Exp-3

Model planning and Building

Aim:

To describe the model planning and building of the whole data set.

Code:

```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.linear_model import LinearRegression
from sklearn.metrices_selection import train_test_split
from sklearn preprocessing import standard scaler.
df = pd.read_csv('advertising.csv')
print (df.head())
print (df.describe())
X = df [['TV', 'Radio', 'Newspaper']]
y = df ['sales']
# split data
x_train, y_test, y_train, y_test = train_test_split
                    (x, y, test_size=0.2, random_state=0)

model = LinearRegression()
model.fit (x_train, y_train)
y_pred = model.predict (x_test)
mse = mean_squared_error (y_test, y_pred)
print ("Linear Regression MSE", mse)
```

Exp:3

Aim:

Code:

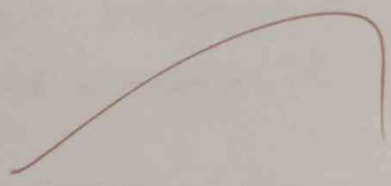
	TV	Radio	Newspaper	Sales
0	230.1	37.8	69.2	22.1
1	44.5	39.3	45.1	10.4
2	17.2	45.9	69.3	12.0
3	151.5	41.3	58.5	16.5
4	180.8	10.8	58.4	17.9

	TV	Radio	Newspaper	Sales
count	200.000000	200.000000	200.000000	200.000000
mean	147.042500	23.264000	30.554000	18.130500
std	85.854236	14.846809	21.778621	5.28852
min	0.700000	0.000000	0.300000	1.600007
25%	79.357300	9.975000	12.787500	11.000000
50%	149.780000	22.900000	25.750000	16.000000
max	296.400000	47.600000	114.000000	27.000000

```
plt.figure(figsize=(8,5))
sns.scatterplot(x=y_test, y=y_pred)
plt.xlabel('Actual sales')
plt.ylabel('predicted sales')
plt.title("Linear Regression: Actual vs predicted")
plt.show()

kmean = KMean(n_cluster=3, random_state=0)
df['cluster'] = kmean.fit_predict(scaled)
plt.figure(figsize=(8,6))
sns.scatterplot(data=df, x='TV', y='Sales',
                hue='cluster', palette='set2')

plt.show()
```



Result:

The program has been executed successfully.