سوال ۱ : الف / علت استفاده از اعتبارسنجی متقابل (Cross_Validation) چیست ؟

هدف از Cross_Validation این است که به چه میزان مدل می‌تواند روی داده هایی که روی آن ها train نشده بود به خوبی پیش‌بینی ها را انجام بدهد.

• مزایای استفاده از این روش : train نشده روی آن

۱) جلوگیری کردن از اورفیت (Overfitting) کردن مدل روی داده های
ها : زمانی که یک مدل بیش از حد به داده های آموزشی وابسته شده و نتایج خوبی را روی
آن ها بدست می‌آورد. اما نمی‌تواند داده های جدید را به خوبی پیش‌بینی کرده باشد
مدل ما overfit کرده است و این یعنی پیچیدگی مدل به صورت غیر ضروری بالا رفته
است. مدل اعتبارسنجی کمک می‌کند تا مدل به گونه‌ای آموزش دیده باشد که
روی داده های test هم به خوبی کارکرده باشد.

• به نوعی در این حالت ما تمام داده ها را به چندین بخش به train و test تقسیم بندی می‌کنیم

۲) استفاده بهینه از داده ها :
می‌دانیم که در حالت عادی داده ها را به دو مجموعه train و test تقسیم بندی کرده بودیم
که این روش می‌تواند منجر به از دست رفتن بخش بزرگی از داده ها برای آموزش مدل بشود.
که با استفاده از اعتبارسنجی متقابل می‌توان از یک داده ها هم برای آموزش و هم برای تست استفاده کرد.
در نتیجه داده های ما به صورت بهینه‌تری استفاده می‌شوند.

• در مورد از روش های اعتبارسنجی :
۱) K-fold — در این روش ما داده ها را به دسته هایی k تایی تقسیم بندی می‌کنیم.
و به هر بار یکی از این دسته ها را به عنوان test ~~train~~ در نظر می‌گیریم
و با باقی دسته ها مدل را train می‌کنیم. و به همین ترتیب مکان test را عوض
می‌کنیم و مدل را ارزیابی می‌کنیم.
به نوعی ما در هر تکرار یکی از بخش ها به عنوان داده های آزمایشی (test) و بقیه را به عنوان train
در نظر می‌گیریم که این فرایند به تعداد k بار تکرار خواهد شد.

مزیت ها ← از داده ها به صورت کارآمد تری استفاده می کنیم
که چون در هر مرحله بخش های مختلفی از داده ها آزمایش می‌شوند مدل به خوبی
روی تمام قسمت ها train می شود.

معایب ← اگر k بزرگ باشد محاسبات به طور قابل توجهی افزایش خواهد یافت

. LOOCV: . leave-One-Out-Cross-Validation:

این روش حالت خاصی از k-fold می باشد در هر بار یک نمونه برای تست استفاده می شود و
و باقی داده ها برای آموزش مدل استفاده می شوند.

و این فرایند باید به تعداد کل نمونه های که داریم اجرا کنیم، به طور مثال اگر n تا نمونه داشته باشیم
باید به تعداد n بار ران کنیم الگوریتم را.

مزیت ها ← این روش بیشترین استفاده از داده های آموزشی را می کند که مدل را به خوبی train
کند
و نتیجه ی نهایی هم دقیق تر بسیار دقیق خواهد بود.

تنها معایب این روش این است که از لحاظ محاسباتی بسیار سنگین است چون داریم روی هر کدام از داده ها
یک بار الگوریتم را اجرا می کنیم.

• قسمت ب) متریک فاصله اقلیدسی :

$$D(x,y) = \sqrt{\sum_{k=1}^{d} (x_k - y_k)^2}$$

• حال اگر هر بعد را در یک مقدار حقیقی ضرب کنیم، باز هم معیار فاصله خواهیم داشت.
به ترتیب خصوصیات یک معیار فاصله را بررسی می‌کنیم: • چون هر بعد را باید در یک a_k ضرب می‌کردیم.

$$D(x',y') = \sqrt{\sum_{k=1}^{d} (a_k x_k - y_k a_k)^2} = \sqrt{\sum_{k=1}^{d} a_k^2 (x_k - y_k)^2} = \sqrt{\sum_{k=1}^{d} a_k^2 (x_k - y_k)^2}$$

① غیر منفی بودن: چون در هر کدام از $(x_k - y_k)^2$ یک عدد مثبت ضرب کرده‌ایم $a_k^2 \geq 0$
پس نتیجه‌ی آن هم یک عدد مثبت خواهد شد چون $D(x,y) \geq 0$ بود.

• چون که $a_k^2 \geq 0$ و $(x_k - y_k)^2 \geq 0$ این برای هر x_k و y_k مقدار فاصله آن همیشه غیر منفی است.

② Identity of Indiscernibles که یعنی $D(x,y) = 0$ اگر و تنها اگر $y = x$ باشد
از آن جا که $a_k \neq 0$ غیر صفر فرض شده‌اند. تنها زمانی مقدار $D(x,y)$ صفر خواهد شد که $x = y$ شده باشد. پس a_kها تاثیری در صفر شدن فاصله ندارند.

③ تقارن: $D(x,y) = D(y,x)$ ← از آن جایی که در رابطه $(x_k - y_k)^2$ با $(y_k - x_k)^2$ برابر است
و a_k به شکل ضریب در هر دو آمده بود. پس باز هم برقرار است.

$$D(x',y') = \sqrt{\sum_{k=1}^{d} a_k^2 (x_k - y_k)^2}$$

$$D(y',x') = \sqrt{\sum_{k=1}^{d} a_k^2 (y_k - x_k)^2}$$

و میدانیم این دو با هم برابر هستند چون

$$(y_k - x_k)^2 = (x_k - y_k)^2$$

④ نابرابری مثلث

$$D(y,z) + D(x,y) \geq D(x,z)$$

← از آن جایی که فاصله اقلیدسی استاندارد نابرابری مثلث را برآورده می‌کند و فقط ضریب a_k به آن افزوده شده است پس همچنان برقرار خواهد بود.

سوال (۲):

الف) تعریف L1 Regularization و L2 Regularization

• هدف این است که مدل را از پیچیدگی زیاد دور نگه داریم: پس یک ترم کمکی به loss اضافه می‌کنیم که آن میزان w_i ها را کنترل کند. پنالتی آن به این صورت تعریف می‌شود:

$$L1\ Penalty = \lambda \sum_{i=1}^{d} |w_i|$$

که در این جا λ ضریب تنظیم کننده یا همان تعیین کننده میزان جریمه است که با افزایش آن میزان وزن‌ها بیشتر کاهش پیدا می‌کند و میزان Regularization هم بیشتر خواهد شد.

ویژگی مهم L1 این است که ممکن است که بعضی از ضریب‌ها را به طور کامل صفر کند. پس L1 یک مدل ساده‌تر با ویژگی‌های کمتر ایجاد می‌کند.

حال در L2 پنالتی به این صورت است که:

$$L2\ Penalty = \lambda \sum_{i=1}^{d} |w_i|^2$$

که در این حالت همه ضرایب را کوچک می‌کند اما به صفر نمی‌رساند. پس برای انتخاب کردن ویژگی‌ها مناسب نیست و برای کنترل کردن پیچیدگی مدل مناسب است.

• L1 معمولا مدل‌های ساده‌تری را ایجاد می‌کند اما L2 ضرایب بزرگ‌تر را کاهش می‌دهد.

ضرایب خطی که در حالت کلی ضرایب یا وزن‌ها در مدل نشان دهنده این است که تا چه اندازه ویژگی‌ها بر خروجی تاثیر دارند و زمانی که ضرایب بزرگ باشد به این معنا خواهد بود که مدل به ویژگی‌های بزرگتر حساس هستند و ... است پس با توجه به بزرگترین ضرایب:

• مدل به ویژگی‌های خاص حساس می‌شود.
• تاثیر ویژگی‌های مختلف کاهش می‌یابد.
• مدل نسبت به نویزها و داده‌های غیرمعمول واکنش نشان می‌دهد.

• فرم بسته متغیر تصمیمی w را اگر داریم می‌بایست L(w) نسبت به w مشتق گرفته و مساوی با صفر قرار دهیم :

$$L(w) = \sum_{i=1}^{n} (w^T x_i - y_i)^2 + \lambda \|w\|_2^2$$

• in <u>Matrix form</u>:

$$L(w) = (Xw - y)^T (Xw - y) + \lambda w^T w$$

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

• حال گرادیان را برابر با صفر قرار دهیم.

$$2X^T(Xw - y) + 2\lambda w = 0$$

$$2X^T(Xw) - 2X^T y + 2\lambda w = 0$$

$$(X^T X + \lambda I) w = X^T y$$

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

بهینه‌ی w •

Scanned with CamScanner

سوال ۳ :

$$D = \{(x_1, y_1), (x_2, y_2) \dots (x_n, y_n)\}$$

• در واقع رابطه احتمالاتی میان x و y به این شکل است که:

$$y_i = w x_i + \varepsilon_i \qquad \varepsilon_i \sim N(0,1) \longrightarrow$$

• نرمال [illegible]

• پس همانطور که گفتیم به دلیل آن که ε_i نویزی از احتمالات نرمال می‌باشد پس توزیع y را به این صورت می‌توان با توجه به میانگین آن نوشت:

$$y_i \sim N(w x_i, 1)$$

• حال اگر به صورت تابع بنویسیم داریم که:

$$P(y_i \mid w) = \frac{1}{\sqrt{2\pi}\,(1)} \exp\left(-\frac{(y_i - w x_i)^2}{2}\right)$$

حال از آن یک likelihood بگیریم:

$$P(D \mid w) = \prod_{i=1}^{n} p(y_i \mid w) = \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{(y_i - w x_i)^2}{2}\right)$$

$$\log(P(D \mid w)) = \sum_{i=1}^{n} \log\frac{1}{\sqrt{2\pi}} + -\left(\frac{(y_i - w x_i)^2}{2}\right)$$

$$= \underbrace{\sum_{i=1}^{n} \left(-\log(\sqrt{2\pi})\right)}_{\text{Constant}} - \frac{1}{2} \sum_{i=1}^{n} (y_i - w x_i)^2$$

$$\Rightarrow +\log P(D \mid w) = -\frac{1}{2} \sum_{i=1}^{n} (y_i - w x_i)^2$$

- **maximizing on the log-liklihood:**

$$\Rightarrow \max_{w} \log P(D|w) = \max_{w} \left(-\frac{1}{2} \sum_{i=1}^{n} (y_i - w x_i)^2 \right)$$

$$= \min \sum_{i=1}^{n} (y_i - w x_i)^2$$

یعنی باید این عبارت minimize
بشه

$$\Rightarrow \arg\max_{w} \log P(D|w) = \arg\min_{w} \sum_{i=1}^{n} (y_i - w x_i)^2$$

• پاسخ سوال چهارم : رابطه‌ی بین ورودی و خروجی را به این صورت مدل سازی کرده‌ایم :

$$y = \exp(wx) = y = e^{wx}$$

مجموعه داده‌ای که در اختیار داریم $\longrightarrow D = \{(x_1,y_1),(x_2,y_2),\dots(x_n,y_n)\}$

• برای تابع هزینه‌ی مجموع مربع خطا داریم :

Cost of SSE → Sum of Squared Errors

$$J(w) = \sum_{i=1}^{n} \left(y_i - \exp(wx_i)\right)^2$$

• بر اساس روش گرادیان کاهشی داریم :

$$w_{t+1} = w_t - \alpha \frac{\partial J(w)}{\partial w}$$

α → learning rate

$$-\frac{\partial J(w)}{\partial w} = \frac{\partial}{\partial w}\left(\sum_{i=1}^{n} \left(y_i - e^{wx_i}\right)^2\right)$$

$$= (2)\sum_{i=1}^{n}\left(y_i - e^{wx_i}\right)\cdot\frac{\partial}{\partial w}\left(y_i - e^{wx_i}\right)$$

$$= 2\sum_{i=1}^{n}\left(y_i - e^{wx_i}\right)e^{wx_i}x_i$$

$$\Rightarrow \underbrace{\boxed{w_{t+1}}}_{\text{پارامتر آپدیت شده}} = w_t + 2\alpha\sum_{i=1}^{n}\left(y_i - e^{(wx_i)}\right)e^{(wx_i)}\cdot x_i$$

$$\frac{\partial J(w)}{\partial w} = 0 \rightarrow \underline{y_i = e^{x_i w}}$$

$$\rightarrow \sum_{i=1}^{n} x_i e^{2wx_i} = \sum_{i=1}^{n} x_i e^{wx_i} e^{wx_i} = \underline{\sum_{i=1}^{n} y_i x_i e^{wx_i}}$$