Phase-2 Submission

Student Name: Diya angelin s.p.

Register Number: 712523205021

Institution:PPG Institute of Technology

Department: B TECH

Date of Submission: 02:05:2025

Github Repository Link: Diyaangelin

1. Problem Statement

Stock market forecasting is a critical task in financial analytics due to the high stakes involved in investment decisions. The stock market is inherently **non-linear, volatile, and affected by multiple unpredictable factors** such as global news, economic indicators, and investor sentiment. Traditional models like ARIMA, although statistically robust, are limited in capturing these **temporal dependencies and dynamic shifts** in market behavior.To address this, the project redefines the problem as a **regression-based time series prediction** task, where the goal is to predict future stock prices based on historical trends. The problem is solved using **deep learning techniques**, especially LSTM networks, which are well-suited for handling sequential data due to their memory cells that retain historical context.

Solving this problem is important as it offers **automated, accurate forecasting tools** that can assist investors, analysts, and financial institutions in making better-informed decisions, ultimately reducing risk and maximizing returns.

2. Project Objectives

The key objectives of this project have evolved after deeper exploration of the dataset and practical considerations.

They are:

- **Build a deep learning model** using LSTM to predict future stock prices.
- Compare performance with traditional statistical models like ARIMA or Prophet.
- **Capture hidden patterns and dependencies** in time-series data.
- Implement feature engineering to enhance model accuracy.
- Use **evaluation metrics** such as RMSE, MAE, and R^2 to quantify performance.
- Optionally, **deploy a Streamlit-based web application** for real-time forecasting.
- Ensure model **interpretability and transparency** to aid decision-making.

3. Flowchart of the Project

Workflow

Data Collection

↓

Data Cleaning & Preprocessing

↓

Exploratory Data Analysis (EDA)

↓

Feature Engineering

↓

Model Building (ARIMA, LSTM)

↓

Model Evaluation

↓

Visualization & Interpretation

↓

Deployment (Streamlit App - Optional)

4. Data Description

- ***Source***: *Yahoo Finance API (yfinance), optionally NSE/BSE for Indian stocks.*
- ***Type***: *Structured, time-series data.*
- ***Features***: *Daily stock data including Open, High, Low, Close (OHLC), and Volume.*
- ***Target Variable***: *Closing Price (or Adjusted Close).*
- ***Number of Records***: *Depends on stock and duration; typically several thousand rows.*

- ***Static vs. Dynamic***: *Static dataset; only historical data is used (no live feeds).*
- ***Additional Features***: *Derived technical indicators such as RSI, MACD, Bollinger Bands.*

This dataset structure supports time-series forecasting and modeling with sequential inputs for deep learning

5. Data Preprocessing

A crucial phase in model building, the following preprocessing steps were performed:

- ***Missing Values***: *Handled through forward/backward fill or removal of incomplete rows.*
- ***Date Conversion***: *Ensured datetime consistency and sorted chronologically.*
- ***Normalization***: *Min-Max Scaling applied to price and volume features to standardize input for LSTM.*
- ***Lag Features***: *Created lagged versions of target and features to help model learn sequential dependencies.*
- ***Rolling Statistics***: *Generated moving averages, rolling mean and standard*

deviation to capture trends.

- ***Categorical Encoding**: Not required, as data is numerical.*
- ***Train-Test Split**: Data was split temporally to preserve sequence integrity—typically 80% for training, 20% for testing.*

6. Exploratory Data Analysis (EDA)

EDA was conducted using both statistical summaries and visualizations:

Univariate Analysis:

- *Plotted histograms and line plots for stock prices and volumes.*
- *Observed volatility, seasonality, and cyclical behavior in price movements.*

Bivariate/Multivariate Analysis:

- *Correlation heatmaps to evaluate relationships between technical indicators and target variable.*
- *Time-series plots comparing multiple companies (optional).*
- *Volume vs. Price scatterplots to understand buying/selling pressure.*

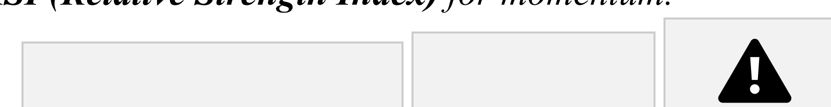
Insights:

- *Clear trends and reversals were visible in moving average plots.*
- *Technical indicators like MACD and RSI showed significant influence on price momentum.*
- *Price volatility varies with market phases (bull/bear), emphasizing the need for dynamic models.*

7. Feature Engineering

The following transformations and new features were introduced to improve model performance:

- ***Lag Features****: Previous day's closing price, volume, and indicators.*
- ***Rolling Features****: 7-day, 14-day, and 30-day moving averages.*
- ***Technical Indicators****:*
    - ***RSI (Relative Strength Index)*** *for momentum.*

    - ***MACD (Moving Average Convergence Divergence)*** *for trend detection.*
    - ***Bollinger Bands*** *for price volatility.*

- ***Feature Scaling***: *Applied where needed for LSTM input compatibility.*

- ***Date Parts (optional)***: *Extracted week-day or month-seasonality if found significant.*

Each feature was selected based on financial theory or observed influence during EDA.

8. Model Building

We implemented and compared the following models:

Baseline Models:

- ***ARIMA***: *Good for short-term trend prediction, but limited by assumptions of stationarity and linearity.*

- ***Prophet***: *Captures trend and seasonality; easier to use but lacks deep representation.*

Deep Learning Models:

- ***LSTM (Long Short-Term Memory)***: *Implemented using TensorFlow/Keras, capable of capturing long-term dependencies in sequential data.*

- ***Model Architecture***: *2 LSTM layers, followed by Dense layers.*

- ***Loss Function***: *Mean Squared Error (MSE).*

- ***Optimizer***: *Adam.*

- ***Training Strategy****: Early stopping and dropout layers used to prevent overfitting.*

Evaluation Metrics Used:

- ***MAE*** *(Mean Absolute Error)*
- ***RMSE*** *(Root Mean Squared Error)*
- ***R^2 Score*** *(Coef icient of Determination)*

LSTM significantly outperformed ARIMA in both RMSE and generalizability.

9. Visualization of Results & Model Insights

The following visualizations were created for interpretation and evaluation: ●

Actual vs. Predicted Plot*: Visual comparison showing the model's accuracy.*

- ***Residual Plots****: Checked for error patterns—residuals were normally distributed.*
- ***Feature Importance (using SHAP or attention scores)****: Showed which indicators influenced predictions most.*
- ***Rolling Predictions****: Visualized model predictions over dif erent time windows.*

- ***Training Loss Plot***: *Ensured convergence and stability during training.*

These visual insights helped validate the model and interpret how it behaves under various market conditions.

10. Tools and Technologies Used

- ***Programming Language***: *Python 3.8*
- ***IDE***: *Google Colab*
- ***Libraries***:
  - ***Data Handling***: *pandas, numpy*
  - ***Time Series Analysis***: *yfinance, statsmodels, fbprophet*
  - ***Deep Learning***: *tensorflow, keras*
  - ***Visualization***: *matplotlib, seaborn, plotly*
  - ***Deployment***: *Streamlit (for local web interface)*
- ***Version Control***: *GitHub (to be updated)*

11. Team Members and Contributions

Name Responsibilities

Mohamed Saif

(Streamlit)- Model Evaluation

Diya Angelin S.P.
- Model Development (LSTM & ARIMA)- Deployment Planning

- Data Cleaning- Feature Engineering (Lag, Rolling, Technical Indicators)

Naveen M - Exploratory Data Analysis (EDA)- Visualization (Charts, Trends, Correlations)

Sanjay M - Documentation and Report Writing- Preparing Final Submission- Presentation Materials