Министерство образования Республики Беларусь

Учреждение образования

“Белорусский государственный университет

информатики и радиоэлектроники”

Факультет информационных технологий и управления

Кафедра интеллектуальных информационных технологий

**Лабораторная работа №3**

по дисциплине «Статистические основы индуктивного вывода»

Выполнил: Поплавский Э. Э.

Группа: xxxxxx

Проверил: Name N. N.

Минск 2023

1. **Цели**

Приобрести навыки построения модели для кластеризации методом 𝑘-средних при обучении без учителя.

1. **Задание**
   1. Выбрать с сайта kaggle.com набор данных в формате .csv, пригодный для построения модели кластеризации, загрузить и подготовить его к дальнейшей обработке. Если набор уже имеет метки классов – удалить метки.
   2. Построить модель кластеризации методом 𝑘-средних
   3. Определить оптимальное количество кластеров, обосновать своё решение
   4. Получить центроиды кластеров и количество вхождений в кластеры
   5. Интерпретировать каждый кластер
   6. Указать какие знания можно получить из набора
   7. Сохранить IPython Notebook
2. **Выполнение**

Для выполнения лабораторной работы был выбран набор данных, описывающий цветы Ириса.

Набор данных о цветках ириса — это многомерный набор данных, представленный британским статистиком и биологом Рональдом Фишером в его статье 1936 года "Использование множественных измерений в таксономических задачах". Иногда его называют набором данных по ирису Андерсона, потому что Эдгар Андерсон собрал данные для количественной оценки морфологических изменений цветков ириса трех родственных видов. Набор данных состоит из 50 образцов каждого из трех видов ирисов (Iris Setosa, Iris Virginica и Iris Versicolor). В каждом образце были измерены четыре признака: длина и ширина чашелистиков и лепестков в сантиметрах.

Этот набор данных стал типичным тестовым примером для многих методов статистической классификации в машинном обучении.

В лабораторной работе реализована кластеризация данных из набора по Виду (species), для кластеризации будут использованы атрибуты видов длинна и ширина чашелистика, длинна и ширина лепестка соответственно - sepal\_length, sepal\_width, petal\_length, petal\_width. Набор данных содержит 150 записей. Пример записей в файле с набором данных представлен на Рисунке 1.

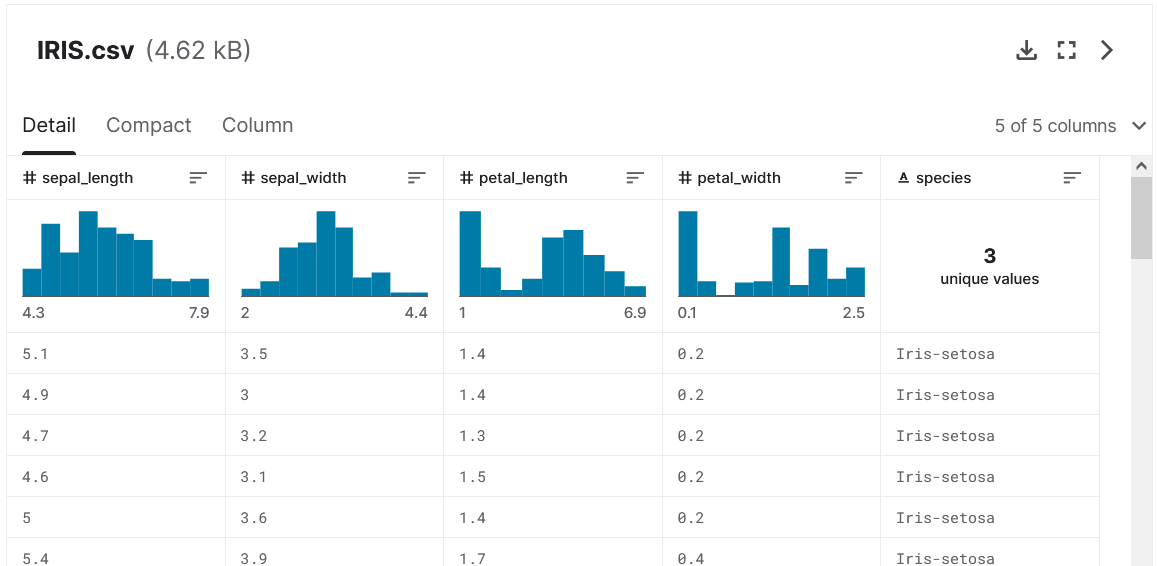


Рисунок 1. Пример данных из выбранного набора.

Далее будет приведен код, использованный для кластеризации данных, а так же сопутствующие вычисления и графики.

Импорт необходимых библиотек

import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

from sklearn.cluster import KMeans

import os

Проверка загружен ли csv файл

print(os.path.exists('path/iris.csv'))

Чтение датасета

PATH = "path/iris.csv"

dataset = pd.read\_csv(PATH)

Факторизация значений в столбцах, содержащих строковые значения

for column in dataset:

    if type(dataset[column][0]) is str:

        dataset[column] = pd.factorize(dataset[column])[0]

Отображение тепловой карты данных представлено на Рисунке 2.

sns.heatmap(round(abs(dataset.corr()), 1,), annot=True)

plt.show()

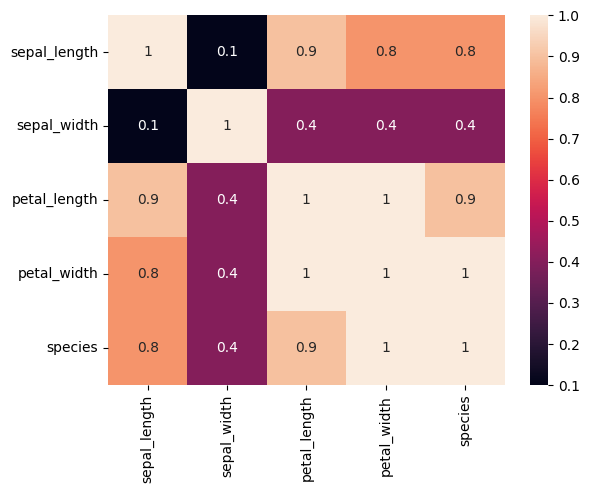


Рисунок 2. Тепловая карта связей столбцов данных в наборе.

Для выбора нужного количества кластеров будет построен график зависимости инерции от количества кластеров (Рисунок 3). Оптимальное количество кластеров, на основе графика – 3.

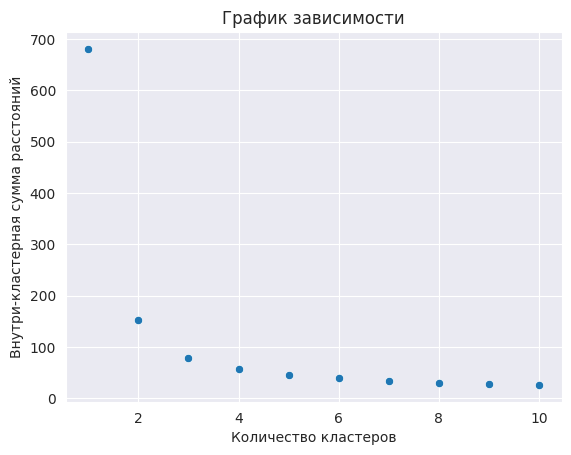


Рисунок 3. График зависимости инерции от количества кластеров.

На графике откладывается некоторая величина, характеризующая качество кластеризации, например, внутри-кластерная сумма расстояний 𝐽 для разного количества кластеров. В прикладных программах для реализации метода 𝑘-средних нужная величина – это искажение или инерция, например в методе KMeans библиотеки sklearn – это значение атрибута inertia\_). Оптимальное количество кластеров соответствует значению 𝑘, после которого величина 𝐽 перестает резко падать. Лучший вариант 𝐽 = 0 достигается при количестве кластеров, совпадающих с числом наблюдений.

Но в действительности обычно нужны стабильные кластеры с большим количеством наблюдений, для которых можно выявить закономерности.

Код для построения графика

inertia = []

for num\_of\_clusters in range(1, 11):

    k\_means = KMeans(n\_clusters=num\_of\_clusters, init="k-means++")

    # init: определяет метод для инициализации центроидов кластера.

    # "kmeans++" - широко используемым методом, который пытается выбрать начальные центроиды,

    # удаленные друг от друга, чтобы повысить шансы найти хорошее решение для кластеризации.

    k\_means.fit(dataset.drop("species", axis=1))

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

sns.set\_style("darkgrid")

sns.scatterplot(x=[x for x in range(1, 11)], y=inertia)

# scatterplot - функция cоздания диаграммы рассеивания

# args - [набор данных, который нужно визуализировать; столбцы, которые будут выступать как оси x и y]

plt.title("График зависимости")

plt.xlabel("Количество кластеров")

plt.ylabel("Внутри-кластерная сумма расстояний")

Код для построения модели, делящей данные на 3 части. Так же код предоставляет средние значения кластеров и количество вхождений в них.

CLUSTERS\_NUM = 3

model = KMeans(n\_clusters=CLUSTERS\_NUM)

model.fit(dataset.drop("species",axis=1,))

# Метод "fit" является ключевым шагом в обучении модели кластеризации.

# K средних значений на наборе данных. Он определяет назначения кластеров для

# каждой точки данных и обновляет центроиды кластера на основе полученных данных.

clusters = pd.DataFrame(

    columns=dataset.columns.drop("species"),

    data=model.cluster\_centers\_

)

clusters["number\_of\_occurrences"] = \

    np.unique(model.labels\_, return\_counts=True)[1]

print(clusters)

Получена следующая таблица (таблица будет меняться каждый раз, даже на одинаковых данных, из-за случайного выбора начальных значений центроидов).

sepal\_length sepal\_width petal\_length petal\_width number\_of\_occurrences

0 5.901613 2.748387 4.393548 1.433871 62

1 5.006000 3.418000 1.464000 0.244000 50

2 6.850000 3.073684 5.742105 2.071053 38

После нескольких запусков вычислений было выяснено, что модель очень хорошо работает на выбранном наборе данных. Каждый раз в итоговой таблице для кластеров определяется одинаковое количество вхождений. Благодаря тому, что изначальные данные были размечены, мы можем посчитать сколько изначально было каждого Вида, а затем сделать выводы, что модель точно соотносит данные определенного типа к нужному кластеру. Сильная взаимосвязь между определяемым и остальными параметрами, отраженная на тепловой карте, оказывает значительное положительное влияние на качество работы модели.

1. **Вывод**

Во время выполнения лабораторной работы были приобретены навыки построения модели для кластеризации методом 𝑘-средних при обучении без учителя. В ходе выполнения был выбран с сайта kaggle.com набор данных в формате .csv, пригодный для построения модели кластеризации. После он был загружен и подготовлен к дальнейшей обработке. Под подготовкой подразумевается факторизация данных и удаления столбца данных, размечающих данные (меток классов). Была построена модель кластеризации методом 𝑘-средних при помощи языка Python и библиотеки KMeans. Так же для корректного построения модели было определено оптимальное количество кластеров (3), при помощи графика зависимости инерции от количества кластеров в модели. После обработки данных были получены центроиды кластеров и количество вхождений в каждый из кластеров. Было определено какие знания можно получить из набора данных, мы соотнесли исходные размеченные данные и полученные значения, получив Видовую интерпретацию каждого и кластеров.