**模型评估**

模型在训练集上的误差通常称为 “训练误差” 或 “经验误差”，而在新样本上的误差称为 “泛化误差”。显然，机器学习的目的是得到泛化误差小的学习器。然而，在实际应用中，新样本是未知的，所以只能使训练误差尽量小。

当模型在训练集上表现很好而在新样本上误差很大时，称为 “过拟合”；反之，模型在训练集上误差就很大时，称为 “欠拟合”。

* 欠拟合通常容易解决，如 **增加数据、增大训练次数、增大学习率或使用更复杂的模型** 等。
* 过拟合则很难解决，常用的方法包括 **简化模型，减少特征，更多的数据，交叉验证，正则化，Dropout，Shuffling** 等。

奥卡姆剃刀定律：如无必要，勿增实体。即机器学习模型越简单，那么久越可能获得泛化误差小的学习器。

所以，为了得到泛化误差小的模型，并避免过拟合，在构建机器模型时，通常将 [数据集拆分](https://feisky.xyz/machine-learning/basic/datasets.html) 为相互独立的训练数据集、验证数据集和测试数据集等，而在训练过程中使用验证数据集来评估模型并据此更新超参数，训练结束中使用测试数据集评估训练好的最终模型的性能。

**评价指标**

评价指标（也称性能度量）是评估机器学习泛化能力的标准，不同的指标往往会导致不同的评判结果。并且，不同的机器学习任务也有着不同的评价指标。

常用的评价指标有

* **错误率**：分类错误的样本数占样本总数的比例。
* **准确率（精度）**：分类正确的样本数占样本总数的比例。在很多情况下，准确率是一个欠佳或具有误导性的指标，比如在不同类型错误具有不同代价时（典型为分类不平衡，即正类别或负类别极其罕见）。
* **对数损失函数（置信度）**：log\\_loss=-\frac{1}{N}\sum\_{i=1}^{N}y\_i log p\_i + (1-y\_i)log(1-p\_i)*log*\_*loss*=−​*N*​​1​​∑​*i*=1​*N*​​*y*​*i*​​*logp*​*i*​​+(1−*y*​*i*​​)*log*(1−*p*​*i*​​)，其中 y\_i*y*​*i*​​ 是指第 i*i* 个样本所属的真实类别（0 或者 1），而 p\_i*p*​*i*​​ 表示第 i*i* 个样本属于类别 1 的概率。
* **查准率和查全率**：查准率是指分类器分类正确的正样本（True Positive，TP）的个数占该分类器所有分类为正样本个数（TP+FP (false positive)）的比例；而查全率是指分类器分类正确的正样本个数（TP）占所有的正样本个数（TP+FN (false negative)）的比例。这两者通常是一对矛盾的变量，**查准率高时，查全率往往偏低**。
* **F1 度量**：基于查准率和查全率的调和平均定义， F1=\frac{2PR}{P+R}*F*1=​*P*+*R*​​2*PR*​​ ，F1\_\beta=\frac{(1+\beta^2)PR}{\beta^2P+R}*F*1​*β*​​=​*β*​2​​*P*+*R*​​(1+*β*​2​​)*PR*​​ ，其中 \beta*β* 是查全率对查准率的相对重要性。
* **受试者工作特征（Receiver Operating Characteristic，ROC）**：ROC 曲线描述真正例率（TPR）与假正例率（FPR）的关系。在比较 ROC 时，需要比较 ROC 曲线下面的面积（Area Under ROC Curve，AUC），**AUC 越大，分类效果越好**。
* **代价敏感错误率**：用于非均等代价的情景，根据任务的领域知识设定代价矩阵，计算的错误率称为代价敏感错误率；而对应的 ROC 曲线叫代价曲线。

**比较检验**

机器学习中，性能比较是比较复杂的

* 我们希望比较的是泛化性能，而实际中只能得到测试集上的性能，两者的结果未必相同
* 测试集上的性能跟测试集本身的选择有很大关系
* 很多机器学习算法本身有一定的随机性，即便用相同参数运行在同一个测试集上，多次的运行的结果也有可能不同

所以，我们需要运用统计假设检验（hypothesis test）得到模型的泛化性能是否在统计意义上较优。常用的方法包括

* 二项检验
* t 检验
* 交叉验证 t 检验
* McNemar 检验
* Friedman 检验
* Nemenyi 后续检验

**偏差与方差**

泛化误差可以分解为偏差、方差与噪声之和，偏差 - 方差分解（bias-variance decomposition）是解释泛化性能的重要工具。

* 偏差度量了算法的期望预测与真实结果的偏离程度。
* 方差度量了同样大小训练集变动导致的性能变化。
* 噪声涉及问题本身的难度。

所以，泛化性能是由学习算法的能力、数据的充分性以及学习任务本身的难度共同决定的。

但偏差与方差是有冲突的，即偏差 - 方差窘境（bias-variance dilemma）。在训练程度不足时，学习器拟合程度不强，训练数据的扰动不足以产生显著变化，此时偏差主导泛化错误率。随着训练程度加深，学习器拟合能力增强，训练数据的扰动逐渐可以被学习器学到，方差逐渐主导泛化错误率。如果继续加深训练，则有可能发生过拟合。

## 分类模型评价度量

### 混淆矩阵（Confusion Matrix）

判定方法：根据不同的模型选不同的参数。

原理：混淆矩阵是一个N X N矩阵，N为分类的个数。假如我们面对的是一个二分类问题，也就是N＝2，我们就得到一个2 X 2矩阵。在学习这个矩阵之前，我们需要知道一些简单的定义。（个人不喜欢这部分的中文翻译，一直是保留英文）

Accuracy(准确度):预测正确的数占所有数的比例。

Positive Predictive Value(阳性预测值) or Precision(精度):阳性预测值被预测正确的比例。

Negative Predictive Value(阴性预测值):阴性预测值被预测正确的比例。

Sensity(灵敏度) or recall(召回率)：在阳性值中实际被预测正确所占的比例。

Specificity(特异度):在阴性值中实现被预测正确所占的比例。

若我们得到一个模型的混淆矩阵如下：

我们可以看出，这个模型的准确度为88%，阳性预测值比较高而阴性预测值较低。对于灵敏度和特异度也是相同。这是因为我们选的阈值而导致的，若我们降低阈值，这两对数值就会变的相近。一般的情况下，我们只关心其中的一个定义度量。例如，在医药公司，一般会更加关心最小化误诊率，也就是他们需要的是高特异度。而在磨损模型中我们更关心的是灵敏度。可以看出混淆模型一般只能用于分类输出型模型中。

### ROC曲线

判定方法：ROC曲线应尽量偏离参考线。

原理：ROC全称为Receiver Operation Characteristic Curve,中文名叫做‘接受者操作特征曲线’，中文名简直就是直翻。ROC曲线其实就是从混淆矩阵衍生出来的图形，其横坐标为1-Specificity,纵坐标为Sensitivity。

上面那条曲线就是ROC曲线，随着阈值的减小，更多的值归于正类，敏感度和1-特异度也相应增加，所以ROC曲线呈递增趋势。而那条45度线是一条参照线，也就是说ROC曲线要与这条曲线比较。

简单的说，如果我们不用模型，直接随机把客户分类，我们得到的曲线就是那条参照线，然而我们使用了模型进行预测，就应该比随机的要好，所以ROC曲线要尽量远离参照线，越远，我们的模型预测效果越好。

### AUC（ROC曲线下面积）

判定方法：AUC应该大于0.5.

原理：ROC曲线是根据与那条参照线进行比较来判断模型的好坏，但这只是一种直觉上的定性分析，如果我们需要精确一些，就要用到AUC，也就是ROC曲线下面积。

看上图，参考线的面积是0.5，ROC曲线与它偏离越大，ROC曲线就越往左上方靠拢，它下面的面积(AUC)也就越大，这里面积是0.869。我们可以根据AUC的值与0.5相比，来评估一个分类模型的预测效果。

### Lift（提升）和Gain（增益）

判定方法：lift应一直大于1.

原理：Lift＝(d/b+d)/(c+d/a+b+c+d).它衡量的是，与不用模型相比，模型的预测能力提升了多少。不利用模型，我们只能利用正例的比例‘c+d/a+b+c+d’这个样本信息来估计正例的比例(baseline model),而利用模型之后，只需要从我们预测为正例的那个样本的子集(b+d)中挑选正例，这时预测的准确率为d/b+d.

由此可见，lift(提升指数)越大，模型的预测效果越好。如果这个模型的预测能力跟baseline model一样，那么这个模型就没有任何意义。

一个好的分类模型，就是要偏离baseline model足够远。在lift图中，表现就是，在depth为1之前，lift一直保持较高的(大于1的)数值，也即曲线足够陡峭。

Gain与Lift相当类似，Gain chart是不同阈值下d/b+d的轨迹，与Lift的区别就在于纵轴刻度的不同。

### K-S图

判定方法：其值在0到100之间，值越大，模型表现越好。

原理：K-S图，英文为Kolmogorov-Smirnov chart,是用来评估分类模型表现的图。更准确的来说，K-S是用来度量阳性与阴性分类区分程度的。若我们把总体严格按照阳性和阴性分成两组，则K-S值为100，如果我们是随机区分阳性与阴性，则K-S值为0.所以分类模型的K-S值都在0到100之间，值越大，模型从阴性数据中区分阳性数据的能力越强。

### 基尼系数

判定方法：基尼系数应大于60%，就算好模型。

原理：基尼系数经常用于分类问题，其可以直接从AUC中得到。其公式为：

Gini ＝ 2\*AUC － 1

评估分类模型：

| **指标** | **描述** | **Scikit-learn函数** |
| --- | --- | --- |
| Precision | 精准度 | from sklearn.metrics import precision\_score |
| Recall | 召回率 | from sklearn.metrics import recall\_score |
| F1 | F1值 | from sklearn.metrics import f1\_score |
| Confusion Matrix | 混淆矩阵 | from sklearn.metrics import confusion\_matrix |
| ROC | ROC曲线 | from sklearn.metrics import roc |
| AUC | ROC曲线下的面积 | from sklearn.metrics import auc |

评估回归模型：

| **指标** | **描述** | **Scikit-learn函数** |
| --- | --- | --- |
| Mean Square Error (MSE, RMSE) | 平均方差 | from sklearn.metrics import mean\_squared\_error |
| Absolute Error (MAE, RAE) | 绝对误差 | from sklearn.metrics import mean\_absolute\_error, median\_absolute\_error |
| R-Squared | R平方值 | from sklearn.metrics import r2\_score |