## Кластеризация

### Цель работы

Применить методы машинного обучения для решения задач кластеризации

### Задания для выполнения

1. Загрузите датасет mall\_customers.
2. Проверьте датасет на наличие пустых значений. Если есть пустые значения, то заменить их на наиболее подходящие.
3. Постройте графическую интерпретацию значений датасета. В качестве x-координаты используйте Annual Income, в качестве y-координаты Spending Score.
4. Определите наилучшее число кластеров с помощью графической зависимости числа кластеров от суммы квадратов расстояний значений датасета до центра кластера.
5. Обучить модель К-средних и предсказать кластер для каждого значения из датасета.
6. Построить графическую интерпретацию из пункта 3, с центрами кластеров.
7. Построить графическую интерпретацию с центрами кластеров, в качестве x-координаты используйте Возраст, в качестве y-координаты Spending Score.
8. Оцените модель с помощью метрик.
9. Сравните эффективность модели кластеризации в зависимости от количества кластеров. Сделайте выводы.

### Методические указания

Для начала работы нам потребуется импортировать необходимые библиотеки:

import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

from sklearn.metrics import silhouette\_score, calinski\_harabasz\_score

import matplotlib.cm as cm

from sklearn.cluster import KMeans

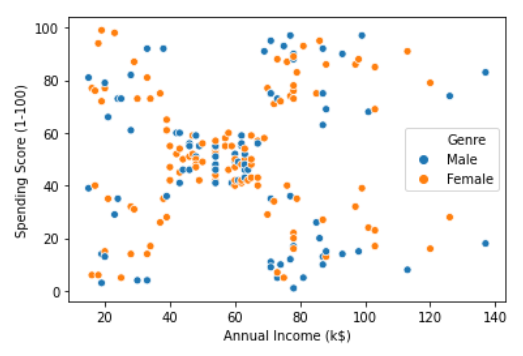
%matplotlib inline

В первую очередь загрузим датасет, выведем первые несколько строк:

df = pd.read\_csv('.../Mall\_Customers.csv')

df.head()

Построим графическую интерпретацию значений датасета. В качестве x-координаты возьмем годовой доход, а в качестве y-координаты оценку трат.



По графику можно предположить оптимальное количество кластеров для хорошего описания массива.

Проверим наши предположения с помощью перебора количества кластеров. Для этого построим на графике взаимосвязь суммы квадратов расстояний значений датасета до центра кластера от количества кластеров.

wcss = []

for i in range(1,11):

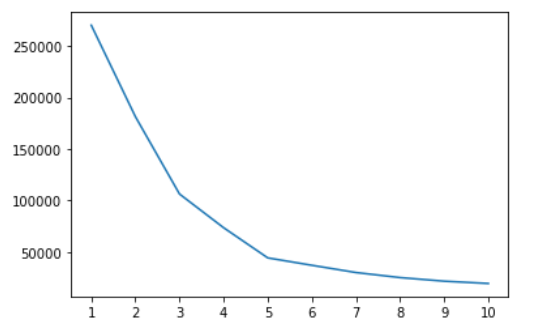
k\_means = KMeans(n\_clusters=i,random\_state=42)

k\_means.fit(x)

wcss.append(k\_means.inertia\_)

plt.plot(range(1,11),wcss)

plt.xticks(range(1,11))



В качестве количества кластеров возьмем число 5. Создадим модель К-средних и обучим её.

Построим графическую интерпретацию данных с центрами кластеров.

plt.scatter(x.Income, x.Score, c=y\_kmeans, s=20, cmap='viridis')

centers = kmeans.cluster\_centers\_

plt.scatter(centers[:, 1], centers[:, 2], c='black', s=200, alpha=0.5);

