scikit-learn 实战之非监督学习

一、实验介绍

1.1 实验内容

非监督学习（英语：Unsupervised learning）是机器学习中十分重要的一个分支。这是本实验课程的第 2 章节，将带你了解如何对聚类效果进行评估。

1.2 实验知识点

- 肘部法则
- 轮廓系数

1.3 实验环境

- python2.7
- Xfce 终端
- ipython 终端

1.4 适合人群

本课程难度为一般，属于初级级别课程，适合具有 Python 基础和线性代数基础，并对机器学习中分类问题感兴趣的用户。

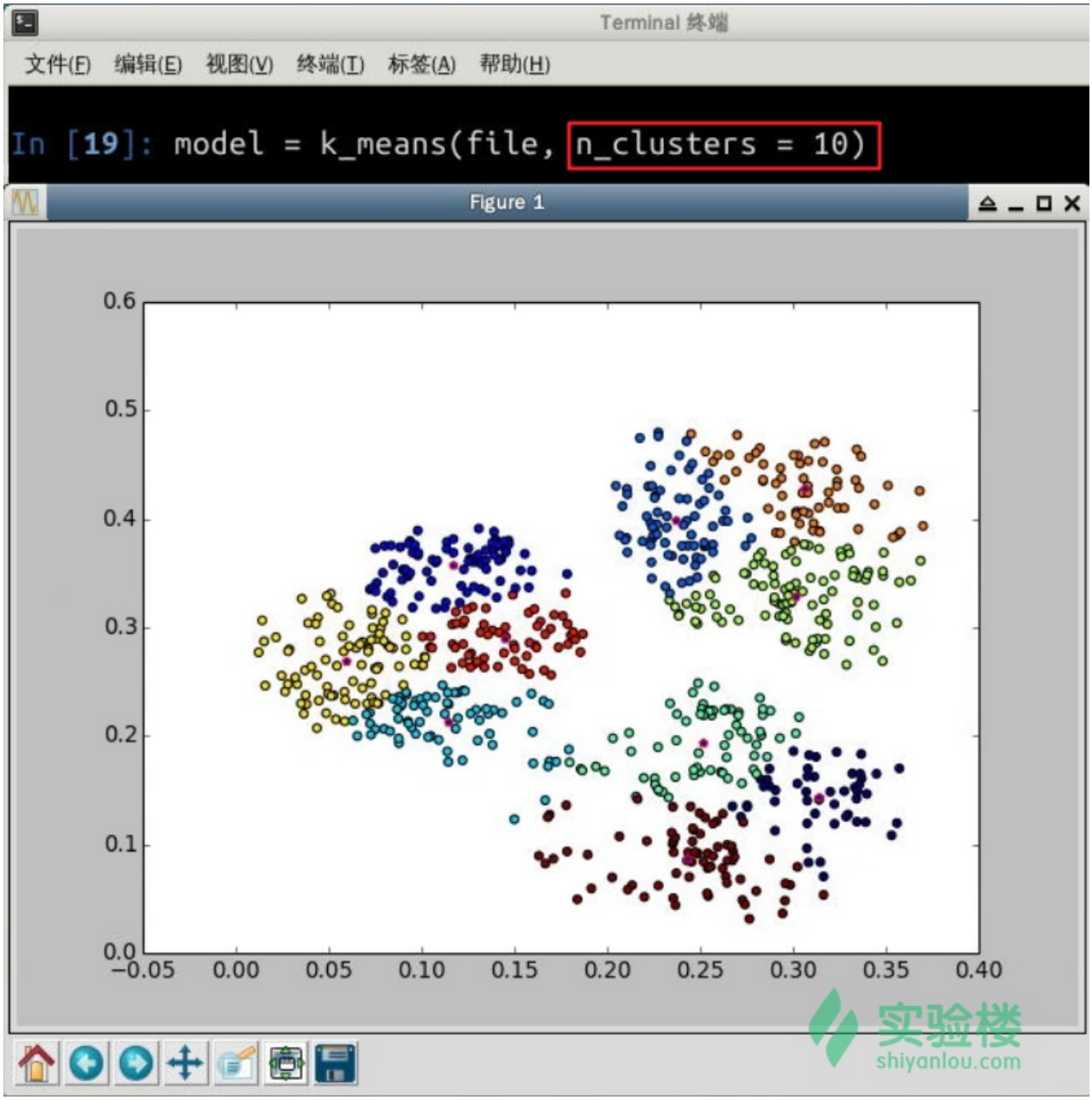
1.5 代码获取

你可以通过下面命令将代码下载到实验楼环境中，作为参照对比进行学习。

```
# 肘部法则
$ wget http://labfile.oss.aliyuncs.com/courses/880/cluster_evaluate_inertia.py
# 轮廓系数
$ wget http://labfile.oss.aliyuncs.com/courses/880/cluster_evaluate_silhouette.py
```

二、K 值怎么选?

在第 1 小节的例子中，我们已经将样本数据聚合成 3 类。但是，你又没有发现一个问题？那就是我们为什么要聚成 3 类？为什么不可以将数据聚成像下图呈现的 10 类?

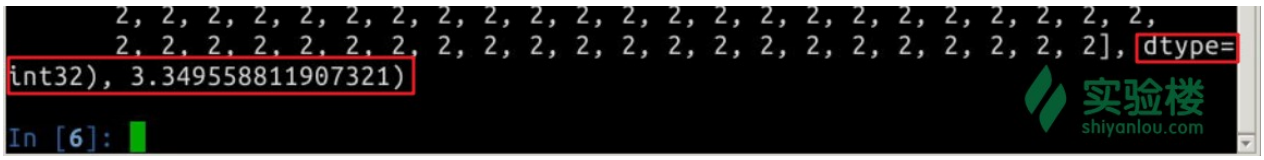


难道是因为，这里的样本数据在空间分布上，看上去有点像 3 种类别吗？那么，当我们遇到肉眼看起来不太好确定类别，或者是高维数据时怎么办呢?

所以，这一系列问题最终可以归结为一个问题，那就是：**当我们在使用 K-Means 聚类时，我们该如何确定 K 值?**

二、肘部法则

这里，介绍一种启发式学习算法，被称之为**肘部法则**。在上文谈到用 `print model` 语句时，它会输出三个数组。其中，前两个数组在进行聚类绘图时已经用到了，但是我们一直没有使用第三个数组（红框标识）。



第三个数组，准确说来只是一个数值。它代表着样本距离最近中心点距离的总和。你可以在大脑里想一下，当我们的 K 值增加时，也就是类别增加时，这个数值应该是会降低的。直到聚类类别的数量和样本的总数相同时，也就是说一个样本就代表一个类别时，这个数值会变成 0。

接下来，我们绘制出这个数值与聚类数量之间的关系曲线。从在线环境左下角**应用程序菜单 > 附件**，打开 ipython 交互式终端。

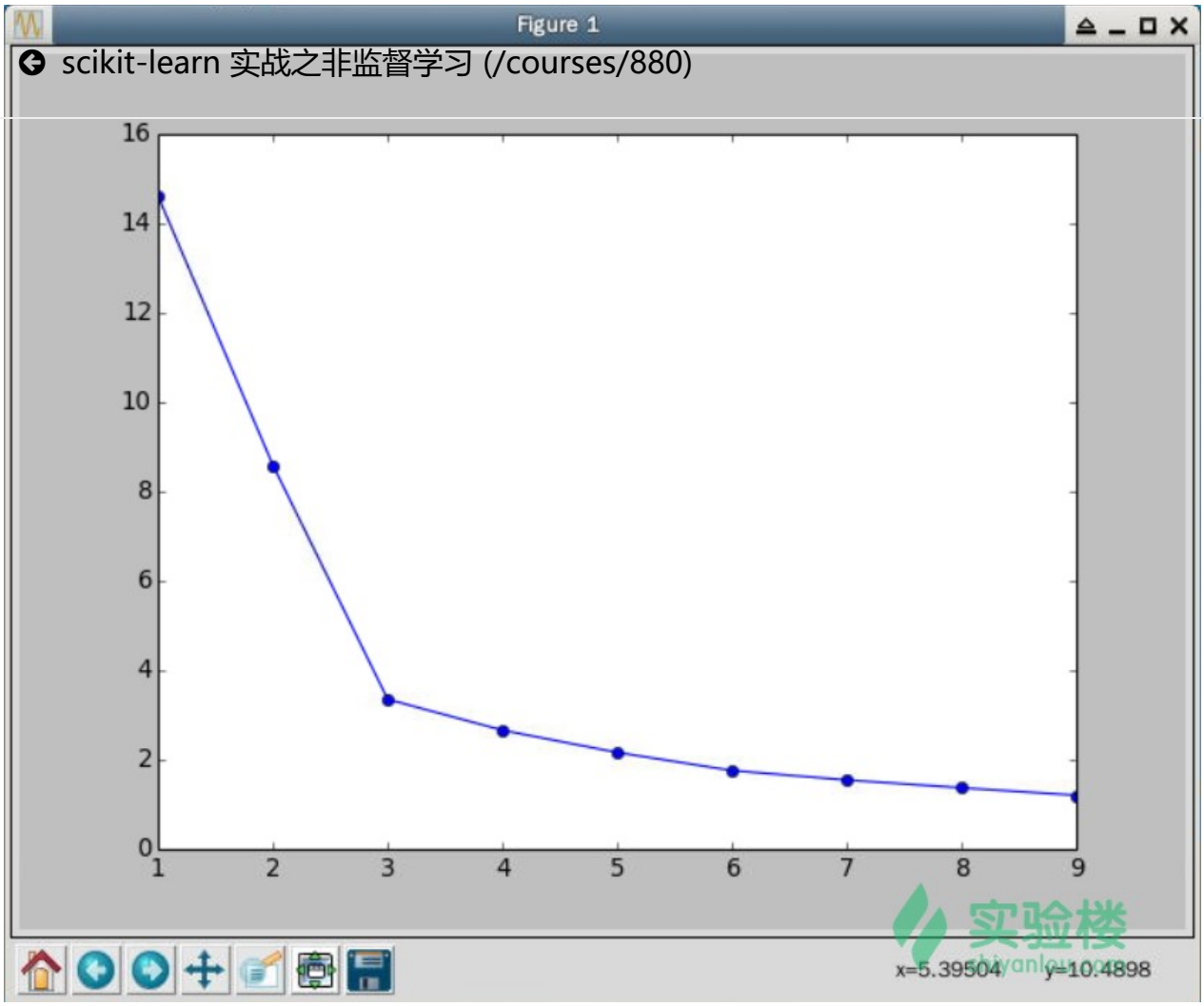
```
from matplotlib import pyplot as plt # 加载绘图模块
from sklearn.cluster import k_means # 加载聚类模块
import pandas as pd # 加载数据处理模块

# 读取数据
file = pd.read_csv("cluster_data.csv", header=0)

index = [] # 横坐标数组
inertia = [] # 纵坐标数组

# K 从 1~ 10 聚类
for i in range(9):
    model = k_means(file, n_clusters=i + 1)
    index.append(i + 1)
    inertia.append(model[2])

# 绘制折线图
plt.plot(index, inertia, "-o")
plt.show()
```



我们可以看到，和预想的一样，样本距离最近中心点距离的总和会随着 K 值的增大而降低。

其中，畸变程度最大的点被称之为「肘部」。我们可以看到，这里的「肘部」明显是 K = 3。这也说明，将样本聚为 3 类的确是最佳选择。

三、轮廓系数

当我们完成一项聚类任务之后，我们需要对聚类效果进行评估。其实，上面提到的肘部法则也算是一种评估方法，它让我们知道当 K 值为多少时，整体聚类的结果更理想。

除了，肘部法则。聚类中还有一种评估方法，叫**轮廓系数**。轮廓系数综合了聚类后的两项因素：内聚度和分离度。内聚度就指一个样本在簇内的不相似度，而分离度就指一个样本在簇间的不相似度。

在 scikit-learn 种，同样也提供了直接计算轮廓系数的方法。下面，绘制出轮廓系数与聚类 K 值变化的折线图。

scikit-learn 实战之非监督学习 (/courses/880)

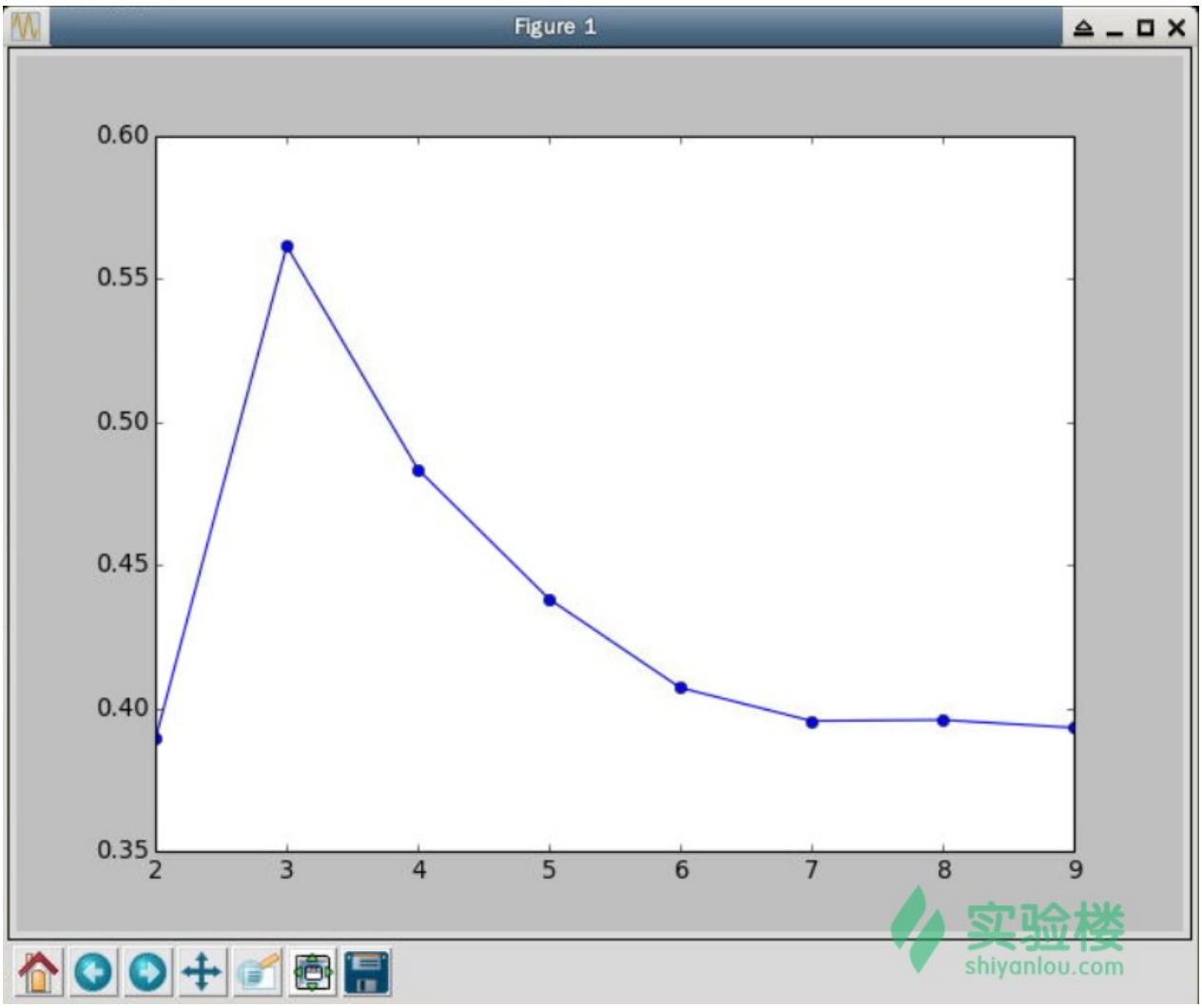
```
from sklearn.metrics import silhouette_score # 导入轮廓系数计算模块

index2 = [] # 横坐标
silhouette = [] # 轮廓系数列表

# K 从 2 ~ 10 聚类
for i in range(8):
    model = k_means(file, n_clusters=i + 2)
    index2.append(i + 2)
    silhouette.append(silhouette_score(file, model[1]))

print silhouette # 输出不同聚类下的轮廓系数

# 绘制折线图
plt.plot(index2, silhouette, "-o")
plt.show()
```



轮廓系数越接近于 1，代表聚类的效果越好。我们可以很清楚的看出，K=3 对应的轮廓系数数组最大，也更接近于 1 。

五、实验总结

本次试验中，我们通过肘部法则和轮廓系数验证了如何选择 K 值。只是 K-Means 聚类实施过程中的第一步。除了 K-Means 之外，还有很多算法都需要提前确定 K 值，会在下一个章节中出现。

上一节：K-Means 聚类算法 (/courses/880/labs/3188/document)

下一节：聚类算法对比与选择 (/courses/880/labs/3190/document)