



KGiSL Institute Of Technology

NAAN MUDHALVAN

Project Title:

Project Sales Analysis

Team Members:

- 1.Ashika.S**
- 2.Dharshana.V**
- 3.Aswitha.M**
- 4.Dinesh Vishnu.S**

PROJECT DESCRIPTION:

PHASE-3 : Analysing On The Product-sales DataSet

OBJECTIVE:

STEPS:

IN GOOGLE COLAB NOTEBOOK:

- Mount the GoogleDrive
- Loading the Dataset to GoogleDrive
- Processing and cleansing the dataset
- Accuracy

#Mount:

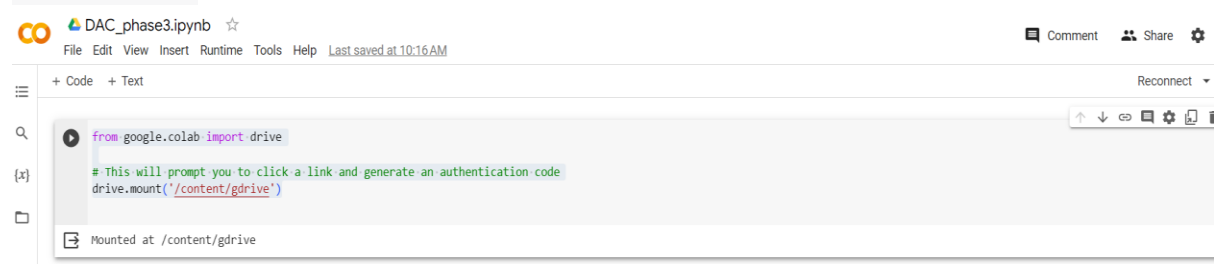
CODE:

```
from google.colab import drive

# This will prompt you to click a link and generate an authentication code

drive.mount('/content/gdrive')
```

OUTPUT:



```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

print(data.head())    # Display the first few rows of the dataset
print(data.info())    # Display data types and non-null counts
print(data.describe()) # Summary statistics
```

```
DAC_phase3.ipynb
File Edit View Insert Runtime Tools Help All changes saved
+ Code + Text
#Processing the data
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
print(data.head()) # Display the first few rows of the dataset
print(data.info()) # Display data types and non-null counts
print(data.describe()) # Summary statistics

0      Unnamed: 0      Date      Q-P1      Q-P2      Q-P3      Q-P4      S-P1      S-P2 \
0      0      13-06-2010      5422      3725      576      907      17187.74      23616.50
1      1      14-06-2010      7047      779      3578      1574      22358.99      4938.86
2      2      15-06-2010      1572      2682      595      1145      4983.24      11199.88
3      3      16-06-2010      5657      2399      3140      1672      17932.09      15209.66
4      4      17-06-2010      1668      3207      2184      708      11627.56      20132.18

0      S-P3      S-P4
0      3121.92      6466.94
1      19392.76      11222.62
2      3224.90      8163.85
3      17018.80      11921.36
4      11837.28      5048.04
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4600 entries, 0 to 4599
Data columns (total 10 columns):
#   Column      Non-Null Count  Dtype
--  --
0   Unnamed: 0      4600 non-null     int64
1   Date           4600 non-null     object
2   Q-P1           4600 non-null     int64
3   Q-P2           4600 non-null     int64
4   Q-P3           4600 non-null     int64
5   Q-P4           4600 non-null     int64
6   S-P3           4600 non-null     float64
7   S-P4           4600 non-null     float64
dtypes: float64(4), int64(5), object(1)
memory usage: 359.5+ KB

0      Unnamed: 0      Q-P1      Q-P2      Q-P3      Q-P4 \
count      4600.000000      4600.000000      4600.000000      4600.000000      4600.000000
mean      2299.500000      4121.849130      2130.281522      3145.740000      1123.500000
std      1328.049949      2244.271323      1089.783705      1671.832231      497.385676
min           0.000000      254.000000      251.000000      250.000000      250.000000
25%      1149.750000      2150.500000      1167.750000      1695.750000      696.000000
50%      2299.500000      4137.000000      2134.000000      3202.500000      1136.500000
75%      3449.250000      6072.000000      3070.250000      4569.000000      1544.000000
max      4599.000000      7998.000000      3998.000000      6000.000000      2000.000000

0      S-P1      S-P2      S-P3      S-P4
count      4600.000000      4600.000000      4600.000000      4600.000000
mean      13066.261743      13505.984848      17049.910800      8010.555000
std       7114.240094      6909.228687      9061.330694      3546.359869
min       805.180000      1591.340000      1355.000000      1782.500000
25%      6817.085000      7403.535000      9190.965000      4962.480000
50%      13114.290000      13529.560000      17357.550000      8103.245000
75%      19248.240000      19465.385000      24763.980000      11008.720000
max      25353.660000      25347.320000      32520.000000      14260.000000
```

```
DAC_phase3.ipynb
File Edit View Insert Runtime Tools Help All changes saved
+ Code + Text
Data columns (total 10 columns):
#   Column      Non-Null Count  Dtype
--  --
0   Unnamed: 0      4600 non-null     int64
1   Date           4600 non-null     object
2   Q-P1           4600 non-null     int64
3   Q-P2           4600 non-null     int64
4   Q-P3           4600 non-null     int64
5   Q-P4           4600 non-null     int64
6   S-P1           4600 non-null     float64
7   S-P2           4600 non-null     float64
8   S-P3           4600 non-null     float64
9   S-P4           4600 non-null     float64
dtypes: float64(4), int64(5), object(1)
memory usage: 359.5+ KB

0      Unnamed: 0      Q-P1      Q-P2      Q-P3      Q-P4 \
count      4600.000000      4600.000000      4600.000000      4600.000000      4600.000000
mean      2299.500000      4121.849130      2130.281522      3145.740000      1123.500000
std      1328.049949      2244.271323      1089.783705      1671.832231      497.385676
min           0.000000      254.000000      251.000000      250.000000      250.000000
25%      1149.750000      2150.500000      1167.750000      1695.750000      696.000000
50%      2299.500000      4137.000000      2134.000000      3202.500000      1136.500000
75%      3449.250000      6072.000000      3070.250000      4569.000000      1544.000000
max      4599.000000      7998.000000      3998.000000      6000.000000      2000.000000

0      S-P1      S-P2      S-P3      S-P4
count      4600.000000      4600.000000      4600.000000      4600.000000
mean      13066.261743      13505.984848      17049.910800      8010.555000
std       7114.240094      6909.228687      9061.330694      3546.359869
min       805.180000      1591.340000      1355.000000      1782.500000
25%      6817.085000      7403.535000      9190.965000      4962.480000
50%      13114.290000      13529.560000      17357.550000      8103.245000
75%      19248.240000      19465.385000      24763.980000      11008.720000
max      25353.660000      25347.320000      32520.000000      14260.000000
```

#CLEANSING THE DATA

```
import pandas as pd
data = data.drop_duplicates()
data.to_csv("cleaned_dataset.csv", index=False)
print(data.describe())
```

OUTPUT:

```
DAC_phase3.ipynb
File Edit View Insert Runtime Tools Help All changes saved
+ Code + Text
[ ]
Product Category
#cleansing the data
import pandas as pd
data = data.drop_duplicates()
data.to_csv("cleaned_dataset.csv", index=False)
print(data.describe())

0      Unnamed: 0      Q-P1      Q-P2      Q-P3      Q-P4 \
count      4600.000000      4600.000000      4600.000000      4600.000000      4600.000000
mean      2299.500000      4121.849130      2130.281522      3145.740000      1123.500000
std      1328.049949      2244.271323      1089.783705      1671.832231      497.385676
min           0.000000      254.000000      251.000000      250.000000      250.000000
25%      1149.750000      2150.500000      1167.750000      1695.750000      696.000000
50%      2299.500000      4137.000000      2134.000000      3202.500000      1136.500000
75%      3449.250000      6072.000000      3070.250000      4569.000000      1544.000000
max      4599.000000      7998.000000      3998.000000      6000.000000      2000.000000

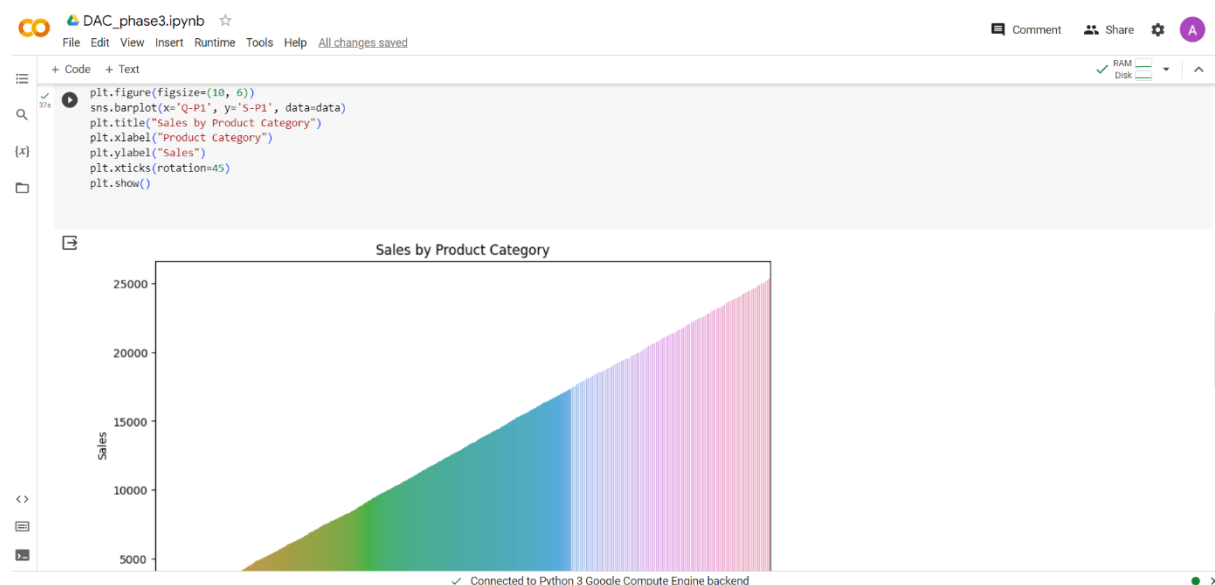
0      S-P1      S-P2      S-P3      S-P4
count      4600.000000      4600.000000      4600.000000      4600.000000
mean      13066.261743      13505.984848      17049.910800      8010.555000
std       7114.240094      6909.228687      9061.330694      3546.359869
min       805.180000      1591.340000      1355.000000      1782.500000
25%      6817.085000      7403.535000      9190.965000      4962.480000
50%      13114.290000      13529.560000      17357.550000      8103.245000
75%      19248.240000      19465.385000      24763.980000      11008.720000
max      25353.660000      25347.320000      32520.000000      14260.000000
```

#ANALYSIS ON DATASET:

CODE:

```
plt.figure(figsize=(10, 6))
sns.barplot(x='Q-P1', y='S-P1', data=data)
plt.title("Sales by Product Category")
plt.xlabel("Product Category")
plt.ylabel("Sales")
plt.xticks(rotation=45)
plt.show()
```

OUTPUT:



ACCURACY:

CODE:

#LOGISTIC REGRESSION

```
from sklearn.datasets import make_classification
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score
from sklearn.linear_model import LogisticRegression
nb_samples = 1000
x, y = make_classification(n_samples=nb_samples, n_features=2,
n_informative=2, n_redundant=0, n_clusters_per_class=1)
xtrain, xtest, ytrain, ytest = train_test_split(x, y, test_size=0.2,
random_state=42)
model = LogisticRegression()
```

```
model.fit(xtrain, ytrain)
```

OUTPUT:

```
max 25353.660000 25347.320000 32520.000000 14260.000000
```

```
#logistic regression
from sklearn.datasets import make_classification
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score
from sklearn.linear_model import LogisticRegression
nb_samples = 1000
x, y = make_classification(n_samples=nb_samples, n_features=2, n_informative=2, n_redundant=0, n_clusters_per_class=1)
xtrain, xtest, ytrain, ytest = train_test_split(x, y, test_size=0.2, random_state=42)
model = LogisticRegression()
model.fit(xtrain, ytrain)
```

```
LogisticRegression
LogisticRegression()
```

CODE:

#ACCURACY OF THE DATA

```
print(accuracy_score(ytest, model.predict(xtest)))
```

OUTPUT:

```
#Accuracy of the data
print(accuracy_score(ytest, model.predict(xtest)))
```

```
0.88
```