机器学习笔记（2）：模型的评估指标

原创 链原力 链原力 2月23日

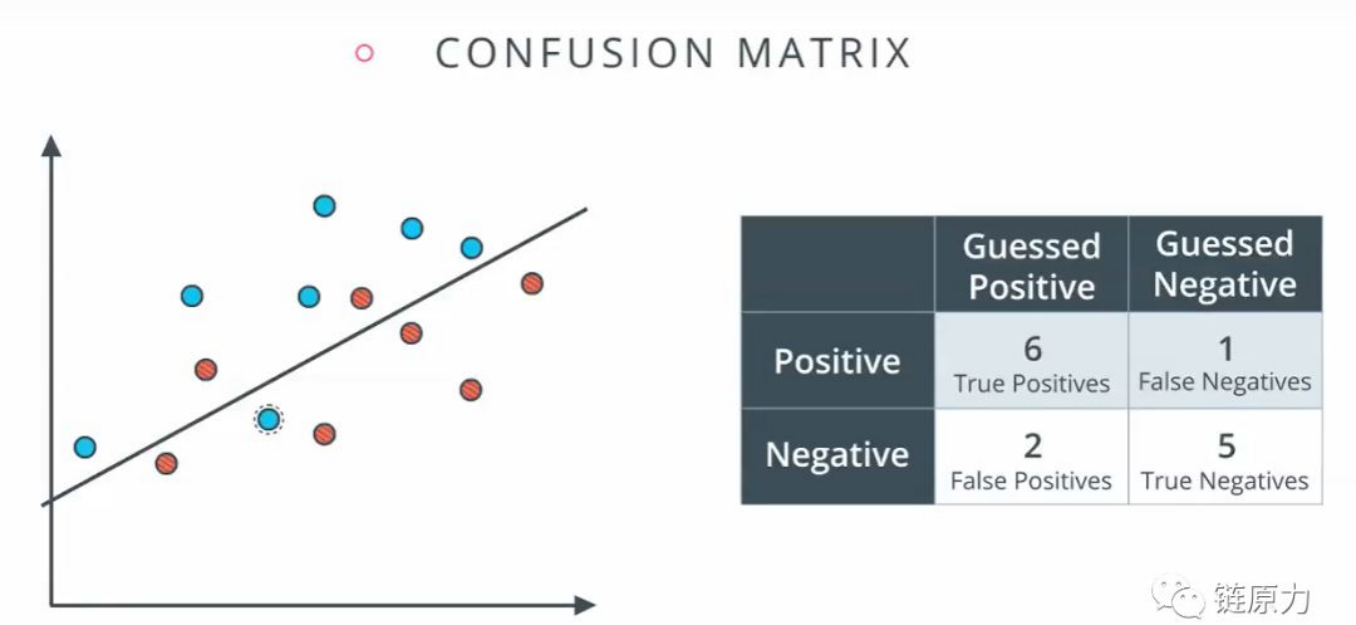
本文来自之前在Udacity上自学机器学习的系列笔记。这是第2篇，介绍了模型的评估指标。

1.评估指标

不同的模型，需要使用不同的指标来评估。下面一一介绍。

1.1.混淆矩阵（Confusion Matrix）

混淆矩阵是一种展示模型结果的矩阵表达方式。如下图所示，蓝点和红点是训练数据，直线是模型的分类结果，直线之上表示分类结果都是蓝点，直线之下表示分类结果都是红点。




```
from sklearn.metrics import confusion_matrix
confusion_matrix(y, guesses)
```

1.2.准确度（Accuracy）

分类准确数与总数的比例。但是精确度不能仅仅作为一些问题的唯一评估指标，例如信用卡欺诈，即使存在一个99%的模型可以将欺诈数据排查出来，但是仍然不是一个好模型，因为它无法捕捉欺诈交易，而模型的目标正是捕捉所有欺诈交易。

```
from sklearn.metrics import accuracy_score
Accuracy = accuracy_score(y, guesses)
```

1.3.精度（Precision）和召回率（Recall）

精度就是判断为Positive且的确为真的与所有判断为Positive的比率；而召回率就是判断为Positive且的确为真的与所有真实Positive的比率。前者追求的是查准率，后者追求的是查全率。下面举两个例子。不同例子所需要使用的指标是不同的。

医院检测一个人是否患病，有四种情况：

判断是否患病	分类为阳性	分类为阴性
真实阳性	真阳性（治疗！）	假阴性（避免！）
真实阴性	假阳性（多检查！）	真阴性（还好！）

医院应该尽力避免“假阴性”，因为如果一个人真实患病但被诊断为没病，就不好了。所以，判断是否患病的模型追求高召回率，判断为阳性且为真的数量占真实阳性数量越大，说明模型越好。

判断一封邮件是否为垃圾邮件，也有四种情况：

判断是否垃圾邮件	分类为垃圾邮件	分类为非垃圾邮件
真实垃圾邮件	真垃圾邮件（过滤！）	假非垃圾邮件（多检查！）
真实非垃圾邮件	假垃圾邮件（避免！）	真非垃圾邮件（还好！）

邮件系统应该尽力避免“假垃圾邮件”，因为如果把一封家人发过来的邮件判断为垃圾邮件，就不好了。判断是否垃圾邮件模型追求高精度。判断为垃圾邮件且为真的数量占判断为垃圾邮件的数量越大，说明模型越好。

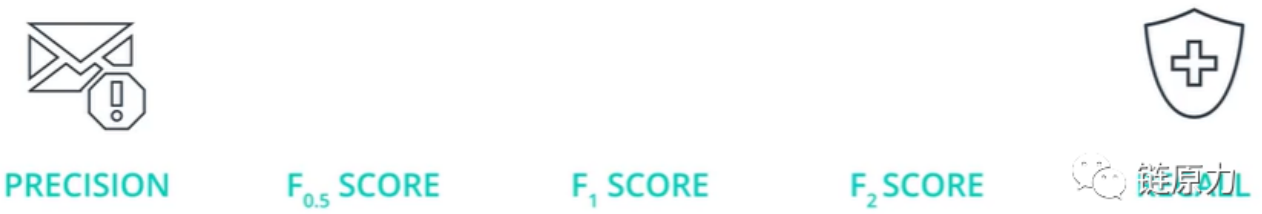
```
from sklearn.metrics import precision_score
from sklearn.metrics import recall_score
Precision = precision_score(y, guesses)
Recall = recall_score(y, guesses)
```

1.4.F1分数和F-beta分数

F1和F-beta分数是综合考虑精度和召回率的指标。具体定义如下：

$$F_1\ SCORE = 2\ \frac{Precision * Recall}{Precision + Recall}$$

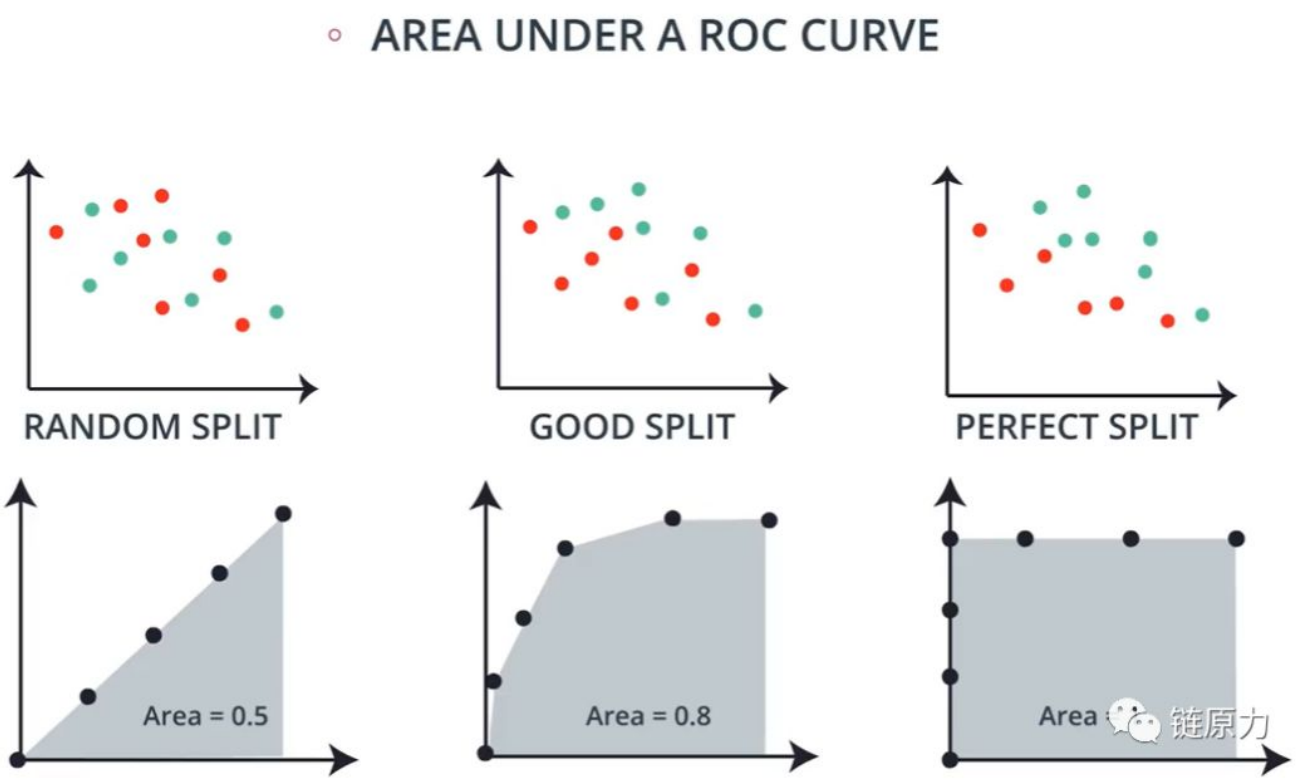
$$F_\beta\ SCORE = (1+\beta^2)\ \frac{Precision * Recall}{\beta^2 * Precision + Recall}$$



从它们的数学定义可以看到，随着beta取不同的值，F-beta在精度和召回率之间移动。当beta=1时，得到一个平衡的考量。为用户推荐一部可能喜欢的电影，我们可以取beta=1；为潜在客户发送促销邮件，我们可以取beta=0.5，这样模型指标更加靠近精度，确保目标的用户群体中有意向的客户越多越好；而检测航空器中的故障部件率，则应该取beta=2，这样模型指标更加靠近召回率，确保尽可能把有故障部件都找出来。

1.5.ROC曲线（Receiver Operator Characteristic Curve）

ROC曲线是这样构成的，我们将判断为Positive且为真的比率作为纵坐标，判断为Positive但为假的比率作为横坐标。然后根据模型对数据的划分，我们在数据间设立一个划分边界，来计算上述的两个比率。如果模型对数据划分的好，那么这些比率点所形成的曲线面积越靠近1，代表模型越好。




```
from sklearn.metrics import roc_curve, roc_auc_curve
```

```
roc_curve(y, guesses)
roc_auc_curve(y, guesses)
```

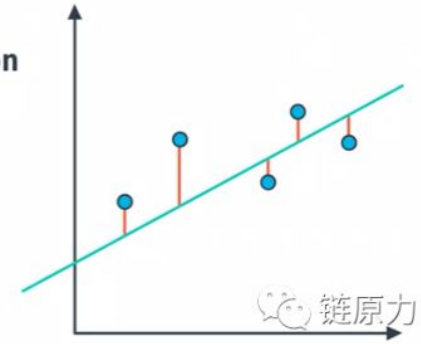
以上模型适用于监督学习的分类模型。

下面介绍适用于监督学习的回归模型的指标。

1.6.平均绝对误差（Mean Absolute Error）

预测值与真实值之差的绝对值，加起来求其平均。

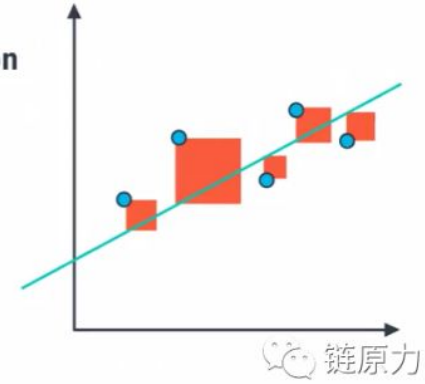
```
from sklearn.metrics import mean_absolute_error
from sklearn.linear_model import LinearRegression

classifier = LinearRegression()
classifier.fit(X,y)

guesses = classifier.predict(X)

error = mean_absolute_error(y, guesses)
```



1.7.均方误差（Mean Squared Error）

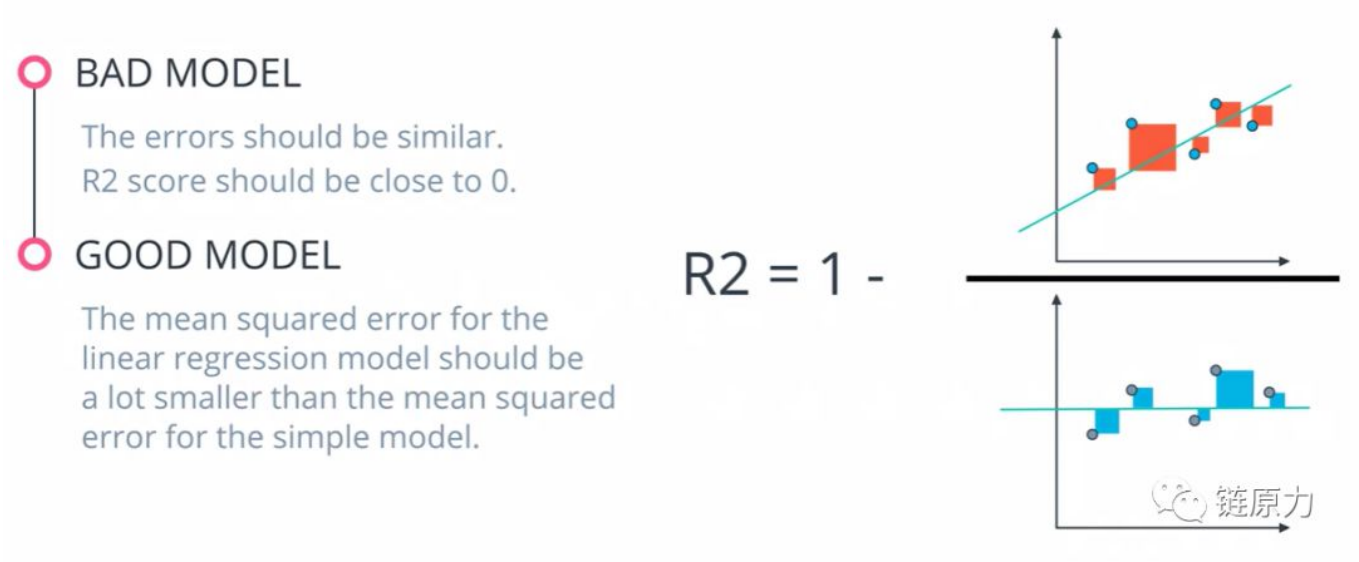
预测值与真实值之差的平方，加起来求其平均。

```
from sklearn.metrics import mean_squared_error
from sklearn.linear_model import LinearRegression

classifier = LinearRegression()
classifier.fit(X,y)

guesses = classifier.predict(X)

error = mean_squared_error(y, guesses)
```



1.8.R2分数（R2 Score）

R2分数通过将我们的模型与最简单的可能模型相比得出。所谓最简单的可能模型，比如说我们要拟合一个散点图，我们可以直接画一条直线穿过这些点。然后求出该直线模型下的均方误差。我们希望期望模型的均方误差比这个直线模型要小，而且是小得多，从而使得R2分数趋向于1。




```
from sklearn_metrics import r2_score
y_true = [1, 2, 4]
y_pred = [1.3, 2.5, 3.7]
r2_score(y_true, y_pred)
```

平均绝对误差、均方误差和R2分数常用于监督学习的回归模型中。