31.Customer Segmentation using Clustering

Program:-

```
import pandas as pd
import numpy as np
from sklearn.cluster import KMeans
from sklearn.preprocessing import StandardScaler
import matplotlib.pyplot as plt
# Sample customer data: CustomerID, Total_Spend, Frequency, Avg_Order_Value
data = {
    "customer_id": [1,2,3,4,5,6,7,8],
    "total_spend": [5000, 800, 1500, 7000, 12000, 3000, 400, 6000],
    "frequency": [50, 10, 20, 60, 90, 30, 5, 55],
    "avg_order_value": [100, 80, 75, 120, 130, 100, 80, 110]}
df = pd.DataFrame(data)
# Select features for clustering
X = df[["total_spend", "frequency", "avg_order_value"]]
# Scale the data
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)
# Apply K-Means clustering
kmeans = KMeans(n_clusters=3, random_state=0)
df["cluster"] = kmeans.fit_predict(X_scaled)
print("Cluster Centers (scaled):")
print(kmeans.cluster_centers_)
print("\nCustomer Segments:")
print(df[["customer_id", "total_spend", "frequency", "avg_order_value", "cluster"]])
# Simple 2D visualization: total_spend vs frequency
plt.scatter(df["total_spend"], df["frequency"], c=df["cluster"])
plt.xlabel("Total Spend")
```

```
plt.ylabel("Frequency")
plt.title("Customer Segmentation using K-Means")
plt.show()
```

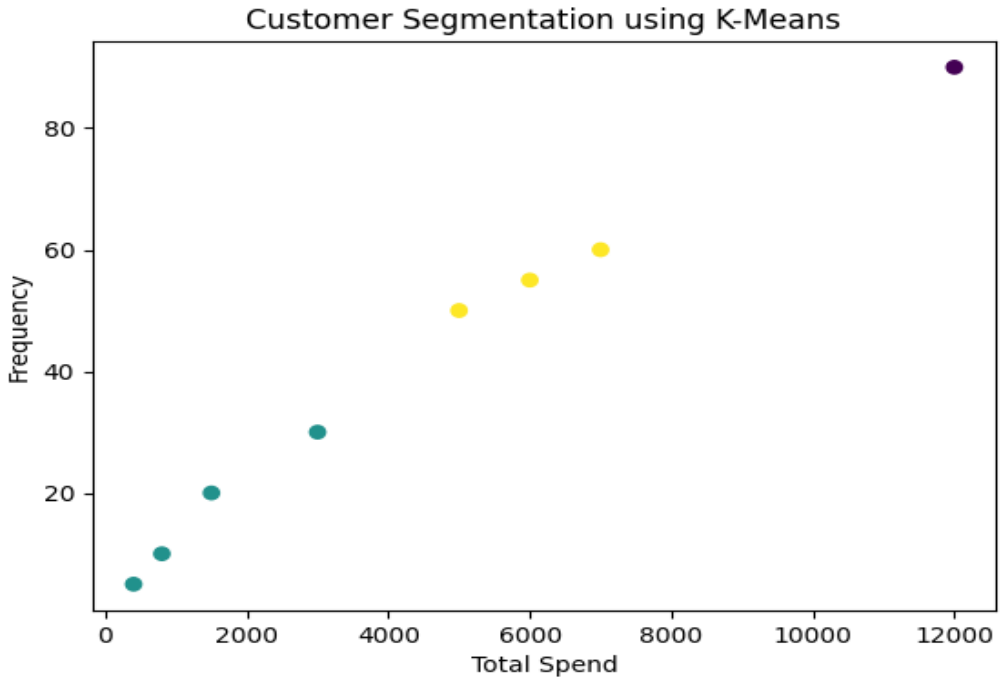
Output:-

Cluster Centers (scaled):

```
[[ 2.06353226  1.84900065  1.6306179 ]
 [-0.8315727  -0.87827531 -0.83194791]
 [ 0.42091952  0.5547002   0.56572458]]
```

Customer Segments:

	customer_id	total_spend	frequency	avg_order_value	cluster
0	1	5000	50	100	2
1	2	800	10	80	1
2	3	1500	20	75	1
3	4	7000	60	120	2
4	5	12000	90	130	0
5	6	3000	30	100	1
6	7	400	5	80	1
7	8	6000	55	110	2



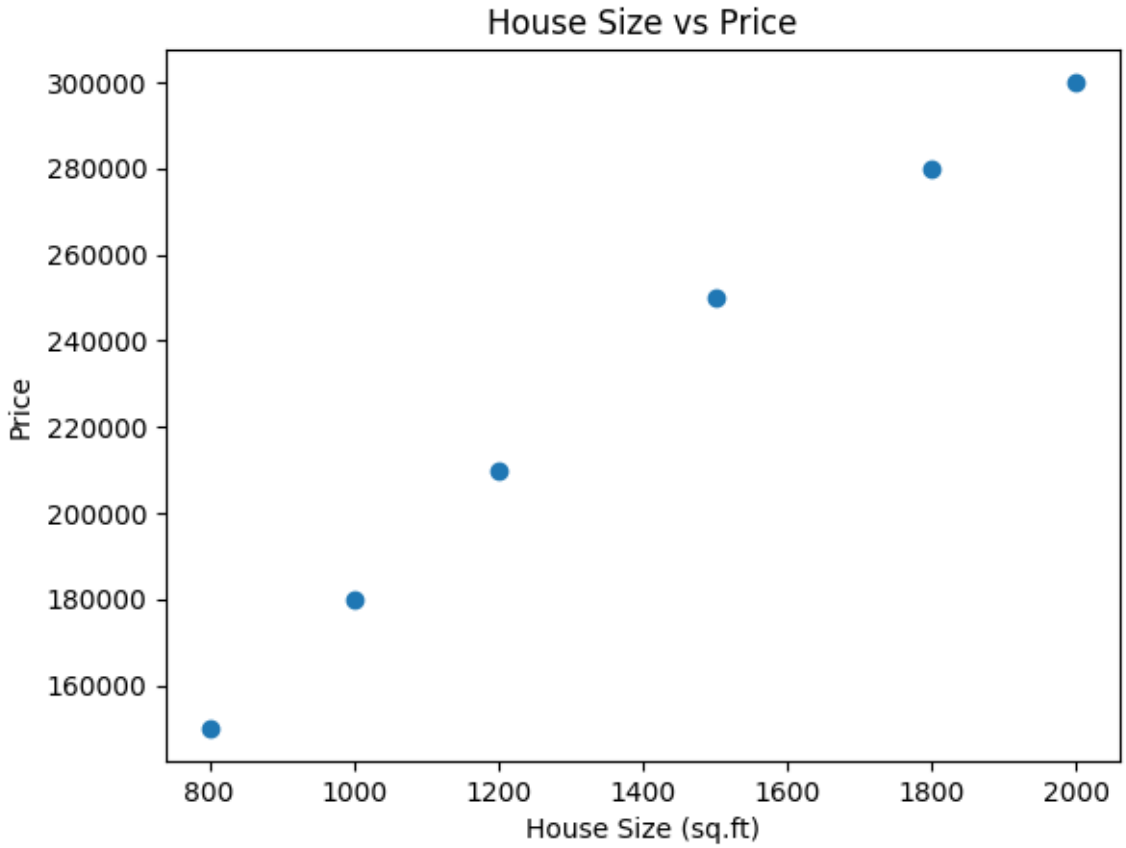
32. House Price Prediction using Linear Regression (Bivariate)

Program:-

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, r2_score
# Sample data: size (sq.ft) and price
data = {
    "size": [800, 1000, 1200, 1500, 1800, 2000],
    "price": [150000, 180000, 210000, 250000, 280000, 300000]
}
df = pd.DataFrame(data)
# Bivariate analysis: scatter plot
plt.scatter(df["size"], df["price"])
plt.xlabel("House Size (sq.ft)")
plt.ylabel("Price")
plt.title("House Size vs Price")
plt.show()
# Prepare data
X = df[["size"]]    # feature matrix
y = df["price"]     # target
# Build model
model = LinearRegression()
model.fit(X, y)
# Predictions
y_pred = model.predict(X)
# Evaluation
mse = mean_squared_error(y, y_pred)
```

```
r2 = r2_score(y, y_pred)
print("Coefficient (slope):", model.coef_[0])
print("Intercept:", model.intercept_)
print("Mean Squared Error:", mse)
print("R² Score:", r2)
```

Output:-



Coefficient (slope): 124.80857580398164

Intercept: 55681.47013782541

Mean Squared Error: 21694742.2154161

R² Score: 0.9923803832219026

33. Car Price Prediction using Multiple Linear Regression

Program:-

```
import pandas as pd
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, r2_score
# Sample car data
data = {
    "engine_size": [1.2, 1.5, 2.0, 2.5, 3.0],
    "horsepower":  [80, 100, 130, 160, 200],
    "fuel_eff":    [18, 17, 15, 13, 11],   # kmpl
    "price":       [600000, 750000, 900000, 1200000, 1500000]}
df = pd.DataFrame(data)
# Features and target
X = df[["engine_size", "horsepower", "fuel_eff"]]
y = df["price"]
# Build model
model = LinearRegression()
model.fit(X, y)
# Predictions
y_pred = model.predict(X)
# Evaluation
mse = mean_squared_error(y, y_pred)
r2 = r2_score(y, y_pred)
print("Coefficients:", model.coef_)
print("Intercept:", model.intercept_)
print("Mean Squared Error:", mse)
print("R² Score:", r2)
```

Output:-

Coefficients: [-750000. 10000. -150000.]

Intercept: 3399999.9999999986

Mean Squared Error: 750000000.0

R^2 Score: 0.992816091954023

34. Treatment Outcome Classification using KNN

Program:-

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score
# Sample medical data
data = {
    "age": [30, 45, 50, 35, 60, 55, 40, 65],
    "bp":  [120,130,140,125,150,145,135,155],   # blood pressure
    "chol":[180,200,220,190,240,230,210,250],   # cholesterol
    "outcome": ["Good","Bad","Bad","Good","Bad","Bad","Good","Bad"]}
df = pd.DataFrame(data)
X = df[["age","bp","chol"]]
y = df["outcome"]
# Train-test split
X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.25, random_state=0)
# KNN model
knn = KNeighborsClassifier(n_neighbors=3)
knn.fit(X_train, y_train)
# Predictions
y_pred = knn.predict(X_test)
# Evaluation
accuracy = accuracy_score(y_test, y_pred)
precision = precision_score(y_test, y_pred, pos_label="Good")
recall = recall_score(y_test, y_pred, pos_label="Good")
f1 = f1_score(y_test, y_pred, pos_label="Good")
```

```
print("Predictions:", y_pred)
print("True labels:", list(y_test))
print("Accuracy:", accuracy)
print("Precision:", precision)
print("Recall:", recall)
print("F1-Score:", f1)
```

Output:-

```
Predictions: ['Bad' 'Bad']
True labels: ['Good', 'Bad']
Accuracy: 0.5
Precision: 0.0
Recall: 0.0
F1-Score: 0.0
```

35. Customer Segmentation using K-Means (Retail Store Data)

Program:-

```
import pandas as pd
from sklearn.cluster import KMeans
from sklearn.preprocessing import StandardScaler
import matplotlib.pyplot as plt
# Sample retail customer data
data = {
    "customer_id": [1,2,3,4,5,6,7,8,9,10],
    "total_spent": [1000, 2000, 500, 7000, 8000, 1200, 300, 4500, 6000, 900],
    "visit_frequency": [5, 10, 3, 25, 30, 7, 2, 18, 22, 4]}
df = pd.DataFrame(data)
X = df[["total_spent", "visit_frequency"]]
# Scale features
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)
# K-Means clustering
kmeans = KMeans(n_clusters=3, random_state=0)
df["cluster"] = kmeans.fit_predict(X_scaled)
print("Customer segments:")
print(df[["customer_id","total_spent","visit_frequency","cluster"]])
# Visualization
plt.scatter(df["total_spent"], df["visit_frequency"], c=df["cluster"])
plt.xlabel("Total Spent")
plt.ylabel("Visit Frequency")
plt.title("Customer Segmentation using K-Means")
plt.show()
```

Output:-

Customer segments:

	customer_id	total_spent	visit_frequency	cluster
0	1	1000	5	0
1	2	2000	10	0
2	3	500	3	0
3	4	7000	25	1
4	5	8000	30	1
5	6	1200	7	0
6	7	300	2	0
7	8	4500	18	2
8	9	6000	22	2
9	10	900	4	0