

Quiz — 3 — Answer

MIR — Spring 2026

Instructor: Dr. Amin Eskandari

Head Teaching Assistants: Hamid Namjoo, Amir Hossein Hemati

Teaching Assistants: Ali Mojahed, Ali Nikvan, Amir Javvi

پاسخ سوال 1

بخش الف) بررسی امکان‌پذیری وجود چنین مدلی

گام ۱: تعریف متغیرهای ماتریس درهم‌ریختگی (Confusion Matrix)

در مسائل بازیابی اطلاعات و طبقه‌بندی دودویی چهار کمیت اصلی داریم:

TP : تعداد اسناد مرتبط که درست به عنوان مرتبط پیش‌بینی شده‌اند

FN : تعداد اسناد مرتبط که اشتباه نامرتبب پیش‌بینی شده‌اند

FP : تعداد اسناد نامرتبب که اشتباه مرتبط پیش‌بینی شده‌اند

TN : تعداد اسناد نامرتبب که درست نامرتبب پیش‌بینی شده‌اند

همچنین:

$$P = TP + FN = 500 \quad (\text{کل اسناد مرتبط})$$

تعداد اسناد نامرتبب:

$$N - P = TN + FP$$

و کل اسناد:

$$N = TP + TN + FP + FN$$

گام ۲: استفاده از مقدار Recall (یا TPR)

Recall یا True Positive Rate برابر است با:

$$TPR = TP / P$$

در صورت سؤال:

$$TPR = 0.80$$

$$P = 500$$

بنابراین:

$$TP = 0.80 \times 500 = 400$$

$$FN = 500 - 400 = 100$$

پس در این سیستم:

$$TP = 400$$

$$FN = 100$$

گام ۳: استفاده از مقدار TNR

TNR یا True Negative Rate:

$$TNR = TN / (TN + FP)$$

طبق صورت سؤال:

$$TNR = 0.96$$

بنابراین:

$$TN / (TN + FP) = 0.96$$

اگر تعداد کل نمونه‌های منفی را با M نشان دهیم:

$$M = N - P$$

پس:

$$TN = 0.96 M$$

$$FP = 0.04 M$$

گام ۴: استفاده از Accuracy

فرمول Accuracy:

$$\text{Accuracy} = (TP + TN) / N$$

طبق سؤال:

$$\text{Accuracy} = 0.95$$

جایگذاری مقادیر:

$$TP = 400$$

$$TN = 0.96M$$

$$N = P + M = 500 + M$$

پس:

$$= 0.95(400 + 0.96M) / (500 + M)$$

گام ۵: حل معادله

$$0.95(500 + M) = 400 + 0.96M$$

$$0.95M = 400 + 0.96M + 475$$

$$0.01M = 75$$

$$M = 7500$$

پس:

$$\text{تعداد اسناد نامرتب} = 7500$$

بنابراین:

$$N = 500 + 7500 = 8000$$

گام ۶: ساخت ماتریس درهم‌ریختگی سازگار

حالا همه مقادیر را محاسبه می‌کنیم

از قبل داریم:

$$TP = 400$$

$$FN = 100$$

همچنین:

$$TN = 0.96 \times 7500 = 7200$$

$$FP = 0.04 \times 7500 = 300$$

پس ماتریس سازگار به صورت زیر است:

$$TP = 400$$

$$FN = 100$$

$$FP = 300$$

$$TN = 7200$$

گام ۷: بررسی Precision

فرمول Precision:

$$\text{Precision} = TP / (TP + FP)$$

$$\text{Precision} = 400 / (400 + 300)$$

$$\text{Precision} = 400 / 700$$

$$\text{Precision} \approx 0.571$$

اما در صورت سؤال گفته شده:

$$\text{Precision} > 0.9$$

پس این مقدار با شرط مسئله سازگار نیست.

گام ۸: بررسی شرط $FPR < FNR$

FPR:

$$FPR = FP / (FP + TN)$$

$$FPR = 300 / 7500 = 0.04$$

FNR:

$$FNR = FN / (FN + TP)$$

$$FNR = 100 / 500 = 0.20$$

در نتیجه:

$$FPR = 0.04$$

$$FNR = 0.20$$

بنابراین:

$$FPR < FNR$$

این شرط برقرار است.

گام ۹: نتیجه‌گیری بخش الف

از سه متریک زیر:

$$TPR = 0.80$$

$$TNR = 0.96$$

$$Accuracy = 0.95$$

مقدار N به طور یکتا برابر 8000 به دست می‌آید.

اما در این حالت Precision برابر 0.571 می‌شود که با شرط:

$$\text{Precision} > 0.9$$

در تضاد است!

پس هیچ ماتریس درهم‌ریختگی‌ای وجود ندارد که تمام شرایط زیر را همزمان داشته باشد:

$$\text{Accuracy} = 0.95$$

$$\text{TPR} = 0.80$$

$$\text{TNR} = 0.96$$

$$\text{Precision} > 0.9$$

$$\text{FPR} < \text{FNR}$$

در نتیجه چنین مدلی اصولاً غیرممکن است و تناقض دقیقاً از شرط Precision ناشی می‌شود!

بخش ب استفاده از MCC برای طراحی تست سازگاری

ضریب هم‌بستگی ماتیو (Matthews Correlation Coefficient) یک معیار جامع است که از هر چهار مقدار ماتریس درهم‌ریختگی استفاده می‌کند.

اگر فقط متریک‌ها گزارش شده باشند (مانند Accuracy، TNR، Recall و Precision)، می‌توان مراحل زیر را انجام داد:

مرحله ۱: بازسازی نسبت‌های ماتریس

از TPR می‌توان نسبت TP و FN را تعیین کرد:

$$\text{TP} = 0.8P$$

$$\text{FN} = 0.2P$$

از TNR نیز نسبت TN و FP مشخص می‌شود:

$$TN = 0.96(N-P)$$

$$FP = 0.04(N-P)$$

مرحله ۲: استفاده از Accuracy برای تخمین نسبت کلاس‌ها
از رابطه:

$$Accuracy = (TP + TN) / N$$

می‌توان نسبت بین تعداد نمونه‌های مثبت و منفی را استخراج کرد.
در این مثال این رابطه منجر شد به:

$$N = 8000$$

مرحله ۳: محاسبه MCC از متریک‌های گزارش شده

پس از بازسازی مقادیر تقریبی FN، FP، TP و TN می‌توان MCC را محاسبه کرد.

اگر MCC محاسبه شده با Precision گزارش شده ناسازگار باشد، نشان می‌دهد که گزارش متریک‌ها
غیرممکن است!

مرحله ۴: بررسی سازگاری Precision

پس از بازسازی ماتریس، Precision واقعی محاسبه می‌شود:

$$Precision = TP / (TP + FP)$$

اگر مقدار به دست آمده با Precision گزارش شده اختلاف جدی داشته باشد، تناقض در گزارش وجود
دارد.

مرحله ۵: نتیجه تست

بنابراین یک تست سازگاری می‌تواند به شکل زیر طراحی شود:

1. بازسازی ماتریس درهم‌ریختگی از TNR و TPR

2. استفاده از Accuracy برای تعیین نسبت کلاس‌ها

3. محاسبه Precision و MCC

4. مقایسه با مقادیر گزارش شده

اگر هیچ ماتریسی نتواند همه متریک‌ها را همزمان وریفای کند، گزارش متریک‌ها ناسازگار است! در مثال این سوال، بازسازی ماتریس نشان می‌دهد Precision واقعی حدود 0.57 است، در حالی که گزارش شده $Precision > 0.9$ است بنابراین تست سازگاری بلافاصله تناقض را آشکار می‌کند.

پاسخ سوال 2

بخش الف) بررسی این‌که آیا می‌توان Accuracy دو مدل A و B را فقط با دانستن متریک‌های:

TPR , FPR , Precision

به صورت تابعی از نسبت کلاس مثبت

$$\pi = P / N$$

نوشت یا نه

و این‌که آیا ممکن است در بعضی π ‌ها تقریباً برابر و در بعضی π ‌ها خیلی متفاوت باشند.

گام ۱: تعریف نسبت مثبت π

ما می‌خواهیم همه چیز را فقط بر حسب نسبت کلاس مثبت بنویسیم:

$$\pi = P / N$$

پس:

$$P = \pi N$$

$$N - P = (1 - \pi)N$$

گام ۲: نوشتن Accuracy به صورت تابعی از π

فرمول Accuracy:

$$\text{Accuracy} = (TP + TN) / N$$

اما:

$$TP = TPR \times P = TPR \times \pi N$$

$$TN = (1 - FPR)(N - P) = (1 - FPR)(1 - \pi)N$$

جایگذاری:

$$\text{Accuracy} = [TPR \times \pi N + (1 - FPR)(1 - \pi)N] / N$$

N حذف می‌شود:

$$\text{Accuracy}(\pi) = TPR \times \pi + (1 - FPR)(1 - \pi)$$

این فرمول برای هر مدل معتبر است.

گام ۳: نوشتن Accuracy برای مدل A و B

مدل A:

$$TPR_A = 0.20$$

$$FPR_A = 0.02$$

بنابراین:

$$\text{Accuracy}_A(\pi) = 0.20\pi + (1 - 0.02)(1 - \pi)$$

$$\text{Accuracy}_A(\pi) = 0.20\pi + 0.98(1 - \pi)$$

$$\text{Accuracy}_A(\pi) = 0.20\pi + 0.98 - 0.98\pi$$

$$\text{Accuracy}_A(\pi) = 0.98 - 0.78\pi$$

مدل B:

$$\text{TPR}_B = 0.80$$

$$\text{FPR}_B = 0.30$$

$$\text{Accuracy}_B(\pi) = 0.80\pi + 0.70(1 - \pi)$$

$$\text{Accuracy}_B(\pi) = 0.80\pi + 0.70 - 0.70\pi$$

$$\text{Accuracy}_B(\pi) = 0.70 + 0.10\pi$$

گام ۴: بررسی این که آیا Accuracy دو مدل می تواند در بعضی π ها نزدیک و در بعضی π ها متفاوت باشد

اختلاف Accuracy:

$$\Delta(\pi) = \text{Accuracy}_A - \text{Accuracy}_B$$

$$\Delta(\pi) = (0.98 - 0.78\pi) - (0.70 + 0.10\pi)$$

$$\Delta(\pi) = 0.28 - 0.88\pi$$

اکنون رفتار تابع را بررسی می کنیم:

وقتی π نزدیک صفر باشد (کلاس مثبت بسیار کم):

$$\Delta \approx 0.28$$

یعنی Accuracy مدل A بالاتر است.

وقتی π بزرگ شود:

Δ کم می‌شود و در نقطه‌ای صفر می‌شود.

حل:

$$0.28 - 0.88\pi = 0$$

$$\pi \approx 0.318$$

تفسیر:

اگر نسبت کلاس مثبت حدود 32 درصد باشد، Accuracy دو مدل تقریباً برابر است

اگر π بسیار کوچک باشد، A بهتر است

اگر π بزرگ باشد، B بهتر است

بنابراین:

بله! ممکن است در بعضی نسبت‌های کلاس Accuracy بسیار نزدیک و در بعضی دیگر کاملاً متفاوت باشد.

این جمله در گزارش رسمی فقط در صورتی درست است که $\pi \approx 0.3$ باشد.

گام ۵ (پاسخ بخش ۳ الف):

در چه شرایطی مدل A سود می‌برد و در چه شرایطی مدل B؟

اگر داده بسیار نامتوازن باشد (π کوچک):

مدل A تقریباً همیشه کلاس منفی را درست می‌زند ($FPR = 0.02$ و $TNR = 0.98$)

ولی مدل B مقدار زیادی FP تولید می‌کند ($FPR = 0.30$)

پس $Accuracy A > Accuracy B$

اگر داده نسبتاً متعادل باشد (π بزرگ):

مدل B مثبت‌ها را خیلی بهتر می‌زند ($TPR = 0.80$ در مقابل 0.20)

و سهم FP زیاد دیگر آن‌قدر غالب نیست

پس $Accuracy\ B > Accuracy\ A$

نتیجه بخش الف

$Accuracy$ دو مدل کاملاً وابسته به π است

• در π نزدیک 0: مدل A برتر است

• در π نزدیک 0.3: شبیه هم می‌شوند

• در π نزدیک 1: مدل B برتر است

پس بله، جمله « $Accuracy$ دو مدل تقریباً برابر است» فقط تحت شرایط خاصی درست است و در بسیاری نسبت‌های دیگر کاملاً غلط است.

بخش ب

تحلیل کیفی درباره‌ی رفتار MCC در بازه‌های مختلف $\pi = P/N$

(بدون محاسبه عددی دقیق)

گام ۱: درک شهودی MCC

MCC حساس است به:

• هر چهار مقدار TP , TN , FP , FN

• توازن بین FP و FN

• توزیع کلاس‌ها

MCC برخلاف Accuracy مدل‌های «همه چیز منفی» یا «همه چیز مثبت» را تنبیه می‌کند.

گام ۲: وقتی π خیلی کوچک است (مثلاً $\pi = 0.01$)

در این حالت داده شدیداً نامتوازن است:

بخش عمده N-P بسیار بزرگ است $\leftarrow$ تصمیم‌گیری روی منفی‌ها مهم‌تر می‌شود.

مقایسه A و B:

مدل A

$FPR_A = 0.02$ (خیلی کم)

$TPR_A = 0.20$ (کم ولی مهم نیست چون P کم است)

مدل B

$FPR_B = 0.30$ (خیلی زیاد $\rightarrow$ FP بسیار زیاد)

$TPR_B = 0.80$

چون P خیلی کم است، FN و TP سهم کمی در MCC دارند.

اما FP بسیار زیاد مدل B، ساختار ماتریس را به هم می‌ریزد.

نتیجه:

در π های خیلی کوچک، مدل A به‌طور ساختاری MCC بالاتری خواهد داشت.

گام ۳: وقتی π نسبتاً متعادل است (مثلاً 0.3 تا 0.6)

اینجا سهم TP و FN بزرگ‌تر می‌شود:

مدل B

TP زیاد $\rightarrow TPR = 0.80$

بزرگ ولی منفی‌ها کمتر از حالت قبلی هستند $\rightarrow FPR = 0.30$

مدل A

TP کم $\rightarrow TPR = 0.20$

اما منفی‌ها حجم کمتری دارند $\rightarrow FPR = 0.02$

در این حالت افزایش TP برای مدل B بسیار مهم‌تر است و باعث افزایش MCC می‌شود.

نتیجه:

در داده‌های نسبتاً متعادل، مدل B پتانسیل MCC بالاتری دارد.

گام ۴: چرا ممکن است مدلی با Accuracy پایین‌تر، MCC بالاتر داشته باشد؟

مثال:

در داده‌های متعادل، مدل A:

TPR بسیار پایین دارد $\ll FN$ زیاد

در نتیجه اگرچه Accuracy ممکن است بد نباشد (به خاطر TN بالا)، ولی FN زیاد MCC را خراب می‌کند!

در همان داده‌ها مدل B:

FPR بالاست و Accuracy را پایین می‌آورد

اما TPR بالا باعث می‌شود TP زیاد باشد

و این توازن FP/FN موجب MCC بالاتر می‌شود

منظور:

Accuracy فقط مجموع تصمیم‌های درست را می‌سنجد، اما MCC وابسته به ساختار خطاست:

اگر یک مدل FP زیاد ولی FN کم داشته باشد (یا برعکس)، Accuracy ممکن است به صورت گمراه‌کننده بالا/پایین شود، اما MCC دقیق‌تر رفتار را نشان می‌دهد.

جمع‌بندی نهایی بخش ب

- در π بسیار کوچک <<< مدل A به طور ساختاری MCC بهتری دارد (چون FP‌های B زیاد است)
- در π متوسط و داده‌های نسبتاً متعادل <<< مدل MCC B بهتری خواهد داشت (چون TP‌های آن بسیار بیشتر است)
- ممکن است مدل Accuracy B پایین‌تری داشته باشد ولی MCC بالاتر، چون MCC ساختار FP/FN را متعادل لحاظ می‌کند

پاسخ سوال 3

بخش الف) نشان دادن این‌که می‌توان متریک‌های ظاهراً عالی داشت اما در عمل عملکرد مدل بسیار بد باشد.

داده‌ها:

نسبت کلاس مثبت (اسپم):

$$\pi = 0.2$$

یعنی:

۲۰٪ ایمیل‌ها اسپم هستند

۸۰٪ ایمیل‌ها غیر اسپم هستند

در گزارش مدل آمده است:

$$\text{Accuracy} > 0.95$$

$$\text{Precision} > 0.9$$

$$\text{NPV} > 0.95$$

اما کاربران می‌گویند ایمیل‌های مهم زیادی از دست می‌روند!
این یعنی تعداد زیادی از نمونه‌های واقعی یک کلاس به اشتباه طبقه‌بندی شده‌اند.

گام ۱: درک معنی FNR

FNR برابر است با:

$$\text{FNR} = \text{FN} / (\text{TP} + \text{FN})$$

یعنی چه نسبتی از نمونه‌های مثبت واقعی توسط مدل از دست رفته‌اند
اگر FNR بالا باشد یعنی:

مدل بخش بزرگی از کلاس مثبت را تشخیص نمی‌دهد

گام ۲: ساختار کیفی یک خانواده از confusion matrix ها
با توجه به اینکه:

$$\pi = 0.2$$

اگر تعداد کل نمونه‌ها N باشد:

$$P = 0.2N$$

$$N - P = 0.8N$$

اکنون ساختاری طراحی می‌کنیم که در آن:

TN بسیار بزرگ

FP بسیار کوچک

TP بسیار کوچک

FN بسیار بزرگ

یعنی مدل تقریباً همه چیز را «منفی» پیش‌بینی می‌کند

به صورت کیفی:

TP = مقدار کم

FN = مقدار بسیار زیاد

TN = مقدار بسیار زیاد

FP = مقدار بسیار کم

در واقع مدل بیشتر ایمیل‌ها را غیر اسپم اعلام می‌کند

گام ۳: چرا Accuracy در این حالت بالا می‌ماند

فرمول Accuracy:

$$\text{Accuracy} = (TP + TN) / N$$

چون:

۸۰٪ داده‌ها منفی هستند

اگر مدل اکثر آن‌ها را درست تشخیص دهد، TN بسیار زیاد می‌شود

در نتیجه:

TN بزرگ <<< Accuracy بالا

حتی اگر بخش بزرگی از مثبت‌ها از دست بروند

گام ۴: چرا Precision می‌تواند بالا بماند

فرمول Precision:

$$\text{Precision} = TP / (TP + FP)$$

اگر مدل فقط تعداد بسیار کمی ایمیل را اسپم اعلام کند، آن‌ها معمولاً واقعاً اسپم خواهند بود
یعنی:

FP بسیار کم

در نتیجه:

Precision بالا می‌ماند.

پس حتی اگر:

TP کم باشد

تا زمانی که FP خیلی کم باشد، Precision بزرگ خواهد بود

گام ۵: چرا NPV نیز بالا می‌ماند

فرمول NPV:

$$NPV = TN / (TN + FN)$$

چون:

TN بسیار زیاد است (به خاطر فراوانی کلاس منفی)

حتی اگر FN زیاد باشد، باز هم نسبت TN غالب است.

در نتیجه:

NPV می‌تواند همچنان بزرگ‌تر از 0.95 باقی بماند

گام ۶: نتیجه ساختار این خانواده از ماتریس‌ها

خانواده‌ای از ماتریس‌ها وجود دارد که ویژگی‌های زیر را دارند:

TN بسیار بزرگ

FP بسیار کوچک

TP بسیار کوچک

FN بسیار بزرگ

در این حالت:

Accuracy بالا

Precision بالا

NPV بالا

اما:

FNR بسیار بالا

یعنی مدل بخش بزرگی از نمونه‌های مثبت را از دست می‌دهد

این دقیقاً همان چیزی است که کاربران گزارش داده‌اند

گام ۷: چرا MCC در این حالت پایین خواهد بود

MCC از هر چهار بخش ماتریس استفاده می‌کند

در سناریوی بالا:

TP بسیار کم

FN بسیار زیاد

در نتیجه:

$TP \times TN$ مقدار نسبتاً کوچک می‌شود

در طرف مقابل:

$FP \times FN$ ممکن است قابل توجه شود

همچنین وجود FN زیاد باعث می‌شود مخرج کسر بزرگ شود

نتیجه:

مقدار MCC کوچک می‌شود

بنابراین:

حتی اگر Accuracy و Precision و NPV بالا باشند، MCC نشان می‌دهد که مدل واقعاً متوازن و قابل اعتماد نیست.

نتیجه بخش الف

می‌توان مدل‌هایی ساخت که:

Accuracy بالا

Precision بالا

NPV بالا

دارند، اما:

FNR بسیار بزرگ است و بسیاری از نمونه‌های مثبت از دست می‌روند

در چنین شرایطی MCC معمولاً مقدار پایینی خواهد داشت و نشان می‌دهد مدل عملکرد واقعی خوبی ندارد

بخش ب) طراحی یک معیار ارزیابی ساده برای جلوگیری از چنین تقلب‌هایی

هدف:

ساخت یک تابع ارزیابی که:

مدل‌های نامتوازن را جریمه کند

مدل‌های متعادل را تشویق کند

گام ۱: متریک‌هایی که باید وزن بالایی داشته باشند

مهم‌ترین متریک‌ها برای جلوگیری از تقلب:

TPR (Recall)

TNR

FNR

FPR

MCC

دلیل:

این متریک‌ها مستقیماً رفتار مدل روی هر دو کلاس را نشان می‌دهند

اگر یکی از کلاس‌ها نادیده گرفته شود، فوراً مشخص می‌شود

گام ۲: متریک‌هایی که می‌توانند وزن کم‌تری داشته باشند

متریک‌هایی که می‌توانند همراه‌کننده باشند:

Accuracy

Precision

NPV

چرا؟

چون در داده‌های نامتوازن ممکن است این مقادیر بالا باشند حتی اگر مدل بسیار بد عمل کند

گام ۳: ایده یک تابع ارزیابی کیفی

یک معیار ساده می‌تواند چنین ساختاری داشته باشد:

امتیاز کلی بالا زمانی داده شود که:

TPR بالا باشد

TNR بالا باشد

MCC بالا باشد

و در عین حال:

FNR و FPR کوچک باشند.

در عمل می توان گفت:

امتیاز مدل بر اساس میانگین یا ترکیب متریک های زیر تعیین شود:

TPR

TNR

MCC

و سپس جریمه هایی اعمال شود اگر:

FNR از یک حد مشخص بیشتر شود

یا FPR از حد مشخصی بیشتر شود

گام ۴: جریمه مدل هایی با FNR بالا

اگر:

FNR بسیار بالا باشد

یعنی:

مدل بخش بزرگی از کلاس مثبت را از دست داده است.

در چنین حالتی حتی اگر:

Accuracy بالا باشد

Precision بالا باشد

مدل باید امتیاز پایینی بگیرد

بنابراین تابع ارزیابی باید به شکل زیر عمل کند:

اگر FNR بزرگ باشد << امتیاز کل شدیداً کاهش یابد

گام ۵: جریمه مدل‌هایی با FPR بالا

اگر:

FPR بزرگ باشد

یعنی مدل تعداد زیادی از نمونه‌های منفی را اشتباه مثبت تشخیص می‌دهد

این مسئله در بسیاری از سیستم‌ها (مثل اسپم فیلتر) مشکل جدی ایجاد می‌کند

بنابراین:

تابع ارزیابی باید FPR بالا را نیز به شدت جریمه کند

گام ۶: جریمه مدل‌هایی با MCC پایین

اگر:

MCC پایین باشد

یعنی مدل بین کلاس‌ها همبستگی واقعی خوبی ندارد

در این حالت ممکن است:

Accuracy ظاهراً خوب باشد

اما ساختار خطاها نامتعادل باشد.

پس:

MCC پایین باید باعث کاهش قابل توجه امتیاز مدل شود

جمع‌بندی نهایی

برای جلوگیری از تقلب در گزارش عملکرد مدل:

متریک‌های زیر باید بیشترین اهمیت را داشته باشند:

TPR

TNR

MCC

FNR

FPR

و متریک‌های زیر باید نقش ثانویه داشته باشند:

Accuracy

Precision

NPV

چنین معیار ارزیابی‌ای باعث می‌شود:

مدل‌هایی که یک کلاس را نادیده می‌گیرند (FNR بالا)

یا خطای زیادی روی کلاس دیگر دارند (FPR بالا)

امتیاز پایینی بگیرند، حتی اگر Accuracy و Precision ظاهراً بسیار خوب باشند!

پاسخ سوال 4

(الف)

$$B_f = \left\lfloor \frac{B}{R} \right\rfloor = \left\lfloor \frac{700}{150} \right\rfloor = 4$$

$$rba = \left\lfloor \frac{i \times R}{B} \right\rfloor = \left\lfloor \frac{37 \times 150}{700} \right\rfloor = \left\lfloor \frac{5550}{700} \right\rfloor = 7$$

$$RBA_{REC} = RBA_{BOF} + rba = 8 + 7 = 15$$

ب) رکورد ۳۷ از بایت ۵۵۵۰ تا ۵۶۹۹ (طول ۱۵۰ بایت) گسترده است. بلاک ۷ (RBA=15) تا بایت ۶۹۹۹ (۷۰۰-۱×۸) ۵۵۹۹ = دقت RBA=8 بلاک صفر فایل است، بلاک rba=7 یعنی RBA=15. بلاک ۱۵ از بایت ۱۰۵۰۰=۷۰۰×۱۵ تا ۱۱۲۹۹=۱۰۵۰۰+۶۹۹ نه، نشانی‌دهی نسبی بلاک نسبت به شروع فایل شروع فایل در RBA=8 است. بلاک rba=7 یعنی بلاک ۱۵. آفست داخل فایل بلاک ۷ ۷۰۰×۷=۴۹۰۰. رکورد ۳۷ از ۵۵۵۰=۱۵۰×۳۷ شروع می‌شود، یعنی آفست ۴۹۰۰=۵۵۵۰-۴۹۰۰ درون بلاک ۱۵. ۱۵۰ بایت بعدی تا بایت ۵۶۹۹=۵۵۵۰+۱۴۹. بلاک ۱۵ از بایت ۴۹۰۰ تا ۵۵۹۹ را پوشش می‌دهد. ۵۶۹۹ < ۵۵۹۹، پس ۱۰۰ بایت به بلاک بعدی (RBA=16, rba=8) می‌ریزد. بنابراین رکورد در دو بلاک قرار دارد. برای خواندن کامل رکورد، سیستم باید دو بلاک RBA=15 و RBA=16 را بخواند ← دو عمل I/O می‌خواهیم.

ج) مطابق جدول ۲-۳ کتاب روحانی رانکوهی، برنامه باید از دستور READ (یا معادل آن) استفاده کند که به طور خودکار از موقعیت جاری می‌خواند. سیستم فایل منطقی مراحل زیر را طی می‌کند:

1. بررسی مجوز دسترسی.
2. بررسی صحت پارامترها.
3. بررسی وجود رکورد در بافر (اگر بلاک مورد نظر قبلاً خوانده شده باشد).
4. در غیر این صورت، تبدیل آدرس نسبی رکورد (۳۸) به RBA و درخواست خواندن بلاک از سیستم فایل فیزیکی.

(د) جوابش در ویس ریکورد شده هست. ([لینک](#))