Slovenská technická univerzita v Bratislave
Fakulta informatiky a informačných technológií
Ilkovičova 3, 842 19 Bratislava 4

umelá inteligencia

zadanie 4

Peter Plevko

2020/2021

Definovanie problému: Zadanie 4a – klasifikácia

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.

Opis riešenia a použitý algoritmus

1:

Na začiatku si zvolím počet bodov ktoré budem generovať pre každú farbu to znamená že keď zadám číslo 5000 vygenerujem 5000 bodov pre červenú zelenú modrú a fialovú to je dokopy 20000 bodov plus 20 počiatočných takže dokopy vygenerujem 20020 bodov.

2:

Keďže pre každé k mam použiť rovnaké body vygenerujem ich vopred. Tieto body generujem tak aby tam neboli žiadne duplicity a aby som mal 5000 správnych bodov to znamená že napríklad pre červene vytvorím 5000 bodov v rozpätí x < +5000 a y < + 5000. Následne vytvorím 5000 bodov nesprávnych to znamená v rozpätí x > +5000 a y > + 5000.

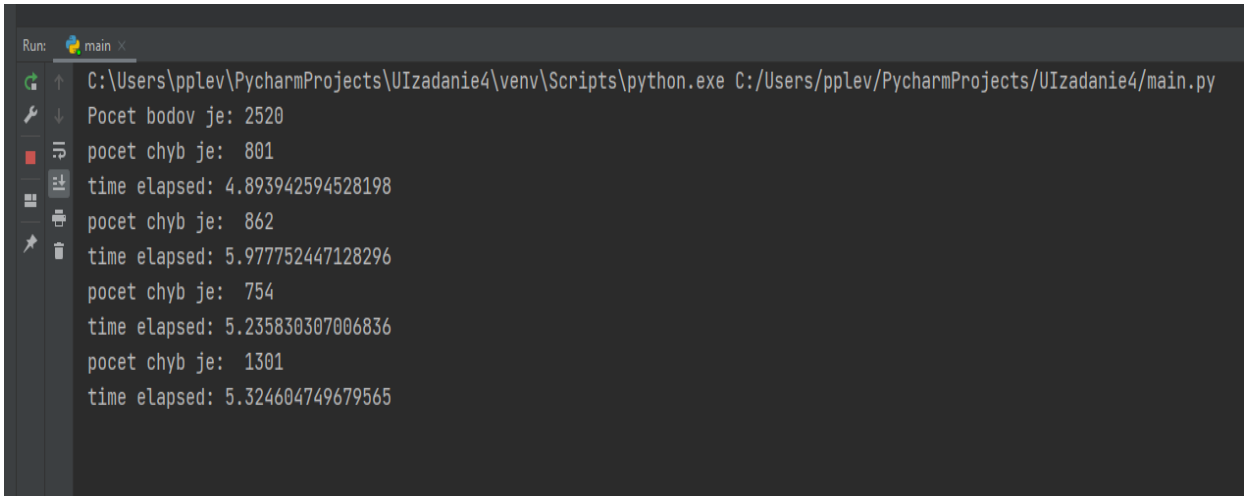
3:

Teraz som v mojom fore ktorý sa vykoná 4 krát pre k: 1, 3, 7, 15. Pre každé k sa vykoná nasledovný algoritmus. Vyberiem náhodnú farbu z farieb: red, green, blue, purple samozrejme vyberám takú ktorá nebola naposledy pridelená. Po pridelení farby sa vyskúša náhoda či vyšlo 99% a mam správnu alebo vyšlo 1% a mam nesprávnu. Podľa toho či mam správnu alebo nesprávnu vyberiem z poľa správnych alebo nesprávnych. Pripočítam color count tejto farby a zavolám moju funkciu classify s parametrami x, y, k v ktorej si vyrátam euklidovskú dĺžku od x a y súradnice môjho terajšieho bodu ktorý chcem klasifikovať so všetkými tréningovými bodmi ktoré mam. Následne sortnem od najmenšej vzdialenosti a vyberiem k najbližších a podľa nich určím farbu tohto bodu. Pridám tento bod do mojich tréningových dát a vychádzam z funkcie clasify. Porovnám farbu ktorú pridelila funkcia classify s farbou ktorú reálne ma mat tento bod ak sa farby rovnajú nerobím nič keď sa farby nerovnajú pripočítam chybu. V momente keď mam počet farieb každej farby rovný poctu ktoré mam vygenerovať končím.

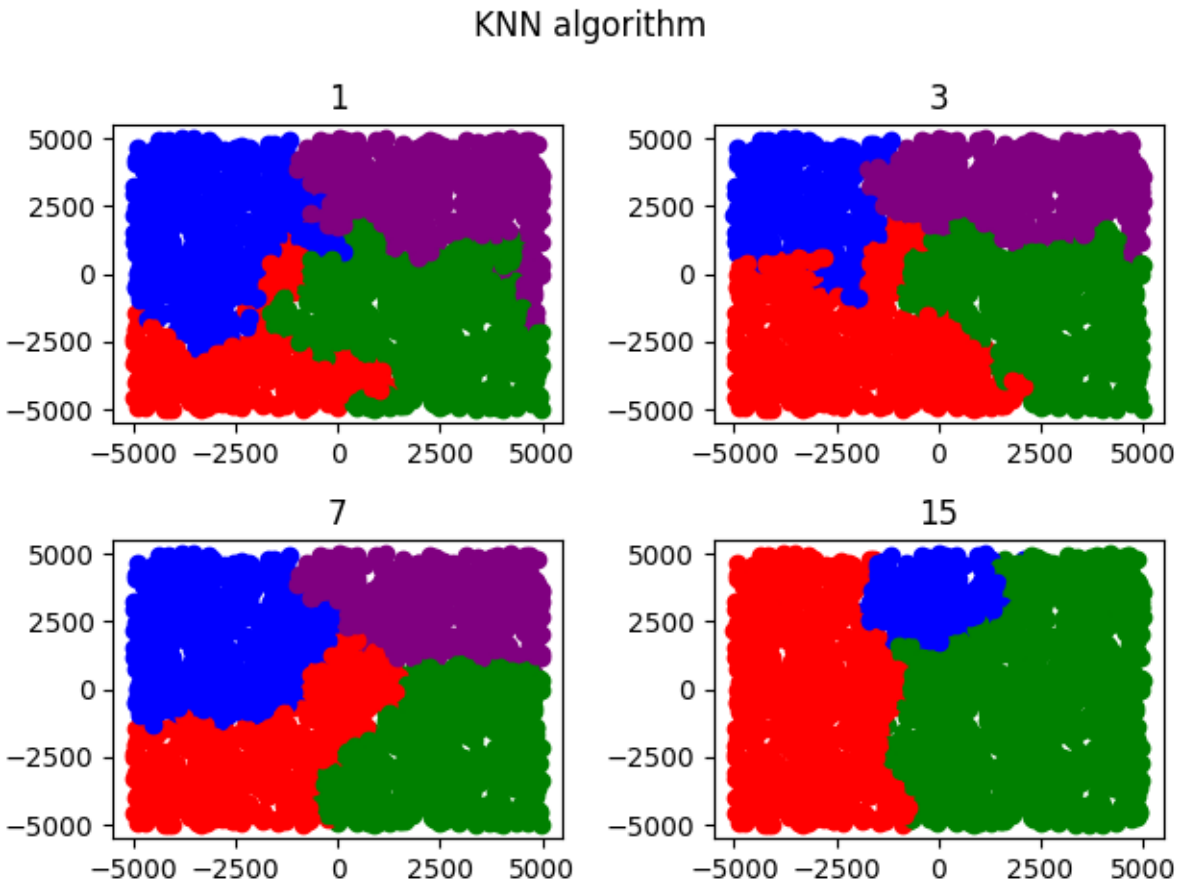
4:

V tomto momente mam už prejdene všetky body pre k a vypíšem čas koľko to trvalo, koľko chyb bolo a ostáva mi to už iba vykresliť. Vykresľujem to pomocou knižnice plt. Vykreslím 4 grafy pre každé k a končím algoritmus.

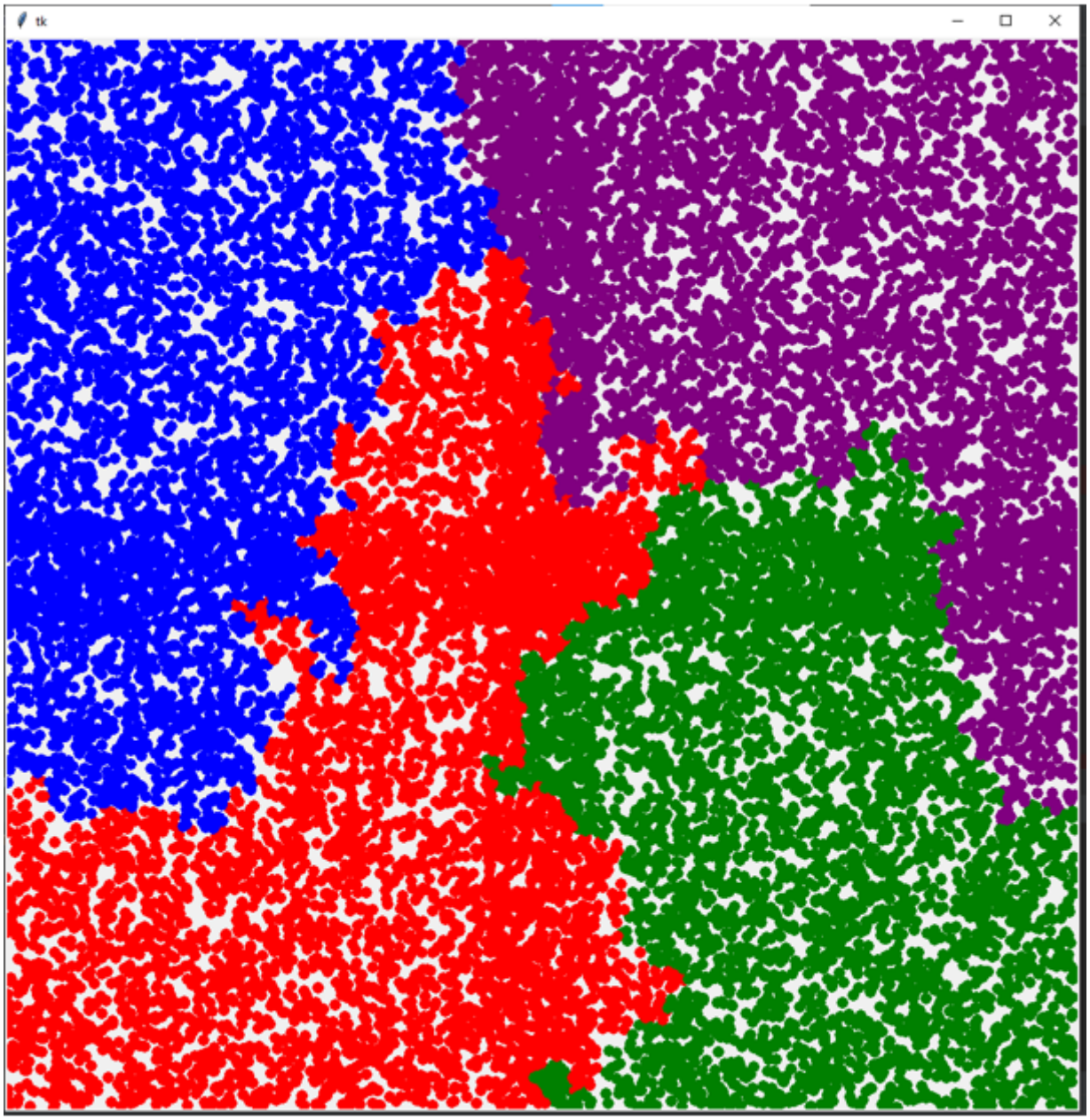
Užívateľské prostredie



Výsledný graf mi zapíše do súboru knn ktorý vyzerá nasledovne:



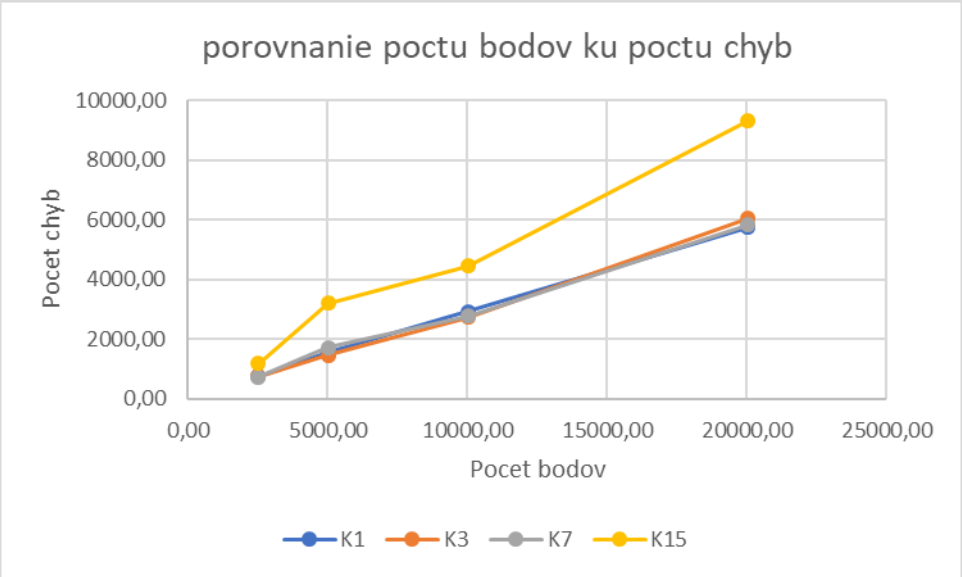
Predchádzajúci výpis môjho programu vyzeral nasledovne ale nevyfarboval plochu celu preto som ho upravil. Aby to vyzeralo lepsie



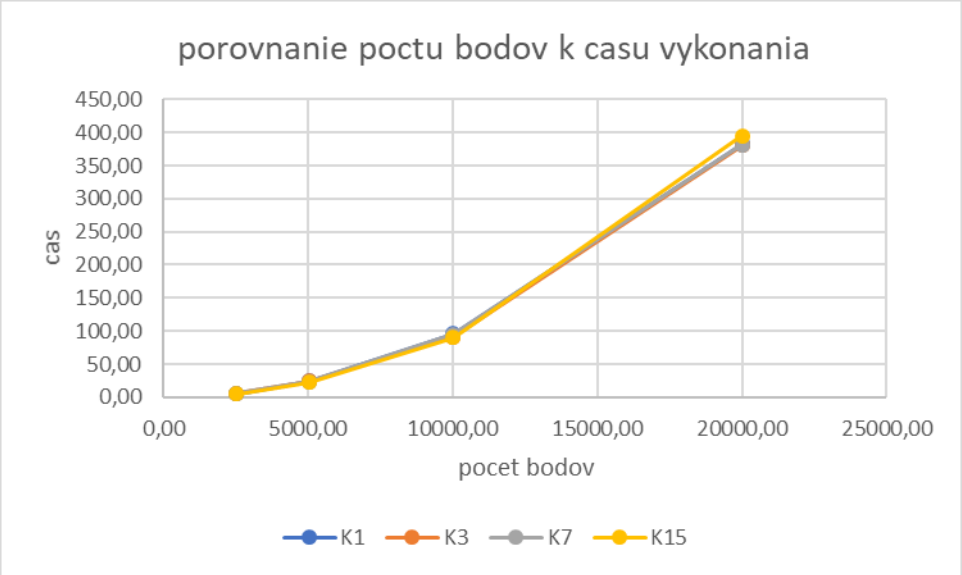
Testovanie

Vykonal som všetky testy 3 krát a spriemeroval ich a z výsledkov vznikli tieto grafy. Testy prebehli pre počet bodov: 20010, 10020, 5020, 2520.

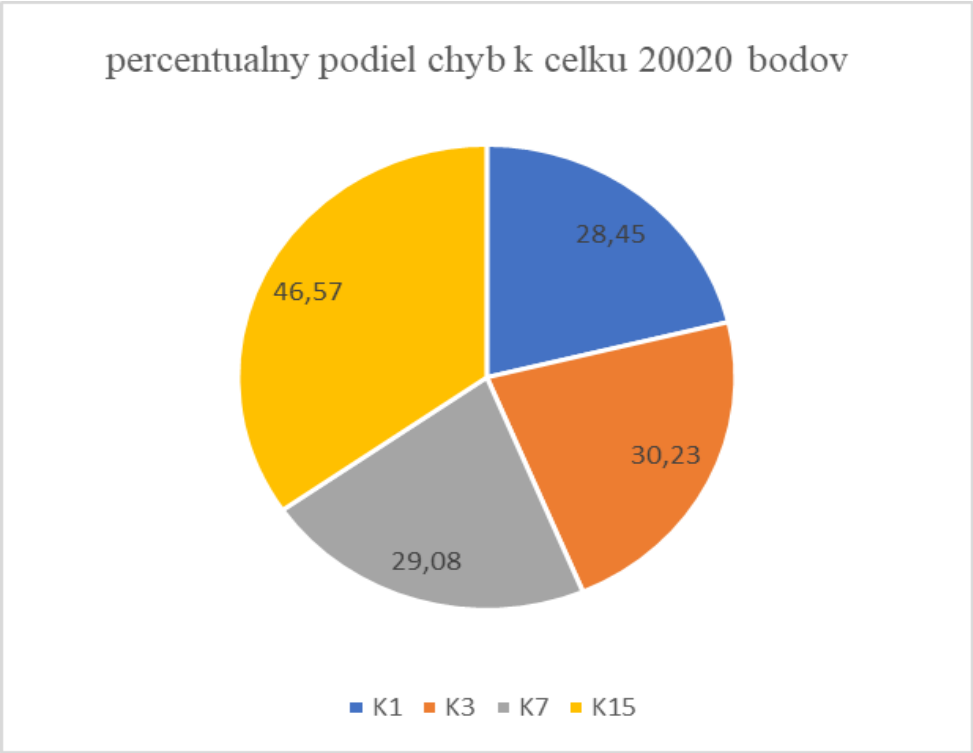
Body pridávam do datasetu.

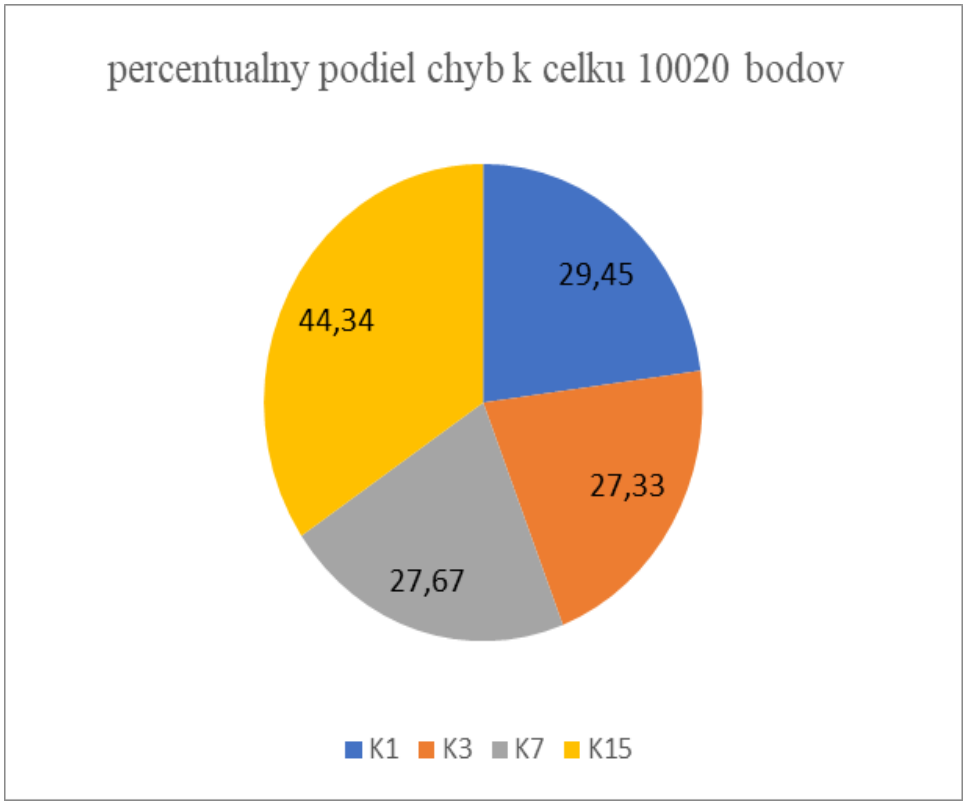


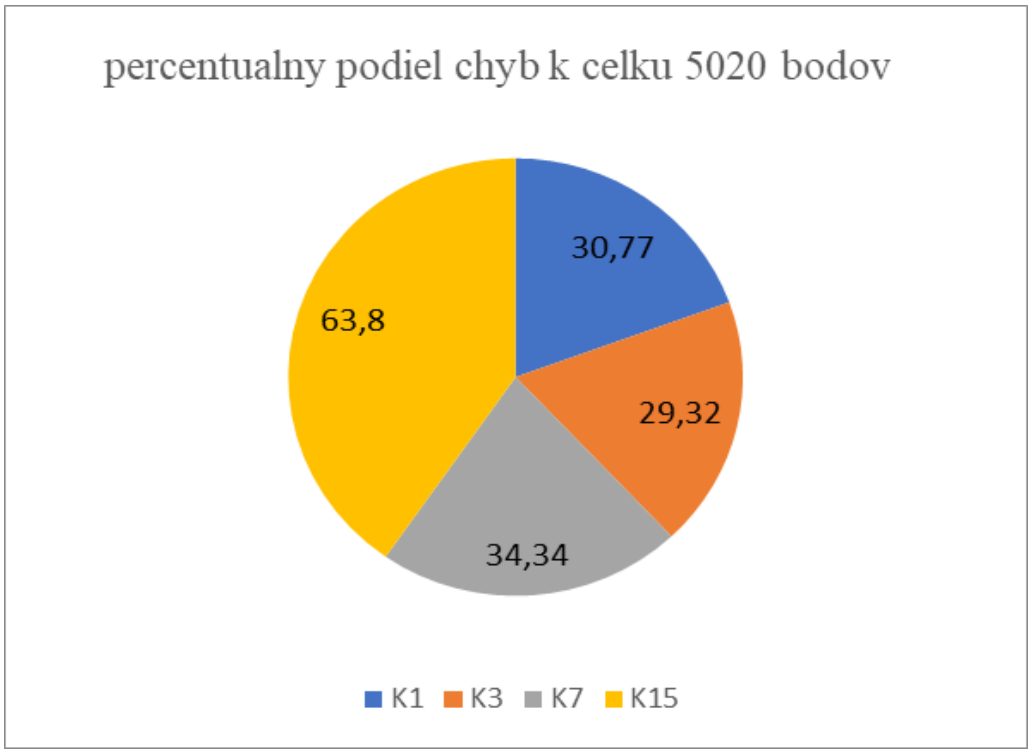
Na tomto grafe vidíme že so stúpajúcim počtom bodov stúpa aj počet chyb najväčší počet chyb je pre k15 čo je aj logické pretože v tomto prípade na začiatku robím k15 a môj cely dataset ma 20 bodov to znamená že jedna farba na obrázku ani nebude.

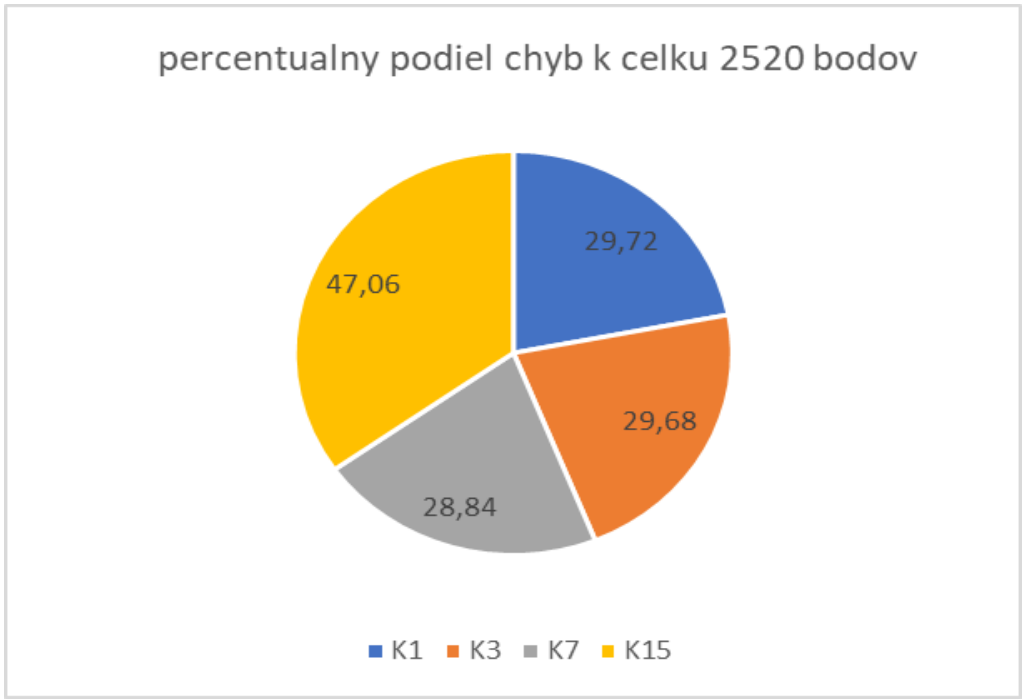


Na tomto grafe vidíme že čas sa zvyšuje s väčším počtom bodov čo je aj logické. Medzi k1 a k3 a k7 a k15 nie je badateľný rozdiel.

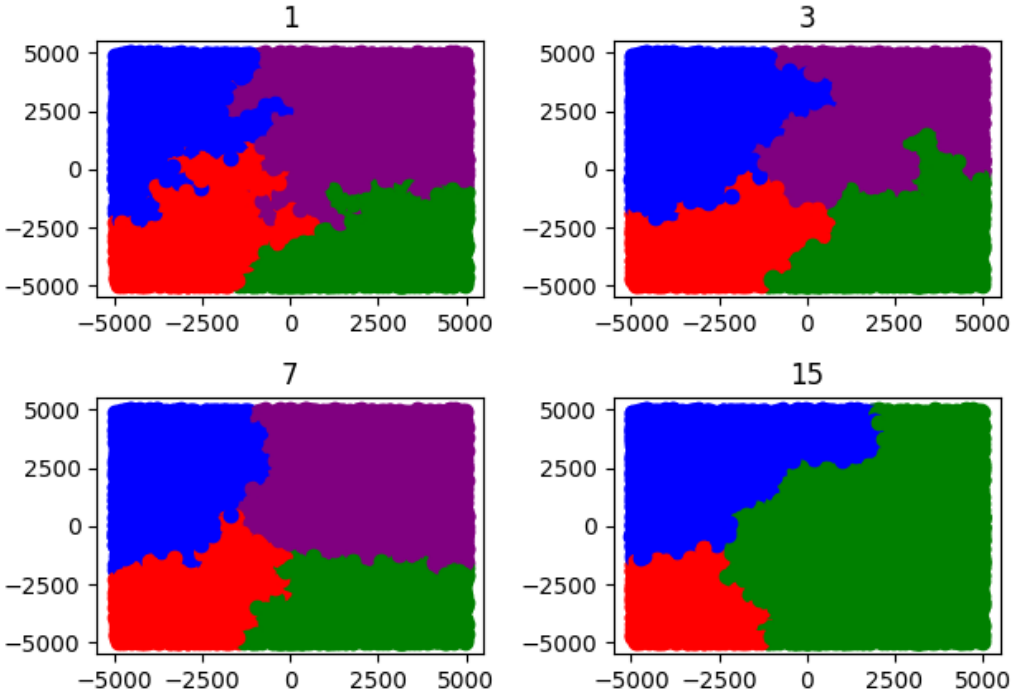

percentualny podiel chyb k celku 20020 bodov
28,45
30,23
29,08
46,57
K1
K3
K7
K15


percentualny podiel chyb k celku 10020 bodov
29,45
27,33
27,67
44,34
K1
K3
K7
K15

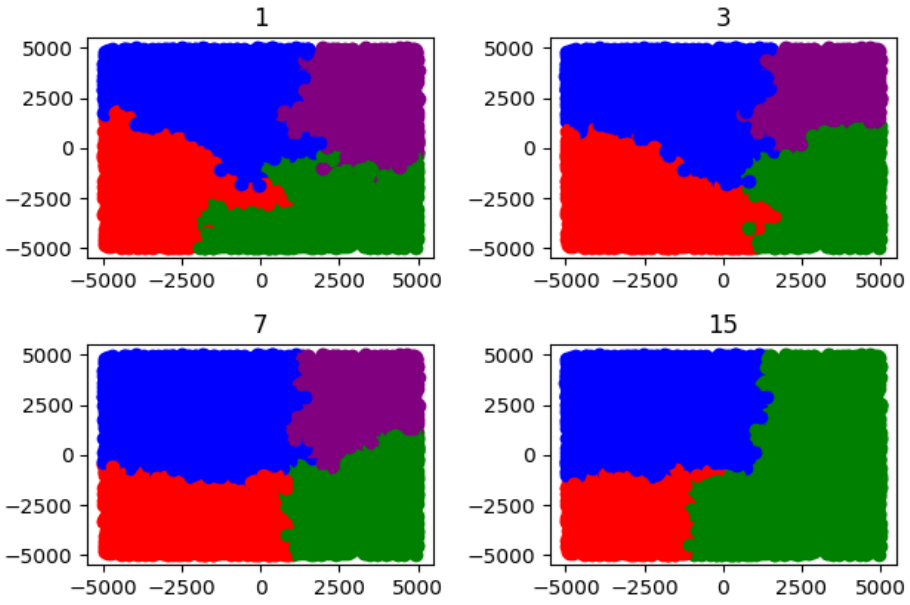

percentualny podiel chyb k celku 5020 bodov
30,77
29,32
34,34
63,8
K1
K3
K7
K15


percentualny podiel chyb k celku 2520 bodov
29,72
29,68
28,84
47,06
K1
K3
K7
K15

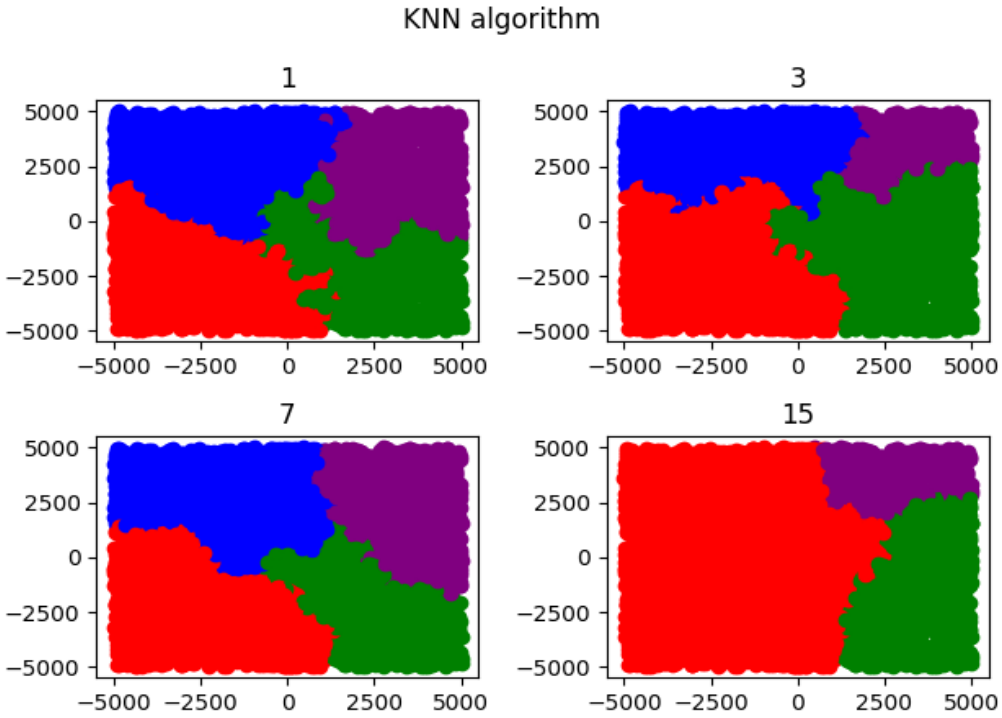
20020


KNN algorithm
1
3
7
15
5000
2500
0
−2500
−5000
−5000 −2500 0 2500 5000

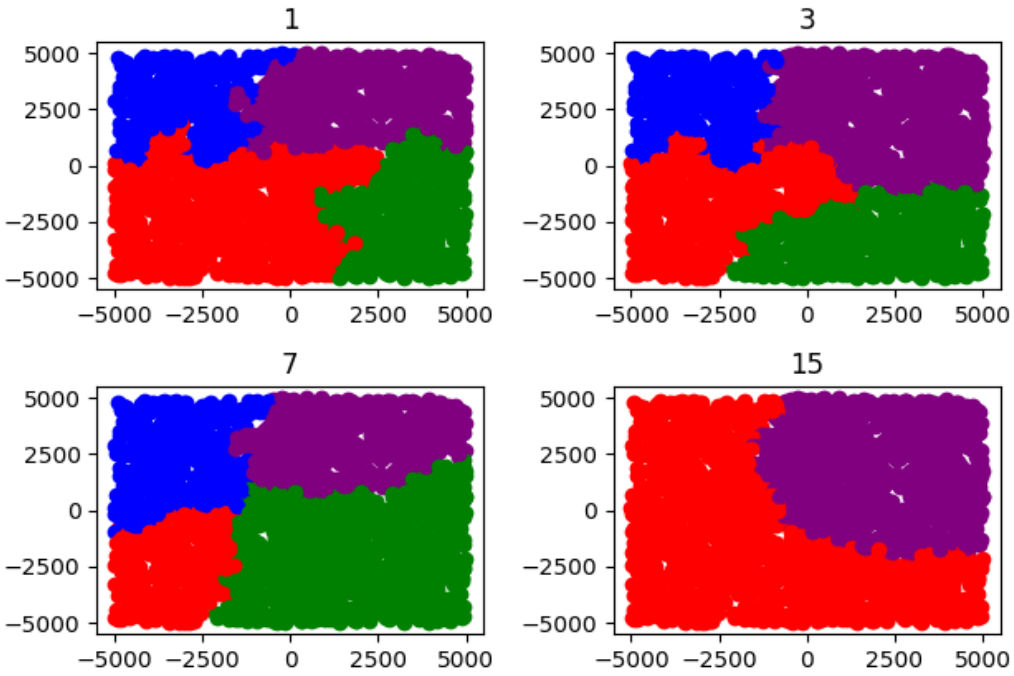
10020


KNN algorithm
1
3
7
15
5000
2500
0
−2500
−5000
−5000 −2500 0 2500 5000

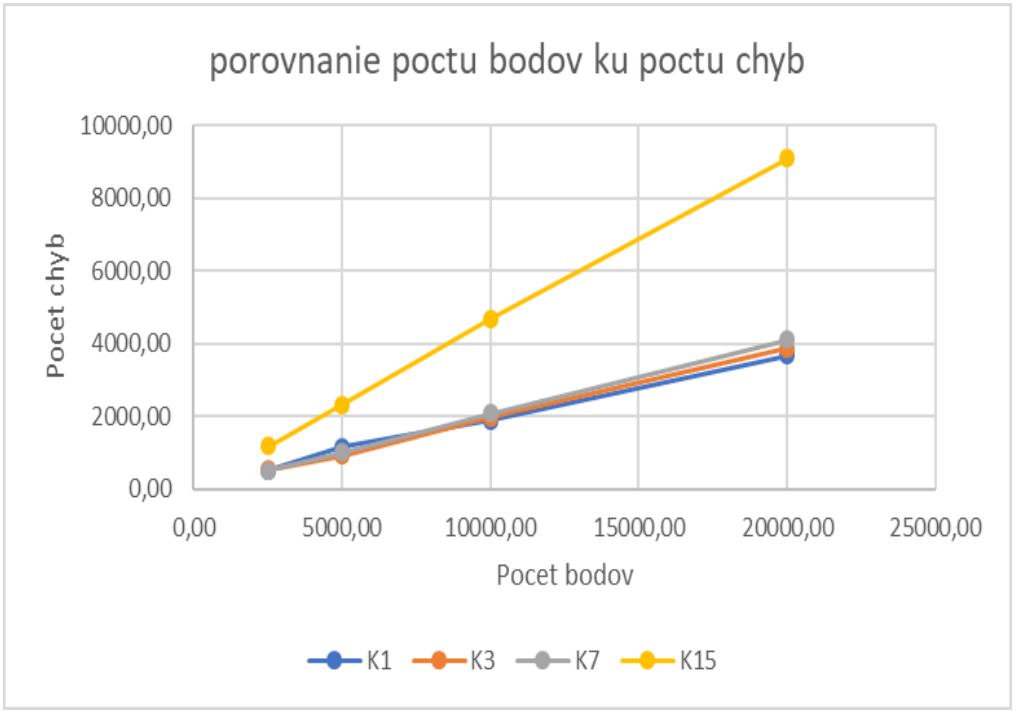
5020 bodov



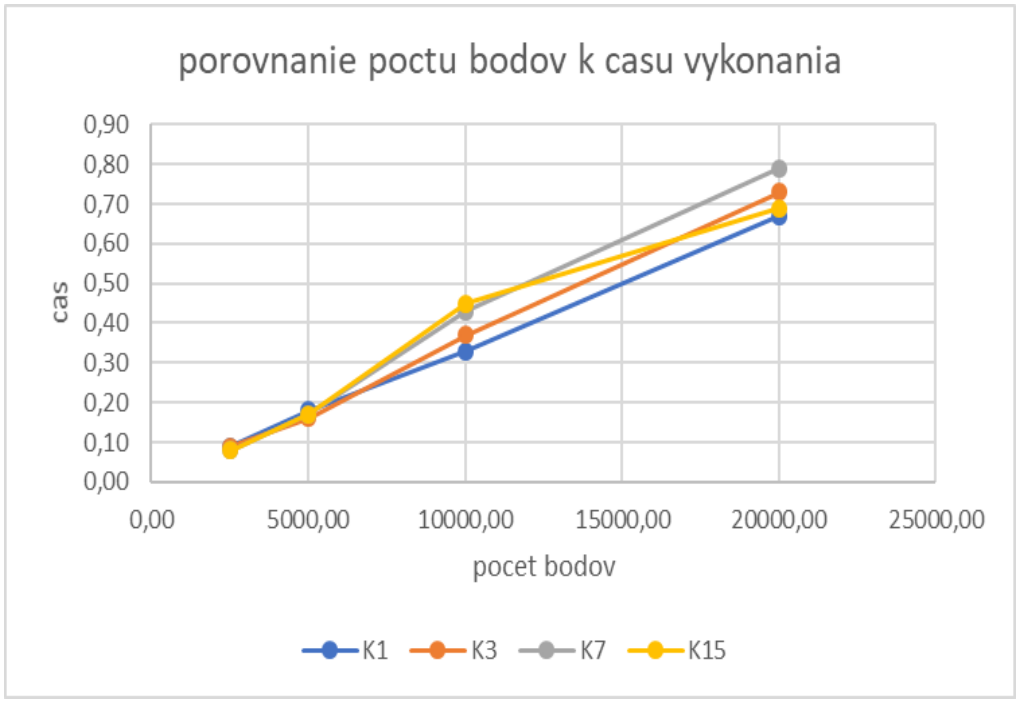
2520



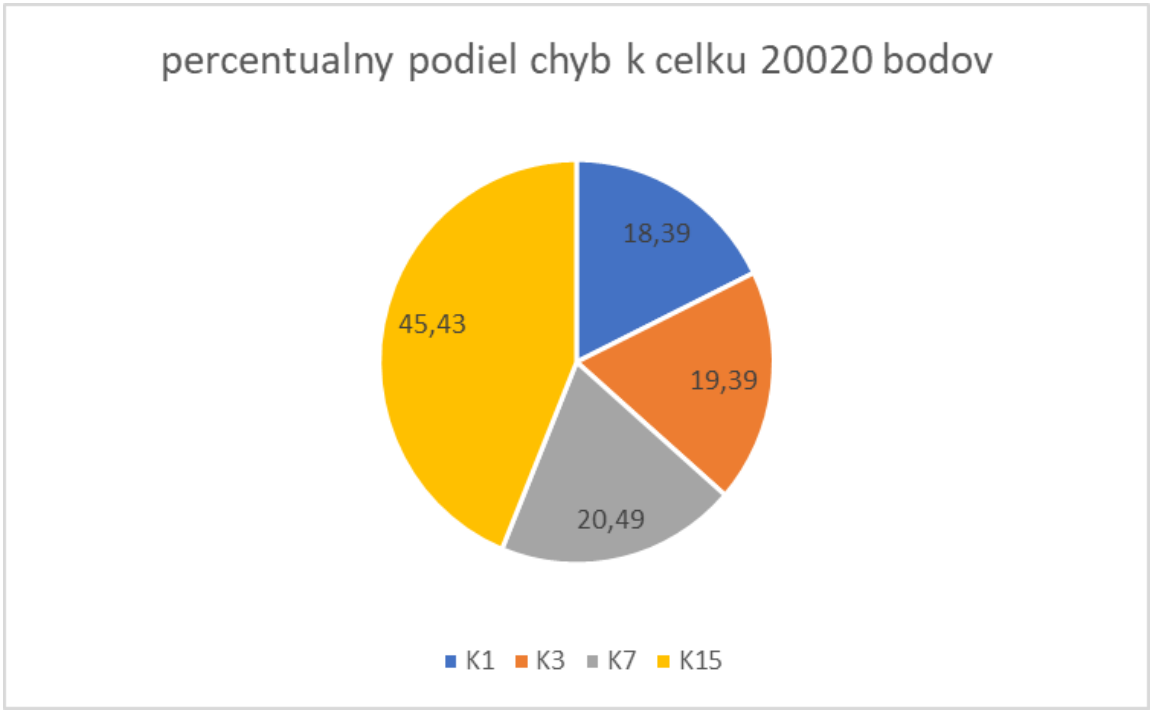
Body nepridávam do datasetu

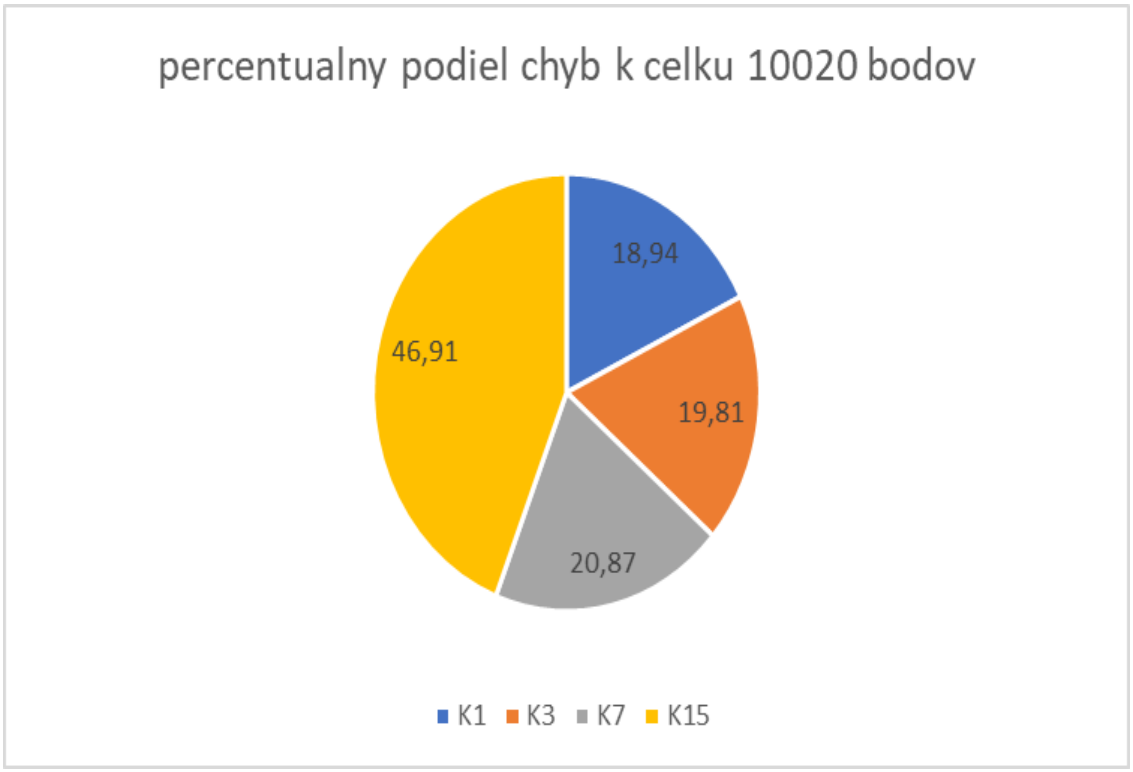


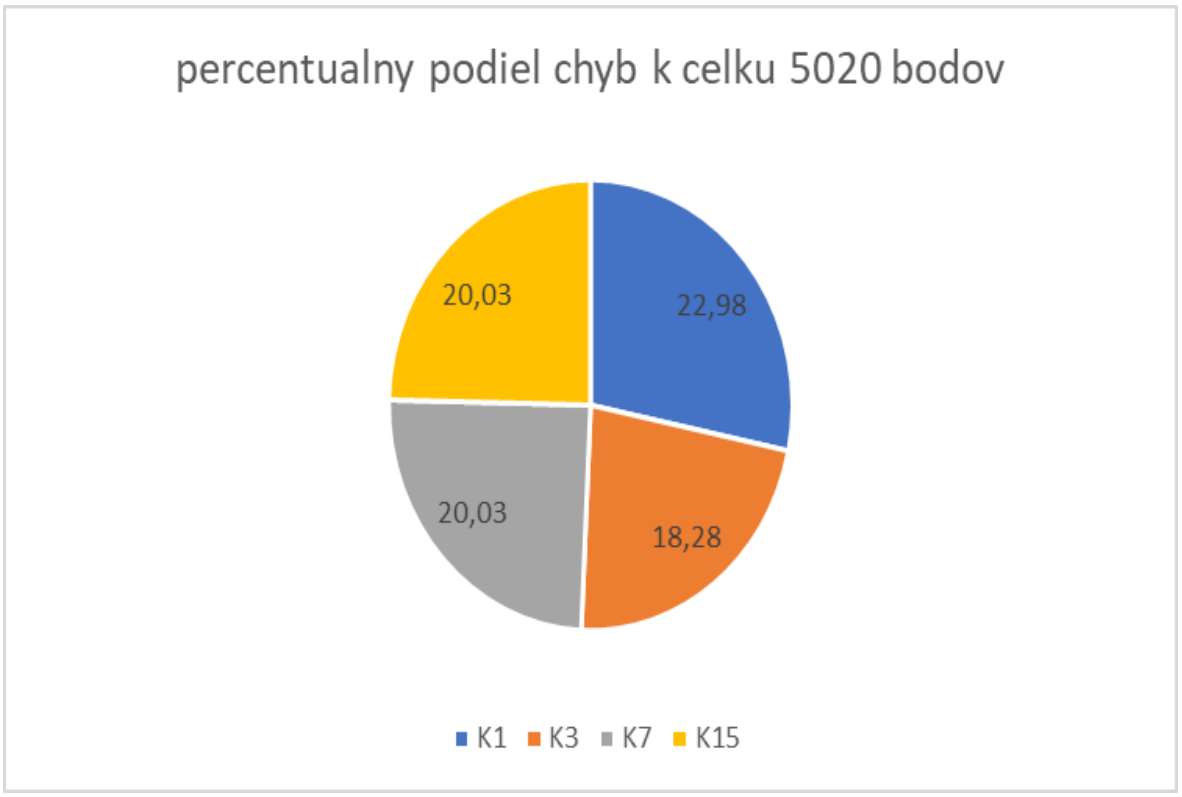
Vidim ze pocet chyb oproti pridavaniu do datasetu je nizsi u vsetkych k okrem K15 tam je priblizne rovnaky. S toho viem odvodit ze k15 je nezmyselna velkost knn.

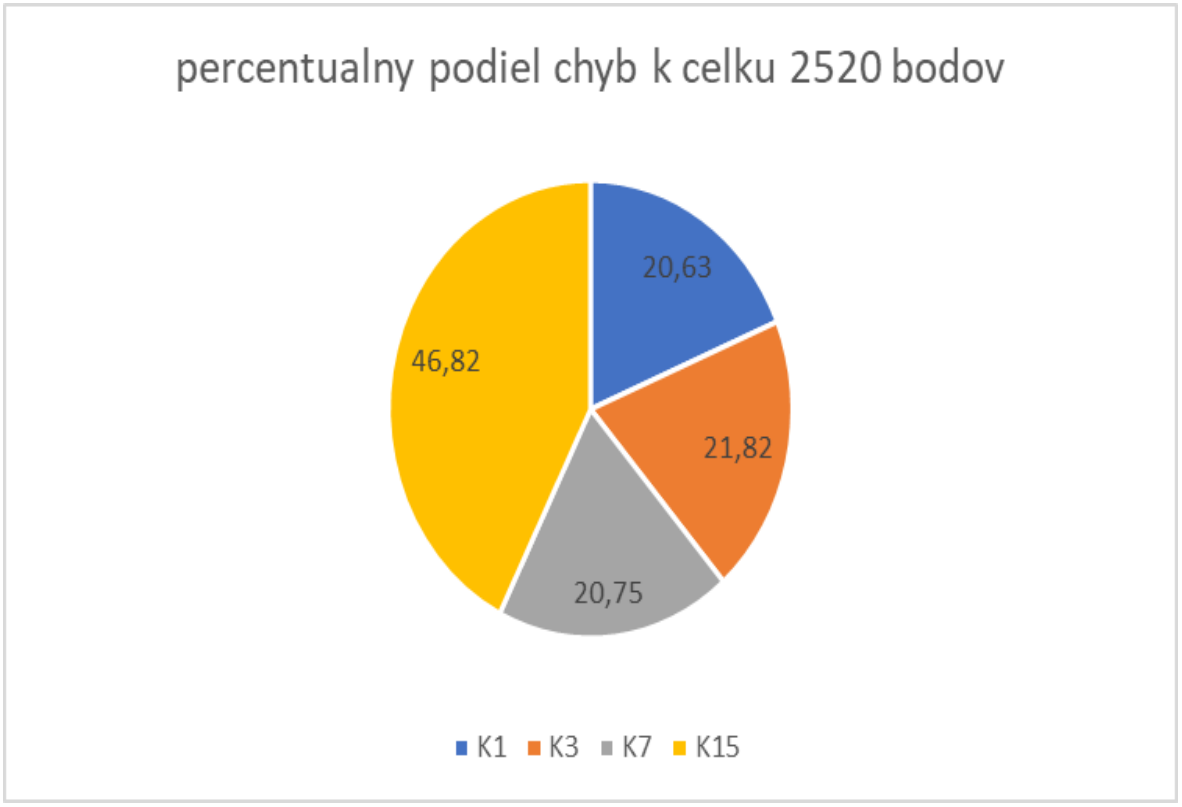


Vidim ze cas vykonania je neporovnatelne rychlejsi a ked si zoberiem do uvahu ze pocet chyb je tiez mensi logicky mi s toho vypliva ze je lepsie mat 20 treningovych bodov a ostatne urcovat z nich a nepridavat ich do datasetu.

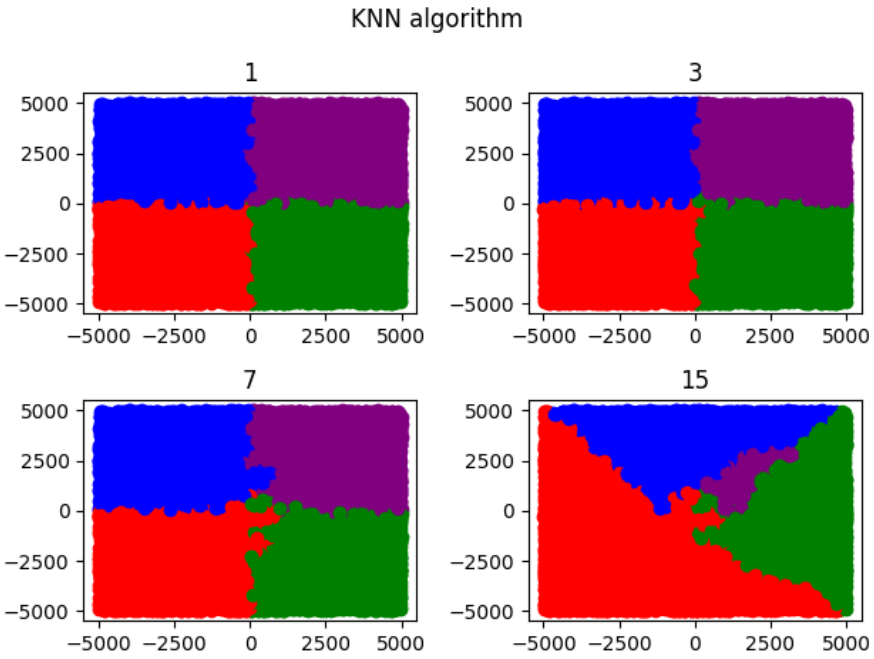

percentualny podiel chyb k celku 20020 bodov
18,39
19,39
20,49
45,43
K1
K3
K7
K15


percentualny podiel chyb k celku 10020 bodov
18,94
19,81
20,87
46,91
K1
K3
K7
K15

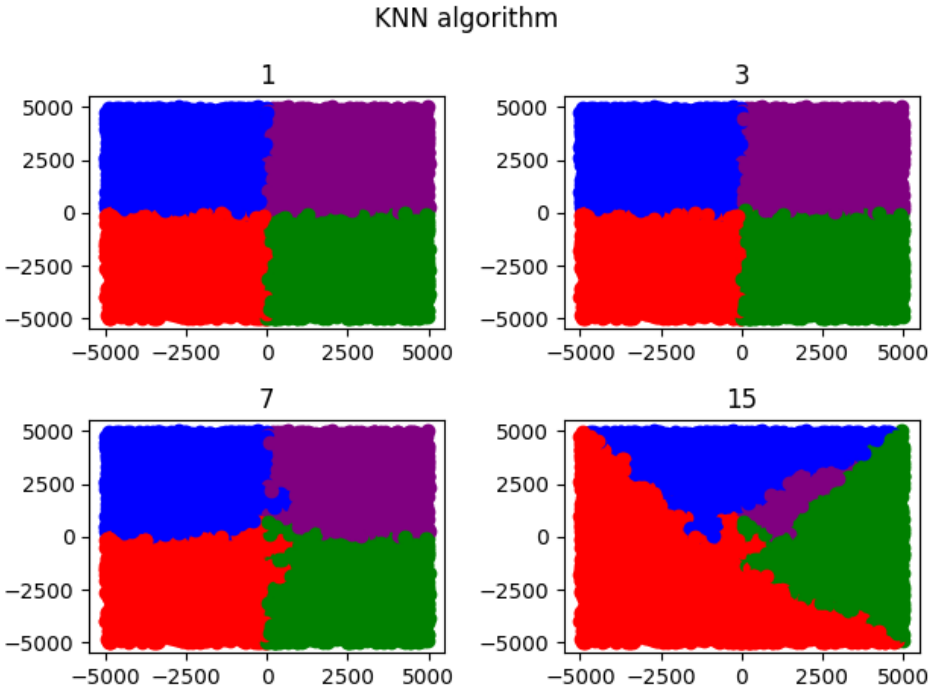

percentualny podiel chyb k celku 5020 bodov
20,03
22,98
20,03
18,28
K1
K3
K7
K15


percentualny podiel chyb k celku 2520 bodov
20,63
46,82
21,82
20,75
K1
K3
K7
K15

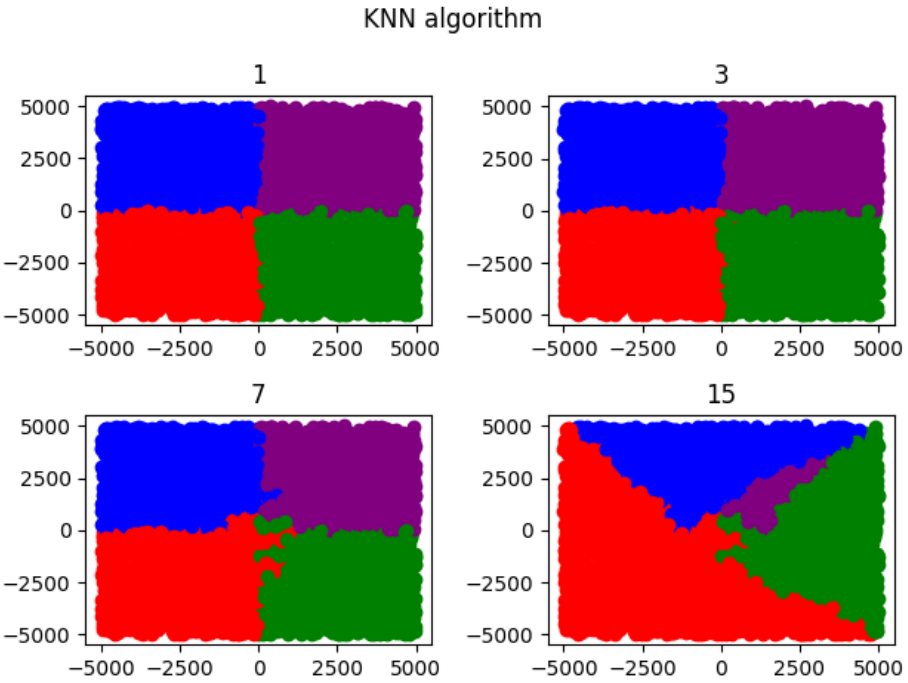
20020 bodov



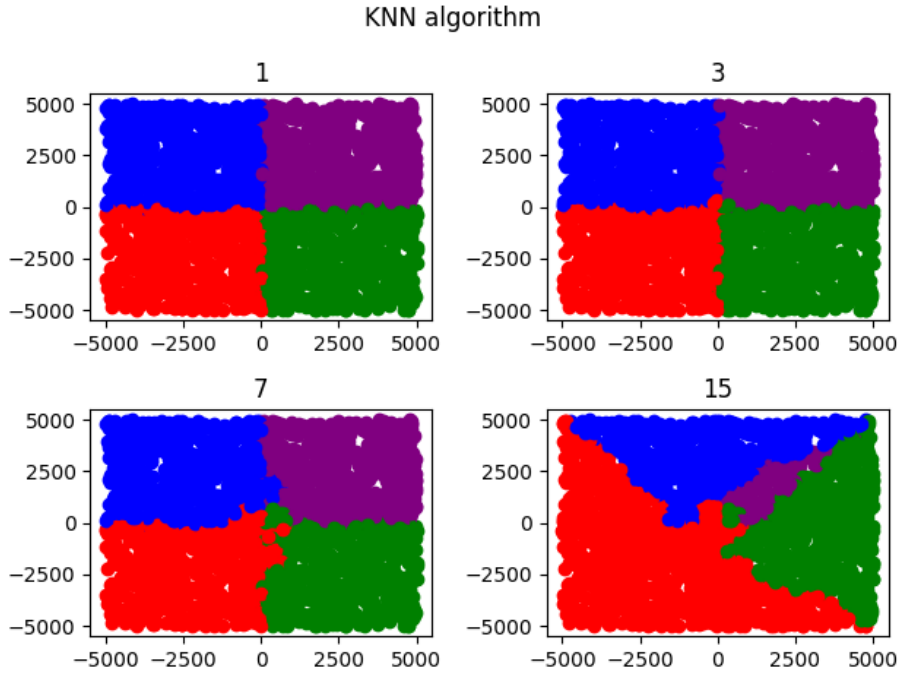
10020 bodov



5020 bodov



2520 bodov



Zhodnotenie

Tento projekt by som nazval ako úspešný riešenie problému je dostatočne rýchle to znamená že program mi zbehne pre najväčší vstup čo je 20020 bodov do 400 sekúnd. Podľa mojich doterajších testov je aj správne. Testovanie môžem robiť jedine vizuálne a porovnávaním si výstupov s mojimi kolegami. Podľa doterajších porovnaní by mal byt môj algoritmus správny.

Závislosť na programovacom prostredí:

Keďže python je vysokoúrovňoví programovací jazyk tak je jasne že tento program by rýchlejšie zbehol v tých nižších napríklad c ale v nom by som si musel ostatne veci implementovať sám.