

TASK 4:SALES PREDICTION USING PYTHON

1.Sales prediction involves forecasting the amount of a product that customers will purchase, taking into account various factors such as advertising expenditure, target audience segmentation, and advertising platform selection.

2.In businesses that offer products or services, the role of a Data Scientist is crucial for predicting future sales. They utilize machine learning techniques in Python to analyze and interpret data, allowing them to make informed decisions regarding advertising costs. By leveraging these predictions, businesses can optimize their advertising strategies and maximize sales potential. Let's embark on the journey of sales prediction using machine learning in Python.

Import required libraries

```
In [43]: import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.metrics import mean_squared_error, r2_score, mean_absolute_error
from sklearn.model_selection import train_test_split
from sklearn.model_selection import train_test_split
from sklearn.impute import SimpleImputer
from sklearn.linear_model import LinearRegression
from sklearn.metrics import accuracy_score
```

Loading and reading the dataset

```
In [44]: Data=pd.read_csv('advertising.csv')
Data
```

Out[44]:

| | TV | Radio | Newspaper | Sales |
|-----|-------|-------|-----------|-------|
| 0 | 230.1 | 37.8 | 69.2 | 22.1 |
| 1 | 44.5 | 39.3 | 45.1 | 10.4 |
| 2 | 17.2 | 45.9 | 69.3 | 12.0 |
| 3 | 151.5 | 41.3 | 58.5 | 16.5 |
| 4 | 180.8 | 10.8 | 58.4 | 17.9 |
| ... | ... | ... | ... | ... |
| 195 | 38.2 | 3.7 | 13.8 | 7.6 |
| 196 | 94.2 | 4.9 | 8.1 | 14.0 |
| 197 | 177.0 | 9.3 | 6.4 | 14.8 |
| 198 | 283.6 | 42.0 | 66.2 | 25.5 |
| 199 | 232.1 | 8.6 | 8.7 | 18.4 |

200 rows × 4 columns

there are 200 rows and 4 columns in dataset

Statistical summary of the numerical columns in the DataFrame

```
In [45]: Data.describe()
```

Out[45]:

| | TV | Radio | Newspaper | Sales |
|-------|------------|------------|------------|------------|
| count | 200.000000 | 200.000000 | 200.000000 | 200.000000 |
| mean | 147.042500 | 23.264000 | 30.554000 | 15.130500 |
| std | 85.854236 | 14.846809 | 21.778621 | 5.283892 |
| min | 0.700000 | 0.000000 | 0.300000 | 1.600000 |
| 25% | 74.375000 | 9.975000 | 12.750000 | 11.000000 |
| 50% | 149.750000 | 22.900000 | 25.750000 | 16.000000 |
| 75% | 218.825000 | 36.525000 | 45.100000 | 19.050000 |
| max | 296.400000 | 49.600000 | 114.000000 | 27.000000 |

```
In [42]: # print the data types of each column  
print(Data.dtypes)
```

```
TV          float64  
Radio       float64  
Newspaper   float64  
Sales       float64  
dtype: object
```

To get information about the data

```
In [33]: Data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 200 entries, 0 to 199
Data columns (total 4 columns):
 #   Column      Non-Null Count  Dtype
---  -
 0   TV          200 non-null    float64
 1   Radio       200 non-null    float64
 2   Newspaper   200 non-null    float64
 3   Sales       200 non-null    float64
dtypes: float64(4)
memory usage: 6.4 KB
```

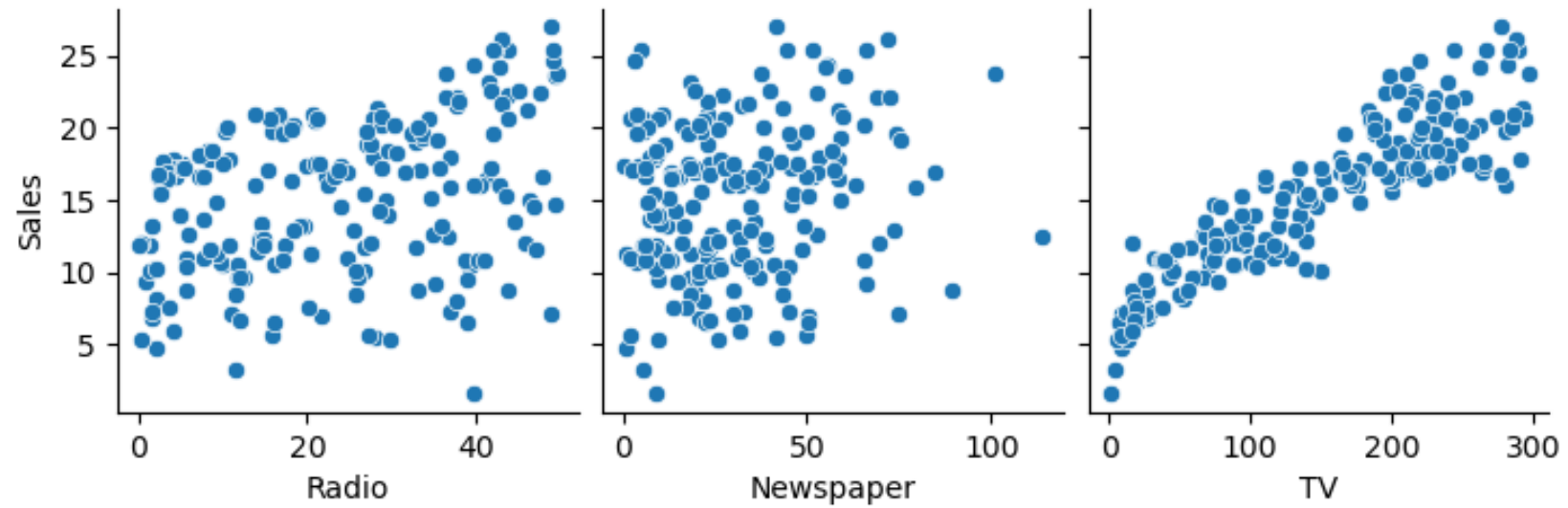
Check missing values

```
In [34]: Data.isnull().sum()
```

```
Out[34]: TV          0
Radio          0
Newspaper      0
Sales          0
dtype: int64
```

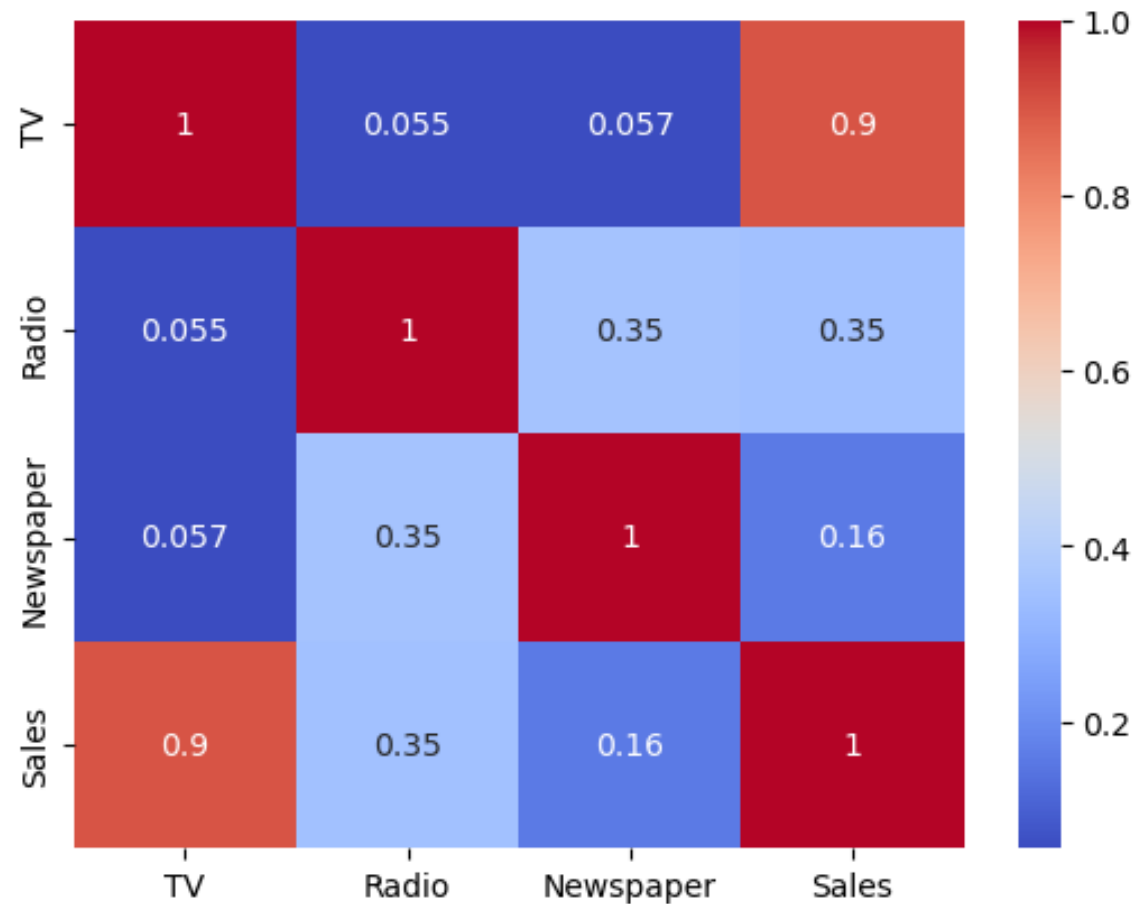
```
In [46]: sns.pairplot(Data,x_vars=["Radio","Newspaper","TV"],y_vars="Sales")
```

```
Out[46]: <seaborn.axisgrid.PairGrid at 0x240a3b90610>
```



```
In [36]: sns.heatmap(Data.corr(), annot=True, cmap='coolwarm')
```

```
Out[36]: <Axes: >
```



```
In [37]: X=Data.drop(["Sales"],axis=1)
y=Data["Sales"]
```

```
In [38]: X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

```
In [39]: model=LinearRegression()
```

```
In [40]: model.fit(X_train,y_train)
```

```
Out[40]:
```

▼ LinearRegression

LinearRegression()

```
In [41]: y_pred=model.predict(X_test)
mse=mean_squared_error(y_test,y_pred)
rmse=np.sqrt(mse)
print(mse)
print(rmse)
```

```
2.9077569102710923
```

```
1.7052146229349232
```

```
In [ ]:
```