

Machine Learning Projects  
for .NET Developers

Apress®

# 机器学习 项目开发实战

[美] Mathias Brandewinder 著  
姚军 译

- 来自.NET专家的声音
- 从数据中学习，让应用更“聪明”

 中国工信出版集团

 人民邮电出版社  
POSTS & TELECOM PRESS

Machine Learning Projects  
for .NET Developers

# 机器学习 项目开发实战

[美] Mathias Brandewinder 著  
姚军 译

人民邮电出版社  
北京

## 图书在版编目 ( C I P ) 数据

机器学习项目开发实战 / (美) 马蒂亚斯·布兰德温  
德尔 (Mathias Brandewinder) 著 ; 姚军译. -- 北京 :  
人民邮电出版社, 2016. 8  
ISBN 978-7-115-42951-3

I. ①机… II. ①马… ②姚… III. ①机器学习  
IV. ①TP181

中国版本图书馆CIP数据核字(2016)第162289号

## 版 权 声 明

Machine Learning Projects for .NET Developers

By Mathias Brandewinder, ISBN: 978-1-4302-6767-6

Original English language edition published by Apress Media.

Copyright © 2015 by Apress Media.

Simplified Chinese-language edition copyright © 2016 by Post & Telecom Press.

All rights reserved.

本书中文简体字版由 Apress Media 授权人民邮电出版社独家出版。未经出版者书面许可, 不得以任何方式复制本书的任何内容。

版权所有, 侵权必究。

- 
- ◆ 著 [美] Mathias Brandewinder
  - 译 姚 军
  - 责任编辑 王峰松
  - 责任印制 焦志炜
  - ◆ 人民邮电出版社出版发行 北京市丰台区成寿寺路 11 号  
邮编 100164 电子邮件 315@ptpress.com.cn  
网址 <http://www.ptpress.com.cn>  
北京艺辉印刷有限公司印刷
  - ◆ 开本: 800×1000 1/16  
印张: 17.5  
字数: 382 千字 2016 年 8 月第 1 版  
印数: 1-2 500 册 2016 年 8 月北京第 1 次印刷
- 著作权合同登记号 图字: 01-2016-4639 号
- 

定价: 59.00 元

读者服务热线: (010)81055410 印装质量热线: (010)81055316  
反盗版热线: (010)81055315

# 内容提要

本书通过一系列有趣的实例，由浅入深地介绍了机器学习这一炙手可热的新领域，并且详细介绍了适合机器学习开发的 Microsoft F#语言和函数式编程，引领读者深入了解机器学习的基本概念、核心思想和常用算法。书中的例子既通俗易懂，同时又十分实用，可以作为许多开发问题的起点。通过对本书的阅读，读者无须接触枯燥的数学知识，便可快速上手，为日后的开发工作打下坚实的基础。本书适合对机器学习感兴趣的.NET 开发人员阅读，也适合其他读者作为机器学习的入门参考书。

# 关于作者

**Mathias Brandewinder** 是 Microsoft F# 最有价值专家 (MVP)，住在加州旧金山，在那里他为 Clear Lines Consulting 工作。作为一名当之无愧的数学极客，他很早就对构建模型帮助其他人利用数据做出更好的决策感兴趣。他拥有商业、经济和运营研究等多个硕士学位，在到达硅谷之后不久便爱上了编程。从 .NET 刚出现时开始，他就专业开发软件，为各行各业开发业务应用程序，重点是预测模型和风险分析程序。

# 关于译者

姚军，曾在多家券商任 IT 经理，在系统集成、数据库、网络系统方面有近 20 年经验，主导及参与了多个大型系统集成项目的需求分析、实施及维护。由于工作原因，在计算机领域涉猎极广，自 2006 年开始，工作之余将大量精力投入 IT 图书的翻译及编著工作，曾参与“全国网络技术水平考试丛书”的编写工作，译作甚丰，如《Python 金融大数据分析》《数据驱动的网络分析》《软件架构师的 12 项修炼：技术技能篇》《增量承诺螺旋模型：系统和软件开发的成功之道》《VMware vCAT 权威指南：成功构建云环境的核心技术和方法》等。

# 关于技术评审

**Scott Wlaschin** 是一位 .NET 开发人员、架构师和作家。他在不同的领域有 20 多年的经验，从高级 UX/UI 到低级数据库实现均有涉猎。他曾经用多种语言编写重要的代码，其中最喜欢的是 Smalltalk、Python 和更新的 F#，他在 [fsharpforfunandprofit.com](http://fsharpforfunandprofit.com) 上发表关于 F# 的博客。

# 致谢

感谢我的父母，我在充满书香的家庭中长大，书籍对今天的我产生了深远的影响。我对它们的热爱是从事这个疯狂项目的部分原因，尽管许多人警告我，这一旅程充满艰辛，我仍努力地尝试写一本自己的书。旅途的艰辛是值得的，对此我满怀骄傲：我也能写一本书了！为此（还有许多其他方面），我必须感谢父母。

孤独的旅程毫无趣味，我很幸运有 3 位出色的同行者：无畏的 Gwenan、智慧的 Scott 和坚定的 Petar。Gwenan Spearing 和 Scott Wlaschin 认真地审核手稿，为我提供了宝贵的反馈意见，保证这个项目处于正确的方向。有了他们，最终的结果大不相同。书中最好的部分源自他们的贡献，而读者找到的任何问题都应归咎于我！我要衷心地感谢 Petar Vucetin，和他成为商业合作伙伴和朋友是一种幸运。他在我情绪不佳的时候首当其冲，在这种时候仍然鼓励我，为我提供完成任务所需的时间和空间。感谢你，伙计！——你是真正的朋友。

本书写作期间还有许多人帮助过我，无法一一列举。感谢帮助我实现本书，提供代码、建议或者溢美之辞的人们——你们知道我指的是谁！特别要感谢 F#社区。这是一个坦率的社区（显然有时候这令人有些烦恼），但更重要的是，认识社区的这么多人是快乐和灵感的源泉。请保持这种令人敬畏的状态！

当然，没有燃料就走不了长路。这段特别旅程的动力是咖啡因，在我看来，旧金山的咖啡馆有最好的玛琪雅朵咖啡，让我在过去的一年半中每天都有好的开始。



# 前言

如果你手里拿着这本书，我就可以认定你是对机器学习感兴趣的.NET 开发人员了。你可能对编写 C#应用程序很熟悉，开发的很有可能是业务线应用程序。以前你可能遇到过 F#，也可能没有。而且，你很有可能对机器学习感到好奇。这一主题每天都见诸报端，因为它和软件工程有着很紧密的联系，但是使用的是不熟悉、看似有些抽象的数学概念。简而言之，机器学习看上去是有趣的主题、值得学习的实用技能，但是从哪里入手难以说清。

本书的意图是作为开发人员的机器学习入门书。我的主要目标是使熟悉代码编写的读者（而不是数学家）容易理解书中的主题。对数学的喜爱当然没有坏处，但是本书通过实用的示例学习核心概念，说明其中的工作原理。

什么是机器学习？机器学习是一种编程艺术，所编写的计算机程序随着可用数据越来越多而更好地执行任务，无须开发人员更改代码。

上述定义相当宽泛，反映了机器学习广泛适用于各个领域这一事实。但是，该定义中的一些具体特征值得更详细说明。机器学习是关于程序编写的学科，这些代码运行于生产环境并执行某项任务，这使它不同于统计学。机器学习是一个跨学科领域，这个主题既和倾向于数学的研究人员相关，也和软件工程师相关。

定义中另一个有趣的部分是数据。机器学习是关于利用可用数据解决实际问题的学科。使用数据是机器学习的关键部分，理解数据、研究如何从中提取有用信息，往往比使用的特定算法更重要。因此，我们将从数据开始了解机器学习。每章都从一个真实的数据集和所要解决的特定问题开始，数据中包含了现实世界中的所有不完善和意外。由此，我们将在这一背景下从头开始构建问题解决方案，在需要的时候介绍思路。在此过程中，我们将创建一个基础，帮助你理解不同思路的组合使用，使你在以后需要的时候更有效率地使用库或者框架。

我们的探索从熟悉的 C#和 Visual Studio 开始，但是在取得进展之后将介绍 F#，这是一种特别适合于机器学习问题的.NET 语言。正如机器学习，函数式编程一开始令人生畏。然而，一旦掌握了诀窍，F#就会变得很简单且极具效率。如果你完全是 F#的初学者，本书将告诉你该语言所需了解的一切，你将在现实、有趣的问题中学习如何高效地使用该语言。

学习过程中，我们将探索各种各样的问题，帮助你理解机器学习能使应用程序变得更好的领域，有些方法可能出人意料。我们将探索图像识别、垃圾邮件过滤器和自我学习游戏以及其他一些问题。而且，在我们共同的旅途上，你将发现机器学习并没有那么复杂，相当简单的模型就能产生令人惊讶的出色结果。最后，你将会发现，机器学习非常有趣！好了，不多啰嗦了，让我们一起对付第一个机器学习问题吧！

# 目录

■ 第1章 256级灰度	1
1.1 什么是机器学习	2
1.2 经典的机器学习问题：图像分类	3
1.2.1 挑战：构建一个数字识别程序	3
1.2.2 机器学习中的距离函数	5
1.2.3 从简单的方法入手	5
1.3 我们的第一个模型（C#版本）	6
1.3.1 数据集组织	6
1.3.2 读取数据	7
1.3.3 计算图像之间的距离	9
1.3.4 编写分类器	11
1.4 那么，如何知道程序有效？	12
1.4.1 交叉验证	12
1.4.2 评估模型质量	13
1.4.3 改进模型	14
1.5 介绍用于机器学习的F#	15
1.5.1 使用F#交互执行进行实时脚本编写和数据研究	15
1.5.2 创建第一个F#脚本	18
1.5.3 剖析第一个F#脚本	19
1.5.4 创建函数管道	22
1.5.5 用元组和模式匹配操纵数据	23
1.5.6 训练和评估分类器函数	24
1.6 改进我们的模型	26
1.6.1 试验距离的另一种定义	26
1.6.2 重构距离函数	27
1.7 我们学到了什么	30

1.7.1	在好的距离函数中能找到什么	30
1.7.2	模型不一定要很复杂	31
1.7.3	为什么使用 F#?	31
1.8	更进一步	32
■ 第 2 章	垃圾邮件还是非垃圾邮件?	33
2.1	挑战: 构建一个垃圾邮件检测引擎	34
2.1.1	了解我们的数据集	34
2.1.2	使用可区分联合建立标签模型	35
2.1.3	读取数据集	36
2.2	根据一个单词决定	38
2.2.1	以单词作为线索	38
2.2.2	用一个数字表示我们的确定程度	39
2.2.3	贝叶斯定理	40
2.2.4	处理罕见的单词	42
2.3	组合多个单词	42
2.3.1	将文本分解为标记	42
2.3.2	简单组合得分	43
2.3.3	简化的文档得分	44
2.4	实现分类器	45
2.4.1	将代码提取到模块中	46
2.4.2	文档评分与分类	47
2.4.3	集合和序列简介	49
2.4.4	从文档语料库中学习	51
2.5	训练第一个分类器	53
2.5.1	实现第一个标记化程序	54
2.5.2	交互式验证设计	54
2.5.3	用交叉验证确立基准	55
2.6	改进分类器	56
2.6.1	使用每个单词	56
2.6.2	大小写是否重要?	57
2.6.3	简单就是美	58
2.6.4	仔细选择单词	59
2.6.5	创建新特征	61
2.6.6	处理数字值	63
2.7	理解分类错误	64

2.8 我们学到了什么? .....	66
■ 第3章 类型提供程序的快乐 .....	67
3.1 探索 StackOverflow 数据 .....	68
3.1.1 StackExchange API .....	68
3.1.2 使用 JSON 类型提供程序 .....	70
3.1.3 构建查询问题的最小化 DSL .....	73
3.2 世界上的所有数据 .....	76
3.2.1 世界银行类型提供程序 .....	76
3.2.2 R 类型提供程序 .....	77
3.2.3 分析数据与 R 数据框架 .....	81
3.2.4 .NET 数据框架 Deedle .....	83
3.2.5 全世界的数据统一起来! .....	84
3.3 我们学到了什么? .....	88
■ 第4章 自行车与人 .....	91
4.1 了解数据 .....	92
4.1.1 数据集有哪些内容? .....	92
4.1.2 用 FSharp.Charting 检查数据 .....	93
4.1.3 用移动平均数发现趋势 .....	94
4.2 为数据适配模型 .....	96
4.2.1 定义简单直线模型 .....	96
4.2.2 寻找最低代价模型 .....	97
4.2.3 用梯度下降找出函数的最小值 .....	98
4.2.4 使用梯度下降进行曲线拟合 .....	99
4.2.5 更通用的模型公式 .....	100
4.3 实施梯度下降的方法 .....	101
4.3.1 随机梯度下降 .....	101
4.3.2 分析模型改进 .....	103
4.3.3 批量梯度下降 .....	105
4.4 拯救者——线性代数 .....	107
4.4.1 宝贝, 我缩短了公式! .....	108
4.4.2 用 Math.NET 进行线性代数运算 .....	109
4.4.3 标准形式 .....	110
4.4.4 利用 MKL 开足马力 .....	111
4.5 快速演化和验证模型 .....	112

4.5.1	交叉验证和过度拟合	112
4.5.2	简化模型的创建	113
4.5.3	在模型中添加连续特征	115
4.6	用更多特征改进预测	117
4.6.1	处理分类特征	117
4.6.2	非线性特征	119
4.6.3	正规化	122
4.7	我们学到了什么?	123
4.7.1	用梯度下降最大限度地减小代价	123
4.7.2	用回归方法预测数字	124
■ 第5章	你不是独一无二的雪花	125
5.1	发现数据中的模式	126
5.2	我们所面临的挑战: 理解 StackOverflow 上的主题	128
5.3	用 K-均值聚类方法找出聚类	132
5.3.1	改进聚类和质心	133
5.3.2	实施 K-均值聚类方法	135
5.4	StackOverflow 标签的归类	138
5.4.1	运行聚类分析	138
5.4.2	结果分析	139
5.5	好的聚类和坏的聚类	141
5.6	重新标度数据集以改进聚类	144
5.7	确定需要搜索的聚类数量	147
5.7.1	什么是“好”的聚类?	147
5.7.2	确定 StackOverflow 数据集的 $k$ 值	148
5.7.3	最终的聚类	150
5.8	发现特征的相关性	151
5.8.1	协方差和相关系数	151
5.8.2	StackOverflow 标签之间的相关性	153
5.9	用主成分分析确定更好的特征	154
5.9.1	用代数方法重新组合特征	155
5.9.2	PCA 工作方式预览	156
5.9.3	实现 PCA	158
5.9.4	对 StackOverflow 数据集应用 PCA	159
5.9.5	分析提取的特征	160
5.10	提出建议	165

5.10.1 简单标签推荐系统	165
5.10.2 实现推荐系统	166
5.10.3 验证做出的推荐	168
5.11 我们学到了什么?	170
■ 第6章 树与森林	171
6.1 我们所面临的挑战：“泰坦尼克”上的生死存亡	171
6.1.1 了解数据集	172
6.1.2 观察各个特征	173
6.1.3 构造决策桩	174
6.1.4 训练决策桩	176
6.2 不适合的特征	177
6.2.1 数值该如何处理?	177
6.2.2 缺失数据怎么办?	178
6.3 计量数据中的信息	180
6.3.1 用熵计量不确定性	180
6.3.2 信息增益	182
6.3.3 实现最佳特征识别	184
6.3.4 使用熵离散化数值型特征	186
6.4 从数据中培育一棵决策树	187
6.4.1 建立树的模型	187
6.4.2 构建决策树	189
6.4.3 更漂亮的树	191
6.5 改进决策树	192
6.5.1 为什么会过度拟合?	193
6.5.2 用过滤器限制过度的自信	194
6.6 从树到森林	195
6.6.1 用k-折方法进行更深入的交叉验证	196
6.6.2 将脆弱的树组合成健壮的森林	198
6.6.3 实现缺失的部分	199
6.6.4 发展一个森林	200
6.6.5 尝试森林	201
6.7 我们学到了什么?	202
■ 第7章 一个奇怪的游戏	205
7.1 构建一个简单的游戏	206

## ■ 目录

7.1.1 游戏元素建模	206
7.1.2 游戏逻辑建模	207
7.1.3 以控制台应用的形式运行游戏	209
7.1.4 游戏显示	211
7.2 构建一个粗糙的“大脑”	213
7.2.1 决策过程建模	214
7.2.2 从经验中学习制胜策略	215
7.2.3 实现“大脑”	216
7.2.4 测试“大脑”	218
7.3 我们能更高效地学习吗?	221
7.3.1 探索与利用的对比	221
7.3.2 红色的门和蓝色的门是否不同?	222
7.3.3 贪婪与规划的对比	223
7.4 无限的瓷砖组成的世界	224
7.5 实现“大脑”2.0	227
7.5.1 简化游戏世界	227
7.5.2 预先规划	228
7.5.3 $\epsilon$ -学习	229
7.6 我们学到了什么?	231
7.6.1 符合直觉的简单模型	231
7.6.2 自适应机制	232

## ■ 第8章 重回数字

8.1 调整代码	233
8.1.1 寻求的目标	234
8.1.2 调整距离函数	235
8.1.3 使用 Array.Parallel	239
8.2 使用 Accord.NET 实现不同的分类器	240
8.2.1 逻辑回归	241
8.2.2 用 Accord 实现简单逻辑回归	242
8.2.3 一对一、一对多分类	244
8.2.4 支持向量机	246
8.2.5 神经网络	248
8.2.6 用 Accord 创建和训练一个神经网络	250
8.3 用 m-brace.net 实现伸缩性	253
8.3.1 用 Brisk 启动 Azure 上的 MBrace	253

8.3.2 用 MBrace 处理大数据集 .....	256
8.4 我们学到了什么? .....	259
■ 第9章 结语 .....	261
9.1 描绘我们的旅程 .....	261
9.2 科学! .....	262
9.3 F#: 函数式风格更有效率 .....	263
9.4 下一步是什么? .....	264



## 第 1 章



# 256 级灰度

## 构建自动识别数字图像的程序

如果你打算建立一个当前技术热点的列表，机器学习当然会名列前茅。然而，虽然这个术语到处出现，但是它的真实含义往往含混不清。它是和“大数据”或者“数据科学”一样的东西吗？它和统计学有何不同之处？表面上，机器学习似乎是一种奇特、令人畏惧的专业，使用令人眼花缭乱的数学知识和算法，和软件工程师的日常活动没有多少共同之处。

在本章以及本书余下的部分中，我的目标是和大家一起完成实际项目，以此阐明机器学习的原理。我们将循序渐进地解决问题，主要是从头开始编写代码。通过讲述这种方法，我们可以理解工作原理的细节，逐步说明广泛适用的核心思路和方法，并帮助你为以后构建专用程序库打下坚实的基础。在第 1 章中，我们将深入探讨一个经典问题——手写数字识别，同时完成以下几件工作：

- 建立适用于大部分机器学习问题的方法论。机器学习模型的开发与标准业务线应用程序有微妙的不同，将带来特殊的挑战。学到本章的最后，你将会理解交叉验证的概念、重要性以及使用方法。
- 帮助你理解如何“考虑机器学习”，以及如何看待机器学习问题。我们将讨论相似性和距离之类的思路，这些思路是大部分算法的核心。我们还将说明，虽然数学是机器学习的重要组成部分，但是这个方面可能被过分强调了，有些核心思路实际上相当简单。我们将从比较简单的算法开始，你会看到，这些算法实际上工作得很好！
- 了解如何用 C# 和 F# 解决问题。我们将从实现 C# 解决方案开始，然后提供 F# 的等价解决方案。F# 是一种特别适合于机器学习和数据科学的 .NET 语言。

在第 1 章就碰上这样的问题，似乎会令人畏缩——但是不要被吓住！从表面上看这个问题很难，但是你会发现，我们仅用相当简单的方法，就能够创建相当有效的解决方案。再说，解决小儿科的问题有什么意思？