رگرسیون لجستیک

به‌همراه مثال‌های کاربردی، توضیحات آماری و کد پایتون

نگارنده: آرمان موجودی

برای دانلود و یا اجرای آنلاین کد پایتون این آموزش به آدرس گیتهاب آموزشها مراجعه کنید.

https://github.com/ML-OilGas/ML_Algorithms/Logistic_Regression

ارتباط با نگارنده و ارائه‌ی نظرات و پیشنهادات:

https://www.linkedin.com/in/arman-mojoodi

به او...

از صبا پرس که ما را همه شب تا دم صبح

بوی زلف تو همان مونس جان است که بود

مقدمه

در آموزش‌های رگرسیون خطی ساده و چندگانه، متغیر خروجی یک مقدار عددی بود و هدف، پیش‌بینی این مقدار بر اساس متغیرهای ورودی بود. با این حال، در بسیاری از مسائل عملی، خروجی یک مقدار عددی نیست، بلکه تنها مشخص می‌کند که هر نمونه به کدام گروه یا کلاس تعلق دارد. به چنین مسائلی، **طبقه‌بندی**[1] گفته می‌شود. برای مثال، تشخیص اینکه یک چاه نفتی است یا گازی، یا تعیین اینکه یک بیمار به جراحی، فیزیوتراپی یا استراحت پزشکی نیاز دارد، همگی نمونه‌هایی از مسائل طبقه‌بندی هستند. در این مسائل، هر مقدار ممکن برای خروجی را یک **کلاس**[2] یا **طبقه** می‌نامند بنابراین در مورد مثال چاه نفتی/گازی، مسئله دو کلاسه است؛ یک کلاس یا طبقه چاه نفتی، کلاس دیگر چاه گازی.

در نگاه اول شاید به نظر برسد که بتوان این کلاس‌ها را با اعداد جایگزین کرد و سپس از رگرسیون خطی استفاده نمود. به عنوان مثال می‌توان چاه نفتی را با شماره ۱ و چاه گازی را با شماره ۲ کدگذاری کرد. و یا در دیتاست درمانی، نیازمند جراحی شماره ۱، فیزیوتراپی شماره ۲ و استراحت پزشکی شماره ۳. اما این رویکرد چند مشکل ایجاد می‌کند:

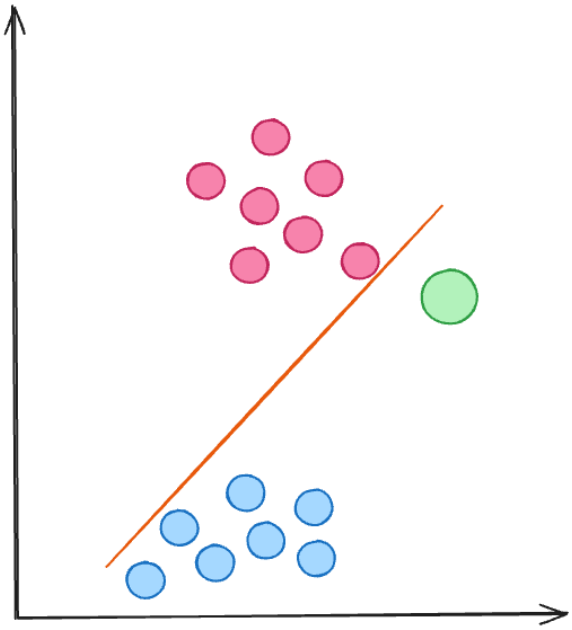
۱- ترتیب کدگذاری مخصوصا برای حالت چند کلاسه در مدل نهایی موثر است. مثلا در دیتاست درمانی مورد بحث، حالت نیازمند جراحی شماره ۱، فیزیوتراپی شماره ۲، استراحت پزشکی شماره ۳ با حالت نیازمند جراحی شماره ۳، فیزیوتراپی شماره ۲، استراحت پزشکی شماره ۱ ممکن است به مدل‌های متفاوتی منجر شوند. علاوه بر این، در این نوع کدگذاری تفاوت حلت نیازمند جراحی با فیزیوتراپی به اندازه تفاوت حالت فیزیوتراپی با استراحت پزشکی درنظر گرفته شده است، هر دو به اندازه ۱ واحد تفاوت دارند. اما آیا واقعا در عمل، تفاوت این کلاس‌ها مشابه هم هستند؟

۲- فارغ از مشکل فوق، اگر کلاس‌ها را با یک خط از هم تفکیک کنیم ممکن است در تعلق نمونه‌ها به هر کلاس خطا رخ دهد. به عنوان مثال به شکل ۱ توجه کنید که دو کلاس آبی و قرمز با یک خط تفکیک شده‌اند. مشاهده می‌کنید که هرچند دایره‌ی سبز به داده‌های قرمز نزدیک‌تر است و انتظار می‌رود به این کلاس تعلق گیرد، اما بر اساس خط رگرسیون در کلاس آبی قرار می‌گیرد.

---

[1] Classification

[2] Class



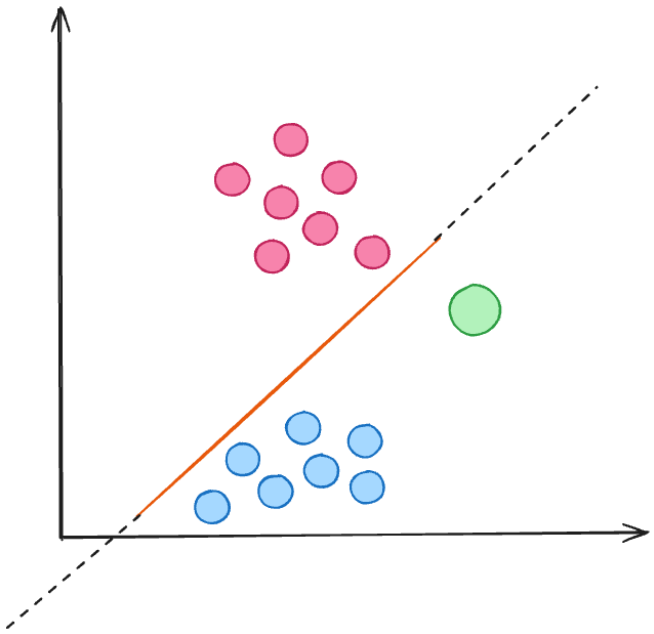
شکل ۱ : تفکیک کلاس‌ها با استفاده از خط و خطا در تعلق نمونه به کلاس

۳- روش رگرسیون خطی براورد صحیحی از احتمال تعلق نمونه به هر کلاس نمی‌دهد.

برای سادگی حالت دو کلاسه را بررسی می‌کنیم. فرض کنید که خروجی شامل دو کلاس k1 و k2 است و می‌خواهیم احتمال تعلق هر نمونه را به یکی از این دو کلاس بدست آوریم. اگر احتمال تعلق هر نمونه به کلاس اول را با استفاده از رگرسیون خطی مدل کنیم، خواهیم داشت.

$$p(x) = \beta_0 + \beta_1 x \qquad (۱)$$

طبق این رابطه، دو کلاس با استفاده از یک خط جدا می‌شوند، مطابق شکل ۲. مشاهده می‌کنید که با امتداد خط رگرسیون که دو کلاس را از هم تفکیک می‌کند احتمال منفی و بیشتر از یک نیز خواهیم داشت که درست نیست.



شکل ۲ : محاسبه احتمال تعلق هر نمونه به کلاس‌ها با استفاده از خط راست

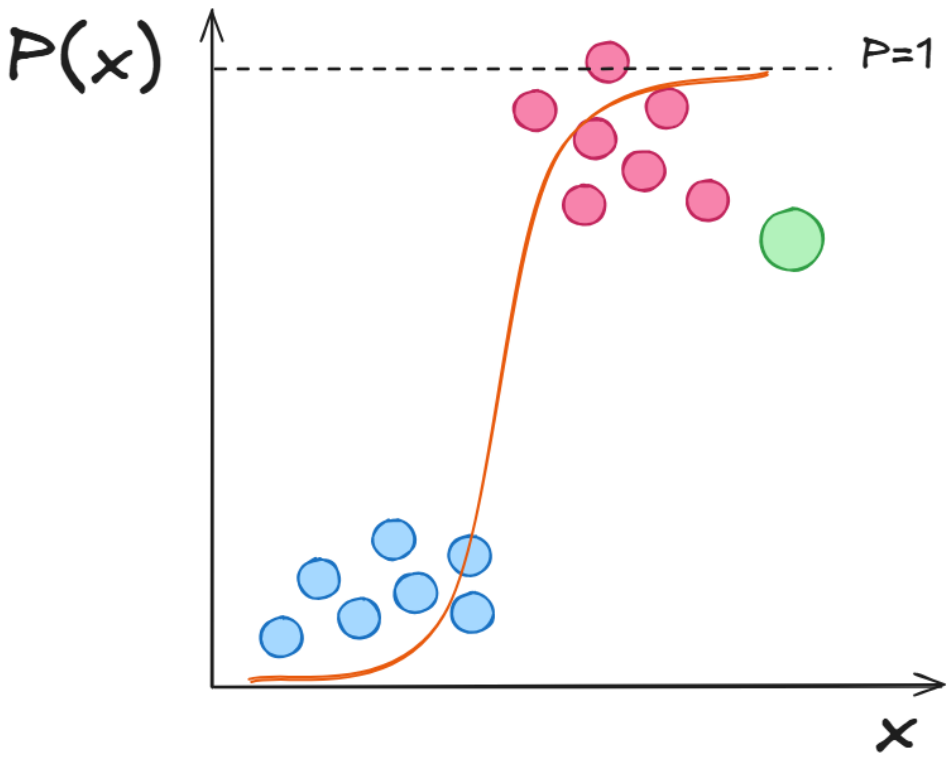
به همین دلیل، برای مسائل طبقه‌بندی از روش‌هایی استفاده می‌شود که مستقیماً برای پیش‌بینی احتمال تعلق نمونه‌ها به کلاس‌ها طراحی شده‌اند. یکی از رایج‌ترین این روش‌ها، **رگرسیون لجستیک** است.

در روش رگرسیون لجستیک[3] بجای خط، احتمال تعلق نمونه به هر کلاس را با استفاده از تابع لجستیک بصورت زیر محاسبه می‌کنیم:

$$p(x) = \frac{e^{\beta_0+\beta_1 x}}{1+e^{\beta_0+\beta_1 x}} \qquad (۲)$$

با استفاده از تابع لجستیک مطمئن هستیم که همواره احتمال بدست آمده بین صفر و یک خواهد بود. تابع لجستیک دارای منحنی S شکل (سیگموئید) است (شکل ۳).

[3] Logistic Regression



شکل ۳ : شمایی از تابع لجستیک

مشاهده می‌کنید که در اینحالت احتمال تعلق به هر کلاس بین دو مقدار صفر و یک است و در اینحالت احتمال تعلق دایره سبز رنگ به کلاس قرمز بیشتر از آبی است.

برای تفسیر بهتر ضرایب مدل، رابطه‌ی (۲) را می‌توان به صورت زیر بازنویسی کرد.

$$\frac{p(x)}{1-p(x)} = e^{\beta_0+\beta_1 x} \qquad (۳)$$

به سمت چپ رابطه‌ی (۳)، $\frac{p(x)}{1-p(x)}$، odds گفته می‌شود. توجه کنید که odds هر مقداری از صفر تا بینهایت را می‌تواند اختیار کند. با لگاریتم گرفتن از رابطه‌ی (۳) به رابطه‌ی زیر میرسیم.

$$\log(odds) = \beta_0 + \beta_1 x \qquad (۴)$$

باتوجه به رابطه‌ی (۴) مشاهده می‌کنید برخلاف رگرسیون خطی، در رگرسیون لجستیک ضرایب مدل مستقیماً بیانگر میزان تغییر احتمال نیستند، بلکه تغییر در لگاریتم Odds را نشان می‌دهند. به عبارت دیگر اثر تغییرات ۱ واحد x بر p(x) برابر با ضریب β_1 نیست. بلکه ضریب β_1 درواقع نشان‌دهنده‌ی اثر تغییرات ۱ واحد x بر لگاریتم odds است. بنابراین براساس ضریب β_1 و باتوجه به علامت مثبت و منفی آن تنها می‌توان گفت که اثر x بر p(x) افزایشی است

یا کاهشـــی. علاوه بر این، باتوجه به رابطه‌ی (۲) و بعلت حضـــور ترم نمایی، مقدار تاثیر ۱ واحد تغییر x بر p(x) به مقدار خود x هم بستگی دارد که این موضوع بصورت شهودی از روی شکل هم مشخص است.

توضـــیحات مربوط به نحوه‌ی تعیین ضـــرایب بتا و تعمیم روابط به حالت چند کلاســـه در بخش جمع‌بندی و نکات تکمیلی بیان شده است.

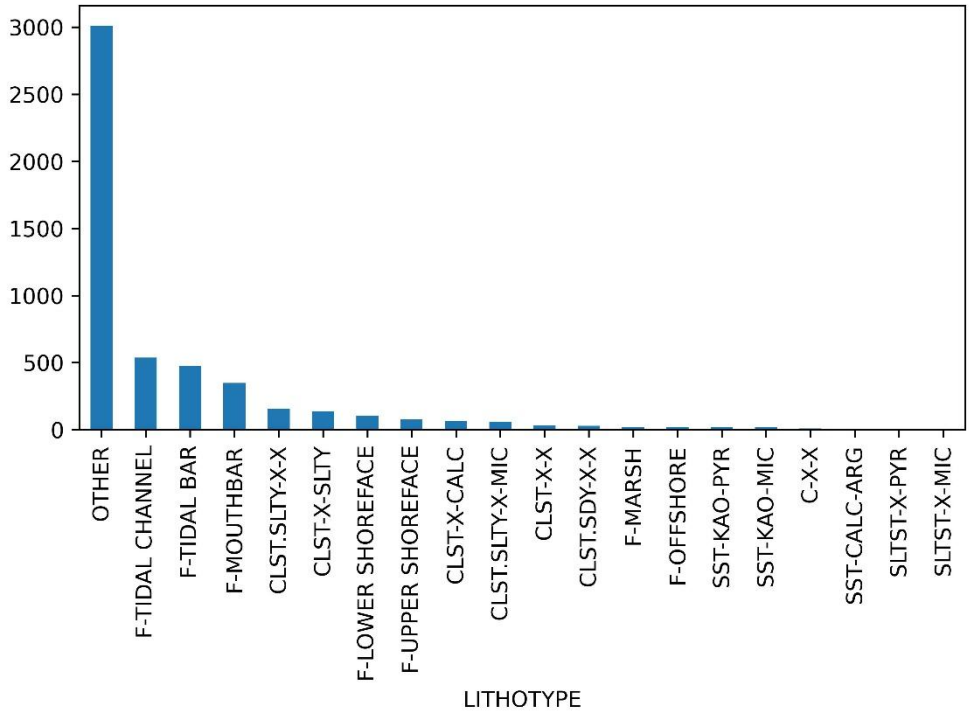
در بخش بعد، برای یک دیتاسـت نفتی واقعی و با اسـتفاده از پایتون، مدل رگرسـیون لجسـتیک محاسـبه و نتایج آن تحلیل می‌شود.

اجرای عملی

در این بخش از مجموعه‌داده‌ی نفتی واقعی **Volve** که در آموزش رگرسیون خطی چندگانه معرفی شد، استفاده می‌کنیم. هدف، پیش‌بینی متغیر **LITHOTYPE** براساس سایر ویژگی‌های پتروفیزیکی و در نتیجه حل یک مسئله‌ی طبقه‌بندی است (رابطه‌ی (5)).

$$LITHOTYPE = f\begin{pmatrix} BVW, CARB_FLAG, COAL_FLAG, PHIF, RHOFL, \\ RHOMA, RW, SAND_FLAG, SW, TEMP, VSH, KLOGH \end{pmatrix} \quad (5)$$

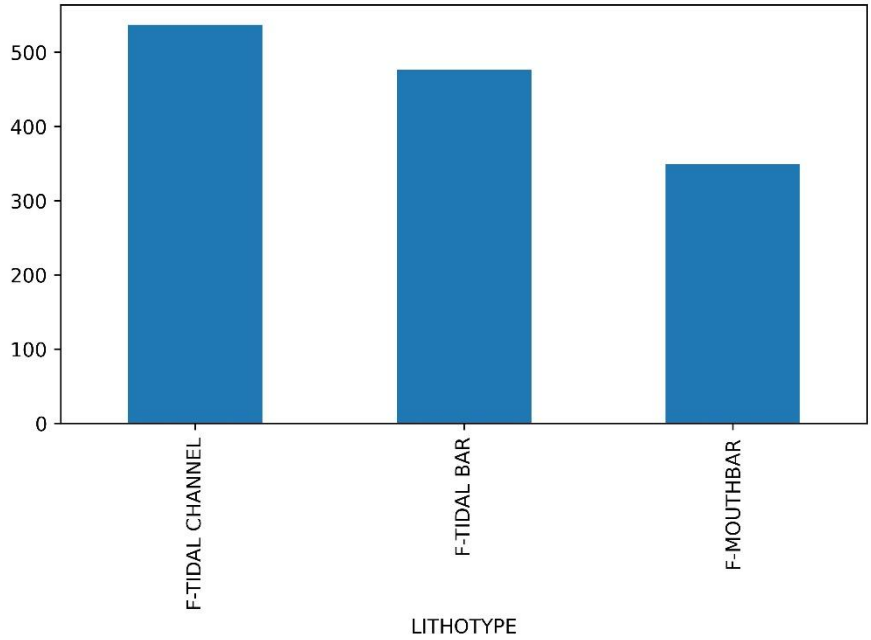
در مجموعه داده‌ی پتروفیزیکی مورد استفاده، چندین کلاس مختلف برای متغیر LITHOTYPE تعریف شده است که در شکل ۴ نمایش داده شده‌اند.



شکل ۴ : دسته‌های مختلف متغیر LITHOTYPE

به‌منظور ساده‌سازی فرآیند تحلیل، تنها کلاس‌هایی نگه داشته شده‌اند که حداقل **۲۵۰ نمونه** دارند و سایر کلاس‌های کم‌نمونه از مجموعه‌داده حذف شده‌اند. همچنین، همان‌گونه که در شکل ۴ مشاهده می‌شود، کلاسی با عنوان OTHER بیشترین فراوانی را دارد، اما اطلاعات زمین‌شناسی معناداری ارائه نمی‌دهد؛ بنابراین این کلاس نیز از مجموعه‌داده حذف شده است. در نتیجه، مدل نهایی تنها بر روی سه کلاس اصلی آموزش داده می‌شود که توزیع آن‌ها در شکل ۵ نشان داده شده است.

در مسائل طبقه‌بندی، اگر متغیر خروجی تنها شامل دو کلاس باشد، مسئله طبقه‌بندی دودویی[4] نامیده می‌شود. اما از آنجا که متغیر LITHOTYPE در این مجموعه‌داده شامل بیش از دو کلاس است، مسئله‌ی حاضر یک **طبقه‌بندی چندکلاسه**[5] محسوب می‌شود.



شکل ۵ : سه کلاس اصلی مورد استفاده برای طبقه‌بندی متغیر LITHOTYPE

برای پیاده‌سازی مدل، از کلاس LogisticRegression موجود در کتابخانه‌ی sklearn استفاده شده است. خروجی اجرای مدل در شکل ۶ نمایش داده شده است.

Classification Report:

	precision	recall	f1-score	support
F-MOUTHBAR	0.40	0.05	0.08	87
F-TIDAL BAR	0.53	0.86	0.66	119
F-TIDAL CHANNEL	0.82	0.85	0.84	135
accuracy			0.65	341
macro avg	0.59	0.58	0.53	341
weighted avg	0.61	0.65	0.58	341

شکل ۶ : خروجی مدل رگرسیون لجستیک

خروجی نمایش‌داده‌شده شامل معیارهای ارزیابی مدل برای هر کلاس و همچنین میانگین عملکرد مدل است که در بخش بعد به تفصیل بررسی خواهند شد.

---

[4] Binary Classification

[5] Multiclass Classification

بحث در نتایج

۱- معیارهای ارزیابی مدل چیست ؟

در مسائل طبقه‌بندی، یکی از ساده‌ترین معیارهای ارزیابی، **دقت کلی**[6] است که برابر است با نسبت تعداد پیش‌بینی‌های صحیح به کل نمونه‌ها. با توجه به تعاریف شکل ۷، رابطه‌ی آن بصورت زیر است.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{6}$$

اگرچه این معیار در نگاه اول مناسب به نظر می‌رسد، اما در بسیاری از مسائل، به‌ویژه زمانی که تعداد نمونه‌های کلاس‌ها با یکدیگر تفاوت زیادی داشته باشد، معیار قابل اعتمادی نیست. برای مثال، فرض کنید در یک مسئله‌ی دودویی، از مجموع یک میلیون نمونه، تنها ۱۰۰ هزار نمونه متعلق به کلاس اول و ۹۰۰ هزار نمونه متعلق به کلاس دوم باشند. در چنین شرایطی، مدلی که تمام نمونه‌ها را متعلق به کلاس دوم پیش‌بینی کند، همچنان دقت کلی برابر با ۹۰ درصد خواهد داشت، در حالی که هیچ‌یک از نمونه‌های کلاس اول را به‌درستی شناسایی نکرده است.

به همین دلیل، برای ارزیابی دقیق‌تر عملکرد مدل، به‌ویژه در مسائل چندکلاسه یا مجموعه‌داده‌های نامتوازن، از ماتریس درهم‌ریختگی[7] استفاده می‌شود. این ماتریس نشان می‌دهد که مدل هر کلاس را با چه دقتی پیش‌بینی کرده و خطاهای آن در کدام کلاس‌ها رخ داده است. در این ماتریس، مقادیر روی قطر اصلی بیانگر پیش‌بینی‌های صحیح و سایر خانه‌ها نشان‌دهنده‌ی خطاهای مدل هستند. نمونه‌ای از ماتریس درهم‌ریختگی برای یک مسئله‌ی طبقه‌بندی دودویی در شکل ۷ نمایش داده شده است.

---

[6] Accuracy
[7] Confusion Matrix

		Predicted class	
		P	N
Actual class	P	TP	FN
	N	FP	TN

شکل ۷ : مثالی از ماتریس درهم‌ریختگی، P مثبت، N منفی، TP مثبت صحیح، FP مثبت کاذب، TN منفی صحیح، FN منفی کاذب

بر اســـاس ماتریس درهم‌ریختگی، معیارهای مختلفی برای ارزیابی عملکرد مدل تعریف می‌شـــوند که مهم‌ترین آن‌ها عبارت‌اند از **دقت**[8]**، بازیابی**[9] **و امتیاز F1** .

دقت

دقت، نســـبت نمونه‌های مثبتی اســـت که به‌درســـتی پیش‌بینی شـــده‌اند به کل نمونه‌هایی که مدل آن‌ها را مثبت تشخیص داده است.

$$Precision = \frac{TP}{TP + FP} \tag{7}$$

دقت بالا نشان می‌دهد که بیشتر پیش‌بینی‌های مثبت مدل، واقعاً صحیح بوده‌اند.

از معیار دقت زمانی استفاده می‌شود که **هزینه‌ی مثبت کاذب (FP)** زیاد باشد؛ یعنی اگر مدل به اشتباه یک نمونه را مثبت تشخیص دهد، پیامدهای نامطلوبی ایجاد شود. برای مثال، فرض کنید از یک مدل طبقه‌بندی برای تشخیص بیمارانی که نیاز به عمل جراحی دارند اســـتفاده می‌شـــود. در این حالت، اعلام اشـــتباه نیاز به جراحی برای یک بیمار می‌تواند منجر به انجام آزمایش‌ها یا اقدامات درمانی غیرضـــروری شـــود؛ بنابراین افزایش دقت اهمیت بیشـــتری پیدا می‌کند.

[8] Precision
[9] Recall

بازیابی

بازیابی، نسبت نمونه‌های مثبتی است که به‌درستی پیش‌بینی شده‌اند به کل نمونه‌هایی که در واقع مثبت بوده‌اند.

$$Recall = \frac{TP}{TP + FN} \tag{8}$$

بازیابی بالا نشان می‌دهد که مدل توانسته بخش بزرگی از نمونه‌های مثبت واقعی را شناسایی کند.

از معیار بازیابی زمانی استفاده می‌شود که **هزینه‌ی منفی کاذب (FN)** زیاد باشد؛ یعنی اگر مدل یک نمونه‌ی مثبت را به اشتباه منفی تشخیص دهد، خسارت قابل توجهی ایجاد شود. برای مثال، در یک سامانه‌ی تشخیص نشت گاز، اگر مدل وجود نشت را تشخیص ندهد، ممکن است خطرات جانی و خسارات مالی سنگینی به همراه داشته باشد. در چنین شرایطی، حتی اگر مدل گاهی هشدار اشتباه صادر کند، معمولاً قابل قبول‌تر از آن است که یک نشت واقعی را تشخیص ندهد.

با مقایسه‌ی روابط دقت و بازیابی مشاهده می‌شود که تفاوت اصلی این دو معیار در مخرج کسر آن‌ها است؛ به همین دلیل، هر یک جنبه‌ی متفاوتی از عملکرد مدل را ارزیابی می‌کنند.

در عمل، معمولاً نمی‌توان دقت و بازیابی را به‌طور هم‌زمان افزایش داد و بهبود یکی از آن‌ها اغلب با کاهش دیگری همراه است. بنابراین، در بسیاری از کاربردها لازم است تعادلی میان این دو معیار برقرار شود.

امتیاز F1

امتیاز F1، میانگین هارمونیک دقت و بازیابی است و زمانی کاربرد دارد که هر دو معیار اهمیت داشته باشند، به‌ویژه در مسائل دارای کلاس‌های نامتوازن.

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{9}$$

در این معیار، تنها زمانی مقدار بالایی حاصل می‌شود که هر دو معیار دقت و بازیابی عملکرد مناسبی داشته باشند؛ بنابراین F1 شاخص مناسبی برای ارزیابی کلی مدل در بسیاری از مسائل طبقه‌بندی محسوب می‌شود.

با توجه به نتایج شکل ۶ مشاهده می‌شود که عملکرد مدل در کلاس **F-TIDAL CHANNEL** بهتر است، در حالی که عملکرد آن برای کلاس‌هایی که تعداد نمونه‌های کمتری دارند ضعیف‌تر است. این موضوع نشان می‌دهد که نامتوازن بودن داده‌ها می‌تواند بر کیفیت پیش‌بینی برخی کلاس‌ها تأثیر بگذارد.

جمع‌بندی و نکات تکمیلی

نحوه‌ی تعیین ضرایب β

در رگرسیون خطی، ضرایب مدل با استفاده از روش کمترین مربعات محاسبه می‌شدند. اما در رگرسیون لجستیک، به دلیل غیرخطی بودن تابع سیگموئید، امکان استفاده از این روش وجود ندارد. به همین دلیل، ضرایب مدل معمولاً با استفاده از روش‌های عددی و تکرارشونده محاسبه می‌شوند. برای تعیین ضرایب β از تابع Likelihood که به آن تابع درست‌نمایی هم گفته می‌شود استفاده می‌کنیم.

هدف از این روش آن است که ضرایب β به گونه‌ای انتخاب شوند که احتمال مشاهده‌ی داده‌های موجود بیشینه شود؛ به عبارت دیگر، برای یک مسئله‌ی دودویی، مدل باید برای نمونه‌هایی که واقعاً به کلاس اول تعلق دارند، احتمال‌هایی نزدیک به ۱ و برای نمونه‌های کلاس دوم، احتمال‌هایی نزدیک به صفر پیش‌بینی کند. تابع Likelihood به صورت زیر تعریف می‌شود.

$$\ell(\beta_0, \beta_1) = \prod_{i:y_i=1} p(x_i) \prod_{i':y_{i'}=0} \left(1 - p(x_{i'})\right) \tag{10}$$

از آنجا که هدف، یافتن ضرایبی است که مقدار این تابع را بیشینه کنند، این روش Maximum Likelihood نامیده می‌شود.

جهت روشن شدن موضوع به مثال ساده‌ی زیر توجه کنید.

فرض کنید سه بیمار داریم که براساس میزان قند خون آنها، به دو دسته‌ی قند خون نرمال (y=0) و قند خون بالا (y=1) تقسیم شده‌اند:

بیمار	قند خون x	واقعیت y
۱	80	0
۲	120	1
۳	150	1

فرض کنید یک مدل با یک سری ضرایب دلخواه ساخته‌ایم. خروجی مدل با این ضرایب بصورت زیر است:

بیمار	واقعیت y	احتمال $P(Y=1)$
۱	0	0.10
۲	1	0.70
۳	1	0.90

این مدل خوب به نظر می‌رسد. چرا؟ چون برای بیماری که واقعاً $y = 1$ بوده، احتمال بالایی داده و برای بیماری که $y = 0$ بوده، احتمال پایینی داده است.

حالا Likelihood چه کار می‌کند؟

برای هر مشاهده می‌پرسیم:

مدل چقدر احتمال داده که **همین چیزی که واقعاً اتفاق افتاده** رخ بدهد؟

برای بیمار اول که واقعاً $y = 0$ است، مدل گفته احتمال $y = 0$:

$$1 - 0.10 = 0.90$$

برای بیمار دوم که واقعاً $y = 1$ است:

$$0.70$$

و برای بیمار سوم:

$$0.90$$

پس Likelihood می‌شود حاصل‌ضرب این احتمال‌ها:

$$L = 0.90 \times 0.70 \times 0.90 = 0.567$$

یعنی این مجموعه ضرایب، با داده‌های واقعی نسبتاً سازگار است.

حالا ضرایب دیگری امتحان کنیم

فرض کنید یک مجموعه ضرایب دیگر امتحان کردیم و خروجی مدل بصورت زیر بدست آمد.

بیمار	واقعیت y	احتمال $P(Y=1)$
۱	0	0.40
۲	1	0.50
۳	1	0.60

حالا احتمال اتفاق واقعی برای هرکدام برابر است با:

$$1 - 0.40 = 0.60,\ 0.50,\ 0.60$$

بنابراین:

$$L = 0.60 \times 0.50 \times 0.60 = 0.18$$

Likelihood این مدل **کمتر** است. پس مدل اول بهتر با داده‌های مشاهده‌شده سازگار بوده است.

بنابراین Maximum Likelihood دقیقاً یعنی چه؟

یعنی:

مجموعه ضرایبی را پیدا کن که Likelihood را بیشینه کند.

یعنی ما مدام دنبال این هستیم:

β_0, β_1 اینگونه انتخاب شوند که $L(\beta_0, \beta_1)$ بیشینه شود.

نکته‌ی خیلی مهم:

ما احتمال واقعی را بیشینه نمی‌کنیم؛ ضرایب مدل را تغییر می‌دهیم تا احتمال داده‌هایی که واقعاً مشاهده کرده‌ایم، بیشترین مقدار را پیدا کند.

تعمیم به حالت چند کلاسه

رگرسیون لجستیک در شکل پایه‌ی خود برای مسائل دودویی طراحی شده است. برای تعمیم این روش به مسائل چندکلاسه، از **تابع Softmax** استفاده می‌شود که احتمال تعلق هر نمونه به تمام کلاس‌ها را به طور هم‌زمان محاسبه می‌کند.

$$\Pr(Y = k \mid X = x) = \frac{e^{\beta_{k0} + \beta_{k1} x_1 + \cdots + \beta_{kp} x_p}}{\sum_{l=1}^{K} e^{\beta_{l0} + \beta_{l1} x_1 + \cdots + \beta_{lp} x_p}} \tag{11}$$

که در آن K تعداد کل کلاس‌ها است.

تابع Softmax از نظر شکل رفتاری مشابه تابع سیگموئید است و خروجی آن نیز به صورت احتمال بیان می‌شود. تفاوت اصلی این دو تابع در آن است که سیگموئید تنها احتمال تعلق یک نمونه به **دو کلاس** را مدل می‌کند، در حالی که Softmax احتمال تعلق همان نمونه را به **تمام کلاس‌ها** به طور هم‌زمان محاسبه و نرمال‌سازی می‌کند، به گونه‌ای که مجموع احتمال‌های به‌دست‌آمده برای همه‌ی کلاس‌ها برابر با یک باشد.