به نام خدا

دانشگاه تهران

دانشکدگان فنی

دانشکده مهندسی برق و کامپیوتر

یادگیری ماشین

تمرین دوم

نام و نام خانوادگی : حسین سیفی

شماره دانشجویی : ۸۱۰۱۰۰۳۸۶

اردی‌بهشت ۱۴۰۲

سوال ۱

الف

با توجه به مقدار داده شده برای y، می‌توان با تغییر شکل معادله و کم کردن مقدار β_0 از طرفین معادله به عبارت زیر برای مقدار خطا رسید:

$$\varepsilon = y - \beta_0$$

با استفاده از عبارت فوق، می‌توان مقدار مجموع مربعات خطا را به شکل زیر نوشت:

$$\sum \varepsilon^2 = \sum (y - \beta_0)^2$$

برای به دست آوردن بهترین مقدار β باید حداقل مقدار خطا را برای عبارت فوق یافت. برای انجام این کار باید نقطه بحرانی تابع فوق را به دست آورد. این عمل بدین شکل انجام می‌گیرد که از تابع فوق نسبت به β_0 مشتق می‌گیریم و برابر صفر قرار می‌دهیم و سپس با جایگذاری مقادیر معلوم، مقدار β_0 را به دست می‌آوریم. محاسبات توصیف شده به شکل زیر انجام می‌گیرد:

$$\frac{d}{d\beta_0} \sum (y - \beta_0)^2 = 0$$

مشتق حاصل مجموع چند عبارت را می‌توان به صورت مجموع مشتق هر کدام از عبارات نوشت:

$$\sum \frac{d}{d\beta_0} (y - \beta_0)^2 = 0$$

مشتق تابع فوق به شکل زیر به دست می‌آید:

$$\sum -2 * (y - \beta_0) = 0$$

به دلیل بی تاثیر بودن مقدار ۲-، آن را از عبارت فوق حذف می‌کنیم و پس از انجام تغییرات مورد نیاز، آن را به شکل زیر بازنویسی می‌کنیم:

$$\sum y = \sum \beta_0$$

در عبارت سمت راست تساوی، β_0 به تعداد نمونه‌های موجود در مجموعه داده‌های آموزش با خود جمع می‌شود. اگر تعداد این داده‌ها را برابر با n در نظر بگیریم، عبارت فوق پس از بازنویسی با شکل زیر درمی‌آید:

$$\sum y = n * \beta_0$$

در نهایت تساوی شکل زیر را به خود می‌گیرد:

$$\beta_0 = \frac{\sum y}{n}$$

برای به دست آوردن مقدار β_0 برای نمونه‌های داده شده، به شکل زیر اعداد داده شده را در عبارت فوق جایگذاری می‌کنیم:

$$\beta_0 = \frac{34 + 47 + 66 + 52 + 80 + 35 + 66 + 48 + 87 + 81}{10} = \frac{596}{10} = 59.6$$

ب

همانند بخش قبلی همین سوال، با تغییر شکل و جابجایی عبارت داده شده در صورت سوال می‌توان به عبارت زیر رسید:

$$\varepsilon = y - \beta_1 x$$

و سپس معادله مجموع مربعات خطا را به شکل زیر به دست آوردیم:

$$\sum \varepsilon^2 = \sum (y - \beta_1 x)^2$$

همانند بخش قبل از مقدار تابع مجموع مربعات خطا نسبت به β_1 مشتق گرفته می‌شود و برابر صفر قرار می‌گیرد تا بهترین مقدار را برای این پارامتر به دست آید:

$$\sum \frac{d}{d\beta_1} (y - \beta_1 x)^2 = 0$$

مشتق عبارت فوق به شکل زیر به دست می‌آید و در ادامه ساده‌سازی انجام می‌گیرد:

$$\sum -2x(y - \beta_1 x) = 0$$

از مقدار ثابت ۲- صرف نظر می‌کنیم(که البته تاثیرگذار هم نیست) و با اعمال تغییرات مورد نیاز، عبارت به شکل زیر نوشته می‌شود:

$$\sum xy = \beta_1 \sum x^2$$

با توجه به اینکه β_1 دارای مقدار ثابت است این امکان وجود داشت تا آن را از سیگما بیرون آوریم و در نهایت عبارت زیر به دست می‌آید:

$$\beta_1 = \frac{\sum xy}{\sum x^2}$$

و با جایگذاری مقادیر نمونه‌های داده شده در عبارت فوق، مقدار β_1 به دست می‌آید:

$$\beta_1 = \frac{(34 * 5) + (47 * 11) + (66 * 18) + (52 * 15) + (80 * 21) + (35 * 6) + (66 * 17) + (48 * 10) + (87 * 24) + (81 * 19)}{5^2 + 11^2 + 18^2 + 15^2 + 21^2 + 6^2 + 17^2 + 10^2 + 24^2 + 19^2}$$

$$\beta_1 = \frac{9774}{2498} = 3.91$$

ج

معادله اول خط فیت شده به کمک رگرسیون را نمایش می‌دهد و رابطه خطی بین متغیر داده شده(X که در مثال فوق همان ساعت مطالعه است) مقدار فیت شده(که برای داده‌های جدید و خارج از مجموعه آموزش به عنوان پیش‌بینی عمل می‌کند) نشان می‌دهد در حالی که معادله

دوم خطی است که رابطه بین جفت X و Yهای مشاهده شده در مجموعه داده آموزش(همان ساعت مطالعه و نمره در مثال فوق) را مدل می‌کند. همچنین در معادله اول همه نمونه‌ها روی یک خط مستقیم قرار نمی‌گیرند و مقادیر b_1 و b_0 با استفاده از مقادیر نمونه‌ها به دست می‌آید و برای ما مقادیری شناخته شده هستند. اما در معادله دوم β_1 و β_0 مقادیر شناخته شده نیستند.

د

همانطور که می‌دانیم بهترین تخمین واریانس با معیار mse در توزیع گوسی منتج به رابطه زیر برای واریانس می‌شود.

$$\sigma^2 = \frac{\sum_{i=1}^{n}(x_i - \mu)^2}{n\_1}$$

برای استفاده از فرمول فوق ابتدا نیاز به محاسبه میانگین داده‌ها داریم:

$$\mu = \frac{34 + 47 + 66 + 52 + 80 + 35 + 66 + 48 + 87 + 81}{10} = 59.6$$

سپس از فرمول معرفی شده برای محاسبه واریانس استفاده می‌کنیم:

$$\sigma^2 = \frac{(34 - 59.6)^2 + (47 - 59.6)^2 + (66 - 59.6)^2 + (52 - 59.6)^2 + (80 - 59.6)^2 + (35 - 59.6)^2 + (66 - 59.6)^2 + (48 - 59.6)^2 + (87 - 59.6)^2 + (81 - 59.6)^2}{9}$$

$$= \frac{3318.4}{9} = 368.71$$

ه

پیش‌بینی دقیق نمره بر اساس دادگان دیده شده، به داده‌های بسیار بیشتری از ۱۰ نمونه احتیاج دارد اما حتی اگر بیشترین تعداد داده ممکن نیز برای یادگیری پارامترهای مدل استفاده شود، باز هم نمی‌توان انتظار داشت که نمره متناظر با هر ساعت مطالعه را به صورت دقیق پیش‌بینی کند. همچنین با توجه به اینکه نقاط داده شده بر روی یک خط قرار ندارند و در بخش الف و ب همین سوال قصد داریم برازش یک خط را بر روی دادگان داده شده انجام دهیم، حتی انتظار دقت ۱۰۰ درصد را بر روی دادگان آموزشی نیز نمی‌توان داشت.(برای مثال سه نمونه (۱۰و۴۸)و (۱۱و۴۷) و (۵و۳۴) بر روی یک خط قرار ندارند.)

سوال ۲

الف

همانطور که می‌دانیم دو منبع خطای بایاس و واریانس برای یک مدل یادگیری ماشین وجود دارند. اگر مدل مورد نظر بیش از اندازه ساده در نظر گرفته شود، برازش داده‌ها بر اساس آن پیش‌فرض انجام می‌گیرد و می‌تواند باعث Underfit شدن مدل شود که باعث افزایش خطای بایاس و کاهش خطای واریانس می‌شود و اگر مدل بیش از اندازه پیچیده در نظر گرفته شود و بر اساس آن برازش انجام گیرد، امکان Overfit شدن وجود دارد و می‌تواند باعث افزایش خطای واریانس و کاهش خطای بایاس شود. برای جلوگیری از Overfit شدن یا Underfit شدن مدل یادگیری ماشین ایجاد شده، باید مقدار پیچیدگی مدل در حدی در نظر گرفته شود که مجموع خطای واریانس و بایاس به حداقل مقدار خود برسد. برای انجام این کار گاهی مدل تا حد زیادی پیچیده در نظر گرفته می‌شود و در مواقع مورد نیاز با استفاده از پارامترهایی خاص، پیچیدگی مدل کاهش می‌یابد. این کاهش پیچیدگی به کمک محدودیت‌هایی که بر روی مقادیر بردار وزن‌ها در نظر گرفته می‌شود انجام می‌گیرد. دو رابطه ریاضی زیر L1 Regularization و L2 Regularization را در نشان می‌دهند:

L1 Regularization(Lasso):

$$w^* = argmin_w \left\{ \sum_{j=1}^{n} \left(\left(\sum_{i=0}^{m} w_i x^i \right) - y_j \right)^2 + \lambda \left|\left| w \right|\right|_1 \right\}$$

L2 Regularization(Ridge Regression):

$$w^* = argmin_w \left\{ \sum_{j=1}^{n} \left(\left(\sum_{i=0}^{m} w_i x^i \right) - y_j \right)^2 + \lambda \left|\left| w \right|\right|_2^2 \right\}$$

در عبارات فوق، پارامتر λ مقدار آزادی بردار وزن‌ها را نشان می‌دهد. در ادامه تفاوت‌های دو عبارت فوق را بررسی می‌کنیم:

۱. تفاوت بزرگ دو عبارت استفاده از نرم ۱ در L1 Regularization و نرم ۲ در L2 Regularization است.
۲. در پاسخ L1 Regularization تعداد زیادی از درایه‌ها مقداری برابر با صفر دارند یعنی ماتریس w اسپارس است در صورتی که در L2 Regularization اینطور نیست و نمی‌توان ضرایبی برابر با صفر داشت.
۳. L2 Regularization فرم بسته دارد اما L1 Regularization فرم بسته ندارد.
۴. L1 برای داده‌های کم و L2 برای داده‌های متراکم مناسب‌تر است.
۵. L1 در مقابل Outlierها نسبت به L2 مقاوم‌تر است.

ب

فرم Quadratic عبارت موجود در صورت سوال که همان رابطه ریاضی L2 Regularization است به شکل زیر قابل بازنویسی است:

$$\beta^* = (Y - X\beta)^T (Y - X\beta) + \lambda \beta^T \beta$$

و عبارات را به شکل زیر باز می‌کنیم:

$$\beta^* = Y^T Y - Y^T X\beta - (X\beta)^T Y + (X\beta)^T X\beta + \lambda \beta^T \beta$$

در ادامه از عبارت فوق نسبت به β مشتق می‌گیریم و برابر با صفر قرار می‌دهیم:

$$\frac{d}{d\beta}\beta^* = 0 - 2X^TY + 2X^TX\beta + 2\lambda\beta = 0$$

سپس معادله را به شکل زیر حل می‌کنیم تا مقدار β به دست آید:

$$-X^TY + X^TX\beta + \lambda\beta = 0$$

$$(X^TX + \lambda I)\beta = X^TY$$

$$\beta = \frac{X^TY}{(X^TX + \lambda I)} = (X^TY)(X^TX + \lambda I)^{-1}$$

در نهایت فرم بسته فرمول داده شده به شکل زیر به دست می‌آید.

$$\beta = (X^TY)(X^TX + \lambda I)^{-1}$$

سوال ۳

الف

همانطور که می‌دانیم مجموع احتمال کلاس‌های متفاوت باید برابر یک شود. بنابراین احتمال کلاس y_k به شکل زیر قابل نمایش است:

$$P(Y = y_k|X) = 1 - \sum_{k=1}^{K-1} P(Y = y_k|X)$$

با توجه به اینکه برای انتخاب برچسب برای نمونه داده X نیاز داریم تا احتمال K کلاس را محاسبه کنیم، برای محاسبه احتمال کلاس Kام نیازی به وجود وزن‌هایی برای این کلاس مانند کلاس‌های دیگر وجود ندارد و می‌توان از احتمال زیر برای آن استفاده کرد:

$$P(Y = y_k|X) = \frac{1}{1 + \sum_{k=1}^{K-1} exp(w_{k_0} + \sum_{i=1}^{d} w_{k_i} X)}$$

و برای محاسبه احتمال سایر کلاس‌ها(k=1,2,...,K-1) می‌توان از رابطه زیر استفاده کرد:

$$P(Y = y_k|X) = \frac{\exp\left(w_{k_0} + \sum_{i=1}^{d} w_{k_i} X\right)}{1 + \sum_{k=1}^{K-1} exp(w_{k_0} + \sum_{i=1}^{d} w_{k_i} X)}$$

و در نهایت رابطه فوق برای $P(Y = y_k|X)$ به دست می‌آید.

ب

طبقه‌بند برچسبی با بیشترین احتمال را برای نمونه X انتخاب می‌کند، بنابراین قانون طبقه‌بند به شکل زیر تعریف می‌شود:

$$\hat{y} = \arg_{y_k} \max\ P(Y = y_k|X)$$

سوال ۴

ابتدا به اثبات رابطه اول می‌پردازیم:

$$\overline{P_n}(x) \sim N(\mu, h_n^2 + \sigma^2)$$

عبارت فوق را می‌توان به شکل امیدریاضی احتمال x نسبت به سایز پنجره پارزن نوشت:

$$\overline{P_n}(x) = E[p_n(x)] = \frac{1}{nh_n} E[\varphi\left(\frac{x - x_i}{h_n}\right)]$$

سپس با استفاده از فرمول امید ریاضی برای متغیر تصادفی گسسته، می‌توان عبارت فوق را به شکل زیر بازنویسی کرد:

$$\overline{P_n}(x) = \frac{1}{h_n} \int_{-\infty}^{+\infty} \varphi\left(\frac{x - q}{h_n}\right) p(q) dq$$

سپس $\varphi(x)$ را با فرمول توزیع نرمال $N(0,1)$ و $p(u)$ را با فرمول توزیع نرمال $N(\mu, \sigma^2)$ جایگزین می‌کنیم:

$$\overline{P_n}(x) = \frac{1}{h_n} \int_{-\infty}^{+\infty} \frac{1}{\sqrt{2\pi}} \exp\left[-\frac{1}{2}\left(\frac{x - q}{h_n}\right)^2\right] * \frac{1}{\sqrt{2\pi}\sigma} \exp\left[-\frac{1}{2}\left(\frac{q - \mu}{\sigma}\right)^2\right] dq$$

در ادامه به ساده‌سازی عبارت فوق تا رسیدن به نتیجه نهایی می‌پردازیم:

$$\overline{P_n}(x) = \frac{1}{2\pi h_n \sigma} \exp\left[-\frac{1}{2}\left(\frac{x^2}{h_n^2} + \frac{\mu^2}{\sigma^2}\right)\right] \int_{-\infty}^{+\infty} \exp\left[-\frac{1}{2} q^2 \left(\frac{1}{h_n^2} + \frac{1}{\sigma^2}\right) - 2q\left(\frac{x}{h_n^2} + \frac{\mu}{\sigma^2}\right)\right] dq$$

اگر در این مرحله تغییر متغیر انجام دهیم و فرض کنیم $\theta^2 = \frac{\sigma^2 h_n^2}{\sigma^2 + h_n^2}$ و همچنین $\alpha = \theta^2(\frac{x}{h_n^2} + \frac{\mu}{\sigma^2})$ باشد، آنگاه:

$$\overline{P_n}(x) = \frac{\sqrt{2\pi\theta}}{2\pi h_n \sigma} \exp\left[-\frac{1}{2}\left(\frac{x^2}{h_n^2} + \frac{\mu^2}{\sigma^2}\right) + \frac{1}{2}\frac{\alpha^2}{\theta^2}\right]$$

و در ادامه به شکل زیر ساده می‌شود:

$$\overline{P_n}(x) = \frac{1}{\sqrt{2\pi} h_n \sigma} \frac{h_n \sigma}{\sqrt{h_n^2 + \sigma^2}} \exp\left[-\frac{1}{2}\left(\frac{x^2}{h_n^2} + \frac{\mu^2}{\sigma^2} - \frac{\alpha^2}{\theta^2}\right)\right]$$

با جایگزینی متغیرهای کمکی، مقدار $\frac{\alpha^2}{\theta^2}$ به شکل زیر به دست می‌آید:

$$\frac{\alpha^2}{\theta^2} = \frac{\theta^4}{\theta^2}\left(\frac{x}{h_n^2} + \frac{\mu}{\sigma^2}\right)^2$$

همچنین عبارت زیر نیز ساده می‌شود:

$$\frac{x^2}{h_n^2} + \frac{\mu^2}{\sigma^2} - \frac{\alpha^2}{\theta^2} = \frac{(xh_n)^2}{(h_n^2 + \sigma^2)h_n^2} + \frac{(\mu\sigma)^2}{(h_n^2 + \sigma^2)\sigma^2} - \frac{2x\mu}{h_n^2 + \sigma^2} = \frac{(x - \mu)^2}{h_n^2 + \sigma^2}$$

و در نهایت با جایگذاری عبارت فوق در عبارت $\overline{P_n}(x)$ عبارت ریاضی زیر به دست می‌آید:

$$\overline{P_n}(x) = \frac{1}{\sqrt{2\pi(h_n^2+\sigma^2)}} \exp\left[-\frac{1}{2}\frac{(x-\mu)^2}{h_n^2+\sigma^2}\right]$$

عبارت فوق رابطه مربوط به توزیع نرمال با میانگین μ و انحراف معیار $h_n^2+\sigma^2$ را نشان می‌دهد. بنابراین:

$$\overline{P_n}(x) \sim N(\mu, h_n^2+\sigma^2)$$

و در نتیجه عبارت مورد نظر اثبات شد.

سپس به سراغ اثبات عبارت دوم می‌رویم:

$$P_n(x) - \overline{P_n}(x) = \frac{1}{2}\left(\frac{h_n}{\sigma^2}\right)\left[1-\left(\frac{x-\mu}{\sigma}\right)^2\right]P(x)$$

با جایگذاری عبارات متناظر با $\overline{P_n}(x)$ و $P_n(x)$ داریم:

$$P_n(x) - \overline{P_n}(x) = \frac{1}{\sqrt{2\pi}\sigma}\exp\left[-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2\right] - \frac{1}{\sqrt{2\pi(h_n^2+\sigma^2)}}\exp\left[-\frac{1}{2}\frac{(x-\mu)^2}{h_n^2+\sigma^2}\right]$$

سپس به ساده‌سازی عبارت فوق می‌پردازیم تا به عبارت زیر برسیم:

$$P_n(x) - \overline{P_n}(x) = P(x)\left\{1 - \frac{1}{\sqrt{1-\left(\frac{h_n}{\sigma}\right)^2}} exp\left[\frac{1}{2}\frac{h_n^2(x-\mu)^2}{h_n^2+\sigma^2}\right]\right\}$$

سپس به عبارت زیر می‌رسیم:

$$P_n(x) - \overline{P_n}(x) = P(x)\left\{1-\left(1-\frac{1}{2}\left(\frac{h_n}{\sigma}\right)^2\right)\left(1-\frac{h_n}{\sigma^2}\frac{(x-\mu)^2}{h_n^2+\sigma^2}\right)\right\}$$

$$= P(x)\left\{1-1+\frac{1}{2}\frac{h_n}{\sigma^2}-\frac{h_n}{2\sigma^2}\frac{(x-\mu)^2}{h_n^2+\sigma^2}\right\}$$

$$= P(x)\frac{1}{2}\left(\frac{h_n}{\sigma}\right)^2\left[1-\frac{(x-\mu)^2}{h_n^2+\sigma^2}\right]$$

$$P_n(x) - \overline{P_n}(x) = \frac{1}{2}\left(\frac{h_n}{\sigma^2}\right)\left[1-\left(\frac{x-\mu}{\sigma}\right)^2\right]P(x)$$

همانطور که مشاهده می‌شود با جایگذاری عبارات مربوطه و ساده‌سازی سمت چپ عبارت داده شده، به سمت راست عبارت مذکور می‌رسیم.

سوال ۵

برای اثبات اینکه یک متریک، فاصله استاندارد است، نیاز داریم تا خواصی مانند مثبت بودن فاصله، صفر بودن فاصله در صورت یکی بودن a , b، نامساوی مثلثی و جابجایی پذیری را برای آن متریک اثبات کنیم. در ادامه هر کدام از این خواص یک به یک اثبات خواهند شد. لازم به ذکر است که متریک تعریف شده پس از ساده‌سازی شکل زیر را به خود می‌گیرد:

$$D(a,b) = \sqrt{\sum_{i=1}^{d} p_i^2 (a_i - b_i)^2}$$

جابجایی پذیری

در این بخش باید اثبات کنیم که:

$$D(a,b) = D(b,a)$$

از سمت چپ عبارت فوق شروع می‌کنیم و تلاش می‌کنیم به سمت راست آن برسیم:

$$D(a,b) = \sqrt{\sum_{i=1}^{d} p_i^2 (a_i - b_i)^2}$$

همانطور که می‌دانیم $x^2 = (-x)^2$ است و بنابراین می‌توان عبارت زیر توان دو را قرینه کنیم:

$$D(a,b) = \sqrt{\sum_{i=1}^{d} p_i^2 (b_i - a_i)^2}$$

عبارت به دست آمده فوق همان سمت راست عبارت اولیه است:

$$\sqrt{\sum_{i=1}^{d} p_i^2 (b_i - a_i)^2} = D(b,a)$$

در نتیجه:

$$D(a,b) = D(b,a)$$

نامساوی مثلث

نامساوی مثلثی به شکل زیر تعریف می‌شود:

$$D(a,b) + D(b,c) \geq D(a,c)$$

می‌توان متریک فوق را به شکل quadratic به شکل زیر بازنویسی کرد:

$$D(a,b) = \left|\left|P(a-b)\right|\right|_2$$

در این صورت نامساوی مثلثی به شکل زیر تعریف می‌شود:

$$||Pa||_2 + ||Pb||_2 \geq ||P(a+b)||_2$$

با جایگذاری w=Pa و z=Pb نامساوی به شکل زیر نوشته می‌شود و ساده‌سازی ادامه می‌یابد:

$$||w||_2 + ||z||_2 \geq ||w+z||_2$$

طرفین عبارت فوق را به توان دو می‌رسانیم:

$$||w||_2^2 + ||z||_2^2 + 2||w||_2||z||_2 \geq ||w||_2^2 + ||z||_2^2 + 2\sum_{i=1}^{d} w_i^d z_i^d$$

که پس از ساده‌سازی به عبارت زیر می‌رسیم:

$$||w||_2||z||_2 \geq \sum_{i=1}^{d} w_i^d z_i^d$$

عبارت فوق طبق نامساوی کوشی-شوارتز یک بدیهی است و در نتیجه نامساوی مثلثی برای این متریک اثبات می‌شود.

صفر شدن فاصله

در صورتی که a=b باشد فرمول فاصله به شکل زیر درمی‌آید:

$$D(a,b) = \sqrt{\sum_{i=1}^{d} p_i^2 (a_i - a_i)^2} = \sqrt{\sum_{i=1}^{d} p_i^2 0} = 0$$

بنابراین متریک تعریف شده یکی دیگر از خواص فاصله استاندارد را داراست.

مثبت بودن فاصله

متریک تعریف شده به صورت ریشه دوم جمع مربعات یکی عبارت است و در نتیجه همیشه عددی مثبت است. بنابراین متریک تعریف شده یکی دیگر از خواص فاصله استاندارد را داراست.

تاثیر بر KNN

تاثیر این متریک بر الگوریتم KNN بدین شکل است که با توجه به مقادیر بردار وزن‌های ضرب شده در ویژگی‌ها، فواصل جدید ممکن است با فاصله اقلیدسی برابر نباشند و البته ممکن است که بردار وزن‌ها با توجه به نیاز مسئله و عدم کفایت فاصله اقلیدسی برای حل آن، محاسبه شود. اما به طور کلی با توجه به عملکرد KNN و نیاز به مشخص کردن نقاطی با حداقل فاصله نسبت به نقطه مورد نظر، این متریک می‌تواند برای KNN مفید باشد و مشکلی ایجاد نخواهد کرد. هر چند همانطور که گفته شد ممکن است نتیجه نهایی با استفاده از فاصله اقلیدسی یکسان نباشد.(همانطور که نتیجه در صورت استفاده از متریک فاصله اقلیدسی و فاصله منهتن یکسان نیست اما هر دو کاربرد خود را دارند.)

سوال ۶

در این سوال طبق فرمول‌های داده شده دادگانی ایجاد می‌شوند و سپس دو نویز متفاوت در دو حالت به داده‌ها اضافه می‌شود. حالت اول شامل نویز سفید گاوسی می‌باشد و در حالت دوم نویز پواسون با $\lambda = 2$ به دادگان اضافه می‌شود. ابتدا دادگان اولیه به شکل زیر به کمک کتابخانه Numpy ایجاد می‌شوند:

```
X = np.arange(-10,10,0.2)
Y = (2*np.cos(X)/-np.pi) + ((2*X)/(2*np.pi)) + (2*np.cos(3*X)/(-3*np.pi))
```

سپس به کمک کتابخانه Numpy به تعداد دادگان ایجاد شده در مرحله قبل، دادگان تصادفی از دو توزیع مذکور به شکل زیر ایجاد می‌شود و به دادگان فوق اضافه می‌شود:

```
white_noise = np.random.normal(0, 1, size=100)
noisy_Y_w = Y + (0.1 * white_noise)
poisson_noise = np.random.poisson(2, size=100)
noisy_Y_p = Y + (0.1 * poisson_noise)
```

در مرحله بعد با استفاده از تابع polyfit از کتابخانه numpy چندجمله‌ای‌هایی با درجه ۱ تا ۱۵ بر روی دو دسته دادگان برازش می‌شوند و ضرایب چند جمله‌ای‌ها با درجه‌های متفاوت برای هر دسته داده در یک لیست ذخیره می‌شود. کد این بخش در باکس زیر ادامه قابل مشاهده است:

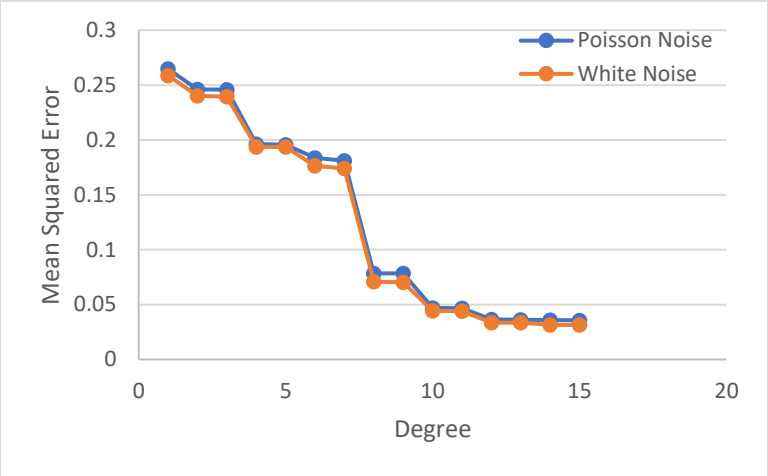
```
coeff1 = list()
coeff2 = list()
for i in range(1,16):
    coeff1.append(np.polyfit(X,noisy_Y_w,deg=i))
    coeff2.append(np.polyfit(X,noisy_Y_p,deg=i))
```

در نهایت مقدار معیار میانگین مربعات خطا[1] برای هر کدام از درجات برازش شده محاسبه می‌شود.

[1] Mean Squared Error(MSE)

الف

در این بخش بهترین و بدترین چندجمله‌ای برازش شده برای هر یک از مجموعه دادگان مشخص می‌شود. در نمودار زیر مقادیر معیار میانگین مربعات خطا برای هر یک از درجات و روی هر دو دسته دادگان ایجاد شده مشخص شده است:



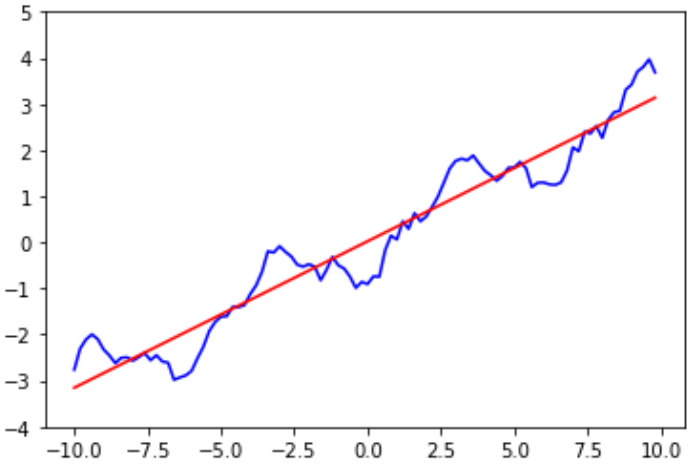
نمودار ۱

همانطور که در نمودار فوق مشخص است بهترین چندجمله‌ای برازش شده برای هر دو حالت دادگان، چند جمله‌ای با درجه ۱۵ است و بدترین چند جمله‌ای درجه‌ای برابر با ۱ دارد.

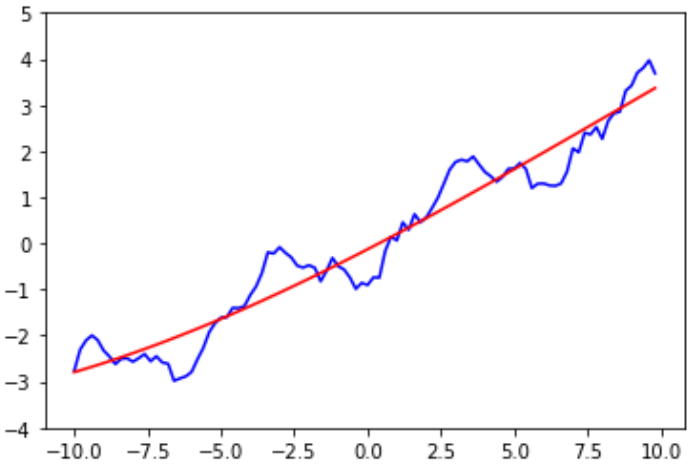
ب

در این قسمت سوال نمودارهای خواسته شده رسم می‌شوند و مقادیر معیار مورد نظر گزارش می‌شوند. همانطور که در قسمت قبل همین سوال گزارش شد، بهترین درجه برای هر دو حالت دادگان ۱۵ و بدترین درجه برابر با ۱ است و بنابراین برای هر کدام از حالات دادگان ۴ نمودار رسم شده است. همچنین مقدار میانگین مربعات خطا برای هر چند جمله‌ای برازش شده در توضیحات هر نمودار مشخص شده است. در هر یک از نمودارهای زیر منحنی آبی مقادیر دادگان واقعی و منحنی قرمز چندجمله‌ای برازش شده را نشان می‌دهد.

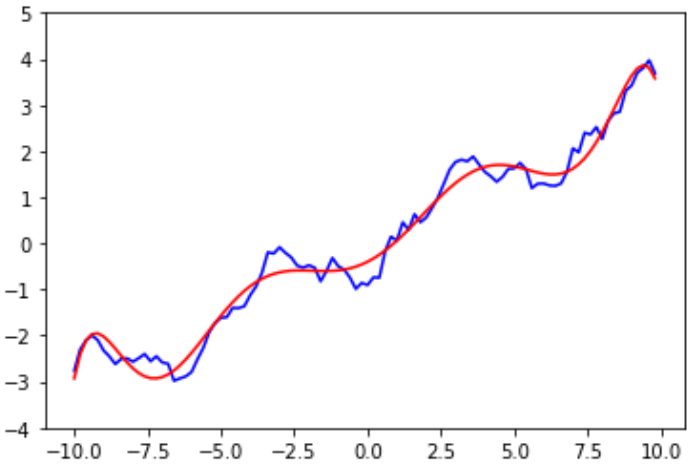
دادگان با نویز سفید گاوسی



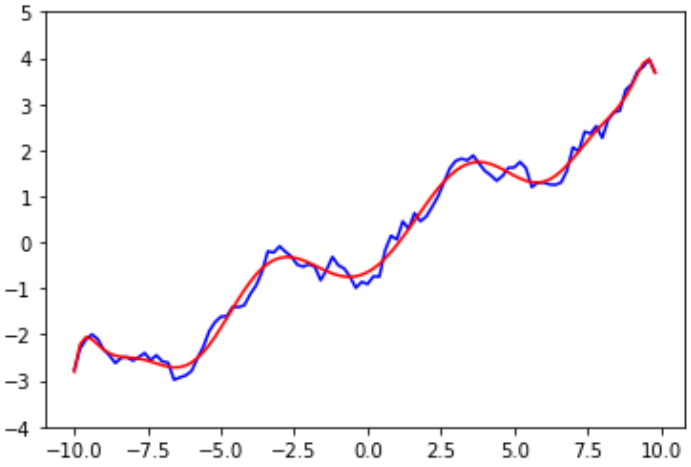
نمودار ۲: بهترین چند جمله‌ای برازش شده با درجه ۱ و MSE=0.258



نمودار ۳: چند جمله‌ای درجه ۳ و MSE=0.239

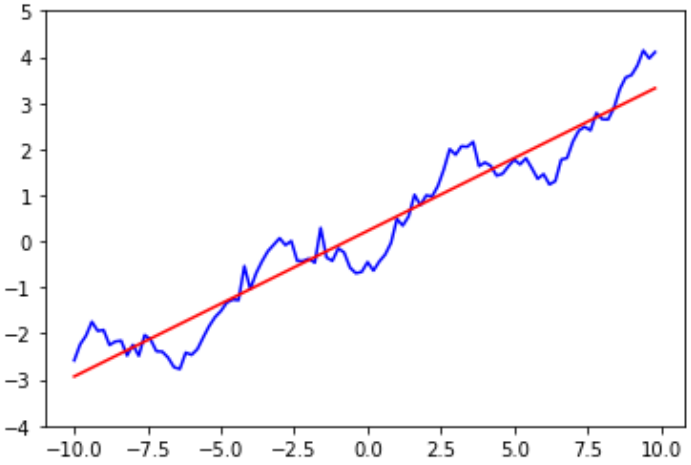


نمودار ٤ : چند جمله‌ای درجه ٨ و MSE=0.070

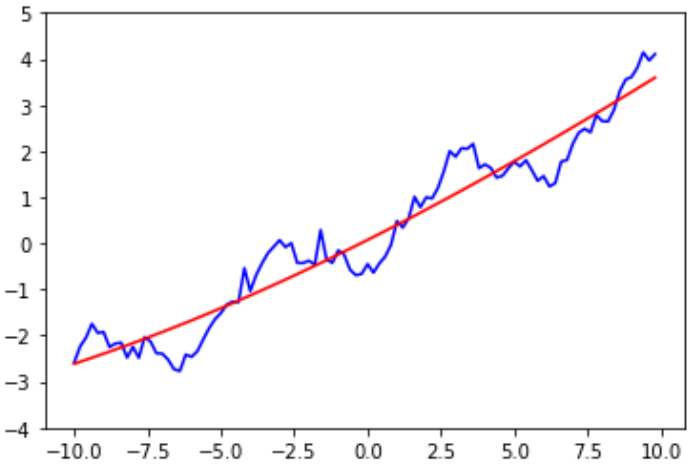


نمودار ٥: بهترین چند جمله‌ای برازش شده با درجه ١٥ و MSE=0.0314

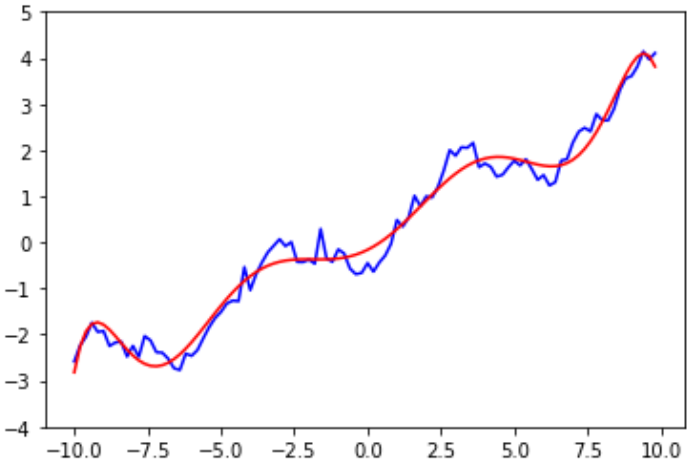
دادگان با نویز پواسون



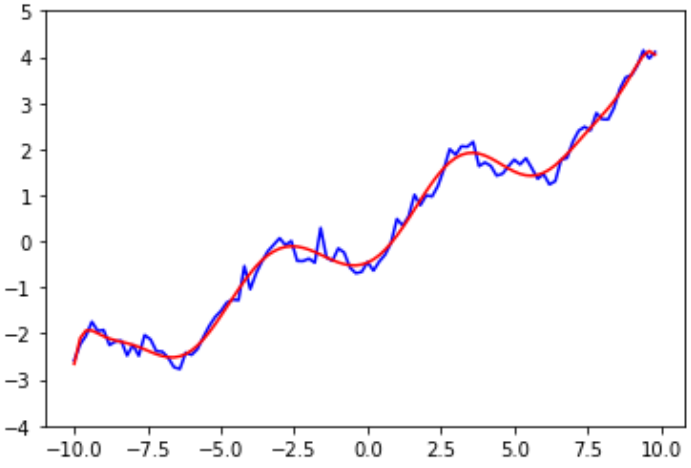
نمودار ٦ : بهترین چند جمله‌ای برازش شده با درجه ١ و MSE=0.264



نمودار ۷: چند جمله‌ای درجه ۳ و MSE=0.245



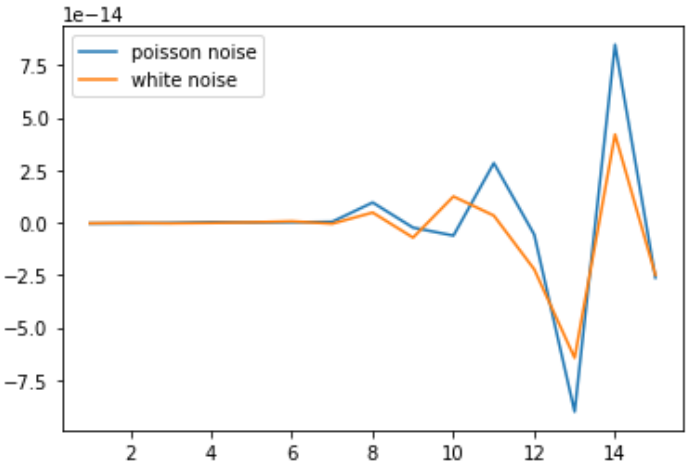
نمودار ۸: چند جمله‌ای درجه ۸ و MSE=0.078



نمودار ۹: بهترین چند جمله‌ای برازش شده با درجه ۱۵ و MSE=0.0356

ج

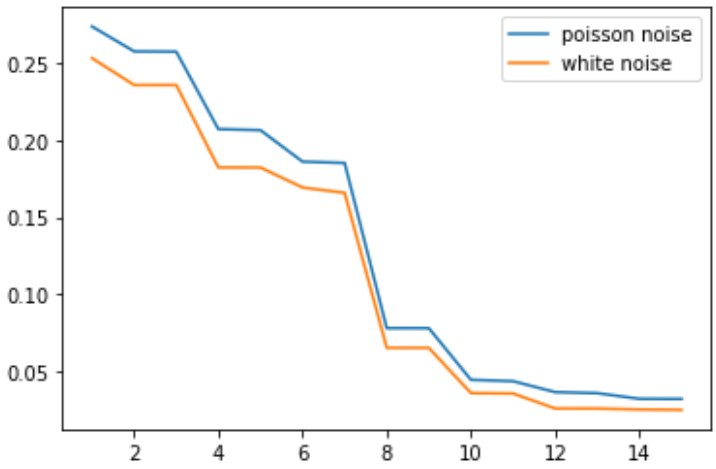
برای مقدار خطای بایاس دو دسته داده نمودار زیر به دست آمده است:



نمودار ۱۰

آنطور که انتظار می‌رفت، بایاس با افزایش پیچیدگی مدل به صورت پیوسته کاهش پیدا نکرد که می‌تواند به دلیل کمبود تعداد نمونه‌های مورد استفاده باشد اما به طور کلی کاهش مقدار بایاس تا چندجمله‌ای با درجه ۱۳ مشهود است.

در ادامه نمودار واریانس قابل مشاهده است:



نمودار ۱۱

در این نمودار نیز انتظار داشتیم تا واریانس به صورت صعودی رشد کند اما کاملا برعکس نمودار به صورت پیوسته نزولی است.

سوال ۷

در این سوال، در دو حالت دادگانی در دو کلاس را ایجاد می‌کنیم و سپس به دسته‌بندی آن‌ها به کمک Logistic Regression می‌پردازیم و در نهایت مرز تصمیم را رسم می‌کنیم. در ابتدا تابعی به شکل زیر برای ایجاد نقاط تصادفی در دایره مورد نظر ایجاد می‌کنیم و توسط دو متغیر r1 و r2 نشان می‌دهیم در صورتی که نیاز به ایجاد دادگان تصادفی به شکل یک حلقه داریم، دادگان در چه بازه‌ای قرار بگیرند.

```
def random_ora(a, b, r1, r2, c):
    Y = list()
    X = np.random.uniform(a-r2,a+r2,c).tolist()
    for x in X:
        if x>a+r1 or x<a-r1:
            lim = np.sin(np.arccos((x-a)/r2))*r2
            Y.append(np.random.uniform(b-lim,b+lim))
        else:
            lim1 = np.sin(np.arccos((x-a)/r1))*r1
            lim2 = np.sin(np.arccos((x-a)/r2))*r2
            rand = np.random.uniform((b+lim1,b-lim2),(b+lim2,b-lim1))
            Y.append(rand[randint(0,1)])
    return np.array(X),np.array(Y)
```

سپس به شکل زیر توابع فراخوانی می‌شوند تا داده‌ها برای حالت اول خواسته شده در دو دسته ایجاد شوند:

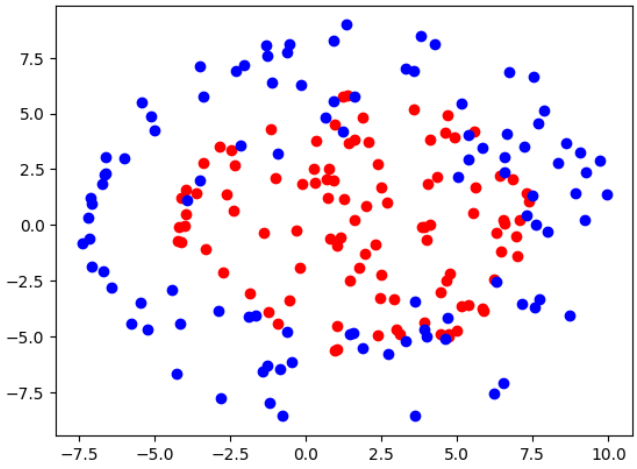
```
X11,Y11 = random_ora(1.5, 0, 0, 6, 100)
X12,Y12 = random_ora(1.5, 0, 4, 9, 100)
```

و دادگان حالت دوم در دو دسته به شکل زیر ایجاد می‌شود:

```
X21 = np.random.normal(1,1,100)
Y21 = np.random.normal(0,1,100)
X22,Y22 = random_ora(1.5, 0, 2, 6, 200)
```

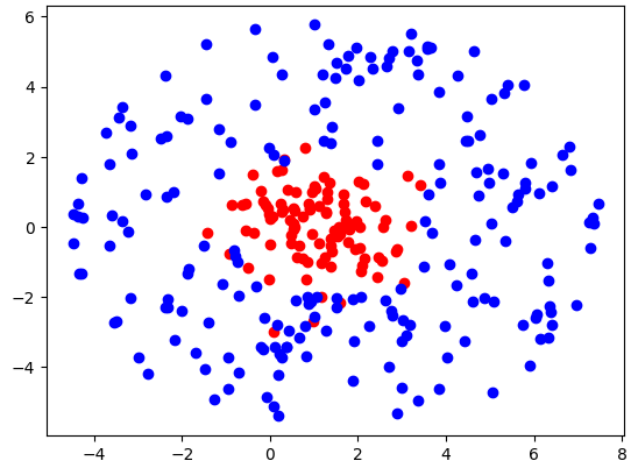
الف

نمودار دادگان ایجاد شده حالت اول به شکل زیر می‌باشد:



نمودار ۱۲

و در حالت دوم نمودار دادگان به شکل زیر درمی‌آید:



نمودار ۱۳

همانطور که در دو نمودار فوق قابل مشاهده است، کلاس‌ها در دادگان حالت دوم دارای تفکیک بیشتری نسبت به دادگان حالت اول می‌باشند و انتظار می‌رود که طبقه بند با دقت بهتری این مجموعه را طبقه‌بندی کند.

ب

در ابتدا نیاز داریم تا در هر دو حالت دادگان هر کلاس را به تعداد ابعاد بالاتری ببریم تا امکان تفکیک آن‌ها توسط طبقه‌بند ایجاد شود. این افزایش بعد، طبق خواسته سوال از ۲ به ۳۵ خواهد بود. عمل خواسته شده توسط تابع زیر انجام می‌گیرد:

```
def add_dim(X,Y):
    i = 3
    df = pd.DataFrame()
    df['x1'] = X
    df['x2'] = Y
    for j in range(2, 8):
      for k in range(0, j+1):
        col_name = 'x' + str(i)
        i += 1
        df[col_name] = df['x1']**(j-k) * df['x2']**k
    return df
```

سپس به تابعی نیاز داریم که داده‌ها را به دو مجموعه آموزش و تست تقسیم بندی کند. تابع مورد نظر به شکل زیر تعریف می‌شود:

```
def train_test_split(X, Y, train_size):
    data = X.tolist()
    label = Y.tolist()
    y_train = list()
    x_train = list()
    y_test = list()
    x_test = list()
    for i in range(len(data)):
        if np.random.uniform(0,1) > train_size:
            x_test.append(data[i])
            y_test.append(label[i])
        else:
            x_train.append(data[i])
            y_train.append(label[i])
    return np.array(x_train), np.array(y_train), np.array(x_test), np.array(y_test)
```

و در مرحله پایانی تعریف توابع مورد نیاز، کلاسی برای ایجاد، آموزش و تست مدل Logistic Regression به شکل زیر تعریف می‌شود. در ابتدا تابع Constructor کلاس تعریف می‌شود که مقادیر تعداد تکرار، λ، تعداد کلاس‌ها و مقدار هایپرپارامتر نرخ یادگیری(alpha) در Attributeهای کلاس ذخیره می‌شود:

```
class logistic_regression():
    def __init__(self, epochs, alpha, n_class, lambda_):
        self.epochs = epochs
        self.alpha = alpha
        self.n_class = n_class
        self.lambda_ = lambda_
```

سپس تابع initial_values به شکل زیر تعریف می‌شود. در ابتدا یک ستون با مقدار ۱ به عنوان ستون اول مجموعه داده اضافه می‌شود تا از نیاز به یادگیری پارامتر w_0 به صورت جداگانه اجتناب شود. سپس یک ماتریس به صورتی که به تعداد ویژگی‌ها ستون داشته باشد و به تعداد کلاس‌های مجموعه دادگان سطر داشته باشد ایجاد می‌شود تا مقدار w برای هر یک از کلاس‌ها به صورت تجمیع شده ذخیره شود:

```
def initial_values(self, X_train):
        X = np.ones((X_train.shape[0],X_train.shape[1]+1))
        X[:,1:] = X_train
        w = np.zeros((X.shape[1], self.n_class))
        return X, w
```

همچنین سه تابع Sigmoid، Cost_function و Gradient_descent مطابق تعریف، به شکل زیر ایجاد می‌شوند:

```
def sigmoid(self, z):
        return 1 / (1 + np.exp(-z))

    def cost_function(self, X, y, h, w):
        m = len(y)
        j_w = (1./2*m) * (((y - h).T @ (y- h)) + self.lambda_ * np.sum(np.square(w)))
        return j_w

    def gradient_descent(self, X, y, h, w):
        m = len(y)
        w = w - self.alpha * (1/m)* ( np.dot(X.T, (h-y)) + self.lambda_ * w )
        return w
```

در قسمت بعد مهم‌ترین تابع این کلاس، تابع برازش، مطابق تعاریف ایجاد می‌شود و مقدار وزن‌های نهایی را بازمی‌گرداند:

```
def fit(self, X_train, y_train):
        self.w = []
        self.cost = []
        X, w = self.initial_values(X_train)
        for i in range(0, self.n_class):
            y = np.where(y_train == np.unique(y_train)[i], 1, 0)
            cost = []
            w_i = w[:, i]
            for epoch in range(self.epochs):
                h = self.sigmoid(X.dot(w_i))
                w_i = self.gradient_descent(X, y, h, w_i)
                cost.append(self.cost_function(X, y, h, w_i))
            self.w.append(w_i)
            self.cost.append(cost)
        return np.array(self.w)
```

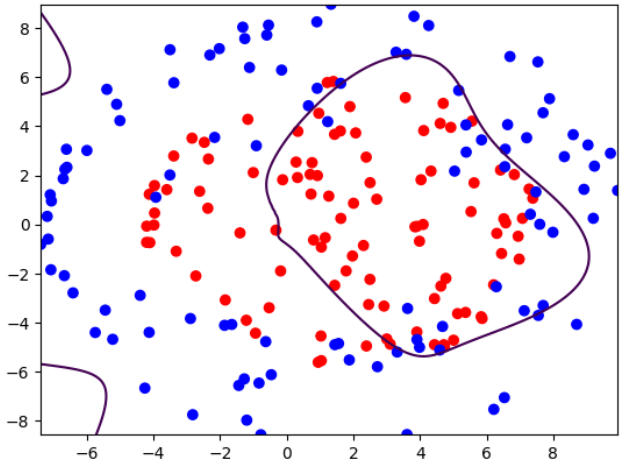
و در نهایت توابع پیش‌بینی برچسب دادگان جدید و محاسبه دقت طبقه‌بند تعریف شده است:

```
def predict(self, X_test):
        classProbabilities = self.sigmoid(np.insert(X_test, 0, 1, axis=1) @ np.array(self.w).T)
        pred = classProbabilities.argmax(axis=1)
        return pred


    def accuracy(self, y, pred):
      return np.mean(pred == y)
```

پس از تعریف کلاس و توابع فوق، برای هر کدام از حالات مجموعه داده توابع به نوبت فراخوانی می‌شوند تا بهترین طبقه‌بندی داده صورت گیرد. طبقه‌بندها برای هر کدام از حالات داده، دقتی به شکل زیر دارند:

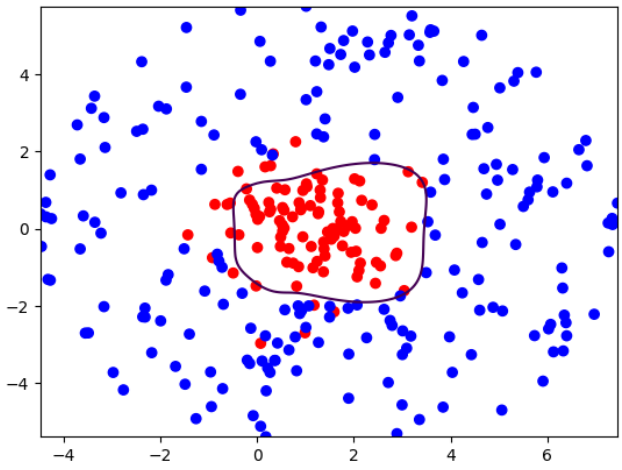
	دادگان حالت اول	دادگان حالت دوم
Accuracy	77.7%	94.36%

مرز تصمیم برای دادگان حالت اول به شکل زیر رسم شده است:



نمودار ۱۴

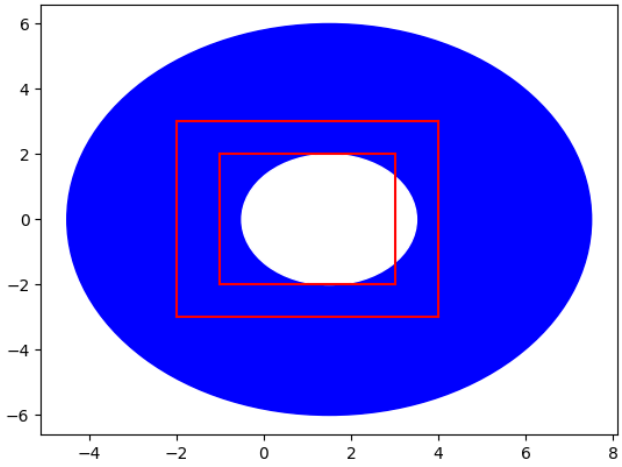
و برای دادگان حالت دوم به شکل زیر درمی‌آید:



نمودار ۱۵

ج

همانطور که در بخش الف همین سوال نیز ذکر شد، دادگان حالت دوم دارای تفکیک پذیری بیشتری نسبت به حالت اول هستند. با توجه به اینکه دادگان حالت دوم از دو دسته تشکیل شده است که محدوده هر دسته در شکل زیر مشخص شده است. دسته اول با مربع قرمز بزرگ و ناحیه آبی رنگ نیز محدوده دسته دوم است.



نمودار ۱۶

همانطور که مشخص است بیشترین تداخل دو دسته این حالت بین مربع کوچکتر و بزرگتر با ناحیه آبی است که طبق تعریف توزیع نرمال، این ناحیه که بین $\mu \pm 2\sigma$ و $\mu \pm 3\sigma$ است تنها شامل ۴.۷ درصد داده‌های دسته اول است. در صورتی که در حالت اول ۵۵ درصد محدوده دادگان دسته دوم با محدوده دادگان دسته اول دارای همپوشانی است.(محاسبات به شکل زیر انجام شده است)

$$Overlap = \frac{\pi(6)^2 - \pi(4)^2}{\pi(6)^2} = \frac{20}{36} = 0.555$$

بنابر توضیحات فوق حتی پیش از انجام طبقه‌بندی نیز انتظار می‌رفت که بر روی دادگان حالت دوم بهتر عمل کند.

اما پس از بررسی نتایج طبقه‌بندی، طبقه‌بند دادگان دوم مطابق انتظار عمل کرد و دقت خوبی ارائه کرد. این عملکرد در نموداری که شامل مرز تصمیم است نیز قابل مشاهده است و عملکرد قابل قبول این طبقه‌بند را نشان می‌دهد. اما طبقه‌بند اول نتوانست مرزر تصمیم خوبی رسم کند و دقت متوسطی نیز به دست آورد که با توجه با درهم‌آمیختگی دادگان این مجموعه و طبقه‌بندی با پیچیدگی نه چندان بالا دقت چندان بدی هم به نظر نمی‌رسد.

سوال ۸

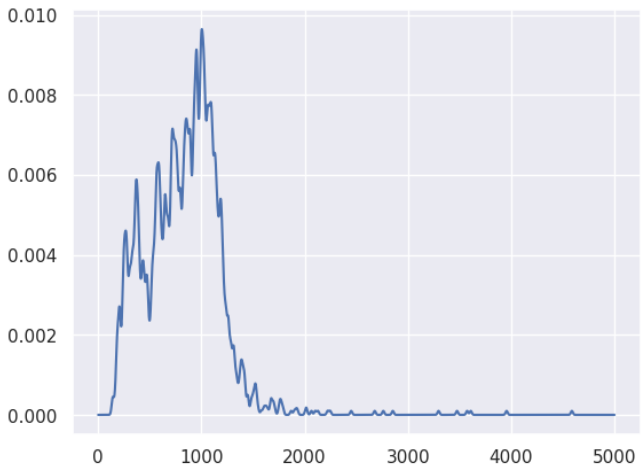
برای پیاده‌سازی این سوال مجموعه داده ted talks در پروژه بارگیری شده است. در ادامه کلاس Parzen به شکل زیر پیاده شده است:

```
class Parzen:

    def __init__(self, h=1.0):

        self.h = h


    def fit(self, X):

        self.X = X

        self.n = len(X)


    def calc_prob(self, data):

        prob = []

        for x in tqdm(data):

            temp = [abs(np.sqrt((math.pi)*2) * (1/self.h)* np.exp((-1/2)*((x-xi)/self.h)**2)) for xi
in self.X]

            prob.append((sum(temp)[0]/self.n))


        return prob
```

و تغییر اندازه پنجره‌های مورد نیاز با تغییر پارامتر h انجام می‌گیرد.

الف

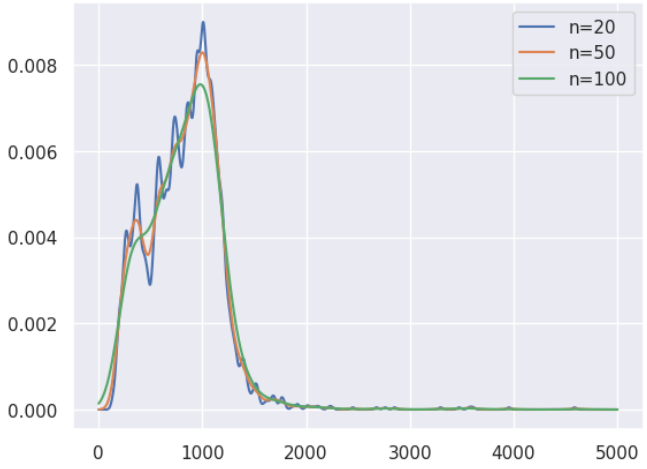
داده‌های ستون duration استخراج شده است و توزیع داده‌ها با اندازه پنجره پارزن ۱۰ به شکل زیر رسم شده است:



نمودار ۱۷

ب

توزیع مقدار duration با استفاده از سه مقدار ۲۰، ۵۰ و ۱۰۰ برای اندازه پنجره پارزن به شکل زیر درمی‌آید:

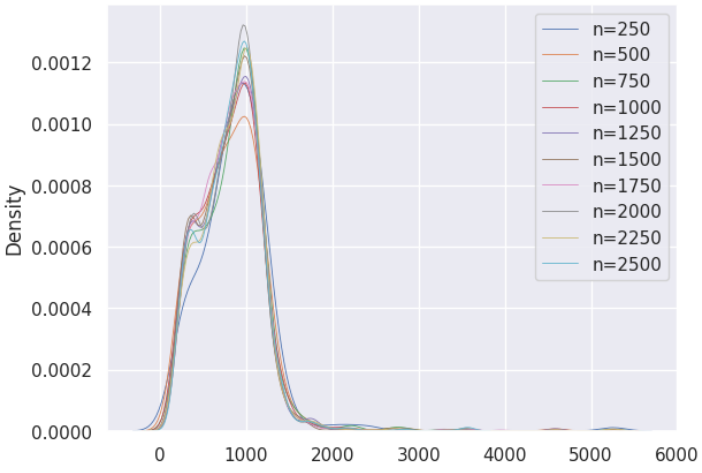


نمودار ۱۸

با توجه به نمودارهای فوق، آنچه به نظر می‌رسد این است که هرچه مقدار n افزایش می‌یابد، نمودار توزیع رسم شده smoothتر می‌شود که یک دلیل آن می‌تواند به دلیل کاهش نویز و قرار گرفتن تعداد بیشتری داده در یک پنجره و کاهش داده‌های نویزی هر پنجره نسبت به داده‌های سالم آن باشد.

ج

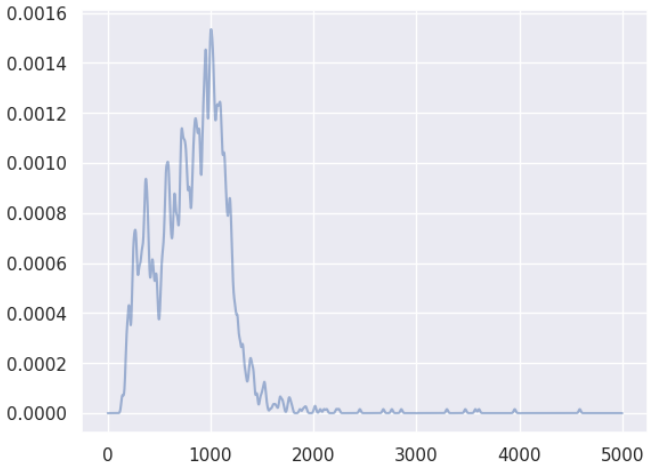
توزیع ستون duration با استفاده از اندازه پنجره‌های متفاوت و استفاده از توابع پیشفرض کتابخانه seaborn در نمودار زیر قابل مشاهده است:



نمودار ۱۹

د

خواسته سوال در بخش الف با استفاده از کتابخانه sklearn نیز پیاده‌سازی شد و در نمودار زیر قابل مشاهده است:



نمودار ۲۰

نمودار رسم شده توسط کتابخانه sklearn در ظاهر تفاوت محسوسی با کلاس پیاده‌سازی ما نداشت و نمودارها تا حد بسیار زیادی مشابه یکدیگر رسم شده‌اند.