**Stock Market Analysis and Prediction**

**1. Introduction**

**- Project Overview:**

The Stock Market Analysis and Prediction project provides an interactive dashboard to visualize and analyze stock performance for 10 companies over the past 40 years. The project utilizes Python for data extraction via the `yfinance` library, predictive modeling using the XGB Regressor algorithm, and Power BI for dashboard creation and visualization.

**- Problem Statement:**

Understanding historical stock data and predicting future trends can be complex. This project simplifies the process by integrating data visualization and machine learning predictions into a user-friendly dashboard.

**- Goals:**

- Fetch and analyze stock market data for key companies.

- Predict future stock prices using machine learning.

- Build an interactive dashboard in Power BI for end-user insights.

**2. Features**

**1. Dashboard 1: Revenue & Stock Performance**

- Key metrics such as trading volume, opening price, closing price, highest and lowest prices.

- Stock price trends with 50-day and 200-day Simple Moving Averages (SMA).

- Comparison table for selected companies showing the latest price, percentage change, and earnings per share (EPS).

- Daily stock price analysis using candlestick charts.

- Prediction table with forecasted closing prices for selected stocks.

**2. Dashboard 2: Business Performance & Market Forecast**

- Revenue distribution of companies visualized using treemaps.

- Top 5 companies by trading volume summarized in a table.

- Comparison of total revenue and net income over quarters via bar charts.

- Predicted vs. actual stock prices for recent days displayed as line charts.

- Maximum close prices by year and company visualized through multi-line charts.

**3. Prerequisites**

**- System Requirements:**

- Operating System: Windows/Linux/MacOS.

- Minimum RAM: 8 GB.

- Disk Space: 10 GB free.

**- Libraries and Frameworks:**

- `yfinance` for fetching stock market data.

- `pandas` and `numpy` for data manipulation.

- `xgboost ` XGBRegressor modeling.

- `joblib` for saving/loading trained models.

- `ta` (Technical Analysis) for calculating indicators like RSI.

- Power BI for dashboard creation and visualization.

**4. Data**

**- Data Sources:**

- Stock data fetched using the `yfinance` Python library.

**- Dataset Description:**

- Companies: 10 key players (e.g., AAPL, MSFT, GOOGL, META, NVDA, AMZN, TSLA, PYPL, INTC, CSCO).

- Timeframe: Historical data spanning 40 years.

- Features: `Date`, `Open`, `High`, `Low`, `Close`, `Volume`, `Adjusted Close`, and additional financial indicators (e.g., SMA, RSI, lagged features).

**5. Project Workflow**

**1. Data Collection:**

- Used the `yfinance` library to fetch historical data for 40 years.

- A script reshaped the data to a company-wise format with features such as Open, High, Low, Close, Volume, and Adjusted Close.

- Added date column as a regular feature and saved the data for Power BI.

**2. Data Cleaning and Preprocessing:**

- Checked for missing values and dropped or imputed them as necessary.

- Calculated technical indicators like Simple Moving Averages (SMA) and Relative Strength Index (RSI).

- Added lagged features (`y\_lag1` and `Volume\_Lag1`) for predictive modeling.

**3. Predictive Modeling:**

- Built and trained a Random Forest regression model to predict future stock prices based on technical indicators and historical data.

- Predictions were generated for the next 7 trading days using a function that iteratively updated input features.

**4. Dashboard Development:**

- Imported the processed data into Power BI using Python scripts.

- Designed interactive visualizations, including charts, tables, and KPIs.

**6. Python Scripts Overview**

**1. Data Fetching Script:**

- Fetches 40 years of historical data for 10 companies using `yfinance`.

- Reshapes and cleans the data, adding relevant columns such as `Date`, `Volume`, and `Adjusted Close`.

- Data is organized company-wise and exported for Power BI.

**2. Prediction Script:**

- Predicts the next 7 trading days' stock prices using a trained xgboost model.

- Features include SMA (10-day and 50-day), RSI, and lagged values (`y\_lag1`, `Volume\_Lag1`).

- Uses the `ta` library for indicator calculation and `joblib` for loading the model.

- Iteratively updates the predictions, recalculating indicators after each step.

**7. Results and Visualizations**

**- Key visualizations in the dashboards:**

- Line charts showcasing SMA trends for stock prices.

- Treemaps for revenue distribution across companies.

- Candlestick charts for daily stock price movements.

- Line charts comparing predicted and actual stock prices.

- Prediction table showing forecasted closing prices for the next 7 days.

**8. Challenges and Solutions**

- Challenge: Handling large datasets spanning 40 years for multiple companies.

Solution: Optimized data processing with company-wise data reshaping and concise feature selection.

- Challenge: Accurate prediction of volatile stock prices.

Solution: Incorporated technical indicators and lagged features into the xgboost model, fine-tuning hyperparameters.

**9. Future Scope**

- Enable real-time data fetching and live dashboard updates.

- Integrate sentiment analysis from financial news and social media to enhance prediction accuracy.

- Add sector and index-level analysis for broader insights.

**10. Conclusion**

The Stock Market Analysis and Prediction project demonstrates an effective combination of data science and visualization. By leveraging Python scripts for data preparation and predictive modeling and Power BI for dashboards, the project delivers actionable insights into historical stock performance and future trends.

**11. Appendix**

**- Glossary:**

- XGB Regressor: A machine learning algorithm used for regression and classification.

- SMA: Simple Moving Average, a technical indicator for stock trends.

- RSI: Relative Strength Index, an indicator to evaluate stock momentum.