برای اینکه به بهترین شکل برای مصاحبه آماده بشید، 100 سوال و جواب کلیدی در مورد الگوریتم K-نزدیکترین همسایه (KNN) و معیارهای ارزیابی مدل در یادگیری ماشین رو براتون آماده کردم. هر جواب حدود 4 خط هست تا برای مرور سریع و پاسخگویی مختصر در مصاحبه مناسب باشه.

K-نزدیکترین همسایه (K-Nearest Neighbors - KNN): سوالات و پاسخ‌ها

مفاهیم پایه KNN

1. KNN چیست و در چه حوزه‌هایی کاربرد دارد؟
KNN یک الگوریتم یادگیری ماشین غیرپارامتریک و نمونه‌محور است که هم برای دسته‌بندی (Classification) و هم رگرسیون (Regression) کاربرد دارد. این الگوریتم با یافتن نزدیک‌ترین همسایگان یک نقطه جدید در مجموعه داده آموزشی، پیش‌بینی می‌کند. سادگی و توانایی کار با مرزهای تصمیم‌گیری غیرخطی از مزایای آن است.
2. KNN یک الگوریتم پارامتریک است یا ناپارامتریک؟ توضیح دهید.
KNN یک الگوریتم غیرپارامتریک (Non-Parametric) است. به این معنی که هیچ پارامتر ثابتی (مانند وزن‌ها در رگرسیون خطی) را در مرحله آموزش یاد نمی‌گیرد. در عوض، تمام داده‌های آموزشی را به خاطر می‌سپارد و در زمان پیش‌بینی، مستقیماً از این نمونه‌ها برای تصمیم‌گیری استفاده می‌کند.
3. چرا به KNN "نمونه‌محور" یا "مبتنی بر حافظه" می‌گویند؟
KNN نمونه‌محور (Instance-Based) یا مبتنی بر حافظه (Memory-Based) است زیرا در مرحله آموزش، فقط داده‌های آموزشی را ذخیره می‌کند و هیچ مدل صریحی نمی‌سازد. برای هر پیش‌بینی جدید، تمام مجموعه داده آموزشی را بررسی می‌کند تا نزدیک‌ترین همسایه‌ها را بیابد و بر اساس آن‌ها تصمیم بگیرد.
4. شعر "تو اول بگو با کیان زیستی، من آنگه بگویم که تو کیستی" چه ارتباطی با KNN دارد؟
این شعر به خوبی مفهوم KNN را توضیح می‌دهد. در KNN، برای تعیین کلاس یا مقدار یک نمونه جدید، مدل "نگاه می‌کند" که "دوستان نزدیک" یا "همسایگان" آن چه ویژگی‌هایی دارند. بر اساس ویژگی‌های غالب این همسایگان نزدیک است که پیش‌بینی نهایی انجام می‌شود.
5. **مراحل اصلی عملکرد KNN برای دسته‌بندی را بیان کنید.**
1. ابتدا یک عدد صحیح K (تعداد همسایگان) را انتخاب می‌کنیم.
2. برای یک نمونه جدید، K نزدیک‌ترین نمونه را از داده‌های آموزشی پیدا می‌کنیم.
3. بر اساس رأی اکثریت کلاس‌های K همسایه (Majority Vote)، کلاس نمونه جدید را پیش‌بینی می‌کنیم.
6. برای جلوگیری از تساوی در رأی‌گیری در دسته‌بندی KNN، چه توصیه‌ای در انتخاب K وجود دارد؟
معمولاً توصیه می‌شود که K را به صورت یک عدد فرد انتخاب کنیم. این کار به جلوگیری از موقعیت‌های تساوی رأی در زمان تصمیم‌گیری برای کلاس نمونه جدید کمک می‌کند، به خصوص در مسائل دسته‌بندی با دو کلاس.
7. KNN چگونه می‌تواند مرزهای تصمیم‌گیری غیرخطی ایجاد کند؟
بر خلاف مدل‌های خطی مانند رگرسیون لجستیک، KNN نیازی به تعریف یک مرز خطی ندارد. مرز تصمیم‌گیری آن به

صورت پویا و بر اساس توزیع نقاط داده در فضای ویژگی شکل می‌گیرد و می‌تواند بسیار پیچیده و غیرخطی باشد تا کلاس‌های غیرخطی تفکیک‌پذیر را جدا کند.

8. منظور از "Voronoi Tessellation" در K=1 در KNN چیست؟
هنگامی که K=1 باشد، مرزهای تصمیم‌گیری KNN به Voronoi Tessellation منجر می‌شود. در این حالت، فضا به "سلول‌های ورونوی" تقسیم می‌شود که هر سلول شامل تمام نقاطی است که به یک نقطه آموزشی خاص (و نه دیگر نقاط) نزدیک‌تر هستند. این مرزها بسیار بریده‌بریده و حساس به نویز هستند.

K و Bias-Variance Trade-off

9. تأثیر K بر مرز تصمیم‌گیری در KNN چیست؟
مقدار K تأثیر مستقیمی بر پیچیدگی مرز تصمیم‌گیری دارد. K کوچک (مثلاً 1) منجر به مرزهای پیچیده و بریده‌بریده می‌شود، در حالی که K بزرگ‌تر مرزهای هموارتر و ساده‌تری ایجاد می‌کند.

10. اگر K=1 باشد، چه مشکلاتی ممکن است در مدل KNN ایجاد شود؟
K=1 باعث می‌شود مدل به شدت مستعد بیش‌برازش (Overfitting) شود. در این حالت، مدل بیش از حد به داده‌های آموزشی خاص و حتی نویز موجود در آن‌ها حساس می‌شود و عملکرد ضعیفی روی داده‌های جدید و دیده نشده خواهد داشت. همچنین به داده‌های پرت (Outliers) بسیار حساس است.

11. افزایش K چه تأثیری بر Bias و Variance مدل KNN دارد؟
با افزایش K، مدل واریانس (Variance) کمتری پیدا می‌کند (یعنی کمتر به نویز و تغییرات کوچک در داده‌ها حساس است) و هموارتر می‌شود. اما در عین حال، بایاس (Bias) آن افزایش می‌یابد، به این معنی که ممکن است سادگی بیش از حد، الگوهای واقعی و پیچیدگی‌های موجود در داده را نادیده بگیرد و دچار کم‌برازش (Underfitting) شود.

12. مفهوم "Bias-Variance Trade-off" را در رابطه با انتخاب K در KNN توضیح دهید.
Bias-Variance Trade-off به این معنی است که نمی‌توان همزمان هم بایاس و هم واریانس مدل را به حداقل رساند. در KNN، K کوچک باعث بایاس کم و واریانس بالا (بیش‌برازش) می‌شود، در حالی که K بزرگ باعث بایاس بالا و واریانس کم (کم‌برازش) می‌گردد. هدف پیدا کردن K بهینه‌ای است که این دو را به بهترین شکل متعادل کند.

13. چگونه مقدار بهینه K را برای KNN تعیین می‌کنیم؟
مقدار بهینه K یک هایپرپارامتر (Hyperparameter) است که معمولاً با استفاده از مجموعه داده اعتبارسنجی (Validation Set) یا تکنیک‌های اعتبارسنجی متقابل (Cross-Validation) تعیین می‌شود. مدل با Kهای مختلف آموزش داده شده و عملکرد آن روی داده‌های ولیدیشن ارزیابی می‌شود تا بهترین K انتخاب شود.

معیارهای فاصله

14. چرا انتخاب معیار فاصله در KNN اهمیت دارد؟
معیار فاصله نحوه تعریف "نزدیکی" بین نقاط داده را مشخص می‌کند. انتخاب نادرست معیار فاصله می‌تواند منجر به شناسایی همسایگان نامربوط شود و در نتیجه، بر دقت و عملکرد نهایی مدل KNN تأثیر منفی بگذارد.

15. رایج‌ترین معیار فاصله در KNN چیست؟ فرمول آن را بنویسید.
فاصله اقلیدسی (Euclidean Distance) رایج‌ترین معیار است.
فرمول برای دو نقطه $x=(x1,...,xd)$ و $x′=(x1′,...,xd′)$ در فضای d بعدی:
$$d(x,x′)=(x1−x1′)2+(x2−x2′)2+⋯+(xd−xd′)2$$

16. فاصله اقلیدسی وزن‌دار (Weighted Euclidean Distance) چیست و چه مزیتی دارد؟
در فاصله اقلیدسی وزن‌دار، به هر بعد (ویژگی) یک وزن wi اختصاص داده می‌شود. این کار به ما اجازه می‌دهد تا به ویژگی‌های مهم‌تر، اهمیت بیشتری در محاسبه فاصله بدهیم. مزیت آن این است که می‌توان تأثیر ویژگی‌های مختلف را بر نزدیکی کنترل کرد.
dw(x,x′)=w1(x1−x1′)2+⋯+wd(xd−xd′)2
17. فاصله مینکوفسکی (Minkowski Distance) چیست؟ ارتباط آن با فواصل اقلیدسی و منهتن را بیان کنید.
فاصله مینکوفسکی یک فرم عمومی‌تر از فواصل اقلیدسی و منهتن است.
فرمول: d(x,x′)=(Σi=1d|xi−xi′|p)1/p
اگر p=1 باشد، به فاصله منهتن (Manhattan Distance) تبدیل می‌شود.
اگر p=2 باشد، همان فاصله اقلیدسی است.
18. فاصله منهتن (Manhattan Distance) چیست؟
فاصله منهتن که به آن فاصله بلوک‌شهری (City Block Distance) یا L1 Norm نیز گفته می‌شود، مجموع قدر مطلق تفاوت‌های مختصات دو نقطه است. این فاصله مانند مسیری است که در یک شبکه مربعی (مثل خیابان‌های منهتن) برای رفتن از یک نقطه به نقطه دیگر باید طی کرد.
19. چه زمانی از فاصله کسینوسی (Cosine Distance) استفاده می‌شود و چرا؟
فاصله کسینوسی (یا شباهت کسینوسی) بر اساس زاویه بین دو بردار عمل می‌کند و به جای اندازه بردارها، به جهت آن‌ها اهمیت می‌دهد. این معیار بیشتر در کاربردهایی مانند پردازش زبان طبیعی (NLP)، سیستم‌های توصیه‌گر و تحلیل اسناد (که جهت بردارها نشان‌دهنده محتوا است) استفاده می‌شود.
20. نرم Lp چیست و چه ارتباطی با فاصله مینکوفسکی دارد؟
نرم Lp یک بردار x به صورت ||x||p=(Σi=1d|xi|p)1/p تعریف می‌شود. فاصله مینکوفسکی در واقع همان نرم Lp بردار تفاوت (x−x′) است. نرم L1 برابر با فاصله منهتن و نرم L2 برابر با فاصله اقلیدسی است.
21. آیا مقیاس‌بندی ویژگی‌ها (Feature Scaling) در KNN مهم است؟ چرا؟
بله، بسیار مهم است. از آنجا که KNN بر اساس فاصله کار می‌کند، ویژگی‌هایی با مقیاس‌های بزرگ‌تر می‌توانند بر محاسبه فاصله سلطه پیدا کرده و تأثیر ویژگی‌های با مقیاس کوچک‌تر را نادیده بگیرند. مقیاس‌بندی (مانند نرمال‌سازی یا استانداردسازی) باعث می‌شود همه ویژگی‌ها به طور مساوی در محاسبه فاصله مشارکت کنند.

KNN برای رگرسیون

22. هدف KNN در مسائل رگرسیون چیست؟
در مسائل رگرسیون، هدف KNN پیش‌بینی یک مقدار پیوسته (عددی) برای یک نمونه جدید است. به جای پیش‌بینی کلاس با رأی اکثریت، میانگین (یا میانه) مقادیر همسایگان نزدیک را محاسبه می‌کند.
23. نحوه پیش‌بینی مقدار در رگرسیون KNN را توضیح دهید.
برای یک نمونه جدید x , ابتدا K نزدیکترین همسایه آن ((x′(1),...,x′(k)) را از داده‌های آموزشی پیدا می‌کنیم. سپس، مقدار پیش‌بینی شده y^ برای x، میانگین (یا گاهی میانه) مقادیر برچسب این K همسایه است:
(y^=k1Σj=1ky′(j
24. مشکل "ناپیوستگی" در تابع تخمین زده شده توسط رگرسیون KNN به چه معناست؟
به دلیل اینکه KNN در رگرسیون، میانگین مقادیر همسایگان را می‌گیرد، تابع پیش‌بینی شده می‌تواند در نقاطی که همسایگان تغییر می‌کنند، ناپیوستگی داشته باشد. این به این معنی است که نمودار رگرسیون ممکن است صاف نباشد و دارای جهش‌هایی باشد.
25. تأثیر K=1 در رگرسیون KNN بر روی "برازش نویز" چگونه است؟

مشابه دسته‌بندی، در رگرسیون نیز اگر K=1 باشد، مدل به شدت به نویز (Noise) حساس است. خط رگرسیون بسیار پرنوسان خواهد بود و هرگونه نوسان یا داده پرت در داده‌های آموزشی، مستقیماً در پیش‌بینی منعکس می‌شود و مدل را دچار بیش‌برازش می‌کند.

26. افزایش K در رگرسیون KNN چگونه می‌تواند منجر به "تخت کردن انتهای منحنی" شود؟
با افزایش K در رگرسیون، مدل هموارتر می‌شود و واریانس آن کاهش می‌یابد. اما اگر K خیلی بزرگ شود، مدل تمایل به کم‌برازش پیدا می‌کند. این امر می‌تواند باعث شود که در انتهای دامنه داده‌ها، منحنی پیش‌بینی شده به جای دنبال کردن روند واقعی داده‌ها، به سمت یک خط صاف میل کند و پیچیدگی‌های اصلی را از دست بدهد.

مزایا و معایب KNN

27. **دو مزیت اصلی KNN را نام ببرید.**
1. **سادگی:** درک و پیاده‌سازی آن بسیار ساده است.
2. **عدم نیاز به آموزش صریح:** فاز آموزشی ندارد، فقط داده‌ها را ذخیره می‌کند.
3. **توانایی ایجاد مرزهای تصمیم‌گیری غیرخطی:** می‌تواند الگوهای پیچیده را یاد بگیرد.

28. **دو عیب اصلی KNN را ذکر کنید.**
1. **هزینه محاسباتی بالا در زمان پیش‌بینی (پیش‌بینی کند):** برای هر نمونه جدید، باید فاصله آن را با تمام نمونه‌های آموزشی محاسبه کند.
2. **حساسیت به ابعاد بالا (Curse of Dimensionality):** در فضاهای با ابعاد زیاد، مفهوم فاصله بی‌معنی می‌شود و عملکرد آن کاهش می‌یابد.
3. **حساسیت به داده‌های نویز و پرت (Outliers):** به خصوص با K کوچک.

29. منظور از "نفرین ابعاد" (Curse of Dimensionality) در رابطه با KNN چیست؟
در فضاهای با ابعاد (تعداد ویژگی‌ها) بالا، داده‌ها بسیار پراکنده می‌شوند و مفهوم "نزدیکی" (که اساس KNN است) بی‌معنی می‌شود. به این معنی که همه نقاط ممکن است از یکدیگر "دور" به نظر برسند و پیدا کردن همسایگان واقعی دشوار گردد، که عملکرد مدل را به شدت کاهش می‌دهد.

30. چرا KNN برای مجموعه داده‌های بسیار بزرگ (Big Data) مناسب نیست؟
به دلیل هزینه محاسباتی بالا در زمان پیش‌بینی. برای هر پیش‌بینی، باید فاصله تا تمام نقاط آموزشی محاسبه شود. در مجموعه داده‌های بسیار بزرگ، این عملیات زمان‌بر و نیازمند حافظه زیادی است، که KNN را ناکارآمد می‌کند.

31. چه روش‌هایی برای بهبود کارایی KNN در مجموعه داده‌های بزرگ وجود دارد؟
استفاده از ساختارهای داده‌ای که جستجوی همسایگان نزدیک را بهینه‌سازی می‌کنند (مانند KD-Tree یا Ball Tree)، یا استفاده از روش‌های کاهش ابعاد (Dimensionality Reduction) مانند PCA قبل از اعمال KNN می‌تواند به بهبود کارایی کمک کند.

32. آیا KNN نسبت به ویژگی‌های نامربوط (Irrelevant Features) حساس است؟ چرا؟
بله، KNN به ویژگی‌های نامربوط حساس است. وجود ویژگی‌های نامربوط می‌تواند مفهوم "نزدیکی" را مخدوش کند، زیرا این ویژگی‌ها در محاسبه فاصله مشارکت می‌کنند اما اطلاعات مفیدی برای دسته‌بندی یا رگرسیون ارائه نمی‌دهند، و ممکن است باعث شوند همسایگان واقعی به درستی شناسایی نشوند.

مقایسه با سایر الگوریتم‌ها

33. تفاوت اصلی KNN با رگرسیون خطی یا لجستیک در چیست؟
رگرسیون خطی/لجستیک الگوریتم‌های پارامتریک هستند که یک مدل صریح (یک خط یا صفحه تصمیم) می‌سازند، در حالی که KNN غیرپارامتریک است و هیچ مدل صریحی نمی‌سازد، بلکه داده‌ها را حفظ می‌کند و در زمان پیش‌بینی از آن‌ها استفاده می‌کند. همچنین، مدل‌های خطی مرزهای تصمیم خطی دارند، در حالی که KNN می‌تواند مرزهای غیرخطی ایجاد کند.
34. چرا KNN برای "کارهای تشخیص الگو" که مرزهای تصمیم‌گیری پیچیده‌ای دارند، مناسب است؟
زیرا KNN ماهیت غیرپارامتریک دارد و نیازی به فرض خاصی درباره توزیع داده‌ها یا شکل مرز تصمیم‌گیری ندارد. این قابلیت به آن اجازه می‌دهد تا مرزهای تصمیم‌گیری غیرخطی و پیچیده‌ای را که با شکل واقعی کلاس‌ها در فضای ویژگی مطابقت دارند، به طور انعطاف‌پذیر یاد بگیرد.
35. آیا KNN برای داده‌های دسته‌ای (Categorical Data) مناسب است؟ اگر نه، چه راهکاری پیشنهاد می‌کنید؟
KNN به طور مستقیم برای داده‌های دسته‌ای با معیارهای فاصله استاندارد (مانند اقلیدسی) مناسب نیست، زیرا این معیارها برای داده‌های عددی طراحی شده‌اند. برای کار با داده‌های دسته‌ای، باید از معیارهای فاصله مخصوص (مثل Gower distance) استفاده کرد یا داده‌های دسته‌ای را به صورت One-Hot Encoding به عددی تبدیل کرد.

نکات پیشرفته و کاربردها

36. منظور از "تابع وزن‌دهی" در KNN چیست؟ مثالی بزنید.
تابع وزن‌دهی (Weighting Function) در KNN (معمولاً در رگرسیون و دسته‌بندی وزن‌دار) برای اختصاص وزن‌های متفاوت به همسایگان استفاده می‌شود. همسایگان نزدیک‌تر وزن بیشتری می‌گیرند، به این معنی که تأثیر بیشتری در تصمیم نهایی دارند.
مثال: وزن می‌تواند با معکوس مربع فاصله ($1/d^2$) متناسب باشد.
37. در چه سناریوهایی استفاده از KNN با وزن‌دهی همسایگان (Weighted KNN) مفید است؟
استفاده از Weighted KNN زمانی مفید است که بخواهیم تأثیر همسایگان دورتر را کاهش دهیم و به همسایگان نزدیک‌تر اهمیت بیشتری بدهیم. این کار می‌تواند حساسیت به نویز را کاهش داده و دقت مدل را بهبود بخشد، به خصوص در مواردی که داده‌های نزدیک، اطلاعات معتبرتری دارند.
38. آیا KNN می‌تواند در سیستم‌های توصیه‌گر (Recommender Systems) استفاده شود؟ چگونه؟
بله، KNN یکی از الگوریتم‌های پایه در سیستم‌های توصیه‌گر مبتنی بر محتوا (Content-Based) یا مبتنی بر همکاری (Collaborative Filtering) است. می‌توان از آن برای یافتن کاربران مشابه یا آیتم‌های مشابه بر اساس نزدیکی در فضای ویژگی‌ها (مثلاً سلیقه کاربران یا ویژگی‌های فیلم‌ها) استفاده کرد و سپس آیتم‌ها را توصیه کرد.
39. **برای مقابله با داده‌های پرت (Outliers) در KNN چه روش‌هایی وجود دارد؟**
1. **افزایش K:** با افزایش K، تأثیر یک نقطه پرت بر رأی اکثریت کاهش می‌یابد.
2. **استفاده از KNN وزن‌دار:** با دادن وزن کمتر به همسایگان دورتر (که ممکن است پرت باشند).
3. **پیش‌پردازش داده‌ها:** شناسایی و حذف یا اصلاح نقاط پرت قبل از اعمال KNN.
40. آیا KNN می‌تواند با داده‌های از دست رفته (Missing Data) کار کند؟
به طور مستقیم خیر. معیارهای فاصله مانند اقلیدسی نیاز دارند که تمام مقادیر ویژگی‌ها موجود باشند. برای کار با داده‌های از دست رفته، باید ابتدا از روش‌های درونیابی (Imputation) برای پر کردن مقادیر گمشده استفاده کرد، یا یک معیار فاصله مخصوص که می‌تواند مقادیر از دست رفته را مدیریت کند، به کار برد.

ارزیابی عملکرد مدل (Performance Metrics): سوالات و پاسخ‌ها

ماتریس درهم‌ریختگی (Confusion Matrix)

41. ماتریس درهم‌ریختگی (Confusion Matrix) چیست و چرا از آن استفاده می‌کنیم؟
ماتریس درهم‌ریختگی جدولی است که عملکرد یک مدل دسته‌بندی را با مقایسه پیش‌بینی‌های مدل با برچسب‌های واقعی نمایش می‌دهد. از آن استفاده می‌کنیم تا علاوه بر دقت کلی، نوع خطاهای مدل (مثلاً تعداد مثبت‌های کاذب یا منفی‌های کاذب) را به وضوح درک کنیم.
42. **در یک مسئله دسته‌بندی باینری، چهار جزء اصلی ماتریس درهم‌ریختگی را نام ببرید و توضیح دهید.**
    1. **True Positive (TP):** مدل به درستی کلاس مثبت را پیش‌بینی کرده است.
    2. **True Negative (TN):** مدل به درستی کلاس منفی را پیش‌بینی کرده است.
    3. **False Positive (FP):** مدل به اشتباه کلاس مثبت را پیش‌بینی کرده است (خطای نوع I).
    4. **False Negative (FN):** مدل به اشتباه کلاس منفی را پیش‌بینی کرده است (خطای نوع II).
43. **منظور از "خطای نوع I" و "خطای نوع II" در ماتریس درهم‌ریختگی چیست؟**
    - **خطای نوع I (Type I Error) = False Positive (FP):** رد کردن فرضیه صفر به اشتباه (مثبت کاذب). مثال: آژیر دزدگیر زنگ می‌زند در حالی که دزدی وجود ندارد.
    - **خطای نوع II (Type II Error) = False Negative (FN):** قبول کردن فرضیه صفر به اشتباه (منفی کاذب). مثال: دزدگیر زنگ نمی‌زند در حالی که دزدی وجود دارد.
44. **در عبارت "True Positive"، "True" به چه معناست و "Positive" به چه معناست؟**
    - **"True" یا "False"** به **درستی پیش‌بینی مدل** اشاره دارد؛ یعنی آیا پیش‌بینی مدل با واقعیت مطابقت دارد یا خیر.
    - **"Positive" یا "Negative"** به **کلاسی که مدل پیش‌بینی کرده است** (یا به کلاسی که واقعی بوده) اشاره دارد.
45. **یک مثال عملی از هر یک از TP, TN, FP, FN در سناریوی تشخیص سرطان ارائه دهید.**
    - **TP:** فرد سرطان دارد و مدل تشخیص می‌دهد سرطان دارد.
    - **TN:** فرد سرطان ندارد و مدل تشخیص می‌دهد سرطان ندارد.
    - **FP:** فرد سرطان ندارد اما مدل به اشتباه تشخیص می‌دهد سرطان دارد (تشخیص کاذب).
    - **FN:** فرد سرطان دارد اما مدل به اشتباه تشخیص می‌دهد سرطان ندارد (عدم تشخیص).

معیارهای اصلی ارزیابی

46. دقت (Accuracy) چیست؟ فرمول آن را بنویسید.
دقت، ساده‌ترین و رایج‌ترین معیار ارزیابی است.
تعریف: نسبت تعداد کل پیش‌بینی‌های صحیح (TP + TN) به کل نمونه‌ها.
فرمول: Accuracy=TotalSamplesTP+TN
47. چرا دقت (Accuracy) به تنهایی معیار مناسبی برای ارزیابی مدل در همه موارد نیست؟
دقت در مجموعه داده‌های نامتوازن (Imbalanced Datasets) گمراه‌کننده است. مدلی که همیشه کلاس اکثریت را پیش‌بینی می‌کند، می‌تواند دقت بالایی داشته باشد اما در تشخیص کلاس اقلیت (که اغلب مهم‌تر است) کاملاً شکست بخورد، مانند

مثال تشخیص سرطان.

48. بازخوانی (Recall) یا حساسیت (Sensitivity) چیست؟ فرمول آن را بنویسید.
تعریف: توانایی مدل در شناسایی صحیح موارد مثبت واقعی. یعنی از تمام مثبت‌های واقعی، چند درصد را مدل به درستی تشخیص داده است.
فرمول: Recall=Sensitivity=TP+FNTP

49. در چه سناریوهایی Recall معیار مهم‌تری است؟ مثالی بزنید.
Recall زمانی مهم‌تر است که هزینه False Negative (FN) (عدم تشخیص موارد مثبت واقعی) بسیار بالا باشد.
مثال: در تشخیص بیماری‌های جدی (مانند سرطان)، نمی‌خواهیم بیماری را از دست بدهیم (FN کم باشد)، حتی اگر منجر به FP بیشتر شود.

50. دقت (Precision) چیست؟ فرمول آن را بنویسید.
تعریف: دقت پیش‌بینی‌های مثبت مدل. یعنی از تمام دفعاتی که مدل "مثبت" پیش‌بینی کرده، چند درصدش واقعاً مثبت بوده است.
فرمول: Precision=TP+FPTP

51. در چه سناریوهایی Precision معیار مهم‌تری است؟ مثالی بزنید.
Precision زمانی مهم‌تر است که هزینه False Positive (FP) (مثبت کاذب) بالا باشد.
مثال: در سیستم‌های فیلترینگ اسپم، نمی‌خواهیم ایمیل‌های مهم را به اشتباه به عنوان اسپم (FP) فیلتر کنیم، حتی اگر منجر به FN بیشتر (برخی اسپم‌ها فیلتر نشوند) شود.

52. تفاوت اصلی بین Recall و Precision را توضیح دهید.
Recall روی پیدا کردن همه مثبت‌های واقعی تمرکز دارد (می‌خواهیم FN کم باشد)، در حالی که Precision روی صحت پیش‌بینی‌های مثبت تمرکز دارد (می‌خواهیم FP کم باشد). این دو معیار اغلب با هم در تعارض هستند؛ بهبود یکی ممکن است به بدتر شدن دیگری منجر شود.

53. ویژگی (Specificity) چیست؟ فرمول آن را بنویسید.
تعریف: توانایی مدل در شناسایی صحیح موارد منفی واقعی. یعنی از تمام منفی‌های واقعی، چند درصد را مدل به درستی تشخیص داده است.
فرمول: Specificity=TN+FPTN

54. چه ارتباطی بین Specificity و False Positive Rate (FPR) وجود دارد؟
False Positive Rate (FPR) همان نرخ مثبت کاذب است که برابر است با FPR=FP+TNFP.
بنابراین، Specificity=1−FPR. این دو معیار مکمل یکدیگر هستند.

معیارهای ترکیبی و جامع

55. F1 Score چیست و چرا از آن استفاده می‌کنیم؟
F1 Score یک معیار ترکیبی است که میانگین هارمونیک Precision و Recall را محاسبه می‌کند. از آن استفاده می‌کنیم تا به طور همزمان هم به صحت (Precision) و هم به کامل بودن (Recall) مدل اهمیت دهیم و یک معیار واحد و متعادل برای ارزیابی ارائه دهیم، به خصوص در داده‌های نامتوازن.

56. چرا در F1 Score از میانگین هارمونیک به جای میانگین حسابی استفاده می‌شود؟
میانگین هارمونیک به عملکرد ضعیف در هر یک از Precision یا Recall "تنبیه" بیشتری می‌دهد. اگر یکی از این دو مقدار خیلی پایین باشد، F1 Score هم پایین می‌آید. در حالی که میانگین حسابی ممکن است عملکرد ضعیف یک معیار را با عملکرد خوب معیار دیگر جبران کند و گمراه‌کننده باشد.

57. فرمول F1 Score را بنویسید.
F1=Precision+Recall2×Precision×Recall

58. منحنی ROC (Receiver Operating Characteristic Curve) چیست؟
منحنی ROC یک نمایش گرافیکی از عملکرد یک مدل دسته‌بندی باینری است. این منحنی، نرخ مثبت واقعی (True Positive Rate - TPR یا همان Recall) را در مقابل نرخ مثبت کاذب (False Positive Rate - FPR) در آستانه‌های مختلف طبقه‌بندی رسم می‌کند.
59. AUC (Area Under the Curve) در AUC-ROC به چه معناست؟
AUC به معنای مساحت زیر منحنی ROC است. این معیار یک عدد واحد ارائه می‌دهد که توانایی کلی مدل در تفکیک (Discrimination) بین کلاس‌های مثبت و منفی را در تمام آستانه‌های ممکن نشان می‌دهد.
60. **تفسیر مقادیر AUC را توضیح دهید (نزدیک به 0.5، نزدیک به 1).**
- **AUC = 0.5:** نشان‌دهنده عملکرد تصادفی (Random Guessing) مدل است؛ یعنی مدل هیچ توانایی در تمایز بین کلاس‌ها ندارد (مانند پرتاب سکه).
- **AUC = 1:** نشان‌دهنده یک دسته‌بند کامل و بدون نقص است که می‌تواند به طور ایده‌آل بین دو کلاس تمایز قائل شود.
- **AUC > 0.5:** نشان‌دهنده عملکرد بهتر از تصادفی است. هرچه AUC به 1 نزدیک‌تر باشد، توانایی تفکیک مدل بهتر است.
61. چرا AUC-ROC یک معیار جامع‌تر نسبت به Accuracy برای ارزیابی مدل‌های دسته‌بندی است؟
AUC-ROC عملکرد مدل را در تمام آستانه‌های طبقه‌بندی ممکن ارزیابی می‌کند، نه فقط در یک آستانه خاص. این ویژگی آن را در برابر نامتوازن بودن کلاس‌ها مقاوم می‌کند و برای مقایسه عملکرد مدل‌ها (بدون نیاز به انتخاب یک آستانه خاص) بسیار مفید است.

ارزیابی در مسائل چندکلاسه (Multi-class Classification)

62. ماتریس درهم‌ریختگی در مسائل دسته‌بندی چندکلاسه چگونه است؟
در مسائل با بیش از دو کلاس (مثلاً K کلاس)، ماتریس درهم‌ریختگی به صورت یک جدول KxK گسترش می‌یابد. ردیف‌ها نشان‌دهنده کلاس‌های واقعی و ستون‌ها نشان‌دهنده کلاس‌های پیش‌بینی شده هستند. هر سلول نشان‌دهنده تعداد نمونه‌هایی است که واقعاً در کلاس سطر بوده‌اند و مدل آن‌ها را در کلاس ستون پیش‌بینی کرده است.
63. چگونه می‌توان Precision و Recall را برای یک کلاس خاص در یک ماتریس درهم‌ریختگی چندکلاسه محاسبه کرد؟
برای محاسبه Precision و Recall برای یک کلاس خاص (i):
- TPi: تعداد نمونه‌های کلاس i که به درستی به کلاس i اختصاص داده شده‌اند (عنصر قطری Cii).
- FPi: تعداد نمونه‌هایی که از کلاس‌های دیگر به اشتباه به کلاس i اختصاص داده شده‌اند (جمع ستون i به جز Cii).
- FNi: تعداد نمونه‌هایی که واقعاً از کلاس i بوده‌اند اما به اشتباه به کلاس‌های دیگر اختصاص داده شده‌اند (جمع سطر i به جز Cii).

سپس فرمول‌های استاندارد Precision و Recall اعمال می‌شوند.
64. **تفاوت "Macro-averaging" و "Micro-averaging" در محاسبه معیارهای عملکرد (مثل F1 Score) در مسائل چندکلاسه چیست؟**
- **Macro-averaging:** معیار عملکرد (مثلاً F1 Score) برای **هر کلاس به صورت جداگانه** محاسبه می‌شود و سپس میانگین این مقادیر برای همه کلاس‌ها گرفته می‌شود. این روش به همه کلاس‌ها وزن یکسانی می‌دهد، صرف نظر از تعداد نمونه‌هایشان.
- **Micro-averaging:** ابتدا **TP، FP و FN** را برای **تمام کلاس‌ها به صورت تجمیعی** جمع‌آوری می‌کند، سپس معیار عملکرد را بر اساس این مقادیر تجمیع شده محاسبه می‌کند. این روش بیشتر تحت تأثیر عملکرد مدل روی کلاس‌های پرتکرار (اکثریت) قرار می‌گیرد.
65. **چه زمانی استفاده از Macro-averaging و چه زمانی استفاده از Micro-averaging توصیه می‌شود؟**

- **Macro-averaging** زمانی توصیه می‌شود که تعداد نمونه‌ها در کلاس‌ها نامتوازن باشد و شما بخواهید عملکرد مدل را روی **کلاس‌های اقلیت** نیز به خوبی ارزیابی کنید، زیرا به همه کلاس‌ها اهمیت یکسانی می‌دهد.
- **Micro-averaging** زمانی توصیه می‌شود که کلاس‌ها نسبتاً متعادل هستند، یا زمانی که عملکرد کلی و تجمیعی مدل (صرف نظر از توزیع کلاس‌ها) برای شما مهم است.

نکات تکمیلی ارزیابی

66. چه معیارهایی علاوه بر Precision، Recall و F1 Score برای ارزیابی مدل‌های دسته‌بندی باینری وجود دارد؟
علاوه بر این‌ها، می‌توان به Youden's J Index (که از Sensitivity و Specificity مشتق می‌شود)، Matthews Correlation Coefficient (MCC) که برای داده‌های نامتوازن مناسب است، و Area Under the Precision-Recall Curve (AUPRC) که به ویژه برای کلاس‌های اقلیت بسیار کوچک مفید است، اشاره کرد.
67. برای ارزیابی مدل‌های رگرسیون (که خروجی پیوسته دارند) از چه معیارهایی استفاده می‌شود؟
معیارهای رایج برای ارزیابی مدل‌های رگرسیون شامل:
   - **Mean Absolute Error (MAE):** میانگین قدر مطلق خطاها.
   - **Mean Squared Error (MSE):** میانگین مربعات خطاها (به خطاهای بزرگتر جریمه بیشتری می‌دهد).
   - **Root Mean Squared Error (RMSE):** ریشه دوم MSE (در همان واحد خروجی است).
   - **R-squared (R2):** ضریب تعیین (توضیح می‌دهد که مدل چقدر از واریانس متغیر وابسته را توضیح می‌دهد).
68. چرا RMSE نسبت به MAE در برخی موارد ترجیح داده می‌شود؟
RMSE به خطاهای بزرگتر جریمه (Penalty) بیشتری می‌دهد زیرا خطاها را مربع می‌کند. این باعث می‌شود که RMSE نسبت به داده‌های پرت (Outliers) حساس‌تر باشد. بنابراین، اگر خطاهای بزرگ برای شما اهمیت بیشتری دارند، RMSE انتخاب بهتری است. MAE جریمه یکسانی برای همه خطاها اعمال می‌کند.
69. چه زمانی استفاده از R-squared (R2) در رگرسیون می‌تواند گمراه‌کننده باشد؟
R2 می‌تواند زمانی گمراه‌کننده باشد که مدل بیش‌برازش شده باشد. افزودن ویژگی‌های بیشتر (حتی نامربوط) به مدل همیشه R2 را افزایش می‌دهد، حتی اگر مدل به واقع عملکرد بهتری روی داده‌های جدید نداشته باشد. برای رفع این مشکل، از Adjusted R2 استفاده می‌شود.
70. منظور از "آستانه طبقه‌بندی (Classification Threshold)" در مدل‌های دسته‌بندی باینری چیست؟
بسیاری از مدل‌های دسته‌بندی (مانند رگرسیون لجستیک یا SVM) یک مقدار احتمال یا امتیاز خروجی می‌دهند. آستانه طبقه‌بندی یک مقدار مرزی (معمولاً 0.5) است که تعیین می‌کند آیا یک نمونه به عنوان مثبت یا منفی طبقه‌بندی شود. تغییر این آستانه بر Precision، Recall و FPR تأثیر می‌گذارد.
71. چه رابطه‌ای بین ROC Curve و تغییر آستانه طبقه‌بندی وجود دارد؟
ROC Curve با تغییر آستانه طبقه‌بندی رسم می‌شود. هر نقطه روی منحنی ROC نشان‌دهنده یک جفت (FPR, TPR) برای یک آستانه خاص است. با جابجایی آستانه از 0 تا 1، می‌توانیم تمام نقاط ممکن (FPR, TPR) را پوشش دهیم و منحنی ROC را رسم کنیم.
72. Precision-Recall Curve (PR Curve) چیست و چه زمانی از آن استفاده می‌شود؟
PR Curve، Precision را در مقابل Recall در آستانه‌های مختلف طبقه‌بندی رسم می‌کند. این منحنی به ویژه برای مجموعه داده‌های نامتوازن (Imbalanced Datasets)، به خصوص زمانی که کلاس مثبت (کلاس اقلیت) بسیار کوچک است، مفیدتر از ROC Curve است، زیرا به طور مستقیم بر عملکرد کلاس مثبت تمرکز می‌کند.
73. Area Under the Precision-Recall Curve (AUPRC) چیست و چه مزیتی دارد؟
AUPRC مساحت زیر منحنی Precision-Recall است. این معیار برای ارزیابی مدل در مجموعه داده‌های نامتوازن، جایی که تمرکز بر عملکرد روی کلاس مثبت (اقلیت) است، بسیار مناسب است. AUPRC بالاتر نشان‌دهنده مدل بهتر است که هم

Precision و هم Recall خوبی دارد.

74. مفهوم "Class Imbalance" (عدم تعادل کلاس) در داده‌ها را توضیح دهید و چرا بر ارزیابی مدل تأثیر می‌گذارد.
Class Imbalance به وضعیتی گفته می‌شود که تعداد نمونه‌های یک کلاس (کلاس اکثریت) در مجموعه داده بسیار بیشتر از تعداد نمونه‌های کلاس دیگر (کلاس اقلیت) باشد. این موضوع بر ارزیابی مدل تأثیر می‌گذارد زیرا معیارهایی مانند Accuracy می‌توانند گمراه‌کننده باشند و مدل ممکن است عملکرد ضعیفی روی کلاس اقلیت (که معمولاً مهم‌تر است) داشته باشد.

75. **برای مقابله با Class Imbalance چه راهکارهایی در مرحله پیش‌پردازش داده‌ها وجود دارد؟**
- **Oversampling (افزایش نمونه‌های کلاس اقلیت):** مانند SMOTE.
- **Undersampling (کاهش نمونه‌های کلاس اکثریت):** مانند Random Undersampling.
- **ترکیب Oversampling و Undersampling.**
- **استفاده از وزن‌دهی کلاس (Class Weighting)** در الگوریتم‌های یادگیری ماشین.

سوالات مفهومی و مقایسه‌ای

76. چرا KNN نیازی به فاز "آموزش" به معنای سنتی (یادگیری پارامترها) ندارد؟
زیرا KNN یک الگوریتم مبتنی بر نمونه است. در فاز "آموزش"، به سادگی تمام داده‌های آموزشی را به خاطر می‌سپارد. هیچ تابع یا پارامتری برای بهینه‌سازی یا یادگیری وجود ندارد. تمام محاسبات و تصمیم‌گیری‌ها در زمان پیش‌بینی یک نمونه جدید انجام می‌شود.

77. چگونه انتخاب K بهینه بر عملکرد مدل بر روی داده‌های ولیدیشن و تست تأثیر می‌گذارد؟
انتخاب K بهینه (که معمولاً بر اساس بهترین عملکرد روی داده‌های ولیدیشن انجام می‌شود) منجر به مدلی می‌شود که تعادل مناسبی بین بایاس و واریانس دارد. این K باید عملکرد کلی خوبی را هم بر روی داده‌های ولیدیشن و هم بر روی داده‌های تست (جدید و دیده نشده) نشان دهد.

78. آیا KNN برای داده‌های با ابعاد بسیار بالا (High-Dimensional Data) مناسب است؟ چرا؟
به طور کلی خیر، به دلیل "نفرین ابعاد". در ابعاد بالا، فاصله بین نقاط به طور یکنواخت بزرگ می‌شود و تمایز بین "نزدیک" و "دور" از بین می‌رود، که باعث می‌شود جستجوی همسایگان واقعی بی‌اثر شود و عملکرد KNN به شدت کاهش یابد.

79. برای مقابله با "نفرین ابعاد" در KNN چه روش‌هایی پیشنهاد می‌کنید؟
استفاده از روش‌های کاهش ابعاد (Dimensionality Reduction) مانند:
- **PCA (Principal Component Analysis)**
- **LDA (Linear Discriminant Analysis)**
- یا انتخاب ویژگی (Feature Selection) برای حذف ویژگی‌های نامربوط یا افزونگی.

80. **چه تفاوتی بین "نرمال‌سازی" و "استانداردسازی" در پیش‌پردازش داده‌ها وجود دارد و کدام یک برای KNN مناسب‌تر است؟**
- **نرمال‌سازی (Normalization - Min-Max Scaling):** مقادیر را به یک محدوده ثابت (معمولاً 0 تا 1) مقیاس می‌کند.
- استانداردسازی (Standardization - Z-score Normalization): داده‌ها را به میانگین 0 و انحراف معیار 1 تبدیل می‌کند.
برای KNN، استانداردسازی معمولاً ترجیح داده می‌شود، زیرا به داده‌های پرت حساسیت کمتری دارد و شکل توزیع داده‌ها را حفظ می‌کند.

81. آیا KNN می‌تواند با انواع مختلف داده‌ها (عددی، دسته‌ای، متنی) کار کند؟ توضیح دهید.
KNN به طور ذاتی برای داده‌های عددی مناسب است. برای داده‌های دسته‌ای نیاز به تبدیل به فرم عددی (مانند One-Hot

Encoding) یا استفاده از معیارهای فاصله خاص دارد. برای داده‌های متنی، ابتدا باید متن‌ها را به بردارهای عددی (مثلاً با TF-IDF یا Word Embeddings) تبدیل کرد.

82. **KNN در مقایسه با درخت تصمیم (Decision Tree) چه مزایا و معایبی دارد؟**
    - **مزایای KNN:** سادگی، توانایی مدل‌سازی مرزهای غیرخطی پیچیده.
    - **معایب KNN:** کندی در پیش‌بینی (برای داده‌های بزرگ)، حساسیت به ابعاد بالا، حساسیت به نویز.
    - **مزایای درخت تصمیم:** سریع در پیش‌بینی، تفسیرپذیری بالا، مدیریت ویژگی‌های دسته‌ای و گمشده.
    - **معایب درخت تصمیم:** مستعد بیش‌برازش (بدون هرس)، ایجاد مرزهای تصمیم‌گیری خطی (مربعی) در فضای ویژگی.
83. **KNN در مقایسه با SVM (Support Vector Machine) چه تفاوت‌های کلیدی دارد؟**
    - **KNN:** الگوریتم نمونه‌محور، غیرپارامتریک، کند در زمان پیش‌بینی، حساس به ابعاد بالا.
    - **SVM:** الگوریتم پارامتریک، مدل صریح می‌سازد، سریع در زمان پیش‌بینی، برای ابعاد بالا با استفاده از Kernel Trick می‌تواند خوب عمل کند. SVM به دنبال یافتن یک ابرصفحه بهینه برای تفکیک کلاس‌هاست.
84. آیا می‌توان از KNN برای مسائل "تشخیص ناهنجاری (Anomaly Detection)" استفاده کرد؟ چگونه؟
بله، می‌توان از KNN برای تشخیص ناهنجاری استفاده کرد. نقاطی که دارای فاصله زیادی از K نزدیکترین همسایه خود هستند (یعنی چگالی پایینی دارند) می‌توانند به عنوان ناهنجاری (Outlier) در نظر گرفته شوند. این روش به "K-Nearest Neighbors Anomaly Detection" معروف است.
85. چگونه "اعتبارسنجی متقابل (Cross-Validation)" می‌تواند در انتخاب K برای KNN کمک کند؟
Cross-Validation (مانند K-Fold Cross-Validation) به ما کمک می‌کند تا عملکرد مدل را برای مقادیر مختلف K به طور robust ارزیابی کنیم. داده‌ها به چند Fold تقسیم می‌شوند و مدل با Kهای مختلف روی بخش‌های مختلف داده آموزش و اعتبارسنجی می‌شود، تا Kی انتخاب شود که بهترین عملکرد میانگین را در Foldها داشته باشد و از بیش‌برازش روی یک مجموعه ولیدیشن خاص جلوگیری شود.
86. آیا KNN نسبت به توزیع داده‌ها (Data Distribution) حساس است؟
KNN فرض خاصی درباره توزیع داده‌ها (مانند نرمال بودن) ندارد، که یک مزیت است (به همین دلیل غیرپارامتریک است). با این حال، به دلیل اینکه بر پایه فاصله کار می‌کند، چگالی داده‌ها در مناطق مختلف فضا می‌تواند بر عملکرد آن تأثیر بگذارد. مناطق کم‌تراکم ممکن است باعث خطای بیشتر شوند.
87. اگر داده‌های آموزشی و تست دارای توزیع متفاوتی باشند، چه تأثیری بر عملکرد KNN خواهد داشت؟
اگر توزیع داده‌های آموزشی و تست متفاوت باشد، عملکرد KNN به شدت کاهش می‌یابد. زیرا KNN به شدت به نزدیکی و شباهت بین نمونه‌ها وابسته است. اگر نمونه‌های تست از فضای ویژگی‌ای باشند که در داده‌های آموزشی پوشش داده نشده، همسایگان مناسبی پیدا نخواهند شد.
88. آیا KNN می‌تواند با "ویژگی‌های مختلط (Mixed Features)" (عددی و دسته‌ای) به طور همزمان کار کند؟
به طور مستقیم خیر. نیاز به تبدیل داده‌های دسته‌ای به فرم عددی (مانند One-Hot Encoding) است. همچنین می‌توان از معیارهای فاصله مخصوص که می‌توانند همزمان با ویژگی‌های عددی و دسته‌ای کار کنند (مانند Gower distance)، استفاده کرد.
89. مفهوم "Lazy Learner" در رابطه با KNN به چه معناست؟
KNN یک "Lazy Learner" (یادگیرنده تنبل) است زیرا هیچ فاز آموزشی صریحی ندارد و هیچ مدل تعمیم‌یافته‌ای (Generalized Model) نمی‌سازد. تمام "یادگیری" و محاسبات، تا زمان دریافت یک نمونه جدید برای پیش‌بینی به تعویق می‌افتد.
90. نقطه مقابل "Lazy Learner" چیست؟ مثالی بزنید.
نقطه مقابل "Lazy Learner" یک "Eager Learner" (یادگیرنده مشتاق) است. یادگیرنده‌های مشتاق، در مرحله آموزش یک مدل صریح و تعمیم‌یافته از داده‌ها می‌سازند و سپس از این مدل برای پیش‌بینی استفاده می‌کنند. مثال‌ها: رگرسیون خطی، درخت تصمیم، شبکه‌های عصبی.

سناریوهای عملی و مشکلات

91. در چه سناریوی عملی، KNN بهترین انتخاب برای مدل‌سازی است؟
KNN برای مجموعه داده‌های کوچک تا متوسط با ابعاد کم، جایی که مرزهای تصمیم‌گیری پیچیده و غیرخطی هستند، یا زمانی که سرعت آموزش (نه پیش‌بینی) مهم است، یک انتخاب مناسب است. همچنین در سیستم‌های توصیه‌گر با داده‌های چگال (Dense Data) می‌تواند مفید باشد.
92. **اگر مدل KNN من دچار "بیش‌برازش" شده باشد، چه اقداماتی برای رفع آن پیشنهاد می‌کنید؟**
1. **افزایش K:** بزرگتر کردن تعداد همسایگان.
2. **کاهش ابعاد (Dimensionality Reduction):** حذف ویژگی‌های نامربوط یا استفاده از PCA.
3. **حذف نویز و داده‌های پرت (Outlier Removal):** پیش‌پردازش داده‌ها.
4. **جمع‌آوری داده‌های آموزشی بیشتر.**
93. **اگر مدل KNN من دچار "کم‌برازش" شده باشد، چه اقداماتی برای رفع آن پیشنهاد می‌کنید؟**
1. **کاهش K:** کوچکتر کردن تعداد همسایگان.
2. **افزودن ویژگی‌های مرتبط (Feature Engineering):** ساخت ویژگی‌های جدید که اطلاعات بیشتری دارند.
3. **انتخاب معیار فاصله مناسب‌تر:** که بتواند الگوهای واقعی‌تر را در داده‌ها شناسایی کند.
94. آیا KNN به مقیاس ویژگی‌ها حساس است؟ اگر بله، چگونه می‌توان این مشکل را حل کرد؟
بله، KNN به مقیاس ویژگی‌ها بسیار حساس است. ویژگی‌هایی با مقادیر عددی بزرگتر، تأثیر نامتناسبی بر محاسبه فاصله خواهند داشت. این مشکل با مقیاس‌بندی ویژگی‌ها (Feature Scaling)، مانند نرمال‌سازی (Normalization) یا استانداردسازی (Standardization)، حل می‌شود.
95. در یک مسئله دسته‌بندی، اگر هزینه False Positive بسیار بیشتر از False Negative باشد، کدام معیار ارزیابی مهم‌تر است و چرا؟
اگر هزینه FP (مثبت کاذب) بیشتر باشد، Precision معیار مهم‌تری است. زیرا Precision بر صحت پیش‌بینی‌های مثبت تمرکز دارد و سعی می‌کند تعداد FPها را کاهش دهد. (مثلاً در فیلتر اسپم، نمی‌خواهیم ایمیل مهمی را به اشتباه اسپم تلقی کنیم).
96. در یک مسئله دسته‌بندی، اگر هزینه False Negative بسیار بیشتر از False Positive باشد، کدام معیار ارزیابی مهم‌تر است و چرا؟
اگر هزینه FN (منفی کاذب) بیشتر باشد، Recall معیار مهم‌تری است. زیرا Recall بر یافتن تمام موارد مثبت واقعی تمرکز دارد و سعی می‌کند تعداد FNها را کاهش دهد. (مثلاً در تشخیص بیماری، نمی‌خواهیم یک بیمار را به اشتباه سالم تشخیص دهیم).
97. **چگونه می‌توانید در کد پایتون، K-Nearest Neighbors را پیاده‌سازی کنید؟ (بدون جزئیات کد، فقط مراحل کلی)**
1. داده‌ها را بارگذاری و پیش‌پردازش کنید (شامل مقیاس‌بندی).
2. داده‌ها را به مجموعه آموزش و تست تقسیم کنید.
3. مدل KNeighborsClassifier یا KNeighborsRegressor از sklearn.neighbors را ایجاد کنید و n_neighbors (همان K) را تنظیم کنید.
4. مدل را روی داده‌های آموزش fit کنید (در واقع فقط داده‌ها را ذخیره می‌کند).
5. روی داده‌های تست predict کنید.
6. عملکرد مدل را با معیارهای مناسب (accuracy, f1-score, MSE و ...) ارزیابی کنید.
98. چه زمانی استفاده از KNeighborsRegressor برای رگرسیون مناسب است؟
KNeighborsRegressor زمانی مناسب است که رابطه بین ویژگی‌ها و متغیر وابسته پیچیده و غیرخطی باشد، یا زمانی که اندازه مجموعه داده نسبتاً کوچک تا متوسط است. همچنین زمانی که نیازی به مدل‌سازی صریح رابطه نیست و میانگین همسایگان پیش‌بینی خوبی ارائه می‌دهد.
99. آیا وزن‌دهی به همسایگان بر اساس فاصله در رگرسیون KNN می‌تواند به بهبود عملکرد کمک کند؟ چگونه؟

بله، می‌تواند کمک کند. در رگرسیون KNN با وزن‌دهی، به جای گرفتن میانگین ساده، یک میانگین وزن‌دار از مقادیر برچسب همسایگان گرفته می‌شود. همسایگان نزدیک‌تر وزن بیشتری دارند و تأثیر بیشتری بر پیش‌بینی نهایی می‌گذارند، که می‌تواند دقت مدل را بهبود بخشد و حساسیت به نویز را کاهش دهد.

100. به عنوان یک یادگیرنده تنبل، KNN چه تأثیری بر زمان آموزش و زمان پیش‌بینی دارد؟

KNN دارای زمان آموزش بسیار سریع (یا تقریباً صفر) است، زیرا فقط داده‌ها را ذخیره می‌کند. اما دارای زمان پیش‌بینی کند است، زیرا برای هر نمونه جدید، باید فاصله آن را با تمام نمونه‌های آموزشی محاسبه کرده و نزدیک‌ترین همسایگان را پیدا کند. این برخلاف یادگیرنده‌های مشتاق است که زمان آموزش طولانی‌تری دارند اما زمان پیش‌بینی بسیار سریع‌تری دارند.

موفق باشید در مصاحبه!