به نام خدا

دانشگاه تهران

دانشکدگان فنی

دانشکده مهندسی برق و کامپیوتر

یادگیری ماشین

تمرین اول

نام و نام خانوادگی : حسین سیفی

شماره دانشجویی : ۸۱۰۱۰۰۳۸۶

اسفند ۱۴۰۱

فهرست

سوال ۱ .... ۳
آ ۳
ب ۳
پ ۵
ت ۵
ث ۵
سوال ۲ .... ۸
سوال ۳ .... ۱۰
آ ۱۰
ب ۱۱
پ ۱۲
ت ۱۲
ث ۱۳
سوال ۴ .... ۱٤
آ ۱٤
ب ۱٤
پ ۱٥
ت ۱٥
ث ۱٥
ج ۱٥
سوال ۵ .... ۱٦
آ ۱٦
طبقه‌بند naïve bayes ۱٦
تفاوت طبقه‌بند بیز و naïve bayes ۱٦
هزینه و مواقع استفاده از naïve bayes ۱۷
ب ۱۷
پیش‌پردازش ۱۷
مرحله آموزش ۱۸
مرحله تست ۲۰
ارزیابی ۲۰
پ ۲۲
سوال ۶ .... ۲٤

سوال ۱

آ

برای هر کدام از دو کلاس ۱ و ۲ رابطه را با جایگذاری $x = \frac{a_1+a_2}{2}$ بازنویسی می‌کنیم و تا حد ممکن ساده می‌کنیم:

$$P(x|\omega_1) = \frac{1}{\pi b}\frac{1}{1+(\frac{\frac{a_1+a_2}{2}-a_1}{b})^2} = \frac{1}{\pi b}\frac{1}{1+(\frac{a_2-a_1}{2b})^2}$$

$$P(x|\omega_2) = \frac{1}{\pi b}\frac{1}{1+(\frac{\frac{a_1+a_2}{2}-a_2}{b})^2} = \frac{1}{\pi b}\frac{1}{1+(\frac{a_1-a_2}{2b})^2}$$

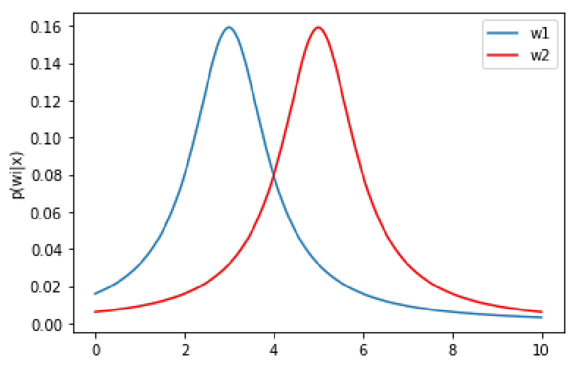
می‌دانیم که عبارت دارای توان دو را می‌توان قرینه کرد و به نتیجه یکسانی برسیم، بنابراین می‌توان از عبارت اول به عبارت دوم رسید:

$$P(x|\omega_1) = \frac{1}{\pi b}\frac{1}{1+(\frac{-(a_2-a_1)}{2b})^2} = \frac{1}{\pi b}\frac{1}{1+(\frac{a_1-a_2}{2b})^2} = P(x|\omega_2)$$

بنابراین نشان دادیم که به ازای $x = \frac{a_1+a_2}{2}$ در تابع توزیع کوشی و در حالتی که $P(\omega 1)=P(\omega 2)$ گزاره مورد نظر اثبات می‌شود:

$$P(x|\omega_1) = P(x|\omega_2)$$

به کمک زبان برنامه نویسی پایتون دو نمودار $P(\omega 1|x)$ و $P(\omega 2|x)$ رسم شدند. برای رسم این نمودارها بازه ۰ تا ۱۰ با گام ۰.۰۰۱ برای متغیر تصادفی x انتخاب شدند و نمودار به شکل زیر به دست آمد. برای رسم این نمودار از مقدار P(x) صرف نظر شده است:



همانطور که در تصویر فوق مشخص است، دو نمودار در نقطه x=4 با یکدیگر تقاطع داشته‌اند که مطابق با گزاره اثبات شده در مرحله قبل، برابر با میانگین مقدار a دو توزیع است.

ب

برای به دست آوردن مقدار احتمال خطا می‌توان از رابطه زیر استفاده کرد:

$$p(error) = \int_{R_1} p(x|w_1)p(w_1)dx + \int_{R_2} p(x|w_2)p(w_2)dx$$

با توجه به اینکه در بخش قبلی سوال مرز تصمیم مشخص شد و در همان حالت است که حداقل احتمال خطا به وجود می‌آید. بنابراین با توجه به مرز تصمیم انتخاب شده، احتمال پیشین برابر برای هر دو کلاس و فرمول داده شده در بخش الف سوال، عبارت فوق را به شکل زیر بازنویسی می‌کنیم:

$$p(error) = \frac{1}{2\pi b}\int_{\frac{a_1+a_2}{2}}^{\infty} \frac{1}{1+(\frac{x-a_1}{b})^2}dx + \frac{1}{2\pi b}\int_{-\infty}^{\frac{a_1+a_2}{2}} \frac{1}{1+(\frac{x-a_2}{b})^2}dx$$

برای حل انتگرال‌های فوق در ابتدا نیاز است که تغییر متغیر صورت گیرد. بنابراین مقدار متغیرهای جدید u_1 و u_2 به شکل $u_1 = \frac{x-a_1}{b}$ و $u_2 = \frac{x-a_2}{b}$ تعریف می‌شوند و در عبارت فوق جایگزین می‌شود. همچنین مقدار du_1 و du_2 به شکل $du_1 = du_2 = bdx$ تعریف می‌شود.

$$p(error) = \frac{1}{2\pi b}\int_{\frac{a_1+a_2}{2}}^{\infty} \frac{1}{1+(u_1)^2}bdu_1 + \frac{1}{2\pi b}\int_{-\infty}^{\frac{a_1+a_2}{2}} \frac{1}{1+(u_2)^2}bdu_2$$

همچنین می‌دانیم که $\int \frac{du}{1+u^2} = \arctan(u)$ است و بنابراین مقدار انتگرال‌های فوق به شکل زیر محاسبه می‌شود و مقادیر مورد نیاز در جای متغیرهای کمکی قرار می‌گیرد:

$$p(error) = \frac{1}{2\pi}\arctan(\frac{x-a_1}{b})\Big|_{\frac{a_1+a_2}{2}}^{\infty} + \frac{1}{2\pi}\arctan(\frac{x-a_2}{b})\Big|_{-\infty}^{\frac{a_1+a_2}{2}}$$

با دقت در نمودار تابع تانژانت وارون می‌توان متوجه شد که این تابع در مثبت بی‌نهایت به مقدار $\frac{\pi}{2}$ و در منفی بی‌نهایت به $-\frac{\pi}{2}$ میل می‌کند. بنابراین با جایگذاری این مقادیر و سایر مقادیر مشخص در عبارت فوق، این عبارت به شکل زیر درمی‌آید:

$$p(error) = \frac{1}{2\pi}\left(\frac{\pi}{2} - \arctan(\frac{a_2-a_1}{2b})\right) + \frac{1}{2\pi}\left(\arctan(\frac{a_1-a_2}{2b}) - (-\frac{\pi}{2})\right)$$

این عبارت به شکل زیر ساده می‌شود:

$$p(error) = \frac{1}{2} - \frac{1}{2\pi}\arctan(\frac{a_2-a_1}{2b}) + \frac{1}{2\pi}\arctan(\frac{a_1-a_2}{2b})$$

با توجه به اینکه مقدار a_2 از a_1 در تعریف مساله بزرگتر است، مقدار موجود در تانژانت وارون سمت چپ همیشه مثبت است و در نتیجه خروجی تانژانت وارون نیز مثبت است اما عبارت درون تانژانت وارون دوم منفی است و مقدار نهایی این عبارت نیز منفی می‌شود. همچنین این تابع خاصیت $\arctan(-u) = -\arctan(u)$ را نیز دارا می‌باشد. در نتیجه می‌توان عبارت دوم را درون قدرمطلق قرار داد و عبارت تانژانت وارون را در منفی ضرب کرد در نتیجه عبارت نهایی پس از ساده سازی به شکل زیر درمی‌آید:

$$p(error) = \frac{1}{2} - \frac{1}{\pi}\arctan\left|\frac{a_2-a_1}{2b}\right|$$

عبارت فوق به عنوان حداقل مقدار خطا به دست می‌آید و گزاره مورد نظر در صورت سوال اثبات می‌شود.

پ

در دو حالت ممکن است که مقدار احتمال خطا بیشتر از حداقل مقدار احتمال خطا باشد. این دو حالت بدین شکل هستند که:

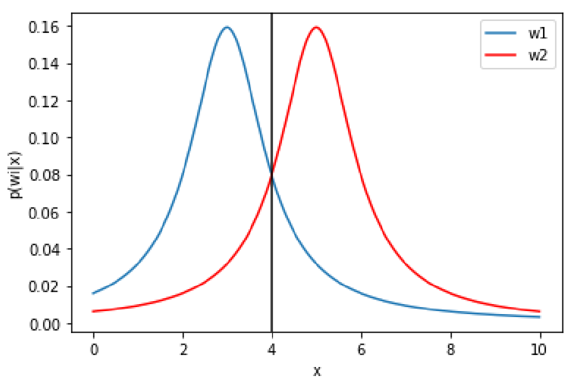
۱. اگر مرز تصمیم به درستی انتخاب نشود می‌تواند منجر به ایجاد مقدار احتمال خطای بسیار بزرگی شود. برای مثال در صورتی که مرز تصمیم در مثبت بی‌نهایت در نظر گرفته شود(تمامی نمونه‌ها متعلق به یک کلاس در نظر گرفته شوند)، یکی از نمودارها به صورت کامل جزو مقدار خطا محاسبه می‌شود.
۲. هر چه اختلاف مقادیر a_1 و a_2 کمتر شود، احتمال خطا بیشتر می‌شود تا جایی که این دو مقدار با یکدیگر برابر شوند. در صورتی که $a_1=a_2$ باشد، عبارت محاسبه شده در بخش قبلی سوال به شکل زیر درمی‌آید:

$$p(error) = \frac{1}{2} - \frac{1}{\pi}\arctan\left|\frac{0}{2b}\right| = \frac{1}{2} - \frac{1}{\pi} * 0 = \frac{1}{2}$$

از آنجایی که مقدار تانژانت وارون که حاوی عبارتی دارای قدر مطلق است نمی‌تواند منفی باشد، صفر کمترین مقداری است که به عنوان خروجی می‌تواند داشته باشد و اگر مرز تصمیم به درستی انتخاب شود اما توزیع مقادیر دو کلاس به شدت به یکدیگر نزدیک باشند، مقدار احتمال خطای این دسته‌بند برابر با ۰.۵ خواهد بود و این بیشینه مقدار خطا است.

ت

طبق اثبات بخش آ و محاسبات انجام شده، مرز تصمیم در نقطه $\frac{a_1+a_2}{2}$ قرار می‌گیرد که با توجه به مقادیر انتخاب شده برای پارامترها، مرز تصمیم در x=4 قرار می‌گیرد و در نمودار زیر قابل مشاهده است:



همچنین با توجه به انتخاب مرز تصمیم در نقطه بهینه، می‌توان از فرمول به دست آمده در قسمت ب همین سوال استفاده کرد تا مقدار احتمال خطا را به دست آورد. این مقدار به صورت زیر به دست می‌آید:

$$p(error) = \frac{1}{2} - \frac{1}{\pi}\arctan\left(\frac{5-3}{1}\right) \approx \frac{1}{2} - \frac{1}{\pi} * 1.1 \approx \frac{1}{2} - \frac{35}{100} = 0.15$$

ث

برای طراحی طبقه‌بند بیزی با مقادیر متفاوت ریسک و یافتن مرز تصمیم می‌توان از فرمول زیر استفاده کرد که در کلاس و کتاب مرجع استفاده شده است:

$$\frac{p(x|w_1)}{p(x|w_2)} >^{w_1} \frac{\lambda_{21} - \lambda_{22}}{\lambda_{12} - \lambda_{11}} \frac{p(w_1)}{p(w_2)}$$

با جایگزینی مقادیر عبارات احتمالی و مقادیر ریسک به عبارت زیر می‌رسیم:

$$\frac{\frac{1}{\pi b}\frac{1}{1+(\frac{x-a_1}{b})^2}}{\frac{1}{\pi b}\frac{1}{1+(\frac{x-a_2}{b})^2}} >^{w_1} \frac{2-0}{1-0} * \frac{0.5}{0.5}$$

با انجام مراحل ساده‌سازی زیر، درنهایت به عبارت نهایی می‌رسیم و می‌توان مقادیر پارامترها را در آن جایگزین کرد:

$$1+(\frac{x-a_2}{b})^2 >^{w_1} 2+2(\frac{x-a_1}{b})^2$$

$$\frac{(x-a_2)^2 - 2*(x-a_1)^2}{b^2} >^{w_1} 1$$

$$\frac{-x^2+2(2a_1-a_2)x+(a_2^2-2a_1^2-b^2)}{b^2} >^{w_1} 0$$

حال می‌توان مقادیر ۱ و ۳ و ۵ را به ترتیب برای پارامترهای b و a_1 و a_2 قرار می‌دهیم:

$$-x^2+2x+6 >^{w_1} 0$$

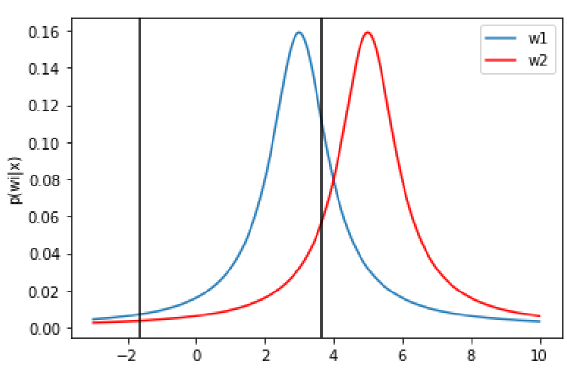
با حل معادله درجه دو فوق به دو ریشه زیر می‌رسیم:

$$x = \begin{cases} 1+\sqrt{7} \approx +3.64 \\ 1-\sqrt{7} \approx -1.64 \end{cases}$$

عبارت فوق به شکل زیر تعیین علامت می‌شود:

x		***-1.64***		***+3.64***	
$-x^2+2x+6$	-		+		-

در نتیجه بین مقادیر ۱.۶۴- و ۳.۶۴+ در کلاس w_1 و دیگر مقادیر در کلاس w_2 قرار می‌گیرند. نمودار دو توزیع فوق و مرزهای تصمیم به شکل زیر می‌باشند:



همانطور که مشخص است این طبقه‌بند که با در نظر گرفتن ریسک دو برابر برای عدم تشخیص کلاس دوم طراحی شده است، در تقابل با کلاس دوم بسیار محتاط‌تر عمل می‌کند و دقت[1] کلاس دوم(نمودار قرمز) را کاهش می‌دهد تا بتواند فراخوانی[2] این کلاس را افزایش دهد و نمونه‌هایی با برچسب واقعی ۲ را به اشتباه با برچسب ۱ تشخیص ندهد. همچنین می‌توان گفت فراخونی کلاس دوم نیز بر فراخوانی کلاس اول ارجحیت دارد.

برای محاسبه خطای این طبقه‌بند مشابه با بخش اول می‌توان عبارت زیر را نوشت:

$$p(error) = \frac{1}{2\pi b}\int_{-\infty}^{-1.64} \frac{1}{1+(\frac{x-a_1}{b})^2}dx + \frac{1}{2\pi b}\int_{3.64}^{\infty} \frac{1}{1+(\frac{x-a_1}{b})^2}dx + \frac{1}{2\pi b}\int_{-1.64}^{3.64} \frac{1}{1+(\frac{x-a_2}{b})^2}dx$$

پس از حل انتگرال‌های فوق مشابه با بخش آ و مقداردهی پارامترها به عبارات زیر می‌رسیم و در ادامه ساده می‌کنیم:

$$p(error) = \frac{1}{2\pi}\arctan{(x-3)}\begin{cases}-1.64\\-\infty\end{cases} + \frac{1}{2\pi}\arctan{(x-3)}\begin{cases}\infty\\3.64\end{cases} + \frac{1}{2\pi}\arctan{(x-5)}\begin{cases}3.64\\-1.64\end{cases}$$

$$p(error) = \frac{1}{2\pi}\left[\arctan(-4.64) + \frac{\pi}{2}\right] + \frac{1}{2\pi}\left[\frac{\pi}{2} - \arctan(0.64)\right] + \frac{1}{2\pi}[\arctan(-1.36) - \arctan(-6.64)]$$

$$p(error) = \frac{1}{2\pi}[\pi - 1.42] \approx 0.28$$

همانطور که مشاهده می‌شود این مقدار از حالتی که به دنبال حداقل کردن ریسک نباشیم، خطای بیشتری را به ما تحمیل می‌کند اما در صورتی که نیاز به طبقه‌بند با ریسک‌های مشخص شده وجود داشته باشد، هدف ما صرفا به حداقل رساندن خطا نیست و به دنبال افزایش دیگر معیارها مانند فراخوانی در یکی از کلاس‌ها (در این مثال کلاس ۲) هستیم.

[1] Precision
[2] Recall

سوال ۲

در ابتدا رابطه تعیین مرز تصمیم را به صورت زیر تشکیل می‌دهیم. مرز تصمیم به ازای مقادیری از x تعریف می‌شود که هر دو توزیع در آن نقطه دارای احتمال پسین یکسانی باشند:

$$p(w_1|x) >^{w_1} p(w_2|x)$$

سپس با استفاده از قانون بیز عبارت فوق را به شکل ضرب احتمال پیشین در likelihood بازنویسی می‌کنیم و از evidence که در هر دو طرف معادله موجود است، صرف نظر می‌کنیم:

$$p(x|w_1) * p(w_1) >^{w_1} p(x|w_2) * p(w_2)$$

با توجه به تساوی احتمالات پیشین در دو سوی نامساوی، از این احتمال نیز صرف نظر می‌کنیم و فرمول توزیع رایلی را در عبارت فوق جایگزین می‌کنیم:

$$\frac{x}{\sigma_1^2}\exp\left(-\frac{x^2}{2\sigma_1^2}\right) >^{w_1} \frac{x}{\sigma_2^2}\exp\left(-\frac{x^2}{2\sigma_2^2}\right)$$

می‌دانیم که لگاریتم جهت نامساوی را عوض نمی‌کند، بنابراین از طرفین نامساوی ln می‌گیریم تا عبارت را نسبت به توان e ساده کنیم:

$$\ln\left(\frac{1}{\sigma_1^2}\exp\left(-\frac{x^2}{2\sigma_1^2}\right)\right) >^{w_1} \ln\left(\frac{1}{\sigma_2^2}exp\left(-\frac{x^2}{2\sigma_2^2}\right)\right)$$

با استفاده از خواص لگاریتم، عبارت فوق را به شکل زیر ساده می‌کنیم:

$$-2\ln(\sigma_1) - \frac{x^2}{2\sigma_1^2} >^{w_1} -2\ln(\sigma_2) - \frac{x^2}{2\sigma_2^2}$$

و در ادامه به شکل زیر ساده می‌شود:

$$\frac{\sigma_1^2 - \sigma_2^2}{2\sigma_1^2\sigma_2^2}x^2 >^{w_1} 2\ln\left(\frac{\sigma_1}{\sigma_2}\right)$$

در این مرحله برای ادامه حل سوال فرض می‌کنیم که مقدار σ_1^2 از σ_2^2 بزرگتر است و به ساده‌سازی ادامه می‌دهیم:

$$x^2 >^{w_1} \frac{4\sigma_1^2\sigma_2^2\ln\left(\frac{\sigma_1}{\sigma_2}\right)}{\sigma_1^2 - \sigma_2^2}$$

$$x >^{w_1} \left|\sqrt{\frac{4\sigma_1^2\sigma_2^2 ln\left(\frac{\sigma_1}{\sigma_2}\right)}{\sigma_1^2 - \sigma_2^2}}\right|$$

در نتیجه در صورتی که مقدار نامساوی فوق صحیح باشد، کلاس ۱ انتخاب می‌شود و در غیر اینصورت کلاس ۲ انتخاب می‌شود. حال اگر حالت مکمل فرض در نظر گرفته شده را در نظر بگیریم، یعنی فرض کنیم که مقدار σ_2^2 از σ_1^2 بزرگتر است، در این حالت جهت نامساوی عوض می‌شود و پس از ساده‌سازی به عبارت زیر می‌رسیم:

$$x^2 <^{w_1} \frac{4\sigma_1^2\sigma_2^2 \ln\left(\frac{\sigma_1}{\sigma_2}\right)}{\sigma_1^2 - \sigma_2^2}$$

$$x <^{w_1} \left|\sqrt{\frac{4\sigma_1^2\sigma_2^2 ln\left(\frac{\sigma_1}{\sigma_2}\right)}{\sigma_1^2 - \sigma_2^2}}\right|$$

در این حالت نیز مانند نتیجه فرض قبلی، در صورتی که نامساوی فوق صحیح باشد، کلاس ۱ انتخاب می‌شود و در غیر اینصورت کلاس ۲ انتخاب می‌شود.

سوال ۳

آ

برای محاسبه مرز تصمیم دادگان موجود در تصویر، ابتدا باید مقدار میانگین را برای ویژگی‌های هر دو کلاس محاسبه کنیم:

$$mean(c_1) = \frac{1}{10}\begin{pmatrix} -2-1.5-1.5-1-0.5+0.5+0.5+1+1.5+1.5 \\ -1+0+1-1+0.5-0.5+0.5-1-0.5+0.5 \end{pmatrix} = \begin{pmatrix} -0.15 \\ -0.15 \end{pmatrix}$$

$$mean(c_2) = \frac{1}{9}\begin{pmatrix} 0+0.5+1+1+1.5+1.5+2+2+2.5 \\ 0.5+2+1+3+0+2+1+3+2 \end{pmatrix} = \begin{pmatrix} 1.33 \\ 1.61 \end{pmatrix}$$

در ادامه به محاسبه واریانس ویژگی‌های هر دو کلاس می‌پردازیم:

$$var(c_1) = \frac{1}{9}\begin{pmatrix} 3.42+1.82+1.82+0.72+0.12+0.42+0.42+1.32+2.72+2.72 \\ 0.72+0.0225+1.32+0.72+0.42+0.12+0.42+0.72+0.12+0.42 \end{pmatrix} = \begin{pmatrix} 1.725 \\ 0.558 \end{pmatrix}$$

$$var(c_2) = \frac{1}{8}\begin{pmatrix} 1.76+0.68+0.10+0.10+0.028+0.028+0.44+0.44+1.36 \\ 1.23+0.15+0.37+1.93+2.59+0.15+0.37+1.93+0.15 \end{pmatrix} = \begin{pmatrix} 0.625 \\ 1.11 \end{pmatrix}$$

سپس به محاسبه مقدار کوواریانس برای هر کلاس می‌پردازیم(در این مرحله از نوشتن محاسبات طولانی صرف نظر شده است):

$$cov_{xy}(c1) = \frac{1}{n-1}\sum_{i=1}^{n}(x_i-\mu_x)(y_i-\mu_y) = -0.116$$

$$cov_{xy}(c2) = \frac{1}{n-1}\sum_{i=1}^{n}(x_i-\mu_x)(y_i-\mu_y) = 0.2$$

در مرحله بعدی به محاسبه مقادیر احتمالات پیشین می‌پردازیم:

$$p(c_1) = \frac{10}{19} = 0.526$$

$$p(c_2) = \frac{9}{19} = 0.473$$

با استفاده از مقادیر محاسبه شده می‌توان ماتریس کوواریانس را برای هر دو کلاس تشکیل داد:

$$\Sigma_1 = \begin{bmatrix} 1.725 & -0.116 \\ -0.116 & 0.558 \end{bmatrix}$$

$$\Sigma_2 = \begin{bmatrix} 0.625 & 0.2 \\ 0.2 & 1.11 \end{bmatrix}$$

حال با استفاده از رابطه زیر می‌توان مرز تصمیم را به دست آورد:

$$g_i(x) = x^t W_i x + w_i^t x + \omega_{i0}$$

در ابتدا مقدار W_i در عبارت فوق را برای هر کلاس محاسبه می‌کنیم:

$$W_1 = -\frac{1}{2}\Sigma_1^{-1} = -\frac{1}{2}\begin{bmatrix} 0.579 & -8.62 \\ -8.62 & 1.79 \end{bmatrix} = \begin{bmatrix} 0.289 & -4.31 \\ -4.31 & 0.895 \end{bmatrix}$$

$$W_2 = -\frac{1}{2}\Sigma_2^{-1} = -\frac{1}{2}\begin{bmatrix}1.6 & 5\\ 5 & 0.9\end{bmatrix} = \begin{bmatrix}0.8 & 2.5\\ 2.5 & 0.45\end{bmatrix}$$

سپس به محاسبه w_i برای هر کلاس می‌پردازیم:

$$w_1 = \Sigma_1^{-1}mean(c1) = \begin{bmatrix}0.579 & -8.62\\ -8.62 & 1.79\end{bmatrix} * \begin{pmatrix}-0.15\\ -0.15\end{pmatrix} = \begin{pmatrix}1.2\\ 1.02\end{pmatrix}$$

$$w_2 = \Sigma_2^{-1}mean(c2) = \begin{bmatrix}1.6 & 5\\ 5 & 0.9\end{bmatrix} * \begin{pmatrix}1.33\\ 1.61\end{pmatrix} = \begin{pmatrix}10.17\\ 8.09\end{pmatrix}$$

حال مقادیر ω_{i0} را برای هر کلاس به دست می‌آوریم:

$$\omega_{10} = -\frac{1}{2}mean^t(c1)w1 - \frac{1}{2}\ln|\Sigma_1| + \ln P(c_1) = -\frac{1}{2}(-0.15, -0.15)\begin{pmatrix}1.2\\ 1.02\end{pmatrix} - \frac{1}{2} * \ln(0.95) + \ln(0.526) = \frac{0.27}{2} + \frac{0.05}{2} - 0.64 = -0.48$$

$$\omega_{20} = -\frac{1}{2}mean^t(c2)w2 - \frac{1}{2}\ln|\Sigma_2| + \ln P(c_1) = -\frac{1}{2}(1.33, 1.61)\begin{pmatrix}10.17\\ 8.09\end{pmatrix} - \frac{1}{2} * \ln(0.65) + \ln(0.473) = \frac{26.5}{2} + \frac{0.43}{2} - 0.74 = 12.725$$

حال تمامی پارامترهای مورد نیاز برای تشکیل معادله مذکور را در اختیار داریم و می‌توانیم معادلات را تشکیل دهیم:

$$g_1(x) = (x_1, x_2)\begin{bmatrix}0.289 & -4.31\\ -4.31 & 0.895\end{bmatrix}\begin{pmatrix}x_1\\ x_2\end{pmatrix} + (1.2, 1.02)\begin{pmatrix}x_1\\ x_2\end{pmatrix} - 0.48$$

$$g_2(x) = (x_1, x_2)\begin{bmatrix}0.8 & 2.5\\ 2.5 & 0.45\end{bmatrix}\begin{pmatrix}x_1\\ x_2\end{pmatrix} + (10.17, 8.09)\begin{pmatrix}x_1\\ x_2\end{pmatrix} + 12.725$$

معادله مرز تصمیم دو کلاس به شرح زیر خواهد بود:

$$(x_1, x_2)\begin{bmatrix}0.289 & -4.31\\ -4.31 & 0.895\end{bmatrix}\begin{pmatrix}x_1\\ x_2\end{pmatrix} + (1.2, 1.02)\begin{pmatrix}x_1\\ x_2\end{pmatrix} - 0.48 >^{w_1} (x_1, x_2)\begin{bmatrix}0.8 & 2.5\\ 2.5 & 0.45\end{bmatrix}\begin{pmatrix}x_1\\ x_2\end{pmatrix} + (10.17, 8.09)\begin{pmatrix}x_1\\ x_2\end{pmatrix} + 12.725$$

ب

مانند قسمت قبلی سوال مقادیر میانگین و کوواریانس محاسبه می‌شوند:

$$mean(c_1) = \frac{1}{10}\begin{pmatrix}-2 - 1.5 - 1.5 - 1 - 0.5 + 0.5 + 0.5 + 1 + 1.5 + 1.5\\ -1 + 0 + 1 - 1 + 0.5 - 0.5 + 0.5 - 1 - 0.5 + 0.5\end{pmatrix} = \begin{pmatrix}-0.15\\ -0.15\end{pmatrix}$$

$$mean(c_2) = \frac{1}{9}\begin{pmatrix}0 + 0.5 + 1 + 1 + 1.5 + 1.5 + 2 + 2 + 2.5\\ 0.5 + 2 + 1 + 3 + 0 + 2 + 1 + 3 + 2\end{pmatrix} = \begin{pmatrix}1.33\\ 1.61\end{pmatrix}$$

$$cov_{xy}(c1) = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \mu_x)(y_i - \mu_y) = -0.116$$

$$cov_{xy}(c2) = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \mu_x)(y_i - \mu_y) = 0.2$$

پ

تغییر مقدار احتمال پیشین تنها روی مقدار ω_{i0} تاثیر گذار خواهد بود و به شکل زیر درمی‌آید:

$$\omega_{10} = -\frac{1}{2}mean^t(c1)w1 - \frac{1}{2}\ln|\Sigma_1| + \ln P(c_1) = -\frac{1}{2}(-0.15, -0.15)\begin{pmatrix}1.2\\1.02\end{pmatrix} - \frac{1}{2} * \ln(0.5) + \ln(0.526) = \frac{0.27}{2} + \frac{0.05}{2} - 0.69 = -0.53$$

$$\omega_{20} = -\frac{1}{2}mean^t(c2)w2 - \frac{1}{2}\ln|\Sigma_2| + \ln P(c_1) = -\frac{1}{2}(1.33, 1.61)\begin{pmatrix}10.17\\8.09\end{pmatrix} - \frac{1}{2} * \ln(0.5) + \ln(0.473) = \frac{26.5}{2} + \frac{0.43}{2} - 0.69 = 12.775$$

و معادلات به شکل زیر درمی‌آیند:

$$g_1(x) = (x_1, x_2)\begin{bmatrix}0.289 & -4.31\\-4.31 & 0.895\end{bmatrix}\begin{pmatrix}x_1\\x_2\end{pmatrix} + (1.2, 1.02)\begin{pmatrix}x_1\\x_2\end{pmatrix} - 0.53$$

$$g_2(x) = (x_1, x_2)\begin{bmatrix}0.8 & 2.5\\2.5 & 0.45\end{bmatrix}\begin{pmatrix}x_1\\x_2\end{pmatrix} + (10.17, 8.09)\begin{pmatrix}x_1\\x_2\end{pmatrix} + 12.775$$

در نهایت معادله مرز تصمیم دو کلاس به شکل زیر خواهد بود:

$$(x_1, x_2)\begin{bmatrix}0.289 & -4.31\\-4.31 & 0.895\end{bmatrix}\begin{pmatrix}x_1\\x_2\end{pmatrix} + (1.2, 1.02)\begin{pmatrix}x_1\\x_2\end{pmatrix} - 0.53 >^{w_1} (x_1, x_2)\begin{bmatrix}0.8 & 2.5\\2.5 & 0.45\end{bmatrix}\begin{pmatrix}x_1\\x_2\end{pmatrix} + (10.17, 8.09)\begin{pmatrix}x_1\\x_2\end{pmatrix} + 12.775$$

ت

رابطه زیر را برای طراحی طبقه‌بند به همراه ریسک در اختیار داریم:

$$\frac{p(x|c_1)}{p(x|c_2)} >^{w_1} \frac{\lambda_{21} - \lambda_{22}}{\lambda_{12} - \lambda_{11}}\frac{p(c_1)}{p(c_2)}$$

$$\frac{p(x|c_1)}{p(x|c_2)} >^{w_1} \frac{a - 0}{2a - 0} * \frac{\frac{10}{19}}{\frac{9}{19}}$$

$$\frac{p(x|c_1)}{p(x|c_2)} >^{w_1} \frac{10}{18}$$

$$18 * p(x|c_1) >^{w_1} 10 * p(x|c_2)$$

با جایگزینی معادلات به دست آمده در بخش قبلی سوال در عبارت فوق، معادله مرز تصمیم با شرایط فوق به دست می‌آید:

$$18 * (x_1, x_2)\begin{bmatrix}0.289 & -4.31\\ -4.31 & 0.895\end{bmatrix}\begin{pmatrix}x_1\\ x_2\end{pmatrix} + (1.2,1.02)\begin{pmatrix}x_1\\ x_2\end{pmatrix} - 0.53 >^{w_1} 10 * (x_1, x_2)\begin{bmatrix}0.8 & 2.5\\ 2.5 & 0.45\end{bmatrix}\begin{pmatrix}x_1\\ x_2\end{pmatrix} + (10.17,8.09)\begin{pmatrix}x_1\\ x_2\end{pmatrix} + 12.775$$

ث

در این بخش نیز همانند بخش پ تغییر مقدار احتمال پیشین تنها روی مقدار ω_{i0} تاثیر گذار خواهد بود و به شکل زیر درمی‌آید:

$$\omega_{10} = -\frac{1}{2}mean^t(c1)w1 - \frac{1}{2}\ln|\Sigma_1| + \ln P(c_1) = -\frac{1}{2}(-0.15, -0.15)\begin{pmatrix}1.2\\ 1.02\end{pmatrix} - \frac{1}{2} * \ln(0.33) + \ln(0.526) = \frac{0.27}{2} + \frac{0.05}{2} - 1.1 = -0.94$$

$$\omega_{20} = -\frac{1}{2}mean^t(c2)w2 - \frac{1}{2}\ln|\Sigma_2| + \ln P(c_1) = -\frac{1}{2}(1.33,1.61)\begin{pmatrix}10.17\\ 8.09\end{pmatrix} - \frac{1}{2} * \ln(0.66) + \ln(0.473) = \frac{26.5}{2} + \frac{0.43}{2} - 0.41 = 13.05$$

$$g_1(x) = (x_1, x_2)\begin{bmatrix}0.289 & -4.31\\ -4.31 & 0.895\end{bmatrix}\begin{pmatrix}x_1\\ x_2\end{pmatrix} + (1.2,1.02)\begin{pmatrix}x_1\\ x_2\end{pmatrix} - 0.94$$

$$g_2(x) = (x_1, x_2)\begin{bmatrix}0.8 & 2.5\\ 2.5 & 0.45\end{bmatrix}\begin{pmatrix}x_1\\ x_2\end{pmatrix} + (10.17,8.09)\begin{pmatrix}x_1\\ x_2\end{pmatrix} + 13.05$$

معادله مرز تصمیم دو کلاس به شکل زیر به دست می‌آید:

$$(x_1, x_2)\begin{bmatrix}0.289 & -4.31\\ -4.31 & 0.895\end{bmatrix}\begin{pmatrix}x_1\\ x_2\end{pmatrix} + (1.2,1.02)\begin{pmatrix}x_1\\ x_2\end{pmatrix} - 0.94 >^{w_1} (x_1, x_2)\begin{bmatrix}0.8 & 2.5\\ 2.5 & 0.45\end{bmatrix}\begin{pmatrix}x_1\\ x_2\end{pmatrix} + (10.17,8.09)\begin{pmatrix}x_1\\ x_2\end{pmatrix} + 13.05$$

سوال ۴

آ

با توجه به توزیع احتمال داده شده و استقلال ویژگی‌ها تابع likelihood به شکل زیر محاسبه می‌شود:

$$p(x|\lambda_i) = p(x_1|\lambda_i)p(x_2|\lambda_i)\ldots p(x_j|\lambda_i) = \prod_{j=1}^{n} p(x_j|\lambda_i)$$

با جایگذاری عبارت توزیع پوآسون به جای عبارات مورد نیاز به تساوی زیر می‌رسیم:

$$p(x|\lambda_i) = \prod_{j=1}^{n}\left(\frac{\lambda_i^{x_j}e^{-\lambda_i}}{x_j!}\right) = \frac{\lambda_i^{\sum_{j=1}^{n}x_j}e^{-n\lambda_i}}{\prod_{j=1}^{n}x_j!}$$

سپس از طرفین معادله لگاریتم می‌گیریم تا تابع log likelihood به دست آید(برای ساده سازی مسئله، لگاریتم با پایه *e* یا همان *ln* گرفته می‌شود):

$$\log\big(p(x|\lambda_i)\big) = \ln\left(\frac{\lambda_i^{\sum_{j=1}^{n}x_j}ne^{-\lambda_i}}{\prod_{j=1}^{n}x_j!}\right) = \left(\sum_{j=1}^{n}x_j\right) * \ln(\lambda_i) - n\lambda_i - \sum_{j=1}^{n}\ln\,(x_j!)$$

برای به دست آوردن مقدار بیشینه تساوی فوق باید از آن نسبت به λ_i مشتق بگیریم و برابر با صفر قرار دهیم و در ادامه ساده می‌کنیم:

$$\frac{\mathrm{dlog}\big(p(x|\lambda_i)\big)}{d\lambda_i} = \frac{\sum_{j=1}^{n}x_j}{\lambda_i} - n = 0$$

$$\sum_{j=1}^{n}x_j = n\lambda_i$$

$$\lambda_i = \frac{\sum_{j=1}^{n}x_j}{n}$$

و در نهایت مقدار بیشینه λ_i که برابر با عبارت فوق است به دست می‌آید.

ب

مانند هر مسئله دیگری در ابتدا تساوی زیر را برای احتمال پسین در اختیار داریم:

$$\mathrm{p}(\lambda|\mathrm{D}) = \mathrm{p}(\mathrm{D}|\lambda)\mathrm{p}(\lambda)$$

سپس عبارات داده شده در بخش آ و ب را در تساوی فوق جایگزین می‌کنیم:

$$\mathrm{p}(\lambda|\mathrm{D}) = \frac{\lambda^{\sum_{j=1}^{n}x_j}e^{-n\lambda}}{\prod_{j=1}^{n}x_j!} * \mathrm{c}\lambda^{\alpha-1}\mathrm{e}^{-\beta\lambda} = c\,\frac{\lambda^{\sum_{j=1}^{n}x_j+\alpha-1}e^{-(n+\beta)\lambda}}{\prod_{j=1}^{n}x_j!}$$

با توجه به اینکه مقدار مخرج کسر فوق یک عبارت ثابت است، می‌توان این عبارت را به شکل زیر نیز بازنویسی کرد:

$$p(\lambda|D) = \frac{c}{\prod_{j=1}^{n} x_j!} \lambda^{\sum_{j=1}^{n} x_j + \alpha - 1} e^{-(n+\beta)\lambda}$$

قسمت کسری عبارت فوق یک مقدار ثابت است و به نظر می‌رسد که تابع احتمال به دست آمده توزیعی شبیه به توزیع گاما شباهت دارد.

پ

بله conjugate prior است چرا که برای احتمال پیشین توزیعی از خانواده گاما داشتیم و برای احتمال پسین نیز به توزیعی از خانواده گاما رسیدیم. کلمه خانواده بدین معنی است که نیاز نیست دو توزیع لزوما یکسان باشند و مقادیر پارامترها و مقدار ثابت دقیقا یکسان باشد و صرفا مهم است که بتوان به شکل توزیع یکسانی آن‌ها را رسم و تفسیر کرد.

ت

صورت سوال مقدار بیشینه توزیع گاما را به ما داده است و با جایگذاری پارامترهای به دست آمده برای توزیع گاما در بخش ب را در فرمول داده شده قرار می‌دهیم:

$$\lambda = \frac{\alpha - 1}{\beta} = \frac{\sum_{j=1}^{n} x_j + \alpha - 1}{n + \beta}$$

ث

بله اگر تعداد داده‌ها به بینهایت میل کند، تخمینگر *MAP* به *MLE* میل می‌کند.

ج

MLE زمانی استفاده می شود که بخواهیم پارامترهای یک توزیع احتمال را بر اساس مجموعه ای از داده‌های مشاهده شده تخمین بزنیم. هدف MLE یافتن مقادیر پارامتری است که تابع احتمال را به حداکثر می‌رساند و تابعی است که نشان می‌دهد داده‌های مشاهده شده چقدر با توزیع احتمال با پارامترهای مشخص داده شده مطابقت دارند. MLE اغلب زمانی استفاده می شود که هیچ اطلاعات قبلی در مورد پارامترهای توزیع موجود نباشد. تخمین MAP زمانی استفاده می شود که بخواهیم پارامترهای یک توزیع احتمال را بر اساس داده های مشاهده شده و برخی اطلاعات قبلی در مورد پارامترها تخمین بزنیم. هدف MAP یافتن مقادیر مناسب برای پارامترهایی است که توزیع احتمال پسین را به حداکثر می‌رساند.

سوال ۵

آ

طبقه‌بند naïve bayes

طبقه‌بند **naïve bayes** یک الگوریتم یادگیری ماشین مبتنی بر احتمالات است که از قانون بیز به منظور طبقه‌بندی داده‌ها استفاده می‌کند. نام "**Naïve**" که در نام این الگوریتم به چشم می‌خورد و در زبان فارسی به معنای "ساده" است این نکته را نشان می‌دهد که در پیاده‌سازی و طراحی این طبقه‌بندها از فرضیاتی ساده کننده استفاده شده است. این فرض ساده کننده، استقلال ویژگی‌های نمونه‌های داده از یکدیگر است که در بسیاری از مواقع در واقعیت صحیح نیست. البته اینکه در این طبقه‌بند از ساده‌سازی به کمک مفروضاتی غیر واقعی استفاده شده است، به معنی ضعف عملکرد این طبقه‌بندها نیست. طبقه‌بند **naïve bayes** در مواقعی که تعداد نمونه‌های مجموعه داده به اندازه‌ای نیست که امکان استفاده از شبکه‌های عصبی با نتیجه‌ای مطلوب را به ما بدهد، عملکرد نسبتا خوبی می‌تواند از خود نشان دهد و نتایج قابل قبولی ارائه کند. از جمله کاربردهایی که این طبقه‌بند بیشترین استفاده را در آن‌ها دارد، پردازش متن و وظایفی مانند تشخیص پیام اسپم است. این طبقه‌بند دارای فاز آموزش است و برای آموزش نیاز به تعدادی داده با برچسب مشخص کننده کلاس وجود دارد. نحوه کار این الگوریتم به شرح زیر است:

۱. ابتدا با توجه به تعداد دادگان مربوط به هر کلاس در مجموعه داده آموزش، احتمال پیشین محاسبه می‌شود که نشان می‌دهد هر داده بدون توجه به مقادیر ویژگی‌ها، چقدر احتمال دارد که به هر یک از کلاس‌ها تعلق داشته باشد.
۲. سپس الگوریتم با توجه به توزیع احتمالی انتخاب شده پیش از آموزش، مقدار پارامترهای توزیع هر ویژگی را برای هر کلاس با توجه به داده‌های آموزش محاسبه می‌کند.
۳. در مرحله تست یا کاربرد، احتمال تعلق هر ویژگی نمونه جدید به توزیع مربوطه از هر کلاس محاسبه می‌شود، سپس با ضرب احتمالات مربوط به هر کلاس در یکدیگر و ضرب در احتمال پیشین مربوط به آن کلاس، احتمال تعلق نمونه به هر کلاس به دست می‌آید و کلاسی با بیشترین احتمال به عنوان برچسب پیش‌بینی شده انتخاب می‌شود.

تفاوت طبقه‌بند بیز و naïve bayes

طبقه‌بند بیز می‌تواند بهترین عملکرد را بین طبقه‌بندهای متفاوت ارائه دهد و به عنوان طبقه‌بند بهینه عمل کند اما این طبقه مشکلاتی را به همراه دارد. مشکلات این طبقه‌بند به شرح زیر می‌باشد:

۱. این طبقه‌بند نیاز به تعداد زیادی داده برای فاز آموزش دارد تا احتمال شرطی ویژگی‌های متفاوت را برای هر کلاس به دست آورد. اگر تعداد داده کم باشد احتمالات به دست آمده دقیق نخواهند بود.
۲. با افزایش تعداد ویژگی‌ها، پیچیدگی مدل بالا می‌رود و باید تعداد حالات مختلفی از ترکیب ویژگی‌ها با یکدیگر را بررسی کند.

این مشکلات باعث می‌شود تا نیاز به یک طبقه‌بند ساده‌تر برای حالاتی که تعداد داده آموزش کم است یا تعداد ویژگی‌ها بسیار زیاد است، حس شود. طبقه‌بند **naïve bayes** فرض‌های ساده کننده‌ای را بر روی طبقه‌بند بیز در نظر گرفت تا حالاتی را طبقه‌بند قبلی نمی‌توانست پوشش دهد را جبران کند. همچنین یک طبقه‌بند ساده‌تر و سبک‌تر به وجود آمد که کاربردهای گسترده‌ای دارد. به طور کلی این فرض‌ها به منظور تخمین ساده‌تر احتمالات و کاهش بار محاسباتی در نظر گرفته شده‌اند. مهمترین فرضی که باعث تمایز این دو طبقه‌بند می‌شود، فرض استقلال ویژگی‌ها[۳] در **naïve bayes** است که توزیع احتمال هر ویژگی در هر کلاس را مستقل از سایر ویژگی‌ها در نظر می‌گیرد. فرض دیگری که در این طبقه‌بند مورد نظر قرار می‌گیرد تساوی اهمیت ویژگی‌های متفاوت است.

[3] IID

هزینه و مواقع استفاده از naïve bayes

Naïve Bayes فرض می‌کند که ویژگی‌ها در هر کلاس مستقل از یکدیگر هستند، در صورتی که اگر ویژگی‌ها واقعاً همبستگی داشته باشند، می‌تواند یک محدودیت مهم باشد و همچنین یک فرض غلط را در نظر گرفته‌ایم. در چنین مواردی، انواع دیگر طبقه‌بندها مانند درخت تصمیم[۴]، جنگل‌های تصادفی[۵] یا ماشین بردار پشتیبان[۶] که این فرض را ندارند، ممکن است عملکرد بهتری داشته باشند.

طبقه‌بند naïve bayes در مواقع متفاوتی می‌تواند مورد استفاده قرار بگیرد و بر سایر طبقه‌بندها ارجحیت داشته باشد. چند مورد از این مواقع به شرح زیر می‌باشند:

۱. تعداد زیاد ویژگی‌ها: naïve bayes می‌تواند مجموعه‌های داده با تعداد زیادی ویژگی را مدیریت کند، و می‌توان آن را به سرعت روی چنین مجموعه‌داده‌هایی آموزش داد.
۲. استقلال ویژگی‌ها: این طبقه‌بند فرض می کند که ویژگی ها با توجه به کلاس مستقل از یکدیگر هستند. اگر این فرض درست باشد، naïve bayes می تواند در طبقه‌بندی داده ها موثر باشد.
۳. داده‌های آموزش محدود: زمانی که داده‌های آموزشی محدود است این طبقه‌بند می‌توان عملکرد خوبی داشته باشد و زمانی مفید است که جمع‌آوری تعداد زیادی داده‌ی برچسب‌گذاری شده امکان پذیر نباشد.
۴. سرعت بالا: در کاربردهایی که به پیش‌بینی با سرعت بالا نیاز داریم این طبقه‌بند می‌تواند گزینه مناسبی باشد چرا که دارای حجم محاسبات نسبتا کمی است و می‌تواند سرعت بالاتری را نسبت به دیگر طبقه‌بندها ارائه دهد.

ب

ابتدا داده‌ها به کمک قطعه کد زیر و با استفاده از کتابخانه pandas در محیط برنامه بارگذاری می‌شوند:

```
data = pd.read_csv('/content/drive/MyDrive/ML_HW1_Data/penguins.csv')
```

پیش‌پردازش

داده‌های این سوال از چند جنبه نیاز به پیش‌پردازش دارند. هر کدام از این جنبه به همراه روش حل آن‌ها در ادامه شرح داده شده‌اند:

۱. وجود داده‌هایی با مقدار غیرمعتبر: تعداد ۶ سطر از داده‌های موجود در این مجموعه داده در همه‌ی ستون‌ها دارای مقدار غیر معتبر x بود و با یافتن سطرهای شامل این مقادیر و سپس حذف این سطرها، این مشکل حل شد. قطعه کد مربوط به این بخش در ادامه الصاق شده است:

```
data = data.drop(data.index[data['culmen_length_mm']=='x'].tolist())
```

۲. به عنوان بخشی از پیش‌پردازش و به منظور تسهیل بخش‌های آتی، برچسب‌های متفاوت دادگان و نام ویژگی‌ها را در دو لیست جدا به کمک قطعه کد زیر ذخیره می‌کنیم:

```
labels = list(set(data['species']))
features = ['culmen_length_mm', 'culmen_depth_mm', 'flipper_length_mm',
'body_mass_g']
```

[4] Decision tree
[5] Random forest
[6] Support vector machine(svm)

۳. در این بخش نیز برچسب‌های عددی برای هر نمونه انتخاب می‌شود تا تحلیل و تفسیر نتایج تسهیل شود:

```
numerical_label = list()

for i in range(data.shape[0]):
    numerical_label.append(labels.index(data.iloc[i]['species']))
data['label'] = numerical_label
```

۴. نوع داده‌ای ویژگی‌ها: ویژگی‌های موجود در این مجموعه داده از نوع عدد بودند اما نوع داده‌ای رشته برای آن‌ها انتخاب شده بود و به عنوان عدد قابل پردازش نبودند. بنابراین پیش از اعمال الگوریتم naïve bayes باید نوع داده‌ای آن‌ها به اعداد اعشاری تغییر می‌کرد. نحوه انجام این کار در قطعه کد زیر قابل مشاهده است:

```
for col in features:
    data[col] = [float(i) for i in list(data[col])]
```

مرحله آموزش

برای پیاده‌سازی الگوریتم naïve bayes ابتدا نیاز داریم تا تعدادی از توابع پایه را پیاده‌سازی کنیم. این توابع شامل موارد زیر است:

۱. تقسیم مجموعه داده به تست و آموزش: در این تابع داده‌ها به صورت تصادفی با احتمالی به اندازه مشخص شده توسط متغیر train_size به مجموعه آموزش اختصاص داده می‌شوند و با احتمالی برابر با `1-train_size` به مجموعه تست اختصاص داده می‌شوند. این تابع ممکن است مجموعه آموزشی دقیقا به اندازه متغیر مشخص شده ایجاد نکند.

```
from random import randint
def train_test_split(data, train_size):
    data = data.values.tolist()
    train = list()
    test = list()
    for i in range(len(data)):
        if randint(0,10)/10 > train_size:
            test.append(data[i])
        else:
            train.append(data[i])
    return train, test
```

۲. دسته‌بندی داده‌ها بر اساس برچسب: برای یافتن توزیع متغیرها ابتدا نیاز است تا داده‌ها بر اساس برچسب هر نمونه دسته‌بندی شود. این عمل با استفاده از دیکشنری‌های زبان پایتون در قطعه کد زیر انجام گرفته است:

```
def group_by_label(data):
    pdata = dict()
    for l in range(len(labels)):
        pdata[l] = list()
    for i in range(data.shape[0]):
        pdata[data.iloc[i]['label']].append(data.iloc[i][features].tolis
t())
    return pdata
```

۳. توابع احتمالاتی: با توجه به ماهیت احتمالاتی الگوریتم naïve bayes، برای پیاده‌سازی این الگوریتم نیاز به توابعی مانند میانگین و انحراف معیار داریم که در قطعه کد زیر نحوه تعریف آن‌ها مشخص شده است:

```
def mean(x):
    return (1/len(x))*sum(x)
def stddev(x):
    m = mean(x)
    x2 = [(i-m)**2 for i in x]
    return (sum(x2)/(len(x)-1))**0.5
```

۴. محاسبه احتمال پیشین: بر اساس تعداد نمونه‌های هر کلاس در مجموعه داده آموزش، احتمال پیشین محاسبه می‌شود. تابع محاسبه احتمال پیشین در قطعه کد زیر قابل مشاهده است:

```
def prior(data, labels, n):
    pci = dict()
    for i in range(len(labels)):
        pci[i] = len(data[i])/n
    return pci
```

۵. تابع چگالی احتمال گاوسی: در قطعه کد زیر تابع چگالی احتمال توزیع گاوسی پیاده شده است:

```
def probability(x, mean, stddev):
    exp = e**( (-0.5) * ( ( (x-mean) /stddev) ** 2) )
    return ( 1 / (stddev * ( 2 * pi ) ** 0.5 ) ) * exp
```

پس از تعریف توابع مورد نیاز، هر کدام در زمان مناسب فراخوانی می‌شوند تا عملیات مورد نظر بر روی داده‌ها انجام گیرد. ابتدا داده‌ها به دو مجموعه آموزش و تست تقسیم می‌شوند. در مرحله دوم داده‌های آموزش گروه بندی می‌شوند، سپس مقدار احتمال پیشین برای هر یک از کلاس‌ها محاسبه می‌شود. پس از انجام عملیات فوق، مقدار میانگین و انحراف معیار برای هر یک از ویژگی‌های هر کلاس به کمک قطعه کد زیر محاسبه می‌شود.(در مجموع ۱۲ میانگین و ۱۲ انحراف معیار محاسبه می‌شود)

```
classes_meanstd = dict()

for k in gtrain.keys():
    temp = list()
    for i in range(4):
        a = mean([item[i] for item in gtrain[k]])
        b = stddev([item[i] for item in gtrain[k]])
        temp.append([a,b])
    classes_meanstd[k] = temp
```

این محاسبات به منزله مرحله آموزش الگوریتم naïve bayes در نظر گرفته می‌شود. این جمله بدین معنی است که با توجه به اینکه به صورت پیش‌فرض توزیع مقادیر ویژگی‌های هر کلاس، گاوسی در نظر گرفته شده است، و با داشتن مقادیر میانگین و انحراف معیار، توزیع گاوسی قابل تشخیص و نمودار آن قابل رسم است، پس مرحله آموزش به پایان رسیده است.

مرحله تست

در مرحله بعدی برای هر نمونه موجود در مجموعه تست باید احتمال تعلق به هر یک از سه کلاس محاسبه شود. احتمال تعلق یک نمونه به هر کلاس بدین شکل محاسبه می‌شود که احتمال تعلق هر کدام از ویژگی‌های نمونه به توزیع به دست آمده ویژگی متناظر با آن در کلاس مربوطه محاسبه می‌شود و با ضرب احتمال پیشین کلاس مورد نظر در احتمالات محاسبه شده برای هر ویژگی در یک کلاس احتمال تعلق آن نمونه به کلاس مشخص شده محاسبه می‌شود. عملیات فوق در قطعه کد زیر قابل مشاهده است:

```
def calc_prob(classes_meanstd, x):

    prob = dict()
    for k in classes_meanstd.keys():
        temp = 1
        i = 0
        for attr in classes_meanstd[k]:
            temp *= probability(x[i], attr[0], attr[1])
            i += 1
        prob[k] = temp
    return prob
```

خروجی تابع فوق سه احتمال به ازای هر نمونه است که هر یک از آن‌ها احتمال تعلق نمونه x به کلاس iام را نشان می‌دهد. با استفاده از تابع argmax تعریف شده در فایل کد، کلاسی با بیشترین احتمال به عنوان برچسب نمونه x انتخاب می‌شود. تابع زیر عملیات فوق را برای تمامی نمونه‌های موجود در مجموعه تست انجام می‌دهد:

```
def predict(classes_meanstd, test):

    predicted = list()
    for x in test:
        prob = calc_prob(classes_meanstd, x)
        print(prob)
        predicted.append(argmax(prob))
    return predicted
pred = predict(classes_meanstd, x_test)
```

تابع فوق به ازای هر نمونه یک برچسب پیش‌بینی می‌کند و لیستی از برچسب‌ها را بازمی‌گرداند. این لیست آماده برای مرحله ارزیابی است.

ارزیابی

در این مرحله ابتدا ماتریس آشفتگی ۳ در ۳ برای مجموعه تست به کمک قطعه کد زیر محاسبه می‌شود.

```
def confusion_matrix(y, pred):
    cm = np.zeros([3,3])
    for i in range(len(y)):
        cm[y[i],pred[i]] += 1
    return cm
cm = confusion_matrix(y_test,pred)
```

خروجی این تابع برای مجموعه تست استفاده شده به شکل زیر خواهد بود:

	Y_pred = 2	Y_pred = 1	Y_pred = 0
y_true = 0	0	0	21
Y_true = 1	0	22	0
Y_true = 2	13	0	1

سپس به کمک تابع زیر، ماتریس آشفتگی 1 vs all برای هر کلاس ایجاد می‌شود:

```
def onevsallcm(cm, c):
    result = np.zeros([2,2])
    result[0,0] = cm[c,c]
    for i in range(3):
        if not i == c:
            result[0,1] += cm[c,i]
            result[1,0] += cm[i,c]
    result[1,1] = np.sum(cm) - result[0,0] - result[0,1] - result[1,0]
    return result
```

نتایج این تابع برای هر کلاس به شرح زیر می‌باشد(جدول‌ها به ترتیب شماره کلاس‌ها می‌باشد):

	Y_pred = +	Y_pred = -
y_true = +	21	0
Y_true = -	1	35

جدول ۱: کلاس ۰

	Y_pred = +	Y_pred = -
y_true = +	22	0
Y_true = -	0	35

جدول ۱: کلاس ۱

	Y_pred = +	Y_pred = -
y_true = +	13	1
Y_true = -	0	43

جدول ۳: کلاس ۲

سپس با استفاده از این ماتریس‌های آشفتگی 1 vs all سایر معیارهای ارزیابی مانند precision، recall و accuracy برای هر کلاس قابل محاسبه است. این عمل به کمک تابع زیر انجام می‌گیرد:

```python
def evaluate(cm):

    metrics = dict()
    for i in range(cm.shape[0]):
        metrics[i] = dict()
        cmi = onevsallcm(cm,i)
        p = cmi[0,0] / (cmi[0,0] + cmi[1,0])
        r = cmi[0,0] / (cmi[0,0] + cmi[0,1])
        acc = (cmi[0,0]+cmi[1,1]) / np.sum(cmi)
        metrics[i]['Precision'] = p
        metrics[i]['Recall'] = r
        metrics[i]['Accuracy'] = acc
        metrics[i]['Confusion Matrix'] = cmi
    return metrics
```

خروجی قطعه کد فوق به شرح زیر می‌باشد:

class	Precision	Recall	Accuracy
0	0.954	1	0.952
1	1	1	1
2	1	0.928	0.982

طبقه‌بند طراحی شده برای کلاس ۱ که داده بیشتری از سایر کلاس‌ها دارد به صورت بی‌نقص عمل می‌کند و بهترین نتایج را برای تمامی معیارهای ارزیابی ارائه می‌دهد، اما برای کلاس ۲ که دارای تعداد داده کمی است دچار مشکل می‌شود و تعدادی از داده‌هایی که در این کلاس قرار دارند را به اشتباه به کلاس ۰ انتساب می‌دهد که باعث کاهش precision کلاس ۰ و در عین حال کاهش recall کلاس ۲ می‌شود. در همین حال، recall کلاس ۰ و precision کلاس ۲ دارای مقدار ۱ هستند که به معنی این هست که تمامی نمونه‌های منتسب به کلاس ۲ در دادگان طلایی نیز دارای برچسب ۲ هستند و همچنین تمامی نمونه‌های کلاس ۰ در دادگان طلایی به درستی با برچسب ۰ تشخیص داده شده‌اند و تنها تعدادی از داده‌های کلاس ۲ که برچسب ۰ را دریافت کرده‌اند باعث کاهش مقادیر معیارهای ارزیابی مذکور در دو کلاس ۰ و ۲ شده‌اند.

پ

در این بخش عملیات بخش ب به کمک کتابخانه scikit-learn انجام گرفته است. ابتدا داده‌ها به شکل زیر به دو مجموعه آموزش و تست تقسیم می‌شوند:

```python
X_train, X_test, y_train, y_test = train_test_split(data[features], data['label'])
```

در مرحله بعدی مدل ایجاد می‌شود و به کمک داده‌های آموزش، آموزش می‌بینند:

```python
model = GaussianNB()

model = model.fit(X_train, y_train)
```

سپس برچسب داده‌های تست پیش‌بینی می‌شود:

```
sklearn_pred = model.predict(X_test)
```

در نهایت به شکل زیر ارزیابی می‌شود و نتایج گزارش می‌شود:

```
cr = classification_report(y_test, sklearn_pred)
```

نتایج ارزیابی این مدل در جدول زیر قابل مشاهده است:

class	Precision	Recall	Accuracy
0	1	1	1
1	1	0.94	0.98
2	0.975	1	0.988

دو طبقه‌بند طراحی شده از ابتدا و طبقه‌بند آماده‌ی کتابخانه sklearn حد زیادی مشابه یکدیگر عمل می‌کنند و دقتی بسیار نزدیک ارائه می‌کنند اما به هر حال تفاوت جزئی بین نتایج دو طبقه‌بند وجود دارد که می‌تواند به دلیل وجود تفاوت در تعداد و انتخاب داده‌های تست و آموزش در دو طبقه‌بند متفاوت باشد. در طبقه‌بندی که از ابتدا و بدون استفاده از کتابخانه‌های مرتبط ایجاد شده است، تقسیم داده‌ها به مجموعه تست و آموزش به صورت متفاوتی با کتابخانه آماده انجام می‌گیرد و در نتیجه می‌تواند به نتایج متفاوت منجر شود. در صورت استفاده از داده‌های یکسان برای دو طبقه‌بند، به احتمال زیاد شاهد نتایج کاملا یکسان از هر دو می‌بودیم.

سوال ۶

برای طراحی طبقه‌بند تصاویر مد نظر این سوال با چالش‌های خاصی روبرو هستیم. در ابتدا با کمک دستور زیر نام تصاویر موجود در پوشه مربوطه را بازیابی می‌کنیم:

```
mypath = '/content/drive/MyDrive/ML_HW1_Data/image/'

images = next(walk(mypath), (None, None, []))[2]
```

سپس هر کدام از تصاویر را با کمک کتابخانه imageio می‌خوانیم و با فرمت آرایه‌های کتابخانه numpy در یک لیست ذخیره می‌کنیم. همچنین از حرف اول نام هر تصویر به عنوان برچسب آن تصویر انتخاب می‌شود:

```
data = [np.array(iio.imread(mypath+name)) for name in images]

labels = [name[0] for name in images]
```

سپس به شکل زیر داده‌ها به دو مجموعه آموزش و تست تقسیم می‌شوند:

```
X_train, X_test, y_train, y_test = train_test_split(data, labels)
```

در مرحله بعد ویژگی‌هایی از هر تصویر استخراج می‌شود. برای طراحی این طبقه بند ویژگی‌های میانگین، انحراف معیار، بیشینه و کمینه هر بعد از سه بعد قرمز، آبی و سبز تصاویر بررسی شدند که در نهایت در حالتی که ویژگی‌های بیشینه و میانگین ابعاد به طور همزمان مورد استفاده قرار گرفتند عملکرد بهتری را از سایر حالات مشاهده کردیم.

```
def extract_feature(X):

  X_f = list()
  for pic in X:
    sample = list()
    for i in range(3):
      sample.append(np.mean(pic[:,:,i]))
      #sample.append(np.min(pic[:,:,i]))
      sample.append(np.max(pic[:,:,i]))
      #sample.append(np.std(pic[:,:,i]))
    X_f.append(sample)
  return X_f
```

در نهایت یک مدل naïve bayes گاوسی ایجاد شد و با ویژگی‌های استخراج شده آموزش دید و برچسب مناسب را برای دادگان تست پیش‌بینی کرد:

```
model = GaussianNB()

model = model.fit(X_train_f, y_train)

pred = model.predict(X_test_f)
```

در نهایت با استفاده از توابع زیر مدل آموزش دیده را ارزیابی می‌کنیم:

```
cr = classification_report(y_test, pred)
confusion_matrix(y_test, pred)
```

ماتریس آشفتگی به دست آمده برای این مدل به شکل زیر می‌باشد:

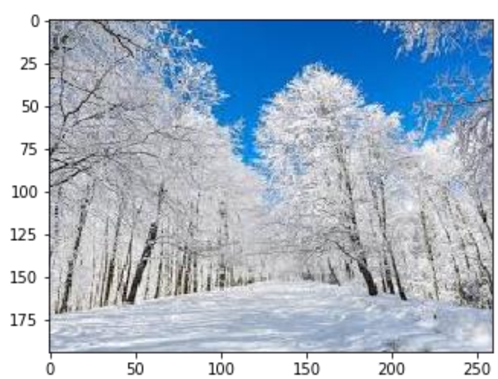
	Y-pred = j	Y_pred = s
Y-true = j	9	1
Y_true = s	0	11

دیگر معیارهای ارزیابی به شرح زیر هستند:

	Precision	Recall	Accuracy
Class = j	1	0.9	-
Class = s	0.92	1	-
Overall	0.96	0.95	0.95

همانطور که مشاهده می‌شود با توجه به تعداد کم داده‌های در دسترس(۸۴ نمونه) و ویژگی‌های نه چندان قدرتمند استخراج شده از هر عکس، مدل عملکرد مناسبی از خود نشان داده است که از ویژگی‌های مهم طبقه‌بند naïve bayes استفاده شده برای این مدل است.

با بررسی ماتریس آشفتگی متوجه می‌شویم که تنها یک داده از کلاس j به اشتباه در کلاس s تشخیص داده شده است. این نمونه در تصویر زیر قابل مشاهده است:



این تصویر بر خلاف تعداد زیادی از تصاویر موجود در کلاس j، رنگ بندی سبز و زرد ندارد و همانند تعداد زیادی از نمونه‌های کلاس s، دارای رنگ‌بندی سفید و آبی است که با توجه به انتخاب ویژگی‌ها بر اساس ابعاد رنگی هر تصویر و عدم توجه به سایر جزئیات، این خطا می‌تواند قابل توجیه باشد.