Slovenská technická univerzita v Bratislave
Fakulta informatiky a informačných technológií

Umelá Inteligencia

# Zadanie č.3

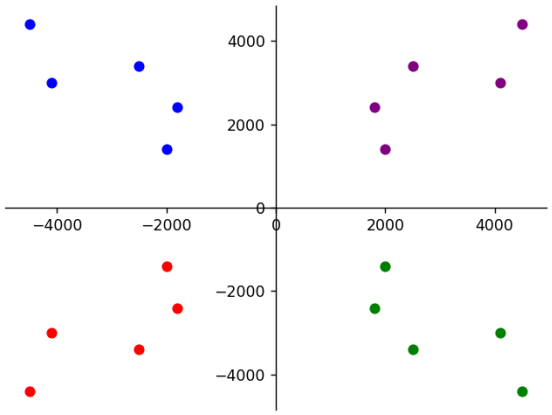
## a) Klasifikácia

Akademický rok 2022/2023

**Meno:** Bc. Irina Makarova
**Cvičiaci:** Ing. Martin Komák, PhD. **Dátum:** 11.12.2022

## Definícia problému

Je daná karteziánska súradnicová sústava, hodnoty na x-ovej a y-ovej osiach sú celočíselné a patria do intervalu -5000 do 5000. V počiatočnom stave sú v nej obsadené 20 vopred definovaných súradníc podľa obrázku nižšie.




Do sústavy treba pridať 10000 bodov z každej zo štyroch kategórií, pričom bod danej farby sa s väčšou pravdepodobnosťou vygeneruje v príslušnom kvadrante. Novo pridané body sa majú klasifikovať pomocou k-NN algoritmu, ktorý spočíva v zaradení bodu do istej kategórie podľa počtu susedných bodov patriacich do rovnakej skupiny. Algoritmus nájde k bodov priestorovo najbližších analyzovanému a priradí mu najfrekventovanejšiu kategóriu v zozname susedných bodov. Efektivita algoritmu sa počítala porovnaním takto určenej kategórie s automaticky pridelenou kategóriou.

## Slovný opis algoritmu

Keďže sa rôzne hodnoty k majú testovať na rovnakých dátach, najprv sa vygeneruje celá množina bodov spolu s vopred zadanými počiatočnými bodmi (celkovo 40020 súradníc). Súradnice všetkých bodov majú byť unikátne, čo sa zaisťuje ukladaním jednotky do 2D poľa veľkosti 10001x10001, kde index riadku dostaneme pripočítaním 5000 k y-ovej súradnici bodu a index stĺpca je súčet 5000 a hodnoty x-ovej súradnice bodu. Hodnota 5000 sa pripočítava z dôvodu odstránenia záporných čísel. Rozloženie bodov v súradnicovej sústave sa teda premietne do matice, čo umožní rýchle overovanie, či sa nevygenerovali duplicitné súradnice.

```
self.area = [[0 for x in range(10001)] for y in range(10001)]
```

Triedy bodov sú v programe zakódované číslami od 0 do 3 v poradí red-green-blue-purple. Súradnice zvyšných 40000 bodov sa generujú náhodné, pričom budú s pravdepodobnosťou 99% patriť do štvorca vyhradeného pre jeho príslušnú kategóriu. Vo funkcii *generateCoords* sa najprv vygeneruje náhodné číslo z intervalu (0; 1), podľa ktorého sa určuje, z akého rozsahu hodnôt sa môžu vygenerovať súradnice nového bodu. Ak je jeho hodnota väčšia ako 0.99, bod sa môže umiestniť na ľubovoľné prázdne miesto v sústave, v opačnom prípade je rozsah súradníc obmedzený.

```python
def generateCoords(pointType):
    percentageOfPointsoGeneratedInItsSquare = 0.99
    percentageOfPointsGeneratedInGivenArea = random.random()
    if percentageOfPointsGeneratedInGivenArea <= percentageOfPointsoGeneratedInItsSquare:
        match pointType:
            case 0:
                return (random.randint(-5000, 499), random.randint(-5000, 499))
            case 1:
                return (random.randint(-499, 5000), random.randint(-5000, 499))
            case 2:
                return (random.randint(-5000, 499), random.randint(-499, 5000))
            case 3:
                return (random.randint(-499, 5000), random.randint(-499, 5000))
    return (random.randint(-5000, 5000), random.randint(-5000, 5000))
```

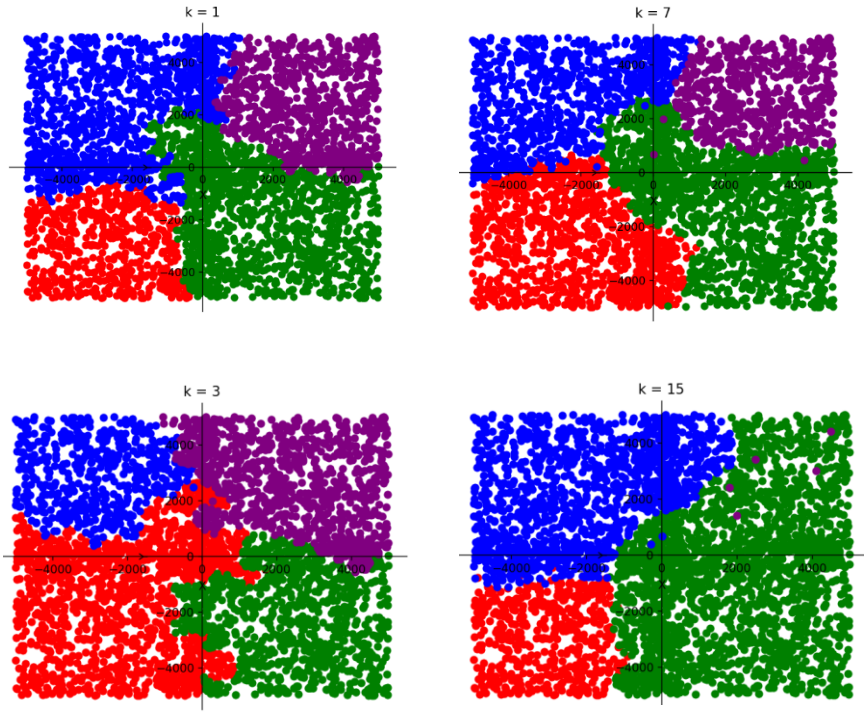
Vygenerované body sa postupne ukladajú do poľa *newPoints*, ktoré je súčasťou triedy *Set*. Z tohto poľa sa následne po jednom vyberajú body, spracúvajú sa vo funkcii *testing*, ktorá volá hlavnú funkciu *classify* a ukladajú sa do poľa už spracovaných bodov, z ktoré sa v ďalšom cykle použijú na hľadanie najbližších susedných bodov pre novo vytvorený bod.

Funkcia *classify* dostáva na vstupe zoznam doteraz spracovaných bodov, hodnotu k a súradnice práve spracúvaného bodu. Najprv sa nej pre každý susedný bod vypočíta Euklidovská vzdialenosť od novo vygenerovaného bodu a následne sa body podľa nej usporiadajú vzostupne. Prvých k najbližších bodov sa použije na určenie triedy spracúvaného bodu, a to takým spôsobom, že sa mu pridelí trieda najfrekventovanejšieho bodu v zozname k bodov.

```python
def classify(x, y, k, vicinity):
    for point in vicinity:
        point.calculateDistance(x, y)
    vicinity.sort(key=lambda el: el.distanceFromNewPoint)
    freq = [0,0,0,0]
    for p in range(k):
        freq[vicinity[p].color] += 1
    return freq.index(max(freq))
```

## Experimenty pre 4000 bodov

Pre každú kategóriu bodov sa vygenerovalo 1000 bodov (celkovo 4000 bodov). Pre hodnoty k z množiny {1,3,7} sa dosahovala úspešnosť 75%. Pri k = 15 úspešnosť klasifikácie výrazne klesá, lebo v prípade väčšieho počtu vzdialených bodov z inej skupiny sa nový bod zaradí do nej, aj keď v jeho blízkom okolí sú body inej farby. Preto nakoniec prevládne jedná kategória.

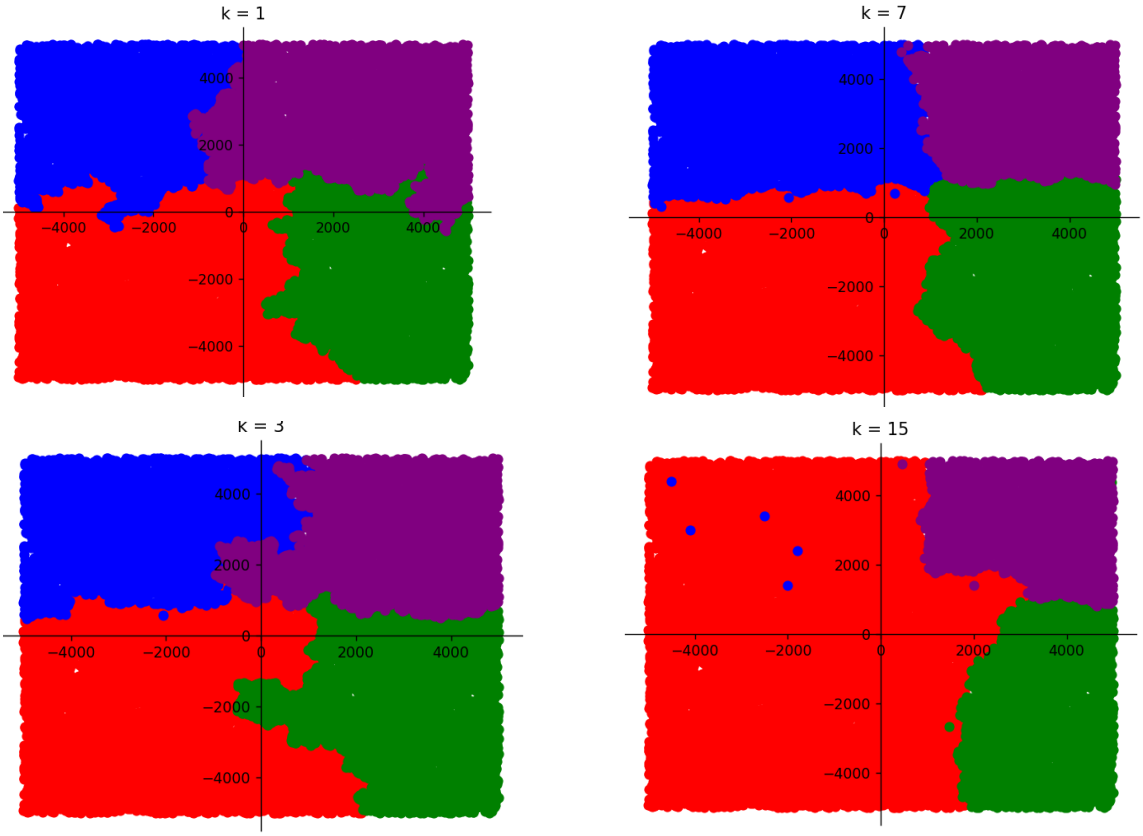



**Čas vykonania algoritmu (v sekundách) a úspešnosť výsledku (v percentách)**

```
k =  1
uspesnost:  75.3
cas vykonania:  12.392529499978991
k =  3
uspesnost:  75.55
cas vykonania:  14.610573900019517
k =  7
uspesnost:  74.925
cas vykonania:  14.873159900016617
k =  15
uspesnost:  59.275
cas vykonania:  15.786510999983875
```

## Experimenty pre 20000 bodov

Pre každú kategóriu bodov sa vygenerovalo 5000 bodov (celkovo 20000 bodov). Pre hodnoty k z množiny {1,3,7} dosahovala úspešnosť podobné hodnoty ako v predchádzajúcom bode. Rovnako pri k = 15 úspešnosť klasifikácie výrazne klesá, lebo sa pri klasifikácii zohľadňuje príliš veľké okolie novo pridaného bodu. Preto nakoniec prevládne jedná kategória.

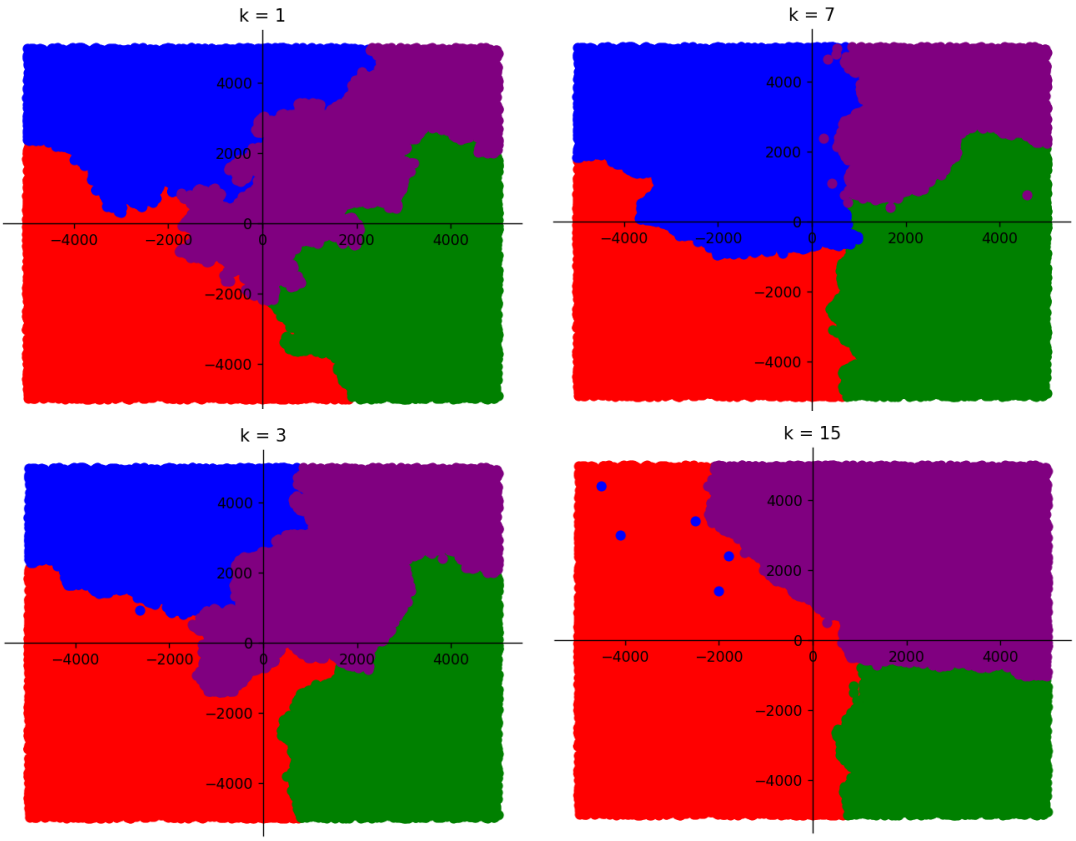



**Čas vykonania algoritmu (v sekundách) a úspešnosť výsledku (v percentách)**

```
k =  1
uspesnost:  75.965
cas vykonania:  410.2959922000009
k =  3
uspesnost:  74.91
cas vykonania:  400.01086430001305
k =  7
uspesnost:  74.02
cas vykonania:  393.3885695999779
k =  15
uspesnost:  50.615
cas vykonania:  391.65713500001584
```

## Experimenty pre 40000 bodov

Pre každú kategóriu bodov sa vygenerovalo 10000 bodov (celkovo 40000 bodov). Najväčšia úspešnosť sa dosiahla pri hodnote k = 7. Pri k = 15 úspešnosť klasifikácie výrazne klesá rovnako ako v predošlých prípadoch, lebo sa pri klasifikácii zohľadňuje príliš veľké okolie novo pridaného bodu, čo znižuje presnosť výsledku. Preto nakoniec prevládne jedná kategória.




**Čas vykonania algoritmu (v sekundách) a úspešnosť výsledku (v percentách)**

```
k =  1
uspesnost:  69.925
cas vykonania:  1758.614166799991
k =  3
uspesnost:  72.5175
cas vykonania:  1761.878411600017
k =  7
uspesnost:  73.675
cas vykonania:  1757.9581518999767
k =  15
uspesnost:  63.144999999999996
cas vykonania:  1760.8327627999824
fin
```