人工智能在信用债投资领域的应用

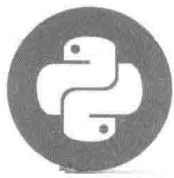
Python语言实践

崔玉征◎著

清華大學出版社

人工智能在信用债投资领域的应用

Python语言实践



崔玉征◎著

清華大學出版社
北京

内 容 简 介

本书共分三部分，第一部分主要讲述机器学习、深度学习和人工智能的基本方法，并给出了使用基于 TensorFlow 后台的 Keras 库做深度学习的实践案例；第二部分主要讲述做信用债投资面临的困难，并给出了实用的解决方案；第三部分主要讲述解决做信用债投资的困难的实用方法，并给出了全部的 Python 源代码。

本书适合在银行、证券、保险、基金等金融机构从事对公信贷和债券投资等工作的相关从业者阅读。

图书在版编目(CIP)数据

人工智能在信用债投资领域的应用：Python 语言实践/崔玉征著. —北京：清华大学出版社，2019

ISBN 978-7-302-51305-6

Ⅰ. ①人… Ⅱ. ①崔… Ⅲ. ①人工智能－应用－债券投资－研究 Ⅳ. ①F830.59-39

中国版本图书馆 CIP 数据核字(2018)第 234146 号

责任编辑：张　伟
封面设计：李召霞
责任校对：宋玉莲
责任印制：刘海龙

出版发行：清华大学出版社
网　　址：http://www.tup.com.cn，http://www.wqbook.com
地　　址：北京清华大学学研大厦 A 座　　邮　　编：100084
社 总 机：010-62770175　　邮　　购：010-62786544
投稿与读者服务：010-62776969，c-service@tup.tsinghua.edu.cn
质量反馈：010-62772015，zhiliang@tup.tsinghua.edu.cn

印 装 者：三河市国英印务有限公司
经　　销：全国新华书店
开　　本：170mm×240mm　　**印　　张**：14.75　　**字　　数**：287 千字
版　　次：2019 年 1 月第 1 版　　**印　　次**：2019 年 1 月第1次印刷
定　　价：69.00 元

产品编号：081058-01

序

我的学生崔玉征先生，拥有中科院模式识别（人工智能）和香港中文大学金融 MBA 两个专业的硕士学位，是典型的复合型人才。他一直致力于通过机器学习等人工智能方法，解决国内资本市场信用债投资时所面临的外部评级虚高、无法真实有效地反映融资主体的信用状况、无法用于融资成本定价等行业痛点。

中国债券市场，自从 2014 年 3 月 5 日第一只信用债“超日债”违约以来，截至现在已经有 228 只信用债发生违约，且违约事件的发生有明显加速的迹象。随着中国经济进入中低速、但高质量增长的新常态，及未来很长一段时间我国可能面临的严峻外部经济环境，可以预判中国债券市场违约事件会频发，违约进入新常态。

美国著名经济学家、诺贝尔经济学奖得主保罗·克鲁格曼教授，提出了著名的“三元悖论”，即开放经济下的政策选择只能同时满足：本国货币政策的独立性、汇率的稳定性和资本的完全流动性中的两个，而无法同时满足全部三个。改革开放 40 年来，中国的宏观经济政策在货币政策相对独立、人民币汇率相对稳定的前提下，主要依靠人口红利、制度红利和资源红利等的不断释放，实现了快速的跨越式发展，发展中出现的问题被高速的经济增长所掩盖。但是，随着这些红利的不断消失，之前不断积累的问题也都已经慢慢地凸显出来了。在金融领域表现得最突出的问题是金融稳定、渐进性改革、定价体系缺失造成的套利机会并存，这三个矛盾构成了中国金融政策的“三元悖论”。也就是说，为了维护中国的金融稳定，政府机构选择了“修修补补”式的渐进性改革，而这种形式的渐进性改革并没有完全遵守金融具有全局性属性的基本原理，这就必然会对定价体系造成扭曲，从而产生大量的套利机会。大量套利机会的存在必然影响中国的金融稳定。因此，这是中国金融政策的“不可能三角”。

问题的存在一直以来都是我们发展的动力，中国经济要想实现高质量的健康发展，需要不断化解中国金融政策面临的“不可能三角”。伴随着人工智能、大数据和移动互联网等技术的不断发展，金融科技不断推动着传统金融业的变革，我们逐渐地看到了解决中国金融政策“不可能三角”的突破口。那就是，通过逐渐建立各层次、各类别资本市场的定价体系，来逐渐消除套利机会

并建设稳定的金融体系。目前，债券市场已经初步具备了建设合理定价体系的条件，这主要是因为随着债券市场违约事件的不断增多，信用利差在不断扩大，尤其是高收益债券的信用利差出现明显扩大的趋势，这就为债券发行主体的定价提供了有利的前提条件。债券市场合理定价的前提是有一个市场化的信用评级体系，该体系可真实有效地反映债券发行主体信用风险的高低。

债券市场现有的外部评级体系，由于监管、财报粉饰较普遍等原因，表现出了明显的“虚高”和评级调整严重滞后的弊端。我认为本书提出的基于场外实时数据并采用机器学习技术的量化评级方法可较好地解决外部评级的弊端。这些场外数据主要包括司法、招聘、股权出质、动产抵押、高管变动、对外投资、实际控制人风险等，完全可以从非财务的侧面反映一家公司真实的经营状况，剔除了由于财务粉饰给我们造成的噪声干扰，仅保留能够真实反映企业还款能力的信号。这在国内资本市场的信用评级领域是一个较大的创新。我期待着这种量化研究方法逐渐得到业内同仁的认可。

何　佳

原证监会规划发展委员会委员、深交所综合研究所所长，南方科技大学教授

2018年9月25日

前言

笔者硕士毕业于中国科学院自动化研究所，专业是模式识别与智能系统。因此，人工智能是笔者的专业！ 10 年前笔者在自动化所读书时，人工智能这个专业远没有像今天这样火爆。 随着 AlphaGo 在人机博弈领域的围棋对弈中不断取得辉煌的成绩，人工智能再次走向辉煌，几乎被很多媒体描述成“无所不能”。

其实，人工智能有强弱之分，即强人工智能和弱人工智能。 强人工智能通常表现为在多个领域可用计算机来代替人类或者比人类做得更好，而弱人工智能通常是指在某一特定领域计算机比人类做得更好，如棋牌类游戏、量化投资等领域。 目前，强人工智能还只是幻想，弱人工智能在某些特定领域已经取得了惊人的成绩。 那么，人工智能可应用于哪些领域呢？ 通常地讲，能产生大量数据且这些数据可实现自动标注的领域，均适用于人工智能。 例如金融行业，它每天都产生大量的数据，且这些数据都可实现自动标注，因为股票要么上涨、要么下跌，债券要么正常兑付、要么违约。 因此，无场景不 AI（人工智能）。

深度学习技术的突破是这次人工智能浪潮的巨大推动力之一。 深度学习的思想跟传统机器学习相比，是一个颠覆性创新，在思维方式上是完全不同的。传统机器学习一般是通过大量数据寻找因果关系，而深度学习一般是通过分层特征提取并通过激活函数寻找关联关系，这正是大数据方法的思维方式。

本书重点介绍的是人工智能方法在信用债投资领域应用的实践方法，提供了大量的 Python 源代码，可供信用债投资者快速建立自己的信用债量化投资信用风险分析体系。 随着国内债券市场的快速发展，目前存量债券已经超过 35 000 只，如此多的投资标的，如果还是采用传统的、基于人工的主观判断和信息筛选，这个工作量是非常巨大的。 自从 2014 年 3 月 5 日国内第一只债券“超日债”违约以来，截至现在已经有 228 只债券发生违约，信用风险越来越受到投资者的重视，建设自己的、能够为投资决策提供有效支持的信用风险计量和管理体系，对广大金融机构来说已经非常迫切。

但是，建设科学、实用的信用债投资分析体系，在国内资本市场的现状下面临很多无法解决的客观困难。 例如，发债主体信息披露不及时、财务粉饰现象普遍存在、违约状态跟盈利状况正相关、违约样本极少，造成样本极不均衡、外部评级过度集中且区分能力很差等现状，作为普通的信用债投资者我们均无法

改变这些现状。我们只能从优化模型架构设计、获取可提供稳定场外数据源且实用的指标等角度入手，来建立自己的信用债投资量化风险分析体系。

本书介绍的信用债投资量化风险分析体系是笔者近10年来经验的总结和升华，书中详细介绍了通过模型架构优化来缓释财报粉饰的模型开发方法，介绍了通过场外真实数据来开发模型的技术，还简单介绍了深度学习技术在信用债投资量化风险分析体系中的应用方法。这些方法既是笔者多年经验的结晶，也是市面上其他同类教材中从来没有介绍过的实用技术总结。

由于财务粉饰现象的普遍存在，基于财务数据的企业信用风险评估经常出现误判，无法挖掘企业真正的投资价值。本书介绍的全部基于企业真实定性数据的信用风险评估方法已经得到了笔者团队的充分验证，效果是非常显著的。这种方法体系和实践方法在国内首屈一指，笔者坚信会给信用债投资分析者很大的启发。

本书共分为八个章节，其中第1章、第2章重点介绍人工智能、机器学习、深度学习的基本理论和方法，也给出了国内资本市场最常见的类别不均衡问题的解决方案；第3章重点介绍基于TensorFlow后台的Keras深度学习架构，并给出实用案例；第4章重点介绍国内债券市场的发展历程和现状；第5章重点介绍国内信用债投资分析面临的困难，并给出了详细的解决方案；第6~8章是本书的开源技术部分，详细讲述了大量的核心技术，包括自动抓取数据、对全市场财务数据的统计检验和分析、对司法等场外数据的统计检验和分析、财务粉饰的本福特统计法则和评分实施方法、用有监督机器学习开发评级模型的方法、用深度学习技术开发评级模型的方法、缓释财报粉饰的评分卡模型架构设计和评分卡模型开发核心代码等，这三章是本书开源技术的核心部分。

在本书的写作过程中，得到了家人、朋友和团队成员的大力支持，没有他们的支持和帮助，笔者不可能心无旁骛地构思本书的写作思路，在此对他们的贡献表示真诚的感谢。本书的前五章和附录部分由笔者完成，第6~8章中的部分Python源代码由团队成员刘志兴等完成初稿，并由笔者完成测试和剩余的逻辑分析部分。

读者若有问题与作者交流，请发邮件至yuzheng.cui@qq.com。扫一扫以下二维码，可获得本书的源代码。

崔玉征

2018年5月8日

目录

第1章 人工智能概述

近年来，人工智能再次成为炙手可热的领域。这次人工智能的蓬勃发展，比以往历次爆发都有更加深远的影响，也产生了令人兴奋不已的结果。但并不是所有的领域都适用于人工智能。概况来说，领域界限清晰、有海量数据、数据可自动标注，能够同时满足这三个条件的细分行业，均适用于人工智能。

例如，人机博弈领域中的棋牌类游戏。以围棋为例，在19×19的棋盘中，白子可用"1"表示，黑子可用"0"表示，再不受其他外界因素的影响，界限清晰，场景也非常简单；每次棋局理论上可产生约10的172次方的棋谱，数据量非常巨大；每次棋局均可产生赢、输、平三类结果中的一个，即目标可自动标注。因此，我们看到了AlphaGo战胜人类围棋顶级大师的惊艳战绩。

再如，在信用债投资领域，同样满足适用人工智能的三个条件。信用债投资重点关注的是企业发行的债券能否按时还本付息，即重点关注的是企业的偿债能力，这个界限非常清晰；目前，资本市场存量债券有30 000多只，且保持每年以10%以上的速度增长。每家企业均有大量的定量数据和定性数据，从这些数据中可反映企业的还款能力。显然，满足海量数据的要求；每只债券到期时，均可分为正常和违约两类，即这些海量数据是可自动标注的。因此，信用债投资分析领域也适用于人工智能。

除此之外，还有非常多的细分领域适用于人工智能。本章重点介绍人工智能的基本概念、原理和实现人工智能的常用技术——机器学习的基本原理。在信用债投资领域中，实现人工智能的相关方法将会在本书的后面章节中详细介绍。

1.1 图灵测试

理解人工智能，必须首先了解什么是图灵测试。"图灵测试"一词来源于计算机科学和密码学的先驱阿兰·麦席森·图灵写于1950年的论文《计算机器与智能》。在这篇论文中，图灵提出了一个用于判断计算机(或程序)是否具有智能的实验，如果计算机(或程序)通过了这个实验，则可以认为这个计算机或程序具备了智能。

图灵测试的基本内容是，如果计算机能在5分钟内回答由人类测试者提出

的一系列问题，且不能被辨别出其机器身份，则该计算机通过测试，说明该计算机具备智能。图灵测试的基本原型，如图 1.1 所示。

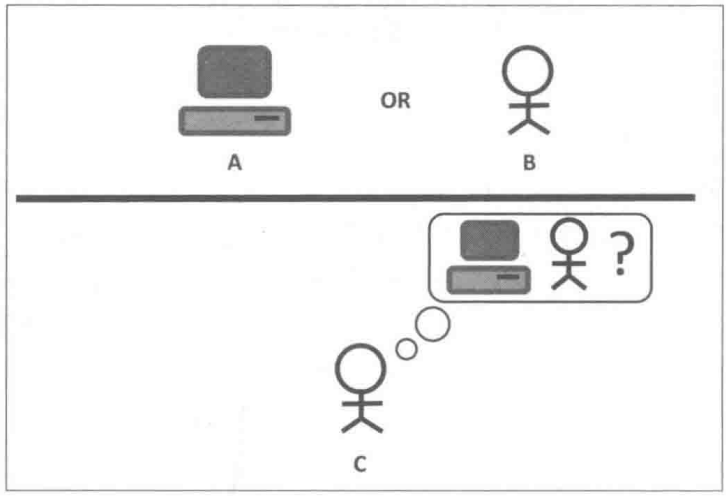


图 1.1　图灵测试的基本原型

在图 1.1 所示的图灵测试的基本原型中，C 代表人类，他通过一定的媒介向被测试者 A 或 B 提问一系列的问题，通过测试者对这些问题的回答，如果 C 无法辨别回答者是人还是机器，则说明被测试者具有智能。

图灵还为这项测试拟定了几个示范性问题，如下所示。

问：请给我写出有关“第四号桥”主题的十四行诗。

答：不要问我这道题，我从来不会写诗。

问：34 957 加 70 764 等于多少？

答：(停 30 秒后)105 721。

问：你会下国际象棋吗？

答：是的。

问：我在 K1 处有棋子 K；你在 K6 处有棋子 K，在 R1 处有棋子 R。轮到你走，你应该下哪步棋？

答：(停 15 秒钟后)棋子 R 走到 R8 处，将军！

由图灵测试的基本原型可知，在某一特定细分领域，即提问者 C 提出的问题仅限于某一特定的狭窄领域，此时通过大量的数据测试和模型训练，是不难通过图灵测试的。这种应用于细分领域的人工智能通常被称为弱人工智能。红极一时的 AlphaGo 就属于弱人工智能的典型应用，也就是说 AlphaGo 只能用于下围棋，不能直接用于其他领域。如果想将 AlphaGo 应用于其他领域，需要用该领域的数据做大量训练后，才有可能适用。

如果提问者 C 提出的问题，不预设场景、可随意提问，被测试者仍能通过图灵测试，则这时通常被称为强人工智能。显然，强人工智能基本是不可能的。因

为这需要机器存储人类有史以来的所有数据，并训练出人类所有可能的知识和推理，这是根本不可能完成的工作。

可见，人工智能是机器智能，不是人类智能，也不是类人智能。机器智能是必须通过提取大量数据，并进行自动标注后计算得到的智能。而除此之外，人类智能还包括不可计算的智能，如情感、直觉、感知等。

1.2 人工智能、机器学习和深度学习

由图灵测试的基本原型可知，人工智能不是一项技术，而是一种技术结果的最终展现。即一系列技术的成果通过了图灵测试，则认为该技术成果具备了智能。实现人工智能有很多种方法，包括规划调度、专家系统、多代理系统、进化计算、机器学习、知识表述、推荐系统等，如图 1.2 所示。

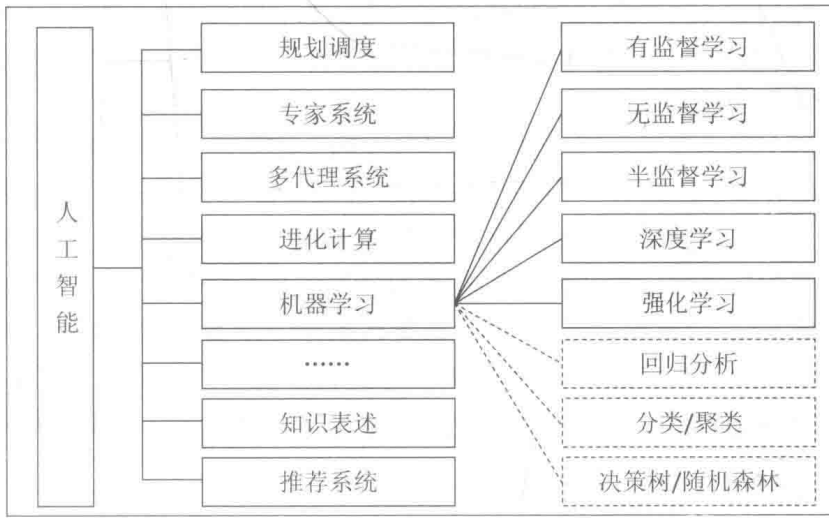


图 1.2 部分人工智能研究分支和研究方法

目前，用机器学习技术实现人工智能的方法产生了令人兴奋的结果。机器学习是一大类技术的统称，又可分为有监督机器学习技术（需要标注数据）和无监督机器学习技术（不需要标注数据），而深度学习又是有监督机器学习技术中的一个重要研究分支。人工智能、机器学习和深度学习的区别与联系如下所示。

1. 机器学习：一种实现人工智能的方法

机器学习最基本的做法，是使用算法来解析数据、从数据中按照一定的规则训练出规律，然后对真实世界中的事件作出决策和预测。与传统的为解决特定任务、硬编码的软件程序不同，机器学习是用大量的数据来“训练”，通过各种算法从数据中学习如何完成任务。举个简单的例子，当我们浏览网站时，经常会出

现电子商务网站广告的商品推荐信息。这是电子商务网站根据你以往的购物、点击和浏览记录，识别出这其中哪些是你真正感兴趣，并且愿意购买的产品，并推荐给你。这样的决策模型，可以帮助电子商务网站为客户提供建议并鼓励购买产品。机器学习直接来源于早期的人工智能领域，传统的算法包括决策树、K均值聚类、贝叶斯分类、支持向量机等。从学习方法上来分，机器学习算法可以分为有监督学习（如分类问题）、无监督学习（如聚类问题）、半监督学习、集成学习、深度学习和强化学习等。传统的机器学习算法在指纹识别、人脸检测、语音识别等领域的应用基本达到了商业化的要求或者特定场景的商业化水平，但每前进一步都异常艰难，直到深度学习算法的出现。

2. 深度学习：一种实现机器学习的技术

深度学习本来并不是一种独立的机器学习方法，其本身也会用到有监督和无监督的机器学习技术来训练深度神经网络模型。但由于近几年该领域发展迅猛，一些特有的学习手段相继被提出（如残差网络），因此越来越多的人将其单独看作一种机器学习的方法。

最初的深度学习是利用深度神经网络来解决特征表达的一种学习过程。深度神经网络本身并不是一个全新的概念，可大致理解为包含多个隐含层的神经网络结构。为了增强深层神经网络的训练效果，人们对神经元的连接方法和激活函数等方面作出相应的调整。其实有不少想法早年间也曾有过，但由于当时训练数据量不足、计算能力落后，因此最终的效果不尽如人意。目前，深度学习令人惊讶地实现了很多任务，使得似乎所有的机器辅助功能都变为可能，如无人驾驶、预防性医疗保健，甚至是更好的电影推荐，都近在眼前，或者即将实现。

3. 三者的区别和联系

机器学习是一种实现人工智能的方法，深度学习是一种实现机器学习的技术。我们就用最简单的方法，可视化地展现出它们的关系，如图1.3所示。

当下深度学习在计算机视觉、自然语言处理领域的应用远超过传统的机器学习方法，并且媒体对深度学习进行了大肆夸大的报道。因此，目前业界有一种错误的、较为普遍的认识，那就是“深度学习最终可能会淘汰掉其他所有机器学习算法”。深度学习，作为目前最热门的机器学习方法，但并不是机器学习的终点。其主要存在以下问题。

（1）深度学习模型需要大量的训练数据，才能表现出良好的效果，但现实生活中往往会遇到小样本问题，此时深度学习方法表现差强人意，传统的机器学习方法就可以很好地处理。

（2）有些领域，采用传统的简单机器学习方法，就可以很好地解决问题了，没必要非得用复杂的深度学习方法。

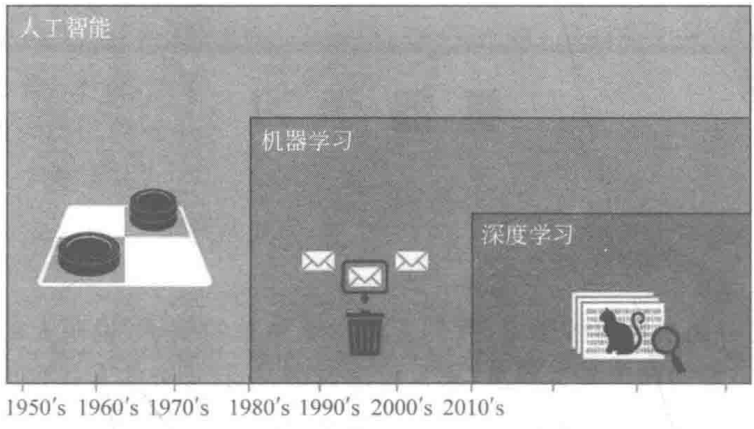


图 1.3 人工智能、机器学习和深度学习的关系

(3) 深度学习的思想，来源于人脑的启发，但绝不是人脑的模拟。举个例子，给一个 4 岁的小孩看一辆自行车之后，再见到哪怕外观完全不同的自行车，小孩也十有八九能作出那是一辆自行车的判断。也就是说，人类的学习过程往往不需要大规模的训练数据，而现在的深度学习方法显然不是对人脑的模拟。

第2章 机器学习

为了更好地了解机器学习，我们首先举个简单的例子，说明人的学习过程。夏天时，我们都去超市买过西瓜，通常通过观察、拍打等方法来找到成熟的甜瓜。观察主要是看西瓜外表的色泽和根蒂的形状，拍打主要是听声音。随着我们经验的不断积累，采用上述方法挑选出“好瓜”的概率越来越大。我们经验不断积累的过程，可用表 2.1 所示的数据集表示。

表 2.1 西瓜数据集

序号	色泽	根蒂	声响	是否好瓜
1	青绿	干缩	浊响	是
2	黑绿	干缩	浊响	是
3	青绿	硬挺	清脆	否
4	黑绿	硬挺	沉闷	否
…	……	……	……	……

通过人脑的不断归纳、总结，我们可以得出“好瓜”的模型是“某种色泽、某种根蒂、某种声响的瓜”。人脑的归纳和总结，就是人的学习过程，这是一个非常复杂的、非线性的神经元学习网络。

那么，机器的学习过程是怎样的呢？我们知道，计算机只能处理数字信号，且只能判断确定的规则，如大于、小于、不等于等。对于模糊的规则，如感知、感觉等则无法判断。举个简单的例子说明机器的学习过程，如下所示。

问：1＋4 等于几？

答：20。

规则反馈：不对，多了。

问：1＋9 等于几？

答：13。

规则反馈：不对，多了。

问：3＋4 等于几？

答：7。

规则反馈：对了。

问：6+9 等于几？

答：13。

规则反馈：不对，少了。

很多很多次以后……

问：2+2 等于几？

答：4。

问：4+5 等于几？

答：9。

这就是机器的学习过程，准确来说这是机器学习方法中最常用的一种，叫有监督机器学习。“监督”的意思，就是数据集训练中的规则反馈。

最开始的几步是对模型的训练，“多了”或“少了”可以理解为训练时的误差监督反馈，模型根据误差调整自身参数，这就是机器学习里常用的反向传播(back propagation)的简单解释。

通过以上示例分析，并结合专业教材对机器学习的定义，此处，我们给出最直观、最易理解的定义，机器学习是一种实现人工智能的科学，主要是通过数据可自动总结规律并改进性能的计算机算法的研究。

2.1 机器学习概述

通常，根据收集的样本是否存在标注，将机器学习分为有监督机器学习和无监督机器学习两类。有监督机器学习所需要的样本集数据，不仅仅包括特征数据，还包括每个样本的一个准确输出值，该输出值即为对该样本的标注。如果收集的样本中没有对每个样本的标注，则该数据集只能用于无监督机器学习。例如，如果我们只收集每只股票的开盘价、收盘价、成交额、成交量、换手率数据，则该数据集只能用于无监督机器学习；而如果我们除此之外还收集每只股票的涨跌信息，并将涨跌信息用于预测，则该数据集适用于有监督机器学习。

2.1.1 有监督机器学习

有监督机器学习中的预测结果可以是连续值，也可以是离散值。根据这样的属性可将有监督机器学习分为回归问题(regression)和分类问题(classification)。回归问题预测的结果一般为连续值，即因变量为数量变量。分类问题预测的结果一般为离散值，即因变量为分类变量。

1. 回归问题

回归问题主要是通过回归算法来探索样本数据与标注数据之间关系的一类问题，而衡量回归算法的规则通常是误差最小。用图 2.1 所示的回归问题示例

图表示，回归问题就是使用计算机算法寻找一条直线（$Y=aX+b$），使得图中的每个红点距离该直线的偏差之和最小。

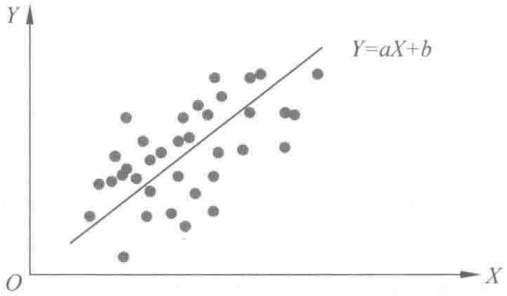


图 2.1　回归问题示例图

在有监督机器学习算法中，常见的回归问题算法有线性回归、逻辑回归、决策树回归、随机森林回归、支持向量机回归等。此处，我们分别以简单的示例说明这些算法的使用方法。

```
#回归问题算法采用的公用数据集
from sklearn import datasets, preprocessing
from sklearn.model_selection import train_test_split
iris=datasets.load_iris()
X, y=iris.data, iris.target
X_train, X_test, y_train, y_test=train_test_split(X, y, random_state
=33)                                                    #随机抽样
scaler=preprocessing.StandardScaler().fit(X_train)   #标准化数据
X_train=scaler.transform(X_train)
X_test=scaler.transform(X_test)
#1. 线性回归
from sklearn.linear_model import LinearRegression
lr=LinearRegression(normalize=True)
lr.fit(X, y)                                  #采用线性回归算法训练模型
y_pred=lr.predict(X_test)                     #用线性回归模型预测
#2. 逻辑回归
from sklearn.linear_model import LogisticRegression
lg=LogisticRegression()
lg.fit(X,y)                                   #采用逻辑回归算法训练模型
y_pred=lg.predict(X_test)                     #用逻辑回归模型预测
#3. 决策树回归
from sklearn.tree import DecisionTreeRegressor
dtr=DecisionTreeRegressor(max_depth=3)
dtr.fit(X,y)                                  #采用决策树回归算法训练模型
y_pred=dtr.predict(X_test)                    #用决策树回归模型预测
```