怎么用 python 做大作业

张璇

北京交通大学电子信息工程学院网络智能实验室

一、介绍

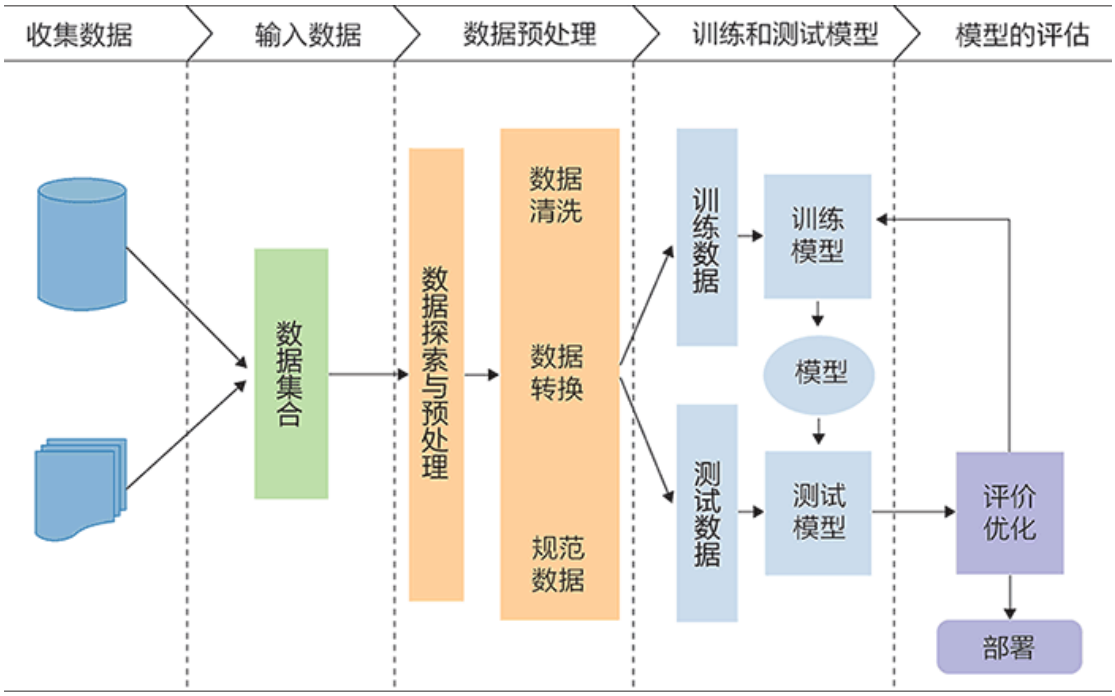
本文假设您有一定的编程经验，比如 Java，C++都可以。本文将提供一些信息，帮助您快速上手 Python 和基于 Python 的机器学习，编出您的第一个 Python 机器学习程序。

为什么要用 Python 呢？使用 Python 最大的好处是有很多现成库，调用这些库中很多现成的函数，可以用几行简单的代码实现丰富的功能。而且相对于 Java 和 C++，Python 非常简单易学，容易上手。

二、步骤

使用 Python 完成一个机器学习项目，要完成哪些**步骤呢？**

请大家首先对机器学习/深度学习的完整流程有一个概念。请大家看这个网页上分享的一张图（https://www.92python.com/view/373.html）。这张图我们也分享在下面。每一个机器学习项目都可以用下图中的流程概括，流程中的每一个步骤也就是我们上课学习的内容。



三、每一个步中，使用的 python 库

那么，在上图所示的每一步中，我们可以用什么 Python 库编程呢？

请大家看这个网页：《【机器学习实战】科学处理鸢尾花数据集》

https://blog.csdn.net/weixin_40431584/article/details/105433366，

该网页是一个简单的例子，用 Python 对一个鸢尾花数据集完成上述机器学习的完整流程。

在这个例子中，数据处理用了 pandas 库，数据可视化用了 matplotlib、seaborn 库，训练决策树模型用了 sklearn 库，评估也用了 sklearn 库。

你会发现，其实每一步骤只需要几行代码就能够实现的。

下面是常用的 python 库：

1. numpy、pandas——数据处理
2. sklearn、tensorflow、pytorch——机器学习/深度学习
3. pyspark ——大数据

四、在库中找到需要的函数

要实现每一步的具体功能，如何在库中找到需要的函数呢？

关于函数最完整的介绍，功能和参数，尽量看官网的官方文档，因为机器学习领域更新特别快，这些库的更新也特别快。官网上的文档是最新的。很多库的文档都有我国的志愿者翻译的中文版。请一定找到它们。

然后可以百度/CSDN/github/… 看看大家是怎么用的。

举个例子：心电图类型识别案例，要使用决策树模型进行分类。那么，调用 sklearn 库的什么决策树模型函数呢？请按下面的步骤

1、打开 sklearn 官方文档：

中文（http://www.scikitlearn.com.cn/）

英文（https://scikit-learn.org/stable/modules/classes.html）

2、找到决策树 DecisionTreeClassifier 函数的说明

中文（http://www.scikitlearn.com.cn/0.21.3/11/）

英文（https://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeClassifier.html#sklearn.tree.DecisionTreeClassifier）

官方文档中有最详细的介绍：

特别是它会举例子告诉你这个函数怎么用，如下图中的代码：

Examples

```
>>> from sklearn.datasets import load_iris
>>> from sklearn.model_selection import cross_val_score
>>> from sklearn.tree import DecisionTreeClassifier
>>> clf = DecisionTreeClassifier(random_state=0)
>>> iris = load_iris()
>>> cross_val_score(clf, iris.data, iris.target, cv=10)
...                             # doctest: +SKIP
...
array([ 1.     ,  0.93...,  0.86...,  0.93...,  0.93...,
        0.93...,  0.93...,  1.     ,  0.93...,  1.      ])
```

上图中的代码中给出了使用这个函数的一般方法。

这个函数的参数怎么设置呢？文档里也有详细的说明，如下图所示：

Parameters:

criterion : *{"gini", "entropy"}, default="gini"*
The function to measure the quality of a split. Supported criteria are "gini" for the Gini impurity and "entropy" for the information gain.

splitter : *{"best", "random"}, default="best"*
The strategy used to choose the split at each node. Supported strategies are "best" to choose the best split and "random" to choose the best random split.

max_depth : *int, default=None*
The maximum depth of the tree. If None, then nodes are expanded until all leaves are pure or until all leaves contain less than min_samples_split samples.

min_samples_split : *int or float, default=2*
The minimum number of samples required to split an internal node:

- If int, then consider `min_samples_split` as the minimum number.
- If float, then `min_samples_split` is a fraction and `ceil(min_samples_split * n_samples)` are the minimum number of samples for each split.

上图中给出了该函数各个参数的详细说明。其中，有些参数上课讲算法时提到过，比如 gini impurity（基尼不纯度）、entropy（熵）、information gain（信息增益）。

设置参数最简单的方法就是：什么参数都不改，就用默认值就好。大多数参数都有默认值（default 项）。请放心，默认值一般来说是经过精心挑选的，在大多数情况下都有很好的性能。

都有些什么函数可以调用呢？文档里有详细的说明，如下图所示：

Methods

`apply`(X[, check_input])	Return the index of the leaf that each sample is predicted as.
`cost_complexity_pruning_path`(X, y[, ...])	Compute the pruning path during Minimal Cost-Complexity Pruning.
`decision_path`(X[, check_input])	Return the decision path in the tree.
`fit`(X, y[, sample_weight, check_input, ...])	Build a decision tree classifier from the training set (X, y).
`get_depth`()	Return the depth of the decision tree.
`get_n_leaves`()	Return the number of leaves of the decision tree.
`get_params`([deep])	Get parameters for this estimator.
`predict`(X[, check_input])	Predict class or regression value for X.
`predict_log_proba`(X)	Predict class log-probabilities of the input samples X.
`predict_proba`(X[, check_input])	Predict class probabilities of the input samples X.
`score`(X, y[, sample_weight])	Return the mean accuracy on the given test data and labels.
`set_params`(**params)	Set the parameters of this estimator.

上图就给出了决策树模型最全的函数方法和它们的功能。

也请大家善用搜索引擎。百度输入关键词（python、sklearn、决策树、...），看看别人对于各种不同的数据集是怎么用的

五、机器学习模型的训练

训练机器学习模型的代码其实很简单。比如上面例子中对鸢尾花数据集使用的决策树就很简单，就几行代码

1. train_test_split 函数划分测试训练集
2. fit 函数训练模型
3. score 函数输出结果准确度。

如下图所示：

3.2 划分训练集和测试集

```
(training_inputs, test_inputs, training_classes,
 test_classes) = train_test_split(all_inputs, all_classes, train_size=0.75, random_state=1)
```

用 75% 和 25% 的比例来划分训练集和测试集，设一个 random_state 是为了在机器学习中每次出的结果一样，便于找问题。

3.3 用模型来学习

```
# 创建决策树分类器
decision_tree_classifier = DecisionTreeClassifier()

# 训练数据
decision_tree_classifier.fit(training_inputs, training_classes)

# 分类精度
decision_tree_classifier.score(test_inputs, test_classes)
```

六、Python 机器学习环境的安装和使用

使用 python，推荐安装 anaconda（一个集成库，里面放了很多很多的常用库），也可以就在华为云平台上跑代码就行~