



## 第 5 章 Matplotlib 绘图

数据可视化能够更加直观地展示数据分布情况和机器学习的结果,一幅精心绘制的图形能够帮助读者在数以千计的零散信息中做出有意义的比较,提炼出使用其他方法不那么容易发现的信息。Python 有着非常丰富且强大的绘图功能,本章将通过讲述创建图形到输出保存图形的整体流程,以及具体修改图形特征来介绍 Matplotlib 模块,向读者逐步呈现 Python 绘图。

### 5.1 Matplotlib 绘图基础

Matplotlib 是 Python 中的 2D 绘图库,也是最著名的 Python 绘图库。虽然 Matplotlib 的代码库很庞大,但是可以通过简单的概念框架和重要的知识来理解和掌握。Matplotlib 中的图形分为以下 4 层结构。

- (1) canvas (画板), 位于最底层, 导入 Matplotlib 库时自动存在。
- (2) figure (画布), 建立在 canvas 之上, 从这一层就能开始设置其参数。
- (3) axes (子图), 将 figure 分成不同块, 实现分面绘图。
- (4) 图表信息 (构图元素), 添加或修改 axes 上的图形信息, 优化图表的显示效果。

为了方便快速绘图, Matplotlib 通过 pyplot 模块提供了一套与 Matlab 类似的命令 API, 这些 API 对应一个个图形元素 (如坐标轴、曲线、文字等), 并可以此对该图形元素进行操作, 而不影响其他部分。创建好画布后, 调用 pyplot 模块所提供的函数, 仅几行代码就可以添加、修改图形元素或在原有图形上绘制新图形。

#### 5.1.1 编程风格

##### 1. 基本流程

根据 Matplotlib 图形的 4 层结构, pyplot 模块绘制图形基本上会遵循一个流程, 使用这个流程可以完成大部分图形的绘制。pyplot 模块基本绘图流程主要分为 3 部分, 如图 5-1 所示。

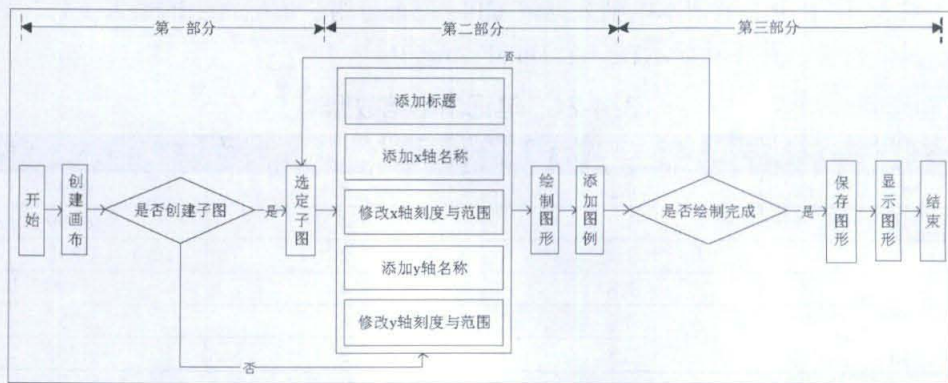


图 5-1 pyplot 模块的基本绘图流程