

بنام خدا

تمرین سری چهارم یادگیری ماشین

مریم رضوانی ۹۹۲۱۱۶۰۰۱۹

سؤال ۱) صحیح و غلط؟؟

الف) کوچک بودن مقدار k باعث پیچیده‌تر شدن مرز تصمیم نمی‌شود و با افزایش مقدار k مرز کلاس پیچیده‌تر می‌شود. غلط؛ برای مثال $k=2$ باعث می‌شود داده را به ۲ قسمت تقسیم کنیم و مدل را ۲ بار آموزش و ارزیابی می‌کنیم. و مقدار داده‌های تست بیشتر و داده‌های آموزشی کمتر می‌شود. این باعث $overfit$ مدل می‌شود. همچنین مرز تصمیم آن پیچیده‌تر و ناهموار تر می‌شود. و اگر مقدار k را بزرگ در نظر بگیریم، در این حالت داده‌های آموزش بیشتر و داده‌های تست کمتر می‌شود که این باعث $underfitting$ می‌شود و مرز تصمیم آن ساده‌تر و صاف شود. بنابراین مقدار k تاثیر مهمی در مرز تصمیم مدل دارد. که عموماً $k=10$ بعنوان پیش فرض در نظر می‌گیرند.

ب) در کلاسیفایر k -nearest neighbors مقدار k از ۱ تا ۵ باعث بیش‌برازش می‌شود. درست؛ اگر k را کم در نظر بگیریم باعث بیش‌برازش می‌شود. مقدار k کوچک ($k=2$) به این معنی است که فقط از یک یا دو همسایگی کناریش برای تعیین کلاس استفاده می‌کند این باعث می‌شود مدل حساس به نویز داده‌ها شود و باعث $overfitting$ شود.

سؤال ۲) توضیح دهید که:

الف) چرا توصیه می‌شود از الگوریتم برای داده‌ها با ویژگی بالا استفاده نشود؟
الگوریتم k -nearest neighbors این الگوریتم برای داده‌ها با ویژگی بالا چالش‌هایی دارد که باعث می‌شود عملکرد آن کاهش یابد. برخی از این چالش‌ها عبارتند از:

- ◆ نفرین بعد: این اصطلاح به این معنی است که هر چه تعداد ویژگی‌ها بیشتر شود، فاصله‌های بین داده‌ها به هم نزدیک تر می‌شوند و تفاوت بین نزدیکترین همسایه و دورترین همسایه کمتر می‌شود. این باعث می‌شود که مفهوم نزدیکی از دست برود و الگوریتم نتواند داده‌ها را به درستی طبقه‌بندی یا رگرسیون کند.
- ◆ بیش‌برازش: این اصطلاح به این معنی است که مدل به داده‌های آموزشی بیش از حد سازگار شده و نتواند به داده‌های جدید خوب تعمیم پیدا کند. این مشکل ممکن است در الگوریتم k -nearest neighbors رخ دهد اگر مقدار k را کوچک انتخاب کنیم یا اگر بین ویژگی‌ها همبستگی وجود داشته باشد.
- ◆ پیچیدگی محاسباتی: این اصطلاح به این معنی است که مدل نیاز به زمان و حافظه زیادی برای اجرا دارد. این مشکل در الگوریتم k -nearest neighbors وجود دارد چون برای هر داده جدید، باید فاصله‌اش را با تمام داده‌های آموزشی محاسبه کنیم و k نزدیکترین همسایه‌اش را پیدا کنیم. این کار با افزایش تعداد ویژگی‌ها و داده‌ها، بسیار زمان‌بر و هزینه‌بر می‌شود.

ب) چگونه الگوریتم k -nearest neighbors با موازنه $bias - variance$ مرتبط شود؟

مقدار k تأثیر مهمی در موازنه $bias - variance$ دارد. اگر مقدار k را کوچک انتخاب کنیم، مثلاً $k=1$ یا $k=2$ ، به این معنی است که مدل فقط از یک یا دو همسایه‌ی نزدیک برای تعیین کلاس یا مقدار استفاده می‌کند. در این حالت، مدل

دارای variance بالا و bias کم است. این به این معنی است که مدل به داده‌های آموزشی بیش از حد سازگار شده و نتواند به داده‌های جدید خوب تعمیم پیدا کند. این مشکل را بیشبرازش (overfitting) می‌گویند.

(ج) کدام یک از معیارهای فاصله زیر نمی‌تواند در k-nearest neighbors استفاده شود؟

- (a) منهتن (b) ماهالونویس (c) جاکارد (d) تشابه کسینوسی (e) همه قابل استفاده اند.
- فاصله شهری (منهتن): این معیار فاصله بین دو نقطه را بر اساس مجموع مقادیر مطلق تفاوت بعدهای آنها محاسبه می‌کند. این معیار برای داده‌هایی که توزیع نرمال ندارند و دارای نویز هستند مناسب است.
 - فاصله ماهالانویس: این معیار فاصله بین دو نقطه را با در نظر گرفتن توزیع داده ها و واریانس بعدهای آنها محاسبه می‌کند. این معیار برای داده‌هایی که توزیع نرمال دارند و دارای وابستگی بین بعدها هستند مناسب است.
 - معیار ژاکارد می‌تواند در الگوریتم k-nearest neighbors برای محاسبه فاصله بین داده‌ها استفاده شود. این معیار برای داده‌هایی که دارای بعدهای دودویی یا نمادین هستند مناسب است. برای مثال، اگر داده‌ها به صورت بردارهای دودویی باشند که نشان دهنده وجود یا عدم وجود یک ویژگی در هر داده هستند، می‌توان از معیار ژاکارد برای محاسبه شباهت بین این بردارها استفاده کرد. این معیار شباهت بین دو مجموعه را بر اساس تقسیم اندازه اشتراک آنها بر اندازه اجتماع آنها می‌سنجد. این معیار در مواردی که داده‌ها به صورت مجموعه‌ای از عناصر هستند مانند کلمات یا برچسب‌ها مورد استفاده قرار می‌گیرد.
 - تشابه کسینوسی: این معیار شباهت بین دو بردار را بر اساس زاویه بین آنها می‌سنجد. این معیار در مواردی که اندازه بردارها مهم نیست و فقط جهت آنها مطلوب است کاربرد دارد.
- پاسخ صحیح گزینه (e) همه قابل استفاده اند است. زیرا همه معیارهای فاصله ذکر شده می‌توانند در k-nearest neighbors استفاده شوند.

(د) در یک مثال و شکل توضیح دهید که چگونه می‌توان برای پیش‌بینی عددی از الگوریتم k-nearest neighbors استفاده کرد؟

برای مثال، فرض کنید مجموعه داده‌ای از دمای هوا در یک شهر در طول یک سال را داشته باشیم. می‌خواهیم از این داده‌ها برای پیش‌بینی دمای هوا در روز آینده استفاده کنیم.

اولین قدم، انتخاب k است. k تعداد نزدیکترین همسایه‌هایی است که برای پیش‌بینی استفاده می‌شوند. k معمولاً یک عدد کوچک مانند 3 یا 5 انتخاب می‌شود.

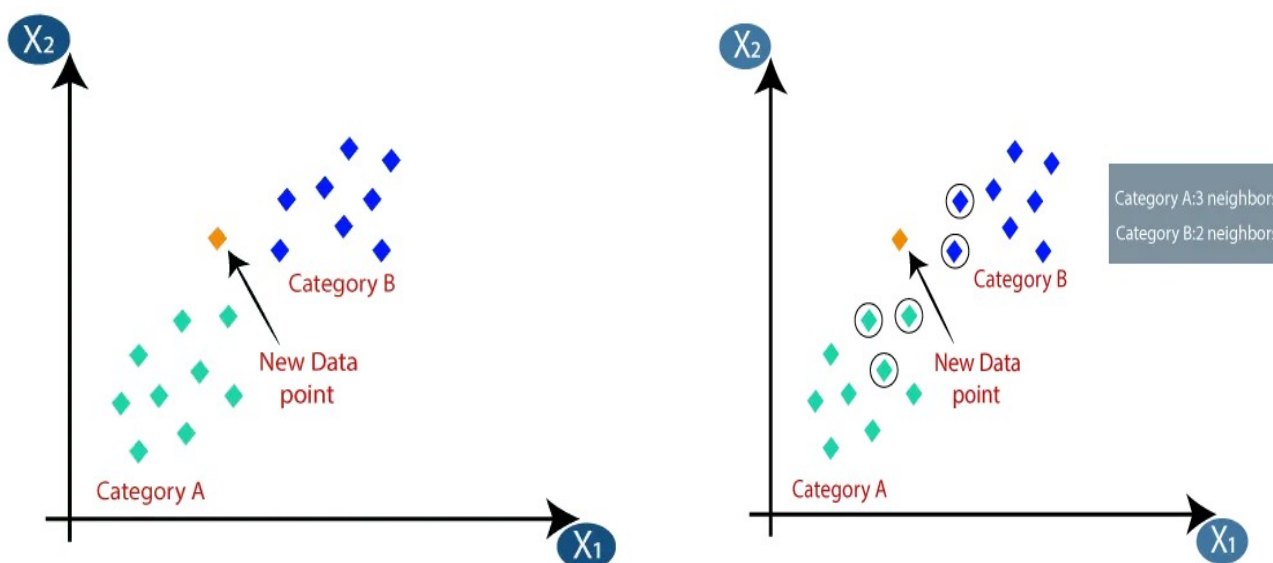
در مرحله بعد، باید فاصله بین هر نقطه داده و نقطه داده جدید را محاسبه کنیم. در این مثال، از فاصله اقلیدسی استفاده خواهیم کرد.

در نهایت، باید k نقطه داده با فاصله کم‌ترین را انتخاب کنیم. سپس، می‌توانیم دمای هوا را با استفاده از میانگین دمای هوا برای این k نقطه داده پیش‌بینی کنیم.

در این مثال، فرض کنید نقطه داده جدید با دمای 20 درجه سانتیگراد است. فاصله بین این نقطه داده و k نقطه داده نزدیک‌ترین آن 1 درجه سانتیگراد است. دمای هوا برای این 4 نقطه داده به ترتیب 20، 21، 22 و 23 درجه سانتیگراد است. بنابراین، میانگین دمای هوا برای این k نقطه داده 21.5 درجه سانتیگراد است. بنابراین، پیش‌بینی ما برای دمای هوا در روز آینده 21.5 درجه سانتیگراد است.

مثال دیگر:

تصور کنید که می خواهید قیمت یک خانه را بر اساس اندازه (مساحت مربع) و مکان (کد پستی) آن پیش بینی کنید. شما یک مجموعه داده حاوی اطلاعاتی در مورد چندین خانه، از جمله اندازه، مکان و قیمت فروش آنها دارید. این مجموعه داده های آموزشی شماست. برای پیش بینی قیمت یک خانه جدید: فاصله بین خانه جدید (که با اندازه و مکان آن نشان داده می شود) و همه خانه ها در داده های آموزشی را محاسبه کنید. شما می توانید از معیارهای فاصله مختلف مانند فاصله اقلیدسی یا فاصله منتهن استفاده کنید. یک مقدار برای k (تعداد نزدیکترین همسایگان) انتخاب کنید. این یک پارامتر مهم است که بر دقت پیش بینی تأثیر می گذارد. خانه های k را در داده های آموزشی پیدا کنید که به خانه جدید نزدیک هستند. میانگین قیمت فروش k نزدیکترین همسایه. این میانگین پیش بینی شما برای قیمت خانه جدید است.



الگوریتم نزدیکترین همسایگان در شکل نارنجی نمایانگر خانه جدید و دایره های مشکی نشان دهنده k نزدیکترین همسایه است. اندازه هر دایره نشان دهنده قیمت فروش خانه است. قیمت پیش بینی شده خانه نوساز میانگین قیمت های فروش دایره های مشکی است.

سؤال ۳) خطای اعتبار سنجی leave one out classifier برای دو دسته بند NN-1 و NN-3 برای داده های زیر چقدر است؟

×	—	×	—	—	—
×		×	—	—	—

NN-1:

هرگاه یکی از نقاط خوشه سمت راست را بیرون بیاوریم، به درستی طبقه‌بندی می‌شود: یک - را نگه داشتیم، نزدیک‌ترین نقطه به نقطه باز شده نیز - است

در خوشه سمت چپ، ما همیشه نقطه باز شده را به اشتباه طبقه‌بندی می‌کنیم. اگر نقطه بیرون نگه داشته شده یکی از +ها در مربع باشد، بسته شدن نقطه باقیمانده سمت راست یک + است، بنابراین طبقه‌بندی ما درست است. اگر نقطه بیرون نگه داشته شده - در وسط باشد، نزدیکترین نقطه با یک + است، بنابراین در این مورد نادرست هستیم.

در مجموع، ما 1 مورد از 10 احتمال را به اشتباه طبقه‌بندی کردیم.

$$\text{Error cross-validation}=1/10$$

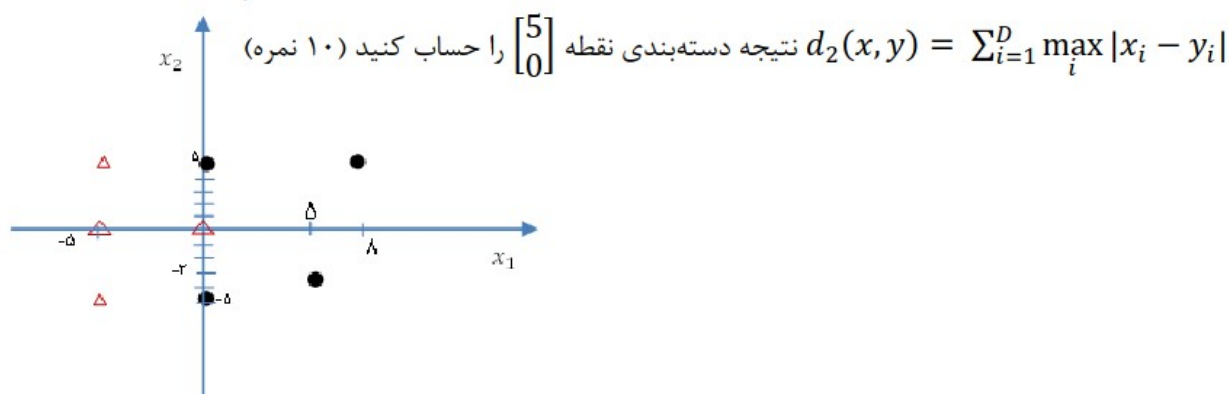
NN-3:

اکنون ما هنوز یک نقطه داده را نگه می‌داریم، اما به سه نزدیک‌تر باقیمانده اجازه می‌دهیم اکثریت رای خود را در مورد نحوه طبقه‌بندی نقطه نگه‌داشته‌شده رأی دهند. خوشه سمت راست کم و بیش یکسان است. در خوشه سمت چپ، اگر هر یک از +ها را در اطراف لبه‌ها نگه داریم، نزدیک‌ترین نقاط باقی مانده یک +، - و + هستند، بنابراین اکثریت رای یک + است، و ما درست می‌گوییم. اگر - را در وسط نگه داریم، نزدیکترین درخت باقی مانده همه + است، بنابراین ما به عنوان + طبقه‌بندی می‌کنیم، که نادرست است.

در مجموع از هر ده بار یکی اشتباه کردیم.

$$\text{Error cross-validation}=1/10$$

۴- در مساله دسته‌بندی شکل زیر به روش 1-NN و 3-NN با معیار فاصله $d_1(x, y) = \max_i |x_i - y_i|$ و



۲۵

مهر
یکشنبه
17 October 2021
۱۰ ربيع الاول ۱۴۴۳

$K=1$, $K=3$

Sample	class	d_1	$d_2 = \sum x_i - y_i $
① (۵, -۲)	طبر	$\max(5-5 , 0-(-2)) = 2$	$(5-5 + 0-(-2)) = 2$
② (۰, -۵)	طبر	$\max(5-0 , 0-(-5)) = 5$	$(5-0 + 0-(-5)) = 10$
③ (۸, ۵)	طبر	$\max(5-8 , 0-5) = 5$	$(5-8 + 0-5) = 10$
④ (۵, ۵)	طبر	$\max(5-5 , 0-5) = 5$	$(5-5 + 0-5) = 5$
⑤ (۰, ۰)	مُلت	$\max(5-0 , 0-0) = 5$	$(5-0 + 0-0) = 5$
⑥ (-۵, ۰)	مُلت	$\max(5-(-5) , 0-0) = 10$	$(5-(-5) + 0-0) = 10$
⑦ (-۵, -۵)	مُلت	$\max(5-(-5) , 0-(-5)) = 10$	$(5-(-5) + 0-(-5)) = 15$
⑧ (-۵, ۵)	مُلت	$\max(5-(-5) , 0-5) = 10$	$(5-(-5) + 0-5) = 15$

$K=1$ $x = \begin{bmatrix} 5 \\ 0 \end{bmatrix}$ (توزیعین حساب) $N_1 \Rightarrow d_1 = 2, d_2 = 2$ Class → طبر

$K=3$ $x = \begin{bmatrix} 5 \\ 0 \end{bmatrix}$ (توزیعین حساب) $N_1 \Rightarrow d_1 = 2, d_2 = 2$ Class, طبر

$y = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$ (توزیعین حساب) $N_2 \Rightarrow d_1 = 5, d_2 = 5$ Class, مُلت → $K=3$ کلاس ۱

$y = \begin{bmatrix} 8 \\ 5 \end{bmatrix}$ (توزیعین حساب) $N_3 \Rightarrow d_1 = 5, d_2 = 10$ Class, طبر

۲۴

مهر
شنبه
16 October 2021
۹ ربيع الاول ۱۴۴۳

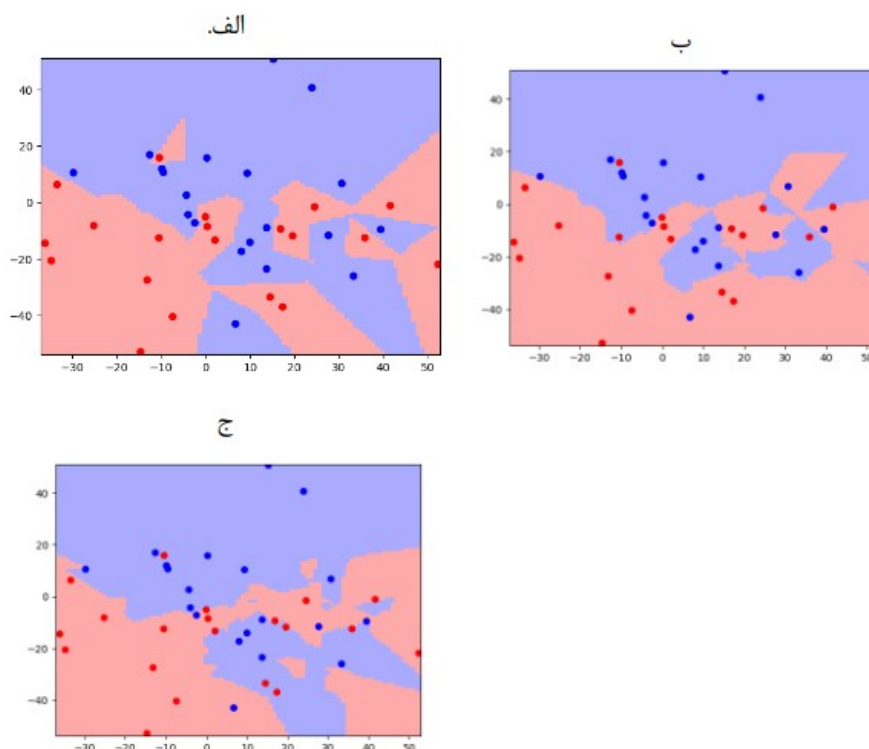
$K=1$ به ازای نمونه اول $d_1 = 2$ توزیعین حساب Class = طبر

$K=3$ به ازای نمونه اول $d_1 = 2 \rightarrow$ Class = طبر
به ازای نمونه دوم $d_1 = 5 \rightarrow$ Class = مُلت
به ازای نمونه سوم $d_1 = 5 \rightarrow$ Class = طبر

$K=1$ $d_2 = 2$ به ازای نمونه اول Class = طبر

$K=3$ N_1 : به ازای نمونه اول $d_2 = 2 \rightarrow$ کلاس طبر
 N_2 : به ازای نمونه دوم $d_2 = 5 \rightarrow$ کلاس مُلت
 N_3 : به ازای نمونه سوم $d_2 = 10 \rightarrow$ کلاس طبر
کلاس نهایی $K=3$ به ازای همه نمونه ها کلاس طبر است.

۵- مقدار K را برای سه کلسیفایر K-Nearest Neighbors زیر با توجه به مرز تصمیم مدل مشخص کنید (۶ نمره)



اگر با دقت تصاویر بالا را بررسی کنید، می بینید که با افزایش مقدار K مرز بین دو کلاس هموارتر می شود. برای تعیین میزان درست K باید دو پارامتر خطای آموزش و خطای اعتبار سنجی را بررسی کنیم.

طبقه‌بندی‌کننده الف) یک مرز تصمیم افقی در $y = 20$ دارد. طبقه‌بندی‌کننده ب) یک مرز تصمیم‌گیری مورب از $(20, 20)$ تا $(-20, -20)$ دارد. طبقه بندی ج) دارای مرز تصمیم عمودی در $x = 10$ است. اگر به نقطه داده در $(10, -20)$ نگاه کنیم، توسط طبقه‌بندی‌کننده الف به عنوان مثبت طبقه‌بندی می‌شود، اما طبقه‌بندی‌کننده ب و ج منفی است. این نشان می‌دهد که سه طبقه‌بندی‌کننده مرزهای تصمیم متفاوتی دارند.

بنظر شکل الف) $K = 1$ را انتخاب کنیم، طبقه بندی کننده به سادگی کلاس نقطه داده را بر اساس برچسب نزدیکترین همسایه خود پیش بینی می کند. این ساده‌ترین شکل طبقه‌بندی K -نزدیک‌ترین همسایگان است، اما اگر داده‌ها به خوبی از هم جدا شوند، می‌تواند بسیار مؤثر باشد. برای تأیید اینکه $K = 1$ انتخاب خوبی است، می‌توانیم به دقت طبقه‌بندی‌کننده‌ها در داده‌های آموزشی نگاه کنیم. جدول زیر دقت طبقه‌بندی‌کننده الف را در داده‌های آموزشی نشان می‌دهد، با این فرض که $K = 1$: تمام داده‌های آموزشی را بدرستی جدا کرده است.

A1 Classifier		
	A	B
1	Classifier	Accuracy
2	A	100%

برای شکل ب) کمی خطا در داده‌های آموزشی افزایش می‌یابد به این معنی که k باید بزرگ‌تر از ۱ باشد. داده‌های قرمز و آبی به خوبی از هم تفکیک نشده‌اند و مقدار k با توجه به خطای آموزشی برابر $k=7$ است. برای شکل ج) هم مشاهده می‌کنیم که مرز تصمیم هموار تر شده و خطای آموزشی افزایش داشته و مقدار $k=10$ است.

سوال ۶) در مورد Radial Basis Function به سؤالات زیر پاسخ دهید.

الف) ویژگی‌های آن را بیان کنید.

- Radial Basis Function یا RBF یک تابع حقیقی است که مقدار آن فقط به فاصله ورودی از یک نقطه ثابت، معمولاً مبدا یا یک مرکز، بستگی دارد. به عبارت دیگر، RBF یک تابع شعاعی است که شکل آن در همه جهات یکسان است.
- RBFها اغلب به عنوان یک مجموعه‌ای از توابع استفاده می‌شوند که یک فضای تابعی از علاقه را تشکیل می‌دهند، به همین دلیل نام RBF به آن‌ها داده شده است. مجموعه‌های RBF معمولاً برای تقریب توابع داده شده استفاده می‌شوند. RBF می‌تواند به عنوان یک تابع پایه برای تقریب توابع غیرخطی استفاده شود. این روش به ما اجازه می‌دهد که یک تابع غیرخطی را با استفاده از ترکیب خطی از RBFها با مراکز و پارامترهای مختلف نمایش دهیم. این روش می‌تواند برای تحلیل داده‌ها، تطبیق سیگنال‌ها، تقریب توابع و مسائل دیگر کاربرد داشته باشد.
- RBFها همچنین به عنوان یک هسته در دسته بندی بردار پشتیبان یا یک شبکه عصبی مصنوعی کاربرد دارند. این ویژگی به الگوریتم KNN و SVM این اجازه را می‌دهد تا با داده‌هایی که مرز تصمیم غیر خطی دارند کار کند. دقت طبقه بندی را افزایش دهند.
- RBFها می‌توانند از نوع چندجمله‌ای، چندضلعی، گوسی، معکوس چندضلعی، معکوس چندجمله‌ای یا بی‌نهایت صاف باشند.
- RBFها می‌توانند یک پارامتر شکل داشته باشند که می‌تواند برای تغییر مقیاس ورودی تابع شعاعی استفاده شود.
- RBF می‌تواند به عنوان یک تابع فعال‌سازی در شبکه‌های عصبی استفاده شود. این شبکه‌های عصبی معمولاً از سه لایه تشکیل شده‌اند: یک لایه ورودی، یک لایه پنهان با توابع فعال‌سازی RBF و یک لایه خروجی با توابع فعال‌سازی خطی یا غیرخطی. این شبکه‌های عصبی می‌توانند با داده‌هایی که دارای الگوها یا روابط غیرخطی هستند کار کنند و معمولاً سریع‌تر از شبکه‌های عصبی چندلایه آموزش می‌بینند.

ب) کاربرد آن برای تقریب تابع را شرح دهید.

RBF می‌تواند برای تقریب هر تابع پیوسته استفاده شود. برای این کار، ابتدا باید داده‌های آموزشی را جمع‌آوری کرد. سپس، باید گره‌های مخفی RBF را به گونه‌ای انتخاب کرد که نقاط داده‌های آموزشی را به خوبی پوشش دهند. در نهایت، باید وزن‌های RBF را با استفاده از روش‌های حداقل مربعات یا حداقل خطای مربعات وزن‌دار، آموزش داد.

RBF می تواند به عنوان یک تابع پایه برای تقریب توابع غیرخطی استفاده شود. این روش به ما اجازه می دهد که یک تابع غیرخطی را با استفاده از ترکیب خطی از RBF ها با مراکز و پارامترهای مختلف نمایش دهیم. این روش می تواند برای تحلیل داده ها، تطبیق سیگنال ها، تقریب توابع و مسائل دیگر کاربرد داشته باشد. فرآیند تقریب تابع با RBF به صورت زیر است:

1. داده های آموزشی را جمع آوری کنید.
2. گره های مخفی RBF را انتخاب کنید.
3. وزن های RBF را آموزش دهید.

ج) آن را با شبکه عصبی مقایسه کرده و نحوه آموزش آن را بیان کنید. RBF و شبکه عصبی هر دو می توانند برای مدل سازی داده های غیرخطی استفاده شوند. با این حال، تفاوت های اساسی بین این دو روش وجود دارد:

	A1	A	B	C
1		ویژگی	RBF	شبکه عصبی
2		نوع تابع پایه	تابع پایه شعاعی	تابع فعال سازی
3		تعداد پارامترهای قابل تنظیم	تعداد گره های مخفی + تعداد وزن ها	تعداد نورون ها در هر لایه + تعداد وزن ها
4		روش آموزش	حداقل مربعات یا حداقل خطای مربعات وزن دار	الگوریتم یادگیری نظارت شده
5				
6				
7				

RBF می تواند به عنوان یک تابع فعال سازی در شبکه عصبی استفاده شود. این شبکه عصبی معمولاً از سه لایه تشکیل شده اند: یک لایه ورودی، یک لایه پنهان با توابع فعال سازی RBF و یک لایه خروجی با توابع فعال سازی خطی یا غیرخطی. این شبکه عصبی می تواند با داده هایی که دارای الگوها یا روابط غیرخطی هستند کار کند و معمولاً سریع تر از شبکه های عصبی چندلایه آموزش می بینند.

برای آموزش یک شبکه عصبی با توابع فعال سازی RBF، می توانیم از روش های زیر استفاده کنیم:

- روش دو مرحله ای: در این روش، ابتدا مراکز و پهنای RBF ها را با استفاده از روش های خوشه بندی یا بهینه سازی تعیین می کنیم. سپس وزن های لایه خروجی را با استفاده از روش کمترین مربعات یا رگرسیون محاسبه می کنیم
- روش تک مرحله ای: در این روش، هم زمان مراکز، پهنای و وزن های RBF ها را با استفاده از روش های گرادیانی یا تکاملی بهینه سازی می کنیم.

د) نحوه انتخاب گره های مخفی چگونه است؟

گره های مخفی در RBF معمولاً مراکز توابع شعاعی هستند که بر روی فضای ورودی پخش می شوند. انتخاب گره های مخفی در RBF یک مسئله مهم است که تأثیر زیادی بر عملکرد و دقت شبکه دارد. برای انتخاب گره های مخفی در RBF می توان از روش های مختلفی استفاده کرد که به دو دسته کلی تقسیم می شوند:

روش های غیربهینه: این روش ها مراکز RBF را بدون در نظر گرفتن خطای شبکه انتخاب می کنند. مثلاً می توان از روش های زیر استفاده کرد:

- انتخاب تصادفی: در این روش، تعداد و مکان مراکز RBF را به صورت تصادفی از مجموعه داده انتخاب می کنیم.
 - انتخاب یکنواخت: در این روش، فضای ورودی را به چندین خانه تقسیم می کنیم و مرکز هر خانه را به عنوان یک مرکز RBF در نظر می گیریم.
 - انتخاب خوشه ای: در این روش، از الگوریتم های خوشه بندی مانند K-means یا Fuzzy C-means برای تقسیم مجموعه داده به چند خوشه استفاده می کنیم و مرکز هر خوشه را به عنوان یک مرکز RBF در نظر می گیریم.
 - روش های بهینه: این روش ها مراکز RBF را با در نظر گرفتن خطای شبکه انتخاب می کنند. مثلاً می توان از روش های زیر استفاده کرد:
 - روش افزایشی: در این روش، ابتدا با یک مرکز RBF شروع می کنیم و سپس به صورت تکراری مراکز RBF را به شبکه اضافه می کنیم تا خطای شبکه کمتر از یک حد مشخص شود. برای انتخاب مراکز RBF میتوان از روش های مختلفی مانند انتخاب نقاط داده با خطای بالا، انتخاب نقاط داده با فاصله بیشتر از مراکز موجود یا انتخاب نقاط داده با تأثیر بیشتر بر خطای شبکه استفاده کرد.
 - روش کاهشی: در این روش، ابتدا با تمام نقاط داده به عنوان مراکز RBF شروع می کنیم و سپس به صورت تکراری مراکز RBF را از شبکه حذف می کنیم تا خطای شبکه کمتر از یک حد مشخص شود. برای حذف مراکز RBF می توان از روش های مختلفی مانند حذف مراکز با وزن کم، حذف مراکز با فاصله کم از مراکز دیگر یا حذف مراکز با تأثیر کم بر خطای شبکه استفاده کرد.
 - روش تطبیقی: در این روش، هم زمان با آموزش وزن های شبکه، مراکز و پهنای RBF ها را نیز به روز می کنیم. برای این کار می توان از روش های گرادینانی یا تکاملی برای بهینه سازی پارامترهای شبکه استفاده کرد.
 - انتخاب گره های مخفی به صورت مبتنی بر الگوریتم های ژنتیک: در این روش، الگوریتم های ژنتیک برای یافتن مجموعه ای از گره های مخفی که بهترین عملکرد را دارند، استفاده می شوند.
- در عمل، معمولاً از ترکیبی از این روش ها برای انتخاب گره های مخفی RBF استفاده می شود.
- سؤال ۷) فرض کنید شما عمل یادگیری را با مجموعه ای از داده ها که به یکی از دو دسته ممکن تعلق دارند انجام داده اید. سپس ناگهان متوجه می شوید که دسته بندی بکار رفته معکوس بوده است (مثال های مثبت بصورت منفی و بالعکس برچسب زده شده اند).
- توضیح دهید که اگر الگوریتم یادگیری شما هریک از موارد زیر باشد چه می توانید بکنید؟
- درخت تصمیم، شبکه عصبی، نزدیکترین همسایه.

نزدیکترین همسایه: الگوریتم های نزدیکترین همسایه برای پیش بینی به نزدیکی نقاط داده تکیه می کنند. معکوس کردن برچسب ها می تواند الگوریتم را گیج کند، زیرا ممکن است یک نقطه داده را با نزدیک ترین همسایگان خود بر اساس برچسب های نادرست مرتبط کند. برای رسیدگی به این وضعیت، الگوریتم نزدیکترین همسایه را می توان تغییر داد تا برچسب های کلاس نزدیکترین همسایه ها را در فرآیند تصمیم گیری خود در نظر بگیرد. این می تواند به کاهش تأثیر معکوس برچسب بر عملکرد الگوریتم کمک کند. به طور کلی، مهم است که از تأثیر بالقوه برگرداندن برچسب بر الگوریتم های یادگیری ماشین آگاه باشید. با درک اینکه چگونه الگوریتم های مختلف با معکوس کردن برچسب کنترل می کنند، می توانید گام های مناسبی برای کاهش اثرات آن بردارید و از استحکام مدل های خود اطمینان حاصل کنید.

نزدیکترین همسایه: اگر دسته بندی داده ها معکوس باشد، شما می توانید الگوریتم نزدیکترین همسایه را بدون تغییر مجدد استفاده کنید. زیرا الگوریتم نزدیکترین همسایه فقط بر اساس فاصله بین داده ها برای انتخاب برچسب عمل می کند و به برچسب های داده ها توجه نمی کند. بنابراین، اگر برچسب ها معکوس باشند، الگوریتم نزدیکترین همسایه همانند قبل عمل می کند و فقط پیش بینی های نهایی را معکوس می کند.

درخت تصمیم: اگر دسته بندی داده ها معکوس باشد، شما می توانید درخت تصمیم را بدون تغییر مجدد استفاده کنید. زیرا درخت تصمیم فقط بر اساس معیارهایی مانند انترپی یا شاخص جینی برای انتخاب شرایط تقسیم داده ها عمل می کند و به برچسب های داده ها توجه نمی کند. بنابراین، اگر برچسب ها معکوس باشند، درخت تصمیم همانند قبل عمل می کند و فقط پیش بینی های نهایی را معکوس می کند.

شبکه عصبی: اگر دسته بندی داده ها معکوس باشد، شما نمی توانید شبکه عصبی را بدون تغییر مجدد استفاده کنید. زیرا شبکه عصبی بر اساس برچسب های داده ها برای بهینه سازی وزن ها و توابع فعال سازی عمل می کند. بنابراین، اگر برچسب ها معکوس باشند، شبکه عصبی دچار خطای بزرگ می شود و پیش بینی های غلط می کند. برای رفع این مشکل، شما می توانید یکی از روش های زیر را انجام دهید:

1. شبکه عصبی را با داده های معکوس مجدداً آموزش دهید. این روش ساده و موثر است، اما زمان بر و هزینه بر است

2. خروجی شبکه عصبی را معکوس کنید. این روش سریع و ارزان است، اما فقط در صورتی کار می کند که شبکه عصبی تنها دو دسته را پیش بینی می کند. در صورتی که شبکه عصبی بیش از دو دسته را پیش بینی می کند، این روش کار نمی کند

سؤال ۸) فرایند تصمیم گیری درخت تصمیم و الگوریتم K-nearest neighbors برای طبقه بندی یک مجموعه داده مقایسه کنید. نقاط قوت و ضعف هر رویکرد را مورد بحث قرار دهید و نمونه هایی از سناریوهایی را ارائه دهید که در آن یک الگوریتم ممکن است از دیگری بهتر عمل کند.

درخت تصمیم:

درخت تصمیم یک الگوریتم یادگیری نظارتی است که از یک درخت ساختار یافته برای نمایش روابط بین ویژگی ها و کلاس ها استفاده می کند. در مرحله آموزش، درخت تصمیم به صورت بازگشتی ساخته می شود. در هر مرحله، یک ویژگی انتخاب می شود که بیشترین تفاوت بین کلاس ها را ایجاد می کند. سپس، داده ها بر اساس این ویژگی تقسیم می شوند و فرایند برای هر زیرمجموعه تکرار می شود.

در مرحله پیش بینی، نمونه جدید از بالا به پایین درخت حرکت می کند و در هر گره، ویژگی مربوطه بررسی می شود. نمونه در نهایت به یک برگ درخت می رسد که کلاس آن را تعیین می کند.

الگوریتم K-nearest neighbors:

الگوریتم K-nearest neighbors یک الگوریتم یادگیری نظارتی است که از مجموعه ای از داده های آموزشی برای تعیین کلاس یک نمونه جدید استفاده می کند. در مرحله آموزش، الگوریتم K-nearest neighbors فقط برچسب های کلاس نمونه های آموزشی را ذخیره می کند.

در مرحله پیش‌بینی، فاصله نمونه جدید با k نمونه آموزشی نزدیک‌ترین به آن محاسبه می‌شود. سپس، کلاسی که بیشترین تعداد نمونه‌های نزدیک به آن را دارد، به عنوان کلاس نمونه جدید تعیین می‌شود.

نقاط قوت و ضعف:

درخت تصمیم:

نقاط قوت:

- انعطاف‌پذیری: درخت تصمیم می‌تواند روابط پیچیده بین ویژگی‌ها و کلاس‌ها را مدل‌سازی کند.
- قابلیت تفسیر: درخت تصمیم می‌تواند به عنوان یک ابزار برای درک روابط بین ویژگی‌ها و کلاس‌ها استفاده شود.
- کارایی: درخت تصمیم می‌تواند به سرعت نمونه‌های جدید را طبقه‌بندی کند.
- درخت تصمیم ساده و قابل فهم است و می‌توان آن را به صورت یک نمودار شاخه‌ای نشان داد.
- درخت تصمیم می‌تواند با داده‌هایی که دارای ویژگی‌های عددی یا نامی هستند کار کند.
- درخت تصمیم می‌تواند با داده‌هایی که دارای مقادیر گم‌شده یا ناهمگن هستند کار کند.

نقاط ضعف:

- حساسیت به داده‌های ناهنجار: درخت تصمیم می‌تواند به داده‌های ناهنجار حساس باشد و این امر می‌تواند منجر به پیش‌بینی‌های نادرست شود.
- احتمال بیش‌برازش: درخت تصمیم ممکن است به داده‌های آموزشی بیش از حد برازش شود و این امر می‌تواند منجر به عملکرد ضعیف در داده‌های جدید شود.
- درخت تصمیم ممکن است به انتخاب شرایط تقسیم داده‌ها حساس باشد و نتایج متفاوتی با تغییر کوچک در داده‌ها بدهد.
- درخت تصمیم ممکن است با داده‌هایی که دارای ویژگی‌های پیوسته و کلاس‌های غیرمتوازن هستند مشکل داشته باشد.
- درخت تصمیم ممکن است با داده‌هایی که دارای وابستگی‌های پیچیده بین ویژگی‌ها هستند مشکل داشته باشد.

الگوریتم K-nearest neighbors

نقاط قوت:

- مقاوم به داده‌های ناهنجار: الگوریتم K-nearest neighbors به داده‌های ناهنجار مقاوم است و این امر می‌تواند منجر به پیش‌بینی‌های دقیق‌تر شود.
- KNN ساده و انعطاف‌پذیر است و نیازی به ساخت یا آموزش یک مدل ندارد.
- KNN می‌تواند با داده‌هایی که دارای ویژگی‌های پیوسته یا نامی هستند کار کند.
- KNN می‌تواند با داده‌هایی که دارای الگوها یا روابط غیرخطی هستند کار کند.
- KNN می‌تواند با تغییر مقدار K بهینه شود و به عنوان یک الگوریتم تطبیقی عمل کند.

نقاط ضعف:

- عدم انعطاف‌پذیری: الگوریتم K-nearest neighbors نمی‌تواند روابط پیچیده بین ویژگی‌ها و کلاس‌ها را مدل‌سازی کند.
- عدم قابلیت تفسیر: الگوریتم K-nearest neighbors به عنوان یک ابزار برای درک روابط بین ویژگی‌ها و کلاس‌ها مفید نیست.
- کندی: الگوریتم K-nearest neighbors می‌تواند برای طبقه‌بندی نمونه‌های جدید کند باشد.
- KNN نیاز به ذخیره و محاسبه تمام داده‌ها دارد و ممکن است برای داده‌های بزرگ زمانبر و حافظه‌بر باشد
- KNN نیاز به انتخاب یک معیار فاصله مناسب برای محاسبه شباهت بین داده‌ها دارد
- KNN نیاز به پیش پردازش داده‌ها مانند مقیاس بندی یا حذف ویژگی‌های نامرتبط دارد.

سناریوهای عملکرد بهتر

درخت تصمیم

- هنگامی که داده‌ها دارای روابط پیچیده بین ویژگی‌ها و کلاس‌ها هستند.
- هنگامی که نیاز به قابلیت تفسیر وجود دارد.
- هنگامی که نیاز به سرعت بالای طبقه‌بندی وجود دارد.

الگوریتم K-nearest neighbors

- هنگامی که داده‌ها دارای داده‌های ناهنجار هستند.
- هنگامی که نیاز به مقاومت به بیش‌برازش وجود دارد.

نتیجه‌گیری

درخت تصمیم و الگوریتم K-nearest neighbors هر دو الگوریتم‌های طبقه‌بندی قدرتمندی هستند که می‌توانند برای طیف گسترده‌ای از کاربردها استفاده شوند. با این حال، این دو الگوریتم از رویکردهای کاملاً متفاوتی برای انجام این کار استفاده می‌کنند و نقاط قوت و ضعف مختلفی دارند. انتخاب الگوریتم مناسب برای یک کاربرد خاص به عوامل مختلفی بستگی دارد، از جمله ماهیت داده‌ها، نیازهای خاص کاربر و منابع در دسترس.

سوال ۹) همانطور که میدانید در knn باید محاسبه‌ای برای کل داده‌ها انجام و نزدیکترین آن را انتخاب نمود و دسته‌بندی را انجام داد. به فصل ۶ کتاب learning from data مراجعه کرده و به سؤالات زیر پاسخ دهید.

الف) مفهوم سازگاری در condensed data set (مجموعه داده متراکم) و عیب آن را بیان کنید. تفاوت آن با سازگاری مجموعه داده آموزش چیست؟

ب) با توجه به مفاهیم قسمت قبل الگوریتم نزدیکترین همسایه متراکم را توضیح دهید.

الف) سازگاری در مجموعه داده‌های فشرده سازگاری در مجموعه داده‌های فشرده به این معنی است که مجموعه داده‌های فشرده باید بتواند نتایجی مشابه مجموعه داده‌های اصلی برای یک کار مشخص ایجاد کند. این یک نیاز مهم برای مجموعه داده‌های فشرده است، زیرا تضمین می‌کند که مجموعه داده‌های فشرده می‌تواند به جای مجموعه داده‌های اصلی بدون کاهش دقت استفاده شود.

دو نوع اصلی سازگاری وجود دارد: سازگاری مجموعه آموزشی: این بدان معناست که مجموعه داده های فشرده و مجموعه داده های اصلی طبقه بندی های یکسانی را در نقاط داده آموزشی ارائه می دهند. سازگاری مجموعه تست: این بدان معنی است که مجموعه داده های فشرده و مجموعه داده های اصلی طبقه بندی های یکسانی را در نقاط داده آزمایشی ارائه می دهند.

سازگاری مجموعه آموزشی یک نیاز ضعیف تر از سازگاری مجموعه تست است. این به این دلیل است که از نقاط داده آموزشی برای آموزش مدل و از نقاط داده آزمون برای ارزیابی مدل استفاده می شود. در نتیجه، مدلی که بر روی یک مجموعه آموزشی که با مجموعه داده های اصلی سازگار است، آموزش داده می شود، نسبت به مدلی که بر روی مجموعه آموزشی که با مجموعه داده های اصلی سازگار نیست، به خوبی تعمیم داده می شود.

- **معایب سازگاری در مجموعه داده های فشرده:** دستیابی به سازگاری در مجموعه داده های فشرده می تواند دشوار باشد، به خصوص برای مجموعه داده های بزرگ و پیچیده. این به این دلیل است که یافتن یک مجموعه داده فشرده که به اندازه کافی کوچک باشد تا کار با آن کارآمد باشد، دشوار است، در حالی که هنوز به اندازه کافی بزرگ باشد تا مهم ترین اطلاعات را از مجموعه داده اصلی بگیرد. علاوه بر این، سازگاری می تواند به قیمت دقت تمام شود. به عنوان مثال، یک مجموعه داده فشرده که با نمونه برداری از مجموعه داده های اصلی ایجاد می شود ممکن است به اندازه مجموعه داده های اصلی دقیق نباشد، زیرا نمونه ممکن است تمام اطلاعات مهم از مجموعه داده های اصلی نباشد.

- سازگاری در مجموعه داده های فشرده چه تفاوتی با سازگاری مجموعه داده های آموزشی دارد؟ سازگاری در مجموعه داده های فشرده مفهومی گسترده تر از سازگاری مجموعه داده های آموزشی است. سازگاری مجموعه داده های آموزشی به سادگی نوعی سازگاری است که مختص نقاط داده آموزشی است. از سوی دیگر، سازگاری در مجموعه داده های فشرده می تواند برای هر نوع داده، از جمله داده های آموزشی، داده های آزمایشی و حتی داده های جدیدی که قبلاً دیده نشده است، اعمال شود.

ب) الگوریتم متراکم نزدیکترین همسایه :

الگوریتم نزدیکترین همسایه متراکم (CNN) یک الگوریتم ساده است که می تواند برای یافتن نزدیکترین نقاط در یک مجموعه داده فشرده استفاده شود. این الگوریتم با افزودن مکرر نزدیک ترین نقطه به مجموعه داده های فشرده کار می کند تا زمانی که مجموعه داده های فشرده با مجموعه آموزشی سازگار شود.

یک شرط ضعیف تر این است که داده های فشرده فقط با مجموعه آموزشی سازگار باشند، به این معنی که داده های فشرده و داده های آموزشی کامل طبقه بندی های یکسانی را در نقاط داده آموزشی ارائه می دهند.

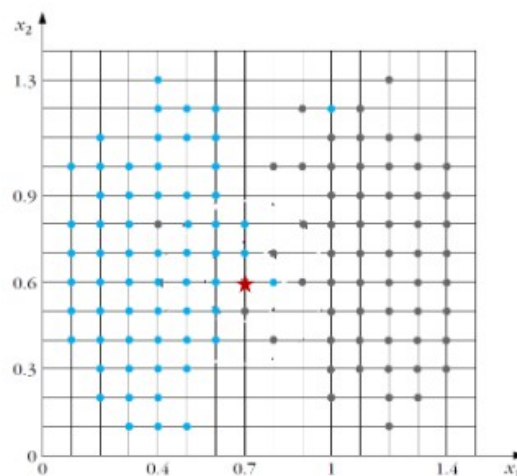
سازگاری مجموعه آموزشی به این معنی است که مجموعه داده های فشرده و مجموعه داده های اصلی طبقه بندی های یکسانی را در نقاط داده آموزشی ارائه می دهند. الگوریتم CNN یک الگوریتم ساده و کارآمد است، اما استفاده از آن برای مجموعه داده های بزرگ می تواند از نظر محاسباتی گران باشد. این به این دلیل است که الگوریتم باید فاصله بین هر نقطه در مجموعه داده های فشرده و هر نقطه در مجموعه داده اصلی را محاسبه کند. در اینجا مثالی از نحوه عملکرد الگوریتم CNN آورده شده است:

1. با یک مجموعه داده فشرده شروع کنید که حاوی تعداد کمی از نقاط از مجموعه داده اصلی است.
2. نزدیکترین نقطه را در مجموعه داده های اصلی به هر نقطه در مجموعه داده های فشرده پیدا کنید.
3. نزدیکترین نقطه را به مجموعه داده های فشرده اضافه کنید.
4. مراحل 2 و 3 را تکرار کنید تا مجموعه داده های فشرده با مجموعه آموزشی سازگار شود.

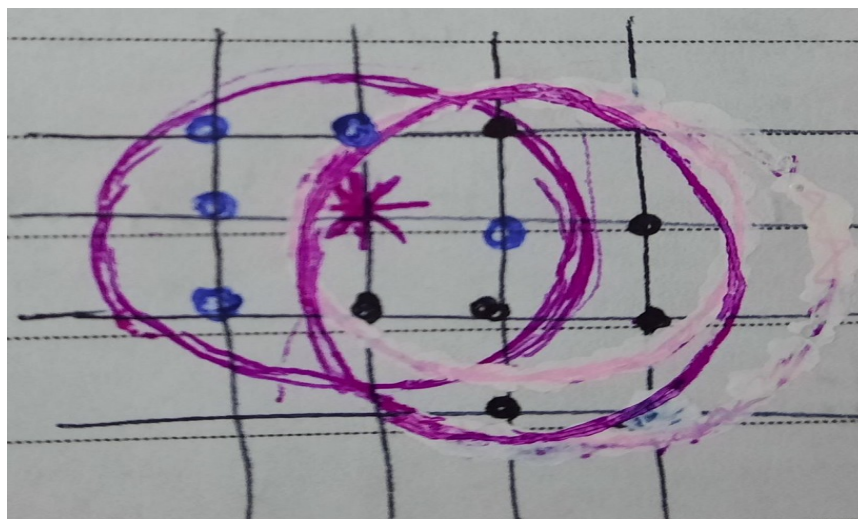
هنگامی که مجموعه داده های فشرده با مجموعه آموزشی سازگار است، می توان آن را به جای مجموعه داده اصلی برای هر کاری که نیاز به استفاده از مجموعه داده اصلی دارد استفاده کرد. نتیجه مجموعه داده های فشرده سازگار ابزار قدرتمندی

است که می تواند برای بهبود کارایی و اثربخشی وظایف پردازش داده ها استفاده شود. با این حال، مهم است که از معایب سازگاری، مانند دشواری دستیابی به سازگاری و پتانسیل کاهش دقت، آگاه باشید. به عبارت دیگر، متن چالش های استفاده از الگوریتم CNN برای مترکم کردن مجموعه های داده در حالی که سازگاری با مجموعه داده اصلی را حفظ می کند، بحث می کند. نویسنده خاطرنشان می کند که الگوریتم CNN همیشه مجموعه داده های فشرده ای را تولید نمی کند که با مجموعه آموزشی سازگار باشد، و اینکه فرآیند فشرده سازی مجموعه داده ها می تواند از نظر محاسباتی گران باشد. با این حال، نویسنده همچنین خاطرنشان می کند که مزایای استفاده از مجموعه داده های فشرده، مانند کاهش هزینه های ذخیره سازی و بهبود عملکرد، اغلب بیشتر از هزینه ها است.

۱۰- دو کلاس با توزیع نقاط بصورت مشکی (کلاس ۱)، و آبی (کلاس ۲) را در نظر بگیرید. طبقه بندی نقطه ستاره به مشخصه (۰.۶، ۰.۷) با روش (تخمین چگالی با k نزدیکترین همسایه) 5NN را مشخص کنید و بگویید که متعلق به کلاس مشکی (کلاس ۱) است یا آبی (کلاس ۲). (۵ نمره امتیازی)



از روش تخمین چگالی با 5NN استفاده می کنیم. در این روش، ابتدا پنج نقطه داده نزدیک ترین به نقطه داده جدید را با توجه به معیار چگالی انتخاب یعنی برای هر کلاس ۵ نمونه نزدیکتر به نقطه ستاره دار را مشخص می کنیم. سپس، برای هر کلاس یک دایره رسم می کنیم که شامل ۵ نقطه مربوط به آن کلاس باشد. و در نهایت کلاسی را که شعاع دایره کمتری دارد انتخاب می کنیم. با توجه به این توضیحات و رسم شکل دایره برای هر کلاس متوجه می شویم که کلاس آبی دایره با شعاع کمتری رسم شده و حجم و چگالی فشرده تری دارد در نتیجه نقطه ستاره دار متعلق به کلاس آبی است.



سؤال ۱۱) فرض کنید مجموعه داده‌ای از قیمت مسکن در یک شهر خاص داریم. با دو ویژگی اندازه خانه و فاصله از مرکز شهر. می‌خواهیم با توجه به این دو ویژگی قیمت خانه نوساز را پیش‌بینی کنیم. اینکار را برای روش *locally weighted regression* توضیح دهید.

برای پیش‌بینی قیمت خانه نوساز با استفاده از روش *locally weighted regression*، ابتدا باید مجموعه داده‌ها را جمع‌آوری کنیم. این مجموعه داده‌ها باید شامل دو ویژگی اندازه خانه و فاصله از مرکز شهر و همچنین قیمت خانه باشد.

در مرحله بعد، باید یک تابع وزن را انتخاب کنیم. تابع وزن تعیین می‌کند که همسایه‌های نزدیک‌تر به نقطه داده جدید تا چه حد بر پیش‌بینی تأثیر می‌گذارند. تابع وزن متداول برای استفاده در *locally weighted regression*، تابع وزن گاوسی است.

در نهایت، باید پارامترهای تابع وزن را آموزش دهیم. این پارامترها معمولاً شامل مقدار میانگین و واریانس تابع وزن هستند.

برای آموزش پارامترهای تابع وزن، می‌توانیم از روش *gradient descent* استفاده کنیم. روش *gradient descent* به ما کمک می‌کند تا پارامترهای تابع وزن را به گونه‌ای تنظیم کنیم که خطای پیش‌بینی را کاهش دهیم.

پس از آموزش پارامترهای تابع وزن، می‌توانیم از آن برای پیش‌بینی قیمت خانه نوساز استفاده کنیم. برای این کار، ابتدا باید فاصله بین نقطه داده جدید و هر یک از نقاط داده آموزشی را محاسبه کنیم. سپس، باید وزن هر نقطه داده را با استفاده از تابع وزن محاسبه کنیم. در نهایت، باید قیمت خانه نوساز را با استفاده از مجموع وزن‌ها و مقادیر قیمت خانه برای نقاط داده آموزشی نزدیک‌ترین به نقطه داده جدید پیش‌بینی کنیم.

در این مثال، دو ویژگی اندازه خانه و فاصله از مرکز شهر برای پیش‌بینی قیمت خانه نوساز استفاده می‌شوند. برای پیش‌بینی قیمت خانه نوساز برای یک خانه با اندازه 100 متر مربع و فاصله 5 کیلومتر از مرکز شهر، ابتدا باید فاصله بین این نقطه داده جدید و هر یک از نقاط داده آموزشی را محاسبه کنیم. سپس، باید وزن هر نقطه داده را با استفاده از تابع وزن گاوسی محاسبه کنیم. در نهایت، باید قیمت خانه نوساز را با استفاده از مجموع وزن‌ها و مقادیر قیمت خانه برای نقاط داده آموزشی نزدیک‌ترین به نقطه داده جدید پیش‌بینی کنیم.

با استفاده از روش *locally weighted regression*، می‌توانیم پیش‌بینی‌های دقیق‌تری نسبت به روش‌های پیش‌بینی خطی داشته باشیم. این به این دلیل است که *locally weighted regression* به ویژگی‌های خاص نقطه داده جدید توجه می‌کند.