第5章

人工智能的关键技术

CHAPTER 5

视频讲解

思想引领

人工智能行业的发展关键在于机器学习、神经网络和深度学习等技术的进步。机器学习是人工智能的重要分支，它通过让计算机系统模仿人类的学习与思考方式来改进自身系统的输出效率与结果。神经网络技术是机器学习的一个分支，它通过系统模仿人类大脑神经元思考与信息传递的过程来进行建模，并通过数据的训练得到可以预测与认知的系统，对相关问题进行决策与输出。深度学习是神经网络的一个子领域，它试图模仿人脑的工作方式，通过构建多层神经网络来处理复杂的数据和任务，它在图像识别、语音识别和自然语言处理等领域取得了显著的成果。总体来说，人工智能的关键技术通过模拟人类的学习、理解和思考过程，实现了对复杂数据的高效处理和决策，为各行各业提供了强大的技术支持。

知识目标

1. 了解机器学习、神经网络以及深度学习的基本概念与应用。
2. 掌握机器学习常用算法。
3. 掌握决策树的概念以及相关工具的使用。

能力目标

1. 能够应用机器学习工具解决实际问题。
2. 能够使用决策树模型解决简单问题。
3. 能够终身学习和自我提升。

职业素养目标

1. 学生应秉持严谨的工作态度和强烈的责任心，对待每个项目

都需认真细致，确保工作的高质量和高效率。

2. 鉴于人工智能领域的迅猛发展，学生应具备持续学习的意愿与能力，不断更新自身的知识与技能，以应对未来的挑战。

3. 学生应树立正确的科技观，深刻认识到科技是推动社会进步的重要力量，同时也要警惕科技可能带来的潜在负面影响。

5.1 机器学习基础

机器学习作为人工智能领域的关键分支，通过使计算机从数据中习得规律和模式，进而实现智能化决策与任务执行。在当今数字化时代，掌握机器学习技术不仅能显著提升个人竞争力，还能为职业发展增添显著优势。众多行业，如金融、医疗、制造等，正广泛运用机器学习技术，以提升效率、降低成本并开创新价值。因此，学习机器学习不仅是顺应行业发展的必然要求，更是把握未来机遇的重要基石。

5.1.1 概述与分类

机器学习在当前计算机科学技术发展中占据举足轻重的地位，其核心功能在于使计算机系统能够通过数据和经验的训练，实现自动学习和持续优化，而无须进行显式的编程。这一学习过程不仅使计算机能够从数据中精确提取模式、自主做出决策，还能逐步提升系统性能，为人们提供高效且可靠的解决方案。

1. 定义与本质

(1) 定义。机器学习是一门专注于算法设计与开发的科学，旨在通过数据和经验，使计算机实现自动学习和持续优化。

(2) 本质。机器学习的核心在于模拟人类的思维与学习过程，通过不断优化算法并进行持续训练，最终构建出逼近物理世界的计算模型。该模型能够高效地从数据中提取知识，并将其应用于全新场景。

2. 发展历程

追溯至 17 世纪，帕斯卡尔与费马等先驱者奠定了早期的直接概率推理理论基础。随后，贝叶斯、拉普拉斯等学者进一步深化了对概率论推理问题的研究。这些理论共同构成了机器学习研究中广泛应用的工具和数学基石。普遍认为，经过多个阶段的理论创新与技术突破，20 世纪中叶成为机器学习领域公认的崛起点。

(1) 早期阶段。在这一阶段，研究者开始探索如何让机器模拟人类的学习过程。该阶段的研究主要集中在模式识别和计算学习理论的基础性问题上。阿兰·图灵提出了图灵测试，作为衡量机器是否具备智能行为的标准。弗兰克·罗森布拉特发明了感知机，这一早期神经网络模型为后续神经网络研究奠定了坚实基础。马文·明斯基则从理论层面剖析了以感知机为代表的神经网络模型的局限性，指出其无法解决异或(XOR)等基本问题，这一发现导致了神经网络研究的暂时性停滞。

（2）中期发展。随着计算机技术的不断进步和数据量的急剧增加，机器学习开始广泛应用于实际问题，如语音识别、图像处理等领域。在这一时期，机器学习算法取得了显著进展，涌现出神经网络、支持向量机等多种先进算法。赫伯特·西蒙等学者在人工智能符号逻辑推理方面做出了开创性贡献，为后续机器学习的发展奠定了坚实的理论基础。塞普·林纳亚等首次系统阐述了自动链式求导方法，这一方法成为著名的反向传播算法的雏形。保罗·沃博斯等则首次提出将 BP 算法的思想应用于神经网络，即多层感知机（MLP），从而有力推动了神经网络技术的进一步发展。

（3）现代繁荣。进入 21 世纪，随着大数据和计算能力的显著提升，机器学习迎来了全新的发展机遇。深度学习技术的蓬勃兴起，极大地推动了机器学习领域的进步，使得机器在图像识别、自然语言处理等领域取得了突破性成果。杰夫·辛顿（Geoffrey Hinton）、杨立昆（Yann LeCun）和约书亚·本吉奥（Yoshua Bengio）三位学者被誉为“深度学习三巨头”，他们在深度学习领域的开创性贡献，极大地促进了当代机器学习技术的发展。

3. 分类

机器学习的分类主要划分四大类，下面详细介绍这四种分类。

（1）监督学习。监督学习是指利用一组带有标签（或标记）的已知数据来训练系统，使其能够对新输入数据进行精确的预测或分类（输出）。

案例解析：在手写数字识别任务中，每张手写数字图片作为输入，相应的数字标签作为输出。通过大量已标注数据的训练，模型能够实现对手写数字图片的精准识别。

以一个简明示例说明，假设存在一组数据：1，3，3.55，3.6，4，4.2，5，6，…，99。需要为每个数据添加相应标记，如：1（int），3（int），3.55（float），3.6（float），4（int），4.2（float），5（int），6（int），…，99（int）。其中，（int）和（float）作为标签，分别指示整数值和小数值。监督学习的核心目标即是借助这些带标签的数据集训练系统，使其能够对新数据（例如 100）进行精确的预测与识别，如图 5-1 所示。

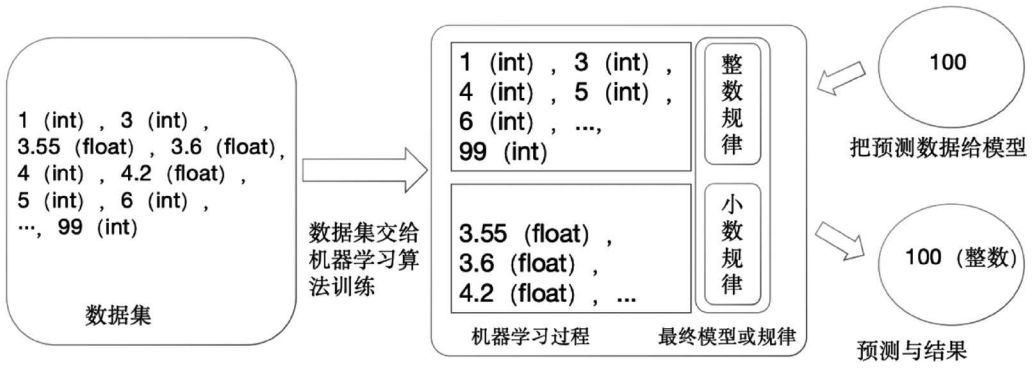


图 5-1　监督学习过程

（2）无监督学习。无监督学习不依赖于带有标签的数据，而是从无标签的数据中自主发现规律、模式和结构。

案例解析：在市场细分中，企业可能会收集大量客户数据，但这些数据没有明确的标签。通过无监督学习，企业能够识别不同客户群体的特征，从而制订有针对性的营销策略。

借用上述监督学习中数据类型分类的例子，无监督学习不对训练数据进行标注，而是通过系统算法进行训练达到相关规律，并对数据（如 100）进行预测与识别，如图 5-2 所示。

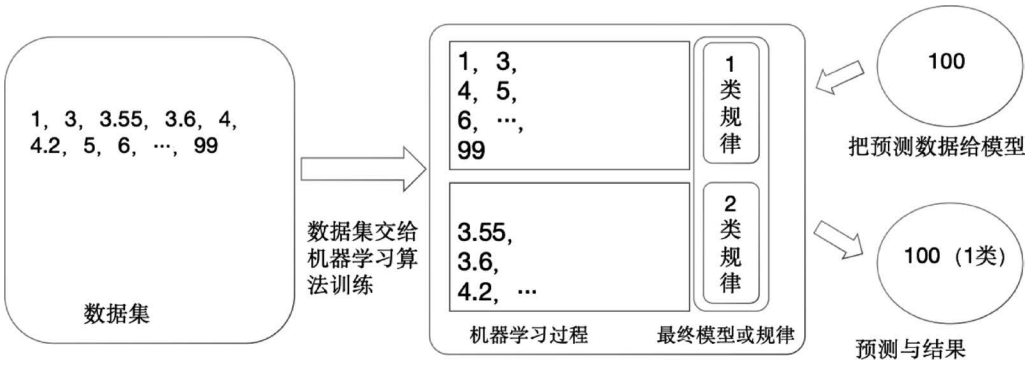


图 5-2　无监督学习过程

（3）半监督学习。在半监督学习中，训练数据既包含少量有标签的样本，也包含大量未标签的样本。

案例解析：在图像分类任务中，可能仅有少量图像拥有明确的标签，而绝大多数图像则缺乏标签。半监督学习能够有效利用这些有限的标签信息，从而显著提升学习效果。

（4）强化学习。强化学习是一种通过与环境的交互，并根据反馈结果不断优化行为策略，从而学习如何达成目标的方法。

案例解析：在游戏领域，例如 AlphaGo 等围棋程序，通过与自己或其他对手进行大量对弈，持续优化策略，最终达到高水平的竞技能力。

5.1.2　常用算法

众多机器学习经典算法各具独特优势，适用于多样化的应用场景。以下是对这些算法的简要概述。

1. 监督学习常用算法

（1）线性回归。线性回归主要用于预测连续值，通过探寻特征与标签之间的线性关系来实现预测。

（2）逻辑回归。逻辑回归适用于解决二分类问题，借助 sigmoid 函数将线性回归模型的输出结果转换为概率值。

（3）支持向量机（SVM）。SVM 通过寻找一个最佳超平面来区分类别，既可用于分类问题，也适用于回归问题。

（4）决策树。决策树通过构建基于特征的决策路径来进行分类或回归预测。

（5）随机森林。随机森林由多个决策树构成，通过集成学习的方式提升预测的准确性和稳定性。

（6）K 近邻算法（KNN）。KNN 依据样本在特征空间中的最近邻样本来进行分类或回归。

（7）朴素贝叶斯。朴素贝叶斯基于贝叶斯定理，通过计算各类别条件概率来预测目标变量。

2. 无监督学习常用算法

(1) K-means 聚类。K-means 是一种广泛应用的聚类算法,通过迭代调整聚类中心,以最小化数据点与聚类中心之间的距离。

(2) 主成分分析。主成分分析旨在降低数据的维度,同时最大限度地保留原始数据的信息。

(3) 关联规则学习。关联规则学习用于揭示数据集中变量之间的关联关系,例如在购物篮分析中识别购买模式。

3. 半监督学习

(1) 图论推理算法。该算法通过分析数据点之间的相似性来构建图结构,进而执行推理和聚类操作。

(2) 拉普拉斯支持向量机。此方法融合了有标签和无标签数据,借助正则化项对模型进行优化,以提升其性能。

4. 强化学习

(1) Q 学习。Q 学习是一种基于表格的离策略学习方法,通过持续更新 Q 值来实现最优策略的学习。

(2) 深度 Q 网络。深度 Q 网络融合了深度学习和 Q 学习的优势,利用神经网络来近似 Q 函数,特别适用于处理大型状态空间的复杂问题。

(3) 策略梯度方法。策略梯度方法直接对策略进行优化,尤其适用于涉及连续动作空间的强化学习问题。

5.1.3　应用前景与未来趋势

随着科技的持续进步和数据的迅猛增长,机器学习作为人工智能的核心分支,正以空前的速度革新生活和工作模式。无论是医疗健康、金融服务,还是智能制造、智慧城市,机器学习的应用领域广泛,未来发展潜力无限。本书将探讨机器学习在多个领域的应用前景及其未来发展趋势。

1. 医疗健康领域

在医疗健康领域,机器学习展现出巨大的潜力。通过分析海量的医疗数据,包括病历、影像和基因组信息,机器学习模型能够有效辅助医生进行疾病诊断、治疗方案推荐以及药物研发。例如,深度学习技术在医学影像分析方面已取得显著进展,能够自动识别 X 光片、CT 扫描和 MRI 中的异常情况,显著提升了诊断的准确性和效率。此外,机器学习还能用于预测疾病的发展趋势,协助医生制订更加个性化的治疗计划。未来,随着可穿戴设备和移动健康技术的不断进步,个人健康数据将愈发丰富,这将为机器学习提供更广阔的发展空间,进一步推动精准医疗的深入发展。

2. 金融服务行业

在金融服务行业，机器学习正逐步重塑传统业务模式。通过深入分析客户的交易行为、信用记录及社交媒体数据，机器学习模型能够更精准地评估信用风险，从而显著提升贷款审批的效率和精确度。此外，机器学习技术还被广泛应用于股票价格预测、投资组合优化以及欺诈行为识别等多个领域，助力金融机构更高效地管控风险并提升盈利能力。展望未来，随着区块链和数字货币技术的不断进步，机器学习在金融监管、智能合约以及去中心化金融等新兴领域的应用将愈发关键。

3. 智能制造领域

在智能制造领域，机器学习作为实现工业 4.0 的核心技术之一，发挥着至关重要的作用。通过深入分析生产过程中的数据，机器学习模型能够实时监控设备状态，精准预测维护需求，有效减少停机时间，从而显著提升生产效率。此外，机器学习在优化生产流程、提升产品质量以及降低生产成本方面也展现出显著优势。展望未来，随着物联网技术的广泛普及和 5G 网络的迅猛发展，机器间的通信将变得更加高效，这无疑将为机器学习在智能制造领域的应用开辟更为广阔的发展空间。

4. 智慧城市建设

在智慧城市的构建过程中，机器学习扮演着举足轻重的角色。通过对城市运行中各类数据的深入分析，如交通流量、能源消耗以及公共安全信息，机器学习模型能够协助城市规划者制定更加科学、合理的决策，从而显著提升城市的运行效率及居民的生活品质。例如，智能交通系统通过实时解析交通数据，优化交通信号灯的调控策略，有效缓解拥堵问题；智能电网则通过预测电力需求，动态调整电力供应，大幅提高能源利用效率。展望未来，随着传感器技术和云计算的持续发展，智慧城市将迈向更高水平的智能化，机器学习亦将在更多领域展现其强大潜力。

5. 教育与人才培养

在教育领域，机器学习同样展现出广阔的应用前景。通过深入分析学生的学习行为和成绩数据，机器学习模型能够为学生量身定制个性化的学习资源和辅导建议，从而助力他们更高效地掌握知识。此外，机器学习技术亦可应用于教师绩效评估、课程优化设计以及教育资源的合理分配等多个层面，有效提升教育的整体质量和公平性。展望未来，随着在线教育和终身学习理念的日益普及，机器学习必将在教育领域发挥愈发关键的作用。

6. 未来发展趋势

随着计算能力的增强和算法的不断创新，机器学习模型将变得更加复杂且强大。深度学习将继续在图像识别、语音处理以及自然语言理解等领域取得显著突破。同时，强化学习作为一种通过试错机制优化决策的方法，将在自动驾驶、游戏设计和机器人控制等领域获得更广泛的应用。此外，联邦学习与隐私保护技术的进步将使机器学习在处理敏感数据时更加安全可靠。

总之，机器学习的未来充满机遇与挑战。伴随着技术的持续进步和应用场景的不断拓展，机器学习将在更多领域发挥关键作用，推动社会进步与发展。然而，也应关注机器学习引发的伦理和社会问题，确保技术发展真正造福全人类。

5.1.4　决策树概念与构建过程

1. 决策树的概念

决策树是一种在机器学习中用于事物分类或回归的学习模型，它体现了对象属性与对象值之间的映射关系。该模型通过节点来表示对象，树中的分支路径则代表某个可能的选择序列，而每个叶节点则对应从根节点到该叶节点所经历的路径所表示的对象值。通常，决策树包含以下几种节点。

(1) 根节点(Root Node)。根节点是所有其他决策的起点，如图 5-3 所示。整棵决策树旨在回答以下问题：在面对具有不确定性的选择或决策问题时，从根节点出发，决策路径及其可能的结果和价值将如何展开。

(2) 事件节点(Event Node)。事件节点标志着面临不确定性结果的时刻，如图 5-4 所示。例如，一家银行正在开发一款新的锁箱应用程序，可能面临以下两种情况：一是投入 8 万元进行产品开发，但最终未能成功上市；二是投入 15 万元，并取得巨大成功，从而获得 100 万元的价值回报。事件的发生伴随着一定的概率，通常只有在满足特定概率条件时才会发生。

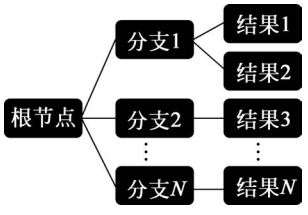


图 5-3　根节点及分支

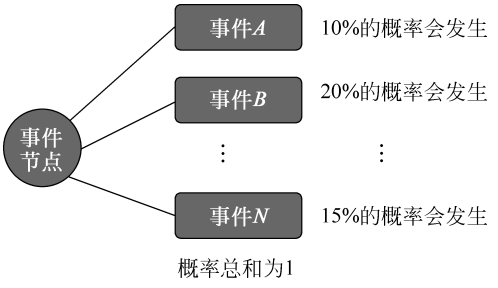


图 5-4　事件节点与事件概率示意图

(3) 决策节点(Decision Node)。决策节点表示可供选择的不同选项，如图 5-5 所示。决策节点在决策过程中扮演关键角色，负责对不同方案进行选择和判断。通过对比各方案的效果或指标，决策节点帮助选择最优方案。以一个简单的购物篮分析为例，决策树可能依据顾客的购买历史、年龄、性别等因素，预测其是否会购买某商品。在此过程中，决策节点可能会根据顾客年龄是否大于或等于 30 岁进行划分，从而形成两个不同的分支。

(4) 终端节点。终端节点(又称为叶子节点)代表事件或决策的最终结果，如图 5-6 所示。在上述锁箱的案例中，尽管该事件经过了全面的评估与详细介绍，但其一个可能的结果(终端节点)为获利 100 万元；而另一个可能的结果(另一终端节点)则表明，若事件失败，银行将至少损失 15 万元的投资。

节点类型分类说明如表 5-1 所示。

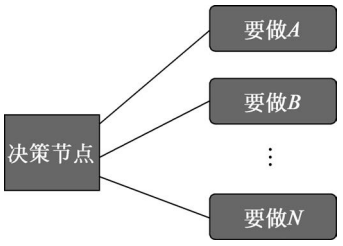


图 5-5　决策节点示意图

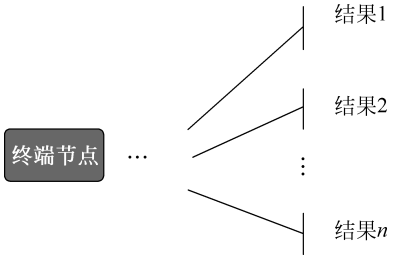


图 5-6　终端节点示意图

表 5-1　节点类型说明

节 点 类 型	手写表示符号	数字表示符号	意　　义
决策节点	方框	方框□	决策选择
事件节点	圆圈	圆圈○	事件发生
终端节点	圆点	线或其他符号 |	最终结果

2. 决策树构建过程

构建决策树时，需细致检查并明确界定各步骤，以便精准绘制各阶段图表。为将其有效应用于实践，需掌握以下要点。

（1）分解过程。尽管构建过程涉及众多步骤，但通常是根据活动性质对各个阶段进行划分。在构建过程中，应将位于决策点之间的活动归为一组。每个决策点均标志着某一阶段的终结及下一阶段的起始。

（2）定义决策。在每个阶段结束时，都会出现一个决策点，项目团队必须对所有可用选项做出选择。例如，在推出新型密码箱产品的过程中，一个阶段可能是进行市场测试。在此阶段的决策点包括：①直接启动全面的产品推广活动；②放弃该产品；③进行第二次市场测试；④对产品进行修改后再次进行市场测试。

（3）估计概率。一旦定义了某个阶段并确定了相关事件，项目团队必须全力以赴为各结果分配相应的概率。这可以依据过往经验、其他环境或事件的经验，或者基于合理的推测等方式进行。所有结果的概率之和必须等于 1，且每个结果的设定都必须兼顾之前的结果。例如，全面推广的客户渗透量可能受到测试营销进展情况等因素的影响。

5.1.5　使用 Excel 工具构建决策树

1. TreePlan 获取与导入

决策树工具一般可以手绘或是使用办公软件、画图软件进行表示，下面介绍一款 Excel 宏文件 TreePlan.xla，可以方便且简单地进行决策树构建。

图 5-7　TreePlan 文件

（1）获取 TreePlan。TreePlan 宏文件可以从 TreePlan 官网进行购买下载，或者从网上下载早前的 TreePlan 试用版。一般地，TreePlan 压缩包解压后一般有三个文件：TreePlan 插件试用版.xla、TreePlan 样例.xls 和 TreePlan 指导书.pdf。相关文件的文件名一般是英文的，为方便案例使用，本书把文件名修改为中文形式，如图 5-7 所示。

（2）把 TreePlan 插件导入 Excel 中。启动 Excel(本例使用 Excel 2016 版本)并打开一个空白的工作表。单击菜单“文件”→“选项”，弹出“Excel 选项”对话框。依次单击“加载项”→“Excel 加载项”，单击“转到”按钮，如图 5-8 所示。弹出“加载宏”对话框，单击“浏览”按钮，如图 5-9 所示。找到 TreePlan.xla，单击“确定”按钮，把插件导入 Excel 当中，如图 5-10 所示。确定“加载项”对话框中 TreePlan 插件选项被选中(打钩)，再单击“加载项”按钮退出对话框。

图 5-8　加载项

图 5-9　浏览宏文件

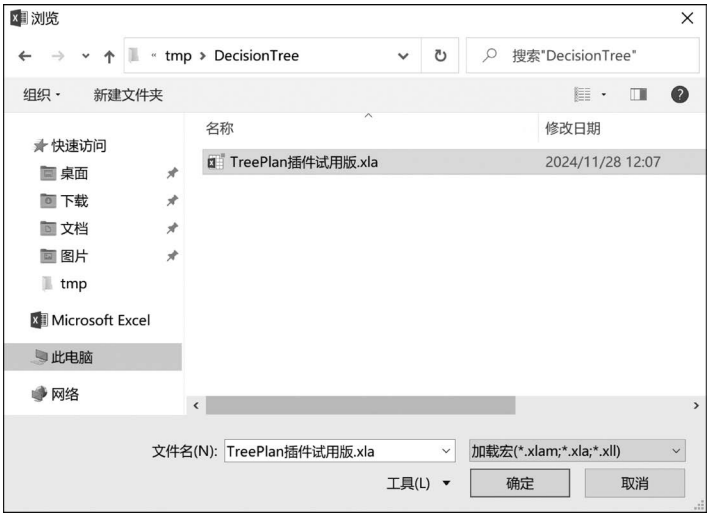


图 5-10　找到宏文件

2. TreePlan 基本使用

把 TreePlan 插件导入后，重新启动 Excel，打开一个工作表，可以看到菜单项中多出“加载项”菜单，单击“加载项”菜单会出现 Decision Tree 菜单命令，如图 5-11 所示。

(1) 创建决策树。在工作表中选择其中一个单元格，如 C5。单击菜单“加载项”中 Decision Tree 按钮，弹出试用通知，单击 I Agree 按钮即可。在 TreePlan New 的对话框中，单击 New Tree 按钮新建决策树，将出现一个决策节点及两个决策分支的默认决策树，适当调整下显示的列宽，让决策树显得美观，如图 5-12 所示。

图 5-11　Excel 决策树菜单

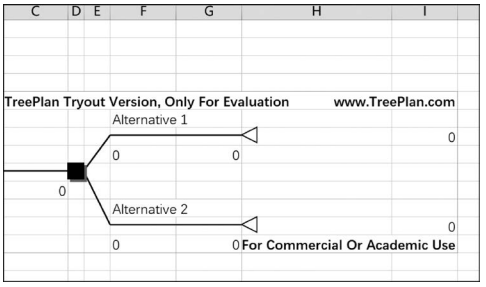


图 5-12　创建决策树

(2) 创建节点与分支。选中终端节点“◁”，单击菜单的 Decision Tree 按钮。弹出对话框，根据需求创建适合的节点类型以及 Branches 分支数。这里保持默认选项，单击 OK 按钮，如图 5-13 所示。在终端节点中创建节点的同时也会创建分支，如图 5-14 所示。

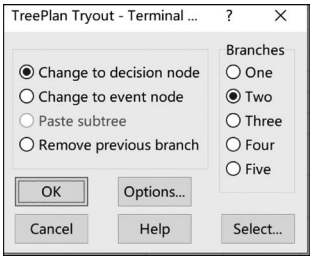


图 5-13　创建节点

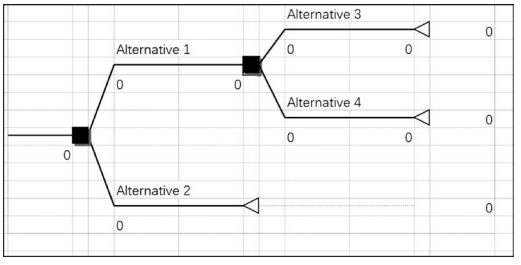


图 5-14　节点创建结果

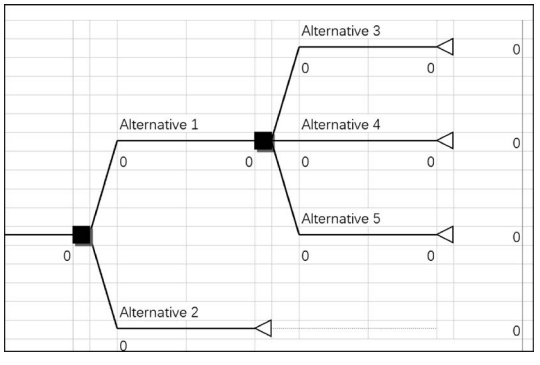


图 5-15　变更或创建分支

(3) 变更或创建分支。在终端节点上创建分支参考上述“创建节点与分支”，在已经存在的其他节点上创建分支，选中要操作的节点(如前面刚增加的节点)，单击 Decision Tree 按钮，弹出对话框，选中 Add branch，单击 OK 按钮即可，如图 5-15 所示。

(4) 删除节点与分支。选中要删除的节点，如 Alternative 3 分支的终端节点，单击 Decision Tree 按钮，弹出对话框。选中 Remove previous branch，单击 OK 按钮，如图 5-16 和图 5-17 所示。

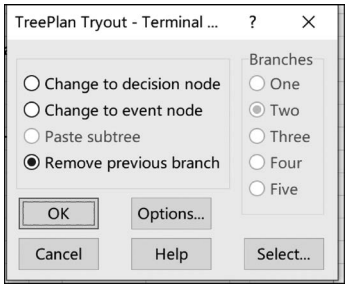


图 5-16　删除节点与分支

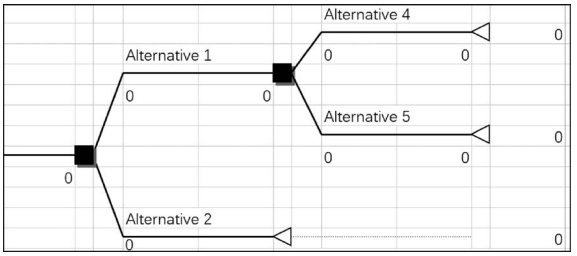


图 5-17　删除节点与分支结果

3. 问题描述

某公司准备投资一种新产品，如果投资项目失败，则会把原本潜在的收益价值损失掉。新的产品必须经过开发、测试、审查等步骤，证明不会对其他产品造成问题，然后才能推向市场。在任何阶段的失败，充其量是花费更多的钱，而在最坏的情况下，将导致完全失败。决策树允许公司量化每个阶段的利弊，同时为公司提供决策框架，以便在项目开始后作出决策。

假设公司从供应商那里得到一个功能不完善的锁箱产品，公司需要决定如何升级这个产品来盈利。公司可以保持产品现状，或在原样的基础上通过增加人手来改进销售流程，或者自行开发产品。如果自行开发产品，可以选择小规模开发，或者全面开发这个产品。

然后，公司设定了开发阶段、决策、不同市场反应的概率，以及预估成本和收入。为了简化理解，特别进行以下说明：如果自行开发新产品，可以开发功能完善的版本，也可以开发最小功能版本；对于功能完善的产品开发完成推向市场，需要增加额外的研发成本 20 万元，但有 45％的概率市场响应良好并为此收益 150 万元，35％的概率市场响应一般并为此收益 15 万元，20％的概率市场响应较差并收益 8000 元；对于最小功能产品推向市场，需要增加额外研发成本 6 万元，但有 10％的概率市场响应良好并为此收益 150 万元，30％的概率市场响应一般并为此收益 7 万元，60％的概率市场响应较差并收益 4000 元。

如果使用现有产品，可以选择不升级产品或通过增加人手改进销售流程的方式。对于保持原样的产品，不需要额外增加成本，但有 70％的概率市场响应良好并为此收益 3 万元，30％的概率市场响应较差并收益 2000 元；对于增加人手改进销售流程的方式，需要花费 4 万元成本，但有 30％的概率市场响应良好并为此收益 50 万元，40％的概率市场响应一般并为此收益 4 万元，30％的概率市场响应较差并收益 5000 元。

4. 利用 TreePlan 构建决策树案例

（1）新建工作表，默认新建决策树，如图 5-18 所示。

（2）调整样式内容。分别选中 Alternative 1 和 Alternative 2 并修改为“开发新产品”和“使用现有产品”，如图 5-19 所示。

（3）为“开发新产品”分支创建“决策节点”以及 2 个分支。选中“开发新产品”分支的终端节点，单击 Decision Tree 按钮，在弹出的对话框中，选中 Change to decision node，在右边的 Branches 分支中选中 Two，单击 OK 按钮，如图 5-20 所示。

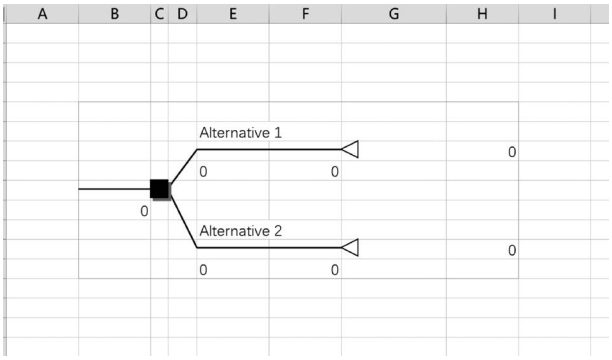


图 5-18　创建默认决策树

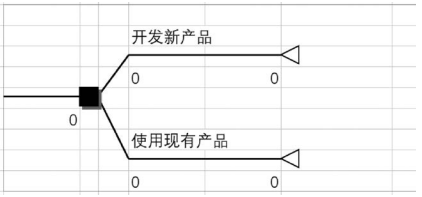


图 5-19　调整样式

分别选中 Alternative 3 和 Alternative 4 并修改为“开发完善的产品”和“开发最小功能产品”，如图 5-21 所示。

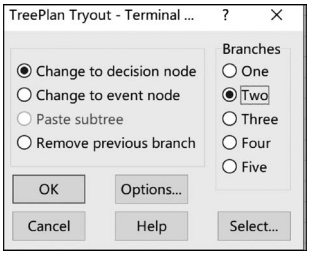


图 5-20　添加分支

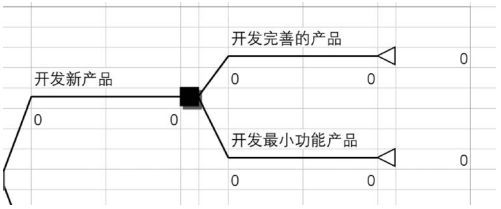


图 5-21　修改样式

(4) 为决策分支设置代价花销(需要付出的成本)。在“开发完善的产品”分支下方有两个“0”，单击左边的“0”设置为“－200000”，左边的“0”修改后，分支下方右边的“0”会自动计算。同时为“开发最小功能产品”分支下方左边的“0”设置为“－60000”，如图 5-22 所示。

(5) 为“开发完善的产品”分支创建“事件节点”以及 3 个分支。选中“开发完善的产品”终端节点，单击 Decision Tree 按钮。在弹出的对话框中，选中 Change to event node，在 Branches 分支中选中 Three，单击 OK 按钮，如图 5-23 所示。

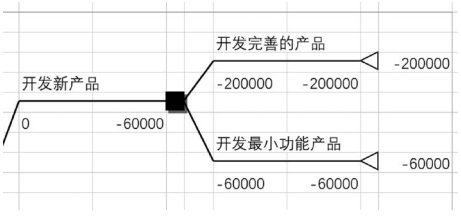


图 5-22　设置成本

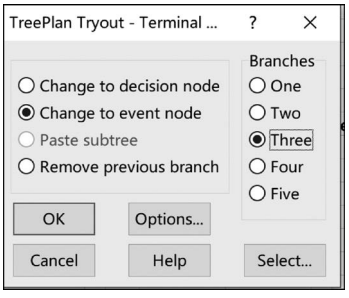


图 5-23　为“开发完善的产品”分支创建 3 个分支

分别选中 Outcome 5、Outcome 6、Outcome 7 并修改为“响应良好”“响应一般”“响应较差”，如图 5-24 和图 5-25 所示。

(6) 为“事件节点”分支设置对应的概率与收益。单击“响应良好”上方的 0.333333 单

元格，设置为 0.45(即 45%的概率)。单击“响应良好”下方的 0 单元格，设置为对应的收益 1500000，如图 5-26 所示。

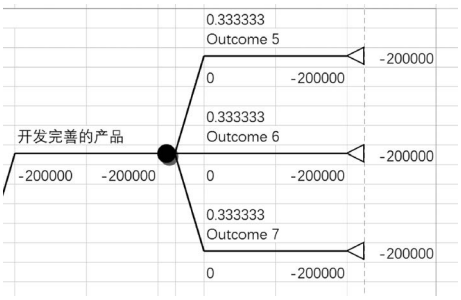


图 5-24　为分支修改样式修改前效果

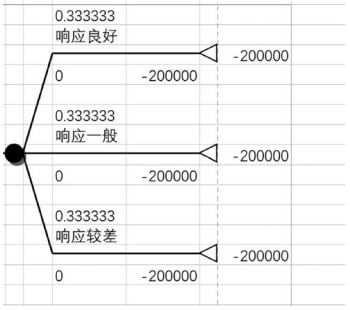


图 5-25　为分支修改样式修改后效果

分别为“响应一般”“响应较差”分支设置其概率(0.35、0.2)与收益(150000、8000)，如图 5-27 所示。

图 5-26　设置概率与收益

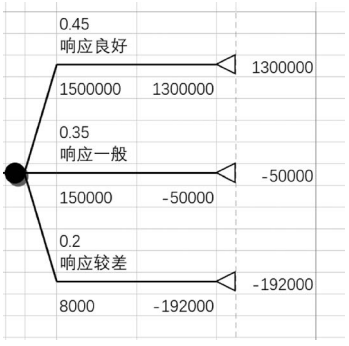


图 5-27　概率与收益修改后效果

(7) 参考前面案例描述与操作步骤，为“开发最小功能产品”分支创建“事件节点”的 3 个分支并设置概率与收益，如图 5-28 所示。

(8) 为“使用现有产品”分支创建“决策节点”以及 2 个分支。选中“使用现有产品”分支的终端节点，单击 Decision Tree 按钮，在弹出的对话框中，选中 Change to decision node，在 Branches 分支中选中 Two，单击 OK 按钮。把分支名称分别设置为“不升级”和“改进销售流程”，为对应分支设置相应的代价花销“0”和“－40000”，如图 5-29 所示。

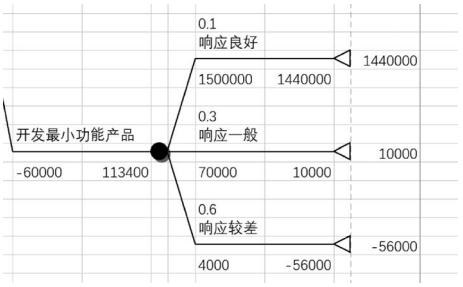


图 5-28　为“开发最小功能产品”分支设置概率与收益

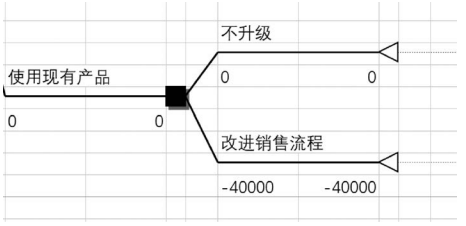


图 5-29　为“使用现有产品”分支设置相关内容

（9）为“不升级”分支创建“事件节点”及 2 个分支，并设置概率与收益，如图 5-30 所示。

（10）为“改进销售流程”分支创建“事件节点”及 3 个分支，并设置概率与收益，如图 5-31 所示。

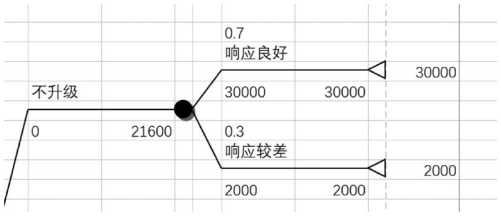


图 5-30　为“不升级”分支设置相关内容

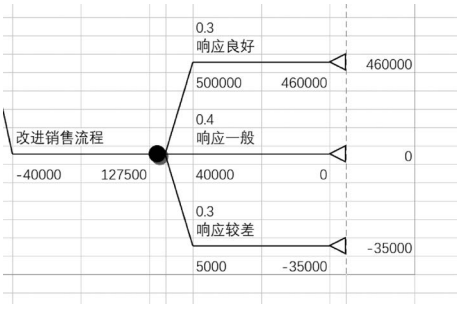


图 5-31　为“改进销售流程”分支设置相关内容

（11）最终得到完整的决策树，如图 5-32 所示。

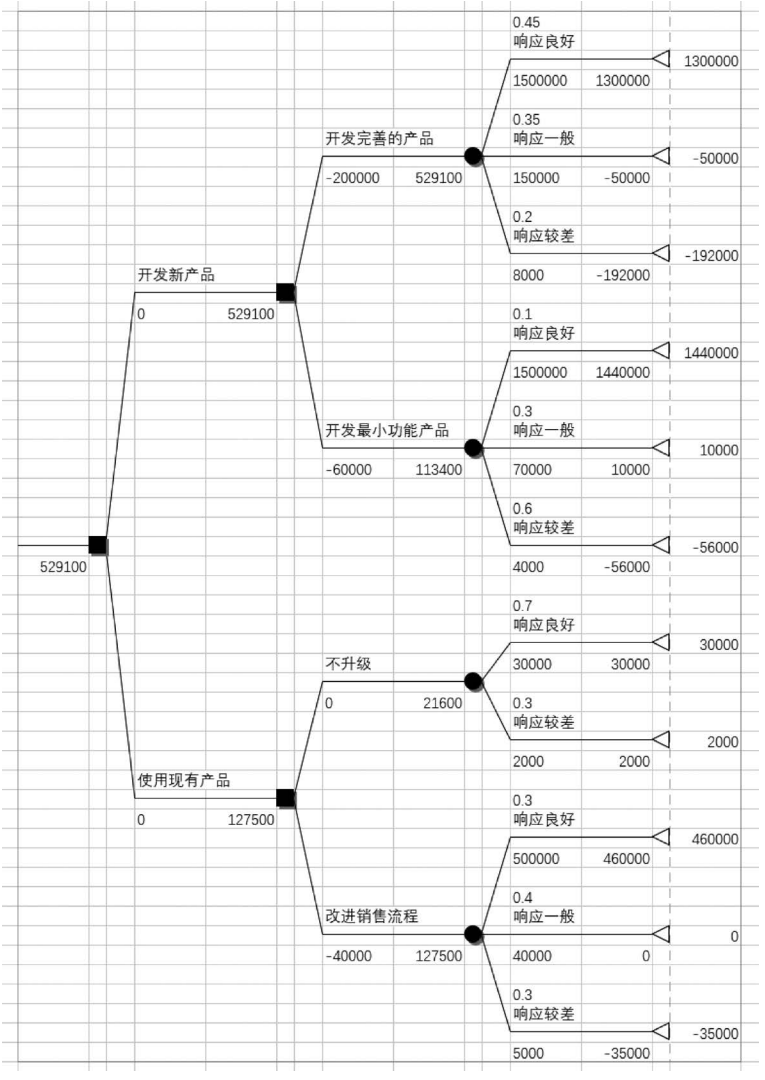


图 5-32　完整决策树

5. TreePlan 决策树结果解释

决策树最右面的单元格的终端值，表示每条从根节点开始到终端节点的分支的最终结果，每个事件分支都有一个可能存在的期望收益，如表 5-2 所示。

表 5-2　TreePlan 终端结果说明（单位：元）

序号	决策分支	事件分支	代价花销	预估成功收益	最终期望收益	最优终端收益
1	开发新产品	开发完善的产品	－200000	1500000	529100	1300000
2		开发最小功能产品	－60000	1500000	113400	1440000
3	使用现有产品	不升级	0	30000	21600	30000
4		改进销售流程	－40000	500000	127500	460000

分支是否好坏，需要看最终的期望收益是否最高。在表 5-2 中，序号 1 的分支的期望收益是 529100 元，是所有分支最高的，代表决策时选用此分支最有可能获利最大。

5.1.6　使用 KNIME 工具构建决策树

1. KNIME 软件下载安装

KNIME 软件在官网下载后，直接安装即可。

2. 问题描述

癌症的确诊通常需要综合多方面数据进行综合诊断。本案例所使用的癌症预测数据集源自 Kaggle 站点的 breast-cancer 乳腺肿瘤数据。该数据集包含 569 条记录，涵盖 id、诊断结果等共计 32 个字段，具体字段信息如表 5-3 所示。通过这些字段，借助 KNIME 平台进行数据处理和模型构建，旨在实现对乳腺肿瘤的初步筛选和诊断。

表 5-3　乳腺肿瘤 breast-cancer 数据集的字段定义

字段名称	字段定义	字段名称	字段定义
diagnosis	诊断结果，B 良性 M 恶性	compactness_se	紧凑度标准差
radius_mean	肿瘤平均半径	concavity_se	凹度标准差
texture_mean	平均纹理	concave points_se	凹点数标准差
perimeter_mean	平均周长	symmetry_se	对称性标准差
area_mean	平均面积	fractal_dimension_se	分形维数标准差
smoothness_mean	平滑度均值	radius_worst	半径最大值
compactness_mean	紧凑度均值	texture_worst	最差纹理特征值
concavity_mean	凹度均值	perimeter_worst	周长最大值
concave points_mean	凹点平均数	area_worst	面积最大值
symmetry_mean	对称性均值	smoothness_worst	平滑度最大值
fractal_dimension_mean	分形维数均值	compactness_worst	紧凑度最大值
radius_se	半径标准差	concavity_worst	凹度最大值
texture_se	纹理标准差	concave points_worst	凹点最大值
perimeter_se	周长标准差	symmetry_worst	对称性最大值
area_se	面积标准差	fractal_dimension_worst	分形维数最大值
smoothness_se	平滑度标准差		

3. 利用 KNIME 构建决策树案例

在 KNIME 中导入数据集，并对数据进行预处理，包括清洗、筛选和转换。通过选择合适的算法和参数设构建模型（本案例选择决策树模型）。在模型训练完成后，对模型进行评估和优化，以提高预测准确性。针对本案例，将重点关注模型的分类效果，即对患者乳腺肿瘤的良性与否（良性或恶性）进行准确区分。通过不断调整或训练，最终得到一个具有较高预测准确率的决策树模型，为乳腺癌的早期诊断提供参考依据。

使用 KNIME 进行数据模型构建时，建议提前规划好工作流程。以下是本案例的工作流程，如图 5-33 所示。

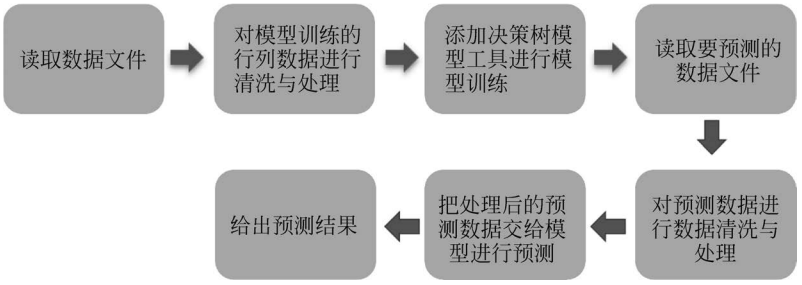


图 5-33　工作流程图

（1）打开 KNIME，新建工作项目。打开 KNIME 软件后，单击 Home 标签页面的 Create new workflow 按钮，如图 5-34 所示。弹出 Create a new workflow 窗口，输入工作项目的名称如 test 后，单击 Create 按钮创建项目，如图 5-35 和图 5-36 所示。

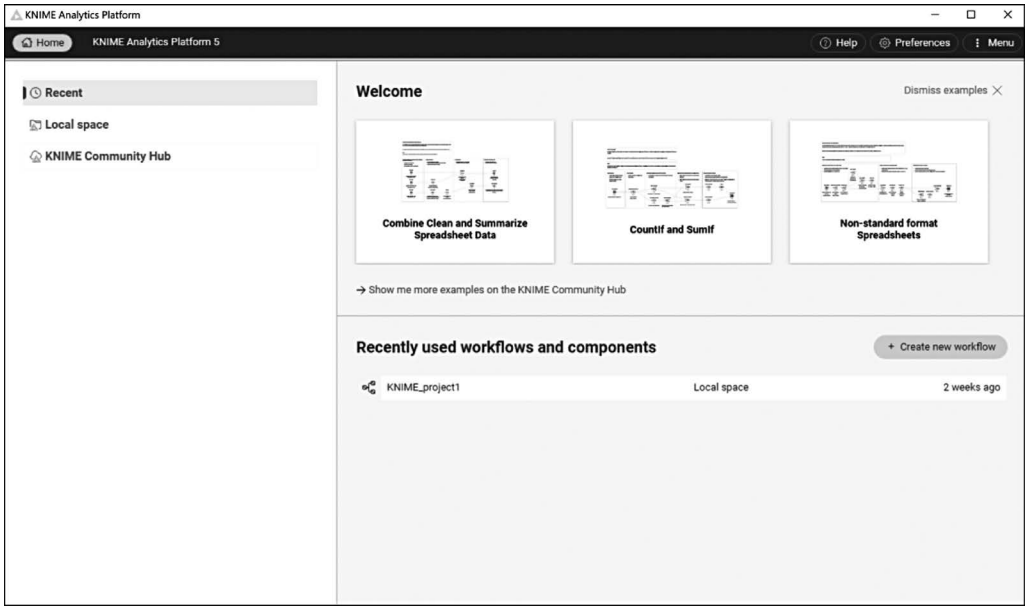


图 5-34　创建项目

（2）根据项目规划，读取数据文件。因为 breast-cancer 的数据文件是 CSV 格式，因此在 KNIME 项目左边的 Nodes 面板中找到 CSV Reader 节点，用鼠标将其拖入右边的工作

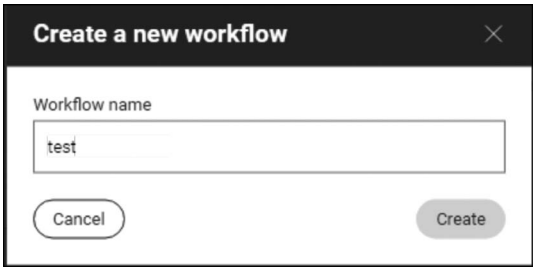


图 5-35　给项目命名

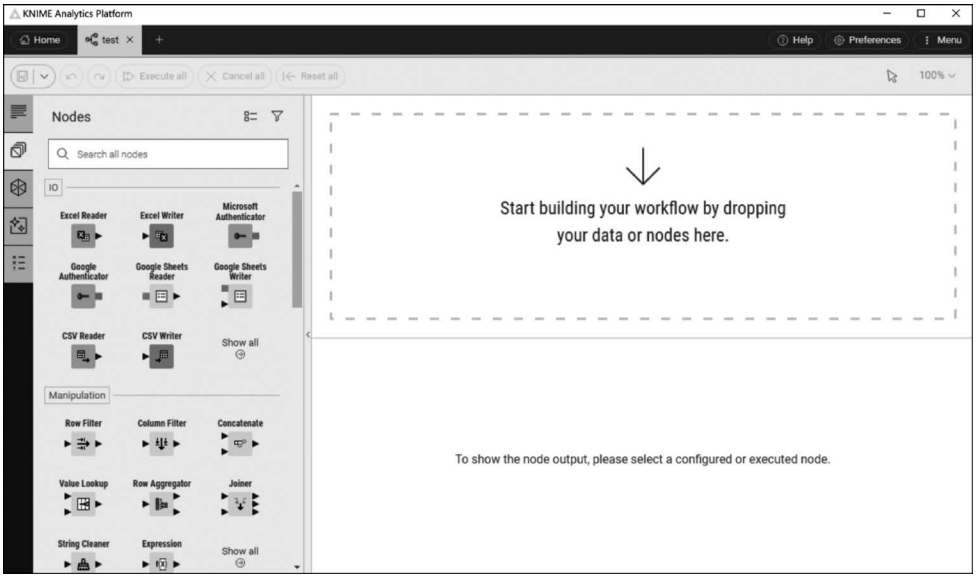


图 5-36　项目创建后的界面

区中，如图 5-37 所示。

双击工作区中的 CSV Reader 节点下方的 Add comment 注释名称，输入“读取文件”。鼠标移动节点上方，此时节点图标上方会出现 4 个图标，分别是 Configure（设置）、Execute（执行）、Cancel（取消）与 Reset（重置），如图 5-38 所示。单击 Configure 图标，弹出配置窗口，设置数据文件的路径和格式，确保数据正确导入，如图 5-39 所示。在弹出的参数设置窗口中，在 Settings 标签页下，找到 Input location 设置组的 File 文件标签，单击 Browse 浏览按钮选择相应的 breast-cancer. csv 数据文件。

图 5-37　拖入 CSV Reader 节点

图 5-38　节点 4 个操作按钮

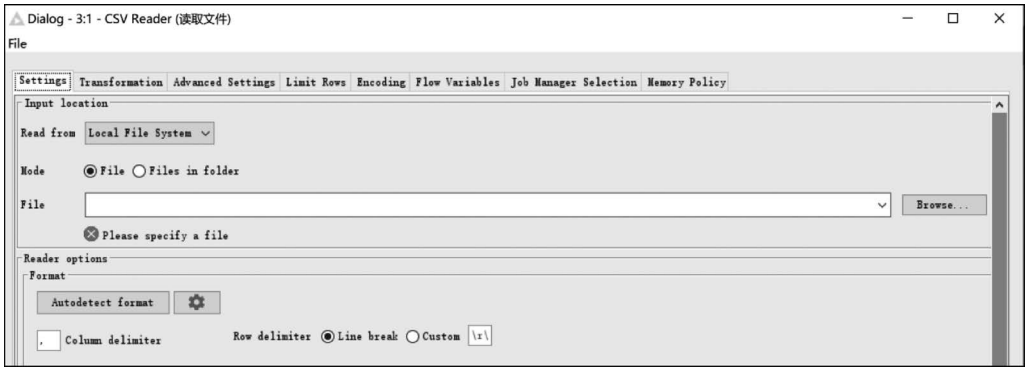
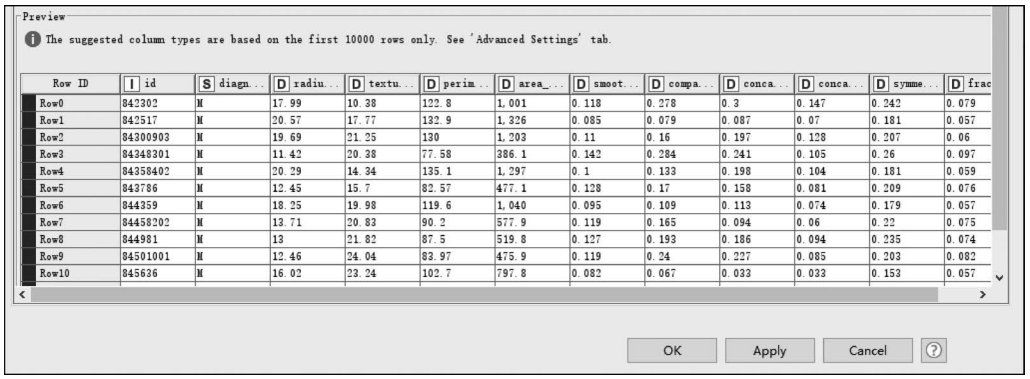


图 5-39　文件配置界面

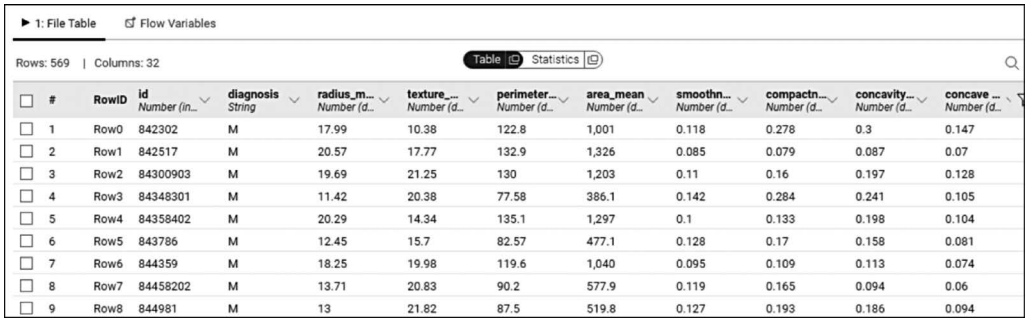
此时，在 Setting 标签页面下方的 Preview 选项组会进行数据预览，如图 5-40 所示。



Row ID	id	diagn...	radiu...	textu...	perim...	area_...	smoot...	compa...	conca...	conca...	symme...	frac
Row0	842302	M	17.99	10.38	122.8	1,001	0.118	0.278	0.3	0.147	0.242	0.079
Row1	842517	M	20.57	17.77	132.9	1,326	0.085	0.079	0.087	0.07	0.181	0.057
Row2	84300903	M	19.69	21.25	130	1,203	0.11	0.16	0.197	0.128	0.207	0.06
Row3	84348301	M	11.42	20.38	77.58	386.1	0.142	0.284	0.241	0.105	0.26	0.097
Row4	84358402	M	20.29	14.34	135.1	1,297	0.1	0.133	0.198	0.104	0.181	0.059
Row5	843786	M	12.45	15.7	82.57	477.1	0.128	0.17	0.158	0.081	0.209	0.076
Row6	844359	M	18.25	19.98	119.6	1,040	0.095	0.109	0.113	0.074	0.179	0.057
Row7	84458202	M	13.71	20.83	90.2	577.9	0.119	0.165	0.094	0.06	0.22	0.075
Row8	844981	M	13	21.82	87.5	519.8	0.127	0.193	0.186	0.094	0.235	0.074
Row9	84501001	M	12.46	24.04	83.97	475.9	0.119	0.24	0.227	0.085	0.203	0.082
Row10	845636	M	16.02	23.24	102.7	797.8	0.082	0.067	0.033	0.033	0.153	0.057

图 5-40　文件预览窗口

单击 OK 按钮导入数据到节点中。此时在工作区面板下方的 File Table 标签已经出现导入的数据的列名，可以单击下方的 Execute 按钮查看数据详细情况，如图 5-41 所示。



#	RowID	id Number (in...	diagnosis String	radius_m... Number (d...	texture_... Number (d...	perimeter... Number (d...	area_mean Number (d...	smoothn... Number (d...	compactn... Number (d...	concavity... Number (d...	concave ... Number (d...
1	Row0	842302	M	17.99	10.38	122.8	1,001	0.118	0.278	0.3	0.147
2	Row1	842517	M	20.57	17.77	132.9	1,326	0.085	0.079	0.087	0.07
3	Row2	84300903	M	19.69	21.25	130	1,203	0.11	0.16	0.197	0.128
4	Row3	84348301	M	11.42	20.38	77.58	386.1	0.142	0.284	0.241	0.105
5	Row4	84358402	M	20.29	14.34	135.1	1,297	0.1	0.133	0.198	0.104
6	Row5	843786	M	12.45	15.7	82.57	477.1	0.128	0.17	0.158	0.081
7	Row6	844359	M	18.25	19.98	119.6	1,040	0.095	0.109	0.113	0.074
8	Row7	84458202	M	13.71	20.83	90.2	577.9	0.119	0.165	0.094	0.06
9	Row8	844981	M	13	21.82	87.5	519.8	0.127	0.193	0.186	0.094

图 5-41　导入文件后的数据

(3) 在确认数据导入无误后，接下来需对数据进行清洗。假设数据集可能存在某个值缺失了，因此对它进行清洗，让其缺失的值设置为默认的某个值。例如，可以选择将缺失值填充为该列的平均值或者 0，这取决于具体的数据特性和业务需求。在这个案例中，统一设置为 0。在 KNIME 中，可以轻松实现这一步骤，通过使用 Missing Value 节点来处理缺失数据。找到并拖曳 Missing Value 节点到工作区，如图 5-42 所示，并注释名称为“清洗”。

按住“读取文件”节点黑色三角图形不动，拖向“清洗”节点左边的黑色三角进行“数据传送”，这样就可以把“读取文件”节点操作完成的数据传递给下一个节点进行操作处理，如图 5-43 所示。

图 5-42　为工作区添加 Missing Value 节点

图 5-43　节点连接

接下来，在“清洗”节点中打开配置窗口，选择 Default 标签，并将下方的 Number(integer)、Number(double)两项的整数值与浮点数值选择为 Fix value，并将缺失值设置为 0。将 String 字符串缺失默认值设置为 Most Frequent Value，如图 5-44 所示。确认设置无误后，单击 Apply 按钮应用更改，单击 OK 按钮退出设置窗口。随后，可以继续对数据进行必要的预处理，如去除重复项、筛选特定列等，为后续数据分析做好准备。

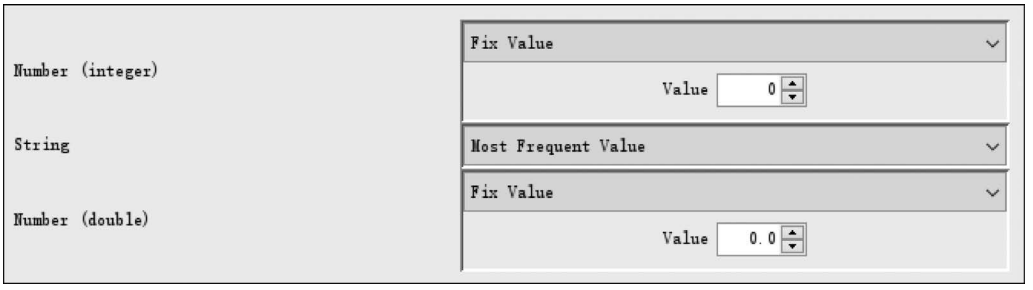


图 5-44　设置空数据的缺失默认值

选择“清洗”节点，单击节点上方 Execute 按钮或按 F7 快捷键，执行清洗节点处理，可以看到工作区下方有结果出现。

(4) 对数据进行预处理。对清洗后的数据，根据需求选取数据值的前 559 行数据进行训练，将后 10 行数据作为测试集。为此，将使用 Row Filter 节点来筛选指定行数的数据。

拖曳 Row Filter 节点至工作区并注释名称为“筛选行”，选中“清洗”节点右边黑色三角拖动连接到“筛选行”节点左边黑色三角进行连接，完成数据流程的搭建，如图 5-45 所示。

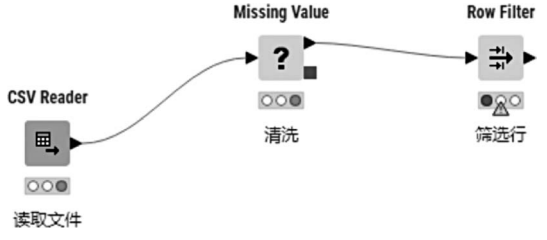


图 5-45　添加筛选行过滤节点

在“筛选行”节点打开配置筛选条件设置窗口。在窗口中单击 Add criterion 添加条件，在 Filter column 筛选列中选中 Row number 列，并在 Operator 比较器中选择≤选项，在 Value 值中设置 559，以确定训练集的范围，如图 5-46 所示。单击 OK 按钮应用筛选条件，关闭配置窗口。此时，工作流中的“筛选行”节点已准备好执行。单击 Execute 按钮执行，筛选行节点会根据设置的条件快速完成数据的划分。在执行完毕后，可在工作区查看到已成功筛选出前 559 行数据作为训练集，如图 5-47 所示。

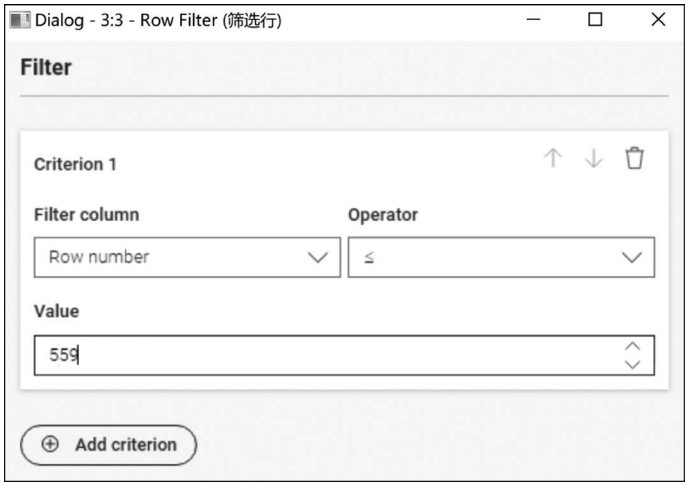
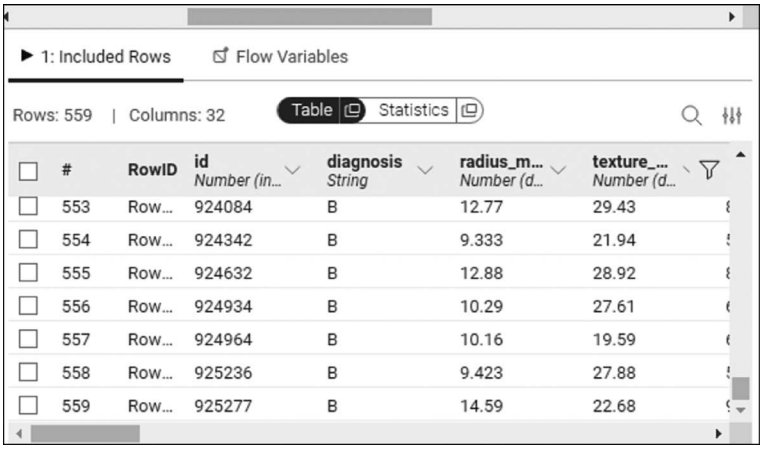


图 5-46　设置筛选行的范围



#	RowID	id Number (in...	diagnosis String	radius_m... Number (d...	texture_... Number (d...
553	Row...	924084	B	12.77	29.43
554	Row...	924342	B	9.333	21.94
555	Row...	924632	B	12.88	28.92
556	Row...	924934	B	10.29	27.61
557	Row...	924964	B	10.16	19.59
558	Row...	925236	B	9.423	27.88
559	Row...	925277	B	14.59	22.68

图 5-47　筛选行结果

（5）构建决策树模型。在 Nodes 面板中找到 Analytics 分类，选择 Decision Tree Learner 决策树模型学习训练节点，并将其拖曳至工作区。将其命名为“决策树训练”，并与“筛选行”节点相连，如图 5-48 所示。

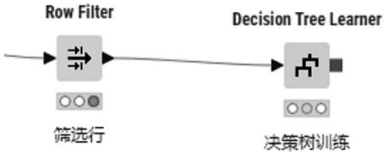


图 5-48　添加决策树模型

打开“决策树训练”节点的配置窗口，设置所需参数。把 Options 选项卡中的 General 组的 Class column 分类列中指定数据集中分类的列，本案例数据集分类列为 diagnosis 诊断结果列。其他参数保留默认值，如

图 5-49 所示。最后单击 Apply 按钮以应用参数设置并退出配置窗口。单击“决策树训练”节点上方的 Execute 按钮以训练模型。

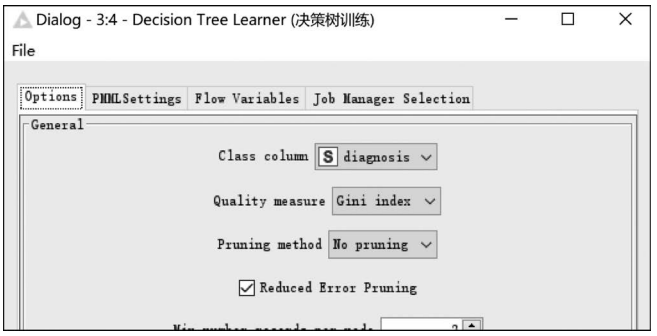


图 5-49　设置模型参数

（6）根据前面的步骤以及项目规划，提供预测数据，并进行初步的清洗与预处理。参考“筛选行”节点的操作，完成数据流程的搭建，如图 5-50 所示。

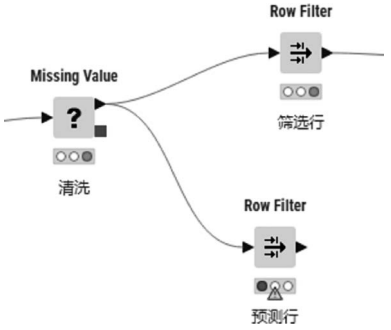


图 5-50　添加预测行筛选节点

① 打开“预测行”节点配置窗口中添加条件，如图 5-51 所示。

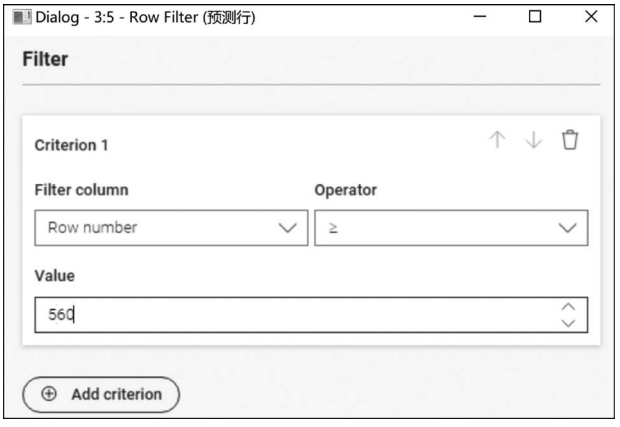


图 5-51　设置预测行范围

② 单击 OK 按钮应用筛选条件，关闭配置窗口。此时，工作流中的“预测行”节点已准备好执行。单击 Execute 按钮执行完毕后，可在工作区查看到预测数据集结果只有 10 条预测数据，如图 5-52 所示。

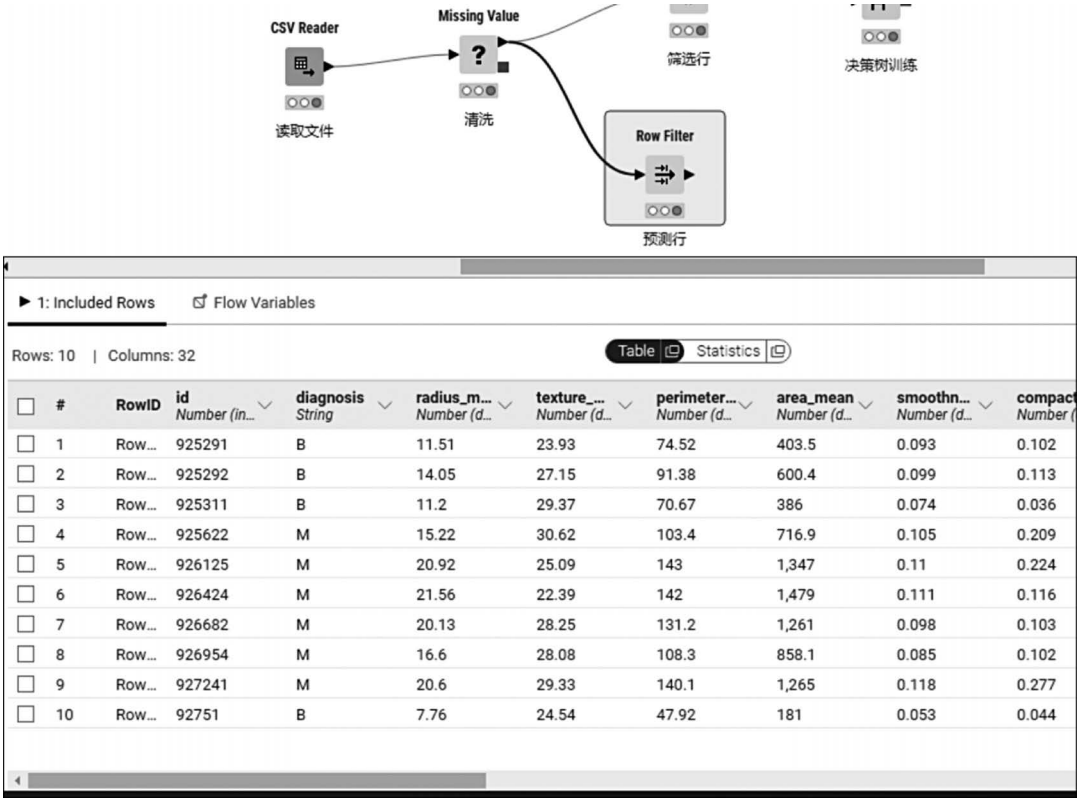


图 5-52　预测行数据

③ 对预测数据进行预处理。本案例中把预测数据集中的 diagnosis 诊断结果列从数据中删除。在 Nodes 面板的搜索框中搜索 column，找到 Column Filter 列筛选节点，将其拖曳至工作区，命名为“删除诊断列”，并与“预测行”节点相连，如图 5-53 所示。

④ 在配置窗口中找到 Includes 列表框，在框中找到 diagnosis 列并单击，单击“<”按钮把 diagnosis 列添加到 Excludes 排除列表框中，如图 5-54 所示，单击 OK 按钮退出配置。

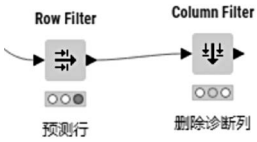


图 5-53　添加删除诊断列节点

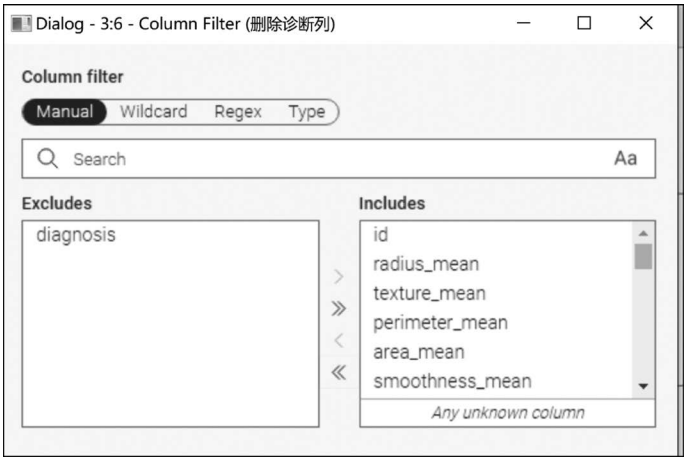
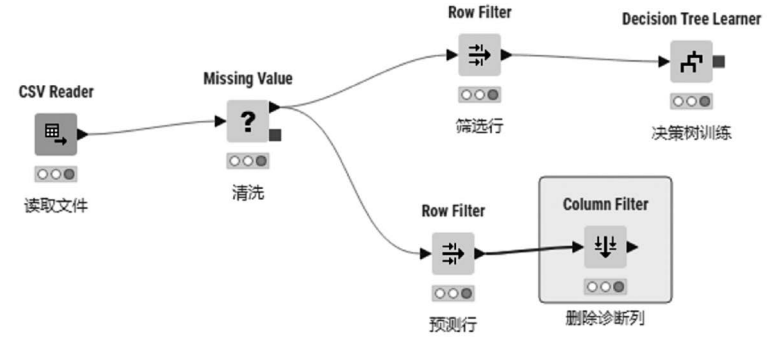


图 5-54　删除 diagnosis 列

⑤ 执行“删除诊断列”节点，并查看结果。此时，结果集中看不到 diagnosis 列的数据，如图 5-55 所示。



▶ 1: Filtered table　Flow Variables

Rows: 10 | Columns: 31　Table　Statistics

#	RowID	id Number (in...	radius_m... Number (d...	texture_... Number (d...	perimeter... Number (d...	area_mean Number (d...
1	Row...	925291	11.51	23.93	74.52	403.5
2	Row...	925292	14.05	27.15	91.38	600.4
3	Row...	925311	11.2	29.37	70.67	386
4	Row...	925622	15.22	30.62	103.4	716.9
5	Row...	926125	20.92	25.09	143	1,347
6	Row...	926424	21.56	22.39	142	1,479
7	Row...	926682	20.13	28.25	131.2	1,261
8	Row...	926954	16.6	28.08	108.3	858.1
9	Row...	927241	20.6	29.33	140.1	1,265
10	Row...	92751	7.76	24.54	47.92	181

图 5-55　删除列后的结果

(7) 随后，将预处理后的数据集接入“预测模型”节点，进行模型预测。

① 在 Nodes 面板中搜索 Decision，找到 Decision Tree Predictor 预测节点，将其拖曳至工作区，并命名为“模型预测”。将“删除诊断列”节点右边黑色三角与“模型预测”节点左边黑色三角相连。将“决策树训练”节点右边的蓝色方框输出端口连接至“模型预测”节点左边蓝色方框输入端口，以提供所需的分类模型，如图 5-56 所示。

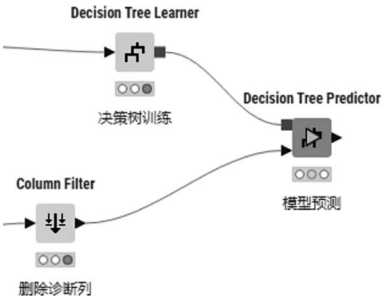


图 5-56　添加预测模型

② 在完成连接后，打开“模型预测”节点的配置窗口，在 Options 选项卡选中 Change prediction column name 以更改预测结果的列名，并在下面输入框中填写“诊断结果”，如图 5-57 所示。单击 Apply 按钮应用配置并退出。

③ 此时，模型预测节点已准备就绪。以下是完整案例的工作流程图，如图 5-58 所示。

④ 执行“模型预测”节点，并查看结果。可以看到结果集最后有一列“诊断结果”列，而

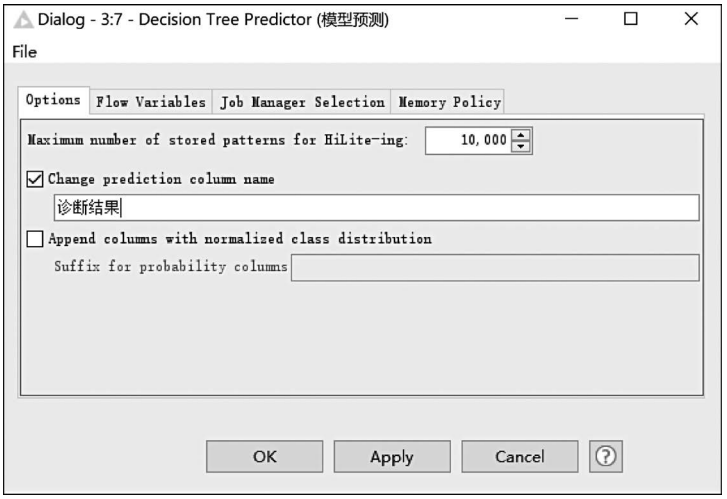


图 5-57　设置预测结果列名

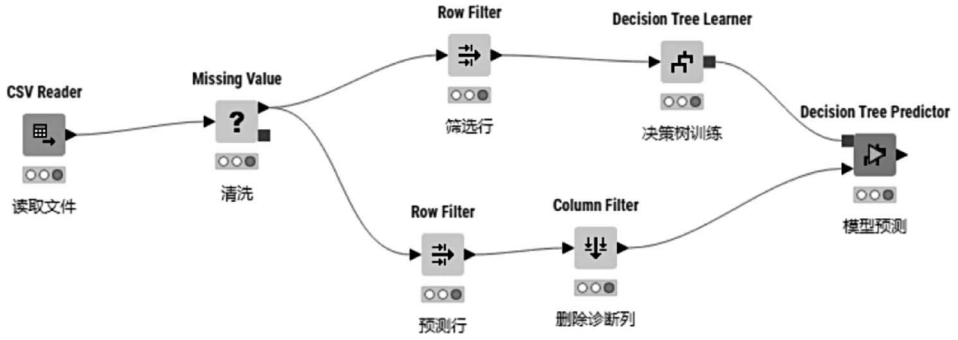


图 5-58　完整工作流程

且给出分类结果，如图 5-59 所示。

Table　Statistics

smoothn... Number (d...	compactn... Number (d...	concavity... Number (d...	concave ... Number (d...	symmetr... Number (d...	fractal_di... Number (d...	诊断结果 String
0.13	0.252	0.363	0.097	0.211	0.087	B
0.124	0.226	0.133	0.105	0.225	0.083	B
0.093	0.055	0	0	0.157	0.059	B
0.142	0.792	1.17	0.236	0.409	0.141	M
0.141	0.419	0.66	0.254	0.293	0.099	M
0.141	0.211	0.411	0.222	0.206	0.071	M
0.117	0.192	0.322	0.163	0.257	0.066	M
0.114	0.309	0.34	0.142	0.222	0.078	M
0.165	0.868	0.939	0.265	0.409	0.124	M
0.09	0.064	0	0	0.287	0.07	B

图 5-59　最终预测结果

4. KNIME 决策树结果解释

前面案例的分类结果提供了初步的预测诊断，通过与原数据对比，预测结果具有较高的

准确率，为后续医疗决策提供了有力的数据支持。

5.2　神经网络

神经网络是机器学习领域的重要分支，通过模拟人脑神经元的连接关系来处理复杂数据。它由大量节点相互连接，形成层次分明的结构，能够捕捉数据的非线性关系并高效进行特征提取。在机器学习中，神经网络广泛应用于分类、回归、聚类等任务，展现出卓越的学习能力和高度适应性。随着计算能力的显著提升和算法的不断创新，神经网络在图像识别、自然语言处理等领域取得了突破性成果，有力推动了机器学习技术的进步。

5.2.1　基本原理

神经网络是一种模拟人脑神经元连接关系的计算模型，通过模拟人脑神经元间的复杂连接和信息传递过程，实现对数据的学习和处理。该网络由大量节点（或称神经元）相互连接构成，每个节点负责接收输入信号，进行加权求和，并通过激活函数生成输出信号，如图 5-60 所示。神经网络通过调整连接权重，学习数据中的模式和规律，进而实现对未知数据的预测和分类。

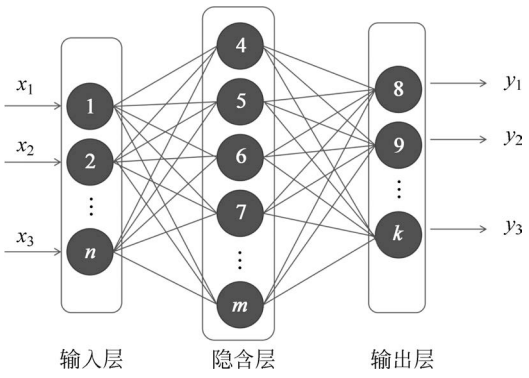


图 5-60　神经网络模型基本原理图

通过一个简明的例子来理解神经网络的基本原理。假设任务是判断一张图片中是否包含猫，可以利用神经网络来完成这项任务。首先，需准备一些训练数据，包括含猫的图片和不含猫的图片。接着，将这些图片作为输入传递给神经网络。

在神经网络中，信息以层次化的方式传递，从输入层起始，经过一个或多个隐藏层，最终抵达输出层。每一层的神经元仅与相邻层的神经元相连，形成前馈网络结构。这种结构使神经网络能够捕捉数据中的复杂非线性关系，并有效进行特征提取和表示。

在猫识别任务中，输入层接收图片的像素值作为输入。随后，这些输入被传递至隐藏层。隐藏层中的神经元对输入进行加权求和，并通过激活函数生成输出，这一过程可视作对图片特征提取的环节。最终，输出层依据隐藏层的输出来判断图片中是否包含猫。

总之，神经网络作为一种强大的机器学习工具，能够处理复杂的非线性关系，并具备良好的泛化能力。通过深入理解其基本原理和结构类型，可以更有效地设计和训练神经网络模型，从而解决各类实际问题。

5.2.2　结构类型

神经网络根据其特定的应用场景和功能需求，被设计成多种不同的结构与类型。以下将介绍目前主流的结构类型。

1. 前馈神经网络

前馈神经网络(Feedforward Neural Networks,FNN)是最基础且广泛应用的一种神经网络结构,其信息流动为单向,从输入层经过隐藏层直接传递至输出层。FNN 特别适用于简单的模式识别任务,诸如图像分类和文本分类等。

例如,利用 FNN 可以识别手写数字。首先,将手写数字图像作为输入数据,随后通过多层神经元的逐层处理,最终获得每个数字类别的概率分布。通过对比这些概率值,即可确定输入图像中所对应的数字。

2. 卷积神经网络

卷积神经网络(Convolutional Neural Networks,CNN)用于处理图像数据,CNN 通过运用卷积层,自动学习图像的空间层次结构特征。卷积操作不仅能有效减少参数数量,还能保留关键的空间信息,这使得 CNN 在图像识别、视频分析等领域取得了显著成效。

例如,利用 CNN 识别交通标志的过程如下:首先,将交通标志图像作为输入数据;接着,通过多个卷积层和池化层的逐层处理,自动提取交通标志的特征表示;最后,借助全连接层对这些特征进行分类,从而确定交通标志的具体类别。

3. 循环神经网络

循环神经网络(Recurrent Neural Networks,RNN)与 FNN 不同,RNN 具备反馈连接,能够允许信息在时间序列中传递。这一特性使得 RNN 特别适合处理序列数据,例如自然语言处理、语音识别以及时间序列预测等问题。然而,传统的 RNN 常常面临梯度消失或梯度爆炸的难题,为此,长短期记忆网络(LSTM)和门控循环单元(GRU)等变体被引入,以有效解决这些问题。

以股票价格预测为例,可以利用 LSTM 网络进行操作。首先,将历史股票价格作为输入数据;接着,借助 LSTM 层的记忆功能,捕捉时间序列中的长期依赖关系;最后,通过全连接层对这些特征进行预测,从而得出未来一段时间的股票价格走势。

4. 生成对抗网络

生成对抗网络(Generative Adversarial Networks,GAN)是一种由两部分组成的网络——生成器和判别器。生成器致力于生成逼真的数据样本,而判别器则负责区分真实样本与生成器产生的假样本。两者通过相互竞争的方式协同进化,最终生成高质量的数据样本。GAN 在数据增强、艺术创作等领域展现出广阔的应用前景。

例如,可以利用 GAN 生成新的图像数据。首先,训练生成器网络以生成逼真的图像;随后,训练判别器网络以区分真实图像与生成器产生的假图像。通过持续迭代训练这两个网络,生成器生成逼真图像的能力将逐步提升。

5. 自编码器

自编码器(Autoencoders)是一种基于无监督学习的神经网络,旨在通过压缩和解压缩输入数据来学习数据的有效表示。自编码器通常由一个编码器和一个解码器组成,其中编

码器负责将输入数据转换为低维编码，而解码器则尝试从该编码中重构原始数据。自编码器在降维、去噪和特征学习等领域具有显著的应用价值。

例如，在人脸识别任务中，首先将人脸图像作为输入数据；接着，编码器将高维的人脸图像转换成低维的人脸特征表示；最后，解码器从这个人脸特征表示中重构出原始的人脸图像。在这一过程中，自编码器能够自动学习到人脸图像的关键特征表示。

除了上述常见的神经网络结构外，还存在许多其他类型的网络结构，如深度信念网络（DBN）、稀疏编码网络和极限学习机（ELM）等，它们各具特色，适用于不同的应用场景和问题。

5.2.3　训练与优化

神经网络的训练过程旨在通过调整网络中的权重和偏置，以最小化损失函数。这一过程通常借助梯度下降算法来完成，其中最广泛应用的是随机梯度下降算法。在训练过程中，需计算损失函数对各个参数的梯度，并沿着梯度的反方向对参数进行更新。此过程将反复进行，直至损失函数收敛至一个较小的值。

为了更深入地理解这一过程，可以借助一个简明的例子来说明。假设存在一个基础的前馈神经网络，其功能是识别手写数字。为了有效训练这一网络，必须定义一个损失函数，用以评估模型输出与实际标签之间的偏差。常见的损失函数包括均方误差和交叉熵损失等。在此例中，选择交叉熵损失作为损失函数。接着，采用随机梯度下降算法对损失函数进行优化。具体操作为：在每次迭代过程中，计算损失函数对各个参数的梯度，并沿梯度的反方向调整参数值。这一过程将反复进行，直至损失函数趋于一个较小的稳定值。

然而，在实际应用场景中，常常会遇到诸如过拟合、欠拟合以及局部最优解等挑战。针对这些难题，可以采取以下策略。

1. 正则化

为了防止过拟合，可以在损失函数中引入正则项，以对模型的复杂度进行惩罚。常见的正则化方法包括 L1 正则化和 L2 正则化等。例如，通过在损失函数中添加 L2 正则项，能够有效约束权重的大小，从而避免模型因过于复杂而导致的过拟合问题。

2. 数据增强

为了丰富训练数据的多样性并增强模型的泛化能力，可以对训练数据进行一系列变换操作。例如，通过对图像进行旋转、缩放和平移等处理，生成新的训练样本。这种做法能够有效提升训练数据的多样性，从而进一步提高模型的泛化能力。

3. 早停法

为避免过拟合并节约计算资源，可在验证集上实时监控模型性能，一旦性能不再提升，即提前终止训练。此方法能有效防止过拟合，同时减少计算资源的消耗。

4. 学习率调度

为了加速收敛并提升模型性能，可以采取动态调整学习率的策略。常见的学习率调度

方法包括固定学习率、逐步递减学习率以及自适应学习率等。例如，借助 Adam 算法，能够实现学习率的自动调整，从而有效提升模型性能。

5. 批量归一化

为了加速训练进程并增强模型稳定性，建议在每层网络之后引入批量归一化层，以标准化输入数据。此举不仅能有效提升训练速度，还能显著提高模型的稳定性。

6. 残差连接

为了解决深度神经网络中的梯度消失问题并提升模型性能，可以在深层网络中引入残差连接，以跳过部分层直接传递信息。

神经网络训练与优化是一个复杂且关键的过程。通过精心挑选损失函数、优化算法和正则化技术，能够显著增强模型表现。同时，需警惕过拟合与欠拟合现象，确保模型具备优秀的泛化能力。

5.3　深度学习

深度学习作为机器学习的关键分支，模仿人脑的神经网络结构和功能，通过构建多层神经网络模型，实现对数据的高效处理和特征提取，如图 5-61 所示。相较于浅层学习，深度学习能自动从海量数据中习得更为复杂和抽象的特征表示，因而在众多领域均取得了卓越成效。然而，深度学习亦面临诸如过拟合、梯度消失等挑战和问题。因此，在实际应用中，需针对具体问题选取恰当的模型和算法，并进行充分的训练与优化。

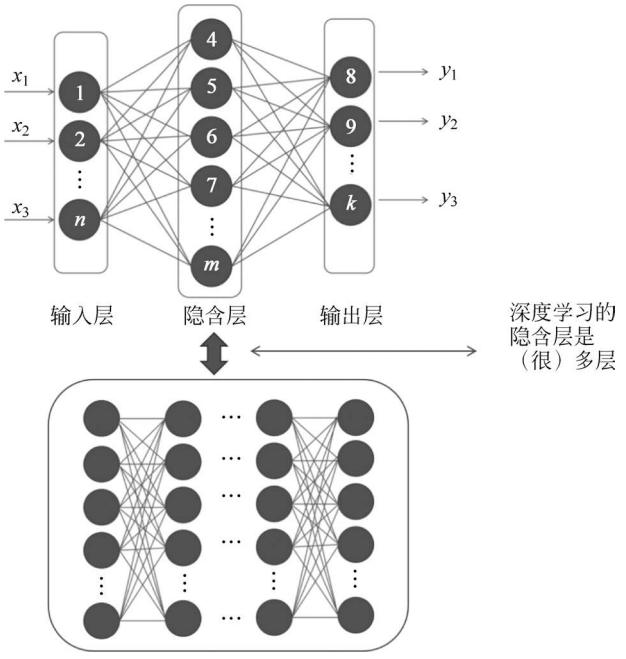


图 5-61　深度学习隐含层的深度模型图

5.3.1　核心技术

1. 神经网络架构

深度神经网络(DNN)是深度学习的基础,涵盖多种网络结构,包括 CNN、RNN、LSTM 和 GRU 等。这些网络结构能够高效处理不同类型的数据,如图像、文本和序列数据。具体而言,CNN 在图像识别任务中表现卓越,而 RNN 则擅长应对自然语言处理和时间序列预测等挑战。

2. 激活函数

激活函数通过引入非线性因素,使神经网络能够更精确地逼近复杂的函数关系。常见的激活函数包括 ReLU、Sigmoid 和 Tanh 等。例如,ReLU 函数在正区间内的导数为常数,这一特性有效缓解了梯度消失问题。

3. 损失函数与优化算法

损失函数用于评估模型预测值与真实值之间的差异,常见的损失函数包括均方误差、交叉熵损失等。优化算法旨在最小化损失函数,常用的优化算法有随机梯度下降(SGD)、Adam 和 RMSprop 等。例如,Adam 算法融合了 Momentum 和 RMSprop 的优势,能够自适应地调节学习率,从而加速收敛并提升模型性能。

4. 正则化与防止过拟合

为防止过拟合,可运用 L1、L2 正则化、dropout 及批量归一化等技术。例如,dropout 技术通过在训练过程中随机剔除部分神经元,降低模型对特定神经元的依赖,进而增强模型的泛化能力。

5. 迁移学习与预训练模型

迁移学习允许将在某一任务上训练好的模型应用于另一相关任务,从而节省时间和计算资源。预训练模型是指在大型数据集上预先训练完毕的模型,可直接用于特定任务或作为新模型的起点进行微调。例如,ResNet 便是一种常用的预训练模型,在图像分类任务中展现出卓越的性能。

5.3.2　应用领域

1. 计算机视觉

深度学习在计算机视觉领域的应用极为广泛,包括图像分类、目标检测、人脸识别及图像分割等多个领域。例如,CNN 在 ImageNet 图像分类竞赛中取得了显著突破,这一进展有力地推动了计算机视觉领域的快速发展。

2. 自然语言处理

深度学习在自然语言处理领域的应用极为广泛，涵盖机器翻译、情感分析、问答系统及文本生成等多个方面。例如，RNN 和 LSTM 在处理序列数据方面展现出显著优势，因此在自然语言处理任务中得到了广泛应用。

3. 语音识别与合成

深度学习在语音识别与合成领域同样取得了显著成效。例如，深度神经网络被广泛应用于声学模型的训练及语音特征的提取；此外，基于深度学习的语音合成技术亦实现了长足进步。

4. 推荐系统

深度学习在推荐系统中的应用日益广泛。例如，协同过滤算法能够利用用户的历史行为数据，精准预测用户的兴趣偏好，从而实现个性化推荐；而基于内容的推荐算法则依据物品的特征信息，进行有针对性的推荐。

5. 强化学习

深度学习与强化学习的融合已成为当前研究的热点方向之一。强化学习是一种依托试错机制来探寻最优策略的机器学习方法；而深度学习则能够通过分析海量数据，自动提炼出有效的特征表示。两者的有机结合，有望在游戏 AI、自动驾驶等前沿领域实现更为显著的突破。

5.3.3 挑战与趋势

1. 数据需求与隐私保护

深度学习依赖于海量的标注数据进行训练，然而在实际应用场景中，获取充足标注数据常常面临诸多挑战。此外，随着数据隐私意识的不断提升，如何有效保障用户隐私也已成为不容忽视的议题。未来，亟须探索更加高效的数据增强技术及先进的隐私保护机制，以应对这些挑战。

2. 模型复杂度与可解释性

深度学习模型通常具备较高的复杂度，难以阐释其内部工作机制及决策过程。这不仅为模型的调试和应用设置了障碍，同时也唤起了对模型可解释性的广泛重视。未来，亟须探索更为简洁且高效的模型结构及解释方法，以提升模型的可解释性与透明度。

3. 泛化能力与鲁棒性

深度学习模型在应对未知数据或干扰时，通常展现出较弱的泛化能力和鲁棒性。这可能导致模型在实际应用场景中产生错误或失效。未来研究需聚焦于开发更为健壮的训练方法和精准的评估指标，以提升模型的泛化能力和鲁棒性。

4. 多模态学习与跨领域应用

随着多媒体数据的广泛普及和跨领域应用需求的不断增长，如何将不同模态的数据有效融合，进行联合学习和推理，已成为一个至关重要的研究方向。未来，亟须探索更为高效的多模态学习算法和跨领域迁移学习方法，以进一步推动深度学习在更多领域的广泛应用和持续发展。

习题 5

1. 什么是机器学习？它和传统编程有什么不同？
2. 你能举一个日常生活中的例子来说明机器学习是如何工作的吗？
3. 在机器学习中，“训练”是什么意思？为什么训练对机器学习模型很重要？
4. 什么是“过拟合”？它是如何影响机器学习模型的？
5. 机器学习中的“特征”是什么？为什么选择正确的特征对构建一个好的模型很重要？
6. 什么是监督学习、无监督学习和强化学习？它们之间有什么区别？
7. 为什么说数据在机器学习中扮演了重要的角色？数据的质和量都对机器学习模型有什么影响？
8. 机器学习有哪些潜在的伦理问题？为什么用户在使用机器学习时需要考虑这些问题？

实训 5

1. 使用一个简单的数据集（如手写数字识别），尝试使用一个在线机器学习平台（例如 Google Colab 或 Kaggle）来训练一个基本的分类模型。记录你的步骤和结果，并解释你是如何评估模型性能的。

2. 选择一个你感兴趣的主题（例如电影评论情感分析），收集相关的数据集，并使用文本处理工具对数据进行预处理。然后，尝试使用一个简单的机器学习算法（如朴素贝叶斯分类器）来对文本数据进行分类。分享你的发现和遇到的挑战。

3. 利用公开的数据，尝试解决一个回归或分类问题（例如房价预测）。使用模型进行训练和预测，并讨论如何改进模型以提高预测准确性。

习题 5