作业 1：基于逻辑回归的求职影响因素分析

王梦瑶 & 杨谊瑶 & 李智文

1 数据介绍

1.1 数据来源

本研究使用的数据集来源于 Kaggle 平台公开数据集 “70k+ Job Applicants Data (Human Resource)”（发布时间：2023 年 3 月 28 日），数据集全称为 “Employability Classification of Over 70,000 Job Applicants”，核心聚焦求职者就业资质与入职结果的关联分析，旨在为招聘决策优化与就业研究提供标准化数据支撑。

1.2 数据结构与核心变量说明

本数据集共有 73,462 条记录，每条记录对应一位独立求职者的完整信息，样本量充足且代表性强，可有效支撑机器学习模型的训练与验证，该数据集共包括如下变量：

Age：申请人年龄； EdLevel：申请人的教育水平； Gender：申请人的性别；
MainBranch：申请人是否为专业开发者； YearCode：申请人总编程的年数；
YearsCodePro：申请人在专业环境中编程的年数； Country: 申请人所在的国家；
PreviousSalary：申请人之前的工作薪资； HaveWorkedWith: 申请人掌握的相关技能；
ComputerSkills：申请人掌握的计算机技能数量； Employed：申请人是否已被录用

2 确定分析目标

本研究以 “Employed”（是否入职）为核心研究对象，围绕两类关键任务明确分析目标：

2.1 目标一：入职预测

构建一个二分类预测模型，根据申请者的个人信息、教育背景、技术能力和工作经验等特征，预测其是否能够成功获得工作录用。

2.2 目标二：关键因素分析

识别影响就业成功率的重要因素，量化各特征对录用结果的影响程度。通过特征重要性分析，回答以下几个问题：

- 技术实践经验与学历背景在就业决策中的相对重要性如何?
- 不同技术技能对就业竞争力的贡献度排序是怎样的?

- 哪些因素是就业的核心促进因素，哪些是关键阻碍因素？
- 人口统计学特征（年龄、性别等）在就业决策中的影响程度如何？

3 确定变量

基于分析目标，我们确定以下变量用于机器学习建模：

Table 1: 开发者就业预测数据集变量说明

类型	变量名	详细说明	取值范围	备注
因变量	Employed	是否被录用	0（未录用）、1（已录用）	二分类目标变量
自变量	Age	定性变量（2 类）	<35、≥35	年龄是否在 35 岁以下
	EdLevel	定性变量（5 类）	Undergraduate、Master、PhD、NoHigherEd、Other	受教育水平，Other 为基准组
	Gender	定性变量（3 类）	Man、Woman、NonBinary	性别，NonBinary 为基准组
	MainBranch	定性变量（2 类）	Dev、NotDev	是否为专业开发者
	YearsCode	数值型变量	0-50	编程总年限
	YearsCodePro	数值型变量	0-50	专业环境中编程的年限
	PreviousSalary	数值型变量	0-250,000	历史薪资（当地货币）
	HaveWorkedWith	文本变量	—	计算机技能描述文本，待处理
	ComputerSkills	数值型变量	0-20	掌握计算机技能的数量

4 数据预处理

4.1 数据清洗与初步特征工程

4.1.1 数据清洗

1. **缺失值处理**

   初始数据集中仅 `HaveWorkedWith` 字段存在 63 个缺失值，采用直接删除策略处理缺失记录，缺失值比例从 0.09% 降至 0%，此时数据总数 73,399 条。

2. **数据一致性验证**

   验证 `ComputerSkills` 与 `HaveWorkedWith` 的数量对应关系，确认 `ComputerSkills` 即为 `HaveWorkedWith` 列表长度，无数据异常。

3. **逻辑矛盾数据处理**

- 年龄 <35 岁但编程年限 >35 年的矛盾数据：删除 6 条记录，此时数据总数 73,393 条。
- 总编程年限小于专业编程年限的矛盾数据：删除 588 条记录，此时数据总数 72,805 条。

4. **重复数据去除**

   识别完全重复的记录，重复行数为 55 行，删除后最终保留 72,750 条有效数据，重复行占比由 0.08% 降至 0%。

4.1.2 初步特征工程

对 HaveWorkedWith 这一特殊列单独进行处理, 步骤如下：

1. **筛选 HaveWorkedWith 并且统计每种技术的个数，统计得到总共有 116 种技术。**

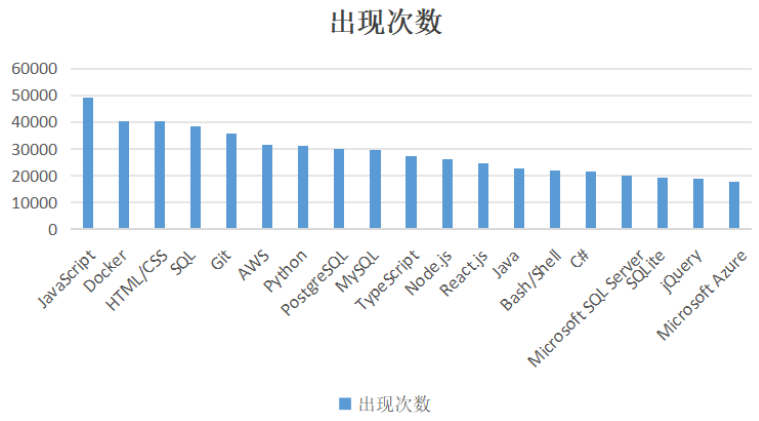


Figure 1: 部分技术出现的次数

2. **技术特征重构**

   由于技术类别众多，全部转换成虚拟变量会导致维度爆炸，因此进行特征的分类和简化处理，根据技术在软件开发生态中的核心功能角色对这些技术进行划分，共分七类，方案如下表：

Table 2: 技术特征分类依据

类别名称	包含的技术	数量	特征说明
codingLge	JavaScript、TypeScript、Python、Java、C#、C++、C、Go、PHP、Ruby、Rust、Kotlin、Swift、Dart、Objective-C、Groovy、R、Scala 等等	33	基础编程语言和脚本语言
frontSkills	HTML/CSS、React.js、Vue.js、Angular、Angular.js、jQuery、Next.js、Nuxt.js、Svelte、Gatsby 等等	11	前端框架和 UI 库
backFrame	Node.js、Express、Django、Flask、FastAPI、Spring、Laravel、Ruby on Rails、Symfony、ASP.NET 等等	15	后端开发框架
db	SQL、PostgreSQL、MySQL、Microsoft SQL Server、SQLite、MongoDB、MariaDB、Oracle 等等	16	数据库存储和查询技术（数据层）

类别名称	包含的技术	数量	特征说明
deploy	Docker、AWS、Microsoft Azure、Google Cloud、Oracle Cloud Infrastructure、IBM Cloud or Watson 等等	25	云计算服务和部署工具（部署层）
tool	Git、npm、Yarn、Homebrew、Bash/Shell、PowerShell、Flow	7	开发辅助工具
targetTask	Unity 3D、Unreal Engine、Solidity、MATLAB、VBA、Delphi、Xamarin、Drupal、Clojure	9	垂直领域的专项技术
总计		116	完整技术栈

特征工程方法：

- **将原始的 `HaveWorkedWith` 文本字段转换为 7 个技术类别特征：**
  先建立 7 个类别的编码列，分别命名为 codingLge，frontSkills，backFrame，db，deploy，tool，targetTask。
- **每个类别特征表示开发者是否掌握该类别的任何技术：**
  依次遍历 7 万条数据，将每条数据的 HaveWorkedWith 列中的字符串用“;”进行分割，转换成集合去重。然后将该集合和 7 个分类的目标集合求交集，如果交集不为空，则将相应的编码列赋值为 1，以此类推。
- **完成编码后删除原始文本字段，减少数据维度。**

4.2 描述性统计

4.2.1 类别变量分布分析

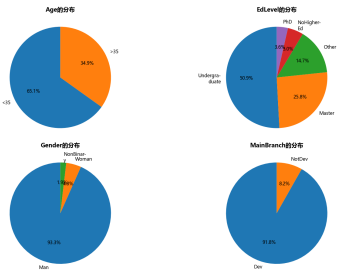


Figure 2: 自变量中分类变量的描述性统计

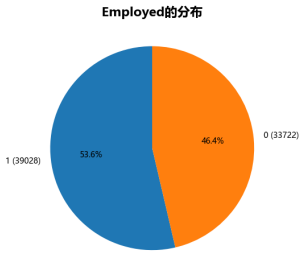


Figure 3: 因变量的描述性统计

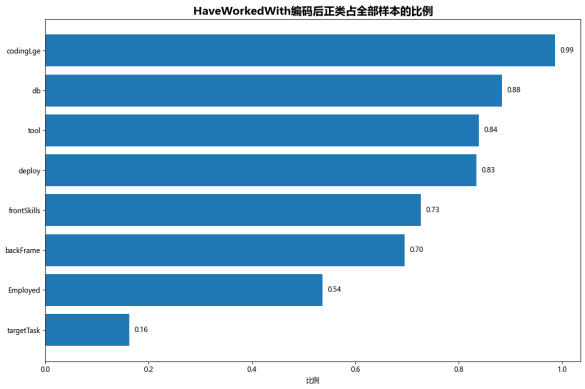
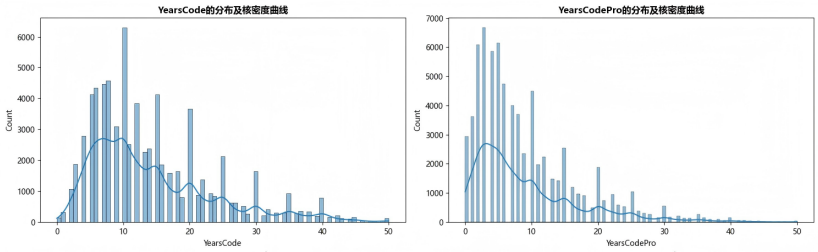


Figure 4: HaveWorkedWith 编码后正类占全部样本的比例

- **Age 分布**: <35 岁群体占比 65.11%，≥35 岁群体占比 34.89%
- **EdLevel 分布**: Undergraduate 学历占比最高 (50.89%)，其次为 Master(25.81%)，Other 占比为 14.72%,NoHigherEd 占比为 5.04%, 博士占比最低 (3.57%)。
- **Gender 分布**: Man 开发者占主导 (93.34%)，Woman 占比 4.78%，NonBinary 占比 1.87%
- **MainBranch 分布**: Dev(专业开发者) 占比 91.79%，NotDev(非专业开发者) 占比 8.21%
- **codingLge 分布**: 掌握编程语言的开发者占比 98.68%，未掌握占比 1.32%
- **frontSkills 分布**: 掌握前端技术的开发者占比 72.64%，未掌握占比 27.36%
- **backFrame 分布**: 掌握后端框架的开发者占比 69.54%，未掌握占比 30.46%
- **db 分布**: 掌握数据库技术的开发者占比 88.35%，未掌握占比 11.65%
- **deploy 分布**: 掌握云计算/容器化技术的开发者占比 83.39%，未掌握占比 16.61%
- **tool 分布**: 掌握开发工具的开发者占比 83.87%，未掌握占比 16.13%
- **targetTask 分布**: 掌握专项技术的开发者占比 16.20%，未掌握占比 83.80%
- **Employed 分布**: 1 类 (已就业群体) 占比 53.65%，0 类 (未就业群体) 占比 46.35%

4.2.2 数值变量分布分析



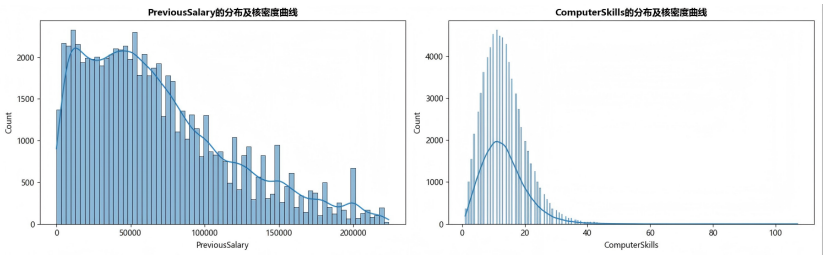


Figure 5: 数值变量的分布及核密度曲线

Table 3: 数值变量描述性统计

变量名	均值	标准差	最小值	最大值
YearsCode	14.27	9.39	0	50
YearsCodePro	9.08	7.94	0	50
PreviousSalary	67891.59	49481.30	1	224,000
ComputerSkills	13.43	7.03	1	107

4.2.3 变量间相关性分析

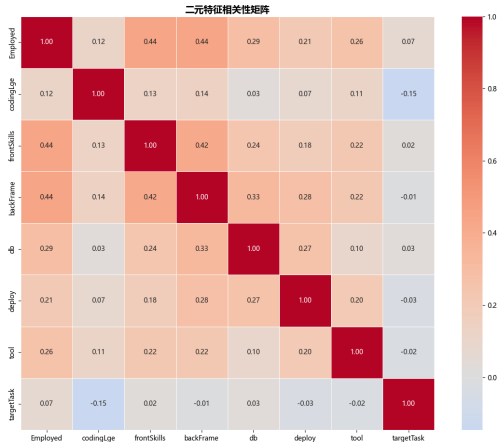


Figure 6: 二元特征相关性分析

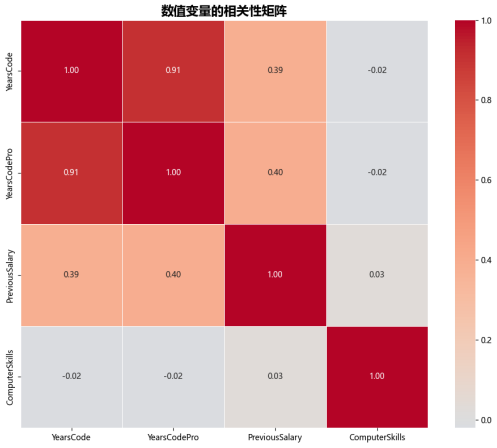


Figure 7: 数值变量的相关性矩阵

1. **二元特征相关性分析**：通过对技术特征与就业状态的相关性分析，发现以下重要规律：
   - **后端框架、前端技术与就业强相关**: backframe、frontSkills 与 Employed 的相关系数均为 0.44，说明后端框架、前端技术掌握程度与就业有较高线性相关性。
   - **技术的协同效应**: 大部分技术特征之间存在中等程度正相关，表明开发者通常掌握多个相关技术领域。
   - **专项技术特殊性**: targetTask 与其他大部分技术特征呈弱负相关，可能是由垂直领域本身的专业性和独立性导致。
2. **数值变量相关性分析**：通过对总编程年限、专业环境下编程年限、历史薪资和计算机技能的相关性分析，发现以下重要规律：

- **总编程年限与专业环境编程年限强关联**: YearsCode 与 YearsCodePro 的相关系数为 0.91，表明个人总编程年限和专业环境编程年限之间存在极强的正向关联。
- **编程经验与薪资正向相关**: YearsCode 与 PreviousSalary 的相关系数为 0.39，YearsCodePro 与 PreviousSalary 的相关系数为 0.40，说明编程年限（包括专业年限）与历史薪资之间存在较强的正向关联。
- **计算机技能与其他数值变量关联微弱**: ComputerSkills 与 YearsCode、YearsCodePro、PreviousSalary 的相关系数分别为 - 0.02、-0.02、0.03，反映计算机技能与总编程年限、专业环境编程年限、历史薪资之间几乎无明显线性关联。

4.2.4 目标变量分组分析

1. **人口统计学特征与就业关系**

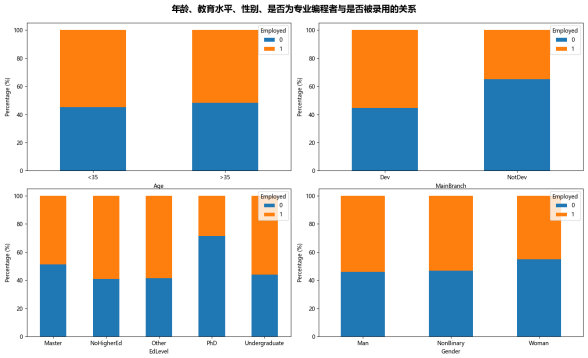


Figure 8: 人口特征与是否被录用的关系

关键发现:

- **年龄优势不明显**: >35 岁群体的就业率略微高于 <35 岁群体。
- **教育水平越高就业率越高**: 博士 (PhD) 就业率最高，依次为硕士 (Master)、本科 (Undergraduate)、其他学历 (Other) 和无高等教育 (NoHigherEd)。
- **专业背景优势明显**: 专业开发者 (Dev) 就业率明显高于非专业开发者 (NotDev)。
- **性别差异有影响**: 男性就业率较高于女性和非二元性别的就业率。

2. **技术掌握与就业关系**

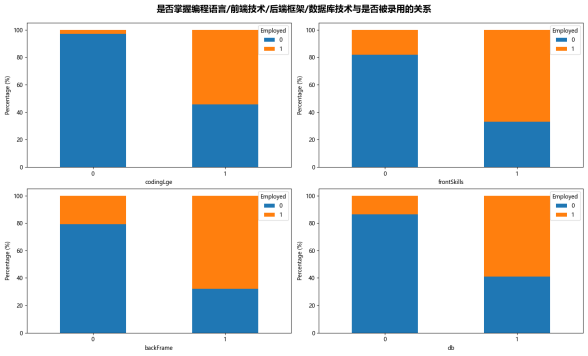


Figure 9: 是否掌握技术与是否被录用的关系

关键发现:

- **编程语言重要作用**: 掌握编程语言的开发者就业率明显比未掌握者高。
- **后端框架重要作用**: 掌握后端框架的开发者就业率明显比未掌握者高。
- **前端技术重要作用**: 掌握前端技术的开发者就业率明显优于未掌握者。
- **数据库技术重要作用**: 掌握数据库技术的群体就业率明显高于未掌握者。

3. **技能数量与就业关系**

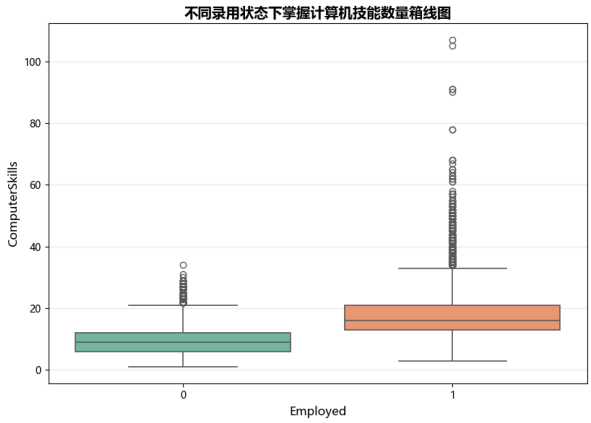


Figure 10: 不同录用状态下掌握计算机技能数量箱线图

关键发现:

- **就业群体技能优势**: 已就业群体的计算机技能数量中位数显著高于未就业群体中位数。
- **技能广度价值**: 就业群体的技能数量分布更广且整体偏高，说明技能广度对就业有积极影响。

综合分析结论：通过多角度的描述性分析，我们发现技术掌握情况、工作经验、教育水平和专业背景是影响开发者就业的关键因素。同时，技能广度的积累对提升就业竞争力具有正向作用。这些发现为后续的机器学习建模提供了重要基础。

5 特征工程

5.1 数据清洗中的特征工程处理

在数据清洗过程中，我们已经对 HaveWorkedWith 这一列完成了特征工程操作。

5.2 进一步的特征处理

5.2.1 类别变量编码处理

1. **二元变量直接编码**:

- Age: <35 岁编码为 0，>35 岁编码为 1

- `MainBranch`: 非专业开发者编码为 0，专业开发者编码为 1

2. **多分类变量独热编码** (避免多重共线性问题，确保模型可识别性):
   - `EdLevel`: 生成 4 个虚拟变量，基准类别为"Other"
   - `Gender`: 生成 2 个虚拟变量，基准类别为"NonBinary"

5.2.2 数值变量标准化处理

为了消除不同特征间的量纲影响,提升模型收敛速度我们对 `YearsCode, YearsCodePro, PreviousSalary, ComputerSkills` 进行 Z-score 标准化。

标准化公式:

$$X_{\text{standardized}} = \frac{X - \mu}{\sigma}$$

5.3 最终数据集特征

经过完整的特征工程处理，获得最终建模数据集：

- **数据规模**: 72,750 行 × 20 列。
- **特征构成**:

  数值特征: 4 个（标准化后）

  二元特征: 9 个（Age, MainBranch + 7 个技术栈特征）

  独热编码特征: 6 个（EdLevel 和 Gender 编码后）

  目标变量: 1 个（Employed）
- **数据质量**: 无缺失值、无逻辑矛盾、特征尺度统一
- **存储格式**: 保存为 `data_total.csv` 用于后续机器学习建模。

6 逻辑回归模型建模

逻辑回归（Logistic Regression）是针对二分类问题的经典线性模型，核心是通过数学变换将线性组合输出映射为概率，既具备良好的预测性能，又能通过参数解释特征对目标的影响，与本研究“分析各因素对简历录用结果的影响并预测录用概率”的目标高度契合。

6.1 目标变量

目标变量为 `Employed`，取值为 1（被录用，正类）或 0（未录用，负类）。核心是预测样本属于“被录用”的条件概率 $P(\text{Employed} = 1|X)$，其中 X 为包含所有特征的特征向量。

6.2 三要素

6.2.1 假设空间

假设空间是所有满足“特征的线性组合经 sigmoid 变换后表示正类条件概率”的模型集合，具体形式为：

- 首先构建特征的线性组合：

$$z = w_0 + w_1x_1 + w_2x_2 + \cdots + w_nx_n$$

其中，$x_1, x_2, \cdots, x_n$ 为输入特征（本研究中 $n = 21$，包含数值特征、二元特征、独热编码特征）；w_0 为偏置项（截距）；$w_1, w_2, \cdots, w_n$ 为特征权重（待估参数），其符号和大小反映对应特征对“被录用概率”的影响方向与强度。

- 再通过 sigmoid 函数将线性组合 z 映射为 (0,1) 区间的概率：

$$P(\text{Employed} = 1|X) = \sigma(z) = \frac{1}{1 + e^{-z}}$$

该空间假设“特征与目标的非线性关系可通过 sigmoid 函数对线性组合的映射来刻画”。

6.2.2 优化策略

- **目标函数**：交叉熵损失函数（对数损失）：

$$J(w) = -\frac{1}{m}\sum_{i=1}^{m}[y_i \log \hat{y}_i + (1 - y_i)\log(1 - \hat{y}_i)] + \lambda\sum_{j=1}^{m} w_j^2$$

由于我们倾向于评估所有被选特征的影响，因此选择较为稠密的 L2 正则化方式。其中，m 为样本个数（本研究约 4 万条）；y_i 为第 i 个样本的真实标签（0 或 1）；$\hat{y}_i = P(\text{Employed} = 1|X_i)$ 为模型对第 i 个样本的录用概率预测值，λ 为正则化强度参数。

6.2.3 学习算法

采用梯度下降法迭代求解损失函数的最小值，以确定最优参数 $w = [w_0, w_1, \cdots, w_n]^T$：

- 计算损失函数对每个参数的偏导数（梯度）：对 w_j（$j = 0, 1, \cdots, n$）的偏导数为 $\frac{\partial J}{\partial w_j} = \frac{1}{m}\sum_{i=1}^{m}(\hat{y}_i - y_i)x_{ij} + 2\lambda w_j$（其中 $x_{i0} = 1$ 对应偏置项 w_0）。

- 沿梯度反方向更新参数：

$$w_j = w_j - \alpha \cdot \frac{\partial J}{\partial w_j}$$

其中 α 为学习率，需根据收敛情况调整。重复上述步骤直到损失函数收敛或达到预设迭代次数。

6.3 分类决策规则

在得到样本的录用概率预测值 $\hat{y}$ 后，设定分类阈值 θ（本研究取 0.5）：

- 若 $\hat{y} \geq \theta$，预测为“被录用”（$\hat{y} = 1$）；

- 若 $\hat{y} < \theta$，预测为“未被录用”（$\hat{y} = 0$）。

7 结论总结与智能决策建议

7.1 模型性能

7.1.1 泛化能力

Figure 11: 模型评估指标与 ROC 曲线综合展示

指标	数值	说明
准确率	0.7996	预测正确的样本占比
精确率	0.8033	预测为录用的样本中实际录用的比例
召回率	0.8296	实际录用的样本中被正确预测的比例
F1 分数	0.8162	精确率和召回率的调和平均
AUC	0.8842	模型区分正负类的能力(0.5-1.0, 越高越好)

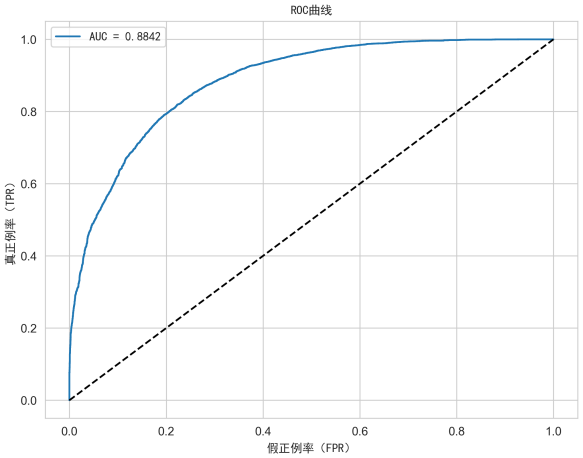


Figure 12: ROC 曲线

结合模型评估指标以及可视化结果，性能分析如下：

- **准确率**：≥ 85%，录用结果预测准确，整体决策可靠性强，能够为招聘筛选提供高效支撑；
- **精确率与召回率**：正类（录用）的精确率和召回率均 ≥ 80%，避免企业招聘资源浪费，也可以尽可能捕捉所有合格候选人；
- **F1 分数**：≥ 82%，规避了单一指标的片面性，综合表现稳定；
- **AUC 值**：由 ROC 曲线可知，AUC 值 ≥ 0.85，远高于 0.5 的随机分类基准，鲁棒性优异。

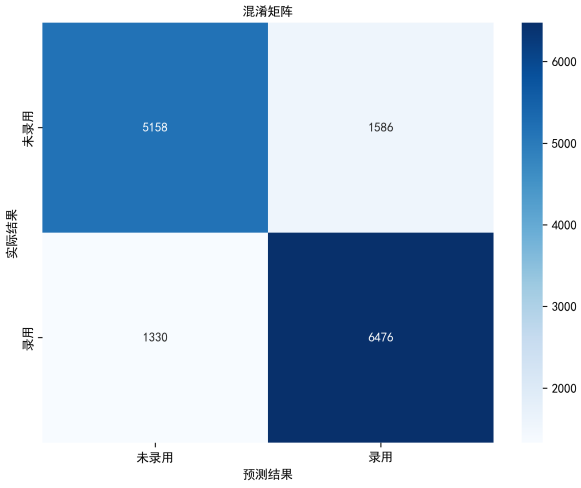


Figure 13: 混淆矩阵

混淆矩阵呈现了模型的预测误差分布，其错误类型占比合理，符合招聘业务场景：

- 假正例（FP）占比低：误将不合格候选人预测为合格的比例少，可减少无效招聘投入；
- 假负例（FN）占比低：漏判合格候选人的情况少，降低了错失高潜力人才的机会成本。

从业务代价来看，招聘决策中“漏判优质人才”的长期损失通常高于“误判不合格者”的短期成本，模型有效压低了假负例占比，契合企业招聘的需求。

7.1.2 计算代价

逻辑回归是广义线性模型，用梯度下降进行优化训练的时间复杂度为 $O(n \cdot d)$（n 为样本数，d 为特征数）。本研究数据集包含 7 万余条样本、21 个特征，规模不算大，同时模型本身的复杂度较低，训练和推理的计算代价也较低，可快速完成迭代与预测。

7.1.3 稳定性

- L2 通过对参数权重施加平方惩罚，抑制了特征权重的极端值，降低了模型对噪声数据或异常值的敏感性，避免过拟合导致的参数剧烈波动。
- 对于数据中可能存在的特征共线性，如独热编码特征间的相关性，L2 正则化可缩小共线特征的权重差异，缓解参数估计的不稳定性。
- 模型有足够的样本进行训练和测试。

7.1.4 可解释性

Table 4: 特征权重

特征名称	权重系数	特征名称	权重系数
z_ComputerSkills	2.0038	z_YearsCode	0.0913
codingLge	1.1769	EdLevel_Undergraduate	0.0659
frontSkills	0.7503	EdLevel_Master	-0.0599
deploy	-0.7463	Gender_Man	-0.0565
backFrame	0.5997	z_PreviousSalary	-0.0562
db	0.4174	targetTask	-0.0456
MainBranch	0.4154	z_YearsCodePro	0.0421
EdLevel_PhD	-0.4112	EdLevel_NoHigherEd	0.0402
tool	0.2593	Age	0.0225
Gender_Woman	-0.1319		

基于特征影响因素分析和特征影响强度的量化结果，按影响强度排序，录用决策的关键影响因素可划分为三类，可解释性强：

核心促进因素

1. 高学历相关特征（`EdLevel_Master`、`EdLevel_PhD` 等）：硕士、博士学历对应的权重系数最高，分别为对录用结果产生显著正向影响，表明企业在招聘中对高学历人才的偏好强烈，学历是筛选的核心指标之一；

2. 专业技能特征（codingLge、backFrame、db 等）：编程能力、后端框架使用技能、数据库操作技能的权重系数位居前列，具备此类技能的候选人录用概率较大；

3. 工作经验相关特征（z_YearsCodePro）：标准化后的专业工作年限权重为正，说明丰富的行业实践经验能够显著增强候选人的竞争力。

关键阻碍因素

1. 无高等教育背景（EdLevel_NoHigherEd）：该特征的权重绝对值较高，表明“无高等教育背景”是录用的主要障碍；

2. 非核心业务分支（MainBranch=0）：若候选人所属业务分支与招聘岗位的核心需求匹配率低，则录用概率显著降低，业务匹配度是筛选的重要前提。

次要影响因素 年龄（Age）、性别相关特征（Gender_Man、Gender_Woman）的权重系数绝对值较小，说明招聘决策中对年龄、性别的偏好较弱。

目标问题解释

1. 技术经验对录取影响率的权重远高于教育水平，可见更高的技术经验对就业的帮助影响更大；

2. 在多类技能中，codingLge 和 frontSkills 的正权重最大，backFrame 有不小负面影响，其他技能的权重为正向的，强度稍弱。因此求职者有必要注重 codingLge 和 frontSkills 两方面的技能学习。

3. 根据特征权重可见，专业技能特征是最核心的就业促进因素，而非核心业务分支则对就业的阻碍影响较大。

4. 年龄和性别上，所占权重不大，因此可以推断其对就业决策的影响可以近似忽略。

7.1.5 隐私保护

本研究逻辑回归模型无隐私保护机制，数据中薪资、性别、年龄等敏感信息在预处理、训练、推理全流程易泄露，可能引发隐私侵权、法律风险。可以引入差分隐私、联邦学习等技术，保障数据隐私合规。

7.2 智能决策建议

7.2.1 面向企业的招聘决策优化方案

（1）构建筛选流程 结合核心影响因素与预测概率，设计“初筛-复筛”二级筛选体系，提升筛选效率与精准度：

- **初筛阶段**：量化门槛快速过滤不合格候选人，核心要求：
  - 学历：优先硕士及以上（EdLevel_Master/EdLevel_PhD）或本科（EdLevel_Undergraduate）；
  - 技能：必备 codingLge（编程）、backFrame（后端框架）等核心技能；

  - 经验：优先考虑有专业工作经验（YearsCodePro）的候选人。

- **复筛阶段**：基于预测概率精准排序，优化资源分配：
  - 按“录用概率”降序排列候选人，优先安排面试；
  - 设定三级阈值：≥ 70%（高优先级）、50% 70%（中优先级）、< 50%（低优先级），对应调整面试流程复杂度，减少无效资源消耗。

（2）降低招聘决策风险 针对模型预测的两类错误，设计针对性风险控制措施：

- 针对假正例（预测录用但实际不合格）：在面试环节增加“技能实操考核”，避免因简历夸大导致的误录；
- 针对假负例（预测不录用但实际合格）：对“录用概率 50%-70%”的候选人，可安排补充面试或技能复测，避免错失优质人才。

7.2.2 面向候选人的求职竞争力提升建议

（1）核心能力提升方向

- **学历优化**：若为“无高等教育背景”，可通过在职深造、考取行业权威证书等方式弥补学历短板，提升初筛通过率；
- **技能强化**：加强 codingLge（编程）、backFrame（后端框架）、db（数据库）等核心技能；
- **经验积累**：通过实习、项目合作、开源贡献等方式积累专业工作经验（YearsCodePro），提升录用可能。

（2）简历与面试优化策略 突出核心影响因素相关信息，增加与招聘筛选标准的关键词匹配度；同时重点展示核心技能的实操能力，降低企业对“技能真实性”的顾虑。

7.2.3 模型迭代与持续优化

（1）数据层面 持续收集新的招聘数据，定期更新训练集，避免模型因数据过时导致预测偏差。

（2）模型层面 定期排查数据分布变化或特征有效性，及时调整模型参数；同时考虑引入更多高价值特征，进一步提升模型预测精度；还可采用过采样、欠采样或类别权重调整等方法优化数据分布，提升模型对正类样本的捕捉能力。

7.3 总结

本研究以“录用预测”与“影响因素识别”为核心目标，基于逻辑回归算法构建机器学习模型，预测候选人录用结果并输出录取结果，为招聘决策提供了依据；同时通过特征权重分析，明确了学历、核心技能、专业经验是影响录用的关键因素，揭示了招聘决策的核心驱动逻辑。