CYCLE - 5 : PROGRAM 2

Given dataset contains 200 records and five columns, two of which describe the customer’s annual income and spending score. The latter is a value from 0 to 100. The higher the number, the more this customer has spent with the company in the past:

Functions to familiarize:

* The purpose of Kmeans.fit() is to train the model with data.
* The purpose of Kmeans.predict() is to apply a trained model to data

==================================================

Q. Using k means clustering create 6 clusters of customers based on their spending pattern.

* Visualize the same in a scatter plot with each cluster in a different color scheme.
* Display the cluster labels of each point.(print cluster indexes)
* Display the cluster centers.
* Use different values of K and visualize the same using scatter plot

**Output:**

**from sklearn.cluster import KMeans**

**from sklearn.datasets import make\_blobs**

**import numpy as np**

**import matplotlib.pyplot as plt**

**import seaborn as sns**

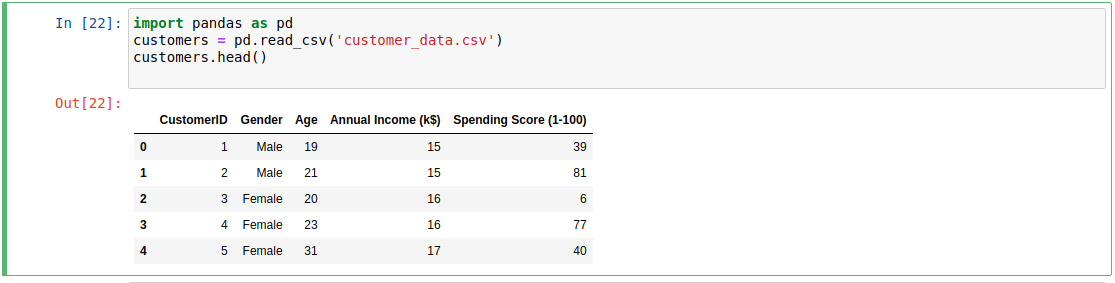
**sns.set()**

**%matplotlib inline**

**import pandas as pd**

**customers = pd.read\_csv('customer\_data.csv')**

**customers.head()**



**points = customers.iloc[:, 3:5].values**

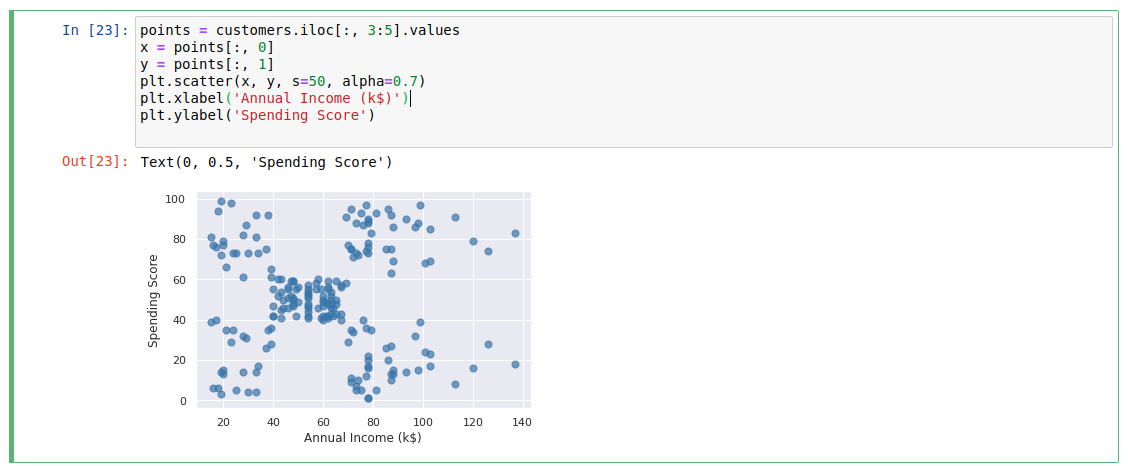
**x = points[:, 0]**

**y = points[:, 1]**

**plt.scatter(x, y, s=50, alpha=0.7)**

**plt.xlabel('Annual Income (k$)')**

**plt.ylabel('Spending Score')**

**kmeans = KMeans(n\_clusters=5, random\_state=0)**

**kmeans.fit(points)**

**predicted\_cluster\_indexes = kmeans.predict(points)**

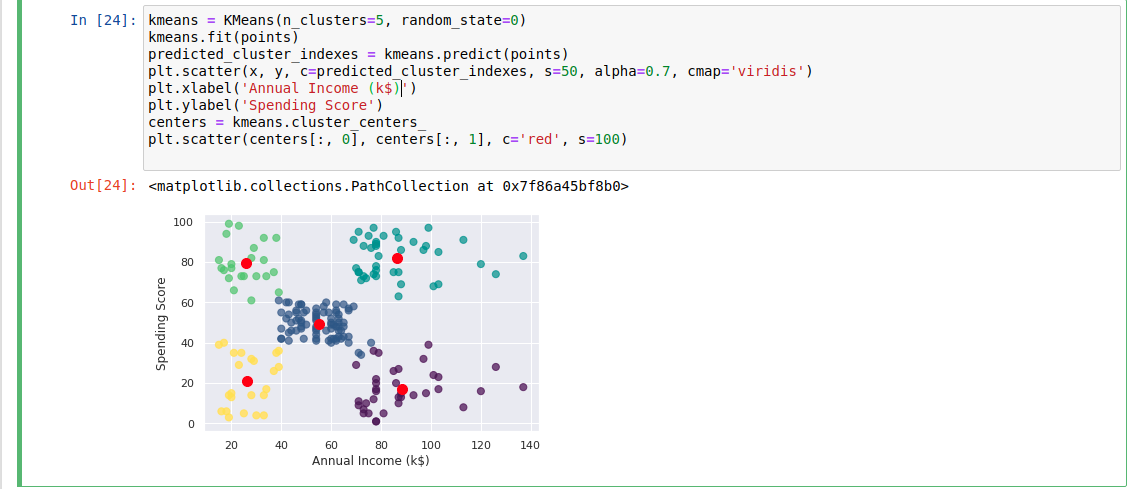
**plt.scatter(x, y, c=predicted\_cluster\_indexes, s=50, alpha=0.7, cmap='viridis')**

**plt.xlabel('Annual Income (k$)')**

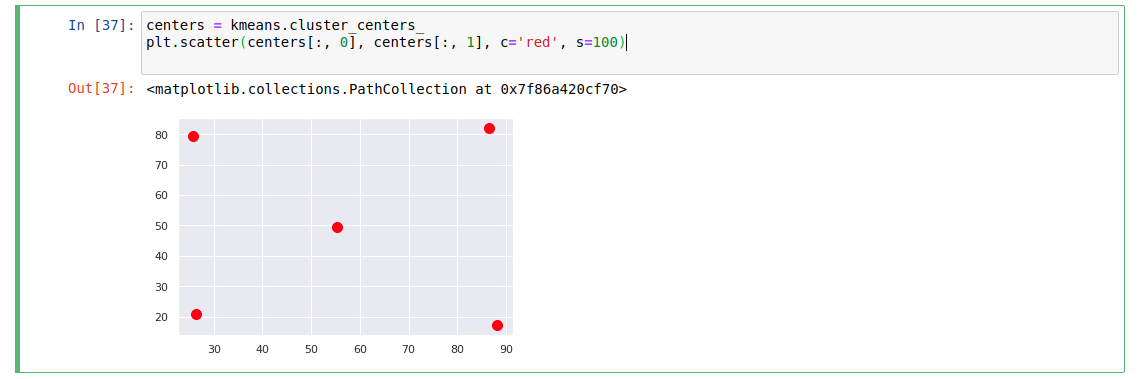
**plt.ylabel('Spending Score')**

**centers = kmeans.cluster\_centers\_**

**plt.scatter(centers[:, 0], centers[:, 1], c='red', s=100)**

**centers = kmeans.cluster\_centers\_**

**plt.scatter(centers[:, 0], centers[:, 1], c='red', s=100)**

****

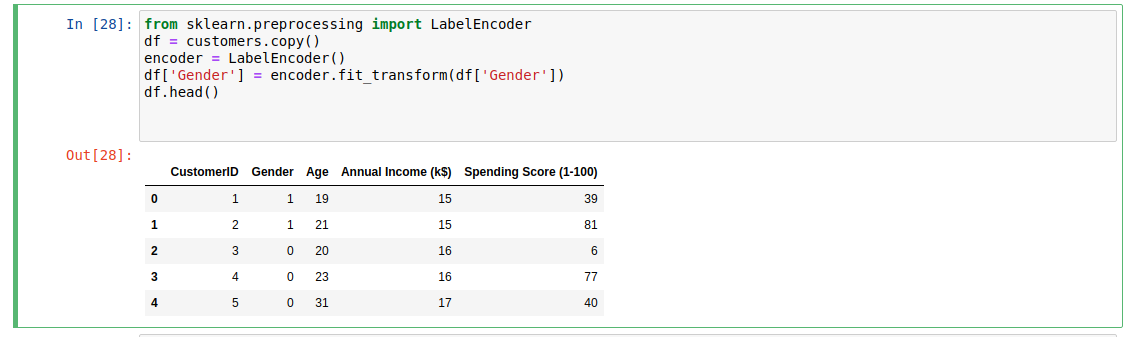
**from sklearn.preprocessing import LabelEncoder**

**df = customers.copy()**

**encoder = LabelEncoder()**

**df['Gender'] = encoder.fit\_transform(df['Gender'])**

**df.head()**



**points = df.iloc[:, 1:5].values**

**inertias = []**

**for i in range(1, 10):**

**kmeans = KMeans(n\_clusters=i, random\_state=0)**

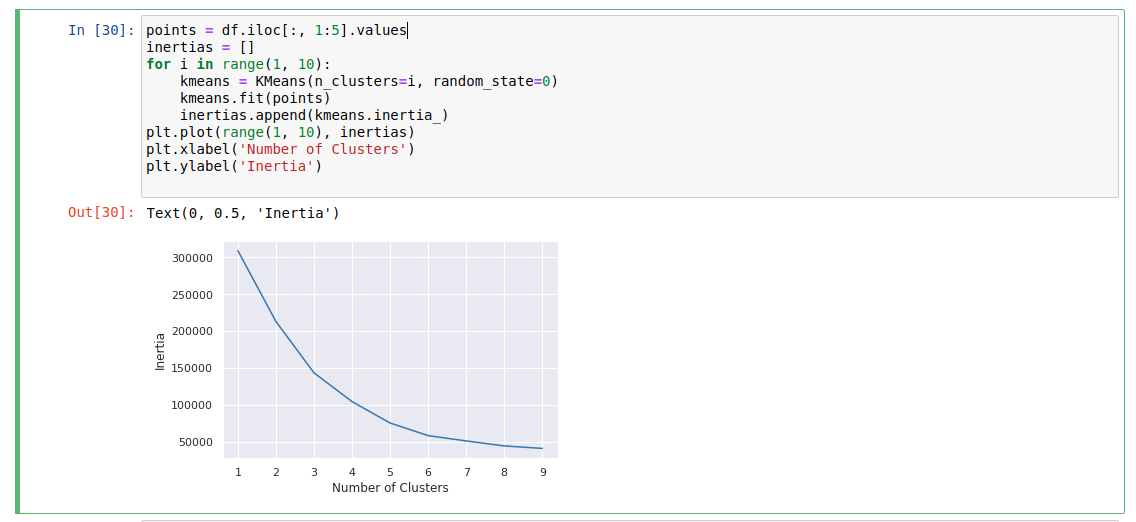
**kmeans.fit(points)**

**inertias.append(kmeans.inertia\_)**

**plt.plot(range(1, 10), inertias)**

**plt.xlabel('Number of Clusters')**

**plt.ylabel('Inertia')**



References:

<https://www.atmosera.com/blog/unsupervised-learning-with-k-means-clustering-part-ii/>

https://nickmccullum.com/python-machine-learning/k-means-clustering-python/