Ebpl-DS-Predicting air quality levels using advanced machine learning algorithms for environmental insights

Student Name: SHARMILA D

Register Number: 510623106048

Institution: C ABDUL HAKEEM ENGINEERING COLLEGE AND TECHNOLGY, MELVISHARAM

Department: ECE

Date of Submission: 09-05-2025

Github Repository Link: https://github.com/abinayakumaravel

1. Problem Statement

Accurately predicting stock prices is one of the most complex and highstakes challenges in finance. This project aims to develop a predictive model using time series analysis and AI algorithms to forecast stock prices based on historical trends and technical indicators. The challenge lies in capturing temporal dependencies, volatility, and market dynamics to produce reliable short-term predictions.

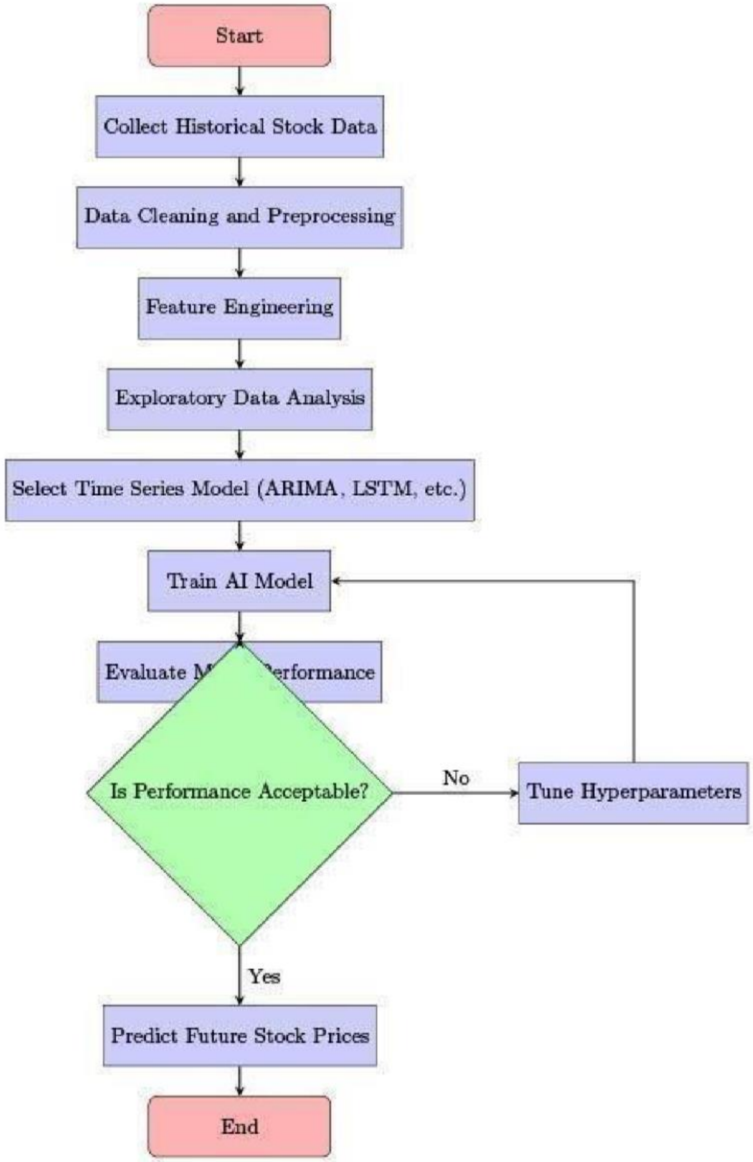
2. Project Objectives

- Develop a time series-based AI model for stock price forecasting.
- Analyze and visualize trends, seasonality, and volatility in historical stock data.
- Extract and engineer relevant technical indicators (e.g., RSI, MACD, moving averages).

- Evaluate multiple models, including ARIMA, LSTM, and Prophet.

- Deploy an interactive interface using Gradio for real-time stock forecasting.

3. Flowchart of the Project Workflow


Start
Collect Historical Stock Data
Data Cleaning and Preprocessing
Feature Engineering
Exploratory Data Analysis
Select Time Series Model (ARIMA, LSTM, etc.)
Train AI Model
Evaluate M erformance
Is Performance Acceptable?
No
Tune Hyperparameters
Yes
Predict Future Stock Prices
End

4. Data Description

- Dataset Name: Historical Stock Prices (e.g., Apple Inc. – AAPL)
- Source: Yahoo Finance / Alpha Vantage API
- Type of Data: Time series (numeric)
- Records and Features: ~2,000+ daily records with OHLCV data (Open, High, Low, Close, Volume)
- Target Variable: Next-day closing price (regression)
- Static or Dynamic: Dynamic dataset (can be updated regularly)
- Attributes Covered: Open, High, Low, Close, Volume, Date

5. Data Preprocessing

- *Fetched daily stock price data using yfinance API.*
- *Converted date columns to datetime format and set as index.*
- *Handled missing values with forward-fill method.*
- *Created lag features and technical indicators (e.g., moving averages).*
- *Normalized features using MinMaxScaler for neural network inputs.*

6. Exploratory Data Analysis (EDA)

- Univariate Analysis: Distribution plots for stock returns and volume.
- Time Series Plots: Price trend, moving average overlays, volatility bands.
- Correlation Heatmaps: Between indicators and closing price.
- Insights:
- Strong autocorrelation in short lags.
- High volume days often precede price swings.

7. Feature Engineering

- Generated rolling statistics: 7-day, 14-day, and 30-day moving averages.
- Derived technical indicators: RSI, MACD, Bollinger Bands • Created lag features to capture historical dependency
- Scaled features for LSTM using MinMaxScaler.

8. Model Building

- **Algorithms Used:**
  - ARIMA (AutoRegressive Integrated Moving Average) ○ Facebook Prophet (trend + seasonality) ○ LSTM (Long Short-Term Memory Neural Network)
- ***Model Selection Rationales :***
  - *ARIMA for baseline statistical modeling. ○ Prophet for capturing trend/seasonality. ○ LSTM for deep learning and long-term dependencies.*
  - *Train-Test Split: Last 20% of data for testing (timebased split)*
- **Evaluation Metrices:**
  - *MAE (Mean Absolute Error) ○ RMSE (Root Mean Squared Error) ○ MAPE (Mean Absolute Percentage Error)*

9. Visualization of Results & Model Insights

- *Forecast vs Actual Charts: Overlay plots of predicted vs actual stock prices.*
- *Residual Analysis: Plotted prediction errors over time.*
- *Feature Importance (for non-deep learning models): SHAP or permutation importance.*
- *Model Comparison: Table of RMSE, MAE across ARIMA, Prophet, and LSTM.*

10. Tools and Technologies Used

- **Programming Language**: Python

- **Environment**: Jupyter Notebook / Google Colab
- **Libraries**: ○ Pandas, numpy for data processing ○ Matplotlib, seaborn, plotly for visualization ○ Scikit-learn, statsmodels, fbprophet, keras/tensorflow for modeling ○ Yfinance for data extraction ○ Gradio for interface deployment

11. Collab Project Code

https://colab.research.google.com/drive/1bYUk7b64h59M-gB6ttqsVLmRcc-OPyFP#scrollTo=Wf5KrEb6vrkR

NAME	ROLE	**RESPONSIBILITIES**
ABINAYA K	Team lead	Oversee project Team lead development, coordinate team activities, ensure timely delivery of milestones, and contribute to documentation and final presentation.
NISHA S	Data collector	Collect data from APIs (e.g., Twitter), manage dataset storage, clean and preprocess text data, and ensure quality of input da
AFRINISHA BEGUM A	Model developer	Build sentiment and emotion classification models, perform feature engineering, and evaluate model performance using matrices
PRIYADARSHINI S	Data analyser	Conduct exploratory data analysis
SHARMILA D	Model developer	Build sentiment and Emotion Model classification models, perform feature Developer engineering, and evaluate model Performance using suitable metrics.

12. Team Members and Contributions