

A cluster of colorful geometric shapes, including triangles and squares in shades of blue, yellow, green, and orange, arranged in a complex, overlapping pattern in the top-left corner.

SKLearn

万永权



SKlearn 简介

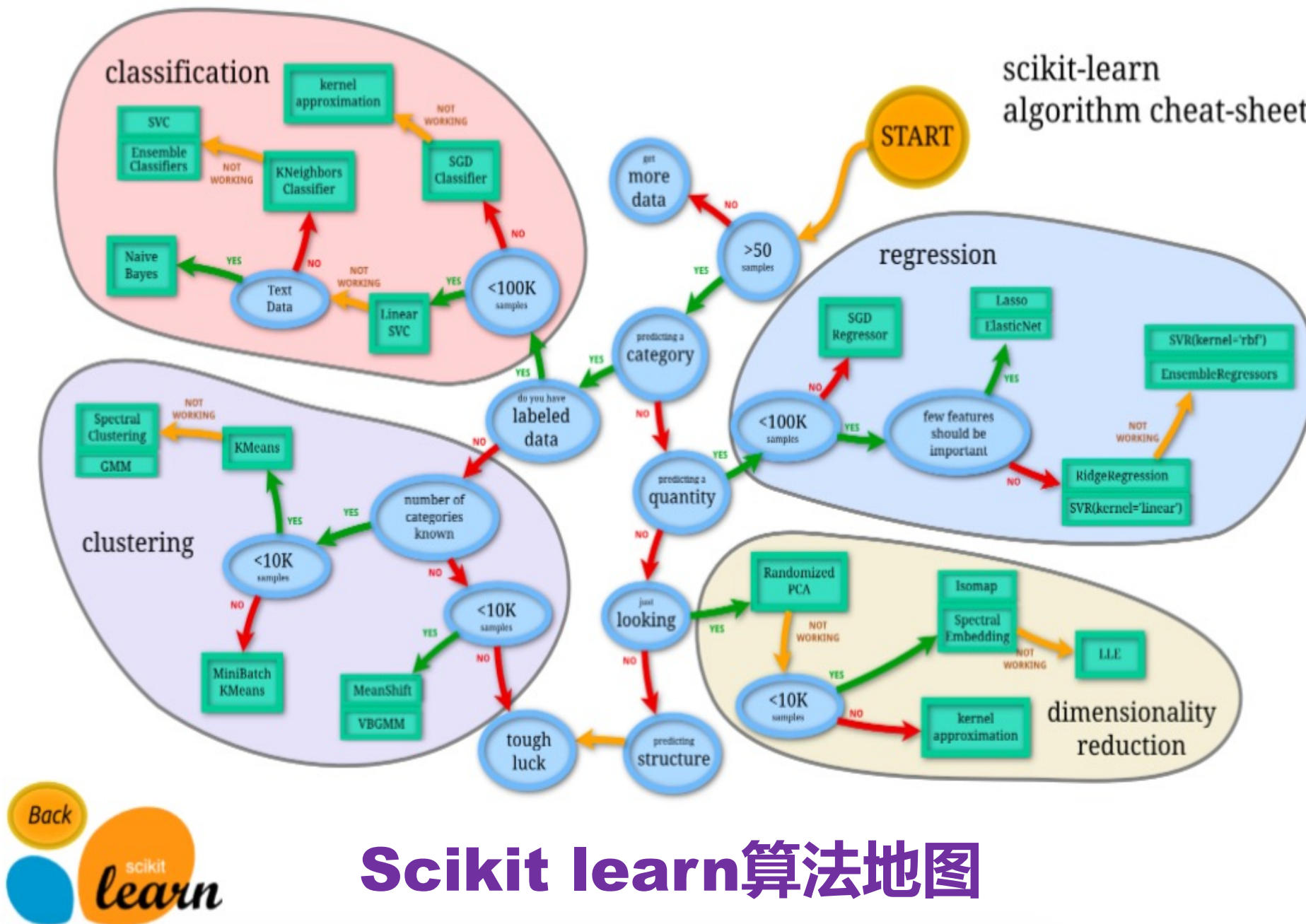
Scikit learn的简称是SKlearn, Python中实现**机器学习的模块**。

建立在 NumPy、SciPy 和 Matplotlib的基础上。SKlearn包含了许多最常见的机器学习算法, 例如分类、回归、聚类、数据降维、数据预处理等。

官方网站: <http://scikit-learn.org>



scikit-learn algorithm cheat-sheet



Scikit learn算法地图



SKlearn的一般步骤

1. 获取数据, 创建数据集
2. 数据预处理
3. 数据集拆分
4. 定义模型
5. 模型评估与选择

4.3.1 SKlearn的一般步骤

1. SKlearn获取数据

SKlearn提供了一个强大的数据库，包含了很多经典数据集。可以通过包含 **SKlearn.datasets** 使用这个数据库。

数据库网址为：<http://scikit-learn.org/stable/modules/classes.html#module-sklearn.datasets>。

数据集	描述
datasets.fetch_california_housing	加载加利福尼亚住房数据集。
datasets.fetch_lfw_people	加载有标签的人脸数据集。
datasets.load_boston	加载波士顿房价数据集。
datasets.load_breast_cancer	加载乳腺癌威斯康星州数据集。
datasets.load_diabetes	加载糖尿病数据集。
datasets.load_iris	加载鸢尾花数据集。
datasets.load_wine	加载葡萄酒数据集。



使用经典的波士顿房价数据集，代码如下：

```
from sklearn.datasets import load_boston  
boston = load_boston()
```

或者

```
from sklearn import datasets  
boston = datasets.load_boston()
```

另一个比较著名的是鸢尾花数据集，调用如下：

```
from sklearn.datasets import load_iris  
data = load_iris()
```

或者

```
from sklearn import datasets  
boston = datasets.load_iris()
```



2. SKlearn数据预处理

SKlearn中的preprocessing模块功能是数据预处理和数据标准化，能完成诸如数据标准化、正则化、二值化、编码以及数据缺失处理等。

函数名称	功能
<code>preprocessing.Binarizer</code>	根据阈值对数据进行二值化
<code>preprocessing.Imputer</code>	插值，用于填补缺失值。
<code>preprocessing.LabelBinarizer</code>	对标签进行二值化
<code>preprocessing.MinMaxScaler</code>	将数据对象中的每个数据缩放到指定范围。
<code>preprocessing.Normalizer</code>	将数据对象中的数据归一化为单位范数。
<code>preprocessing.OneHotEncoder</code>	使用one-Hot方案对整数特征编码。
<code>preprocessing.StandardScaler</code>	通过去除均值并缩放到单位方差来标准化。
<code>preprocessing.normalize</code>	将输入向量缩放到单位范数。
<code>preprocessing.scale</code>	沿某个轴标准化数据集。

【例】使用SKlearn的preprocessing模块对数据进行标准化处理。

【例】使用preprocessing的MinMaxScaler类，将数据缩放到固定区间 [0, 1]。

【例】使用preprocessing的StandardScaler标准化类。

3. SKlearn数据集拆分

可以使用SKlearn提供的`train_test_split`方法，按照比例将数据集分为测试集和训练集，格式：

```
X_train,X_test, y_train, y_test =  
cross_validation.train_test_split(train_data,train_target,test_size=0.4, random_state=0)
```

参数解释：

- ◆ `train_data`：要划分的样本特征数据
- ◆ `train_target`：要划分的样本结果
- ◆ `test_size`：测试集占比，默认值为0.3即预留30%测试样本。如果是整数的话就是测试集的样本数量。
- ◆ `random_state`：是随机数的种子。随机数种子的实质是该组随机数的编号。在需要重复试验的时候，使用同一编号能够得到同样一组随机数。比如随机数种子的值为1、其他参数相同的情况下，每次得到的随机数是相同的。如果每次需要不一样的数据，则`random_state`设置为None。



4.3.2 SKlearn模型选择与算法评价

1. SKlearn定义模型

针对不同的问题，选择合适的模型是非常重要的。如何确定学习模型，既涉及到模型的功能，还需要考虑不同数据量的情况。



2. 使用模型进行训练和预测

模型建立之后，需要使用数据集进行学习，称为训练。SKlearn的模型中大都提供了fit函数可以进行学习训练。



3. SKlearn的模型评估手段

sklearn.metrics模块中提供了一些性能指标。

函数名	功能
metrics.f1_score()	计算调和均值F1指数
metrics.precision_score()	计算精确度
metrics.recall_score()	计算召回率
metrics.roc_auc_score()	根据预测分数计算接收机工作特性曲线下的计算区域 (ROC/AUC)
metrics.precision_recall_fscore_support()	计算每个类的精确度, 召回率, F1指数和支持
metrics.classification_report()	根据测试标签和预测标签, 计算分类的精确度, 召回率, F1指数和支持指标

metrics.mean_absolute_error()	平均绝对误差回归损失
metrics.mean_squared_error()	均方误差回归损失
metrics.r2_score()	R2回归分数函数
model_selection.cross_validate()	通过交叉验证评估指标, 并记录适合度/得分时间
model_selection.cross_val_score()	通过交叉验证评估分数
model_selection.learning_curve()	学习曲线
model_selection.validation_curve()	验证曲线

【例3.71】 查看iris数据集。

```
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
df = pd.read_csv('iris.csv', header=1)
X = df.iloc[:, [0, 2]].values      #
#前50个样本(setosa类别)
plt.scatter(X[:50, 0], X[:50, 1], color='red')
#中间50个样本(versicolor类别)
plt.scatter(X[50:100, 0], X[50:100, 1], color='blue')
#后50个样本的散点图(Virginica 类别)
plt.scatter(X[100:, 0], X[100:, 1], color='green')
plt.xlabel('petal length')
plt.ylabel('sepal length')
#图例位于左上角
plt.legend(loc=2)
plt.show()
```

