Zadanie 4a – klasifikácia

❖ Riešený problém

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, Y)* a vždy má unikátne súradnice. 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). 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*.

❖ Opis riešenia

▪ Implementáčne prostredie

Svoje zadanie som riešila pomocou programovacieho jazyka **Python 3.9.0** v prostredí PyCharm 2020.2.3. Naviac pre efektívne riešenie som použila ďalšie väčšinou defaultne knižnice:
math a copy (pre rôzne matematické operácie)
random (pre generovanie bodov na začiatku)
time (na vyrátanie trvalosti programu a aj efektivity)
matplotlib (na kreslenie výsledkov klasifikácie) https://matplotlib.org/

▪ Použitý algoritmus

Na vyriešenie problému som použila **KNN-algoritmus**, ktorý sa spúšťa vo funkcií **compareMap**(početElementovVTriede, mapa, trieda1, trieda2, trieda3, trieda4, k). Hlavným princípom jeho fungovania je to, že my musíme vyhľadať pre každý nový vložený bod K najbližších susedov a prideliť nášmu bodu tu triedu, ktorá je pridelená väčšine týchto susedov (teda klasifikovať ho podľa väčšiny), za čo je zodpovedná funkcia **classify**(x, y, k). V našom prípade triedy sú farbičky, diaľku medzi dvomi bodmi rátame pomocou Euklidovej vzdialenosti, a K najbližších susedov budeme ukladať v pole susedov (iné vzdialenosti naukladáme). Čiže najbližších K susedov ja hľadám pomocou skúšania diaľky medzi všetkými bodmi. Ak vyrátaná diaľka k nejakému bodu je kratšia, ako najväčšia v pole susedov, tak tie body vymením a pridám momentálny do poľa susedov.
Na začiatku programu používateľ ma možnosť zvoliť si čo sa bude vykonávať: **testovanie** na veľkom množstve údajov alebo **spustenie algoritmu s vlastnými parametrami**.

▪ Používateľské rozhranie

Hlavnú informáciu o výsledkoch vypisujem do **konzoly**. Taktiež ju ukladám do súboru **results_test.txt** aby sa v budúcnosti jednoduchšie načítavali získané výsledky a dali sa do tabuliek/grafov a porovnávali sa.
Vytvorenú mapu ja hneď vykresľujem a ukladám do toho istého priečinku, kde sa nachádza aj hlavny program. Po každej uplne vykonanej klasifikácii sa dá pozrieť na vytvorený graf.

❖ Otestovanie programu

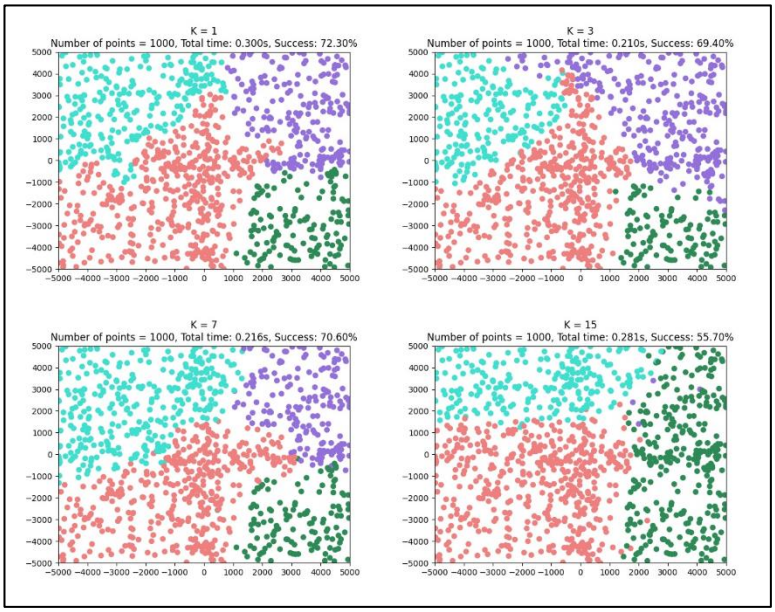
▪ Možnosti nastavenia parametrov

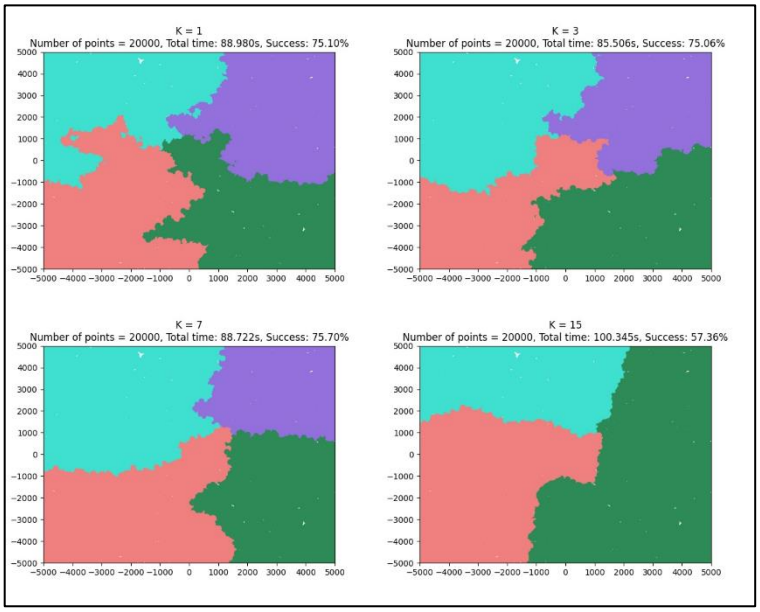
Svoj program overujem na veľkom teste, ktorý zahŕňa **rôzne nastavenia počtu generovaných** bodov (1tis, 20tis, 40tis a 60tis) pre **všetky možnosti K** (1, 3, 7 alebo 15). Pre každú veľkosť testu generujem rovnaké body, ktoré v budúcnosti budú vložené v mapu a klasifikované. Žiadny z nich sa neopakuje a ma svoje unikátne súradnice (to som zabezpečila tým, že zo začiatku vygenerujem všetky body spolu a pozriem si, ci sa nezopakujú, a potom rozdelím ich podľa farbičiek naspäť). Všetky výsledky

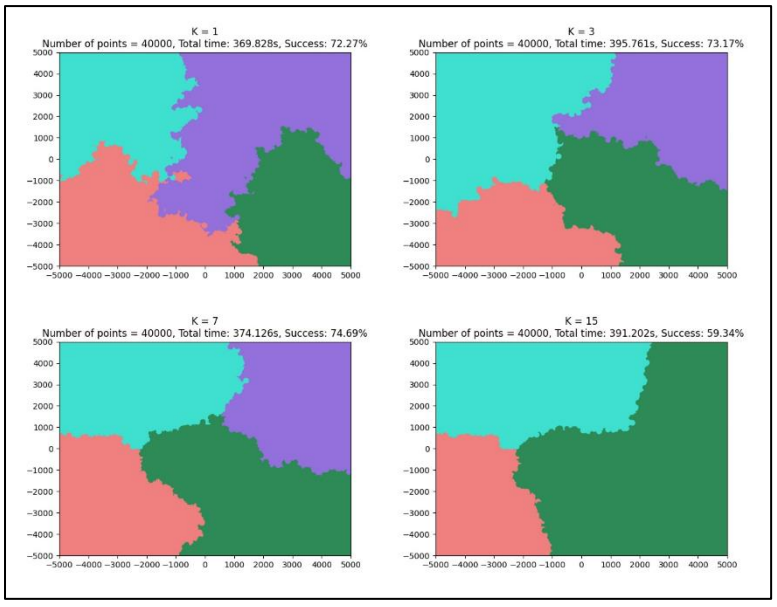
zapisujem (ako to bolo uvedené pred tým) do konzoly, results_test.txt a aj ukladám obrázky v adresári.

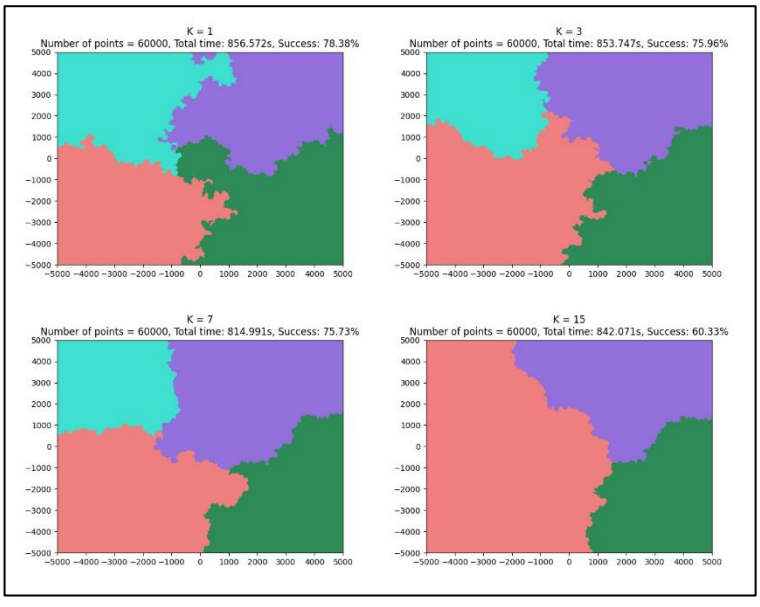
- **Porovnanie dosahovaných výsledkov pre viacero nastavení parametrov**

Nižšie sú uvedené grafy s rôznymi možnosťami parametrov ako som spomínala predtým.









Bohužiaľ, testy boli vykonané iba jeden krát, preto uvedené dáta nie sú veľmi objektívne. Avšak, snažila som sa spustiť test na rôznych počtoch bodov, čo dokazuje moje očakávania, že:

- čím menšie je K, tým hranice oblastí sú viac nestabilné (majú väčšie výbežky)
- v prípadoch, že K sa rovná 1, 3 alebo 7 úspešnosť je približne rovnaká (v mojom prípade bolo víťazné K = 1 a K = 7)
- pre hodnotu K = 15 hranice tried sú viac hladké a rovné (teda nemajú také prepady ako pri K = 1), avšak často sa stáva, že nejaká trieda (alebo triedy) úplne zanikne. To je spôsobené hlavne tým, že pri veľkom K sa zachytáva viac bodov z ostatných tried.
- z pozorovania ja som zistila, že najvhodnejšia by bola hodnota K medzi 3 a 7

Priemerný čas trvania klasifikácie pre 1000 bodov je 0,25 sekundy, pre 20 tis. bodov 90,89 sekúnd, pre 40 tis. bodov 382,73 sekúnd a pre 60 tis. bodov 841,85 sekúnd. Čas nerastie lineárne, ale skôr exponenciálne.

Number of inserted points	K	Duration (in sec)	Success
1000	1	0.300	72.30 %
	3	0.210	69.40 %
	7	0.216	70.60 %
	15	0.281	55.70 %
20000	1	88.980	75.10 %
	3	85.506	75.06 %
	7	88.722	75.70 %
	15	100.345	57.36 %
40000	1	369.828	72.27 %
	3	395.761	73.17 %
	7	374.126	74.69 %
	15	391.202	59.34 %
60000	1	856.572	78.38 %
	3	853.747	75.96 %
	7	814.991	75.73 %
	15	842.071	60.33 %

❖ Zhodnotenie

Celkovo môj program funguje korektne a správne, a preto považujem celý projekt za úspešný. Zdalo by sa aj pridať nejaké vylepšenie (napríklad prehľadávať susedov nejakou šachovnicou alebo niečo podobné), ale aj bez toho môj kód dokáže pomerne rýchlo nájsť kompletné riešenie. Naviac spúšťala som ho aj na väčších počtoch bodov (ako som to uviedla vyššie) a vždy to stíhalo klasifikovať ich do nejakého adekvátneho časového limitu (napríklad najväčší test na 60tis trval menej ako 1hod). Inak ešte by sa zdalo zopakovať experimenty viackrát, aby boli hodnoty menej náhodné, a teda výsledky by boli viac presné.