Ex 03

Model Planning and Building

Aim: To implement and Compare simple machine learning models.

Program Code:

```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.linear_model import
from sklearn.metrics import mean_squared_error
from sklearn.cluster import Kmeans
df = pd.read.csv('advertising.csv')
print( df.head())
print( df.describe())
x = df.['Sales']
model = LinearRegression()
model.fit (x_train, y_train)
mse = mean_squared_error (y_test, y_pred)
print('Linear Regression MSE:", mse)
plt.figure( figsize = (8,5))
sns.Scatterplot (x = y_test, y = y_pred)
```

Output:

TV	Radio	newspaper	Sales
230.1	37.8	69.2	22.1
44.5	39.3	45.1	10.4
17.5	45.9	69.3	12.0
151.5	41.3	58.5	16.5
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	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

Linear Regression MSE: 4.522525620419 1

```
plt.label ("Actual Sales")
plt.ylabel ('Predicted Sales')
plt.title("Linear Regression: Actual vs
                                Predicted Sales")
plt.show()
scaler = standard Sales()
Scaled= Scaler.fit_transform(df[['Tv','Radio',
                                'Newspaper']])
plt.['cluster'] = Kmeans.fit.predict(Scaled)
plt.figure (figsize=(8,6))
sns.Scatterplot (data=df, x='Tv', y='Sales',
    hue='Cluster', palette= 'Set2')
plt.title('K-means Clustering: Tv Budget vs
                                Sales")
plt.show()
```

Result:

Thus the simple machine learning models has been implemented and completed successfully.