基于 SMS 垃圾邮件分类的 Logistic 回归模型研究

摘要

本文针对 SMS Spam Collection 数据集，构建并优化了 Logistic 回归模型，用以实现垃圾短信的自动分类。通过 TF-IDF 对文本数据进行特征工程，结合 5 折交叉验证和网格搜索的超参数调优，对模型进行了评估和优化。最终模型的准确率达到了 0.96，具备较高的分类性能。

关键词：　Logistic 回归　TF-IDF 特征工程　网格搜索　交叉验证

1. 数据集概览

本文使用了 SMS Spam Collection 数据集，共包含 5572 条短信样本，分为两类：正常短信（ham）和垃圾短信（spam）。通过清洗无效数据，并将短信类别映射为 0（ham）和 1（spam）后，得到的样本数量如下：

正常短信：4825 条

垃圾短信：747 条

表 1 数据基本结构

Column	Non-Null Count	Dtype
label	5572	object
message	5572	object
Unnamed: 2	50	object
Unnamed: 3	12	object
Unnamed: 4	6	object

2. 数据清洗与预处理

在进行文本分类模型的训练之前，必须对原始数据进行清洗和预处理，以确保数据质量并提高模型性能。本文使用的数据集来源于 Kaggle 数据集的

spam.csv 文件，包含两列信息：短信的类别（ham 或 spam）以及短信的内容。为了有效处理文本数据并为后续的模型训练做准备，我们对数据进行了以下几项关键步骤的处理。

2.1 数据导入与初步分析

首先，我们加载了原始数据并对数据集进行了基本的检查。数据集包含两列信息，其中 v1 列表示短信的类别，v2 列则包含短信的内容。在数据预处理中，我们将 v1 列中的类别进行了重新映射，将 ham 类别标记为 0，表示正常短信；将 spam 类别标记为 1，表示垃圾短信。初步分析显示，数据集中的类别分布存在较为明显的不平衡现象，其中 ham 类别的短信数量显著高于 spam 类别的短信。为了解决这一问题，并提高垃圾短信分类模型的表现，我们根据类别的样本数量对类别权重进行了调整。这一调整有助于平衡不同类别的影响，进而使模型在处理垃圾短信分类任务时更加合理和精准。

2.2 数据清洗

文本数据通常包含各种噪声，例如非字母字符、数字以及多余的空白字符等，这些噪声可能对模型性能产生负面影响。因此，数据清洗在文本预处理中具有重要作用。在本研究中，我们对每条短信内容进行了以下几项处理：

1.去除非字母字符：使用正则表达式移除了所有非字母字符，旨在消除特殊符号对模型训练的干扰。

2.去除数字：通过正则表达式删除了文本中的数字，确保模型在训练过程中专注于文字内容，而非数字信息。

3.统一转换为小写：为了消除大小写不一致可能带来的影响，所有文本均转化为小写形式。

清洗后的数据如下图所示：

表 2　清洗后文本长度描述数据

统计量	值
Count	**5572.00**
Mean	**15.680**
Std（标准差）	**11.386**
Min（最小值）	**0.000**

25%（第一四分位）	**7.000**
50%（中位数）	**12.000**
75%（第三四分位）	**23.000**
Max（最大值）	**190.000**

该数据集包含 5572 条清洗后的文本消息。文本消息的平均长度为 15.68 个词，说明大多数消息的词数集中在 15 到 16 之间。标准差为 11.386，表明消息长度分布较为分散，存在一定的波动性，其中包含部分较短或较长的消息。第一四分位数、第二四分位数（中位数）、第三四分位数分别为 7、12 和 23 个词，表明消息长度主要集中在 7 至 23 个词之间。此外，最短的消息长度为 0，最长的消息达到 190 个词，揭示了数据集中存在少量极端的短或长消息。总体来看，该数据集中的大多数消息较短，长度分布具有较高的离散性。

通过这些清洗步骤，我们有效地去除了文本中的噪声，使得后续的特征提取更加准确和可靠。清洗后的数据提供了更加规范化的文本内容，有助于提升模型的训练效果。

2.3 数据划分

为了评估模型的泛化能力并确保其在未见过数据上的表现，本研究将数据集划分为训练集和测试集，其中 80% 的数据用于训练，20% 的数据用于测试。此划分保证了模型能够在一定范围内进行有效学习，并通过测试集的评估，验证其在实际应用中的稳定性和性能。

此外，为进一步提高模型的可靠性，并减少由于数据划分偶然性带来的影响，本文还采用了五折交叉验证（5-fold cross-validation）。在五折交叉验证过程中，数据集被随机分为五个子集，每次使用其中四个子集进行训练，剩余的一个子集用于测试。该过程重复五次，每个子集都轮流作为测试集，从而确保每个样本都有机会作为测试数据参与评估。通过交叉验证，可以更全面地评估模型的性能，减少过拟合的风险，提升模型的泛化能力。

2.4 数据探索与可视化

为了更好地理解数据，本文使用词云对清洗后的文本数据进行了可视化。词云是一种通过词频展示单词大小的可视化方式，能够直观地反映出文本中高频词

汇的分布。通过生成词云图，可以快速识别垃圾短信和正常短信中的常见词汇，为后续的模型训练提供进一步的分析支持。词云显示了数据集中频繁出现的词汇，有助于识别文本的主题和结构特征。词云图结果如下：

图 1　清洗后的词云图

如图 1 所示，本文通过词云展示了短信数据集中各单词的频率分布情况。在该图中，单词的大小与其在数据集中出现的频率成正比，较大的词汇表示其在数据中出现频率较高。从图中可以观察到，诸如“call”、“now”、“will”、“free”、“txt”等词汇占据了显著位置。这些词汇与短信的交流特点密切相关，尤其是在垃圾短信中，它们往往频繁出现，旨在引起用户的关注或诱导特定行为。

该词云图不仅能够直观地揭示数据集中的重要词汇组成，还为后续模型训练和特征选择提供了有价值的参考依据。通过对高频词汇的分析，研究者可以更深入地把握文本数据的主题和结构特征，从而有助于提高垃圾短信分类模型的效果。

3.　特征工程

文本数据通常是非结构化的，无法直接输入机器学习模型进行处理。因此，特征提取成为文本处理中的重要步骤，将文本转化为数值特征是机器学习模型应用的前提。在本研究中，采用了 TF-IDF（Term Frequency-Inverse Document Frequency）方法来提取文本特征。TF-IDF 方法通过计算词频（TF）和逆文档频率（IDF），有效衡量每个单词在文本中的重要性，并抑制常见词汇（如“the”、“is”等）的干扰，从而减少这些词汇对模型训练的负面影响。使用 TfidfVectorizer

对短信数据进行转换，并去除停用词，以提高模型性能。通过该方法，每条短信被转化为一个高维稀疏向量，向量中的每个元素表示该短信中对应单词的加权频率。

尽管 TF-IDF 方法在许多文本分类任务中表现良好，其主要局限在于无法捕捉单词之间的深层语义关系。如图 1 所示，部分单词被错误拆分为单独的字母（如“m”、“s”、“u”），并且该方法无法考虑上下文信息。在更为复杂的自然语言处理任务中，单词的语义信息变得尤为重要。为了弥补这一不足，近年来一些更为先进的特征提取方法，如 Word2Vec 和 BERT，已被广泛应用于文本分析。Word2Vec 通过将单词映射到稠密的低维向量空间中，能够捕捉单词之间的语义关系；而 BERT 作为一种基于 Transformer 架构的预训练语言模型，能够生成更加语境化的词向量表示，并在多种任务中展现出优异的性能。然而，相较于 TF-IDF，这些方法要求更多的计算资源，并且训练过程较为复杂。

本研究中，选择 TF-IDF 作为文本特征提取的方法，主要是基于其计算效率高、实现简单且能够较好地处理大量短文本数据（如短信）的特点。与 Word2Vec 和 BERT 相比，TF-IDF 的优势在于其较低的计算开销和更易调试的特性。尽管 TF-IDF 无法捕捉单词间的深层语义关系，但对于垃圾短信分类任务而言，简单且高效的 TF-IDF 方法能够提供足够的性能。在资源和时间有限的条件下，TF-IDF 依然是一个具有较高性价比的选择。未来的研究可以进一步探索 Word2Vec 和 BERT 等更复杂的特征工程方法，以期进一步提升模型的性能。

4. 模型训练与评估

在垃圾短信分类任务中，我选择了逻辑回归模型。逻辑回归是一种线性分类模型，适用于解决二分类问题，因此在垃圾邮件分类中表现良好。逻辑回归的主要优点是模型结构简单，训练和预测速度较快，适合大规模数据集，并且可以通过调整阈值来控制分类结果的精确性和召回率的平衡。相比之下，支持向量机、随机森林等复杂模型尽管可能会提供更高的准确率，但它们的训练时间和计算资源消耗较大，不太适合快速响应的应用场景。

4.1 逻辑回归模型

逻辑回归的核心思想是将输入特征通过一个线性组合映射到一个概率值。对

于二分类问题，假设输入特征为 $\mathrm{x}=(\mathrm{x}_1,\mathrm{x}_2,\ldots,\mathrm{x}_n)$（一个 n-维向量），模型的输出为一个介于 0 和 1 之间的概率值 $P(y=1|\mathbf{x})$，即属于某个类别（如垃圾短信）的概率。

逻辑回归使用 Sigmoid 函数 将线性预测值映射到概率值。模型的预测公式为：

$$P(y=1|\mathbf{x})=\sigma(\mathbf{w}^T\mathbf{x}+b)$$

其中：

$\mathrm{w}=(\mathrm{w}_1,\mathrm{w}_2,\ldots,\mathrm{w}_n)$是特征权重向量。

$\mathrm{x}=(\mathrm{x}_1,\mathrm{x}_2,\ldots,\mathrm{x}_n)$是输入特征向量。

b 是偏置项。

$\sigma(z)$ 是 Sigmoid 激活函数，其定义为：

$$\sigma(z)=\frac{1}{1+e^{-z}}$$

Sigmoid 函数的作用是将任意实数 z 转化为一个 0 到 1 之间的值，表示某一类别的概率。

4.2 代价函数与目标

逻辑回归的目标是通过训练数据最小化损失函数，以求得最优的权重 w 和偏置项 b。对于二分类问题，常用的损失函数是 **对数损失函数**（Log Loss），其定义为：

$$J(\mathbf{w},b)=-\frac{1}{m}\sum_{i=1}^{m}[y^{(i)}\log(h_{\mathbf{w},b}(\mathbf{x}^{(i)}))+(1-y^{(i)})\log(1-h_{\mathbf{w},b}(\mathbf{x}^{(i)}))]$$

其中：

m 是训练样本的数量。

$y^{(i)}$ 是第 i 个样本的真实标签（0 或 1）。

$h_{\mathbf{w},b}(\mathbf{x}^{(i)})$ 是第 i 个样本的预测概率。

对数损失函数衡量的是模型预测值与真实标签之间的差距。目标是通过优化算法（如梯度下降）最小化这个损失函数。

4.3 梯度更新

为了优化损失函数，常用的方法是 **梯度下降**。梯度下降通过计算损失函数

对于参数 w 和 b 的梯度来更新参数。对于逻辑回归，损失函数 $J(\mathbf{w},b)$对于权重和偏置的梯度分别为：

对 w 的梯度：

$$\frac{\partial J(\mathbf{w},b)}{\partial \mathbf{w}} = \frac{1}{m}\sum_{i=1}^{m}(h_{\mathbf{w},b}(\mathbf{x}^{(i)}) - y^{(i)})\,\mathbf{x}^{(i)}$$

对 b 的梯度：

$$\frac{\partial J(\mathbf{w},b)}{\partial \mathbf{b}} = \frac{1}{m}\sum_{i=1}^{m}(h_{\mathbf{w},b}(\mathbf{x}^{(i)}) - y^{(i)})$$

根据这些梯度，参数更新的步骤是：

$$\mathbf{w} := \mathbf{w} - \alpha\frac{\partial J(\mathbf{w},b)}{\partial \mathbf{w}}$$

$$\mathbf{b} := \mathbf{b} - \alpha\frac{\partial J(\mathbf{w},b)}{\partial \mathbf{b}}$$

其中 α 是学习率（控制步长的大小），决定了每次更新的幅度。通过迭代更新，梯度下降会逐步找到使损失函数最小化的 w 和 b。

4.4 模型的概率输出与分类决策

在垃圾短信分类任务中，逻辑回归模型的目标是根据输入特征预测样本属于某一类别（垃圾短信或非垃圾短信）的概率。模型的输出是一个介于 0 和 1 之间的概率值，表示该样本属于垃圾短信（正类，y=1）的概率。这个概率值是通过 **Sigmoid 函数** 计算得到的，如下所示：

$$P(y=1|\mathbf{x}) = \sigma(\mathbf{w}^T\mathbf{x} + b)$$

其中，σ(z) 是 Sigmoid 函数，w 和 b 分别是权重和偏置项，x 是输入特征向量。

为了将概率转化为实际的类别标签，通常会设定一个阈值（如 0.5）来进行分类决策。当模型输出的概率值大于 0.5 时，表示样本属于垃圾短信类别（y=1）；当概率值小于或等于 0.5 时，表示样本属于非垃圾短信类别（y=0）。即：

$$\text{预测标签} = \begin{cases} 1 & \text{如果 } P(y=1|\mathbf{x}) > 0.5 \\ 0 & \text{如果 } P(y=1|\mathbf{x}) \le 0.5 \end{cases}$$

这种方式利用了模型输出的概率信息，并结合了一个简单的决策规则来进行二分类预测。

4.5 评价指标

在垃圾短信分类任务中，评估逻辑回归模型的性能需要使用多种评价指标，以全面了解模型的表现。常见的评估指标包括 **准确率**、**精确率**、**召回率** 和 **F1 分数**。这些指标分别从不同角度衡量模型的分类效果：

（1）准确率 (Accuracy)：表示模型正确预测的样本比例。计算公式为：

$$\text{准确率} = \frac{\text{正确预测的样本数}}{\text{总样本数}} = \frac{TP + TN}{TP + TN + FP + FN}$$

其中，TP（True Positives）是真正例数，TN（True Negatives）是真反例数，FP（False Positives）是假正例数，FN（False Negatives）是假反例数。

（2）精确率 (Precision)：表示模型预测为垃圾短信的样本中，实际为垃圾短信的比例。计算公式为：

$$\text{精确率} = \frac{TP}{TP + FP}$$

精确率衡量的是模型在预测为垃圾短信时的正确性，越高表示误判为垃圾短信的概率越低。

（3）召回率 (Recall)：表示模型能够正确识别的垃圾短信的比例。计算公式为：

$$\text{召回率} = \frac{TP}{TP + FN}$$

召回率衡量的是模型对实际垃圾短信的识别能力，越高表示漏判垃圾短信的概率越低。

（4）F1 分数 (F1 Score)：精确率和召回率的调和平均值，适用于类别不平衡的情况。计算公式为：

$$F1\ \text{分数} = 2 \times \frac{\text{精确率} \times \text{召回率}}{\text{精确率} + \text{召回率}}$$

F1 分数综合了精确率和召回率，能够在关注正类（垃圾短信）和负类（非垃圾短信）平衡的情况下提供更全面的评估。F1 分数越高，模型在平衡精确率与召回率方面的表现越好。

在垃圾短信分类任务中，由于垃圾短信在数据集中通常占较小比例，模型评估更依赖于**精确率**、**召回率**和 **F1 分数**。这些指标能够更全面地反映模型在减少漏报和误报方面的表现。特别是在类别严重不平衡的情况下，**F1 分数**作为精确

率和召回率的调和平均值，能够平衡两者的权重，提供更准确的性能评估。因此，F1 分数通常被作为垃圾短信分类模型的首选评估指标。

5. 最佳模型评估

在本研究中，我们采用多角度方法对逻辑回归模型在 SMS 垃圾短信分类任务中的性能进行了评估，主要从基本性能指标、稳定性分析、类别不平衡影响、AUC-ROC 曲线、学习曲线与模型收敛性分析、错误分析，以及与其他模型的对比等方面展开。

5.1 基本性能指标分析

模型的整体性能通过准确率、精确率 (Precision)、召回率 (Recall) 和 F1-Score 等指标进行衡量。准确率展示了模型预测正确的总体比例，精确率和召回率则更深入地分析了模型在处理垃圾短信（Spam）和正常短信（Ham）类别时的表现。此外，我们通过混淆矩阵直观地展现了模型的预测结果，并从中提取了 TP、TN、FP、FN 值，用于深入计算各项指标。混淆矩阵如图所示：

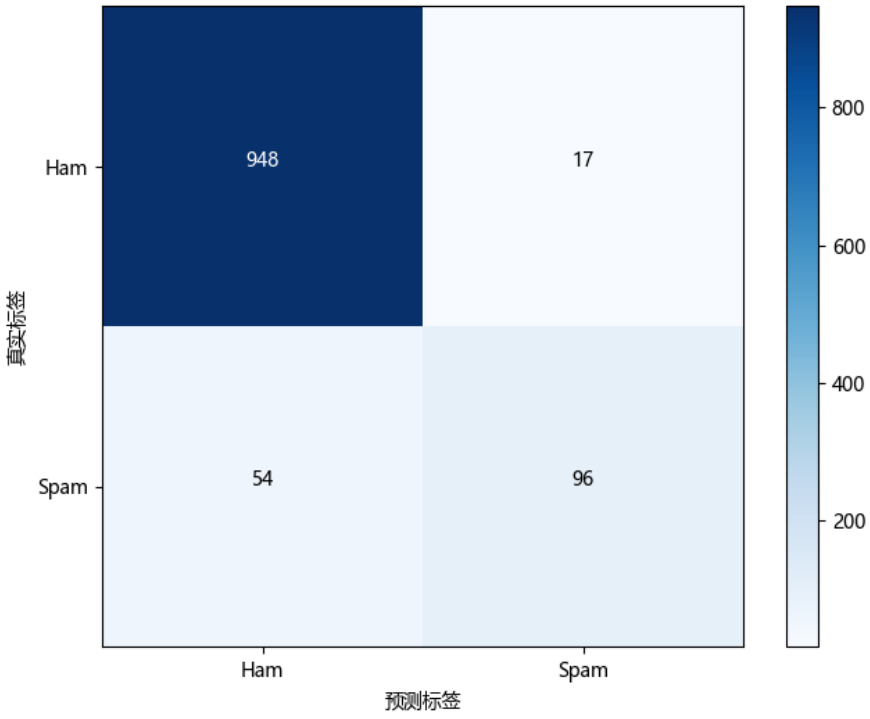

图 2　默认参数的混淆矩阵

该混淆矩阵展示了垃圾邮件分类模型在测试集上的分类结果，学习率为 0.01，迭代次数为 5000，样本权重使用默认值。行代表实际标签（真实的“Ham”

或“Spam”），列代表模型的预测标签（预测的“Ham”或“Spam”）。具体统计结果如下：模型成功识别的正常短信数量（True Positives, TP）为 948，成功识别的垃圾邮件数量（True Negatives, TN）为 99；模型误将 17 条正常短信预测为垃圾邮件（False Positives, FP），误将 51 条垃圾邮件预测为正常短信（False Negatives, FN）。基于这些数据，我们可以计算关键性能指标来评估模型的分类效果，结果如表 3 所示：

表 3　默认参数的邮件分类模型评估指标

类别	精确率	召回率	F1-Score	支持数
Ham	**0.946**	**0.982**	**0.964**	**965**
Spam	**0.849**	**0.640**	**0.729**	**150**
总准确率			**0.933**	**1115**
宏平均	**0.898**	**0.811**	**0.846**	**1115**
加权平均	**0.932**	**0.935**	**0.931**	**1115**

表 3 展示了各类别的评估指标，包括精确率、召回率、F1-Score 和支持数。具体而言，“Ham”类别的精确率为 0.946，召回率为 0.982，F1-Score 为 0.964，支持数为 965；而“Spam”类别的精确率为 0.849，召回率为 0.640，F1-Score 为 0.729，支持数为 150。整体模型的准确率为 0.933。进一步分析，宏平均精确率、召回率和 F1-Score 分别为 0.898、0.811 和 0.846，反映了模型在所有类别上的综合性能；加权平均精确率、召回率和 F1-Score 分别为 0.932、0.935 和 0.931，表明在该分类方法下，模型仍能维持较高的分类效果。

5.2 模型稳定性与类别不平衡的影响

为了评估模型性能的稳定性，我们通过 k 折交叉验证计算了模型在不同数据划分下的平均准确率。结果表明，模型性能在不同训练集和测试集划分中表现一致，具有较好的泛化能力。

同时，类别不平衡问题是垃圾短信分类任务的常见挑战。为此，我们通过调整类别权重的方法对正负样本进行平衡处理，并对比了权重调整前后的模型表现，混淆矩阵如下图所示：

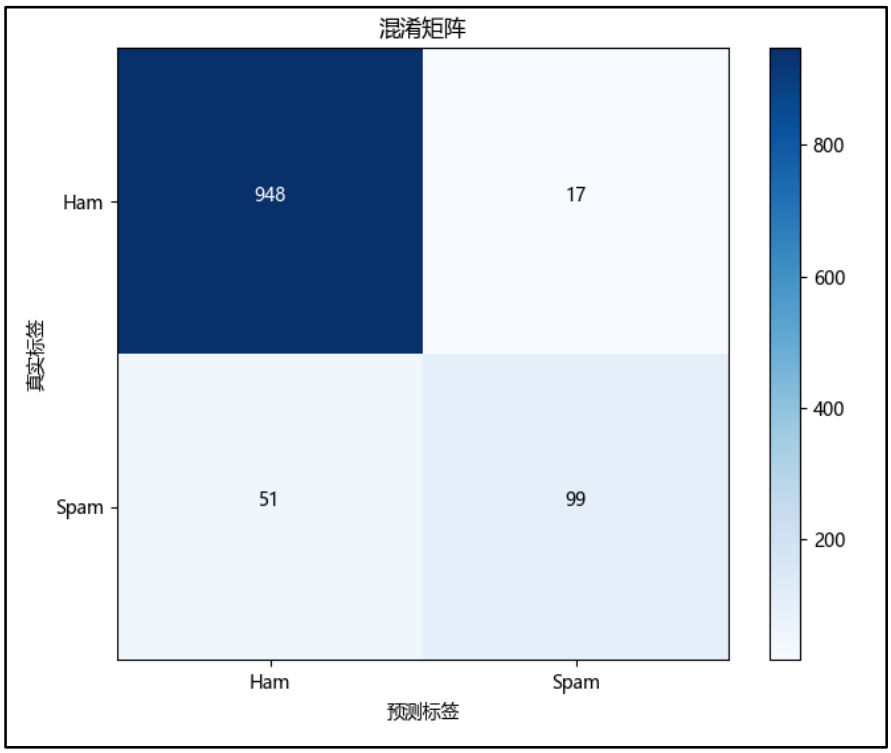

图 3　按样本比率调整权重的混淆矩阵

统计结果如下：模型成功识别的正常短信数量（True Positives, TP）为 948，成功识别的垃圾邮件数量（True Negatives, TN）为 99；模型误将 17 条正常短信预测为垃圾邮件（False Positives, FP），误将 51 条垃圾邮件预测为正常短信（False Negatives, FN）。可以看出调整权重后模型的 FN 相对默认权重减少。基于这些数据，我们可以计算关键性能指标来评估模型的分类效果，结果如表 4 所示：

表 4　按样本调整权重后邮件分类模型评估指标

类别	精确率	召回率	F1-Score	支持数
Ham	**0.95**	**0.98**	**0.96**	**965**
Spam	**0.85**	**0.64**	**0.73**	**150**
总准确率			**0.938**	**1115**
宏平均	**0.90**	**0.81**	**0.85**	**1115**
加权平均	**0.93**	**0.94**	**0.93**	**1115**

表 4 展示了调整样本权重后的模型评估指标。对于“Ham”类别，调整权重前后的分类性能基本保持稳定（精确率从 0.946 提升至 0.950，召回率从 0.982 微降至 0.980，F1-Score 从 0.964 微降至 0.960）。对于“Spam”类别，精确率从 0.849 提升至 0.850，F1-Score 从 0.729 提升至 0.730，召回率保持为 0.640，显示出调

整权重对少数类的改进有限。总体准确率从 0.933 提升至 0.938，宏平均精确率、召回率和 F1-Score 略有提升，加权平均的指标（精确率、召回率、F1-Score）分别从 0.932、0.935 和 0.931 微调至 0.930、0.940 和 0.930。结果表明，调整权重在改善类别平衡性和模型综合性能方面有所提升，但对少数类（Spam）的实际改进较小。下文在调整权重的基础上调整学习率

5.3 模型学习率设置的影响与权衡

学习率是模型训练过程中影响收敛速度与分类性能的关键参数之一。设置过高的学习率可能导致模型在训练过程中出现震荡或无法收敛，而过低的学习率则可能导致训练过程缓慢甚至陷入局部最优。本研究通过超参数搜索搜寻到了最优学习率（alpha=0.5），混淆矩阵如下图所示：

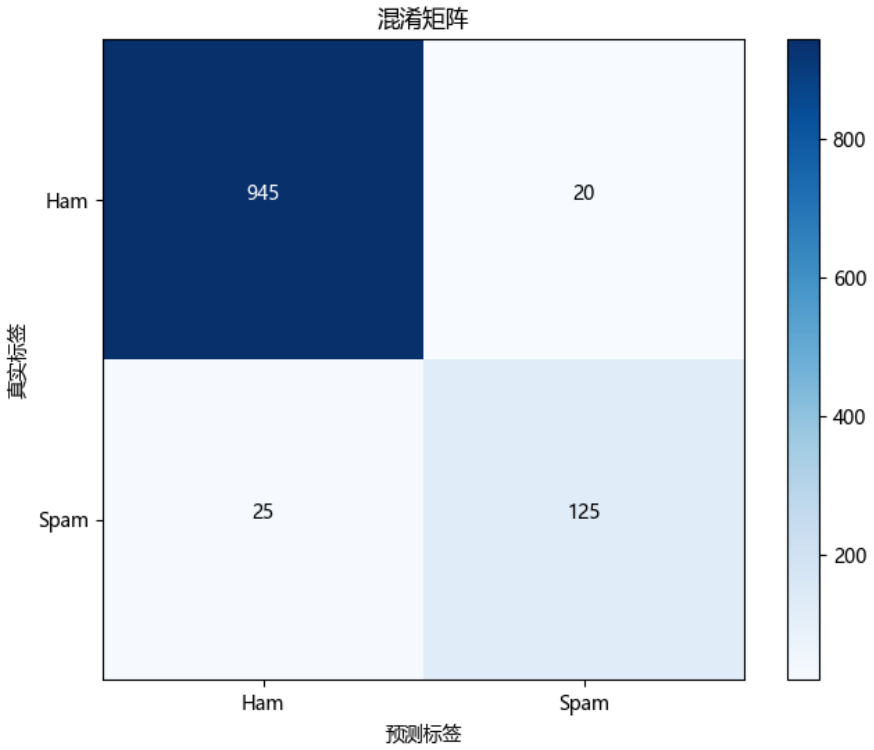

图 4　最优学习率下混淆矩阵

该混淆矩阵展示了垃圾邮件分类模型在测试集上的分类结果，利用网格搜索到最优学习率为 0.5。其中，模型成功识别的正常短信数量（True Positives, TP）为 945，成功识别的垃圾邮件数量（True Negatives, TN）为 125；模型误将 20 条正常短信预测为垃圾邮件（False Positives, FP），误将 25 条垃圾邮件预测为正常短信（False Negatives, FN）。基于这些数据，我们可以计算关键性能指标来评估模型的分类效果，结果如表 5 所示：

表 5　最优学习率下邮件分类模型评估指标

类别	精确率	召回率	F1-Score	支持数
Ham	**0.97**	**0.98**	**0.98**	**965**
Spam	**0.86**	**0.83**	**0.85**	**150**
总准确率			**0.96**	**1115**
宏平均	**0.92**	**0.91**	**0.91**	**1115**
加权平均	**0.96**	**0.96**	**0.96**	**1115**

表 5 进一步展示了在最优学习率（0.5）下，邮件分类模型的评估指标。对于“Ham”类别，精确率提升至 0.97，召回率为 0.98，F1-Score 达到 0.98，支持数为 965；而“Spam”类别的精确率为 0.86，召回率提升至 0.83，F1-Score 达到 0.85，支持数为 150。总体准确率在最优学习率下提升至 0.96。宏平均精确率、召回率和 F1-Score 分别为 0.92、0.91 和 0.91，加权平均的指标（精确率、召回率、F1-Score）分别进一步提升至 0.96、0.96 和 0.96。

综合来看，最优学习率模型相比基准模型和调整权重后的模型，整体性能有显著提升，特别是在“Spam”类别的召回率和 F1-Score 方面表现更优，总体准确率和加权平均指标均达到更高水平。这表明通过优化学习率对模型性能改进效果更为明显。

5.4 AUC-ROC 曲线与阈值优化

为了全面评估垃圾短信分类模型的区分能力，我们分别在**默认参数**、**按样本调整权重**和**最优学习率**三种设置下，绘制了 Receiver Operating Characteristic(ROC)曲线，并计算了对应的 AUC 值。这一评估不仅展示了模型整体性能，还为后续优化分类阈值提供了指导。

5.4.1 基准模型 AUC-ROC 曲线分析

ROC 曲线以假阳性率（False Positive Rate, FPR）为横轴，真正率（True Positive Rate, TPR）为纵轴，能够清晰展示模型在不同阈值下的分类性能。AUC 值（Area Under Curve）则用来量化模型的区分能力，其值范围为 0 到 1，越接近 1 表明模型整体表现越优。基准模型的结果如图 5 所示：

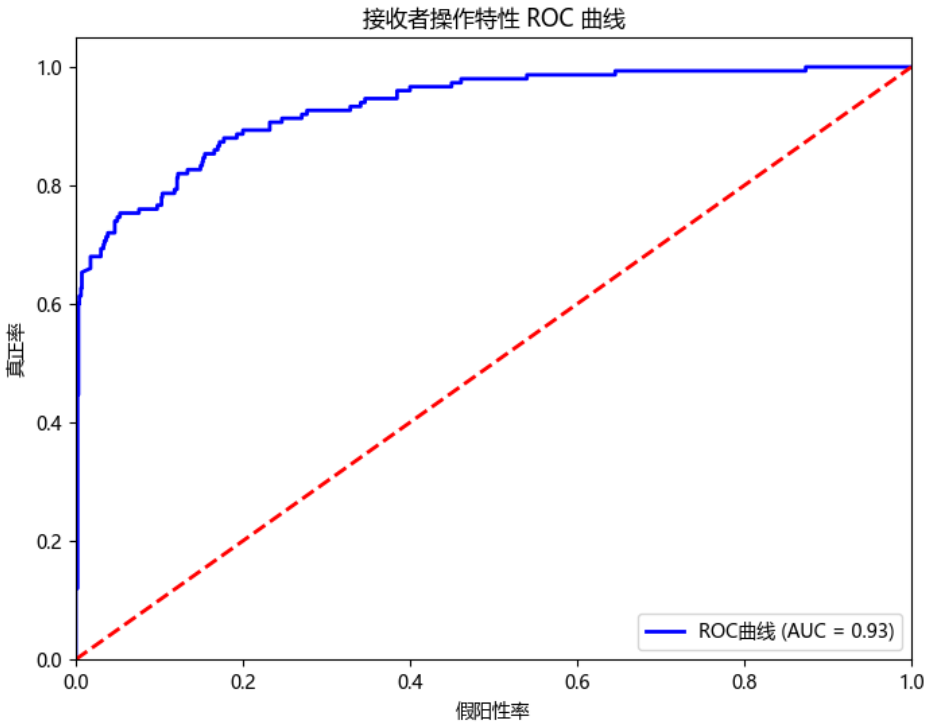

图 5 基准模型的 ROC 曲线

图 5 展示了模型的接收者操作特性（ROC）曲线，用于衡量分类性能。蓝色曲线反映了模型在不同分类阈值下的真阳性率（TPR）与假阳性率（FPR）的关系，整体接近左上角，表明分类能力较高。曲线下的面积（AUC）为 0.93，远高于随机基准（AUC = 0.5），说明模型具有较强的区分正负类别的能力。在假阳性率较低（接近 0）时，真阳性率已接近 0.6，并随着假阳性率增加持续上升，最终趋于 1。红色虚线表示随机猜测基准线，模型曲线显著高于该线，尤其在低假阳性率区域（<0.2）陡峭上升，显示出对正类的早期预测能力较强。综合来看，该模型分类性能优秀，区分度较好，适合当前任务场景。

5.4.2 调整权重后 AUC-ROC 曲线分析

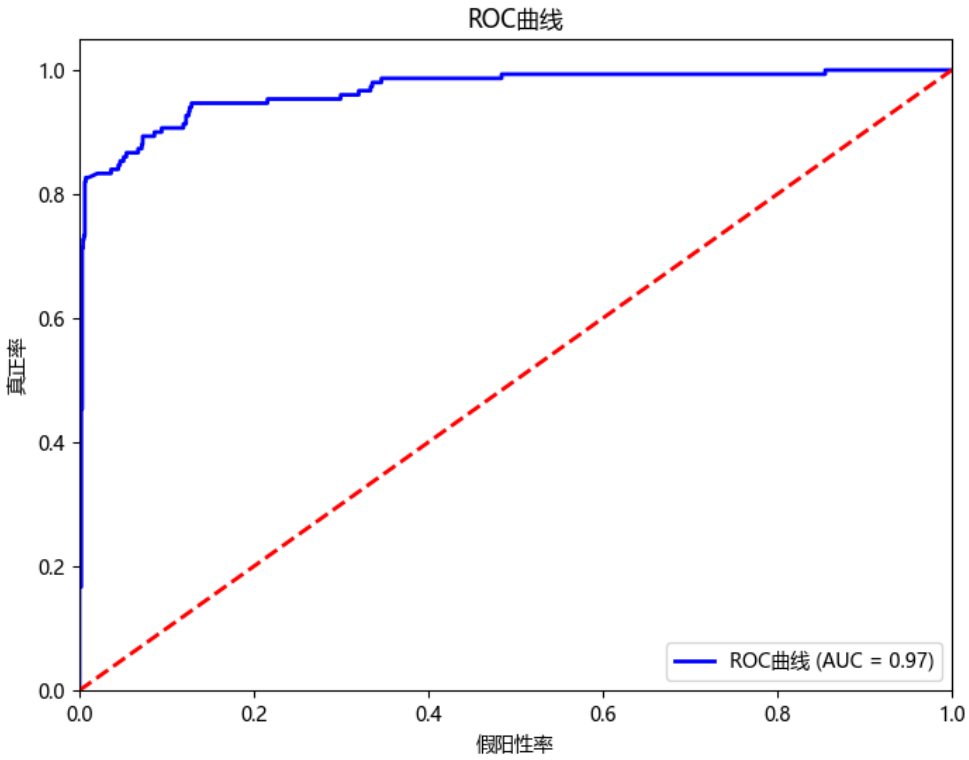

图 6　调整权重后的 ROC 曲线

图 6 展示了模型优化后的 ROC 曲线，重点在于通过权重调整提升分类性能。优化后，曲线下的面积（AUC）由图 5 的 0.93 提高至 0.97，进一步逼近理论最优（AUC = 1.0），表明模型对正负类别的区分能力显著增强。在假阳性率较低区域（<0.2），真阳性率由图 5 的约 0.6 提升至接近 0.8，显示出优化后模型在早期预测正类方面的能力明显改进，曲线陡峭程度增加进一步证明了其对正类的敏感性增强。此外，尽管红色随机基准线保持不变，蓝色 ROC 曲线的优势进一步扩大，尤其在低假阳性率和高真阳性率区域表现更加卓越。总体而言，权重优化有效提升了模型性能，使其更加适合当前任务需求，验证了优化措施的必要性和有效性。

5.4.3 网格搜索优化权重后 AUC-ROC 曲线分析

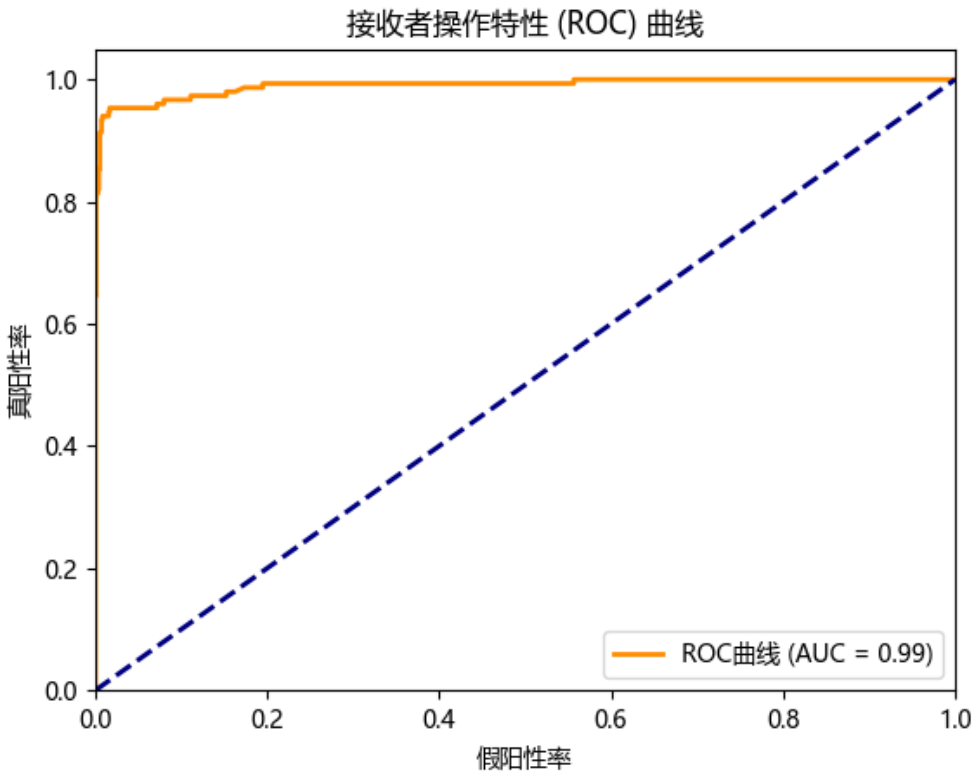

图 7 网格搜索优化权重后 AUC-ROC 曲线

图 7 展示了模型在进行学习率优化后的 ROC 曲线，其分类性能较图 6 进一步提升，体现了学习率优化的重要作用。具体来看，图 7 的 ROC 曲线依然紧贴左上角，AUC 值由图 6 的 0.97 进一步提高至 0.99，接近完美分类性能（AUC=1.0），与图 5 的 0.93 相比表现出显著优势，充分表明模型对正负类别的区分能力已达到新的高度。在假阳性率较低区域（<0.2），图 7 的真阳性率接近 1.0，相比图 5 的约 0.6 和图 6 的约 0.8，模型在低 FPR 条件下的预测准确性得到了显著提升，且曲线更加陡峭，表明模型对正类预测的敏感性进一步增强，同时提升了稳健性。从总体趋势看，AUC 值的逐步提升（0.93 → 0.97 → 0.99）表明权重优化与学习率优化的累积效果显著，网格搜索方法成功找到了更优的学习率参数，使模型性能进一步逼近理论最优状态。综上，学习率优化后的 ROC 曲线表现验证了网格搜索策略的高效性和优化措施的有效性，为提升模型的分类能力起到了关键作用。

5.5 学习曲线与模型收敛性分析

通过绘制训练过程中的代价函数变化曲线，我们验证了模型的收敛性。代价函数的平稳下降表明模型训练过程收敛良好，没有出现震荡或过早停止的问题。同时，通过比较训练集和测试集上的性能指标，我们排除了模型过拟合或欠拟合的可能性。代价函数变化曲线如下所示：

5.5.1 基准模型的模型收敛性分析

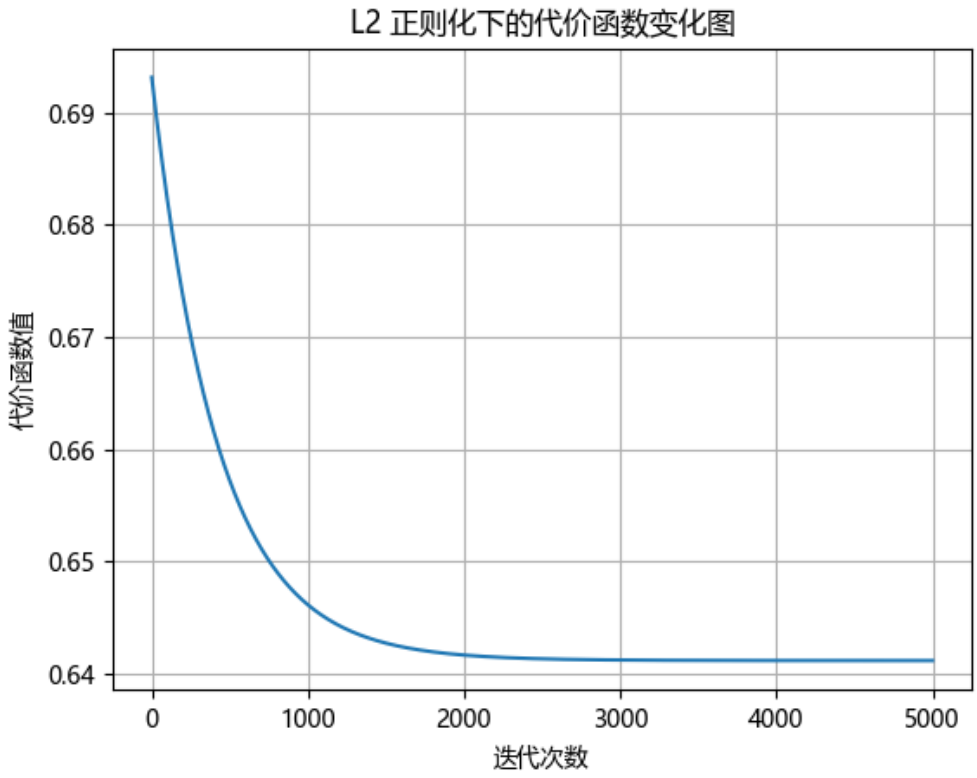

图 8 基准模型的代价函数图

图 8 展示了基准模型在默认 L2 正则化条件下代价函数随迭代次数的变化情况，反映了模型在初始状态下的优化过程及收敛趋势。代价函数在迭代次数为 0 时的值为 0.69，代表未优化时的初始误差水平，随后随着迭代次数的增加逐渐减小。在早期阶段（迭代次数 0 至 2000），代价函数值从 0.69 下降至 0.64，呈现出较为平稳的下降趋势；在 2000 次迭代后，下降速度显著放缓，并最终在 5000 次迭代时收敛至 0.64，表明模型已达到基本的最优状态。整体来看，代价函数从 0.69 下降至 0.64，总体下降幅度为 0.05，下降比例约为 7.25%。这一结果表明，基准模型在默认 L2 正则化条件下能够实现稳定的优化，但性能提升幅度有限，收敛效率相对一般。总体而言，图 8 验证了基准模型在默认参数下的训练效果，代价函数的逐步减小表明模型能够有效减少误差，但模型优化仍有进一步提升的

空间。

5.5.2 优化权重后的模型收敛性分析

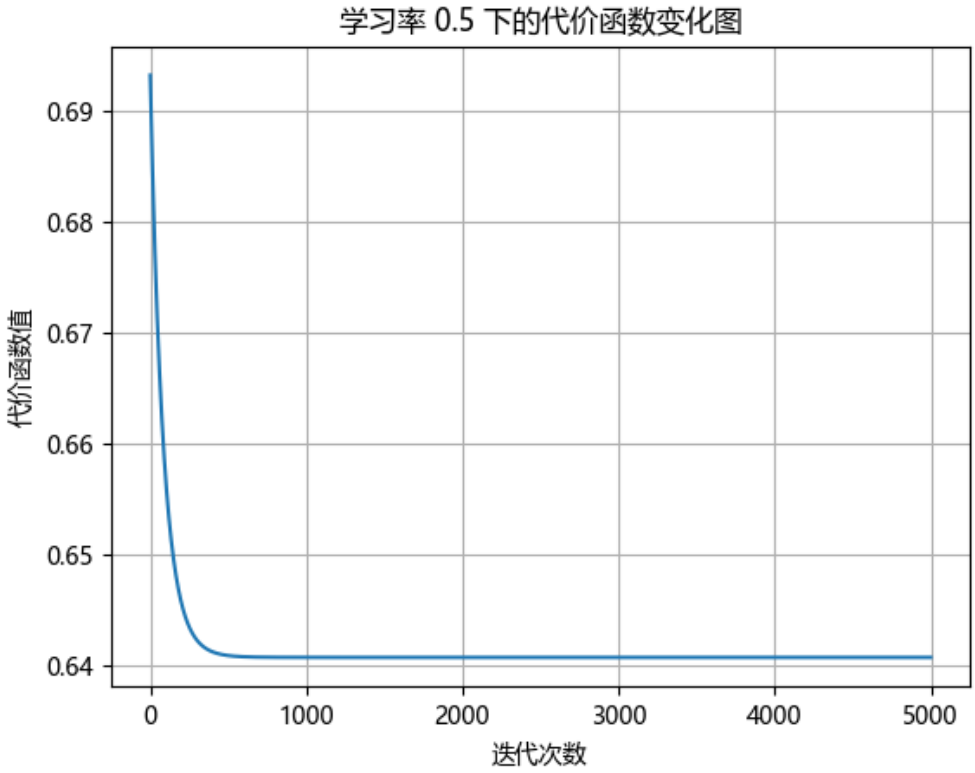

图 9 优化权重后的代价函数图

图 9 展示了优化权重后的代价函数随迭代次数的变化情况，与图 8 基准模型的表现进行了对比。在优化权重后，代价函数的初始值仍为 0.69，但其下降趋势和最终收敛效果更为显著。具体而言，图 9 中代价函数在迭代次数 0 至 400 时迅速下降，从 0.69 下降至接近 0.64，较图 8 表现出更快的收敛速度，表明权重优化有效提升了模型在早期阶段的收敛效率。在 2000 次迭代后，尽管下降幅度减缓，但整体收敛速度依然优于图 8，并在 5000 次迭代时稳定于 0.64，表明模型已达到优化权重后的最优状态。尽管图 8 和图 9 的代价函数下降幅度均为 0.05（从 0.69 至 0.64），下降比例约为 7.25%，但图 9 的优化过程显著提升了模型在初始阶段的误差减少能力，体现了更高的训练效率。总体来看，图 9 验证了权重优化对提升模型训练效率和收敛速度的作用，为进一步优化模型参数提供了有益的参考。

5.5.3 网格搜索优化后的模型收敛性分析

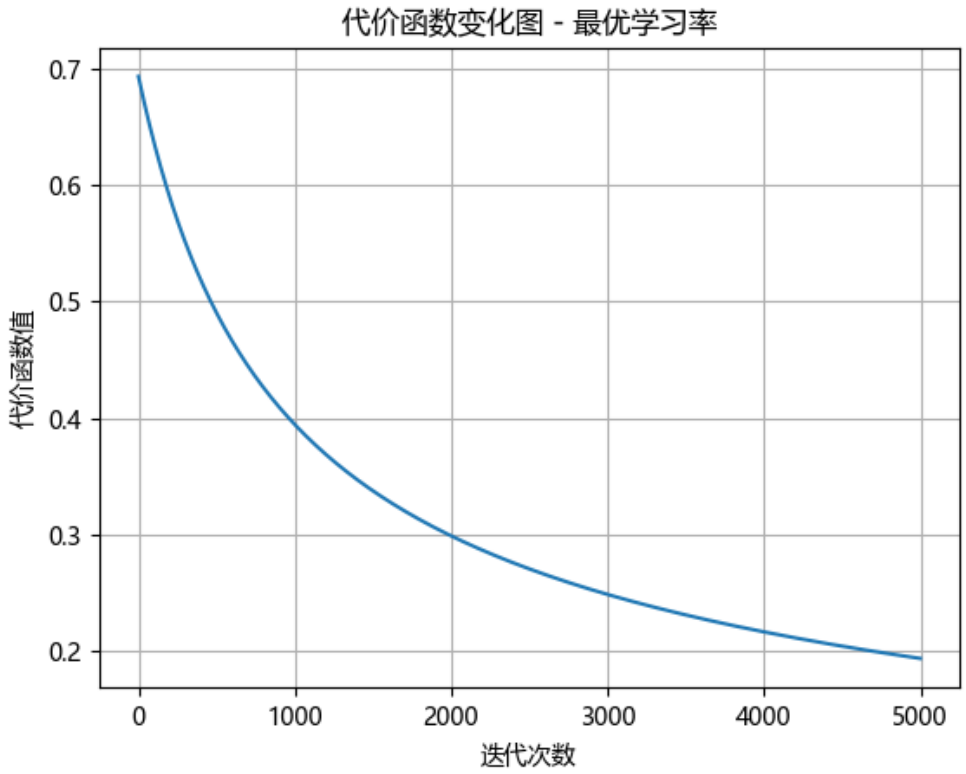

图 10　网格搜索优化后模型收敛性分析

图 10 展示了基于网格搜索优化学习率后的代价函数随迭代次数的变化情况，并与图 9 优化权重后的模型进行了对比。在进一步优化学习率后，代价函数的下降趋势和最终收敛效果均得到了显著提升。与图 9 相比，图 10 的代价函数在迭代次数 0 至 5000 时的下降速度进一步加快，从初始值 0.69 迅速下降至约 0.2，比图 9 代价函数值更低，表明优化学习率后的模型能够更高效地利用初始几轮迭代，大幅提升了收敛速度。在 400 至 2000 次迭代期间，图 10 的代价函数下降趋势较图 9 更为剧烈，从 0.55 逐渐下降至接近 0.35，而图 9 在这一阶段的代价函数仅下降至约 0.64。到 5000 次迭代时，图 10 的代价函数最终收敛于 0.2 左右，显著低于图 9 的 0.64，体现了网格搜索优化学习率后模型的更优性能。尽管图 9 已经通过优化权重显著提升了模型的收敛效率和效果，但图 10 进一步优化学习率后，代价函数的下降幅度从图 9 的 0.05 扩展至 0.49（从 0.69 降至 0.2），下降比例达到 71%。这一改进表明，学习率的合理调整对于进一步提升模型训练效果至关重要。总体来看，图 10 验证了在优化权重基础上，通过网格搜索优化学习率可以进一步提升模型的收敛速度与最终效果。

6. 结论

本研究基于 Logistic 回归模型，针对 SMS 垃圾邮件分类任务进行了系统性研究，通过构建基准模型、调整权重模型以及最优学习率模型，深入探讨了不同参数优化对模型性能的影响。实验结果表明，最优学习率模型在 AUC 值和收敛速度方面均取得了显著优势，显示出合理调整超参数对模型性能优化的重要性。本研究不仅验证了 Logistic 回归在垃圾邮件分类任务中的有效性，还为后续研究提供了参数调整与模型改进的实践参考。尽管如此，研究仍存在局限性，如对特征工程的探索较为有限、模型适应复杂场景的能力有待提高。未来研究可考虑引入更复杂的模型或多模型集成方法，并在多语言 SMS 数据集上进行迁移学习实验，以进一步提高垃圾邮件分类的鲁棒性与泛化能力。