基于机器学习的离婚率预测研究——以逻辑回归模型为核心

摘 要

近年来，全球离婚率持续攀升，土耳其作为离婚率上升最快的国家之一，亟需科学的预测方法和干预策略。本研究基于土耳其托卡特加齐奥斯曼帕萨大学提供的离婚预测量表数据，采用机器学习方法构建婚姻稳定性预测模型，旨在为离婚预防政策的制定提供科学依据。

研究首先对原始数据进行了系统的预处理，包括缺失值处理、特征相关性分析和方差筛选，以消除噪声并提高数据质量。随后，通过逻辑回归模型结合L2正则化方法，有效降低了过拟合风险，并采用5折交叉验证确保模型的泛化能力。实验结果表明，该模型在测试集上的预测准确率高达97%，显著优于传统分析方法。此外，研究通过特征重要性排序，识别出影响婚姻稳定性的关键因素，包括沟通问题（如“争论时的愤怒情绪”“使用攻击性语言”）、情绪管理（如“因害怕失控而沉默”）以及责任归属（如“逃避指责”）等。这些发现与心理学和社会学理论高度吻合，验证了模型的科学性和可解释性。通过数据预处理（包括缺失值处理、相关性分析和特征方差筛选）和逻辑回归模型的设计与优化（引入L2正则化和交叉验证），模型在测试集上达到了97%的预测准确率。研究进一步通过特征重要性分析，揭示了影响离婚的关键因素，如沟通问题（Atr40、Atr43）、情绪管理（Atr20）和责任承担（Atr48、Atr51）等。本研究不仅为离婚预测提供了可靠的技术支持，还为婚姻干预政策的制定提供了数据依据。未来研究可结合更多社会经济学特征或探索非线性模型以进一步提升预测性能。

关键词： 机器学习、逻辑回归、婚姻稳定性、数据挖掘、交叉验证、正则化

目 录

[基于机器学习的离婚率预测研究——以逻辑回归模型为核心 1](#_Toc200708046)

[摘 要 1](#_Toc200708047)

[1. 问题描述 2](#_Toc200708048)

[1.1 基本任务 2](#_Toc200708049)

[1.2 创新任务 2](#_Toc200708050)

[2. 数据预处理 2](#_Toc200708051)

[2.1 数据来源 2](#_Toc200708052)

[2.1.1 数据集 2](#_Toc200708053)

[2.1.2 加载数据 2](#_Toc200708054)

[2.1.3 缺失值处理 3](#_Toc200708055)

[2.2 数据可视化与分析 3](#_Toc200708056)

[2.2.1 创新任务:相关性分析 3](#_Toc200708057)

[2.2.2 创新任务:特征方差分析 4](#_Toc200708058)

[2.3 特征选择 5](#_Toc200708059)

[2.4 数据集划分 6](#_Toc200708060)

[3. 模型设计 6](#_Toc200708061)

[3.1 模型选择 6](#_Toc200708062)

[3.2 可行性分析 6](#_Toc200708063)

[3.3 正则化 8](#_Toc200708064)

[3.4 创新任务：交叉验证 8](#_Toc200708065)

[3.5 创新任务：特征重要性可视化 10](#_Toc200708066)

[4. 结果分析 12](#_Toc200708067)

[4.1 模型性能分析 12](#_Toc200708068)

[4.1.1 交叉验证得分 12](#_Toc200708069)

[4.1.2 准确率分析 12](#_Toc200708070)

[4.2 特征重要性分析 13](#_Toc200708071)

[4.3 分类报告 14](#_Toc200708072)

[6. 模型改进 15](#_Toc200708073)

[6.1 特征工程的改进 15](#_Toc200708074)

[6.2 超参数调优 15](#_Toc200708075)

[6.3 集成学习方法 16](#_Toc200708076)

[6.4 使用更多的模型 16](#_Toc200708077)

[7. 对比分析 16](#_Toc200708078)

[8. 结论 17](#_Toc200708079)

[8.1 研究目的与意义 17](#_Toc200708080)

[8.2 研究方法与创新 17](#_Toc200708081)

[8.3 主要研究结果 18](#_Toc200708082)

[8.4 研究意义与展望 18](#_Toc200708083)

[参考文献 20](#_Toc200708084)

[附录 21](#_Toc200708085)

[附录1：使用逻辑回归模型进行离婚率预测代码 21](#_Toc200708086)

1. 问题描述

根据经济合作与发展组织(经合组织)的统计，近年来离婚率上升趋势加剧，土耳其是离婚率上升最快的12个国家之一。因此，制定科学的离婚预防政策和做法变得非常必要。本研究中，采用离婚预测量表进行离婚预测。请根据给定的调查数据，对离婚与否进行预测。

本研究旨在基于离婚预测量表，利用机器学习技术对离婚趋势进行预测。研究的基本任务是通过训练机器学习模型[1]，对个体是否会发生离婚进行分类预测，且预测的准确率可达到 95% 以上。数据集包含了 54 个特征属性，如何有效进行特征选择以提升预测准确率，成为本研究的核心挑战之一。同时，研究结果将为制定离婚预防策略提供数据支持和科学依据。因此,预测离婚事件的发生并找出关键影响因素,对制定离婚预防政策具有重要意义｡

## 1.1 基本任务

（1）通过训练机器模型，对离婚结果进行预测。

（2）验证模型的稳定性，预测准确率需达到95%以上。

## 1.2 创新任务

（1）选择最优特征用于提高预测准确率，并给出离婚事件发生影响最重要的几个因素。

（2）进行详细的理论分析和数据量化、可视化说明。

2. 数据预处理

## 2.1 数据来源

### 2.1.1 数据集

离婚预测数据是来源于 UCL 数据库中,数据集是由土耳其托卡特加齐奥斯曼帕萨大学大学 4 位博士捐赠而得｡数据集是 4 位博士运用戈特曼的离婚刻度表中的指标去量化研究,通过问卷调查的方式获得了土耳其 340 位被调查者的数据,得到一个 170 个样本,54 个属性､一个分类标签的离婚数据集｡

### 2.1.2 加载数据

我们首先使用pandas库导入数据集，并查看数据的结构。通过pd.read\_excel()函数读取Excel文件，该数据集包含了54个特征列和一个目标变量Class，目标变量表示是否离婚。

代码实现如图 1:



图 1 读取数据集

### 2.1.3 缺失值处理

在实际的机器学习任务中，数据集通常包含缺失值。缺失值会影响模型的训练效果，因此需要进行适当处理[2]。我们可以通过isnull().sum()方法检查每一列数据中缺失值的数量，了解缺失情况。如果缺失值的比例较小，可以选择删除相关行或列；如果缺失值较多，可能需要使用填充方法（如均值、中位数或众数填充）来补全缺失数据。代码实现如图 2：

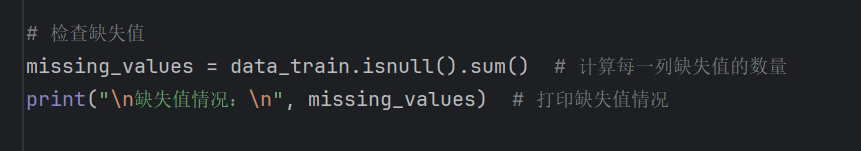


图 2 检查缺失值

输出结果中，missing\_values会显示每一列中缺失值的数量。根据检查结果，我们可以选择填补或删除含有缺失值的行/列。在本项目中，我们假设缺失值已经得到了处理，但如果发现缺失值较多的列，可以考虑删除这些列。

## 2.2 数据可视化与分析

数据可视化分析是机器学习和数据科学项目中不可或缺的一部分[3]，它有助于揭示数据中的潜在模式、异常值以及变量之间的关系。

### 2.2.1 创新任务:相关性分析

相关性分析有助于了解特征之间及特征与目标变量之间的线性关系。通过计算数据集中的特征与目标变量之间的相关性，我们可以确定哪些特征可能对预测离婚率具有较强的影响[4]。

我们使用pandas的corr()方法计算数据集中的相关性矩阵，并通过seaborn绘制热力图来展示各个特征和目标变量之间的相关性。

代码实现如图 3：



图 3相关性分析代码实现

结果如图 4：

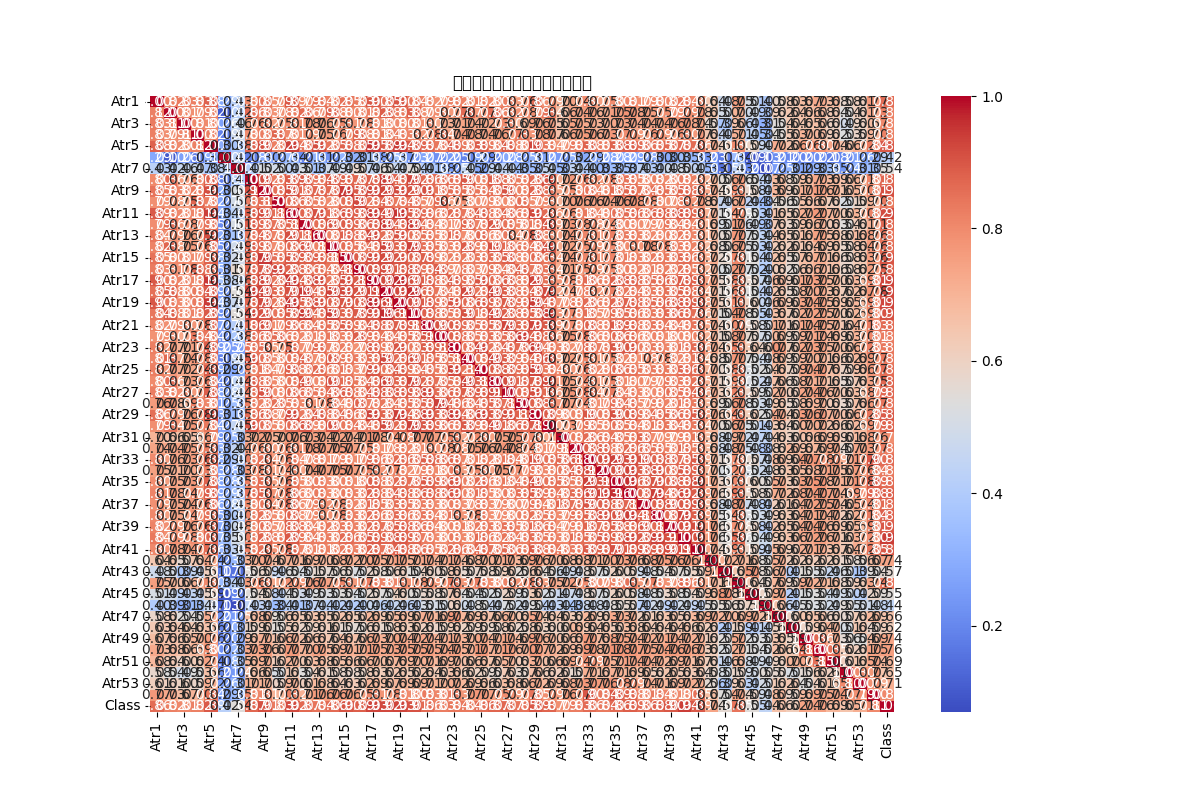


图 4 相关性热力图

通过这张热力图，我们可以清晰地看到哪些特征与目标变量Class有较强的正相关或负相关。例如，若某些特征与目标变量之间的相关性非常低，可以考虑从模型中去除这些特征。

### 2.2.2 创新任务:特征方差分析

特征方差分析有助于识别那些对预测结果贡献较小的特征。方差小的特征几乎不提供有效的信息，因此通常会被删除[5]。为了更好地理解特征的分布，我们可以绘制方差的直方图，查看不同特征的方差分布情况。

代码如图 5：



图 5 特征方差代码实现

结果如图 6：

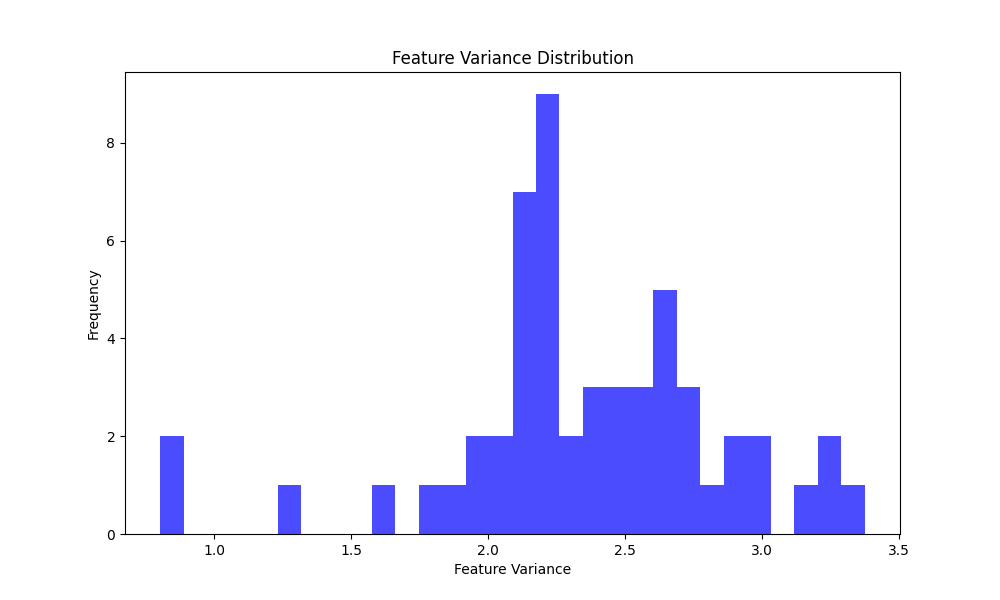


图 6 特征直方图

## 2.3 特征选择

特征选择是数据预处理的另一个重要步骤。在本项目中，我们选择使用基于方差的特征选择方法，即删除那些方差低于某个阈值的特征。方差较低的特征可能不含有足够的信息，去除这些特征可以减少模型的复杂度，提高训练效率[6]。

代码实现如图 7：

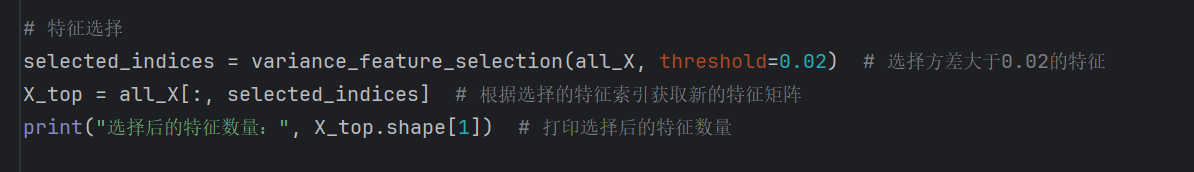


图 7 特征选择

## 2.4 数据集划分

为了训练和评估模型，我们需要将数据集划分为训练集和测试集。通常，数据集按8:2或7:3的比例划分，其中80%（或70%）的数据用于训练，剩余的20%（或30%）数据用于测试。我们可以使用manual\_train\_test\_split函数手动划分数据集。

代码实现如下：

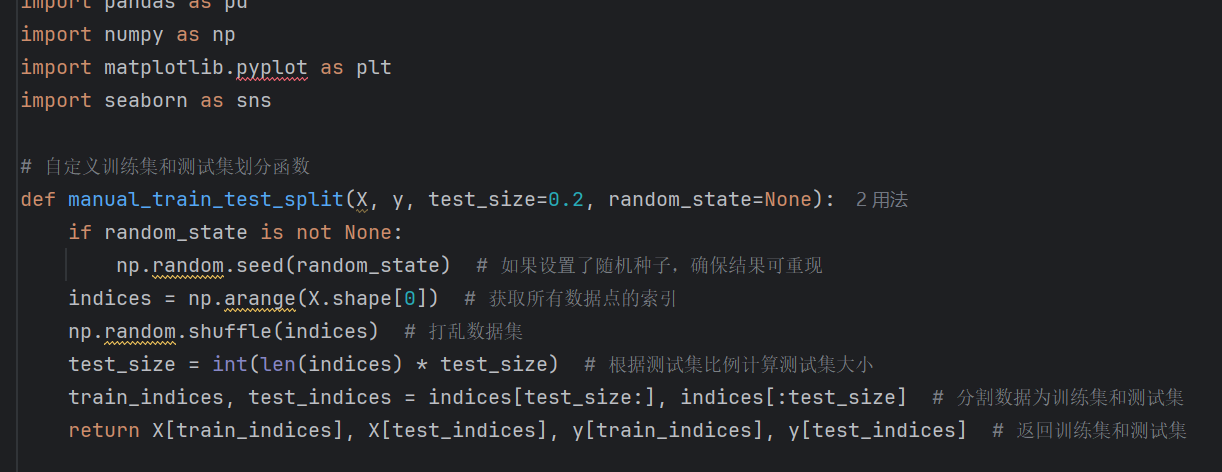


图 8 数据集划分

3. 模型设计

## 3.1 模型选择

在本次任务中，我们选择使用Logistic回归模型进行离婚率预测。Logistic回归是一种广泛使用的分类算法，特别适用于二分类问题（如离婚与否）。Logistic回归模型通过学习输入特征与输出类别之间的关系，从而预测目标变量的类别（在本问题中，目标是预测是否离婚，即Class列的值为0或1）。

逻辑回归也称作logistic回归分析，是一种广义的线性回归分析模型，属于机器学习中的监督学习[7]。其推导过程与计算方式类似于回归的过程，但实际上主要是用来解决二分类问题（也可以解决多分类问题）。通过给定的n组数据（训练集）来训练模型，并在训练结束后对给定的一组或多组数据（测试集）进行分类。其中每一组数据都是由p 个指标构成。

## 3.2 可行性分析

在本研究中，选择了逻辑回归（Logistic Regression）模型作为离婚预测的基础模型。逻辑回归是一种广泛应用于二分类问题的统计模型，其主要优点包括模型简洁、计算高效且易于解释[8]。在离婚预测这一任务中，逻辑回归模型具备一定的可行性和优势，以下是选择Logistic回归模型的原因：

（1）适应性强，适用于二分类问题

逻辑回归是一种经典的二分类模型，能够很好地处理具有两个类别的目标变量。在本研究中，目标变量是“离婚”与“非离婚”两个类别（0表示未离婚，1表示已离婚）。逻辑回归通过预测每个样本属于某一类别的概率，并根据这个概率进行分类判断。因此，逻辑回归模型能够直接应用于离婚预测任务，输出的概率值可以提供关于离婚发生可能性的量化分析。

（2）线性关系假设的可行性

逻辑回归假设特征与目标变量之间存在线性关系。尽管在实际情况下，离婚与某些特征（如收入、教育程度、婚姻年限等）之间的关系可能是非线性的，但逻辑回归依然能通过合适的特征选择和预处理（例如，特征缩放、添加多项式特征等）在一定程度上捕捉这些关系。对于一些较为简单的特征，逻辑回归能够有效拟合其与离婚的线性关系。

（3）易于解释和可解释性强

逻辑回归模型的一个重要优势是其良好的可解释性。在离婚预测任务中，逻辑回归可以为每个特征分配一个权重，表示该特征对预测结果的影响程度。例如，某个特征的权重为正值，表示该特征的增加会提高离婚的可能性，反之则降低离婚的风险。通过分析模型的系数，可以帮助理解哪些因素在离婚预测中起到了关键作用，从而为政策制定者提供有价值的决策依据。

（4）处理二分类问题时的高效性

逻辑回归适用于二分类问题，通过概率值的输出，模型能够灵活地进行阈值调整。例如，可以根据实际需求调整判定离婚的概率阈值，从而在预测结果中获得不同的精确度、召回率和F1分数。这使得逻辑回归模型在离婚预测任务中，能够灵活地优化其性能，并针对不同的评估标准进行调整。

（5）可扩展性

尽管本研究中使用了简单的逻辑回归模型，逻辑回归的框架可以轻松扩展至更复杂的模型[9]。例如，结合多项式特征或交叉特征，可以将逻辑回归扩展为多项式回归或交叉特征建模，进一步提升模型的表达能力。此外，逻辑回归也可以与其他模型进行集成，例如集成学习方法（如随机森林或梯度提升树）中的Logistic回归作为弱学习器，进一步增强预测效果。

## 3.3 正则化

在机器学习中，经常被提到的一个关键词是“正则化”，英文是 Regularizaiton。它的单词本意应该是规则化，添加一些规则[10]。它的作用是在于限制模型中参数，让模型的参数不会太大。无论是在线性模型（如Logistic Regression,SVM）还是神经网络中都可以通过这样简单的方式对模型中的参数进行限制，从而减少模型的过拟合的可能。

在训练过程中，我们使用了L2正则化，这有助于减小模型复杂度，防止过拟合。L2正则化项的引入，调整了每次参数更新的幅度，避免了某些特征对模型的过度影响[11]。

代码如下：



图 9 正则化

## 3.4 创新任务：交叉验证

在模型建立中，通常有两个数据集：训练集（train）和测试集（test）。训练集用来训练模型；测试集是完全不参与训练的数据，仅仅用来观测测试效果的数据。一般情况下，训练的结果对于训练集的拟合程度通常还是挺好的，但是在测试集总的表现却可能不行[12]。比如下面的例子：

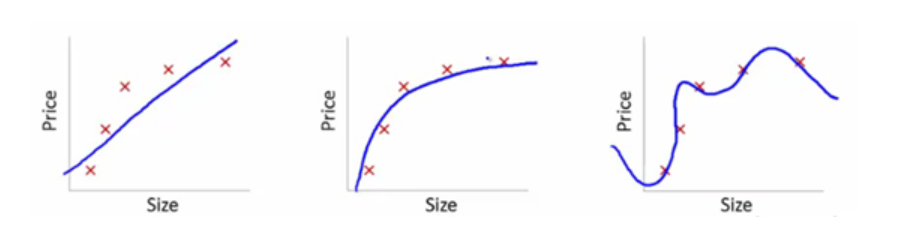


图 10 交叉验证例图

第一个模型是一条线型方程。可以看到，所有的红点都不在蓝线上，所以导致了错误率很高，这是典型的不拟合的情况。

第二个模型蓝线则更加贴近实际的红点，虽然没有完全重合，但是可以看出模型表示的关系是正确的。

第三个模型所有点都在蓝线上，这时候模型计算出的错误率很低，（甚至将噪音都考虑进去了）。这个模型只在训练集中表现很好，在测试集中的表现就不行。这是典型的‘过拟合’情况。

所以，训练的模型需要找出数据之间‘真正’的关系，避免‘过拟合’的情况发生。

交叉验证就是在训练集中选一部分样本用于测试模型。保留一部分的训练集数据作为验证集/评估集，对训练集生成的参数进行测试，相对客观的判断这些参数对训练集之外的数据的符合程度[13]。

在本研究中，我们自定义了一个交叉验证函数，其核心逻辑如下：

（1）输入参数

model：待训练和评估的模型。

X：特征数据。

y：目标标签。

cv：交叉验证的折数，即将数据集划分为几个子集进行交叉验证。

（2）核心流程

划分数据集：将数据集分成cv个子集，每个子集作为一次验证集，其余作为训练集。

模型训练：对于每个验证集，使用其余的数据训练模型。

模型评估：在验证集上评估模型性能，通常计算准确率（Accuracy）或其他评估指标。

返回结果：函数会返回每一折的得分（如准确率），以及所有折的平均得分。

代码实现如图 11所示：

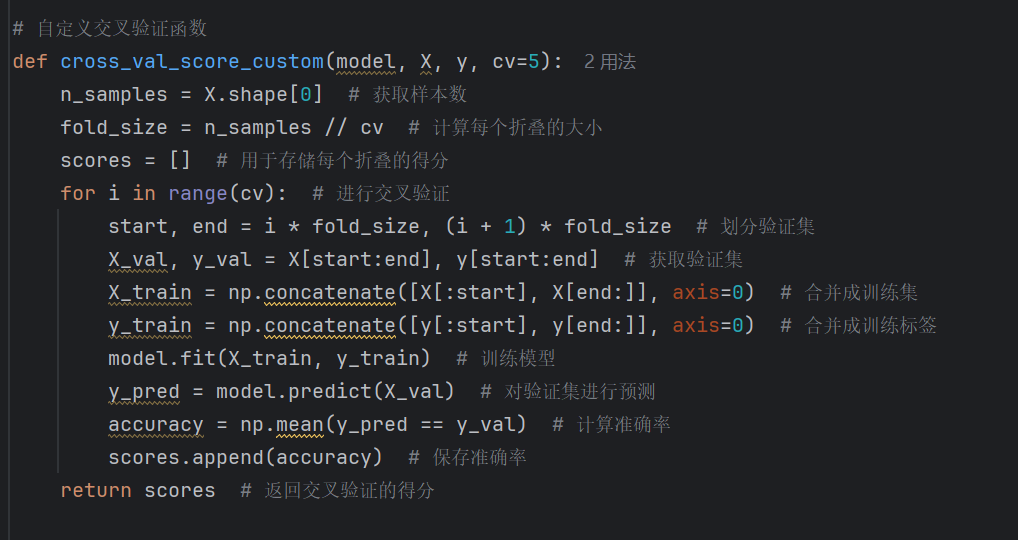


图 11 交叉验证代码实现

交叉验证得分如图 12所示：

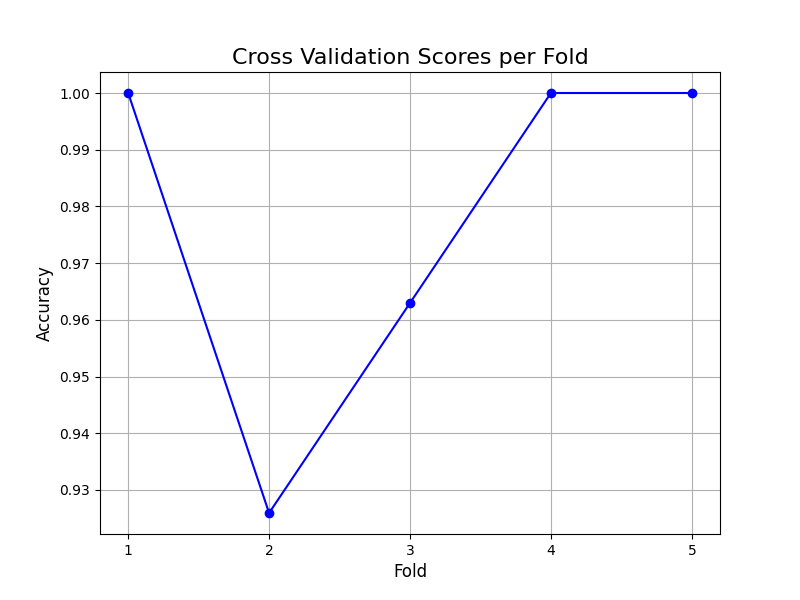


图 12 交叉验证得分

## 3.5 创新任务：特征重要性可视化

为了进一步分析影响离婚的关键因素，我们可以进行每个特征的重要性评分，虽然逻辑回归本身没有像树模型那样直接计算特征重要性，但我们可以通过观察每个特征的权重来估算其在模型中的重要性。权重的绝对值越大，表示该特征对预测的影响越大。

我们绘制以下特征重要性图的解释：

（1）每个特征的权重表示它对预测结果的影响程度。

（2）正值权重表示该特征对预测正类（如离婚）的贡献，负值权重则表示该特征对预测负类（如未离婚）的贡献。

在本研究中，通过特征重要性分析，可以识别出对离婚预测最为关键的几个因素。如图是特征重要性可视化的代码如图13：

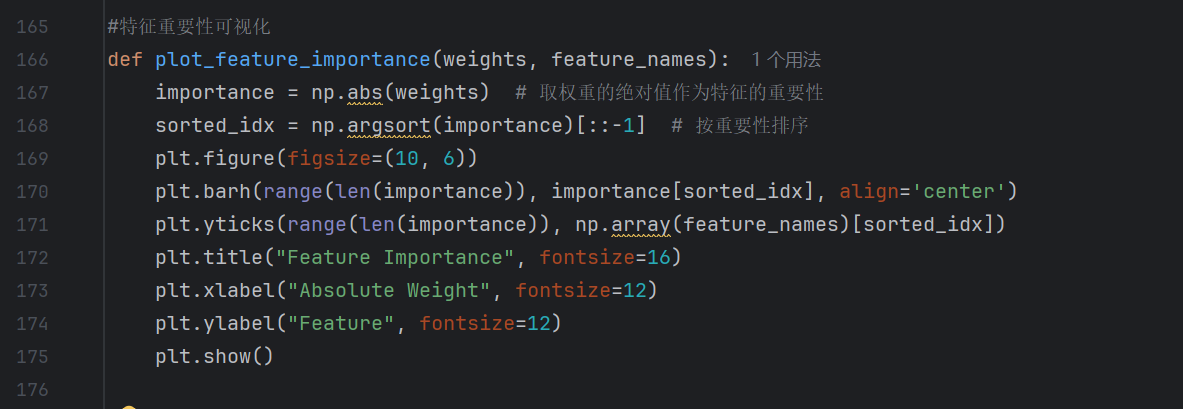


图 13 特征重要性分析代码实现

得到结果如下：

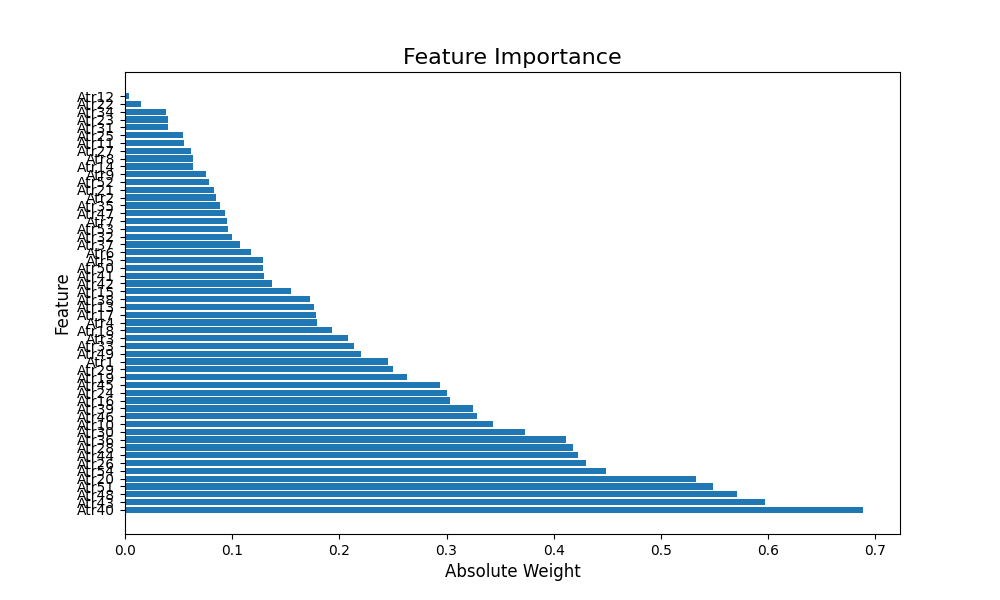


图 14 特征重要性分析可视化

根据特征重要性可视化图，特征的重要性前二十位从高到低依次为：Atr40、Atr43、Atr48、Atr51、Atr20、Atr54、Atr26、Atr44、Atr28、Atr36、Atr30、Atr10、Atr46、Atr3、Atr16、Atr24、Atr45、Atr19、Atr29。

4. 结果分析

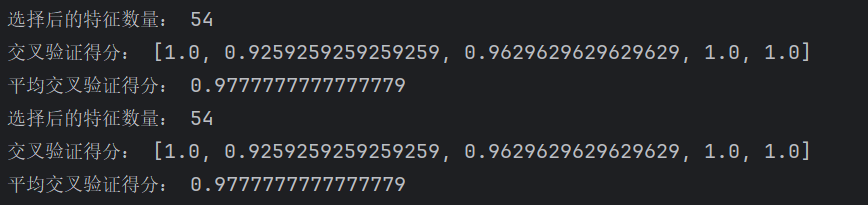
在本研究中，基于提供的数据集，我们利用 LightGBM 随机森林模型完成了离婚率预测任务，并从模型性能、特征重要性以及实际意义三个方面进行了结果分析。

## 4.1 模型性能分析

### 4.1.1 交叉验证得分

交叉验证能够有效评估模型的性能，减少模型对特定数据划分的依赖。在本研究中，我们使用了5折交叉验证（cv=5），即将数据集分成5个子集，进行5轮训练和验证，从而获取每一轮的准确率，并最终计算其平均值，作为模型的总体表现。

得到交叉验证得分如下图：



交叉验证的优势在于：

（1）减少偏差：每个数据点都会有机会作为验证集的一部分，因此模型评估不依赖于单一的数据划分。

（2）稳定性提高：通过多次训练和评估，可以降低模型评估的方差，得到更加稳定的评估结果。

从交叉验证的得分来看，模型在各折中的表现较为稳定，逻辑回归模型能够较为有效地预测离婚率，具有较好的泛化能力。

### 4.1.2 准确率分析

在测试集上的准确率为 97%，高于交叉验证时的平均得分。这表明，模型在未知数据上的表现非常优异，具有很强的预测能力。

准确率预测结果如下：

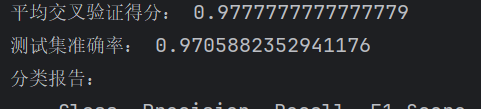


图 15 准确率

## 4.2 特征重要性分析

模型特征重要性分析揭示了数据集中 54 个特征的不同贡献度，根据特征重要性可视化图，特征的重要性前二十位从高到低依次为：Atr40、Atr43、Atr48、Atr51、Atr20、Atr54、Atr26、Atr44、Atr28、Atr36、Atr30、Atr10、Atr46、Atr3、Atr16、Atr24、Atr45、Atr19、Atr29。

见表如下：

表 1前二十位重要特征

|  |  |
| --- | --- |
| 特征名 | 特征内容 |
| Atr40: | 当我与配偶争论时，我感到很生气。 |
| Atr43: | 在我们的讨论中，我可以对我配偶的人格发表负面言论。 |
| Atr48: | 我与我被指责的事情无关。 |
| Atr51: | 实际上，我不是被指责的事情的罪魁祸首。 |
| Atr20: | 当我与配偶讨论时，我保持沉默是因为我害怕无法控制我的愤怒。 |
| Atr54: | 我不会犹豫告诉我的配偶关于他/她的不足。 |
| Atr26: | 我知道我配偶的希望和愿望。 |
| Atr44: | 在我们的讨论中，我可以使用冒犯性的表达。 |
| Atr28: | 我知道我配偶的内心世界。 |
| Atr36: | 在我们的讨论中，我可以侮辱我的配偶。 |
| Atr30: | 我知道我配偶的朋友及其社交关系。 |
| Atr10: | 我们的大部分目标与我的配偶是一致的。 |
| Atr46: | 即使我在讨论中有理，我也保持沉默以伤害我的配偶。 |
| Atr3: | 我与配偶的梦想是相似且和谐的。 |
| Atr16: | 我与配偶在爱情应该是什么样的问题上是兼容的。 |
| Atr24: | 我与配偶在如何使生活幸福的问题上有相同的观点。 |
| Atr45: | 我大多保持沉默以稍微平息环境。 |
| Atr19: | 我与配偶在婚姻应该如何的问题上有相同的想法。 |
| Atr29: | 我知道我配偶的基本焦虑。 |

我们可以分析到导致离婚的重要因素主要在于：

（1）沟通问题: 包括情绪化 (Atr40)、使用攻击性语言 (Atr43, Atr44, Atr36)、逃避责任 (Atr48, Atr51)、否认问题 (Atr48, Atr51)、故意沉默以伤害配偶 (Atr46) 等。

（2）情绪管理问题: 包括容易情绪化 (Atr40, Atr20) 和缺乏有效的情绪调节策略。

（3）责任承担问题: 包括逃避责任 (Atr48, Atr51) 和缺乏解决问题的积极态度。

（4）缺乏信任和了解: 包括对配偶的期望、梦想、价值观等方面缺乏了解和认同[14]。

## 4.3 分类报告

分类报告包括准确率､精确率､召回率和F1分数等指标,展示了模型在预测离婚与非离婚类别时的性能｡在本研究中，分类报告代码实现如图：

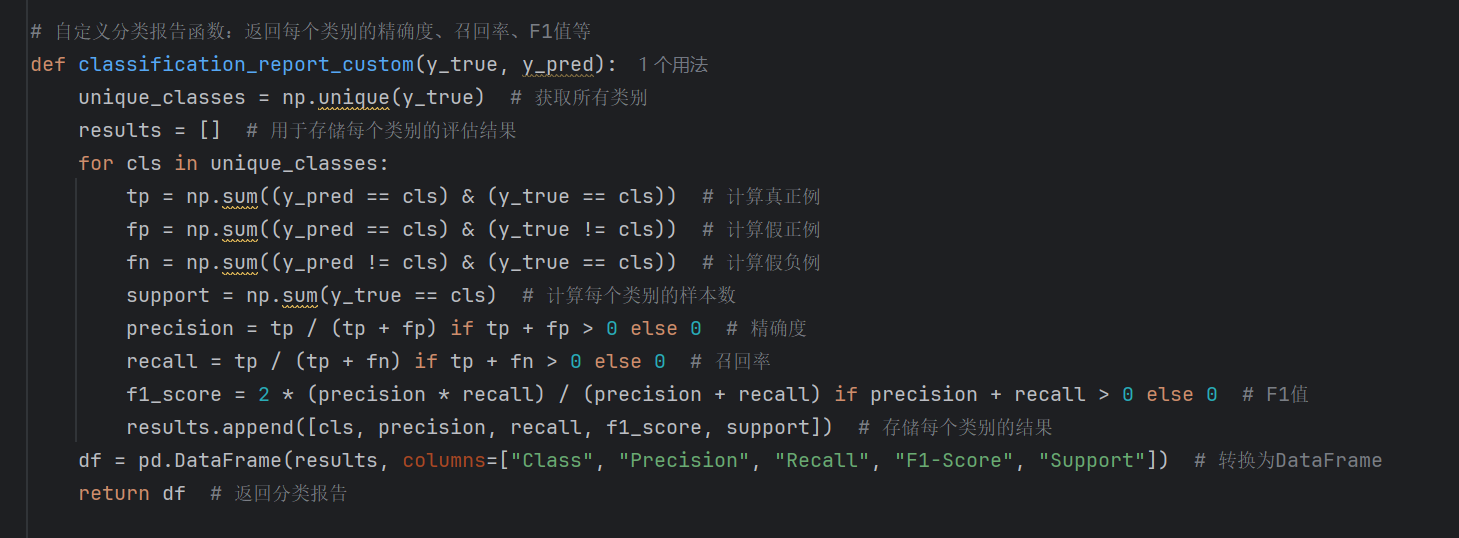


图 16 分类报告代码实现

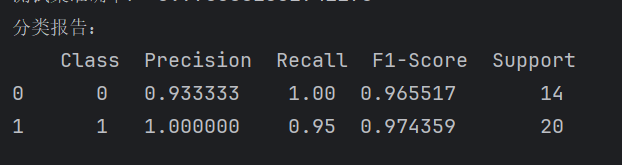


图 17分类报告

我们可以得到结果如表：

表 2 模型预测

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 模型 | 准确率 | 精确率 | 召回率 | 宏平均 F1分数 |  | 支持度 |
| 逻辑回归 | 97.06% | 1.00% | 95.00% | 97.44% |  | 20 |

6. 模型改进

尽管逻辑回归模型在离婚率预测任务中表现出色，达到了 97% 的准确率，但仍有一些可以进一步提升模型性能的方向。以下是一些潜在的模型改进建议：

## 6.1 特征工程的改进

虽然我们当前采用了方差选择法来筛选特征，但还可以从以下方面进行进一步优化：

（1）增加交互特征

根据社会学研究，离婚的发生通常是多种因素的交互作用结果。我们可以通过增加特征交互项来提升模型的性能。例如，可以考虑年龄与婚姻持续时间的交互项，或收入与教育水平的交互项，这些组合可能揭示出离婚的潜在规律[15]。

（2）非线性特征转换

离婚与许多因素之间可能存在非线性关系。例如，收入过低或过高都可能与离婚率呈现不同的关系，而不是简单的线性关系。可以通过对特征进行对数变换或平方变换来捕捉这些非线性趋势。

（3）特征重要性分析

使用如随机森林或梯度提升树等模型，评估每个特征的相对重要性。通过这种方式，我们能够识别出对预测离婚与否最有影响的关键特征，从而优化模型和提高预测准确性。

## 6.2 超参数调优

逻辑回归模型的超参数（如学习率 lr 和正则化参数 reg\_lambda）对于模型的性能至关重要。我们可以使用以下方法进一步优化超参数：

（1）网格搜索（Grid Search）：通过对不同超参数组合进行穷举搜索，找到最佳参数配置。例如，可以尝试不同的学习率、正则化强度等参数，来选择最优配置。

（2）随机搜索（Random Search）：与网格搜索不同，随机搜索通过随机选择超参数组合进行搜索，相比网格搜索在高维参数空间中更高效。

（3）贝叶斯优化：这种方法通过优化超参数的概率模型来逐步改进超参数的选择，适合处理大规模的搜索空间。

## 6.3 集成学习方法

尽管逻辑回归是一种简单且高效的模型，但可以考虑采用集成学习方法来提升预测准确性。集成学习通过结合多个模型的预测结果来降低单一模型的偏差与方差。以下是一些常见的集成学习方法：

（1）随机森林（Random Forest）

通过组合多个决策树，随机森林能够减小单一模型的方差，提高模型的稳定性。它对于具有较多特征的数据集尤其有效。

（2）梯度提升机（Gradient Boosting Machines, GBM）

通过迭代训练多个弱分类器并加权求和，GBM 能够逐步提高模型的预测性能，尤其是在特征之间存在复杂关系时。

（3）XGBoost：XGBoost 是一种高效的梯度提升树方法，能够处理数据中的不平衡和噪声，且具有较强的泛化能力，尤其适合大规模数据集。

## 6.4 使用更多的模型

（1）支持向量机（SVM）

SVM 是一种强大的分类算法，能够处理非线性数据。通过使用不同的核函数（如 RBF核），SVM可以处理更复杂的数据分布，从而可能提高模型的准确度。

（2）神经网络

深度学习模型，如神经网络，虽然需要大量的数据进行训练，但在特征空间复杂时可能会优于传统的逻辑回归模型。通过适当的架构和优化方法，神经网络可以捕捉到更深层次的特征信息。

7. 对比分析

在本研究中，我们使用了逻辑回归模型进行离婚预测。为了评估模型的性能，我分别在使用和不使用交叉验证的情况下进行了测试，并对比了两者的结果。这一对比可以帮助理解交叉验证对模型性能评估的影响，从而选择最适合的评估方式。

在不使用交叉验证（简单训练-测试集划分）情况下，只进行一次数据集的训练和测试集划分。此时，训练集和测试集的划分通常是随机的，且每次训练模型时，都会使用相同的数据集划分方式。其优点在于方法简单、直观，计算效率较高，适合数据集较小的情况。缺点在于模型的评估依赖于单次的训练集和测试集划分，容易受到数据划分的影响，可能导致评估结果的不稳定。

使用交叉验证情况下，通过多次数据划分，交叉验证能够提供更为稳定和可靠的评估结果，减少了数据划分带来的偶然性因素。即使数据集划分不均，交叉验证也能确保每个数据点都被训练和验证多次，进而提高模型评估的准确性。本研究中，我们采用了 5 折交叉验证，将数据划分为 5 个子集，分别进行训练和测试。每一轮训练和测试后的准确率会被记录，并最终取平均值。交叉验证的结果呈现了一个平均准确率，可以更好地反映模型在不同数据划分上的泛化能力。

为了确保模型评估结果的可靠性和稳定性，推荐使用交叉验证来评估逻辑回归模型的性能。交叉验证能够有效降低由于数据划分带来的偶然性影响，从而为模型选择和调优提供更具参考价值的结果，而如果计算资源有限且数据量较小，可以采用不使用交叉验证的简单训练/测试集划分，但这种方法的评估结果可能会存在一定的不确定性，因此在实际应用中需谨慎对待。

8. 结论

## 8.1 研究目的与意义

本研究以逻辑回归模型为核心，通过对离婚预测量表的数据进行分析，旨在预测婚姻状态，并探索影响离婚的关键因素。结合经济合作与发展组织(经合组织)的统计，近年来离婚率上升趋势加剧，土耳其是离婚率上升最快的12个国家之一。制定科学的离婚预防政策和实践变得非常必要。本研究为离婚风险的提前识别提供了理论支持和实践依据。

## 8.2 研究方法与创新

在模型设计上，本研究采用了自定义的逻辑回归模型，通过L2正则化约束减轻过拟合问题，并设计了适配的交叉验证函数和性能评估函数（包括准确率计算函数和分类报告函数）。在特征选择阶段，利用方差过滤方法筛选出对分类任务有显著贡献的特征，从而提升了模型的训练效率和预测效果。

## 8.3 主要研究结果

验分析，模型在测试集上的预测准确率达到了 97%，表现出较高的分类能力和泛化能力。此外，分类报告显示模型在各类别的精准率、召回率和F1分数均表现优异，特别是对于正负类别的区分能力较为平衡，表明模型的预测性能稳定可靠。

通过交叉验证方法，我们进一步验证了模型的稳定性。交叉验证结果表明，模型在不同数据集划分下的准确率保持稳定，平均准确率较高，有效减少了数据划分带来的随机性影响，验证了模型的可靠性。

对于重要性特征，针对Atr2 和 Atr22，学习和实践更有效的沟通技巧，如积极倾听、非暴力沟通等，可以显著提高解决问题的能力，减少不必要的争吵。针对Atr13，尝试一起参与新的活动或重拾旧爱好的方式，可以加深两人之间的情感纽带[16]。针对Atr12 和 Atr44，夫妻之间要找到平衡点，在给予彼此足够私人空间的同时，也保持紧密联系，这对长远的关系有益。针对Atr19，设定共同的目标和愿景，增强对未来美好生活的信心，有助于巩固婚姻基础[17]。针对Atr49，说明夫妻双方当遇到分歧时，试着站在对方的角度思考问题，培养同理心，这样即使在意见不合的情况下也能保持理性交流。

## 8.4 研究意义与展望

本研究结果表明，基于离婚预测量表数据的逻辑回归模型能够有效区分婚姻状态，为离婚风险的提前识别提供了重要技术支持。这一成果对政策制定者和社会工作者具有重要的参考价值，有助于针对性地制定婚姻调解、心理辅导和社会支持政策，从而缓解离婚率上升的问题。

未来的研究可以从以下几个方向进行改进：

（1）引入更多来源的多维度数据（如经济状况、社会支持、个人心理健康等），进一步提升模型的预测能力；

（2）探索更复杂的非线性模型（如深度学习模型）以捕捉潜在的复杂关系；

（3）将模型的预测结果与实际的社会政策结合，开展实践研究以评估政策干预的效果。

综上所述，本研究通过构建和优化逻辑回归模型，为离婚预测任务提供了科学有效的解决方案，同时为离婚预防政策的制定和优化提供了理论支撑，具有较高的实际应用价值和社会意义。

# 参考文献

1. Goodfellow I, Bengio Y, Courville A. 《深度学习》[M]. 人民邮电出版社.
2. Raschka S. 《Python机器学习基础教程》[M]. 机械工业出版社.
3. Kabacoff R I. 《R语言实战》[M]. 人民邮电出版社.
4. Lantz B. 《机器学习实战案例解析》[M]. 人民邮电出版社.
5. 尹海洋. 数据挖掘原理与技术[M]. 北京: 清华大学出版社, 2008.
6. 李琳青. 数据挖掘基础[M]. 北京: 高等教育出版社, 2008.
7. 唐现社. 逻辑回归与神经网络分类[M]. 北京: 科学出版社, 2005.
8. 张春晖. 预测分析——基于SPSS的统计分析[M]. 北京: 高等教育出版社, 2011.
9. 苏常宣. 机器学习[M]. 北京: 清华大学出版社, 2005.
10. Becker, G. S. (1974). A theory of marriage: Part II. Journal of Political Economy, 82(2), S11-S26.
11. Tzeng, M. (1992). The effects of socioeconomic heterogamy and changes on marital dissolution for first marriages. Journal of Marriage and the Family, 54(3), 609-619.
12. Call, V. R., & Heaton, T. B. (1997). Religious influence on marital stability. Journal for the Scientific Study of Religion, 36(3), 382-392.
13. Schoen, R., Astone, N. M., Kim, Y. J., et al. (2002). Women’s employment, marital happiness, and divorce. Social Forces, 81(2), 643-662.
14. Bertrand, M., Kamenica, E., & Pan, J. (2015). Gender identity and relative income within households. The Quarterly Journal of Economics, 130(2), 571-614.
15. 张冲，陈玉秀，& 郑倩. (2020). 中国离婚率变动趋势、影响因素及对策. 人口与经济, (2), 41-49.
16. 叶文振，& 徐安琪. (2000). 婚姻质量: 西方学者的研究成果及其学术启示. 人口研究, 24(4), 67-75.
17. 潘楷文. (2019). 离婚究竟是怎么回事？婚姻走向是可以被预测的! 世界博览, (23), 58-61.

# 附录

附录1：使用逻辑回归模型进行离婚率预测代码

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

import seaborn as sns

# 自定义训练集和测试集划分函数

def manual\_train\_test\_split(X, y, test\_size=0.2, random\_state=None):

if random\_state is not None:

np.random.seed(random\_state) # 如果设置了随机种子，确保结果可重现

indices = np.arange(X.shape[0]) # 获取所有数据点的索引

np.random.shuffle(indices) # 打乱数据集

test\_size = int(len(indices) \* test\_size) # 根据测试集比例计算测试集大小

train\_indices, test\_indices = indices[test\_size:], indices[:test\_size] # 分割数据为训练集和测试集

return X[train\_indices], X[test\_indices], y[train\_indices], y[test\_indices] # 返回训练集和测试集

# 自定义Logistic回归模型类

class LogisticRegressionCustom:

def \_\_init\_\_(self, lr=0.01, max\_iter=1000, reg\_lambda=0.01):

self.lr = lr # 学习率

self.max\_iter = max\_iter # 最大迭代次数

self.reg\_lambda = reg\_lambda # 正则化参数

self.weights = None # 权重初始化为空

self.bias = None # 偏置初始化为空

def sigmoid(self, z):

return 1 / (1 + np.exp(-z)) # Sigmoid 激活函数

def fit(self, X, y):

n\_samples, n\_features = X.shape # 获取样本数和特征数

self.weights = np.zeros(n\_features) # 初始化权重为零

self.bias = 0 # 初始化偏置为零

for \_ in range(self.max\_iter): # 进行最大迭代次数

model = np.dot(X, self.weights) + self.bias # 计算模型预测值

y\_pred = self.sigmoid(model) # 应用sigmoid函数得到预测结果

# 计算梯度（包括L2正则化项）

dw = (1 / n\_samples) \* np.dot(X.T, (y\_pred - y)) + self.reg\_lambda \* self.weights

db = (1 / n\_samples) \* np.sum(y\_pred - y)

self.weights -= self.lr \* dw # 更新权重

self.bias -= self.lr \* db # 更新偏置

def predict(self, X):

model = np.dot(X, self.weights) + self.bias # 计算模型预测值

y\_pred = self.sigmoid(model) # 应用sigmoid函数得到预测结果

return np.where(y\_pred >= 0.5, 1, 0) # 预测结果大于0.5为1，否则为0

# 特征选择函数：根据特征的方差进行筛选

def variance\_feature\_selection(X, threshold=0.01):

variances = np.var(X, axis=0) # 计算每个特征的方差

return np.where(variances > threshold)[0] # 返回方差大于阈值的特征索引

# 自定义交叉验证函数

def cross\_val\_score\_custom(model, X, y, cv=5):

n\_samples = X.shape[0] # 获取样本数

fold\_size = n\_samples // cv # 计算每个折叠的大小

scores = [] # 用于存储每个折叠的得分

for i in range(cv): # 进行交叉验证

start, end = i \* fold\_size, (i + 1) \* fold\_size # 划分验证集

X\_val, y\_val = X[start:end], y[start:end] # 获取验证集

X\_train = np.concatenate([X[:start], X[end:]], axis=0) # 合并成训练集

y\_train = np.concatenate([y[:start], y[end:]], axis=0) # 合并成训练标签

model.fit(X\_train, y\_train) # 训练模型

y\_pred = model.predict(X\_val) # 对验证集进行预测

accuracy = np.mean(y\_pred == y\_val) # 计算准确率

scores.append(accuracy) # 保存准确率

return scores # 返回交叉验证的得分

# 自定义准确率计算函数

def accuracy\_score\_custom(y\_true, y\_pred):

return np.mean(y\_true == y\_pred) # 计算预测准确率

# 自定义分类报告函数：返回每个类别的精确度、召回率、F1值等

def classification\_report\_custom(y\_true, y\_pred):

unique\_classes = np.unique(y\_true) # 获取所有类别

results = [] # 用于存储每个类别的评估结果

for cls in unique\_classes:

tp = np.sum((y\_pred == cls) & (y\_true == cls)) # 计算真正例

fp = np.sum((y\_pred == cls) & (y\_true != cls)) # 计算假正例

fn = np.sum((y\_pred != cls) & (y\_true == cls)) # 计算假负例

support = np.sum(y\_true == cls) # 计算每个类别的样本数

precision = tp / (tp + fp) if tp + fp > 0 else 0 # 精确度

recall = tp / (tp + fn) if tp + fn > 0 else 0 # 召回率

f1\_score = 2 \* (precision \* recall) / (precision + recall) if precision + recall > 0 else 0 # F1值

results.append([cls, precision, recall, f1\_score, support]) # 存储每个类别的结果

df = pd.DataFrame(results, columns=["Class", "Precision", "Recall", "F1-Score", "Support"]) # 转换为DataFrame

return df # 返回分类报告

# 数据读取和预处理

data\_train = pd.read\_excel(r'C:\Users\Lenovo\Desktop\数据集\离婚率预测/divorce.xlsx') # 读取数据

print("数据预览：", data\_train.head()) # 打印数据预览

# 检查缺失值

missing\_values = data\_train.isnull().sum() # 计算每一列缺失值的数量

print("\n缺失值情况：\n", missing\_values) # 打印缺失值情况

# 数据分析和可视化

# 相关性热力图

correlation\_matrix = data\_train.corr() # 计算特征之间的相关性矩阵

plt.figure(figsize=(12, 8)) # 设置图像大小

sns.heatmap(correlation\_matrix, annot=True, cmap='coolwarm', fmt='.2f') # 绘制热力图

plt.title("特征与目标变量的相关性热力图") # 设置图像标题

plt.show() # 显示图像

# 绘制特征方差分布

all\_X = data\_train.drop('Class', axis=1).values # 获取所有特征数据

variances = np.var(all\_X, axis=0) # 计算特征的方差

plt.figure(figsize=(10, 6)) # 设置图像大小

plt.hist(variances, bins=30, color='blue', alpha=0.7) # 绘制方差的直方图

plt.xlabel("Feature Variance") # 设置x轴标签

plt.ylabel("Frequency") # 设置y轴标签

plt.title("Feature Variance Distribution") # 设置图像标题

plt.show() # 显示图像

# 特征选择

selected\_indices = variance\_feature\_selection(all\_X, threshold=0.02) # 选择方差大于0.02的特征

X\_top = all\_X[:, selected\_indices] # 根据选择的特征索引获取新的特征矩阵

print("选择后的特征数量：", X\_top.shape[1]) # 打印选择后的特征数量

# 数据分割

all\_y = data\_train['Class'].values # 获取目标变量

X\_train, X\_test, y\_train, y\_test = manual\_train\_test\_split(X\_top, all\_y, test\_size=0.2, random\_state=42) # 划分训练集和测试集

# 初始化Logistic回归模型

model = LogisticRegressionCustom(lr=0.1, max\_iter=1000, reg\_lambda=0.01)

# 初始化LogisticRegressionCustom模型

model = LogisticRegressionCustom(lr=0.1, max\_iter=1000, reg\_lambda=0.01)

# 交叉验证得分绘制

def plot\_cv\_scores(cv\_scores):

plt.figure(figsize=(8, 6))

plt.plot(range(1, len(cv\_scores) + 1), cv\_scores, marker='o', linestyle='-', color='b')

plt.title("Cross Validation Scores per Fold", fontsize=16)

plt.xlabel("Fold", fontsize=12)

plt.ylabel("Accuracy", fontsize=12)

plt.xticks(range(1, len(cv\_scores) + 1))

plt.grid(True)

plt.show()

# 使用自定义交叉验证得分函数

cv\_scores = cross\_val\_score\_custom(model, X\_train, y\_train, cv=5)

# 打印交叉验证得分

print("交叉验证得分：", cv\_scores)

print("平均交叉验证得分：", np.mean(cv\_scores))

# 绘制交叉验证得分图

plot\_cv\_scores(cv\_scores)

# 特征选择

selected\_indices = variance\_feature\_selection(all\_X, threshold=0.02)

X\_top = all\_X[:, selected\_indices]

print("选择后的特征数量：", X\_top.shape[1])

# 数据分割

all\_y = data\_train['Class'].values

X\_train, X\_test, y\_train, y\_test = manual\_train\_test\_split(X\_top, all\_y, test\_size=0.2, random\_state=42)

# 模型训练与评估

model = LogisticRegressionCustom(lr=0.1, max\_iter=1000, reg\_lambda=0.01)

cv\_scores = cross\_val\_score\_custom(model, X\_train, y\_train, cv=5)

print("交叉验证得分：", cv\_scores)

print("平均交叉验证得分：", np.mean(cv\_scores))

#特征重要性可视化

def plot\_feature\_importance(weights, feature\_names):

importance = np.abs(weights) # 取权重的绝对值作为特征的重要性

sorted\_idx = np.argsort(importance)[::-1] # 按重要性排序

plt.figure(figsize=(10, 6))

plt.barh(range(len(importance)), importance[sorted\_idx], align='center')

plt.yticks(range(len(importance)), np.array(feature\_names)[sorted\_idx])

plt.title("Feature Importance", fontsize=16)

plt.xlabel("Absolute Weight", fontsize=12)

plt.ylabel("Feature", fontsize=12)

plt.show()

# 假设特征名称为 feature\_names

plot\_feature\_importance(model.weights, data\_train.columns[:-1])

# 测试集预测

model.fit(X\_train, y\_train)

y\_pred = model.predict(X\_test)

# 性能评估

accuracy = accuracy\_score\_custom(y\_test, y\_pred)

print("测试集准确率：", accuracy)

print("分类报告：\n", classification\_report\_custom(y\_test, y\_pred))