INTRODUCTION

Product sales analysis is an indispensable tool for modern businesses striving for success in an increasingly competitive market. By delving insights into customer preferences, market trends, and the performance of their products. This project explores the significance of product sales analysis, its key components, and its vital role in shaping business strategies.

DESCRIPTION

Analyse sales data to identify top-selling products, peak sales periods, and customer preferences, aiding in inventory management and marketing strategies.

IBM Cognos tool

The IBM Cognos tool is used for analysing the files such as csv files and other files to visualize data from them.

TOP SELLING PRODUCTS

1. Average of product P1 is nearly 3000
2. Average of product P2 is nearly 2000
3. Average of product P3 is nearly 4000
4. Average of product P4 is over 1000
5. Comparing the 4 products, product P3 has sold more than other products.
6. So, product P3 can be considered as Top selling product.
7. The below graph represents the total sales units for 4 products on every year(2011-2023) .
8. P3>P1>P2>P4 is the order of top selling products.

PEAK SALES PERIOD

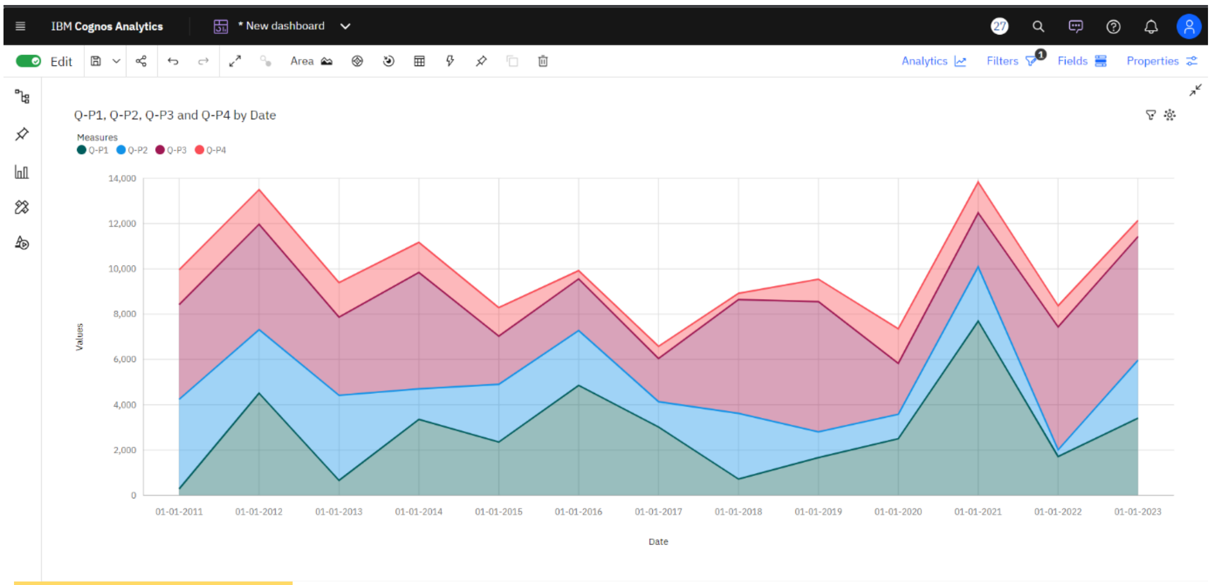
1. From the Below graph we can assume that the period 2020-2022 has the highest peak sales period for the four products .

2. Next to that 2011-2013 has the highest sales period.
3. Year 2022 has the highest sales year and 2011 is more or less has similar sales as year 2022.

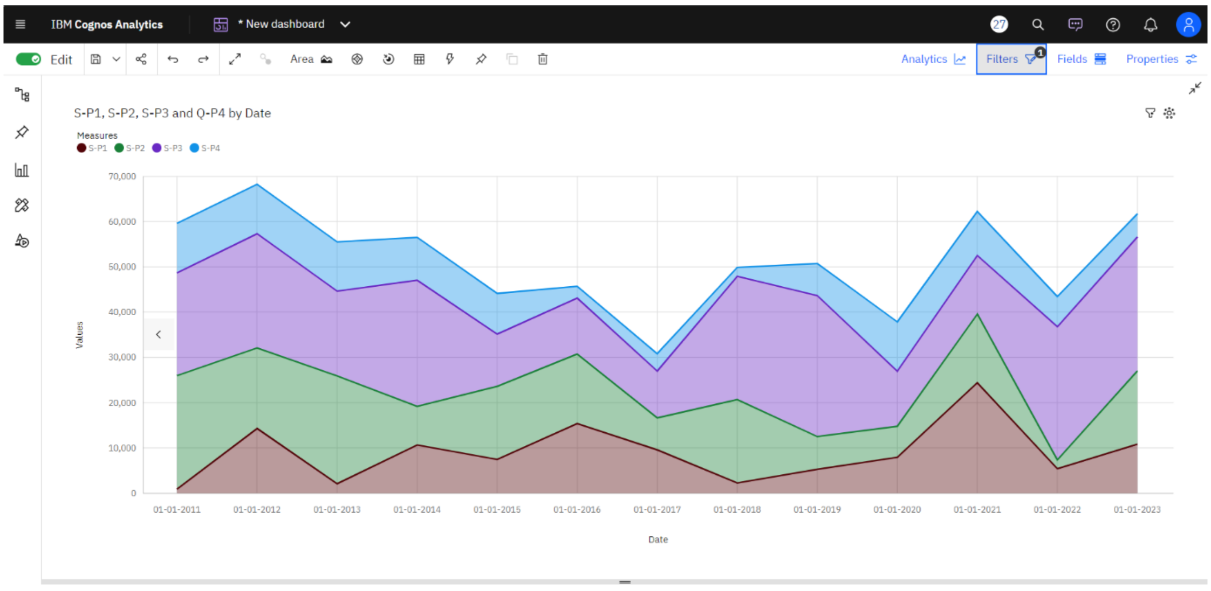
CUSTOMER PREFENCES

Product P3 has more customer preferences.

TOTAL SELLING UNIT



TOTAL REVENUE



DATA CLEANING AND PROCESSING

The data cleaning and data processing process done in python using the below code.

# Import necessary libraries

```
import pandas as pd
from sklearn.preprocessing import StandardScaler
from sklearn.impute import SimpleImputer
```

# Load the dataset

```
df = pd.read_csv('statsfinal.csv')
```

# Step 1: Handle Missing Data ,Identify and handle missing values

```
df.dropna(inplace=True)
```

# Drop rows with missing values

#Alternatively, use imputation for missing values

```
imputer = SimpleImputer(strategy='mean')
df['column_with_missing_values'] =
imputer.fit_transform(df[['column_with_missing_values']])
```

# Step 2: Remove Duplicate Records

```
df.drop_duplicates(inplace=True)
```

# Step 3: Standardize Data

# Standardize numerical data using StandardScaler

```
scaler = StandardScaler()
df[['numeric_column1', 'numeric_column2']] =
scaler.fit_transform(df[['numeric_column1', 'numeric_column2']])
```

# Step 4: Encoding Categorical Variables

# Convert categorical variables to numerical using one-hot encoding

```
Sdf = pd.get_dummies(df, columns=['categorical_column'])
```