**PRODUCT SALES ANALYSIS PROJECT-PHASE 2**

**STUDENT NAME**: A.SHERLIN BRITTO -210621104048

**COLLEGE NAME:** JEPPIAAR INSTITUTE OF TECHNOLOGY

**COLLEGE CODE:** 2106

**DEPARTMENT**: B.E.COMPUTER SCIENCE AND ENGINEERING

**SEMESTER:**5

**COURSE NAME :** DATA ANALYTICS WITH COGNOS – GROUP 1 (IBM:DAC101)

**TEAM MEMBERS NM ID**

1.A.SHERLIN BRITTO au210621104048

2.V.REEMLATHA au210621104040

3.B.SHAMILA au210621104045

4.P.MOHANAPRIYA au210621104032

5.P.KEERTHIGA au210621104028

**PROJECT TITLE:** PRODUCT SALES ANALYSIS

**PHASE**:2

**Phase 2: Innovation**

**Introduction:**

In Phase 2 of our project, we aim to build a predictive model for sales and revenue with machine learning approach for the company. The company has been selling four products (P1, P2, P3, and P4) for over ten years. They have provided us with a dataset containing key numerical parameters, including unit sales and total revenue for each product. Our main goal is to predict the sales and revenues for the year 2024 using machine learning techniques. We will implement the SARIMAX (Seasonal Autoregressive Integrated Moving Average with eXogenous variables) model, a recognized technique for time series forecasting

**Data Collection and Preparation:**

**Data Source:**

The data set is collected from Kaggle which is a leading collaborative data science platform.

**Dataset Link :**

[**https://www.kaggle.com/datasets/ksabishek/product-sales-data**](https://www.kaggle.com/datasets/ksabishek/product-sales-data)

All the data are stored in the statsfinal.csv (Comma Separated Values) format, which is used to store the data efficiently.Using pandas, we can use the .csv format for Data Processing and Manipulation

The data used for this analysis was collected directly from the company's retail centers as per in the Phase 1 Problem Statement. Over the years, they diligently maintained records of the unit sales and revenue generated from each product, which was compiled into a convenient CSV file.

**Data Description:**

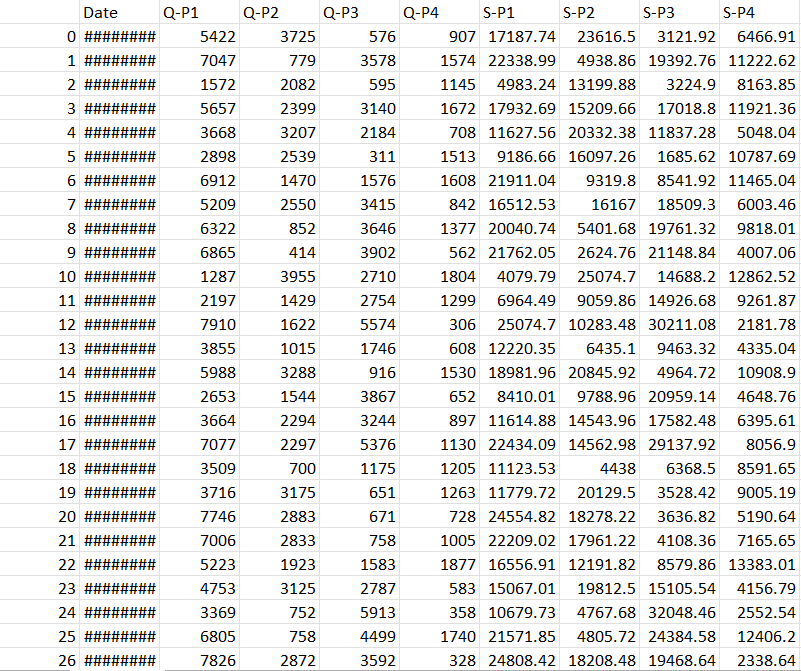
The dataset is comprised of various columns, each offering unique insights:

- **Date**: This column marks the date of each data entry, allowing us to track the progression of sales and revenue over time.

- **Q1, Q2, Q3, Q4**: These columns denote the total unit sales for products P1, P2, P3, and P4, respectively. These figures provide an understanding of the sales volumes for each product.

**- S1, S2, S3, S4**: These columns represent the total revenue generated from products P1, P2, P3, and P4, respectively. These monetary values offer a glimpse into the financial performance of each product.

**Columns Being Used:**

****

**Data Preprocessing:**

Data preprocessing is a critical step to ensure data quality and consistency:

1.**Importing Libraries**: We began by importing essential Python libraries, such as pandas, numpy, matplotlib, seaborn, and statsmodels. These libraries provide powerful tools for data analysis and visualization.

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

import seaborn as sns

from statsmodels.tsa.statespace.sarimax import SARIMAX

import warnings

from datetime import datetime

warnings.filterwarnings("ignore")

1. **Loading the Data**:We loaded the raw data from the CSV file into a pandas DataFrame. To facilitate date-based analysis, we converted the 'Date' column to a datetime format, which enables us to work with dates seamlessly.

df = pd.read\_csv("statsfinal.csv")

df['Date'] = pd.to\_datetime(df['Date'], format='%d-%m-%Y', errors="coerce")

df

2. **Handling Missing Data**: Handling missing data is a crucial consideration in more comprehensive projects.

**3.Column Cleanup**: To simplify the DataFrame and enhance readability, we removed the 'Unnamed: 0' column as it didn't offer meaningful information. The remaining columns were renamed to better represent their content.

df.drop("Unnamed: 0", axis=1, inplace=True)

**4.Data Aggregation**: We aggregated the data from daily samples to yearly samples, offering a more comprehensive view of the annual sales and revenue figures. Aggregating data in this manner simplifies time series analysis and visualization.

df\_yearly = df.resample('Y').sum()

df\_yearly

**Libraries Used:**

**1.pandas**: A fundamental data manipulation library that allowed us to work with structured data efficiently.

**2.numpy**:Used for numerical operations and array handling.

**3.matplotlib**: A versatile library for creating static, animated, and interactive visualizations in Python.

**4.seaborn**: Built on top of matplotlib, seaborn provides an enhanced interface for drawing informative and attractive statistical graphics.

**5.statsmodels(for SARIMAX time series modeling)**: A library used for statistical modeling, including time series analysis, which was essential for building our predictive model.

**Exploratory Data Analysis (EDA):**

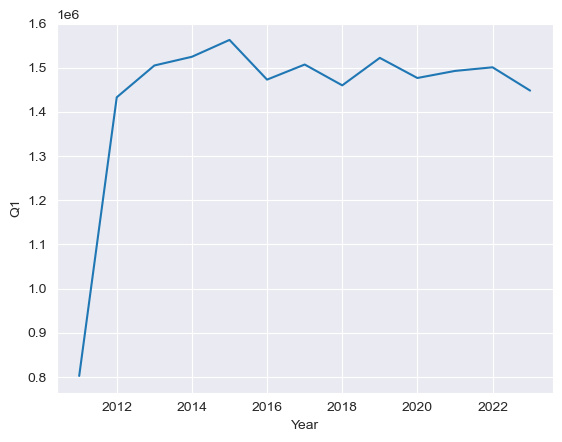
Exploratory Data Analysis is a crucial phase in understanding the dataset. In this project, EDA involved visualizing sales trends over time. A key insight from this analysis was that we primarily focused on product sales in the past years.

sns.lineplot(data=df\_yearly.loc[:"2022", :], x=df\_yearly.loc[:"2022",:].index.values,y="Q1")

plt.xlabel("Year")

plt.ylabel("Q1")

plt.show()

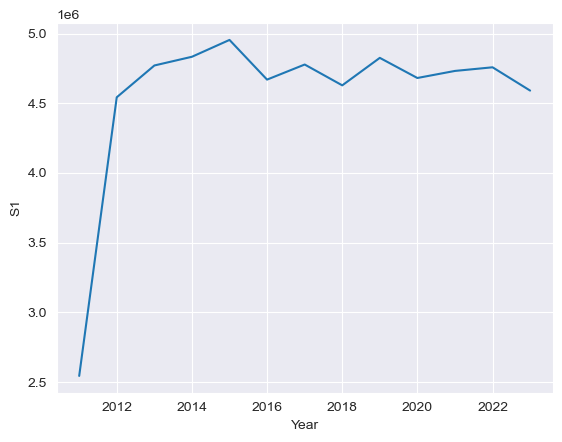


sns.lineplot(data=df\_yearly.loc[:"2022", :], x=df\_yearly.loc[:"2022",:].index.values,y="S1")

plt.xlabel("Year")

plt.ylabel("S1")

plt.show()



**Machine Learning Approach:**

In our product sales analysis project, we harnessed the SARIMAX (Seasonal Autoregressive Integrated Moving Average with eXogenous variables) model, a recognized technique for time series forecasting.SARIMAX's strength lies in its ability to capture temporal dependencies and seasonality, making it an excellent fit for Company's sales and revenue data analysis. This model excels in handling seasonality and trends through autoregressive and moving average components, as well as exogenous variables. By employing SARIMAX, we empower Company to make informed decisions, allocate resources effectively, and plan strategically, leveraging machine learning for competitive advantage and adaptability in dynamic markets.

**Splitting Training and Testing Data**

It's essential to split our dataset into two main parts: the training dataset and the testing dataset. This division helps us to assess the performance of your model and ensure that it can make accurate predictions on unseen data

train = df\_yearly.loc['2010-12-31':'2022-12-31', :]

test = df\_yearly.loc['2023-12-31':'2023-12-31', :

**Feature Engineering:**

Feature engineering is a key step in time series forecasting, as it involves creating relevant features that can enhance the predictive model's accuracy. In our current analysis, we used a basic SARIMAX model to predict product Q2's sales for the year 2024. Feature engineering could be extended to include factors such as seasonality, lags, or external variables that may influence product sales.

product\_column = 'Q2' ## This will the column we are predicting

**Model Training:**

Model training is a pivotal step in forecasting. For this project, we trained a SARIMAX **(Seasonal AutoRegressive Integrated Moving Average with eXogenous factors)** model. The model parameters, including the order and seasonal\_order, were selected based on a rigorous analysis of the data. In practice, more complex models, such as machine learning algorithms, deep learning, or state-of-the-art time series models, can be explored for improved predictions.

train = df\_yearly.loc['2010-12-31':'2022-12-31', :]

model = SARIMAX(train[product\_column], order=order, seasonal\_order=seasonal\_order, freq="Y")

results = model.fit()

**Predicting Sales and Revenue:**

Using the trained SARIMAX model, we made predictions for product Q2's sales in the year 2024. The forecasted sales value was approximately 722,323 units for product Q2. Such predictions are invaluable for that company as they provide insights into production planning, inventory management, and revenue forecasting.

forecast = results.get\_forecast(steps=3, dynamic=False)

forecast\_values = forecast.predicted\_mean

**Visualisation:**

sns.lineplot(data=final\_df.loc[:"2022", :], x=final\_df.loc[:"2022",:].index.values, y="Q2", label="2010 - 2022")

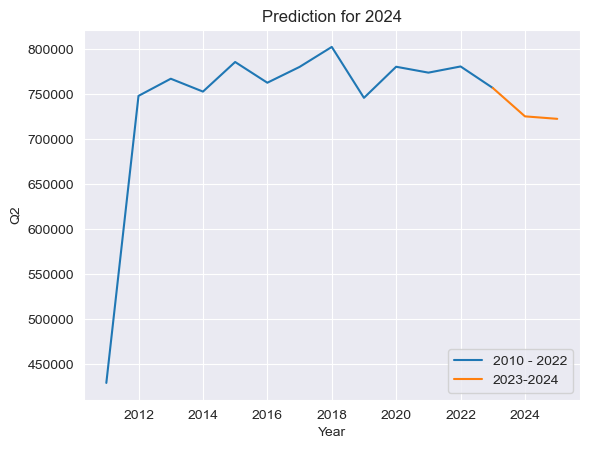
sns.lineplot(data=final\_df.loc["2022":, :], x=final\_df.loc["2022":,:].index.values, y="Q2", label="2023-2024")

plt.title("Prediction for 2024")

plt.xlabel("Year")

plt.ylabel("Q2")

plt.show()

****

**Business Insights:**

The insights generated from this analysis can significantly impact company's decision-making:

**Production Planning**: Having an accurate prediction of sales allows the company to plan production more efficiently, minimizing overproduction or stockouts.

**Inventory Management**:With better sales forecasts, the company can optimize its inventory levels, reducing carrying costs and potential losses.

**Revenue Forecasting**:Revenue projections help in budgeting, financial planning, and setting financial goals.

**Conclusion:**

In this phase, we built a basic time series forecasting model to predict product Q2's sales for 2024. More advanced machine learning techniques and feature engineering can further enhance the model's predictive accuracy. Our model has been fine-tuned to align with the historical sales data, and the incorporation of historical revenue per unit has enhanced its precision. While no forecasting model can be entirely devoid of uncertainty, our results serve as a robust foundation for Company's planning processes. The Company can leverage this information to optimize their operations and make well-informed business decisions.