COMPUTING SUBJECT:	Machine Learning
TYPE:	WORK ASSIGNMENT
IDENTIFICATION:	Unsupervised learning & clustering
COPYRIGHT:	*Michael Claudius &Jens Peter Andersen*
DEGREE OF DIFFICULTY:	Medium
TIME CONSUMPTION:	3-6 hours
EXTENT:	< 200 lines
OBJECTIVE:	KMeans**,** Applying cluster inertias Using Scikit-Learn KMeans score & Silhouette score & DBScan
COMMANDS:	**KMeans, .inertia, plot**

Introduction

The Mission

Explore Scikit-Learns facilities on unsupervised learning and clustering.

Prerequisities

You have already solved the KMeans program from the book.

The problem

You have been hired as a data scientist for a client, which is a major mall in your area. They want assistance to segment their customers on basis on some data they have collected. The data has the following attributes:

- CustomerID: Customers ID in the malls database
- Gender: Customers gender
- Age: Customers age
- Annual income: Customers annual $-income in thousands
- Spending score: A measure 1-100 on how willing the customer is to spend money

Hint: Utilize the programs on KMeans and DBScan at your teacher's home page.
Alternatively get inspired from the code in the Jupyter Notebook for chapter 9. To be found in https://github.com/ageron/handson-ml2

Setup

Step 1: Setup actions

Start Jupyter Notebook and make a new notebook:ULMallCustomers
Import needed libraries:

```
import numpy as np
import matplotlib.pyplot as plt import pandas as pd
from sklearn.cluster import KMeans
from sklearn.metrics import silhouette_score from sklearn.cluster import DBSCAN
```

Establish the dataset as a Dataframe:

1. Download mall customer database from https://www.kaggle.com/shwetabh123/mall-customers and save it in the notebook folder.
2. Load the 'Mall_Customers.csv' dataset into the notebook.
   Suggestion: Load it in as a Dataframe (pandas.core.frame.DataFrame)

Find customer segments on income and spending

Step 2: Extract and inspect the dataset

Actions:

1. Extract '*annual income'* and '*spending score'* in an array (`numpy.ndarray`) with 2 columns for each of them.
2. Do a Scatter-Plot on this income-spending dataset.
   Code suggestion:
   ```
   plt.scatter(X[:, 0], X[:, 1], s = 25, c = 'black') plt.title('Plot of Income-Spending') plt.xlabel('Annual Income (k$)') plt.ylabel('Spending Score (1-100)') plt.show()
   ```
3. Do you spot any clusters by inspecting the diagram?

Step 3: KMeans and inertia

Actions:

1. Perform `KMeans.fit(..)` on the income-spending dataset over a range from 1 to 10 clusters.
   Tip; Make an array of KMeans with k = 1 up to k=10
2. Set up a loop and save the inertia in a list.
3. Print out all the inertia (divide by 1000 to get easy overview)
4. Keep the silhouette scores (`silhouette_score`) in a list.

Step 4: Elbow analysis: Determine number of clusters from inertia

Actions:

1. Plot the inertia curve – inertia vs. number of clusters.
2. Analyse the curve:
   a. How many clusters does it suggest?
   b. Explain the concepts of 'underfit' and 'overfit' in this context
3. What customer segmentation would you suggest to your client?

Step 5: Visualize and interpret the clusters

Actions:
1. Plot the clusters – if you like assign a color to each instance according to its' cluster label (1, 2, 3, ….).

Code suggestion:
```
plt.scatter(X[y_kmeans == 0, 0], X[y_kmeans == 0, 1], s = 25, c = 'red', label = 'Cluster 1')
plt.scatter(X[y_kmeans == 1, 0], X[y_kmeans == 1, 1], s = 25, c = 'blue', label = 'Cluster 2')
plt.scatter(kmeans.cluster_centers_[:, 0], kmeans.cluster_centers_[:, 1], s = 100, c = 'yellow', label = 'Centroids')
plt.title('Clusters of customers')
plt.xlabel('Annual Income (k$)') plt.ylabel('Spending Score (1-100)') plt.legend()
plt.show()
```

2. Interpret the clusters in by characterising each. You may then assign meaningful name labels to the clusters.

Step 6: Silhoutte score: Determine number of clusters from silhouette scores

Actions:
1. Plot the curve – silhouette score vs. number of clusters.
2. Analyse the curve: How many clusters does it suggest?
3. What customer segmentation would you suggest to your client on this basis?

Optional: Find customer segments on age and spending

Step 7: Extract and inspect another dataset (age-spending score)

Actions:
1. Extract 'age' and 'spending score' in an array (`numpy.ndarray`) with 2 columns for each of them.
2. Plot this income-spending dataset.

Step 8: Repeat steps 3 to 6 on this dataset (age-spending score)

Actions: Repeat steps 3, 4, 5, and 6.

Optional: Density and outliers

Step 9: Visualize density and possible outliers

Actions:
1. Perform DBSCAN on the income-spending dataset
   Suggestion: Try with parameters epsilon=8, min_samples=3

2. Run different epsilon 4 8 12 16 with a fixed number of min_samples
   Run different min_samples=3 5 10 f20 for fixed number of epsilon = 8

3. Display clusters and outliers graphically

4. Evaluate the clusters found with KMeans in the light of the information generated with DBSCAN