Project Report

Project Title: **Time Series Analysis and Forecasting for Stock Market**

Objective:

The primary objective of this project was to gain practical exposure to Time Series Analysis and apply forecasting techniques on real-world stock market data.

The goals included:
1. Understanding data preprocessing steps specific to time series data.
2. Applying statistical and machine learning models for forecasting.
3. Evaluating and comparing model performance using standard error metrics (MSE, MAE, RMSE).
4. Visualizing trends, seasonality, and residuals to gain deeper insights.

Dataset Description:

- **Source:** The dataset consisted of historical daily prices of multiple stocks and ETFs, sourced from a publicly available stock market dataset.
  Link - https://www.kaggle.com/datasets/jacksoncrow/stock-market-dataset

- **Components:** Each record included Date, Open, High, Low, Close, Adjusted Close, and Volume columns.

- **Size:** The dataset covered thousands of stocks with a long historical time range, providing sufficient data for robust time series modeling.

- **Preprocessing:**
  1. Checked and handled missing values and duplicate rows.
  2. Converted date columns to datetime format for time-indexing.
  3. Filtered relevant stocks (e.g., NASDAQ-listed, non-ETFs).
  4. Selected Apple Inc. (AAPL) as the representative stock for forecasting.

Exploratory Data Analysis (EDA):

Key EDA steps included:

- **Line Plots:** Visualized Close and Adjusted Close prices over time to identify trends and volatility.
- **Rolling Statistics:** Plotted 30-day rolling mean and standard deviation to assess smoothened trends and detect periods of high or low volatility.
- **STL Decomposition:** Applied Seasonal-Trend decomposition using Loess (STL) to break down the series into Trend, Seasonal, and Residual components.

Stationarity Checks:

To apply models like ARIMA and SARIMA, ensuring stationarity is critical:

1. Augmented Dickey-Fuller (ADF) Test:

- The initial p-value was high (>0.99), indicating non-stationarity.
- After first-order differencing, the p-value dropped near zero, indicating the differenced series was stationary.

2. KPSS Test:

- The initial test indicated a trend component (non-stationarity).
- Second-order differencing confirmed stationarity with p-value ~0.1.

Models Implemented:

1. Naive Forecast:

- Used as a baseline for comparison.
- Simply forecasted all future values equal to the last observed value from the training data.
- Expected to have the highest error among all models.

2. ARIMA (5,2,2):

- A classic statistical forecasting model that combines Auto-Regressive, Integrated (Differencing), and Moving Average components.
- Parameters (p=5, d=2, q=2) were chosen based on ACF/PACF plots and grid search.
- Delivered one of the lowest MSE and RMSE scores.

3. SARIMA (1,2,1)(1,0,1,7):

- Extended ARIMA with seasonal terms, useful for stock prices that might exhibit weekly or monthly seasonal effects.
- Seasonal order (1,0,1,7) models weekly patterns.
- Performed nearly as well as ARIMA, validating mild seasonality.

4. Facebook Prophet:

- An automated time series model by Meta (Facebook).
- Easy to implement but for this dataset, Prophet overfit trend and seasonality poorly, resulting in higher errors.

5. LSTM Neural Network:

- A deep learning approach tailored for sequence prediction.
- 2 LSTM layers with 64 and 32 units respectively, with Dropout layers.
- Data was scaled using MinMaxScaler for optimal training.
- Trained with early stopping to avoid overfitting.
- LSTM matched ARIMA's performance.

Train-Test Split:

- Data was split with the last 365 days as the test set.
- Models were trained on the remaining historical data.
- All predictions and metrics were calculated using this consistent split.

Highlighted: Model Evaluation Summary

Model	MSE	MAE	RMSE
ARIMA(5,2,2)	83.155757	5.498516	9.118978
LSTM	83.235476	5.492099	9.123348
SARIMA(1,2,1)(1,0,1,7)	83.360252	5.512264	9.130184
Naive	115.738311	7.763578	10.758174
Prophet	32046.524492	178.936098	179.015431

Interpretation of Results:

- **Best Models**: ARIMA(5,2,2) and LSTM were the best performers.

- **SARIMA:** Marginal improvement with seasonal components.
- **Naive Benchmark:** Higher error, as expected.
- **Prophet:** Poorer fit for this high-frequency data.

Tools & Technologies Used:

1. Python 3
2. Pandas, NumPy
3. Matplotlib, Seaborn
4. Statsmodels
5. Facebook Prophet
6. TensorFlow & Keras
7. Scikit-learn

Key Learnings:

- Data preprocessing and stationarity testing are fundamental.
- Model selection must align with data behavior.
- Evaluation metrics provide concrete means to compare models.
- Sophisticated models may not always outperform statistical baselines.

Challenges Faced:

- Ensuring stationarity with multiple differencing rounds.
- Tuning LSTM hyperparameters.
- Managing run time and computing cost.

Conclusion:

This project successfully demonstrated how time series analysis can be applied to forecast stock market trends using both classical statistical models and modern deep learning techniques. By collaboratively comparing ARIMA, SARIMA, Prophet, and LSTM models, we showed that while advanced neural networks have great potential, well-tuned traditional models can deliver equally strong results for financial time series. Overall, this project strengthened our team's skills in data preparation, model development, evaluation, and insight generation, providing us with valuable practical experience for solving real-world forecasting challenges.