# Dokumentácia ku Klastrovaniu

## Jakub Fridrich

**Zadanie**: Z2b, klastrovanie

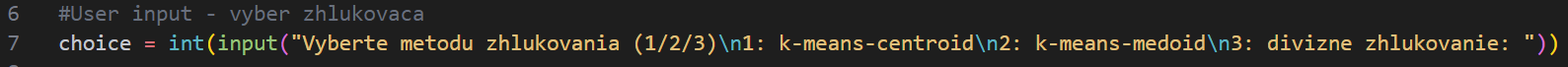
**Použité IDE**: VS Code

**Použité knižnice**: tkinter, numpy, random, time

**Súbor s kódom**: Z2b.py

## **Vstupné údaje**

Po spustení programu je používateľom zadaná verzia zhlukovača.



## **Reprezentácia údajov**

**Body (Points)**:

**Pole array\_points** obsahuje všetky body, ktoré sú generované v rámci určitého rozsahu súradníc. Každý bod je reprezentovaný dvojicou súradníc (x, y), čo vytvára 2D priestor pre clustering.

**Typ dát**: array\_points je zoznam súradníc, kde každý bod je dvojčlenný zoznam, napríklad [(x1, y1), (x2, y2), ...].

**Rozsah súradníc**: Súradnice bodov sa pohybujú v rozsahu od -5000 do 5000.

**Centrá klastrov (Centers)**:

**Centroidy a Medoidy**: V závislosti od výberu metódy sa klastre reprezentujú prostredníctvom centroidov alebo medoidov. **Centroid** je priemer všetkých bodov v klastri a nie je nevyhnutne jedným z bodov. **Medoid** je skutočný bod v klastri, ktorý minimalizuje priemernú vzdialenosť k ostatným bodom v klastri.

**Inicializácia centier**: Centrá sa inicializujú pomocou **k-means++ metódy**, aby boli centrá rozmiestnené **efektívnejšie**. Táto inicializácia je uložená buď v zozname **centroids\_id** pre centroidy alebo **medoids\_id** pre medoidy.

**Typ dát**: Zoznam dvojíc súradníc podobne ako pri bodoch, ale obsahuje len toľko centier, koľko je požadovaných klastrov (napríklad [(c1\_x, c1\_y), (c2\_x, c2\_y), ...]).

**Klastre (Clusters)**:

Klastre sú reprezentované ako zoznam zoznamov. Každý klaster je zoznam bodov, ktoré k nemu patria.

**Pri k-means**: Pre každý klaster sa body priraďujú na základe najbližšieho centra (centroidu alebo medoidu). Klastre sa aktualizujú každou iteráciou, kým sa centrá stabilizujú.

**Pri divíznom zhlukovaní**: Klastre sú inicializované s jedným veľkým klastrom obsahujúcim všetky body, ktorý sa následne delí na dva menšie klastre. Tento proces pokračuje, kým sú všetky klastre dostatočne kompaktné (pri vyhovujúcej priemernej vzdialenosti 500).

**Typ dát**: Klastre sú reprezentované ako zoznam zoznamov, napríklad [[klaster\_1\_body], [klaster\_2\_body], ...], kde každý vnútorný zoznam obsahuje body patriace do konkrétneho klastra.

**Matica vzdialeností (Distances)**:

Matica vzdialeností vypočítaná vo funkcii dist\_calc ukladá vzdialenosti každého bodu k všetkým centrám, čím umožňuje efektívne priradenie bodov k najbližšiemu centru.

**Typ dát**: Matica je numpy 2D pole s rozmermi (num\_points, num\_clus), kde každý riadok predstavuje bod a každý stĺpec vzdialenosť tohto bodu k určitému centru.

**Vizualizácia pomocou Tkinter**:

* Program vykresľuje body a centrá klastrov na plátno pomocou knižnice Tkinter. Každý bod sa zobrazuje ako malý kruh a centrum klastra je zobrazený ako kruh, pričom klastre sú odlíšené farebne.

## **Klastrovanie K-means (centroid) algoritmom**

K-means je algoritmus zhlukovania, ktorý pracuje na princípe priraďovania bodov k *k* rôznym zhlukom (klastrom) podľa ich podobnosti. Pre verziu, kde stredom klastrov je **centroid**, algoritmus funguje v programe nasledovne:

**Inicializácia centier:** Náhodne vyberieme *k* bodov v priestore ako počiatočné centrá klastrov (centroidy). Na efektívnejšie rozmiestnenie centier bude použité **K-means++**.

**Priraďovanie bodov:** Každý bod v priestore priradíme k najbližšiemu centru (podľa Euklidovskej vzdialenosti), čím vytvoríme *k* klastrov.

**Aktualizácia centroidov:** Pre každý klaster vypočítame nový centroid ako aritmetický priemer všetkých bodov v danom klastri. Tento nový centroid sa stane novým stredom klastra.

**Opakovanie priraďovania bodov a aktualizácie centroidov:** Kroky priraďovania a aktualizácie opakujeme, kým sa centroidy prestanú výrazne meniť alebo kým algoritmus dosiahne preddefinovanú podmienku.

## **Generovanie bodov**

Po úspešnom user-inpute sa spustí funkcia **init\_20()**, ktorá vygeneruje 20 unikátnych bodov do 2D priestoru. Tento priestor má rozmery X a Y, v intervaloch od -5000 do +5000.

Následne po vygenerovaní týchto 20 bodov sa spustí funkcia **generate\_more(count)**, ktorá vygeneruje ďalších *count-*bodov, v tomto prípade 40 000. Tieto body sú generované nasledovným spôsobom:

1. Náhodne sa vyberie jeden zo **všetkých doteraz vytvorených** bodov v 2D priestore. (nie len z prvých 20)
2. Vygeneruje sa náhodné číslo **X\_offset** v intervale od -100 do +100
3. Vygeneruje sa náhodné číslo **Y\_offset** v intervale od -100 do +100
4. Pripočíta sa X\_offset a Y\_offset k súradniciam bodu a ak súradnice presahujú 2D priestor, generujú sa nové offsety až kým nebudú korektné.
5. Nový bod sa pridá do 2D priestoru, ktorý bude mať súradnice ako náhodne vybraný bod v kroku 1, pričom tieto súradnice budú posunuté o X\_offset a Y\_offset

Obrázok, na ktorom je text, snímka obrazovky, softvér, multimediálny softvér

Automaticky generovaný popis

## **Inicializácia centier**

Program využíva inicializáciu centier metódou **k-means++.** Táto metóda bola použitá pre efektívne rozmiestnenie centier, čo aj zvyšuje úspešnosť programu.

V prípade, že bolo zvolené k-means centroid zhlukovanie, program vygeneruje náhodné súradnice v rozhraní súradníc.

Ak však bolo zvolené k-means medoid zhlukovanie, program vyberie náhodný bod z už existujúcich bodov.

Každý ďalší centroid alebo medoid je vybraný na základe **pravdepodobnosti úmernej vzdialenosti** od najbližšieho už existujúceho centra.

Pre každý bod sa spočíta jeho najmenšia vzdialenosť ku každému už zvolenému centru a na základe týchto vzdialeností sa vytvorí rozdelenie pravdepodobností.

Náhodne sa vyberie nové centrum, kde pravdepodobnosť výberu každého bodu závisí od jeho vzdialenosti (body, ktoré sú ďalej od existujúcich centier, majú väčšiu šancu stať sa novými centrami)

Celý proces zabezpečuje funkcia **kmeans\_pp (arr, k, choice)**

Obrázok, na ktorom je text, snímka obrazovky, softvér

Automaticky generovaný popis

## **Priraďovanie bodov**

Ako prvé sa vypočítajú vzdialenosti bodov a centier pomocou Euklidovej vzdialenosti a rozšírenie o novu os (pre umožnenie broadcastu s **centers\_array**), a to vo funkcii **dist\_calc(array\_centers)**.

Obrázok, na ktorom je text, snímka obrazovky, softvér, písmo

Automaticky generovaný popis

Následne sa pre každý bod nájde index najbližšieho centra vďaka funkcii **clustering(dists)** a priradia sa k príslušnému klasteru pomocou **for-loopu** vo funkcii **kcent\_clustering()** alebo **kmed\_clustering().**

Obrázok, na ktorom je text, snímka obrazovky, písmo

Automaticky generovaný popis

Obrázok, na ktorom je text, snímka obrazovky, písmo, rad

Automaticky generovaný popis

## **Aktualizácia centier**

Obrázok, na ktorom je text, písmo, snímka obrazovky

Automaticky generovaný popis

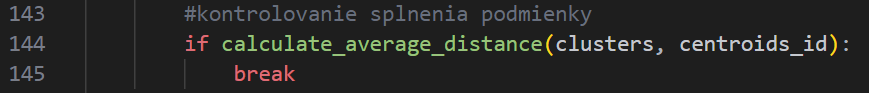
Obrázok, na ktorom je text, písmo, snímka obrazovky

Automaticky generovaný popis

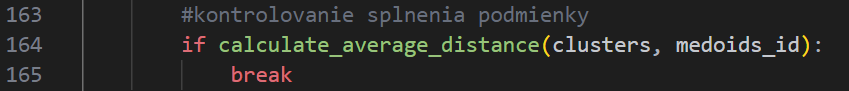
## **Kontrolovanie splnenia podmienky**

Podmienka úspešného ukončenia programu sa v programe kontroluje pomocou funkcie **calculate\_average\_distance**. Táto funkcia vypočíta **priemernú vzdialenosť bodov od centra** v každom klastri. Ak sú všetky priemerné vzdialenosti v rámci klastrov **menšie alebo rovné stanovenému prahu (500)**, algoritmus sa považuje za **úspešne ukončený**, pretože sa dosiahla dostatočná zhlukovacia presnosť. Ak nie je úspešne dosiahnutý prah (500) do limitu iterácii (20), program sa považuje za neúspešný.

*Volanie vo funkcii kcent\_clustering():*



*Volanie vo funkcii kmed\_clustering():*

**

Obrázok, na ktorom je text, multimediálny softvér, softvér, snímka obrazovky

Automaticky generovaný popis

## **Klastrovanie Divíznym zhlukovaním**

Divízne zhlukovanie na princípe **postupného delenia klastrov**, pričom začíname s jedným veľkým klastrom obsahujúcim všetky body a postupne ho rozdeľujeme na menšie klastre, kým každý klaster nedosiahne požadovanú presnosť, to je v každom klastri priemernú vzdialenosť bodov od stredu viac ako 500.

**Inicializácia:** Vytvorí sa počiatočný klaster obsahujúci všetky body.

**Výpočet priemernej vzdialenosti**: Pre každý klaster sa vypočíta centroid (stred klastra) a priemerná vzdialenosť všetkých bodov v tomto klastri od jeho centroidu.

**Rozdelenie klastrov**: Ak je priemerná vzdialenosť bodov od centroidu väčšia než stanovený prah, klaster sa rozdelí na dva menšie klastre.

**Opakovanie**: Tento proces sa opakuje pre každý nový klaster, kým všetky klastre nesplnia podmienku maximálnej priemernej vzdialenosti.

**Ukončenie**: Ak sú všetky klastre dostatočne kompaktné, algoritmus sa zastaví.

Obrázok, na ktorom je text, snímka obrazovky, softvér

Automaticky generovaný popis

Obrázok, na ktorom je text, snímka obrazovky, softvér

Automaticky generovaný popis

## **Vizualizácia**

Vizualizácia je realizovaná funckiou **draw\_clusters(),** ktorá je volaná v každom type zhlukovača. Táto funkcia vykresli všetky body a finálne klastre, pričom každý bod je farebne odlíšený farbami zo zoznamu **cluster\_color.** Každý bod je transformovaný pomocou škálovacieho faktora (**scaling\_down=20**) a posunutý do stredu plátna tak, aby boli všetky body viditeľné.

Obrázok, na ktorom je text, snímka obrazovky

Automaticky generovaný popis

## **Testovanie zhlukovačov**