Github Link: https://github.com/snekha435/NM-snekhavalli.git

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

PHASE-2

1. Problem Statement

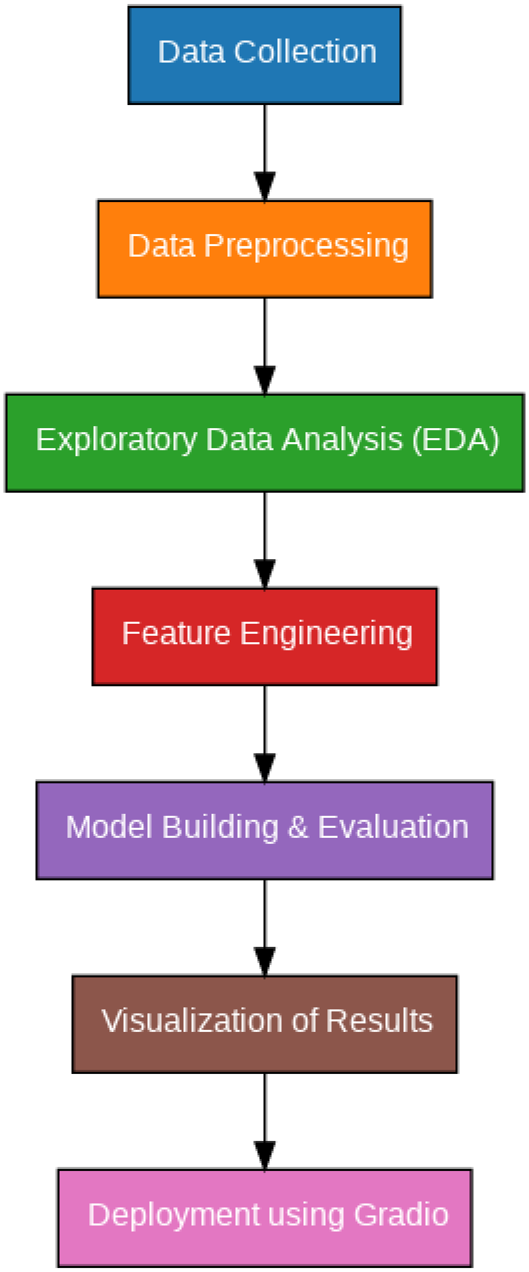
The inherent volatility and complexity of financial markets make accurate stock price prediction a significant challenge. Traditional methods often fall short in capturing the intricate patterns and dependencies within the vast amounts of historical and real-time data. This unpredictability poses risks for investors and financial institutions, highlighting the need for more sophisticated and data-driven approaches. This project aims to address this challenge by leveraging the power of Artificial Intelligence and time series analysis techniques to develop a robust and insightful stock price predition model. The goal is to provide a valuable tool for understanding market dynamics and potentially informing investment decisions.

2. Project Objectives

The primary objective of this project is to develop and evaluate an AI-driven model capable of predicting future stock prices with a reasonable degree of accuracy. Specific objectives include:

- **Data Acquisition and Preparation:** To gather relevant historical stock price data and preprocess it for time series analysis.
- **Exploratory Data Analysis (EDA):** To understand the underlying characteristics of the stock price data, identify trends, seasonality, and potential anomalies.
- **Feature Engineering:** To create meaningful features from the raw time series data that can enhance the predictive power of the model.
- **Model Building:** To implement and train various time series forecasting models, including traditional statistical models and advanced machine learning/deep learning models.
- **Model Evaluation:** To rigorously evaluate the performance of the developed models using appropriate time series evaluation metrics.
- **Visualization and Interpretation:** To effectively visualize the predicted stock prices and gain insights into the factors influencing the model's predictions.
- **Technology Exploration:** To utilize relevant tools and technologies for data handling, model development, and deployment.

3. Flowchart of the Project Workflow



4. Data Description

- **Dataset Name:** Stock Market Performance Dataset

- **Source:** News & Financial Market Reports (e.g., Google data, financial news websites)
- **Type of Data:** Structured tabular data
- **Records and Features:** ~150 company records, 4 features (Company, Price, Change, % Change)
- **Target Variable:** % Change (numeric – can be used for classification or regression)
- **Static or Dynamic:** Dynamic dataset (market values change daily)
- **Attributes Covered:**
  - Company Name
  - Stock Price
  - Daily Price Change
  - Percentage Change

- **Dataset link : https://tradingeconomics.com/united-states/stock-market**

5. Data Preprocessing

This stage involves cleaning and preparing the data for analysis and model building. Key steps include:

- **Handling Missing Values:** Identifying and addressing missing data points (e.g., imputation using mean, median, or more sophisticated techniques).
- **Outlier Detection and Treatment:** Identifying and handling extreme values that might skew the analysis or model training (e.g., using statistical methods like IQR or Z-score, or domain-specific knowledge).
- **Data Type Conversion:** Ensuring all data types are appropriate for analysis.
- **Stationarity Checks:** Testing for stationarity in the time series data using methods like the Augmented Dickey-Fuller (ADF) test or Kwiatkowski-Phillips-Schmidt-Shin (KPSS) test.
- **Transformation for Stationarity:** Applying transformations like differencing, logarithmic transformation, or seasonal decomposition of time series (STL) to make the data stationary if required by the chosen models.
- **Data Scaling:** Scaling numerical features to a similar range (e.g., using MinMaxScaler or StandardScaler) to improve model performance and prevent dominance of features with larger magnitudes.
- **Splitting Data:** Dividing the data into training, validation, and testing sets for model development and evaluation, ensuring temporal order is maintained.

6. Exploratory Data Analysis (EDA)

This section focuses on understanding the characteristics and patterns within the preprocessed stock price data. Common EDA techniques include:

- **Time Series Plots:** Visualizing the stock price over time to identify trends, seasonality, and volatility.
- **Descriptive Statistics:** Calculating summary statistics like mean, median, standard deviation, minimum, and maximum for different variables.
- **Autocorrelation Function (ACF) and Partial Autocorrelation Function (PACF) Plots:** Analyzing the correlation between a time series and its lagged values to identify the order of ARIMA models or inform feature engineering.
- **ACF:** Plots the correlation between the time series and its lags. $\rho_k = \frac{\text{Cov}(Y_t, Y_{t-k})}{\text{Var}(Y_t)}$
- **PACF:** Plots the partial correlation between the time series and its lags, removing the influence of intermediate lags.
- **Decomposition Plots:** Separating the time series into its trend, seasonal, and residual components.
- **Distribution Plots (Histograms, Box Plots)**: Examining the distribution of stock prices and other relevant variables.
- **Volatility Analysis:** Visualizing and analyzing the changing volatility of the stock over time.
- **Correlation Analysis (Heatmaps):** Exploring the relationships between different variables in the dataset (if external data is included).

7. Feature Engineering

Enhance model prediction by generating new features:

- **Lagged Variables:** Past stock prices (e.g., price at t-1, t-5, t-10).
- **Moving Averages:** Trend smoothing via rolling averages:
- **SMA:** $\text{SMA}t = \frac{1}{n} \sum{i=0}^{n-1} Y_{t-i}$
- **EMA:** $\text{EMA}t = \alpha Y_t + (1-\alpha) \text{EMA}{t-1}$
- **Volatility Measures:** Historical volatility using rolling standard deviation:
- **Historical Volatility (n-day):** $\sigma_t = \sqrt{\frac{1}{n-1} \sum_{i=1}^{n} (R_{t-i+1} - \bar{R})^2}$ (where R = daily returns).
- **Technical Indicators:** Common analysis tools (RSI, MACD, Bollinger Bands, Stochastic Oscillator).
- **Time-Based Features:** Day, month, year, quarter.
- **Interaction Terms:** Combinations of existing features.

- **External Data Features:** Sentiment, macroeconomic data (if used).

8. Model Building

Developing and training prediction models:

- **Classical Time Series:** ARIMA (and SARIMA), Exponential Smoothing (Simple, Holt's, Holt-Winters).
- **Machine Learning:** Linear Regression, SVR, Random Forest, Gradient Boosting (XGBoost, LightGBM).
- **Deep Learning:** RNNs (LSTM, GRU), TCNs.
- **Each model involves:**
  - Architecture: Core principles.
  - Hyperparameter Tuning: Optimization (Grid/Random/Bayesian Search).
  - Training: Fitting to data.
  - Validation: Tuning and preventing overfitting.

9. Visualization of Results & Model Insights:

- **Predicted vs. Actual:** Compare test set predictions to actual values.
- **Residual Analysis:** Analyze error plots for model assumptions and bias.
- **Error Distribution:** View error histograms/density plots.
- **Performance Metrics:** Report MSE, RMSE, MAE, MAPE, R^2.
- **Feature Importance:** Identify key predictive features.
- **Scenario Analysis (Optional):** Explore prediction variations with different inputs.

10. Tools and Technologies Used

- **Programming Language:** Python
- **Data Analysis and Manipulation Libraries:** Pandas, NumPy
- **Time Series Analysis Libraries:** Statsmodels, Prophet
- **Machine Learning Libraries:** Scikit-learn
- **Deep Learning Libraries:** TensorFlow, Keras, PyTorch
- **Data Visualization Libraries:** Matplotlib, Seaborn, Plotly
- **Development Environment:** Jupyter Notebooks, VS Code, Google Colab
- **Version Control:** Git, GitHub.

11. Team Members and Contributions

- ○ *Data cleaning and EDA* – Vishwabharathi.S
- ○ *Feature engineering – Ramkishor.S*
- ○ *Model development – Snekhavalli.K*
- ○ *Documentation and reporting – Safeera Nowsheen.M*