

GitHub Link : <https://github.com/skalaiyaran2005>

Project Title

Cracking the Market Code with AI-Driven Stock Price Prediction Using Time Series Analysis

• Problem Statement

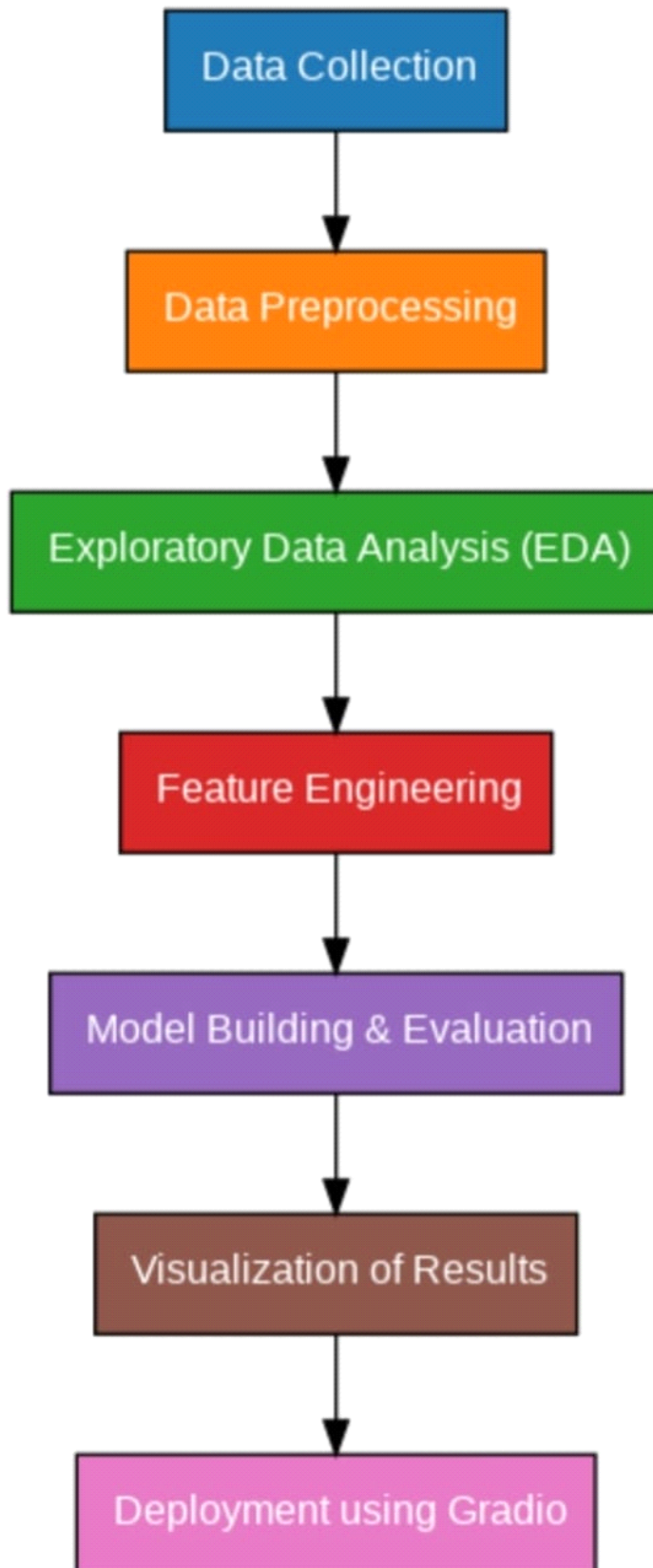
Forecasting stock prices is a complex and highly valuable challenge in financial markets due to their volatile, nonlinear, and dynamic nature. The aim of this project is to use historical time series data and AI-based models to predict future stock prices. Accurate forecasting models can support informed investment decisions, reduce risks, and potentially increase profitability for traders and analysts.

• Project Objectives

- Develop a machine learning model to accurately predict stock prices using time series data.
- Analyze historical price patterns and identify influential indicators.
- Compare different algorithms (ARIMA, LSTM, Prophet) for forecasting effectiveness.
- Visualize trends and predictions to enhance interpretability.
- Build an interactive interface (optional: Streamlit or Gradio) for real-time prediction demonstration.

• Flowchart of the Project Workflow

3. Flowchart of the Project Workflow



- **Data Description**

- Dataset Name: Stock Price Dataset
- Source: Kaggle
- Type of Data: Time Series
- Attributes: Date, Open, High, Low, Close, Volume
- Target Variable: Closing Price
- Nature of Dataset: Dynamic, real-world financial data
- Dataset Name: <https://www.kaggle.com/>
- dataset link : <https://www.kaggle.com/search?q=stock+price+prediction>

- **Data Preprocessing**

- Handled missing values in price columns.
- Converted date into datetime format and set as index.
- Performed log-scaling or differencing for stationarity (if using ARIMA).
- Normalized data (for LSTM model).
- Engineered features like moving averages and RSI.

- **Exploratory Data Analysis (EDA)**

- Line plots for price trends over time.
- Rolling mean and standard deviation for trend detection.
- Autocorrelation and Partial Autocorrelation plots.

- Volatility analysis using standard deviation and price range.

- **Feature Engineering**

- Created lag-based features (e.g., Close_t-1, Close_t-2).
- Derived technical indicators like:
 - 7-day and 21-day Moving Averages
 - Relative Strength Index (RSI)
 - MACD
- Differencing for stationarity in ARIMA.

- **Model Building**

- ARIMA Model for linear trends in stationary series.
- LSTM Model for capturing long-term dependencies and patterns.
- Facebook Prophet for trend + seasonality modeling.

- **Evaluation Metrics**

- RMSE (Root Mean Squared Error)
- MAE (Mean Absolute Error)
- MAPE (Mean Absolute Percentage Error)
- R² Score (for LSTM models)

- **Visualization of Results & Model Insights**

- Actual vs Predicted closing prices for test data.
- Residual plots for error analysis.
- Feature importance (if using models like XGBoost).
- Comparison plots of model performance.

- **Tools and Technologies Used**

- Language: Python
- IDE: Jupyter Notebook / Google Colab
- Libraries: pandas, numpy, matplotlib, seaborn, scikit-learn, keras, statsmodels, fbprophet
- Optional Interface: Streamlit or Gradio for user interaction

- **Team Members and Contributions**

- S. Kalaiyarasan
 - Data Collection and Preprocessing
 - EDA and Feature Engineering
- N. Manikandan
 - LSTM Model Development
 - ARIMA Implementation
 - Visualization
- S. Gokul

- Evaluation and Model Comparison
- Documentation and Report Preparation