Проект по "Размити множества и приложения"

Автор: Любослав Карев

Въведение в решавания проблем и цел на проекта

Целта на проекта бе да се създаде програма, която използва размити клъстеризиращи алгоритми върху даден набор от данни, да се направи визуализация на тези данни, както и да се дефинират размити лингвистични правила, на база на резултатите от клъстеризацията.

Теоретична постановка и използван алгоритъм: Описание на приложения алгоритъм: дефиниции и извеждания, необходими за реализацията на поставената цел.

Алгоритъмът използван за клъстеризация е C-Means. Алгоритъмът се реализира в няколко стъпи:

1. Избор на брой клъстери
2. На случаен принцип се поставя степен на принадлежност на всеки един вектор от данните, към всеки един от клъстерите
3. Изчислява се центъра на всеки един от клъстерите
4. За всеки вектор от данните, се преизчислява неговата степен на принадлежност към всеки от клъстерите
5. Ако разликата между старите и новоизчислените коефициенти е по-голяма от предварително зададена константа, алгоритъма се връща на стъпка 3

   Център на клъстер се изчислява по следния начин:

$$c_k = \frac{\sum_{x \in X} w_k(x)^m x}{\sum_{x \in X} w_k(x)^m}$$

където k е клъстера, за който се изчислява центъра, X са данните ни, а m е параметър, показващ "размитостта" на множеството.

Степен на принадлежност на елемент i към клъстер j изчисляваме по следния начин:

$$w_{ij} = \frac{1}{\sum_{k=1}^{c} \left(\frac{||x_i - c_j||}{||x_i - c_k||}\right)^{\frac{2}{m-1}}}$$

Лингвистичните правила са представени с функцията на принадлежност на Бел, като в общия случай:

$$bell(x; a, b, c) = \frac{1}{1 + \left|\frac{u-c}{a}\right|^{2b}}$$

В конкретната задача, за параметър c ще приемаме центъра на дадения клъстер c_i. Наклонът b ще бъде оставен като параметър равен на 2 (с възможност за промяна). Широчината a пък ще е разстоянието между конкретния клъстер c_i и най-близкия до него ($|c_i - c_j|$)

Описание на данните, предпроцесна обработка

Използваните данни са Iris данните - представящи три представителя на семейството цветя Iris - Iris-virginica, Iris-versicolor и Iris-setosa. Всеки от индивидите е представен със следните пет характеристики:

- височина на чашелистче (`sepal_height`)
- широчина на чашелистче (`sepal_width`)
- височина на венчелистче (`petal_height`)
- широчина на венчелистче (`petal_width`)
- клас към който принадлежи екземпляра (`class`)

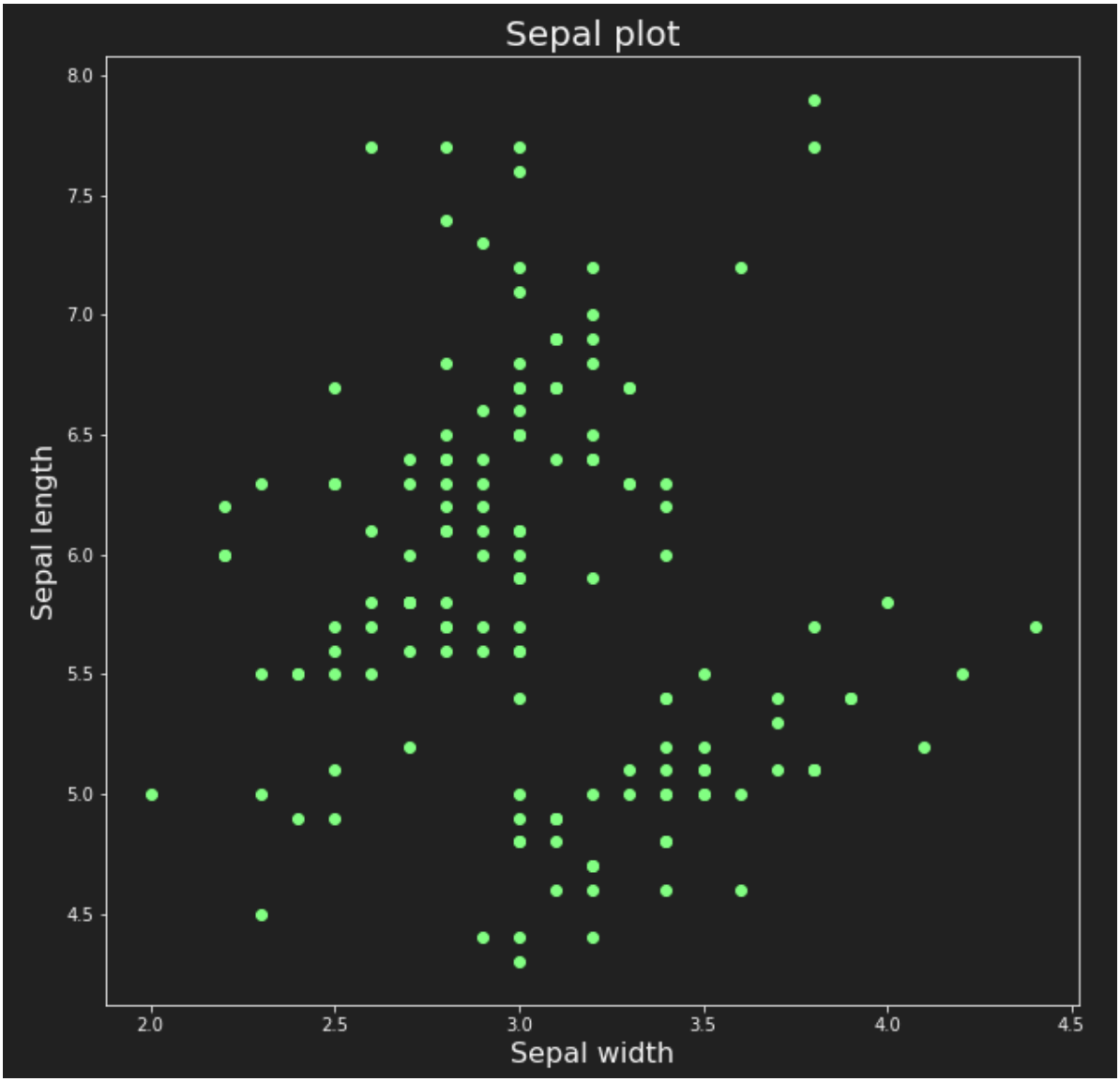
	sepal_length	sepal_width	petal_length	petal_width	class
0	5.1	3.5	1.4	0.2	Iris-setosa
1	4.9	3.0	1.4	0.2	Iris-setosa
2	4.7	3.2	1.3	0.2	Iris-setosa
3	4.6	3.1	1.5	0.2	Iris-setosa
4	5.0	3.6	1.4	0.2	Iris-setosa
5	5.4	3.9	1.7	0.4	Iris-setosa
6	4.6	3.4	1.4	0.3	Iris-setosa
7	5.0	3.4	1.5	0.2	Iris-setosa
8	4.4	2.9	1.4	0.2	Iris-setosa

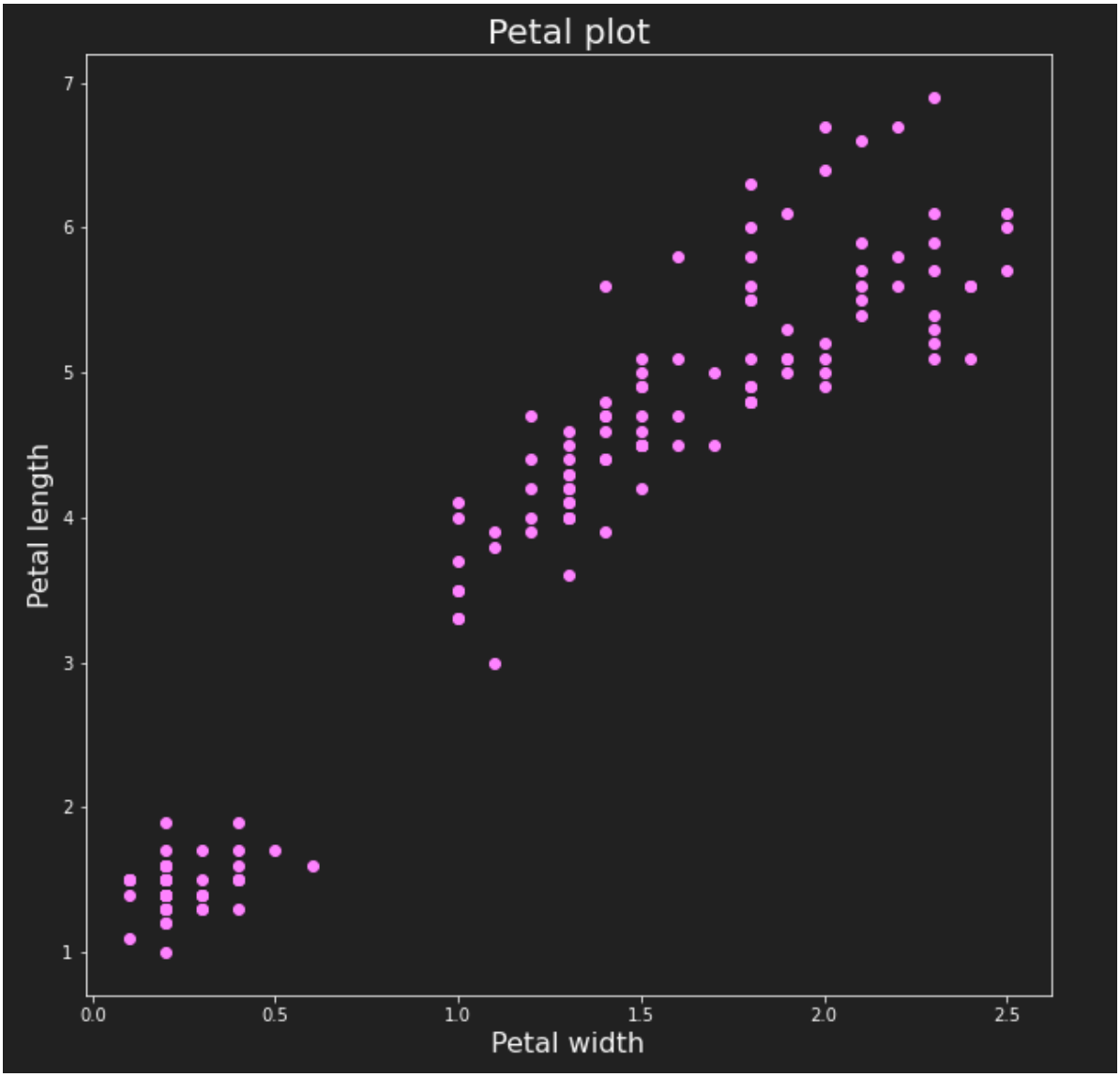
В базата от данни разполагаме с 150 екземпляра, разпределени равномерно между трите класа. Като част от предварителната обработка, ще премахнем информацията за класа от данните.

Оттук нататък вместо "чашеличстче и венчелистче" ще използваме английските термини "sepal" и "petal"

Експериментални/симулационни резултати: Представят се получени резултати и се визуализират - графично и/или таблично. Анализ на резултатите.

Нека първо визуализираме наличните данни - на долните две диаграми са показани графично `sepal width` / `sepal_length`, както и `petal_width` / `petal_length`.


Sepal plot
Sepal length
Sepal width
8.0
7.5
7.0
6.5
6.0
5.5
5.0
4.5
2.0
2.5
3.0
3.5
4.0
4.5



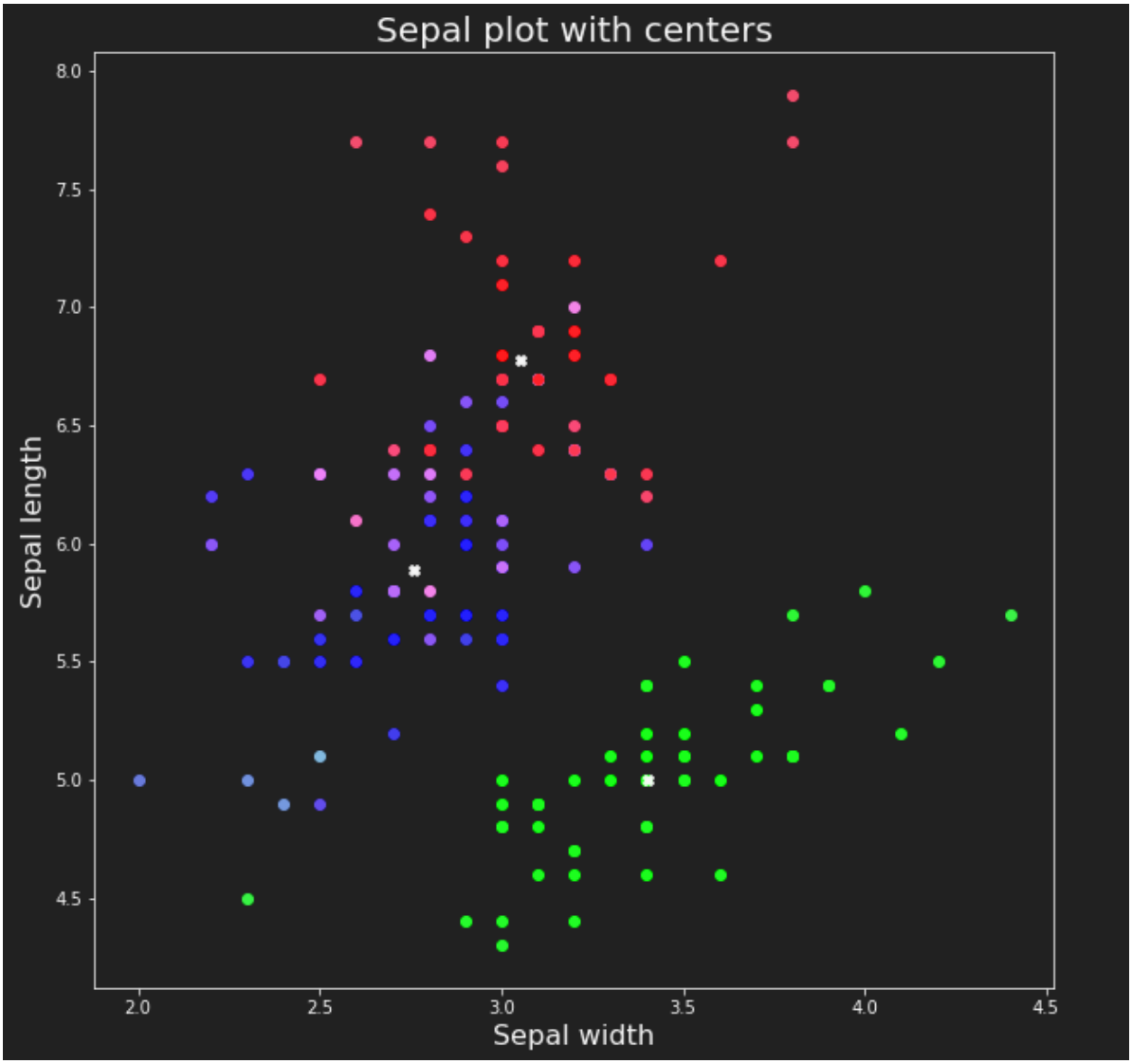
С помощта на библиотеката `scikit fuzzy`, стартираме алгоритъма Fuzzy C-Means, със следните параметри:

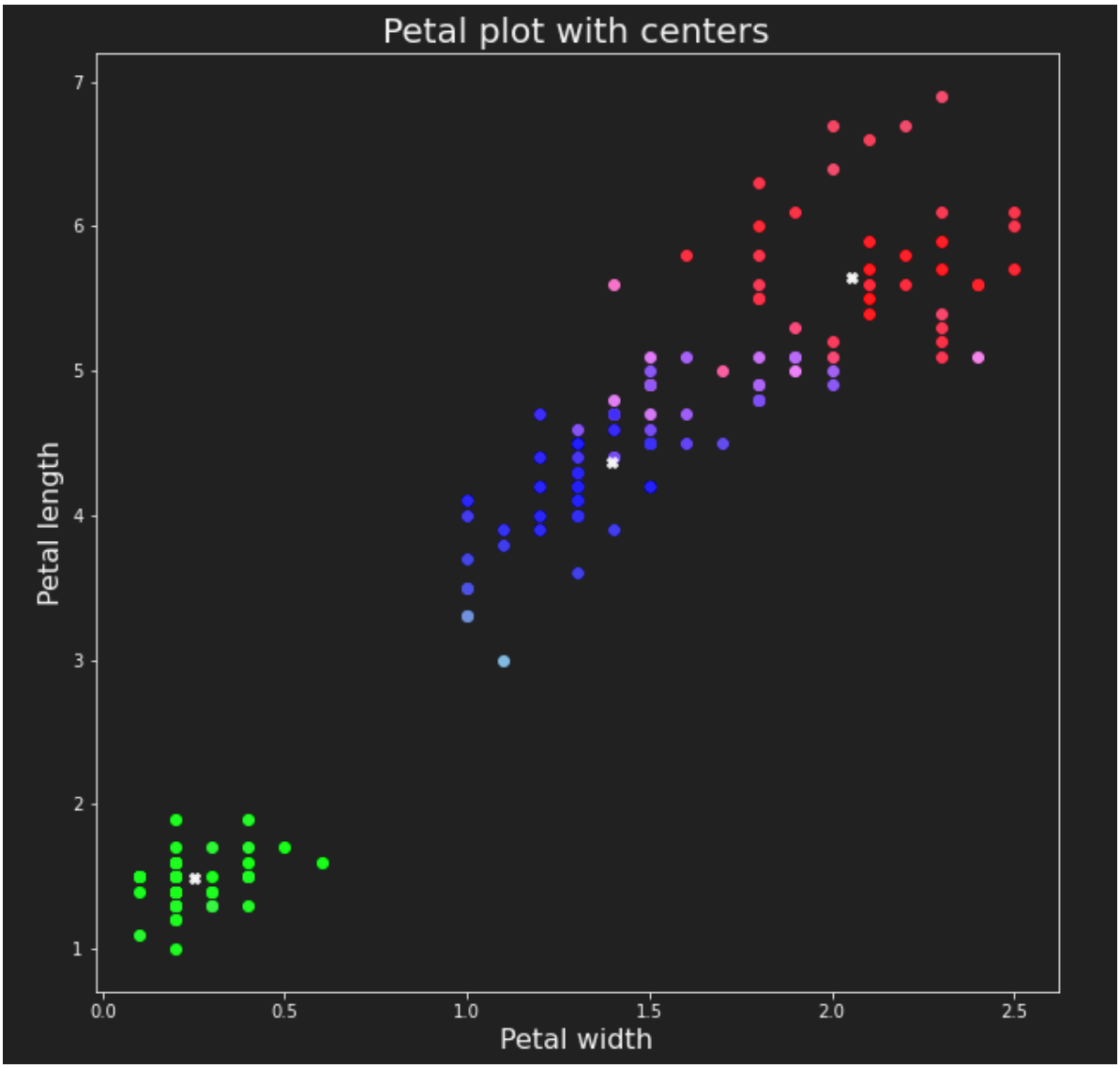
- Брой на клъстери: 3
- Хиперпараметър m: 2
- Разлика, при която алгоритъма спира: 0.0001
- Максимален брой итерации: 1000

Алгоритъмът ни връща откритите центъри на клъстерите, както и матрица, съдържаща степента на принадлежност на всеки от записите в данните към трите клъстера:

	sepal_length	sepal_width	petal_length	petal_width
0	6.775107	3.052427	5.646900	2.053603
1	5.889190	2.761231	4.364241	1.397439
2	5.003561	3.403036	1.485001	0.251541

Графичното преставяне на изглежда по следния начин:


Sepal plot with centers
Sepal length
Sepal width
8.0
7.5
7.0
6.5
6.0
5.5
5.0
4.5
2.0
2.5
3.0
3.5
4.0
4.5



Оцветяването на всяка от точките зависи от нивото на принадлежност към всеки от клъстерите - ако дадена точка принадлежи изцяло към един клъстер (например този в долния край), цветът ѝ ще бъде изцяло зелен. Другите два клъстера са оцветени в син и червен цвят - тези точки, които пренадлежат и към единия, и към другия клъстер са в розово-лилав цвят. Центрите на трите клъстера са отбелязани с бели символи X.

На база на тези клъстери можем да дефинираме няколко лингвистични променливи:

- `sepal_length_short`
- `sepal_length_normal`
- `sepal_length_long`
- `sepal_width_short`
- `sepal_width_normal`
- `sepal_width_long`
- `petal_length_short`
- `petal_length_normal`
- `petal_length_long`
- `petal_width_short`
- `petal_width_normal`
- `petal_width_long`

Тези лингвистични променливи са дефинирани с функция на принадлежност на Бел: центъра на кривата е съответната координата от клъстерния център, широчината на кривата е разликата между центъра на клъстера и центъра на най-близкия клъстер, а наклонът е хиперпараметър, със стойност 1.

Основни изводи (идеи за бъдеща работа; възможни приложения на използваните подходи и реализиран алгоритъм).

Една от основните идеи за бъдеща разработка, е добавяне на използването на обобщения модул поненс, за извършване на разсъждения на база на горе-споменатите лингвистични правила. Пример за такъв извод би бил:

`if petal_width is short and petal_length is long and sepal_width is normal and sepal_height is long, then flower is Iris-setosa`

Друго разширение, би било прилагането на вид класификация, чрез използването на вече получените клъстери. За целта, трябва да дадем клас към всеки от откритите центрове на клъстери. Ако се появи нов индивид, за който нямаме информация за класа му, можем да изчислим принадлежността му към всеки от другите клъстери. Ако принадлежността на индивида към даден клъстер надвишава дадена константа, то можем да класифицираме индивида към даден клас.

Списък на използваната литература

- https://archive.ics.uci.edu/ml/datasets/Iris
- https://www.kaggle.com/prateekk94/fuzzy-c-means-clustering-on-iris-dataset
- http://ce.sharif.edu/courses/92-93/1/ce957-1/resources/root/Lectures/Lecture4&5.pdf

Приложение: Код на програмната реализация

https://github.com/lyubolp/fuzzy-clusterization/blob/main/fuzzy_classification.ipynb