



دانشگاه علم و صنعت ایران

تمرین اول

نام درس: داده کاوی پیشرفته

استاد درس: دکتر بهروز مینایی

نام: محمد حقیقت

شماره دانشجویی: 403722042

گرایش: هوش مصنوعی

دانشکده: مهندسی کامپیوتر

نیم سال دوم 1403-1404

سوال اول

Classification - 1

هدف (سوال تجاری): پیش‌بینی این که آیا یک مشتری در پرداخت وام خود دچار مشکل خواهد شد؟

ورودی: داده‌های مشتری مثل سن، درآمد، نوع شغل، تاریخچه حساب، بدهی‌های معوقه، رفتار بازپرداخت وام‌های قبلی.

خروجی: یک برچسب با عنوانی:

- ریسک بالا
- ریسک متوسط
- ریسک پایین

کاربرد: مدیران می‌توانند از این مدل برای تصمیم‌گیری در مورد تایید درخواست وام یا تنظیم شرایط وام بر اساس سطوح ریسک استفاده کنند.

Regression - 2

هدف (سوال تجاری): بانک در سال آینده از وام‌های یک مشتری چقدر درآمد بهره خواهد داشت؟
ورودی:

- مقدار وام
- نرخ بهره
- مدت زمان وام
- رفتار بازپرداخت گذشته مشتری
- شاخص‌های اقتصادی (تورم، نرخ بیکاری)

خروجی: پیش‌بینی عددی از درآمد بهره مورد انتظار از وام‌های مشتری.

کاربرد:
از مدیریت ریسک حمایت می‌کند با شناسایی مشتریانی که ممکن است به درآمد بهره کمتر از حد

انتظار کمک کنند.

به استراتژی‌های قیمت‌گذاری وام کمک می‌کند و به بانک این امکان را می‌دهد که نرخ‌های بهره را به طور پویا تنظیم کند.

Clustering -3

هدف (سوال تجاری): آیا می‌توانیم مشتریان را بر اساس رفتار مالی‌شان به بخش‌های معنادار تقسیم کنیم؟

وروودی: موجودی حساب‌ها، فراوانی تراکنش‌ها، انواع خریدها، تاریخچه وام‌ها و عادات پس‌انداز مشتری.

خروجی: خوش‌هایی از مشتریان با رفتارهای مالی مشابه، مانند:

- سرمایه‌گذاران ثروتمند (High-Net-Worth Investors)
- وام‌گیرندگان مکرر (Frequent Borrowers)
- پس‌اندازکنندگان صرفه‌جو (Budget-Conscious Savers)
- حرفه‌ای‌های جوان (Young Professionals)

کاربرد: بانک می‌تواند استراتژی‌های بازاریابی را سفارشی کند، محصولات مالی هدفمند طراحی کند و خدمات مشتری را با تمرکز بر نیازهای هر بخش بهبود بخشد.

Association analysis -4

هدف (سوال تجاری): کدام محصولات بانکی به طور مکرر توسط مشتریان به طور همزمان استفاده می‌شوند؟

وروودی: داده‌های تراکنش، استفاده مشتری از محصولات (مانند حساب‌های پس‌انداز، وام‌ها، کارت‌های اعتباری، حساب‌های سرمایه‌گذاری).

خروجی: قوانین انجمنی، مانند:

مشتریان دارای وام مسکن ۷۰٪ احتمال دارند که یک حساب پس‌انداز بازنیستگی باز کنند.

مشتریانی که کارت اعتباری سفر دارند، اغلب برای وام‌های شخصی درخواست می‌دهند.

کاربرد: مدیران می‌توانند از این بینش‌ها برای فروش متقابل و بسته‌بندی محصولات مالی استفاده کنند، که به بهبود حفظ مشتری و افزایش درآمد کمک می‌کند.

سوال دوم

(A)

استخراج قوانین انجمنی (Association rule mining) یک تکنیک داده‌کاوی است که برای شناسایی روابط بین اقلام در دیتاست های بزرگ، به ویژه در پایگاه‌های داده تراکنشی استفاده می‌شود. این تکنیک الگوهایی را به صورت قوانین اگر آنگاه (if-then) کشف می‌کند.

مثلاً "اگر یک مشتری لپتاپ بخرد، احتمالاً ماوس هم می‌خرد. این قوانین بر اساس معیارهایی مانند:

- پشتیبانی (Support): چند بار اقلام به طور مشترک ظاهر می‌شوند
- اعتماد (Confidence): احتمال خرید آیتم B اگر آیتم A خریداری شود
- تقویت (Lift): قدرت رابطه نسبت به شانس تصادفی تولید می‌شوند

این تکنیک به طور گستره‌هایی در زمینه‌هایی مانند تحلیل سبد خرید، تشخیص تقلب و سیستم‌های توصیه‌گر در خرده‌فروشی و بانکداری به کار می‌رود.

با وجود مفید بودن آن، استخراج قوانین انجمنی با چالش‌ها و مشکلات متعددی هنگام اعمال بر روی داده‌های تراکنشی مواجه است. کمبود داده یک مشکل عمده است، زیرا بسیاری از تراکنش‌ها شامل ترکیب‌های منحصر به فرد اقلام هستند که یافتن الگوهای قوی را دشوار می‌کند.

مقیاس‌پذیری چالش دیگری است زیرا تحلیل پایگاه‌های داده بزرگ با میلیون‌ها تراکنش می‌تواند از نظر محاسباتی پرهزینه باشد. علاوه بر این، ممکن است قوانین اضافی یا بی‌اهمیت تولید شوند که شناسایی بینش‌های واقعاً ارزشمند را دشوار می‌کند.

مسائل دیگر شامل مدیریت داده‌های پویا (چرا که رفتار مشتریان با گذشت زمان تغییر می‌کند) و انتخاب آستانه‌های معنادار برای پشتیبانی و اعتماد به نفس به منظور جلوگیری از تولید قوانین بیش از حد یا کم است. غلبه بر این چالش‌ها نیازمند الگوریتم‌های کارآمد، تنظیم دقیق پارامترها و دانش خاص حوزه است.

(B)

تفاوت بین Fuzzy Association Rule Mining و Traditional Association Rule Mining

استخراج قوانین انجمنی به روش سنتی با داده‌های دقیق (دوتایی) کار می‌کند، جایی که یک آیتم یا در یک تراکنش وجود دارد یا وجود ندارد این روابط بین اقلام را با استفاده از معیارهایی مانند Confidence، Support و Lift پیدا می‌کند. با این حال، با داده‌های پیوسته یا مبهم مشکل دارد، زیرا به آستانه‌های سختگیرانه نیاز دارد (به عنوان مثال، یک تراکنش یا باید با قاعده مطابقت داشته باشد یا نداشته باشد).

از سوی دیگر قانون وابستگی فازی داده‌های پیوسته و نامشخص را با استفاده از منطق فازی پردازش می‌کند. به جای حضور یا عدم حضور دقیق یک مورد، اجازه می‌دهد که عضویت جزئی در یک مجموعه وجود داشته باشد. این در سناریوهای دنیای واقعی مفید است که در آن‌ها مقادیر به طور دقیق "بله" یا "خیر" نیستند بلکه در یک دامنه وجود دارند (مثلاً "درآمد متوسط" به جای فقط "بالا" یا "پایین").

مثال استخراج قواعد وابستگی سنتی:

یک بانک تراکنش‌ها را تحلیل می‌کند و می‌باید:

قاعده: مشتریانی که موجودی حساب آنها بالای ۱۰,۰۰۰ دلار است، احتمالاً برای دریافت کارت اعتباری اقدام می‌کنند.

تفسیر: اگر یک مشتری بیش از ۱۰,۰۰۰ دلار داشته باشد، در این قاعده حساب می‌شود. اگر آنها ۹۹۹۹ دلار داشته باشند، شامل نمی‌شوند.

این سختگیری می‌تواند منجر به از دست دادن بینش‌های ارزشمند شود زیرا کسی که ۹,۹۹۹ دلار دارد ممکن است مشابه کسی که ۱۰,۰۰۱ دلار دارد رفتار کند.

مثال استخراج قوانین وابستگی فازی:

به جای یک آستانه سخت، بانک از منطق فازی برای تعریف سطوح تعادل استفاده می‌کند:

قانون: مشتریان با موجودی بالا در حساب احتمالاً برای دریافت کارت اعتباری اقدام می‌کنند

تفسیر:

اگر موجودی ۵۰۰۰ دلار باشد، عضویت "موجودی بالا" ممکن است ۳٪ باشد (نسبتاً بالا).

اگر موجودی ۱۵,۰۰۰ دلار باشد، عضویت ۹٪ است (بالا).

اگر موجودی ۲۵,۰۰۰ دلار باشد، عضویت ۱٪ (کاملاً بالا) است.

این مزها را نرم می‌کند و انتقالات تدریجی را به تصویر می‌کشد، که منجر به تصمیم‌گیری‌های انعطاف‌پذیرتر و شبیه‌تر به انسان می‌شود.

(C)

دو کاربرد بالقوه برای استخراج قوانین وابستگی فازی:

1- تحلیل رفتار مشتری در بانکداری

کاربرد: شناسایی الگوهای خرج کردن و عادات مالی مشتریان.

مثال: به جای طبقه‌بندی سخت‌گیرانه مشتریان به عنوان "مشتریان پرخرج" یا "مشتریان کم‌خرج"، قوانین فازی می‌توانند آن‌ها را به دسته‌های "کم"، "متوسط" و "زیاد" با درجات عضویت مختلف تقسیم‌بندی کنند.

مزیت: به بانک‌ها کمک می‌کند تا پیشنهادات را شخصی‌سازی کنند، محدودیت‌های اعتباری پویا تعیین کنند و ارزیابی ریسک را با دقت بیشتری بهبود بخشد.

2- تشخیص پزشکی و پیشنهاد روش‌های درمانی

کاربرد: تحلیل علائم بیماران و نتایج آزمایش‌های پزشکی.

مثال: به جای تعریف "قند خون بالا" به عنوان مقدار بیشتر از 140 mg/dL ، قوانین فازی می‌توانند سطوح را به "طبیعی"، "مرزی" و "بالا" با گذارهای نرم‌تر دسته‌بندی کنند.

مزیت: ارائه تشخیص‌های دقیق‌تر، جلوگیری از آستانه‌های سخت و امکان برنامه‌ریزی درمانی شخصی‌سازی‌شده برای بیماران.

(D)

دو چالش رایج در Fuzzy Association Rule Mining

1- تعیین توابع عضویت مناسب

چالش: در داده‌کاوی قوانین انجمنی فازی، تعریف توابع عضویت (که مشخص می‌کند مقادیر چگونه به مجموعه‌های فازی تعلق دارند) بسیار مهم است. انتخاب نوع تابع (مثل مثلثی، ذوزنقه‌ای یا گوسی) و پارامترهای آن می‌تواند ذهنی و وابسته به حوزه مورد مطالعه باشد.

مثال: اگر بخواهیم سطح درآمد را به "کم"، "متوسط" و "زیاد" دسته‌بندی کنیم، باید مرزهای بین این دسته‌ها را با دقت تنظیم کنیم. اگر این مرزها به درستی تعیین نشوند، ممکن است منجر به طبقه‌بندی نادرست و استخراج قوانین غیر دقیق شود.

راه حل: استفاده از دانش کارشناسان یا تکنیک‌های بهینه‌سازی خودکار (مانند الگوریتم‌های ژنتیکی) برای تنظیم دقیق توابع عضویت.

2- پیچیدگی محاسباتی و انفجار قوانین (Rule Explosion)

چالش: داده‌کاوی قوانین انجمنی فازی محاسبات بیشتری نسبت به روش‌های سنتی نیاز دارد، زیرا هر آیتم در پایگاه داده می‌تواند به چندین مجموعه فازی با درجات مختلف عضویت تعلق داشته باشد. این امر باعث افزایش هزینه محاسباتی و تولید بیش از حد قوانین می‌شود.

مثال: در یک فروشگاه، اگر قیمت محصولات به "ارزان"، "متوسط" و "گران" دسته‌بندی شود، یک تراکنش ممکن است به چندین قانون فازی منجر شود، که در نتیجه زمان پردازش افزایش می‌یابد.

راه حل: استفاده از تکنیک‌های هرس (Pruning) یا تعیین آستانه‌های بالاتر برای مقادیر پشتیبانی (Confidence) و اطمینان (Support) به منظور کاهش تعداد قوانین غیر ضروری.

سوال سوم

(A)

1- مدیریت داده‌های گمشده (Missing Data):

گاهی اوقات دیتاست شامل مقادیر خالی یا گمشده هستند. شما می‌توانید این مقادیر را پر کنید (مثلاً با میانگین مقادیر دیگر) یا ردیف‌ها/ستون‌هایی با مقادیر گمشده زیاد را حذف کنید.

مثال: در یک دیتاست بیمارستانی، اگر برخی از مقادیر "فشار خون" ثبت نشده باشند، می‌توانید آن‌ها را با میانگین فشار خون سایر بیماران جایگزین کنید.

2- نرمال‌سازی داده‌ها یا مقیاس‌بندی:

این مرحله مقادیر را به یک مقیاس مشترک تبدیل می‌کند تا هیچ ویژگی خاصی بر مدل غلبه نکند. مثلاً تبدیل مقادیر به بازه‌ای بین ۰ و ۱.

مثال: در پیش‌بینی قیمت خانه، ویژگی‌هایی مانند "متراژ" ممکن است خیلی بزرگ‌تر از "تعداد اتاق‌ها" باشد، بنابراین مقیاس‌بندی کمک می‌کند تا تاثیر آن‌ها متعادل شود.

3- کدگذاری داده‌های دسته‌ای:

وقتی داده‌ها شامل دسته‌ها (مثل "قرمز"، "سبز"، "آبی") هستند، باید آن‌ها را به مقادیر عددی تبدیل کنید تا مدل یادگیری ماشین بتواند آن‌ها را پردازش کند.

مثال: در تحلیل بازخورد مشتریان، می‌توانید "راضی" را به ۱ و "ناراضی" را به ۰ تبدیل کنید تا مدل بتواند داده‌ها را تحلیل کند.

(B)

در داده‌کاوی، نویز و داده‌های پرت دو مفهوم مهم هستند که گاهی با هم اشتباه گرفته می‌شوند.

نویز در داده‌ها:

نویز به داده‌های غیرواقعی یا اشتباه گفته می‌شود که معمولاً به خاطر خطاهای اندازه‌گیری، مشکلات سنسورها، یا ورود نادرست داده‌ها ایجاد می‌شوند. نویز می‌تواند دقت مدل را کاهش دهد چون اطلاعات نادرست وارد تحلیل می‌شود.

مثال: در یک دیتاست دمای هوا، اگر به اشتباه دمای 1000 درجه ثبت شود، این مقدار یک نویز است که احتمالاً ناشی از خطای سنسور است.

داده‌های پرت (Outliers):

داده‌های پرت نقاطی هستند که به طرز قابل توجهی با بقیه داده‌ها متفاوت هستند، اما ممکن است واقعی و معنادار باشند. داده‌های پرت همیشه نویز نیستند؛ گاهی آن‌ها اطلاعات ارزشمندی درباره رخدادهای نادر دارند.

مثال: اگر میانگین حقوق کارکنان در یک شرکت 10 میلیون تومان باشد، ولی حقوق مدیرعامل 200 میلیون تومان باشد، این مقدار یک داده پرت است، اما لزوماً نویز نیست، چون می‌تواند یک واقعیت باشد.

تفاوت اصلی:

نویز معمولاً داده‌ای اشتباه یا بی‌معنی است که باید حذف یا اصلاح شود، اما داده‌های پرت می‌توانند واقعی باشند و گاهی شامل اطلاعات مهمی هستند.

(C)

نمونه‌گیری تصادفی ساده (Simple Random Sampling) :

در این روش، هر نمونه از جماعت با احتمال برابر انتخاب می‌شود. این کار مثل بیرون کشیدن کارت‌ها از یک جعبه به‌طور تصادفی است.

مزیت: ساده و آسان برای پیاده‌سازی.

عیب: اگر جماعت بسیار متنوع باشد، نمونه ممکن است نماینده واقعی کل جماعت نباشد.

مثال: اگر بخواهید نظر مردم درباره یک محصول جدید را بررسی کنید و از بین 1000 نفر به‌طور تصادفی 100 نفر را انتخاب کنید، این نمونه‌گیری تصادفی ساده است.

نمونه‌گیری طبقه‌بندی‌شده (Stratified Sampling) :

در این روش، جماعت به گروه‌های همگن (طبقه‌ها) تقسیم می‌شود و سپس از هر طبقه، نمونه‌های تصادفی برداشته می‌شود. این کار باعث می‌شود که هر زیرگروه در نمونه نهایی نماینده داشته باشد.

مزیت: دقیق‌تر و نماینده‌تر برای جماعت‌های نامتوازن.

عیب: نیاز به شناخت و تقسیم درست طبقه‌ها دارد.

مثال: اگر بخواهید نظر دانشآموزان درباره کیفیت غذا در مدرسه را بررسی کنید و مدرسه شامل مقاطع ابتدایی، راهنمایی و دبیرستان باشد، می‌توانید از هر مقطع به نسبت تعداد دانشآموزان، نمونه‌گیری کنید.

چه زمانی نمونه‌گیری طبقه‌بندی شده موثرتر است؟

زمانی که جمعیت به زیرگروه‌های مشخص و متفاوت تقسیم می‌شود، نمونه‌گیری طبقه‌بندی شده دقیق‌تر است. به‌ویژه اگر اندازه یا ویژگی‌های زیرگروه‌ها نامتعادل باشند، این روش کمک می‌کند که تحلیل منصفانه‌تر و قابل‌اعتماد‌تر باشد.

(D)

: (Curse of Dimensionality) نفرین ابعاد

وقتی تعداد ویژگی‌های داده (یا ابعاد) زیاد می‌شود، داده‌ها در فضای بسیار بزرگی پخش می‌شوند. این پراکندگی باعث مشکلاتی می‌شود:

فاصله بین نقاط افزایش می‌یابد: نقاط داده‌ها از هم دورتر می‌شوند، و این کار یادگیری الگوها را سخت‌تر می‌کند.

نیاز به داده‌های بیشتر: برای پوشش دادن تمام ترکیب‌های ممکن از ویژگی‌ها، باید حجم زیادی داده داشته باشید، و گرنه مدل دچار overfitting می‌شود.

کاهش عملکرد مدل: الگوریتم‌های یادگیری ماشین، مثل KNN یا شبکه‌های عصبی، در ابعاد بالا به سختی می‌توانند مرزهای تصمیم‌گیری دقیقی پیدا کنند.

مثال: فرض کنید می‌خواهید یک مدل برای تشخیص بیماری بسازید و 1000 ویژگی مختلف (مثل فشار خون، قند، سابقه بیماری و...) دارید. این تعداد زیاد ویژگی ممکن است مدل را گیج کند و نتایج نادرست بدهد.

چگونه PCA مشکل را حل می‌کند؟

تحلیل مولفه‌های اصلی (PCA) یک روش کاهش ابعاد است که ویژگی‌ها را به مجموعه‌ای جدید از ویژگی‌های مستقل تبدیل می‌کند، به‌طوری که بیشترین اطلاعات داده‌ها حفظ شود:

ایجاد مولفه‌های جدید: PCA محورهای جدیدی پیدا می‌کند که ترکیبی خطی از ویژگی‌های اولیه هستند. این مولفه‌ها طوری انتخاب می‌شوند که حداقل واریانس داده‌ها را پوشش دهند.

انتخاب مولفه‌های مهم: بعد از محاسبه مولفه‌ها، می‌توانید فقط مولفه‌هایی را نگه دارید که بیشترین اطلاعات را دارند و مولفه‌های با اهمیت کمتر را حذف کنید.

вшرده‌سازی داده‌ها: در نهایت، داده‌ها به فضای جدید و کم‌بعدتر نگاشت می‌شوند که تحلیل و یادگیری ماشین راحت‌تر انجام شود.

مثال: در همان مسئله پزشکی، PCA ممکن است از 1000 ویژگی اولیه، 20 مؤلفه اصلی بسازد که 90% اطلاعات کل داده‌ها را نگه می‌دارند. این کار باعث می‌شود مدل سریع‌تر و دقیق‌تر شود!

مزایای استفاده از PCA:

کاهش پیچیدگی: تعداد ابعاد کمتر = یادگیری سریع‌تر و ساده‌تر.

کمک به جلوگیری از بیشبرازش (Overfitting): ابعاد کمتر، مدل را عمومی‌تر و مقاوم‌تر می‌کند.

بهبود بصری‌سازی: داده‌ها در 2 یا 3 بعد، راحت‌تر تحلیل و مشاهده می‌شوند.

(E)

فرمول min-max normalization

$$x' = \frac{x - \min(x)}{\max(x) - \min(x)}$$

$$X_{\min} = 12$$

$$X_{\max} = 60$$

$$[12, 25, 30, 45, 60]$$

$$12 \rightarrow \frac{12-12}{60-12} = \frac{0}{48} = 0$$

$$25 \rightarrow \frac{25-12}{60-12} = \frac{13}{48} = 0.27$$

$$30 \rightarrow \frac{30-12}{60-12} = \frac{18}{48} = 0.375$$

$$45 \rightarrow \frac{45-12}{60-12} = \frac{33}{48} = 0.6875$$

$$60 \rightarrow \frac{60-12}{60-12} = \frac{48}{48} = 1$$

در نهایت:

[0 , 0.27 , 0.375 , 0.6875 , 1]

سوال چهارم

(A)

مقایسه معماری چندلایه‌ای انبار داده (Data Warehouse) با معماری سنتی سیستم‌های OLTP

معماری چندلایه‌ای انبار داده (Data Warehouse)

انبار داده معمولاً دارای معماری چندلایه‌ای است که شامل لایه‌های زیر می‌شود:

1. لایه منبع داده (Data Source Layer) : شامل داده‌های استخراج شده از سیستم‌های عملیاتی(OLTP)، فایل‌های خارجی و منابع دیگر.
2. لایه یکپارچه‌سازی (Integration Layer) : شامل فرایندهای ETL (استخراج، تبدیل و بارگذاری) برای تمیز کردن، تبدیل و ذخیره داده‌ها در انبار داده.
3. لایه ذخیره‌سازی (Storage Layer) : پایگاه داده مرکزی که داده‌های تاریخی و پردازش شده را نگه می‌دارد.
4. لایه پردازش و تحلیلی (Processing & Analytical Layer) : شامل موتورهای پردازش داده و ابزارهای OLAP برای تحلیل داده‌ها.
5. لایه ارائه و نمایش (Presentation Layer) : داشبوردها، گزارش‌ها و ابزارهای BI برای نمایش داده‌ها به کاربران.

این معماری به تحلیل داده‌ها در حجم بالا و از زوایای مختلف کمک می‌کند.

معماری سنتی سیستم‌های OLTP (Online Transaction Processing) سیستم‌های OLTP بر پردازش سریع تراکنش‌های روزمره تمرکز دارند و معمولاً دارای معنای سه‌لایه‌ای هستند:

1. **لایه رابط کاربری (Presentation Layer)**: شامل اپلیکیشن‌ها و وب‌سایت‌هایی که کاربران از طریق آن‌ها تراکنش انجام می‌دهند.
 2. **لایه منطقی (Business Logic Layer)**: شامل قوانین کسب‌وکار و پردازش‌های مربوط به تراکنش‌ها.
 3. **لایه پایگاه داده (Database Layer)**: پایگاه داده عملیاتی که به صورت نرم‌السازی شده ذخیره‌سازی می‌شود تا تراکنش‌ها را با حداقل افزونگی و حداکثر کارایی انجام دهد.
- سیستم‌های OLTP برای پردازش سریع و به‌روز بودن داده‌ها طراحی شده‌اند.

چرا OLAP موضوع محور (Subject-Oriented) است و OLTP کاربرد محور (Application-Oriented)؟

OLAP داده‌ها را بر اساس موضوعات خاصی مانند فروش، بازاریابی، مالی، منابع انسانی و غیره سازماندهی می‌کند. این سیستم‌ها داده‌ها را از منابع مختلف جمع‌آوری و ترکیب کرده و امکان تحلیل چندبعدی را فراهم می‌کنند.

مثال: بررسی روند فروش یک محصول در مناطق مختلف در ۵ سال گذشته.

OLTP بر پردازش تراکنش‌های روزمره و عملیاتی متمرکز است. ساختار پایگاه داده در OLTP برای پردازش سریع تراکنش‌ها بهینه شده و شامل جداول نرم‌السازی برای جلوگیری از افزونگی داده‌ها می‌باشد.

مثال: ثبت سفارش در یک فروشگاه آنلاین، ذخیره اطلاعات مشتری در بانک، ثبت نمرات دانشجویان در سیستم دانشگاهی.

(B)

Data Cube چیست؟

داده‌کوب یا Data Cube یک نمایش چندبعدی از داده‌ها در پردازش تحلیلی برخط (OLAP) است که امکان انجام تحلیل‌های پیچیده را فراهم می‌کند. این ساختار به کاربران اجازه می‌دهد داده‌ها را از زوایای مختلف بررسی کنند.

برای مثال، در یک فروشگاه خردۀ فروشی، داده‌کوب می‌تواند داده‌های فروش را با ابعاد زیر ذخیره کند:

فروش بر اساس منطقه جغرافیایی

مقایسه فروش بین دسته‌های مختلف محصولات

تحلیل فروش در بازه‌های زمانی مختلف

اجزای داده‌کوب (Data Cube)

یک داده‌کوب شامل دو مولفه اصلی است:

معیارها (Measures) یا واقعیت‌ها (Facts):

شامل ارزش‌های عددی هستند که متريک‌های کسب‌وکار مانند کل فروش، سود، درآمد را نشان می‌دهند.

این مقادیر در جدول واقعیت (Fact Table) ذخیره می‌شوند.

ابعاد (Dimensions):

ویژگی‌هایی که زاویه دید تحلیل داده‌ها را تعیین می‌کنند، مانند زمان، منطقه، دسته‌بندی محصول.

این ویژگی‌ها در جداول ابعادی (Dimension Tables) ذخیره می‌شوند.

سلول‌های داده‌کوب:

هر سلول نشان‌دهنده یک ترکیب خاص از مقادیر ابعاد با مقدار معیار مرتبط است.

مثال: مقدار فروش برای "لوازم الکترونیکی" در "نیویورک" در "زانویه ۲۰۲۴" برابر با ۵۰,۰۰۰ دلار است.

یک شرکت خردهفروشی میخواهد عملکرد فروش را بر اساس مناطق جغرافیایی و دسته‌بندی محصولات در طول زمان تحلیل کند. بهترین طرح پایگاه داده بستگی به پیچیدگی و ارتباط بین داده‌ها دارد.

پیشنهاد: استفاده از طرح ستاره‌ای (Star Schema)

چرا طرح ستاره‌ای؟

ساده و کارآمد: شامل یک جدول واقعیت مرکزی و چندین جدول ابعادی است.

اجرای سریع پرس‌وچوها: به دلیل ساختار غیرنرم‌الشده، بازیابی داده‌ها سریع‌تر از طرح‌های دیگر است.

قابل فهم و مدیریت آسان: کاربران کسب‌وکار و تحلیل‌گران می‌توانند به راحتی داده‌ها را جستجو و تحلیل کنند.

ساختار طرح ستاره‌ای برای تحلیل فروش:

جدول واقعیت: Sales_Fact (نگهداری اطلاعاتی مانند کل فروش، سود، درآمد)

جداول ابعادی:

Time_Dimension (تاریخ، ماه، سال)

Region_Dimension (شهر، ایالت، کشور)

Product_Dimension (دسته‌بندی، زیر دسته، برنده)

(C)

نقش OLAP در تحلیل نرخ بازپذیری بیماران

OLAP ابزاری برای تحلیل چندبعدی داده‌ها است که به تصمیم‌گیرندگان کمک می‌کند تا الگوهای کلی در داده‌های بیماران را شناسایی کنند.

مراحل استفاده از OLAP:

تحلیل روند بازپذیری بیماران در طول زمان: بررسی تعداد بیمارانی که در ۳۰ روز پس از ترخیص دوباره بستری شده‌اند.

مقایسه نرخ بازپذیری بر اساس بیماری‌ها و شرایط پزشکی: بررسی اینکه بیماران با بیماری‌های قلبی، دیابت، عفونت‌ها و... چقدر احتمال بازپذیری دارند.

بررسی تأثیر بیمارستان‌ها و پزشکان مختلف بر بازپذیری بیماران: مقایسه عملکرد بیمارستان‌های مختلف و تأثیر پزشکان بر میزان بازگشت بیماران.

بررسی تأثیر درمان‌های خاص بر کاهش نرخ بازپذیری: مقایسه بیمارانی که برنامه‌های مراقبتی ویژه دریافت کرده‌اند با بیمارانی که این برنامه‌ها را نداشته‌اند.

عملیات OLAP برای تحلیل نرخ بازپذیری:

Slice: بررسی داده‌های بازپذیری برای یک بیمارستان خاص در یک بازه زمانی مشخص.

Dice: بررسی نرخ بازپذیری در میان بیماران دیابتی بالای ۶۰ سال در دو بیمارستان مختلف.

Drill-down: تجزیه نرخ بازپذیری از ماهانه به روزانه برای تحلیل دقیق‌تر.

Roll-up: جمع‌بندی نرخ بازپذیری از بیمارستان‌های خاص به کل منطقه جغرافیایی.

نقش داده‌کاوی در پیش‌بینی نرخ بازپذیری بیماران
داده‌کاوی از تکنیک‌های یادگیری ماشین و الگوریتم‌های پیش‌بینی برای کشف الگوهای مخفی در
داده‌های بیماران استفاده می‌کند.

مراحل داده‌کاوی برای پیش‌بینی نرخ بازپذیری:

جمع‌آوری داده‌ها از انبار داده: اطلاعات بیماران، شرایط پزشکی، درمان‌های دریافت‌شده، داروها،
مدت بستری، میزان قند خون، فشار خون و...

پاک‌سازی و پردازش داده‌ها: حذف داده‌های ناقص یا نادرست و استانداردسازی اطلاعات.

انتخاب متغیرهای کلیدی: متغیرهایی مانند سن، بیماری‌های مزمن، نوع درمان، مدت بستری، سابقه
بستری‌های قبلی و وضعیت بیمه.

اجرای الگوریتم‌های داده‌کاوی:

درخت تصمیم (Decision Tree): برای دسته‌بندی بیماران با احتمال بالای بازپذیری.

شبکه‌های عصبی (Neural Networks): برای تشخیص الگوهای پیچیده در داده‌ها.

رگرسیون لجستیک (Logistic Regression): برای پیش‌بینی احتمال بازپذیری بیماران.

خوشه‌بندی (Clustering): برای شناسایی گروه‌هایی از بیماران که در معرض خطر بازپذیری قرار دارند.

تولید مدل پیش‌بینی و ارزیابی دقیق آن: مقایسه مدل‌های مختلف برای انتخاب بهترین مدل پیش‌بینی.

چگونه OLAP و داده‌کاوی با هم کار می‌کنند؟

OLAP به تحلیل‌گران کمک می‌کند تا روندها و عوامل تأثیرگذار بر بازپذیری بیماران را درک کنند.

داده‌کاوی از این اطلاعات برای ساخت مدل‌های پیش‌بینی و پیشنهاد راهکارهای کاهش نرخ بازپذیری استفاده می‌کند.

مثال همکاری OLAP و داده‌کاوی:

تحلیل‌گران با استفاده از OLAP مشاهده می‌کنند که بیماران قلبی بالای ۶۵ سال در بیمارستان X بیشترین میزان بازپذیری را دارند.

داده‌کاوی این داده‌ها را تحلیل کرده و مشخص می‌کند که کمبود برنامه‌های مراقبتی پس از ترخیص، عامل اصلی بازپذیری این بیماران است.

مدل داده‌کاوی می‌تواند پیش‌بینی کند که چه بیمارانی با احتمال زیاد دوباره بستری خواهند شد و به پزشکان هشدار دهد تا اقدامات پیشگیرانه انجام دهند.

سوال پنجم

(A)

بهبود برچسب‌گذاری داده‌ها با مدل‌های زبانی بزرگ (Large Language Models) می‌توانند به طور قابل توجهی کارایی و دقت برچسب‌گذاری داده‌ها در مجموعه‌های آموزشی را به ویژه در محیط‌های یادگیری نیمه‌نظرات شده و ضعیف‌نظرات شده بهبود بخشنند. در اینجا چند نکته کلیدی در مورد چگونگی کمک LLM‌ها به این فرآیند آورده شده است:

تولید خودکار برچسب (Automated Label Generation): LLM‌ها می‌توانند برچسب‌هایی برای داده‌های بدون برچسب تولید کنند و با درک زمینه و معناشناختی، این کار را انجام دهند. این موضوع به ویژه در یادگیری نیمه‌نظرات شده مفید است، جایی که مقدار کمی داده برچسب‌گذاری شده می‌تواند مدل را راهنمایی کند تا برچسب‌هایی برای مجموعه‌های بزرگ‌تر داده‌های بدون برچسب استنباط کند.

ادغام یادگیری فعال (Active Learning Integration): در سناریوهای یادگیری فعال، LLM‌ها می‌توانند نمونه‌های اطلاعاتی را شناسایی کنند که نیاز به برچسب‌گذاری دارند. با تجزیه و تحلیل عدم قطعیت در پیش‌بینی‌ها، LLM‌ها می‌توانند اولویت‌بندی کنند که کدام نقاط داده باید برچسب‌گذاری شوند، و بدین ترتیب فرآیند برچسب‌گذاری را بهینه‌سازی کنند و هزینه‌های مربوط به برچسب‌گذاری دستی را کاهش دهند.

تکنیک‌های نظرات ضعیف (Weak Supervision Techniques): LLM‌ها می‌توانند در تنظیمات ضعیف‌نظرات شده به کار گرفته شوند، جایی که از هنگارهای خاص دامنه برای استنباط برچسب‌ها استفاده می‌کنند. این رویکرد اجازه می‌دهد تا منابع مختلف اطلاعاتی ادغام شوند و فرآیند برچسب‌گذاری را بدون نیاز به مداخله دستی گسترش دهد.

مهندسی پرامپت (Prompt Engineering): با ظهور LLM‌ها، مهندسی پرامپت به عنوان یک تکنیک برای استخراج دانش از مدل‌ها بدون نیاز به آموزش مجدد آن‌ها به وجود آمده است. این امکان را فراهم می‌کند که محققان درک مدل را بررسی کرده و برچسب‌ها را بر اساس پرامپت‌های خاص تولید کنند، که فرآیند برچسب‌گذاری را کارآمدتر و متناسب با نیازهای مجموعه داده می‌سازد.

کنترل کیفیت (Quality Control): LLM‌ها همچنین می‌توانند در حفظ کیفیت داده‌ها کمک کنند و با شناسایی ناهماهنگی‌ها یا خطاهای برچسب‌گذاری شده، به این امر بپردازند. با تجزیه و تحلیل مجموعه داده‌های برچسب‌گذاری شده، LLM‌ها می‌توانند مشکلات احتمالی را علامت‌گذاری کنند و اطمینان حاصل کنند که داده‌های آموزشی قابل اعتماد و با کیفیت بالا باقی می‌مانند، که برای عملکرد مدل بسیار حیاتی است.

(B)

چالش‌ها و مزایای استفاده از LLM‌ها برای پیش‌پردازش داده‌ها

استفاده از مدل‌های زبانی بزرگ (LLM‌ها) برای پیش‌پردازش داده‌ها، بهویژه در وظایفی مانند پاک‌سازی، استخراج ویژگی و تبدیل داده‌ها، چالش‌ها و مزایای خاصی را به همراه دارد، بهویژه زمانی که با داده‌های غیرساختاری یا پر از نویز مواجه هستیم.

مزایا:

پاک‌سازی خودکار داده‌ها (Automated Data Cleaning): LLM‌ها می‌توانند در شناسایی و اصلاح خطاهای در مجموعه‌های داده کمک کنند، مانند اشتباهات املایی یا فرمتبندی نامنظم. این خودکارسازی می‌تواند به طور قابل توجهی زمان و تلاش مورد نیاز برای پاک‌سازی دستی داده‌ها را کاهش دهد.

استخراج ویژگی (Feature Extraction): LLM‌ها در درک زمینه و معنا بسیار قوی هستند، که به آن‌ها این امکان را می‌دهد که ویژگی‌های معناداری را از داده‌های غیرساختاری، مانند متن، استخراج کنند. این قابلیت می‌تواند کیفیت ویژگی‌های مورد استفاده در مدل‌های یادگیری ماشین را بهبود بخشد و منجر به عملکرد بهتر شود.

تبدیل داده‌ها (Data Transformation): LLM‌ها می‌توانند فرآیند تبدیل داده‌ها به فرمتهای مناسب برای تحلیل را تسهیل کنند. به عنوان مثال، آن‌ها می‌توانند متن خام را به داده‌های ساختاری تبدیل کنند و با شناسایی موجودیت‌ها و روابط کلیدی، تحلیل و مدل‌سازی را آسان‌تر کنند.

مدیریت داده‌های پر از نویز (Handling Noisy Data): LLMها برای کار با حجم زیادی از داده‌ها، از جمله ورودی‌های غیرساختاری و پر از نویز طراحی شده‌اند. توانایی آن‌ها در یادگیری از زمینه به آن‌ها کمک می‌کند تا نویز را فیلتر کرده و بر اطلاعات مرتبط تمرکز کنند و کیفیت کلی داده‌ها را بهبود بخشنند.

چالش‌ها

پیچیدگی پیاده‌سازی (Complexity of Implementation): ادغام LLMها در خطوط لوله پیش‌پردازش داده‌های موجود می‌تواند پیچیده باشد. سازمان‌ها ممکن است با چالش‌هایی در سازگاری سیستم‌های خود برای بهره‌برداری مؤثر از قابلیت‌های LLM مواجه شوند.

نیاز به منابع زیاد: LLMها برای آموزش و استنتاج به منابع محاسباتی قابل توجهی نیاز دارند. این می‌تواند مانعی برای سازمان‌های کوچک‌تر یا آن‌هایی باشد که دسترسی محدودی به منابع محاسباتی با عملکرد بالا دارند.

کیفیت خروجی: در حالی که LLMها می‌توانند کیفیت داده‌ها را بهبود بخشنند، آن‌ها بی‌نقص نیستند. خطر تولید خروجی‌های نادرست یا مغرضانه وجود دارد، بهویژه اگر داده‌های آموزشی شامل تعصبات یا نادرستی‌ها باشد. این می‌تواند به پیچیدگی‌های بیشتری در مرحله پیش‌پردازش منجر شود.

وابستگی به داده‌های آموزشی: اثربخشی LLMها به شدت به کیفیت و تنوع داده‌های آموزشی وابسته است. اگر داده‌های آموزشی به‌طور کافی نمایانگر دامنه هدف نباشند، عملکرد مدل ممکن است تحت تأثیر قرار گیرد و منجر به نتایج ضعیف در پیش‌پردازش داده‌ها شود.

(C)

مدل‌های زبانی بزرگ (LLM‌ها) می‌توانند به طور قابل توجهی تجسم داده‌ها را در زمینه هوش مصنوعی متمرکز بر داده تقویت کنند و به طور خودکار بینش‌ها و خلاصه‌هایی از مجموعه‌های داده پیچیده تولید کنند. در اینجا نحوه کمک آن‌ها و همچنین محدودیت‌های احتمالی آن‌ها آورده شده است:

تقویت‌های ارائه شده توسط LLMها

تولید بینش‌های خودکار (Automatic Insight Generation): LLMها می‌توانند حجم زیادی از داده‌ها را تجزیه و تحلیل کرده و الگوها یا روندهای معناداری را استخراج کنند. با پردازش داده‌های متنه، آن‌ها می‌توانند یافته‌های کلیدی را خلاصه کنند و این امر به کاربران کمک می‌کند تا بدون نیاز به بررسی داده‌های خام، مجموعه‌های داده پیچیده را درک کنند.

خلاصه‌های زبان طبیعی (Natural Language Summaries): LLMها می‌توانند تجسم‌های داده را به توصیف‌های زبان طبیعی تبدیل کنند. این قابلیت به کاربران اجازه می‌دهد تا به سرعت پیامدهای داده‌های ارائه شده در نمودارها یا گراف‌ها را درک کنند و تصمیم‌گیری بهتری بر اساس اطلاعات تجسم شده انجام دهند.

تحلیل زمینه‌ای (Contextual Analysis): با استفاده از درک خود از زمینه، LLMها می‌توانند بینش‌هایی ارائه دهند که پیامدهای گسترده‌تری از داده‌ها را در نظر می‌گیرند. این می‌تواند شامل شناسایی همبستگی‌ها یا ناهنجاری‌هایی باشد که ممکن است به راحتی از طریق بازرسی بصری قابل مشاهده نباشند.

کاوش تعاملی داده (Interactive Data Exploration): LLMها می‌توانند تعامل کاربر با تجسم‌های داده را با اجازه دادن به کاربران برای پرسیدن سوالات به زبان طبیعی تقویت کنند. مدل سپس می‌تواند تجسم‌ها یا خلاصه‌های مرتبط را بر اساس پرسش‌ها تولید کند و کاوش داده را شهودی‌تر کند.

محدویت‌های احتمالی وابستگی به LLMها

کیفیت بینش‌ها (Quality of Insights): اثربخشی LLMها در تولید بینش‌ها به شدت به کیفیت داده‌های ورودی وابسته است. اگر داده‌ها پر سرو صدا یا مغرضانه باشند، بینش‌های تولید شده نیز ممکن است نادرست باشند و به نتایج نادرستی منجر شوند.

مسائل تفسیرپذیری (Interpretability Issues): در حالی که LLMها می‌توانند خلاصه‌هایی تولید کنند، استدلال پشت بینش‌های آن‌ها همیشه ممکن است شفاف نباشد. کاربران ممکن است در اعتماد به بینش‌های تولید شده بدون درک منطق زیرین آن‌ها با چالش مواجه شوند.

بیش‌برازش به داده‌های آموزشی (Overfitting to Training Data): LLMها ممکن است بینش‌هایی تولید کنند که الگوهای موجود در داده‌های آموزشی خود را منعکس کنند و نه داده‌های واقعی که در

حال تجزیه و تحلیل هستند. این می‌تواند به نتایج گمراه‌کننده‌ای منجر شود اگر مدل به مثال‌های خاصی بیش‌برازش کند و نتواند به طور مؤثر تعمیم یابد.

نیاز به منابع: استفاده از LLM‌ها برای وظایف تجسم داده می‌تواند نیاز به منابع زیادی داشته باشد و به قدرت محاسباتی و زمان قابل توجهی نیاز دارد که ممکن است برای همه سازمان‌ها قابل اجرا نباشد.

سوال ششم

(A)

مقدار شباهت کسینوسی (cosine similarity) از -1 تا 1 متغیر است.

- ۱ نشان می‌دهد که دو بردار یکسان هستند (یعنی به یک جهت اشاره می‌کنند)
- ۰ به این معنی است که بردارها عمود بر هم هستند (یعنی هیچ شباهتی ندارند).
- 1 نشان می‌دهد که بردارها کاملا در جهت مخالف هستند.

برای بیشتر کاربردهای دنیای واقعی (مانند تشابه متنی و سیستم‌های توصیه‌گر)، مقادیر شباهت کسینوسی معمولاً بین ۰ و ۱ متغیر است زیرا مؤلفه‌های بردار (مانند فراوانی کلمات) معمولاً غیرمنفی هستند.

(B)

اگر دو شی دارای مقدار 1 برای شباهت کسینوسی باشند، لزوماً به این معنی نیست که آن‌ها کاملاً یکسان هستند، بلکه به این معنی است که آن‌ها در یک جهت یکسان در فضای برداری قرار دارند.

توضیح:

شباهت کسینوسی فقط زاویه بین دو بردار را در نظر می‌گیرد، نه اندازه آن‌ها را. بنابراین، اگر دو بردار دارای مقدار کسینوسی ۱ باشند، این نشان می‌دهد که آن‌ها دقیقاً در یک جهت قرار دارند، اما ممکن است طول (اندازه) آن‌ها متفاوت باشد.

(C)

رابطه بین شباهت کسینوسی و همبستگی (Correlation)

شباهت کسینوسی (Cosine Similarity) و همبستگی پیرسون (Pearson Correlation) هر دو معیارهایی برای سنجش شباهت یا ارتباط بین دو بردار هستند، اما تفاوت‌هایی کلیدی دارند که به میانگین و انحراف معیار داده‌ها مربوط می‌شود.

شباهت‌ها:

هر دو معیار جهت بردارها را در نظر می‌گیرند، نه بزرگی آن‌ها

هم شباهت کسینوسی و هم همبستگی پیرسون بر اساس زاویه بین بردارها محاسبه می‌شوند. اگر دو بردار در یک جهت قرار داشته باشند، مقدار هر دو معیار نزدیک به ۱ خواهد بود، و اگر در جهت مخالف باشند، مقدارشان نزدیک به -۱ می‌شود.

تفاوت‌های کلیدی:

تأثیر میانگین و انحراف معیار

شباهت کسینوسی میانگین و مقیاس داده‌ها را در نظر نمی‌گیرد این معیار فقط زاویه بین بردارها را محاسبه می‌کند، بدون اینکه تفاوت در مقدارهای میانگین و پراکندگی داده‌ها را در نظر بگیرد.

همبستگی پیرسون میانگین و انحراف معیار را حذف می‌کند

در همبستگی، هر مقدار از میانگین خود کسر و بر انحراف معیار نرمال‌سازی می‌شود.

بنابراین، همبستگی پیرسون به تغییرات نسبی داده‌ها توجه دارد، نه صرفاً جهت بردارها.

مواردی که شباهت کسینوسی و همبستگی پیرسون برابرند

اگر بردارها میانگین صفر داشته باشند، یعنی مقدارهای ایشان از مقدار میانگین خود تفرقی شده باشند، آنگاه شباهت کسینوسی و همبستگی پیرسون یکسان خواهند بود.

دلیل این موضوع این است که نرمال‌سازی همبستگی پیرسون دقیقاً همان اثری را دارد که شباهت کسینوسی روی داده‌های بدون میانگین اعمال می‌کند.

مواردی که این دو معیار متفاوت هستند

اگر دو بردار دارای میانگین‌های متفاوتی باشند (مثلاً یکی در مقیاس بزرگ‌تر از دیگری باشد)، شباهت کسینوسی همچنان ممکن است مقدار بالایی بدهد، اما همبستگی پیرسون به دلیل محاسبه با میانگین‌ها ممکن است مقدار متفاوتی بدهد.

همبستگی پیرسون زمانی که دو بردار مقدارهای مشابهی دارند اما در مقیاس‌های مختلف هستند (مثلاً یکی دو برابر دیگری است)، مقدار ۱ را نمی‌دهد، اما شباهت کسینوسی مقدار ۱ خواهد داشت.

(D)

رابطه بین فاصله اقلیدسی و شباهت کسینوسی برای بردارهای با نرم L₂ برابر با ۱

کاهش فاصله اقلیدسی با افزایش شباهت کسینوسی

همان‌طور که شباهت کسینوسی افزایش می‌یابد (به ۱ نزدیک می‌شود)، فاصله اقلیدسی کاهش پیدا می‌کند.

وقتی شباهت کسینوسی ۱ است (بردارها دقیقاً در یک جهت هستند)، فاصله اقلیدسی صفر می‌شود.

رابطه غیرخطی بین فاصله اقلیدسی و شباهت کسینوسی

این رابطه خطی نیست؛ بلکه یک روند نزولی غیرخطی دارد.

افزایش شباهت کسینوسی در مقادیر بالا (مثلاً از ۰٪ به ۱۰٪) تاثیر بیشتری بر کاهش فاصله اقلیدسی دارد نسبت به مقادیر پایین‌تر.

هرچه شباهت کسینوسی بیشتر باشد، فاصله اقلیدسی کمتر خواهد بود.

این رابطه زمانی دقیق‌تر است که بردارها دارای نُرم L_2 برابر با ۱ باشند، زیرا در این حالت فاصله اقلیدسی تنها تابعی از زاویه بین بردارها است.

برای بردارهای نرمال‌شده، شباهت کسینوسی و فاصله اقلیدسی اطلاعات مشابهی را ارائه می‌دهند، اما در مقیاس‌های مختلف.

(E)

رابطه بین فاصله اقلیدسی و همبستگی پیرسون برای بردارهای استاندارد شده (میانگین = ۰، انحراف معیار = ۱)

کاهش فاصله اقلیدسی با افزایش همبستگی پیرسون

همان‌طور که همبستگی پیرسون افزایش می‌یابد (به ۱ نزدیک می‌شود) فاصله اقلیدسی کاهش پیدا می‌کند.

وقتی همبستگی پیرسون ۱ است (یعنی دو بردار کاملاً همبسته هستند) فاصله اقلیدسی صفر می‌شود.

اگر همبستگی برابر -۱ باشد (یعنی دو بردار کاملاً در جهت مخالف باشند) فاصله اقلیدسی به مقدار بالایی می‌رسد.

رابطه غیرخطی بین فاصله اقلیدسی و همبستگی پیرسون

این رابطه خطی نیست بلکه یک روند نزولی غیرخطی دارد.

همانند شباهت کسینوسی در شکل 1(a)، کاهش فاصله اقلیدسی زمانی که همبستگی نزدیک به ۱ است، سریع‌تر اتفاق می‌افتد.

هرچه همبستگی پیرسون بیشتر باشد، فاصله اقلیدسی کمتر خواهد بود.

در بردارهای استاندارد شده (با میانگین ۰ و انحراف معیار ۱)، فاصله اقلیدسی و همبستگی پیرسون اساساً اطلاعات مشابهی را ارائه می‌دهند، زیرا هر دو به زاویه بین بردارها وابسته می‌شوند.

برای داده‌های استاندارد، فاصله اقلیدسی و همبستگی پیرسون می‌توانند به جای یکدیگر استفاده شوند، زیرا رابطه‌ای مشخص بین آن‌ها وجود دارد.

(F)

تعریف شباخت کسینوسی

شباخت کسینوسی بین دو بردار x و y به صورت زیر تعریف می‌شود:

$$\cos(\theta) = \frac{x \cdot y}{|x||y|}$$

از آنجا که هر بردار دارای طول واحد (نرم برابر ۱) است، یعنی $1 = \|x\| = \|y\|$ فرمول بالا ساده می‌شود:

$$\cos(\theta) = x \cdot y$$

تعریف فاصله‌ی اقلیدسی

فاصله اقلیدسی بین x و y به صورت زیر تعریف می‌شود:

$$d(x, y) = \|x - y\|$$

با محاسبه‌ی مربع فاصله اقلیدسی:

$$d^2(x, y) = (x - y) \cdot (x - y)$$

با استفاده از خاصیت ضرب داخلی:

$$d^2(x, y) = x \cdot x - 2x \cdot y + y \cdot y$$

از آنجا که $1 = \|x\|^2 = \|y\|^2$ داریم:

$$d^2(x, y) = 1 - 2(x \cdot y) + 1$$

$$d^2(x, y) = 2(1 - x \cdot y)$$

رابطه‌ی نهایی:

چون $x \cdot y = \cos(\theta)$ جایگذاری می‌کنیم:

$$d^2(x, y) = 2(1 - \cos(\theta))$$

با گرفتن جذر دو طرف:

$$d(x, y) = \sqrt{2(1 - \cos(\theta))}$$

شباخت کسینوسی میزان زاویه‌ی بین دو بردار را اندازه‌گیری می‌کند.

فاصله‌ی اقلیدسی میزان فاصله‌ی مستقیم بین دو نقطه را نشان می‌دهد.

اگر بردارها دارای طول واحد باشند، شباخت کسینوسی و فاصله اقلیدسی مستقیماً به هم مرتبط هستند و می‌توان یکی را از دیگری محاسبه کرد.

این رابطه در یادگیری ماشین و بازیابی اطلاعات بسیار کاربرد دارد، زیرا در شرایطی که بردارها نرمال‌سازی شده باشند می‌توان از این دو معیار به‌طور جایگزین استفاده کرد.

(G)

تعریف استانداردسازی و مفروضات

فرض کنید دو بردار داده‌ای $Y = (y_1, y_2, \dots, y_n)$ و $X = (x_1, x_2, \dots, x_n)$ داریم. اگر داده‌ها استاندارد شده باشند، هر مقدار جدید به صورت زیر محاسبه می‌شود:

$$x'_i = \frac{x_i - \bar{x}}{s_x}, \quad y'_i = \frac{y_i - \bar{y}}{s_y}$$

که در آن :

\bar{x} و \bar{y} میانگین‌های بردارهای x و y هستند.

s_x و s_y انحراف معیارهای بردارها هستند.

پس از استانداردسازی، بردارهای جدید x' و y' ویژگی‌های زیر را دارند:

میانگین آنها صفر است و واریانس آنها برابر با ۱ است.

تعریف همبستگی پیرسون

همبستگی پیرسون بین دو بردار به صورت زیر تعریف می‌شود:

$$r = \frac{\sum x'_i y'_i}{n}$$

با توجه به اینکه بردارهای استاندارد شده دارای میانگین صفر و واریانس یک هستند، این رابطه به صورت ضرب داخلی ساده می‌شود:

$$r = \frac{x' \cdot y'}{n}$$

که در آن $x' \cdot y'$ ضرب داخلی دو بردار استاندارد شده است.

محاسبه فاصله اقلیدسی بین بردارهای استاندارد شده

فاصله اقلیدسی بین دو بردار استاندارد شده به صورت زیر تعریف می‌شود:

$$d(x', y') = \|x' - y'\|$$

مربع این فاصله برابر است با:

$$d^2(x', y') = (x' - y') \cdot (x' - y')$$

با استفاده از خواص ضرب داخلی:

$$d^2(x', y') = x' \cdot x' - 2x' \cdot y' + y' \cdot y'$$

از آنجا که بردارهای استاندارد شده واریانس واحد دارند داریم:

$$x' \cdot x' = n, y' \cdot y' = n$$

پس داریم:

$$\begin{aligned} d^2(x', y') &= n + n - 2(x' \cdot y') \\ &= 2n - 2(x' \cdot y') \end{aligned}$$

چون می‌دانیم که $x' \cdot y' = nr$ جایگذاری می‌کنیم:

$$d^2(x', y') = 2n - 2nr$$

فاکتور می‌گیریم:

$$d^2(x', y') = 2n(1 - r)$$

سپس با گرفتن جذر به رابطه زیر می‌رسیم:

$$d(x', y') = \sqrt{2n(1 - r)}$$

رابطه نهایی

بنابراین، فاصله اقلیدسی بین دو بردار استاندارد شده به همبستگی پیرسون مرتبط است:

$$d(x', y') = \sqrt{2n(1 - r)}$$

که در آن:

$d(x', y')$ فاصله اقلیدسی بین بردارهای استاندارد شده است.

۲ ضریب همبستگی پیرسون است.

n تعداد داده‌ها است.

تفسیر رابطه

اگر $r = 1$ (همبستگی کامل مثبت) باشد $d = 0$ یعنی دو بردار دقیقاً یکسان هستند.

اگر $r = 0$ (عدم همبستگی) باشد، $d = \sqrt{2n}$ که نشان می‌دهد بردارها به بیشترین فاصله ممکن در فضای استاندارد شده رسیده‌اند.

اگر $-1 < r < 0$ (همبستگی کامل منفی) باشد $d = \sqrt{4n}$ که $d = 2\sqrt{n}$ یعنی بردارها در جهت کاملاً مخالف قرار دارند.

سوال هفتم

(A)

در این الگوریتم برای یافتن K همسایه نزدیک یک data object طراحی شده است. این الگوریتم به صورت زیر کار می‌کند:

برای هر شی مراحل زیر اجرا می‌شود:

- فاصله این شی با تمامی اشیای دیگر محاسبه می‌شود.
- فاصله‌ها بر اساس مقدارشان به ترتیب نزولی مرتب می‌شوند.
- همچنین باید مشخص کنیم که هر فاصله مربوط به کدام شی است.
- K شی اول در لیست مرتب شده به عنوان همسایگان نزدیک انتخاب و برگردانده می‌شوند.

- این فرآیند برای همه اشیای داده‌ای تکرار می‌شود.

در این الگوریتم فاصله‌ها به صورت نزولی مرتب شده و زمانی که ما K تای اول لیست را به عنوان همسایه برگردانیم در واقع آن شی‌هایی برگردانده می‌شود که بیشترین فاصله را با شی‌ما دارند و عملاً همسایه نیستند.

(B)

اگر در مجموعه داده اشیای تکراری وجود داشته باشد، این الگوریتم ممکن است با مشکلاتی روبرو شود:

از آنجایی که تابع فاصله فقط برای اشیای کاملاً یکسان مقدار 0 برمی‌گرداند، اگر یک شیء چندین کپی تکراری داشته باشد، الگوریتم ممکن است آن‌ها را به عنوان نزدیک‌ترین همسایگان انتخاب کند، حتی اگر این کار منطقی نباشد.

در حضور داده‌های تکراری، ممکن است الگوریتم به جای انتخاب همسایگان متنوع، فقط نسخه‌های تکراری یک شی را انتخاب کند. این مسئله می‌تواند باعث کاهش دقیقی الگوریتم شود، به‌ویژه در کاربردهایی مانند طبقه‌بندی (classification) که نیاز به تنوع همسایگان دارد.

هنگام مرتب‌سازی فاصله‌ها، همه اشیای تکراری که فاصله‌ی صفر دارند، در بالای لیست قرار می‌گیرند. در نتیجه، الگوریتم ممکن است تقریباً فقط اشیای تکراری را به عنوان نزدیک‌ترین همسایگان بازگرداند، در حالی که اشیای دیگر که ممکن است مرتبط‌تر باشند، نادیده گرفته می‌شوند.

اگر یک شی دارای چندین نسخه تکراری باشد، این امکان وجود دارد که دیگر اشیای معنادار ولی کمی دورتر از لیست همسایگان حذف شوند. این مسئله می‌تواند عملکرد الگوریتم را در کاربردهایی مثل خوشبندی (clustering) یا سیستم‌های توصیه‌گر (recommendation systems) تضعیف کند.

(C)

برای رفع این مشکل در الگوریتم KNN می‌توان تغییرات زیر را اعمال کرد
نادیده گرفتن اشیای کاملاً تکراری هنگام انتخاب همسایگان:

پس از مرتب‌سازی فاصله‌ها، می‌توان اشیایی که فاصله‌ی آن‌ها ۰ است و دقیقاً مشابه شی مورد بررسی هستند را رد کرد و به انتخاب از میان اشیای منحصر‌به‌فرد ادامه داد.

استفاده از مجموعه (Set) یا دیکشنری برای ذخیره همسایگان منحصر‌به‌فرد

به جای اینکه فقط اولین K مقدار از لیست مرتب‌شده را انتخاب کنیم، می‌توان از یک مجموعه یا دیکشنری برای اطمینان از انتخاب اشیای منحصر‌به‌فرد استفاده کرد.

کاهش تأثیر اشیای تکراری با یک رویکرد وزنی

به جای حذف کامل اشیای تکراری، می‌توان تأثیر آن‌ها را کاهش داد تا در انتخاب همسایگان، اهمیت کمتری داشته باشند.

تغییر معیار مرتب‌سازی

به جای مرتب‌سازی فقط بر اساس فاصله، می‌توان اشیای منحصر‌به‌فرد را در اولویت قرار داد تا در صورت وجود تعداد زیادی داده‌ی تکراری، همسایگان معنادارتری انتخاب شوند.

سوال هشتم

(A)

```
from sklearn.datasets import load_wine
import pandas as pd

data = load_wine()

df = pd.DataFrame(data.data, columns=data.feature_names)

df['target'] = data.target

print("First 5 rows of the dataset:")
df.head()
```

دیتاست رو لود کردیم و نمایش دادیم

	alcohol	malic_acid	ash	alcalinity_of_ash	magnesium	total_phenols	flavanoids	nonflavanoid_phenols	proanthocyanins	color_intensity	hue	od280/od315_of_diluted_wines	proline	target	
0	14.23	1.71	2.43		15.6	127.0	2.80	3.06	0.28	2.29	5.64	1.04	3.92	1065.0	0
1	13.20	1.78	2.14		11.2	100.0	2.65	2.76	0.26	1.28	4.38	1.05	3.40	1050.0	0
2	13.16	2.36	2.67		18.6	101.0	2.80	3.24	0.30	2.81	5.68	1.03	3.17	1185.0	0
3	14.37	1.95	2.50		16.8	113.0	3.85	3.49	0.24	2.18	7.80	0.86	3.45	1480.0	0
4	13.24	2.59	2.87		21.0	118.0	2.80	2.69	0.39	1.82	4.32	1.04	2.93	735.0	0

```
print("\nDataset Information:")
df.info()
```

اطلاعات دیتاست را چاپ کردیم:

```
Dataset Information:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 178 entries, 0 to 177
Data columns (total 14 columns):
 #   Column           Non-Null Count  Dtype  
 ---  --  
 0   alcohol          178 non-null    float64
 1   malic_acid       178 non-null    float64
 2   ash               178 non-null    float64
 3   alcalinity_of_ash 178 non-null    float64
 4   magnesium         178 non-null    float64
 5   total_phenols     178 non-null    float64
 6   flavanoids        178 non-null    float64
 7   nonflavanoid_phenols 178 non-null    float64
 8   proanthocyanins  178 non-null    float64
 9   color_intensity   178 non-null    float64
 10  hue               178 non-null    float64
 11  od280/od315_of_diluted_wines 178 non-null    float64
 12  proline           178 non-null    float64
 13  target             178 non-null    int64  
dtypes: float64(13), int64(1)
memory usage: 19.6 KB
```

```
print("\nClass Distribution:")
print(df['target'].value_counts())
```

توزیع کلاس ها به شکل زیر است

```
Class Distribution:
target
1    71
0    59
2    48
Name: count, dtype: int64
```

```
print("\nFeature Names:", data.feature_names)
print("Target Class Names:", data.target_names)
```

نام ویژگی‌های دیتاست و نام دسته‌بندی‌های شراب را نمایش دادیم.

```
Feature Names: ['alcohol', 'malic_acid', 'ash', 'alcalinity_of_ash', 'magnesium', 'total_phenols', 'flavanoids', 'nonflavanoid_phenols', 'proanthocyanins', 'color_intensity', 'hue', 'od280/od315_of_diluted_wines', 'proline']
Target Class Names: ['class_0', 'class_1', 'class_2']
```

(B)

```
min_values = df.min()
feature_with_smallest_values = min_values.idxmin()
print("\nFeature with the smallest values:", feature_with_smallest_values)

max_values = df.max()
feature_with_largest_values = max_values.idxmax()
print("Feature with the largest values:", feature_with_largest_values)

range_values = max_values - min_values
feature_with_largest_range = range_values.idxmax()
print("Feature with the largest range:", feature_with_largest_range)
```

```
Feature with the smallest values: target
Feature with the largest values: proline
Feature with the largest range: proline
```

در این دیتاست:

ویژگی با کوچک‌ترین مقدار : **target**

ویژگی با بزرگ‌ترین مقدار : **proline**

ویژگی با بزرگ‌ترین دامنه تغییرات : **proline**

ویژگی **target** کوچک‌ترین مقادیر را دارد، در حالی که **proline** هم بیشترین مقدار را دارد و هم گسترده‌ترین دامنه تغییرات را نشان می‌دهد.

(C)

```
correlation_matrix = df.corr()
print("\nCorrelation Matrix:")
correlation_matrix
```

چاپ ماتریس همبستگی

	alcohol	malic_acid	ash	alcalinity_of_ash	magnesium	total_phenols	flavonoids	nonflavanoid_phenols	proanthocyanins	color_intensity	hue	od280/od315_of_diluted_wines	proline	target
alcohol	1.000000	0.094397	0.211545	-0.310235	0.270798	0.289101	0.236815	-0.155929	0.136698	0.546364	-0.071747	0.072343	0.643720	-0.328222
malic_acid	0.094397	1.000000	0.164045	0.288500	-0.054575	-0.335167	-0.411007	0.292977	-0.220746	0.248985	-0.561296	-0.368710	-0.192011	0.437776
ash	0.211545	0.164045	1.000000	0.443367	0.286587	0.128980	0.115077	0.186230	0.009652	0.258887	-0.074667	0.003911	0.223626	-0.049643
alcalinity_of_ash	-0.310235	0.288500	0.443367	1.000000	-0.083333	-0.321113	-0.351370	0.361922	-0.197327	0.018732	-0.273955	-0.276769	-0.440597	0.517889
magnesium	0.270798	-0.054575	0.286587	-0.083333	1.000000	0.214401	0.195784	-0.256294	0.236441	0.199950	0.055398	0.066004	0.393351	-0.209179
total_phenols	0.289101	-0.335167	0.128980	-0.321113	0.214401	1.000000	0.864564	-0.449935	0.612413	-0.055136	0.433681	0.699549	0.498115	-0.719163
flavonoids	0.236815	-0.411007	0.115077	-0.351370	0.195784	0.864564	1.000000	-0.537900	0.652692	-0.172379	0.543479	0.787194	0.494193	-0.847498
nonflavanoid_phenols	-0.155929	0.292977	0.186230	0.361922	-0.256294	-0.449935	-0.537900	1.000000	-0.365845	0.139057	-0.262640	-0.503270	-0.311385	0.489109
proanthocyanins	0.136698	-0.220746	0.009652	-0.197327	0.236441	0.612413	0.652692	-0.365845	1.000000	-0.025250	0.295544	0.519067	0.330417	-0.499130
color_intensity	0.546364	0.248985	0.258887	0.018732	0.199950	-0.055136	-0.172379	0.139057	-0.025250	1.000000	-0.521813	-0.428815	0.316100	0.265668
hue	-0.071747	-0.561296	-0.074667	-0.273955	0.055398	0.433681	0.543479	-0.262640	0.295544	-0.521813	1.000000	0.565468	0.236183	-0.617369
od280/od315_of_diluted_wines	0.072343	-0.368710	0.003911	-0.276769	0.066004	0.699949	0.787194	-0.503270	0.519067	-0.428815	0.565468	1.000000	0.312761	-0.788230
proline	0.643720	-0.192011	0.223626	-0.440597	0.393351	0.498115	0.494193	-0.311385	0.330417	0.316100	0.236183	0.312761	1.000000	-0.633717
target	-0.328222	0.437776	-0.049643	0.517859	-0.209179	-0.719163	-0.847498	0.489109	-0.499130	0.265668	-0.617369	-0.788230	-0.633717	1.000000

ماتریس همبستگی (correlation matrix) مقدار همبستگی بین هر دو ویژگی را نشان می‌دهد.
مقدار همبستگی بین -1 تا 1 متغیر است:

- 1 یا نزدیک به 1: نشان‌دهندهٔ همبستگی مثبت قوی است (افزایش یک ویژگی باعث افزایش دیگری می‌شود).
- -1 یا نزدیک به -1: نشان‌دهندهٔ همبستگی منفی قوی است (افزایش یک ویژگی باعث کاهش دیگری می‌شود).
- 0 یا نزدیک به 0: نشان‌دهندهٔ عدم همبستگی بین دو ویژگی است.

Features with the strongest correlation: ('flavonoids', 'total_phenols')

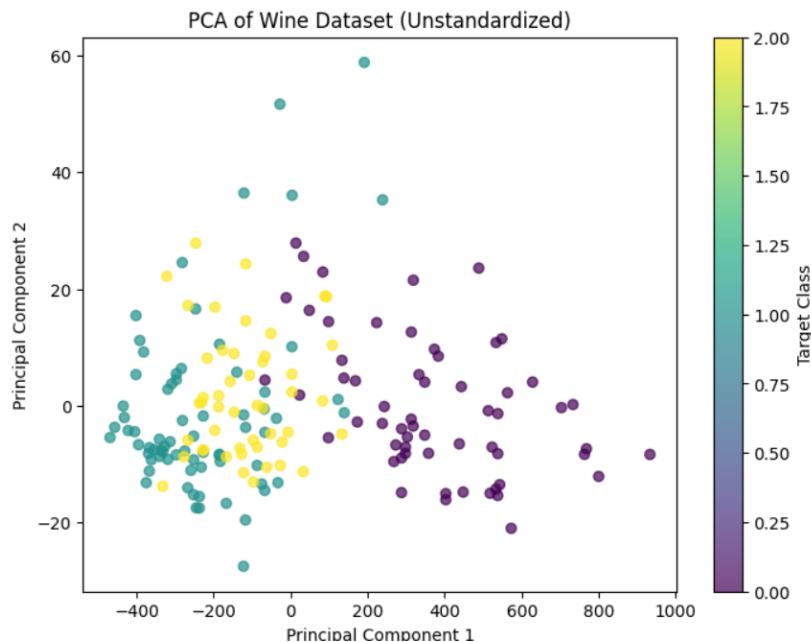
ویژگی‌هایی که قوی‌ترین همبستگی را دارند، عبارت‌اند از:
0.86 با مقدار همبستگی Flavonoids و 0.78 با مقدار همبستگی flavonoids و od280/od315_of_diluted_wines

این بدان معناست که مقدار این ویژگی‌ها به شدت به یکدیگر وابسته هستند؛ به عبارت دیگر، افزایش مقدار یکی از آن‌ها معمولاً با افزایش مقدار دیگری همراه است.

(D)

```
pca = PCA(n_components=2)
principal_components = pca.fit_transform(df.drop(columns=['target']))
pca_df = pd.DataFrame(data=principal_components, columns=['PC1', 'PC2'])
pca_df['target'] = df['target']

plt.figure(figsize=(8,6))
plt.scatter(pca_df['PC1'], pca_df['PC2'], c=pca_df['target'], cmap='viridis', alpha=0.7)
plt.xlabel('Principal Component 1')
plt.ylabel('Principal Component 2')
plt.title('PCA of Wine Dataset (Unstandardized)')
plt.colorbar(label='Target Class')
plt.show()
```



رسم نمودار PCA

در نمودار پراکندگی، دو مؤلفه اصلی PC1 و PC2 روی محورهای مختصات رسم شده‌اند. هر نقطه نشان‌دهنده یک نمونه شراب است، و رنگ نقاط بر اساس کلاس هدف (نوع شراب) مشخص شده است.

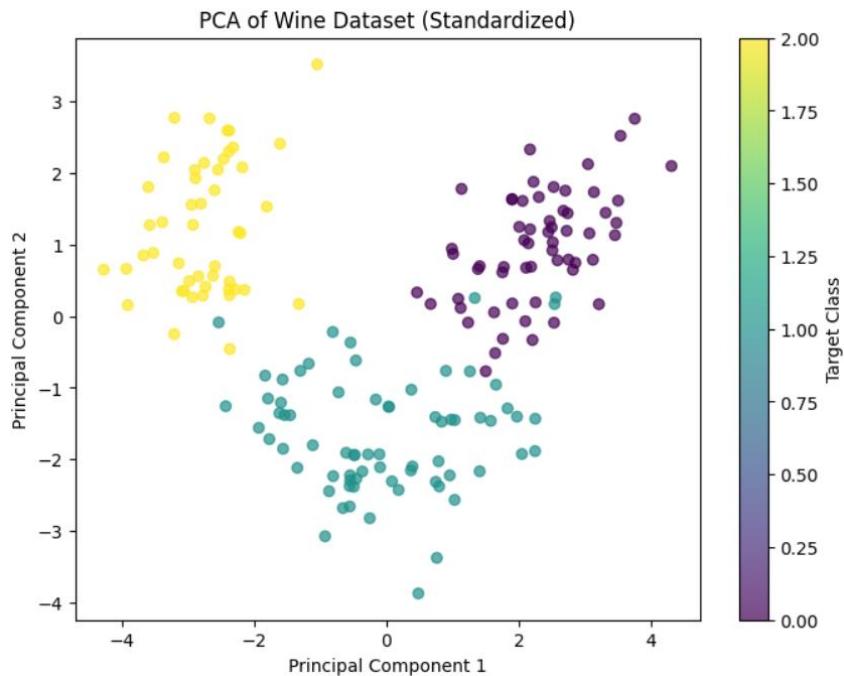
(E)

```
from sklearn.preprocessing import StandardScaler
scaler = StandardScaler()
scaled_data = scaler.fit_transform(df.drop(columns=[ 'target' ]))

pca = PCA(n_components=2)
principal_components = pca.fit_transform(scaled_data)

pca_df = pd.DataFrame(data=principal_components, columns=[ 'PC1', 'PC2' ])
pca_df['target'] = df['target']

plt.figure(figsize=(8,6))
plt.scatter(pca_df['PC1'], pca_df['PC2'], c=pca_df[ 'target' ], cmap='viridis', alpha=0.7)
plt.xlabel('Principal Component 1')
plt.ylabel('Principal Component 2')
plt.title('PCA of Wine Dataset (Standardized)')
plt.colorbar(label='Target Class')
plt.show()
```



توضیح خروجی تحلیل مؤلفه‌های اصلی (PCA) روی داده‌های استاندارد شده:

استانداردسازی داده‌ها

قبل از اجرای PCA داده‌ها استاندارد شده‌اند، به این معنی که میانگین هر ویژگی از مقدار آن کم شده و سپس بر انحراف معیار تقسیم شده است. این کار باعث می‌شود که تمامی ویژگی‌ها دارای مقیاس یکسان باشند و هیچ ویژگی‌ای تأثیر نامتناسبی روی تحلیل نداشته باشد.

اجرای PCA

روی داده‌های استاندارد شده اجرا شده و دو مؤلفه اصلی PC1 و PC2 استخراج شده‌اند.

رسم نمودار

نمودار پراکندگی نمایش‌دهنده دو مؤلفه اصلی است که هر نقطه نماینده یک نمونه شراب می‌باشد. رنگ هر نقطه بر اساس کلاس شراب (target) مشخص شده است.

مقایسه با PCA بدون استانداردسازی

در اینجا، تفکیک داده‌ها معمولاً بهتر از حالت بدون استانداردسازی است، زیرا PCA روی داده‌های با مقیاس یکسان اجرا شده است.

استانداردسازی باعث می‌شود که ویژگی‌هایی با مقادیر بزرگ‌تر، تاثیر بیشتری روی مولفه‌های اصلی نداشته باشند.