Scikit-learn

数据嗨客

Scikit-learn（0.19.1）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.

符号标记

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

基本建模流程

```
① 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 加载数据
boston = datasets.load_boston()
X = boston.data
y = boston.target
③ 训练集-测试集划分
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3)
④ 数据预处理
scaler = preprocessing.StandardScaler().fit(X_train)
X_train = scaler.transform(X_train)
X_test = scaler.transform(X_test)
⑤ 模型构建与拟合
lr = LinearRegression()
lr.fit(X_train, y_train)
⑥ 模型预测与评价
y_pred = lr.predict(X_test)
r2_score(y_test, y_pred)
```

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
```

训练集-测试集划分

```
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.

数据预处理

标准化

```
from sklearn.preprocessing import StandardScaler
① 构建转换器实例
scaler = StandardScaler()
② 拟合及转换
scaler.fit_transform(X_train)
```

部分数据预处理方法	对应的sklearn的类
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

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（Wrapper），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.

有监督学习算法

回归

```
from sklearn.linear_model import LinearRegression
① 构建模型实例
lr = LinearRegression(normalize=True)
② 训练模型
lr.fit(X_train, y_train)
③ 作出预测
y_pred = lr.predict(X_test)
```

部分回归算法	对应的sklearn的类
LASSO	linear_model.Lasso
Ridge	linear_model.Ridge
ElasticNet	linear_model.ElasticNet
回归树	tree.DecisionTreeRegressor

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)
```

决策树分类算法示例，对于二分类问题，y_prob为每个样本预测为“0”和“1”类的概率.

部分分类算法	对应的sklearn的类
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

集成

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)
```

部分集成算法	对应的sklearn的类
AdaBoost	ensemble.AdaBoostClassifier
	ensemble.AdaBoostRegressor
基于梯度提升	ensemble.GradientBoostingClassifier
	ensemble.GradientBoostingRegressor

无监督学习算法

聚类

sklearn.cluster模块包含了一系列无监督聚类算法.

```
from sklearn.cluster import KMeans
① 构建聚类实例
kmeans = KMeans(n_clusters=3, random_state=0)
② 拟合
kmeans.fit(X_train)
③ 预测
kmeans.predict(X_test)
```

部分聚类算法	对应的sklearn的类
DBSCAN	cluster.DBSCAN
层次聚类	cluster.AgglomerativeClustering
谱聚类	cluster.SpectralClustering

降维

```
from sklearn.decomposition import PCA
pca = PCA(n_components=2)
pca.fit_transform(X_train)
```

通过主成分分析（PCA）将原始数据映射到2维空间.

部分降维方算法	对应的sklearn的类
线性判别分析	discriminant_analysis
	LinearDiscriminantAnalysis
核主成分分析	decomposition.KernelPCA
局部线性映射	manifold.LocallyLinearEmbedding
t-SNE	manifold.TSNE

模型评价

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
metrics.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，基于排版的关系，以下不再重复写这两个参数.

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.

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.

交叉验证及超参数调优

交叉验证

```
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.

超参数调优——网格搜索

```
from sklearn.model_selection import GridSearchCV
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.

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.

Yan Xiaodong
数 据 嗨 客 出 品