

Zadanie 2B Dokumentácia

Použité algoritmy:

Začneme generáciou bodov, kde každý bod má svoje vlastné súradnice (nebudú dva body na tom istom mieste), tieto body ukladáme do listu `coordinates`. Potom si používateľ vyberie aký algoritmus bude použitý.

Algoritmus k-means, kde stred je centroid:

Konvertuje vstupné súradnice na numpy pole pre jednoduchšiu manipuláciu a začne výberom náhodného centroidu. Potom začne cyklus, ktorý sa ukončí ak dosiahneme 100% úspešnosť (keď žiaden klaster nemá priemernú vzdialenosť bodov od stredu viac ako 500.) alebo dosiahneme maximálny limit iterácií (100). V tomto cykle sa priradia body na základe najbližšieho centroidu do poľa `assigned_clusters[]`, aktualizuje centroidy na základe priemerných súradníc bodov v každom klasteri, skontroluje či sme dosiahly 100% úspešnosť, ak áno ukončí cyklus, ak nie pridá nový centroid (centroidy sa pridávajú do poľa `centroids []`) a opakuje cyklus.

Algoritmus k-means, kde stred je medoid:

Konvertuje vstupné súradnice na numpy pole pre jednoduchšiu manipuláciu a začne výberom náhodného medoidu. Potom začne cyklus, ktorý sa ukončí ak dosiahneme 100% úspešnosť (keď žiaden klaster nemá priemernú vzdialenosť bodov od stredu viac ako 500.) alebo dosiahneme maximálny limit iterácií (100). V tomto cykle vypočíta vzdialenosti bodov od medoidov pomocou štvorcovej euklidovskej vzdialenosti, vytvorí zoznam klastrov, kde každý klaster obsahuje body priradené k príslušnému medoidu `assigned_clusters[]`, ďalej aktualizuje medoidy následovne: Vypočíta celkovú vzdialenosť od každého bodu k ostatným v klasteri a nový medoid je bod s najmenšou celkovou vzdialenosťou k ostatným bodom v klasteri. Nakoniec skontroluje či sme dosiahly 100% úspešnosť, ak áno ukončí cyklus, ak nie pridá nový medoid (medoidy sa pridávajú do poľa `medoids []`) a opakuje cyklus.

Algoritmus divízneho zhľukovania, kde stred je centroid:

Začne výpočtom centroidu pre všetky body, uloží ho do poľa `centroids[]` spolu aj s osou rozdelenia (začínáme s x-ovou), ďalej vytvorí zhľuk pre všetky body a začne cyklus, ktorý sa ukončí ak dosiahneme 100% úspešnosť (keď žiaden klaster nemá priemernú vzdialenosť bodov od stredu viac ako 500.) alebo dosiahneme maximálny limit iterácií (100). Následne zistí ktorý zhľuk je najväčší a rozdelí ho na základe uloženej osi centroidu potom inicializuje nové centroidy a zhľuky. Následne pre rozdelené časti vytvorí centroid a pridá zhľuk, nahradí pôvodný zhľuk novými a použije KD-Tree na efektívne priradovanie bodov k najbližším centroidom. Aktualizuje zhľuky na základe priradení z KD-Tree, prepočíta pozície centroidov a skontroluje či sme dosiahly 100% úspešnosť, ak áno ukončí cyklus, ak nie opakuje cyklus.

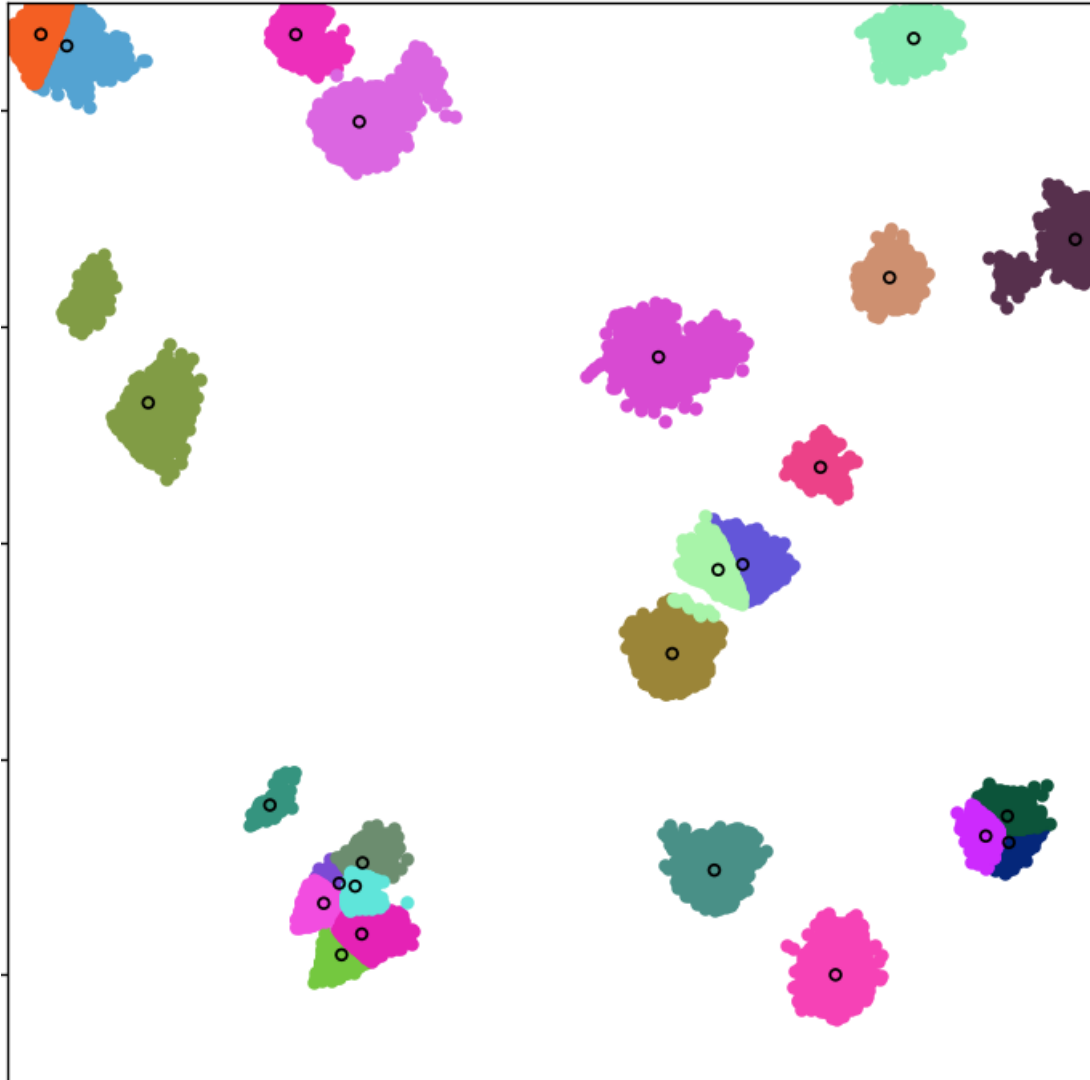
Použité knižnice:

matplotlib.pyplot, random, numpy, time, scipy.spatial.distance, scipy.spatial

Dosahované výsledky:

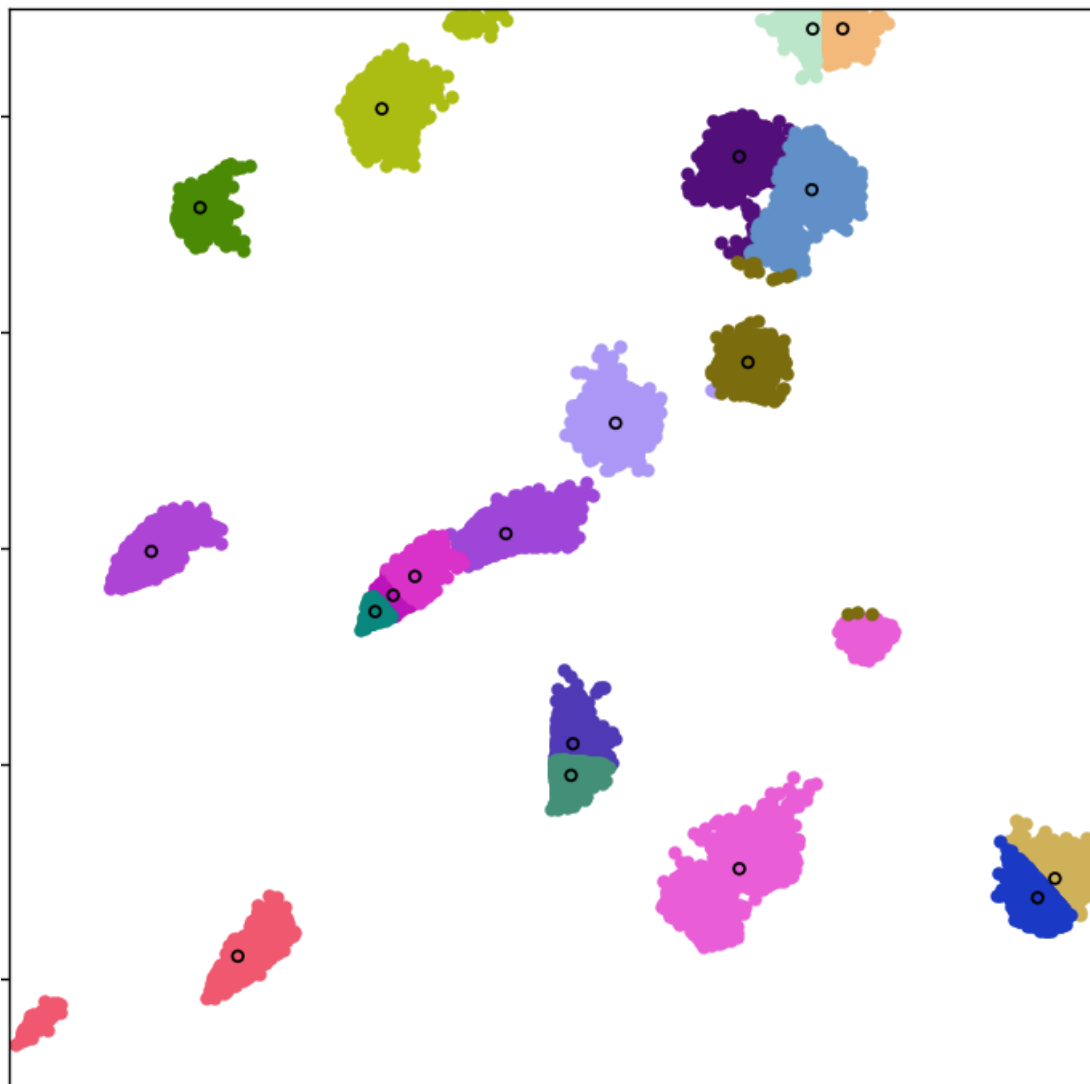
Testovanie prebiehalo na 40000 bodoch.

Algoritmus k-means, kde stred je centroid:



Počet klastrov: 25

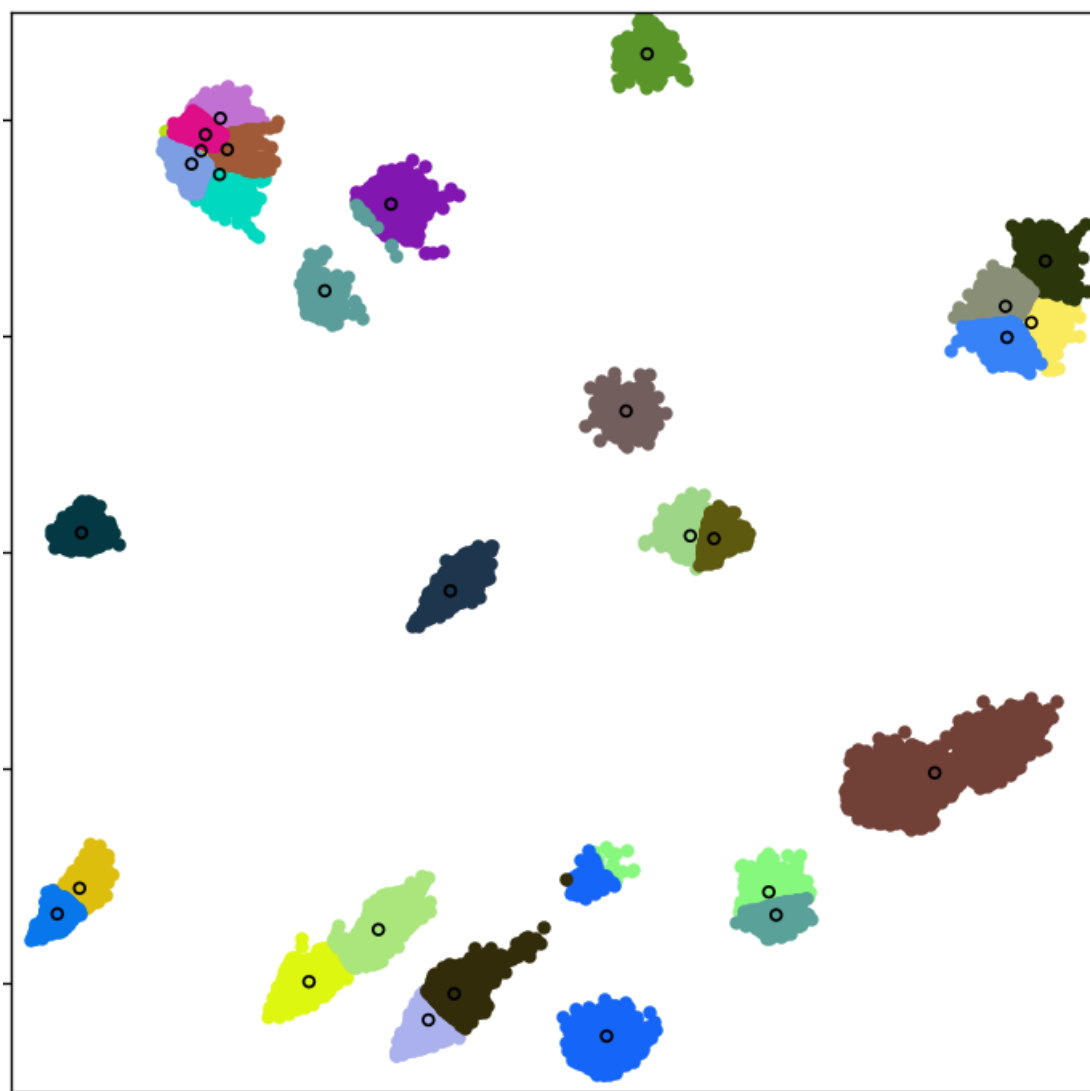
Algoritmus bežal 2.67 sekúnd.



Počet klastrov: 19

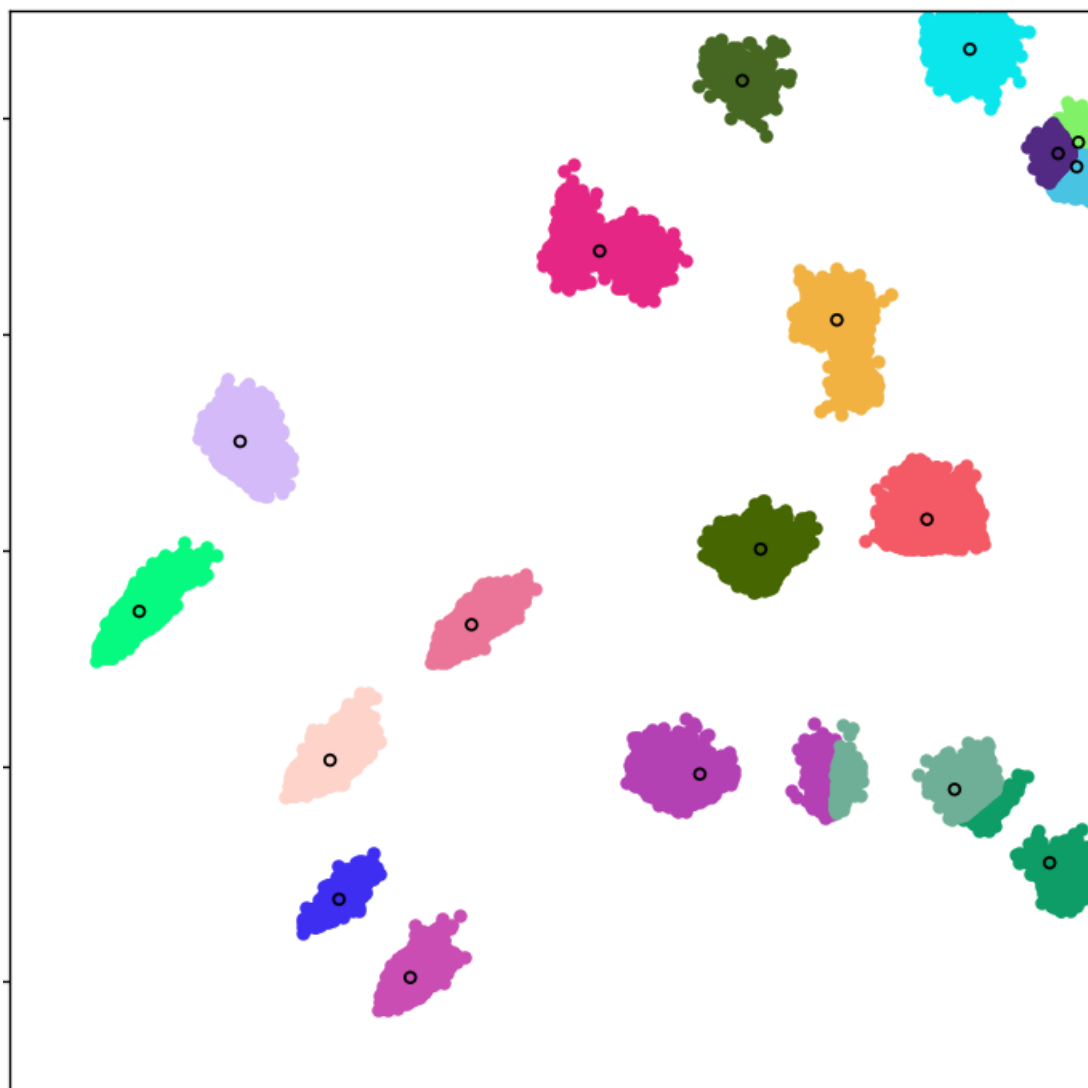
Algoritmus bežal 2.2 sekúnd.

Algoritmus k-means, kde stred je medoid:



Počet klastrov: 28

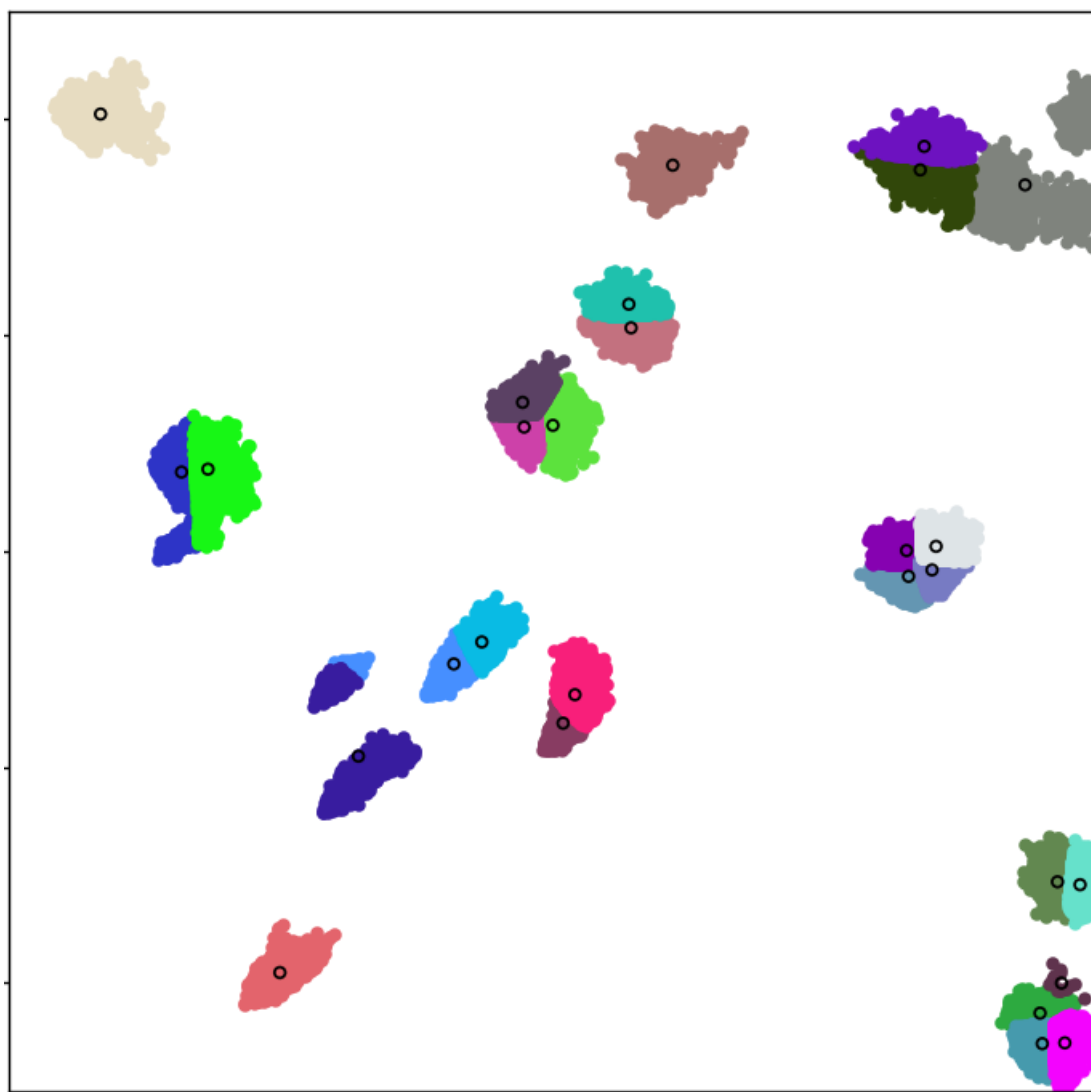
Algoritmus bežal 2 minúty a 1.25 sekundy.



Počet klastrov: 18

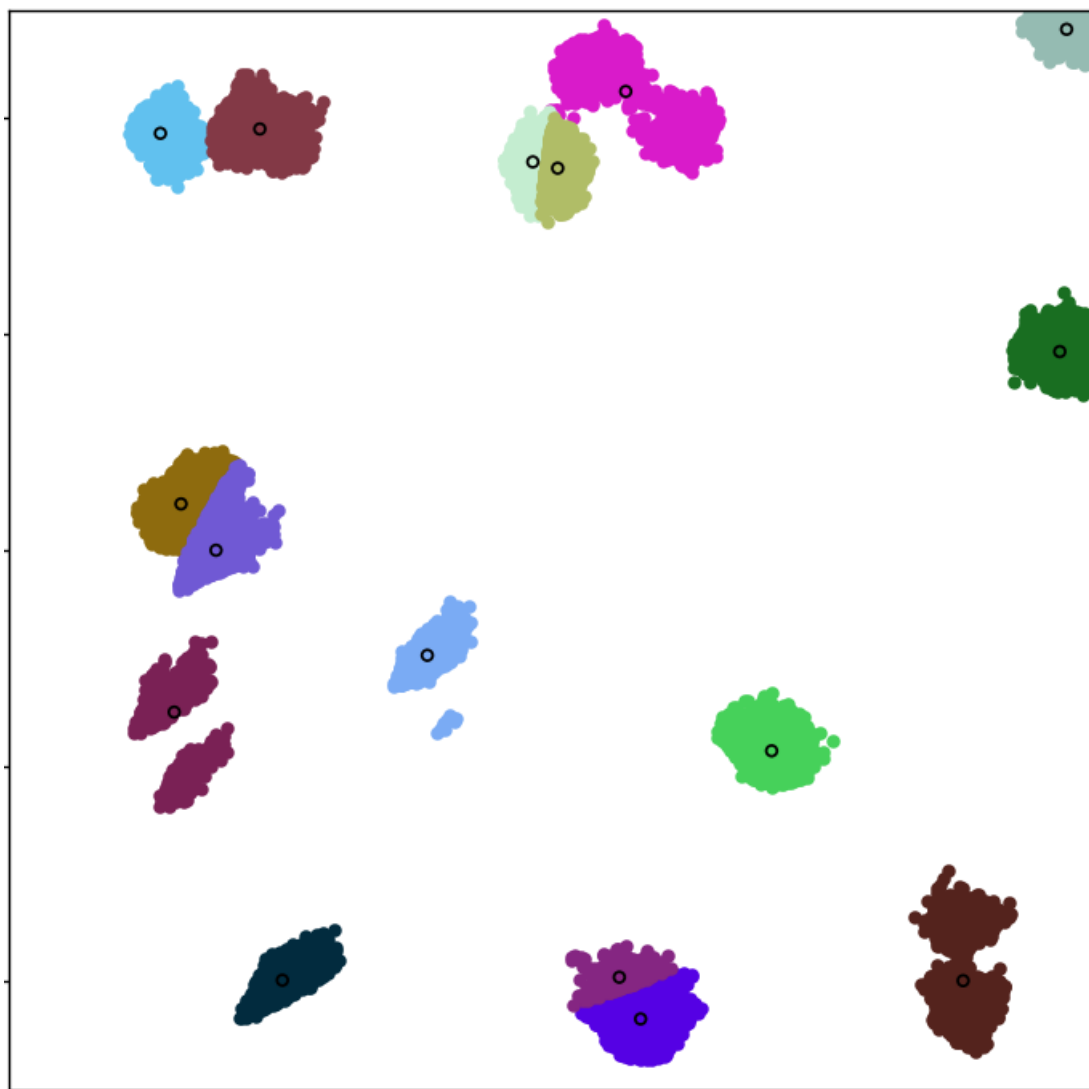
Algoritmus bežal 1 minútu a 52.71 sekúnd.

Algoritmus divízneho zhľukovania, kde stred je centroid:



Počet klastrov: 29

Algoritmus bežal 7.4 sekúnd.



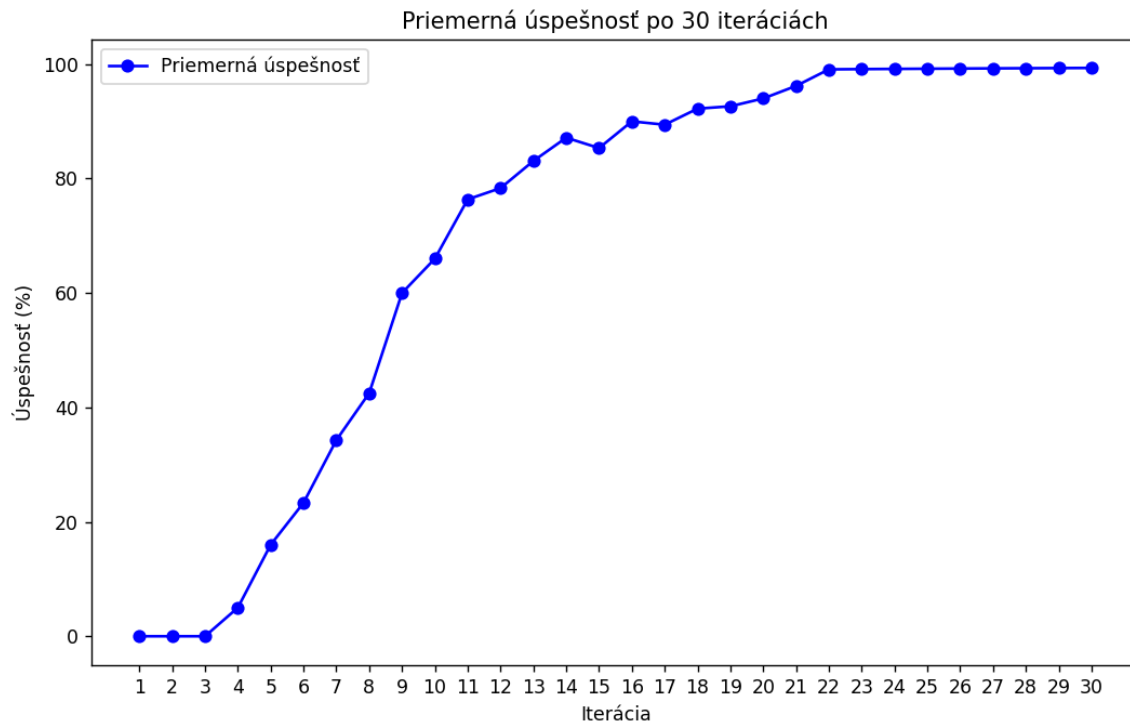
Počet klastrov: 15

Algoritmus bežal 4.15 sekúnd.

Zhodnotenie výsledkov:

Budeme vyhodnocovať priemerné výsledky algoritmov na vzorke 30-tich výsledkov.

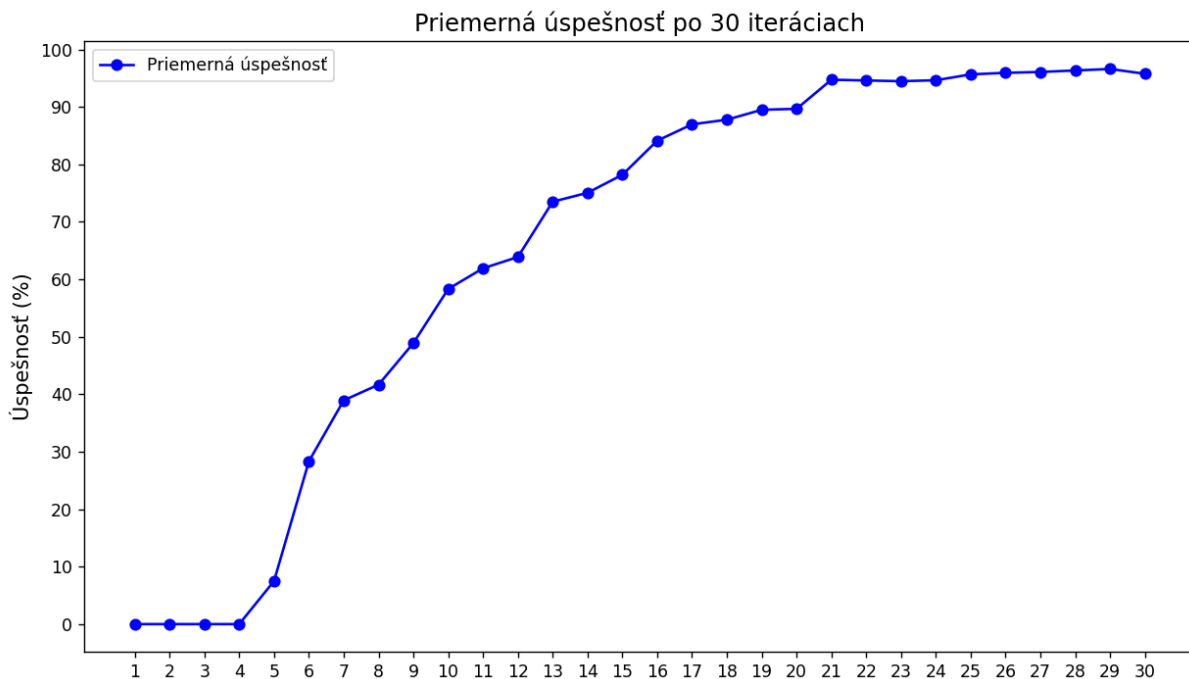
Algoritmus k-means, kde stred je centroid:



Priemerný počet klastrov: 29

Priemerný beh algoritmu: 4.31 sekundy

Algoritmus celkovo dosahoval dobré výsledky za najrýchlejší čas, aj keď sa niekedy stal výkyv bol malý.

Algoritmus k-means, kde stred je medoid:

Priemerný počet klastrov: 42

Priemerný beh algoritmu: 2 minúty a 9.48 sekundy

Tento algoritmus mal najdlhší priemerný čas behu a zároveň najväčší priemerný počet klastrov a zároveň jeho hodnoty obsahovali najviac výkyvov.

Algoritmus divízneho zhľukovania, kde stred je centroid:

Priemerný počet klastrov: 39

Priemerný beh algoritmu: 10.04 sekundy

Tento algoritmus bol druhý najrýchlejší a mal druhý najlepší priemerný počet klastrov. Algoritmus prejavoval najväčšie výkyvy ale nie tak často a zároveň prichádzal na najlepšie riešenia občasne.

Celkové zhodnotenie:

Celkovo by som vyhodnotil prvý algoritmus za najlepší, bežal najrýchlejšie a preukazoval dobré výsledky pravidelne. Za najhorší algoritmus by som vyhodnotil druhý, bol najpomalší a pravidelne preukazoval výkyvy hodnôt. Tretí algoritmus by som prirovnal „žolíkovi“, môže prísť s veľmi dobrým riešením alebo s veľmi zlým, ale celkovo preukazoval akceptovateľné hodnoty.