Martin Hric
ID = 111696

SLOVENSKÁ TECHNICKÁ UNIVERZITA V BRATISLAVE

FAKULTA INFORMATIKY A INFORMAČNÝCH TECHNOLÓGIÍ

UMELÁ INTELIGENCIA

4b Klastrovanie

Vypracoval: Martin Hric

Cvičiaci: Ing. Juraj Vincúr

Ak. rok: 2021/2022

Martin Hric
ID = 111696

OBSAH

Martin Hric
ID = 111696

ZADANIE

4b

Máme 2D priestor, ktorý má rozmery X a Y, v intervaloch od -5000 do +5000. Tento 2D priestor vyplňte 20 bodmi, pričom každý bod má náhodne zvolenú polohu pomocou súradníc X a Y. Každý bod má unikátne súradnice (t.j. nemalo by byť viacej bodov na presne tom istom mieste).

Po vygenerovaní 20 náhodných bodov vygenerujte ďalších 20000 bodov, avšak tieto body nebudú generované úplne náhodne, ale nasledovným spôsobom:

1. Náhodne vyberte jeden zo **všetkých** doteraz vytvorených bodov v 2D priestore.
   Ak je bod príliš blízko okraju, tak zredukujete príslušný interval v nasledujúcich dvoch krokoch.
2. Vygenerujte náhodné číslo X_offset v intervale od -100 do +100
3. Vygenerujte náhodné číslo Y_offset v intervale od -100 do +100
4. Pridajte nový bod 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

Vašou úlohou je naprogramovať zhlukovač pre 2D priestor, ktorý zanalyzuje 2D priestor so všetkými jeho bodmi a rozdelí tento priestor na k zhlukov (klastrov). Implementujte rôzne verzie zhlukovača, konkrétne týmito algoritmami:

- k-means, kde stred je centroid
- k-means, kde stred je medoid
- aglomeratívne zhlukovanie, kde stred je centroid
- divízne zhlukovanie, kde stred je centroid

Vyhodnocujte úspešnosť/chybovosť vášho zhlukovača. Za úspešný zhlukovač považujeme taký, v ktorom žiaden klaster nemá priemernú vzdialenosť bodov od stredu viac ako 500.

Vizualizácia: pre každý z týchto experimentov vykreslite výslednú 2D plochu tak, že označkujete (napr. vyfarbíte, očíslujete, zakrúžkujete) výsledné klastre.

Dokumentácia musí obsahovať opis konkrétne použitých algoritmov a reprezentácie údajov. V závere zhodnoťte dosiahnuté výsledky ich porovnaním.

Poznámka: Je vhodné použiť rôzne optimalizácie pre dostatočne efektívnu prácu Vášho zhlukovača. Napríklad pre aglomeratívne zhlukovanie je možné použiť 2-rozmernú maticu vzdialenosti dvojíc bodov. Naplnenie takejto matice má kvadratickú zložitosť. Potom sa hľadá najbližšia dvojica (najmenšie číslo v matici), to má opäť kvadratickú zložitosť, ale nenásobia sa tie časy, ale sčítavajú. Po výbere najbližšej dvojice túto dvojicu treba zlúčiť a tým sa zníži veľkosť matice o 1 (lebo sa zníži počet zhlukov o 1) Pri tom sa aktualizujú len vzdialenosti pre tento nový zhluk (len jeden stĺpec/riadok, zvyšok matice ostáva nezmenený).

Martin Hric
ID = 111696

ÚVOD

V tomto zadaní som mal za úlohu naprogramovať zhlukovač, kde zo začiatočných 20 náhodne vygenerovaných bodov v intervale <-5000,5000> vybrať náhodne 1 a k nemu s offsetom <-100,100> pre x,y vygenerovat ďalších N bodov, min. 20 000 bodov. Po vygenerovaní všetkých bodov tieto body vytvoria zhluky, ktoré som mal ohodnotiť pomocou algoritmov K-Means s centroidom, K-Means s monoidom, aglomeratívne a divizívne so stredmi ako centroid. Daný zhlukovač mám ohodnotiť ako úspešný , ak je jeho priemerná vzdialenosť bodov v rámci každého klastra(zhluku) od jeho stredu menšia ako 500. Potom mám aj vizualizovať dané zhluky , ktoré som prv rozmýšľal , že rovnako ako v 3. Zadaní to spravím vlastnoručne cez tkinter, ale potom som sa rozhodol radšej pre knižnicu matplotlib.

Opis algoritmov

K-means

Pri k-means ako prvé vyberiem 20 jedinečných bodov, ktoré určím ako stredy zhlukov. Následne priradím každý bod najbližšiemu klastru, čim si vytvorím prvé ohodnotenie bodov. Po ohodnotení bodov vyrátam nový stred klastra. Ak má byť stred centroid, tak ho vyrátam ako priemernú vzdialenosť všetkých x-ových súradníc, čím dostanem x-ovú súradnicu centroidu a rovnako vyrátam priemer všetkých y-ových súradníc, a tak získam y-ovú súradnicu centroidu. Centroid je iba fiktívny stred, tento bod reálne vykreslený nie je. Naopak ak má byť stred medoid, tak toto je už reálny bod, ktorý môžeme povedať, že to je najbližší bod k všetkým bodom, teda bod s najmenším súčtom euklidovských vzdialeností k všetkým ostatným bodom. Keď už mám aktualizovaný stred klastra, tak znova priradím všetky body k danému klastru. Tento cyklus priraďovania bodov a vyrátavania nového stredu vykonávam až pokiaľ sa pri prechode jednou iteráciou nič nezmení, čo znamená, že už máme finálne ohodnotenie v danom experimente.

Divizívne zhlukovanie

V divíznom zhlukovaní využívam k-means algoritmus so stredom ako centroid. Na začiatku mám 1 klaster, ktorý rozdelím na 2 klastre. Následne ak ešte nemám požadovaný počet klastrov, tak vyberiem klaster, ktorý má najväčší priemer vzdialeností v rámci klastra a ten znova rozdelím na 2 klastre. Tento cyklus sa opakuje až pokiaľ nedosiahnem zadaný počet klastrov.

Martin Hric
ID = 111696

Aglomeratívne zhlukovanie

Pri tomto zhlukovaní nastávajú komplikácie. Ja som ho totižto implementoval, ale pre 20 000 bodov čo je vasa minimálna požiadavka mi to nezbehne, resp. zbehne ale taký čas ja nie som ochotný čakať, mal som to pustené celú noc, celú noc mi na stolíku štartovalo lietadlo a ráno keď som vstal, som si našiel notebook vypnutý. Pravdepodobne nejaký bluescreen alebo akási chyba nastala a mi vypla notebook.

Vzhľadom na to, že som to nerobil úplne skoro nemal som už čas čakať na tento jeden algoritmus kým zbehne a nejako ho testovať či to naozaj funguje a robí čo ma, príde mi to ako výsmech aby som robil nejaký algoritmus, ktorý keď som pozeral na internet a od ostatných spolužiakov že to trvá +- 24 hodín, závisí od implementácie. Veď to sa nedá nijak testovať…

Nestíhal som to aj kvôli tomu, keďže každý predmet na tejto fakulte si myslí, že je len jeden jediný a najhlavnejší, ale takých predmetov mám 6 každý semester.

Je mi jasné, že mi strhnete body za to, že nemám tento algoritmus, ja len toľko k tomu, že mal som ho spravený, pre málo bodov mi zbehol, fungoval a robil čo mal ale podmienkou je aby zbehol pre 20 000 bodov a je jedno za aký čas, no a pre toľko bodov som nebol už ďalej ochotný čakať takže otestované to nie je.

Martin Hric
ID = 111696

Postrehy

Do tejto časti by som rád napísal moje postrehy, resp zmeny oproti zadaniu.

Odovzdávam kód, kde je to implementované presne podľa zadania na stránke predmetu. Ja ale osobne nesúhlasím s offsetom, pre generovanie bodov, pretože <-100,100> je veľmi málo v okolí jedného len z 20 bodov, kde ceý interval je až <-5000,5000>.

Takéto generovanie pre 20 000bodov predstavuje priemerne 1 000 bodov v okolí každého z počiatočných 20 bodov a ak máme natrepať 1 000 bodov pre <-100,100> tak to ďakujem pekne.

Toto generovanie vytvára štvorce a nie nejaký zhluk bodov, ktorý je v strede hustejší a po okrajoch redší. Pozeral som na stackoverflow a našiel som funkciu random.gauss(), ktorá tento problém vyriešila ale samozrejme nastavoval som aj väčší offset a to 500.

Ďalej zmenu, ktorú som robil bola tá, že som vo funkcii result(), ktorá mi vyhodnocovala úspešnosť klastrovanie hodnotu z 500 som menil na 1000, keďže 500 je veľmi málo a aj pre vyššie počty klastrov sa stávalo, že pri tak veľkom intervale a tak málo začiatočných bodoch nejaký zhluk bol +-1000 od stredu.

Čiže nabudúce roky, by som určite navýšil počet začiatočných bodov aspoň na 50.
(50 x 100(offset) = 5000(celý interval))

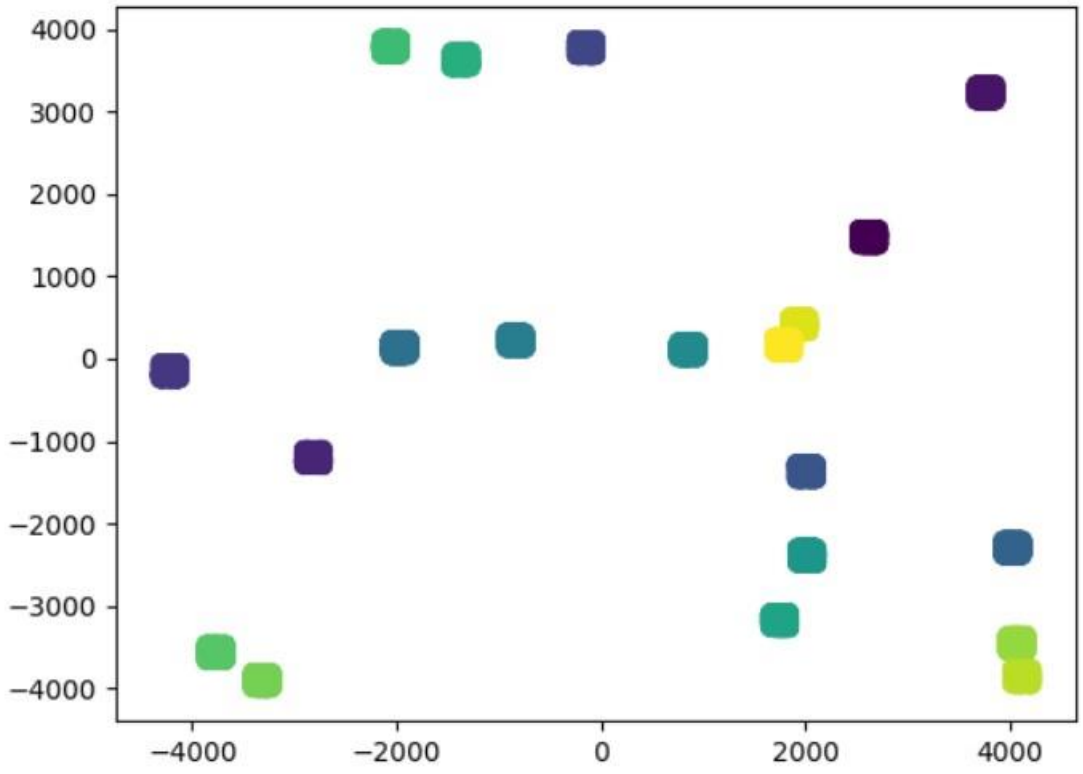
Martin Hric
ID = 111696

TESTOVANIE

Počas testovania som sa rozhodol porovnávať priemerné vzdialeností v rámci klastra a zároveň spriemerovať tieto priemery do jedného čísla, ktoré považujem ako finálne ohodnotenie zhlukovača v danom experimente. Na začiatok som len ukázal, ako to vyzerá pre hodnoty, ktoré sú dané zo zadania. Potom som to zmenil na parametre:

INTERVAL :5000, počet bodov :20 000, OFFSET :1000 (v programe mám implementované, že ak nový bod s offsetom je za hranicami interval, tak jeho hodnota sa otočí aby bol v intervale), 1000 kvôli tomu tomu aby nevznikali veľké biele miesta a boli aj prepojené niekedy dané klastre. Tie štvorčeky som nechal, na tom nezáleží veľmi.

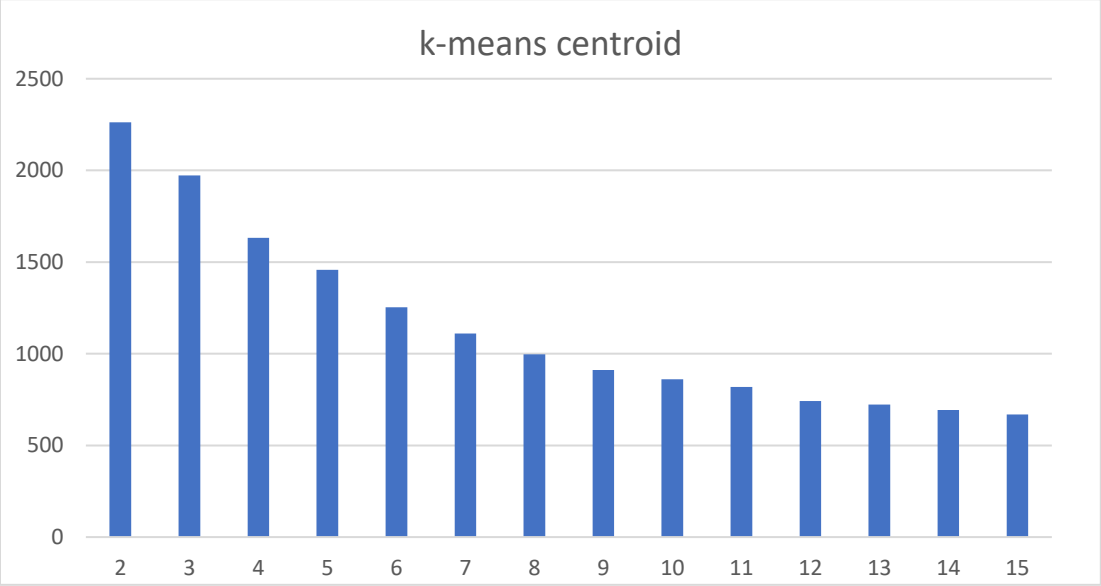
Testovanie som vykonával pre k=2 až k=15.



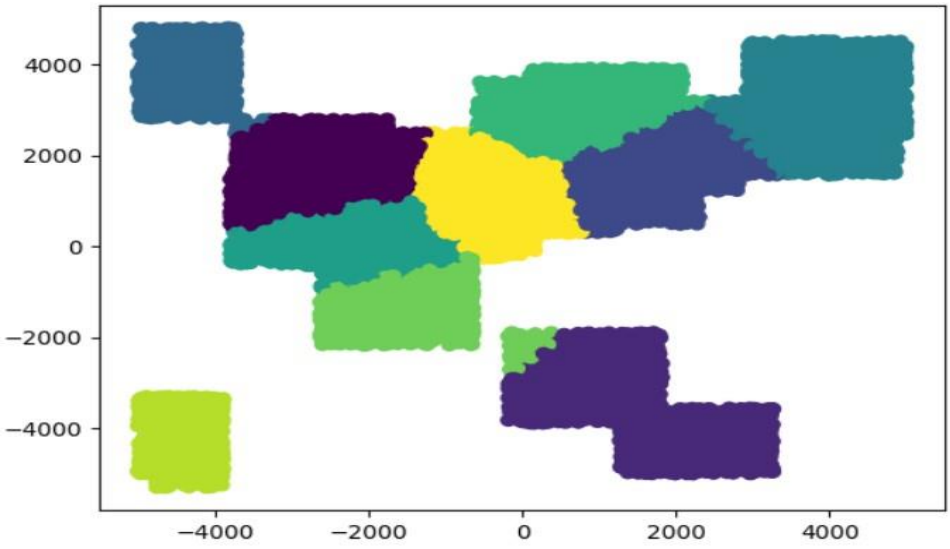
Takto to vyzerá, pre 20 000 bodov, a offset 100….

Martin Hric
ID = 111696

Ako prvé by som porovnal závislosť priemernej vzdialenosti od rastúceho k, pri jednotlivých zhlukovacích algoritmoch.



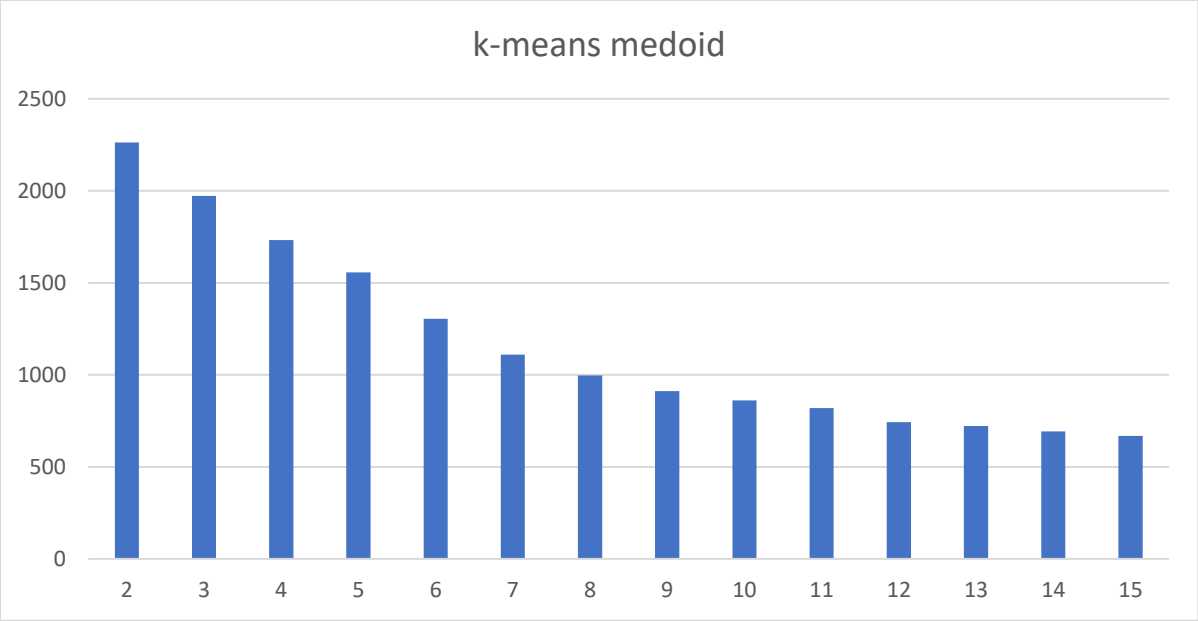
Vývoj algoritmu k-means, kde stred klastra je centroid, môžeme vidieť na grafe na obrázku 1. Výsledný graf je zostavený z piatich spriemerovaných vzoriek. Prvú vec, ktorú si všimneme hneď, a ktorú by sme aj očakávali je, že so zvyšujúcim sa k sa znižuje aj priemerná vzdialenosť v rámci klastrov. Od k=10 si môžme všimnúť, že hodnot začínajú pomaly stagnovať. To sa nazýva lakeť grafu.



Vyzobrazenie z testovania pre k=10.

Martin Hric
ID = 111696

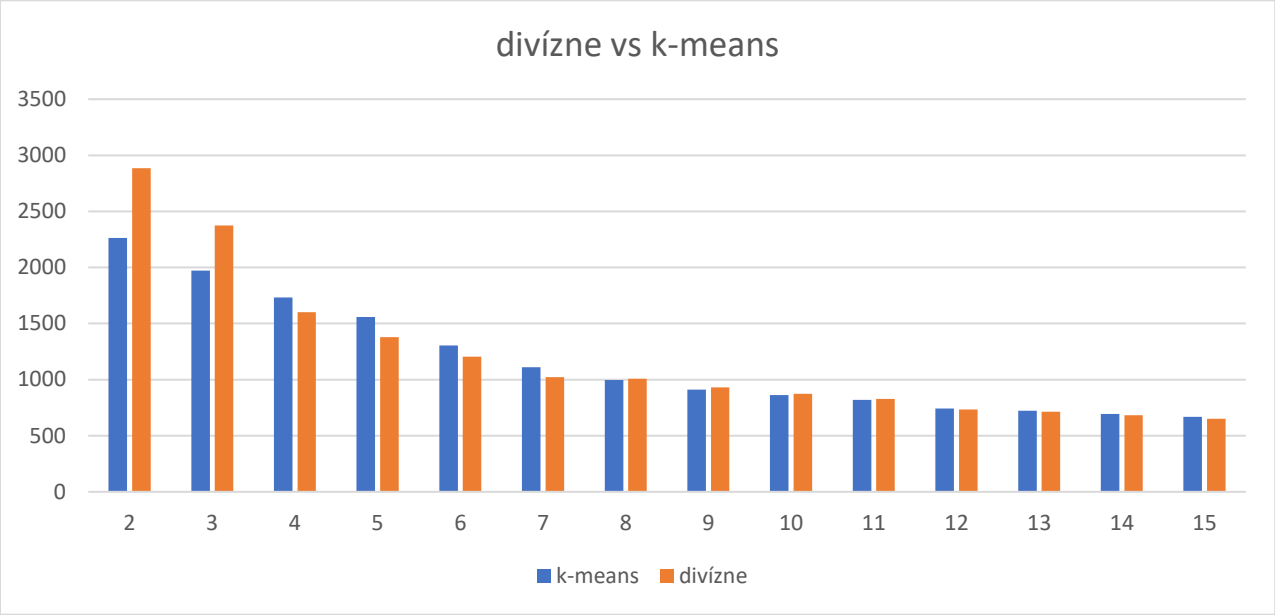
K-Means s medoidom, rovnako 5 spriemerovaných testov. Keďže je to rovnaký algoritmus , výsledy sú podobné ako centroid. Priemerne však dosahoval medoid o trochu lepšie výsledky, keďže ak klaster zahrňoval aj nejakých „outlierov“, tak títo mali oveľa väčší dopad na súradnice centroidu ako medoidu.



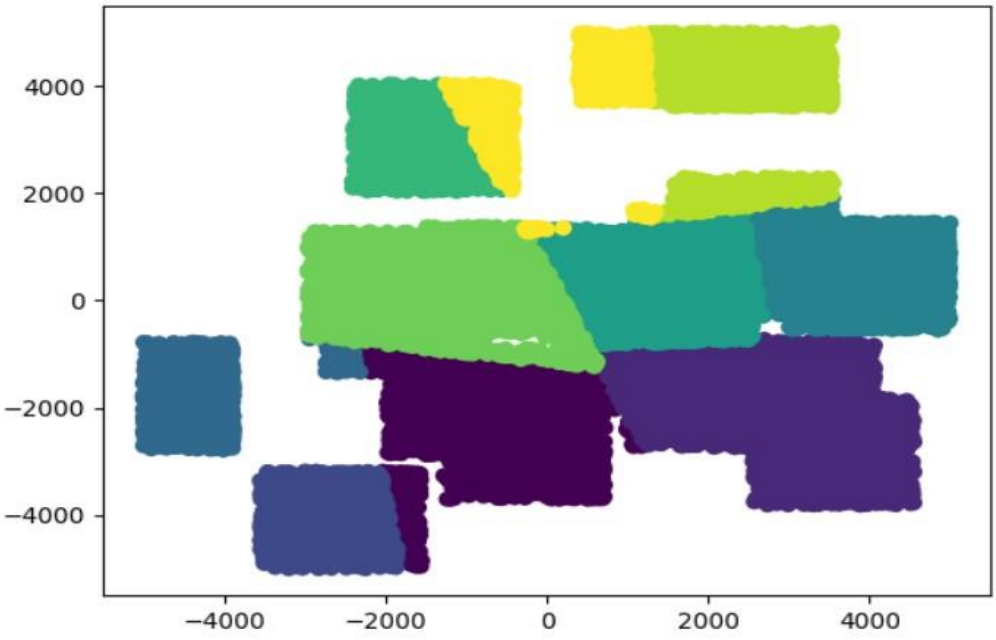
Pri väčšom počte klastrov však môžeme vidieť, že už aj tieto malé rozdiely sa stratili a priemery sú približne rovnaké. Dôvodom, prečo sú pri väčšom počte klastrov rozdiely už rovnaké je, že klastre sú už menšie, a teda aj outsiderov je tam menej, respektíve sú bližšie ako napríklad pri počte klastrov rovný 3. Avšak aj keď nám medoid dával lepšie výsledky, časovo bol na tom horšie nakoľko sme tam stred hľadali ako bod najbližšie ku všetkým bodom, čo má časovú zložitosť O(n2), kde n je počet prvkov v klastri. To mi pri 20020 prvkoch trvalo približne 5-6 minút. Naopak ak je stred centroid, tak časová zložitosť pre nájdenie nového stredu klastru je iba O(n), pretože iba raz prejdem celý klaster a zrátam súradnice a následne ich spriemerujem, čo mi trvalo približne 6 sekúnd.

Martin Hric
ID = 111696

Na rovnakej množine bodov ako pri predošlých dvoch testoch som vykonal aj experiment s divíznym zhlukovačom. Výsledky som ale zahrnul do spojeného grafu ,kde som navzájom porovnal k-means a divízne zhlukovanie.



Divízne zhlukovanie v porovnaní s k-means dosahovalo zo začiatku mierne horšie výsledky. Neviem prečo.



Vyobrazenie z divízneho pre k=10.

Martin Hric
ID = 111696

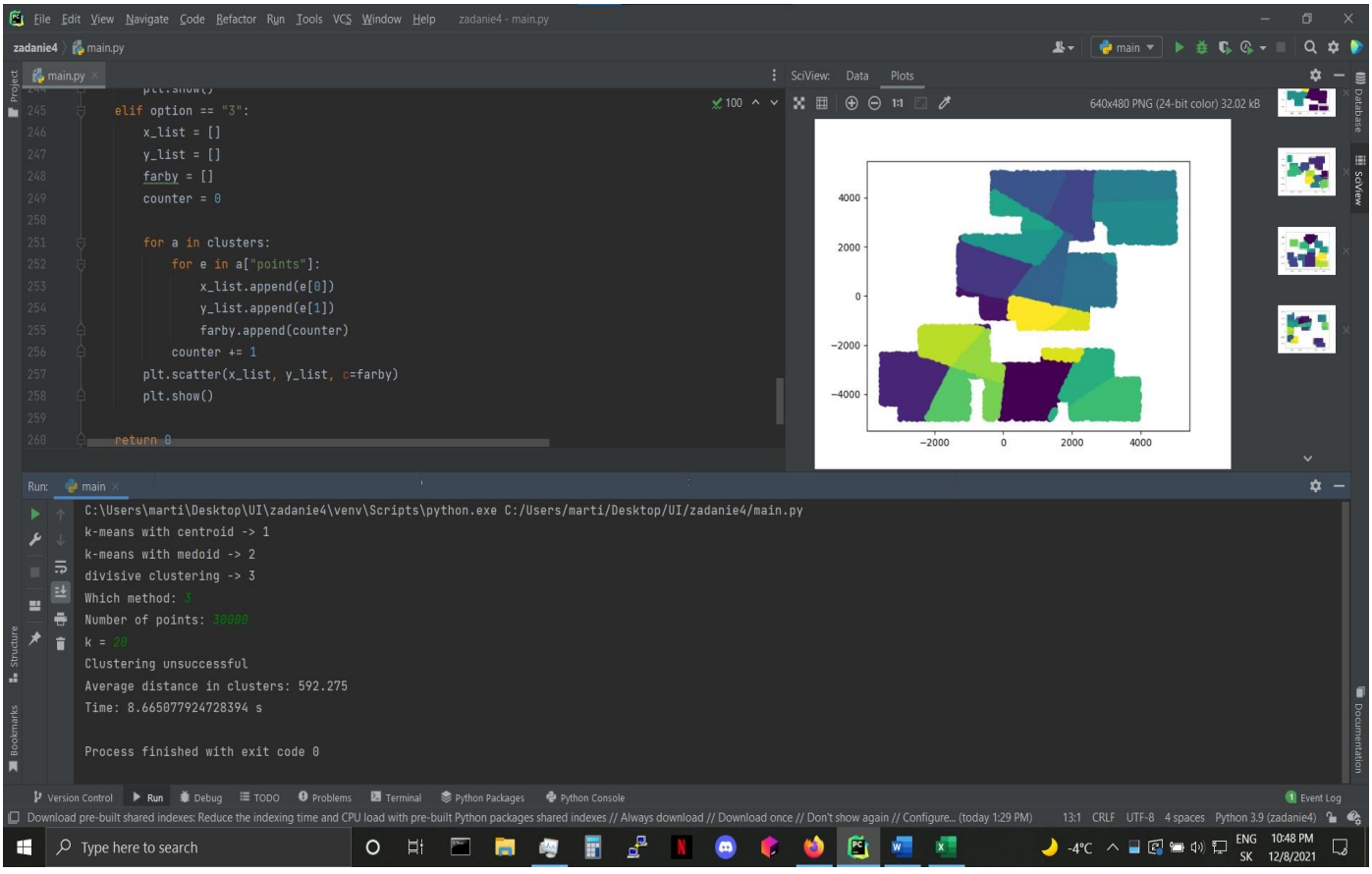
INTERFACE

Na začiatku si používateľ zvolí , ktorý algoritmus chce použiť (aglomeratívne som zmazal.)

Potom zvolí počet bodov a nakoniec koľko klastrov sa ma vytvoriť.

```
k-means with centroid -> 1
k-means with medoid -> 2
divisive clustering -> 3
Which method: 3
Number of points: 20000
k = 10
Clustering unsuccessful
Average distance in clusters: 968.845
Time: 3.766885995864868 s
```

Celé to vyzerá takto. Na pravej strane mi plt.scatter() zobrazuje zhluky.



Martin Hric
ID = 111696

ZÁVER

Nevýhodou k-means je, že vo veľkej miere záleží od počiatočného stavu, ako sú určené prvé klastre, a tak koľko-krát pustíme algoritmus, toľko rôznych výsledkov dostaneme. Výhodou ale je jeho časová zložitosť, ktorá záleží od počtu klastrov, bodov a času potrebného na výpočet vzdialenosti dvoch bodov.

Najrýchlejšie je divízne, najpomalšie je aglomeratívne no zato pri tých mojích pár malých testoch mi prišlo najlepšie z hľadiska zhlukovania. Pre 20 000 bodov mi to ale nikdy nezbehlo takže som to netestoval.

Na záver by som ešte rád podotkol, o koľko náročnejšie bolo zadanie 4b oproti 4a.

Pozrel som si čo treba robiť v 4a a je to neporovateľne jednoduchšie… Samozrejme vy pán cvičiaci, za to nijak nemôžte a aj nás ste upozorňovali aby sme takéto veci písali do evaluácie, určite tam všetko napíšem, i keď neviem aký to bude mať výsledok, keďže od starších spolužiakov som počul, že už niekoľko rokov sa študenti sťažujú na pána Kohútku, žiadne zmeny ale zatiaľ nenastali tak neviem či tá celá evaluácia ma vôbec nejaký zmysel…