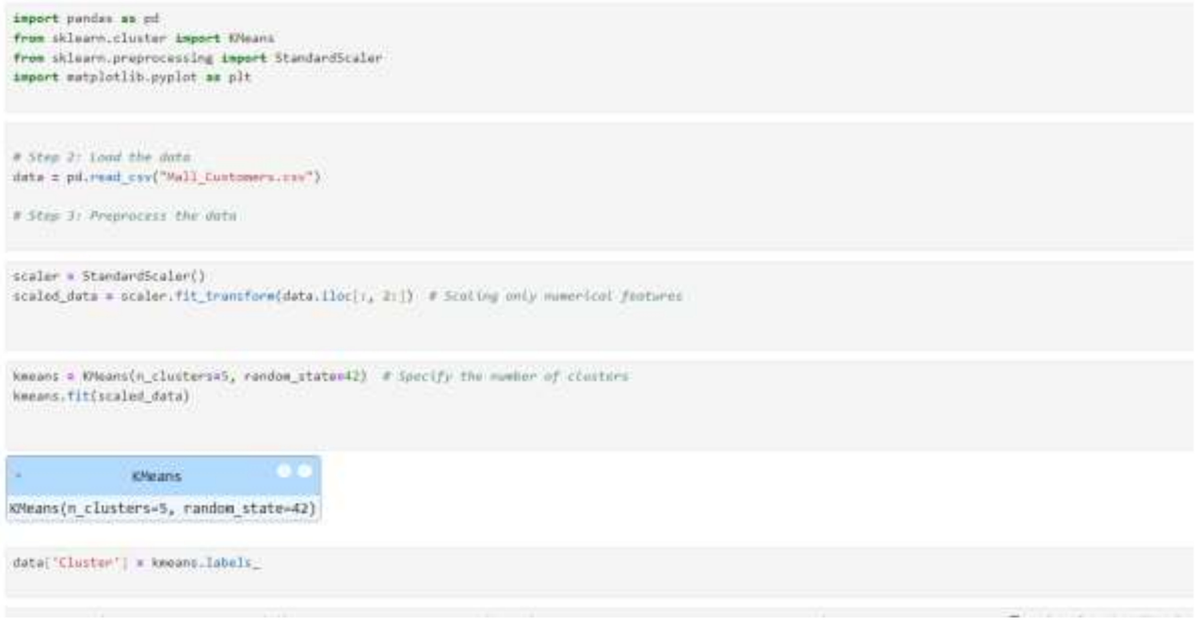
Name: Abhishek Mali
Roll No.: 213070
PRN No.: 22320024

Tutorial No.6

Implement following algorithms using Python on suitable data sets. i. K-Means ii.K-Medoids.



```
import pandas as pd
from sklearn.cluster import KMeans
from sklearn.preprocessing import StandardScaler
import matplotlib.pyplot as plt
```

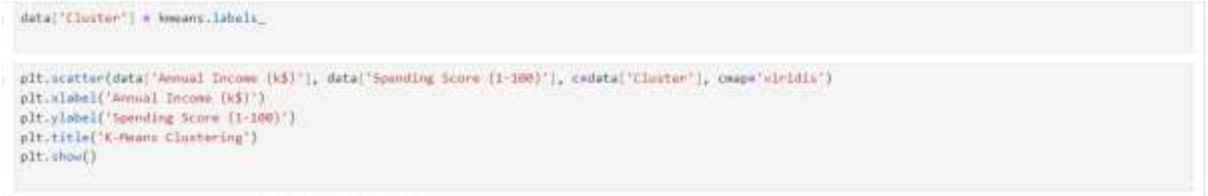
```
# Step 2: Load the data
data = pd.read_csv("Mall_Customers.csv")

# Step 3: Preprocess the data
```

```
scaler = StandardScaler()
scaled_data = scaler.fit_transform(data.iloc[:, 2:])  # Scaling only numerical features
```

```
kmeans = KMeans(n_clusters=5, random_state=42)  # Specify the number of clusters
kmeans.fit(scaled_data)
```

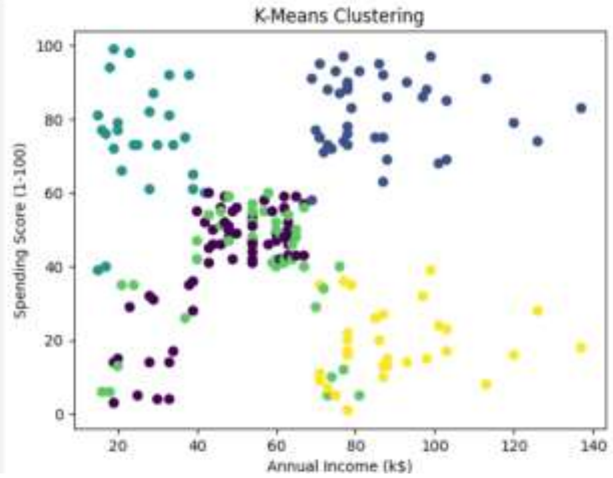
```
KMeans
KMeans(n_clusters=5, random_state=42)
```

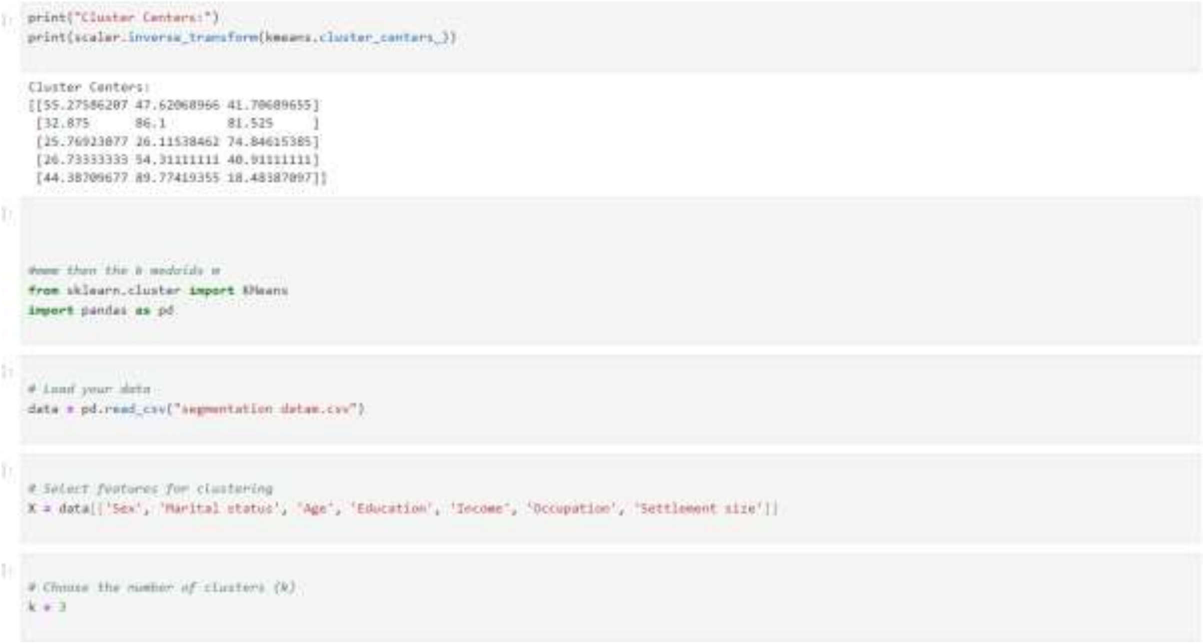
```
data['Cluster'] = kmeans.labels_
```



```
data['Cluster'] = kmeans.labels_
```

```
plt.scatter(data['Annual Income (k$)'], data['Spending Score (1-100)'], c=data['Cluster'], cmap='viridis')
plt.xlabel('Annual Income (k$)')
plt.ylabel('Spending Score (1-100)')
plt.title('K-Means Clustering')
plt.show()
```





```
print("Cluster Centers:")
print(scaler.inverse_transform(kmeans.cluster_centers_))
```

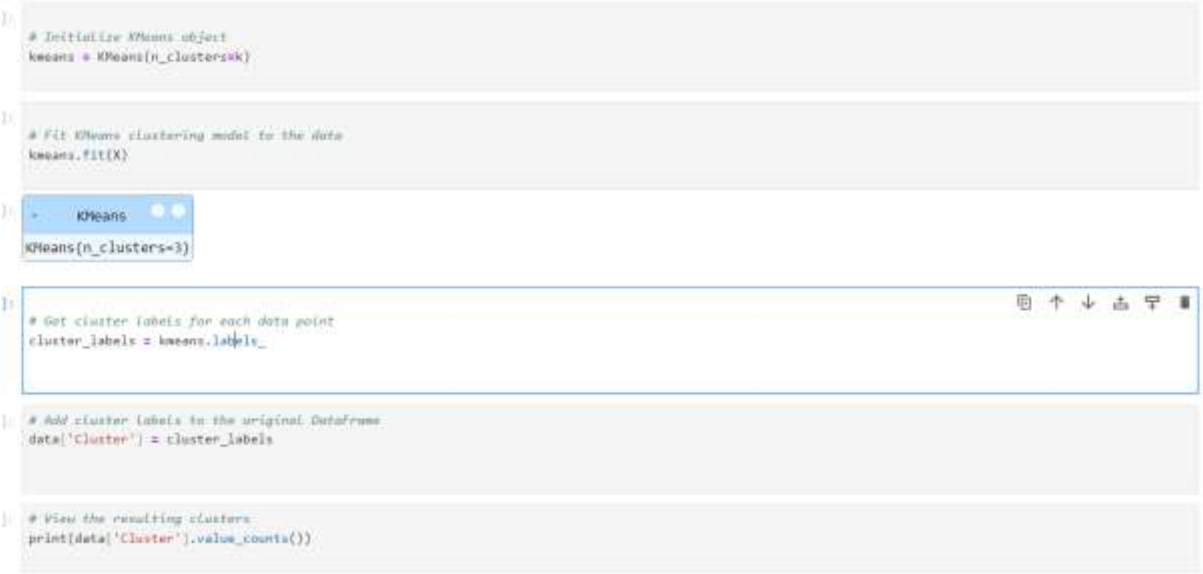
```
Cluster Centers:
[[55.27586207 47.62068966 41.70689655]
 [32.875      86.1        81.525     ]
 [25.76923077 26.11538462 74.84615385]
 [26.73333333 54.31111111 40.91111111]
 [44.38709677 89.77419355 18.48387097]]
```

```
#now then the k medoids #
from sklearn.cluster import KMeans
import pandas as pd
```

```
# Load your data
data = pd.read_csv("segmentation datam.csv")
```

```
# Select features for clustering
X = data[['Sex', 'Marital status', 'Age', 'Education', 'Income', 'Occupation', 'Settlement size']]
```

```
# Choose the number of clusters (k)
k = 3
```



```
# Initialize KMeans object
kmeans = KMeans(n_clusters=k)
```

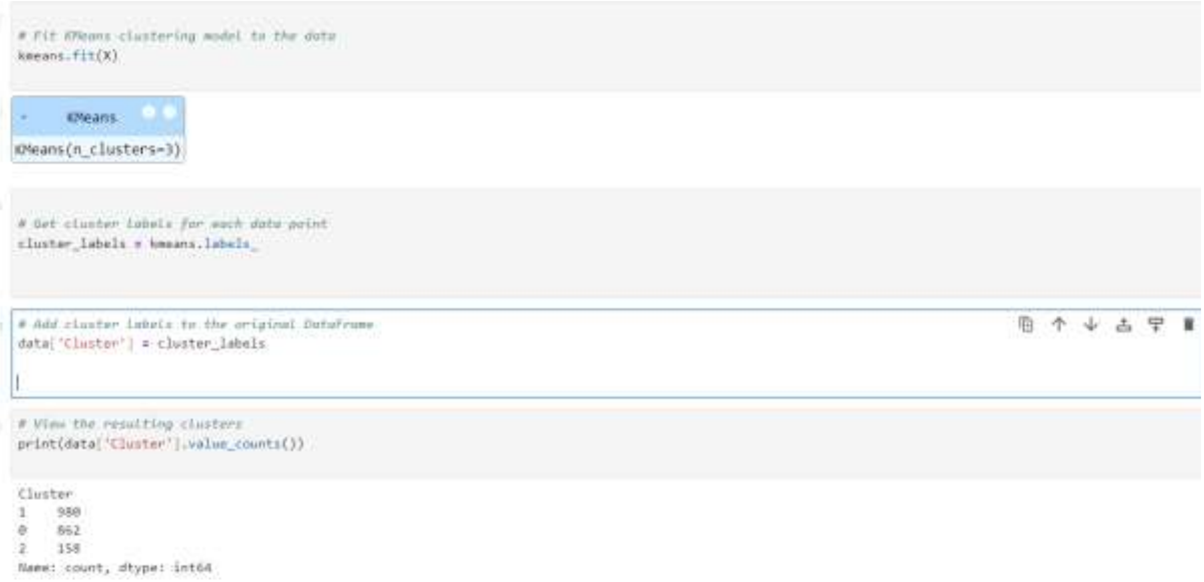
```
# Fit KMeans clustering model to the data
kmeans.fit(X)
```

```
KMeans
KMeans(n_clusters=3)
```

```
# Get cluster labels for each data point
cluster_labels = kmeans.labels_
```

```
# Add cluster labels to the original DataFrame
data['Cluster'] = cluster_labels
```

```
# View the resulting clusters
print(data['Cluster'].value_counts())
```



```
# Fit KMeans clustering model to the data
kmeans.fit(X)
```

```
KMeans(n_clusters=3)
```

```
# Get cluster labels for each data point
cluster_labels = kmeans.labels_
```

```
# Add cluster labels to the original DataFrame
data['Cluster'] = cluster_labels
```

```
# View the resulting clusters
print(data['Cluster'].value_counts())
```

```
Cluster
1    980
0    862
2    158
Name: count, dtype: int64
```