Apple Stock Price Prediction

Mentor & Team Member Details

Mentor :- Dilawar Basha & Prajwal

Team Members:-

- Falak Hashmi
- Shanmukha Sasank Nandula
- Jemiyah Lijo
- PALAMALA PENCHALA SWAMY
- Chandana S
- Pankaj Kumar Singh
- Kasturi Sai Ganaamruth

Objective

The primary objective of the Apple Stock Price Prediction Project is to accurately forecast the future stock prices of Apple Inc. (AAPL) using historical stock market data. By analyzing past trends, seasonal patterns, and price movements, this project aims to develop a reliable predictive model that can assist investors, analysts, and financial planners in making informed decisions. The project leverages statistical methods such as ARIMA/SARIMA, and machine learning algorithms like XGBoost or LSTM, to capture the underlying structure of the time series data. Ultimately, the goal is to provide data-driven insights into future price movements, helping to identify potential investment opportunities and minimize risks in stock trading.

Dataset Details : Apple Inc. Stock Prices

The dataset contains historical stock price data for **Apple Inc. (AAPL)**. It is a time series dataset, typically used for financial analysis and forecasting tasks. Contains 2011 rows and 6 columns.

[5]:

Date	Open	High	Low	Close	Adj Close	Volume
2012-01-03	58.485714	58.928570	58.428570	58.747143	50.765709	75555200
2012-01-04	58.571430	59.240002	58.468571	59.062859	51.038536	65005500
2012-01-05	59.278572	59.792858	58.952858	59.718571	51.605175	67817400
2012-01-06	59.967144	60.392857	59.888573	60.342857	52.144630	79573200
2012-01-09	60.785713	61.107143	60.192856	60.247143	52.061932	98506100
...	...	...	...	...	...	...
2019-12-23	280.529999	284.250000	280.369995	284.000000	282.054138	24643000
2019-12-24	284.690002	284.890015	282.920013	284.269989	282.322266	12119700

[6]:

```
#No.of rows & columns
data.shape
```

[6]: (2011, 6)

Key Columns:

- **Date** – The trading date (daily frequency).
- **Open** – Price of the stock at market open.
- **High** – Highest price reached during the trading day.
- **Low** – Lowest price reached during the trading day.
- **Close** – Price of the stock at market close.
- **Adj Close** – Adjusted close price accounting for dividends and stock splits.
- **Volume** – Number of shares traded on that day.

Data Preprocessing

To prepare the Apple stock data for time series forecasting, several preprocessing steps were performed:

1. Aligning with U.S. Business Days:
*A Custom BusinessDay calendar was created using the US Federal Holiday Calendar to define valid trading days, excluding weekends and U.S. public holidays.
*A date range (my_range) was generated from 2012-01-03 to 2019-12-30 using this custom calendar.

2. Handling Non-Trading Days:
*The dataset (data) was compared against my_range to identify any mismatched dates.
*Additional custom holidays (like Good Friday and market-closed days not included in the standard calendar) were manually added to ensure alignment.

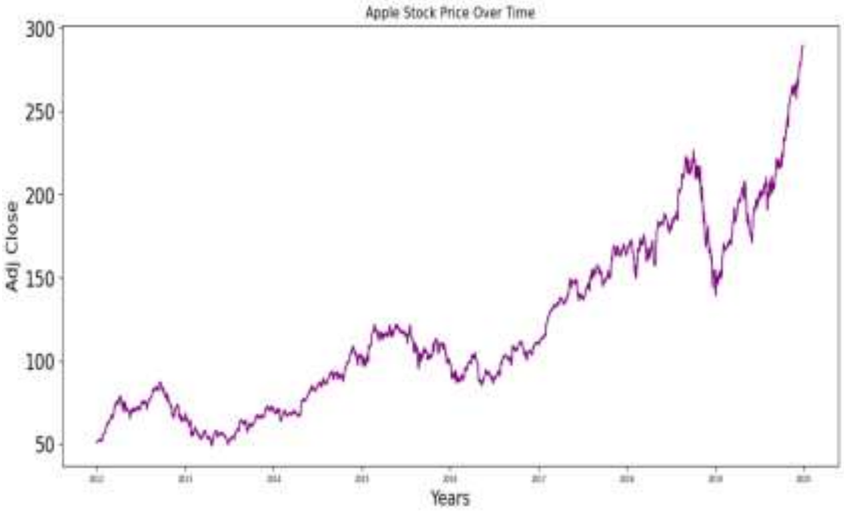
3. Filtering by Frequency:
*The dataset's frequency was set to the refined Custom Business Day calendar, ensuring consistent time intervals for modeling.
*Non-trading days were excluded, and the index was set to valid business dates only.

4. Dropping Unnecessary Columns:
*Columns not required for forecasting (Open, High, Low, Close, Volume) were removed.
*Only the Adj Close column was retained, as it reflects stock prices adjusted for splits and dividends.

5. Saving the Cleaned Dataset:
*The cleaned and preprocessed dataset was saved as new_Apple_data.csv for future use in modeling

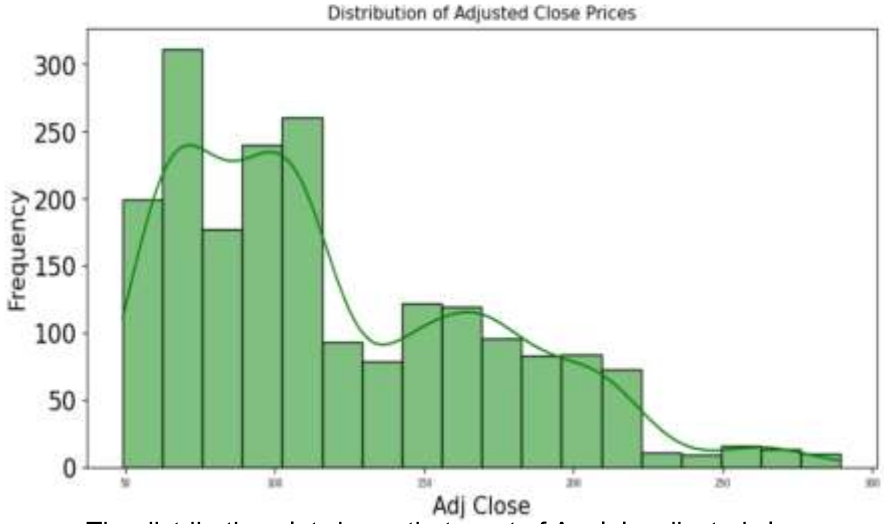
Exploratory Data Analysis (EDA)

Apple Stock Price Over Time



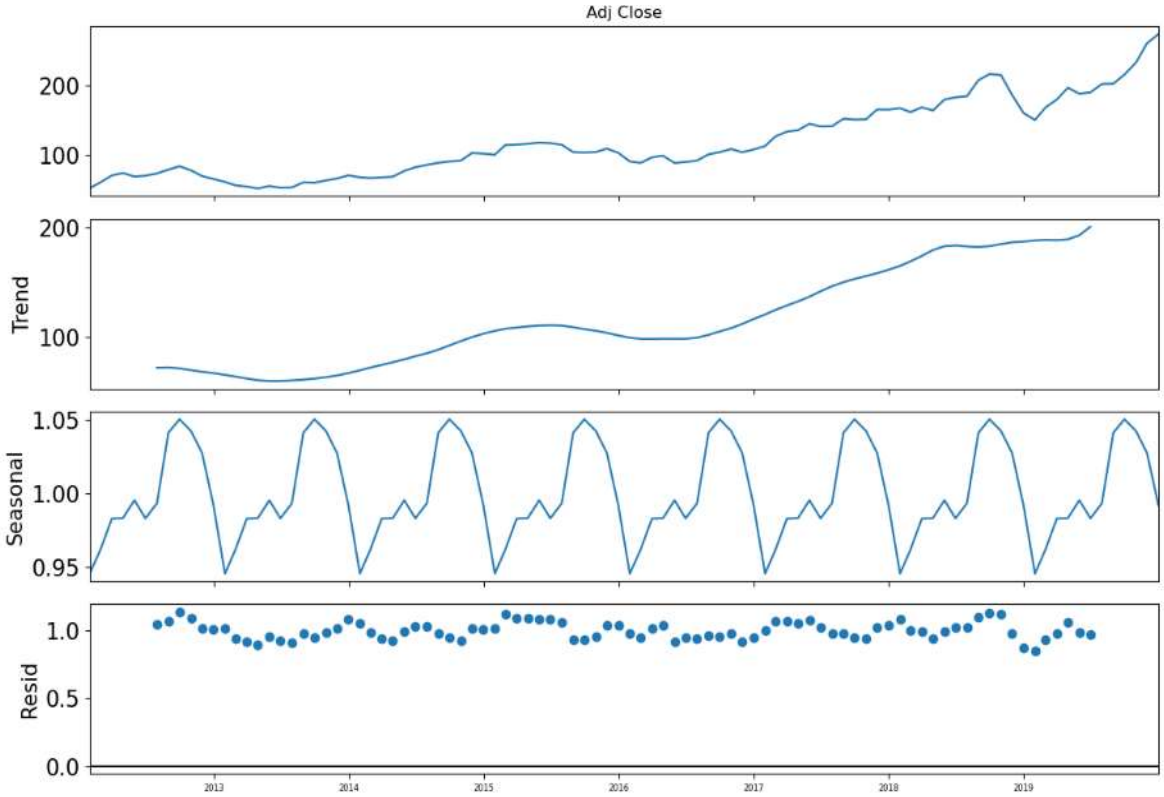
The chart shows a steady rise in Apple's adjusted stock price from 2012 to 2019, with periods of short-term volatility. It highlights strong long-term growth, especially after 2017, indicating increasing investor confidence and market performance.

Distribution of Adjusted Close Prices



The distribution plot shows that most of Apple's adjusted close prices fall between \$50 and \$150, with a right-skewed pattern. Higher prices are less frequent, indicating consistent growth over time with fewer occurrences of high-price peaks.

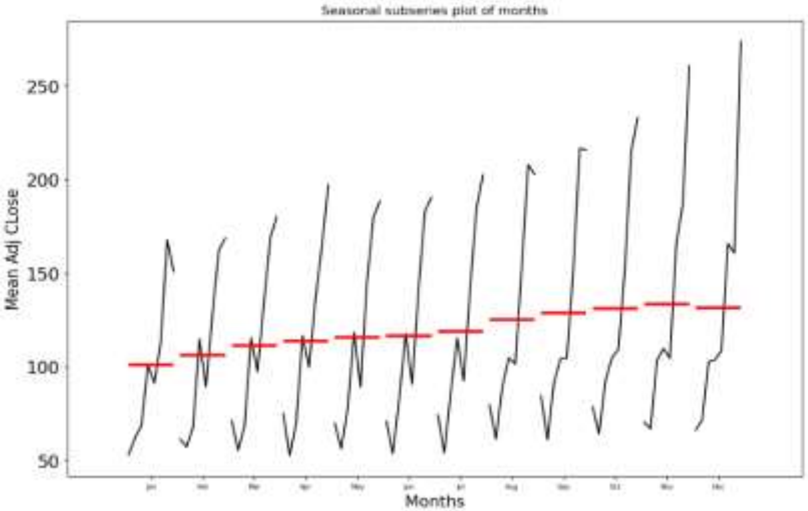
Time Series Decomposition of Apple Stock Prices(EDA)



The decomposition plot shows a clear upward trend in Apple's stock prices over time, with strong yearly seasonality. The residuals remain relatively stable, indicating that the multiplicative model captures both the trend and seasonal patterns effectively.

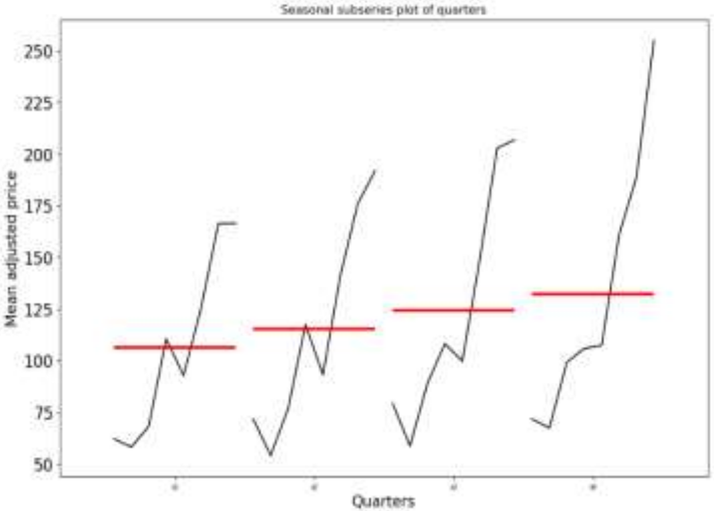
Exploratory Data Analysis (EDA)

Monthly Seasonal Subseries Plot



The seasonal subseries plot shows a clear yearly upward trend in Apple’s average stock prices, with noticeable month-wise patterns. Prices tend to rise sharply toward year-end, especially in November and December, indicating strong seasonal effects.

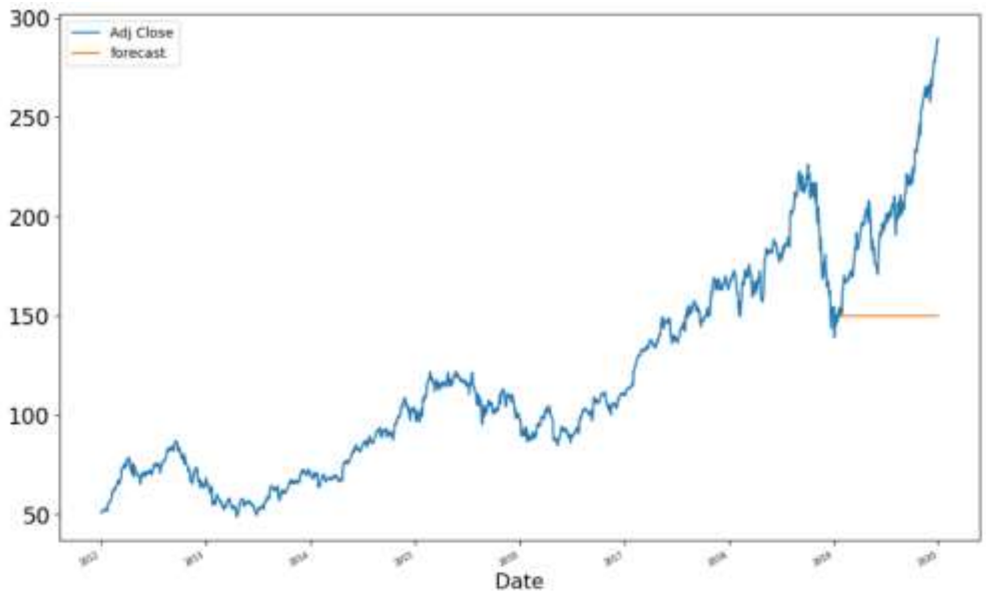
Quarterly Seasonal Subseries Plot



The quarterly subseries plot reveals a consistent upward trend across all quarters, with the highest stock prices typically observed in Q4. This suggests strong year-end performance, likely influenced by product launches and holiday season demand.

Model Building using ARIMA

An ARIMA(1,1,1) model was used to forecast Apple’s adjusted stock prices. The model achieved a MAPE of 26.37% and RMSE of 66.82, indicating moderate prediction accuracy with noticeable forecasting error compared to advanced models.




```
#Mean Absolute Percentage Error (MAPE) - ARIMA
mape1 = np.mean(np.abs(df['forecast'] - df['Adj Close'])/np.abs(df['Adj Close']))
mape1
```

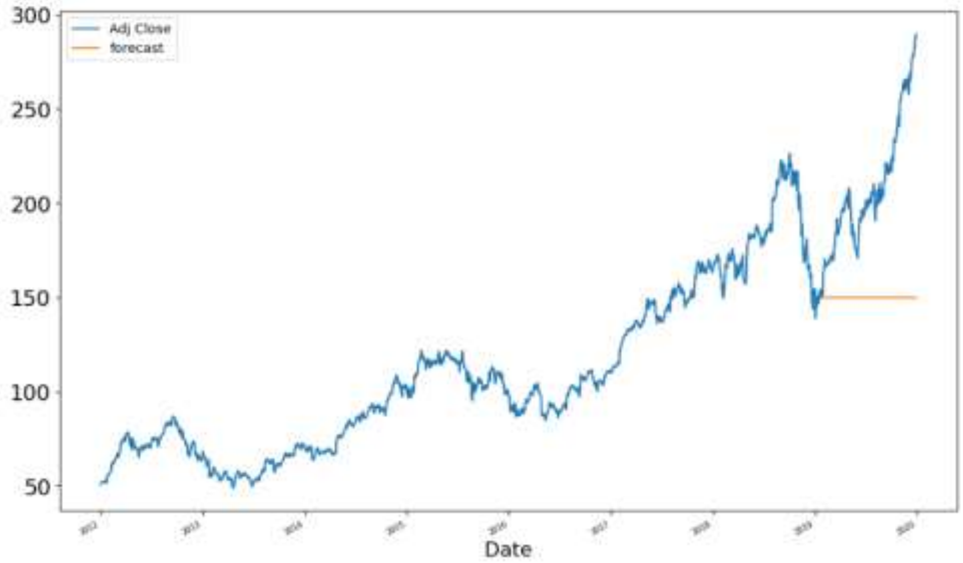
```
np.float64(0.26374136638963713)
```

```
#Root Mean Squared Error (RMSE) - ARIMA
rmse1 = np.mean((df['forecast'] - df['Adj Close'])**2)**.5
rmse1
```

```
np.float64(66.82204794662974)
```

Model Building using SARIMA

The SARIMA model was implemented using the SARIMAX class to forecast Apple's stock price. The plot shows actual vs. predicted values, A SARIMA(1,1,1)(1,1,1,30) model was applied to forecast Apple's stock prices. It achieved a **MAPE of 23.45%** and **RMSE of 59.95**, showing improved accuracy over the ARIMA model by better capturing seasonality and reducing prediction error.




```
#Mean Absolute Percentage Error (MAPE) - ARIMA
mape2 = np.mean(np.abs(df0['forecast1'] - df0['Adj Close'])/np.abs(df0['Adj Close']))
mape2
```

```
np.float64(0.23450258280984498)
```

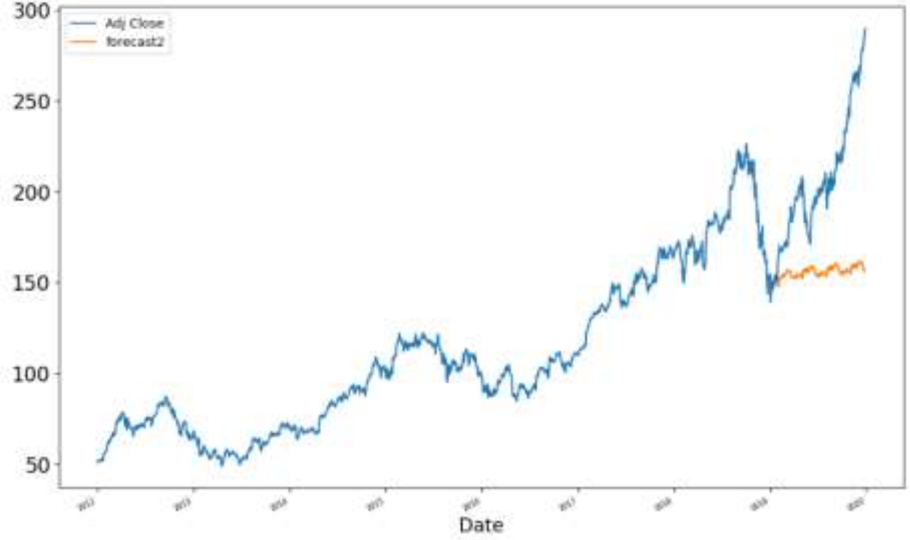
```
#Root Mean Squared Error (RMSE) - ARIMA
rmse2 = np.mean((df0['forecast1'] - df0['Adj Close'])**2)**.5
rmse2
```

```
np.float64(59.951730238991104)
```

SARIMA Model(Quarterly)

A second SARIMA model was trained with a seasonal period of 63 to improve forecasting. The plot compares actual prices with both SARIMA forecasts. This helps evaluate which seasonal configuration better captures Apple's stock price patterns over time.

The second SARIMA model (seasonal period = 63) achieved a MAPE of 23.56% and RMSE of 60.82. While slightly less accurate than SARIMA-30, it still provides a competitive forecast and helps analyze the impact of different seasonal settings.



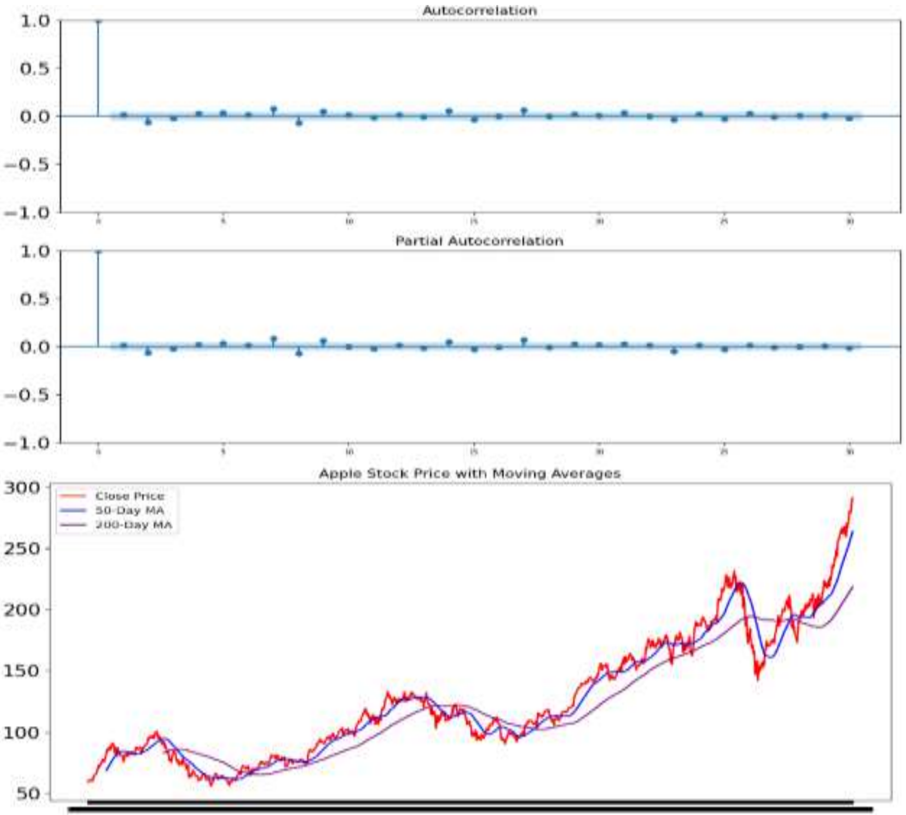

```
#Mean Absolute Percentage Error (MAPE) - ARIMA
mape3 = np.mean(np.abs(df1['forecast2'] - df1['Adj Close'])/np.abs(df1['Adj Close']))
mape3
```

```
np.float64(0.2355776244295463)
```

```
rmse3 = np.mean((df1['forecast2'] - df1['Adj Close'])**2)**.5
rmse3
```

```
np.float64(60.822064649401234)
```

Model Building (XG Boost)



The plot shows Apple’s stock price alongside its 50-day and 200-day moving averages. It highlights both short-term and long-term trends, helping identify momentum shifts. Crossovers between the averages indicate potential buy or sell signals.

The ACF and PACF plots show minimal autocorrelation, suggesting weak linear dependence in daily returns. The moving average plot confirms a strong long-term upward trend in Apple’s stock, with 50-day and 200-day MAs supporting trend analysis.

Model Building (XG Boost)

Modeling: Predicting Future Prices

```
# Selecting features for prediction (Close, 50_MA, 200_MA)
features = ['50_MA', '200_MA']
target = 'Close'
```

```
from sklearn.metrics import mean_squared_error, r2_score
import numpy as np

# Initialize the XGBoost model
xgb_model = XGBRegressor()

# Train the XGBoost model
xgb_model.fit(X_train, y_train)

# Make predictions
y_pred = xgb_model.predict(X_test)

# Calculate RMSE and R²
rmse = np.sqrt(mean_squared_error(y_test, y_pred))  # Manual square root
r2 = r2_score(y_test, y_pred)

print(f"RMSE: {rmse}")
print(f"R²: {r2}")
```

```
RMSE: 2.9128138840745987
R²: 0.9967758794958307
```

The XGBoost regressor was trained on Apple stock features to predict closing prices. It provided RMSE and R^2 scores, showing strong model performance with enhanced prediction accuracy compared to linear and ARIMA-based models, thanks to its gradient boosting capability. The XGBoost model achieved an **RMSE of 2.91** and an **R^2 score of 0.997**, indicating highly accurate predictions with minimal error. It significantly outperforms linear and ARIMA-based models, effectively capturing patterns in Apple's stock price data.

Model Evaluation and Selection

	Models	Root Mean Squared Error
0	SARIMA(Monthly)	59.951730
1	SARIMA(Qauterly)	60.822065
2	ARIMA	66.822048

SARIMA monthly model has better RMSE score comparison to other

Summary of RMSE Comparison

Among all models:

- SARIMA (Monthly) has the lowest RMSE (59.95),
- Followed by SARIMA (Quarterly) (60.82),
- And ARIMA performs the worst (66.82).

🔍 Conclusion:

SARIMA (Monthly) outperforms others in terms of predictive accuracy, making it the best statistical model for Apple stock forecasting in this analysis.

Deployment

Apple Stock Price Forecasting using SARIMA

Upload your stock CSV file with 'Adj Close' column

Drag and drop file here
Limit 200MB per file • CSV

Browse files

AAPL.csv 143.8KB

Data Preview

Date	Open	High	Low	Close	Adj Close	Volume
2019-12-23 00:00:00	280.53	284.25	280.37	284	282.0541	24643000
2019-12-24 00:00:00	284.69	284.89	282.92	284.27	282.3223	12119700
2019-12-26 00:00:00	284.82	289.98	284.7	289.91	287.9236	23280300
2019-12-27 00:00:00	291.12	293.97	288.12	289.8	287.8144	36566500
2019-12-30 00:00:00	289.46	292.69	285.22	291.52	289.5226	36028600

Select number of days to forecast

38

7 60

Out put...

Forecasted Prices (Next 30 Days)

	predicted_mean
2015	289.76
2016	290.6139
2017	290.6395
2018	290.8542
2019	291.5634
2020	291.985
2021	291.6176
2022	291.923
2023	292.243
2024	292.9604
2025	293.333

Forecast Plot

Challenges Faced During the Project

- Understanding Project Requirements

  As a beginner, it was initially difficult to fully grasp the expectations, scope, and flow of a real-time stock forecasting project.
- Confidence in Decision-Making

  As a fresher, lack of confidence in model selection or analytical decisions made it harder to contribute assertively in team discussion
- Data Gaps and Non-Trading Days

  Handling missing dates due to weekends, holidays, and market closures required careful frequency alignment and holiday calendar adjustments.
- Seasonality Detection

  Identifying the right seasonal order (monthly vs. quarterly) in SARIMA models was complex and required trial and error.
- Model Selection and Tuning

  Choosing between ARIMA, SARIMA and XGBoost involved comparing multiple metrics and tuning hyperparameters for best performance.
- Computational Time

  Fitting SARIMA models with large seasonal orders (like 63) took significant time and memory.

Thank You...