**Phase-2 Submission Template**

**Student Name:** YUVARAJ.C

**Register Number:** 422223106041

**Institution:** Surya Group of institutions

**Department:** B.E/ECE.2nd year

**Date of Submission:** [8/5/2025]

**Github Repository Link:** [https://github.com/Yuvaraj2101/Cracking-the-Market-Code-AI-Driven-Stock-Price-Prediction-Using-Time-Series-Analysis.git]

### **1. Problem Statement**

* The stock market is highly volatile, making price prediction a complex task.
* Conventional statistical models struggle with nonlinear trends and noise.
* This project aims to use AI, particularly deep learning, to forecast stock prices.
* The focus is on time series data, where historical prices influence future values.
* This is a regression problem, as we aim to predict continuous values.
* Accurate predictions can help investors make informed decisions.
* We use LSTM networks due to their effectiveness in handling sequential data.
* Solving this problem can improve automated trading systems and risk management.
* AI can offer insights beyond human analysis, especially with large data.
* This work contributes to smarter, data-driven investment strategies.

### **2. Project Objectives**

1. To predict future stock prices using AI-driven time series models.
2. Leverage LSTM networks to learn from past stock trends.
3. Achieve high prediction accuracy using historical and technical indicators.
4. Improve model interpretability with visual analytics and feature analysis.
5. Compare AI models with traditional time series methods like ARIMA.
6. Ensure the model generalizes well across different stock datasets.
7. Evaluate model performance using RMSE and R² metrics.
8. Deploy the model for real-time or batch prediction use cases.
9. Explore feature importance to understand market influencers.
10. Refine goals based on EDA findings and model outcomes.

### **3. Flowchart of the Project Workflow**



**2. Data Preprocessing – Clean and prepare time series**

**3. EDA – Visualize trends and correlations**

**4. Feature Engineering – Add technical indicators**

**5. Model Selection – LSTM, ARIMA, Random Forest**

**6. Training and Validation – Split and fit models**

**7. Evaluation – Use RMSE, MAE, R²**

### **4. Data Description**

### ***Dataset source****: Yahoo Finance (or Kaggle API-based dataset).*

### ***Data includes:*** *Date, Open, High, Low, Close, Volume.*

### ***Data type:*** *Time-series, structured format.*

### ***Size: Approx****. 2500–5000 rows depending on stock/time range.*

### ***Target variable:*** *‘Close’ price for next day prediction.*

### *Data spans over multiple years for trend analysis.*

### ***Frequency:*** *Daily stock prices.*

### *Static dataset but can be updated via API.*

### ***Additional features****: moving averages, RSI, MACD.*

### *Used primarily for supervised learning tasks.[*

### **5. Data Preprocessing**

* *Removed missing or null values using forward fill.*
* *Handled date-time formats and indexed by date.*
* *Removed duplicate records and ensured chronological order.*
* *Generated technical indicators like 5-day MA, 20-day MA, RSI.*
* *Normalized continuous features to scale input for LSTM.*
* *Encoded categorical data if any (e.g., stock sector).*
* *Removed extreme outliers using IQR or z-score method.*
* *Created lag features for time dependency.*
* *Ensured no data leakage during train-test split.*
* *Saved clean dataset for modeling phase*

### **6. Exploratory Data Analysis (EDA)**

*Univariate analysis showed seasonality in price changes.*

*Line plots revealed bullish and bearish trends over time.*

*High correlation found between opening and closing prices.*

*Volume spikes aligned with major price changes.*

*Boxplots identified price outliers during market crashes.*

*RSI and MACD indicated momentum shifts in trends.*

*Moving averages smoothed out short-term fluctuations.*

*Heatmap showed strong feature correlation.*

*EDA revealed key factors impacting future prices.*

*Insights used to select features for the model.*

### **7. Feature Engineering**

* *Created lag-based features for 1, 5, and 10-day delays.*
* *Added technical indicators: MA, EMA, RSI, Bollinger Bands.*
* *Encoded time elements: day of week, month (for trend detection).*
* *Removed redundant or highly correlated features.*
* *Generated target feature: next day closing price.*
* *Applied log transformation to reduce skewness.*
* *Grouped data into windows for LSTM input format.*
* *Tested polynomial features but limited their use due to overfitting.*
* *Standardized all features before feeding into the model.*
* *Final dataset optimized for temporal learning.*

### **8. Model Building**

### Used LSTM for sequential modeling of stock data.

1. Compared with Random Forest Regressor and ARIMA.
2. LSTM captured long-term dependencies better than others.
3. Data split into 70% train and 30% test using time-based splitting.
4. LSTM trained with multiple epochs and tuned hyperparameters.
5. Evaluated models using RMSE, MAE, and R².
6. LSTM achieved lowest RMSE (~2.5 on test set).
7. Random Forest lagged in trend generalization.
8. ARIMA underperformed on nonlinear fluctuations.

10.LSTM selected as final model for prediction.

### **9. Visualization of Results & Model Insights**

### Plotted actual vs predicted stock prices for visual comparison.

LSTM predictions closely tracked real price movement.

Residual plots showed minimal bias and variance.

Feature importance from Random Forest showed RSI, Volume as key.

Line graphs illustrated prediction error margins.

Time series decomposition highlighted trend, seasonality, noise.

Used rolling forecast plots to visualize future estimates.

Comparative bar chart showed LSTM outperformed baselines.

Added confidence intervals to show prediction range.

Visuals aided interpretation of model strengths.

### **10. Tools and Technologies Used**

**Programming Language: Python**

**IDE: Google Colab & Jupyter Notebook**

**Libraries: pandas, numpy, matplotlib, seaborn, sklearn, keras, tensorflow**

**Time Series Tools: statsmodels, fbprophet (optional)**

**Visualization: matplotlib, seaborn, Plotly**

**Version Control: GitHub**

**Dataset Source: Yahoo Finance API / Kaggle**

**Model Deployment (optional): Flask / Streamlit**

**Documentation: MS Word / Markdown**

**Others: XGBoost (for comparison if needed)**

### **11. Team Members and Contributions**

* + *[Yuvaraj.C]: Data Collection & EDA*
  + *[Santhosh kumar.M] Feature Engineering & ARIMA Modeling*
  + *[Sanjai.k]: LSTM Model & Evaluation*
  + *[Jyothi.R]: Visualizations & Report WritingMember*