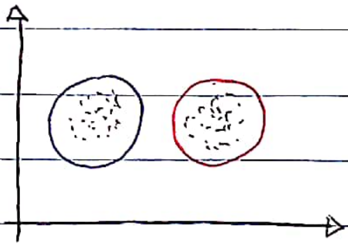
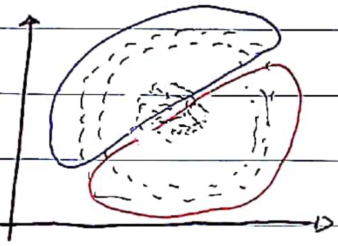
زهرا سهیشتل ۹۸۱۰۵۱۳۸ تمرین سری ۲ ML

سوال ۱.۱)

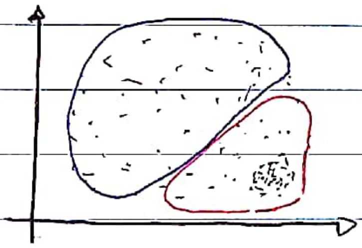
الف) کلاسترها به صورت زیر خواهند بود.



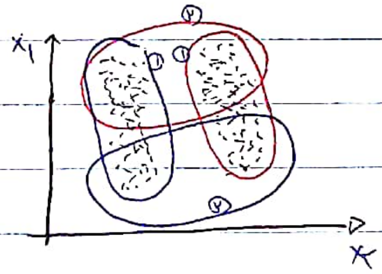
دو دسته نقطه به صورت جدا از هم به خوبی به شکل مقابل دسته‌بندی می‌شوند.



با توجه به تقارن شکل، اجرای clustering منجر به دسته‌بندی به شکل مقابل شده؛ نیمی از نقاط در دو دسته به صورت شکل مانند شکل می‌شوند.



به دلیل چگالی نقاط در سمت راست، مرکز در آن ناحیه کلاستر تشکیل خواهد شد. و تعدادی نقاط که به این مرکز در ناحیه چگال‌تر نزدیک‌تر هستند نیز در این cluster قرار می‌گیرند و شکلی مانند شکل مقابل ایجاد می‌شود.



با توجه به نحوه انتخاب رندم نقاط اولیه به عنوان مرکز، هر یک از حالات زیر ممکن است اتفاق بیفتد. به یکی از ۲ حالت زیر cluster می‌شوند.

⇐ در حالت آخر feature scaling در راستای x_1 باعث متمرکز شدن دیتاها می‌شود و باعث می‌شود چیزی شبیه به شکل اول شود که نقاط به صورت کروی‌تر شده‌اند و در نتیجه حالت بد ② در شکل آخر پیش نیاید.

سوال ۱.۲)

در inertia فقط فاصله های درون کلاسترها درنظر گرفته میشود درصورتی که در silhouette score علاوه بر فواصل درون کلاسترها، فاصله‌های بین کلاسترها نیز درنظر گرفته میشوند.

در inertia، با رشد k، خطا بصورت نزولی و کاهشی باید داشت خوب نیست و ما به دنبال حالت بهینه k هستیم.
همچنین silhouette score قابلیت مقایسه k را نیز فراهم میکند.
اما در silhouette نمودار شکل خاصی ندارد که مثل inertia نقطه زانویی که تقسیم‌بندی راحت‌تر باشد.

ابتدا اینکه نمودار خوب است که تمام cluster plot ها از نظر پیوسته باشند، که این موضوع در شکل های k=4 و k=3 رعایت نشده پس خوب نیستند.
از بین نمودارهای باقی مانده k=5 و k=4، میدانیم نمودار خوب است که اندازه‌های کلاسترها به صورت یونیفرم باشد و هم‌اندازه. پس نمودار سوم با k=5 مناسب‌ترین نمودار میباشد و درنتیجه تعداد خوشه‌ها = 5 انتخاب مناسب میباشد.

سوال ۱.۳)

active learning

حالتی است که در آن الگوریتم لرنینگ به صورت تعاملی و interactively میتواند درخواست label جدید دهد در مراحل اجرای الگوریتم.
یک حالت خاص از semi_supervised محسوب میشود.
به این صورت که ابتدا خوشه‌بندی میکنیم و به صورت interactively از user میخواهیم به خوشه‌ها label دهد. اگر نمونه‌های داخل cluster نتوانیم label را با قطعیت بدهیم، به آن label نمیدهیم.
در ادامه با استفاده از data های label شده مدل میسازیم و همینطور ادامه میدهیم.

Semi supervised learning:

یک dataset بزرگ داریم که فقط بخشی از آن label خورده‌اند و باقی label ندارند. حال می‌خواهیم با ترکیبی از supervised و unsupervised این مدل را train کنیم. به دلیل هزینه فقط تعدادی کمی داده را لیبل می‌توانیم بزنیم.

فرض کنید می‌توانیم k تا query بدهیم (در واقع فرآیند train کردن که k به نسبت حجم دیتا مقدار خیلی کمی دارد. به این صورت عمل می‌کنیم که ابتدا به روش unsupervised کل دیتا را به k خوشه cluster می‌کنیم سپس k تا کوئری که می‌توانیم بزنیم (از نقاط نزدیک به مراکز خوشه‌ها استفاده می‌کنیم و آن‌ها را درخواست می‌دهیم و این label را به بقیه اعضای cluster تعمیم می‌دهیم. حال از هر cluster یک نقطه به نماینده (نقاط نزدیک مرکز خوشه) انتخاب می‌کنیم و مدل جدید را روی این k نقطه train می‌کنیم به روش supervised.

سوال ۲)

PCA whitening

در این روش سعی می‌کنیم ابتدا با مقدار کردن دیتا برروی eigenvector ها همبستگی میان feature ها را از بین ببریم و سپس با نرمالایز کردن dataset واریانس هر feature را $\boxed{1}$ کنیم، که این کار را با تقسیم کردن هر component به جذر متناظرش eigen value انجام می‌دهیم. $X_{m\times n}$

ابتدا ماتریس X' را به صورتی می‌سازیم که zero mean باشد، این طور که در واقع هر ستون ماتریس X را از میانگین آن ستون کم می‌کنیم تا ماتریس X' حاصل شود.

حال ماتریس کوواریانس را به صورت $C_{n\times n} = \frac{1}{m} X' X'^T$ قابل محاسبه است.

حال با استفاده از PCA (روش قطری‌سازی) eigenvector های ماتریس C را محاسبه می‌کنیم

$$\Rightarrow C_{n\times n} = U_{n\times n} \Sigma_{n\times n} U^T_{n\times n}$$

که ماتریس $U = \begin{bmatrix} | & & | \\ u_1 & \cdots & u_n \\ | & & | \end{bmatrix}$ شامل eigen vector های ماتریس C باشد.

و ماتریس قطری Σ شامل eigenvalue ها به روی قطر اصلی آن باشد.

بنابراین ماتریس $\tilde{X}$ را به صورت $\tilde{X}_{n\times n} = U^T X'$ حاسبه کنیم که در واقع $\tilde{X}$ تصویر داده‌ها بر بردارهای eigen vector ها باشد.

حال برای PCA یعنی کاهش بعد ماتریس $\tilde{X}$ به K سطر اول آن را در نظر می‌گیریم و بقیه سطرها صفر می‌شوند. که آن‌ها را در نظر نمی‌گیریم و ماتریس $\tilde{X}$ را به صورت زیر با بعد Kxn در نظر می‌گیریم.

$$\tilde{X} = \begin{bmatrix} -\tilde{x}_{(1)}- \\ \vdots \\ -\tilde{x}_{(n)}- \end{bmatrix} = \begin{bmatrix} -\tilde{x}_{(1)}- \\ \vdots \\ -\tilde{x}_{(k)}- \\ 0 \\ \vdots \\ 0 \end{bmatrix} \Rightarrow \tilde{X} = \begin{bmatrix} -\tilde{x}_{(1)}- \\ \vdots \\ -\tilde{x}_{(k)}- \end{bmatrix}_{Kxn}$$

حال برای اینکه variance هر feature برابر شود چون به آن‌ها متفاوت آن تقسیم می‌کنیم. حال داده‌های جدید $X_{whitened}$ را به صورت زیر تعریف می‌کنیم:

$$X_{whitened} = \Sigma^{-1/2}(U^T X')$$

زیرا $\Sigma^{-1/2}$ عکس ماتریس قطری شامل جذر eigen valve ها باشد.

که باعث در $\tilde{X}$ اعمال شود.

$$cov(X_{whitened}) = (\Sigma^{-1/2} U^T x')(\Sigma^{-1/2} U^T X')^T$$

حال ثابت می‌کنیم $I = cov(X_{whitened})$ به این صورت که

$$X_w X_w^T = \Sigma^{-1/2} U^T x' x'^T U \Sigma^{-1/2}$$

$$= \Sigma^{-1/2} U^T \underbrace{U}_{I} \Sigma \underbrace{U^{-1}}_{I} U \Sigma^{-1/2}$$

$$= \Sigma^{-1/2} \Sigma \Sigma^{-1/2} = I$$

از اثبات نیز در نظر نمی‌گیریم طوری که همین صفر باشد.

سوال ۳.۲)

برای به دست آوردن نقطه optimal در Ridge Regression از تابع cost آن مشتق گرفته و برابر صفر قرار می‌دهیم.

$$L(w) = \|Xw-y\|_2^2 + \lambda\|w\|_2^2$$
$$= (Xw-y)^T(Xw-y) + \lambda w^T w$$
$$= (w^T X^T - y^T)(Xw-y) + \lambda w^T w$$
$$= w^T X^T Xw - w^T X^T y - y^T Xw + y^T y + \lambda w^T w$$

$$\frac{\partial L(w)}{\partial w} = X^T Xw - X^T y + \lambda w$$

$$\frac{\partial L(w)}{\partial w} = 0 \Rightarrow X^T y = X^T Xw + \lambda w$$
$$(X^T X + \lambda I)w = X^T y$$
$$w^* = (X^T X + \lambda I)^{-1} X^T y$$

$w^* = (X^T X + \lambda I)^{-1} X^T y$ در نتیجه نقطه optimal برابر خواهد بود با

روش دیگر: خواهیم $\lambda\|w\|_2^2$ را صفر کنیم $L(w) = \|Xw-y\|_2^2$ را صفر کنیم

این $L(w)$ را می‌توانیم به صورت $\left\| B \right\|_2^2$ در نظر گرفته شده B به صورت $B = \begin{bmatrix} Xw-y \\ \sqrt{\lambda}\, w \end{bmatrix}$ باشد. حال می‌خواهیم B را فرم $Aw-b$ بنویسیم.

$$B = \begin{bmatrix} X \\ \sqrt{\lambda} I \end{bmatrix} w - \begin{bmatrix} y \\ 0 \end{bmatrix} = Aw - b$$

(A و b در زیر براکت‌ها)

که می‌خواهیم: $\|Aw-b\|_2^2$ را صفر کنیم که همان LSE است که جواب min آن

$$A^T A w^* = A^T b$$

شد که داشتیم.

پس داریم:

$$A^T A w^* = A^T b \qquad A = \begin{bmatrix} X \\ \sqrt{\lambda} I \end{bmatrix} \qquad b = \begin{bmatrix} y \\ 0 \end{bmatrix}$$

$$A^T A = \begin{bmatrix} X^T & \sqrt{\lambda} I \end{bmatrix} \begin{bmatrix} X \\ \sqrt{\lambda} I \end{bmatrix} = X^T X + \lambda I$$

$$A^T b = \begin{bmatrix} X^T & \sqrt{\lambda} I \end{bmatrix} \begin{bmatrix} y \\ 0 \end{bmatrix} = X^T y$$

$$\Rightarrow A^T A w^* = A^T b \Rightarrow (X^T X + \lambda I) w^* = X^T y$$

$$\Rightarrow \boxed{w^* = (X^T X + \lambda I)^{-1} X^T y}$$

سوال ۴)

الف) dimensionality reduction باعث کاهش overfitting میشود.
کاهش بعد باعث کاهش تعداد متغیرها میشود و کاهش تعداد متغیرها باعث کاهش پیچیدگی مدل میشود و overfitting را کاهش میدهد.
در این صورت که حضور دارد در PCA سعی میشود noise دیتا از بین برود و مدل را با ترکیب بخش های مهم دیتا را حفظ کند.

ب) data augmentation methods

=> data augmentation for image classification
یک راه ایجاد تصاویر جدید با استفاده از چرخش و وارونه سازی تصاویر میباشد.
راه دیگر استفاده از generative adversarial net برای تولید تصاویر مصنوعی جدید.

=> data augmentation for speech recognition
تولید داده های مصنوعی از MFCC های گفتاری باعث بهبود تشخیص یک
سیستم از گفته ها از طریق انتقال یادگیری در داده های مصنوعی از طریق CNN در سطح
کاراکتر RNN ایجاد شده است شود.

=> data augmentation for signal processing
جهت generative adversarial Network ها (به عنوان GCGAN) برای
تولید سیگنال های الکتروانسفالوگرافی مصنوعی استفاده کرده مطابق
سیگنال های نشان داده شده توسط مبتلایان به پارکینسون است.

استفاده از convolutional Neural Network برای تشخیص احساسات بسته
به EEG در تشخیص بهتر احساسات کمک میکند.

مدل Open AI GPT-2 قادر به تولید سیگنال های بیولوژیکی مصنوعی مانند EEG
و EMG میباشد.

سوال (۵) مساله را در حالت کلی n بعدی اثبات میکنیم $x, y \in \mathbb{R}^n$

(۵.۱) ابتدا با توجه به اینکه $\cos(x,y) = \frac{\langle x \cdot y \rangle}{|x||y|}$ میتوانیم تابع کرنل را به صورت زیر بازنویسی کنیم.

$$K(x,y) = (1+\cos(\angle x,y))^2 = 1 + 2\cos(\angle x,y) + \cos^2(\angle x,y)$$

$$= 1 + 2\frac{\langle x \cdot y \rangle}{|x||y|} + \frac{\langle x \cdot y \rangle \cdot \langle x,y \rangle}{|x|^2|y|^2}$$

$$= 1 + 2\frac{\sum_{i=1}^{n} x_i y_i}{|x||y|} + \frac{\left(\sum_{i=1}^{n} x_i y_i\right)\left(\sum_{j=1}^{n} x_j y_j\right)}{|x|^2|y|^2}$$

حال میتوانیم با صورت زیر تابع feature space ϕ را تعریف کنیم

$$\phi(x) = \begin{bmatrix} 1 \\ \sqrt{2}\frac{x_1}{|x|} \\ \vdots \\ \sqrt{2}\frac{x_n}{|x|} \\ \frac{x_1 x_1}{|x|^2} \\ \frac{x_1 x_2}{|x|^2} \\ \vdots \\ \frac{x_n x_n}{|x|^2} \end{bmatrix} \begin{matrix} \} \Rightarrow 1 \\ \} \Rightarrow \text{به تعداد } x_i \text{ها} \\ \} \Rightarrow \text{تمام حالات ممکن } x_i x_j \end{matrix}$$

حال داریم:

$$\langle \phi(x) \cdot \phi(y) \rangle = 1 + \sum_{i=1}^{n} \frac{\sqrt{2}x_i}{|x|} \frac{\sqrt{2}y_i}{|y|} + \sum_{i=1}^{n}\sum_{j=1}^{n} \frac{x_i x_j}{|x|^2} \frac{y_i y_j}{|y|^2}$$

$$= 1 + 2\frac{\sum_{i=1}^{n} x_i y_i}{|x||y|} + \frac{\sum_{i}\sum_{j}(x_i y_i)(x_j y_j)}{|x|^2|y|^2}$$

$$= 1 + 2\frac{\langle x \cdot y \rangle}{|x||y|} + \frac{(\sum x_i y_i)(\sum x_j y_j)}{|x|^2|y|^2}$$

$$= 1 + 2\frac{\langle x,y \rangle}{|x||y|} + \frac{\langle x,y \rangle \langle x,y \rangle}{|x|^2|y|^2}$$

$$= K(x,y) \Rightarrow K(x,y) = \langle \phi(x) \cdot \phi(y) \rangle \quad \checkmark$$

(۵.۲.۱) ماتریس K را خواهیم ثابت کنیم نیمه‌مثبت معین است کافی است ثابت کنیم:

$$\forall v \in \mathbb{R}^m ; \quad v^T K v \geq 0$$

بدین صورت شروع می‌کنیم:

$$v^T K v = [v_1 \ \cdots \ v_m] \begin{bmatrix} | & & | \\ c_1 & \cdots & c_m \\ | & & | \end{bmatrix} \begin{bmatrix} v_1 \\ \vdots \\ v_m \end{bmatrix}$$

$$v^T K v = \begin{bmatrix} \langle v, c_1 \rangle & \cdots & \langle v, c_m \rangle \end{bmatrix} \begin{bmatrix} v_1 \\ \vdots \\ v_m \end{bmatrix}$$

$$v^T K v = \sum_{i=1}^{m} \langle v, c_i \rangle v_i$$

$$= \sum_{i=1}^{m} \left(\sum_{j=1}^{m} v_j K_{j,i} \right) v_i$$

$$= \sum_{i=1}^{m} \sum_{j=1}^{m} v_i v_j K_{j,i}$$

$$= \sum_{i=1}^{m} \sum_{j=1}^{m} v_i v_j \phi(x^{(j)})^T \phi(x^{(i)})$$

$K_{j,i} = \langle \phi(x^{(j)}) \cdot \phi(x^{(i)}) \rangle = \phi(x^{(j)})^T \phi(x^{(i)})$

$$= \left\langle \left(\sum_{j=1}^{m} v_j \phi(x^{(j)}) \right) \cdot \left(\sum_{i=1}^{m} v_i \phi(x^{(i)}) \right) \right\rangle$$

$$= \left\| \sum v_i \phi(x^{(i)}) \right\|_2^2 \geq 0$$

پس نشان دادیم: $\forall v \in \mathbb{R}^m ; \ v^T K v \geq 0$

$\Rightarrow$ K نیمه‌مثبت معین است.

سوال ۲.۲.۵) خواهیم نشان دهیم اگر K یک ماتریس نیمه مثبت معین باشد.
آن‌گاه می‌توان $K(x,z) = \langle \phi(x) \cdot \phi(z) \rangle$ نوشت.

اگر K ماتریس نیمه معین مثبت باشد آن‌گاه متقارن نیز هست و می‌توان آن را قطری کرد
و بنابراین $K = PDP^T$ نوشت که در آن D ماتریس قطری با مقادیر ویژه
روی قطر آن و P ماتریس متعامد شامل بردارهای ویژه می‌باشد.

$$D = \begin{bmatrix} \lambda_1 & & 0 \\ & \ddots & \\ 0 & & \lambda_m \end{bmatrix} \quad \text{و} \quad P = \begin{bmatrix} | & | & & | \\ v_1 & v_2 & \cdots & v_m \\ | & | & & | \end{bmatrix}$$

باید $\phi(x)$ را تعریف کنیم. طوری که $K(x,z) = \langle \phi(x) \cdot \phi(z) \rangle$

ابتدا هر دو طرف ماتریس K را براساس فرم PDP^T می‌نویسیم.

$$K = \begin{bmatrix} v_1 & \cdots & v_m \end{bmatrix} \begin{bmatrix} \lambda_1 & & 0 \\ & \ddots & \\ 0 & & \lambda_m \end{bmatrix} \begin{bmatrix} v_1^T \\ \vdots \\ v_m^T \end{bmatrix}$$

$$K = \begin{bmatrix} \lambda_1 v_1 & \cdots & \lambda_n v_m \end{bmatrix} \begin{bmatrix} v_1^T \\ \vdots \\ v_m^T \end{bmatrix}$$

سطر iام ↓ ستون jام

$$K_{ij} = \langle r_i \cdot c_j \rangle$$

$$r_i = \begin{bmatrix} \lambda_1 v_i^{(1)} & \cdots & \lambda_m v_i^{(m)} \end{bmatrix}, \quad c_j = \begin{bmatrix} v_j^{(1)} \\ \vdots \\ v_j^{(m)} \end{bmatrix}$$

$$K_{ij} = \sum_{l=1}^{m} \lambda_l v_i^{(l)} v_j^{(l)} \quad (*)$$

حال کافی است $\phi(x_i)$ را به صورت زیر تعریف کنیم:

$$\phi(x_i) = \begin{bmatrix} \sqrt{\lambda_1} v_i^{(1)} \\ \vdots \\ \sqrt{\lambda_m} v_i^{(m)} \end{bmatrix}$$

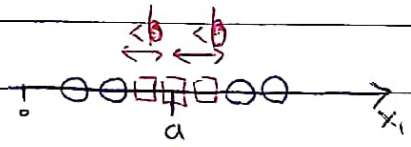
در این صورت داریم:

$$\langle \phi(x_i) \cdot \phi(x_j) \rangle = \sum_{l=1}^{n} \sqrt{\lambda_l} v_i^{(l)} \sqrt{\lambda_l} v_j^{(l)} \overset{(*)}{=} K_{ij}$$

$$\Rightarrow \langle \phi(x_i) \cdot \phi(x_j) \rangle = K_{ij} \quad \blacksquare$$

سوال 6)

(۴.۱



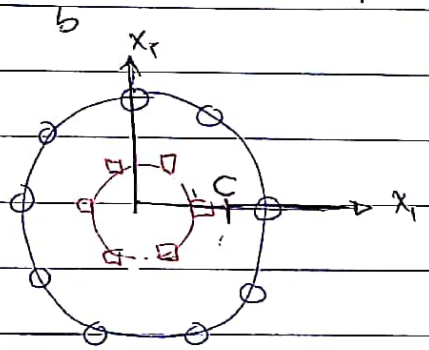
$\phi_1(x)$ را به صورت زیر تعریف می‌کنیم

$$\phi_1(x) = |x_1 - a|$$

حال کافی است یک threshold مانند b تعریف کنیم تا به راحتی cluster شوند.

در این صورت خواهیم داشت:

که داده‌ها خوشه‌بندی شوند.



با توجه به فاصله نقاط از مرکز داده‌ها را cluster می‌کنیم به این صورت که:

$$\phi_2(x_1, x_2) = (x_1^2 + x_2^2)$$

حال کافی است یک threshold به صورت c در نظر بگیریم. آن‌گاه

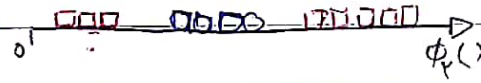
داده‌هایی که $\phi_2(x_1, x_2) < c$ در یک خوشه قرار می‌گیرند

و داده‌هایی که $\phi_2(x_1, x_2) > c$ در خوشه دیگر قرار می‌گیرند.

در این حالت ترکیبی از دو حالت بالا را داریم، به این صورت که ابتدا تبدیل اول

یعنی اعمال ϕ_2 بر روی داده‌ها باعث می‌شود داده‌ها مانند زیر شده باشد

که فاصله از مرکز را نشان می‌دهد.



حال با اعمال تبدیل ϕ_1 بر روی نتیجه قبل به خوبی آن‌ها را خوشه‌بندی کند.

پس تابع تبدیل در این حالت به فرم $\phi_1(\phi_2(x_1, x_2))$ خواهد بود و یک threshold

کافی است داده‌ها را خوشه‌بندی کند.

(۴.۲

$$(x_1-a)^2+(x_2-b)^2-r^2=0$$

$$x_1^2-2ax_1+a^2+x_2^2-2bx_2+b^2-r^2=0$$

$$(x_1^2)+(x_2^2)-2a(x_1)-2b(x_2)+(b^2+a^2-r^2)=0$$

پس در feature space که به صورت (x_1^2, x_2^2, x_1, x_2) تعریف شود معادله‌ی دایره به صورت یک معادله یک خط در فضای (x_1^2, x_2^2, x_1, x_2) می‌باشد با ضرایب $(-2b, -2a, 1, 1)$ و عرض از مبدأ $(b^2+a^2-r^2)$ می‌باشد.

پس در این فضای جدید linearly seprable می‌باشد.

سوال ۷)

می‌دانیم در کرنل خطی در SVM رابطه‌ی آن به صورت زیر می‌باشد

$$\min \quad \frac{1}{2}\|w\|_2^2 + C \sum (1 - y^{(i)} (w^T x^{(i)} + b))$$

که در آن C در واقع hyperparameter ای است که کنترل می‌کند میزان اهمیت میان اندازه margin و دقت cluster کردن داده‌ها.

هرچه C بزرگتر یعنی به اینکه داده‌ها درست cluster شده باشند (یعنی در سمت درست ابرصفحه و در فاصله‌ی درستی از آن (بزرگترین داخل حاشیه یا خارج آن) قرار گرفته باشند) اهمیت بیشتری نسبت به margin size می‌دهد.

و هرچه C کوچکتر به بیشینه کردن حاشیه اهمیت بیشتری می‌دهد و ضریب اشتباه پیش‌بینی کردن نمونه‌ها کمتر و اثر آن‌ها نیز در تابع loss کمتر می‌شود.

linear
C=10 ⇒ f
C=1 ⇒ b
C=0.1 ⇒ c

بنابراین در حالت‌های کرنل خطی یعنی شکل‌های b، f، c داریم:
شکل f کوچکترین حاشیه و در نتیجه بزرگترین C یعنی C=10 مربوط است.
شکل c بزرگترین حاشیه را دارد و در نتیجه کوچکترین C یعنی C=0.1 مربوط است.
و در آخر شکل b با حاشیه متوسط متناظر C متوسط یعنی C=1 می‌باشد.

در کرنل RBF داریم که λ hyperparameter میزان هموار بودن مرزها را کنترل می‌کند
بدین صورت که هرچه λ مقدار کمتری داشته باشد مرزها شکل smooth تری هستند

RBF
$\lambda = 10 \Rightarrow a$
$\lambda = 1 \Rightarrow e$
$\lambda = 0.1 \Rightarrow d$

بدین صورت داریم در شکل‌ها با RBF Kernel : یعنی شکل‌های (a, d, e)

مرزها در شکل d، smooth تر هستند چون متناظر با کمترین λ یعنی $\lambda = 0.1$ می‌باشد.
مرزها در شکل a، کمترین میزان smoothness را دارند چون متناظر با بیشترین λ یعنی $\lambda = 10$ می‌باشد.
و در آخر مرزها شکل e از لحاظ smoothness میزانی بین دو شکل قبل و در نتیجه متناظر با λ با مقدار میانه یعنی $\lambda = 1$ می‌باشد.
البته تاثیر C نیز مانند قبل می‌باشد که C بیشتر margin کمتر.

سوال ۸)

الف) در رابطه با تاثیر اندازه training set به biase-var tradeoff داریم:
هرچه اندازه training set بیشتر باشد با دیدن نمونه‌های بیشتر مدل دقیق‌تر می‌شود و در واقع اطلاعات جدید اضافه شده و باعث فیت تر شدن مدل بر روی دیتا می‌شود که در نتیجه بایاس کمتر و واریانس بیشتر می‌شود

و در حالتی که اندازه n کمتر، بایاس بیشتر و واریانس کمتر می‌شود.

با افزایش سایز training set باعث شباهت بین هر نمونه می‌شود و باعث overfitting می‌شود و بایاس کم و واریانس زیاد می‌شود.

ب) در k-fold هر نقطه یکبار تست می‌شود و k-1 دفعه train می‌شود. در روش MCCV تقسیم‌بندی‌های بیشتری امتحان می‌شوند البته همه‌ی این تقسیم‌بندی‌ها استفاده نمی‌شوند.

میانگین گرفتن از نرخ خطا MSE در k-fold باعث می‌شود مدل نامطلوب با واریانس زیاد داشته باشیم، در MCCV یک تخمین از بایاس و واریانس کمتر می‌دهد.