Nicole Burns

D214—Data Analytics Capstone

A. Research Question (justification, context, and hypothesis):

"Can a SARIMA model effectively forecast Apple Inc's quarterly revenue with a model accuracy of >80%?"

Since its inception in 1976, Apple has grown to become the first company with a 3 trillion-dollar market value (Vlastelica, 2023). Its continuing growth makes Apple of keen interest to stakeholders such as investors, analysts, and competitors. These stakeholders make critical decisions based on the company's financial health and growth trajectory. The ability to predict future sales with a high degree of accuracy can inform a range of strategic decisions, from investment strategies to marketing and resource allocation. This study seeks to evaluate whether a SARIMA model can effectively forecast Apple's quarterly revenue with an accuracy of >80%.

The following are the hypotheses:

Null hypothesis: H0—A SARIMA model cannot effectively forecast Apple's quarterly revenue at a model accuracy of > 80%.

Alternative hypothesis: H1—A SARIMA model can effectively forecast Apple's quarterly revenue at a model accuracy of > 80%.

B. Data Collection Process:

The data needed for the analysis is available via the Yahoo Finance website. The data is found under the 'Income Statement' and 'Quarterly' tabs. The data available spans from quarters ending in 1985 to 2023. Prior to data cleaning and transformation, the dataset contained 51 rows representing financial data names and 156 columns representing the dates each quarter ended. Names include the following:

- Total Revenue
- Cost of Revenue
- Gross Profit
- Operating Expense
- Operating Income
- Net Non-Operating Interest Income Expense
- Interest Income Non-Operating
- Interest Expense Non-Operating
- Total Other Finance Cost
- Pretax Income
- Tax Provision

- Net Income Common Stockholders
- Net Income
- Diluted NI Available to Com Stockholders
- Basic EPS
- Diluted EPS
- Total Operating Income as Reported
- Total Expenses
- Net Income from Continuing & Discontinued Operation
- Normalized Income
- Interest Income
- Interest Expense
- Net Interest Income
- EBIT Quantitative
- Reconciled Cost of Revenue
- Reconciled Depreciation
- Net Income from Continuing Operation Net Minority Interest
- Total Unusual Items Excluding Goodwill
- Total Unusual Items
- Normalized EBITDA
- Tax Rate for Calcs
- Tax Effect of Unusual Items

One of the advantages of this data-gathering methodology includes the dataset being publicly available. However, the disadvantage includes accessibility and exportation being limited. Individuals will need a subscription to view all dates available or download the dataset. I circumvented this issue by signing up for a free trial.

C. Data Extraction and Preparation:

The analysis will be performed in Jupyter Notebook using Python. The advantages of using Python include it being a very versatile programming language that can be used for a variety of tasks. Such includes machine learning techniques, the method being used for this analysis (SudoPurge, 2021). Python also has libraries specifically designed for data analysis, and time series forecasting, such as Pandas, NumPy, Statsmodels, and Scikit-learn, among others. It also is widely used in the industry and is supported by many organizations (GFG, 2023). While I didn't encounter any issues using Python, one disadvantage includes its consumption of a considerable amount of memory, causing performance issues when handling large datasets.

```
import pandas as pd

df = pd.read_csv(r'C:\Users\nki46\Documents\WGU\D214\AAPL_quarterly_financials.csv')
df.columns
```

```
Index(['name', 'ttm', '12/31/2023', '09/30/2023', '06/30/2023', '03/31/2023',
       '12/31/2022', '09/30/2022', '06/30/2022', '03/31/2022',
       ...
       '12/31/1987', '09/30/1987', '06/30/1987', '03/31/1987', '12/31/1986',
       '09/30/1986', '06/30/1986', '03/31/1986', '12/31/1985', '09/30/1985'],
      dtype='object', length=156)
```

```
df.head()
```

	name	ttm	12/31/2023	09/30/2023	06/30/2023	03/31/2023	12/31/2022	09/30/2022	06/30/2022	03/31/2
0	TotalRevenue	385,706,000,000	119,575,000,000	89,498,000,000	81,797,000,000	94,836,000,000	117,154,000,000	90,146,000,000	82,959,000,000	97,278,000,
1	\tOperatingRevenue	385,706,000,000	119,575,000,000	89,498,000,000	81,797,000,000	94,836,000,000	117,154,000,000	90,146,000,000	82,959,000,000	97,278,000,
2	CostOfRevenue	212,035,000,000	64,720,000,000	49,071,000,000	45,384,000,000	52,860,000,000	66,822,000,000	52,051,000,000	47,074,000,000	54,719,000,
3	GrossProfit	173,671,000,000	54,855,000,000	40,427,000,000	36,413,000,000	41,976,000,000	50,332,000,000	38,095,000,000	35,885,000,000	42,559,000,
4	OperatingExpense	55,013,000,000	14,482,000,000	13,458,000,000	13,415,000,000	13,658,000,000	14,316,000,000	13,201,000,000	12,809,000,000	12,580,000,

5 rows × 156 columns

The data preparation process included the importation of the CSV file mentioned in the previous section.

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 51 entries, 0 to 50
Columns: 156 entries, name to 09/30/1985
dtypes: object(156)
memory usage: 62.3+ KB
```

Upon inspection to verify that the CSV was properly imported, .info() was used to inspect the number of rows, columns, and their respective datatypes.

```
# Transposing the DataFrame to make columns rows, and rows columns
df_transposed = df.T

new_header = df_transposed.iloc[0]
df_transposed = df_transposed[1:]
df_transposed.columns = new_header

df_transposed.reset_index(inplace=True)
df_transposed.rename(columns={'index': 'Date'}, inplace=True)


df_transposed.head()
```

name	Date	TotalRevenue	\tOperatingRevenue	CostOfRevenue	GrossProfit	OperatingExpense	\tSellingGeneralAndAdministration	\tResearchAndDev
0	ttm	385,706,000,000	385,706,000,000	212,035,000,000	173,671,000,000	55,013,000,000	25,111,000,000	29,90
1	12/31/2023	119,575,000,000	119,575,000,000	64,720,000,000	54,855,000,000	14,482,000,000	6,786,000,000	7,69
2	09/30/2023	89,498,000,000	89,498,000,000	49,071,000,000	40,427,000,000	13,458,000,000	6,151,000,000	7,30
3	06/30/2023	81,797,000,000	81,797,000,000	45,384,000,000	36,413,000,000	13,415,000,000	5,973,000,000	7,44
4	03/31/2023	94,836,000,000	94,836,000,000	52,860,000,000	41,976,000,000	13,658,000,000	6,201,000,000	7,45

5 rows × 52 columns

The dataset needed to be transposed, making the dates the row values, and the names the column values.

```
# Converting numerical strings to floats, removing commas, and reducing values

for col in df_transposed.columns[1:]:
    df_transposed[col] = df_transposed[col].str.replace(',', '').astype(float)/ 1_000_000


df_transposed.head()
```

name	Date	TotalRevenue	\tOperatingRevenue	CostOfRevenue	GrossProfit	OperatingExpense	\tSellingGeneralAndAdministration	\tResearchAndDevelopm
0	ttm	385706.0	385706.0	212035.0	173671.0	55013.0	25111.0	299(
1	12/31/2023	119575.0	119575.0	64720.0	54855.0	14482.0	6786.0	76
2	09/30/2023	89498.0	89498.0	49071.0	40427.0	13458.0	6151.0	73(
3	06/30/2023	81797.0	81797.0	45384.0	36413.0	13415.0	5973.0	74
4	03/31/2023	94836.0	94836.0	52860.0	41976.0	13658.0	6201.0	74

5 rows × 52 columns

Upon verifying that the transposing was completed successfully, the numerical values were converted from objects to floats, commas were removed, and the values were reduced to improve readability.

```
# Keeping only the 'Date' and 'TotalRevenue' columns
df = df_transposed[['Date', 'TotalRevenue']]

df.head()
```

name	Date	TotalRevenue
0	ttm	385706.0
1	12/31/2023	119575.0
2	09/30/2023	89498.0
3	06/30/2023	81797.0
4	03/31/2023	94836.0

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 155 entries, 0 to 154
Data columns (total 2 columns):
 #   Column        Non-Null Count  Dtype
---  ------        --------------  -----
 0   Date          155 non-null    object
 1   TotalRevenue  155 non-null    float64
dtypes: float64(1), object(1)
memory usage: 2.6+ KB
```

```
null_values = df.isnull()
null_values.sum()
```

```
name
Date            0
TotalRevenue    0
dtype: int64
```

All columns not needed for the analysis were removed and I verified whether any null values were present.

```
from matplotlib.dates import DateFormatter

# Removing 'ttm'
df = df[df['Date'] != 'ttm']

# Converting the 'Date' column to datetime
df['Date'] = pd.to_datetime(df['Date'])

# Setting the 'Date' column as the index
df.set_index('Date', inplace=True)

# Sorting the DataFrame by the index (Date)
df.sort_index(inplace=True)
```

Lastly, row ‘ttm’ (trailing twelve months) was removed. The ‘Date’ column’s format was changed from ‘object’ to ‘datetime’, used as the index, and sorted.

D. Data Analysis (advantage & disadvantage):

My decision to use a SARIMA model was based on the characteristics of the data, which displayed a clear trend and seasonality, both of which are well-handled by SARIMA models. A SARIMA model can account for seasonality within the data and an ARIMA cannot (Patra, 2023). The initial ADF test confirmed the need for differencing, indicating that SARIMA was a fitting choice. An advantage of using a SARIMA model includes it being well-suited for the dataset because it can model both the non-stationary trend and the seasonality present in Apple's revenue data. One disadvantage of using a SARIMA model is it can become quite complex and may not capture sudden changes in the trend or seasonality or external factors which might limit their forecasting accuracy in the case of market shifts or new product introductions.

My analysis process included:

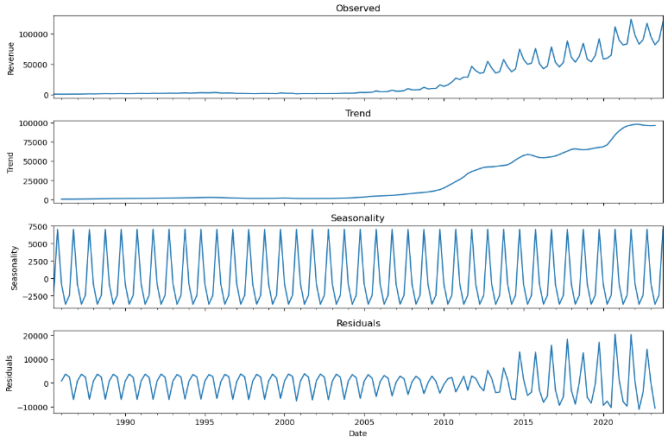
1. Stationarity Testing:

- I started by using the Augmented Dickey-Fuller (ADF) test to check for stationarity in the time series data. This is an essential step in time series analysis because most models require the data to be stationary.
- The high p-value suggested that the series is non-stationary.

```
ADF Statistic: 0.734467
p-value: 0.990501
```

2. Seasonal Decomposition:

- I then applied seasonal decomposition to separate the time series into trend, seasonal, and residual components. This helps to understand underlying patterns in the data.
- I also calculated the mean and standard deviation of the residuals, which aids in understanding the noise in the data after accounting for trend and seasonality.




```
Mean of residuals: -9.571826191322677
Standard deviation of residuals: 5938.660747067528
```

3. The SARIMA model:

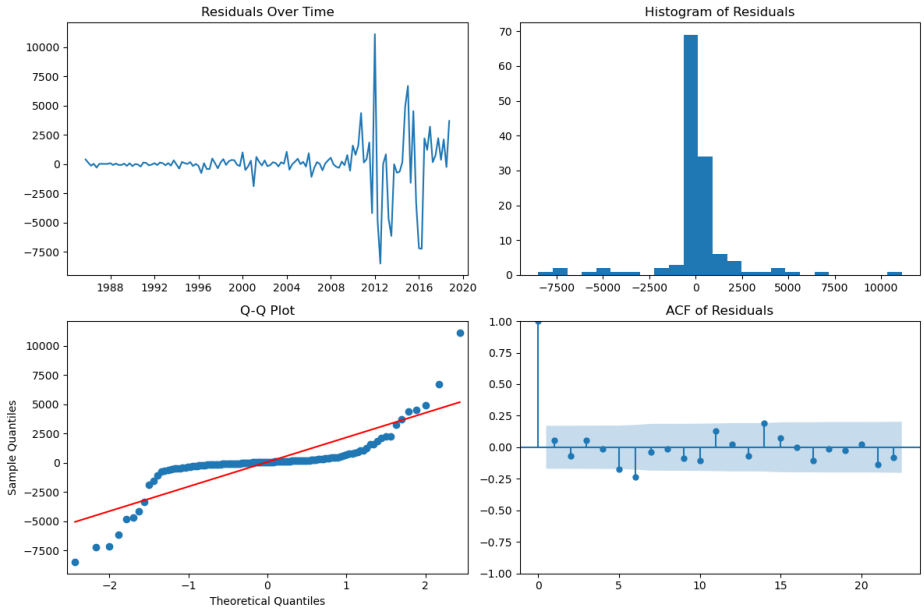
- Given the seasonality and non-stationarity in the data, I chose a Seasonal Autoregressive Integrated Moving Average (SARIMA) model, which is designed to handle both seasonality and non-stationarity.
- I used the 'auto_arima' function to obtain the best parameters for the SARIMA model.

```
Best model:  ARIMA(0,1,0)(3,1,0)[4]
Total fit time: 2.308 seconds
                                SARIMAX Results
==========================================================================================
Dep. Variable:                                 y   No. Observations:                  133
Model:             SARIMAX(0, 1, 0)x(3, 1, 0, 4)   Log Likelihood               -1164.699
Date:                           Sun, 07 Apr 2024   AIC                           2337.397
Time:                                   12:44:45   BIC                           2348.805
Sample:                                 09-30-1985 HQIC                          2342.033
                                      - 09-30-2018
Covariance Type:                              opg
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
ar.S.L4       -0.1171      0.043     -2.700      0.007      -0.202      -0.032
ar.S.L8        0.1771      0.089      1.986      0.047       0.002       0.352
ar.S.L12       0.4943      0.055      8.939      0.000       0.386       0.603
sigma2      5.048e+06   3.02e+05     16.703      0.000    4.46e+06    5.64e+06
===================================================================================
Ljung-Box (L1) (Q):                   0.43   Jarque-Bera (JB):               439.16
Prob(Q):                              0.51   Prob(JB):                         0.00
Heteroskedasticity (H):             325.32   Skew:                             0.06
Prob(H) (two-sided):                  0.00   Kurtosis:                        12.07
===================================================================================

Warnings:
[1] Covariance matrix calculated using the outer product of gradients (complex-step).
```

4. Model Fitting:

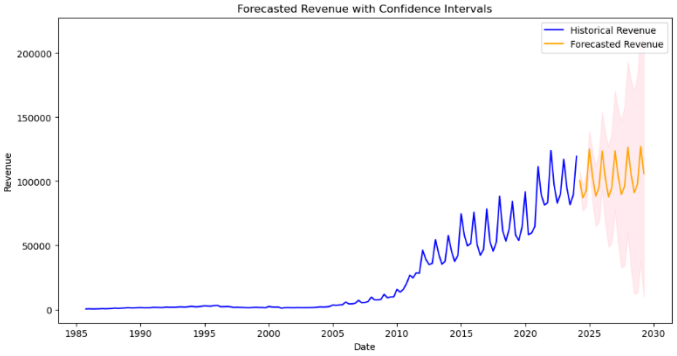
- After the SARIMA model was fitted to the data, the model parameters suggest that both seasonal and non-seasonal components are significant. The Ljung-Box test and the histogram of residuals suggest that the residuals are random, which is a good sign. However, the Q-Q plot indicates some deviation from normality, especially in the tails.



5. Forecasting:

- Using the SARIMA model, I forecasted future revenues and visualized these with historical data, showing confidence intervals.
- I created forecasts and accompanying confidence intervals for 21 quarters into the future.

	forecast	lower_conf	upper_conf
2024-03-31	100560.835646	93530.280597	107591.390696
2024-06-30	87177.751504	77235.045202	97120.457805
2024-09-30	92677.781766	80500.503215	104855.060316
2024-12-31	125028.775509	110967.665411	139089.885608
2025-03-31	103409.080289	84066.575343	122751.585235
2025-06-30	88422.613153	64958.807453	111886.418853
2025-09-30	95106.768864	68144.427380	122069.110348
2025-12-31	123521.037847	93464.665167	153577.410527
2026-03-31	102143.192417	67309.059978	136977.324856
2026-06-30	87697.419488	48666.047146	126728.791830
2026-09-30	94986.659785	52167.513705	137805.805865
2026-12-31	123820.476256	77522.414578	170118.537934
2027-03-31	103844.556369	51776.986280	155912.126459
2027-06-30	89726.729630	32468.077926	146985.381333
2027-09-30	96155.050569	34138.322707	158171.778431
2027-12-31	126539.296410	60104.396888	192974.195933
2028-03-31	105800.284312	32815.563664	178785.004960
2028-06-30	91159.941080	12166.631101	170153.251060
2028-09-30	97755.186774	13179.087343	182331.286205
2028-12-31	127033.871120	37221.343217	216846.399024
2029-03-31	106028.086861	9396.713327	202659.460396



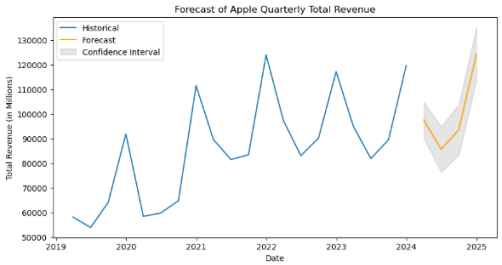
6. Error Metric Calculations:

- To evaluate the accuracy of my model, I calculated the error metrics such as MSE, RMSE, MAE, and MAPE.
- MAPE or the percentage of error resulted in 33.0%, accuracy is inferred as ~67%

```
Mean Absolute Error (MAE): 24082.046444487834
Mean Squared Error (MSE): 855268411.3690422
Root Mean Squared Error (RMSE): 29244.972411835886
Mean Absolute Percentage Error (MAPE): 33.0366258353486%
```

E. Results & Recommendations:

The forecasting plot showing the predicted revenue for future quarters along with confidence intervals displays wide confidence intervals suggesting considerable uncertainty in the forecasts, especially as the forecast increases. Because of this, I decided to try experimenting with focusing on short-term historical values and projections. Taking a look at the last five years and projecting a year into the future yielded the following results:




```
Mean Absolute Error (MAE): 3804.4542257005705
Mean Squared Error (MSE): 15017110.674127325
Root Mean Squared Error (RMSE): 3875.1916951458447
Mean Absolute Percentage Error (MAPE): 3.9749686522290744%
```

The new SARIMA model results show an error percentage of 3.97% or ~96% accuracy. These results suggest the model has more proficiency in short-term forecasting, which can be a

limitation. Forecasting too far into the future creates uncertainty that can be due to unexpected changes in the market. A recommended course of action would be to use the SARIMA model as a starting point for forecasting revenue, forecasting mainly into the near future. However, the model should always only be used as a guide, not a prediction, and combined with business intuition and other market insights. Keeping the previous statements in mind, two directions for future study are as follows:

1. Incorporate external factors such as consumer sentiment, competitor strategies, economic indicators, or product launch impact to create a more powerful model in accounting for external shocks or trends.
2. Consider pairing other forecasting techniques, or machine learning models, with the SARIMA model to leverage the strengths of different approaches and potentially improve accuracy.

F. Sources:

GfG. (2023, June 12). SAS vs R vs Python. GeeksforGeeks. Retrieved April 3, 2024, from https://www.geeksforgeeks.org/sas-vs-r-vs-python/

Patra, T. D.-D. (2023, September 24). *Sarima vs Arima for Timeseries Analysis Model*. Medium. Retrieved April 3, 2024, from https://dhirajpatra.medium.com/sarima-vs-arima-for-timeseries-analysis-model-a600ab544b1f

SudoPurge. (2021, March 16). *Python vs. R for data science*. Medium. Retrieved April 3, 2024, from https://towardsdatascience.com/python-vs-r-for-data-science-cf2699dfff4b

Vlastelica, R., & Bloomberg. (2023, June 30). *Apple just made history by becoming the first company with a $3 trillion market value-'all powered by a device people look at 4 hours a day'*. Fortune. Retrieved April 3, 2024, from https://fortune.com/2023/06/30/apple-history-3-trillion-market-value/

Yahoo! (2024, April 1). *Apple Inc. (AAPL) income statement*. Yahoo! Finance. Retrieved April 1, 2024, from https://finance.yahoo.com/quote/AAPL/financials