Title : Time Series Forecasting

Introduction

This project presents a Stock Price Forecasting platform implemented as a full-stack web application using FastAPI for the backend and HTML/CSS/JavaScript with Chart.js for the frontend. The application enables users to retrieve historical stock data, perform comprehensive exploratory data analysis (EDA), build and evaluate forecasting models including ARIMA, SARIMA, Random Forest, and XGBoost, generate predictions with confidence intervals, and export data for visualization in Tableau. The goal is to provide in-depth insights into stock price trends and assist users in making data-driven decisions.

Approach

The project follows a structured approach to implement stock price forecasting:

1. **Data Collection**: Fetching historical stock data from Yahoo Finance using yfinance.
2. **Exploratory Data Analysis (EDA)**:
    - Basic statistics (mean, median, std, min, max, range).
    - Stationarity check using the Augmented Dickey-Fuller (ADF) test.
    - Seasonality and trend analysis via decomposition.
    - Autocorrelation (ACF) and Partial Autocorrelation (PACF) plots.
    - Time window analysis (daily, weekly, monthly averages and volatilities).
    - Pattern heatmap for hour vs. day of week.
    - Visualizing historical stock prices and other charts.
3. **Model Selection & Training**:
    - **ARIMA (AutoRegressive Integrated Moving Average)**: A standard time-series forecasting model for univariate data.
    - **SARIMA (Seasonal ARIMA)**: An extension of ARIMA that incorporates seasonality.
    - **Random Forest**: A machine learning ensemble method with feature engineering (returns, moving averages, volatility, momentum, lags).
    - **XGBoost**: An advanced gradient boosting model for improved accuracy.
    - Hyperparameter tuning and feature creation for optimal performance.
4. **Forecasting**:
    - Predicting stock prices for user-selected periods (next day, week, month, quarter).
    - Generating 95% confidence intervals based on historical volatility.
    - Visualizing forecasts with historical data, including interactive charts.
5. **Performance Evaluation**:
    - Metrics used: Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), Mean Absolute Percentage Error (MAPE).
    - For ML models: Additional R^2 score; for statistical models: AIC/BIC.

    - Insights on trend, expected return, volatility, and model confidence.
6. **Export**:
    - Generating CSV files with enhanced data (e.g., moving averages, returns) for Tableau visualization.

Methodology

1. **Data Preprocessing:**
    - The application allows users to select a stock symbol (e.g., AAPL, GOOGL), start date, and end date via a dropdown and date inputs.
    - The stock data is retrieved using the yfinance library.
2. **Data Cleaning:**
    - Handling Missing Values: Missing values are filled using forward-fill and back-fill methods.
    - Outlier Detection: Extreme outliers are handled to prevent model skew.
    - Ensuring Data Integrity: Duplicates are removed, dates are standardized to UTC, and non-numeric values are managed.
    - Cleaning report generated: total records, missing filled, duplicates removed.
3. **Stationarity Check:**
    - ADF Test: Determines if the stock price time series is stationary or requires differencing.
    - Autocorrelation Plot: Analyzes time dependencies in the data.
4. **Model Training:**

    ARIMA Model
    - Trained with order parameters (p, d, q), where:
        - p (AutoRegressive term) determines the number of lag observations.
        - d (Differencing order) ensures stationarity.
        - q (Moving Average term) defines the size of the error component.
    - Forecasting performed using the fitted model.

    SARIMA Model

    - Extends ARIMA with seasonal order (P, D, Q, s) to capture seasonal patterns in stock prices.

    Random Forest and XGBoost Models

    - Feature engineering: Returns, log returns, moving averages (5,10,20), volatility (5,10), momentum (5,10), volume features, time-based features (day of week, month, quarter), and lags (1-10).
    - Trained on 80/20 split, with recursive forecasting for future steps.
    - Feature importance calculated.

5. **Forecasting & Visualization**
    - Users select model type and forecast period via dropdowns.
    - Predictions generated with confidence intervals, visualized using Chart.js (line charts for historical/forecast, bar for ACF/PACF, etc.).
    - Detailed tables and insights provided (e.g., trend: upward/downward, expected return %).
6. **Performance Evaluation**
    - Predicted values compared against test data or in-sample fits.
    - Evaluation metrics:
        - MAE (Mean Absolute Error)
        - RMSE (Root Mean Squared Error)
        - MAPE (Mean Absolute Percentage Error)
        - R^2 (for ML models)
        - AIC/BIC (for statistical models)

Results & Conclusion

- The application successfully forecasts stock prices with improved accuracy using a combination of statistical and ML models.
- Seasonal and trend patterns are captured effectively, with XGBoost and Random Forest often outperforming ARIMA/SARIMA on complex datasets.
- Future improvements could include real-time data integration, additional models like LSTM, or ensemble methods for hybrid forecasting.

Technologies Used

- **Python**: FastAPI (backend API), Pandas, NumPy, Matplotlib, Statsmodels (ARIMA/SARIMA), Scikit-Learn (Random Forest), XGBoost, yfinance (data retrieval).
- **Frontend**: HTML/CSS/JavaScript, Chart.js (interactive visualizations).
- **Other**: Uvicorn (server), CORS middleware for cross-origin requests.

Deployment

- The application can be deployed using Uvicorn on platforms like Render, Vercel, or AWS for real-time stock forecasting access via web browser.

This project is a practical implementation of **time-series forecasting** using statistical models, aiming to provide insights into stock market trends.