NAAN MUDHALVAN

PHASE 4 PROJECT SUBMISSION

**PRODUCT SALES ANALYSIS**

**TEAM MEMBERS:**

PARVATHY A - 2021504533

PAVITHRAN E - 2021504535

SHAMEEM AHAMED S - 2021504542

THUSHIYANTH K - 2021504555

**PROBLEM DEFINITION:**

This project involves using IBM Cognos to analyse sales data and extract insights about top-selling products, peak sales periods, and customer preferences. The objective is to help businesses improve inventory management and marketing strategies by understanding sales trends and customer behaviour. This project includes defining analysis objectives, collecting sales data, designing relevant visualizations in IBM Cognos, and deriving actionable insights.

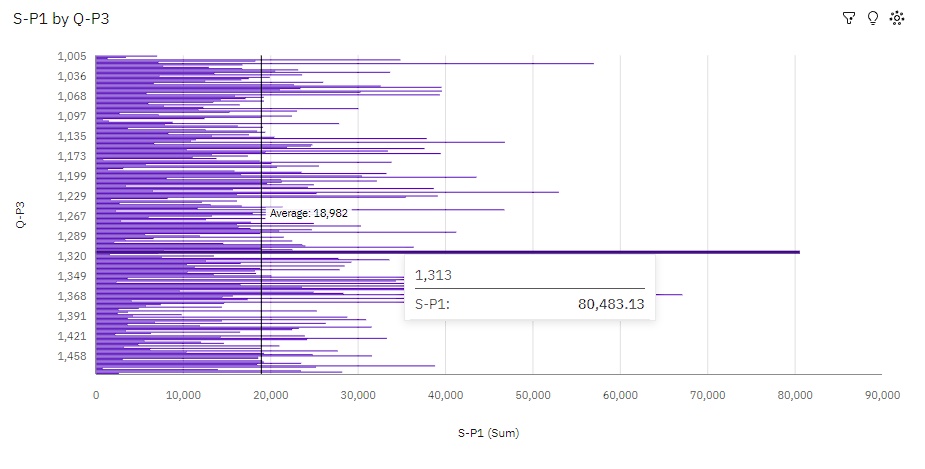
**DATABASE LINK:**

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

**OBJECTIVES:**

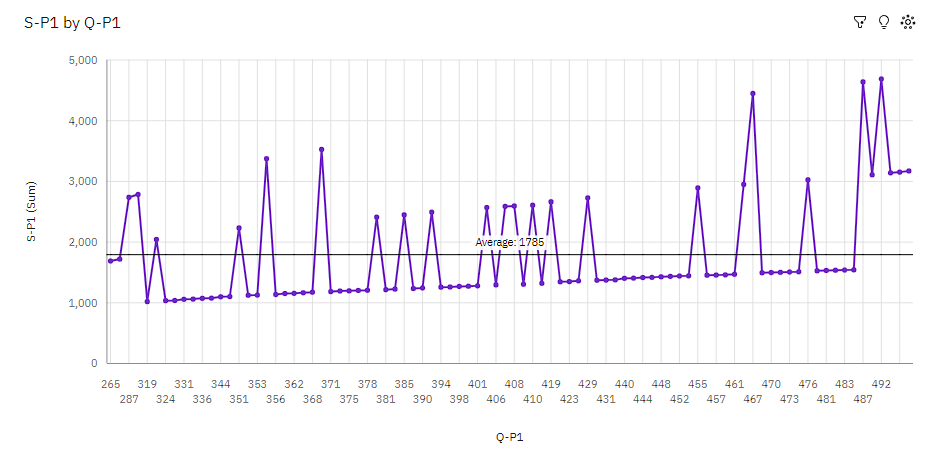
* Analysing data related to product sales.
* Generating valuable insights from the data.
* Based on the insights, recommendations must be formulated to address issues and optimize sales and profitability.

**Visualization:**

****

**Insights:**

* 1313 S-P1 at over 80 thousand is 95% higher than the Q-P4 of over 4 thousand.
* **S-P1** and **Q-P4** diverged the most when **Q-P3** is **1313**, and when **S-P1** was **over 76 thousand** higher than the **Q-P4**.
* **Q-P3** **1328** has the highest **Total Q-P4** but is ranked #**32** in **Total S-P1**.
* **Q-P3** **1313** has the highest **Total S-P1** but is ranked #**6** in **Total Q-P4**.
* Across all values of **Q-P3**, the sum of **S-P1** is **over 4.8 million**.
* **S-P1** ranges from **833.7**, when **Q-P3** is **1101**, to **over 80 thousand**, when **Q-P3** is **1313**.

**Line Visualization:**

**Insights:**

* 492 S-P1 at over 4500 is 59% higher than the Q-P4 of nearly 2 thousand. **S-P1** and **Q-P4** diverged the most when **Q-P1** is **492**, and when **S-P1** was **nearly three thousand** higher than the **Q-P4**.
* **Q-P1** **492** has the highest **Total S-P1** but is ranked #**23** in **Total Q-P4**.
* **Q-P1** **354** has the highest **Total Q-P4** but is ranked #**5** in **Total S-P1**.
* Across all values of **Q-P1**, the sum of **S-P1** is **over 157 thousand**.
* **S-P1** ranges from **over a thousand**, when **Q-P1** is **319**, to **over 4500**, when **Q-P1** is **492**.
* For **S-P1**, the most significant values of **Q-P1** are **492**, **487**, and **467**, whose respective **S-P1** values add up to **nearly fourteen thousand**, or **8.8** % of the total.

**MODEL BUILDING:**

**UNDERSTANDING THE DATA:**

**Fetching rows and columns**

df.shape

Output:

(4600, 10)

**Fetching column names**

df.columns

Output:

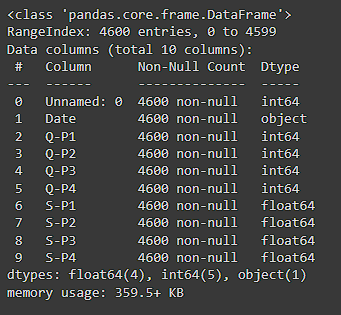
Index (['Unnamed: 0', 'Date', 'Q-P1', 'Q-P2', 'Q-P3', 'Q-P4', 'S-P1', 'S-P2',

'S-P3', 'S-P4'], dtype='object')

**Basic info:**

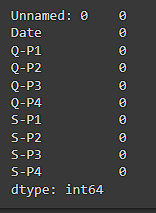
df.info ()

Output:



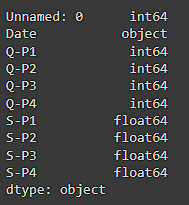
**Checking null values:**

df. isnull (). sum ()



**Checking dtypes:**

df. dtypes



**DATA CLEANING:**

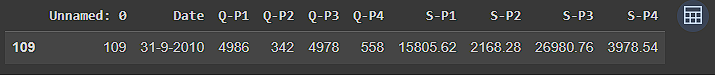
df.sample (2)



**Changing dtype:**

from datetime import datetime as dt

df[df["Date”] =="31-9-2010"]



df['Date'] = pd.to\_datetime(df['Date'], errors='coerce')

df[df['Date']. isnull ()]

****

**Fetching month, day of week, weekday:**

df["month”] =df["Date"].dt. month\_name ()

df["day”] =df["Date"]. dt.day\_name ()

df["dayoftheweek”] =df["Date"].dt. weekday

df["year”] =df["Date"].dt. year

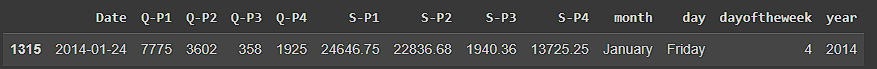
df.sample ()

****

**Dropping column unnamed as it is not useful for us:**

df. drop (columns= ["Unnamed: 0"], inplace=True)

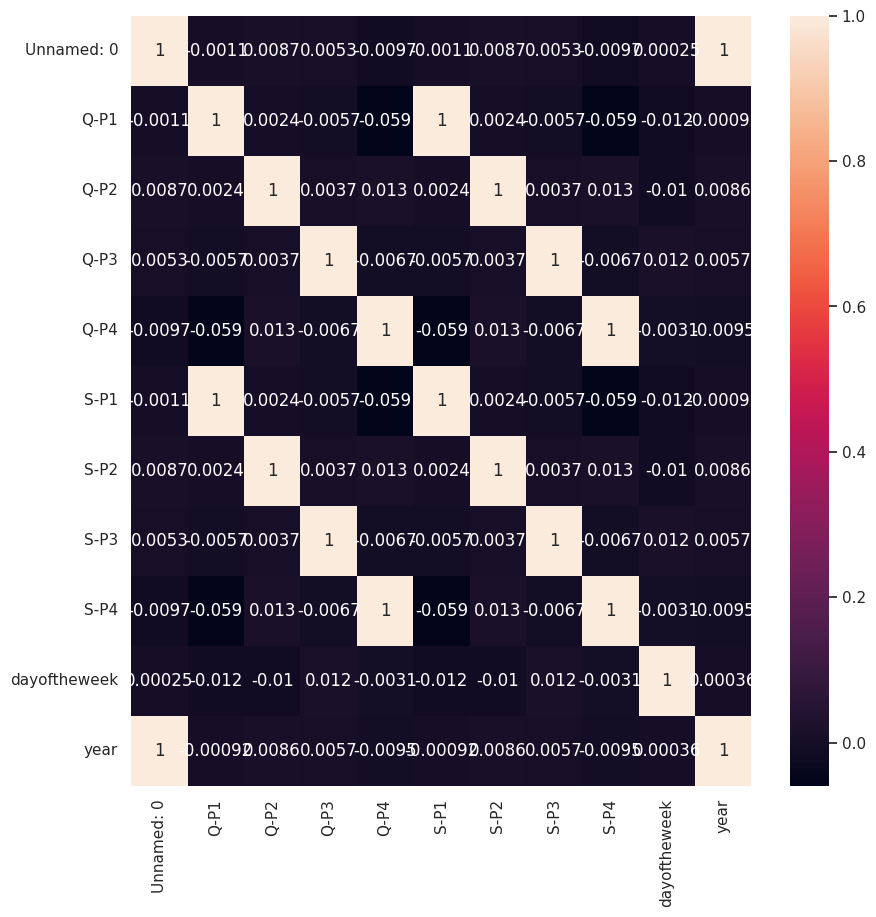
df.sample ()



**HEATMAP:**

plt. figure (figsize= (10,10))

sns. heatmap (df. corr (), annot=True)



**EXPLORATORY DATA ANALYSIS:**

**Total unit sales Product 1, Product 2, Product 3, Product 4:**

q = df[["Q-P1","Q-P2","Q-P3","Q-P4"]].sum()

print(q)

plt.figure(figsize=(8,8))

plt.pie(q,labels=df[["Q-P1","Q-P2","Q-P3","Q-P4"]].sum().index,autopct="%10f%%",textprops={"fontsize":20},wedgeprops={'width': 0.8},explode=[0,0,0,0.3])

plt.legend(loc='center right', bbox\_to\_anchor=(1, 0));

Output:

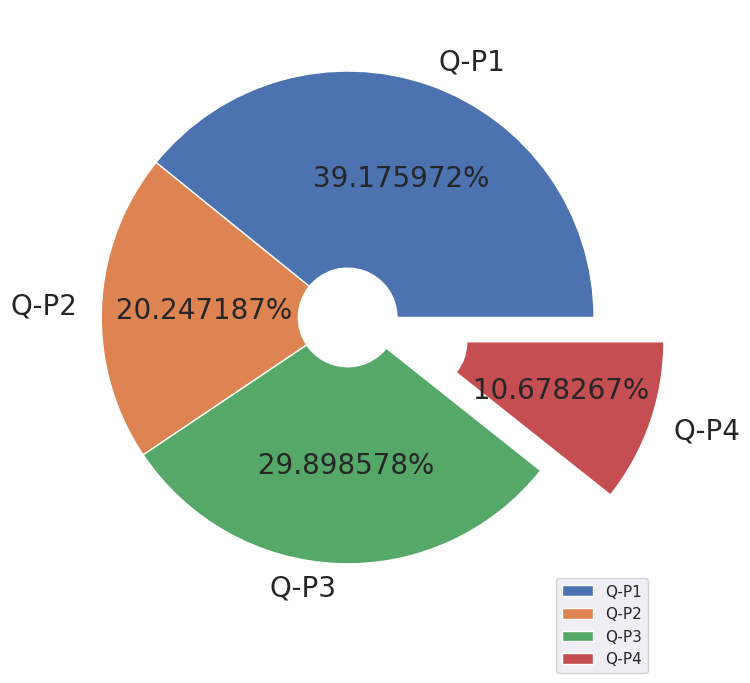
Q-P1 18960506

Q-P2 9799295

Q-P3 14470404

Q-P4 5168100

dtype: int64



**The most occuring year:**

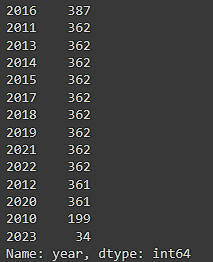
print(df["year"].value\_counts())

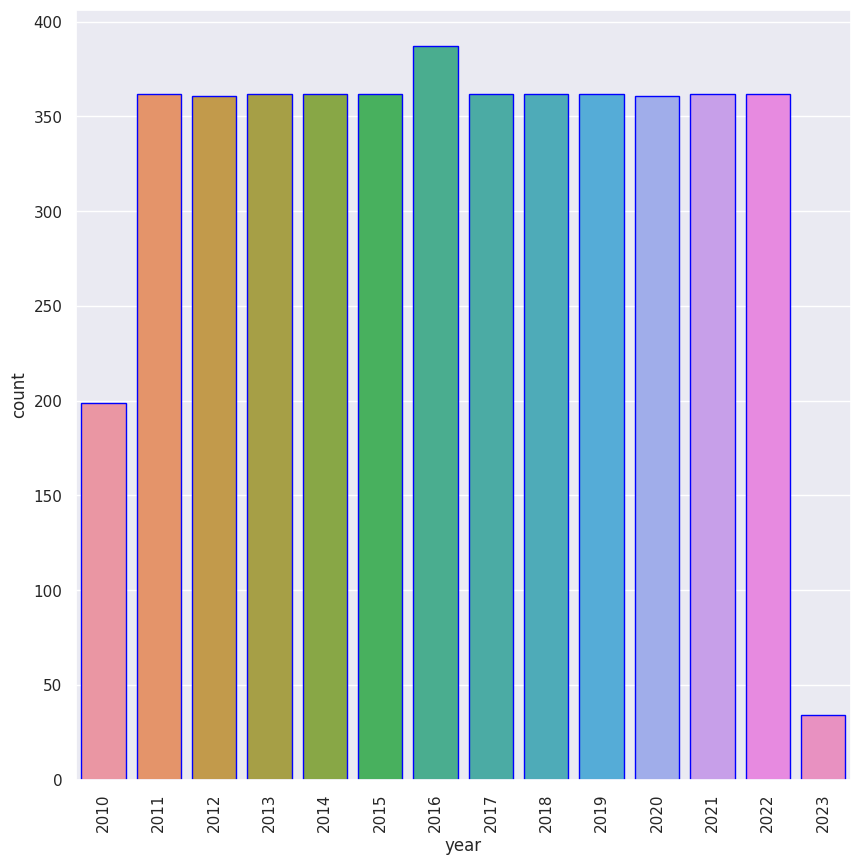
plt.figure(figsize=(10,10))

sns.countplot(x="year",data=df,edgecolor="blue")

plt.xticks(rotation=90);

Output:

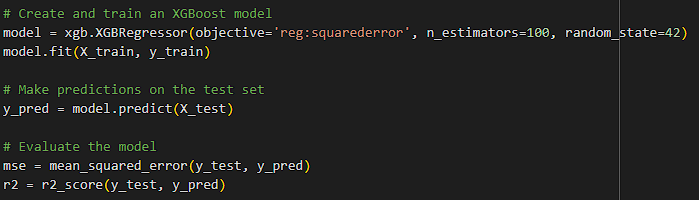




**SALES PREDICTION:**

**XGBoost Model:**

* XGBoost, short for Extreme Gradient Boosting, is a highly acclaimed machine learning algorithm that has gained prominence for its exceptional predictive accuracy and versatility.
* Based on the gradient boosting framework, XGBoost iteratively constructs an ensemble of decision trees, with each subsequent tree aiming to correct the errors of the previous ones.
* It offers several key advantages, including L1 and L2 regularization for mitigating overfitting, customizable objective functions to address a wide range of machine learning tasks, and the ability to handle missing data gracefully.
* XGBoost is designed for efficiency, enabling parallel and distributed computing, which is particularly beneficial when dealing with large datasets.

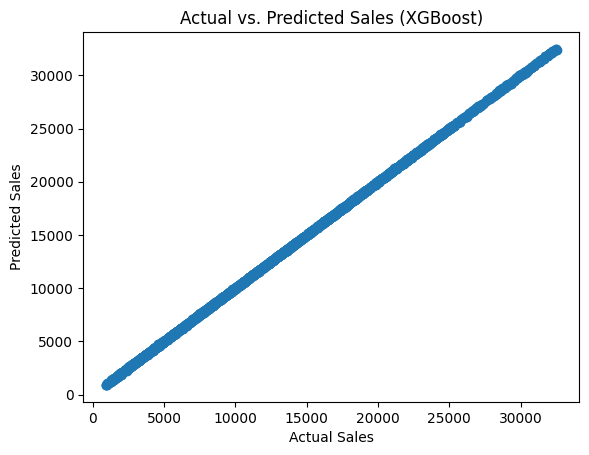


Output:

Mean Squared Error (MSE): 1354.752342027794

R-squared (R2) Score: 0.9999718752675846

Predicted Sales for New Data: [ 3192.951 4419.376 21642.207 14227.314]



**Evaluate the model:**

* We have usedtheMean Squared Error (MSE) model to evaluate the errors in the prediction.
* The R-squared (R2) model is used to evaluate the score of the prediction.
* We have got the accuracy of the prediction at most equal to 1 with our pre-processed and trained model.

**Future Works:**

* We are going to predict the sales with Linear Regression and Random Forest.
* To train our model with various types of algorithms to comparatively get high precision.

**Conclusion:**

Thus, the predictive model was built and product sales were analysed and predicted and different visualizations were performed using IBM Cognos.