

شناسایی آماری الگو

تمرین های سری چهار

فرهاد دلیرانی

۹۶۱۳۱۱۲۵

dalirani@aut.ac.ir

dalirani.1373@gmail.com

تمام کدها با پایتون 3.6 نوشته شده‌اند.

همچنین از پکیج‌های زیر استفاده کرده‌ام:

numpy -

matplotlib -

البته برای راحتی در نصب پایتون 3.6 و پکیج‌های مربوط به دیتاساینس که numpy و matplotlib هم جزیی از آن پکیج‌ها هستند از Anaconda 5.0.0 استفاده کرده‌ام که همه‌ی موارد گفته شده را بدون دردسر و سختی نصب می‌کند. تنها کافی است آن را از <https://www.anaconda.com/download> دانلود کنید و Installer باقی کار را انجام می‌دهد.

زبان برنامه نویسی: پایتون 3.6

پکیج‌ها: پکیج‌های گفته شده را برای راحتی در نصب با Anaconda نصب کردم.

ورژن Anaconda من: Anaconda 5.0.0 For Linux Installer که البته همین ورژن برای سایر

سیستم عامل‌ها هم موجود است.

محیط برنامه نویسی: pyCharm Community Edition

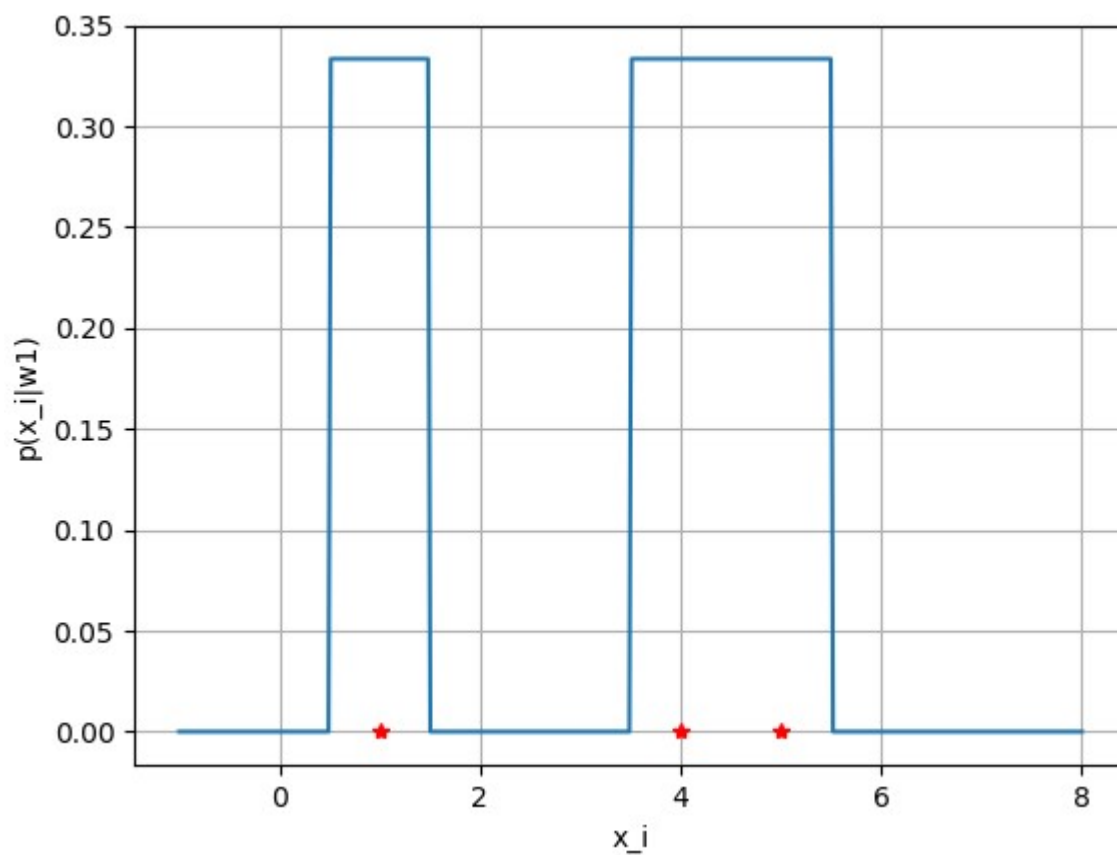
سوال ۱)

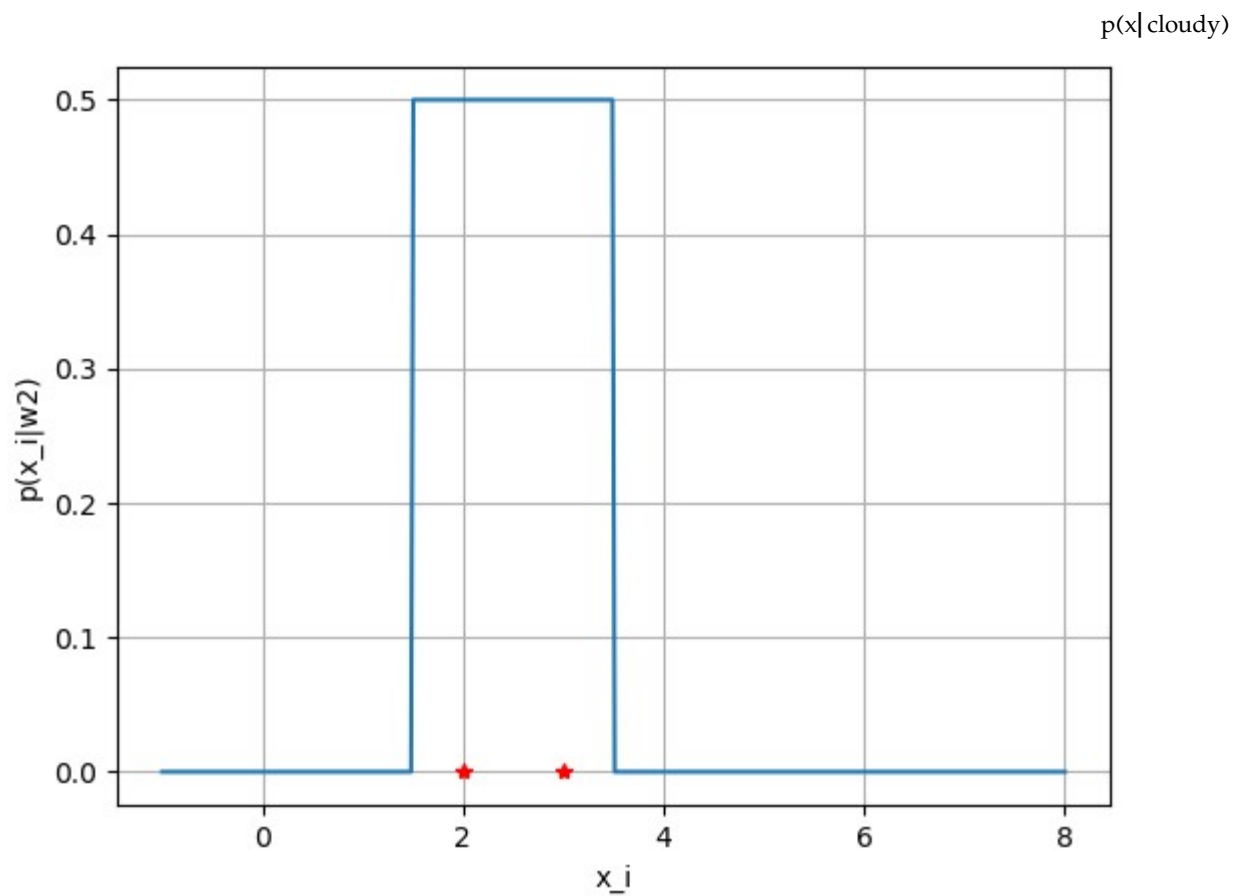
این سوال را به دو روش حل کرده‌ام:

روش اول:

کد پایتونی نوشته‌ام که با استفاده از رابطه‌های داده شده در سوال میزان $p(x|w_1)$ و $p(x|w_2)$ را برای 400 عدد بین 1- و 8 محاسبه می‌کند و در نهایت نمودارها را رسم می‌کند. کد مربوط به این سوال در problem1 و در فایل problem1.py موجود است. خروجی حاصل از اجرای آن کد را در تصویر زیر مشاهده می‌کنید:

$P(x|\text{sunny})$:



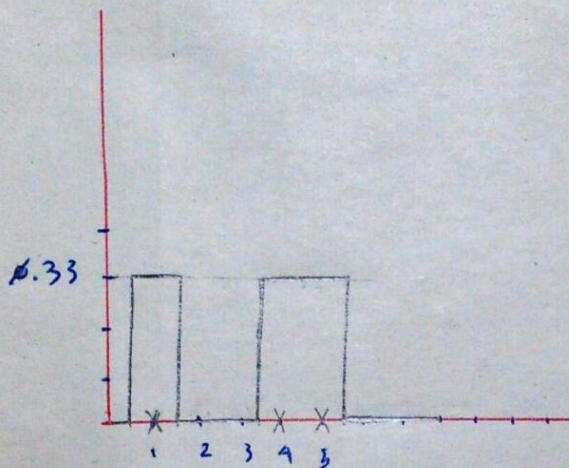


روش دوم:

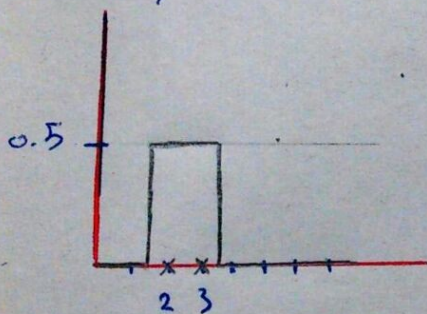
$$p(x) = \frac{1}{n} \sum_{i=1}^n \frac{1}{h^d} p\left(\frac{x-x_i}{h}\right)$$

$$= \sum_{i=1}^n \frac{1}{nh^d} p\left(\frac{x-x_i}{h}\right)$$

$\Rightarrow p(x)$ is sum of n
box of height $\frac{1}{nh^d}$



sunny \rightarrow 3 box with ~~height~~ height $\frac{1}{nh^d} = \frac{1}{3 \times 1} = 0.33$



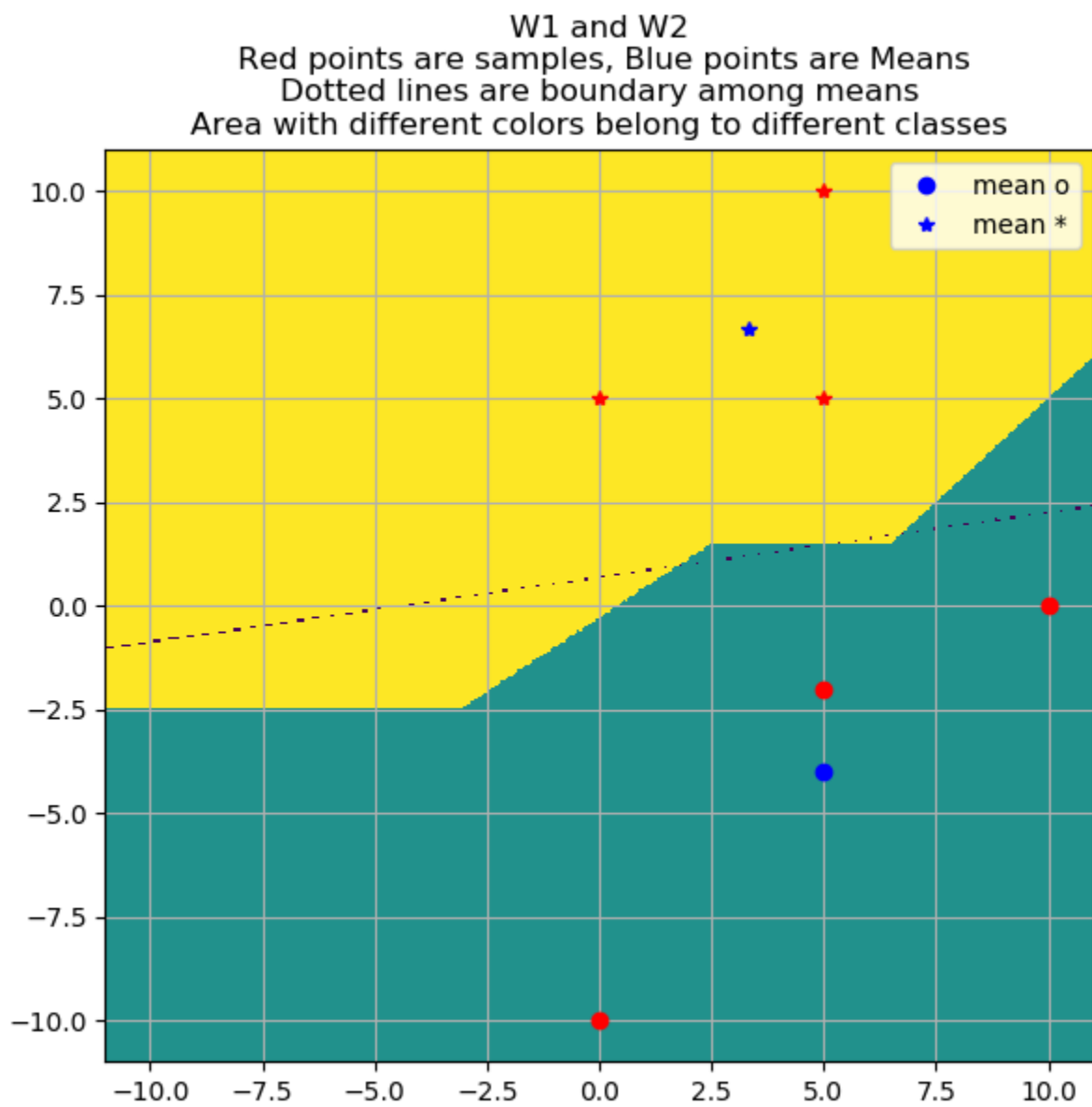
cloudy \rightarrow 2 box height of each is $\frac{1}{nh^d} = \frac{1}{2 \times 1} = 0.5$

سوال ۲)

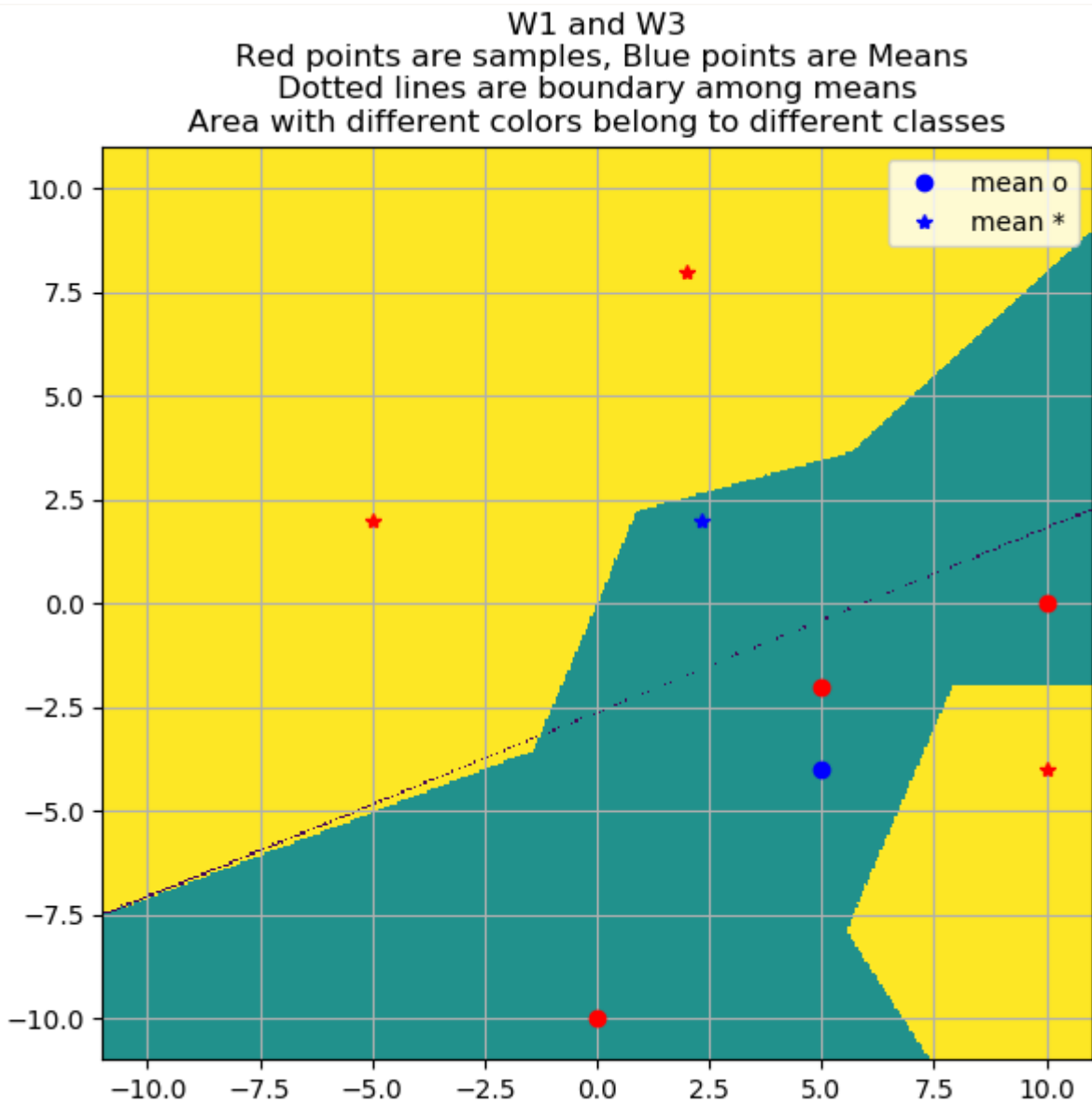
برای حل این سوال کد پایتونی نوشته‌ام که در پوشه‌ی problem2 و در فایل problem2.py قرار دارد. تابع `draw_decision_boundary_of_knn(wi, wj, wk, title)` را نوشته‌ام که sample های دو و یا سه کلاس را می‌گیرد و نمونه‌های را با رنگ قرمز رسم می‌کند و نمونه‌های هر کلاس را برای تمایز با شکل‌های مختلفی همچون ستاره، دایره و مثلث رسم می‌کند. همین‌طور میانگین هر کلاس را با رنگ آبی رسم می‌کند. ناحیه‌های متعلق به هر کلاس را با رنگ متفاوتی نمایش می‌دهد و محل تلاقی رنگ‌های مختلف مرز تصمیم KNN است. مرز بین میانگین کلاس‌ها را با خط‌های بریده مشخص می‌کند.

نتایج حاصل از اجرای کد را در زیر مشاهده می‌کنید:

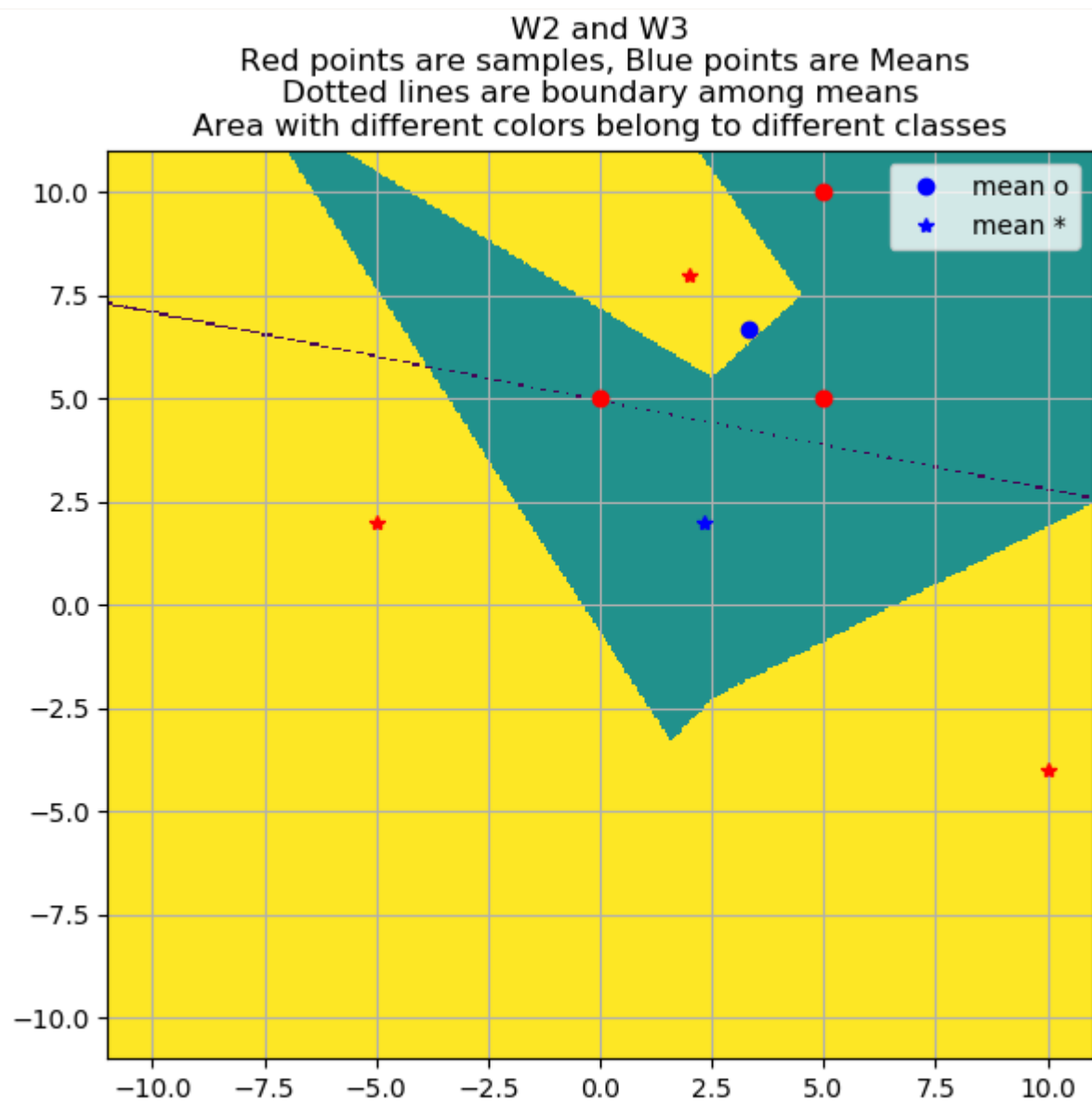
بخش a)



بخش (b)



بخش c)



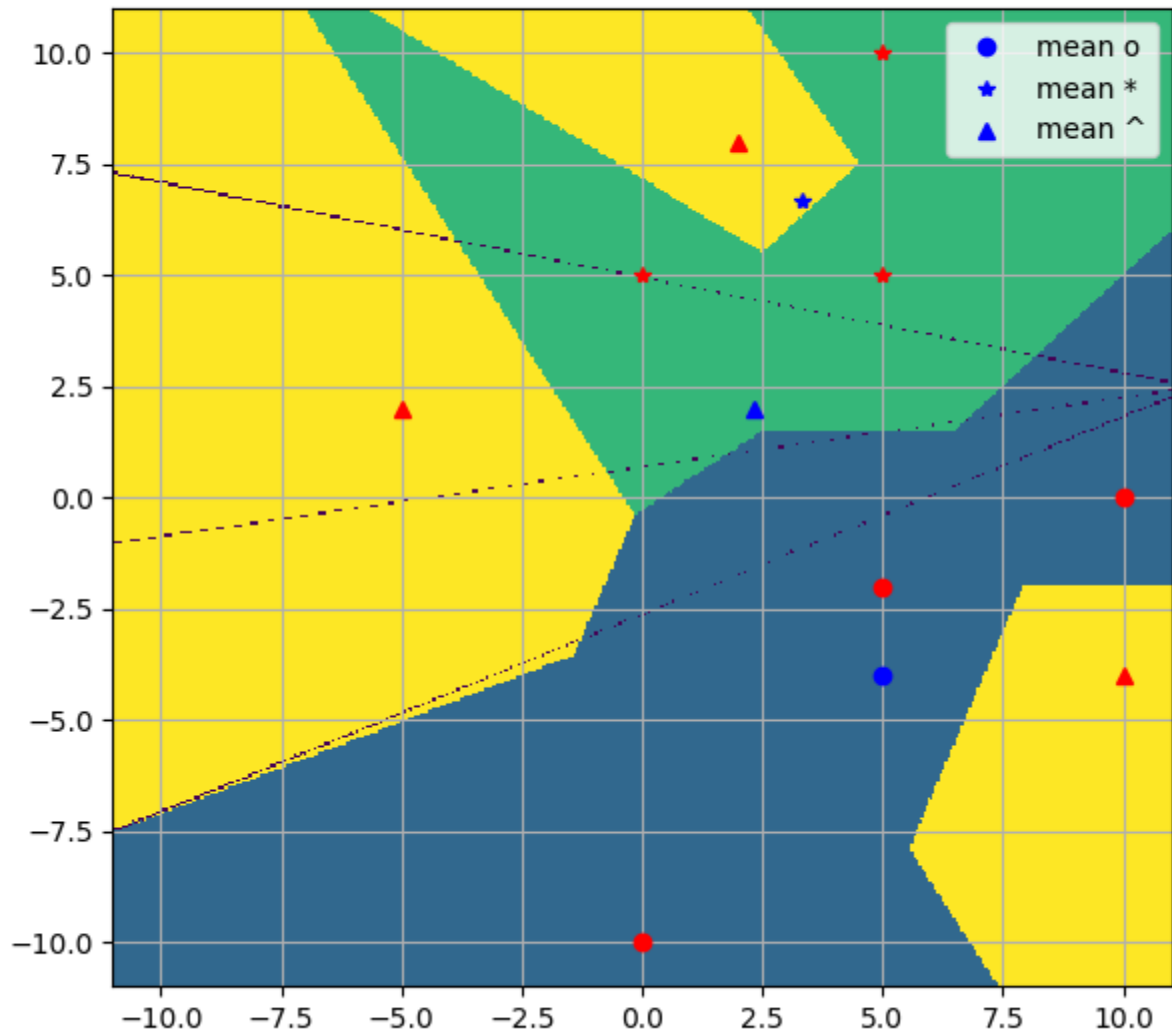
بخش d)

W1, W2 and W3

Red points are samples, Blue points are Means

Dotted lines are boundary among means

Area with different colors belong to different classes



سوال ۳)

کدهای مربوط به این سوال problem3 و در پوشه‌ی problem3.py است. برای این سوال تابع‌های زیر را نوشته‌ام:

`phi(x, xi, h)`

این تابع مقدار تابع parzen به مرکز x را برای یک نقطه محاسبه می‌کند.

`p_x_given_wj(wj, x, h)`

این تابع با استفاده از تابع قبل مقدار $p(x|w_j)$ را با استفاده از روش parzen تخمین می‌زند.

`classify(dataset, x, h)`

این تابع مقدار $p(x|w_i)p(w_i)$ را برای تمام کلاس‌ها محاسبه می‌کند و کلاس نقطه‌ی x را پیدا می‌کند.

خروجی حاصل از اجرای کد بالا را مشاهده می‌کنید:

```
h=1,      x=[0.5, 1.0, 0.0],      predict=w2
h=1,      x=[0.31, 1.51, -0.5],  predict=w2
h=1,      x=[-0.3, 0.44, -0.1],  predict=w2
h=0.1,    x=[0.5, 1.0, 0.0],      predict=w2
h=0.1,    x=[0.31, 1.51, -0.5],  predict=w2
h=0.1,    x=[-0.3, 0.44, -0.1],  predict=w2
```

سوال 4)

بخش a,b,c)

برای حل این سوال کد پایتونی نوشته‌ام که در پوشه‌ی problem4 و در فایل problem4.py است. تابع‌های زیر را نوشته‌ام:

`univariate_normal(mean, variance, x)`

این تابع یک نقطه را می‌گیرد و احتمال آن را در یک توزیع نرمال استاندارد برمی‌گرداند. از این تابع برای رسم توزیع نرمال استفاده کرده‌ام.

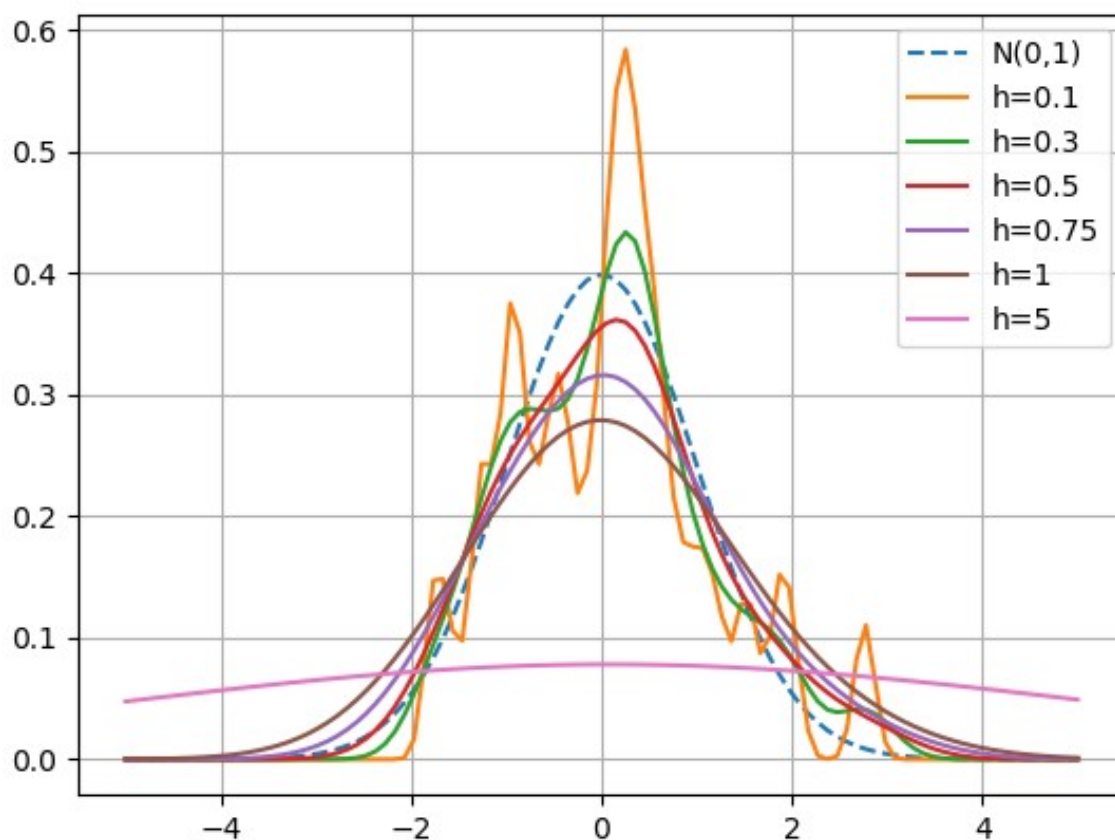
`phi(x, xi, h)`

این تابع مقدار پنجره‌ی parzen به مرکز x را در نقطه‌ی xi محاسبه می‌کند.

`estimate_px(dataset, x, h)`

این تابع با استفاده از روش تخمین parzen احتمال $p(x)$ را حساب می‌کند.

با استفاده از این تابع‌ها برای مقدارهای متفاوت h توزیع نرمال اولیه را تخمین زده‌ام. بعد از اجرای کد خروجی زیر به دست می‌آید:



خط چین آبی رنگ توزیع نرمال استاندارد است. هر چه میزان h کمتر می‌شود توزیع تخمین زده شده

sharp تر می شود و دچار برآمدگی های بیشتر و تیزتری می شود و هرچه h بزرگ می شود توزیع تخمین زده شده smooth تر می شود.

سوال (5)

کدهای مربوط به این سوال در پوشه‌ی problem5 و در فایل problem5.py قرار دارد. برای انجام این سوال از تابع‌هایی که برای سوال ۴ نوشته‌ام استفاده کرده‌ام.

`univariate_normal(mean, variance, x)`

این تابع یک نقطه را می‌گیرد و احتمال آن را در یک توزیع نرمال استاندارد برمی‌گرداند. از این تابع برای رسم توزیع نرمال استفاده کرده‌ام.

`phi(x, xi, h)`

این تابع مقدار پنجره‌ی parzen به مرکز x را در نقطه‌ی xi محاسبه می‌کند.

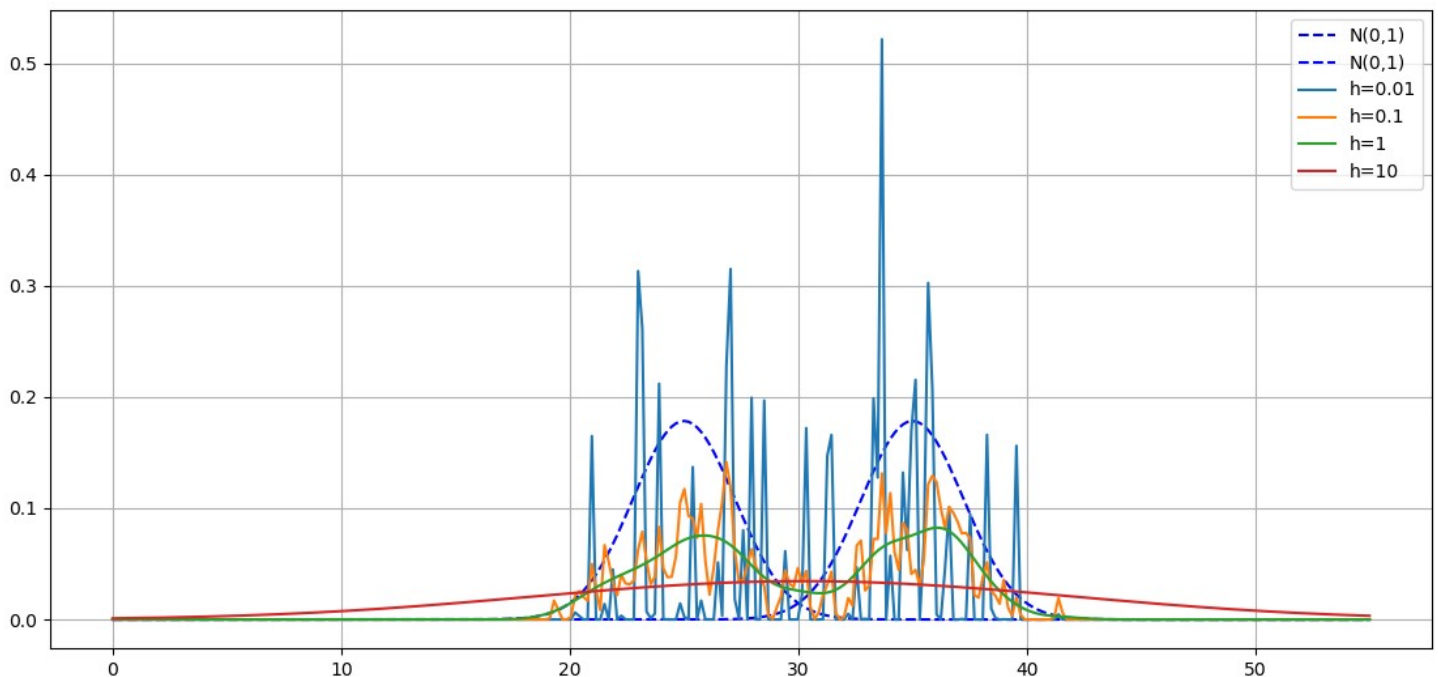
`estimate_px(dataset, x, h)`

این تابع با استفاده از روش تخمین parzen احتمال $p(x)$ را حساب می‌کند.

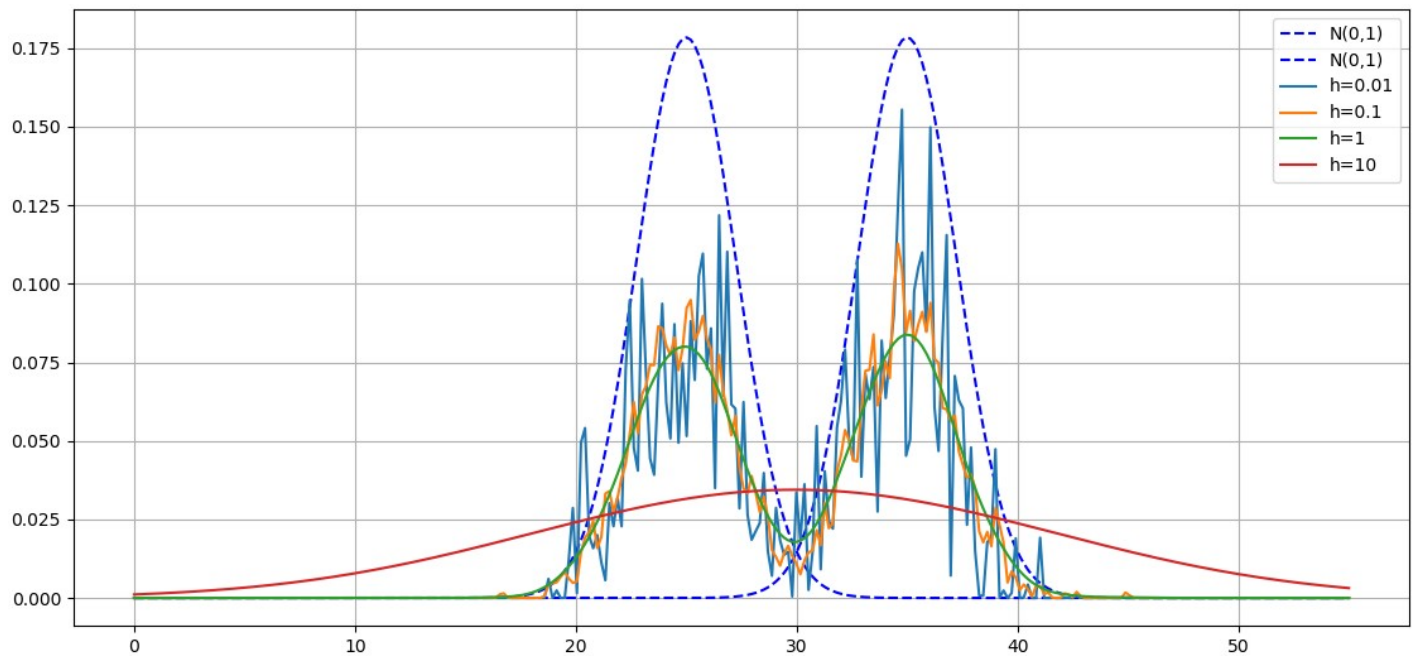
بخش a,b)

بعد از اجرای کد برای بخش a و b خروجی‌های زیر به دست می‌آید:

برای 200 نمونه:



برای 2000 نمونه:



بخش C:

هر چه h بزرگ تر شده است تخمین smooth تر شده است و تعداد قله ها و برآمدگی هایش کم شده است. هر چه h کوچک می شود تخمین sharp تر می شود و تعداد قله ها و برآمدگی ها بیشتر می شود.

هر چه تعداد samples ها بیشتر می شود تخمین ها نسبت به متناظر خود در تعداد sample های کم تر، smooth تر می شوند و برآمدگی ها آن ها کم تر و همین طور نرم تر می شود.