Github Link: https://github.com/kavindass/project--Cracking-the-market-code-.git

Project Title: Cracking the market code with AI-driven stock price prediction using time series analysis

PHASE-2

- **Problem Statement**

The stock market is a highly complex and dynamic system where prices are driven by various factors such as economic indicators, geopolitical events, market sentiment, and company performance. Despite extensive research, accurately forecasting stock prices remains a significant challenge due to the unpredictable nature of the market and the massive amount of data involved. Traditional methods like fundamental analysis and statistical models, though useful, struggle to account for the nonlinear relationships and temporal dependencies in market data, often leading to limited forecasting accuracy.

- **Project Objectives**

Predict Stock Prices: Forecast future stock prices for companies like Apple (AAPL), Microsoft (MSFT), and Tesla (TSLA) using historical price data.

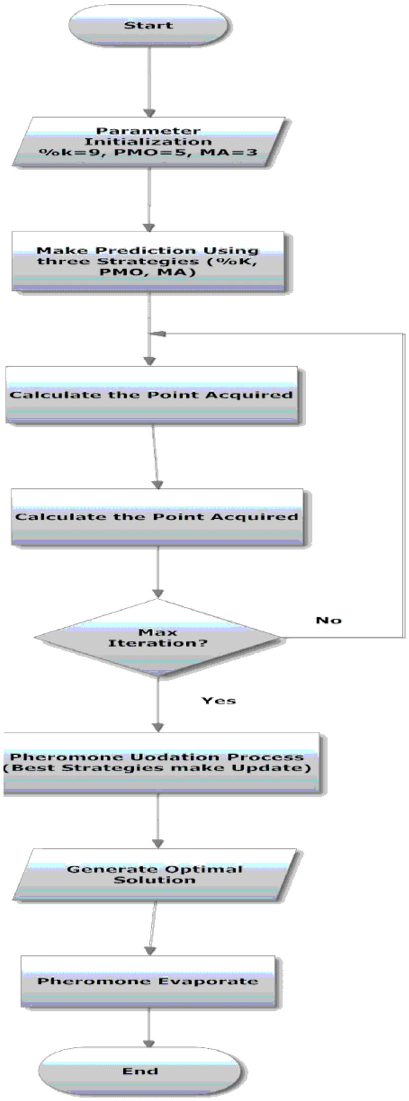
Implement Deep Learning Models: Leverage time series analysis with models such as LSTM, GRU, and Transformer to capture temporal dependencies in stock data.

Enhance Accuracy with Technical Indicators: Integrate features like moving averages (SMA, EMA), RSI, and MACD to improve model performance.

Data Preprocessing: Clean and normalize stock data, handling missing values and ensuring proper feature scaling for model readiness.

Evaluate Model Performance: Assess predictive accuracy using metrics like MAE, RMSE, and R^2 to ensure generalizability on unseen data

- **Flowchart of the Project Workflow**


Start
Parameter Initialization %k=9, PMO=5, MA=3
Make Prediction Using three Strategies (%K, PMO, MA)
Calculate the Point Acquired
Calculate the Point Acquired
Max Iteration?
No
Yes
Pheromone Uodation Process (Best Strategies make Update)
Generate Optimal Solution
Pheromone Evaporate
End

- **Data Description**
  - Source: Historical stock price data for companies like Apple (AAPL), Microsoft (MSFT), and Tesla (TSLA) from 2010 to 2023, sourced from platforms like Yahoo Finance and Alpha Vantage.
  - Key Features:
  - Date: Trading date.
  - Open, High, Low, Close: Daily price information.
  - Volume: Number of shares traded.
  - Adjusted Close: Price adjusted for dividends and stock splits.
  - Data Characteristics:
  - Daily frequency with ~3,000–4,000 data points per company.
  - Missing values were handled using forward/backward fill methods.

- Feature Engineering: Technical indicators like SMA, EMA, RSI, and MACD were added to improve predictive power.
- Dataset Link: https://www.kaggle.com/datasets/niszarkiah/job-market-insights-dataset

- **Data Preprocessing**
- Missing Data Handling: Used forward-fill and backward-fill techniques to handle missing values and maintain the temporal sequence.
- Feature Engineering: Added technical indicators (SMA, EMA, RSI, MACD) and lag features (past price/volume) to capture market trends.
- Normalization: Applied Min-Max scaling to normalize stock prices and volume data for consistency across features.
- Train-Test Split: Split the dataset chronologically into training (80%) and testing (20%) sets to preserve the time series nature.
- Sequence Generation: Converted the time series data into sequences for LSTM/GRU models using a sliding window approach.
- **Exploratory Data Analysis (EDA)**
- Summary Statistics: Calculated mean, median, and standard deviation for stock features (Open, Close, Volume) to understand their distributions.
-
- Missing Data: Visualized missing values using heatmaps and handled them with forward-fill or backward-fill methods.
-
- Correlation Analysis: Used a heatmap to check correlations between stock price features and identify potential relationships.
-
- Visualization: Plotted stock price trends and technical indicators (SMA, EMA, RSI) to spot patterns, trends, and volatility.

- **Feature Engineering**

Technical Indicators:

Added SMA, EMA, RSI, and MACD to capture trends, momentum, and market signals.

Lag Features:

Created lag features using previous stock prices (e.g., last 5 days) to capture temporal dependencies.

Rolling Features:

Calculated rolling averages and standard deviations to capture recent trends and volatility.

Price & Volume Changes:

Added daily returns and rolling volume averages to track stock price changes and market activity.

- **Model Building**

Model Selection:

Chose deep learning models like LSTM and GRU for their ability to capture temporal dependencies in time series data.

Data Preparation:

Transformed the data into sequences using a sliding window approach for LSTM/GRU models, where past stock data is used to predict future prices.

Model Architecture:

Built models with multiple layers: LSTM/GRU layers followed by Dense layers for regression tasks, with activation functions and dropout for regularization.

Hyperparameter Tuning:

Used techniques like grid search or random search to optimize hyperparameters, including learning rate, number of layers, and batch size.

Model Evaluation:

Evaluated models using metrics like MAE, RMSE, and R^2 to measure prediction accuracy and model generalization.

- **Visualization of Results & Model Insights**

- Prediction vs Actual: Plotted predicted stock prices against actual values to visually compare model performance.

- Error Analysis: Visualized prediction errors to identify patterns or biases in the model's predictions.

- Loss Curves: Plotted training and validation loss to check for overfitting or underfitting.

- Performance Metrics: Used metrics like MAE, RMSE, and R^2 to evaluate model accuracy and robustness.

- **Tools and Technologies Used**
- Programming Language:
- Python for data analysis, model development, and visualization.
- Libraries:
- Pandas and NumPy for data manipulation.
- Matplotlib and Seaborn for data visualization.
- Scikit-learn for preprocessing and model evaluation.
- Deep Learning Framework:
- TensorFlow and Keras for building and training LSTM/GRU models.
- Data Source:
- Yahoo Finance and Alpha Vantage for stock price data.
- Model Evaluation:
- Matplotlib for visualizing prediction vs. actual price trends.
- **Team Members and Contributions**

[kishore] – Project Lead & Model Development: Led the project, developed and fine-tuned deep learning models (LSTM, GRU), and handled the overall architecture of the solution.

[kavindass] – Data Collection & Preprocessing: Collected historical stock data from Yahoo Finance and Alpha Vantage, and performed data cleaning, normalization, and feature engineering.

[karthick] – Exploratory Data Analysis (EDA) & Visualization: Conducted EDA, visualized stock price trends, and identified key patterns, and created visualizations for model evaluation.

[kumar] – Model Evaluation & Performance Analysis: Evaluated model performance using metrics like MAE, RMSE, and R^2, and provided insights on model results, errors, and optimization.