

```
#importing the libraries
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd

#import dataset
dataset= pd.read_csv("advertising_sales_data.csv")
dataset.drop(['Campaign'], axis=1, inplace=True)
print(dataset)
x=dataset.iloc[:, :-1].values
y=dataset.iloc[:, -1].values
```

```

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 x 4 columns]

```
#imputing for missing values
from sklearn.impute import SimpleImputer
imputer= SimpleImputer(missing_values=np.nan, strategy= 'mean')
imputer.fit(x)
x= imputer.transform(x)
```

```
#calculating average amount spent on TV advertising
average_tv_spending = np.mean(x[:, 0])
print(f"Average amount spent on TV advertising: {average_tv_spending}")
```

```
Average amount spent on TV advertising: 147.0425
```

Average amount spent on TV advertising: 147.0425

```
#Correlation between radio advertising expenditure and product sales
correlation_radio_sales =np.corrcoef(x[:,1].astype(float), y.astype(float), rowvar=False)
print(f"Correlation between radio advertising expenditure and product sales: {correlation_radio_sales}")
```

```
Correlation between radio advertising expenditure and product sales: [[1.          0.34965074]
 [0.34965074 1.          ]]
```

This shows pretty poor correlation between the radio advertising expenditure and product sales of only- 0.35

```
#Advertisinig with the strongest impact- Check for correlation with TV and Newspaper
correlation_TV_sales =np.corrcoef(x[:,0].astype(float), y.astype(float), rowvar=False)
print(f"Correlation between TV advertising expenditure and product sales: {correlation_TV_sales}")
correlation_Newspaper_sales =np.corrcoef(x[:,2].astype(float), y.astype(float), rowvar=False)
print(f"Correlation between Newspaper advertising expenditure and product sales: {correlation_Newspaper_sales}")
```

```
Correlation between TV advertising expenditure and product sales: [[1.          0.90120791]
 [0.90120791 1.          ]]
Correlation between Newspaper advertising expenditure and product sales: [[1.          0.15796003]
 [0.15796003 1.          ]]
```

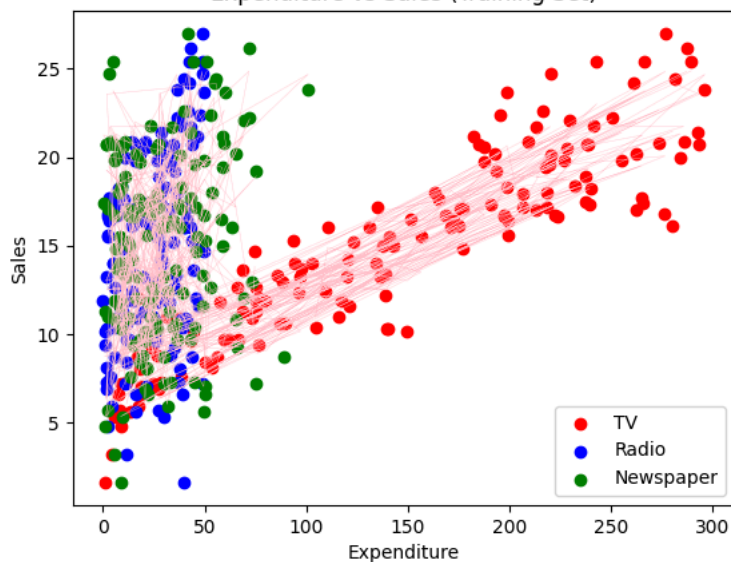
From last 2 codes we conclude: TV has the highest impact with a strong correlation of 0.901

```
#Splitting dataset in Training set and Test Set
from sklearn.model_selection import train_test_split
#creating 4 variables-
x_train, x_test, y_train, y_test= train_test_split(x,y, test_size= 0.25, random_state=1)
#Training the model
from sklearn.linear_model import LinearRegression
regressor= LinearRegression()
regressor.fit(x_train, y_train)
#Predicting the Test set results
y_pred= regressor.predict(x_test)
```

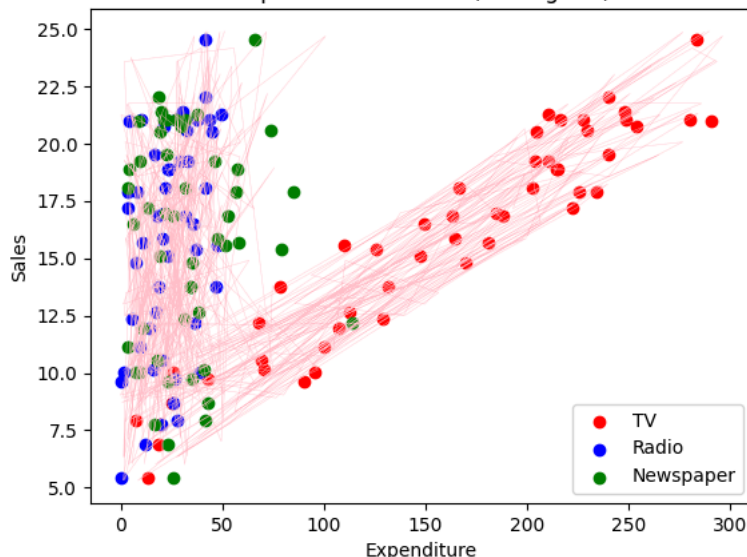
```
plt.scatter(x_train[:,0], y_train, color='red', label="TV")
plt.plot(x_train, regressor.predict(x_train), color='pink', linewidth=0.25)
plt.scatter(x_train[:,1], y_train, color='blue', label="Radio")
plt.scatter(x_train[:,2], y_train, color='green', label="Newspaper")
plt.title("Expenditure vs Sales (Training Set)")
plt.xlabel("Expenditure")
plt.ylabel("Sales")
plt.legend()
plt.show()
plt.scatter(x_test[:,0], y_pred, color='red', label= "TV")
plt.plot(x_train, regressor.predict(x_train), color='pink', linewidth=0.25)
plt.scatter(x_test[:,1], y_pred, color='blue', label= "Radio")
plt.scatter(x_test[:,2], y_pred, color='green', label="Newspaper")
plt.title("Expenditure vs Sales (Testing Set)")
plt.xlabel("Expenditure")
plt.ylabel("Sales")
plt.legend()
plt.show()
```




Expenditure vs Sales (Training Set)



Expenditure vs Sales (Testing Set)



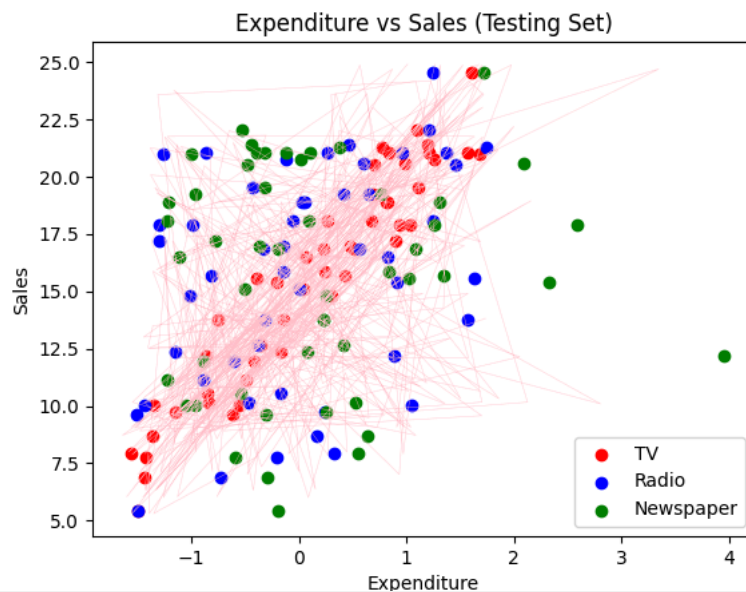
```
#making the prediction for sales when: of advertising expenditures: $200 on TV, $40 on Radio, and $50 on Newspaper  
regressor.predict([[200,40,50]])
```

 array([19.72797923])

```
#Normalising the dataset  
from sklearn.preprocessing import StandardScaler  
sc= StandardScaler()  
x_train= sc.fit_transform(x_train)  
x_test= sc.transform(x_test)  
print(x_train)  
#Now performing the Linear Regression  
from sklearn.linear_model import LinearRegression  
regressor= LinearRegression()  
regressor.fit(x_train, y_train)  
#Predicting the Test set results  
y_pred= regressor.predict(x_test)
```

 [Show hidden output](#)

```
#plotting with Normalised Values  
plt.scatter(x_train[:,0], y_train, color='red', label="TV")  
plt.plot(x_train, regressor.predict(x_train), color='pink', linewidth=0.25)  
plt.scatter(x_train[:,1], y_train, color='blue', label="Radio")  
plt.scatter(x_train[:,2], y_train, color='green', label="Newspaper")  
plt.title("Expenditure vs Sales (Training Set)")  
plt.xlabel("Expenditure")  
plt.ylabel("Sales")  
plt.legend()  
plt.show()  
plt.scatter(x_test[:,0], y_pred, color='red', label= "TV")  
plt.plot(x_train, regressor.predict(x_train), color='pink', linewidth=0.25)  
plt.scatter(x_test[:,1], y_pred, color='blue', label= "Radio")  
plt.scatter(x_test[:,2], y_pred, color='green', label="Newspaper")  
plt.title("Expenditure vs Sales (Testing Set)")  
plt.xlabel("Expenditure")  
plt.ylabel("Sales")  
plt.legend()  
plt.show()
```



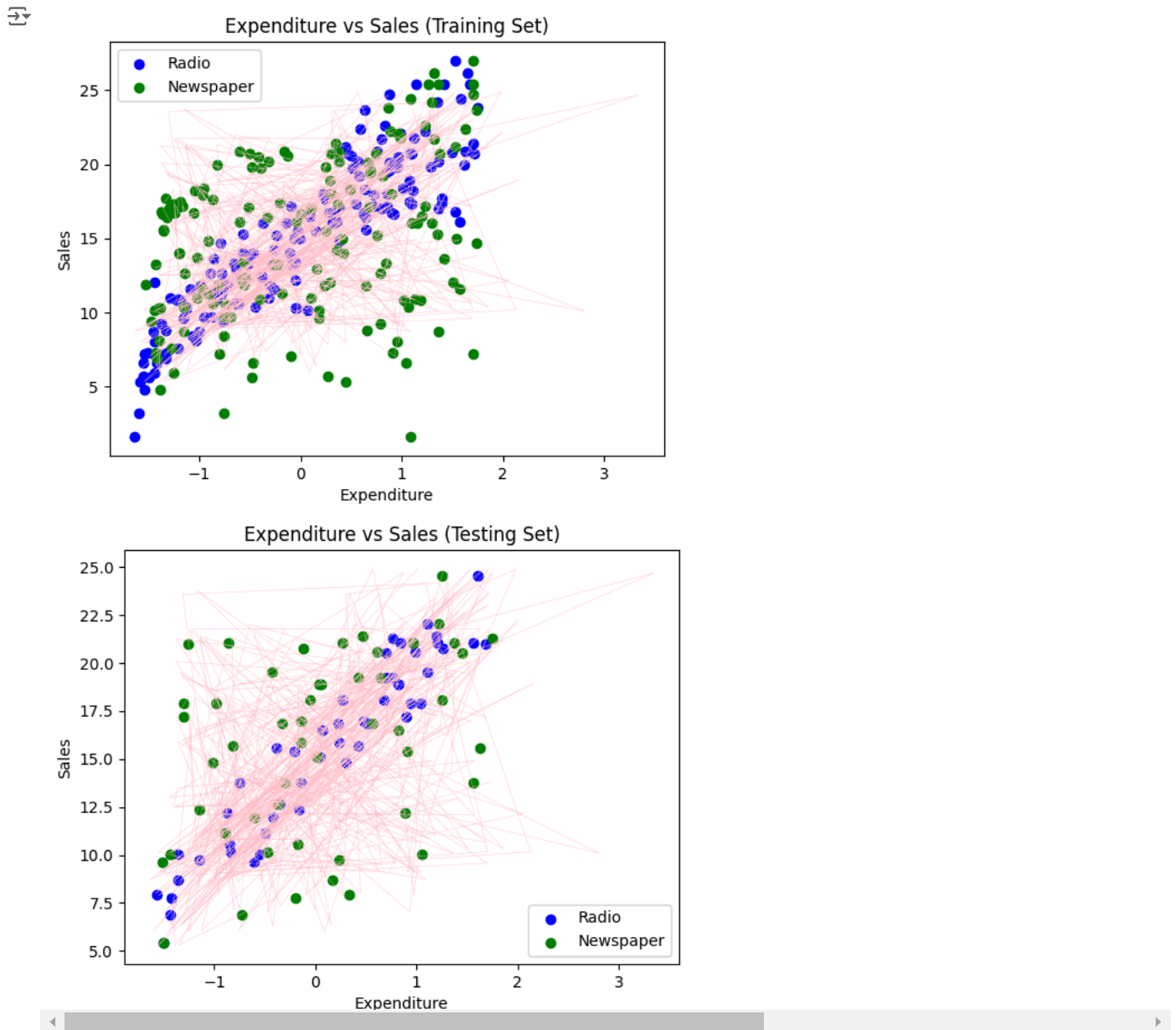
```
#using only radio and newspaper for predictions of sales
dataset= pd.read_csv("advertising_sales_data.csv")
dataset.drop(['Campaign', 'TV'], axis=1, inplace=True)
x=dataset.iloc[:, :-1].values
y=dataset.iloc[:, -1].values

#imputing for missing values
from sklearn.impute import SimpleImputer
imputer= SimpleImputer(missing_values=np.nan, strategy= 'mean')
imputer.fit(x)
x= imputer.transform(x)

#Splitting dataset in Training set and Test Set
from sklearn.model_selection import train_test_split
#creating 4 variables-
x_train, x_test, y_train, y_test= train_test_split(x,y, test_size= 0.25, random_state=1)
#Training the model
from sklearn.linear_model import LinearRegression
regressor= LinearRegression()
regressor.fit(x_train, y_train)
#Predicting the Test set results
y_pred= regressor.predict(x_test)
```

```
#plotting only radio and newspaper
plt.plot(x_train, regressor.predict(x_train), color='pink', linewidth=0.25)
plt.scatter(x_train[:,0], y_train, color='blue', label="Radio")
plt.scatter(x_train[:,1], y_train, color='green', label="Newspaper")
plt.title("Expenditure vs Sales (Training Set)")
plt.xlabel("Expenditure")
plt.ylabel("Sales")
plt.legend()
plt.show()

plt.plot(x_train, regressor.predict(x_train), color='pink', linewidth=0.25)
plt.scatter(x_test[:,0], y_pred, color='blue', label= "Radio")
plt.scatter(x_test[:,1], y_pred, color='green', label="Newspaper")
plt.title("Expenditure vs Sales (Testing Set)")
plt.xlabel("Expenditure")
plt.ylabel("Sales")
plt.legend()
plt.show()
```



```
#normalising only for radio and newspaper
#Normalising the dataset
from sklearn.preprocessing import StandardScaler
sc= StandardScaler()
x_train= sc.fit_transform(x_train)
x_test= sc.transform(x_test)
#Now performing the Linear Regression
from sklearn.linear_model import LinearRegression
regressor= LinearRegression()
regressor.fit(x_train, y_train)
#Predicting the Test set results
y_pred= regressor.predict(x_test)
```

```
#plotting with Normalised Values for radio and newspaper
```

```
plt.plot(x_train, regressor.predict(x_train), color='pink', linewidth=0.25)
plt.scatter(x_train[:,0], y_train, color='blue', label="Radio")
plt.scatter(x_train[:,1], y_train, color='green', label="Newspaper")
plt.title("Expenditure vs Sales (Training Set)")
plt.xlabel("Expenditure")
plt.ylabel("Sales")
plt.legend()
plt.show()
```

```
plt.plot(x_train, regressor.predict(x_train), color='pink', linewidth=0.25)
plt.scatter(x_test[:,0], y_pred, color='blue', label= "Radio")
plt.scatter(x_test[:,1], y_pred, color='green', label="Newspaper")
plt.title("Expenditure vs Sales (Testing Set)")
plt.xlabel("Expenditure")
plt.ylabel("Sales")
plt.legend()
plt.show()
```

