

روش ها و معیارهای ارزیابی الگوریتم یادگیری ماشین

معیار های متنوعی برای ارزیابی کارایی الگوریتم های یادگیری ماشین، الگوریتم های طبقه بندی و رگرسیون وجود دارد. باید در انتخاب معیارهای ارزیابی کارایی ML دقت به خرج دهیم زیرا:

چگونگی اندازه گیری و مقایسه کارایی الگوریتم های ML کاملا به معیارهایی که انتخاب می کنید وابسته است.

چگونگی وزن دادن به اهمیت ویژگی های مختلف در نتایج، کاملا تحت تاثیر معیار هایی است که انتخاب می کنید.

معیارهای کارایی برای مسائل طبقه بندی (classification): طبقه بندی و الگوریتم های مربوط به آن را در بخش های قبلی مورد بررسی قرار دادیم. در اینجا، در باره معیارهای متنوع کارایی که برای ارزیابی پیش بینی ها در مسائل طبقه بندی استفاده می شود را بررسی می کنیم.

ماتریس سردرگمی (confusion matrix): زمانی که خروجی شامل دو نوع کلاس و یا بیشتر باشد، این ساده ترین راه برای اندازه گیری کارایی یک مساله طبقه بندی است. ماتریس سردرگمی چیزی جز یک جدول با دو بعد نیست. مقدار واقعی (actual value) و پیش بینی شده (predicted value). همان طور که در شکل زیر نشان داده شده است، هر دو بعد دارای مثبت-صحیح (TP) ، منفی-صحیح (TN) ، مثبت-غلط (FP) و منفی-غلط (FN) است.

Actual

		1	0
Predicted	1	True Positives (TP)	False Positives (FP)
	0		True Negatives (TN)

اصطلاحات مربوط به ماتریس سردرگمی در ادامه توضیح داده شده است.

مثبت-صحیح (TP): زمانی است که هر دو کلاس واقعی و پیش بینی از نقاط داده ۱ است.

منفی-صحیح (TN): زمانی است که هر دو کلاس واقعی و پیش بینی از نقاط داده صفر است.

مثبت-غلط (FP): زمانی است که کلاس واقعی از نقطه داده صفر و کلاس پیش بینی ۱ است.

منفی-غلط (FN): زمانی است که کلاس واقعی از نقطه داده ۱ و کلاس پیش بینی صفر است.

برای محاسبه ماتریس سردرگمی مربوط به مدل طبقه بندی خود، می توانیم از تابع `confusion_matrix` از `sklearn.metrics` استفاده کنیم.

دقت طبقه بندی (classification accuracy): این متداول ترین معیار کارایی برای الگوریتم های

طبقه بندی است. می توان آن را در قالب تعداد پیش بینی های صحیح صورت گرفته به نسبت همه پیش

بینی های صورت گرفته، تعریف کنیم. با کمک رابطه زیر و با استفاده از ماتریس سردرگمی می توان آن را

محاسبه کرد.

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

برای محاسبه دقت مدل طبقه بندی خود می توانیم از تابع `accuracy_score` از `sklearn.metrics` استفاده کنیم.

گزارش طبقه بندی (classification report): این گزارش شامل امتیاز دقت (Precisions)، یاد آوری (Recall)، F1 و پشتیبانی (Support) است که در ادامه توضیح داده می شوند.

دقت: دقت استفاده شده در بازیابی اسناد می تواند به صورت تعداد اسناد صحیح بازگردانده شده توسط مدل ML ما، تعریف شود. با کمک رابطه زیر و با استفاده از ماتریس سردگمی می توان آن را به سادگی محاسبه کرد.

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

یادآوری (recall) یا حساسیت (sensitivity): یادآوری را می توان در قالب تعداد مثبت های بازگردانده شده توسط مدل ML تعریف کرد. با کمک رابطه زیر و با استفاده از ماتریس سردگمی می توان آن را به سادگی محاسبه کرد.

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

اختصاصی بودن (Specificity): در مقابل یادآوری، اختصاصی بودن را می توان در قالب تعداد منفی های بازگردانده شده توسط الگوریتم ML تعریف کرد. با کمک رابطه زیر و با استفاده از ماتریس سردگمی می توان آن را به سادگی محاسبه کرد.

$$Specificity = \frac{TN}{TN + FP}$$

پشتیبانی: پشتیبانی را می توان در قالب تعداد نمونه های پاسخ صحیح که در هر کلاس از مقادیر هدف قرار می گیرد، تعریف کرد.

امتیاز F1: این امتیاز، میانگین هارمونیک از دقت و یادآوری را به ما می دهد. به بیان ریاضی، امتیاز F1 میانگین وزن دار از دقت و یادآوری است. بهترین مقدار برای F1، ۱ و بدترین مقدار صفر است. با کمک رابطه زیر می توان امتیاز F1 را محاسبه کرد.

$$F1 = 2 * (precision * recall) / (precision + recall)$$

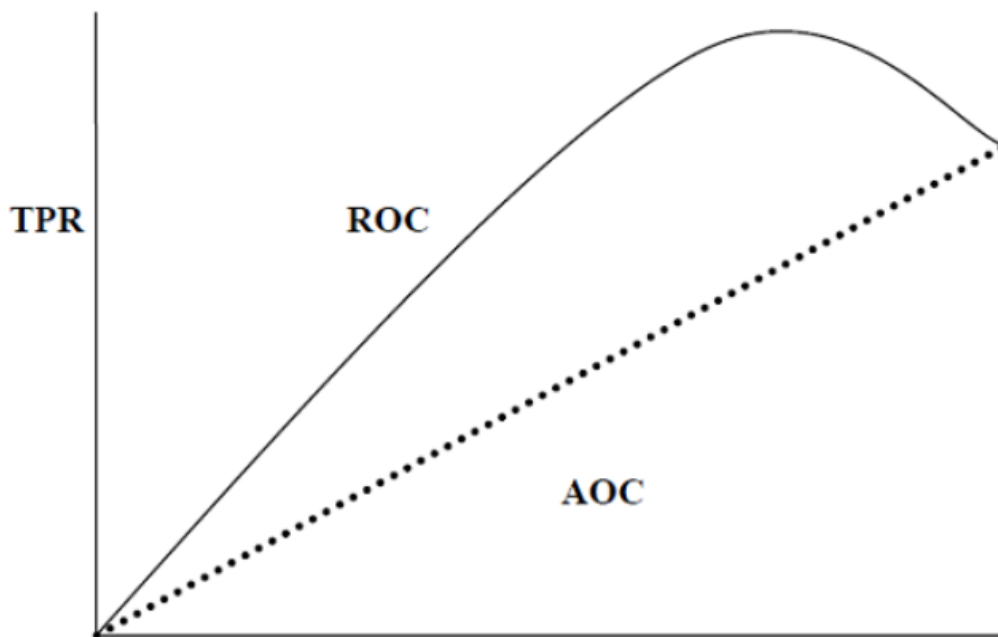
امتیاز F1 یک مشارکت نسبی برابر از دقت و یادآوری دارد. برای دریافت گزارش طبقه بندی از مدل طبقه بندی، می توان از تابع `classification_report` از `sklearn.metrics` استفاده کرد.

مساحت زیر نمودار ROC (Area Under Curve) (AUC): ROC (Receiver Operating Characteristic)

ROC (Receiver Operating Characteristic)– AUC (Area Under Curve) یک معیار کارایی، بر اساس مقادیر آستانه متغیر برای مسائل طبقه بندی است. همانطور که از نام آن بر می آید، ROC یک منحنی احتمالی است و AUC تفکیک پذیری را اندازه گیری می کند. به بیان ساده تر، معیار AUC-ROC، درباره قابلیت مدل در زمینه تشخیص کلاس ها به ما اطلاعات می دهد. هرچه AUC بالاتر باشد، مدل بهتر است.

به بیان ریاضی، با رسم TPR (True Positive Rate)، می توان آن را ساخت یعنی حساسیت یا یادآوری در مقابل FPR (False Positive Rate) یعنی اختصاصی بودن در مقادیر آستانه متنوع. شکل زیر ROC، شامل TPR در محور y و FPR در محور x را نشان می دهد.

Mathematically, it can be created by plotting TPR (True Positive Rate) i.e. Sensitivity or recall vs FPR (False Positive Rate) i.e. 1-Specificity, at various threshold values. Following is the graph showing ROC, AUC having TPR at y-axis (and FPR at x-axis –



برای محاسبه AUC-ROC می توان از تابع `roc_auc_score` از `sklearn.metrics` استفاده کرد.

LOGLOSS(logarithmic loss): همچنین فقدان رگرسیون لجستیک یا فقدان cross-entropy نیز نامیده می شود. به طور کلی روی احتمال تخمین و اندازه گیری کارایی یک مدل طبقه بندی، تعریف می شود که ورودی یک مقدار احتمالی بین صفر و ۱ است. با تمایز دادن آن و دقت، بهتر می توان آن را فهمید. همانطور که می دانیم دقت شمارش پیش بینی (مقدار پیش بینی = مقدار واقعی) های انجام شده در مدل ما است در حالی که **Log Loss** میزان عدم اطمینان از پیش بینی ما بر اساس میزان تفاوت آن از برچسب اصلی است. با کمک مقدار **Log Loss** می توانیم دید دقیق تری از کارایی مدل خود داشته باشیم. برای محاسبه **Log Loss** میتوان از تابع `log_loss` از `sklearn.metrics` استفاده کرد.

مثال: مثال زیر یک راهکار ساده در پایتون را معرفی می کند که یک دید درباره چگونگی استفاده از معیارهای کارایی مطرح شده فوق را روی مدل طبقه بندی دودویی به ما می دهد.

```
from sklearn.metrics import confusion_matrix
from sklearn.metrics import accuracy_score
from sklearn.metrics import classification_report
from sklearn.metrics import roc_auc_score
from sklearn.metrics import log_loss
X_actual = [1, 1, 0, 1, 0, 0, 1, 0, 0, 0]
Y_predic = [1, 0, 1, 1, 1, 0, 1, 1, 0, 0]
results = confusion_matrix(X_actual, Y_predic)
print ('Confusion Matrix :')
print(results)
print ('Accuracy Score is',accuracy_score(X_actual,
Y_predic))
print ('Classification Report : ')
print (classification_report(X_actual, Y_predic))
print('AUC-ROC:',roc_auc_score(X_actual, Y_predic))
print('LOGLOSS Value is',log_loss(X_actual,
Y_predic))
```

خروجی:

```
Confusion Matrix :
[
  [3 3]
  [1 3]
]
Accuracy Score is 0.6
Classification Report :
              precision          recall       f1-score
support
      0              0.75            0.50            0.60
      6              0.50            0.75            0.60
      4
micro avg              0.60            0.60            0.60
10
macro avg              0.62            0.62            0.60
10
weighted avg          0.65            0.60            0.60
10
AUC-ROC:  0.625
```

LOGLOSS Value is 13.815750437193334

معیارهای کارایی برای مسائل رگرسیون: درباره رگرسیون و الگوریتم های آن در بخش های قبل

صحبت کردیم. در اینجا، معیارهای متنوع کارایی که برای ارزیابی پیش بینی ها در مسائل رگرسیون مورد استفاده قرار میگیرند را معرفی می کنیم.

میانگین خطای مطلق (mean absolute error - MAE) : این ساده ترین معیار خطا است که در

مسائل رگرسیون مورد استفاده قرار میگیرد. اساسا MAE مجموع میانگین اختلاف مطلق بین مقادیر واقعی و پیش بینی است. به بیان ساده تر، با کمک MAE می توان فهمید که پیش بینی ها چقدر اشتباه هستند. MAE جهت گیری مدل را مشخص نمی کند، یعنی هیچ نظری درباره کم کارا بودن (underperformance) یا پر کارا بودن (overperformance) مدل نمی دهد. رابطه زیر فرمول محاسبه MAE است.

$$MAE = \frac{1}{n} \sum |Y - \hat{Y}|$$

در اینجا، Y برابر است با مقادیر خروجی واقعی و \hat{Y} مقادیر خروجی پیش بینی شده است. برای محاسبه MAE می توان از تابع `mean_absolute_error` از `sklearn.metrics` استفاده کرد.

خطای میانگین مربع (mean square error - MSE) : شبیه MAE است، با این تفاوت که

مربع تفاوت مقادیر خروجی واقعی و پیش بینی شده را به جای استفاده از مقدار مطلق، قبل از محاسبه مجموع همه آنها محاسبه می کند. این اختلاف در رابطه زیر نشان داده شده است.

$$MSE = \frac{1}{n} \sum (Y - \hat{Y})^2$$

در اینجا، Y برابر است با مقادیر خروجی واقعی و \hat{Y} مقادیر خروجی پیش بینی شده است. برای محاسبه MSE می توان از تابع `mean_squared_error` از `sklearn.metrics` استفاده کرد.

مربع R: به طور کلی از معیار مربع R برای اهداف توضیحی استفاده می شود. این معیار یک نشانه ای از خوبی یا تناسب یک مجموعه از مقادیر خروجی پیش بینی شده با مقادیر خروجی واقعی ارائه می کند. رابطه زیر به فهم آن کمک می کند.

$$R^2 = 1 - \frac{\frac{1}{n} \sum_{i=1}^n (Y_i - \hat{Y}_i)^2}{\frac{1}{n} \sum_{i=1}^n (Y_i - \bar{Y}_i)^2}$$

در عبارت ریاضی فوق، MSE صورت کسر است و مخرج کسر واریانس در مقادیر Y است. برای محاسبه مقدار مربع R می توان از تابع `r2_score` از `sklearn.metrics` استفاده کرد.

مثال: مثال زیر یک راهکار ساده در پایتون را معرفی می کند که یک دید درباره چگونگی استفاده از معیارهای کارایی مطرح شده فوق را روی مدل رگرسیون به ما می دهد.

```
from sklearn.metrics import r2_score
from sklearn.metrics import mean_absolute_error
from sklearn.metrics import mean_squared_error
X_actual = [5, -1, 2, 10]
Y_predic = [3.5, -0.9, 2, 9.9]
print ('R Squared =', r2_score(X_actual, Y_predic))
print ('MAE =', mean_absolute_error(X_actual,
Y_predic))
print ('MSE =', mean_squared_error(X_actual,
Y_predic))
```

خروجی:

```
R Squared = 0.9656060606060606
MAE = 0.42499999999999999
MSE = 0.56749999999999999
```