Slovenská technická univerzita

Fakulta informatiky a informačných technológií

Umelá inteligencia - Zadanie č. 4

Klasifikácia

autor: Jakub Hajdu

cvičiaci: Ing. Ivan Kapustík

akad. rok 2020/21

Jakub Hajdu
ID: 105340

Zadanie úlohy

Máme 2D priestor, ktorý má rozmery *X* a *Y*, v intervaloch od −5000 do +5000. V tomto priestore sa môžu nachádzať body, pričom každý bod má určenú polohu pomocou súradníc *X* a *Y*. Každý bod má unikátne súradnice (t.j. nemalo by byť viacej bodov na presne tom istom mieste). Každý bod patrí do jednej zo 4 tried, pričom tieto triedy sú: red (*R*), green (*G*), blue (*B*) a purple (*P*). Na začiatku sa v priestore nachádza 5 bodov pre každú triedu (dokopy teda 20 bodov). Súradnice počiatočných bodov sú:

R: [−4500, −4400], [−4100, −3000], [−1800, −2400], [−2500, −3400] a [−2000, −1400]

G: [+4500, −4400], [+4100, −3000], [+1800, −2400], [+2500, −3400] a [+2000, −1400]

B: [−4500, +4400], [−4100, +3000], [−1800, +2400], [−2500, +3400] a [−2000, +1400]

P: [+4500, +4400], [+4100, +3000], [+1800, +2400], [+2500, +3400] a [+2000, +1400]

Vašou úlohou je naprogramovať klasifikátor pre nové body – v podobe funkcie classify(int *X*, int *Y*, int *k*), ktorá klasifikuje nový bod so súradnicami *X* a *Y*, pridá tento bod do nášho 2D priestoru a vráti triedu, ktorú pridelila pre tento bod. Na klasifikáciu použite *k*-NN algoritmus, pričom *k* môže byť 1, 3, 7 alebo 15.

Na demonštráciu Vášho klasifikátora vytvorte testovacie prostredie, v rámci ktorého budete postupne generovať nové body a klasifikovať ich (volaním funkcie classify). Celkovo vygenerujte 40000 nových bodov (10000 z každej triedy). Súradnice nových bodov generujte náhodne, pričom nový bod by mal mať zakaždým inú triedu (dva body vygenerované po sebe by nemali byť rovnakej triedy):

- R body by mali byť generované s 99% pravdepodobnosťou s X < +500 a Y < +500
- G body by mali byť generované s 99% pravdepodobnosťou s X > −500 a Y < +500
- B body by mali byť generované s 99% pravdepodobnosťou s X < +500 a Y > −500
- P body by mali byť generované s 99% pravdepodobnosťou s X > −500 a Y > −500

Návratovú hodnotu funkcie classify porovnávajte s triedou vygenerovaného bodu. Na základe týchto porovnaní vyhodnoťte úspešnosť Vášho klasifikátora pre daný experiment.

Experiment vykonajte 4-krát, pričom zakaždým Váš klasifikátor použije iný parameter *k* (pre *k* = 1, 3, 7 alebo 15) a vygenerované body budú pre každý experiment rovnaké.

Vizualizácia: pre každý z týchto experimentov vykreslite výslednú 2D plochu tak, že vyfarbíte túto plochu celú. Prázdne miesta v 2D ploche vyfarbite podľa Vášho klasifikátora.

V závere zhodnoťte dosiahnuté výsledky ich porovnaním.

Jakub Hajdu
ID: 105340

Reprezentácia údajov

Najpodstatnejšou časťou reprezentácie dát programu je slovník **points_dict**, v ktorom sú uložené všetky body stavového priestoru. Kľúčom sú súradnice x a y v podobe tuple. Hodnotou je objekt bodu triedy **Point**, ktorá obsahuje atribúty reálnej vygenerovanej farby, farby pridelenej klasifikátorom a zoznam najbližších bodov.

Najbližšie body sú v zoznamoch ukladané ako objekty triedy **Neighbor**, ktoré obsahujú odkaz na svoj vlastný objekt bodu a vzdialenosť od bodu, v ktorého zozname sa nachádzajú. Z dôvodu šetrenia pamäte je dĺžka týchto zoznamov vždy prispôsobená najvyššiemu zadanému k, takže sa vždy ukladá len potrebný počet najbližších bodov.

Pre rýchlejšiu vizualizáciu je lepšie, keď sú funkcii **scatter()** z použitej knižnice **matplotlib** body zadané naraz. Tento postup urýchlil vykresľovanie grafov pri 20 tisícoch bodov tisícnásobne. Súradnice a farby sa preto pred vykreslením ukladali do jednoduchých zoznamov **x_list, y_list** a **color_list**.

Popis použitého algoritmu

Program začína v hlavnom while cykle pre menu aplikácie a najprv načítava vstupné body zo súboru (20 bodov zo zadania je zapísaných v súbore *init.txt*, obdobne by sa dali načítavať iné množiny inicializačných bodov pri dodržaní formátu aký má tento súbor). Ďalej sa programu ako vstupné údaje zadáva počet bodov na vygenerovanie a klasifikáciu a všetky hodnoty k, pre ktoré má vykonať n klasifikáciu na tejto jednej množine bodov.

Po načítaní vstupných údajov sa začnú generovať body v zadanom množstve a s pravdepodobnosťami súradníc podľa farieb v súlade so zadaním. Na generovanie slúži funkcia **generate_points().** V tejto funkcii sa zároveň overuje duplicitné generovanie, aby sa predišlo vygenerovaniu viacerých bodov s rovnakými súradnicami. Súradnice podľa jednotlivých farebných tried sú definované vo funkcii **new_point()**, ktorá vracia náhodne generované súradnice zo správneho rozsahu. Priradenie farebného označenia (R, G, B, P) zabezpečuje pomocná funkcia **color_by_mod()**.

Po vygenerovaní každého bodu sú do jeho zoznamu priradené objekty okolitých bodov spolu s vypočítanými vzdialenosťami od aktuálneho bodu. Úlohu výpočtu a priraďovania vzdialeností bodov spĺňa funkcia **calculate_distances(),** v ktorej sa vypočíta vzdialenosť od každého bodu dovtedy existujúceho v stavovom priestore (Euklidovskou vzdialenosťou pomocou funkcie **euclidean()**), pričom sa tieto body zoraďujú podľa vzdialenosti od najkratšej a udržiava sa len potrebný počet bodov s cieľom šetrenia pamäte. Pre každý bod sa brali do úvahy iba body, ktoré dovtedy existovali, aby klasifikácia prebiehala správnym spôsobom

Po vygenerovaní všetkých bodov a priradení najbližších susedov sa pristupuje k samotnej klasifikácii vo funkcii **knn()**, kde sa v cykle volá pre každý generovaný bod funkcia **classify()**, ktorá priradí bodu farbu na základe toho, ktorá farba je v okolí k najbližších bodov najviac zastúpená. Početnosti farieb sú ukladané do slovníka, ktorý je následne roztriedený v poradí od najväčšej početnosti po najmenšiu a klasifikovanému bodu sa priradí farba prvého prvku v slovníku (najpočetnejšia farba). Farba z klasifikátora sa porovnáva so skutočnou farbou bodu a výstupom funkcie **knn()** je samotná percentuálna úspešnosť klasifikátora.

Vizualizácia prebieha v hlavnom cykle programu (ak je zapnutá možnosť VISUALIZE) pomocou funkcie **scatter()** z knižnice **matplotlib**, ktorá má vopred implementované aj potrebné škálovanie plochy grafu. Ako vstup sú tvorené zoznamy súradníc x, y a farieb jednotlivých bodov.

Jakub Hajdu
ID: 105340

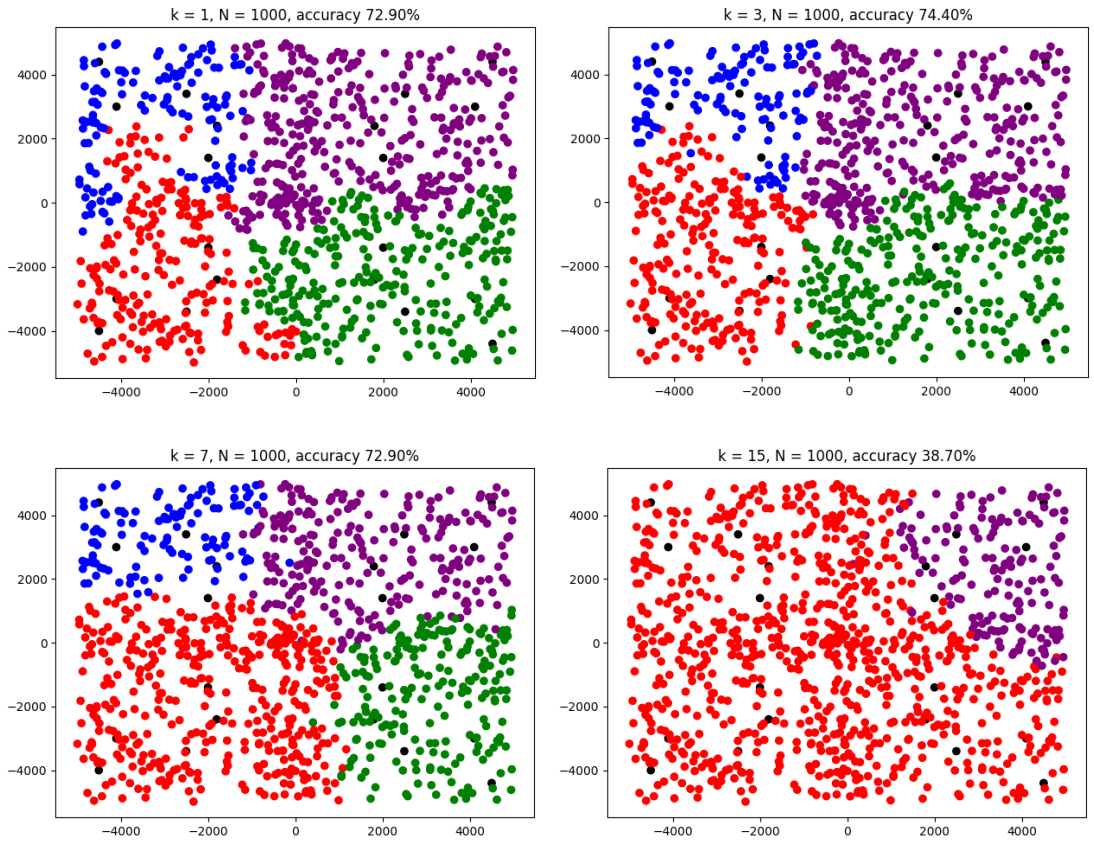
Testovanie

Testovanie v úvodných fázach písania programu spočívala hlavne v kontrolných výpisoch pre kontrolu správnosti funkcionality generovania bodov, výpočtu vzdialeností medzi bodmi, tvorenia zoznamov vzdialeností, ich zoraďovania a správneho posudzovania farieb okolitých bodov.

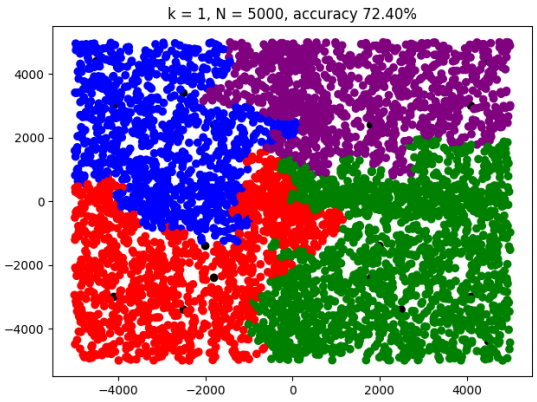
Po overení správneho fungovania klasifikačného algoritmu bol program spúšťaný pre rôzne počty bodov, zakaždým pre všetky štyri zadané hodnoty k (1, 3, 7, 15) a výsledky boli porovnávané kvantitatívne pomocou percentuálnej úspešnosti klasifikátora, ako aj vizuálne pomocou vizualizácie grafmi.

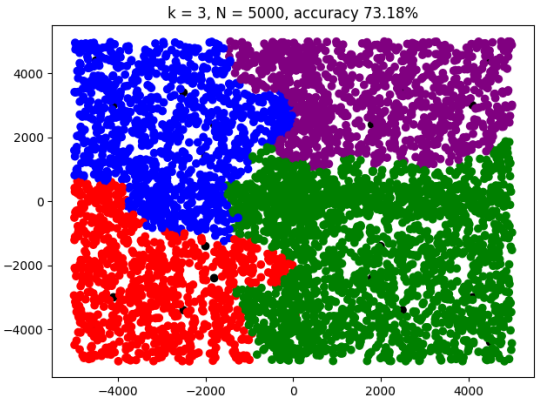
Klasifikácia pre jednotlivé štvorice hodnôt k bola vždy spúšťaná na rovnakej množine bodov, pričom boli dodržané pravdepodobnosti generovania bodov jednotlivých farebných tried v rozsahoch súradníc zo zadania. Nižšie je možné vidieť príklad textového výpisu na porovnanie percentuálnej úspešnosti štyroch zadaných hodnôt k a výstupné grafy pre počty bodov 1000, 5000, 10000 a 20000.

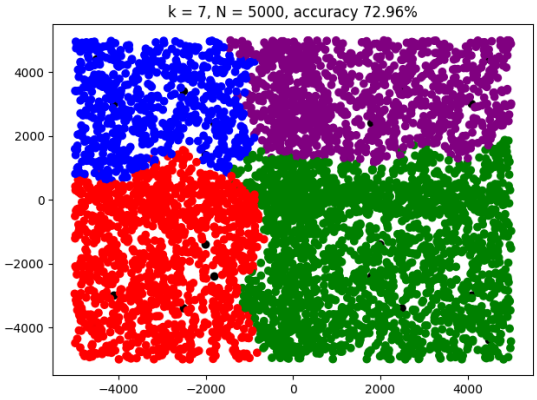
```
[STARTING] k-nn(1)
[DONE] k-nn: Correctly classified points: 70.89 %

[STARTING] k-nn(3)
[DONE] k-nn: Correctly classified points: 71.30 %

[STARTING] k-nn(7)
[DONE] k-nn: Correctly classified points: 70.93 %

[STARTING] k-nn(15)
[DONE] k-nn: Correctly classified points: 37.10 %
```

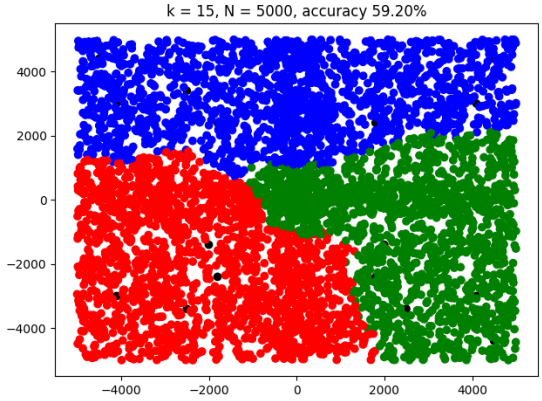


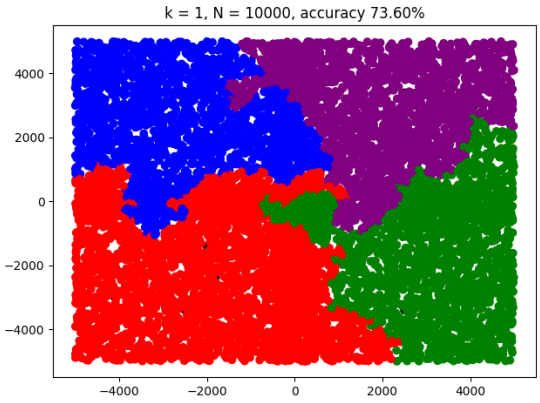
Jakub Hajdu
ID: 105340

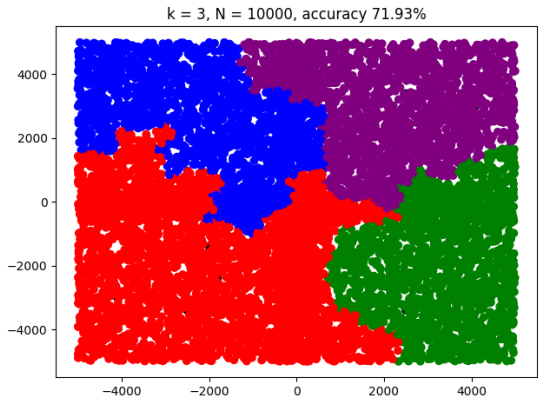


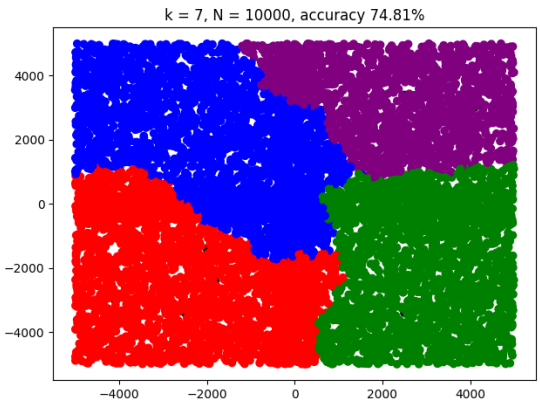


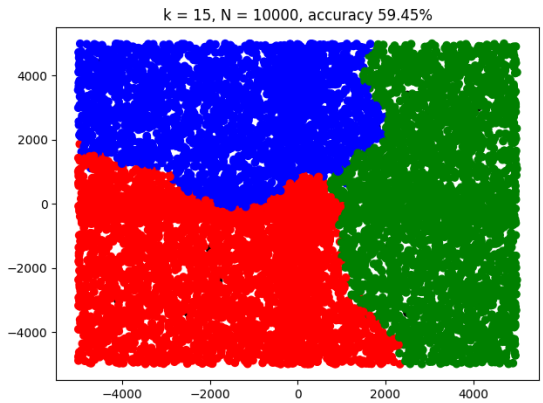




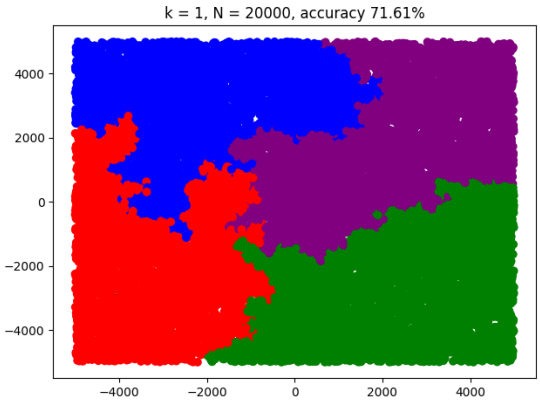


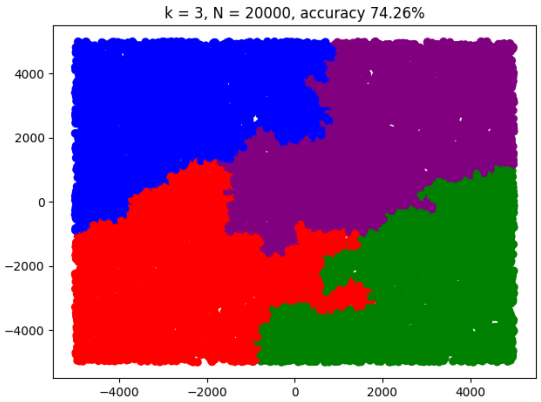


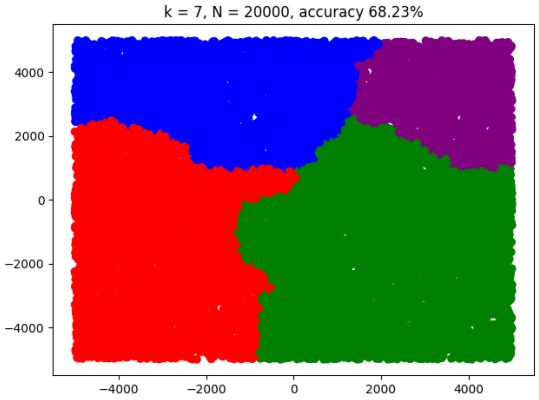


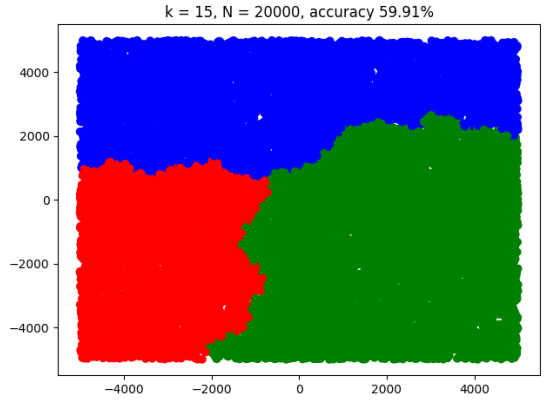


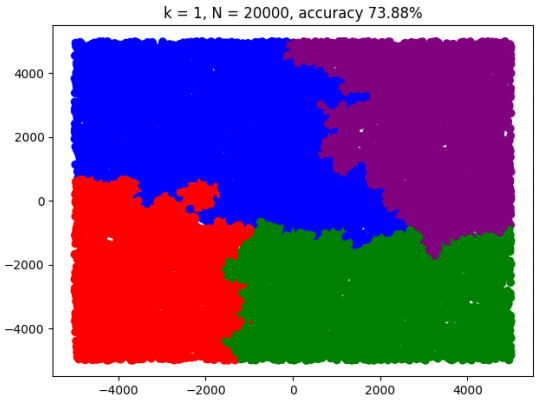
Jakub Hajdu
ID: 105340

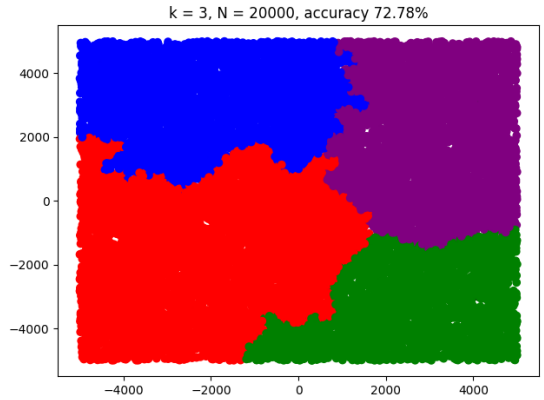


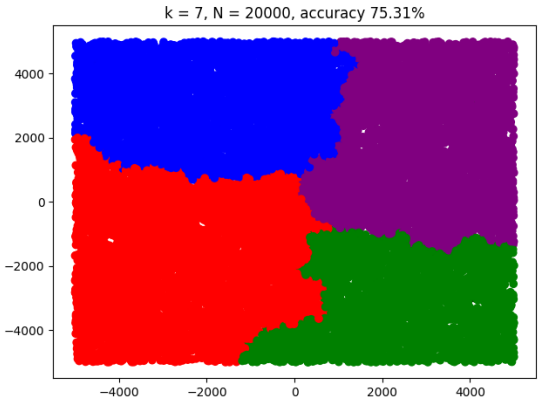


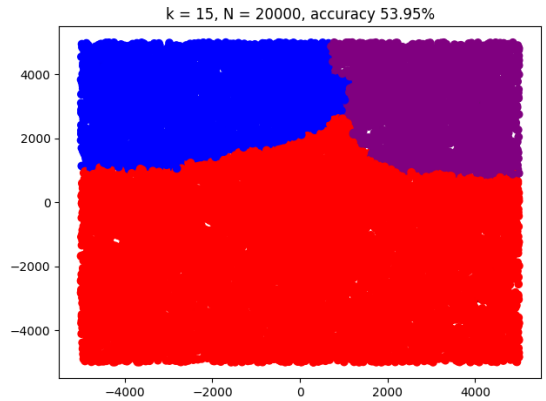












Jakub Hajdu
ID: 105340

Zhodnotenie

Program bol otestovaný na zadaných 20 tisícoch bodov, ako aj na menších množinách (napríklad 100, 1000, 2500, 5000 a 10000). Vo všetkých prípadoch vykazoval algoritmus výrazne najhoršiu úspešnosť pri k = 15. Tá sa v priemere pohybovala od 35% do 60%. Úspešnosť ostatných hodnôt (1, 3, 7) sa navzájom veľmi nelíšila a takmer vždy dosahovala okolo 70% až 76%. Po zhodnotení väčšieho množstva spustení zakaždým pre tieto štyri hodnoty vyšlo najavo, že najlepšiu úspešnosť dosahovali klasifikácie pre k = 3 a k = 7, pričom medzi týmito bol rozdiel rádovo v desatinách percenta.

Voľba implementačného prostredia

Program je napísaný v jazyku Python. Prostredím pre vývoj bol Pycharm vo verzii 2020.2.3 s verziou python 3.8. Inicializačný súbor **init.txt** bol vytvorený v NotePade ako jednoduchý txt súbor.

Jakub Hajdu
ID: 105340

Opis používateľského rozhrania

Hlavným prostredím používateľského rozhrania je konzola, vykreslené grafy s vizualizáciou sa zobrazujú v novom okne. Po spustení sa zobrazí úvodná možnosť pre načítanie súboru alebo ukončenie programu.

```
Load from file = 1
Quit = 0
```

Po zadaní „0“ program skončí, zadaním možnosti „1“ sa zobrazí výzva na meno inicializačného súboru.

```
Please enter the name of the file (without extension)
```

Ak bol zadaný neplatný názov súboru, program na to upozorní, inak pokračuje na voľbu počtu bodov.

```
[INIT] Loading initial points from file...
[INIT] Done

Specify the number of points to be classified
```

Následne je možné zapnúť alebo vypnúť vizualizáciu grafmi.

```
Graph ON = 1
Graph OFF = 0
```

Nakoniec je potrebné zadať ešte hodnoty k pre knn algoritmus oddelené medzerami.

```
1
Specify the k values [k1 k2 k3 ... kN]
```

Po vygenerovaní a klasifikovaní zadaného počtu bodov program vypíše úspešnosti zadaných hodnôt k.

```
100 %
[DONE] Point generator

[STARTING] k-nn(1)
[DONE] k-nn: Correctly classified points: 72.10 %

[STARTING] k-nn(3)
[DONE] k-nn: Correctly classified points: 74.20 %

[STARTING] k-nn(7)
[DONE] k-nn: Correctly classified points: 76.80 %

[STARTING] k-nn(15)
[DONE] k-nn: Correctly classified points: 61.70 %
```

V prípade zapnutej vizualizácie sa zobrazia aj príslušné grafy, ktoré je možné uložiť.