МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение

высшего образования «Южно-Уральский государственный университет»

(национальный исследовательский университет)

Высшая школа электроники и компьютерных наук

Кафедра системного программирования

кластеризация данных

(название практической работы)

ОТЧЕТ

по практической работе № 2

по дисциплине «Интеллектуальные системы и технологии»

Выполнил:

студент группы КЭ–401

/ Д.А. Иванов /

(подпись)

« » 2024 г.

Проверил:

преподаватель кафедры СП

/ С.М. Абдуллаев /

(подпись)

« » 2024 г.

Оглавление

[Введение 3](#_Toc178778583)

[1. Кластеризации данных 4](#_Toc178778584)

[2. Иерархическая кластеризация 6](#_Toc178778585)

[3. итерационная кластеризация 9](#_Toc178778586)

[4. Понижение размерности с методом t-SNE 10](#_Toc178778587)

[библиографический список 12](#_Toc178778588)

# Введение

Линейная регрессия - самый простой инструмент выявления зависимости между различными числовыми признаками. Часто линейную регрессию относят к методам машинного обучения, однако на самом деле – это не так. Восстановить линейную регрессию – то есть линейную зависимость между различными числовыми данными – можно с помощью аналитического решения. Тем не менее, так как линейная регрессия решает ту же задачу, что и машинное обучение - задачу восстановления зависимости между данными - для регрессии применяются те же термины и понятия, что и для машинного обучения.

Однако, не всегда требуется восстановить полную зависимость. Иногда достаточно определить то, к какому "типу" или "классу" относится объект. Задачи такого рода называются задачами классификации.

Обучение без учителя – класс методов машинного обучения для поиска закономерностей в наборе данных. Кластеризация – разделение большого множества объектов на кластеры – классы, внутри которых объекты похожи между собой. Кластеризация –не контролируемое обучение без учителя, поскольку в таких задачах отсутствует целевая переменная и разметка. Алгоритм пытается структурировать данные, о которых мало что известно заранее.

# Кластеризации данных

Для составления прогнозов воспользуемся классическим набором данных ирисов Фишера. Датасет представляет набор из 150 записей с пятью атрибутами в следующем порядке: длина чашелистика (sepal length), ширина чашелистика (sepal width), длина лепестка (petal length), ширина лепестка (petal width) и класс, соответствующий одному из трех видов: Iris Setosa, Iris Versicolor или Iris Virginica, обозначенных соответственно 0, 1, 2. Наш алгоритм должен принимать четыре свойства одного конкретного цветка и предсказывать, к какому классу (виду ириса) он принадлежит. Имеющиеся в наборе данных метки можно использовать для оценки качества предсказания.

Для решения задач кластеризации используются Python, библиотеки scikit-learn, для загрузки и обработки набора данных и matplotlib, для визуализации. На листинге 1 представлен программный код для исследования исходного набора данных.

Листинг 1 – Исследование набора данных

# Импортируем библиотеки

from sklearn import datasets

import matplotlib.pyplot as plt

# Загружаем набор данных

iris\_df = datasets.load\_iris()

# Методы, доступные для набора данных

print(dir(iris\_df))

# Признаки

print(iris\_df.feature\_names)

# Метки

print(iris\_df.target)

# Имена меток

print(iris\_df.target\_names)

# Разделение набора данных

x\_axis = iris\_df.data[:, 0] # Sepal Length

y\_axis = iris\_df.data[:, 1] # Sepal Width

# Построение

plt.xlabel(iris\_df.feature\_names[0])

plt.ylabel(iris\_df.feature\_names[1])

plt.scatter(x\_axis, y\_axis, c=iris\_df.target)

plt.show()

В результате запуска программы вы увидим изображение представленное на рисунке 1.

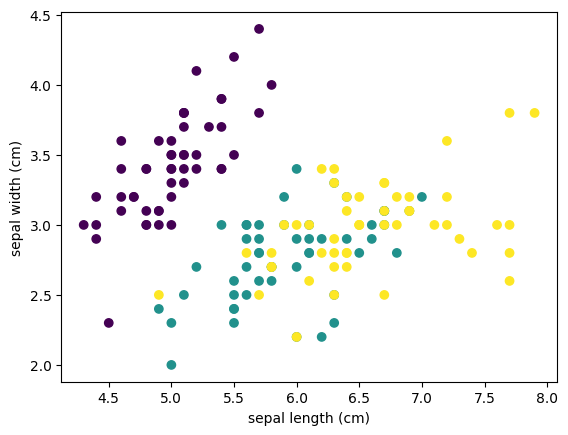


Рисунок 1 – Результат кластеризации

На диаграмме фиолетовым цветом обозначен вид Setosa, зеленым – Versicolor и желтым – Virginica.

# Иерархическая кластеризация

Иерархическая кластеризация, как следует из названия, представляет собой алгоритм, который строит иерархию кластеров. Этот алгоритм начинает работу с того, что каждому экземпляру данных сопоставляется свой собственный кластер. Затем два ближайших кластера объединяются в один и так далее, пока не будет образован один общий кластер.

Результат иерархической кластеризации может быть представлен с помощью дендрограммы.

На листинге 2 представлен код для составления иерархии кластеров и вывода дендограммы.

Листинг 2 – Иерархическая кластеризация

# Подключение библиотек

from scipy.cluster.hierarchy import linkage, dendrogram

from sklearn import datasets

import matplotlib.pyplot as plt

# Создание полотна для рисования

fig = plt.figure(figsize=(15, 30))

fig.patch.set\_facecolor('white')

# Загрузка набора данных "Ирисы Фишера"

iris = datasets.load\_iris()

# Реализация иерархической кластеризации при помощи функции linkage

mergings = linkage(iris.data, method='ward')

# Построение дендрограммы. Разными цветами выделены автоматически определенные кластеры

R = dendrogram(mergings, labels=[iris.target\_names[i] for i in iris.target], orientation = 'left', leaf\_font\_size = 12)

# Отображение дендрограммы

plt.show()

На рисунке 2 представлена дендограмма на основе иерархии кластеров.

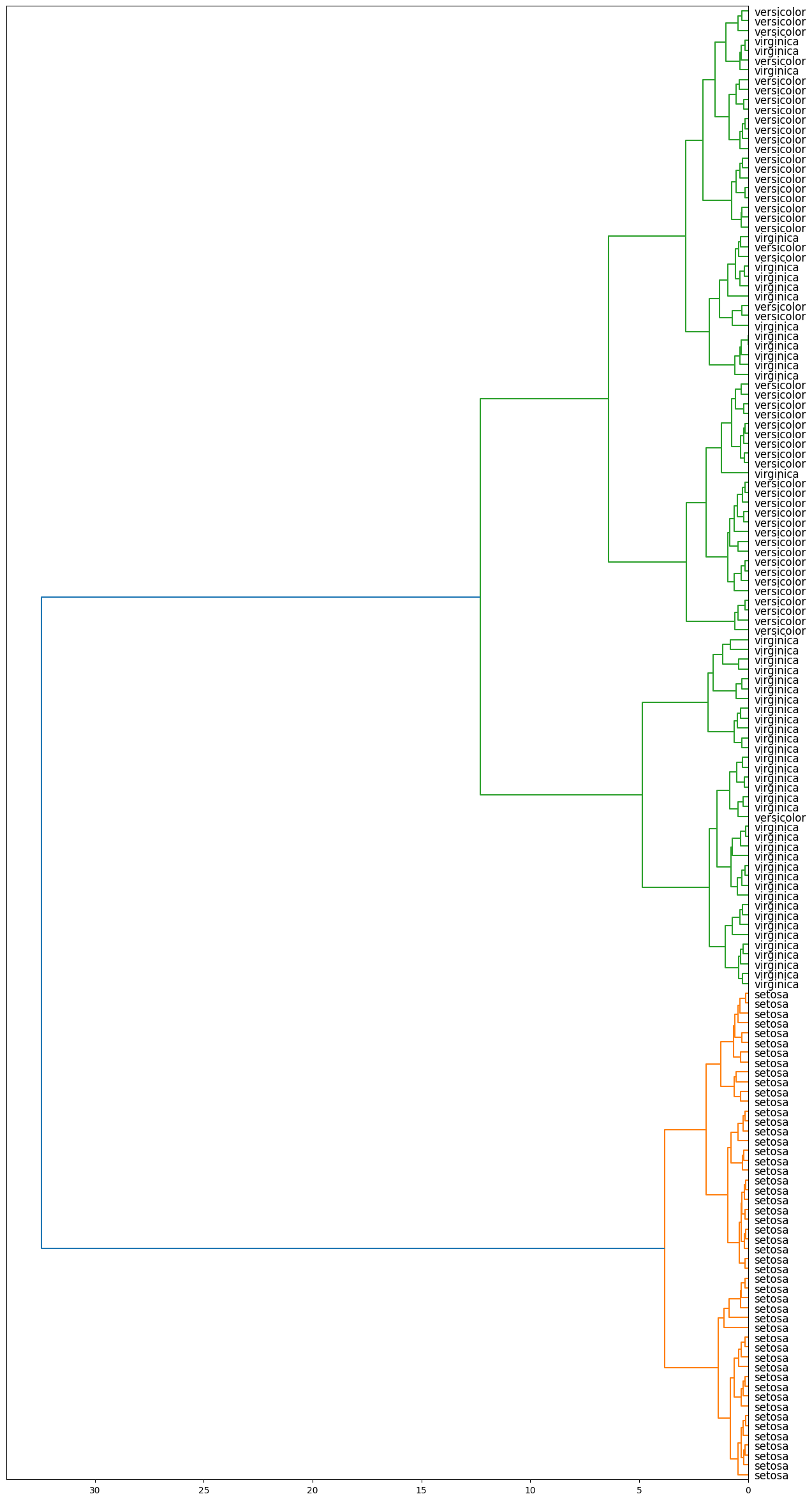


Рисунок 2 – Дендограмма

Алгоритм кластеризации точно выделил класс Iris setosa и заметно отделил вид Iris virginica от Iris versicolor.

Иерархические алгоритмы применяются, когда истинное количество групп не известно. Перед использованием иерархических методов необходимо исключить выбросы и пронормировать исходные данные, т.е. привести их к единой шкале измерений. Это необходимо по следующим причинам. Суть подхода заключается в измерении попарных расстояний между объектами и последовательном объединении или наоборот, дроблении, где соседние объекты похожи друг на друга.

# итерационная кластеризация

Наиболее популярным алгоритмом кластеризации данных является метод k-средних. Это итеративный алгоритм кластеризации, основанный на минимизации суммарных квадратичных отклонений точек кластеров от центроидов (средних координат) этих кластеров.

Первоначально выбирается желаемое количество кластеров. Известно, что в используемом наборе данных есть 3 класса, так что в данном случае k = 3.

Теперь случайным образом из входных данных выбираются три элемента выборки, в соответствие которым ставятся три кластера, в каждый из которых теперь включено по одной точке, каждая при этом является центроидом этого кластера.

Далее происходит поиск ближайшего соседа текущего центроида. Точка добавляется к соответствующему кластеру и пересчитывается положение центроида с учетом координат новых точек. Алгоритм заканчивает работу, когда координаты каждого центроида перестают меняться. Центроид каждого кластера в результате представляет собой набор значений признаков, описывающих усредненные параметры выделенных классов.

На рисунке 3 представлена схема работы этого метода.

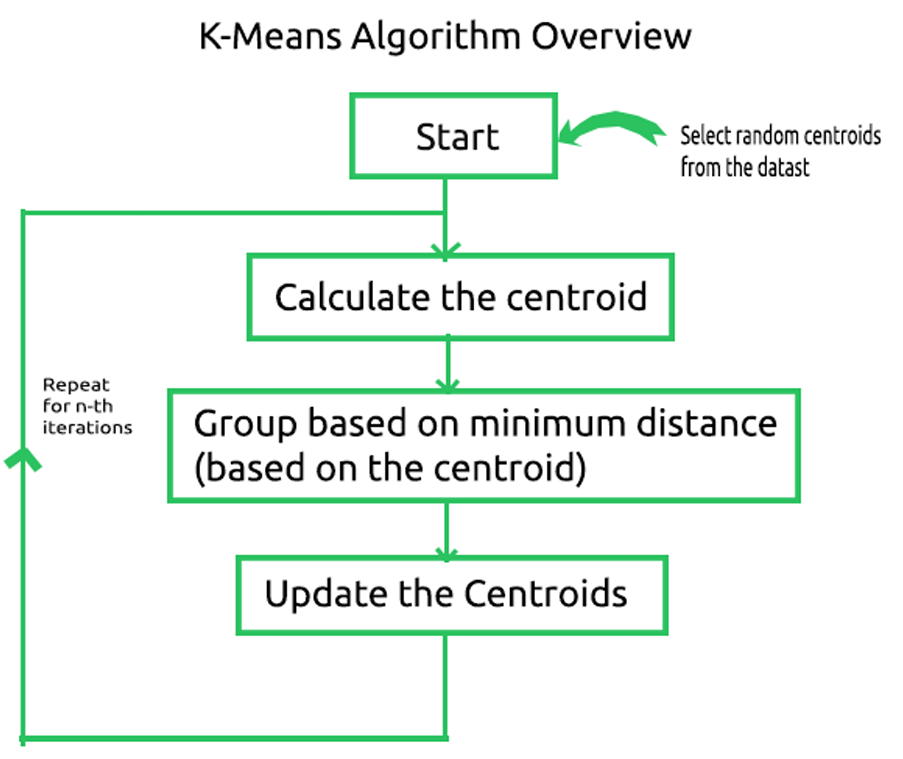


Рисунок 3 – Блок-схема метода k-средних

# Понижение размерности с методом t-SNE

Метод t-SNE (t-distributed stochastic neighbor embedding) представляет собой один из методов обучения без учителя, используемых для визуализации, например, отображения пространства высокой размерности в двух- или трехмерное пространство. t-SNE расшифровывается как распределенное стохастическое соседнее вложение.

Метод моделирует каждый объект пространства высокой размерности в двух- или трехкоординатную точку таким образом, что близкие по характеристикам элементы данных в многомерном пространстве (например, датасете с большим числом столбцов) проецируются в соседние точки, а разнородные объекты с большей вероятностью моделируются точками, далеко отстоящими друг от друга.

На листинге 3 представлен код для моделирования методом t-SNE.

Листинг 3 – Метод t-SNE

# Импорт библиотек

from sklearn import datasets

from sklearn.manifold import TSNE

import matplotlib.pyplot as plt

# Загрузка датасета

iris\_df = datasets.load\_iris()

# Определяем модель и скорость обучения

model = TSNE(learning\_rate=100)

# Обучаем модель

transformed = model.fit\_transform(iris\_df.data)

# Представляем результат в двумерных координатах

x\_axis = transformed[:, 0]

y\_axis = transformed[:, 1]

plt.scatter(x\_axis, y\_axis, c=iris\_df.target)

plt.show()

На рисунке 4 представлена результат этой программы.

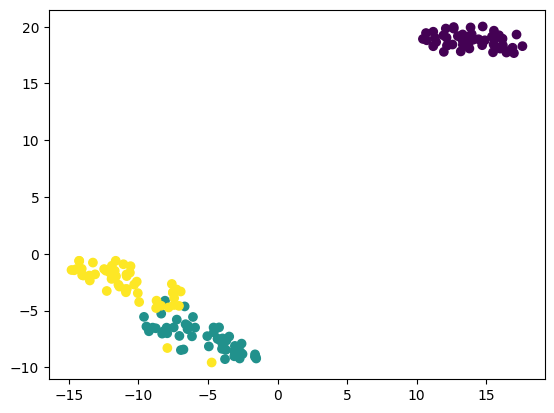


Рисунок 4 – Метод t-SNE

# библиографический список

1. Халил, З.Х., Абдуллаев С.М. Pегрессионные прогнозы урожайности орошаемых озимых культур с использованием спутниковыx вегетационных индексов: модели, предикторы и эксперименты //Вестник Южно-Уральского государственного университета. Серия "Компьютерные технологии, управление, радиоэлектроника".–2021.–Том 21 №2.– C.23-35 DOI:10.14529/ctcr210203
2. Салал, Я. Абдуллаев С.М Приложение теорем «No Free Lunch» в задаче ансамблевой классификации несбалансированных выборок данных. //Информатизация и связь.–2020 № 6.– C.46-50
3. Салал Я.Х. Абдуллаев, С.М., «Программный комплекс для прогноза когнитивного потенциала учащихся» //Свидетельство Роспатента о государственной регистрации программы для ЭВМ. Заявка № 2021617282 от 13 мая 2021, правообладатель: ФГАОУ ВО "ЮУрГУ (НИУ)
4. https://habr.com/ru/articles/79756/
5. <https://biomolecula.ru/articles/klasternyi-analiz-i-dilemma-biologicheskogo-polzovatelia>
6. <https://proglib.io/p/unsupervised-ml-with-python>
7. Agrawal R., Imieliński T., Swami A. Mining association rules between sets of items in large databases // Proceedings of the 1993 ACM SIGMOD international conference on Management of data - SIGMOD '93. — 1993. — ISBN 0897915925. — doi:10.1145/170035.170072
8. Amigó E, Gonzalo J, Artiles J et al. A comparison of extrinsic clustering evaluation metrics based on formal constraints.// Information Retrieval Journal 12009. 2(4):461486. DOI:10.1007/s10791-008-9066-8