SLOVENSKÁ TECHNICKÁ UNIVERZITA

Fakulta informatiky a informačných technológií

v Bratislave

Umelá inteligencia – zadanie č.4

Jakub Taraba

Prednášajúci: Ing. Lukáš Kohútka, PhD.

Cvičiaci: Ing. Ivan Kapustík

Čas cvičení: Štvrtok 14:00

Jakub Taraba
AIS ID: 103159

Zadanie

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 20000 nových bodov (5000 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 $Y > -500$

(Zvyšné jedno percento bodov je generované v celom priestore.)

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.

Jakub Taraba
AIS ID: 103159

Reprezentácia údajov

Údaje, resp. body sú reprezentované pomocou triedy Point.

```
class Point:
    def __init__(self, x, y, color):
        self.coords = (x, y)
        self.color = color
        self.euclidean_distance = 0
```

Tieto body sú potom uložené v zozname, aby som vedel pracovať s tými istými bodmi pre rôzne K.

Opis algoritmu

1. Vygenerujem si všetky body s 99% pravdepodobnosťou a s 1% pravdepodobnosťou, na to slúži funkcia **generate_n_color_points(n, color, percentage)**
2. Náhodne si určím aká farba bude vygenerovaná, aby som potom neskôr mohol zistiť, či klasifikátor vyhodnotil bod s farbou korektne alebo s chybou
3. Začnem s generáciou bodu, pre každý bod si určím pravdepodobnosť jeho výskytu
4. Na základe pravdepodobnosti si vyberem bod z poľa 99% alebo 1% bodmi
5. Bod klasifikujem
6. Určím, či klasifikácia prebehla v poriadku a ak nie, pripočítam to ako chybu
7. Opakujem do bodu 2. až dokým nemám vygenerovaný dostatočný počet bodov

```
for i in range(n):
    coords = (random.randint(x1, x2), random.randint(y1, y2))
    point = Point(coords[0], coords[1], color)

    # ochrana proti generovaniu duplicitnych bodov
    while point in points:
        coords = (random.randint(x1, x2), random.randint(y1, y2))
        point = Point(coords[0], coords[1], color)

    points.append(point)

return points
```

Časť z funkcie generate_n_color_points, kde je vidieť ako generujem body do polí

```
while True:
    color = random.choice(colors)
    if color != last_color:
        break
```

Náhodné určenie farby

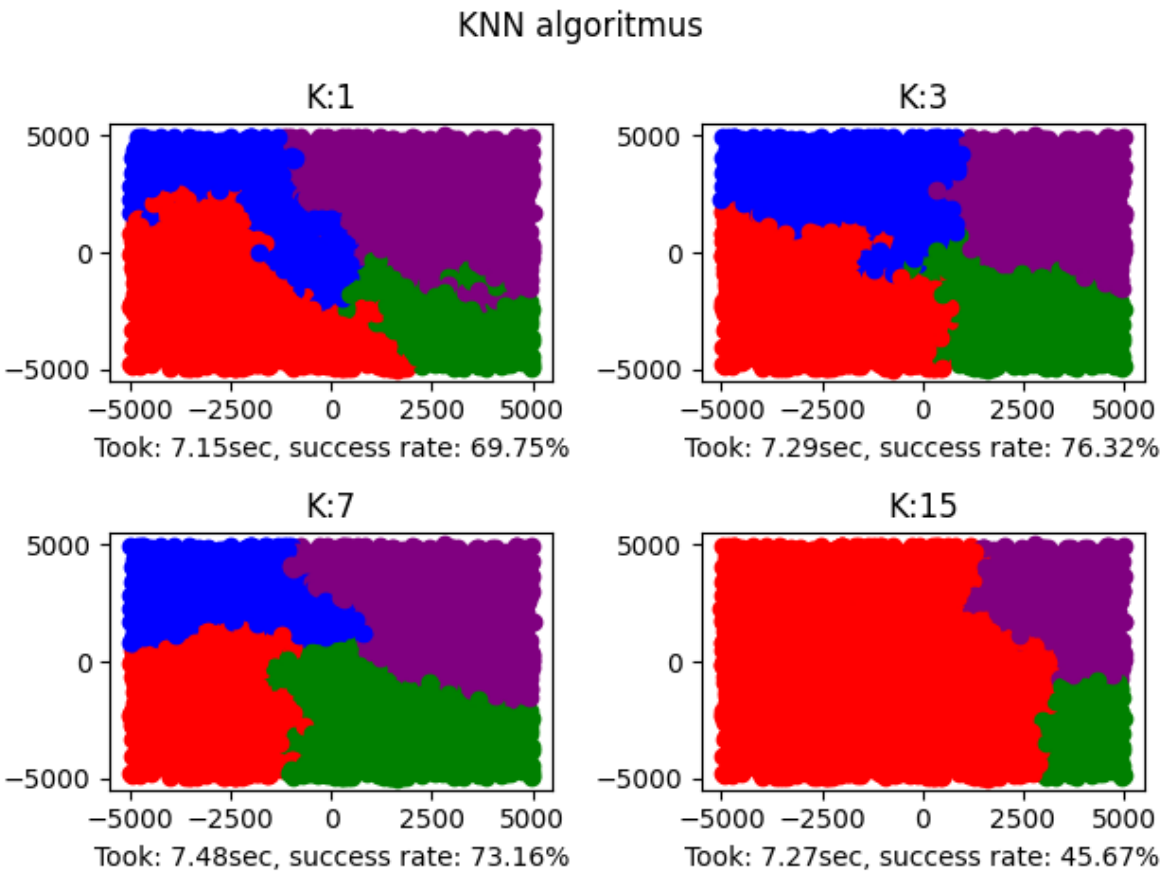
```
red_point = red_points_99[red_points] if percentage == 99 else red_points_1[red_points]
red_points += 1
classified_color = classify(red_point.coords[0], red_point.coords[1], knn_values[i])
```

Výber bodu a jeho klasifikácia

Jakub Taraba
AIS ID: 103159

Ovládanie programu

Program stačí spustiť, vybrať si počet bodov pre **jednu farbu** a počkať kým sa vygenerujú body a grafy. Program priebežne vypisuje pre aké K momentálne generuje body. Po úspešnom zbehnutí programu sa uloží súbor „vygenerovane_knn_grafy_k<CISLO>.png“, kde <CISLO> je počet bodov pre jednu farbu.

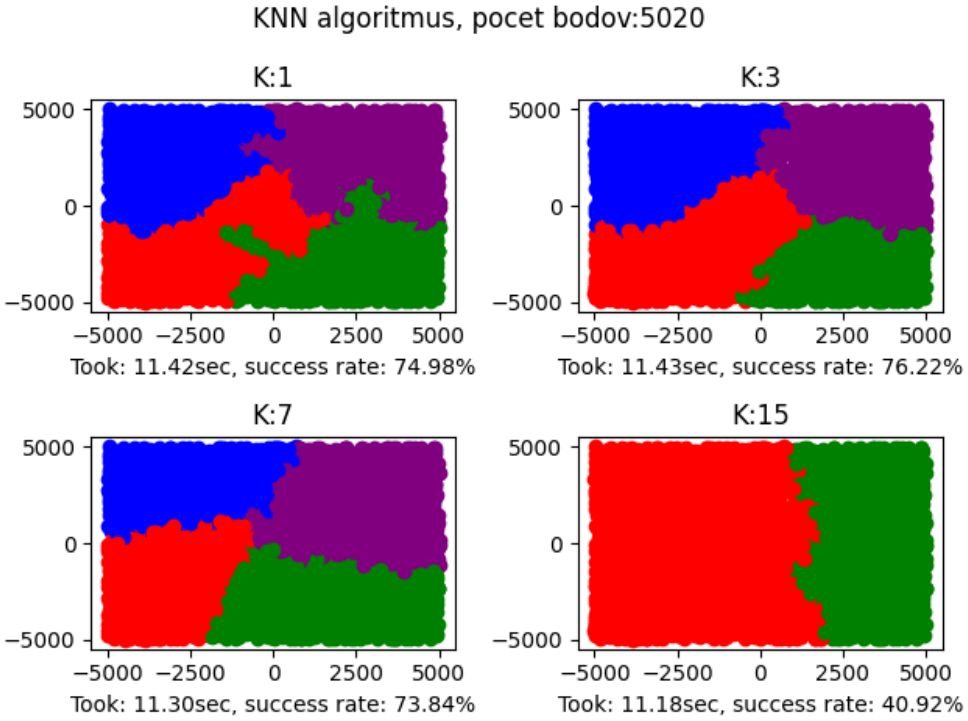


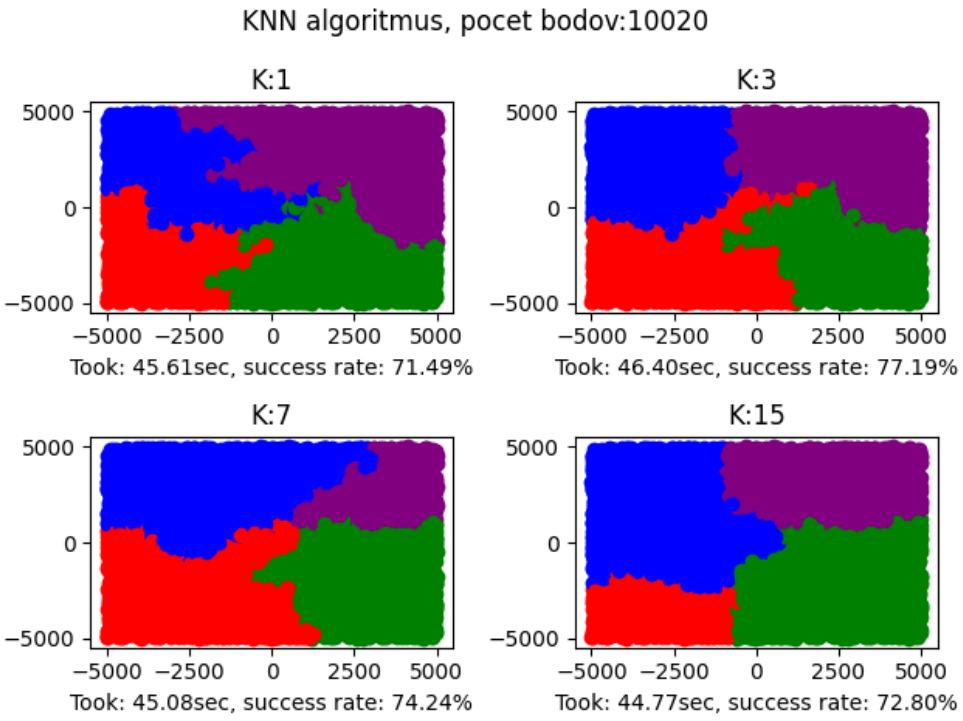
Ukážka vygenerovaného obrázku

Výsledky

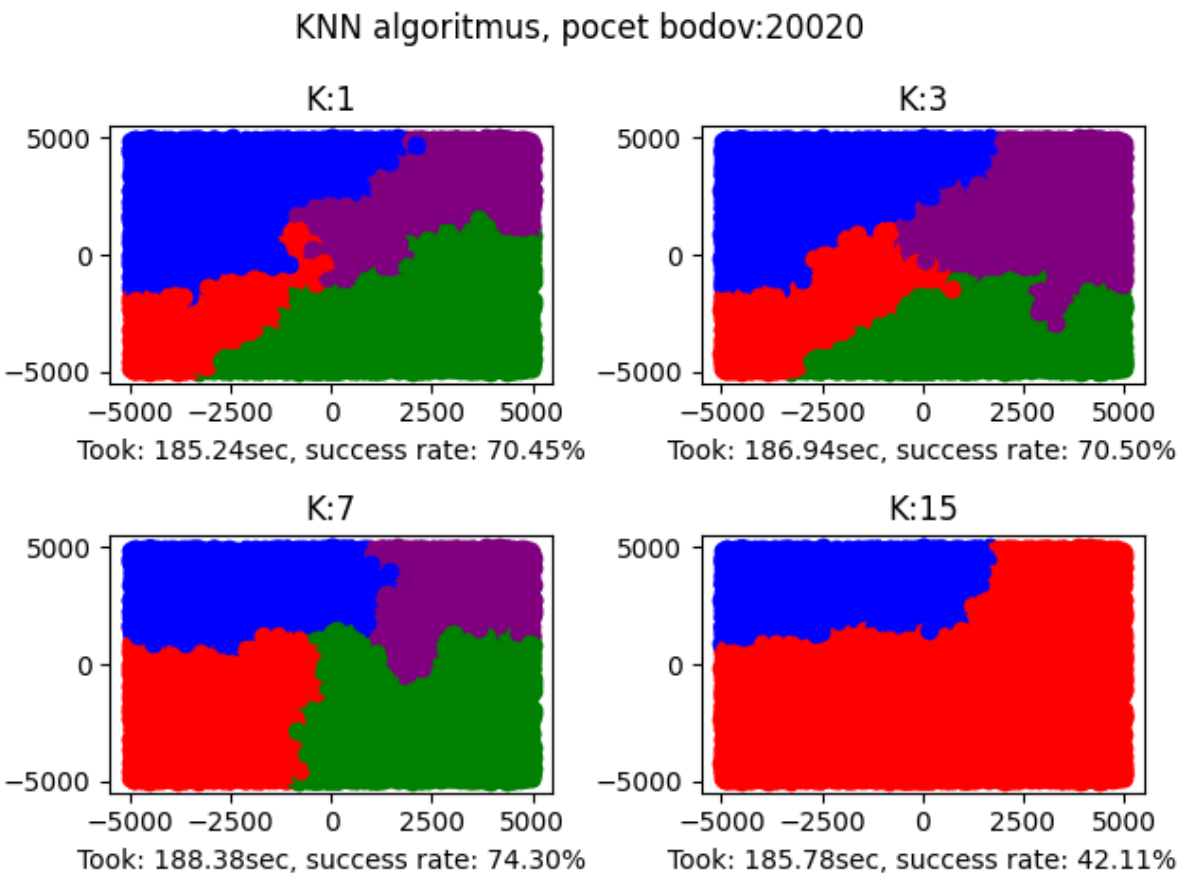
Program som testoval pre 5 000, 10 000 a 20 000 bodov. Z výsledkov je možné vidieť, že K1, K3, K7 mali podobný success rate, ale pri K15 success rate výrazne klesá. Doba behu programu pre všetky K je rovnaká, pretože počet bodov je rovnaký. Taktiež pri K15 sa často stáva, že „dominujú“ 2 alebo 3 farby, veľmi zriedkavo je pri K15 vidieť všetky 4 farby.

Jakub Taraba
AIS ID: 103159





Jakub Taraba
AIS ID: 103159



Zhodnotenie

Program bez problémov zbehne pre 20 000 bodov, trvá to okolo 12 minút. Ďalšie optimalizácie by sa mohli urobiť v tom, aby program vypočítaval iba vzdialenosti tým bodom, ktoré sú v blízkosti bodu. Teda napr. hľadať iba take body, ktoré sú v tom istom "štvorci", to by program určite urýchlilo.