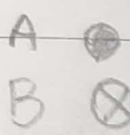
810198358

امیرمهدی انصاری پور

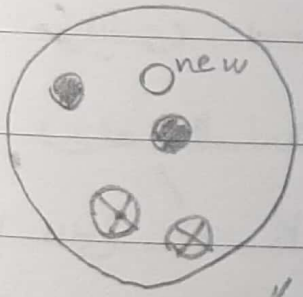
6) مزایای weighted :

در روش weighted، recordهایی که به newrecord نزدیک‌تر هستند وزن بیشتری می‌گیرند و در حالیکه در روش unweighted تمامی همسایه‌های نزدیک وزن‌های یکسانی دارند.

و دور وجود ندارد (همانند شکل زیر) وزن new مانع از فاصله‌ای با وزن A، به B predict میشود.)



در روش weighted احتمال بروز tie خیلی کمتر از unweighted است.



در شکل زیر، وزن new در روش weighted کلاس A

میشود اما در روش unweighted، چنین حالتی را tie نامیدیم و new به

کلاس منفی هم هست

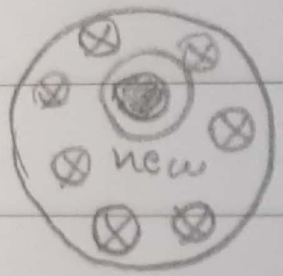
هیچ کدام از کلاس‌های A، B predict نمی‌شود.

ادامه سوال 6 قسمت weighted

البته در بیش weighted هم سلست حالتی پیش بیاید که فاصله یک رکورد

با train یا new record صفر باشد، در نتیجه وزن بی نهایت برای آن یک

رکورد بدست می آید (حاشیه سلل بیرد، کلاس A برای new).



در این حالت بقیه رکوردها در نظر گرفته نمی شوند، هر چند که

فواصلی شان نیز باشد (مثل بیش weighted)

7) الگوریتم DM باید بتواند کلاس‌های rare (کلاس‌های با تعداد نمونه کم) را هم پیش‌بینی کند.

لذا برای همین dataset باید balance شود. همین طور database

هم محدود است و برخی مواقع دسترسی به برخی رکوردها ناممکن است.

9) معایب استفاده از K کوچک: نسبت prediction دقت تاثیر noise، outlier و خطا قرار گیرد. برای مثال اگر K با مساوی 1 انتخاب کنیم، الگوریتم فقط یک رکورد از train را بعنوان نتیجه برمی گرداند (overfitting یا حفظ کردن (memorizing حافظه رخ میدهد)

مزایای استفاده از K کوچک: هزینه محاسباتی الگوریتم در مقایسه با حالتی که K مقادیر بزرگی می گیرد کاهش می یابد (دقت شود KNN الگوریتم instance-based است، یعنی به ازای هر رکورد جدید test، الگوریتم درباره آن میشود.)

10) برای اولویت دادن به ویژگی (attribute)هایی که ارتباط بیشتری با target.variable دارند. زیرا KNN بصورت default همه ویژگی ها را یکسان

ادامه ۱۵) بنابراین امکانش هست که چندین رکورد در attribute های مهم به newrecord بسیار نزدیک باشند، اما در سایر ویژگی های غیر خروجی فاصله داشته باشند. در اینصورت اگر از streck axes برای میترهای مهم رخ ندهد نتیجه prediction اشتباه میشود.

امیرمهدی انصاری پور 810198358

12) با درست کردن dataset از رکورد های R2:R10 و ران کردن الگوریتم KNN به ازای k = 2:

```
R3 = c(0.13, 0.19, 1)
R4 = c(0.65, 0, 1)
R5 = c(0.06, 0.92, 0)
R6 = c(0.38, 0.39, 0)
R7 = c(0.72, 0.19, 0)
R8 = c(0.75, 1, 1)
R9 = c(0.63, 0.9, 1)
R10 = c(1, 0.48, 1)

data = rbind(R2, R3, R4, R5, R6, R7, R8, R9, R10)
dimnames(data) = list(c("R2 (Bad loss)", "R3 (Bad loss)", "R4 (Bad loss)", "R5 (Bad loss)", "R6 (Good ri
                        ,
                        c("Age(MMN)", "Income(MMN)", "Marital(MMN)"))

trueClass = c("Bad loss", "Bad loss", "Bad loss", "Bad loss", "Good risk", "Good risk", "Good risk", "Go

knn(data, new, cl= trueClass, k = 2, prob = TRUE)
```

```
> data
                Age(MMN) Income(MMN) Marital(MMN)
R2 (Bad loss)       0.25        0.01            1
R3 (Bad loss)       0.13        0.19            1
R4 (Bad loss)       0.65        0.00            1
R5 (Bad loss)       0.06        0.92            0
R6 (Good risk)      0.38        0.39            0
R7 (Good risk)      0.72        0.19            0
R8 (Good risk)      0.75        1.00            1
R9 (Good risk)      0.63        0.90            1
R10 (Good risk)     1.00        0.48            1
> knn(data, new, cl= trueClass, k = 2, prob = TRUE)
[1] Good risk
attr(,"prob")
[1] 0.5
Levels: Bad loss Good risk
>
```

همانطور که دیده میشود، کلاس رکورد 1 Good risk پیش بینی شده است.