目录

第 5 章

监督学习：高级算法

这是监督学习过程的第二部分。 您现在正在研究监督学习生态系统中更先进的机器学习算法。

提示：这些算法通常用于工业化的机器学习生态系统，因为它们解决了许多常见的问题，您将在示例和它们的解决方案中看到这些问题。

Boosting（元算法）

机器学习术语“Boosting”是指一系列算法，这些算法使弱学习者适应强学习者。 当特征不能产生足够强的学习者来学习算法时，这些方法是有用的。Boosting 算法使用一组低精度分类器来创建一个高精度分类器。

基本概念（图 5-1）是将机器学习链连接成一个处理链，以获得改进的结果。

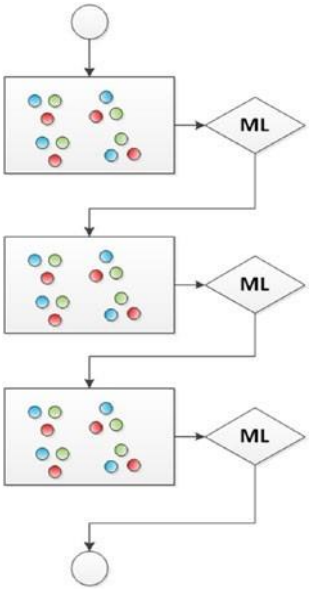

图 5-1。ML Boosting 概念图

这是一个复杂的介绍。 我将快速解释弱学习者的概念。

“弱学习者”是一种ML算法（用于回归/分类），它提供的精度略优于随机猜测。

Ada Boost(适应性Boost)

Ada Boost或Adaptive Boosting是一种使数据科学家能够增强数据处理结果以获得更好的结果的方法。

Ada Boost分类器是一个元估计器，它首先在原始数据集上拟合分类器，然后在同一数据集上拟合分类器的附加副本，但调整错误分类实例的权重，使后续分类器更多地关注困难的情况。

例：

在您的 Jupyter 软件中打开示例 Jupyter 笔记本：第 005 章示例 001A.ipynb。

使用Part A，您将激活所需的核心库。

以下三个导入对于您需要执行的工作非常重要。

```
from sklearn.ensemble import AdaBoostClassifier
```

Ada Boost 分类器使用 Ada Boost-SAMME 算法集。

这个具体的例子是使用 SAMME 离散升压算法。

你需要的下一个结构是你想要 boost 的机器学习算法。

您正在使用支持向量分类(SVC)算法。

```
from sklearn.preprocessing import StandardScaler
```

数据工程使用标准定标器来准备特征。

```
from sklearn.preprocessing import StandardScaler
```

变压器通过删除均值和缩放到单位方差来标准化特征。

既然你有了基本的工具，我将向你展示如何构建一个 boosting 解决方案。

B 部分和 C 部分提供您需要的数据。 我建议我们用我的 rose 上的数据。

在 D 部分中，您使用变压器执行所需特性的缩放。

```
transformer = StandardScaler(copy=True, with_mean=True, with_std=False)
```

这将转换数据以支持更好的培训过程。

变压器结果为：

```
Features: 3
Samples: 420
Scale: None
Mean: [5.82341667 3.07140952 1.22077619]
variance: None
```

In Part E you create the SVC ready for the AdaBoost to use.

```
svc=SVC(max_iter=5000,
    gamma='auto',
    class_weight='balanced',
    probability=True,
    kernel='linear',
    random_state=0,
    verbose=False)
```

这个 SVC 将被 Ada Boost 使用。

在 F 部分中，您将执行一个 boost 周期：

```
clf1=Ada Boost Classifier((algorithm='SAMME',
      n_estimators=1,
      base_estimatorn_estimators=svc,
      learning_raten_estimators=1,
      random_staten_estimators=0)
clf1.fit(X_train_scale, y_train)
score1 = clf1.score(X_test_scale,y_test)
```

评分结果：67.22%

在 G 部分中，您执行 5 个周期，得分为：67.7778%-改进！

在 H 部分中，您执行 10 个周期，得分为：73.3333%-改进！

在 I 部分中，您执行 20 个周期，得分为：73.3333%-没有改进？

警告　　提升只会提高到一个点，然后你只是浪费时钟周期而没有有效的增益。

您已经成功地将 67.222%提高到 73.3333%，这已经改进了 9.09%。

注意　机器学习可以是本书中的任何一种机器学习算法。

你可以关闭你已经成功完成的笔记本，你的第一个 boost 例子。

在Jupyter软件中打开示例Jupyter笔记本：第05章示例001B.ipynb。

我们将完成另一个示例，但是这个示例使用不同的基本机器学习算法和定标器。

请参阅 D 部分被称为健壮的 Scaler 的定标器。 这个定标器使用对异常值具有鲁棒性的统计数据来转换特征，因为它删除中值并根据分位数范围对数据进行缩放。

E 部分是您可以使用额外的树分类器作为基本机器学习过程的点。

在 I 部分中，您将使用 1000 周期的 boost。

提示 要使 Jupyter 笔记本显示 1000 个周期，您需要执行此修改：

```
%javascript

IPython.OutputArea.prototype._should_scroll=function
（lines）{

return false;

}
```

正如您在 J 部分中所观察到的，您实现了从 70.270%到 97.200%的分数改进，因此提高了 38.324！ 这就是 boost 的力量。

你可以关闭这个笔记本。
接下来，我们将研究Boost解决方案的值域结果。
打开示例Jupyter Notebook：第5章示例001C.ipynb
执行完整的笔记本。
随着解决方案的推进，您实现了69.440%到82.220%的改进。
我想让你看看在boost1到5之间缺乏改进。
如果我们绘制预测值域，您将观察到值域中有一个主要的值移位。
您将得到以下四个结果（图5-2和图5-3）。

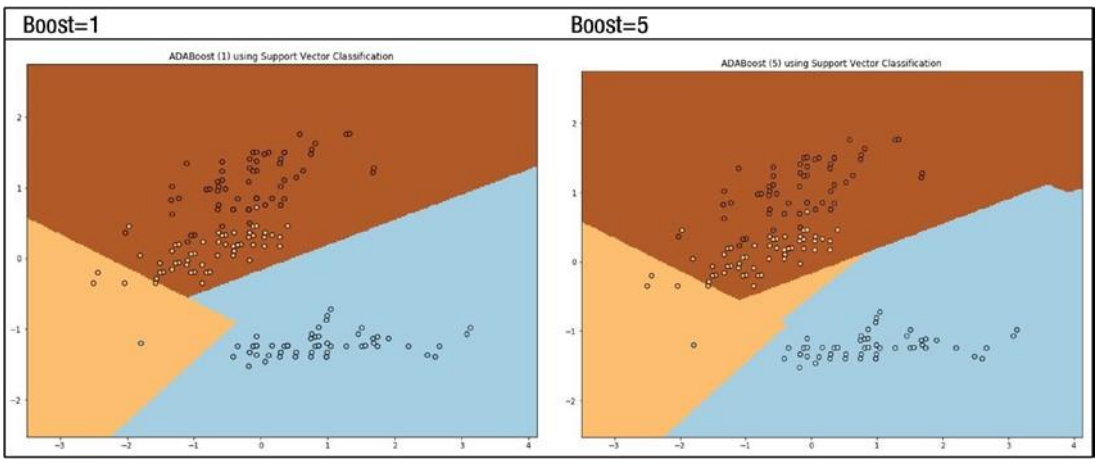

图 5-2。Ada Boost 改变了值域

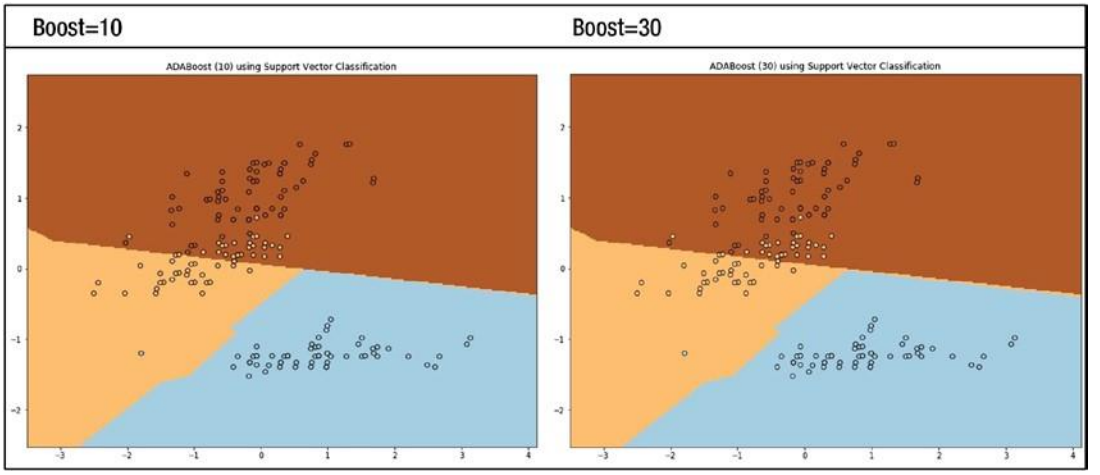

图5-3。 AdaBoost 不会更改值域

警告：

由于域边界上缺少测试数据，该移位不会出现在分数中，因为您没有在这些边缘边界中进行测试。

始终检查您的数据是否覆盖了整个值域！

您可以关闭notebook。做得好，您现在已经可以执行 boosting 了。

渐变树增强

梯度提升是回归和分类解决方案的机器学习技术，以弱预测模型（特征决策树）组合的形式生成预测模型。

例子：

打开Jupyter Notebook 的示例：Jupyter 软件中的第 005 章，示例 002.ipynb

核心引擎为：

from sklearn.ensemble import GradientBoostingClassifier

H 部分中的以下命令激活了Boosting：

clf = GradientBoostingClassifier(** params)

你的结果将如 Figure 5-4 所示。

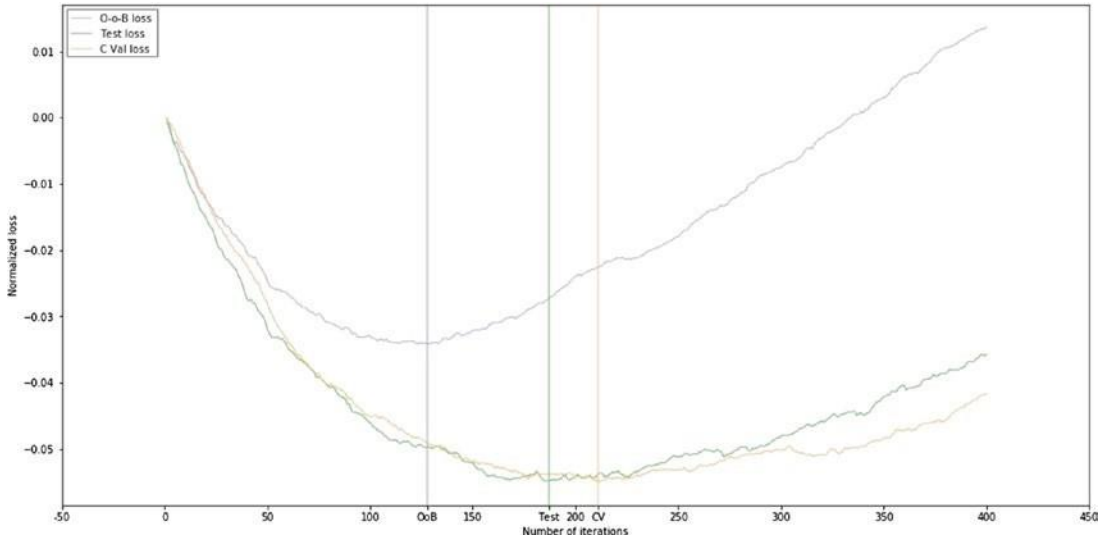

图5-4。 渐变树增强

您现在可以关闭 Notebook。

XGBoost

XGBoost 提供了一个梯度提升框架。它作为机器学习库非常成功。

该过程支持分布式处理框架 Apache Hadoop、Apache Spark 和 Apache Flink。这使得此框架具有高度。可扩展性，也因此是使用最多的库之一

Python 库是使用以下方法安装的：

conda install -c conda-forge xgboost

或

conda install -c anaconda py-xgboost （这一个是我的个人选择）

例子：

打开 Jupyter 软件中 Jupyter Notebook 的示例 ： 第五章 示例 003.ipynb。

在 A 部分中，加载 xgboost 引擎：

from xgboost import XGBClassifier

在 C 部分中，您通过以下方法使用它：

```
xc = XGBClassifier(max_depth=12,
        learning_rate=0.05,
        n_estimators=1968,
        nthread=8)
```

您的结果：

精度： 82.05128%

您可以使用E 部分打开机器学习的输出。

您现在可以关闭 笔记本。

这是算法示例结束的提示。

现在，您应该能够理解如何使用过程增强算法来帮助机器学习去改进模型的培训。模型训练是IML 建模部分的最重要的部分。

接下来，我们将研究一个正在出现的算法，即 TensorFlow。

TensorFlow

TensorFlow 是一个开源软件库，用于跨一系列任务进行数据流编程。它是一个象征性的数学库，也用于机器学习应用程序，如神经网络。

TensorFlow 由 Google Brain 团队于 2015 年 11 月开发。TensorFlow 可以在多个 CPU 和 GPU 上运行。

提示 谷歌的 The Machine Learning Crash Course（MLCC）培训课程将使您能够获得对 TensorFlow更好的基本准备 .

如果你想阅读更多关于背景， 我建议你访问： https://www tensorflow.org/learn

提示 TensorFlow 社区是一个强大而发达的群体。

---

https://www.tensorflow.org/tutorials

提示 作为 TensorFlow 是由谷歌大脑团队开发的。我建议使用：
https://colab.research.google.com/notebooks/welcome.ipynb

您只需上传示例即可使用它们。

注意 在本书中我已经用我自己的TensorFlow安装。

将 TensorFlow 库安装到 Python 中：

conda install -c conda-forge tensorflow

我将向您展示使用 TensorFlow 进行机器学习的基本构建基块。

TensorFlow 在将数据处理成有意义的信息方面具有巨大的能力。

打开Jupyter 软件中的示例 Jupyter Notebook：Chapter 005 示例 004.ipynb

注意 您将直接使用 TensorFlow。

本章的稍后部分，我将向您展示如何使用Keras 作为接口。

在 A 部分中，您将注意到 TensorFlow 接口将 TensorFlow 引入为 tf。

在 B 部分中，您将执行一些基本数学内容，以此向您介绍基本概念。

声明常量如下：a = tf.constant(5)。这是完成 a =5 的。

因此，让我们执行以下计算：

a=5.000, b=10.000, c=20.000, d=12.000, e=89.000
x=5.000 + 10.000 + 20.000
Addition with constants: x = 35
y=5.000 x 10.000 x 20.000
Multiplication with constants: y = 1000

z=2.000 ^ 12.000
Power with constants: z = 4096
s=sqrt(89.000)
Square root with constants: s = 9

我将向您展示如何使用 TensorFlow 处理矩阵。

创建产生 1x3 矩阵的常量操作。该
操作作为节点添加到默认图中。构造函数返回的值表示常量操作的输出。

Create Constant that produces a 1x4 matrix.

matrix1 = tf.constant([[10., 11., 12.,13.]])

Create another Constant that produces a 4x1 matrix.

matrix2 = tf.constant([[14.],[15.],[16.],[17.]])

创建以 " 矩阵1"和"矩阵2"作为输入的 Matmul
操作。返回的值"product"表示矩阵乘法的结果。

product = tf.matmul(matrix1, matrix2)

为了运行matmul 操作，我们调用‘run（ ）’方法，传递"product"。

操作的输出以 "结果" 作为numpy ‘ndarray’对象返回。

```
with tf.Session() as sess:
    result = sess.run(product)
    print(result)
```

您的结果：
[[718.]]
调查 G 部分计算以测试系统的功能：更改i=1000 并看看它的厉害吧！
恭喜您能够执行基本的 TensorFlow 处理。

预测 TensorFlow 目前是机器学习生态系统中的主要主力。
我预测，到2025年，这个具有伴随生态系统的图书馆将成为一个领导系统。

在开始运行 下一个示例之前，以下术语非常重要：

纪元

纪元是给定数据样本的完整迭代。纪元数是算法要循环或迭代的时间量。纪元数直接影响训练步骤的结果。纪元太少可能导致算法仅 达到局部最小值，使用的迭代数越多，当您达到全局最小值或至少达到更好的局部最小值时会产生更好的结果。

层

机器学习模型被组装为被称为层的逻辑链的集成和堆叠。具有确定功能目的的层分离可产生更简单和可重用的层。

TensorFlow 提供了一组良好的预配置有用 common 层， 为您的机器学习提供高效和有效的处理环境。

您的模型将由这些层创建的多个有用的操作层组成。

估计

估计器是TensorFlow最具伸缩性和面向生产的模型类型，它使您能够使用预先制作的估计器，使您能够将处理工作带到比TensorFlow基础支持更高的概念级别。这是TensorFlow 的一个功能，它非常成功。

检查站

检查点是在训练期间创建的模型的版本，然后保存供以后使用。这对于确定新模型对旧模型的成功与否很有用。

现在，您可以验证在其他数据处理示例中如何使用 TensorFlow。

打开示例notebook：

- Jupyter Notebook: Chapter 005 Example 005A.ipynb
- 在 Chapter-005-Example-005A-01.txt 中查看结果
- Jupyter Notebook: Chapter 005 Example 005B.ipynb
- 0.9333 的 准确率
- Jupyter Notebook: Chapter 005 Example 005C.ipynb
- 在 Chapter-005-Example-005C-01.txt 中查看结果

完成这些工作簿后，您就已经使用 TensorFlow 完成了。

贝叶斯统计

贝叶斯统计学是统计学理论，基于贝叶斯对概率的解释，概率表示事件发生可信度的大小，随着新信息的收集而变化，而不是基于频率或有倾向的固定值。

伯努利试验

伯努利试验是一个随机试验，只有两个结果，通常标记为"成功"或"失败"，其中每次试验成功的概率完全相同。成功的概率由 θ 给出，这是介于 *0* 和 1 之间的数字。因此 θ ∈[0，1]。

打开 Jupyter 软件中的示例 Jupyter Notebook：第 005 章示例 006A.ipynb。

在此示例中，我们将使用名为"头或尾"的游戏来演示你怎样可以预测如何处理"成功"或"失败"的方法。这可能是：

您赢得了客户或失去了他们。

您的产品工作或失败。

机器人捡起草莓或掉在地上。

您可以将此应用到任何双输出方法。

现在运行完整的笔记本，定义硬币人头面为成功！

您的输出是：

```
Perform Test (001):      0 trials,      0 heads,      0 tails ->  0.0000 %
Perform Test (002):      2 trials,      1 heads,      1 tails -> 50.0000 %
Perform Test (003):     10 trials,      5 heads,      5 tails -> 50.0000 %
Perform Test (004):     20 trials,      9 heads,     11 tails -> 45.0000 %
Perform Test (005):     50 trials,     23 heads,     27 tails -> 46.0000 %
Perform Test (006):    500 trials,    246 heads,    254 tails -> 49.2000 %
Perform Test (007):   1000 trials,    488 heads,    512 tails -> 48.8000 %
Perform Test (008):  10000 trials,   4957 heads,   5043 tails -> 49.5700 %
```

您的图形结果如图 5-5 所示。

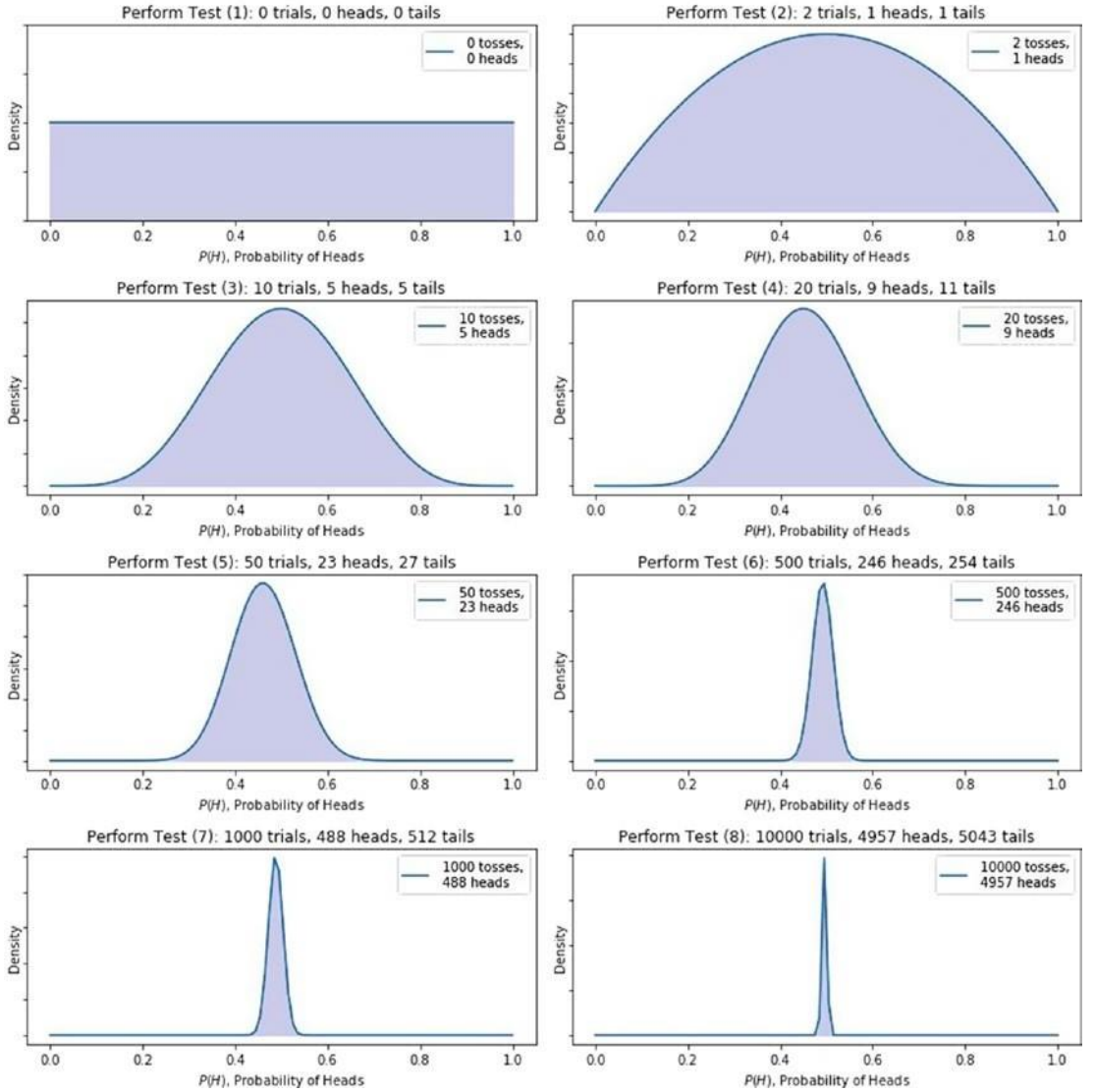

图5-5。 伯努利试验

你可以观察到，随着试验数量的增加，"人头的概率"接近你期望的 50%，。您可以保存笔记本，也可以试用：

number_of_trials = [0, 2, 10, 20, 50, 500, 1000, 10000]

参见： Chapter 005 示例 006A.ipynb 的一个有 20 条轨迹的斐波纳契曲线。。

接下来，我们将讨论执行 IML 所需的常识。

基于案例的推理

基于案例的推理（CBR）（ 图5-6）通常来

说，是基于对过去类似问题的解决方案来解决新问题的过程。

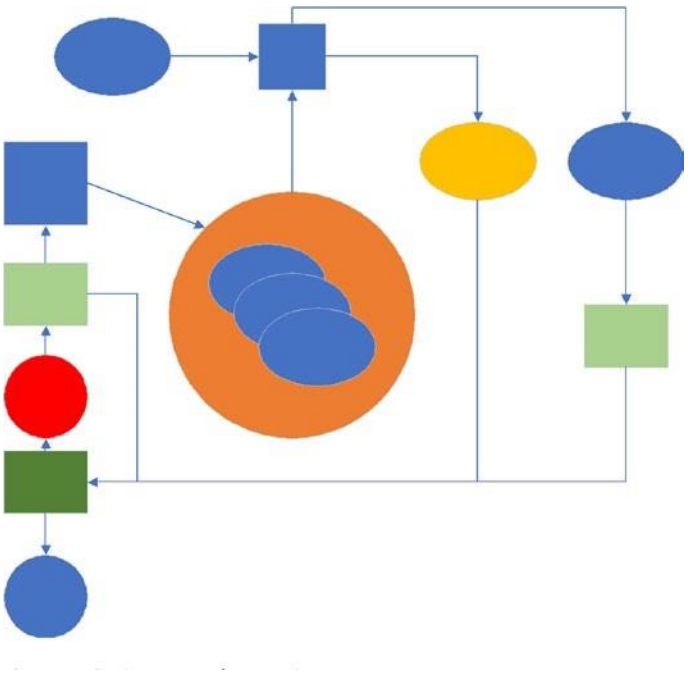

图5-6。 基于案例的推理

基于案例的推理已经正式化，用于计算机推理作为一个四步走的过程。

检索

给定目标问题，从与解决它相关的内存大小中检索。案例包括问题、其解决方案，以及对

有关解决方案如何派生的解释。例如，假设Augs想准备肉桂咖啡。作为一个新手咖啡师，他还记得最相关的经验是他成功地做了纯咖啡。他遵循做纯咖啡的程序，以及根据这个方法做出的解释，构成了Augs的参考案例。

重用

将解决方案从上一个案例映射到目标问题。这可能涉及根据需要来调整解决方案以适应新的情况。在咖啡示例中，安格斯必须调整他的参考解决方案，包括添加肉桂。

修改

将前一个解决方案映射到目标情况后，在真实世界（或模型）中测试新解决方案，并在必要时进行修改。假设 Angus 通过将肉桂棒加入热水来调整他的咖啡解决方案。混合后，他发现咖啡有一些肉桂棒，而不仅仅是味道 ——一个不理想的效果。应该做如些修改：慢慢地向咖啡中添加水，直到过滤后，然后用它来煮咖啡。

保留

成功适应目标问题后，将生成的经验存储为内存中的新情况。因此，Angus 记录了他制作肉桂咖啡的新程序，从而丰富了他的一套储存经验，并使它对未来的咖啡制作更有经验。

强化学习

强化学习 （RL） （图 5-7） 是机器学习的分支。它被认为是监督和非监督学习的混合体。它基于试验和错误的方式模拟人类学习，就像 Angus 从一开始不会做到现在可以做肉桂咖啡。我们将在第九章中详细讨论这个问题。

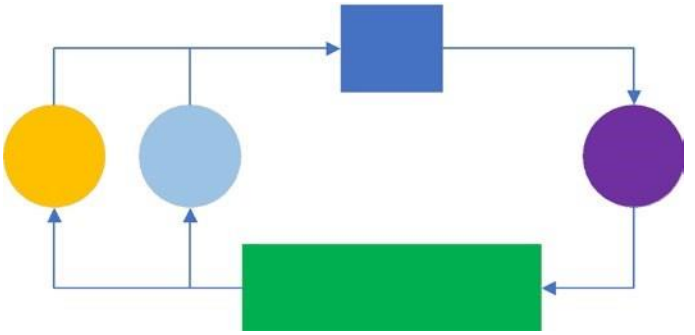

图5-7。 强化学习

归纳逻辑编程

归纳逻辑编程（ILP） 是在机器学习和逻辑编程的交汇处形成的一个研究领域。

高斯过程回归

在概率理论和统计学中，高斯过程是一个随机过程，因此随机变量的每个有限集合都有一个多变量正态分布，即每个值的有限线性组合都是正态分布的。高斯过程的二分法是随机变量的联合分布和连续域内函数的分布，例如时间或空间。

涉及高斯过程的机器学习算法使用惰性学习和
点（内核函数）之间的相似性度量，以便从训练数据中预测不可见点的值。

例子：

打开Jupyter 软件中的示例 Jupyter Notebook：第 5 章示例 007.ipynb。

我用了 sklearn.gaussian_process.GaussianProcessRegressor。

我还使用此机器学习过程演示如何向函数 y = 1.968 *（x * np.cos（x））
的观测值添加噪声如何提高合理预测值的概率。

对于 A 部分，您只需使用观测值，即无噪声。

```
gp = GaussianProcessRegressor(kernel=kernel,
                              optimizer='fmin_l_bfgs_b',
                              alpha=1e-10,
                              n_restarts_optimizer=10,
                              random_state=0)
```

在 B 部分中，通过 Tikhonov 正则化将噪声添加到高斯进程回归器引擎的 α 超参数中：

```
y = f(X).ravel(); dy = 0.5 + 1.0 * np.random.random(y.shape);
noise = np.random.normal(0, dy); y += noise
gp = GaussianProcessRegressor(kernel=kernel,
                              optimizer='fmin_l_bfgs_b',
                              alpha=dy ** 2,
                              n_restarts_optimizer=10,
                              random_state=0)
```

您的结果如图 5-8 和 图 5-9 所示。

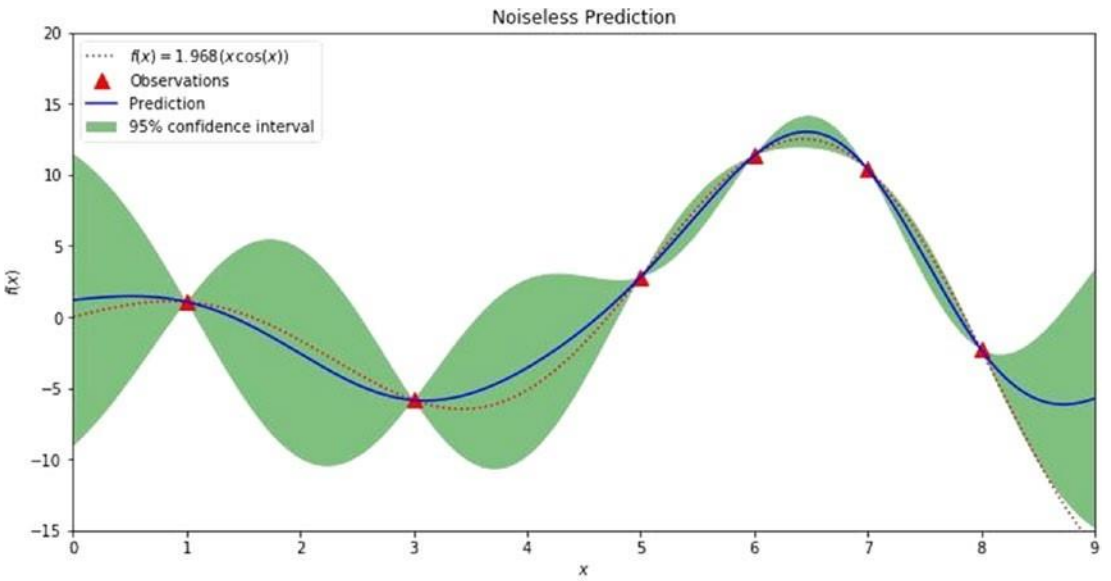

图5-8。 高斯过程回归（1）

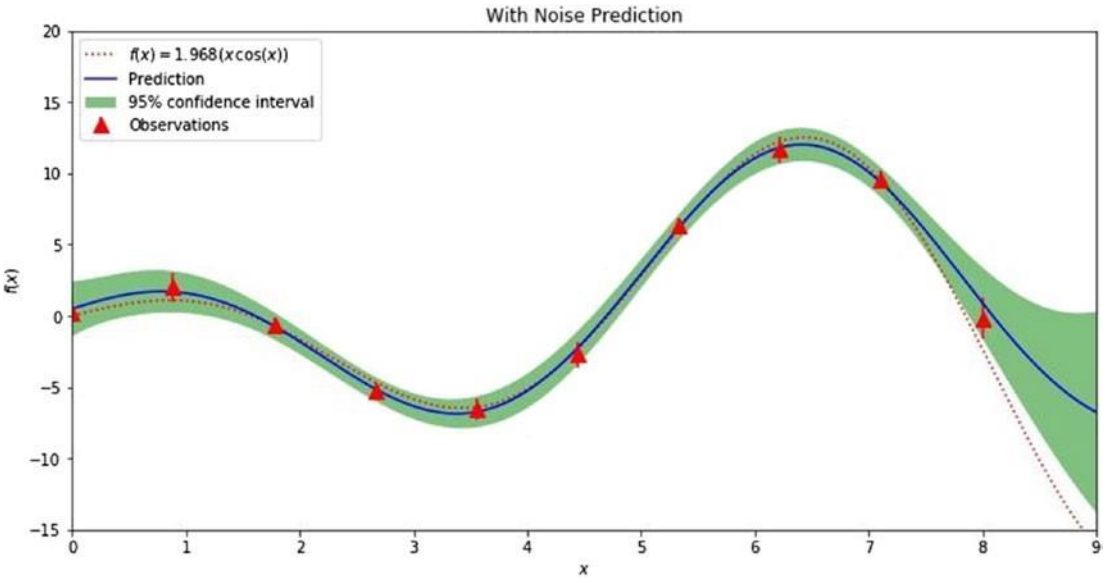

图5-9。 高斯过程回归（2）

您可以清楚地看到，增加噪音的改进是显著的。

提示 通过添加噪声对许多实际机器学习训练集进行惩罚是保证模型训练更多观测结果并提高预测质量的良好做法。

您现在可以关闭笔记本。

下面的示例将介绍更多技术来增加噪声以求解更复杂的要求。

打开 Jupyter 软件中的示例 Jupyter Notebook：第 005 章示例 008.ipynb。

A 部分的解决方案是涵盖核心观测值，但对异常值不起作用。

您的结果如图 5-10 所示。

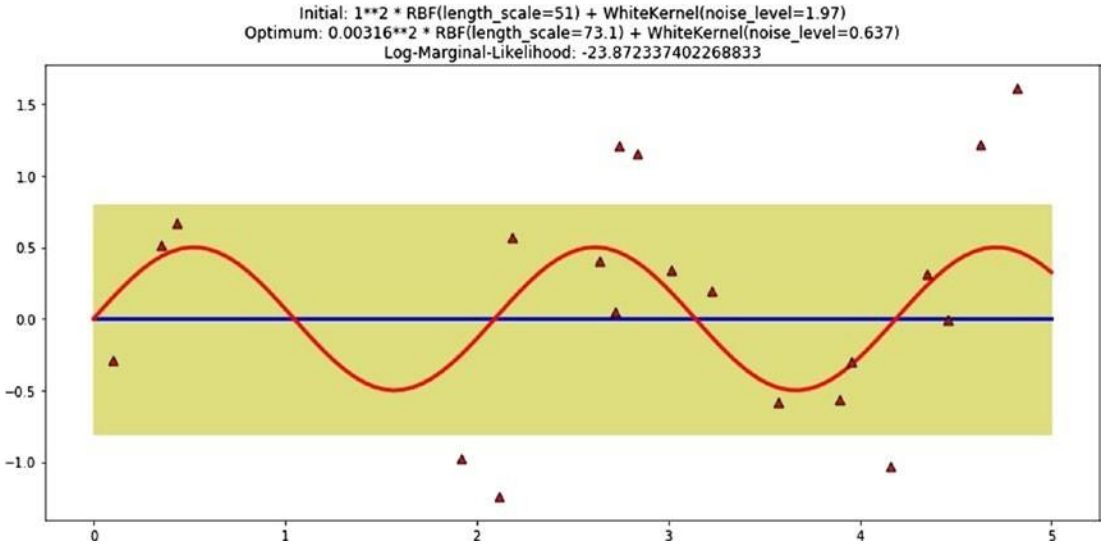

图5-10。 使用蒂霍诺夫正则化的高斯过程回归 （1）

概率带也导致预测问题，并且更加不准确，得分仅为 -4.3556%。

您需要通过内核函数更好地添加噪声。

B 部分解决方案能以更好的方式覆盖核心观测值和异常值。

概率带更好，并且得分为 66.6001%。

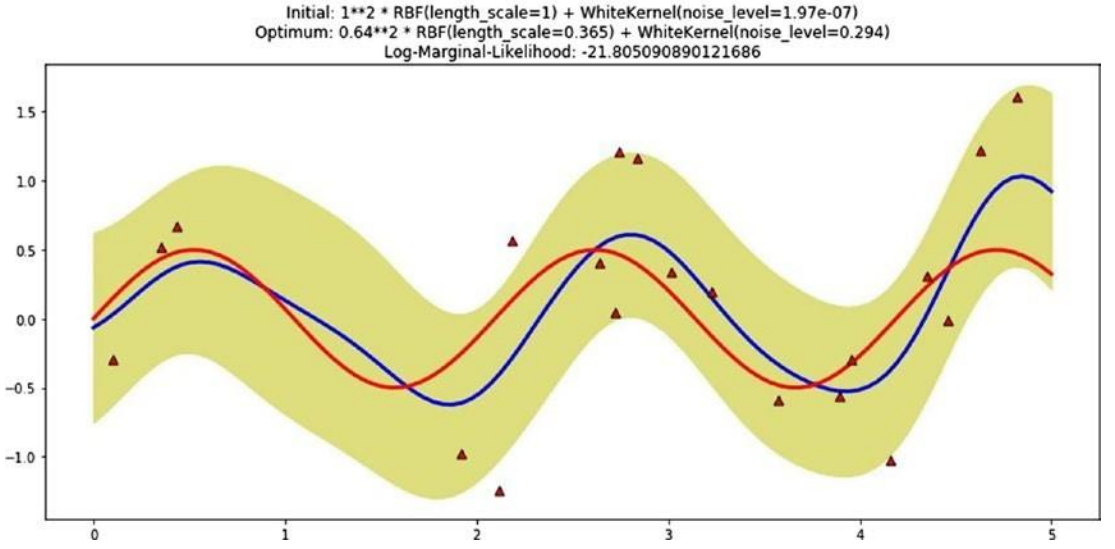

图5-11。 使用蒂霍诺夫正则化的高斯过程回归 （1）

C 部分解决方案以更好的方式涵盖核心观测值和异常值。概率带更好，分数为66.6001%。

您可以关闭此Notebook。

现在，您应该了解，可以通过添加一部分噪声来改进模型的预测。

内核密度估计器（Kernel Density Estimators）

内核密度估计是一种非参数化方法，用于估计连续随机变量的概率密度函数（PDF）。它是非参数化的，因为它不假定变量的任何基础分布。

例子：

打开Jupyter 软件中的示例Jupyter Notebook：第 005 章示例 009.ipynb。

在此示例中，我们正在调查一组已知值，即手写数字;然后我们推断出一组我们期待的数字。

您将使用主体复合分析 （PCA） 过程，使用数据的奇异值分解来执行线性维度缩减，以投影到较低维空间中。

之后，您将使用网格搜索交叉验证五个周期执行内核 密度估计。

{'cv': 10,

'error_score': 'raise-deprecating',

'estimator__algorithm': 'auto',

'estimator__atol': 0,

'estimator__bandwidth': 1.0,

'estimator__breadth_first': True,

'estimator__kernel': 'gaussian',

'estimator__leaf_size': 40,

'estimator__metric': 'euclidean',

'estimator__metric_params': None

'estimator__rtol': 0,

'estimator': KernelDensity(algorithm='auto', atol=0, bandwidth=1.0,
breadth_first=True,

kernel='gaussian',leaf_size=40,metric='euclidean',

metric_params=None, rtol=0),

'fit_params': None,

'iid': True,

'n_jobs': -1,

'param_grid': {'bandwidth': array([0.1 , 0.1274275 , 0.16237767, 0.20691381,
0.26366509, 0.33598183, 0.42813324, 0.54555948, 0.6951928 ,
0.88586679, 1.12883789, 1.43844989, 1.83298071, 2.33572147,
2.97635144, 3.79269019, 4.83293024, 6.15848211, 7.8475997 ,
10.]),

'leaf_size': array([35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51,
52, 53, 54]),

'kernel': array(['gaussian', 'tophat'], dtype='<U8'),

'algorithm': array(['kd_tree', 'ball_tree'], dtype='<U9')},

'pre_dispatch': '2*n_jobs',

'refit': True,

'return_train_score': 'warn',

'scoring': None,

'verbose': 1}

最好的参数是：

{'algorithm': 'kd_tree',
'bandwidth': 3.79269019073225,
'kernel': 'gaussian',
'leaf_size': 51}

网格搜索为解决方案的训练数据找到了最佳参数集。

结果如图 5-12所示。

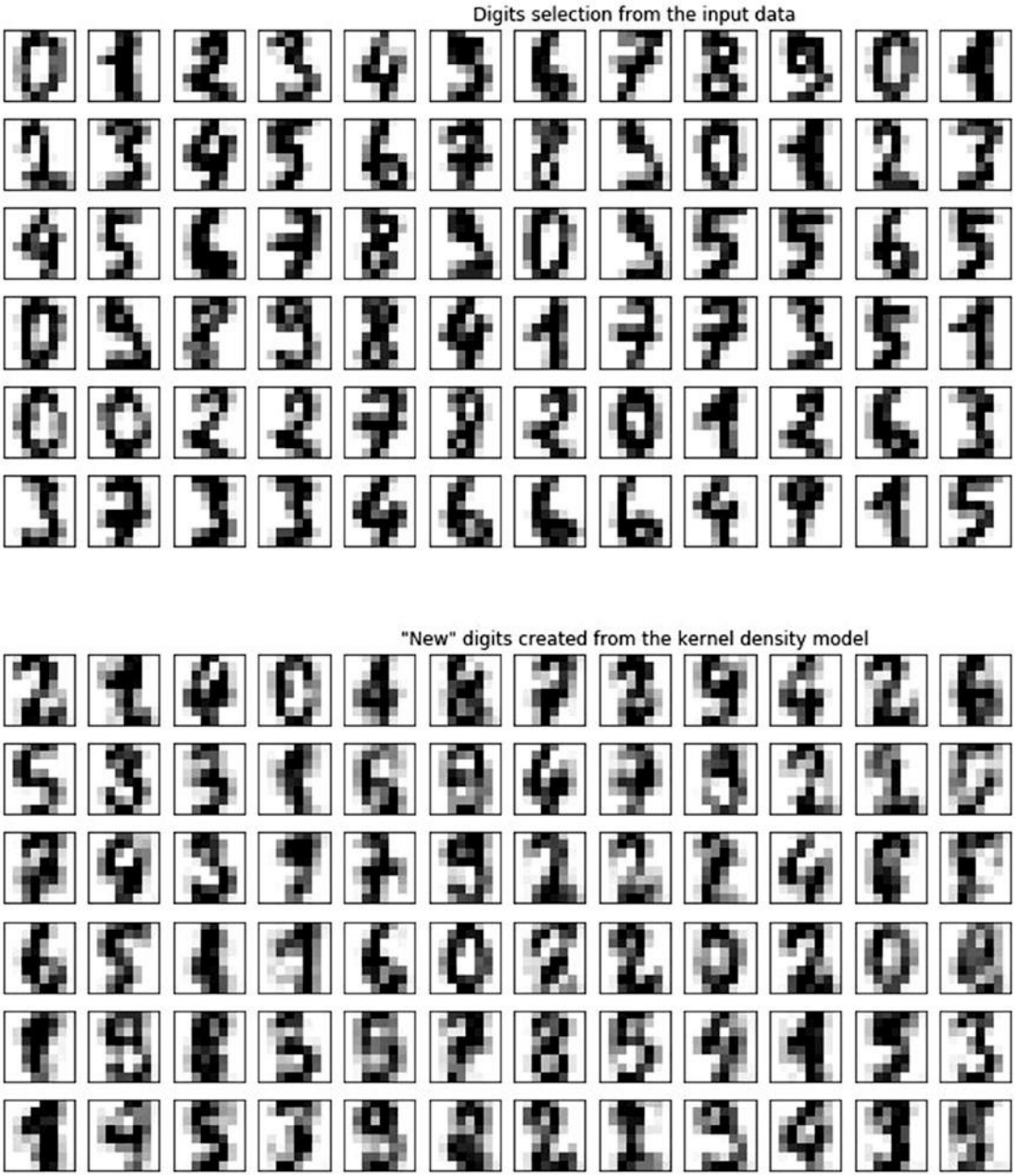

图5-12。 使用网格搜索交叉验证的内核密度估计

生成的第二个数字块是 0、1、2、3、4、5、6、7、8 或 9 范围内手写数字的模型的可能匹配项。

您现在可以成功使用内核密度估计。

请关闭笔记本。

马维三维可视化器

接下来，我想向您介绍我用于数据科学和机器学习项目的Mayavi科学数据三维可视化器。

参见： https://docs.enthought.com/mayavi/mayavi/

要在Python 中使用 Mayavi 3D 科学数据可视化和绘图，您需要：

conda install -c anaconda mayavi

打开Jupyter 软件中的示例 Jupyter Notebook：第 02 章示例 010A.ipynb。

此可视化工具工具包使您能够创建非常详细的 3D 可视化效果。

如果运行 Notebook，您将在结果目录中获得以下 3D 图像（图 5-13）。)

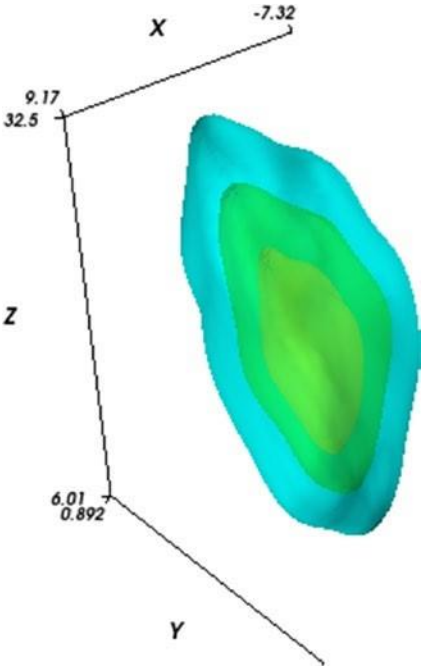

图5-13。 *使用 Mayavi 3D 引擎进行内核密度估计*

注意 您可以通过放大和缩小这些图像来交互...或旋转它们。

如果您观察：第 02 章示例 010B.ipynb 和第 02 章 示例 010C.ipynb，您将看到可视化的复杂性。

这是一个很好的可视化工具 。请花时间去 了解马哈维的能力 。

随机林

随机林是树预测变量的组合，因此每个树都依赖于独立采样的随机矢量的值，并且林中所有树的分布相同。当林中的树的数变大时，林的泛化误差将收敛到一个限制。

树分类器林的泛化误差取决于林中单个树的强度及其之间的相关性。

例子：

打开Jupyter 软件中的示例 Jupyter Notebook：第 005 章 示例 011.ipynb。

运行完整的Notebook。

您正在查看两个重要的随机林引擎。

您的结果将是：

随机森林分类器引擎得到：

得分 83.140%

混淆矩阵为：（tn=2101，fp=396，fn=447，tp=2056）

外树分类器引擎实现：

得分 87.080%

混淆矩阵为： （tn=2205， fp =292， fn=354， tp=2149）

这表明，对于此特定数据集， 额外树分类器更好。

您现在可以关闭笔记本，因为我们接下来观察它如何针对一些真实数据的表现。

打开Jupyter 软件中的示例 Jupyter 笔记本：第 005 章示例012.ipynb。

运行完整的笔记本。

您的结果将对三种颜色的玫瑰的数据进行分类。

预测物种	弗洛里本达（白色）	罗莎·科尔德西（红色）	玫瑰花（蓝色）

实际物种红斑狼疮（蓝色）			
	0	0	41
弗洛里本达（白色）	53	4	0
罗萨·科尔德西 （红色）	0	45	0

这四个功能对结果的影响如下：

Leaf Length (mm) : 9.3581%
Leaf Width (mm) : 3.2167%
Stem length (mm) : 43.2882%
Stem width (mm) : 44.1370%

您现在可以关闭笔记本。

您已完成随机林。

接下来，我将向您展示如何处理不太完美的数据，需要一些预处理才能有用。

处理不平衡的数据集

不平衡的类使"准确性"不在考虑范围。这是机器学习中一个令人惊讶的常见问题（分类中的特定问题），发生在每个类中观测值不成比例的数据集中。

标准精度不再可靠地测量性能，这使得模型训练更加棘手。

我将向您展示如何通过向您展示如何预处理数据集以修复数据结构中的不平衡来处理这些不准确的数据集。

对于我们的第一个示例，我们将解释这些目标变量如何具有三个类。

R 表示右边重，也就是说， *var3* * var4 > var1 * var2

L 表示左边重，也就是说，*var3* ∗ var4 < var1 ∗ var2

B 表示平衡，也就是说，*var3* ∗ var4 = var1 ∗ var2

使用 不平衡学习 库

conda install -c conda-forge imbalanced-learn

我们将使用的不平衡学习库由Fernando Nogueira于 2014 年 8 月启用。

打开Jupyter 软件中的示例 Jupyter Notebook：第 005 章示例 013.ipynb

对于此示例，您将使用多个标准数据集，使您能够将机器学习引擎及其行为与这些数据组进行比较。

您的结果如图 5-14 所示。

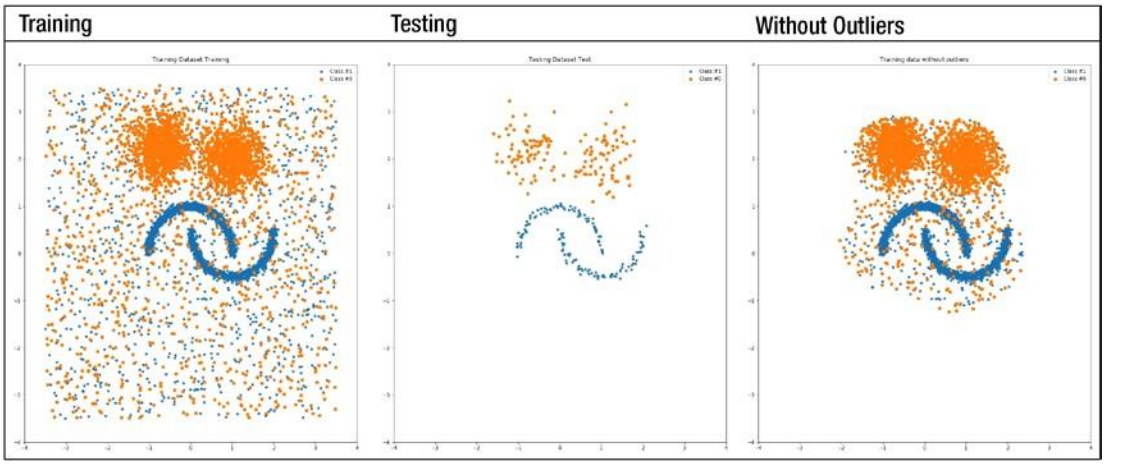

图5-14。 处理不平衡的数据

不平衡类出现在许多域中，包括以下域。

欺诈检测

欺诈检测是一个很有趣的问题。事实是，欺诈性交易易在组织数据中是罕见的；
实际上，它们只占有
组织活动的很小一部分。挑战在于，如果没有合适的工具和系统，一小部分活动会迅速变成财产损失。

罪犯很狡猾。由于老式的欺诈计划一般不能得
手，欺诈者已经学会了改变策略。好消息是，随着机器学习的改进，系统可以学习、调整和发现新的模式来预防诈骗。

垃圾邮件过滤

电子邮件筛选是处理电子邮件，并根据指定的条件组织它。
大多数高级状态是自动处理传入消息。

疾病筛查

在医学中，筛查是一种在人群中用于识别无症状的个人中是否存在尚未确诊的疾病的策略。通常包括症状前或未识别的症状性疾病的个体。筛查测试相对较少，因为它们对实际身体健康的人很熟悉。

广告点击

广告点击率 （CTR）是单击特定链接的用户与查看页面、电子邮件或广告的总用户数的比率。它通常用于衡量特定网站的在线广告活动是否成功以及电子邮件广告活动的有效性 。

应用

生物信息学

生物信息学是一个跨学科的领域，它
改进了研究生物数据的方法和软件工具。它是信息工程、生物学、计算机科学、数学和统计学的跨学科领域，用于检测和检验生物数据。生物信息学已经通过机器学习技术使用，并
应用先进的数学和统计技术。生物信息学是为在DNA研究中识别候选者的基因和单核苷酸多肽转性（SNPs）而建立的。机器学习处理的极端速度和容量
使这成为一位优秀的数据科学家和机器学习工程师的非凡工具。

工业化机器学习在这一领域的应用已经超过了几年前几项技术的有效发现;如果没有机器学习的进步，我们现在正在进行的研究水平是不可能的。

数据库营销

数据库营销
是一种直接营销形式，它使用客户数据库生成直接营销通信的目标列表。数据库
包括客户的姓名和地址、电话号码、电子邮件、信息请求、
购买历史记录，以及可合法、准确地收集的任何其他相关数据，在客户在
日常生活中执行基本业务活动时收集这些数据库的信息。

这些重要的数据库在如今时代被珍藏为高度排他性的公司资产和在监督学习世界中备受追捧的资源。

人机回圈

然而，由于对培训受监督学习系统的数据的依赖性，机器学习的到来也为人类创造了新的就业机会。这些过程称为"人机回圈"，此技术使 ML 能够对明确分类的数据执行监督学习;但是对于那些模棱两可或是明显的离群值的项目，则由人工操作员来决定。

然后，此新决策将添加到训练数据中，并重新训练模型。

数据标记已成为全球的大企业。许多合法数据公司收集个人信息，并出售从许多公共来源收集的客户数据。公共记录只是一种选择，但人们通过社交媒体和商业网站一次又一次自愿分享的信息为客户数据库带来了相当丰富的收获。

这就造成了这样一种情况，即拥有并不违法但通过非法手段获得的数据可以在数据源所有者之间传递，而不必知道非法数据库现在是如何与专注于公共信息的合法数据库混合在一起的。然后，这些个人数据就成为了公众可获得性的根深蒂固的基础，即使它本来不应该公开的。

最终结果可能是几乎完整的个人信息和生活历史可供出售，并以非常实惠的价格出售。

警告

在简单地将数据接受到机器学习环境中之前，请确保您具有全面的数据沿袭和处理这些数据的合法权限。

我建议你看看一般数据保护条例 2016/679 等法规。它是欧盟（EU）法律2018年《欧洲经济区和数据保护法》中关于欧盟内部所有个人的数据保护和隐私的法规。

在美国，请参阅《联邦贸易委员会法》（15 U.S.C.§§41-58）（FTC法案），了解数据保护规则和原则，包括数据控制者的义务和数据主体的同意；访问个人数据或反对收集个人数据的权利；以及安全要求。它还包括cookies和垃圾邮件、第三方的数据处理以及数据的国际传输。这些规则解释了权力数据监管机构的优点，它的强制执行权。制裁和补救措施。

警告 出错会使您和您的公司面临罚款和其他法律诉讼。

机器学习方法

我想向您介绍一个跨行业的数据挖掘标准流程，并以其首字母缩略词被称为为 CRISP-DM 。这是一个数据挖掘过程模型，它定义了数据科学家的通用方法，并用于处理机器学习项目。机器学习的七个步骤如图 5-15 所示。

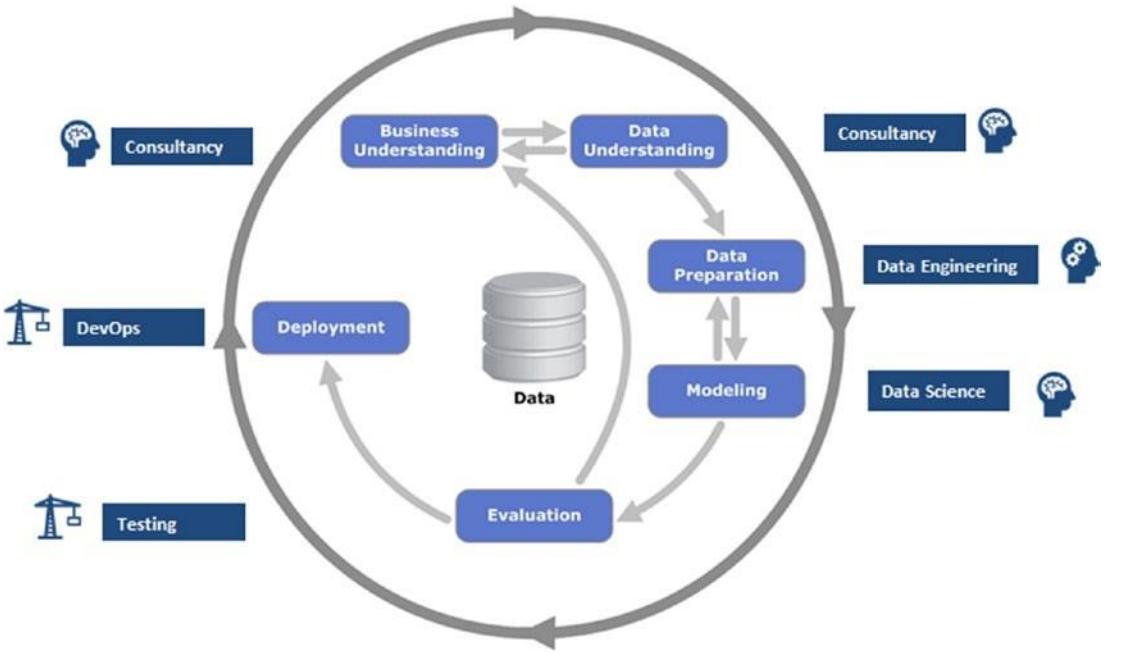

图5-15。 数据挖掘跨行业标准流程 （CRISP-DM）

首先，Crisp - dm 中有哪些人，他们都在做什么？

数据科学家

数据科学家使团队能够选择和实施最有效和高效的算法和机器学习。数据科学家将数据转换为可操作的事物。

数据工程师

数据工程师使团队能够提取、转换所需数据并将其加载到数据科学结构中，并开发用于建模的数据功能。

DevOps

DevOps通过自动化、连续测试和集成，确保受控的、受管理的发布到生产中，以实现持续有效的投资回报。

CRISP-DM 循环

参见图 5-15。

业务理解

CRISP-
DM的第一部分要求您详细了解业务的需求。你很可能会收到竞争或冲突的需求，关于什么是真正的需求来自业务。如果你不能让这个阶段百分之百正确地工作，那么你最多只能对错误的问题给出正确的答案。确保您收集了所有相关的功能和非功能需求，以确保您对业务的期望有一个完整的看法。

项目所需的输出是什么？

此时需要检查的主要项目是确定业务需求的优先级，并将相关的期望输出聚集在一起，以提高团队的数据科学和数据工程能力的吞吐量。

请注意 我使用Agile scrum-style的项目管理方法，您只需将所有期望的输出放在工作积压中。然后，作为一个团队，我们将积压工作处理成逻辑上的输出。

在这个过程中，你的工作清单就完成了。

设定目标

在此阶段，您必须从业务角度找到主要目标。这个核心或主要目标是你必须始终回答的唯一答案。

示例：如果您的主要目标是：“如何阻止高价值客户离开？”

你的所有目标必须支持这一个问题。

次要目标可能是：

- 谁是我的高价值客户？
- 他们为什么要离开？
- 如何阻止他们？

系统目标必须支持这些主要问题。

- 需要哪些数据？
- 数据的格式是什么？

生成项目计划

该项目需要任何其他ML解决方案。此时，您的标准项目管理规则将适用。我通常使用Agile项目，因为它对我的客户很有效。

业务成功标准

您必须确定 企业的最低标准是什么，才能在项目上取得成功。

我使用SMART标准：

- Special – 精确定位您的成就。比如，减少 1，000 名客户离开是好事。
- Measurable = 所需的结果是否可量化？你能确定现在处于什么阶段吗？
- Achievable – 目标可能吗？每月阻止 100，000 个客户不是一个合理的目标。
- Realistic – 确保目标是可能的。
- Time bound – 必须有时间达到目标。

评估当前情况

详细调查当前业务状况的所有方面。

确保每个人都接受你的发现是开始的起点。

确保每个参与者都了解标准。

确定数据目标

确定对机器学习过程的详细目标。始终与业务需求相关。

业务成功标准

描述能够实现业务目标的项目预期输出。

数据科学、数据工程和机器学习成功标准

用技术术语定义项目成功结果的标准。为此，解释以什么作为成功标准以及如何衡量成功标准。

生成项目计划

为所有数据处理目标以及这些目标如何帮助实现业务目标制定一个计划。你的计划应该规定在项目的其余部分要执行的步骤，包括机器学习工具的初始选择和新技术的开发。

数据理解

CRISP-DM 过程的第二阶段是确定数据和元数据的真实状态。了解来源（数据从何而来？）和血统（在使用之前对数据做了哪些处理？）。

浏览数据

这包括您将使用什么查询、可视化数据和做数据报告。这些可能包括：

关键属性的分布（例如，预测任务的目标属性）

两个或几个属性之间的关系

简单聚合的结果

有效子群的性质

简单的统计分析

这些分析必须有助于或改进数据描述和质量报告，并将其纳入进一步分析所需的转换和其他数据准备步骤中。

数据探索报告

Perform 结构化数据探索和检查所有数据子集。

验证数据质量

检查数据的质量，解决以下问题：

- 数据是否完整（是否涵盖所需的所有项目）？
- 基础数据是否正确，或者是否包含错误，如果存在错误，它们常见吗？
- 数据中是否有缺失的值？如果是，它们是如何表示的，它们在哪里发生，它们有多常见？

数据质量报告

生成数据质量验证。数据质量问题的解决方案通常在很大程度上取决于数据和业务知识。

数据准备

选择您的数据

确定用于分析的数据。

包容/排斥的理由

确定要包括或排除的数据以及这些决策的详细信息。

清理数据

通过清理数据质量，您将节省分析技术的下游处理，通过建模来估计缺失的数据。

数据清理报告

描述您为解决数据质量问题而做出的决策和行动。

功能工程

规划功能工程以提取从一个或多个现有属性构造的新属性

建 模

选择建模技术

作为建模的第一步，您将选择要使用的实际建模技术。尽管在业务理解阶段您可能已经选择了 工具，但在此阶段，您将选择特定的建模技术，例如使用 C5.0 构建决策树，或者使用回传神经网络。如果应用了多种技术，则用每种技术分别执行此任务。

测试设计

计划将样本数据集处理为培训、测试和验证数据集所需的操作。

生成模型

在准备好的数据集上运行模型以创建一个或多个模型。

参数设置

任何模型都需要可调整的特定参数。列出参数及其所选值，以及选择参数设置的基本原理。

模型

这些是建模工具生成的实际模型。

模型评估

总结所生产模型的结果（例如，在准确性方面），并按彼此的关系
对它们的质量进行排名。

修订的参数设置

计划模型评估中的改进，修改参数设置，并为下一次建模运行调整参数设置。反复进行模型构建和评估，直到您坚信自己找到了最好的模型。

评价

数据科学和机器学习结果评估

根据业务成功标准总结评估结果，包括关于项目是否已达到初始业务目标的最终声明。

认可的型号

在根据业务成功标准评估模型后，符合所选标准的生成模型将成为已认可的模型。

流程审查

总结流程审查，突出遗漏的活动和应重复的活动。

确定下一步

计划评估和流程审查后;您现在决定如何前进。

部署

总结部署策略，包括必要的步骤以及如何执行这些步骤。

监控和维护计划

总结监控和维护策略，包括必要的步骤以及如何执行这些步骤。

体验文档

总结项目期间取得的重要经验。

好;您已到达 CRISP-DM 方法的末尾。

你如何使用这个新知识？

现在，您已经拥有 CRISP-DM
的基本标准方法，我想向您介绍一个处理系统，我个人一直在使用过去 10
多年。它一直是两个理学硕士项目的核心研究课题和
我的博士学位使用新的研究过程的基本原则。

我的数据科学团队也使用它。

我还从数据科学主管的职位上教导客户并告知客户如何使用这个过程。

它涵盖在以下书籍：

- Practical Hive: A Guide to Hadoop’s Data Warehouse System by Scott Shaw, Andreas François Vermeulen, Ankur Gupta, and David Kjerrumgaard (Apress, 2016).

- Practical Data Science: A Guide to Building the Technology Stack for Turning Data Lakes into Business Assets by Andreas François Vermeulen (Apress, 2018).

快速信息工厂生态系统

信息工厂生态系统是我用于个人处理开发的技术惯例。本书的加工路线将在此基础上制定，但您并不一定要专门使用它。我在本章中讨论的工具对您来说是没有限制的。这些工具可以用于适合您特定生态系统的任何配置或排列。

我建议你开始建立一个你自己的生态系统，或者干脆采用我的生态系统。您已经具备了熟悉并能够熟练地部署一组工具的先决条件。

请记住，您的数据湖将有自己的属性和功能。因此，请采用您的工具，以处理您所工作的生态系统中的特定特征。

R-A-P-T-O-R 使用数据湖的数据科学过程

什么是 R - a - p - t - o - r？

这代表Retrieve-Assess-Process-Transform-Organize-Report。参见图 5-16。

图5-16。 R-A-P-T-O-R 图

Rapid Information Factory（RIF）的R-A-P-T-O-R引擎通过检索（导入外部数据源）、评估（数据质量）、过程（合并）、转换（真相的单一版本）、组织（特征工程）和报告（business insight）的数据管道来引导数据湖中的数据。

R-A-P-T-O-R 引擎（图 5-16）来回传送数据，因为它在数据湖中的不同区域之间分布。

Retrieve

将数据从源系统检索到数据工程环境中。 这是源系统中的数据。

Assess

评估检索中的数据，以确定任何数据质量问题，并修复数据以确保更好的数据。

Process

处理评估中的数据合并同类数据生成当前“单一版本的真相”

Transform

将当前“单一版本的真相”的快照转换为历史存储数据保险库。数据特征工程在这里生成数据科学辅助资料。

将“数据仓库”组织成一组数据集合，以便数据科学模型进行处理。生成数据仓库、数据集市、培训、测试和结果数据集。

将数据科学的结果报告为为标准报告工具所使用的每个数据科学案例预先批准的可交付成果。

什么是数据湖？

数据湖是存储大量原始数据的存储库。它以本地格式存储数据，以满足未来的需求。在本书中，您将了解到为什么这对实际数据科学和工程解决方案极其重要。写数据仓库模式将数据存储在预定义的数据库、表和记录结构中，而数据湖在基于读的架构上使用限制较少的模式来存储数据。数据湖中的每个数据元素都被分配了一个独特的标识符，并用一组全面的元数据标记进行了标记。

数据湖通常是使用分布式数据对象存储来部署的，以能够启用schema-on-read结构。这意味着业务分析和数据挖掘工具访问数据时不需要复杂的模式。使用schema-on-read方法支持按原样加载数据并开始从中即时获取值。

我将在第6章、第7章、第8章和第9章中讨论并提供更多关于使用读存储模式方法的原因的详细信息。

我们在云上的部署是一个经济高效的解决方案，使用 Amazon Simple Storage Service （Amazon S3） 来存储数据湖的基本数据。

我将演示您如何使用云技术来提供数据科学工作。这是这本书中不需要去云的例子，因为他们将很容易在个人笔记本电脑上处理。

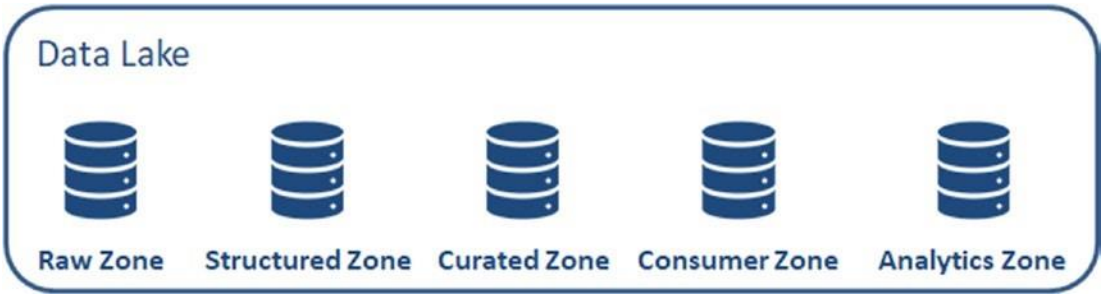

图5-17。 数据湖六个区

这种方法可以利用现有的技术堆栈，从而降低风险和成本。

数据湖区

参见图 5-17。

原始区域是数据湖数据之外的所有数据的入口点。它是几种数据提取处理解决方案的端点。

结构化区域用于将原始数据转换为增强的数据源。该区域的数据采用统一格式，以帮助下一个区域的处理能力。任何数据质量问题在该区域得到解决。

固化区域是数据湖中当前唯一真实的区域。此区域中的数据保管库和数据仓库支持从结构化区域合并数据源。"实时"数据科学模型的结果存储在此 区域中。

消费者区是存储业务见解的不同数据集市的区域，可供最终用户可视化其业务见解。这是大多数商务人士提问的主要区域。

提示

服务级别协议（SLA）只有在客户服务区的其他服务级别得到批准后，才有能力维护服务区。

分析区是数据湖的“沙盒”。在DevOps将其转移到策展专区之前，该专区用于设计、开发和培训新的和新颖的数据科学和机器学习解决方案。

什么是数据保管库？

由Dan Linstedt设计的数据保管库是一种特意构建的数据库建模方法，用于控制来自多个操作系统的数据的长期历史存储。

数据保险存储过程将读数据湖上的模式转换为写入时的模式。

数据仓库被设计成读查询请求模式，然后针对数据湖执行。

我还看到将结果存储为编写架构格式，以保留结果以供将来查询使用。

这两种技术的技术都在第8章中讨论。此时，我只希望您了解制定数据保管库所需的基本结构。

该结构由三个基本数据结构构建：集线器、链接和卫星。

让我们来看一下具体的数据结构，以了解它们为什么是强制性的。

枢纽

枢纽包含一个具有低更改倾向的唯一业务密钥列表。枢纽包含每个枢纽
项的代理项密钥和业务密钥来源的元数据分类。

集线器是数据保管库的核心骨干，我将在第9章中更详细地讨论如何以及为什么使用此结构。

链接

使用链接表对业务键之间的关联或事务进行建模。这些表本质上是多对多的联接表，具有特定的附加元数据。

链接是枢纽
之间的单一关系，以确保准确记录业务关系，从而完成实际业务的数据模型。

在第九章中，我将解释你是如何以及为什么需要特定关系。

卫星

枢纽
和链接构成了模型的结构，但不存储数据的时间特征或描述性特征。这些特征存储在适当的表中，这些表被称为卫星。

卫星是存储有关业务特征的综合级别信息的结构，通常是完整数据仓库数据结构中容量最大的。在第9章中，我将解释这些结构如何以及为什么能够很好地模拟实际业务特征。

枢
纽、链路和卫星的适当组合支持数据科学家构建和存储前提业务关系。作为一个数据建模者这是一个高度需求的技能。

在第9章中详细讨论了写入数据结构时向这种模式的转换，以指出为什么特定结构支持这种处理方法。

我将在第9 章解释 ...为什么您需要特定的枢纽，链接和卫星。

接下来做什么？

您已成功完成第 5章。

您现在已经具备了良好的工作知识，了解监督式机器学习过程。

您熟悉 CRISP-DM 方法，这可确保你的机器学习能力可以从实验和假设进步到IML。

作为机器学习顾问，我通过 RIF 分享了自己的知识和历程。

这一过程将在我们第15章全面工业化项目的中介绍，我们将在其中介绍技术的实际应用和RIF的框架

接下来，我将在第6章中介绍无监督的机器学习。