scikit-learn 实验报告

111172 徐鸿飞

1. 摘要

一般来说，一个学习问题通常会考虑一系列 n 个 样本 数据，然后尝试预测未知数据的属性。 如果每个样本是 多个属性的数据（比如说是一个多维记录），就说它有许多“属性”，或称 features(特征) 。

我们可以将学习问题分为几大类:

监督学习 ，其中数据带有一个附加属性，即我们想要预测的结果值（ 点击此处 转到 scikit-learn 监督学习页面）。这个问题可以是:

分类 ：样本属于两个或更多个类，我们想从已经标记的数据中学习如何预测未标记数据的类别。 分类问题的一个示例是手写数字识别，其目的是将每个输入向量分配给有限数目的离散类别之一。我们通常把分类视作监督学习的一个离散形式（区别于连续形式），从有限的类别中，给每个样本贴上正确的标签。

回归 ：如果期望的输出由一个或多个连续变量组成，则该任务称为 回归 。 回归问题的一个示例是预测鲑鱼的长度是其年龄和体重的函数。

无监督学习，其中训练数据由没有任何相应目标值的一组输入向量 x 组成。这种问题的目标可能是在数据中发现彼此类似的示例所聚成的组，这种问题称为 聚类 ，或者，确定输入空间内的数据分布，

称为 密度估计 ，又或从高维数据投影数据空间缩小到二维或三维以进行 可视化

Scikit-learn（以前称为 scikits.learn，也称为 sklearn）是针对 Python 编程语言的免费软件 机器学习库。它具有各种分类，回归和聚类算法，包括支持向量机，随机森林，梯度提升，k 均值和 DBSCAN，并且旨在使用 Python 进行数值科学计算。

Sklearn 包含了很多种机器学习的方式:

- Classification 分类
- Regression 回归
- Clustering 非监督分类
- Dimensionality reduction 数据降维
- Model Selection 模型选择
- Preprocessing 数据预处理

本次实验主要是使用 Scikit-learn 库进行一些简单的实验。我主要使用了这个库研究了一下日期和合作/冲突的联系。

2. 目的

学习使用 Scikit-learn 库，学习使用 sklearn 库进行机器学习的一个流程。

3. 软硬件环境

硬件	I7-7500u 8g 笔记本
软件	Python 3.7.6
	Scikit-learn 0.23.1

	Numpy 1.18.5
	Matplotlib 3.2.1
IDE	PyCharm

4. 数据量

Google GDELT 数据集。主要使用到的字段有 C MonthYear：记录事件发生的年月，格式为 YYYYMM；30 QuadClass：这个字段指定事件类型主要分类，所有事件将被划分为以下四个分类之一：1=口头合作，2=物质合作，3=口头冲突，4=物质冲突。

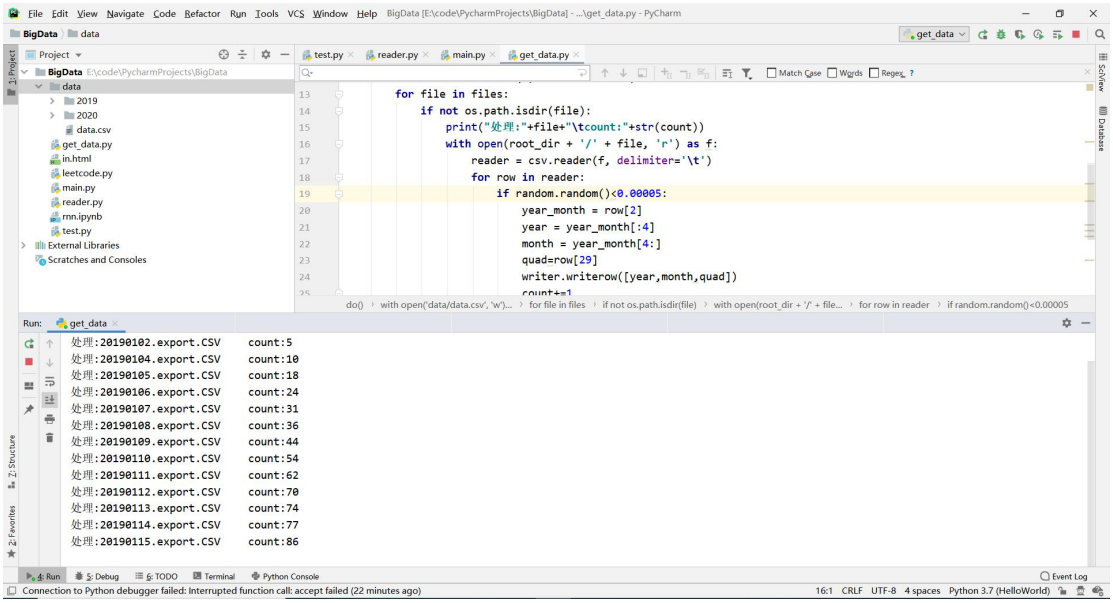
5. 方法/算法

一般的 sklearn 使用的过程为：

创建数据->建立模型—>训练—>预测

数据的提取

从 2019 年这个文件夹中按概率提取出所有的年月与 QuadClass：



划分出属性和标签：

```
def get_x_y(data):
    for i in range(len(data)):
        data[i] = list(map(int, data[i]))
    data = np.array(data)
    data_x = data[:, :2]
    data_y = data[:, 2]
    return data_x, data_y
```

在数据分析过程中，为了保证模型在实际系统中能够起到预期作用，一般需要将样本分成独立的三部分：

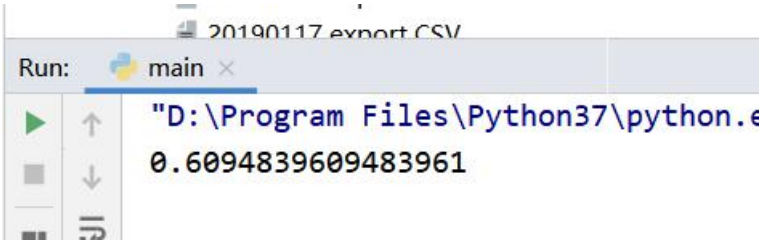
- 训练集（train set）：用于估计模型。
- 验证集（validation set)：用于确定网络结构或者控制模型复杂程度的参数。
- 测试集（test set）：用于检验最优的模型的性能。
- 典型的划分方式是训练集占总样本的 50%，而验证集和测试集各占 25%。

使用 train_test_split 函数可以简单的划分训练集和测试集，而且会帮助打乱。

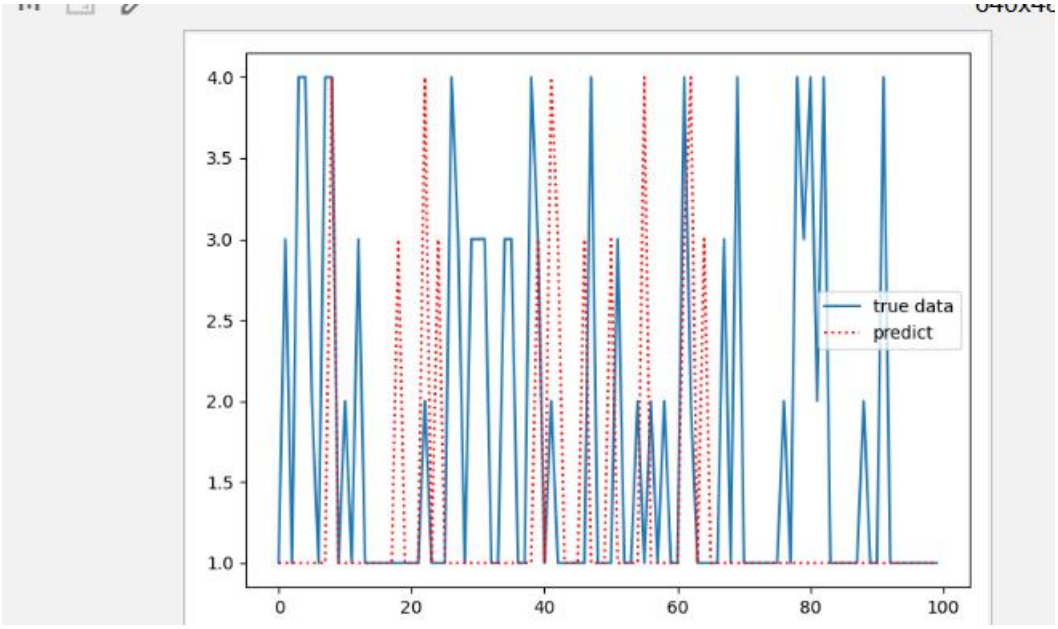
使用 knn 分类进行训练和预测

```
def do():
    data = read_file('./data/new.csv')
    data_x, data_y = get_x_y(data)
    x_train, x_test, y_train, y_test = train_test_split(data_x, data_y, test_size=0.3)
    knn = KNeighborsClassifier()
    knn.fit(x_train, y_train)
    print(knn.score(x_test, y_test))
    y_predict = knn.predict(x_test)
    plt.plot(y_test[100:200], label='true data')
    plt.plot(y_predict[100:200], 'r:', label='predict')
    plt.legend()
    plt.show()
```

6. 结果和分析



使用 knn 进行这个预测的准确度不高，主要是确实时间和和平/冲突的关系不是很大；但是可见也有一定的联系。



只能说每个月每种事件发生的可能性是不一样的，而且分布不是很均匀。

7. 结论

Sklearn 库只需要简单的几行代码就可以帮助我们实现大部分的机器学习的功能。简单高效的数据挖掘和数据分析工具，可在各种环境中重复使用。