机器学习库Scikit-learn

本章目录

01 Scikit-learn概述

02 Scikit-learn主要用法

1.Scikit-learn概述

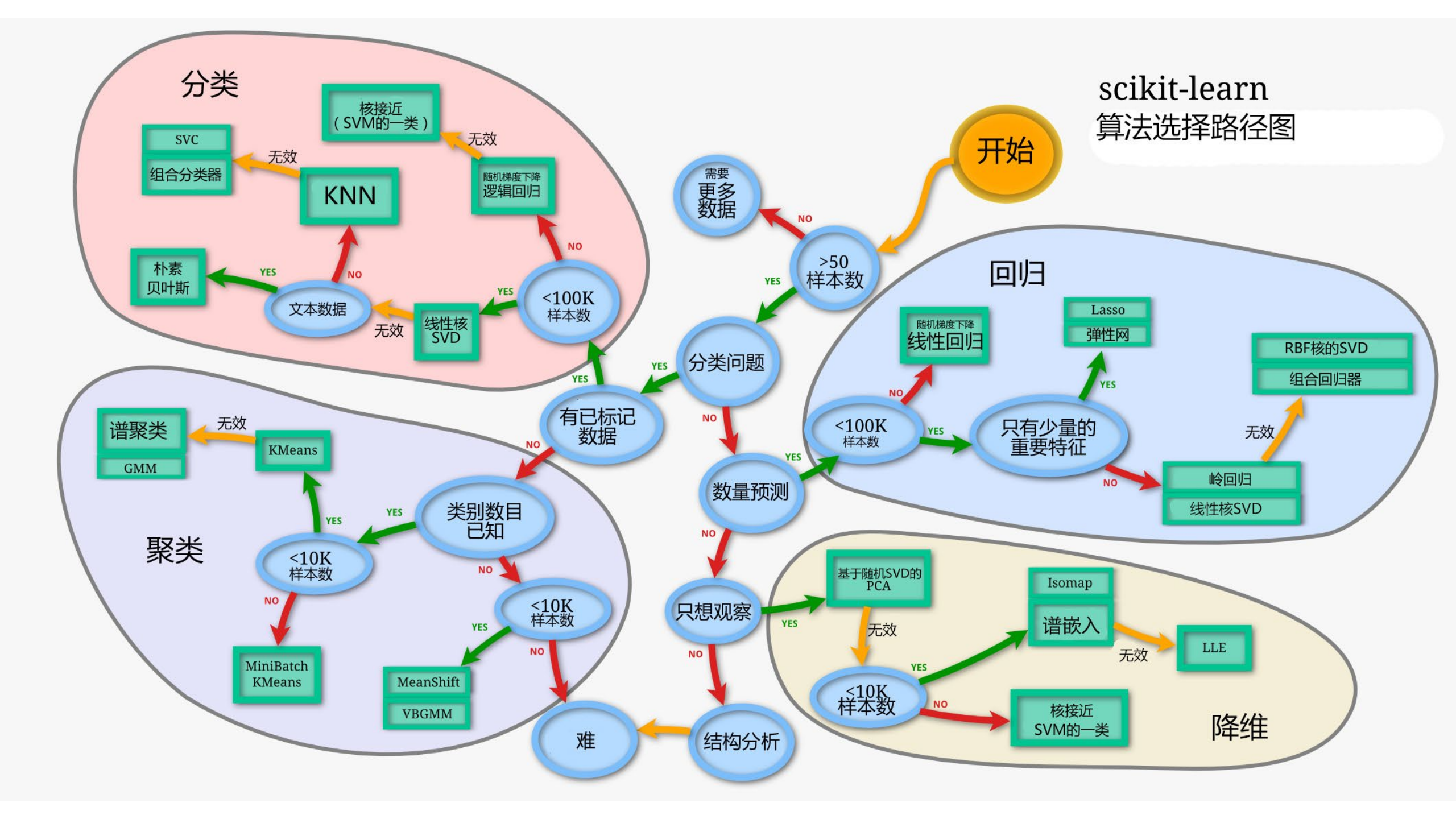
01 Scikit-learn概述

02 Scikit-learn主要用法

1.Scikit-learn概述

Scikit-learn是基于NumPy、 SciPy和 Matplotlib的开源Python机器学习包,它封装了一系列数据预处理、机器学习算法、模型选择等工具,是数据分析师首选的机器学习工具包。

自2007年发布以来，scikit-learn已经成为Python重要的机器学习库了，scikit-learn简称sklearn，支持包括分类，回归，降维和聚类四大机器学习算法。还包括了特征提取，数据处理和模型评估三大模块。


scikit-learn
算法选择路径图
开始
>50 样本数
需要更多数据
分类问题
有已标记数据
<100K 样本数
文本数据
朴素贝叶斯
线性核SVD
随机梯度下降逻辑回归
核接近（SVM的一类）
KNN
SVC
组合分类器
分类
类别数目已知
<10K 样本数
MiniBatch KMeans
MeanShift
VBGMM
KMeans
谱聚类
GMM
聚类
数量预测
<100K 样本数
随机梯度下降线性回归
只有少量的重要特征
Lasso
弹性网
岭回归
线性核SVD
RBF核的SVD
组合回归器
回归
只想观察
基于随机SVD的PCA
<10K 样本数
谱嵌入
Isomap
LLE
核接近SVM的一类
降维
结构分析
难
YES
NO
无效

2.Scikit-learn主要用法

01 Scikit-learn概述

02 Scikit-learn主要用法

2.Scikit-learn主要用法

符号标记

`X_train`	训练数据.
`X_test`	测试数据.
`X`	完整数据.
`y_train`	训练集标签.
`y_test`	测试集标签.
`y`	数据标签.

2.Scikit-learn主要用法

基本建模流程

导入工具包

```
from sklearn import datasets, preprocessing
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import r2_score
```

2.Scikit-learn主要用法

加载数据

✓Scikit-learn支持以NumPy的arrays对象、Pandas对象、SciPy的稀疏 矩阵及其他可转换为数值型arrays的数据结构作为其输入，前提是数据必须是数值型的

✓sklearn.datasets模块提供了一系列加载和获取著名数据集如鸢尾花、波士顿房价、Olivetti人脸、MNIST数据集等的工具，也包括了一些toy data如S型数据等的生成工具

```
from sklearn.datasets import load_iris

iris = load_iris()
X = iris.data
y = iris.target
```

2.Scikit-learn主要用法

数据划分

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test =  train_test_split(X, y,
random_state=12, stratify=y,  test_size=0.3)
```

将完整数据集的70%作为训练集，30%作为测试集，并使得测试集和训练集中各类别数据的比例与原始数据集比例一致（stratify分层策略），另外可通过设置 shuffle=True 提前打乱数据

2.Scikit-learn主要用法

数据预处理

使用Scikit-learn进行数据标准化

```
from sklearn.preprocessing import StandardScaler
```

构建转换器实例

```
scaler = StandardScaler()
```

拟合及转换

```
scaler.fit_transform(X_train)
```

Z-Score标准化

$$x^* = \frac{x - \mu}{\sigma}$$

$$\sigma^2 = \frac{1}{m}\sum_{i=1}^{m}(x^{(i)} - \mu)^2$$

$$\mu = \frac{1}{m}\sum_{i=1}^{m}x^{(i)}$$

处理后的数据均值为0，方差为1

2.Scikit-learn主要用法

数据预处理

使用Scikit-learn进行数据变换

最小最大标准化	MinMaxScaler
One-Hot编码	OneHotEncoder
归一化	Normalizer
二值化（单个特征转换）	Binarizer
标签编码	LabelEncoder
缺失值填补	Imputer
多项式特征生成	PolynomialFeatures

归一化（最大 - 最小规范化）

$$x^* = \frac{x - x_{\min}}{x_{\max} - x_{\min}}$$

将数据映射到[0,1]区间

2.Scikit-learn主要用法

特征选择

```
from sklearn import feature_selection as fs
fs.SelectKBest(score_func, k)
```

过滤式（Filter），保留得分排名前k的特征（top k方式）

```
fs.RFECV(estimator,  scoring="r2")
```

封装式（Wrap- per），结合交叉验证的递归特征消除法，自动选择最优特征个数

```
fs.SelectFromModel(estimator)
```

嵌入式（Embedded），从 模型中自动选择特征，任何具有coef_或者feature_importances_的 基模型都可以作为estimator参数传入

2.Scikit-learn主要用法

监督学习算法-回归

```
from sklearn.linear_model import LinearRegression
```

构建模型实例

```
lr = LinearRegression(normalize=True)
```

训练模型

```
lr.fit(X_train, y_train)
```

作出预测

```
y_pred = lr.predict(X_test)
```

LASSO	linear_model.Lasso
Ridge	linear_model.Ridge
ElasticNet	linear_model.ElasticNet
回归树	tree.DecisionTreeRegressor

2.Scikit-learn主要用法

监督学习算法-分类

```
from sklearn.tree import DecisionTreeClassifier
clf = DecisionTreeClassifier(max_depth=5)
clf.fit(X_train, y_train)
y_pred = clf.predict(X_test)
y_prob = clf.predict_proba(X_test)
```

使用决策树分类算法解决二分类问题， y_prob 为每个样本预测为“0”和“1”类的概率

1.Scikit-learn概述

监督学习算法-分类

逻辑回归	`linear_model.LogisticRegression`
支持向量机	`svm.SVC`
朴素贝叶斯	`naive_bayes.GaussianNB`
K近邻	`neighbors.NearestNeighbors`

2.Scikit-learn主要用法

监督学习算法-集成学习

sklearn.ensemble模块包含了一系列基于集成思想的分类、回归和离群值检测方法.

```
from sklearn.ensemble import RandomForestClassifier
clf = RandomForestClassifier(n_estimators=20)
clf.fit(X_train, y_train)
y_pred = clf.predict(X_test)
y_prob = clf.predict_proba(X_test)
```

AdaBoost	ensemble.AdaBoostClassifier ensemble.AdaBoostRegressor
基于梯度提升	ensemble.GradientBoostingClassifier ensemble.GradientBoostingRegressor

2.Scikit-learn主要用法

无监督学习算法

sklearn.cluster模块包含了一系列无监督聚类算法.

```
from sklearn.cluster import KMeans
```

构建聚类实例

```
kmeans = KMeans(n_clusters=3, random_state=0)
```

拟合

```
kmeans.fit(X_train)
```

预测

```
kmeans.predict(X_test)
```

2.Scikit-learn主要用法

无监督学习算法-降维

sklearn.decomposition 模块包含了一系列无监督降维算法

```
from sklearn.decomposition import PCA
```

导入PCA库，设置主成分数量为3，n_components代表主成分数量

```
pca = PCA(n_components=3)
```

训练模型

```
pca.fit(X)
```

投影后各个特征维度的方差比例(这里是三个主成分)

```
print(pca.explained_variance_ratio_)
```

投影后的特征维度的方差

```
print(pca.explained_variance_)
```

2.Scikit-learn主要用法

无监督学习算法-聚类

DBSCAN	`cluster.DBSCAN`
层次聚类	`cluster.AgglomerativeClustering`
谱聚类	`cluster.SpectralClustering`

2.Scikit-learn主要用法

评价指标

sklearn.metrics模块包含了一系列用于评价模型的评分函数、损失函数以及成对数据的距离度量函数.

```
from sklearn.metrics import accuracy_score
accuracy_score(y_true, y_pred)
```

对于测试集而言，y_test即是y_true，大部分函数都必须包含真实值y_true和预测值y_pred.

2.Scikit-learn主要用法

评价指标

回归模型评价

`metrics.mean_absolute_error()` | 平均绝对误差MAE

`metrics.mean_squared_error()` | 均方误差MSE

`metrics.r2_score()` | 决定系数R^2.

2.Scikit-learn主要用法

评价指标

分类模型评价

`metrics.accuracy_score()`	正确率
`metrics.precision_score()`	各类精确率
`metrics.f1_score()`	F1 值
`metrics.log_loss()`	对数损失或交叉熵损失
`metrics.confusion_matrix`	混淆矩阵
`metrics.classification_report`	含多种评价的分类报告

2.Scikit-learn主要用法

评价指标

分类模型评价

metrics.accuracy_score()	正确率.
metrics.precision_score()	各类精确率.
metrics.f1_score()	F1值.
metrics.log_loss()	对数损失或交叉熵损失.
metrics.confusion_matrix	混淆矩阵.
metrics.classification_report	含多种评价的分类报告.

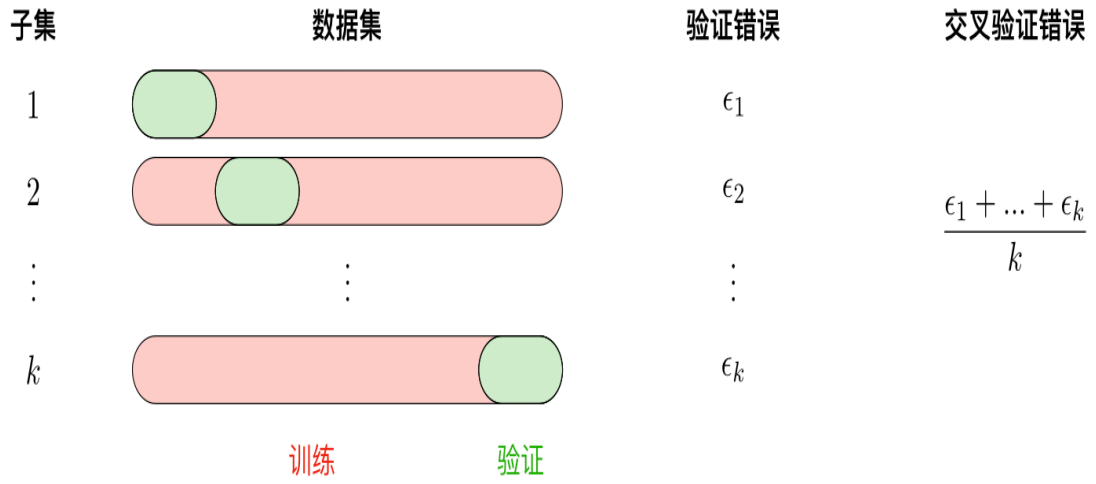
2.Scikit-learn主要用法

交叉验证及超参数调优

```
from sklearn.model_selection import cross_val_score
clf = DecisionTreeClassifier(max_depth=5)
scores = cross_val_score(clf, X_train, y_train,
            cv=5, scoring='f1_weighted')
```

使用5折交叉验证对决策树模型进行评估，使用的评分函数为F1值

sklearn提供了部分带交叉验证功能的模型类如LassoCV、LogisticRegressionCV等，这些类包含cv参数



2.Scikit-learn主要用法

交叉验证及超参数调优

超参数调优——网格搜索

```
from sklearn.model_selection import GridSearchCV
from sklearn import svm
svc = svm.SVC()
params = {'kernel':['linear', 'rbf'], 'C':[1, 10]}  grid_search =
GridSearchCV(svc, params, cv=5)  grid_search.fit(X_train, y_train)
grid_search.best_params_
```

在参数网格上进行穷举搜索，方法简单但是**搜索速度慢**（超参数较多时），且不容易找到参数空间中的局部最优

2.Scikit-learn主要用法

交叉验证及超参数调优

超参数调优: 随机搜索

```
from sklearn.model_selection import RandomizedSearchCV
from scipy.stats import randint
svc = svm.SVC()
param_dist = {'kernel':['linear', 'rbf'], 'C':randint(1, 20)}
random_search = RandomizedSearchCV(svc, param_dist, n_iter=10)
random_search.fit(X_train, y_train)
random_search.best_params_
```

在参数子空间中进行随机搜索，选取空间中的100个点进行建模（可从scipy.stats常见分布如正态分布norm、均匀分布uniform中随机采样得到），时间耗费较少，更容易找到局部最优

参考文献

[1] https://scikit-learn.org/stable/tutorial/basic/tutorial.html ，scikit-learn (sklearn) 官方文档

[2] https://sklearn.apachecn.org/ ， scikit-learn (sklearn) 官方文档中文版