به نام خدا

تمرین سری٤ علوم داده

پارسا آقاعلی ٤٠٠٥٢١٠٧٢

سوال ۱

Regularizationیکی از تکنیک‌های مهم در یادگیری ماشین است که برای بهبود عملکرد مدل و جلوگیری از **Overfitting** استفاده می‌شود Overfitting .زمانی رخ می‌دهد که مدل، به جای یادگیری الگوهای کلی موجود در داده‌ها، نویز یا جزئیات خاص داده‌های آموزشی را بیش از حد تطبیق می‌دهد. این امر باعث می‌شود مدل روی داده‌های جدید عملکرد ضعیفی داشته باشد.

Regularizationبا افزودن یک **پنالتی** (جریمه) به تابع هزینه، مدل را مجبور می‌کند پیچیدگی خود را کاهش دهد و وزن‌های غیرضروری را کوچک کند یا به صفر نزدیک کند. به عبارت دیگر، این روش کمک می‌کند مدل به جای تمرکز روی جزئیات بیش از حد داده‌های آموزشی، به الگوهای کلی‌تر و عمومی‌تر توجه کند.

مکانیسم Regularization

در یادگیری ماشین، مدل‌ها معمولاً با **تابع هزینه** آموزش داده می‌شوند که هدف آن به حداقل رساندن خطای پیش‌بینی است . Regularizationیک ترم اضافی به تابع هزینه اضافه می‌کند که میزان پیچیدگی مدل را کنترل می‌کند. به صورت کلی، Regularization به دو نوع رایج تقسیم می‌شود:

1. L1 Regularization (Lasso)

در این روش، ترم Regularization برابر با مجموع مقادیر مطلق وزن‌ها است:

$$J(\theta) = Loss + \lambda \sum_{i=1}^{n} |\theta_i|$$

- **ویژگی اصلی** :باعث می‌شود برخی از وزن‌ها دقیقاً برابر با صفر شوند. این ویژگی باعث **انتخاب ویژگی** می‌شود، زیرا وزن صفر به معنای حذف آن ویژگی است.
- **کنترل پیچیدگی مدل** :مدل را ساده‌تر می‌کند زیرا تعداد ویژگی‌های غیرضروری کاهش می‌یابد.

2. L2 Regularization (Ridge)

در این روش، ترم Regularization برابر با مجموع مربعات وزن‌ها است:

$$J(\theta) = Loss + \lambda \sum_{i=1}^{n} \theta_i^2$$

- **ویژگی اصلی** :وزن‌ها را به مقادیر کوچکتر نزدیک می‌کند ولی آن‌ها را صفر نمی‌کند.
- **کنترل پیچیدگی مدل** :مدل را روان‌تر و کمتر حساس به نویز می‌کند.

چگونه Regularization باعث جلوگیری از Overfitting می‌شود؟

۱. **کاهش پیچیدگی مدل** Regularization :وزن‌های بزرگ را جریمه می‌کند و مدل را مجبور می‌کند تا از وابستگی بیش از حد به برخی ویژگی‌ها یا نویزها خودداری کند.

۲. **جلوگیری از تطابق بیش از حد با داده‌های آموزشی** :با کوچک کردن وزن‌ها، مدل نمی‌تواند داده‌های آموزشی را بیش از حد به خاطر بسپارد، در نتیجه، تعمیم‌پذیری روی داده‌های جدید افزایش می‌یابد.

۳. **کنترل انعطاف‌پذیری مدل** Regularization :مدل را از یادگیری توابع پیچیده (مانند چندجمله‌ای‌های مرتبه بالا) باز می‌دارد.

۴. **حذف ویژگی‌های غیرضروری** L1 Regularization :می‌تواند ویژگی‌هایی را که اهمیت کمتری دارند، حذف کند و به این ترتیب، مدل را ساده‌تر کند.

Regularization یک ابزار قدرتمند برای جلوگیری از Overfitting است که با محدود کردن وزن‌ها، تعادل مناسبی بین **خطای بایاس (Bias)**و **واریانس (Variance)** برقرار می‌کند. انتخاب نوع (L1 یا L2) Regularization بسته به مسئله و نیاز مدل متفاوت است، اما هر دو در بهبود تعمیم‌پذیری و کاهش حساسیت مدل به نویز داده‌ها مؤثر هستند.

سوال ۲

برای **Feature Selection**یا انتخاب ویژگی، روش **L1 Regularization**که به آن **Lasso** نیز گفته می‌شود، مناسب‌تر است. این به دلیل ویژگی خاص L1 در حذف وزن‌های غیرضروری و ساده‌تر کردن مدل است. در ادامه به توضیح کامل این موضوع می‌پردازیم.

تفاوت L1 و L2 در Regularization

در Regularization ، هدف این است که وزن‌های (Coefficients) مدل محدود شوند تا از پیچیدگی بیش از حد مدل جلوگیری شود. این کار از طریق اضافه کردن یک **Penalty Term** به تابع هزینه انجام می‌شود.

1. L1 Regularization (Lasso):

- در این روش، **مجموع مقادیر مطلق وزن‌ها** به عنوان جریمه به تابع هزینه اضافه می‌شود :

$$J(\theta) = |_i\lambda \sum_{i=1}^{n} |\theta + Loss$$

- **ویژگی کلیدی** :این جریمه باعث می‌شود برخی از وزن‌ها دقیقاً به **صفر** برسند. زمانی که وزن یک ویژگی صفر شود، آن ویژگی از مدل حذف می‌شود. به همین دلیل L1 به‌صورت مؤثری انتخاب ویژگی انجام می‌دهد.

2. L2 Regularization (Ridge):

- در این روش، **مجموع مربعات وزن‌ها** به عنوان جریمه به تابع هزینه اضافه می‌شود :

$$J(\theta) = \frac{2}{i}\lambda \sum_{i=1}^{n} \theta + Loss$$

- **ویژگی کلیدی** L2 :وزن‌ها را کوچک می‌کند اما آن‌ها را به صفر نمی‌رساند. به همین دلیل L2 نمی‌تواند ویژگی‌های غیرضروری را حذف کند، بلکه تنها تأثیر آن‌ها را کاهش می‌دهد.

چرا L1 برای Feature Selection مناسب‌تر است؟

۱. **حذف کامل ویژگی‌های غیرضروری:**

- L1 Regularizationوزن ویژگی‌های کم‌اهمیت را به صفر می‌رساند. وقتی وزن یک ویژگی صفر باشد، آن ویژگی دیگر تأثیری در پیش‌بینی مدل ندارد و می‌توان آن را از مدل حذف کرد. این عمل انتخاب ویژگی را به‌صورت خودکار انجام می‌دهد.

۲. **ساده‌تر کردن مدل:**

- با حذف ویژگی‌های غیرضروری، مدل ساده‌تر می‌شود و از **Overfitting**جلوگیری می‌کند. این ویژگی به‌ویژه زمانی مفید است که با مجموعه داده‌های با ابعاد بالا کار می‌کنیم.

۳. **تعامل مستقیم با انتخاب ویژگی:**

- درL1 Regularization ، جریمه بر اساس مقادیر مطلق وزن‌ها اعمال می‌شود. این باعث می‌شود که مدل به جای کاهش همه وزن‌ها، وزن‌های خاصی را به صفر برساند و در نتیجه به‌طور مستقیم روی انتخاب ویژگی تأثیر بگذارد.

چرا L2 برای Feature Selection مناسب نیست؟

- L2 Regularizationوزن‌ها را کوچک می‌کند اما هیچ‌کدام از آن‌ها را به صفر نمی‌رساند. به همین دلیل، حتی ویژگی‌هایی که اهمیت کمتری دارند، همچنان تأثیرگذار باقی می‌مانند.
- اگر هدف شما حذف کامل ویژگی‌های غیرضروری باشد، L2نمی‌تواند این کار را انجام دهد. به همین دلیل، L2 برای **Feature Selection** مناسب نیست، اما برای **کاهش واریانس** و جلوگیری از Overfitting بسیار مفید است.

مثال ساده:

فرض کنید مجموعه داده‌ای با ۱۰ ویژگی داریم و تنها ۳ ویژگی واقعاً در پیش‌بینی تأثیرگذار هستند. اگر از **L1 Regularization** استفاده کنیم، مدل وزن ۷ ویژگی غیرضروری را به صفر می‌رساند و آن‌ها را حذف می‌کند. اما اگر از **L2 Regularization** استفاده کنیم، وزن تمام ویژگی‌ها تنها کاهش می‌یابد و هیچ‌کدام به صفر نمی‌رسد.

نتیجه‌گیری:

- **L1 Regularization (Lasso)** برای **Feature Selection** مناسب‌تر است زیرا می‌تواند وزن برخی ویژگی‌ها را به صفر برساند و آن‌ها را از مدل حذف کند. این ویژگی باعث می‌شود مدل ساده‌تر و تعمیم‌پذیرتر شود.
- **L2 Regularization (Ridge)** وزن‌ها را کوچک می‌کند ولی آن‌ها را حذف نمی‌کند. بنابراین برای **انتخاب ویژگی** مناسب نیست، اما برای کاهش پیچیدگی مدل و جلوگیری از Overfitting مفید است.

سوال ۳

Transfer Learningو **Domain Adaptation** هر دو روش‌هایی هستند که در یادگیری ماشین برای استفاده از دانش موجود در یک دامنه برای بهبود عملکرد در یک دامنه جدید استفاده می‌شوند. با این حال، این دو مفهوم اهداف، کاربردها و فرضیات متفاوتی دارند. در ادامه به تعریف، تفاوت‌ها و مقایسه این دو روش پرداخته می‌شود.

Transfer Learningیا یادگیری انتقالی، فرآیندی است که در آن دانش به دست آمده از یک دامنه منبع (source domain) به یک دامنه هدف (target domain) منتقل می‌شود. این روش به‌ویژه زمانی مفید است که دامنه منبع و دامنه هدف به هم مرتبط باشند، اما داده‌های موجود در دامنه هدف کم یا محدود باشند.

مثال:

فرض کنید یک مدل برای تشخیص تصاویر حیوانات (مانند گربه‌ها و سگ‌ها) آموزش داده شده است. می‌توان از وزن‌ها و ویژگی‌های یادگرفته‌شده در این مدل برای تشخیص تصاویر پرندگان استفاده کرد، بدون نیاز به آموزش کامل مدل از ابتدا.

ویژگی‌های کلیدی:

- دامنه منبع و دامنه هدف می‌توانند متفاوت باشند، اما باید شباهت‌هایی بین آن‌ها وجود داشته باشد.
- ممکن است فضای ویژگی‌ها و توزیع داده‌ها بین دو دامنه متفاوت باشد.
- هدف: انتقال دانش عمومی که قبلاً یاد گرفته شده است، مانند ویژگی‌های لبه‌ها، اشکال، یا الگوهای متنی.

Domain Adaptationیا انطباق دامنه، یکی از شاخه‌های خاص **Transfer Learning**است که به‌طور خاص برای زمانی طراحی شده است که دامنه منبع و دامنه هدف، فضای ویژگی مشابهی داشته باشند، اما توزیع داده‌ها بین آن‌ها متفاوت باشد. هدف Domain Adaptationاین است که مدل بتواند توزیع داده‌های دامنه هدف را یاد بگیرد و عملکرد بهتری در این دامنه ارائه دهد.

مثال:

فرض کنید یک مدل برای تشخیص دست‌خط انگلیسی آموزش دیده است. اگر بخواهیم این مدل را برای تشخیص دست‌خط فارسی استفاده کنیم، باید مدل را با داده‌های جدید تطبیق دهیم زیرا توزیع داده‌ها (شکل و ساختار دست‌خط) تغییر کرده است.

ویژگی‌های کلیدی:

- فضای ویژگی‌ها در دامنه منبع و دامنه هدف یکسان است.
- تفاوت اصلی در توزیع داده‌ها (data distribution) است.
- هدف: مدل را با توزیع داده‌های دامنه هدف سازگار کند.

تفاوت‌های اصلی Transfer Learning وDomain Adaptation

ویژگی	Transfer Learning	Domain Adaptation
هدف	انتقال دانش کلی از یک دامنه به دامنه دیگر	سازگار کردن مدل با تفاوت در توزیع داده‌ها
شباهت دامنه‌ها	دامنه منبع و دامنه هدف ممکن است متفاوت باشند	فضای ویژگی‌ها مشابه است، اما توزیع داده‌ها متفاوت است
نیاز به داده‌های جدید	ممکن است به داده‌های جدید نیازی نباشد یا داده‌های کم کافی باشد	نیاز به داده‌های دامنه هدف برای تطبیق مدل
پیاده‌سازی	استفاده از مدل از قبل آموزش‌دیده و fine-tuning روی دامنه جدید	تغییر وزن‌ها یا اضافه کردن تنظیمات برای انطباق با داده‌ها
کاربرد	دامنه‌های مختلف با شباهت کلی، مثلاً از تصاویر حیوانات به تصاویر وسایل نقلیه	دامنه‌های مشابه با داده‌های متفاوت، مثلاً از تصاویر روز به تصاویر شب

شباهت‌ها

۱. **هدف مشترک :**هر دو روش برای استفاده مؤثر از دانش موجود در یک دامنه به منظور بهبود عملکرد در دامنه هدف طراحی شده‌اند.

۲. **کاربرد در یادگیری کم‌داده :**هر دو روش زمانی مفید هستند که داده‌های دامنه هدف محدود باشند.

۳. **ارتباط با یادگیری عمیق :**در هر دو روش، از مدل‌های از قبل آموزش‌دیده استفاده می‌شود (مانند شبکه‌های عصبی) و آن‌ها را برای کاربردهای جدید تنظیم می‌کنند.

کاربردهای رایج

Transfer Learning:

- استفاده از مدل‌های از قبل آموزش‌دیده مانند ResNet یا BERT برای مسائل مختلف.
- تشخیص اشیا در تصاویر، طبقه‌بندی متن، تشخیص گفتار.

Domain Adaptation:

- ترجمه ماشینی (Machine Translation) برای زبان‌های متفاوت.
- تشخیص اشیا در شرایط نوری متفاوت (مثلاً از تصاویر روز به تصاویر شب).
- یادگیری در شرایط خاص مانند انتقال از داده‌های شبیه‌سازی‌شده به داده‌های واقعی.

نتیجه

- **Transfer Learning**برای مسائل کلی‌تر و انتقال دانش عمومی میان دامنه‌های مختلف استفاده می‌شود. این روش زمانی مفید است که دامنه منبع و دامنه هدف تفاوت‌های زیادی داشته باشند.
- **Domain Adaptation**شاخه‌ای خاص از Transfer Learning است که زمانی استفاده می‌شود که تفاوت‌ها تنها در **توزیع داده‌ها** باشد. این روش بیشتر روی سازگاری مدل تمرکز دارد.

سوال ٤

الگوریتم **K-means**یکی از پرکاربردترین روش‌های خوشه‌بندی است که هدف آن تقسیم داده‌ها به kخوشه است. k تعداد خوشه‌ها را مشخص می‌کند و تأثیر مستقیمی بر نتایج خوشه‌بندی دارد. انتخاب تعداد مناسب kیکی از چالش‌های اصلی در استفاده از این الگوریتم است، زیرا مقدار اشتباه می‌تواند منجر به **Overfitting**یا **Underfitting**شود.

روش‌های انتخاب K

۱. **روش (Elbow) خم آرنج:**

- این روش یکی از رایج‌ترین روش‌ها برای انتخاب Kاست. در این روش، معیار **مجموع مربعات خطا درون خوشه‌ها (Within-Cluster Sum of Squares - WCSS)**محاسبه می‌شود.
- **WCSS**بیانگر فاصله داده‌ها از مرکز خوشه خود است

$$WCSS = \sum_{i=1}^{K} \sum_{x \in C_i} ||x - \mu_i||^2$$

- مقدار Kرا طوری انتخاب می‌کنیم که کاهش WCSS پس از آن مقدار، ناچیز باشد. این نقطه به "خم آرنج" نمودار معروف است.
- **مزیت :**روشی ساده و قابل تفسیر.
- **معایب :**گاهی ممکن است نقطه خم واضح نباشد.

۲. روش:Silhouette Score

- این معیار، کیفیت خوشه‌بندی را اندازه‌گیری می‌کند **Silhouette Score** .برای هر داده مشخص می‌کند که چقدر به خوشه خود نزدیک است و چقدر از خوشه‌های دیگر فاصله دارد :

$$S = \frac{b - a}{\max(a, b)}$$

  - S:امتیاز سیلوئت برای هر داده.
  - a:میانگین فاصله داده از سایر نقاط در خوشه خود.
  - b:میانگین فاصله داده از نقاط نزدیک‌ترین خوشه دیگر.
- مقدار Kکه بالاترین مقدار Silhouette Score را دارد، انتخاب می‌شود.
- **مزیت :**ارزیابی کیفیت خوشه‌بندی.
- **معایب :**زمان‌بر برای مجموعه داده‌های بزرگ.

۳. روش:Gap Statistic

- این روش توزیع داده‌ها در خوشه‌های واقعی را با توزیع داده‌ها در خوشه‌های تصادفی مقایسه می‌کند.
- **مزیت :**شناسایی بهینه‌ترین تعداد خوشه‌ها با در نظر گرفتن توزیع داده‌ها.

- **معایب :**محاسبات پیچیده‌تر نسبت به روشElbow

۴. **روش‌های مبتنی بر(Domain Knowledge) دانش دامنه:**

- در برخی موارد، تعداد خوشه‌ها بر اساس دانش قبلی در مورد داده‌ها تعیین می‌شود.
- **مثال :**در تحلیل مشتریان، ممکن است تعداد خوشه‌ها برابر با تعداد گروه‌های مختلف بازار باشد.

Overfittingدر الگوریتمK-means

۱. **مفهوم Overfitting درK-means :**

- Overfittingزمانی رخ می‌دهد که تعداد خوشه‌ها (K) بیش از حد بزرگ انتخاب شود. در این حالت:
  - هر خوشه بسیار کوچک شده و حتی ممکن است تنها شامل یک یا دو داده باشد.
  - خوشه‌ها دیگر نمایانگر الگوهای عمومی داده‌ها نیستند و به جای آن به نویز یا جزئیات خاص داده‌ها وابسته می‌شوند.

۲. **چرا Overfitting در K-means رخ می‌دهد؟**

- با افزایش K، مقدار **WCSS**کاهش می‌یابد زیرا هر داده به خوشه‌ای نزدیک‌تر به خود اختصاص می‌یابد.
- این کاهش می‌تواند گمراه‌کننده باشد، زیرا مدل بیش از حد به داده‌های آموزشی وابسته می‌شود و تعمیم‌پذیری آن کاهش می‌یابد.

۳. **تأثیرOverfitting :**

- مدل توانایی تشخیص الگوهای کلی در داده‌ها را از دست می‌دهد.
- عملکرد مدل روی داده‌های جدید کاهش می‌یابد.

جلوگیری از Overfitting درK-means

۱. **انتخاب مناسب :k**

- استفاده از روش‌های ارزیابی مانندElbow ، Silhouette Scoreیا.Gap Statistic
- بررسی کیفیت خوشه‌ها و جلوگیری از انتخاب kبیش از حد بزرگ.

۲. **(Cross-Validation)اعتبارسنجی متقابل:**

- داده‌ها را به بخش‌های آموزشی و آزمون تقسیم کنید و بررسی کنید که آیا خوشه‌بندی در داده‌های آزمون نیز قابل تعمیم است.

۳. **(Regularization)تنظیم‌سازی:**

- افزودن محدودیت یا جریمه برای تعداد زیاد خوشه‌ها.

۴. **دانش دامنه:**

- استفاده از دانش قبلی درباره داده‌ها برای محدود کردن مقدار Kبه یک بازه معقول.

نتیجه‌گیری

- انتخاب تعداد مناسب k در الگوریتم K-means از اهمیت بالایی برخوردار است. روش‌هایی مانند Elbow و Silhouette Scoreمی‌توانند در شناسایی مقدار بهینه کمک کنند.
- Overfittingدر K-means زمانی رخ می‌دهد که تعداد خوشه‌ها بیش از حد بزرگ انتخاب شود و مدل الگوهای عمومی را از دست بدهد.
- برای جلوگیری ازOverfitting ، باید از روش‌های ارزیابی و دانش دامنه استفاده کرد و k را به شکلی انتخاب کرد که تعادل مناسبی بین دقت مدل و تعمیم‌پذیری آن برقرار شود.

سوال ۵

Bias-Variance Tradeoffبه تعادلی میان **Bias**(خطای سیستماتیک) و **Variance**(حساسیت مدل به تغییرات داده‌ها) اشاره دارد. این مفهوم نشان می‌دهد که برای داشتن یک مدل با عملکرد بهینه، باید تعادلی بین ساده بودن و پیچیده بودن مدل برقرار کنیم. اگر این تعادل برقرار نشود، مدل دچار **Underfitting**یا **Overfitting**می‌شود.

(Bias)خطای سیستماتیک:

- Biasزمانی ایجاد می‌شود که مدل ساده باشد و نتواند الگوهای پیچیده داده را یاد بگیرد.
- مدل‌هایی با Bias بالا معمولاً دچار **Underfitting**هستند.
- **مثال :**یک مدل رگرسیون خطی برای داده‌های غیرخطی.

(Variance)واریانس:

- Varianceزمانی رخ می‌دهد که مدل بیش از حد پیچیده باشد و به جزئیات یا نویز داده‌های آموزشی وابسته شود.
- مدل‌هایی با Variance بالا معمولاً دچار **Overfitting**هستند.

- **مثال :**یک مدل رگرسیون چندجمله‌ای با درجه بسیار بالا برای داده‌های ساده.

:Tradeoff

- **Biasبالا :**منجر به Underfitting می‌شود و مدل روی داده‌های آموزشی و داده‌های جدید عملکرد ضعیفی دارد.
- **Varianceبالا :**منجر به Overfitting می‌شود و مدل روی داده‌های آموزشی عملکرد خوب اما روی داده‌های جدید عملکرد ضعیفی دارد.
- هدف، یافتن مدلی است که هم Bias و هم Variance را در سطحی متعادل نگه دارد.

یک نمونه از مقایسه با کمک MAE وMSE

برای بررسی Bias وVariance ، از دو معیار مهم خطا استفاده می‌کنیم:

۱. **:(MAE (Mean Absolute Error**میانگین قدرمطلق اختلافات بین مقادیر واقعی و پیش‌بینی شده.

۲. **:(MSE (Mean Squared Error**میانگین مربع اختلافات بین مقادیر واقعی و پیش‌بینی شده.

فرضیات:

- سه مدل مختلف داریم:
  - **مدل ساده(Bias بالا) :**این مدل داده‌ها را ساده‌سازی می‌کند.
  - **مدل پیچیده(Variance بالا) :**این مدل به نویز داده‌ها وابسته است.
  - **مدل متعادل :**مدلی که تعادل بین Bias و Variance برقرار می‌کند.

نمونه	مقدار واقعی (y_actual)	پیش‌بینی مدل ساده (y_simple)	پیش‌بینی مدل پیچیده (y_complex)	پیش‌بینی مدل متعادل (y_balanced)
1	10	8	10.5	9.8
2	20	15	19.8	20.1
3	30	25	29.5	30.2
4	40	35	38.7	40.0

محاسبه MAE وMSE

۱. **فرمولMAE :**

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_{actual} - y_{predicted}|$$

۲. فرمول**MSE** :

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_{actual} - y_{predicted})^2$$

محاسبات برای هر مدل:

۱. مدل ساده(Bias بالا):

$$MAE = \frac{|10-8| + |20-15| + |30-25| + |40-35|}{4} = \frac{2+5+5+5}{4} = 4.25$$

$$MSE = \frac{(10-8)^2 + (20-15)^2 + (30-25)^2 + (40-35)^2}{4} = \frac{4+25+25+25}{4} = 19.75$$

۲. مدل پیچیده(Variance بالا):

$$MAE = \frac{|10-10.5| + |20-19.8| + |30-29.5| + |40-38.7|}{4} = \frac{0.5+0.2+0.5+1.3}{4} = 0.625$$

$$MSE = \frac{(10-10.5)^2 + (20-19.8)^2 + (30-29.5)^2 + (40-38.7)^2}{4} = \frac{0.25+0.04+0.25+1.69}{4} = 0.5575$$

۳. مدل متعادل:

$$MAE = \frac{|10-9.8| + |20-20.1| + |30-30.2| + |40-40|}{4} = \frac{0.2+0.1+0.2+0.0}{4} = 0.125$$

$$MSE = \frac{(10-9.8)^2 + (20-20.1)^2 + (30-30.2)^2 + (40-40)^2}{4} = \frac{0.04+0.01+0.04+0.0}{4} = 0.0225$$

نتایج نهایی:

مدل	MAE	MSE
مدل ساده	4.25	19.75
مدل پیچیده	0.625	0.5575
مدل متعادل	0.125	0.0225

تحلیل نتایج:

۱. مدل ساده(Bias بالا):

- MAEو MSE بالا هستند.
- این مدل به دلیل سادگی بیش از حد، Biasبالایی دارد و دچار **Underfitting**است.

۲. مدل پیچیده(Variance بالا):

- MAEو MSE نسبت به مدل ساده پایین‌تر هستند.
- اما این مدل ممکن است به دلیل پیچیدگی بالا، روی داده‌های آموزشی خوب عمل کند اما روی داده‌های جدید ضعیف باشد.(**Overfitting**)

۳. مدل متعادل:

- MAEو MSE کمترین مقدار را دارند.
- این مدل بهترین تعادل را بین Bias و Variance برقرار کرده و تعمیم‌پذیری بهتری دارد.

محدودیت‌های MAE وMSE

1. MAE (Mean Absolute Error):

- **مزیت :**نسبت به مقادیر پرت (Outliers) مقاوم‌تر است.
- **محدودیت :**تغییرات بزرگ خطا را کمتر برجسته می‌کند و نمی‌تواند به خوبی اختلافات کوچک و بزرگ را نشان دهد.

2. MSE (Mean Squared Error):

- **مزیت :**مقادیر بزرگ‌تر خطا را برجسته می‌کند و برای مدل‌هایی که نیاز به توجه به خطاهای بزرگ دارند مناسب است.

- **محدودیت :**بسیار حساس به داده‌های پرت است و ممکن است مدل بیش از حد روی این مقادیر متمرکز شود.

نتیجه:

- Bias-Variance Tradeoffنشان می‌دهد که تعادل بین سادگی و پیچیدگی مدل بسیار حیاتی است.
- برای ارزیابی عملکرد مدل‌ها، استفاده از معیارهایی مانند MAE و MSE ضروری است، اما باید با توجه به کاربرد، محدودیت‌های آن‌ها نیز در نظر گرفته شود.
- مدل متعادل، بهترین تعمیم‌پذیری را داشته و عملکرد مناسبی در دنیای واقعی ارائه می‌دهد.

سوال ٦

افزایش بیش از حد پیچیدگی مدل ماشین لرنینگ برای افزایش دقت لزوماً نتیجه مطلوبی ندارد و حتی می‌تواند تأثیر منفی داشته باشد. دلیل این امر به **Bias-Variance Tradeoff**و مسائلی مانند **Overfitting**و **Underfitting**بازمی‌گردد:

١. **Underfitting:**

- زمانی رخ می‌دهد که مدل بیش از حد ساده باشد و نتواند الگوهای موجود در داده را به درستی یاد بگیرد.
- در این حالت، مدل دقت کمی دارد و هم روی داده‌های آموزشی و هم روی داده‌های تست عملکرد ضعیفی ارائه می‌دهد.

٢. **Overfitting:**

- زمانی رخ می‌دهد که مدل بیش از حد پیچیده باشد و علاوه بر یادگیری الگوهای کلی، نویز و جزئیات خاص داده‌های آموزشی را نیز یاد می‌گیرد.
- در این حالت، مدل روی داده‌های آموزشی عملکرد بسیار خوبی دارد، اما روی داده‌های جدید (تست) عملکرد ضعیفی خواهد داشت.

بنابراین، افزایش پیچیدگی مدل بدون محدودیت به دلیل خطر Overfitting نمی‌تواند راه‌حل مناسبی برای افزایش دقت باشد. در عوض، باید به دنبال مدلی متعادل با پیچیدگی مناسب بود.

عوامل مهم برای دستیابی به حداکثر دقت در مدل ماشین لرنینگ

برای دستیابی به دقت حداکثری، باید به عوامل مختلفی توجه کرد که تعادل بین پیچیدگی مدل، کیفیت داده‌ها، و روش‌های بهینه‌سازی را تضمین کند.

١. کیفیت داده‌ها

- **تمیز بودن داده‌ها :**داده‌های ورودی باید از نویز، داده‌های پرت و داده‌های ناقص پاک‌سازی شوند.
- **نمایش مناسب داده‌ها :**اگر داده‌ها به درستی نمایش داده نشده باشند (مانند مقیاس‌بندی نامناسب)، حتی مدل‌های پیچیده نیز ممکن است عملکرد ضعیفی داشته باشند.
- **ویژگی‌های مهم(Feature Selection) :** انتخاب ویژگی‌های مرتبط و حذف ویژگی‌های غیرضروری یا کم‌اهمیت می‌تواند عملکرد مدل را بهبود بخشد.

۲. اندازه دیتاست

- **داده‌های کافی :**مدل‌های پیچیده مانند شبکه‌های عصبی برای جلوگیری از Overfitting به حجم بالایی از داده‌های آموزشی نیاز دارند. اگر داده‌ها کافی نباشند، مدل پیچیده به نویز داده‌ها وابسته می‌شود.
- **توزیع مناسب :**داده‌ها باید نماینده خوبی از کل فضای مسئله باشند تا مدل بتواند به درستی تعمیم‌پذیری پیدا کند.

۳. انتخاب مدل مناسب

- مدل‌های ساده‌تر مانند **رگرسیون خطی** برای مسائل خطی مناسب هستند، در حالی که مسائل پیچیده‌تر به مدل‌هایی مانند **شبکه‌های عصبی عمیق** یا **جنگل تصادفی (Random Forest)** نیاز دارند.
- پیچیدگی مدل باید با پیچیدگی مسئله همخوانی داشته باشد.

٤. تنظیم‌سازی(Regularization)

- Regularizationتکنیکی است که برای کنترل پیچیدگی مدل استفاده می‌شود. این روش با اضافه کردن جریمه‌ای به تابع هزینه، از Overfitting جلوگیری می‌کند.
  - **L1 Regularization:**برخی وزن‌ها را به صفر می‌رساند و برای انتخاب ویژگی مناسب است.
  - **L2 Regularization:**وزن‌ها را کوچک می‌کند اما به صفر نمی‌رساند.

٥. (Cross-Validation)اعتبارسنجی متقابل

- اعتبارسنجی متقابل تکنیکی است که برای ارزیابی عملکرد مدل استفاده می‌شود. این روش مدل را روی داده‌های مختلف آزمایش می‌کند تا از تعمیم‌پذیری آن اطمینان حاصل شود.
- **K-Fold Cross-Validation:**مجموعه داده به Kبخش تقسیم شده و مدل روی هر بخش به طور جداگانه تست می‌شود.

٦. توازن Bias وVariance

- مدل نباید بیش از حد ساده (Bias بالا)یا بیش از حد پیچیده (Variance بالا)باشد. باید مدلی انتخاب شود که تعادل مناسب بین این دو برقرار کند.

۷.بهینه‌سازی هایپرپارامترها

- بسیاری از مدل‌ها دارای پارامترهایی هستند که باید به صورت دستی یا خودکار تنظیم شوند. این پارامترها بر پیچیدگی مدل تأثیر می‌گذارند.
    - **مثال :**تعداد لایه‌ها و نورون‌ها در شبکه عصبی، تعداد درخت‌ها در جنگل تصادفی.

۸. افزایش داده

- در مسائل یادگیری عمیق، تکنیک‌های افزایش داده مانند چرخاندن تصاویر، تغییر مقیاس، و افزودن نویز می‌توانند به بهبود عملکرد مدل کمک کنند.

۹. یادگیری تدریجی

- استفاده از مدل‌های از قبل آموزش‌دیده روی مسائل مشابه(مانند ResNet یا BERT)می‌تواند به کاهش نیاز به داده‌های زیاد کمک کند و دقت را افزایش دهد.

۱۰.معیارهای ارزیابی مناسب

- انتخاب معیار مناسب برای ارزیابی مدل مهم است. برای مسائل مختلف ممکن است معیارهای متفاوتی مانند **MAE، MSE، Accuracy، Precision، Recall، F1-Score**استفاده شود.
    - معیارهای ارزیابی باید با هدف مسئله همخوانی داشته باشند.

دلیل عدم افزایش بی‌نهایت پیچیدگی

۱. **Overfitting:** با افزایش بیش از حد پیچیدگی، مدل به نویز داده‌های آموزشی وابسته می‌شود و روی داده‌های تست یا داده‌های جدید عملکرد ضعیفی خواهد داشت.

۲. **نیاز به داده‌های زیاد :**مدل‌های پیچیده به داده‌های بسیار زیاد نیاز دارند تا از Overfitting جلوگیری کنند. در دیتاست‌های کوچک، افزایش پیچیدگی مشکل‌ساز است.

۳. **افزایش هزینه محاسباتی :**پیچیدگی بیشتر باعث افزایش زمان و منابع محاسباتی مورد نیاز برای آموزش مدل می‌شود.

نتیجه

- افزایش پیچیدگی مدل به تنهایی نمی‌تواند دقت را به طور پایدار افزایش دهد. در عوض، باید به تعادل بین پیچیدگی مدل و کیفیت داده‌ها توجه کرد.
- برای دستیابی به حداکثر دقت، عوامل متعددی مانند کیفیت داده‌ها، انتخاب مدل مناسب، استفاده ازRegularization ، و ارزیابی مدل با Cross-Validation باید در نظر گرفته شوند.
- بهترین مدل، مدلی است که تعادل مناسبی بین Bias و Variance برقرار کند و بتواند به خوبی روی داده‌های جدید تعمیم دهد.

سوال ۷

Precisionو **Recall**دو معیار کلیدی برای ارزیابی عملکرد مدل‌های یادگیری ماشین هستند، به‌ویژه در مسائلی که داده‌ها دارای کلاس‌های نامتعادل هستند (مانند تشخیص تقلب یا بیماری) **Tradeoff** .میان Precision و Recall زمانی رخ می‌دهد که بهبود یکی از این معیارها باعث کاهش دیگری شود. در ادامه، این مفاهیم با یک مثال توضیح داده می‌شوند.

تعریف Precision وRecall

(Precision)دقت:

- Precisionبه ما می‌گوید که از میان تمام نمونه‌هایی که به عنوان "مثبت" پیش‌بینی شده‌اند، چند درصد واقعاً مثبت هستند.
- فرمول :

$$Precision = \frac{TP}{TP + FP}$$

  - TP:تعداد پیش‌بینی‌های مثبت درست.(True Positives)
  - FP:تعداد پیش‌بینی‌های مثبت اشتباه.(False Positives)

(Recall)بازخوانی:

- Recallبه ما می‌گوید که از میان تمام نمونه‌های واقعی مثبت، چند درصد توسط مدل به درستی شناسایی شده‌اند.
- فرمول :

$$Recall = \frac{TP}{TP + FN}$$

  - TP:تعداد پیش‌بینی‌های مثبت درست.(True Positives)
  - FN:تعداد نمونه‌های مثبت واقعی که مدل نتوانسته آن‌ها را شناسایی کند.(False Negatives)

در بسیاری از مسائل، نمی‌توان به طور همزمان هم Precision بالا و هم Recall بالا داشت. افزایش Precision ممکن است باعث کاهش Recallشود و بالعکس. این موضوع معمولاً به دلیل تنظیم **Threshold(آستانه تصمیم‌گیری)** در مدل است:

- با کاهش Threshold :مدل تعداد بیشتری از نمونه‌های مثبت واقعی (TP) را شناسایی می‌کند(Recall بالا)، اما ممکن است تعداد نمونه‌های مثبت اشتباه (FP) نیز افزایش یابد(Precision پایین‌تر).
- با افزایش Threshold :مدل دقیق‌تر عمل می‌کند و Precision افزایش می‌یابد، اما ممکن است برخی از نمونه‌های مثبت واقعی را از دست بدهد(Recall پایین‌تر).

مثال برای Precision-Recall Tradeoff

فرض کنید در یک سیستم تشخیص اسپم (Spam Detection) ایمیل‌ها به دو دسته تقسیم می‌شوند:

- **ایمیل اسپم (مثبت)**
- **ایمیل غیر اسپم (منفی)**

مدلی طراحی کرده‌ایم که ۱۰۰ ایمیل را طبقه‌بندی کرده و نتایج زیر به دست آمده است:

ایمیل	اسپم واقعی(Spam)	غیر اسپم واقعی(Not Spam)
پیش‌بینی اسپم	40	10
پیش‌بینی غیر اسپم	10	40

Confusion Matrix:

$$\begin{bmatrix} TP = 40 & FP = 10 \\ FN = 10 & TN = 40 \end{bmatrix}$$

محاسبه Precision و:Recall

۱. **(Precision)دقت:**

$$Precision = \frac{TP}{TP + FP} = \frac{40}{40 + 10} = 0.8 = 80\%$$

۲. **(Recall)بازخوانی:**

$$Recall = \frac{TP}{TP + FN} = \frac{40}{40 + 10} = 0.8 = 80\%$$

تغییر Threshold و تاثیر آن بر Precision وRecall

- اگر **Threshold**را کاهش دهیم:
  - ایمیل‌های بیشتری به عنوان اسپم طبقه‌بندی می‌شوند.
  - Recallافزایش می‌یابد (زیرا FN کاهش می‌یابد)، اما ممکن است Precision کاهش یابد (زیرا FP افزایش می‌یابد)
- اگر **Threshold**را افزایش دهیم:
  - ایمیل‌های کمتری به عنوان اسپم طبقه‌بندی می‌شوند.
  - Precisionافزایش می‌یابد (زیرا FP کاهش می‌یابد)، اما ممکن است Recall کاهش یابد (زیرا FN افزایش می‌یابد)

نتیجه

Precision/Recall Tradeoff به ما کمک می‌کند تا بسته به نیاز مسئله، تعادلی میان Precision و Recall ایجاد کنیم:

- اگر خطای False Positive (FP) هزینه بیشتری داشته باشد (مانند طبقه‌بندی اشتباه یک ایمیل مهم به عنوان اسپم)، Precision مهم‌تر است.
- اگر خطای False Negative (FN) هزینه بیشتری داشته باشد (مانند از دست دادن یک ایمیل اسپم واقعی)، Recallمهم‌تر است.

سوال ۸

Confusion Matrix:

با استفاده از اطلاعات مسأله، می‌توان عناصر Confusion Matrix را به صورت زیر تعریف کرد:

	مدل مورد علاقه (مثبت واقعی)	مدل غیرمورد علاقه (منفی واقعی)
پیشنهادی فروشنده (مثبت پیش‌بینی‌شده)	TP=4	FP=96
پیشنهادی نشده (منفی پیش‌بینی‌شده)	FN=1	TN=0

محاسبه Precision وRecall :

(Precision)دقت:

Precisionنشان می‌دهد از میان مدل‌هایی که فروشنده پیشنهاد داده (مثبت پیش‌بینی‌شده)، چند درصد واقعاً موردعلاقه خریدار بوده‌اند.

$$\frac{TP}{TP + FP} = Precision$$

$$4\% = 0.04 = \frac{4}{100} = \frac{4}{96 + 4} = Precision$$

(Recall)بازخوانی:

Recallنشان می‌دهد از میان مدل‌های واقعی موردعلاقه خریدار (مثبت واقعی)، چند درصد توسط فروشنده پیشنهاد داده شده‌اند.

$$\frac{TP}{TP + FN} = Recall$$

$$80\% = 0.8 = \frac{4}{5} = \frac{4}{1 + 4} = Recall$$

Confusion Matrixتکمیل‌شده:

$$\begin{bmatrix} FP = 96 & TP = 4 \\ TN = 0 & FN = 1 \end{bmatrix}$$

تحلیل نتایج:

۱. **Precision (4%)**

- تنها ۴ درصد از مدل‌هایی که فروشنده پیشنهاد داده است واقعاً موردعلاقه خریدار بوده‌اند. این مقدار نشان‌دهنده دقت پایین فروشنده در شناسایی مدل‌های موردعلاقه خریدار است.

۲. **Recall (80%)**

- 80درصد از مدل‌های موردعلاقه خریدار توسط فروشنده پیشنهاد داده شده‌اند. این مقدار نشان می‌دهد فروشنده توانسته بخش زیادی از مدل‌های مثبت واقعی را شناسایی کند.

تفسیرPrecision-Recall Tradeoff

در این مثال، فروشنده توانسته Recall بالایی (۸۰٪) داشته باشد، اما Precision بسیار پایینی (۴٪) دارد. این نشان می‌دهد فروشنده سعی کرده است همه مدل‌های ممکن را پیشنهاد دهد تا مطمئن شود مدل‌های موردعلاقه خریدار نیز در میان آن‌ها باشد، اما بسیاری از پیشنهادات او اشتباه بوده است.

برای بهبود Precision و کاهش تعداد پیشنهادات اشتباه(False Positives) ، فروشنده باید تمرکز بیشتری بر کیفیت پیشنهادات خود بگذارد و تعداد مدل‌های پیشنهادی را محدودتر کند. این کار ممکن است Recall را کمی کاهش دهد، اما Precision را افزایش می‌دهد.

نتیجه:

- Precisionپایین و Recall بالا نشان‌دهنده یک Tradeoff (تعادل) است که فروشنده باید بسته به نیاز و هدف خود آن را تنظیم کند.
- اگر هدف ارائه مدل‌های دقیق‌تر باشد، باید روی افزایش Precision کار کرد. اگر هدف پوشش تمام مدل‌های ممکن باشد، باید روی Recall تمرکز کرد.

سوال ۹

ما با یک مسئله تشخیص کلاهبرداری (Fraud Detection) در سیستم‌های مالی مواجه هستیم. نسبت داده‌های مربوط به کلاهبرداری به داده‌های انتقال سالم **1به ۱۰۰۰۰** است، یعنی این مسئله یک **دیتاست بسیار نامتعادل (Imbalanced Dataset)** دارد.

در چنین شرایطی، اگر مدل به سادگی همه نمونه‌ها را به عنوان انتقال سالم (Class 0) پیش‌بینی کند، مدل می‌تواند دقت بسیار بالایی Accuracy)نزدیک به ۱۰۰ (%داشته باشد، اما این مدل بی‌ارزش خواهد بود زیرا توانایی شناسایی کلاهبرداری (Class 1) را ندارد. بنابراین، باید روش‌هایی برای مقابله با این نامتعادلی و ارزیابی دقیق‌تر مدل اتخاذ کنیم.

۱. آموزش مدل:

الف. آماده‌سازی داده‌ها

۱. **پاک‌سازی داده‌ها:(Data Cleaning)**
   - حذف مقادیر پرت (Outliers) و نویزهای غیرضروری.
   - تکمیل داده‌های گمشده.(Missing Data)

۲. **مقیاس‌بندی داده‌ها(Feature Scaling) :**
   - استفاده از روش‌هایی مانند **StandardScaler**یا **MinMaxScaler**برای یکنواخت کردن مقیاس ویژگی‌ها.

۳. **مهندسی ویژگی‌ها(Feature Engineering) :**

- ساخت ویژگی‌های جدید مرتبط با کلاهبرداری، مانند تعداد تراکنش‌های مشکوک در یک بازه زمانی، محل تراکنش، یا مقدار تراکنش.
- حذف ویژگی‌های غیرضروری که تأثیر منفی بر مدل دارند.

ب. مقابله با عدم‌تعادل داده‌ها

برای مقابله با نامتعادل بودن داده‌ها می‌توان از روش‌های زیر استفاده کرد:

۱. **(Oversampling) افزایش داده‌های کلاس اقلیت:**

- استفاده از تکنیک‌هایی مانند **SMOTE (Synthetic Minority Over-sampling Technique)** برای ایجاد نمونه‌های مصنوعی از داده‌های کلاس کلاهبرداری.

۲. **(Undersampling)کاهش داده‌های کلاس اکثریت:**

- حذف برخی از داده‌های کلاس اکثریت (انتقالات سالم) برای متعادل‌سازی نسبت کلاس‌ها.

۳. **استفاده از وزن‌دهی کلاس‌ها:(Class Weighting)**

- در برخی الگوریتم‌ها (مانند Logistic Regression یا Random Forest)، می‌توان وزن بیشتری به کلاس اقلیت (کلاهبرداری) نسبت داد تا مدل به آن توجه بیشتری داشته باشد.

۴. **ایجاد یک دیتاست ترکیبی:**

- ترکیب داده‌های oversampled و undersampled برای دستیابی به یک مجموعه داده متعادل.

ج. انتخاب مدل مناسب

- استفاده از مدل‌هایی که در مواجهه با دیتاست‌های نامتعادل عملکرد خوبی دارند، مانند:
  - **Random Forest**
  - **Gradient Boosting مانند XGBoost یا LightGBM**
  - **Neural Networks شبکه‌های عصبی** با تنظیم مناسب.
- این مدل‌ها معمولاً توانایی بیشتری در تشخیص کلاهبرداری دارند.

د:Cross-Validation .

- برای جلوگیری از Overfitting و ارزیابی تعمیم‌پذیری مدل، از روش‌هایی مانند **K-Fold Cross-Validation** استفاده کنید.

۲. ارزیابی مدل:

در مسئله‌ای با دیتاست نامتعادل، **Accuracy**معیار مناسبی برای ارزیابی مدل نیست، زیرا ممکن است مدل با پیش‌بینی همه نمونه‌ها به عنوان انتقال سالم (Class 0) دقت بالایی کسب کند. در عوض، معیارهای زیر برای ارزیابی مناسب‌تر هستند:

الف Precision .وRecall

۱. **(Precision)دقت:**

- نشان می‌دهد از میان تراکنش‌هایی که مدل به عنوان کلاهبرداری شناسایی کرده، چه درصدی واقعاً کلاهبرداری هستند

$$\frac{TP}{TP + FP} = Precision$$

۲. **(Recall)بازخوانی:**

- نشان می‌دهد از میان تمام تراکنش‌های واقعی کلاهبرداری، چه درصدی توسط مدل شناسایی شده‌اند.

$$\frac{TP}{TP + FN} = Recall$$

۳. **F1-Score:**

- میانگینی از Precision و Recall است که در مسائلی با دیتاست نامتعادل کاربرد بیشتری دارد.

$$\frac{Precision \cdot Recall}{Precision + Recall} \cdot 2 = F1$$

ب:Confusion Matrix .

- برای بررسی دقیق عملکرد مدل از Confusion Matrix استفاده کنید که شامل مقادیر زیر است:
  - **True Positives (TP)**: تعداد کلاهبرداری‌های درست شناسایی‌شده.
  - **False Positives (FP)**: تعداد تراکنش‌های سالمی که به اشتباه به عنوان کلاهبرداری شناسایی شده‌اند.
  - **True Negatives (TN)**: تعداد تراکنش‌های سالمی که درست شناسایی شده‌اند.
  - **False Negatives (FN)**: تعداد کلاهبرداری‌هایی که مدل شناسایی نکرده است.

ج AUC-ROC وPR Curve

- **AUC-ROC (Area Under the Receiver Operating Characteristic Curve):**
  - معیاری است که توانایی مدل در جداسازی کلاس‌ها را نشان می‌دهد.

- مقدار نزدیک به ۱ نشان‌دهنده عملکرد خوب مدل است.

- **Precision-Recall Curve (PR Curve):**

- برای مسائل نامتعادل مناسب‌تر است و نشان می‌دهد Precision و Recall چگونه با تغییر Threshold تغییر می‌کنند.

.3راهکار برای تنظیم و بهبود مدل:

۱. **تنظیمThreshold :**

- مقدار پیش‌فرض Threshold (مانند ۰.۵) ممکن است مناسب نباشد Threshold .را طوری تنظیم کنید که Recall یا Precision بسته به نیاز مسئله بهینه شود.

۲. **(Regularization)تنظیم‌سازی:**

- استفاده از تکنیک‌های تنظیم‌سازی مانند **L1**یا **L2 Regularization**در مدل‌هایی مانند Logistic Regression یا شبکه‌های عصبی برای جلوگیری از.Overfitting

۳. **تفسیر مدل:**

- از ابزارهایی مانند SHAP یا LIME استفاده کنید تا بفهمید مدل چگونه تصمیم‌گیری می‌کند و آیا ویژگی‌های تأثیرگذار منطقی هستند یا خیر.

نتیجه:

- برای تشخیص کلاهبرداری در سیستم‌های مالی با دیتاست نامتعادل، نیاز است از روش‌های مقابله با نامتعادلی مانند **Oversampling**، **Class Weighting**و استفاده از معیارهای ارزیابی مناسب (مانندPrecision ، Recallو-F1 Scoreاستفاده کنیم.
- معیارهایی مانند Accuracy به دلیل نسبت نامتعادل کلاس‌ها نمی‌توانند عملکرد واقعی مدل را نشان دهند.
- مدل‌هایی مانند Random Forest و Gradient Boosting برای این نوع مسائل مناسب هستند و با استفاده از ابزارهایی مانند Cross-Validation و تنظیم Threshold می‌توان عملکرد آن‌ها را بهبود داد.

سوال ۱۰

1. Receiver Operating Characteristic (ROC):

- **منحنی ROC** نموداری است که توانایی مدل در جداسازی کلاس‌های مثبت و منفی را نشان می‌دهد. این منحنی رابطه بین نرخ **True Positive Rate (TPR)**و **False Positive Rate (FPR)**را برای مقادیر مختلف آستانه (Threshold) به تصویر می‌کشد.

محورهای منحنیROC

- **True Positive Rate (TPR):**

$$\frac{TP}{TP + FN} = TPR$$

این معیار نشان می‌دهد که از بین تمام نمونه‌های مثبت واقعی، چه تعداد به درستی شناسایی شده‌اند.

- **False Positive Rate (FPR):**

$$\frac{FP}{FP + TN} = FPR$$

این معیار نشان می‌دهد که از بین تمام نمونه‌های منفی واقعی، چه تعداد به اشتباه به عنوان مثبت شناسایی شده‌اند.

- **منحنی :ROC** با تغییر مقدارThreshold ، مقادیر TPR و FPR تغییر کرده و نقاط مختلفی در نمودار ایجاد می‌شود. این نقاط به یکدیگر متصل می‌شوند تا منحنی ROC شکل گیرد.

2. Area Under the Curve (AUC):

- **(AUC)مساحت زیر منحنی :**معیاری برای ارزیابی مدل است که نشان‌دهنده مساحت زیر منحنی ROC است.
- **تفسیرAUC**
  - مقدار AUC بین ۰ و ۱ قرار دارد.
  - **AUC = 0.5:**مدل تصادفی است و هیچ قدرتی برای تمایز بین کلاس‌ها ندارد (مانند پرتاب سکه).
  - **AUCنزدیک به ۱ :**مدل بسیار خوب عمل می‌کند و قدرت بالایی در جداسازی کلاس‌های مثبت و منفی دارد.
  - **AUC < 0.5:**مدل عملکردی بدتر از تصادفی دارد.

تفسیر AUC در مسائل طبقه‌بندی باینری

- اگر **AUC = 0.7**باشد، این بدان معناست که در ۷۰٪ موارد، مدل می‌تواند نمونه‌های مثبت را از نمونه‌های منفی بهتر از حد تصادفی تمایز دهد.
- اگر **AUC = 0.9**باشد، مدل بسیار بهتر عمل می‌کند و در ۹۰٪ موارد نمونه‌های مثبت را به درستی از نمونه‌های منفی جدا می‌کند.
- **مقایسه ۰.۷ و ۰.۹:**
  - AUC = 0.9نشان‌دهنده عملکرد بهتر مدل نسبت به AUC = 0.7 است.
  - در عمل، AUCبالاتر از ۰.۸ معمولاً به عنوان عملکرد خوب در نظر گرفته می‌شود.

Precision-Recall Curveو تفاوت آن باROC

1. Precision-Recall Curve:

- این منحنی رابطه بین **Precision**و **Recall**را برای مقادیر مختلف Threshold نشان می‌دهد.
- **دقت:**

$$\frac{TP}{TP + FP} = Precision$$

درصد نمونه‌های مثبت پیش‌بینی شده که واقعاً مثبت هستند.

- **Recall :**

$$\frac{TP}{TP + FN} = Recall$$

درصد نمونه‌های مثبت واقعی که به درستی شناسایی شده‌اند.

2. تفاوت بین Precision-Recall Curve و ROC:

ویژگی	ROC	Precision-Recall Curve
محور افقی	False Positive Rate (FPR)	Recall
محور عمودی	True Positive Rate (TPR)	Precision
کاربرد	مناسب برای دیتاست‌های متعادل	مناسب برای دیتاست‌های نامتعادل

بیشتر بر نمونه‌های مثبت (کلاس اقلیت) تمرکز دارد	به تعداد داده‌های منفی حساس است (زیرا FPR استفاده می‌شود)	**حساسیت به داده‌های منفی**

3.مثال:

- در مسائلی مانند **تشخیص کلاهبرداری** یا **تشخیص بیماری‌های نادر** که داده‌ها بسیار نامتعادل هستند (کلاس مثبت بسیار کمتر از کلاس منفی)، **Precision-Recall Curve**بهتر از ROC است. زیرا:
  - FPRدر ROC به تعداد نمونه‌های منفی وابسته است و ممکن است ارزیابی مناسبی برای چنین مسائلی ارائه ندهد.
  - Precision-Recall Curveتمرکز بیشتری بر نمونه‌های مثبت دارد و می‌تواند در این شرایط ارزیابی دقیق‌تری ارائه دهد.

نتیجه‌گیری:

- **ROC Curve**و **AUC**ابزارهایی قدرتمند برای ارزیابی مدل در مسائل طبقه‌بندی هستند، به‌ویژه زمانی که داده‌ها متعادل باشند.
- **Precision-Recall Curve**برای مسائلی که کلاس‌ها نامتعادل هستند، معیار بهتری است زیرا تمرکز بیشتری بر عملکرد مدل روی کلاس مثبت دارد.
- انتخاب مناسب منحنی و معیار ارزیابی به ویژگی‌های داده و هدف مسئله بستگی دارد.

سوال ۱۱

K-fold cross-validationیک روش محبوب برای ارزیابی عملکرد مدل‌های یادگیری ماشین است. در این روش، داده‌ها به Kبخش یا Foldتقسیم می‌شوند و مدل به طور مکرر روی بخش‌های مختلف داده آموزش و آزمایش می‌شود. هدف این روش، استفاده بهتر از داده‌ها برای ارزیابی تعمیم‌پذیری مدل و کاهش وابستگی به یک تقسیم خاص از داده‌ها است.

مراحل فرآیند:K-fold Cross-Validation

۱. **تقسیم داده‌ها به Kبخش:**
   - داده‌ها به Kبخش با اندازه تقریباً برابر تقسیم می‌شوند.
   - هر بخش یک **Fold**نامیده می‌شود.

۲. **اجرای فرآیندCross-Validation**
   - در هر مرحله، یکی از Kها Fold به عنوان مجموعه **آزمون (Test)** انتخاب می‌شود.

- بقیه K−1K-1K−1 Foldها به عنوان مجموعه **آموزش (Train)** استفاده می‌شوند.
- مدل روی داده‌های آموزشی آموزش داده می‌شود و روی داده‌های آزمون ارزیابی می‌شود.

۳. **تکرار برای تمامFold ها:**

- این فرآیند برای تمام K Foldها تکرار می‌شود، به‌طوری که هر Fold دقیقاً یک بار به عنوان مجموعه آزمون و K−1K-1K−1بار به عنوان مجموعه آموزش استفاده می‌شود.

۴. **محاسبه میانگین عملکرد:**

- پس از اجرای فرآیند برای تمامFold ها، معیارهای ارزیابی) مانندAccuracy ، Precision، Recall، یا (MSE محاسبه می‌شوند.
- میانگین این معیارها به عنوان عملکرد کلی مدل در نظر گرفته می‌شود.

مزایا و معایب K-fold Cross-Validation در مقایسه با تقسیم ساده داده‌ها

مزایا:

۱. **استفاده بهینه از داده‌ها:**

- تمام داده‌ها در فرآیند آموزش و آزمون استفاده می‌شوند، که منجر به ارزیابی دقیق‌تر می‌شود.
- در مقایسه با تقسیم ساده(Train-Test Split) ، هیچ داده‌ای کنار گذاشته نمی‌شود.

۲. **کاهش وابستگی به تقسیم تصادفی:**

- در روشTrain-Test Split ، عملکرد مدل ممکن است به نحوه تقسیم داده‌ها وابسته باشد. در مقابل، K-fold Cross-Validationاین مشکل را کاهش می‌دهد.

۳. **ارزیابی تعمیم‌پذیری:**

- این روش تخمینی بهتر از عملکرد مدل روی داده‌های جدید ارائه می‌دهد، زیرا از تمام بخش‌های داده برای آزمون استفاده می‌کند.

۴. **کاهشOverfitting :**

- به دلیل استفاده از بخش‌های مختلف داده برای آزمون، مدل از داده‌های بیش از حد محدود برای ارزیابی تأثیر نمی‌پذیرد.

معایب:

۱. **زمان و منابع محاسباتی:**

- این روش به Kبار آموزش و آزمون مدل نیاز دارد که می‌تواند برای مدل‌های پیچیده زمان‌بر و پرهزینه باشد.

۲. **پیچیدگی بیشتر:**

- نسبت به روش Train-Test Split پیچیده‌تر است و به تنظیمات دقیق‌تر نیاز دارد.

۳. **احتمال افزایش واریانس:**

- اگر داده‌ها بسیار متنوع باشند، ممکن است عملکرد مدل درFold های مختلف بسیار متفاوت باشد.

چگونه مقدار Kرا انتخاب کنیم؟

انتخاب مقدار Kیک تصمیم مهم در فرآیند Cross-Validation است و به ویژگی‌های داده‌ها و منابع محاسباتی موجود بستگی دارد.

معیارهای انتخاب :K

۱. **مقادیر رایج برای :K**

- معمولاً K=5یا K=10استفاده می‌شود.
- این مقادیر یک تعادل مناسب بین دقت و هزینه محاسباتی ارائه می‌دهند.

۲. **حجم داده‌ها:**

- اگر داده‌های کمی در اختیار داریم، می‌توان از مقادیر بزرگ‌تر برای Kاستفاده کرد .این روش برای داده‌های کوچک مناسب است اما هزینه محاسباتی بالایی دارد.
- اگر داده‌های زیادی در اختیار داریم، مقادیر کوچک‌تر کافی هستند.

۳. **هزینه محاسباتی:**

- برای مدل‌های پیچیده‌تر که آموزش آن‌ها زمان‌بر است، انتخاب مقادیر کوچک‌تر Kمی‌تواند هزینه محاسباتی را کاهش دهد.

۴. **تنوع داده‌ها:**

- اگر داده‌ها بسیار متنوع هستند یا نامتعادل هستند، انتخاب مقادیر بزرگ‌تر برای Kمی‌تواند به کاهش واریانس کمک کند.

مثال عملی برایK-fold Cross-Validation

فرض کنید یک مجموعه داده با ۱۰۰۰ نمونه دارید و K=5انتخاب کرده‌اید:

۱. داده‌ها به ۵ بخش ۲۰۰تایی تقسیم می‌شوند.

۲. در هر مرحله، ۴ بخش (۸۰۰ نمونه) برای آموزش و ۱ بخش (۲۰۰ نمونه) برای آزمون استفاده می‌شوند.

۳. این فرآیند ۵ بار تکرار می‌شود تا تمام داده‌ها به عنوان مجموعه آزمون استفاده شوند.

۴. میانگین عملکرد مدل روی این ۵ تکرار به عنوان نتیجه نهایی در نظر گرفته می‌شود.

نتیجه‌گیری

- K-fold Cross-Validationیک روش استاندارد برای ارزیابی تعمیم‌پذیری مدل است که استفاده بهینه‌تری از داده‌ها نسبت به روش Train-Test Split ارائه می‌دهد.
- انتخاب مقدار Kبه حجم داده‌ها، پیچیدگی مدل و منابع محاسباتی بستگی دارد.
- اگرچه این روش هزینه محاسباتی بیشتری دارد، اما ارزیابی دقیق‌تر و پایدارتر از عملکرد مدل فراهم می‌کند.

سوال ۱۲

قسمت اول: مقایسه سیستم پیشنهاد دهنده جدید با سیستم قدیمی

برای اینکه نشان دهیم مدل پیشنهادی ما بهتر از مدل قدیمی است، باید معیارهای مشخص و قابل اندازه‌گیری تعریف کنیم. در اینجا چند معیار کلیدی برای مقایسه سیستم جدید و قدیمی پیشنهاد می‌شود:

معیارهای ارزیابی عملکرد مدل:

۱. **دقت پیشنهادات(Precision)**

- این معیار نشان می‌دهد که از میان تمام کالاهایی که سیستم پیشنهاد داده، چه تعداد واقعاً مرتبط بوده‌اند.
- فرمول:

$$\frac{TP}{TP + FP} = Precision$$

  - TP:تعداد پیشنهادات درست (کالاهایی که مشتری واقعاً به آن‌ها علاقه داشته است).
  - FP:تعداد پیشنهادات اشتباه (کالاهایی که مشتری به آن‌ها علاقه نداشته است).

۲. **بازخوانی پیشنهادات(Recall)**

- این معیار نشان می‌دهد که از میان تمام کالاهایی که مشتری واقعاً به آن‌ها علاقه داشته، چه تعداد توسط سیستم شناسایی شده‌اند.

$$\frac{TP}{TP + FN} = Recall$$

  - FN:تعداد کالاهای مرتبطی که سیستم پیشنهاد نداده است.

۳. **F1-Score**

- میانگین هارمونیک بین Precision و Recall که برای ارزیابی تعادل بین این دو معیار مفید است.
- فرمول:

$$\frac{Precision \cdot Recall}{Precision + Recall} \cdot 2 = F1$$

۴. **نرخ کلیک(Click-Through Rate - CTR)**

- این معیار نشان می‌دهد که چه درصدی از پیشنهادات ارائه شده توسط سیستم توسط مشتریان کلیک شده‌اند.
- فرمول :

$$\frac{\text{تعداد کلیک‌ها روی پیشنهادات}}{\text{تعداد کل پیشنهادات ارائه شده}} = CTR$$

۵. **نرخ تبدیل(Conversion Rate - CR)**

- درصد کالاهایی که مشتری پس از کلیک خریداری کرده است.
- فرمول :

$$\frac{\text{تعداد خریدها}}{\text{تعداد کلیک‌ها}} = CR$$

۶. **زمان پردازش**

- مقایسه زمان مورد نیاز برای تولید پیشنهادات توسط سیستم جدید در مقابل سیستم دستی قبلی.
- کاهش زمان پردازش می‌تواند یکی از مزایای اصلی سیستم جدید باشد.

روش انجام آزمایش

۱. **تعیین گروه کنترل و آزمایش:**

- مشتریان را به دو گروه تقسیم کنید :
  - **گروه کنترل :**که همچنان از سیستم دستی قدیمی استفاده می‌کنند.
  - **گروه آزمایش :**که از سیستم پیشنهاد دهنده جدید استفاده می‌کنند.

۲. **جمع‌آوری داده‌ها:**

- برای هر گروه، داده‌های مرتبط با معیارهای فوق را جمع‌آوری کنید (مانند تعداد کلیک‌ها، خریدها، و دقت پیشنهادات).

۳. **تحلیل نتایج:**

- میانگین معیارهای محاسبه شده برای دو گروه را مقایسه کنید.
- از آزمون‌های آماری استفاده کنید تا اطمینان حاصل شود که تفاوت‌ها معنادار هستند.

قسمت دوم: محاسبه زمان بازگشت سرمایه(ROI)

Return on Investment (ROI)یک معیار کلیدی برای مدیران است که سودآوری پروژه را نسبت به هزینه‌های آن اندازه‌گیری می‌کند. برای محاسبه ROI باید هزینه‌ها و منافع حاصل از سیستم جدید مشخص شوند.

فرمول محاسبهROI

$$ROI = \frac{\text{سود خالص}}{\text{هزینه کل}} \times 100$$

مراحل تخمینROI

۱. تخمین هزینه‌ها:

هزینه‌ها شامل موارد زیر است:

- **هزینه توسعه :**هزینه برنامه‌نویسی، طراحی مدل، و جمع‌آوری داده‌ها.
- **هزینه زیرساخت :**هزینه‌های مربوط به سرورها، پایگاه داده‌ها، و نگهداری سیستم.

- **هزینه آموزش کارمندان :**برای استفاده و مدیریت سیستم جدید.
- **هزینه بهره‌برداری :**هزینه‌های جاری برای اجرای سیستم.

۲.تخمین منافع:

منافع حاصل از سیستم جدید شامل موارد زیر است:

- **افزایش درآمد :**
  - افزایش فروش به دلیل پیشنهادات بهتر (محاسبه از نرخ تبدیل و فروش حاصل از آن).
- **کاهش هزینه‌ها :**
  - کاهش هزینه‌های نیروی انسانی که قبلاً برای پیشنهادات دستی استفاده می‌شد.
  - کاهش زمان پردازش و بهبود کارایی.

۳. محاسبهROI :

- **سود خالص :**تفاوت بین منافع و هزینه‌ها.
- **ROI:** نسبت سود خالص به هزینه کل، به صورت درصد.

روش تخمین زمان بازگشت سرمایه:

- برای تخمین زمان بازگشت سرمایه، باید **Break-Even Point**را محاسبه کرد.
- **فرمول :**

$$\text{زمان بازگشت سرمایه} = \frac{\text{هزینه کل}}{\text{سود خالص ماهانه}}$$

- به عنوان مثال :
  - هزینه کل پروژه: ۱ میلیارد تومان.
  - سود خالص ماهانه 200 :میلیون تومان.
  - زمان بازگشت سرمایه : ۵ ماه.

فرضیات منطقی:

۱. فرض شده که داده‌های مربوط به فروش، کلیک‌ها و نرخ تبدیل به صورت دقیق ثبت شده‌اند.

۲. فرض شده که سیستم جدید قابلیت یکپارچگی کامل با زیرساخت‌های موجود را دارد.

۳. فرض شده که نرخ بازگشت مشتریان به دلیل پیشنهادات بهبود یافته افزایش می‌یابد.

نتیجه‌گیری:

- برای اثبات بهتر بودن سیستم جدید نسبت به سیستم قدیمی، معیارهایی مانندPrecision ، Recall، CTR و نرخ تبدیل باید محاسبه و مقایسه شوند.
- ROIیک معیار کلیدی برای نشان دادن سودآوری سیستم جدید است. محاسبه آن با تخمین هزینه‌ها و منافع سیستم انجام می‌شود.
- نشان دادن زمان بازگشت سرمایه به مدیران، تصمیم‌گیری در مورد اجرای سیستم را تسهیل می‌کند و ارزش پروژه را برای کسب‌وکار مشخص می‌سازد.