kaggle

Create

Home

Competitions

Datasets

Models

Code

Discussions

Learn

More

Your Work

VIEWED

EDITED

View Active Events

SOURAV GK · 1H AGO · 2 VIEWS · PRIVATE

0 Edit

Sales Prediction using Linear Regression

Python · Advertising Dataset

Notebook Input Output Logs Comments (0) Settings

Run
16.0s

Version 3 of 3

Add Tags

In [1]:

```
#importing the needed libraries
import warnings
warnings.filterwarnings('ignore')
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
```

In [2]:

```
#importing and reading the dataset
Data = pd.read_csv('/kaggle/input/advertising-dataset/advertising.csv')
Data.head()
```

Out[2]:

	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

In [3]:

```
#general information about the data we are working with
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
```

In [4]:

```
#all the information gathered from the data
Data.describe()
```

Out[4]:

	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

What we can observe from this information is -

1. TV section has the highest mean value

2. Radio has the lowest average value

In [5]:

```
#Performing Data Cleaning
#checking for non-related values(null)
Data.isnull()
```

Out[5]:

	TV	Radio	Newspaper	Sales
0	False	False	False	False
1	False	False	False	False
2	False	False	False	False
3	False	False	False	False
4	False	False	False	False
...	...	...	...	...
195	False	False	False	False
196	False	False	False	False
197	False	False	False	False
198	False	False	False	False
199	False	False	False	False

200 rows × 4 columns

In [6]:

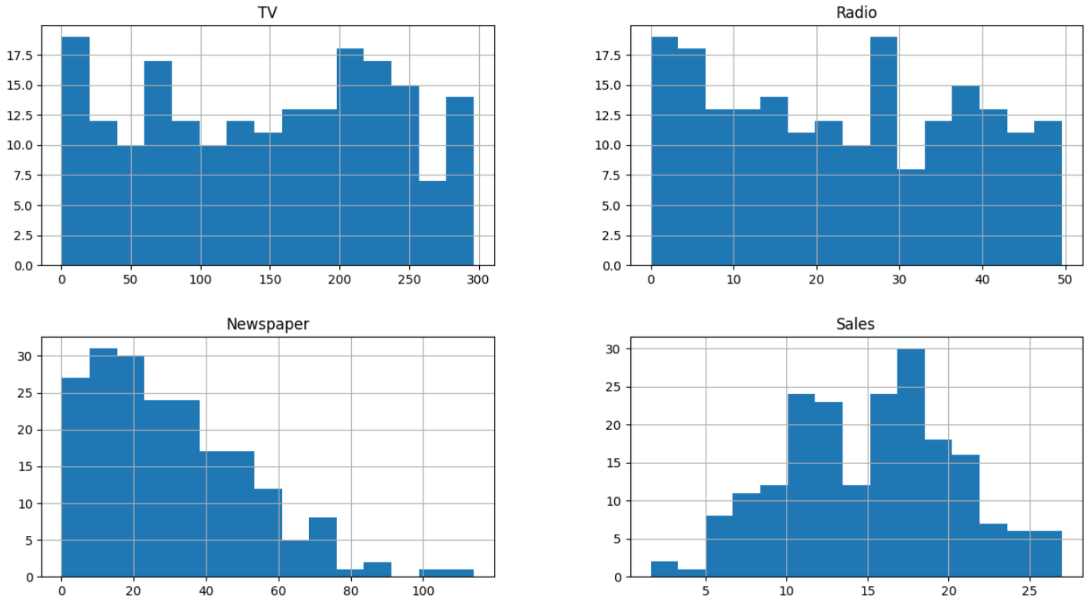
```
#GRAPHICAL ANALYSIS
```

In [7]:

```
Data.hist(bins = 15, figsize = (15, 8))
```

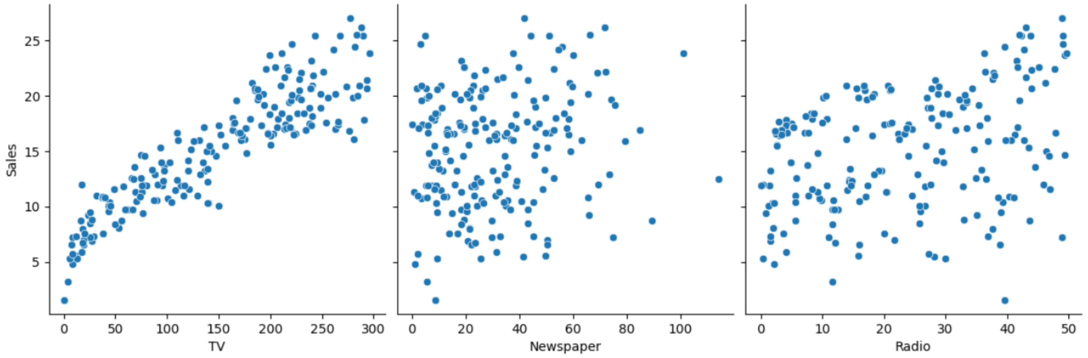
Out[7]:

```
array([[<Axes: title={'center': 'TV'}>,
        <Axes: title={'center': 'Radio'}>],
       [<Axes: title={'center': 'Newspaper'}>,
        <Axes: title={'center': 'Sales'}>]], dtype=object)
```



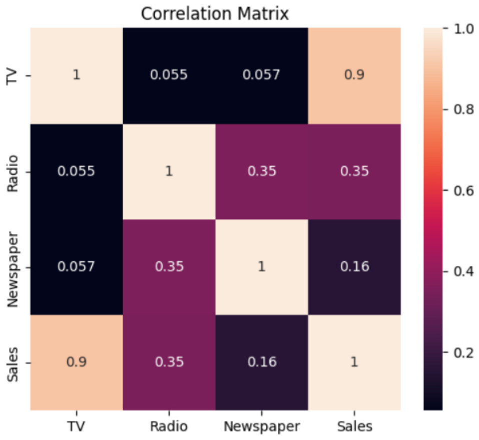
In [8]:

```
#using Scatter Plot to check the relation with other variables
sns.pairplot(Data, x_vars=['TV', 'Newspaper', 'Radio'], y_vars='Sales', height=4, aspect=1, kind='scatter')
plt.show()
```



In [9]:

```
#Understanding how the variables are related to each other
plt.figure(figsize = (6,5))
sns.heatmap(Data.corr(),annot = True)
plt.title('Correlation Matrix')
plt.show()
```



We can notice from the heatmap that the variable - TV is more realted to variable - Sales.

In [10]:

```
#Building the model
#Performing Linear Regression using TV as the feature variable
from sklearn.model_selection import train_test_split
X = Data[['TV','Radio','Newspaper']]
Y = Data[['Sales']]
X_train,X_test,Y_train,Y_test = train_test_split(X,Y,train_size = 0.9,test_size = 0.1,random_state = 0)
```

In [11]:

```
print(X_train)
```

```
        TV  Radio  Newspaper
183  287.6   43.0       71.8
145  140.3    1.9        9.0
45   175.1   22.5       31.5
159  131.7   18.4       34.6
60    53.5    2.0       21.4
..     ...    ...        ...
67   139.3   14.5       10.2
192   17.2    4.1       31.6
117   76.4    0.8       14.8
47   239.9   41.5       18.5
172   19.6   20.1       17.0

[180 rows x 3 columns]
```

In [12]:

```
print(Y_train)
```

```
     Sales
183   26.2
145   10.3
45    16.1
159   12.9
60     8.1
..     ...
67    13.4
192    5.9
117    9.4
47    23.2
172    7.6

[180 rows x 1 columns]
```

In [13]:

```
print(X_test)
```

```
        TV  Radio  Newspaper
```

```
18    69.2  20.5       18.3
170   50.0  11.6       18.4
107   90.4   0.3       23.2
98   289.7  42.3       51.2
177  170.2   7.8       35.2
182   56.2   5.7       29.7
5      8.7  48.9       75.0
146  240.1   7.3        8.7
12    23.8  35.1       65.9
152  197.6  23.3       14.2
61   261.3  42.7       54.7
125   87.2  11.8       25.9
180  156.6   2.6        8.3
154  187.8  21.1        9.5
80    76.4  26.7       22.3
7    120.2  19.6       11.6
33   265.6  20.0        0.3
130    0.7  39.6        8.7
37    74.7  49.4       45.7
74   213.4  24.6       13.1
```

In [14]:

```
print(Y_test)
```

```
     Sales
18    11.3
170    8.4
107   12.0
98    25.4
177   16.7
182    8.7
5      7.2
146   18.2
12     9.2
152   16.6
61    24.2
125   10.6
180   15.5
154   20.6
80    11.8
7     13.2
33    17.4
130    1.6
37    14.7
74    17.0
```

In [15]:

```
from sklearn.linear_model import LinearRegression
model = LinearRegression()
model.fit(X_train,Y_train)
```

Out[15]:

```
▾ LinearRegression
LinearRegression()
```

In [16]:

```
model.coef_
```

Out[16]:

```
array([[ 0.05362697,  0.11604755, -0.00237763]])
```

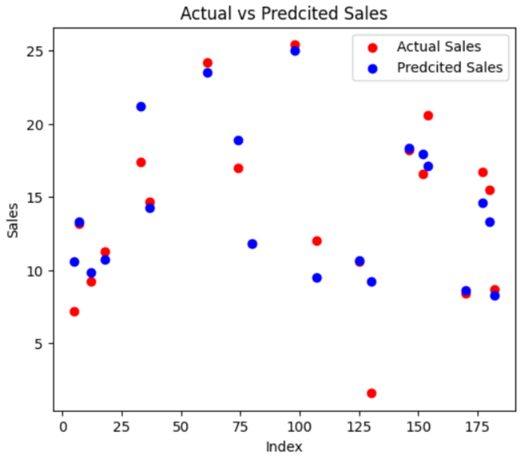
In [17]:

```
result = model.predict(X_test)
print(result)
```

```
[[10.69679209]
 [ 8.63409325]
 [ 9.47787307]
 [24.97315229]
 [14.59913061]
 [ 8.25503268]
 [10.61329861]
 [18.35263967]
 [ 9.84324624]
 [17.91717704]
 [23.48824352]
 [10.63439394]
 [13.33031489]
 [17.14750297]
 [11.79289057]
 [13.34325515]
 [21.21390351]
[21.21398351]
[ 9.26267781]
[14.28036739]
[18.91796044]]
```

In [18]:

```
plt.figure(figsize = (6,5))
plt.scatter(X_test.index,Y_test,color = 'red',label = 'Actual Sales')
plt.scatter(X_test.index,result,color = 'blue',label = 'Predcited Sales')
plt.xlabel('Index')
plt.ylabel('Sales')
plt.title('Actual vs Predcited Sales')
plt.legend()
plt.show()
```



In [ ]:

Continue exploring

Input
1 file

Output
0 files

Logs
16.0 second run - successful

Comments
0 comments