Západočeská univerzita v Plzni
Fakulta aplikovaných věd
Katedra kybernetiky

Semestrální práce

Základy strojového učení a rozpoznávání

KKY/ZSUR

Yauheni Petrachenka
28. ledna 2025

1 Zadání

V mailu jsou potřebná data k řešení semestrální práce. Jedná se o množinu dvojdimenzionálních vektorů. Vaším úkolem je

1) automaticky určit počet tříd

a) metodou shlukové hladiny (automaticky naleznìte hladinu h)

b) metodou řetězové mapy (zkuste několik různých počátků)

c) metodou MAXIMIN

a metody vzájemnì porovnat

2) metodou k-means rozdělit všechna data do zjištěného počtu tříd - porovnat nerovnoměrné binární dělení s přímým dělením do cílového počtu tříd

3) na výsledné rozdělení dat do jednotlivých tříd z bodu 2 vyzkoušet iterativní optimalizaci

4) na základě informací od učitele (informace o zařazení trénovacích dat do jednotlivých tříd ω_i z bodu 2 popřípadě informace z bodu 3) natrénovat

a) Bayesův klasifikátor - tady nepředpokládám explicitně řešení té hranice (kuželoseèky) stačí odhadnout parametry jednosložkového normálního rozložení a nějakým dostatečně jemným rastrem ohodnotit body v prostoru (kde se vyskytují trénovací data), kam který bod má nejvìtší pravděpodobnost.

b) vektorovou kvantizaci - kde velikost kódové knihy bude rovna počtu zjištìných tříd. Podobně jako v pøedchozím bodě zakreslete pomocí rastru body v prostoru (trénovacích dat) odpovídající jednotlivým vzorům

c) klasifikátor podle nejbližšího souseda - vyzkoušejte klasifikaci podle jednoho a podle dvou nejbližších sousedů a podobně jako v předchozím bodě zakreslete pomocí rastru body v prostoru (trénovacích dat), které klasifikujeme do jednotlivých tříd

d) klasifikátor s lineárními diskriminačními funkcemi - porovnejte potřebný počet iterací při použití Rosenblattova alg., a upravené metody konstantních přírůstků pro několik zvolených konstant učení. Podobně jako v předchozím bodě zakreslete pomocí rastru body v prostoru (trénovacích dat), které klasifikujeme do jednotlivých tříd.

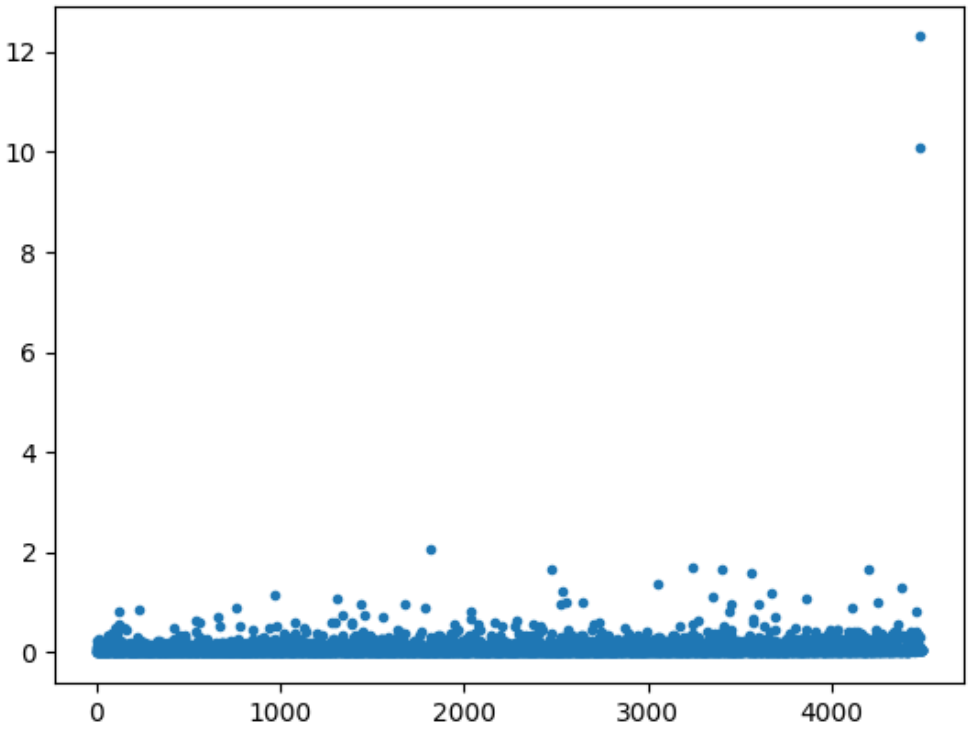
5) na základě informací od učitele (informace o zařazení trénovacích dat do jednotlivých tříd ω_i) natrénujte jednoduchou neuronovou síť pro úlohu klasifikace. Vyzkoušejte několik topologií sítě a různé způsoby trénování (SGD $\times$ batch GD). Úlohu můžete řešit v libovolném jazyce. Ale pokud zvolíte nestandardní řešení (mimo C++, MATLAB, Python), tak mě musíte před odevzdáním informovat, abych se na vás připravil. Výsledky a postup prezentujte krátkou zprávou - záměrně jsem zvolil dimenzi 2, aby se řešení úloh dalo pěkně zobrazit (očekávám spoustu obrázkù)

2 Automatické určování počtu tříd

2.1 Metoda shlukové hladiny

Při opakování algoritmu si můžete všimnout, že ke skoku dochází na samém konci algoritmu. Také kvůli sjednocení shluků vždy dostaneme o jeden skok méně, než je počet tříd (počet tříd = počet skoků + 1). Pro zjištění počtu tříd v kódu byla použita

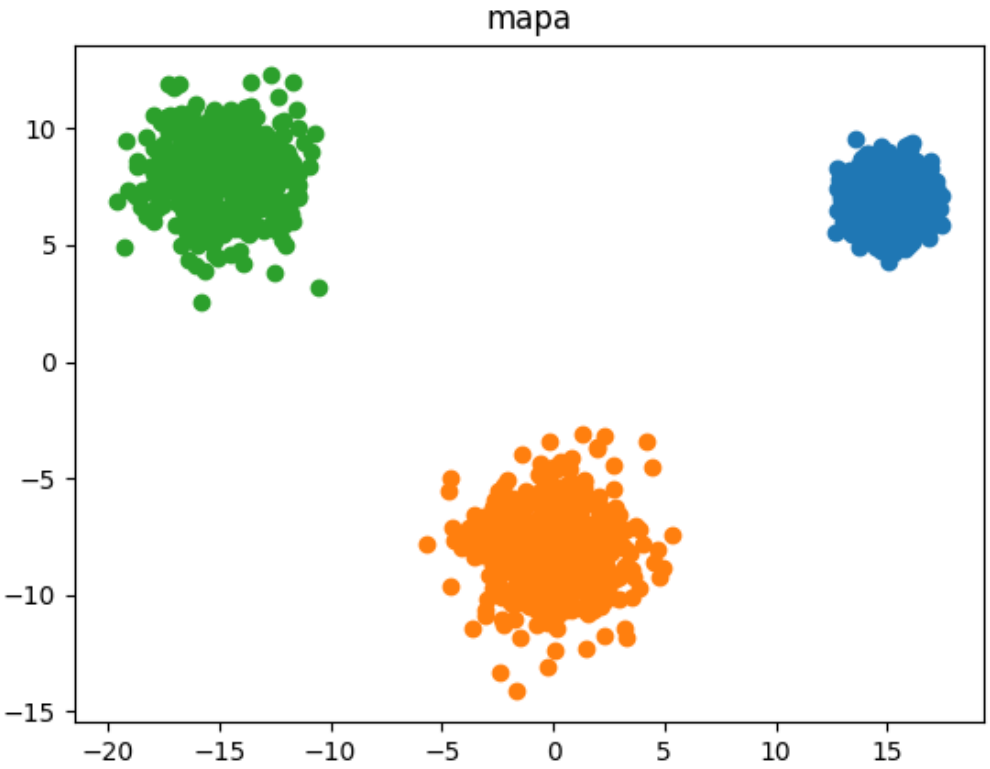
následující podmínka: pokud je skok větší než 70% z maximálního skoku, zvyšeme počet tříd o 1. Nevýhodou této metody je rychlost, metoda je pomalá.



Obrázek 1: Metoda shlukové hladiny

2.2 Metoda řetězové mapy

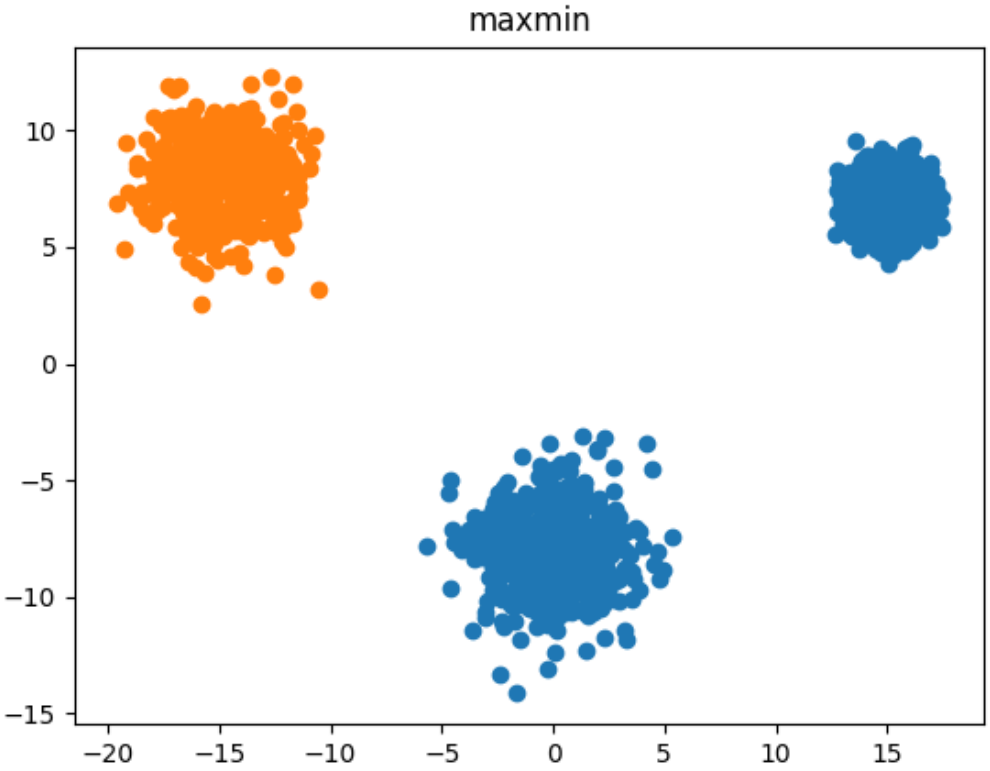
Po několika spuštěních metody si můžeeme všimnout, že výsledek nezávisí na počátečním bodě. Pro určení počtu tříd byla použita tato podmínka: pokud je vzdálenost větší než průměr vzdálenosti násobený 175, zvýšíme počet tříd o 1. První nevýhodou této metody je ruční výběr konstanty pro zvýšení počtu tříd (v našem případě 175). Druhou nevýhodou metody je její rychlost. Metoda je pomalá. Také jsem se snažil rozdělit body podle tříd. Metoda dobře klasifikuje body do tříd.



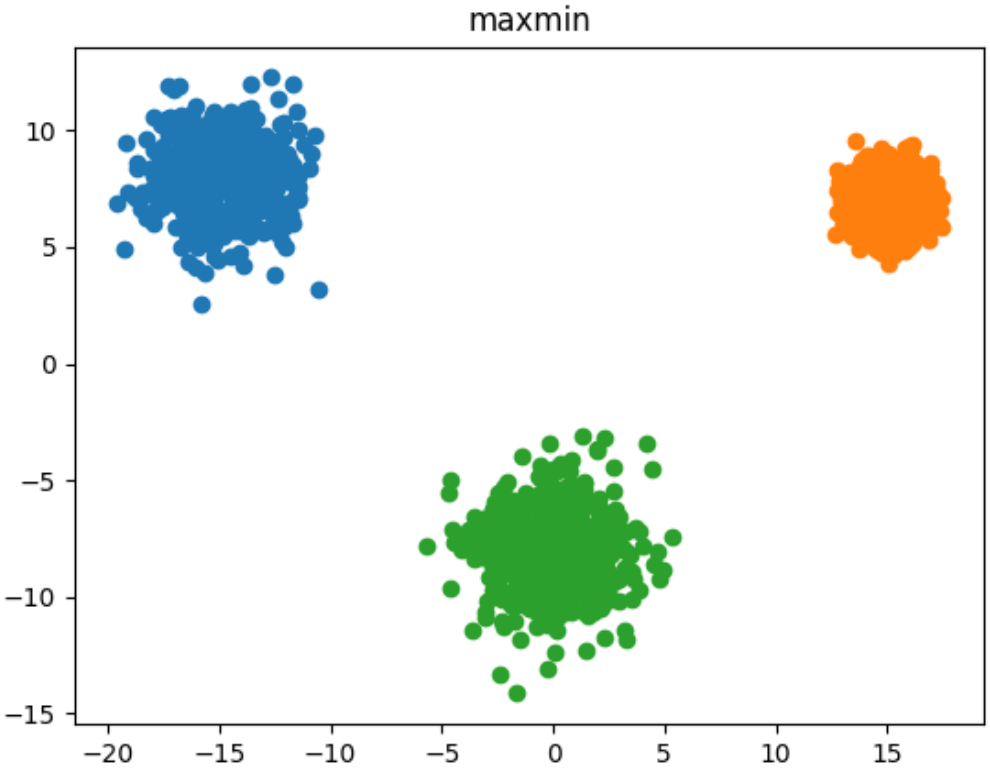
Obrázek 2: Metoda řetězové mapy

2.3 metoda maxmin

Po několika spuštěních metody si můžeeme všimnout, že výsledek závisí na počátečním bodu a na volbě parametru q.(viz. obrázek 3, 4) Pro danou sadu dat jsem zvolil q = 0,8 . Metoda je dostatečně rychlá. Taky jsem vyzkoušel klasifikovat body dané sady dat do tříd (viz. obrázek 3, 4). Většinou dostáneme správný výsledek klasifikace.



Obrázek 3: Nesprávný výsledek metody maxmin



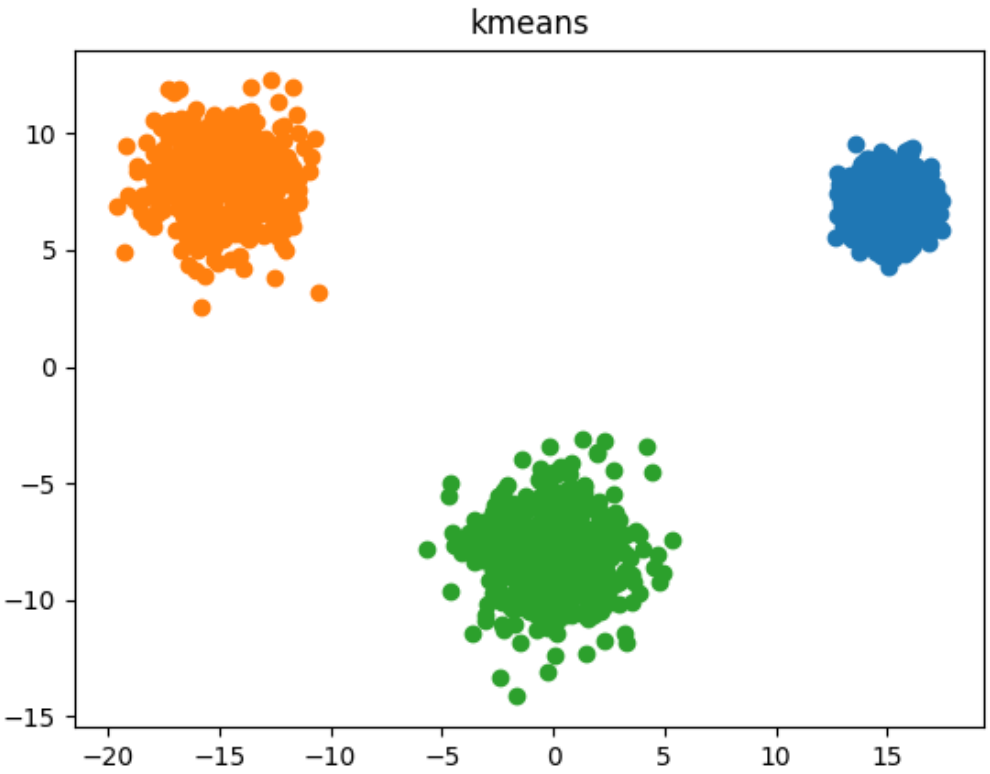
Obrázek 4: Správný výsledek metody maxmin

Nejrychlejší metoda je maxmin, ale ne vždy dává správný výsledek, na rozdíl od metody shlukové hladiny a metody řetězové mapy. Výsledek všech tří metod závisí na parametru q.

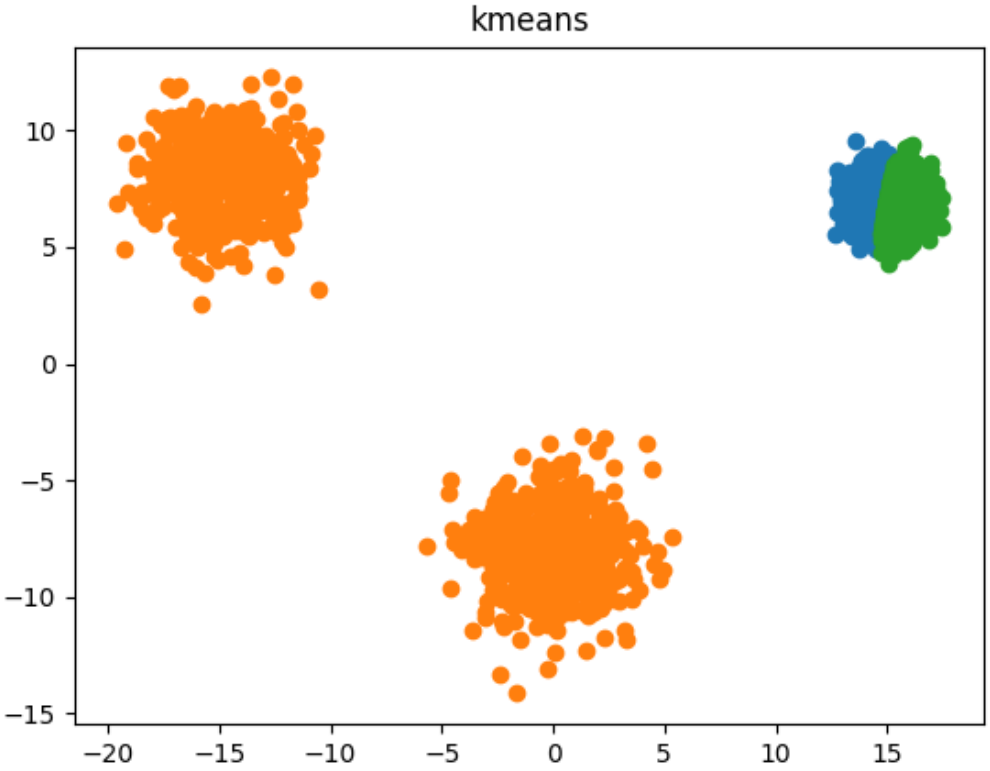
3 Metoda K-means

3.1 Přímé dělení do cílového počtu tříd

Výsledek metody závisí na počátečnéch bodech. Algoritmus bude konvergovat k lokálnímu minimu, který je určen počátečními body. Můžeme mít více než jedno lokální minimum, kvůli tomu algoritmus může dokonvergovat ke špatnému výsledku(viz. obr. 6). Tento problém můžeme vyřešit mnohonásobným spouštěním algoritmu(viy. obr. 5).



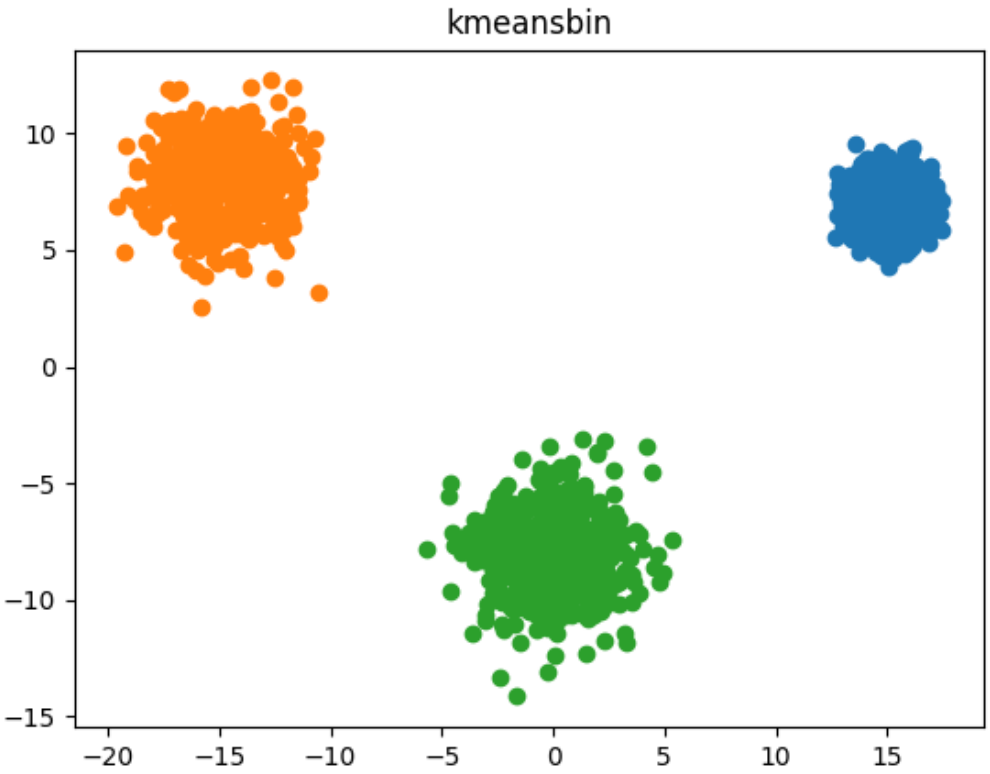
Obrázek 5: Správný výsledek kmeans



Obrázek 6: Nesprávný výsledek metody kmeans

3.2 Nerovnoměrné binární dělení

Binární dělení částečně řeší problém lokálního minima, protože pravděpodobnost dostat lokální minimum, při dělení do dvou tříd, je míň.



Obrázek 7: Kmeans binární dělení

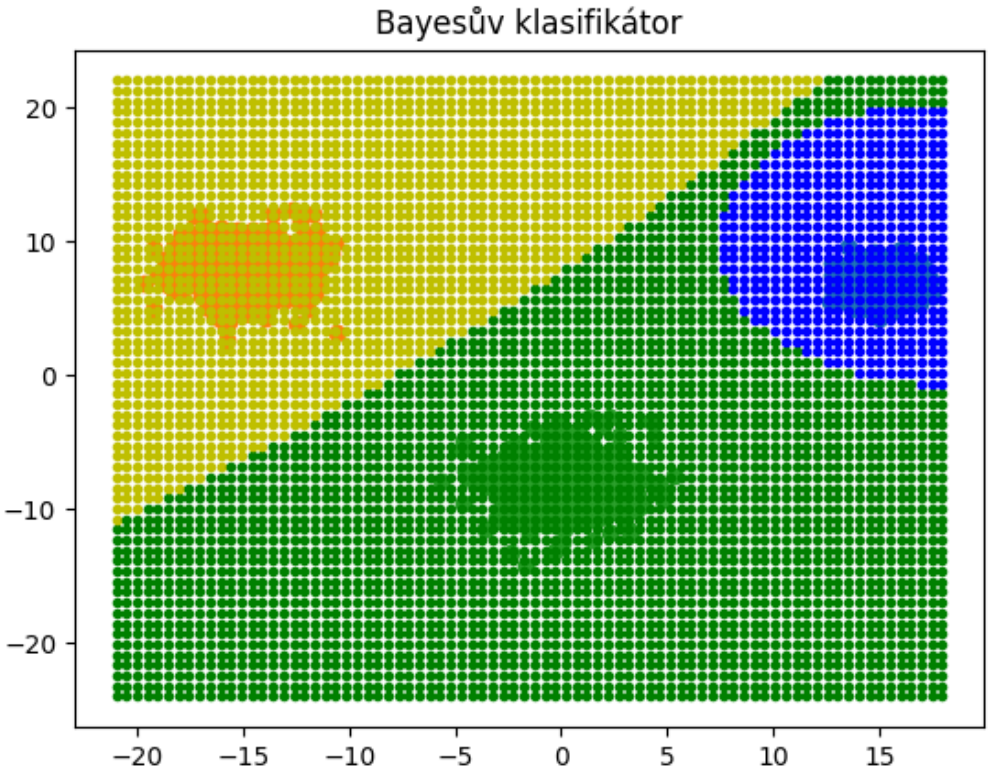
4 Iterativní optimalizace

Iterativní optimalizace má stejný princip práce jako kmeans. Proto metoda neshopná vyřešit problém lokálního minima a bude mít stejný výsledek jako metoda kmeans.

5 Trénování klasifikátorů

5.1 Bayesův klasifikátor

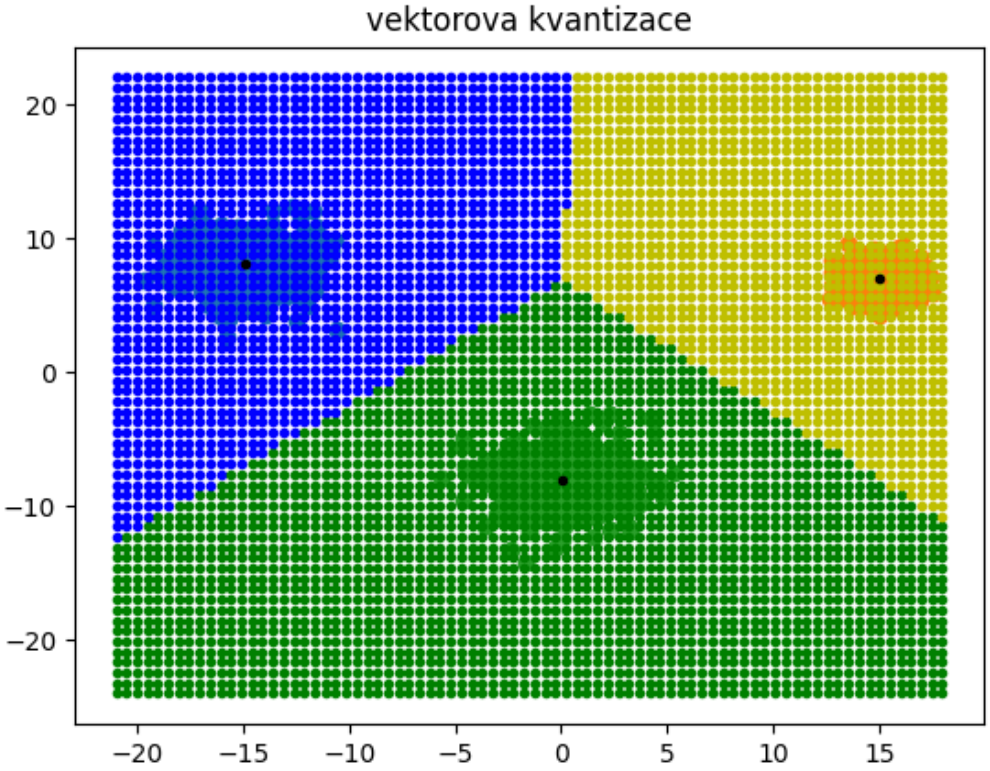
Je hezký vidět, že celková pravděpodobnost ovlivněná apriorní pravděpodobností.



Obrázek 8: Bayesův klasifikátorí

5.2 Vektorová kvantizace

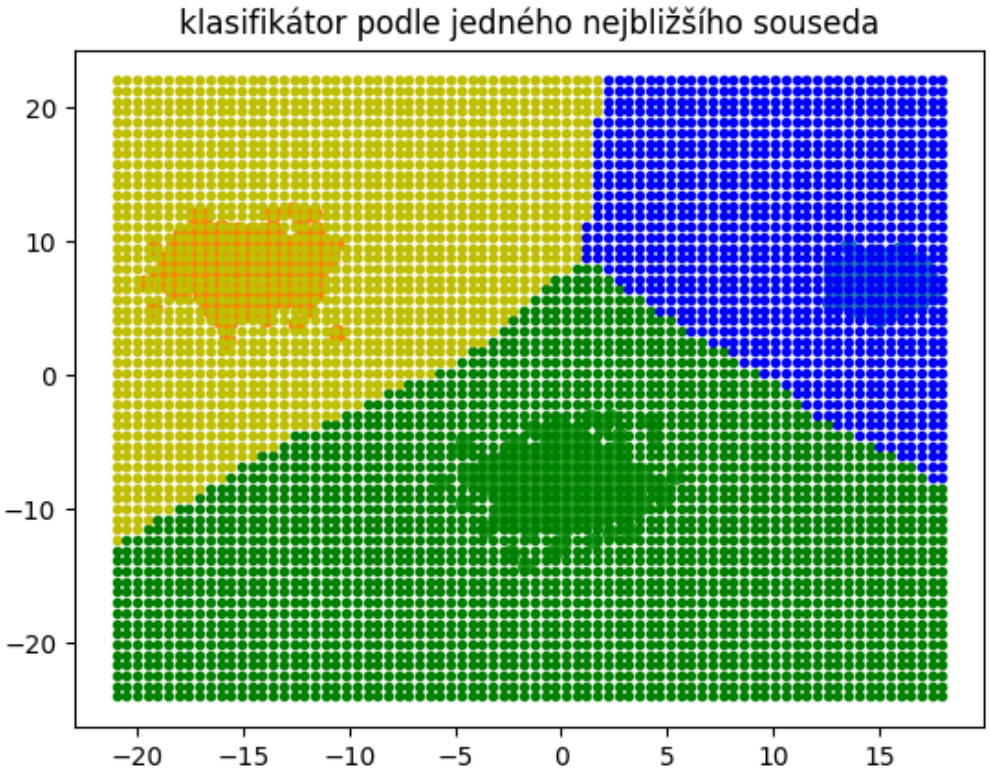
Metoda funguje dobře. Výsledek je podobný na výsledek NN klasifikáce.



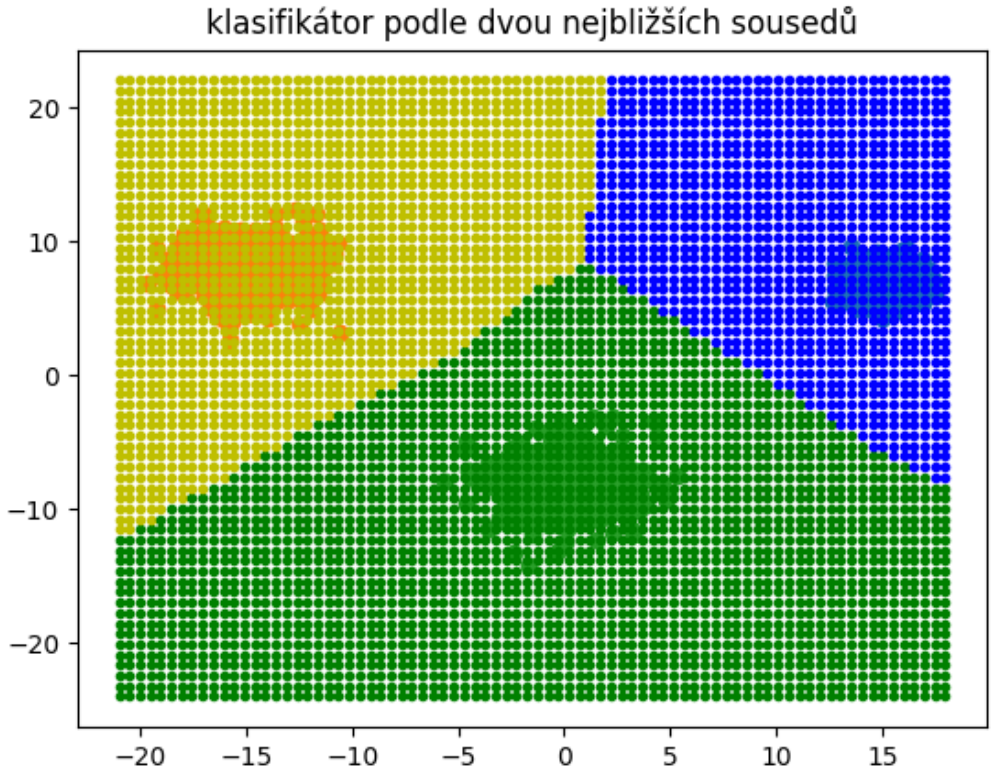
Obrázek 9: Vektorová kvantizace

5.3 Klasifikátor podle nejbližšího souseda

Z obrázků je vidět, že se zvýšením počtu sousedů, separace tříd konverguje k přímé čáře.



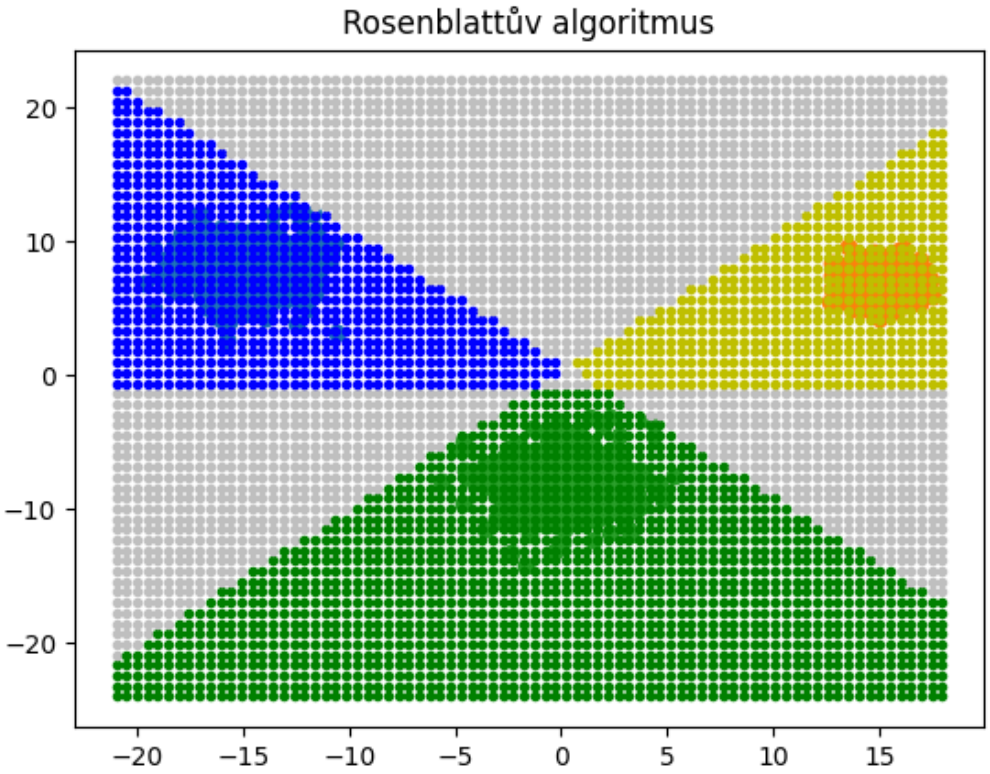
Obrázek 10: Klasifikátor podle jedného nejbližšího souseda



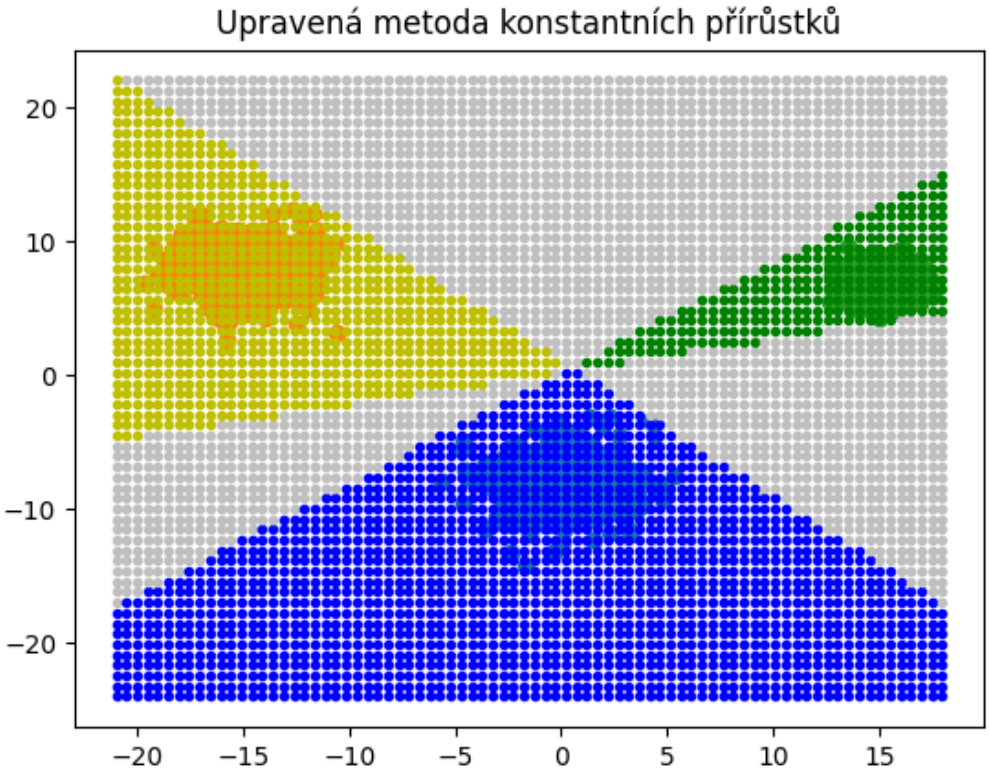
Obrázek 11: Klasifikátor podle dvou nejbližších sousedů

5.4 Klasifikátor s lineárními diskriminačními funkcemi

Porovnáme ě metody: Rosenblattův algoritmus a upravená metoda konstantních přírůstků (UMKP). UMKP má větší počet iterací než Rosenblattův alg. Můžeme všimnout, že kvůli počtu iterací UMKP má přesnější hranici vzhledem ke trenovácí množině.



Obrázek 12: Rosenblattův algoritmus

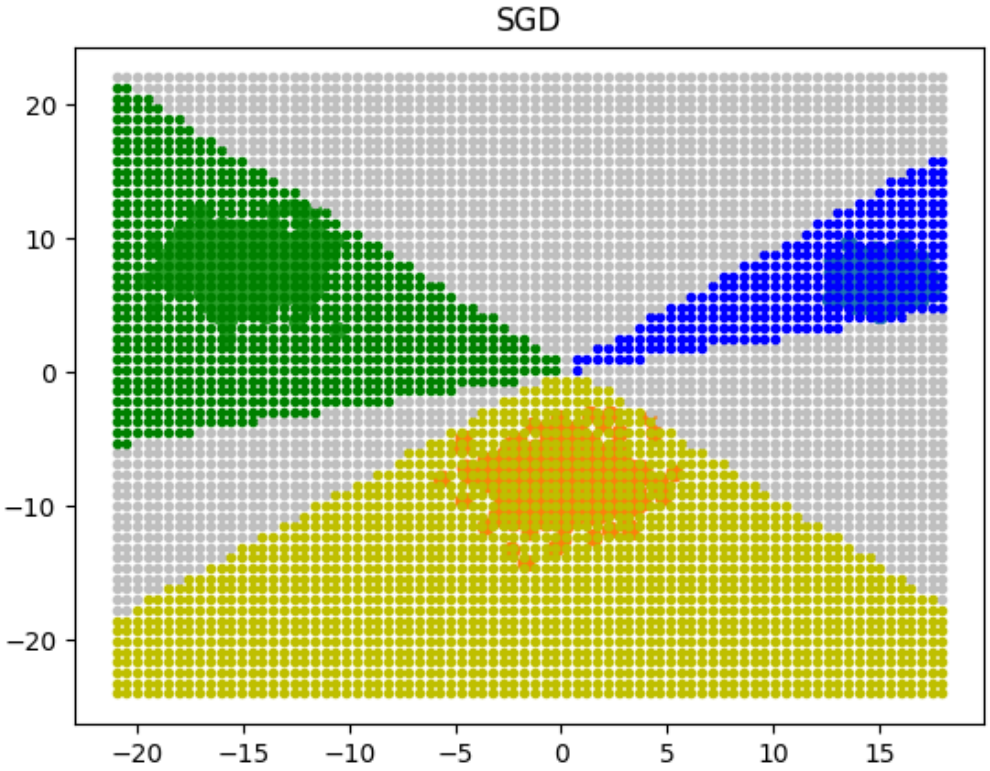


Obrázek 13: UMKP

6 neuronová síť

6.1 SGD

SGD trenování je založeno na algoritmu Rosenblatta, proto má stejné vlastnosti a podobný výsledek.



Obrázek 14: SGD

6.2 BGD

Jsme předpoklad nulové chyby, proto vznikl cyklus v trenování (2-2-4 chyby). Z tohoto důvodu nemáme konečný výsledek trenování a nemůžeme ohodnotit rastr.