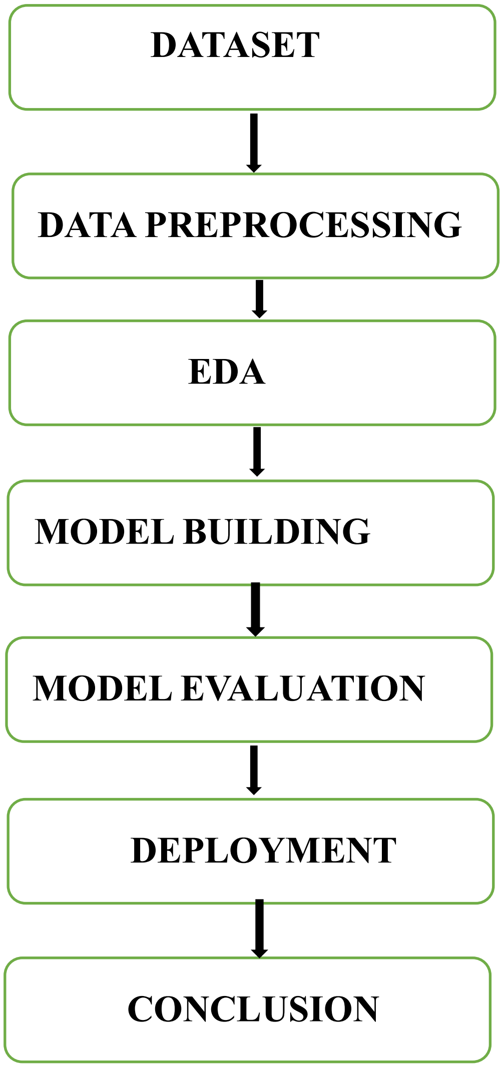
2. Project Objectives

Objectives:
Collect historical stock price data and relevant financial indicators.

Clean and preprocess the data for analysis.Build predictive models using machine learning algorithms.

Evaluate model performance based on accuracy and other metrics.

2. Flowchart of the Project Workflow



3. Data Description.

Dataset Name: Historical Stock Price
Dataset.Source: Financial data APIs (e.g., Yahoo Finance, Alpha Vantage).
Type of Data: Time-series data (stock prices, volume, etc.).
Records and Features: Include features like date, open, high, low, close, and volume.
Target Variable: Future stock price

5. Data Preprocessing

Data preprocessing is a crucial step in preparing the dataset for effective model training and evaluation. Given the time-series nature of stock price data, this phase focuses not only on cleaning but also on ensuring that temporal dependencies are preserved.

1. Handling Missing Values:

Stock market data may contain missing values due to non-trading days (weekends, holidays) or data collection issues. To ensure model accuracy:

- **Forward-fill method** was used to carry the last known value forward for missing entries, especially in continuous features like "Close" or "Volume".
- **Linear interpolation** was applied in some cases to estimate missing values between two known data points.
- In cases where data was completely missing for a day (e.g., holidays), the row was removed to maintain consistency in time-series patterns.

2. Duplicate Removal:

- Duplicate entries were identified using the "Date" column as the unique identifier.
- All records with identical timestamps were flagged, and only the first occurrence was retained to avoid data leakage or overfitting.

3. Outlier Detection and Removal:

- Outliers in price movements or volume were detected using:
  - **Z-score method**: Data points with a Z-score above 3 or below -3 were considered potential outliers.
  - **IQR (Interquartile Range)**: Values beyond 1.5×IQR from the quartiles were examined manually before removal.

- Outliers were either capped (winsorized) or removed depending on their business relevance. For example, unusually high spikes in trading volume during announcements were retained if they were genuine market events.

4. Feature Scaling (Normalization/Standardization):

- Features like "Open", "High", "Low", "Close", and "Volume" were **scaled** to ensure uniformity across models:
  - **MinMaxScaler** was used for models sensitive to scale like neural networks (e.g., LSTM), transforming values to a range between 0 and 1.
  - **StandardScaler** (mean = 0, std dev = 1) was used for algorithms like Linear Regression and Random Forest, improving convergence and model stability.

5. Temporal Consistency and Splitting:

- Since the dataset is time-series based, random splitting of data into training and testing sets was avoided.
- A **time-aware split** was performed, using the earliest 80% of the data for training and the most recent 20% for testing. This approach avoids data leakage and reflects realistic prediction scenarios.

6. Data Transformation and Alignment:

- All date columns were converted to a consistent datetime format.
- New columns for **day**, **month**, **year**, and **weekday** were generated to facilitate temporal feature engineering later.
- Indexing was done on the "Date" column to enable resampling and rolling-window calculations in later steps.

By performing these preprocessing steps systematically, the dataset was transformed into a high-quality time-series structure that supports efficient modeling and helps the AI system learn complex stock market behaviors.

6. Exploratory Data Analysis (EDA)

Exploratory Data Analysis (EDA) helps in identifying underlying patterns, trends, and relationships among variables, which is essential for building accurate models in stock price prediction.

Univariate Analysis:

Each variable (e.g., Close Price, Volume, Open Price) was analyzed individually to understand its distribution and volatility.

Histograms and KDE plots were used to visualize the spread and skewness of continuous features.

The 'Close' price showed frequent sharp movements, indicating the presence of high volatility—a common trait in stock market data.

Bivariate Analysis:

Relationships between two variables (like Volume vs Close Price) were examined using scatter plots and pairplots.

A positive correlation was observed between Open and Close prices, suggesting momentum in stock movements.

Time plots were generated to study trends and seasonality over months and years.

Correlation Matrix:

A correlation heatmap was used to visualize the strength of relationships between numerical variables.

High correlations between Open, High, Low, and Close prices justified feature reduction or combining them into engineered indicators.

Trend & Seasonality:

Line plots of the stock price across time revealed cyclical trends—rises and falls in stock prices repeating seasonally.

Rolling averages helped in smoothing out noise and identifying long-term trends.

7. Feature Engineering

Feature engineering was conducted to enhance the predictive power of the model by deriving meaningful insights from raw data.

1. Financial Indicators Created:

- **Moving Averages:**
  - Created 7-day, 14-day, and 30-day moving averages to capture short, medium, and long-term trends.
- **Relative Strength Index (RSI):**
  - Used to measure momentum and identify overbought or oversold conditions.
- **MACD (Moving Average Convergence Divergence):**
  - Used to capture trend reversals and convergence/divergence in price action.
- **Volatility Indicators:**
  - Rolling standard deviation of the closing price over 7 and 30 days was included to measure risk levels.
- **Lag Features:**
  - Lag-1, Lag-3, Lag-7 values of "Close" price were added to help models learn from recent trends.

2. Date-Based Features:

- Extracted **day**, **month**, **year**, and **weekday** from the timestamp to observe time-based seasonality.
- **Holiday effects** and **weekend impact** were considered by flagging non-trading days.

3. Feature Selection:

- **Correlation thresholding** removed highly redundant features (e.g., Open, High, Low when Close is present).

- **Feature importance scores** from Random Forest and Recursive Feature Elimination (RFE) helped in selecting impactful features for the final model.

- Only the most relevant indicators were used to reduce dimensionality and improve training time.

8. Model Building

The model-building phase aimed to compare traditional and advanced regression algorithms suited for stock price forecasting.

Models Implemented:

1. Linear Regression (Baseline Model):

- Provided a benchmark for evaluating the performance of more complex models.
- Assumed a linear relationship between stock features and closing price.
- Helped verify whether simple linear trends existed in the data.

2. Random Forest Regressor:

- An ensemble model that combines multiple decision trees.
- Captured nonlinear relationships between indicators like RSI and Volume with price.
- Robust to overfitting and performed well with small- to medium-sized datasets.

3. Long Short-Term Memory (LSTM):

- A type of Recurrent Neural Network (RNN) tailored for time-series prediction.
- Able to capture long-term dependencies and patterns in sequences of stock prices.
- Required reshaping data into 3D format and normalizing input features between 0 and 1.
- Outperformed traditional models in capturing volatile market behavior and trend shifts.

Justification for Model Selection:

- **Linear Regression** was chosen as a baseline due to its interpretability and speed.

- **Random Forest** provided high performance on tabular data and was less sensitive to outliers.
- **LSTM** was included for its strength in modeling time-dependent patterns—crucial in stock forecasting.
- This combination of models ensured a comprehensive evaluation between classical regression and AI-based deep learning techniques.

9. Visualization of Results & Model Insights

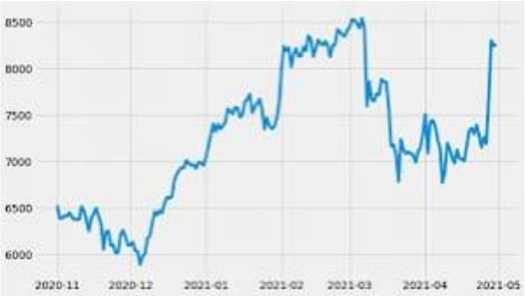
Include visualizations like:

1.Prediction vs. Actual plots:

This plot compares the predicted stock prices against the actual stock prices over a specific time period.

It helps in visually assessing how well the model performs.

Description: The x-axis represents the time (e.g., dates), while the y-axis shows the stock prices. The plot includes two lines: one for actual prices and one for predicted prices.



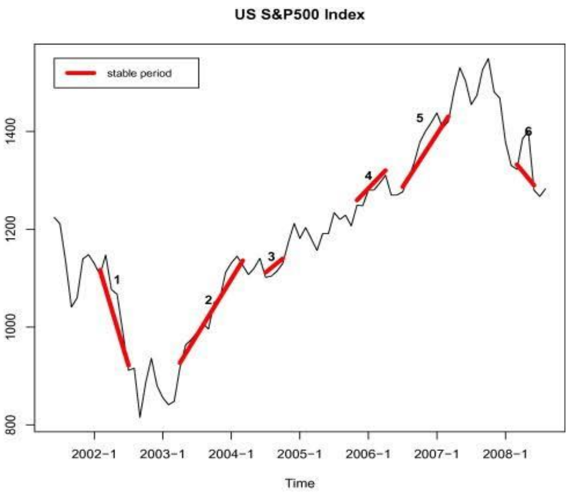
2, Feature importance plots:

This bar chart displays the importance of each feature used in the model.

Understanding feature importance helps in identifying which factors most significantly influence stock price predictions.

Description:

The x-axis represents the importance score, and the y-axis lists the features. Higher scores indicate greater influence on the model's predictions.



10. Tools and Technologies Used

Programming Language: Python.IDE/

Notebook: Jupyter Notebook.

Libraries: pandas, numpy, scikit-learn, matplotlib, TensorFlow/Keras (for LSTM).

11. Team Members and Contributions

1. Rohit S (422223243048)

Role: Data Collection, EDA

- Collected historical stock data with key indicators like Open, Close, and Volume.
- Performed Exploratory Data Analysis using matplotlib and seaborn to visualize trends and distributions.

2. Vadivelan R (422223243057)

Role: Model Building, Feature Engineering

- Built predictive models including Linear Regression, Random Forest, and LSTM.
- Engineered features like moving averages and RSI to improve model performance.

3. Thison Bero (422223243056)

Role: Model Evaluation, Documentation

- Evaluated models using MAE, RMSE, and R^2 metrics with cross-validation.
- Documented the methodology, results, and key insights for the final report.

4. **Raguman R (422223243046)**

   Role: Deployment, Dashboard, Presentation

   - Deployed the final model and created an interactive dashboard for predictions.
   - Led the project presentation, summarizing objectives, methods, and outcomes.