智能软件开发方向基础

第三章 模型的选择与性能评价

张朝晖

2022~2023学年第二学期

河北师范大学软件学院
Software College of Hebei Normal University

序号	内容
1	概述
2	机器学习的基本概念
3	模型的选择与性能评价
4	数据的获取、探索与准备
5	近邻模型------分类、回归
6	决策树模型------分类、回归
7	集成学习------分类、回归
8	(朴素)贝叶斯模型------分类
9	聚类
10	特征降维及低维可视化(PCA, t-SNE)
11	总复习

课程主要参考书

主要内容

1. 模型的学习能力与泛化能力
   - 学习能力、泛化能力
   - 学习误差(或训练误差)、测试误差
   - 欠学习(欠拟合，underfitting)
   - 过学习(过拟合，overfitting)
2. 交叉验证
3. (基于测试集的)模型评价
   - 二分类/多分类模型、回归模型

河北师范大学软件学院
Software College of Hebei Normal University

1.1 学习能力

- 学习得到的模型关于**训练样本集**的预测能力
- 学习能力的评价：
  训练误差(或经验误差)，基于损失函数的经验风险

设学习得到的模型为：$Y=\hat{f}(X)$

训练样本集 $D_{\text{train}}=\{(x_i,y_i),i=1,\ldots,N_{\text{train}}\}$

训练误差：$R_{\text{emp}}(\hat{f})=\frac{1}{N_{\text{train}}}\sum_{i=1}^{N_{\text{train}}}L\left(y_i,\hat{f}(x_i)\right)$

河北师范大学软件学院
Software College of Hebei Normal University

面向分类：

注意此处的指示函数

0-1损失 $L(Y,f(X))=I(Y\neq f(X))=\begin{cases}1 & Y\neq f(X)\\ 0 & Y=f(X)\end{cases}$

$$R_{\text{emp}}(\hat{f})=\frac{1}{N_{\text{train}}}\sum_{i=1}^{N_{\text{train}}}L\left(y_i,\hat{f}(x_i)\right)=\frac{1}{N_{\text{train}}}\sum_{i=1}^{N_{\text{train}}}I\left(y_i,\hat{f}(x_i)\right)$$

训练样本的预测错误率

面向回归：

平方损失函数 $L(Y,f(X))=[Y-f(X)]^2$

$$R_{\text{emp}}(\hat{f})=\frac{1}{N_{\text{train}}}\sum_{i=1}^{N_{\text{train}}}L\left(y_i,\hat{f}(x_i)\right)=\frac{1}{N_{\text{train}}}\sum_{i=1}^{N_{\text{train}}}[y_i-\hat{f}(x_i)]^2$$

均方误差MSE

河北师范大学软件学院
Software College of Hebei Normal University

1.2 泛化能力(generalization ability)

- 学得的模型关于**未知样本**的预测能力

- **泛化误差：**

设学习得到的模型为：$Y=\hat{f}(X)$

该模型关于未知样本的预测误差，即为**泛化误差**,它是学习得到的模型的**期望风险**。

$$R_{\exp}(\hat{f})=\int_{\mathcal{X}\times\mathcal{Y}} L\left(y,\hat{f}(\boldsymbol{x})\right)P(\boldsymbol{x},y)\,d\boldsymbol{x}dy$$

随机变量X,Y的函数的数学期望
--统计意义上的损失值的平均

1.2 泛化能力(generalization ability)-续

- **测试误差**

泛化误差难以估计，实际以**测试误差**评价**模型泛化能力**

给定测试集 $D_{\text{test}}=\{(\boldsymbol{x}_j,y_j),j=1,\ldots,N_{test}\}$

测试误差：

$$R_{\text{test}}(\hat{f})=\frac{1}{N_{\text{test}}}\sum_{j=1}^{N_{test}} L\left(y_j,\hat{f}(\boldsymbol{x}_j)\right)$$

例：对于“分类问题”的0-1损失，测试误差就是模型关于测试集的**预测错误率**：

$$Err_{test}=\frac{1}{N_{\text{test}}}\sum_{i=1}^{N_{test}} I\left(y_i\neq\hat{f}(\boldsymbol{x}_i)\right)$$

注意此处的指示函数

与**预测错误率**对应的是**预测正确率**：

$$Acc_{test}=1-Err_{test}=\frac{1}{N_{\text{test}}}\sum_{i=1}^{N_{test}} I\left(y_i=\hat{f}(\boldsymbol{x}_i)\right)$$

指示函数举例：

$$I(\mathrm{a}=b)=\begin{cases}1 & \text{若a}=b\\ 0 & \text{若a}\neq b\end{cases}\quad I(\mathrm{a}\neq b)=\begin{cases}1 & \text{若a}\neq b\\ 0 & \text{若a}=b\end{cases}\quad I(\mathrm{a}\geq b)=\begin{cases}1 & \text{若a}\geq b\\ 0 & \text{若a}<b\end{cases}$$

1.3 过拟合与模型选择

当采用规模有限的训练集学习模型时，因模型过于复杂，使得学得的模型对**训练集预测性能很好**，而关于**测试集的预测性能很差**，称该现象为**过拟合(overfitting).**

例：基于给定的训练集，进行**M次多项式函数**拟合

M次多项式函数：

$$f_M(x;\boldsymbol{\omega})=\omega_0+\omega_1x+\omega_2x^2+\cdots+\omega_Mx^M=\sum_{j=0}^{M}\omega_jx^j$$

当M给定时，需要估计的参数向量为：

$$\boldsymbol{\omega}=[\omega_0\quad\ldots\quad\omega_M]^T$$

训练集 $D_{\text{train}}=\{(\boldsymbol{x}_i,y_i),i=1,\ldots,N_{\text{train}}\}$
测试集 $D_{\text{test}}=\{(\boldsymbol{x}_k,y_k),k=1,\ldots,N_{test}\}$

采用**经验风险最小化策略**学习模型：

- **损失函数形式**----平均平方误差MSE

- **最小二乘准则**

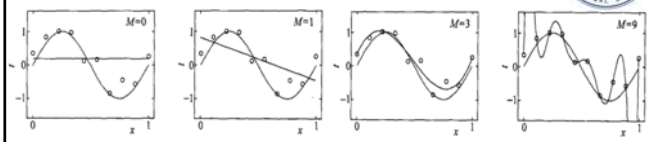
(即：预测误差的平方和最小，或MSE最小)

$$E(\boldsymbol{\omega})=\frac{1}{N_{\text{train}}}\sum_{i=1}^{N_{\text{train}}}[y_i-\hat{y}_i(\boldsymbol{\omega})]^2$$
$$=\frac{1}{N_{\text{train}}}\sum_{i=1}^{N_{\text{train}}}[y_i-f_M(x;\boldsymbol{\omega})]^2$$

$$\boldsymbol{\omega}^*=\underset{\boldsymbol{\omega}}{\operatorname{argmin}}\,E(\boldsymbol{\omega})$$

如何求解？

讨论：不同模型复杂度M的多项式函数拟合结果



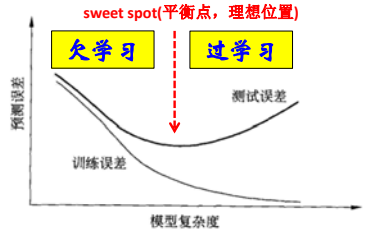
随着模型复杂程度(多项式阶数M)的增加：
模型的训练误差逐渐减小，甚至趋0；
但模型的测试误差先减小，后增加。

M是多项式函数拟合的超参数！

➔多项式函数拟合，应选择合适的M值。

对于多项式函数的拟合来说，M值的确定过程就是模型的选择过程。

右图：
训练误差、测试误差与模型复杂程度之间的关系



随着模型复杂程度的增加：
训练误差逐渐减小，甚至为0；
测试误差先减小，达到最小值后，又增大。

当模型过于复杂时，将产生过拟合。

为避免过拟合，应选择适当复杂度的模型。

常采用两种方式

- 方式1. 采用结构风险最小化策略，构建含有正则项的目标函数
  本讲暂略。
- 方式2. 基于交叉验证方式的模型选择

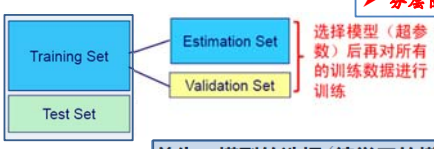
交叉验证不仅用于模型选择，还用于最终模型的评价。

主要内容

1. 模型的学习能力与泛化能力
2. 交叉验证
   - 2.1 数据集的划分方式
   - 2.2 基于交叉验证的模型选择与评价
3. (基于测试集的)模型评价
   - 二分类/多分类模型、回归模型

数据集D的两种层次划分及适用场合
以监督式模型学习为例

如何划分：
- 随机打乱
- 分层随机打乱



首先，模型的选择(待学习的模型的超参数调节)
训练集{估计集(学习)
验证集(评价)}

然后，基于选择的模型结构，学习得到最终模型
{训练集(超参数固定之后，用于学习最终模型)
测试集(评价最终模型)}

完整的模型学习过程

主要内容

1. 模型的学习能力与泛化能力
2. 交叉验证
   - 2.1 数据集的划分方式
   - 2.2 基于交叉验证的模型选择与评价
3. (基于测试集的)模型评价
   - 二分类/多分类模型、回归模型、聚类模型的评价

https://scikit-learn.org/stable/modules/classes.html#module-sklearn.model_selection

sklearn.model_selection: Model Selection

User guide: See the Cross-validation: evaluating estimator performance, Tuning the hyper-parameters of an estimator and Learning curve sections for further details.

Splitter Classes

model_selection.GroupKFold([n_splits])	K-fold iterator variant with non-overlapping groups.
model_selection.GroupShuffleSplit([...])	Shuffle-Group(s)-Out cross-validation iterator
model_selection.KFold([n_splits, shuffle, ...])	K-Folds cross-validator
model_selection.LeaveOneGroupOut()	Leave One Group Out cross-validator
model_selection.LeavePGroupsOut(n_groups)	Leave P Group(s) Out cross-validator
model_selection.LeaveOneOut()	Leave-One-Out cross-validator
model_selection.LeavePOut(p)	Leave-P-Out cross-validator
model_selection.PredefinedSplit(test_fold)	Predefined split cross-validator
model_selection.RepeatedKFold(*[, n_splits, ...])	Repeated K-Fold cross validator.
model_selection.RepeatedStratifiedKFold(*[, ...])	Repeated Stratified K-Fold cross validator.
model_selection.ShuffleSplit([n_splits, ...])	Random permutation cross-validator
model_selection.StratifiedKFold([n_splits, ...])	Stratified K-Folds cross-validator.
model_selection.StratifiedShuffleSplit([...])	Stratified ShuffleSplit cross-validator
model_selection.StratifiedGroupKFold([...])	Stratified K-Folds iterator variant with non-overlapping groups.
model_selection.TimeSeriesSplit([n_splits, ...])	Time Series cross-validator

思考：

- 如何基于交叉验证，进行模型的选择？
- K-fold CV用于模型选择/评价的具体实现过程？
- K-fold CV 中的K值一般为多少？
- 何时使用LOOCV？何时使用K-fold CV？
- 如何基于交叉验证，进行模型的评价？
- 最终要使用的模型应怎么得到？

河北师范大学软件学院
Software College of Hebei Normal University

以监督式学习系统为例，首先**考察最终模型泛化能力的评估方式**
然后**再理解如何在生成最终模型之前，用交叉验证法进行模型选择**。

给定已知答案的**数据集** $\boldsymbol{D}=\{(x_i,y_i),i=1,\ldots,N\}$

- 训练集 $\boldsymbol{D}_{train}$--模型的学习
- 测试集 $\boldsymbol{D}_{test}$--模型泛化能力的评价

$\boldsymbol{D}=\boldsymbol{D}_{train}\cup\boldsymbol{D}_{test}$

数据集 $\boldsymbol{D}$ 划分的几种典型实现方式
[1]留出法(*hold-out*)
[2]交叉验证(*cross validation*)
[3]自助法(*bootstrapping*)

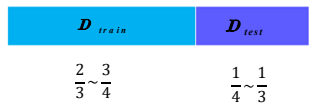
如何划分：
- **“随机打乱”—回归**
- **分层随机打乱—分类**

河北师范大学软件学院
Software College of Hebei Normal University

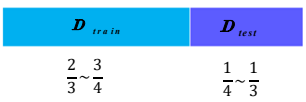
[1]留出法(*hold-out*)、留出法交叉验证(*hold-out* cv)

$\boldsymbol{D}=\boldsymbol{D}_{train}\cup\boldsymbol{D}_{test}$
$\Phi=\boldsymbol{D}_{train}\cap\boldsymbol{D}_{test}$

数据集**随机划分**尽量**保持数据分布的一致性**
A. 单独一次随机划分，估计结果不够稳定可靠
B. 应多次随机划分，重复评估，**取结果的均值及标准差!!!**
(*hold-out cross-validation*)



河北师范大学软件学院
Software College of Hebei Normal University



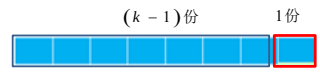
矛盾：
- **训练集规模应足够大**，以便使模型的学习，尽可能在模型学习过程中，让其见识更为丰富的样本多样性；但会导致测试集内样本多样性降低，基于测试集的评价结果不够稳定、准确，无法较好近似模型的泛化能力。
- **测试集规模应足够大**，以便使测试误差更接近泛化误差；但会导致训练集规模降低，使得训练得到的模型与基于整个数据集得到的模型差别较大，降低了评估结果的保真性。

河北师范大学软件学院
Software College of Hebei Normal University

[2]k-倍交叉验证
(k-*fold cross validation*，k-*fold CV*)–*rotation estimation*

$\boldsymbol{D}=\boldsymbol{D}_{train}\cup\boldsymbol{D}_{test}$　　$\Phi=\boldsymbol{D}_{train}\cap\boldsymbol{D}_{test}$

数据集的随机划分尽量**保持数据分布的一致性**
随机打乱$\boldsymbol{D}$，均分成k等份
k=?　3,5,10



- **单轮**k-倍交叉验证
- **多轮**k-倍交叉验证

悲观、有偏估计

留一法交叉验证(*leave-one-out cross-validation*，*LOO-CV*)
无偏估计

河北师范大学软件学院
Software College of Hebei Normal University

例：基于单轮K-折交叉验证错误率，评价某分类模型的预测性能

STEP1. 将样本集D分层随机打乱，均分成互斥的K等份：$D = D_1 \cup \cdots \cup D_K$

STEP2. for j=1,2,...,K, do:

以样本集D_j为测试集D_{test}，其余部分构成训练集D_{train}=D\D_j

以D_{train}学习得到模型$model_j$评价D_{test}，得到**测试错误率**$Error_j$

STEP3.估计上述K个错误率的均值及标准差:

$$\mu(Error) = \frac{1}{K}\sum_{j=1}^{K} Error_j$$

$$\sigma(Error) = \left[\frac{1}{K}\sum_{j=1}^{K}\left(Error_j - \mu(Error)\right)^2\right]^{\frac{1}{2}}$$

STEP4. 输出K-折交叉验证的错误率估计结果：$\mu(Error) \pm \sigma(Error)$

注意： 对于分类问题，面向交叉验证的数据集划分应采用分层方式(即： Stratified k-fold cross validation)，确保训练集、测试集的类别分布一致。

例： Comparison between Support Vector Machines, the Kernel Fisher Discriminant (KFD), a single radial basis function classifier (RBF), AdaBoost (AB), and regularized AdaBoost (ABR) on 13 different benchmark datasets (see text). Best result in bold face, second best in italics.

平均错误率±错误率标准差

	SVM	KFD	RBF	AB	AB_R
Banana	11.5±0.07	**10.8±0.05**	**10.8±0.06**	12.3±0.07	*10.9±0.04*
B.Cancer	*26.0±0.47*	**25.8±0.46**	27.6±0.47	30.4±0.47	26.5±0.45
Diabetes	*23.5±0.17*	**23.2±0.16**	24.3±0.19	26.5±0.23	23.8±0.18
German	**23.6±0.21**	*23.7±0.22*	24.7±0.24	27.5±0.25	24.3±0.21
Heart	**16.0±0.33**	*16.1±0.34*	17.6±0.33	20.3±0.34	16.5±0.35
Image	*3.0±0.06*	3.3±0.06	3.3±0.06	**2.7±0.07**	**2.7±0.06**
Ringnorm	1.7±0.01	**1.5±0.01**	1.7±0.02	1.9±0.03	*1.6±0.01*
F.Sonar	**32.4±0.18**	*33.2±0.17*	34.4±0.20	35.7±0.18	34.2±0.22
Splice	10.9±0.07	10.5±0.06	*10.0±0.10*	10.1±0.05	**9.5±0.07**
Thyroid	4.8±0.22	**4.2±0.21**	4.5±0.21	*4.4±0.22*	4.6±0.22
Titanic	**22.4±0.10**	23.2±0.20	23.3±0.13	*22.6±0.12*	*22.6±0.12*
Twonorm	3.0±0.02	**2.6±0.02**	2.9±0.03	3.0±0.03	*2.7±0.02*
Waveform	*9.9±0.04*	*9.9±0.04*	10.7±0.11	10.8±0.06	**9.8±0.08**

河北师范大学软件学院 Software College of Hebei Normal University

例：基于留一法折交叉验证正确率，评价某分类模型的预测性能

给定样本集$D = \{(x_i, y_i), i = 1, \ldots, N\}$

STEP1.初始化计数器Counter=0

STEP2. for j=1,2,⋯,N, do:

以第j个样本(x_j, y_j)为测试样本，其余样本构成训练集D_{train}=D\$\{(x_i, y_i)\}$

以D_{train}学习得到模型$model_j$预测x_j的类别输出为$\hat{y}_j$

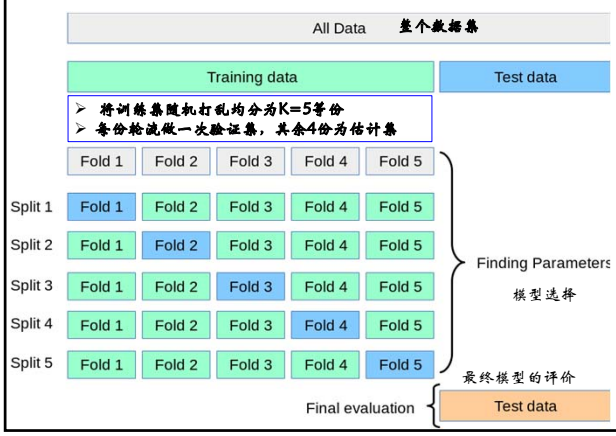
若$\hat{y}_j$与y_j一致，则Counter=Counter+1

STEP3.输出**留一法交叉验证的正确率**：

$$Acc = \frac{Counter}{N} \times 100\%$$

河北师范大学软件学院 Software College of Hebei Normal University

基于单轮K-折交叉验证的模型选择



[3]自助法 (***bootstrapping***)

bootstrap sampling

数据集$\boldsymbol{D} = \{(x_i, y_i), i = 1, \ldots, N\}$ --初始数据集

训练集$\boldsymbol{D}_{train}$--模型的学习

方式1. 对初始数据集$\boldsymbol{D}$有放回的随机抽取N次，每次抽取1个样本得自助数据集$\boldsymbol{D}'$

方式2. 从原始数据集$\boldsymbol{D}$内随机抽取样本“若干(?)?

测试集$\boldsymbol{D}_{test}$--模型泛化能力的评价

对**数据集**$\boldsymbol{D}$中没有被抽取到的样本集 $\boldsymbol{D}_{test} = \boldsymbol{D} \setminus \boldsymbol{D}'$

初始数据集内，样本未被抽取的概率 $\lim_{N\to\infty}\left(1-\frac{1}{N}\right)^N = \frac{1}{e} \approx 0.368$

自助法估计，也称"包外估计"(out-of-bag estimation)，例：**包外错误率**

河北师范大学软件学院 Software College of Hebei Normal University

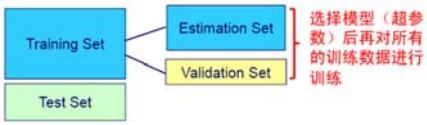
自助法常用于**集成学习**(如：bagging、随机森林)时，**集成模型的性能评价**。基本过程：

- 每个版本的训练集D_{train}对应一个个体模型的学习
- 针对样本集D内每个样本，当其作为某个版本训练集的包外样本时，基于相应的个体模型，对该**包外测试样本**的输出预测，记录其**预测结果**及其作为**包外测试样本的次数**。
- 综合样本集D内每个样本的预测结果
  例：对于分类问题，以投票方式决定每个样本的最终预测类
- 最终，生成集成模型的最终评价结果

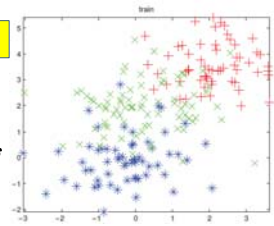
河北师范大学软件学院 Software College of Hebei Normal University

思考：如何基于交叉验证方式，进行模型的选择？

例：基于给定训练集$D_{train}=\{(x_i,y_i),i=1,\ldots,N\}$，针对C=3的分类问题，采用交叉验证方式进行KNN分类模型的选择

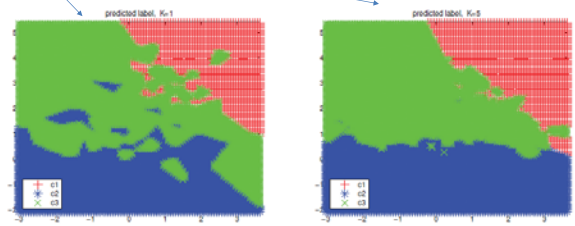


训练集



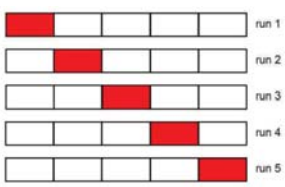
近邻数K值不同时的分类结果

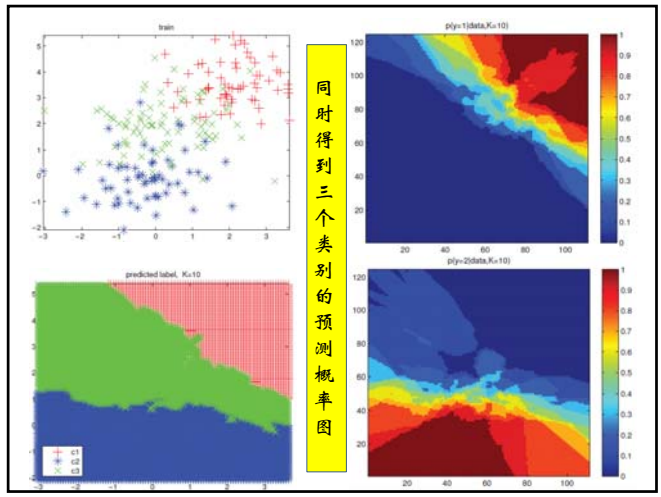
K=1　　K=5



基于**单轮5-fold cross validation**进行近邻数K值的优选

训练集=估计集∪验证集





同时得到三个类别的预测概率图

主要内容

1. **模型的学习能力与泛化能力**
2. **交叉验证**
3. **(基于测试集的)模型评价**

3.1 基于测试集的二分类模型评价

两种情况：(1)非对称类别：是vs.非

(2)对称类别：第1类vs.第2类

3.2 基于测试集的多分类模型的评价

3.3 基于测试集的实值函数预测模型评价

https://scikit-learn.org/stable/modules/model_evaluation.html

Scoring Classification	Function	Comment
'accuracy'	metrics.accuracy_score	
'balanced_accuracy'	metrics.balanced_accuracy_score	
'top_k_accuracy'	metrics.top_k_accuracy_score	
'average_precision'	metrics.average_precision_score	
'neg_brier_score'	metrics.brier_score_loss	
'f1'	metrics.f1_score	for binary targets
'f1_micro'	metrics.f1_score	micro-averaged
'f1_macro'	metrics.f1_score	macro-averaged
'f1_weighted'	metrics.f1_score	weighted average
'f1_samples'	metrics.f1_score	by multilabel sample
'neg_log_loss'	metrics.log_loss	requires predict_proba support
'precision' etc.	metrics.precision_score	suffixes apply as with 'f1'
'recall' etc.	metrics.recall_score	suffixes apply as with 'f1'
'jaccard' etc.	metrics.jaccard_score	suffixes apply as with 'f1'
'roc_auc'	metrics.roc_auc_score	
'roc_auc_ovr'	metrics.roc_auc_score	
'roc_auc_ovo'	metrics.roc_auc_score	
'roc_auc_ovr_weighted'	metrics.roc_auc_score	
'roc_auc_ovo_weighted'	metrics.roc_auc_score	

https://scikit-learn.org/stable/modules/classes.html#module-sklearn.metrics

Classification metrics

See the Classification metrics section of the user guide for further details.

metrics.accuracy_score(y_true, y_pred, *[, ...])	Accuracy classification score.
metrics.auc(x, y)	Compute Area Under the Curve (AUC) using the trapezoidal rule.
metrics.average_precision_score(y_true, ...)	Compute average precision (AP) from prediction scores.
metrics.balanced_accuracy_score(y_true, ...)	Compute the balanced accuracy.
metrics.brier_score_loss(y_true, y_prob, *)	Compute the Brier score loss.
metrics.classification_report(y_true, y_pred, *)	Build a text report showing the main classification metrics.
metrics.cohen_kappa_score(y1, y2, *[, ...])	Cohen's kappa: a statistic that measures inter-annotator agreement.
metrics.confusion_matrix(y_true, y_pred, *)	Compute confusion matrix to evaluate the accuracy of a classification.
metrics.dcg_score(y_true, y_score, *[, k, ...])	Compute Discounted Cumulative Gain.
metrics.det_curve(y_true, y_score[, ...])	Compute error rates for different probability thresholds.
metrics.f1_score(y_true, y_pred, *[, ...])	Compute the F1 score, also known as balanced F-score or F-measure.
metrics.fbeta_score(y_true, y_pred, *, beta)	Compute the F-beta score.
metrics.hamming_loss(y_true, y_pred, *[, ...])	Compute the average Hamming loss.
metrics.hinge_loss(y_true, pred_decision, *)	Average hinge loss (non-regularized).
metrics.jaccard_score(y_true, y_pred, *[, ...])	Jaccard similarity coefficient score.
metrics.log_loss(y_true, y_pred, *[, eps, ...])	Log loss, aka logistic loss or cross-entropy loss.
metrics.matthews_corrcoef(y_true, y_pred, *)	Compute the Matthews correlation coefficient (MCC).

https://scikit-learn.org/stable/modules/classes.html#module-sklearn.metrics

metrics.matthews_corrcoef(y_true, y_pred, *)	Compute the Matthews correlation coefficient (MCC).
metrics.multilabel_confusion_matrix(y_true, ...)	Compute a confusion matrix for each class or sample.
metrics.ndcg_score(y_true, y_score, *[, k, ...])	Compute Normalized Discounted Cumulative Gain.
metrics.precision_recall_curve(y_true, ...)	Compute precision-recall pairs for different probability thresholds.
metrics.precision_recall_fscore_support(...)	Compute precision, recall, F-measure and support for each class.
metrics.precision_score(y_true, y_pred, *[, ...])	Compute the precision.
metrics.recall_score(y_true, y_pred, *[, ...])	Compute the recall.
metrics.roc_auc_score(y_true, y_score, *[, ...])	Compute Area Under the Receiver Operating Characteristic Curve (ROC AUC) from prediction scores.
metrics.roc_curve(y_true, y_score, *[, ...])	Compute Receiver operating characteristic (ROC).
metrics.top_k_accuracy_score(y_true, y_score, *)	Top-k Accuracy classification score.
metrics.zero_one_loss(y_true, y_pred, *[, ...])	Zero-one classification loss.

河北师范大学软件学院 Software College of Hebei Normal University

二分类的情况之一：
非对称类别(是 vs. 非)

(人脸vs.非人脸； 垃圾邮件vs.非垃圾邮件)

- 什么是混淆矩阵？如何计算混淆矩阵？
- 如何利用混淆矩阵生成若干评价指标？
- 什么是ROC曲线？如何利用有限规模测试样本绘制ROC曲线？
- 如何基于ROC曲线进行模型的评价与比较？
- 如何计算ROC曲线下面积(即：AUC值，也称AUC_ROC)？
- 什么是P-R曲线？如何利用有限规模测试样本绘制P-R曲线？
- 如何基于P-R曲线实现模型的定性或定量评价？
- 如何计算P-R曲线下面积(即：AUC_PR)？
- 哪种曲线更适合非对称类别分类模型的性能评价？为什么？

河北师范大学软件学院 Software College of Hebei Normal University

方式1.基于混淆矩阵的两类别分类模型的性能评价

方式2.基于ROC曲线/P-R曲线的性能评价

河北师范大学软件学院 Software College of Hebei Normal University

方式1.基于混淆矩阵的两类别分类模型的性能评价

混淆矩阵： confusion matrix

(1)样本的两种自然类别状态

通常设定感兴趣的一类为正类

类别标号	类别各种名称			
1	Positive (P)	正	阳性	Case Samples (病理样本)
0或-1或2	Negetive (N)	负	阴性	Control Samples (对照样本)

河北师范大学软件学院 Software College of Hebei Normal University

(2)两类决策的混淆矩阵(confusion matrix)

正确分类；错误分类；决策阈值

自然状态 True Value / Actual Value	预测输出(Predicated Outcome) Positive (Predicated 1)	Negetive (Predicated 0 or -1)
Positive (True 1)	a: True Positive(TP) 真阳性 真正类 Hits	b: False Negetive(FN) 假阴性 Misses 假负类
Negetive (True 0 或-1)	c: False Positive(FP) 假阳性 假正类 False Alarms	d: True Negetive (TN) 真阴性 真负类 True Rejections

概率形式(正确分类；错误分类)----条件概率

自然状态 True Value Actual Value	预测输出(Predicated Outcome)	
	Positive (Predicated 1)	Negetive (Preducated 0 or -1)
Positive (True 1)	Sn--灵敏度 P(Predicated P|True P)	β--假阴性率 P(Predicated N|True P)
Negetive (True 0 或-1)	α--假阳性率 P(Predicated P|True N)	Sp--特异度 P(Predicated N|True N)

$$P(Predicated\ \ P\,|\,True\ \ P)+P(Predicated\ \ N\,|\,True\ \ P)=1$$
$$P(Predicated\ \ P\,|\,True\ \ N)+P(Predicated\ \ N\,|\,True\ \ N)=1$$

(3)基于两类别混淆矩阵的评价指标

[1](阳性类)查准率,精度(Precision,P)　$Precision=\frac{TP}{TP+FP}$

[2](阳性类)查全率,召回率(Recall,R),灵敏度(Sensitivity)

命中率，真阳性率(TruePositiveRate)　$S_n=Recall=\frac{TP}{TP+FN}$

[3]特异度(Specificity),真阴性率(TrueNegetiveRate)　$S_p=\frac{TN}{TN+FP}$

[4]假阴性率(FalseNegetiveRate),漏报率(MissedRate)　$\beta=\frac{FN}{FN+TP}$
第2类错误率

[5]假阳性率(FalsePositiveRate),虚警率(FalseAlarmRate)　$\alpha=\frac{FP}{TN+FP}$
第1类错误率

[6](阳性类的)F_β-Score以及F_1-Score

----Precision与Recall的调和平均

$$F_\beta=\frac{1}{\frac{1}{\beta^2+1}\left(\beta^2\frac{1}{R}+1\frac{1}{P}\right)}=\frac{(\beta^2+1)P\cdot R}{\beta^2P+R}$$

$0<\beta<1$,更看重P

$\beta>1$,　更看重R

β=1时，$F_1=\frac{2}{\frac{1}{R}+\frac{1}{P}}=\frac{2P\cdot R}{P+R}=\frac{2TP}{2TP+FP+FN}=\frac{2TP}{测试样本总数+TP-TN}$

[7]马修相关系数(Matthews Correlation Coefficient,MCC)

类别不均衡的两类别分类，公平度量模型预测能力

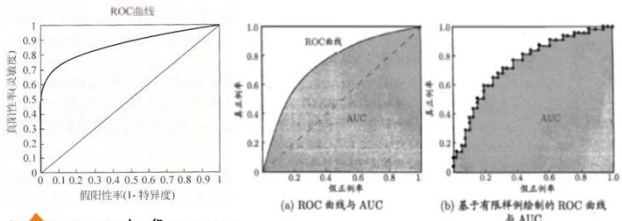
$$MCC=\frac{TP\cdot TN-FP\cdot FN}{\sqrt{(TP+FN)(TN+FP)(TP+FP)(TN+FN)}}$$

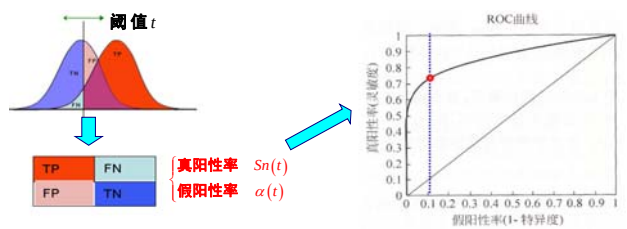
方式1.基于混淆矩阵的两类别分类模型的性能评价

方式2.基于ROC曲线/P-R曲线的性能评价

(1)ROC曲线(*Receiver Operating Characteristic* Curves)及绘制

The ROC curve is a graphical plot of the sensitivity, or true positive rate (y-axis), vs. false positive rate (1 − specificity or 1 − true negative rate) (x-axis), for a binary classifier system as its discrimination threshold is varied.





决策阈值$t\Rightarrow$

特异度(*Specificity*,真阴性率)　$Sp(t)=\frac{TN(t)}{FP(t)+TN(t)}$

假阴性率(第二类错误率)　$\beta(t)=\frac{FN(t)}{FN(t)+TP(t)}$

假阳性率(第一类错误率，误报率)　$\alpha(t)=1-Sp(t)=\frac{FP(t)}{FP(t)+TN(t)}$

真阳性率(*Sensitivity*,敏感度，召回率)　$Sn(t)=1-\beta(t)=\frac{TP(t)}{FN(t)+TP(t)}$

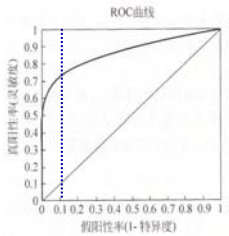
(2)基于ROC曲线的分类器性能评价

第一，根据分类器的设计指标要求，由ROC曲线确定真实分类器的工作点

类条件密度已知，改变决策阈值，得ROC曲线；

ROC曲线上每个点，均对应一对(灵敏度，特异度)；

根据分类器对两种指标要求，确定曲线工作点



ROC曲线越往左上靠，ROC曲线越优，

存在一个平衡点，使这个平衡点就是max(TPR-FPR)所对应的分类器阈值

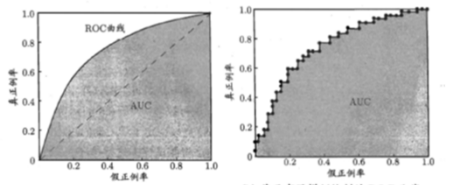
(2)基于ROC曲线的分类器性能评价

第二，类别数=2的分类器性能评价-AUC

AUC--ROC曲线下的面积AUC (Area Under ROC Curves)

ROC曲线越靠近坐标系的左上角，AUC值越大，分类器性能越好

$$AUC = \frac{1}{2}\sum_{i=1}^{N-1}((\alpha_{i+1}-\alpha_i)(R_{i+1}+R_i))$$ AUC-ROC



(a) ROC 曲线与 AUC　(b) 基于有限样例绘制的 ROC 曲线与 AUC

河北师范大学软件学院 Software College of Hebei Normal University

自然状态 True Value Actual Value	预测输出(Predicated Outcome) Positive (Predicated 1)	Negetive (Preducated 0 or -1)
Positive (True 1)	Sn P(Predicated P|True P)	β P(Predicated N|True P)
Negetive (True 0 或-1)	α P(Predicated P|True N)	Sp P(Predicated N|True N)

例：疾病诊断

“positive(阳性类)”--疾病　　“negative(阴性类)”--正常

(1)不希望漏诊疾病

希望结果：　真阳性率=1

不希望结果：真阴性率降低(极端值=0)　假阳性率升高(极端值=1)

ROC曲线靠近左上角

(2)不希望误判为疾病

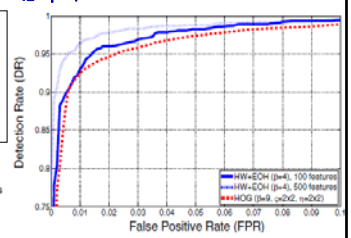
希望结果：　真阴性率=1，假阳性率=0

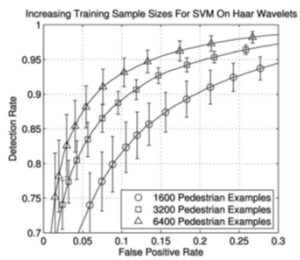
不希望结果：真阳性率降低(极端值=0)　假阴性率升高(极端值=1)

河北师范大学软件学院 Software College of Hebei Normal University

(2)基于ROC曲线的分类器性能评价

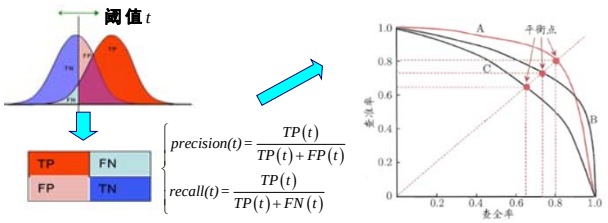
- 可用于不同分类器性能比较
- 可针对同种分类器，进行特征选择、样本规模等优劣比较。





- 可以使用AUC值，单独评价每个特征的类鉴别能力
- 可以结合多个特征的多条ROC曲线横向比较各特征的类区分性

(3)P-R曲线(Precision-Recall Curve)及其AUC-PR



- 定性比较：如果一个学习器的P-R曲线被另一学习器的P-R曲线完全包住，可断言后者性能优于前者。例如：A、B学习器优于学习器C。
- 定量评价方式1：借助不同P-R曲线下方的面积值，进行横向比较
- 定量评价方式2：结合平衡点或者F1值。

----平衡点(BEP, break even point)是P-R曲线与直线P=R的交点。

P=R值越大，学习器性能较好。

----F1 = 2 * P * R /（P + R）。F1值越大，该学习器性能越好。

二分类的情况之二：

对称类别(第1类vs.第2类)

(例：猫vs.狗)

- 什么是混淆矩阵？该混淆矩阵与“非对称的二分类”混淆矩阵的区别？
- 如何产生混淆矩阵？如何基于混淆矩阵产生评价指标？
- 什么是ROC曲线？如何利用有限规模测试样本绘制ROC曲线？
- 对于对称类别的二分类问题，一个测试集可以同时得到几条ROC曲线？
- 如何结合这些ROC曲线进行分类模型的性能评价？

河北师范大学软件学院 Software College of Hebei Normal University

(1) 混淆矩阵(confusion matrix)

第1类；第2类；决策阈值

$A=[a_{ij}]_{2\times 2}$

真实类别	预测类别	
	第1类	第2类
第1类	a_{11}	a_{12}
第2类	a_{21}	a_{22}

a_{ij}----参与测试的样本中，真实类别为第i类，但预测为第j类的样本数

测试集 $D_{\text{test}}=\{(x_i,y_i),i=1,\ldots,N_{test}\}$

总体预测错误率：

$$Err_{test}=\frac{1}{N_{\text{test}}}\sum_{i=1}^{N_{test}}I(y_i\neq\hat{y}_i)\times 100\%$$

总体预测正确率：

$$Acc_{test}=\frac{1}{N_{\text{test}}}\sum_{i=1}^{N_{test}}I(y_i=\hat{y}_i)\times 100\%$$

(2)基于两类别混淆矩阵的评价指标

[1]总体正确率,准确率,准确度(OverallAccuracy) $Accuracy=\frac{\sum_{i=1}^{2}a_{ii}}{\sum_{i=1}^{2}\sum_{j=1}^{2}a_{ij}}\times 100\%$

[2]总体错误率(ErrorRate) $Error=\frac{a_{12}+a_{21}}{\sum_{i=1}^{2}\sum_{j=1}^{2}a_{ij}}\times 100\%=1-Accuracy$

[3]第i类查准率,精度(Precision,P) $P_i=\frac{a_{ii}}{\sum_{j=1}^{2}a_{ji}}\quad i=1,2$

[4]第i类查全率,召回率(Recall,R),正确率 $R_i=\frac{a_{ii}}{\sum_{j=1}^{2}a_{ij}}=Acc_i\quad i=1,2$

[5]第i类F_β-Score以及F_1-Score--P_i与R_i的调和平均

$$F_\beta^{(i)}=\frac{(\beta^2+1)P_i\cdot R_i}{\beta^2P_i+R_i}\qquad i=1,2$$

$0<\beta<1$,更看重P_i

$\beta>1$, 更看重R_i

β=1时，$F_1^{(i)}=\frac{2P_i\cdot R_i}{P_i+R_i}$

[6]宏查准率、宏查全率 $Macro\_P=\frac{1}{2}\sum_{i=1}^{2}P_i$ $Macro\_R=\frac{1}{2}\sum_{i=1}^{2}R_i$

[7]C=2个类别的宏平均F_1-Score(Macro-averaging F_1)

$$Macro\_F_1=\frac{2\times Macro\_P\times Macro\_R}{Macro\_P+Macro\_R}$$

[8]各类预测正确率的几何平均--类别不均衡的评价

$$Acc_{G-mean}=\left(\prod_{i=1}^{2}Acc_i\right)^{\frac{1}{2}}$$

[9]马修相关系数

(Matthews Correlation Coefficient,MCC)

类别不均衡的两类别分类，公平度量模型预测能力

$$MCC=\frac{a_{11}\cdot a_{22}-a_{12}\cdot a_{21}}{\sqrt{(a_{11}+a_{12})(a_{21}+a_{22})(a_{11}+a_{21})(a_{12}+a_{22})}}$$

主要内容

多类别分类(C>2)

- 如何结合测试集的类别标签信息以及预测标签，产生多类别分类的混淆矩阵？
- 如何结合C阶混淆矩阵，产生若干评价指标？
- 一个C阶混淆矩阵可生成C个“第i类vs.非第i类”二阶混淆矩阵

- 一个测试集可以同时得到几条ROC曲线？
- 如何结合这些ROC曲线进行分类模型的性能评价？

(1) 混淆矩阵(confusion matrix)

C阶矩阵： $A = [a_{ij}]_{C\times C}$

a_{ij}----测试样本集内，真实类别为第i类，但预测为第j类的样本数

真实类别为第i类的样本总数：$\sum_{j=1}^{C} a_{ij}$

预测类别为第j类的样本总数：$\sum_{i=1}^{C} a_{ij}$

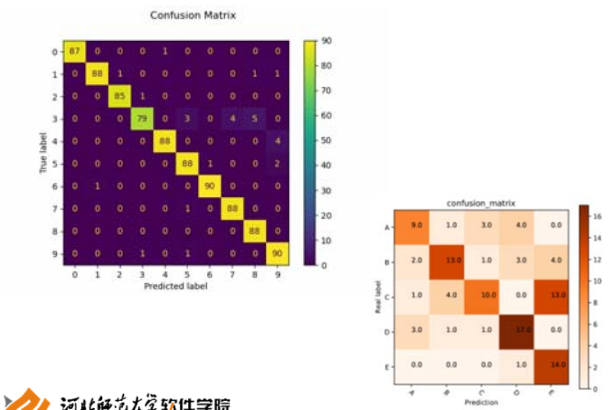
测试样本总数： $\sum_{i=1}^{C}\sum_{j=1}^{C} a_{ij}$

例：基于混淆矩阵的手写体数字识别

混淆矩阵为10行*10列

	class j predicted by a classifier									
true class i	'0'	'1'	'2'	'3'	'4'	'5'	'6'	'7'	'8'	'9'
'0'	97	0	0	0	0	0	1	0	0	1
'1'	0	98	0	0	1	0	0	1	0	0
'2'	0	0	96	1	0	1	0	1	0	0
'3'	0	0	2	95	0	1	0	0	1	0
'4'	0	0	0	0	98	0	0	0	0	2
'5'	0	0	0	1	0	97	0	0	0	0
'6'	1	0	0	0	0	1	98	0	0	0
'7'	0	0	1	0	0	0	0	98	0	0
'8'	0	0	0	1	0	0	1	0	96	1
'9'	1	0	0	0	3	1	0	0	0	95

sklearn.metrics.ConfusionMatrixDisplay



测试集 $D_{\text{test}}=\{(x_i, y_i), i = 1, \ldots, N_{test}\}$

总体预测错误率：

$$Err_{test} = \frac{1}{N_{\text{test}}} \sum_{i=1}^{N_{test}} I(y_i \neq \hat{y}_i) \times 100\%$$

总体预测正确率：

$$Acc_{test} = \frac{1}{N_{\text{test}}} \sum_{i=1}^{N_{test}} I(y_i = \hat{y}_i) \times 100\%$$

(2)基于多类别混淆矩阵的评价指标

[1]总体正确率,准确率,准确度(OverallAccuracy) $Accuracy=\frac{\sum_{i=1}^{C} a_{ii}}{\sum_{i=1}^{C}\sum_{j=1}^{C} a_{ij}}\times 100\%$

[2]总体错误率(ErrorRate) $Error = 1 - Accuracy$

[3]第i类查准率,精度(Precision,P) $P_i = \frac{a_{ii}}{\sum_{j=1}^{C} a_{ji}}$ $i = 1,\ldots,C$

[4]第i类查全率,召回率(Recall,R),正确率 $R_i = \frac{a_{ii}}{\sum_{j=1}^{C} a_{ij}} = Acc_i$ $i = 1,\ldots,C$

[5]宏查准率、宏查全率 $Macro\_P = \frac{1}{C}\sum_{i=1}^{C} P_i$ $Macro\_R = \frac{1}{C}\sum_{i=1}^{C} R_i$

[6]第i类F_β-Score以及F_1-Score--P_i与R_i的调和平均

$$F_\beta^{(i)}=\frac{(\beta^2+1)P_i\cdot R_i}{\beta^2 P_i+R_i}\qquad i=1,\ldots,C$$

$0<\beta<1$,更看重P_i

$\beta>1$, 更看重R_i

β=1时，$F_1^{(i)}=\frac{2P_i\cdot R_i}{P_i+R_i}$

[7]C个类别的宏平均F_1-Score(Macro-averaging F_1)

$$Macro\_F_1=\frac{2\times Macro\_P\times Macro\_R}{Macro\_P+Macro\_R}$$

[8]由C阶混淆矩阵，得到C个"第i类$vs.$非第i类"混淆矩阵

将这C个混淆矩阵取平均，得到平均混淆矩阵

微查准率/微精度($Precision$) $\quad micro\_P=\frac{\overline{TP}}{\overline{TP}+\overline{FP}}$

微查全率/微召回率($Recall$) $\quad micro\_R=\frac{\overline{TP}}{\overline{TP}+\overline{FN}}$

微$F_1-Score$ $\quad micro\_F_1=\frac{2micro\_P\cdot micro\_R}{micro\_P+micro\_R}$

[9] *Kappa*系数(*Cohen's kappa*)

Cohen's kappa measures the agreement between two raters who each classify N items into C mutually exclusive categories.

$$K=\frac{p_0-p_e}{1-p_e}=\frac{\frac{\sum_{i=1}^{C}a_{ii}}{\sum_{i=1}^{C}\sum_{j=1}^{C}a_{ij}}-\frac{\sum_{i=1}^{C}\left[\sum_{j=1}^{C}a_{ij}\sum_{k=1}^{C}a_{ki}\right]}{\left(\sum_{i=1}^{C}\sum_{j=1}^{C}a_{ij}\right)^2}}{1-\frac{\sum_{i=1}^{C}\left[\sum_{j=1}^{C}a_{ij}\sum_{k=1}^{C}a_{ki}\right]}{\left(\sum_{i=1}^{C}\sum_{j=1}^{C}a_{ij}\right)^2}}\qquad -1\le K\le 1$$

考虑两个极端：
没有错分；完全错分

理想情况，完全一致，$K=1$;

K值越小，越不一致，甚至为负.

基于kappa系数一致性的不同等级
- [0,0.20) 极低一致性(slight)
- [0.20,0.40)一般一致性(fair)
- [0.40~0.60) 中等的一致性(moderate)
- [0.6~0.80) 高度的一致性(substantial)
- [0.8~1] 几乎完全一致(almost perfect)

关于p_0, p_e的具体说明：

(1) p_0--实际一致率(总体预测正确率) $p_0=\frac{\sum_{i=1}^{C}a_{ii}}{\sum_{i=1}^{C}\sum_{j=1}^{C}a_{ij}}$

(2) p_e--理论一致率(机遇一致性)

1-p_e--非机遇一致率

p_e表示一个样本被抽取的类别与决策类别一致的概率

$$p_e=\sum_{i=1}^{C}p_{ei}=\frac{\sum_{i=1}^{C}\left[\sum_{j=1}^{C}a_{ij}\sum_{k=1}^{C}a_{ki}\right]}{\left(\sum_{i=1}^{C}\sum_{j=1}^{C}a_{ij}\right)^2}=\sum_{i=1}^{C}\frac{\sum_{j=1}^{C}a_{ij}}{\sum_{i=1}^{C}\sum_{j=1}^{C}a_{ij}}\cdot\frac{\sum_{k=1}^{C}a_{ki}}{\sum_{i=1}^{C}\sum_{j=1}^{C}a_{ij}}$$

第i类样本被观测到的概率

样本决策为第i类的概率

被抽取的类别与决策类别一致的概率

观测样本为第i类并同时被决策为第i类的概率

主要内容

1. 模型的学习能力与泛化能力
2. 交叉验证
3. (基于测试集的)模型评价
   - 3.1 基于测试集的二分类模型评价
   - 3.2 基于测试集的多分类模型的评价
   - 3.3 基于测试集的实值函数预测模型评价

说明：聚类模型的评价在后续聚类模型学习中给出。

Regression metrics

See the Regression metrics section of the user guide for further details.

metrics.explained_variance_score(y_true, ...)	Explained variance regression score function.
metrics.max_error(y_true, y_pred)	The max_error metric calculates the maximum residual error.
metrics.mean_absolute_error(y_true, y_pred, *)	Mean absolute error regression loss.
metrics.mean_squared_error(y_true, y_pred, *)	Mean squared error regression loss.
metrics.mean_squared_log_error(y_true, y_pred, *)	Mean squared logarithmic error regression loss.
metrics.median_absolute_error(y_true, y_pred, *)	Median absolute error regression loss.
metrics.mean_absolute_percentage_error(...)	Mean absolute percentage error (MAPE) regression loss.
metrics.r2_score(y_true, y_pred, *[, ...])	R^2 (coefficient of determination) regression score function.
metrics.mean_poisson_deviance(y_true, y_pred, *)	Mean Poisson deviance regression loss.
metrics.mean_gamma_deviance(y_true, y_pred, *)	Mean Gamma deviance regression loss.
metrics.mean_tweedie_deviance(y_true, y_pred, *)	Mean Tweedie deviance regression loss.
metrics.d2_tweedie_score(y_true, y_pred, *)	D^2 regression score function, percentage of Tweedie deviance explained.
metrics.mean_pinball_loss(y_true, y_pred, *)	Pinball loss for quantile regression.

测试样本集 $D_{Test}=\{(x_i,y_i),i=1,\cdots,N\}$

1.均方误差（*Mean Squared Error*，*MSE*）

sklearn.metrics.mean_squared_error

$$MSE(\boldsymbol{y},\hat{\boldsymbol{y}})=\frac{1}{N}\sum_{i=1}^{N}[y_i-\hat{y}_i]^2$$

2. 均方根误差（*Root Mean Squaed Error*，*RMSE*）

$$RMSE(\boldsymbol{y},\hat{\boldsymbol{y}})=\sqrt{\frac{1}{N}\sum_{i=1}^{N}[y_i-\hat{y}_i]^2}=\sqrt{MSE}$$

3. 平均绝对误差（*Mean Absolute Error*，*MAE*）

sklearn.metrics.mean_absolute_error

$$MAE(\boldsymbol{y},\hat{\boldsymbol{y}})=\frac{1}{N}\sum_{i=1}^{N}|y_i-\hat{y}_i|$$

4. 中值绝对误差（*Median Absolute Error*）

最大绝对误差（*Max Absolute Error*）

sklearn.metrics.median_absolute_error
sklearn.metrics.max_error

$$Median\_AE(\boldsymbol{y},\hat{\boldsymbol{y}})=\underset{i\in\{1,\ldots,N\}}{\text{median}}\{|y_i-\hat{y}_i|\}$$

$$Max\_AE(\boldsymbol{y},\hat{\boldsymbol{y}})=\max_{i\in\{1,\ldots,N\}}\{|y_i-\hat{y}_i|\}$$

5. 均方根对数误差（*Root Mean Squaed Logarithmie Error*，***RMSLE***）

sklearn.metrics.mean_squared_log_error

$$RMSLE(\boldsymbol{y},\hat{\boldsymbol{y}})=\sqrt{\frac{1}{N}\sum_{i=1}^{N}[\log(y_i+1)-\log(\hat{y}_i+1)]^2}$$
$$=\sqrt{\frac{1}{N}\sum_{i=1}^{N}\left[log\frac{1+y_i}{1+\hat{y}_i}\right]^2}$$

该评价指标的应用前提：

- 样本的“标签值”为重尾分布(Heavy-tailed distribution)
  此时，先进行对数运算再取RMSE,可有效避免模型评价时，少数极大值的影响。
- 此外，应确保标签值与预测值为非负。

6. 平均绝对百分比误差（*Mean Absolute Percentage Error*，*MAPE*）

sklearn.metrics.mean_absolute_percentage_error

$$MAPE(\boldsymbol{y},\hat{\boldsymbol{y}})=100\%\times\frac{1}{N}\sum_{i=1}^{N}\frac{|y_i-\hat{y}_i|}{\max\{\varepsilon,|y_i|\}}$$

为避免被0除，引入小正数

该指标的取值范围[0,+∞)

- 若MAPE =0，则预测模型为完美模型；
- 若MAPE > 100 %, 则预测模型为劣质模型。

注意：当真值存在取值0情况时，出现分母0除问题，上述评价指标将不可用。

7. 对称平均绝对百分比误差
（*Symmetric Mean Absolute Percentage Error*，*SMAPE*）

$$SMAPE(\boldsymbol{y},\hat{\boldsymbol{y}})=100\%\times\frac{1}{N}\sum_{i=1}^{N}2\left|\frac{y_i-\hat{y}_i}{|y_i|+|\hat{y}_i|}\right|$$

该指标的取值范围[0,+∞)

- 若SMAPE =0，则预测模型为完美模型；
- 若SMAPE > 100 %, 则预测模型为劣质模型。

注意：当真值及预测值同为0时，出现分母0除问题，上述评价指标将不可用。

8. 决定系数

（可决系数*coefficient of determination*：R^2）

sklearn.metrics.r2_score

$$R^2(\boldsymbol{y},\hat{\boldsymbol{y}})=1-\frac{MSE(y,\hat{y})}{Var(y)}=1-\frac{\frac{1}{N}\sum_{i=1}^{N}[y_i-\hat{y}_i]^2}{\frac{1}{N}\sum_{i=1}^{N}[y_i-\bar{y}]^2}$$
$$=1-\frac{SSE}{SST}=1-\frac{\sum_{i=1}^{N}[y_i-\hat{y}_i]^2}{\sum_{i=1}^{N}[y_i-\bar{y}]^2}$$

决定系数取值在$(-\infty,1]$之间：

- 越接近于1，说明模型的预测效果越好；
- 越接近于0，说明模型的预测效果越差；
- 若取值为负值，说明模型的效果非常差。

9. 解释方差 (*Explained Variance*)

sklearn.metrics.explained_variance_score

$$explained\_variance\_score(\boldsymbol{y}, \hat{\boldsymbol{y}}) = 1 - \frac{Var(y - \hat{y})}{Var(y)}$$

解释方差取值在$(-\infty, 1]$之间：

- 越接近于1，说明模型的预测效果越好；
- 越小于1，说明模型的预测效果越差.

本章思考题(1)

1. 什么是模型的学习能力？泛化能力？
2. 区分：训练集、测试集、估计集、验证集。
3. 结合分类与回归问题，给定已知答案的样本集，如何基于K-fold CV或LOOCV进行模型的评价？如何得到最终的模型？K值一般多大？
4. 给定已知答案的训练样本集，请结合分类与回归问题，如何基于K-fold CV或LOOCV进行模型的选择？K值一般多大？
5. 区分hold out , bootstrapping, K-fold CV与LOOCV的使用场合。

本章思考题(2)

1.对于两类别(非对称类别/对称类别)、多类别的分类问题，如何分类模型对测试集的预测结果，得到相应的混淆矩阵？上述情况下的混淆矩阵有什么区别？如何计算混淆矩阵？如何以图的方式可视化混淆矩阵？

2. 如何针对上述三种情况，利用混淆矩阵生成若干评价指标？

3. 什么是ROC曲线？如何利用有限规模测试样本绘制ROC曲线？
   如何基于ROC曲线进行模型的评价与比较？
   如何计算ROC曲线下面积(即：AUC值，也称AUC_ROC)？

4. 什么是P-R曲线？如何利用有限规模测试样本绘制P-R曲线？
   如何基于P-R曲线实现模型的定性或定量评价？
   如何计算P-R曲线下面积(即：AUC_PR)？

5. 哪种曲线更适合非对称类别分类模型的性能评价？为什么？

本章思考题(3)

1. 理解并掌握回归模型的常用评价指标：

- 决定系数、方差解释比
- 平均绝对误差、最大绝对误差、中值绝对误差
- 平均平方误差、均方根误差
- 均方根对数误差及应用场合
- 平均绝对百分比误差、对称平均绝对百分比误差
- …