Financial Time-Series Anomaly Detection

Objective:
To build a tool that identifies anomalies in stock price trends using financial indicators and machine learning to detect unusual market activities or potential manipulations.

1. Dataset Preprocessing Steps

- **Data Collection:**
  Historical stock price data for selected companies (e.g., AAPL, MSFT, TSLA) was obtained from Yahoo Finance and loaded from local CSV files. These files contain daily Open, High, Low, Close, and Volume data, which served as the foundation for further analysis.
- **Feature Engineering:**
  - **Datetime Formatting:** Converted timestamp fields to datetime format and set as index for time-series analysis.
  - **Handling Missing Values:** Missing data was forward-filled using the last known value (`ffill`) to preserve temporal consistency.
  - **Technical Indicators Calculated:**
    - Simple Moving Average (SMA)
    - Exponential Moving Average (EMA)
    - Relative Strength Index (RSI)
    - Bollinger Bands (Upper and Lower)
    - Moving Average Convergence Divergence (MACD)

- **Normalization:**
  Applied Min-Max scaling to bring features to a common scale, especially for model input.

```
from sklearn.preprocessing import MinMaxScaler

features_to_normalize = ['SMA_20', 'EMA_20', 'RSI', 'BB_High', 'BB_Low', 'Close', 'Open', 'High', 'Low', 'Adj Close', 'Volume']
scaler = MinMaxScaler()
df_normalized = df[features_to_normalize].dropna()
normalized_values = scaler.fit_transform(df_normalized)
df_normalized_temp = pd.DataFrame(normalized_values, index=df_normalized.index, columns=features_to_normalize)
df.update(df_normalized_temp)

print(df.head())

features = df[['Close', 'SMA_20', 'EMA_20', 'RSI']].dropna()
model = IsolationForest(contamination=0.01, random_state=42)
anomaly_predictions = model.fit_predict(features)

df['anomaly_if'] = 0
df.loc[features.index, 'anomaly_if'] = anomaly_predictions
df['anomaly_if'] = df['anomaly_if'].map({1: 0, -1: 1})
```

```
            Unnamed: 0     Open       High    Low  Close  Adj Close  \
Date
2001-01-01           0  16.5000  16.500000  16.50  16.50  12.229188
2001-01-02           1  15.9875  16.299999  15.91  16.25  12.043896
2001-01-03           2  15.8775  15.947500  15.50  15.90  11.784488
2001-01-04           3  16.1250  16.875000  15.75  16.50  12.229188
2001-01-05           4  16.5000  16.500000  16.50  16.50  12.229188

               Volume  SMA_20  EMA_20  RSI  BB_High  BB_Low  anomaly_if
Date
2001-01-01        0.0     NaN     NaN  NaN      NaN     NaN         NaN
2001-01-02  1607584.0     NaN     NaN  NaN      NaN     NaN         NaN
2001-01-03   506560.0     NaN     NaN  NaN      NaN     NaN         NaN
2001-01-04   894416.0     NaN     NaN  NaN      NaN     NaN         NaN
2001-01-05        0.0     NaN     NaN  NaN      NaN     NaN         NaN
```

2. Model Selection and Rationale

- **Unsupervised Anomaly Detection:**
  - **Isolation Forest:** Chosen for its efficiency on high-dimensional data and suitability for anomaly detection without labeled data.
  - **DBSCAN:** Used to identify clusters and noise (outliers) in multidimensional indicator space.
- **Forecasting-Based Anomaly Detection:**
  - **LSTM (Long Short-Term Memory):** Recurrent neural network suitable for capturing temporal dependencies and trends in financial time-series data.
  - **Prophet (by Facebook):** Used for quick deployment of forecasting models that account for seasonality and trends, useful for comparing predicted vs actual prices.
- **Rationale:**
  Isolation Forest and DBSCAN allow detecting statistical anomalies based on pattern deviations. Forecasting models help flag anomalies based on deviation from expected trends.

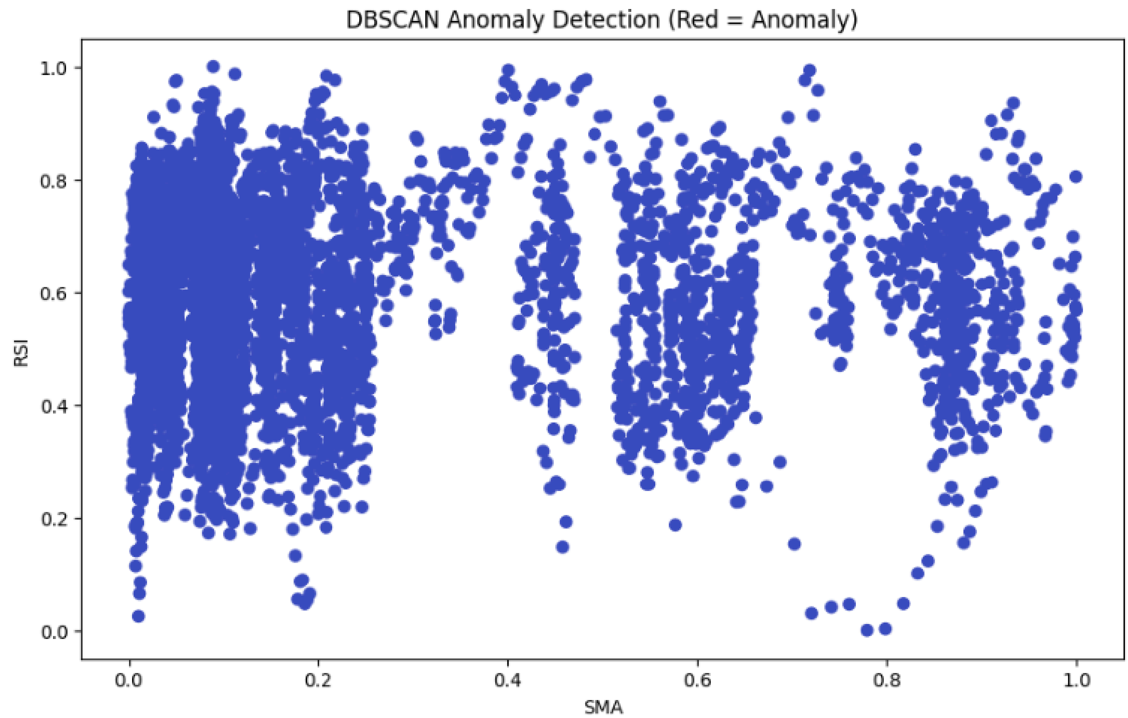
3. Challenges Faced and Solutions

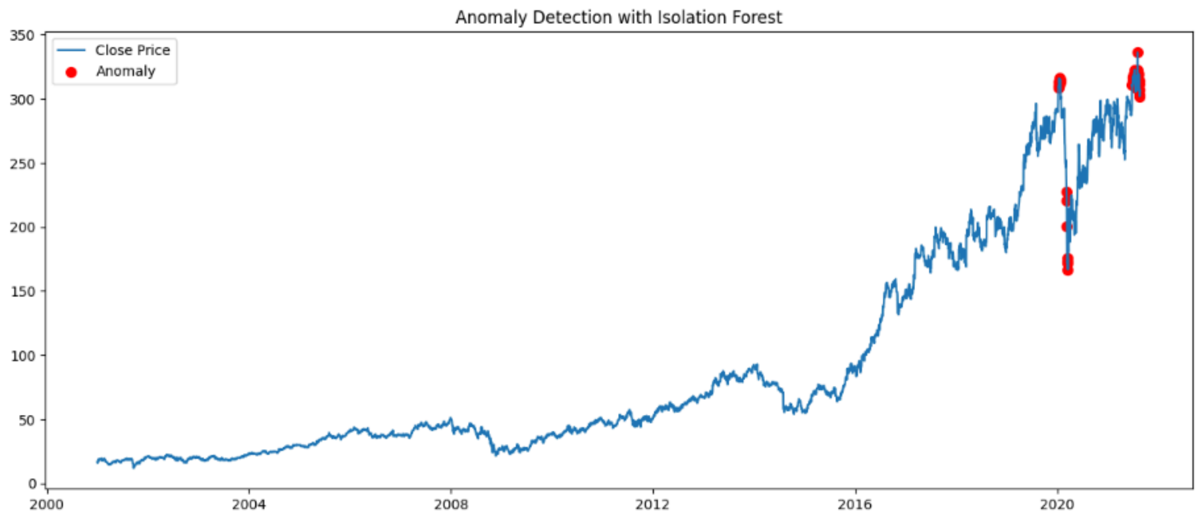
- **Challenge:** Stock prices are highly volatile and noisy.
  Solution: Applied smoothing techniques (e.g., SMA, EMA) and used rolling windows to reduce noise impact.
- **Challenge:** Hyperparameter tuning for DBSCAN (epsilon, min_samples) was sensitive.
  Solution: Performed grid search on a range of values and used visualization (k-distance plot) to identify optimal `eps`.
- **Challenge:** LSTM required significant training time and tuning.
  Solution: Used a smaller time window and reduced sequence length for efficient training. Additionally, Prophet was used as a lightweight alternative.
- **Challenge:** Visualizing multi-company anomalies.
  Solution: Created interactive plots using Plotly to overlay anomalies on historical price charts for each stock.

4. Results with Visualizations and Interpretations

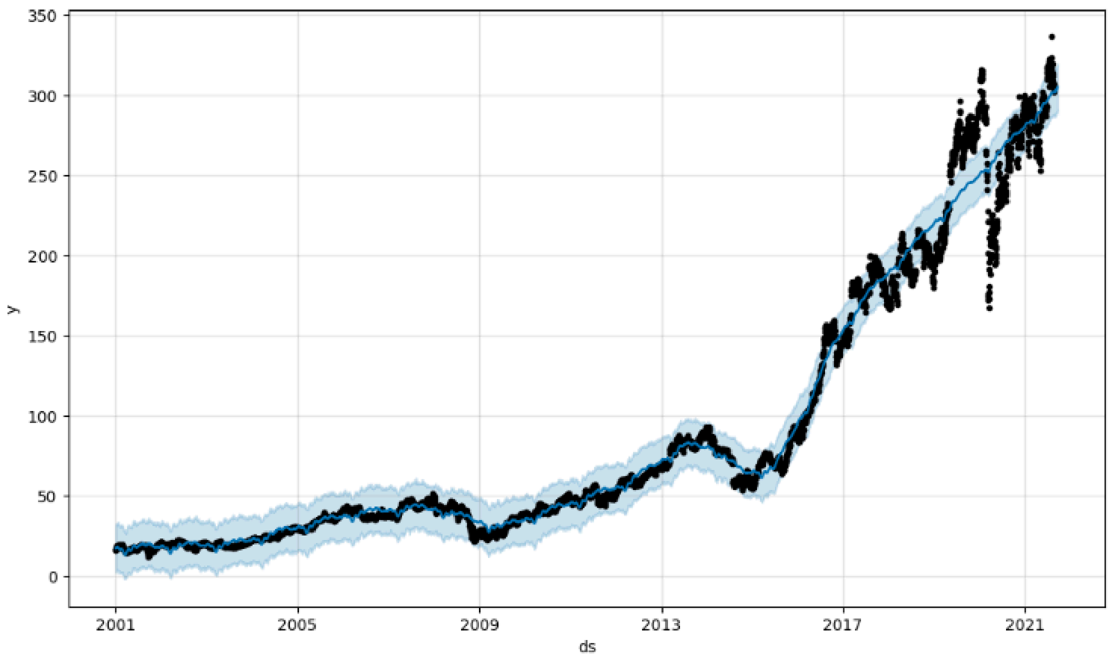
- **Anomaly Detection (Isolation Forest & DBSCAN):**
  Plots revealed spikes or drops in stock prices where models marked points as anomalies (red dots). These often aligned with known news events or earnings releases.

```
Detected anomalies: 0
```





- **Forecasting Anomalies (LSTM/Prophet):**
  - Forecasted trends (blue line) were overlaid with actual stock prices.
  - Significant deviations were flagged as anomalies.
  - Example: Sudden drop in TSLA stock was not captured by forecast, flagged as anomaly.



- **Interpretation:**
  - Most anomalies corresponded to earnings announcements, SEC filings, or macroeconomic news.
  - The approach can assist investors or analysts in early identification of irregular behavior.

Outcome

A functional anomaly detection pipeline was developed combining statistical, unsupervised, and time-series forecasting methods. It provides visual insights into unusual price movements that may indicate market manipulation or unexpected events.