

# به نام خدا

سبحان کوشکی جهرمی 810101496

AI\_A4

سوال 1 ) اگر داده‌های آموزشی شما ناقص هستند و برخی ویژگی‌ها یا اطلاعات در آن‌ها وجود ندارد، باید به چندین جنبه مهم توجه کنید تا مطمئن شوید که مدل شما عملکرد خوبی دارد. این موارد شامل داده‌های گمشده، داده‌های اشتباه، یا عدم دسترسی به برخی ویژگی‌های مهم است. در ادامه، چندین استراتژی و پیشنهاد برای رفع این مشکلات آمده است:

1. شناسایی و درک مشکل

- بررسی کامل داده‌ها: اولین گام بررسی دقیق داده‌ها است. اطمینان حاصل کنید که می‌دانید چه ویژگی‌هایی ناقص یا گمشده هستند و چگونه این مشکل ممکن است بر مدل‌سازی شما تأثیر بگذارد.

- ارزیابی اهمیت ویژگی‌ها: تحلیل کنید که آیا ویژگی‌های گمشده برای مدل‌سازی شما حیاتی هستند یا خیر. ممکن است برخی ویژگی‌ها نسبت به بقیه اهمیت بیشتری داشته باشند.

2. استفاده از جایگزین‌ها (Imputation)

- جایگزینی با مقادیر آماری: برای ویژگی‌های عددی، می‌توانید از میانگین، میانه، یا مد برای پر کردن مقادیر گمشده استفاده کنید. این روش ساده ولی موثر است.

- جایگزینی مبتنی بر همبستگی: اگر بین ویژگی‌ها همبستگی وجود دارد، می‌توانید از این ارتباطها برای پر کردن مقادیر گمشده استفاده کنید.

- مدل‌های پیچیده‌تر: روش‌هایی مانند K-Nearest Neighbors (KNN) یا Multiple Imputation by Chained Equations (MICE) می‌توانند برای پر کردن مقادیر گمشده مفید باشند.

3. استفاده از منابع داده‌ی دیگر

- ترکیب داده‌ها: اگر به منابع داده‌ی دیگری دسترسی دارید که شامل ویژگی‌های مورد نیاز شما است، می‌توانید داده‌ها را ترکیب کنید.

- جستجو در داده‌های خارجی: برخی اوقات می‌توانید از منابع داده‌ی عمومی یا پایگاه‌های داده‌ی دیگر برای پر کردن ویژگی‌های گمشده استفاده کنید.

4. مهندسی ویژگی‌ها

- ایجاد ویژگی‌های جدید: اگر برخی ویژگی‌ها گمشده‌اند، ممکن است بتوانید ویژگی‌های جدیدی ایجاد کنید که اطلاعات مشابهی ارائه دهند.

- ترکیب ویژگی‌ها: ترکیب ویژگی‌های موجود برای ایجاد ویژگی‌های جدید می‌تواند راهی برای جبران داده‌های گمشده باشد.

5. استفاده از نشانگرهای باینری

- نشانگر داده‌های گمشده: ایجاد یک ویژگی باینری که نشان دهد آیا یک ویژگی گمشده است یا خیر، می‌تواند به مدل شما کمک کند تا تفاوت‌ها را بهتر درک کند.

6. استفاده از الگوریتم‌های مقاوم در برابر داده‌های ناقص

- الگوریتم‌های مقاوم: برخی الگوریتم‌های یادگیری ماشین نسبت به داده‌های ناقص حساسیت کمتری دارند. از این نوع الگوریتم‌ها استفاده کنید.

- استفاده از مدل‌های Ensemble: مدل‌های Ensemble معمولاً نسبت به داده‌های ناقص مقاوم‌تر هستند و نتایج بهتری در مواجهه با داده‌های ناقص ارائه می‌دهند.

7. مستندسازی مشکلات و استراتژی‌های شما

- مستندسازی برای آینده: تمام مشکلات داده‌ها و روش‌هایی که برای حل آن‌ها استفاده کرده‌اید را مستندسازی کنید تا در آینده به مشکل مشابهی برخورد نکنید.

با این راهکارها، می‌توانید داده‌های ناقص را مدیریت کنید و مدل‌های دقیق‌تری ایجاد کنید.

2) عدم تعادل در توزیع داده‌ها میان کلاس‌ها در مسائل طبقه‌بندی می‌تواند یک چالش مهم باشد و اغلب منجر به عملکرد ضعیف مدل‌ها، به‌ویژه برای کلاس‌های کمتر نمایان‌شده، می‌شود. برای رفع این مشکل، روش‌ها و استراتژی‌های مختلفی وجود دارد که به بهبود عملکرد مدل‌ها در مواجهه با داده‌های نامتعادل کمک می‌کند. در ادامه، چندین روش برای برخورد با عدم تعادل داده‌ها در کلاس‌ها آمده است:

1. باز نمونه‌گیری داده‌ها

- افزایش نمونه (Oversampling): این روش شامل افزایش تعداد نمونه‌ها در کلاس‌های کمتر نمایان‌شده با تکثیر نمونه‌های موجود یا ایجاد داده‌های جدید است. تکنیک‌هایی مانند SMOTE (Synthetic Minority Over-sampling Technique) به‌عنوان یک روش هوشمندانه برای افزایش نمونه عمل می‌کنند.

- کاهش نمونه (Undersampling): این روش شامل کاهش تعداد نمونه‌ها در کلاس‌های پرنمایان‌شده است. این کار می‌تواند با حذف تصادفی نمونه‌های اضافی یا انتخاب نمونه‌های نمایانگر انجام شود.

2. استفاده از الگوریتم‌های مقاوم در برابر داده‌های نامتعادل

- الگوریتم‌های Ensemble: تکنیک‌هایی مانند Random Forest و Boosting (مثل XGBoost) اغلب می‌توانند با داده‌های نامتعادل بهتر کار کنند، زیرا هر مدل ضعیف در این الگوریتم‌ها می‌تواند با داده‌های مختلف آموزش داده شود.

- الگوریتم‌های مبتنی بر هزینه (Cost-sensitive Learning): برخی الگوریتم‌ها می‌توانند هزینه‌های متفاوتی را برای اشتباهات طبقه‌بندی تعیین کنند، به‌طوری‌که خطا در کلاس‌های کمتر نمایان‌شده گران‌تر است.

- الگوریتم‌های یادگیری از نمونه‌برداری‌های متوازن: الگوریتم‌هایی مانند Balanced Random Forest یا Balanced Bagging Classifier می‌توانند داده‌ها را در هر مرحله از ساخت مدل متوازن کنند.

3. استفاده از معیارهای ارزیابی مناسب

- معیارهای جایگزین برای ارزیابی: در داده‌های نامتعادل، استفاده از معیارهایی مانند دقت (Accuracy) ممکن است گمراه‌کننده باشد. معیارهایی مانند F1-Score، Precision، Recall، یا AUC-ROC اغلب شاخص‌های بهتری برای ارزیابی عملکرد مدل در داده‌های نامتعادل هستند.

- ماتریس Confusion: تجزیه و تحلیل دقیق ماتریس Confusion می‌تواند به شناسایی مشکلات مدل در طبقه‌بندی کلاس‌های خاص کمک کند.

#### 4. استفاده از تکنیک‌های داده‌افزایی (Data Augmentation)

- تولید داده‌های جدید: با استفاده از روش‌های داده‌افزایی، می‌توانید داده‌های جدیدی ایجاد کنید که کلاس‌های کمتر نمایان‌شده را تقویت کنند.
- تبدیل داده‌ها: تکنیک‌هایی مانند چرخش، تغییر اندازه، یا تغییر رنگ (برای داده‌های تصویری) می‌توانند داده‌های بیشتری برای کلاس‌های کمتر نمایان‌شده ایجاد کنند.
- 5. برقراری تعادل در داده‌های آموزشی و تست
  - مطمئن شدن از تعادل در داده‌های تست: در حالی که ممکن است داده‌های آموزشی را بازنمونه‌برداری کنید، اطمینان حاصل کنید که داده‌های تست بازتابی از شرایط دنیای واقعی هستند. این کار کمک می‌کند تا از بیش‌برازش (Overfitting) جلوگیری کنید.
  - با ترکیب این روش‌ها، می‌توانید با مشکلات ناشی از عدم تعادل در توزیع داده‌ها مقابله کنید و عملکرد مدل‌های خود را در مواجهه با چنین شرایطی بهبود بخشید.

**3) وجود نویز در داده‌ها یکی از چالش‌های رایج در تجزیه و تحلیل داده‌ها و یادگیری ماشین است.** نویز می‌تواند به انواع مختلفی رخ دهد، از جمله داده‌های نادرست یا خطاهای انسانی، اختلال‌های محیطی، یا انحراف‌های تصادفی. نویز می‌تواند تأثیر منفی بر عملکرد مدل‌های یادگیری ماشین داشته باشد، از جمله کاهش دقت و قابلیت اعتماد، و افزایش خطر بیش‌برازش (Overfitting). برای مقابله با نویز، چندین استراتژی وجود دارد که می‌تواند به بهبود کیفیت داده‌ها و مدل‌ها کمک کند. در ادامه، چندین روش برای مدیریت نویز در داده‌ها ارائه شده است:

##### 1. پیش‌پردازش داده‌ها

- حذف داده‌های نویزی یا نادرست: بررسی داده‌ها برای شناسایی و حذف مواردی که به وضوح نادرست یا نویزی هستند.
- استانداردسازی و نرمال‌سازی: این کار می‌تواند به کاهش تأثیر نویز ناشی از تفاوت در مقیاس داده‌ها کمک کند.
- حذف ویژگی‌های پرت: شناسایی و حذف ویژگی‌هایی که مقادیر پرت زیادی دارند یا به نظر می‌رسد نویز در آنها زیاد است.

##### 2. استفاده از الگوریتم‌های مقاوم در برابر نویز

- الگوریتم‌های مقاوم: برخی الگوریتم‌ها مانند Random Forest یا الگوریتم‌های مبتنی بر درخت، معمولاً نسبت به نویز مقاوم‌تر هستند.
- الگوریتم‌های Ensemble: تکنیک‌هایی مانند Bagging و Boosting می‌توانند با کاهش تأثیر نویز بر مدل نهایی، کمک کنند.

##### 3. استفاده از روش‌های آماری

- فیلترهای آماری: استفاده از روش‌های آماری برای حذف داده‌های نویزی یا نادرست.
- استفاده از مدل‌های Robust: مدل‌هایی که در برابر داده‌های پرت و نویز مقاوم هستند.

##### 4. تکنیک‌های حذف نویز

- فیلترینگ: استفاده از فیلترهایی مانند فیلتر میانگین متحرک یا فیلتر گاوسی برای صاف کردن داده‌های نویزی.

- روش‌های کاهش نویز: روش‌هایی مانند PCA (Principal Component Analysis) یا SVD (Singular Value Decomposition) می‌توانند به حذف نویز از داده‌ها کمک کنند.

5. داده‌افزایی و مجموعه‌های داده مصنوعی

- افزایش داده‌ها: ایجاد داده‌های مصنوعی برای تقویت مجموعه داده و کاهش تأثیر نویز.

- تکنیک‌های داده‌افزایی: استفاده از روش‌هایی مانند افزایش داده‌های تصویری با تغییر زاویه، اندازه، یا اعمال نویز مصنوعی می‌تواند به کاهش تأثیر نویز در مدل کمک کند.

6. مدیریت بیش‌برازش (Overfitting)

- استفاده از تکنیک‌های جلوگیری از بیش‌برازش: مانند Dropout، Regularization، یا Early Stopping.

- Cross-Validation: استفاده از تکنیک‌های اعتبارسنجی متقابل برای اطمینان از اینکه مدل بر داده‌های نویزی بیش‌برازش نمی‌کند.

7. آنالیز دقیق داده‌ها

- تجزیه و تحلیل داده‌ها برای شناسایی نویز: با استفاده از روش‌های تجزیه و تحلیل داده‌ها برای شناسایی نویز و تعیین منبع آن.

- تجسم داده‌ها: استفاده از روش‌های بصری برای شناسایی نویز و پرت‌ها.

با استفاده از این استراتژی‌ها، می‌توانید تأثیر نویز در داده‌های خود را کاهش داده و مدل‌های دقیق‌تر و قابل‌اعتمادتری ایجاد کنید.

**4** وجود ویژگی‌های همبسته (Correlated Features) در داده‌ها می‌تواند مشکلاتی را در مدل‌سازی ایجاد کند. این مشکلات شامل افزایش پیچیدگی مدل، افزایش خطر بیش‌برازش (Overfitting)، و کاهش کارایی برخی الگوریتم‌هاست. برای مقابله با این مسئله، می‌توانید از روش‌ها و استراتژی‌های مختلفی استفاده کنید. در ادامه، راهکارهایی برای مدیریت ویژگی‌های همبسته آمده است:

1. شناسایی ویژگی‌های همبسته

- محاسبه ماتریس همبستگی: استفاده از ماتریس همبستگی (مانند ضریب همبستگی پیرسون) برای تعیین ارتباط بین ویژگی‌ها. این می‌تواند به شناسایی ویژگی‌های دارای همبستگی بالا کمک کند.

- تجسم داده‌ها: استفاده از روش‌هایی مانند heatmap یا scatterplot matrix برای تجسم همبستگی بین ویژگی‌ها.

2. حذف یا ترکیب ویژگی‌های همبسته

- حذف ویژگی‌های همبسته: اگر ویژگی‌هایی همبستگی بالایی دارند، می‌توانید یکی از آنها را حذف کنید تا پیچیدگی مدل کاهش یابد.

- ترکیب ویژگی‌ها: گاهی می‌توانید ویژگی‌های همبسته را ترکیب کنید تا یک ویژگی جدید بسازید که اطلاعات مشابهی را ارائه می‌دهد.

3. استفاده از روش‌های کاهش ابعاد

- PCA (Principal Component Analysis): این روش ویژگی‌های همبسته را به چند مؤلفه اصلی تبدیل می‌کند که می‌توانند داده‌ها را با کمترین از دست دادن اطلاعات فشرده کنند.

- LDA (Linear Discriminant Analysis): این روش به شما امکان می‌دهد تا ویژگی‌ها را بر اساس تمایز میان کلاس‌ها کاهش دهید.

4. استفاده از الگوریتم‌های مقاوم در برابر ویژگی‌های همبسته

- الگوریتم‌های مبتنی بر درخت: الگوریتم‌هایی مانند Random Forest و Decision Tree معمولاً در برابر ویژگی‌های همبسته مقاوم‌تر هستند، زیرا آنها به‌طور خودکار مهم‌ترین ویژگی‌ها را انتخاب می‌کنند.

- الگوریتم‌های Ensemble: این الگوریتم‌ها به دلیل ساختار چندمدله، اغلب در برابر ویژگی‌های همبسته انعطاف‌پذیرتر هستند.

5. استفاده از تکنیک‌های کاهش بیش‌برازش

- Regularization: روش‌هایی مانند L1 یا L2 Regularization می‌توانند به جلوگیری از تأثیر منفی ویژگی‌های همبسته کمک کنند.

- Dropout: این تکنیک در شبکه‌های عصبی می‌تواند به کاهش تأثیر ویژگی‌های همبسته کمک کند.

6. استفاده از ویژگی‌های مهم (Feature Importance)

- بررسی اهمیت ویژگی‌ها: با استفاده از الگوریتم‌هایی مانند Random Forest یا XGBoost، می‌توانید اهمیت ویژگی‌ها را بررسی کنید و ویژگی‌هایی را که بیشترین تأثیر را دارند، حفظ کنید.

- حذف ویژگی‌های کم‌اهمیت: با شناسایی ویژگی‌های کم‌اهمیت، می‌توانید ویژگی‌های همبسته را که اطلاعات زیادی اضافه نمی‌کنند، حذف کنید.

با استفاده از این روش‌ها، می‌توانید تأثیر ویژگی‌های همبسته را کاهش داده و مدل‌های بهتری ایجاد کنید. همچنین، این روش‌ها به بهبود تفسیر مدل‌ها و جلوگیری از بیش‌برازش کمک می‌کنند.

سوال 2) برای در نظر گرفتن تأثیر آزمون دادن بر روی نمره آزمون، مشاور تحصیلی می‌تواند از یک مدل رگرسیون خطی چندگانه استفاده کند. در مدل رگرسیون خطی چندگانه، چندین متغیر مستقل برای پیش‌بینی یک متغیر وابسته استفاده می‌شود. در این مورد، مشاور می‌تواند علاوه بر ساعات مطالعاتی، تعداد آزمون‌های تمرینی را به عنوان متغیر مستقل دیگر در نظر بگیرد.

فرض کنیم که:

(x\_1) نشان‌دهنده ساعات مطالعاتی باشد.

(x\_2) نشان‌دهنده تعداد آزمون‌های تمرینی باشد.

بنابراین، معادله رگرسیون خطی جدید می‌تواند به صورت زیر باشد:

$$[y = a + b_1 * x_1 + b_2 * x_2]$$



## 6. روش‌های تصادفی (Randomized Methods)

این روش‌ها از عنصر تصادفی برای بهبود اکتشاف استفاده می‌کنند و می‌توانند در مواردی که گزاردیان‌ها ممکن است گیر کنند، مفید باشند.

### سوال سوم

	Predicted Negative	Predicted Positive
Actual Positive	300	20
Actual Negative	30	200

$$\text{Recall} = \text{TP} / \text{TP} + \text{FN}$$

$$\text{Precision} = \text{TP} / \text{TP} + \text{FP}$$

$$\text{Accuracy} = \text{TP} + \text{TN} / \text{TP} + \text{TN} + \text{FP} + \text{FN}$$

$$\text{F1-score} = 2 * \text{precision} * \text{recall} / \text{precision} + \text{recall}$$

```
Recall: 0.9375
Precision: 0.9090909090909091
Accuracy: 0.9090909090909091
F1-score: 0.923076923076923
```

### سوال KNN :





- \*\*داده‌های با نویز زیاد\*\* : داده‌های پر از نویز یا موارد پرت (Outliers) می‌توانند SVM را دچار مشکل کنند، زیرا این داده‌ها بر مرز تصمیم‌گیری تأثیر می‌گذارند.

3. کرنل‌ها و نقش آنها در طبقه‌بندی

SVM معمولاً به عنوان یک الگوریتم خطی شناخته می‌شود، اما با استفاده از کرنل‌ها، این الگوریتم می‌تواند مرزهای تصمیم‌گیری غیرخطی ایجاد کند.

- \*\*کرنل چیست؟\*\* کرنل‌ها توابعی هستند که امکان انتقال داده‌ها را از یک فضای ویژگی به یک فضای ویژگی دیگر فراهم می‌کنند، به طوری که داده‌ها در فضای جدید ممکن است به شکل خطی جداسازی باشند. به عبارت دیگر، کرنل‌ها اجازه می‌دهند تا SVM مسائل پیچیده‌تر را حل کند.

- \*\*نقش کرنل‌ها در طبقه‌بندی\*\* : در مواردی که داده‌ها در فضای ویژگی اصلی به صورت خطی قابل جداسازی نیستند، کرنل‌ها با ایجاد نقشه‌برداری غیرخطی، مرزهای خطی را در فضای ویژگی جدید ممکن می‌کنند. با این کار، SVM می‌تواند در حل مسائل غیرخطی نیز موثر باشد.

- \*\*انواع کرنل‌ها\*\* :

- \*\*کرنل چندجمله‌ای (Polynomial Kernel)\*\* : یک تبدیل غیرخطی بر اساس چندجمله‌ای است.

- \*\*کرنل گوسی (RBF - Radial Basis Function Kernel)\*\* : یک کرنل معروف که بر اساس فاصله اقلیدسی بین نمونه‌ها عمل می‌کند.

- \*\*کرنل سیگموئید (Sigmoid Kernel)\*\* : بر اساس تابع سیگموئید است و در شبکه‌های عصبی نیز به کار می‌رود.

4. تفاوت بین Soft SVM و Hard SVM

- \*\*Hard SVM Classifier\*\* : این نوع SVM فرض می‌کند که داده‌ها خطی جداسازی هستند و مرز تصمیم‌گیری دقیقاً باید بدون هیچگونه خطا یا داده‌های اشتباه ایجاد شود. این روش تلاش می‌کند یک حاشیه بزرگ بین کلاس‌ها ایجاد کند، اما اگر داده‌ها دارای نویز یا موارد پرت باشند، ممکن است با مشکلاتی مواجه شود.

- \*\*Soft SVM Classifier\*\* : این نوع SVM اجازه می‌دهد برخی نمونه‌ها در ناحیه حاشیه یا حتی در سمت دیگر مرز قرار بگیرند. با استفاده از یک پارامتر تنبیه (Regularization Parameter) می‌توان میزان مجاز خطا را تنظیم کرد. این انعطاف‌پذیری باعث می‌شود که Soft SVM برای داده‌هایی با نویز یا داده‌های غیرخطی مناسب‌تر باشد.

تفاوت کلیدی بین این دو رویکرد در میزان مجاز خطا و انعطاف‌پذیری آنهاست. Soft SVM بیشتر برای داده‌های دنیای واقعی مناسب است، در حالی که Hard SVM برای داده‌های بسیار تمیز و کاملاً خطی جداسازی کاربرد دارد.

ماشین‌های بردار پشتیبانی (SVM) بیشتر به عنوان یک الگوریتم طبقه‌بندی شناخته می‌شوند، اما می‌توانند برای مسائل رگرسیون نیز استفاده شوند. در این حالت، از آنها به عنوان ماشین‌های بردار پشتیبانی برای رگرسیون (Support Vector Regression - SVR) یاد می‌شود. در اینجا به نحوه استفاده از SVM در مسائل رگرسیون و تفاوت آن با کاربردهای طبقه‌بندی می‌پردازیم.

1. ایده کلی در SVR

در رگرسیون، هدف پیش‌بینی یک متغیر پیوسته است، برخلاف طبقه‌بندی که هدف تعیین کلاس یا دسته است. در SVR، هدف ایجاد یک تابع رگرسیون است که خطای پیش‌بینی را به حداقل برساند، در حالی که حاشیه‌ای ایجاد می‌کند که در آن خطاها به اندازه خاصی قابل قبول هستند.

2. تفاوت بین SVR و SVM در طبقه‌بندی

در SVM برای طبقه‌بندی، مرز تصمیم‌گیری (Decision Boundary) به گونه‌ای ایجاد می‌شود که دو کلاس به طور حداکثر از هم جدا شوند. اما در SVR، هدف این است که یک خط یا یک سطح رگرسیون ایجاد شود که از یک حاشیه خاص فراتر نرود. این حاشیه به عنوان "خطای مجاز" در نظر گرفته می‌شود.

### 3 پارامترهای اصلی در SVR

- **\*\* $\epsilon$ \*\***: این پارامتر تعیین می‌کند که چه میزان خطا در پیش‌بینی مجاز است. نمونه‌هایی که خطای آنها در این محدوده است، به عنوان بردارهای پشتیبانی در نظر گرفته نمی‌شوند.

- **\*\* $C$ \*\*** (Regularization Parameter): این پارامتر میزان مجاز بودن خطاهای خارج از حاشیه را کنترل می‌کند. پارامتر بزرگتر  $C$  به معنای تحمل کمتر خطا و تمایل بیشتر به مدل‌های پیچیده‌تر است.

### 4. استفاده از کرنل‌ها در SVR

مشابه SVM برای طبقه‌بندی، SVR نیز می‌تواند از کرنل‌ها استفاده کند. کرنل‌ها امکان ایجاد رگرسیون‌های غیرخطی را فراهم می‌کنند. کرنل‌های رایج در SVR عبارت‌اند از:

- **\*\*کرنل گوسی (RBF - Radial Basis Function Kernel)\*\***: کرنلی است که بیشترین استفاده را دارد و برای مسائل پیچیده و غیرخطی مفید است.

- **\*\*کرنل چندجمله‌ای (Polynomial Kernel)\*\***: برای مسائل رگرسیون با الگوهای چندجمله‌ای مناسب است.

### 5. مراحل اجرای SVR

مراحل اصلی استفاده از SVR برای مسائل رگرسیون به شرح زیر است:

1. **\*\*انتخاب کرنل\*\***: بسته به ماهیت داده‌ها، کرنل مناسب را انتخاب کنید. کرنل گوسی اغلب انتخاب پیش‌فرض است.
2. **\*\*تنظیم پارامترها\*\***: پارامترهای  $\epsilon$  و  $C$  را تنظیم کنید. مقدار  $\epsilon$  خطای مجاز و  $C$  میزان منظم‌سازی را تعیین می‌کند.
3. **\*\*آموزش مدل\*\***: داده‌های آموزشی را به مدل SVR بدهید تا مدل رگرسیون ایجاد شود.
4. **\*\*پیش‌بینی\*\***: از مدل SVR برای پیش‌بینی مقادیر متغیر پیوسته برای داده‌های جدید استفاده کنید.