Data Analytics Design for Product Sales Analysis with IBM Cognos

Title: Innovation Phase_2

Task: Import the dataset and perform data cleaning & data analysis

1. Introduction:

Data Analytics with Cognos Product Sales Analysis provides organizations with valuable insights into their sales performance. However, to enhance this analytical capability, incorporating machine learning algorithms is essential. This document explores how machine learning can be integrated to predict future sales trends and customer behaviors more accurately.

2. Problem Statement:

In traditional sales analysis, past data is used to make informed decisions about future sales and customer behaviors. While this approach is valuable, it is limited in its ability to adapt to dynamic market conditions and emerging trends. Machine learning algorithms offer the potential to predict future sales trends and customer behaviors more accurately, thereby empowering organizations to make proactive decisions.

3. Notebook

Types of Problems in Data Science

1. Classification
2. Regression
3. Clustering
4. Natural Language Processing
5. Recommendation Systems
6. Image Recognition
7. Big Data and Distributed Computing

Classification

Involves categorizing data points into predefined classes or categories.

Eg: Classifying emails as spam or not spam, identifying whether a patient has disease or not, categorizing images of animals into species

Concepts for classification:

Logistic Regression: Statistical model that predicts the probability of a binary outcome(eg:yes/no)

Decision Trees: Tree Like structure that make decisions by evaluating features at each node

Random Forests: Ensembles of multiple decision trees to improve accuracy and reduce overfitting.

Support Vector Machines (SVM): Powerful algorithm for bianry and multiclass classification by finding the optimal hyperplane taht best seperates classes.

Neural Networks: Deep Learning Models composed of layers of interconnected neurons, capable of handling complex classification tasks.

Regression

Involves preidcting a continuous numerical value. Eg: Predicting housing prices based on features, forecasting future sales, or estimating the temparature based on Historical Data.

Concepts for regression:

Linear Regression: Statistical technique that models the relationship betweena dependent variable and one or more independent variable

Polynomial Regression:Extends linear regression by fitting a polynomial equation to the data.

Ridge Regression and lasso Regression: Techniques that add regularization to linear regression models to prevent overfitting.

Neural Networks: Deep Learning Models composed of layers of interconnected neurons, capable of handling complex classification

Clustering

Involves grouping of similar data points without predefined categories.

Eg: Customer Segmentation for marketing or clustering documents by topic

Concepts for Clustering:

K-Means Clustering: A partitioning method that divides data into K clusters based on similarity.

Hierarchical Clustering: Builds a tree-like hierarchy of clusters, useful for exploring data at different levels.

DBSCAN(Density-Based Spatial Clustering of Applications with Noise): Clusters data points based on their density, suitable for irregularly shaped clusters.

Notebook Link:

https://colab.research.google.com/drive/15_IxEf7I-775x_aI30dNyvpzFauJgO83#scrollTo=I5nDp_Ww5zcO

The Data import

https://drive.google.com/file/d/1-SGBoro0m1_mbcUEkxu7K4uzeUBG4mQC/view?usp=sharing

About Dataset

Greetings , fellow analysts ! REC corp LTD. is a small-scale business venture established in India. They have been selling FOUR PRODUCTS for OVER TEN YEARS . The products are P1, P2, P3 and P4. They have collected data from their retail centers and organized it into a small csv file , which has been given to you. The excel file contains about 8 numerical parameters :

- Q1- Total unit sales of product 1
- Q2- Total unit sales of product 2
- Q3- Total unit sales of product 3
- Q4- Total unit sales of product 4
- S1- Total revenue from product 1
- S2- Total revenue from product 2
- S3- Total revenue from product 3
- S4- Total revenue from product 4

Understanding the Data

Fetching rows and columns

fetching column names

Basic info

Checking null values

Checking Dtypes

Basic statistical info

CODE

```
df.shape
df.columns
df.info()
df.isnull().sum()
df.dtypes
df.duplicated().sum()
df.describe().T
```

Cleaning the Data Code

Changing dtype

Filling the NaT values with average of time

fetching month,day of week, weekday

Dropping column unnamed as it is not useful for us

```
df.sample(2)
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()]
df["Date"].fillna(df["Date"].mean(),inplace=True)
df['Date'].isnull().sum()
df.dtypes
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()
df.drop(columns=["Unnamed: 0"],inplace=True)
df.sample()
df.corr().T
plt.figure(figsize=(10,10))
sns.heatmap(df.corr(),annot=True)
for i in df.columns:
```

```
print(i,"---------",df[i].unique())
```

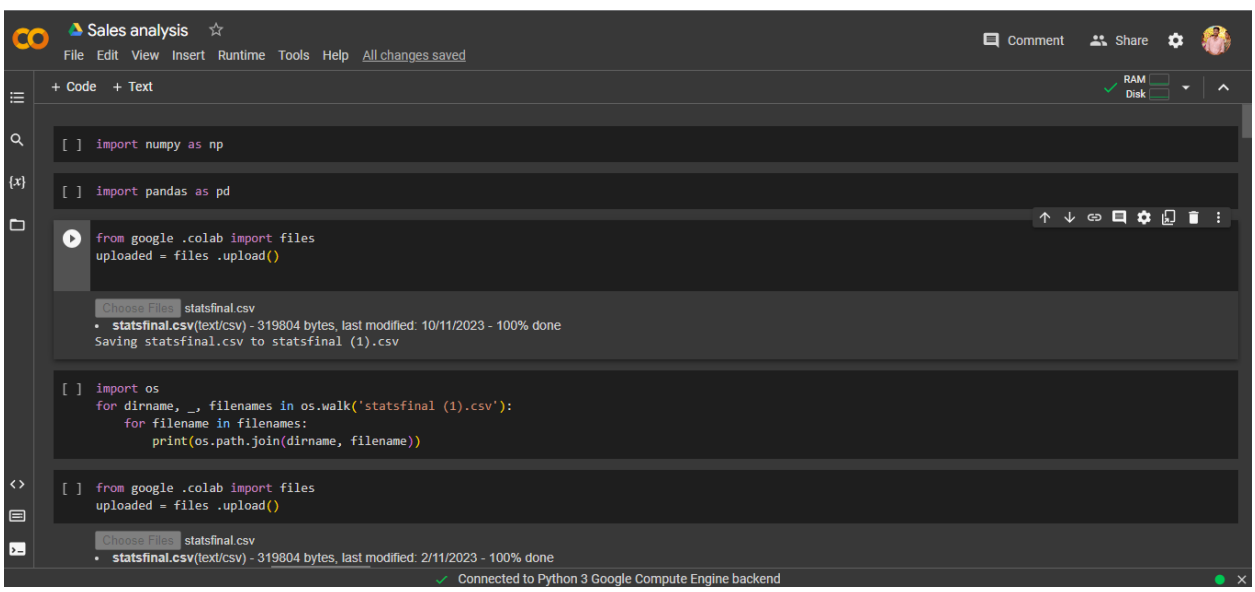
Data Analysis

Analysis the Data through the Python code

Sales analysis

https://colab.research.google.com/drive/1d3PCu5_NhTyP80NYDCE7BUkgj3mwzrt_?usp=sharing

Sample Output



Sales analysis

File Edit View Insert Runtime Tools Help All changes saved

Comment Share

+ Code + Text

```
Saving statsfinal.csv to statsfinal.csv
```

```python
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
pd.options.display.max_columns=50
sns.set(style="darkgrid")
```

```python
df=pd.read_csv("statsfinal.csv")
df.head(5)
```

	Unnamed: 0	Date	Q-P1	Q-P2	Q-P3	Q-P4	S-P1	S-P2	S-P3	S-P4
0	0	13-06-2010	5422	3725	576	907	17187.74	23616.50	3121.92	6466.91
1	1	14-06-2010	7047	779	3578	1574	22338.99	4938.86	19392.76	11222.62
2	2	15-06-2010	1572	2082	595	1145	4983.24	13199.88	3224.90	8163.85
3	3	16-06-2010	5657	2399	3140	1672	17932.69	15209.66	17018.80	11921.36
4	4	17-06-2010	3668	3207	2184	708	11627.56	20332.38	11837.28	5048.04

```python
df.shape
```

Connected to Python 3 Google Compute Engine backend

Sales analysis

File Edit View Insert Runtime Tools Help All changes saved

Comment Share

+ Code + Text

4	4	17-06-2010	3668	3207	2184	708	11627.56	20332.38	11837.28	5048.04

```python
df.shape
```

```
(4600, 10)
```

```python
df.columns
```

```
Index(['Unnamed: 0', 'Date', 'Q-P1', 'Q-P2', 'Q-P3', 'Q-P4', 'S-P1', 'S-P2',
       'S-P3', 'S-P4'],
      dtype='object')
```

```python
df.info()
```

```
<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
```

Connected to Python 3 Google Compute Engine backend

Sales analysis

File Edit View Insert Runtime Tools Help All changes saved

Comment Share

+ Code + Text

RAM Disk

```
[ ]  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
```

```
df.isnull().sum()
```

```
Unnamed: 0    0
Date          0
Q-P1          0
Q-P2          0
Q-P3          0
Q-P4          0
S-P1          0
S-P2          0
S-P3          0
S-P4          0
dtype: int64
```

```
df.dtypes
```

```
Unnamed: 0      int64
Date           object
Q-P1            int64
```

Connected to Python 3 Google Compute Engine backend

Sales analysis

File Edit View Insert Runtime Tools Help All changes saved

Comment Share

+ Code + Text

```
S-P2          0
S-P3          0
S-P4          0
dtype: int64
```

```
df.dtypes
```

```
Unnamed: 0      int64
Date           object
Q-P1            int64
Q-P2            int64
Q-P3            int64
Q-P4            int64
S-P1          float64
S-P2          float64
S-P3          float64
S-P4          float64
dtype: object
```

```
df.duplicated().sum()
```

```
0
```

```
df.describe().T
```

Connected to Python 3 Google Compute Engine backend

Sales analysis

File Edit View Insert Runtime Tools Help All changes saved

Comment Share

+ Code + Text

```
0
```

```
df.describe().T
```

	count	mean	std	min	25%	50%	75%	max
Unnamed: 0	4600.0	2299.500000	1328.049949	0.00	1149.750	2299.500	3449.250	4599.00
Q-P1	4600.0	4121.849130	2244.271323	254.00	2150.500	4137.000	6072.000	7998.00
Q-P2	4600.0	2130.281522	1089.783705	251.00	1167.750	2134.000	3070.250	3998.00
Q-P3	4600.0	3145.740000	1671.832231	250.00	1695.750	3202.500	4569.000	6000.00
Q-P4	4600.0	1123.500000	497.385676	250.00	696.000	1136.500	1544.000	2000.00
S-P1	4600.0	13066.261743	7114.340094	805.18	6817.085	13114.290	19248.240	25353.66
S-P2	4600.0	13505.984848	6909.228687	1591.34	7403.535	13529.560	19465.385	25347.32
S-P3	4600.0	17049.910800	9061.330694	1355.00	9190.965	17357.550	24763.980	32520.00
S-P4	4600.0	8010.555000	3546.359869	1782.50	4962.480	8103.245	11008.720	14260.00

```
df.sample(2)
```

Unnamed: 0	Date	Q-P1	Q-P2	Q-P3	Q-P4	S-P1	S-P2	S-P3	S-P4

Connected to Python 3 Google Compute Engine backend

Sales analysis

File Edit View Insert Runtime Tools Help All changes saved

Comment Share

+ Code + Text

RAM Disk

```
df.sample(2)
```

	Unnamed: 0	Date	Q-P1	Q-P2	Q-P3	Q-P4	S-P1	S-P2	S-P3	S-P4
670	670	17-04-2012	7189	1385	3309	320	22789.13	8780.90	17934.78	2281.60
2388	2388	06-01-2017	7837	2404	2399	975	24843.29	15241.36	13002.58	6951.75

```
[ ] from datetime import datetime as dt
    df[df["Date"]=="31-9-2010"]
```

	Unnamed: 0	Date	Q-P1	Q-P2	Q-P3	Q-P4	S-P1	S-P2	S-P3	S-P4
109	109	31-9-2010	4986	342	4978	558	15805.62	2168.28	26980.76	3978.54

```
[ ] df['Date'] = pd.to_datetime(df['Date'], errors='coerce')
    df[df['Date'].isnull()]
```

```
<ipython-input-23-78db3d189fbd>:1: UserWarning: Parsing dates in DD/MM/YYYY format when dayfirst=False (the default) was specified. This may lead to inconsistent
  df['Date'] = pd.to_datetime(df['Date'], errors='coerce')
```

	Unnamed: 0	Date	Q-P1	Q-P2	Q-P3	Q-P4	S-P1	S-P2	S-P3	S-P4
109	109	NaT	4986	342	4978	558	15805.62	2168.28	26980.76	3978.54

Connected to Python 3 Google Compute Engine backend