Phase-2 Submission Template

Student Name: Noorul Akthar A

Register Number: 2303617710421304

Institution: Government college of engineering Salem

Department: Computer science and engineering

Date of Submission: 10-05-2025

Github Repository Link: [Update the project source code to your Github Repository]

---

1. Problem Statement

- The stock market is highly volatile and difficult to predict using traditional methods.
- Investors often struggle with making informed decisions due to unpredictable price fluctuations.
- Traditional statistical methods fail to capture nonlinear patterns in financial data.

2. Project Objectives

- To predict future stock prices using AI models trained on historical market data.
- To implement and compare multiple time series forecasting models, such as ARIMA, LSTM, and Facebook Prophet.
- To analyse historical stock trends for patterns, seasonality, and anomalies.
- To provide a data-driven decision support system for investors and traders.

3. Flowchart of the Project Workflow

Start

↓

Collect and Preprocess Data

↓

Apply NLP Techniques

↓

Feature Extraction and Selection

↓

Train Fake News Detection Model

↓

Evaluate Model Performance

↓

Deploy System for Real-World Use

↓

Monitor and Improve Accuracy

↓

End

4. Data Description

- **Dataset name and origin:** Stock Market Historical Data

- **Type of data: -** Numerical Time Series Data

- **Number of records: -** hundreds of thousands of records depending on the stock and the time period considered (e.g., for one stock over 5 years, there could be ~1,200+ records for daily data).

- **Features:** -

  1. Date
  - Description: The date of stock data entry. This serves as the time index for the time series model.
  - Type: Datetime

  2. Open
  - Description: The price at which the stock opened on the given trading day.
  - Type: Numerical (float)

  3. High
  - Description: The highest price reached during the trading day.

- **Target variable: - -** Close or Adjacent Close:

  The target variable for prediction is typically the Close price (or Adjacent Close) because it represents the stock's final trading value at the end of each day. This is the value investors care most about when evaluating stock performance.

5. Data Preprocessing

- **Handle missing values:**
  Missing values can occur due to incomplete or irregular data collection.
  Method: Use techniques like:
  Forward Fill: Use the last known valid value to fill missing data.
  Backward Fill: Use the next valid value to fill the missing data.
  Interpolation: Estimate missing values based on surrounding data points.
  Dropping rows: If missing values are minimal, rows with missing data.

- **Remove or justify duplicate records:**
  Duplicate entries can occur if data is collected multiple times for the same date or stock.
  Method: Identify and remove duplicate records.

- **Detect and treat outliers:**
  Outliers can skew the data and affect model performance.
  Method:
  Statistical Method: Use z-scores or IQR (Interquartile Range) to detect outliers.
  Visualization: Box plots, histograms, or scatter plots can help detect outliers visually.

- **Convert data types and ensure consistency:**
  Ensure that each column has the correct data type for analysis.
  Method: Convert columns into the appropriate types:
  Date column to datetime format for proper time series handling.
  Volume, Open, High, Low, Close, Adjacent Close to numeric types.

- **Encode categorical variables:**
  In stock price data, we may not have direct categorical columns. However, for modeling purposes, you might include sector or company names (in case of multi-stock predictions).

Method: Use Label Encoding or One-Hot Encoding to convert categorical data into numeric format.

- **Normalize or standardize features where required:**

  Text Preprocessing (if applicable):

  In some cases, you may want to analyse news headlines or financial reports alongside stock price data.

  Method: Perform text preprocessing for news data:

  Tokenization: Split text into words.

  Stop-word Removal: Remove common words (like “the”, “is”).

  Lowercasing: Convert text to lowercase for uniformity.

  Stemming/Lemmatization: Reduce words to their base form (e.g., "running" → "run").

6. Exploratory Data Analysis (EDA)

- **Univariate Analysis:**
  Univariate analysis examines each feature individually to understand its distribution, central tendency, and variance.

- **Bivariate/Multivariate Analysis*:***

  Bivariate and multivariate analyses explore relationships between two or more variables to uncover patterns, correlations, or trends.

  Correlation analysis:
  Strong positive correlation between Open and Close prices, which makes sense because stock prices are closely tied to each other at the start and end of the trading day.
  Low correlation between Volume and stock prices may indicate that trading volume doesn’t always affect price movement directly.

Scatter Plots:
The relationship between High and Low prices will show whether there are large price fluctuations on particular days.

- **Insights Summary:**
  After performing the univariate and bivariate/multivariate analyses, the following insights can be derived:

  If the distribution is skewed, consider applying transformations (e.g., log transformations) to normalize the data.
  The Volume of shares traded may have significant spikes on certain days (e.g., earnings reports, stock splits), which can be identified via boxplots and histograms.

7. Feature Engineering

Text Length:
Purpose: This feature can be useful when analysing textual data, such as news headlines, financial reports, or social media posts, to understand how the length of text relates to stock price movement.
Example: Longer articles might indicate in-depth reports or important events, potentially affecting the stock price. Combine or split columns (e.g., extracting date parts).

Text Vectorization: Purpose: Text vectorization is essential when working with textual data like news articles, reports, or social media content. The goal is to convert text into numerical representations that can be used as input to machine learning models.

Methods:

Bag of Words (Bow): Represent text as a collection of words, ignoring grammar and word order but keeping multiplicity

TF-IDF (Term Frequency-Inverse Document Frequency): Weighs words based on their frequency in a document relative to their frequency in the entire dataset, helping emphasize important words.

8. Model Building

- **Models Selected:**

When building predictive models for stock price prediction, it's crucial to select models that can handle time series data effectively, as stock prices are sequential and exhibit patterns like trends and seasonality.

- **Models Considered:**

1. Linear Regression:

- Justification: A simple and interpretable model that can establish a linear relationship between stock price and features (e.g., volume, previous prices).

- Usage: Predicts a continuous value (stock price) based on input features.

2. Decision Trees:

- Justification: Can capture non-linear relationships between features. They provide a clear decision-making path and are easy to interpret.

- Usage: Predicts stock prices based on decision rules created from features.

3. Random Forest:

Justification: An ensemble method that improves over decision trees by averaging multiple trees to reduce variance and prevent overfitting.

Usage: Helps make more robust predictions by combining many decision trees for better generalization.

9. Visualization of Results & Model Insights

The goal of this project is to use AI-driven models for stock price prediction, leveraging historical stock data, financial reports, and news articles. By applying time series analysis and machine learning algorithms, we aim to predict future stock movements with high accuracy.

Key Takeaways:

Time series models like LSTM and XGBoost perform well with sequential data and can capture complex relationships in stock prices.

Feature engineering (e.g., text vectorization, sentiment analysis) and proper data preprocessing play a significant role in model performance.

The evaluation metrics like MAE, RMSE, R^2, and AUC are crucial in assessing model effectiveness.

10. Tools and Technologies Used

- **Programming Language:** Python
- **IDE/Notebook:** Google Collab, Jupiter Notebook, VS Code, etc
- **Libraries:** pandas, NumPy, seaborn, matplotlib, scikit-learn, XG Boost, etc.

11. Team Members and Contributions:

- Data cleaning- Keerthi Roshan B
- Model development – Noorul Akthar A
- Feature engineering – Santhosh SV
- Documentation and reporting – Ragul K