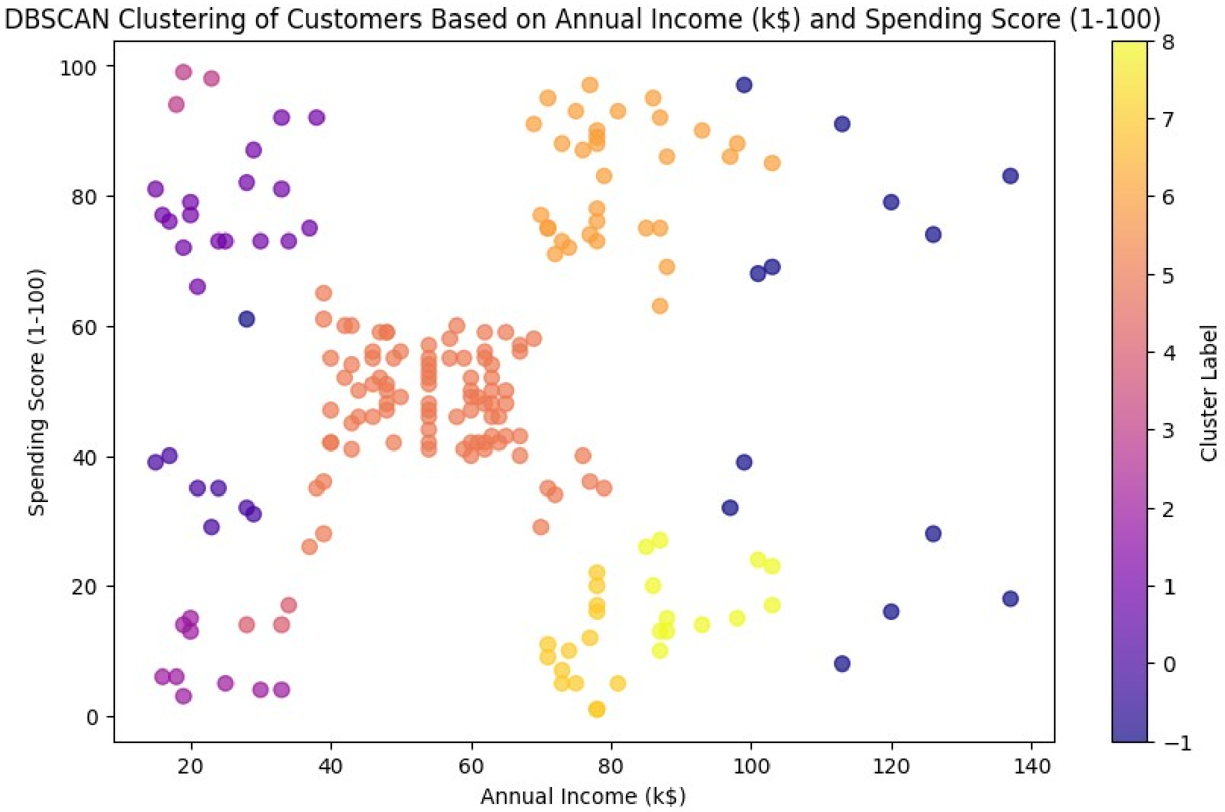
- Name: Deep Salunkhe
- Roll No.:21102A0014
- SEM-7 ML Lab8 Github Link

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.cluster import DBSCAN
from sklearn.preprocessing import StandardScaler

# Load the dataset (replace 'data.csv' with your actual file path)
df = pd.read_csv('/content/Mall_Customers.csv')

# Inspect the first few rows
df.head()
```

```
{"summary":"{\n  \"name\": \"df\",\n  \"rows\": 200,\n  \"fields\": [\
n    {\n      \"column\": \"CustomerID\",\n      \"properties\": {\n
\"dtype\": \"number\",\n        \"std\": 57,\n        \"min\": 1,\n
\"max\": 200,\n        \"num_unique_values\": 200,\n
\"samples\": [\n          96,\n          16,\n          31\
n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"Gender\",\n      \"properties\": {\n        \"dtype\":
\"category\",\n        \"num_unique_values\": 2,\n        \"samples\":
[\n          \"Female\",\n          \"Male\"\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"Age\",\n      \"properties\": {\n
\"dtype\": \"number\",\n        \"std\": 13,\n        \"min\": 18,\n
\"max\": 70,\n        \"num_unique_values\": 51,\n        \"samples\":
[\n          55,\n          26\n        ],\n        \"semantic_type\":
\"\",\n        \"description\": \"\"\n      }\n    },\n    {\n
\"column\": \"Annual Income (k$)\",\n      \"properties\": {\n
\"dtype\": \"number\",\n        \"std\": 26,\n        \"min\": 15,\n
\"max\": 137,\n        \"num_unique_values\": 64,\n
\"samples\": [\n          87,\n          101\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"Spending Score (1-100)\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
25,\n        \"min\": 1,\n        \"max\": 99,\n
\"num_unique_values\": 84,\n        \"samples\": [\n          83,\n
39\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    }\n  ]\
n}","type":"dataframe","variable_name":"df"}
```

```
# Select the relevant features for clustering
X = df[['Annual Income (k$)', 'Spending Score (1-100)']]

# Standardize the features
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)
```

```
# Apply DBSCAN
dbscan = DBSCAN(eps=0.3, min_samples=3)
dbscan_labels = dbscan.fit_predict(X_scaled)

# Add the cluster labels back to the original dataframe
df['Cluster'] = dbscan_labels

# Visualize clusters
plt.figure(figsize=(10, 6))
plt.scatter(X['Annual Income (k$)'], X['Spending Score (1-100)'],
c=dbscan_labels, cmap='plasma', s=50, alpha=0.7)
plt.title('DBSCAN Clustering of Customers Based on Annual Income (k$)
and Spending Score (1-100)')
plt.xlabel('Annual Income (k$)')
plt.ylabel('Spending Score (1-100)')
plt.colorbar(label='Cluster Label')
plt.show()
```

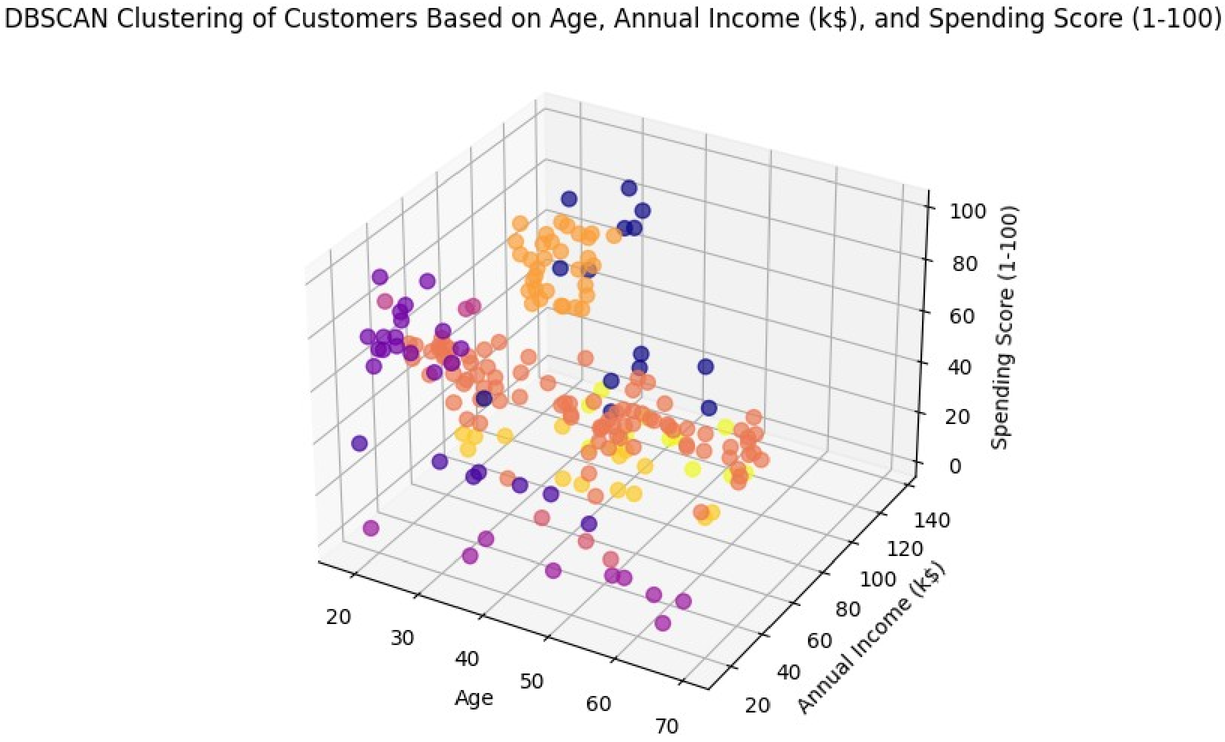


```
# Count the number of clusters (excluding noise)
num_clusters = len(set(dbscan_labels)) - (1 if -1 in dbscan_labels
else 0)

# Count the number of noise points
num_noise = list(dbscan_labels).count(-1)
```

```
print(f"Number of clusters: {num_clusters}")
print(f"Number of noise points: {num_noise}")
```

```
Number of clusters: 9
Number of noise points: 14
```

```
# 3D plot if you have more features
from mpl_toolkits.mplot3d import Axes3D

fig = plt.figure(figsize=(10, 6))
ax = fig.add_subplot(111, projection='3d')
ax.scatter(df['Age'], df['Annual Income (k$)'], df['Spending Score (1-
100)'], c=dbscan_labels, cmap='plasma', s=50, alpha=0.7)
ax.set_title('DBSCAN Clustering of Customers Based on Age, Annual
Income (k$), and Spending Score (1-100)')
ax.set_xlabel('Age')
ax.set_ylabel('Annual Income (k$)')
ax.set_zlabel('Spending Score (1-100)')
plt.show()
```



```
# Count the number of clusters (excluding noise)
num_clusters = len(set(dbscan_labels)) - (1 if -1 in dbscan_labels
else 0)

# Count the number of noise points
```

```python
num_noise = list(dbscan_labels).count(-1)

print(f"Number of clusters: {num_clusters}")
print(f"Number of noise points: {num_noise}")
```

```
Number of clusters: 9
Number of noise points: 14
```