به نام خدا

عنوان:

تکلیف نهم درس یادگیری ماشین

استاد:

دکتر پدرام

نویسنده:

محمدعلی مجتهدسلیمانی

تاریخ:

۱۴۰۳/۰۹/۱۸

سوال ۱

❖ KNN:

الگوریتم KNN یک الگوریتم بدون پارامتر، نمونه محور و نظارت شده از الگوریتم های یادگیری ماشین هست که هم برای کلاس بندی و هم برای رگرسیون مورد استفاده قرار میگیرد. بدون پارامتر به این معنی هست که ما تخمینی دقیقی از توزیع ها حاکم بر داده ها از قبل نداریم و نمیتوانیم از رویکرد روش های پارامتری که توزیع بر داده ها را میدانستیم صرفا به دنبال یک سری پارامتر مثل وزن بودیم استفاده بکنیم. نمونه محور به این معنی هست که در زمان یادگیری، این الگوریتم نمونه های آموزشی را ذخیره و حفظ میکند به جای اینکه سعی کند یادگیری تابع discriminative را داشته باشد.

نحوه کار کردن KNN مشخص است، نقاطی که نزدیک تر بهم دیگر باشند احتمالا متعلق به یک کلاس خواهند بود. در فاز یادگیری، KNN صرفا تمام نقاط داده را ذخیره میکند همراه با برچسب متناظر آنها برای کلاس بندی یا مقدار متناظر آنها برای رگرسیون، در این فاز هیچ مدلی ساخته نمیشود یا ما تخمین پارامتری نخواهیم داشت. در فاز پیشبینی وقتی که داده جدید بدون برچسب وارد میشود، KNN فاصله این نقطه با همه نقاط را بدست میاورد که در این گام معمولا از فاصله اقلیدوسی کمک میگیرد. بعد از این مرحله الگوریتم K تا نزدیک ترین از همسایه های خودش را انتخاب میکند. برای پیشبینی کردن اگر مسائل ما کلاس بندی باشد الگوریتم کلاسی را میان k همسایه انتخاب میکند که اکثریت را داشته باشند به عنوان مثال اگر میان ۵ همسایه اگر ۳ نفر از کلاس A و ۲ نفر از کلاس B باشند الگوریتم کلاس A را انتخاب خواهد کرد به عنوان کلاس مد نظر برای داده جدید. اگر مسئله رگرسیون باشد، الگوریتم میانگین یا یک میانگین وزن دار را میان همسایگان محاسبه میکند و به داده جدید اتلاق میکند. KNN یک رویکرد lazy محسوب میشود زیرا در حین آموزش اصلا مدلی ساخته نمیشود صرفا داده های آموزشی ذخیره میشوند و محاسبات و یافتن همسایگی به مرحله ورود داده جدید موکول میشوند. این روند باعث میشود که روند آموزش بسیار سریعتر باشد و نسبت به داده هایی آموزشی کاملا منعطف باشد.

❖ RBF:

شبکه RBF یک نوع شبکه عصبی feedforward هست که از توابع پایه شعاعی به عنوان تابع فعالیت خود استفاده میکند. RBF برای توابع تخمینی و کلاس بندی الگو ها مناسب هستند.

یک شبکه RBF دارای معماری ۳ لایه است. لایه ورودی: این لایه ورودی را دریافت میکند، تعداد نود ها در این لایه برابر با بعد داده های ورودی ما خواهد بود. لایه مخفی: این لایه، لایه اصلی شبکه خواهد بود که از توابع پایه شعاعی استفاده خواهد کرد. در این لایه ما با ۲ پارامتر مراکز و عرض سروکار داریم. لایه خروجی: این لایه خروجی شبکه را محاسبه میکند که معمولا از یک تابع خطی به عنوان تابع فعالیت استفاده میکند به عنوان مثال جمع وزن دار خروجی لایه های مخفی.

محاسبات RBF در لایه مخفی صورت میگیرد. هر نورون/واحد مخفی فاصله بردار ورودی را از یکی از مراکز محاسبه میکند، معمولا از فاصله اقلیدوسی برای محاسبات استفاده میشود. یک تابع RBF رایج در این لایه برای محاسبه خروجی Gaussian خواهد بود ، فرمول آن در ادامه نوشته شده است:

$$\phi(||x - \mu||) = \exp(- ||x - \mu||^2 / (2\sigma^2))$$

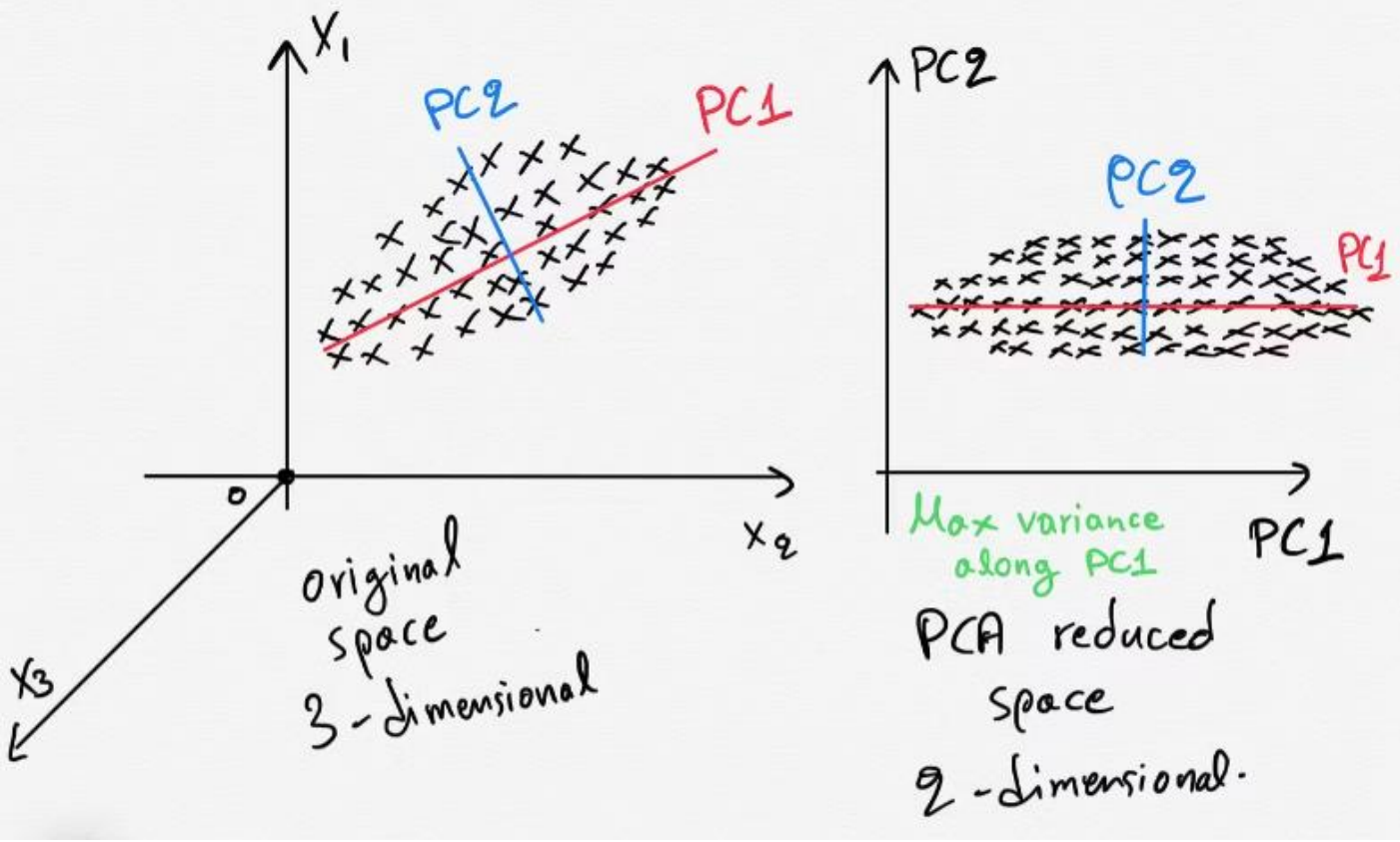
که در اینجا فی یا ϕ، RBF ما خواهد بود.

بعد از این مرحله محاسبات لایه خروجی انجام میشود که حاصل جمع وزن دار خروجی های لایه میانی بعلاوه بایاس خواهد بود. خروجی بیانگر این است که یک ورودی چه قدر به یک ناحیه نزدیک خواهد بود. برای آموزش یک شبکه RBF، ۳ پارامتر باید یاد گرفته شوند: ۱. مراکز ۲. عرض ۳. وزن ها.

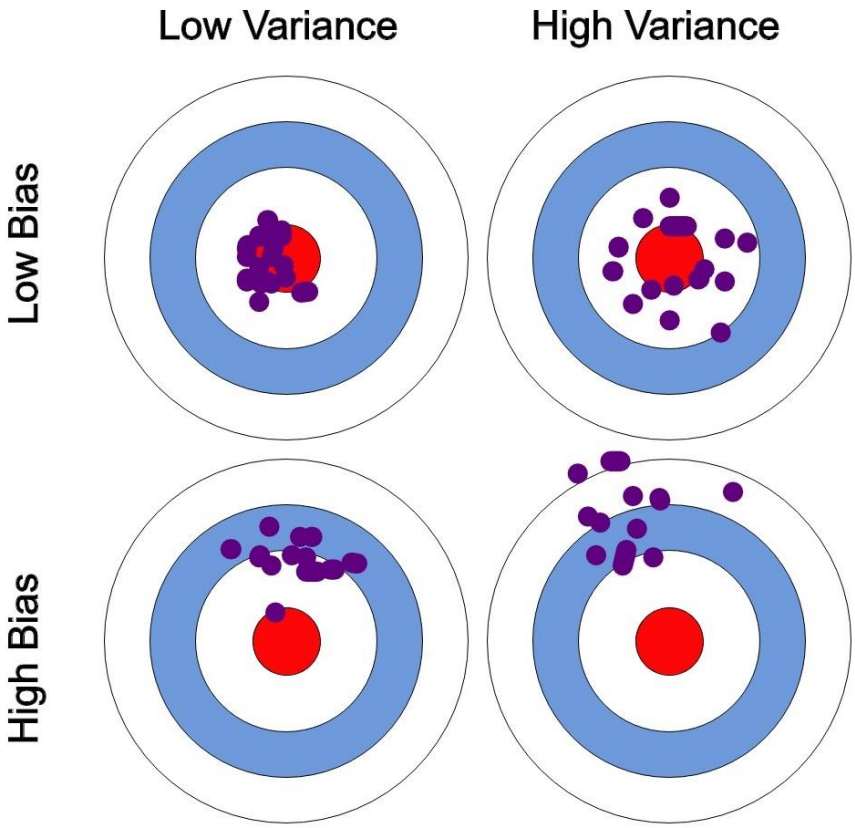
شبکه های RBF یک الگوریتم eager محسوب میشوند. به این معنی که در حین فرآیند یادگیری یک مدل ساخته میشود و پارامتر های عرض، وزن ها و مراکز یاد گرفته خواهند شد.

سوال ۲

الف) (غلط) شبکه های RBF معمولا داده ها را به یک بعد بزرگتر یا مساوی از بعد فعلی نگاشت میکنند در حین فرآیند لایه مخفی و بعد از آن بعد را کاهش میدهند در لایه خروجی تا خروجی مطلوب تولید شود و نمیتوان همیشه تضمین کرد که خروجی شبکه بعدی بزرگتر مساوی از بعد داده ها داشته باشد.



ب) (صحیح) یک عرض خیلی کوچک در Gaussian RBF به این معنی است که تابع فعالیت ما خیلی باریک خواهد بود و به صورت تیغ یا مثلثی شکل پدیدار میشوند، این باعث میشود که RBF فقط نسبت به داده هایی که خیلی به مراکز نزدیک هستند فعال میشود. این منجر به بیش بردازش خواهد شد زیرا فضای کوچیکی پوشش داده میشود و مدل نسبت به یک سری از داده های آموزشی خیلی حساس میشود و نسبت به داده هایی جدید و دیده نشده تعمیم پذیری ندارد در این مقطع مدل high variance و low bias خواهد بود.



ج) (صحیح) به صورت کلی هر چه قدر مقدار k را افزایش بدهیم مدل کمتر نسبت به نویز حساس خواهد بود و مقاوم تر میشود اما ممکن است باعث شود یک سری از الگو ها از دست بروند و مدل به سمت underfitting میل کند. این به این دلیل است که از اونجایی که ما در بین همسایگی ها اکثریت را در نظر میگیریم نسبت به یک داده نویز تنها مقاوم تر خواهد بود و تاثیر کمتری از آن خواهد گرفت.

د) (غلط) وقتی bias بالا باشد فرضیه ما نسبت به داده های قوی تر خواهد بود. در ۵ همسایگی یک مرز تصمیم گیری نرم تری خواهیم داشت و کمتر منعطف خواهد بود و این باعث میشود bias بالا برود و از طرف دیگر variance کاهش پیدا کند.

و) (غلط) تمام محاسبات KNN در زمان ارزیابی صورت میگیرد نسبت به داده هایی جدید برای اینکه باید فاصله را نسبت به همه داده ها انتخاب کند و نزدیک ترین همسایگان را بردار و در زمان آموزش صرفا داده ها را ذخیره میکند پس زمان آموزش آن خیلی کمتر از تست کردن آن است چون عملا فاز آموزشی وجود ندارد.

سوال ۳

برای مقادیر کم k، الگوریتم low bias و high variance خواهد بود به این معنی که مرز تصمیم گیری خیلی منعطف خواهد بود و به راحتی به داده های آموزشی fit میشود و حساسیت به داده ها خیلی بالا میرود این باعث بیش برداز میشود از اونجایی که اگر در همسایگی یک داده نویزی وجود داشته باشد تاثیر بسیاری در فرآیند پیشبینی میگذارد.

برای مقادیر بزرگ k، الگوریتم low variance و high bias خواهد بود به این معنی که مدل نسبت به داده های نویزی بسیار مقاوم تر خواهد بود بخاطر اینکه صرفا یک یا دو داده حتی اگر نویزی باشند تاثیری زیادی نمیگذارند و اکثریت ملاک است. این باعث میشود رز تصمیم گیری نرم تر و کمتر پیچیده باشد و انعطاف کمتری داشته باشد این روند ممکن است باعث underfitting شود.

در کل یک trade-offیی بین این مقادیر کوچک و بزرگ وجود خواهد داشت.

در انتخاب مقدار k هم تعداد ویژگی و هم تعداد کلاس ها موثر خواهند بود. اگر کلاس های ما زیاد باشد باید مقدار k بالا باشد تا مطمئن شویم از هر کلاس به تعداد کافی نماینده وجود خواهد داشت. اگر ویژگی ها زیاد باشد به علت curse of dimensionality نزدیک ترین همسایه شاید زیاد معنی نداشته باشد و سخت تر باشد بفهمیم دقیقا کدام نقاط نزدیک تر هستند.

تعداد مجموعه نقاط آموزشی هم در انتخاب k تاثیر گذار خواهند بود. اگر اندازه مجموعه کوچک باشد با مقادیر پایین k هم میتوانیم عملکرد خوبی داشته باشیم. اگر مجموعه ما بزرگ باشد نیاز است مقادیر k هم بزرگتر شوند تا تعمیم پذیری بالاتر برود.

بهترین رویکرد در انتخاب k استفاده از k-cross validation است.

سوال ٤

الف)

خیر، KNN عموما نمیتواند بر روی داده های نا متوازن خوب عمل کند. به این دلیل که KNN بر این اساس عمل میکند که نقاط داده های نزدیک به هم به احتمال زیاد به یک کلاس تعلق دارند. در یک مجموعه نا متوازن کلاسی که اکثریت را دارا است بیشترین نماینده و نقاط را در همسایگی دارا است پس وقتی میخواهیم برای داده جدید پیشبینی انجام بدهیم داده های کلاس اکثریت، بیشترین نقاط را دارا هستند حتی اگر داده تست ما نزدیک به کلاس اقلیت باشد باز تحت تاثیر کلاس اکثریت قرار میگیرد، در نتیجه این اتفاق پیشبینی ما برای داده های جدید ضعیف خواهد بود. اگر بتوانیم یک ضریب وزن داریم به همسایگی ها بدهیم تا تاثیر کلاس اقلیت بالاتر رود شاید بتوان این روند را بهتر کرد.

ب)

خیر، استفاده خام از KNN برای ویژگی های رسته ای پیشنهاد نمیشود. به این دلیل که KNN به فاصله متری مانند فاصله اقلیدوسی برای شباهت میان نقاط داده ها و ورودی اتکا میکند. این معیار برای داده هایی عددی طراحی شده است و اگر برای داده هایی رسته ای همینجوری اقدام کنیم منطقی نخواهد بود. مثلا فرق بین دیپلم با دکترا چی خواهد بود؟ آیا این فرق بیشتر از فرق دیپلم با دکترا خواهد بود.

راه حل encode کردن ویژگی های رسته ای خواهد بود، تا تبدیل به نمایندگان عددی برای اون رسته شوند تا بتوانیم مفهوم فاصله را پیاده سازی کنیم و استفاده کنیم.

یکی از روش های رایج one-hot encoding خواهد بود که یک ویژگی binary برای هر رسته میسازیم به عنوان مثال اگر مقدار این ویژگی متناظر با رسته خودش باشد ۱ باشد و باقی ویژگی ما ۰ شود، مثال:

- Diploma -> [1, 0, 0, 0]
- Bachelor's -> [0, 1, 0, 0]
- Master's -> [0, 0, 1, 0]
- PhD -> [0, 0, 0, 1]

البته عیب این روش افزایش بعد آن خواهد بود برای رسته های بیشتر.

یک روش دیگر **ordinal encoding** است، به این معنی که میایم و یک مقدار یکتا به هر رسته اتلاق میکنیم.

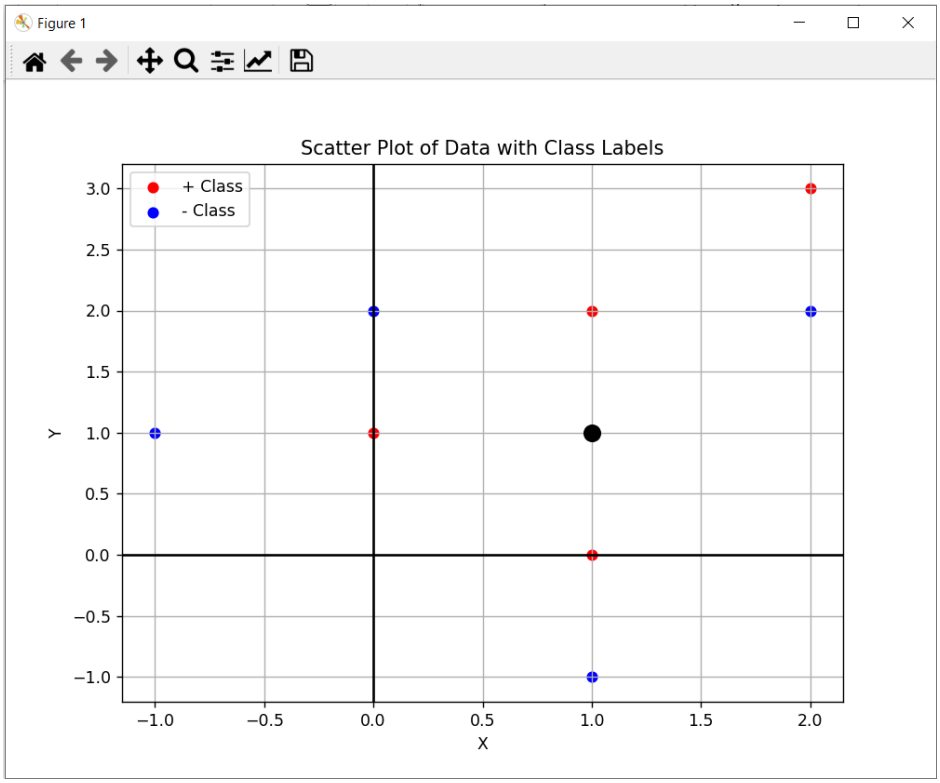
مثال:

- Diploma -> 1
- Bachelor's -> 2
- Master's -> 3
- PhD -> 4

و روش های بیشتر دیگری وجود دارد.

سوال ۵

الف)



نقاط قرمز برای دادهای کلاس "+" و نقاط آبی برای کلاس "–" خواهند بود. نقطه سیاه، داده خواسته شده از ما در صورت سوال است.

ب)

با استفاده از فرمول زیر فاصله اقلیدوسی را برای نقاط **(x1,y1)** و **(x2,y2)** محاسبه میکنیم:

$$distance = sqrt((x2 - x1)^2 + (y2 - y1)^2)$$

حاصل محاسبه فاصله نقطه **(1,1)** با تمام نقاط داده:

X	Y	Class	Distance
-1	1	-	**2**
0	1	+	**1**
0	2	-	**1.41**
1	-1	-	**2**
1	0	+	**1**
1	2	+	**1**
2	2	-	**1.41**
2	3	+	**2.24**

با توجه به محاسبات با در نظر گرفتن ۳ همسایگی نقاط زیر نزدیک ترین همسایه ها خواهند بود:

(0,1) with distance 1 (Class +)

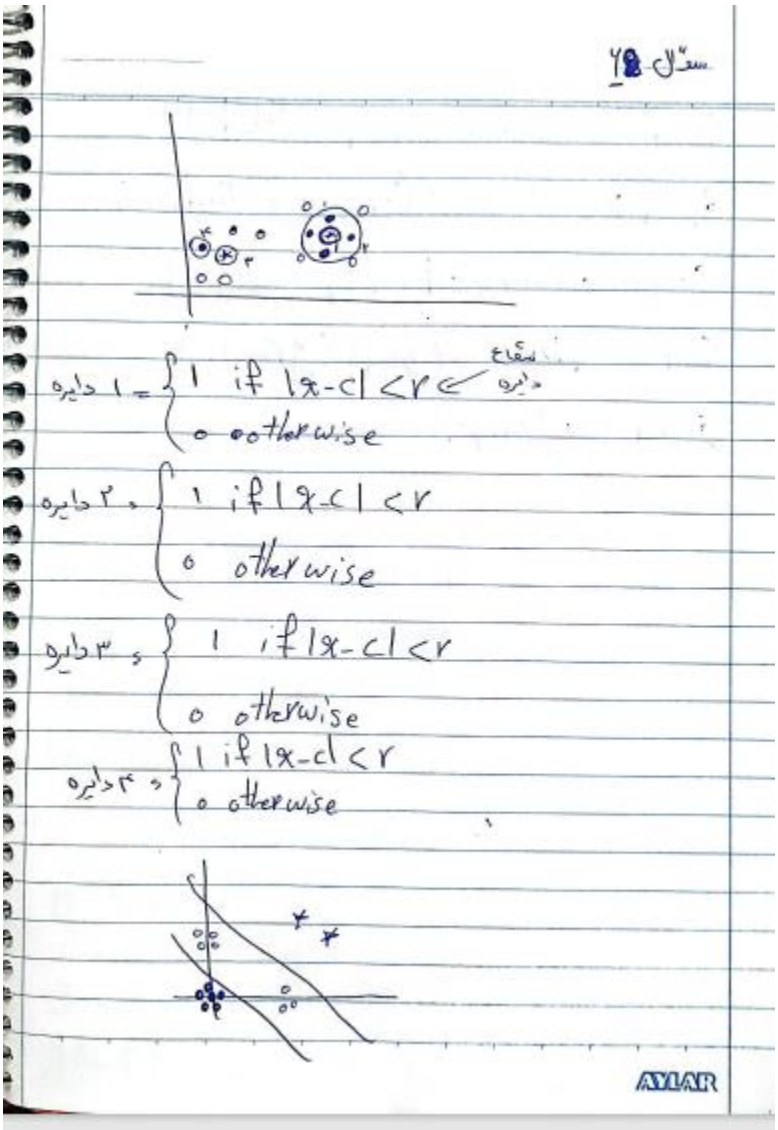
(1,0) with distance 1 (Class +)

(1,2) with distance 1 (Class +)

با محاسبه اکثریت در همسایگی میتوانیم متوجه شویم که داده جدید بر اساس ۳ همسایگی متعلق به کلاس "+" خواهد بود.

ج) از میان هشت داده ای که داریم باید هفت همسایه نزدیک را انتخاب بکنیم به علت **k=7** پس با توجه به جدول قبل از میان ۷ همسایه ۳ همسایه متعلق به کلاس "+" خواهد بود و ٤ همسایه متعلق به کلاس "– " خواهد بود. بنابراین نقطه **(1,1)** با ۷ همسایگی متعلق به کلاس "–" خواهد بود بر اساس اکثریت همسایگی.

سوال ٦



با توجه به شکل توزیع داده ها حداقل به ٤ کرنل نیاز داریم برای اینکه در نگاشت به فضای جدید داده ها بتوانند جدایی پذیر خطی باشند. حداقل ۲ کرنل برای ستاره و ۲ کرنل برای دایره های آبی نیاز داریم. نحوه محاسبات به این شکل است که اگر در داخل دایره باشد ۱ میشود در فضای جدید و گرنه صفر میشود. برای

دایره های آبی هم در خوشه سمت راست دقیقا همین شکلی است البته به صورت دقیق باید بیشتر از شعاع کرنل برای ستاره و کمتر از شعاع خود دایره آبی باشد تا در فضای جدید به ۱ نگاشت شود و از ستاره ها و دایره های تو خالی جدا شود.

سوال ۷

همانطور که مشاهده میشود ما ٤ ناحیه در صورت سوال داریم که هر ٤ ناحیه پراکندگی داده ها شبیه به تابع XOR است. میدانیم تابع XOR به صورت ذاتی جدایی پذیر خطی نیست و نمیتوان با یک خط داده ها را از همدیگر تفکیک کرد. مجموعه ای جدایی پذیر خطی خواهد بود که بتوان با ترسیم یک خط به طور خیلی خوبی کلاس ها را از همدیگر جدا کنیم. به صورت واضح مشاهده میشود که این امکان در این مجموعه وجود ندارد.

مزیت شبکه های RBF دقیقا در این مقطع است که به خوبی میتواند داده های غیر خطی را با نگاشت به یک فضای جدید با بعد بزرگتر مساوی از بعد فعلی داده ها به صورت جدایی پذیر خطی تبدیل کند که با یک خط یا یک صفحه بتوان آن را جدا کرد. این مسئله با استفاده از توابع پایه شعاعی ممکن میشود.

نگاشت به یک فضای با بعد بالاتر از طریق اندازه گیری فاصله هر ورودی با مرکز سنجیده میشود. خروجی شبکه مقدار نزدیکی یک داده نسبت به مراکز خواهد بود که باعث میشود داده به فضای جدیدی نگاشت شود با بعد بالاتر.

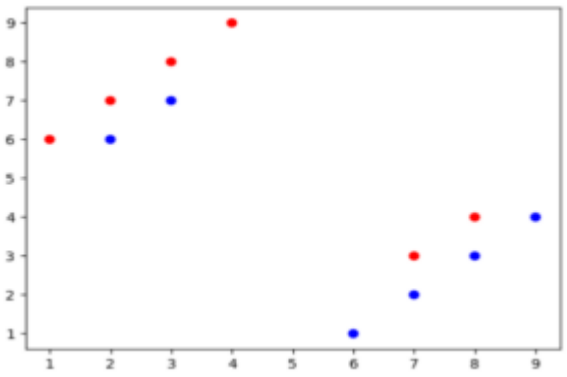
در فضای تبدیل شده با بعد بالاتر حالا داده ها به صورت خطی جدایی پذیر خواهند بود.

پس به صورت کلی داده ها جدایی پذیر خطی نیستند اما با استفاده از RBF و نگاشت به فضای با بعد بالاتر داده جدایی پذیر خطی خواهند بود.

برای هر ناحیه میتوانیم ۲ تابع گوسی استفاده کنیم و نگاشت را انجام بدهیم و در مجموع برای ٤ ناحیه به ۸ تابع گوسی نیاز خواهیم داشت تا نگاشت انجام شود.

سوال ۸

روش **leave one out** یک نوعی از روش های **cross validation** برای ارزیابی میزان یادگیری و عملکرد الگوریتم ما خواهد بود. این روش وقتی استفاده میشود که حجم داده های آموزشی ما کم است. در این روش هر بار یکی از داده ها به عنوان تست از مجموعه آموزشی جدا میشود. یادگیری روی سایر مجموعه صورت میگیرد و در نهایت با اون یک موردی که از مجموعه جدا شده است تست صورت میگیرد. برای یافتن خطا از میانگین تکرار های مختلف را میگیریم. این روش **bias** بسیار پایینی دارد یا ندارد بخاطر اینکه تمام مجموعه داده برای آموزش در هر **iteration** صورت میگیرد. همچنین بر خلاف **k-fold-cross validation**، که باعث وجود پارتیشن بندی تصادفی مجموعه آموزشی میشد این رویکرد **deterministic** خواهد بود به این معنی که هر بار اجرا شود نتیجه یکسان است.



```
k = 1: Error Rate = 0.8333
k = 2: Error Rate = 0.8333
k = 3: Error Rate = 0.6667
k = 4: Error Rate = 0.8333
k = 5: Error Rate = 0.3333
k = 6: Error Rate = 0.3333
k = 7: Error Rate = 0.3333
k = 8: Error Rate = 0.5000
k = 9: Error Rate = 0.3333
k = 10: Error Rate = 0.8333
k = 11: Error Rate = 1.0000
PS C:\Users\Lenovo>
```

با توجه به نقاط داده شده، مقادیر مناسب برای k، ۵، ٦، ۷ و ۹ خواهد بود که کمترین خطا را دارا هستند و مقدار ۱۱ برای k بدترین مقدار خواهد بود.

سوال ۹

در این رویکرد برخلاف رویکرد متداول که در آن اندازه فاصله مهم است، در اینجا زاویه بین ۲ بردار مهم است. در واقع کسینوس اندازه زاویه بین ۲ بردار برای شباهت میسنجیم.

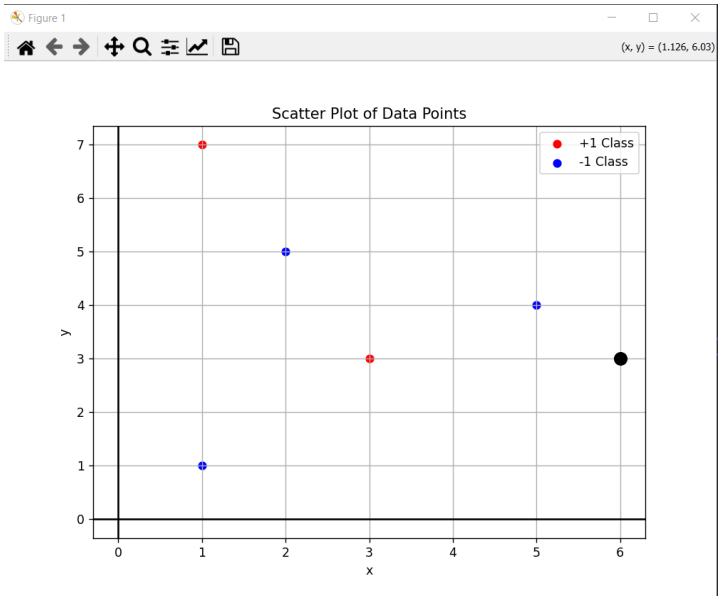
در این رویکرد مقدار ۱ به این معنی هست که دو بردار کاملا شبیه هم هستند. مقدار ۰ به این معنی است که شبیه به هم نیستند (متعامد هستند). مقداری بین ۰ و ۱ درجه شباهت را نشان میدهد.

با توجه به توضیحات گفته شده در انتخاب ۳ همسایگی باید بیشترین مقادیر را برداریم که به ترتیب d1,d2 و d3 خواهند بود با مقادیر 1,0.95 و 0.94. اکثریت این همسایگان را کلاس B تشکل داده است پس نمونه جدید جز کلاس B خواهد بود.

d_1	A	1
d_2	B	0.95
d_3	B	0.94

در انتخاب ۵ همسایگی، اکثریت همسایگان جز کلاس A هستند، پس نمونه جدید جز کلاس A خواهد بود.

سوال ۱۰



نقاط قرمز بیانگر کلاس +۱ و نقاط آبی بیانگر کلاس −۱ هستند. نقطه سیاه، داده خواسته شده در صورت سوال است.

بخش محاسبات:

- **d((6, 3), (1, 1)) = |6 - 1| + |3 - 1| = 5 + 2 = 7**
- **d((6, 3), (1, 7)) = |6 - 1| + |3 - 7| = 5 + 4 = 9**

- **d((6, 3), (3, 3))** = |6 - 3| + |3 - 3| = 3 + 0 = 3
- **d((6, 3), (5, 4))** = |6 - 5| + |3 - 4| = 1 + 1 = 2
- **d((6, 3), (2, 5))** = |6 - 2| + |3 - 5| = 4 + 2 = 6

جدول نقاط و فاصله:

X	Y	Class	Distance to point
1	1	-1	**7**
1	7	+1	**9**
3	3	+1	**3**
5	4	-1	**2**
2	5	-1	**6**

برای در نظر گرفتن ۳ همسایگی نقاط زیر جز نزدیک ترین همسایگی خواهند بود:

- (5, 4) with distance 2 (Class -1)
- (3, 3) with distance 3 (Class +1)
- (2, 5) with distance 6 (Class -1)

با توجه به این ۳ همسایگی، اکثریت را کلاس -۱ تشکیل میدهند پس نقطه **(6,3)** جز کلاس -۱ خواهد بود.

سوال ۱۱

روش های متعددی برای انتخاب همسایگی ها وجود دارد.

۱. روش انتخاب تصادفی:

در این روش ما به صورت تصادفی یک زیر مجموعه از نقاط داده مجموعه آموزشی به عنوان مراکز انتخاب میکنیم. مشکل این روش این است که نمیتواند نماینده خوبی از توزیع حاکم بر داده ها باشد و باعث ضعف در عملکرد شبکه میشود. به عنوان مثال تمام مراکز از یک خوشه در یک گوشه استفاده شود و بقیه نواحی دست نخورده بدون مرکز باقی بمانند.

۲. الگوریتم خوشه بندی **(k-mean)**:

از الگوریتم میانگین **k** روی داده های آموزشی استفاده میکنیم و مراکز پیدا شده توسط این الگوریتم، به عنوان مراکز شبکه در نظر گرفته میشوند. این روش سعی میکند با انتخاب این مراکز تمام فضا را پوشش دهد و نمایندگان بهتری انتخاب بکند. اگر چه از اونجا که این روش تمایل دارد خوشه ها را کروی در نظر بگیرد شاید برای همه توزیع ها خوب نباشد.

۳. گرادیان کاهشی:

اگر قصد داشته باشیم که مراکز هم در حین یادگیری انتخاب شوند مانند وزن ها میتوانیم روی آنها هم یادگیری داشته باشیم با استفاده از گرادیان کاهشی زیرا مقادیر مراکز میتوانند ثابت نباشند و در حین یادگیری تنظیم شوند، عیب این روش این است که پیچیدگی به محاسبات ما اضافه میکند.

توزیع داده ها هر چه قدر پیچیده تر باشد به مراکز بیشتری نیاز داریم.

توزیع داده ها، پیچیدگی داده ها، تعداد نمونه ها، ابعاد ورودی همگی در انتخاب مراکز سهیم هستند. اگر تعداد داده ها خیلی کم باشد میتوانیم همان ها را به عنوان مراکز در نظر بگیریم.