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。这是实验课程的第一章节，将带你了解什么是非监督学习？并学会用 K-Means 算法完成一个聚类实验。

1.2 实验知识点

- 非监督学习概念
- K-Means 聚类

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，并对机器学习中聚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。

```
$ wget http://labfile.oss.aliyuncs.com/courses/880/k_means_cluster.py
```

二、什么是非监督学习？

什么是非监督学习？笼统来讲，它和监督学习是一个相对的概念。在监督学习的过程中，我们需要对训练数据打上标签，这是必不可少的一步。而非监督学习中，就不再不需要提前对数据进行人工标记。

举个例子，比如我们现在有一堆动物的照片。在监督学习中，我们需要提前对每张照片代表的动物进行标记。这一张是狗，那一张是猫，然后再进行训练。最后，模型对于新输入的照片，就能分清楚动物的类别。

当进行非监督学习时，就不需要提前对照片进行标记了。我们只需要将所有的训练样本照片「喂」给算法即可。注意，这个时候和监督学习有一些不同，非监督学习只能识别出训练样本里包含了几种类别的动物，而并不能直接告诉你这只是猫，那一只是狗。但是，这里的类别数量一般都不会太大，你可以手动对类别进行标记，再将数据用于其他用途。

上面这个例子中，非监督学习识别出样本包含几种类别，就是我们通常所说的「聚类」。

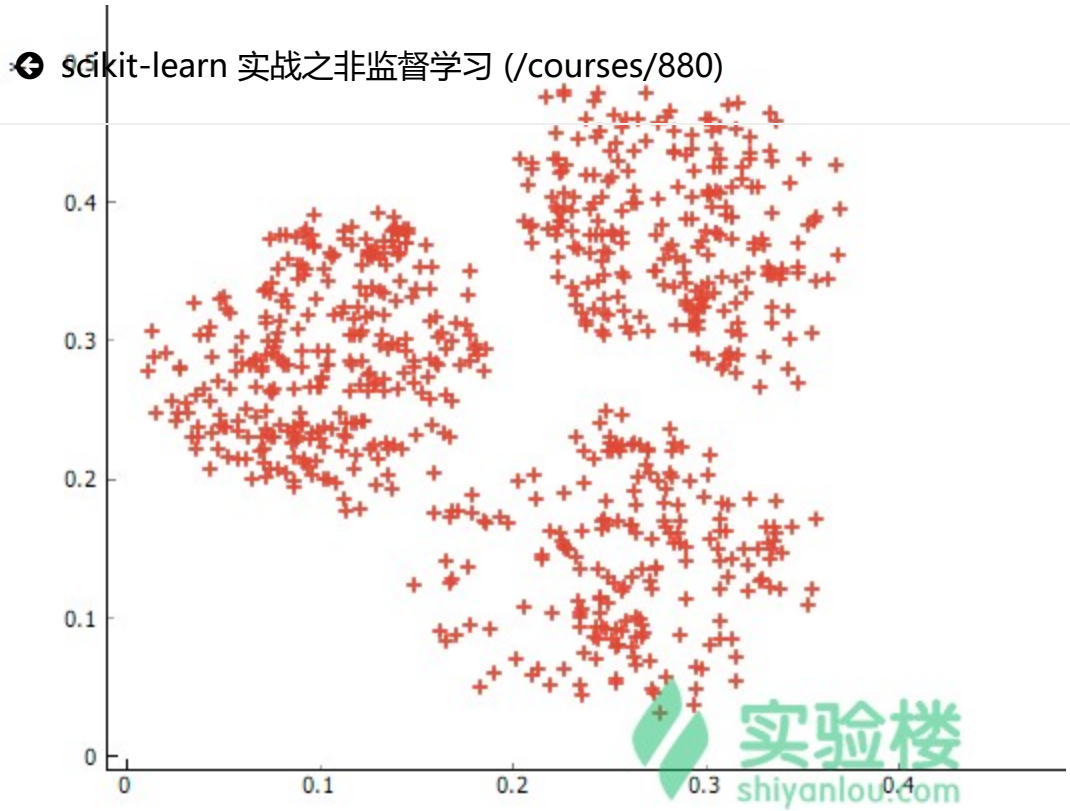
三、K-Means 聚类

监督学习被用于解决分类和回归问题，而非监督学习主要是用于解决聚类问题。聚类，顾名思义就是将具有相似属性或特征的数据聚合在一起。聚类算法有很多，最简单和最常用的就算是 K-Means 算法了。

K-Means，中文译作 K-均值算法。从它的名字来讲，K 代表最终将样本数据聚合为 K 个类别。而「均值」代表在聚类的过程中，我们计算聚类中心点的特征向量时，需要采用求相邻样本点特征向量均值的方式进行。例如，我们将 X1=(x1, y1), X2=(x2, y2), X3=(x3, y3) 聚为一类时，中心点坐标 O(o1, o1) 为：o1 = (x1+x2+x3)/3, o2=(y1+y2+y3)/3。

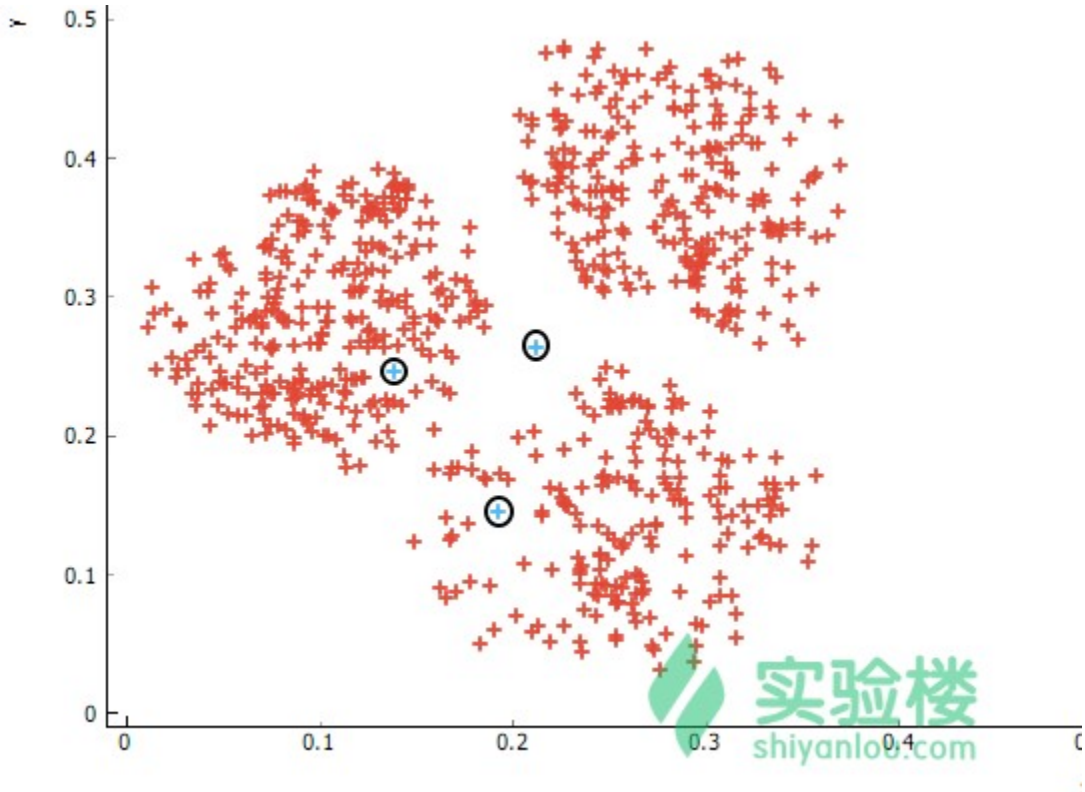
四、聚类过程

K-Means 算法在应用时，相对来上面的例子要复杂一些。现在，假设有如下图所示的一组二维数据。接下来，我们就一步一步演示 K-Means 的聚类过程。

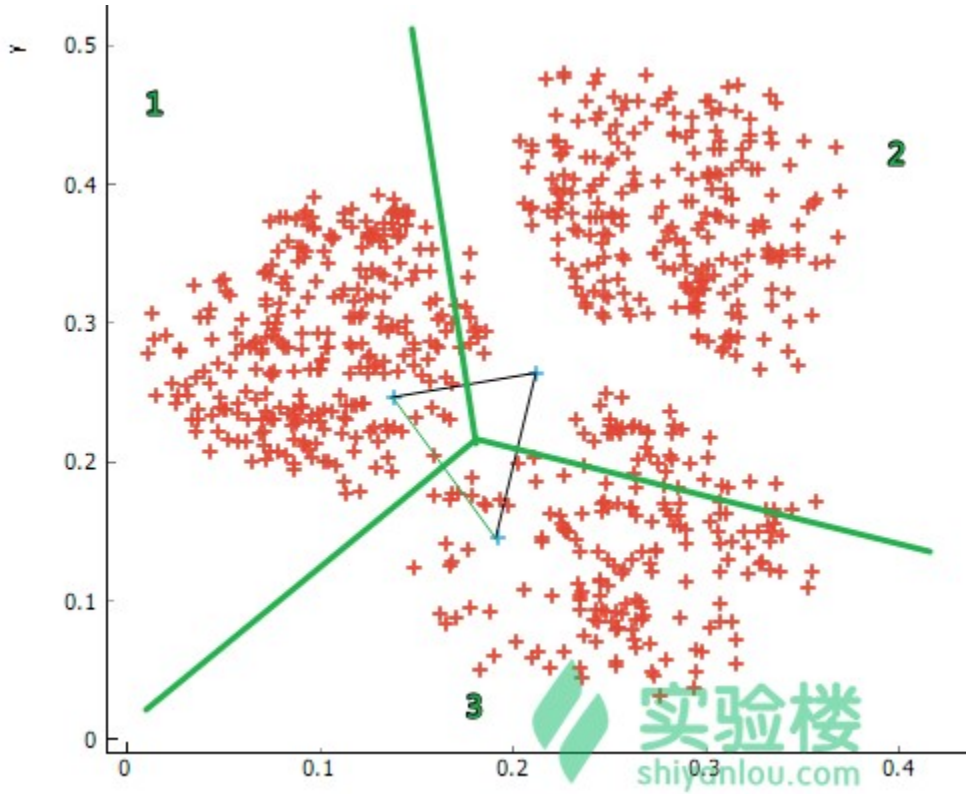


第一步，确定要聚为几类？也就是 K 值。假设，这里我们想将样本聚为 3 类。当然，你也完全可以将其聚为 2 类或 4 类，不要受到视觉上的误导。

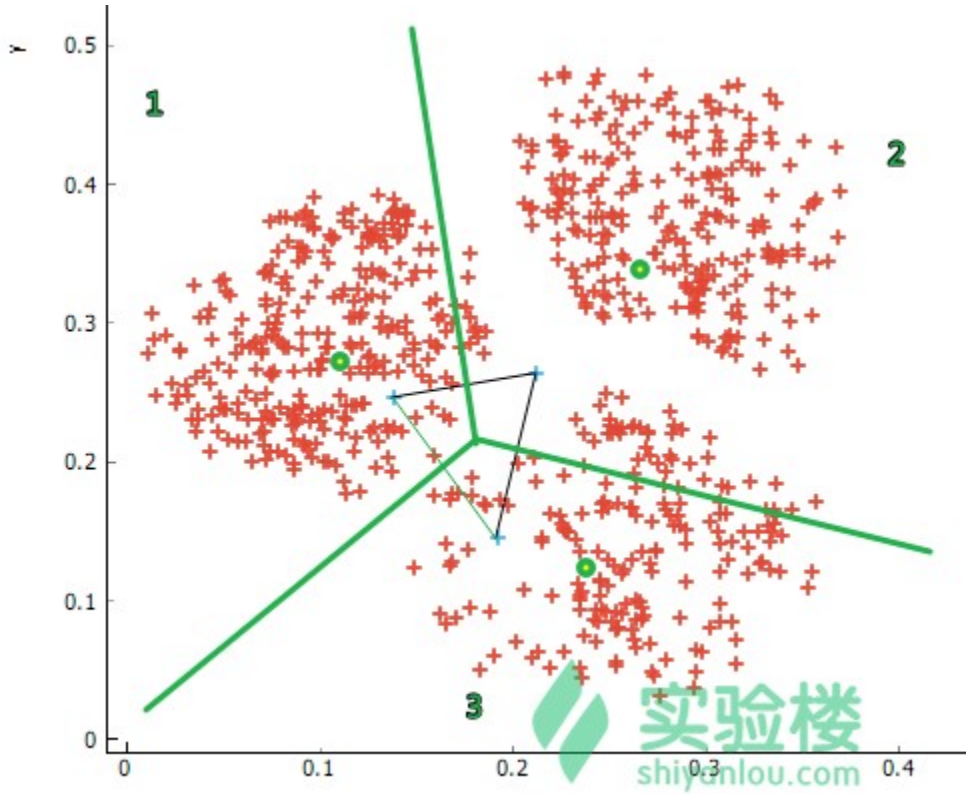
这里，我们以 3 类为例。当确定聚为 3 类之后，我们在特征空间上，**随机**初始化三个类别中心。



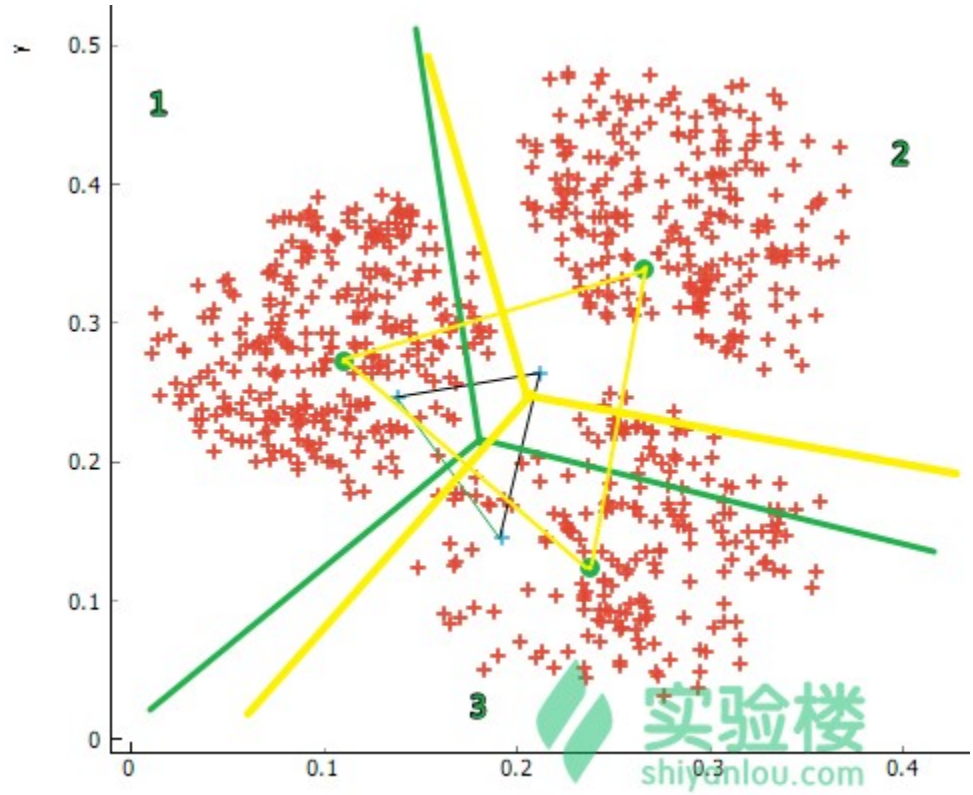
第二步，我们依据这三个随机初始化的中心，将现有样本按照与最近中心点之间的距离进行归类。图中绿线将全部样本划分为三个类别。（中间小三角形是参考线，可以忽略。）



这样，我们的样本被划为为三个区域。现在，我们就要用到上面提到的均值来重新求解 3 个区域对应的新的样本中心。



如上图所示，假设我们求解的新样本中心为三个绿点所示的位置。然后，又重新回到上一步，根据这三个中心重新划分样本，再求解中心。



依次迭代，直到样本中心变化非常小时终止。最终，就可以将全部样本聚类为三类。

五、算法实验

接下来，我们以 scikit-learn 提供的 K-Means 算法为例进行实验。实验样本就采用上面的进行算法过程演示的样本数据。

首先，我们打开 Xfce 终端，通过下面的命令获取实验所需的 csv 数据文件。

```
# 获取实验数据集
$ wget http://labfile.oss.aliyuncs.com/courses/880/cluster_data.csv
```

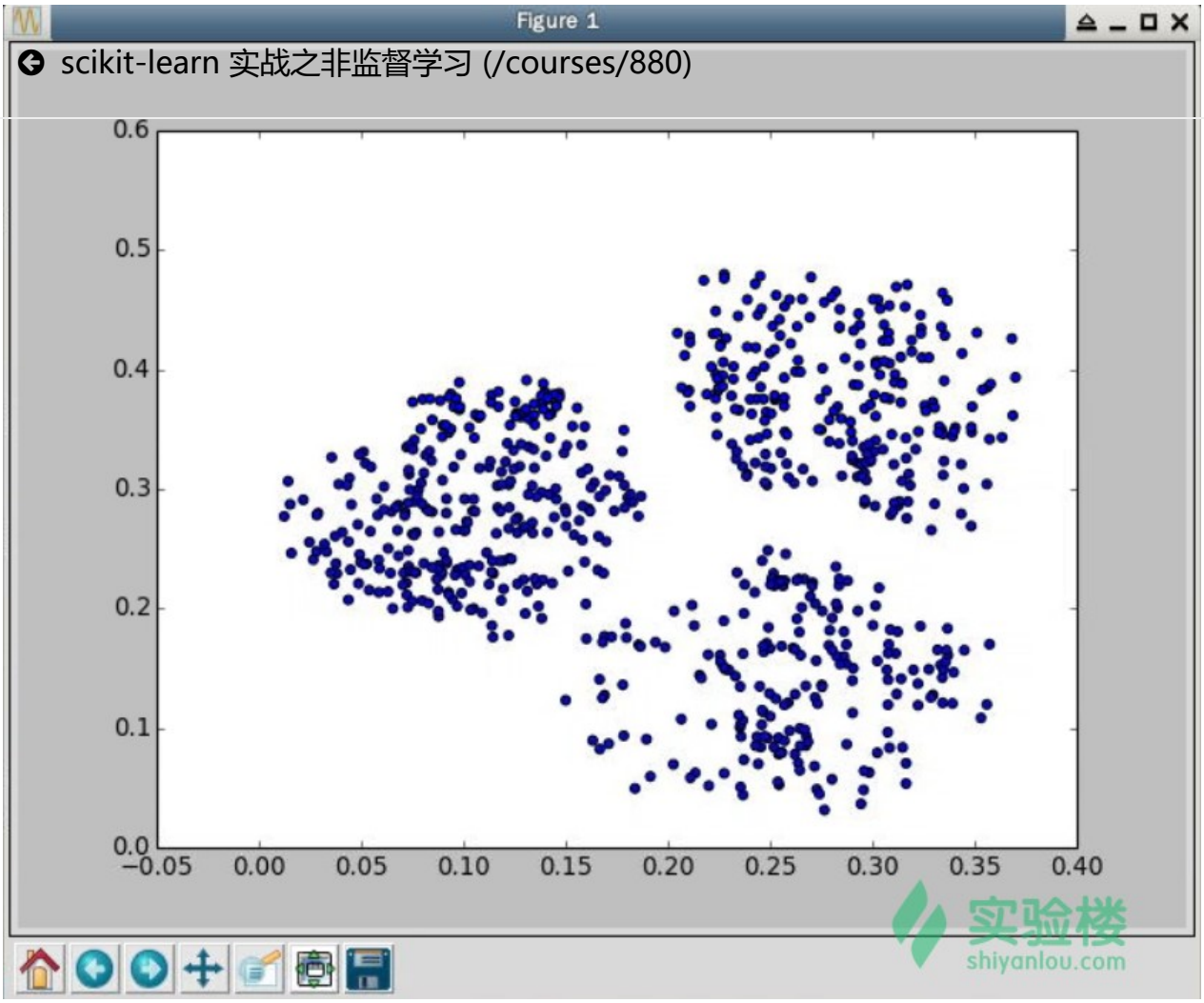
数据下载完成后，我们通过实验环境左下角的**应用程序菜单 > 附件**，打开 `ipython` 终端开始编写 python 代码。

首先，我们导入 Pandas 数据处理模块，用来解析 `csv` 数据文件，并查看文件的组成结构。

```
import pandas as pd # 导入数据处理模块

file = pd.read_csv("cluster_data.csv", header=0) # 导入数据文件
print file # 输出文件
```

可以看到，文件包含两列，也就是对应在特征空间的 x, y 坐标。接下来，我们用 Matplotlib 将数据绘制成散点图。

```
X = file['x'] # 定义横坐标数据
y = file['y'] # 定义纵坐标数据

from matplotlib import pyplot as plt # 导入绘图模块

plt.scatter(X, y) # 绘制散点图
plt.show() # 显示图
```



接下来，开始导入 K-Means 方法进行聚类。scikit-learn 中的聚类算法都包含在 `sklearn.cluster` 方法下。

```
from sklearn.cluster import k_means # 导入 K-Means 方法

model = k_means(file, n_clusters = 3) # 建立聚类模型
```

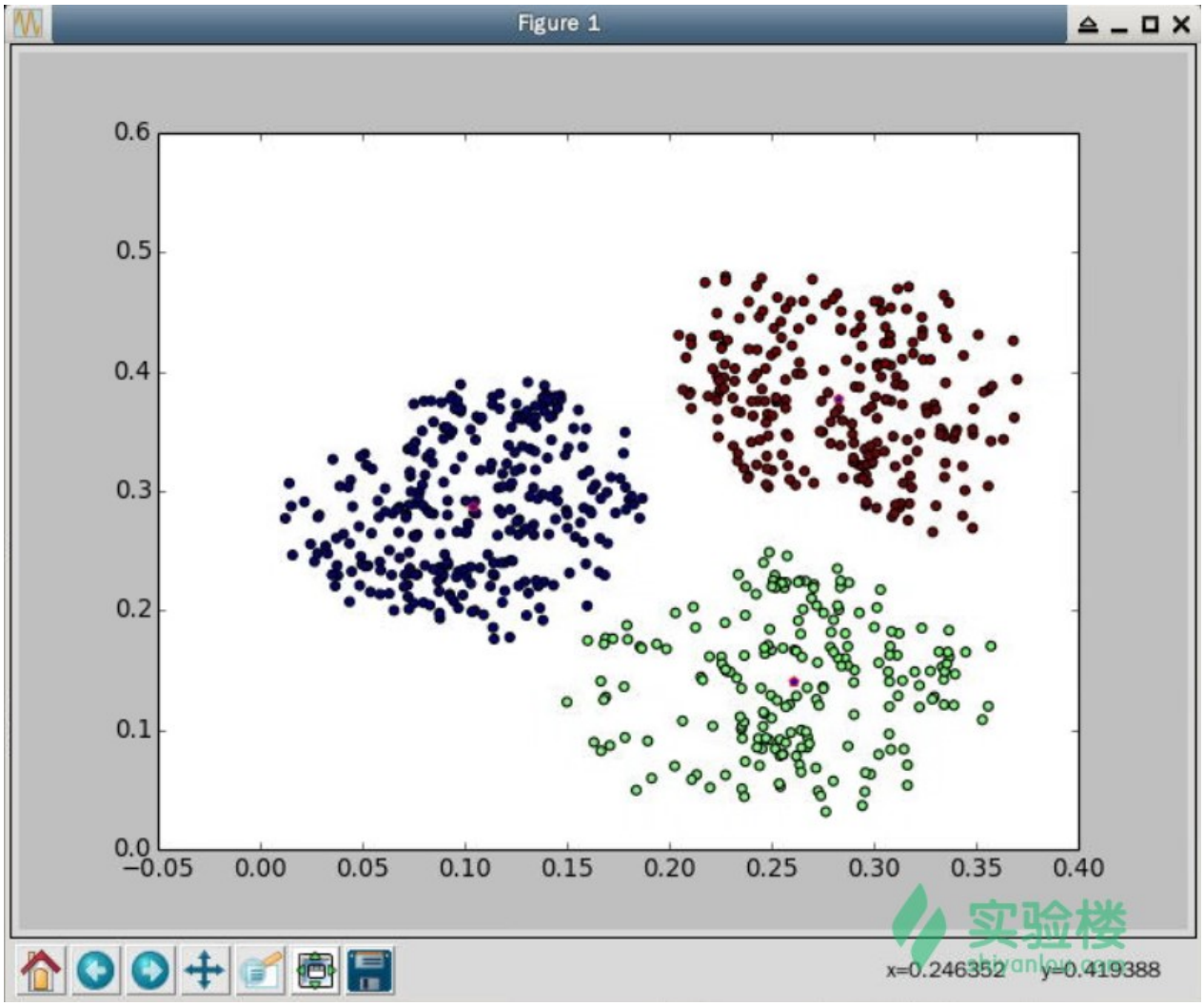
执行完这一步之后，我们可以用 `print model` 看看聚类结果。

```
In [12]: model = k_means(file, n_clusters = 3)

In [13]: print model
(array([[ 0.1040997 ,  0.28663691],
       [ 0.26098231,  0.14011151],
       [ 0.28288786,  0.37671499]]), array([0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
```

我们可以看到，model 输出的结果包含三个数组（截图不全）。其中，第一个数组表示**三个聚类中心点坐标**。第二个数组表示**样本聚类后类别**，第三个数组表示样本**距最近聚类中心的距离总和**。

接下来，我们就将聚类的结果绘制出来

```
cluster_centers = model[0] # 聚类中心数组
cluster_labels = model[1] # 聚类标签数组

plt.scatter(X, y, c=cluster_labels) # 绘制样本并按聚类标签标注颜色

# 绘制聚类中心点，标记成五角星样式，以及红色边框
for center in cluster_centers:
    plt.scatter(center[0], center[1], marker="p", edgecolors="red")

plt.show() # 显示图
```



可以看到，聚类的结果已经显示出来了，聚类中心也做了相应标记。效果还是非常不错的。

六、实验总结

非监督学习是机器学习中十分重要的分支之一。实际生活中，我们会遇到大量的非监督学习问题。因为对样本数据进行人工标记是一件非常繁重的工作。许多时候，我们都会先使用非监督学习对大量的样本进行聚类标注，然后再用标注之后的样本去进行监督学习。

七、课后习题

使用 sklearn.datasets.load_iris() 方法加载鸢尾花数据集，并仅加载特征数据使用 K-Means 完成聚类。然后对比聚类结果和实际的分类情况。

下一节：K 值选择与聚类评估 (/courses/880/labs/3189/document)