

Лабораторная работа № 3

Распознавание образов с помощью обучения без учителя

1. Кластеризация данных методом k-средних (**KMeans**)
 1. Загрузить данные из файла `data_clustering.txt`.
 2. Визуально оценить распределение данных и задать количество кластеров, которое необходимо передать экземпляру `KMeans`.
 3. Алгоритм кластеризации `KMeans` итеративный, поэтому необходимо задать количество итераций `n_init` (по умолчанию равен 8).
 4. Отобразить график с результатами кластеризации (например, используя `plt.scatter`)
2. Кластеризация методом сдвига среднего (**MeanShift**)
 1. Загрузить данные из файла `data_clustering.txt`.
 2. Алгоритм кластеризации `MeanShift` итеративный, поэтому необходимо задать количество итераций `n_init` (по умолчанию равен 300), также необходимо задать размер окна для анализа соседних точек (`bandwidth`)
 3. Отобразить график с результатами кластеризации (например, используя `plt.scatter`)
3. Кластеризация иерархическим методом (**AgglomerativeClustering**)
 1. Загрузить данные из файла `data_clustering.txt`.
 2. Провести кластеризацию иерархическим методом, в качестве расстояния использовать евклидово расстояние.
 3. Самостоятельно определить значение параметра `linkage` (как считать расстояние между парой кластеров)
 4. Отобразить график с результатами кластеризации (например, используя `plt.scatter`)
4. Сравнить качество кластеризации, полученное тремя разными способами в пунктах 1-3 с помощью индекса Дэвиса-Булдина (`sklearn.metrics.davies_bouldin_score`)
5. Оценка качества кластеризации с помощью силуэтных оценок (`sklearn.metrics.silhouette_score`)
 1. Загрузить данные из файла `data_quality.txt`.
 2. Оценить качество кластеризации (методом k-средних), используя силуэтную оценку для разного числа кластеров.
 3. Построить гистограмму зависимости рассчитанной метрики от количества классов. Определить оптимальное количество классов.