МИНОБРНАУКИ РОССИИ

САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ

ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ «ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)

Кафедра Вычислительной техники

**ОТЧЕТ**

**по лабораторной работе №2**

по дисциплине «Машинное обучение»

Тема: «Кластеризация»

Студент группы 2308 Попов Н.А.

Преподаватель Турсуков Н.О.

Санкт-Петербург, 2024

## ОГЛАВЛЕНИЕ

[ТЕОРИЯ 3](#_Toc5733)

[ПРЕДОБРАБОТКА ДАННЫХ 4](#_Toc5734)

[МОДЕЛЬ K-MEANS 6](#_Toc5735)

[МОДЕЛЬ ИЕРАРХИЧЕСКОЙ КЛАСТЕРИЗАЦИИ 8](#_Toc5736)

[МОДЕЛЬ DBSCAN 9](#_Toc5737)

[ВЫВОД 11](#_Toc5738)

# ТЕОРИЯ

**Задача кластеризации.** Задача кластеризации в машинном обучении — это процесс группировки набора объектов (данных) в кластеры, так что объекты в одном кластере более схожи друг с другом, чем с объектами из других кластеров. Кластеризация относится к категории обучения без учителя, поскольку в отличие от обучения с учителем, где данные имеют метки, в кластеризации меток нет.

**Цели кластеризации:**

* Выявление скрытых структур в данных.
* Сегментация данных для дальнейшего анализа.
* Снижение размерности данных для визуализации.

**Методы кластеризации:**

* Иерархическая кластеризация – подход, который начинает с отдельных объектов и последовательно объединяет их.
* K-средних (K-means) – метод, который делит данные на K кластеров на основе минимизации среднеквадратичной ошибки.
* DBSCAN – метод, основанный на плотности, который выделяет кластеры по плотным областям данных и игнорирует шум.

**Метрики оценки качества кластеризации:**

* Силуэт: измеряет, насколько хорошо объекты в одном кластере отделены от объектов других кластеров.
* Коэффициент Дэвиса–Боллдина: оценивает компактность и разделимость кластеров.

# ПРЕДОБРАБОТКА ДАННЫХ

 **Удаление лишних столбцов**  
Удален столбец Unnamed: 0, который не содержит значимой информации для анализа:

data.drop('Unnamed: 0', axis=1, inplace=True)

 **Выбор значимых признаков**  
Оставлены только значимые признаки:

* Осадки (Rainfall\_mm)
* Использование удобрений (Fertilizer\_Used)
* Использование орошения (Irrigation\_Used)
* Урожайность в тоннах с гектара (Yield\_tons\_per\_hectare)

data = data[['Rainfall\_mm', 'Fertilizer\_Used', 'Irrigation\_Used', 'Yield\_tons\_per\_hectare']]

 **Нормализация данных**  
Для приведения данных к одному масштабу применено стандартизирующее преобразование.

scaler = StandardScaler()

scaled\_data = scaler.fit\_transform(data)

 **Разделение данных**  
Данные разделены на следующие наборы:

* Основной набор для подготовки (data\_pre) — первые 900,000 строк.
* Тестовый набор (data\_test) — последние строки после 900,000.
* В пределах основного набора данные разделены на:
  + Тренировочный (data\_train) — первые 630,000 строк.
  + Валидационный (data\_val) — оставшиеся строки (630,000–900,000).

data\_pre, data\_test = data[:900000], data[900000:]

data\_train = data\_pre[:630000]

data\_val = data\_pre[630000:900000]

 **Нормализованные наборы данных**  
Созданы аналогичные наборы данных в нормализованном виде:

* Основной нормализованный набор (data\_pre\_scaled)
* Тестовый нормализованный набор (data\_test\_scaled)
* Тренировочный нормализованный набор (data\_train\_scaled)
* Валидационный нормализованный набор (data\_val\_scaled)

data\_pre\_scaled, data\_test\_scaled = scaled\_data[:900000], scaled\_data[900000:]

data\_train\_scaled = scaled\_data[:630000]

data\_val\_scaled = scaled\_data[630000:900000]

 **Анализ корреляции признаков**  
Построена корреляционная матрица для анализа взаимосвязей между признаками. Для визуализации применена цветовая шкала coolwarm:

corr = data.corr()

corr.style.background\_gradient(cmap='coolwarm')

Полученная корреляционная матрица поможет определить силу взаимосвязи между входными данными и целевой переменной. (Рис. 1)

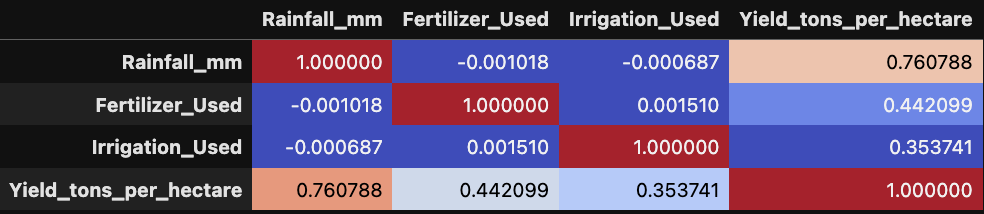


Рисунок 1 – Корреляционная матрица

# МОДЕЛЬ K-MEANS

Основная идея алгоритма заключается в том, чтобы минимизировать внутрикластерное расстояние, то есть расстояние между точками в одном кластере (или инерцию).

Для определения оптимального числа кластеров можно использовать метод «локтя». Он основан на анализе зависимости между числом кластеров и ошибкой (или инерцией) кластеризации – рисунок 4.

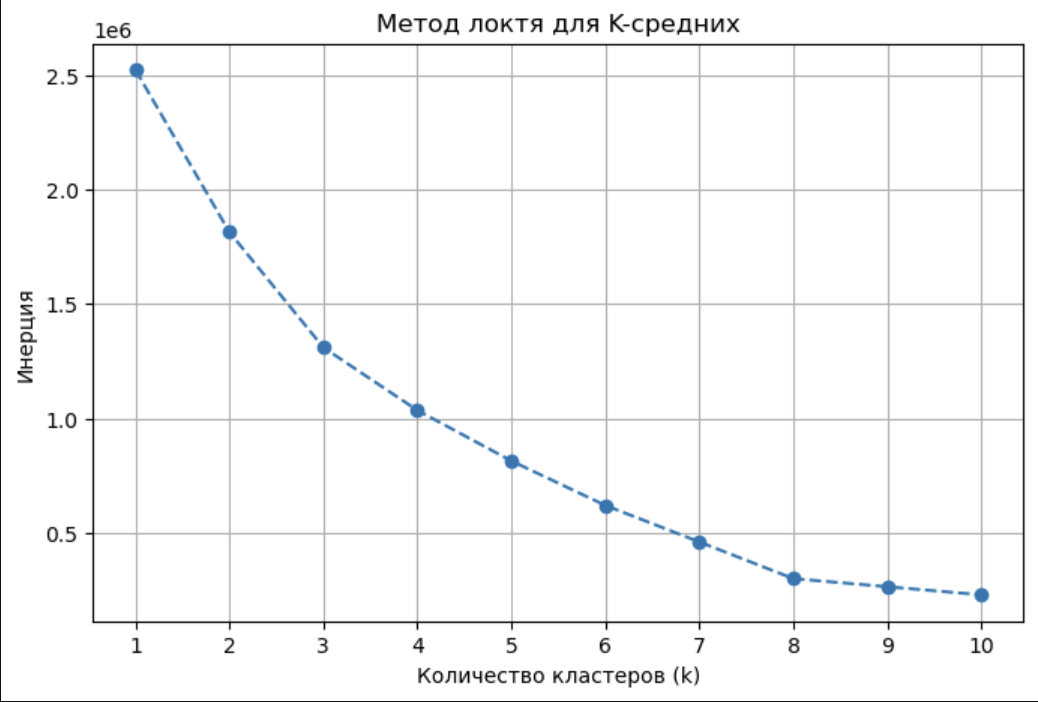


Рисунок 4.

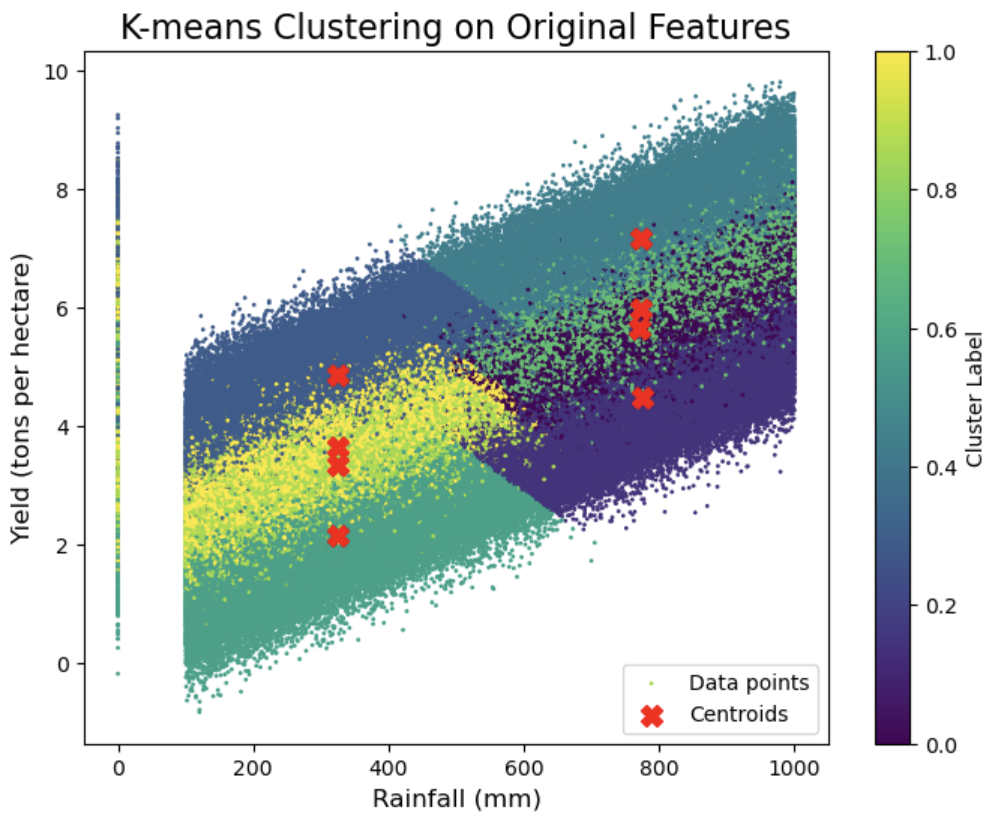
По методу «локтя» определим возможные значения числа кластеров и визуализируем разбиения – рисунок 5. 

Рисунок 5.

Для 8 кластеров получим метрики: DBI = 0.58, S = 0.55.

# МОДЕЛЬ ИЕРАРХИЧЕСКОЙ КЛАСТЕРИЗАЦИИ

Метод создает иерархическую структуру (дерево), которая показывает, как объекты объединяются в кластеры на разных уровнях. Алгоритм начинает с небольших кластеров (обычно с кластеров, состоящих из одного объекта) и постепенно объединяет их в кластеры побольше:

1. Создаём столько кластеров, сколько у нас объектов в выборке, каждый объект — в своём отдельном кластере.
2. Повторяем итеративно слияние двух ближайших кластеров, пока не выполнится критерий останова.

Обычно кластеризацию проводят вплоть до одного кластера, включающего в себя всю выборку, а затем анализируют получившуюся иерархическую дендрограмму. На дендрограмме мы можем визуально заметить, в какой момент происходит скачок расстояний между кластерами, и попытаться определить «естественное» количество кластеров в нашей задаче.

Визуализируем дендограмму – рисунок 6.

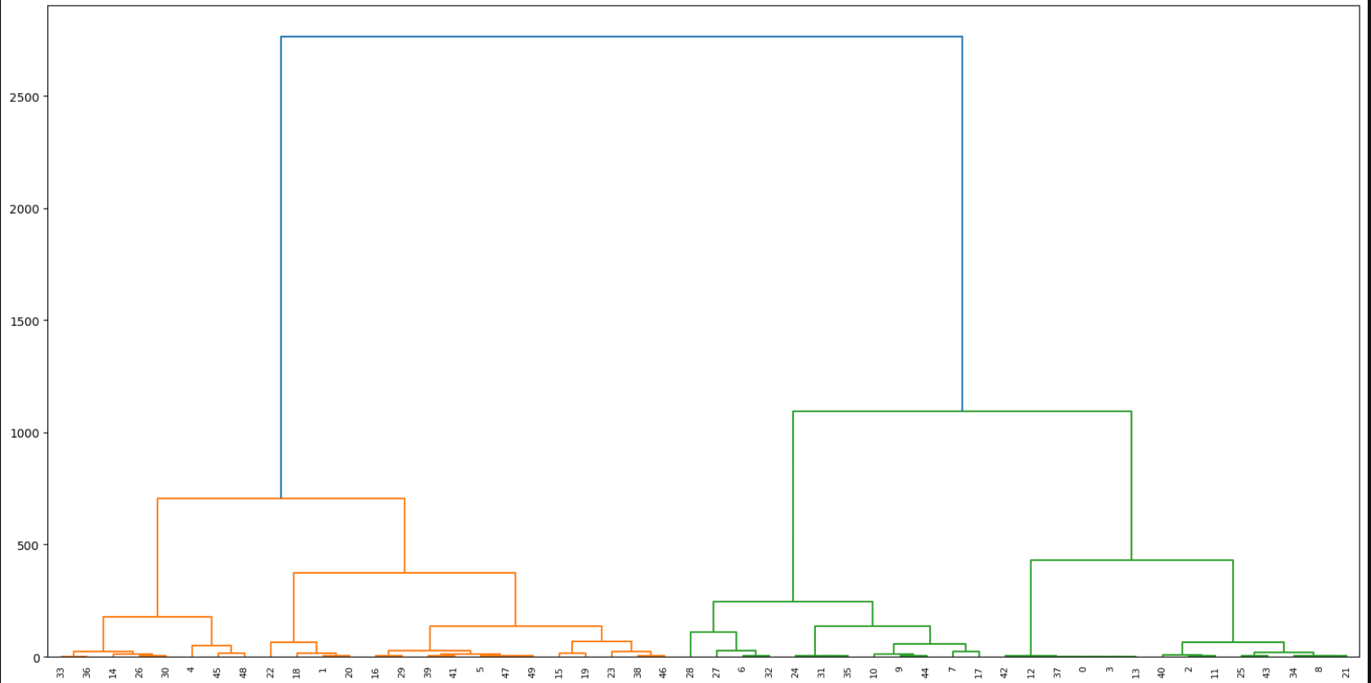


Рисунок 6.

Метрики для 8 кластеров: DBI = 0.87, S = 0.46.

# МОДЕЛЬ DBSCAN

DBSCAN (Density-Based Spatial Clustering of Applications with Noise) — это алгоритм кластеризации, который группирует точки на основе их плотности. Он особенно эффективен для обнаружения кластеров произвольной формы и может справляться с шумом в данных.

DBSCAN классифицирует все объекты на 3 типа:

1. **ядровые** - имеют не менее n соседей внутри шара радиусом ϵ;
2. **пограничные** - находятся в ϵ-шаре хотя бы одной ядровой точки, но сами содержат имеют менее n соседей в ридиусе ϵ;
3. **шумовые** - не являются ни ядровыми, ни пограничными.

Алгоритм:

1. Алгоритм начинает с произвольной точки и ищет все её соседние точки в пределах радиуса ε.
2. Если найдено достаточно соседей (≥ minPts), формируется новый кластер.
3. Алгоритм продолжает добавлять соседние точки к кластеру, пока не будут обработаны все точки.

DBSCAN очень чувствителен к значению параметров n и ϵ, поэтому вся сложность заключается в правильном подборе этих гиперпарметров.

Немного "поигравшись" с параметрами вручную можно получить не самое плохое разделение – рисунок 9.

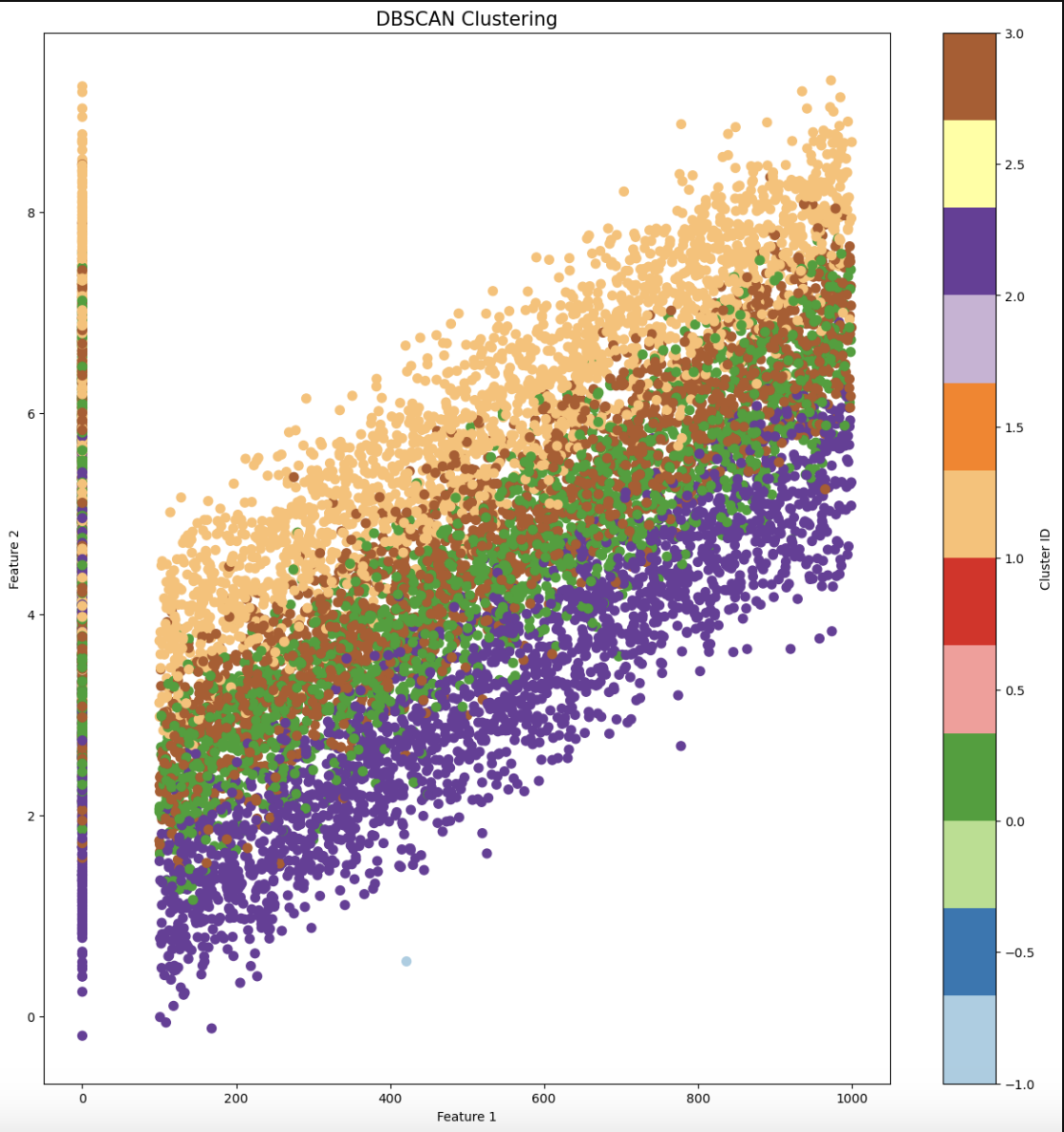


Рисунок 9.

К сожалению, идеального разбиения с помощью DBSCAN вряд ли удастся добиться в моей задаче, так как алгоритм плохо справляется с "размытыми" границами кластеров.

Метрики для 8 кластеров: DBI = 1.07, S = 0.42.

# ВЫВОД

В рамках лабораторной работы были использованы три метода кластеризации: **k-means**, **иерархическая кластеризация** и **DBSCAN**. Каждый из них имеет свои особенности и области применения, зависящие от структуры данных.

1. **Метод k-means** продемонстрировал высокую эффективность при выделении четко очерченных кластеров. Этот метод отлично подходит для задач, где кластеры имеют сходную плотность и форму. Применение k-means позволило получить четкие границы кластеров, что упростило визуализацию и анализ распределения данных.
2. **Иерархическая кластеризация** показала себя особенно полезной при необходимости изучения иерархической структуры данных. Этот метод позволил не только сгруппировать данные, но и выявить взаимосвязи между отдельными кластерами на разных уровнях агрегации. Полученные "размытые" границы кластеров обеспечили гибкость в интерпретации результатов.
3. **DBSCAN**, предназначенный для работы с шумовыми точками и кластерами произвольной формы, не оправдал ожиданий в данной задаче. Его эффективность оказалась ограничена из-за высокой чувствительности к параметрам и трудностей при работе с неравномерной плотностью данных.

На основании проведенного анализа можно сделать вывод, что методы **k-means** и **иерархической кластеризации** оказались наиболее подходящими для поставленной задачи. Их способность выделять кластеры с "размытыми" границами обеспечила более точное и информативное распределение данных, чем метод DBSCAN.