# Phase-2

***Student Name:*** *Kishore P*

***Register Number:*** *712523205032*

***Institution****: PPG Institute of Technology*

***Department:*** *B. Tech IT(II)*

***Date of Submission:*** *10/05/2025*

***Githup Repository Link:*** *https://github.com/kishore-200007/NM_KISHORE-P_DS*

***Project title:*** Cracking the Market Code with AI-driven Stock Price Prediction using Time Series Analysis

## 1.Problem Statement

***Objective:***

*The primary goal is to develop an AI-powered system capable of accurately forecasting stock prices by analyzing historical market data. This involves leveraging time series analysis techniques to identify patterns and trends in stock price movements, enabling more informed investment decisions.*

***Challenges:***

- ***Market Volatility:*** *Stock prices are influenced by a multitude of unpredictable factors, including economic indicators, geopolitical events, and investor sentiment, making accurate prediction inherently difficult.* • ***Non-Stationary Data:*** *Financial time series data often exhibit nonstationarity, meaning their statistical properties change over time. This violates the assumptions of many traditional forecasting models, necessitating advanced methods to handle such complexities.*

- ***Noise and Irregularities:*** *The presence of noise and outliers in stock price data can obscure underlying patterns, posing a significant hurdle for predictive modelling.*

***Approach:***

- ***Data Collection and Preprocessing:*** *Gather historical stock price data, ensuring it's cleaned and normalized to handle missing values and outliers.*
- ***Model Selection:*** *Employ advanced AI models suitable for time series forecasting, such as Long Short-Term Memory (LSTM) networks, which are adept at capturing temporal dependencies in sequential data.*
- ***Feature Engineering:*** *Incorporate relevant features like trading volume, moving averages, and technical indicators to enhance model performance.*
- ***Model Training and Evaluation:*** *Train the selected models on historical data and evaluate their performance using appropriate metrics like Mean Squared Error (MSE) or Mean Absolute Percentage Error (MAPE).*

***Expected Outcomes:***

- *A robust AI-driven model that can predict future stock prices with a reasonable degree of accuracy.*
- *Insights into the effectiveness of different time series analysis techniques and AI models in stock price prediction.*
- *A framework that can be adapted and extended to other financial instruments or markets.*

***Significance:***

- *curate stock price prediction models can serve as valuable tools for investors, traders, and financial analysts, aiding in risk management and strategic decision-makinAg. By harnessing the power of AI and time*

*series analysis, this research aims to contribute to the development of more sophisticated financial forecasting tools.*

## 2. Project Objectives

1. ***Develop a Predictive AI Model***

   - *Build an AI system (e.g., LSTM, GRU, Transformer models) that can accurately forecast stock prices based on historical time series data.*

2. ***Analyze Time Series Data***

   - *Use time series techniques (like ARIMA, SARIMA, Prophet, etc.) to model trends, seasonality, and noise in stock price movements.*

3. ***Feature Engineering and Data Enrichment***

   - *Extract important features (technical indicators, moving averages, volume data) to improve the predictive power of the model.*

4. ***Improve Prediction Accuracy***

   - *Optimize the model to minimize errors (using metrics like RMSE, MAPE) and achieve reliable short-term and long-term forecasting.*

5. ***Handle Financial Data Challenges***

- *Address problems like non-stationarity, volatility, missing data, and outliers through advanced preprocessing and model tuning.*

6. ***Model Evaluation and Benchmarking***

- *Compare AI models (e.g., deep learning vs traditional models) to find the most effective method for stock price prediction.*

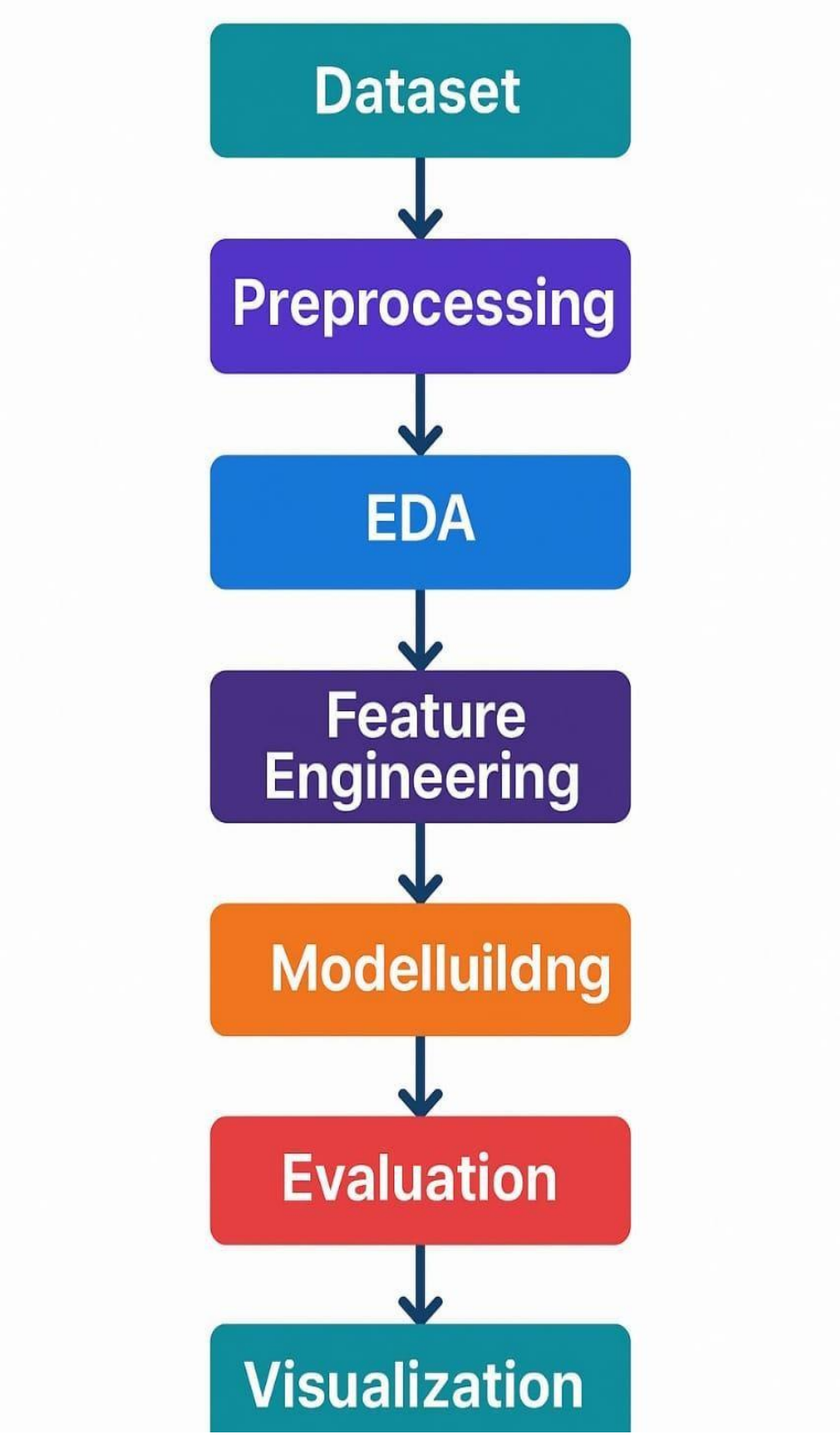
7. ***Create a Scalable and Adaptable Framework***

- *Build a solution that can be generalized to different stocks, indices, and even other financial markets.*

8. ***Generate Actionable Insights***

- *Provide visualizations, trend forecasts, and potential investment signals to assist investors and analysts.*

## 3.Flowchart of the Project Workflow



## 4.Data Description

1. ***Source of Data***
   - *Historical stock market data collected from public financial databases such as Yahoo Finance, Google Finance, Quandl, Alpha Vantage, or other APIs.*

2. ***Time Period***
   - *Data typically spans multiple years (e.g., 5–10 years) to capture long-term trends, seasonal patterns, and market cycles.*

3. ***Frequency of Data***
   - *Data can be collected at different intervals:*
     - ***Daily*** *(most common for stock prediction)*
     - ***Hourly*** *or* ***minute-wise*** *(for intraday trading models)*

4. ***Main Features (Columns)***
   - ***Date/Time*** *— timestamp for each record*
   - ***Open Price*** *— price at market open*
   - ***High Price*** *— highest price of the day*
   - ***Low Price*** *— lowest price of the day*
   - ***Close Price*** *— price at market close*
   - ***Adjusted Close Price*** *— close price adjusted for dividends and splits*
   - ***Volume*** *— number of shares traded*

5. ***Additional Engineered Features*** *(Optional)*
   - *Moving Averages (e.g., 10-day, 50-day)*
   - *RSI (Relative Strength Index)*
   - *MACD (Moving Average Convergence Divergence)*
   - *Bollinger Bands*
   - *Volatility Indicators*

6. ***Data Quality Considerations***
    - *Handle missing values, sudden jumps due to splits or dividends, and market holidays.*
    - *Normalize or scale the data before feeding it to AI models (e.g., using MinMaxScaler).*

## 5. Data Preprocessing

1. **Missing Value Handling**
    - *Apply **forward-fill** (pandas.DataFrame.ffill()) or **interpolation** to impute missing prices.*
    - *Drop rows if missing data is extensive and cannot be reasonably filled.*

2. **Outlier Detection and Correction**
    - Use **IQR method** or **Z-score** to detect abnormal price spikes.
    - Correct for stock splits or dividends using **Adjusted Close** values.

3. **Stationarization**
    - Apply **first-order differencing** (df.diff()) to remove trends.
    - Use **Augmented Dickey-Fuller Test (ADF Test)** to verify stationarity.

4. **Normalization / Scaling**
    - Scale features using:
        - **MinMaxScaler** (sklearn.preprocessing.MinMaxScaler) — scales data to [0,1] range.
        - **StandardScaler** — mean = 0, variance = 1 (especially for models sensitive to feature magnitude).

5. **Feature Engineering**

- Generate technical indicators:
  - **Moving Averages** (Simple, Exponential)
  - **Relative Strength Index (RSI)**
  - **MACD (Moving Average Convergence Divergence)**
  - **Volatility Metrics** (rolling standard deviation)

**6.Temporal Sequence Building**

- For LSTM/GRU models:
  - Create **X, y sequences**: e.g., past 60 days → predict next day (sliding window approach).
  - Reshape input to **3D arrays**: (samples, timesteps, features).

**7.Train-Test Split**

- **Chronological split** (not random): Training set: older data
  - Test set: most recent unseen data

**8.Data Saving**

- *Store pre-processed datasets using **Pickle**, **Feather**, or **Parquet** formats for faster loading during model training.*

## 6. Exploratory Data Analysis (EDA)

*EDA is a fundamental step in data science that involves analyzing datasets to summarize their main characteristics, often with visual methods. In the context of stock price prediction, EDA helps in understanding the underlying patterns, trends, and anomalies in the data, which is crucial for building effective predictive models.*


**Key EDA Steps for Stock Price Time Series**

1. **Data Loading and Structure Examination**

   - Load the dataset using libraries like pandas.
   - Inspect the first few rows with df.head() and check data types and missing values using df.info ().

2. **Time Series Decomposition**

   - Decompose the time series into trend, seasonality, and residual components using methods like STL decomposition.

   - Visualize these components to understand underlying patterns.

3. **Statistical Summary**

- Generate summary statistics using df.describe() to understand the distribution of features like mean, median, standard deviation, etc.
- Identify any skewness or kurtosis in the data.

4. **Missing Value Analysis**

- Identify missing values using df.isnull(). sum().
- Decide on strategies for handling missing data, such as imputation or removal.

5. **Outlier Detection**

- Use statistical methods like Z-scores or IQR to detect outliers.
- Visualize outliers using box plots or scatter plots.

6. **Feature Engineering**
   - Create new features like moving averages, relative strength index (RSI), or moving average convergence divergence (MACD) to capture market trends.
   - These features can provide additional insights into market behaviour.

7. **Correlation Analysis**

- Compute correlation matrices to understand relationships between different features.
- Use heatmaps to visualize these correlations.

8. **Visualization**

- Plot time series data to observe trends and patterns over time.

- Use libraries like matplotlib or plotly for interactive visualizations.

### Sample Visualizations

- **Closing Price Over Time**: A line plot showing the stock's closing price over the selected period.
- **Volume Traded**: Bar plots depicting the trading volume on different days.
- **Moving Averages**: Overlaying short-term and long-term moving averages on the closing price to identify trends.
- **Correlation Heatmap**: A heatmap showing the correlation between different technical indicators.

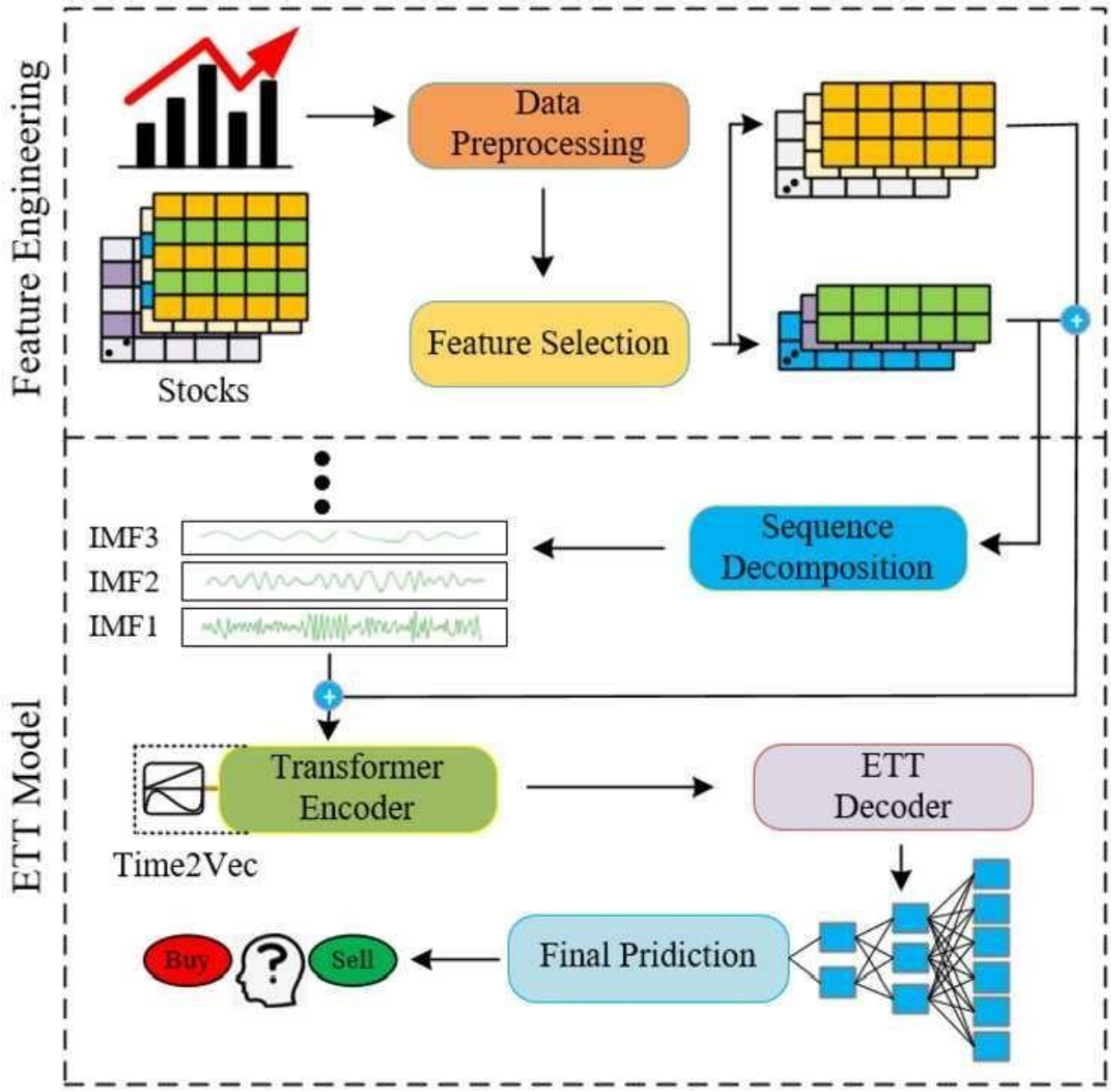
### Further Reading and Resources

- Time Series Forecasting: A Practical Guide to Exploratory Data Analysis
- Exploratory Data Analysis on Stock Market Data
- Stock Price EDA (Time Series Analysis) - Kaggle

## 7. Feature Engineering

### *Feature Engineering Overview*

*Feature engineering involves creating new input variables (features) from raw stock price data to enhance the predictive power of machine learning models. In the context of stock price prediction, effective feature engineering captures underlying patterns, trends, and market behaviours, enabling models to make more accurate forecasts.*




***Key Feature Engineering Techniques***

1. ***Lagged Features***

- *Introduce previous time steps as features (e.g., closing price 1 day ago, 5 days ago) to provide temporal context.*
- *Helps models learn from past behaviours to predict future movements.*

2. ***Rolling Statistics***
   - *Compute moving averages, rolling standard deviations, and other statistics over a window of time.*
   - *Captures short-term trends and volatility.*
   - *Example: 7-day moving average of closing prices.*

3. ***Technical Indicators***
   - *Utilize established financial metrics to gauge market conditions. Examples include:*
     - ***Relative Strength Index (RSI):*** *Indicates overbought or oversold conditions.*
     - ***Moving Average Convergence Divergence (MACD):*** *Shows trend direction and momentum.*
     - ***Stochastic Oscillator:*** *Compares closing price to its price range over a period.*
     - ***Average True Range (ATR):*** *Measures market volatility.*
   - *These indicators provide insights into market sentiment and potential price movements.*

4. ***Price Transformations***
   - *Apply mathematical transformations to stabilize variance and make data more suitable for modelling.*
   - *Common transformations include logarithmic and percentage changes.*

5. ***Time-Based Features***
   - *Extract temporal components such as day of the week, month, quarter, or year.*
   - *Helps capture seasonal effects and cyclical patterns in the market.*

6. ***Sentiment Analysis (Optional)***
    - *Analyze market sentiment from news articles, social media, or financial reports.*
    - *Quantify sentiment as a feature to gauge public perception and its potential impact on stock prices.*

## 8.Model Building

***Model building is a pivotal phase in developing an AI-driven stock price prediction system. It involves selecting appropriate algorithms, training them on historical data, and fine-tuning to achieve optimal predictive performance.***

### *1.Model Selection*

- *Choosing the right model depends on the data characteristics and prediction goals. Common models include:*

- ***ARIMA (Autoregressive Integrated Moving Average):*** *Suitable for univariate time series data with trends and seasonality.*

- ***LSTM (Long Short-Term Memory):*** *A type of recurrent neural network adept at capturing long-term dependencies in sequential data.*

- ***GRU (Gated Recurrent Unit):*** *Similar to LSTM but with a simpler architecture, often requiring less computational resources.*

- ***Hybrid Models:*** *Combining models like CNN-LSTM with XGBoost to leverage both deep learning and gradient boosting techniques.*

### *2. Model Training*

*Training involves feeding the model with preprocessed historical stock data:*

- ***Data Splitting:*** *Divide data into training, validation, and test sets to evaluate model performance effectively.*
- ***Sequence Preparation:*** *For models like LSTM, structure the data into sequences (e.g., using the past 60 days to predict the next day).*
- ***Hyperparameter Tuning:*** *Adjust parameters such as learning rate, number of layers, and neurons to optimize performance.*

### *1. Model Evaluation*

*Assess the model's predictive accuracy using metrics like:*

- ***RMSE (Root Mean Square Error):*** *Measures the model's prediction error magnitude.*
- ***MAE (Mean Absolute Error):*** *Calculates the average absolute difference between predicted and actual values.*
- ***$R^2$ Score:*** *Indicates the proportion of variance explained by the model.*

## 9.Visualization of Results & Model Insights

- *Effective visualization is crucial in evaluating and interpreting the performance of AI-driven stock price prediction models. It aids in understanding model accuracy, identifying patterns, and communicating findings to stakeholders.*

### *1. Predicted vs. Actual Stock Prices*

- ***Line Plots**: Plotting actual and predicted stock prices over time helps in visually assessing the model's forecasting accuracy.*
- ***Residual Plots**: Displaying the difference between actual and predicted values can highlight patterns in prediction errors.*

### *2. Performance Metrics Visualization*

- ***Error Metrics**: Visual representations of metrics like RMSE (Root Mean Square Error), MAE (Mean Absolute Error), and MAPE (Mean Absolute Percentage Error) provide quantitative measures of model performance.*
- ***Comparative Bar Charts**: When evaluating multiple models, bar charts can effectively compare their performance metrics side by side.*

### *3. Feature Importance Analysis*

- ***Feature Importance Plots**: For models like XGBoost, visualizing feature importance scores helps in understanding which features contribute most to the predictions.*
- ***SHAP Values**: SHAP (Shapley Additive Explanations) plots offer insights into how each feature impacts individual predictions, enhancing model interpretability.*

*4.* ***Time Series Decomposition***

- ***Trend, Seasonality, and Residuals:*** *Decomposing the time series into these components and visualizing them can reveal underlying patterns and assist in model selection and tuning.*

*5.* ***Interactive Dashboards***

- ***Real-Time Monitoring:*** *Utilizing tools like Plotly Dash or Tableau to create interactive dashboards allows for dynamic exploration of model predictions and performance over time.*

## 10.Tools and Technologies Used

***1.Programming Languages & Development Environments***

- ***Python****: The primary language for data science and machine learning tasks.*
- ***Jupiter Notebooks****: An interactive environment for developing and presenting data analysis projects.*
- ***Google Collab****: A cloud-based platform that allows for running Python code in the browser with free access to GPUs.*

***2.Data Handling & Preprocessing Libraries***

- ***Pandas****: For data manipulation and analysis.*
- ***NumPy****: For numerical computations.*
- ***Scikit-learn****: For preprocessing tasks like scaling, encoding, and splitting datasets.*

### *3.Time Series Analysis Tools*

- ***Stats models****: For statistical models like ARIMA.*
- ***Facebook Prophet****: Designed for forecasting time series data with seasonality.*
- ***Nixtla's TimeGEN-1****: A recent advancement enabling zero-shot forecasting, allowing predictions on unseen datasets without retraining.*

### *4.Machine Learning & Deep Learning Frameworks*

- ***TensorFlow & Kera's****: For building and training deep learning models like LSTM and GRU.*
- ***PyTorch****: An alternative deep learning framework known for its flexibility.*
- ***XGBoost****: An optimized gradient boosting library for supervised learning tasks.*
- ***Scikit-learn****: Also provides implementations for algorithms like Support Vector Machines (SVM) and Random Forests.*

### *5.Visualization Tools*

- ***Matplotlib & Seaborn****: For static data visualizations.*
- ***Plotly****: For interactive plots and dashboards.*
- ***Tableau****: A business intelligence tool for creating interactive and shareable dashboards.*

### *6.Model Interpretation & Explainability*

- ***SHAP (Shapley Additive Explanations)****: For interpreting the output of machine learning models.*
- ***LIME (Local Interpretable Model-agnostic Explanations)****: For explaining the predictions of any classifier.*

### *7.Data Sources*

- ***Yahoo Finance API****: For fetching historical stock data.*
- ***Alpha Vantage****: Provides real-time and historical stock market data.*
- ***Quandl:*** *Offers a wide range of financial, economic, and alternative datasets.*

## 11. Team Members and Contributions

| Team Member | Contribution |
|---|---|
| DHARSHINI N | Data Cleaning |
| DHARANI A | EDA |
| KISHORE P | Feature engineering |
| LAKSHINE S | Model development |
| ASHOK KUMAR R | Documentation and reporting |