一、课题综述

1.1. 课题说明

2252445 王泓烨，数据预处理与特征提取，撰写实验报告。
2252085 朱亚琨，手写逻辑回归模型，撰写实验报告。
2250409 罗尹泽，手写 SVM 模型，撰写实验报告。
2252699 王鉴戈，手写朴素贝叶斯模型，撰写实验报告。
（注：排序与贡献无关，贡献度均分）

1.2. 课题目标

本项目的目标是通过特征工程和数据预处理，对目标数据集进行可视化分析并处理，并尝试构建多种机器学习模型，包括逻辑回归、SVM、朴素贝叶斯模型，来解决电子邮件二分类问题，将电子邮件分为垃圾邮件和正常邮件两类。

研究过程中我们将原始文本数据进行预处理以及探索性分析，并将文本转换为**降维的特征向量**；对逻辑回归、SVM、朴素贝叶斯模型进行**手写实现**；针对数据集本身的数据特性如类别不平衡问题对手写模型进行**优化改进，并实现结果可视化**；对不同的传统机器学习方法得到的结果进行**比较分析**。

1.3. 课题数据集

本课题使用的数据集为来自 Kaggle 竞赛的 SMS 垃圾邮件收集数据集 (https://www.kaggle.com/datasets/uciml/sms-spam-collection-dataset/data)。该数据集包含从某英国论坛中手动提取的 425 条 SMS 垃圾邮件以及 NUS SMS 语料库中 3375 条随机选择的合法邮件，共 3800 条邮件样本。将样本按 8:2 随机分成训练集和测试集，供项目使用。

二、实验报告设计

2.1. 数据准备

在 kaggle 网站中下载 SMS 垃圾邮件收集数据集。.csv 文件的第一列为邮件编号，第二列为邮件的具体内容。包含从某英国论坛中手动提取的 425 条 SMS 垃圾邮件以及 NUS SMS 语料库中 3375 条随机选择的合法邮件，共 3800 条邮件样本。

2.2. 数据预处理

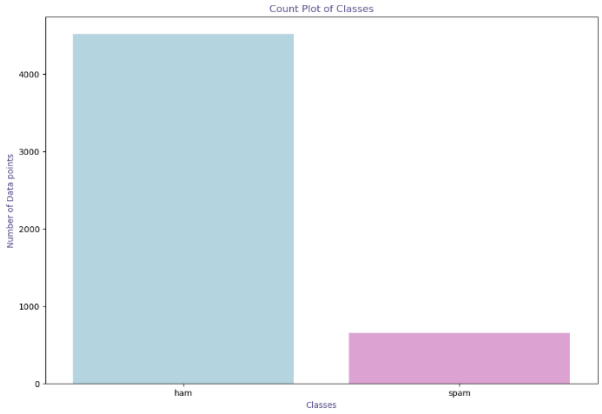
2.2.1 数据清洗

清除数据集中的空值和重复值，并将表格列名改为更易于理解的 target 和 text，得到下面结果。

	target	text
0	ham	Go until jurong point, crazy.. Available only in bugis n great world la e buffet... Cine there got amore wat...
1	ham	Ok lar... Joking wif u oni...
2	spam	Free entry in 2 a wkly comp to win FA Cup final tkts 21st May 2005. Text FA to 87121 to receive entry question(std txt rate)T&C's apply 08452810075over18's
3	ham	U dun say so early hor... U c already then say...
4	ham	Nah I don't think he goes to usf, he lives around here though

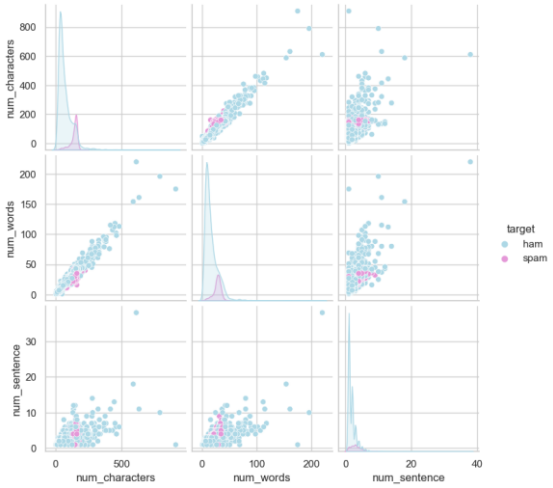
2.2.2 数据探索性分析

统计数据集中垃圾邮件和合法邮件的比例，绘制条形图。



看到合法邮件的样本量明显多于垃圾邮件。

统计邮件内容中的 characters、words、sentences 数量， 得到 characters、words、sentences 数量的相关图如下。



字符数与单词数存在强烈的线性相关性，尤其在 ham 数据中，这表明大多数消息都是由相似长度的单词组成的。从颜色的分布来看，垃圾消息（spam）在每个特征的分布上通常与正常消息（ham）存在差异，例如字符数和单词数相对较多、分布更加集中。

由于篇幅问题，其他探索性分析在 preprocess_data.ipynb 中。

2.2.3 数据标准化

数据标准化的主要步骤包括：

1、转换为小写

2、分词

3、去除特殊字符和停用词

4、进行词干提取。

经过数据标准化后的数据展示如下：

	target	text	num_characters	num_words	num_sentence	transformed_text
0	0	Go until jurong point, crazy.. Available only in bugis n great world la e buffet... Cine there got amore wat...	111	24	2	go jurong point crazi avail bugi n great world la e buffet cine got amor wat
1	0	Ok lar... Joking wif u oni...	29	8	2	ok lar joke wif u oni
2	1	Free entry in 2 a wkly comp to win FA Cup final tkts 21st May 2005. Text FA to 87121 to receive entry question(std txt rate)T&C's apply 08452810075over18's	155	37	2	free entri 2 wkli comp win fa cup final tkt 21st may text fa 87121 receiv entri question std txt rate c appli 08452810075over18
3	0	U dun say so early hor... U c already then say...	49	13	1	u dun say earli hor u c alreadi say
4	0	Nah I don't think he goes to usf, he lives around here though	61	15	1	nah think goe usf live around though

2.2.4 提取 TF-IDF 特征

使用 TfidfVectorizer 对文本数据进行特征提取，将标准化后的文本转换为数组，根据词汇在所有文本中的重要性保留 3000 个最重要的特征（词汇）。数组中的数值表示某个词在该文本中的 TF-IDF 值。

将数据集按 8:2 分为训练集和测试集，并保存在 train_test_data.pkl 中。

2.3. 模型搭建

2.3.1. 逻辑回归模型

逻辑回归是一种广泛应用于二分类问题的统计学习方法。它通过拟合一个 Sigmoid 函数来估计输入特征与输出标签之间的关系。逻辑回归模型的核心在于最大化似然函数，从而找到最佳的模型参数。

逻辑回归模型的基本形式如下：

$$P(y=1 \mid x;\theta)=\frac{1}{1+e^{-(\theta^T x)}}$$

其中，θ 是模型参数，x 是输入特征向量，y 是标签。

为了防止过拟合，我们在模型中加入了 L2 正则化项：

$$J(\theta)=-\frac{1}{m}\sum_{i=1}^{m}\left[y^{(i)}\log\left(h_\theta(x^{(i)})\right)+(1-y^{(i)})\log\left(1-h_\theta(x^{(i)})\right)\right]+\frac{\lambda}{2m}\sum_{j=1}^{n}\theta_j^2$$

其中，λ 是正则化参数。

具体模型搭建代码在 code/Logistic.ipynb 中。

2.3.2. SVM 模型

1）线性支持向量机

线性支持向量机（Linear SVM）主要用于线性可分的数据，目标是找到一个 最优超平面，能够最大化地分离不同类别的数据点。其目标函数可以表示为：

$$\min \frac{1}{2}\|w\|^2$$

在满足以下约束条件下：

$$y_i(w\cdot x_i+b)\geq 1,\quad \forall i$$

2）非线性支持向量机

当数据线性不可分时，可以使用 软间隔 和 核技巧 来处理：

• 软间隔：引入松弛变量，允许一部分样本点位于决策边界内。目标函数变为：

$$\min \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n}\xi_i$$

• 核技巧：将输入数据映射到高维空间，在高维空间中进行线性分类。

常用的核函数包括：

① 线性核（Linear Kernel）：$K(x_i, x_j) = x_i \cdot x_j$

② 多项式核（Polynomial Kernel）：$K(x_i, x_j) = (1 + x_i \cdot x_j)^p$

③ 高斯核（RBF Kernel）：$K(x_i, x_j) = \exp\left(-\frac{\|x_i - x_j\|^2}{2\sigma^2}\right)$

2.3.3 朴素贝叶斯模型

Naive Bayes（朴素贝叶斯）是一种基于贝叶斯定理的概率分类算法。朴素假设指的是假定各个特征之间是条件独立的，即给定类别的情况下，每个特征的概率分布相互独立。对于输入数据 $X=\{x_1,x_2,...,x_n\}$，朴素贝叶斯模型通过最大化条件概率 $P(Y \mid X)$ 预测目标类别 Y，其核心思想可以用以下公式表示：

$$P(Y|X) = \frac{P(Y)\cdot\prod_{i=1}^{n}P(x_i|Y)}{P(X)}$$

其中 $P(Y \mid X)$ 表示给定输入 X 的类别 Y 的后验概率。模型选择具有最大概率的类别作为预测结果。

实验中我们手写实现了 2 个朴素贝叶斯分类器：

（1）多项式 Naive Bayes (Multinomial Naive Bayes)：

- 假定特征是离散的计数值（如词频）。适用于词袋模型中的词频或特征出现的次数。
- 在条件概率估计中，假设每个特征值的概率与类别的条件概率成比例，适合处理文本数据中的词频分布。

（2）伯努利 Naive Bayes (Bernoulli Naive Bayes)：

- 假定特征是二值的（存在或不存在）。在每个类别下，特征的取值为“1”或“0”，分别表示特征的出现和缺失。
- 适用于稀疏特征和文本数据中关键词是否出现的情况。

在实现的过程中，为了避免朴素贝叶斯分类器会出现的“零概率”问题，我们采用**Lidstone 平滑**方法来避免这个问题。Lidstone 平滑的主要思想是给每个事件增加一个小的平滑常数 α，使得所有事件的概率估计不会为零。

在 Lidstone 平滑中，条件概率的计算公式为：

$$P(X_i|C) = \frac{\text{count}(X_i, C) + \alpha}{\text{count}(C) + \alpha \cdot N}$$

其中，$count(X_i,C)$ 是特征 X_i 在类别 C 中出现的次数；$count(C)$ 是类别 C 中所有特征的总计数；N 是特征的总数（如词汇表大小）；α 是一个平滑参数，取值范围为 α>0。

具体搭建过程详见 code/Bayes.ipynb

2.4. 模型训练测试

2.4.1. 逻辑回归

（1）数据准备

我们使用了预处理后的数据集 train_test_data.pkl，其中包含训练集和测试集的数据。为了保证模型的泛化能力，我们对特征进行了标准化处理。

（2）模型训练

我们使用批量梯度下降（GD）和随机梯度下降（SGD）训练逻辑回归模型。

批量梯度下降（GD）

批量梯度下降使用整个训练数据集来计算每次迭代的梯度，确保每次更新都是基于所有数据的平均梯度。这种方法收敛稳定，但计算成本较高。

随机梯度下降（SGD）

随机梯度下降每次仅使用一部分数据来计算梯度，计算效率较高，适合大规模数据集。然而，由于每次更新依赖于较少的数据，可能会导致更新路径较为震荡。

（3）模型测试

我们使用测试集数据对训练好的模型进行评估，包括计算准确率、精确率、召回率和 F1 分数，并绘制混淆矩阵、分类报告和 ROC 曲线。

2.4.2. SVM

（1）数据准备

我们使用了预处理后的数据集 train_test_data.pkl，其中包含训练集和测试集的数据。为了保证模型的泛化能力，我们对特征进行了标准化处理。把 0，1 数据映射为-1，1，并使用 PCA 将 3000 维的数据降为 200 维。

（2）模型训练

我们使用了不同的核函数（线性核、多项式核和径向基核）对 SVM 模型进行了训练，并分别测试了不同的惩罚参数 C 值。

（3）模型测试

我们对不同核函数和不同 C 值下训练的 SVM 模型进行了测试，并比较了手动实现与 scikit-learn 调包实现的测试准确率及运行时间。

2.4.3 朴素贝叶斯模型

（1）数据预处理

①数据加载与分割：将已提取好特征的文本数据 train_test_data.pkl 数据按照 8:2 的比例分为训练集和测试集。

②特征提取与二值化处理：

- 对多项式 Naive Bayes 使用词频特征表示，即每个特征是词在文档中出现的次数。
- 对伯努利 Naive Bayes 进行二值化处理，将每个特征转为“存在”或“缺失”状态。

（2）模型训练

分别使用多项式 Naive Bayes 和伯努利 Naive Bayes 模型，并设置不同的 α\alphaα 值（Lidstone 平滑）。训练过程包括以下步骤：

- **训练数据：**将训练数据输入模型，计算每个类别下特征的条件概率，并根据类别频率更新先验概率。
- **平滑处理：**使用不同的 α 值（如 0.1、0.3、0.5、1.0、2.0、5.0），对每个特征的条件概率进行 Lidstone 平滑。

（3）模型测试

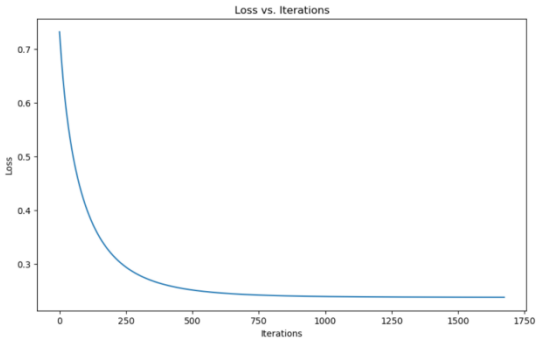
在测试集上评估模型的表现。计算测试集的 Accuracy、F1 Score、Precision 和 Recall 评价指标，并与库实现的分类器进行了对比。

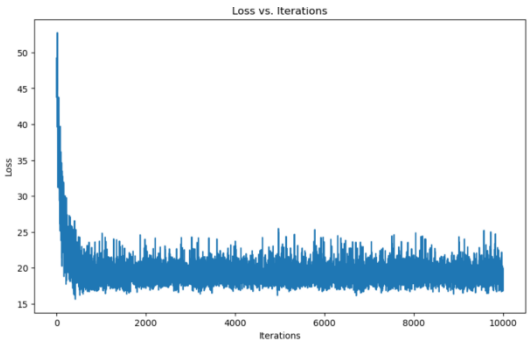
2.5. 结果可视化

2.5.1. 逻辑回归

（1） 损失函数曲线

批量梯度下降（GD） 随机梯度下降（SGD）

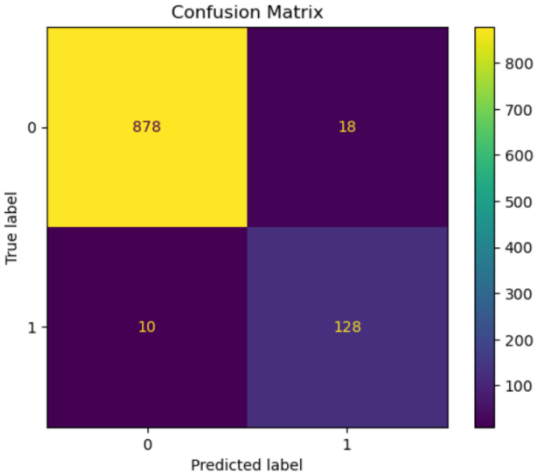


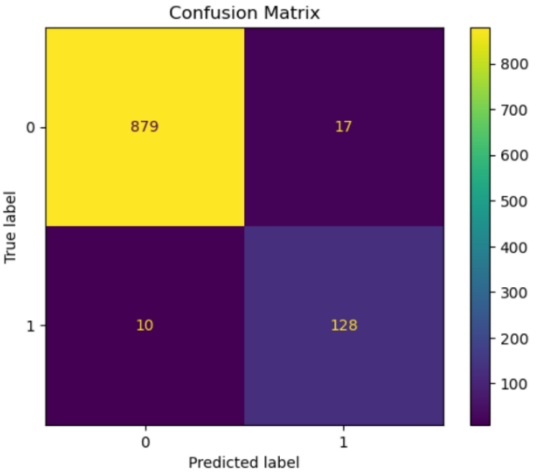


批量梯度下降（GD）的损失函数在初期迅速下降，随后逐渐趋近于一个稳定的最小值。这种平滑且单调递减的趋势表明 GD 能有效找到全局最优解，但每次迭代需要计算整个训练集的梯度，导致收敛速度较慢。**随机梯度下降（SGD）**的损失函数在初期也迅速下降，但在达到一定点后出现较大波动。这是由于 SGD 每次迭代只使用一个或一小部分样本，导致梯度方向不稳定，损失函数呈锯齿状变化。尽管如此，SGD 仍能在多次迭代后接近一个较低的损失值。

（2） 混淆矩阵

批量梯度下降（GD） 随机梯度下降（SGD）





批量梯度下降（GD） 和 **随机梯度下降（SGD）** 的混淆矩阵非常相似，在分类任务上的表现相当。

（3） 分类报告

批量梯度下降（GD）

Classification Report:

	precision	recall	f1-score	support
0	0.99	0.98	0.98	896
1	0.88	0.93	0.90	138
accuracy			0.97	1034
macro avg	0.93	0.95	0.94	1034
weighted avg	0.97	0.97	0.97	1034

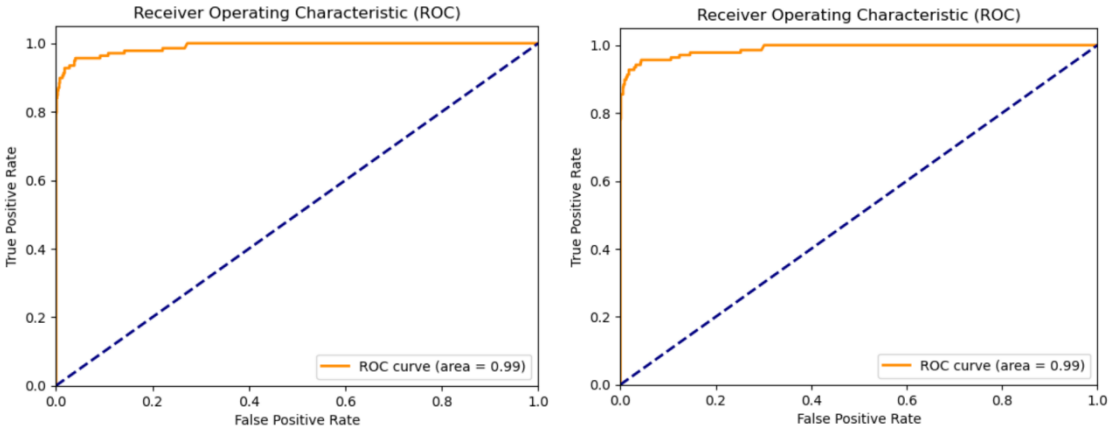
随机梯度下降（SGD）

Classification Report:

	precision	recall	f1-score	support
0	0.99	0.98	0.98	896
1	0.88	0.93	0.90	138
accuracy			0.97	1034
macro avg	0.94	0.95	0.94	1034
weighted avg	0.97	0.97	0.97	1034

（4） ROC 曲线

批量梯度下降（GD）　　随机梯度下降（SGD）

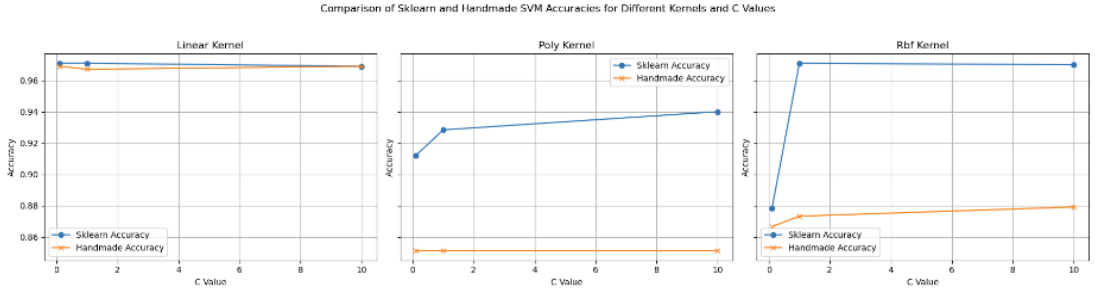


批量梯度下降（GD）和随机梯度下降（SGD）两种方法在垃圾邮件分类任务上均表现出色。它们的精度、召回率和 F1 分数都非常高，特别是对于类别 0 的识别，精确度达到了 0.99。AUC 值均为 0.99，表明模型具有出色的区分能力。

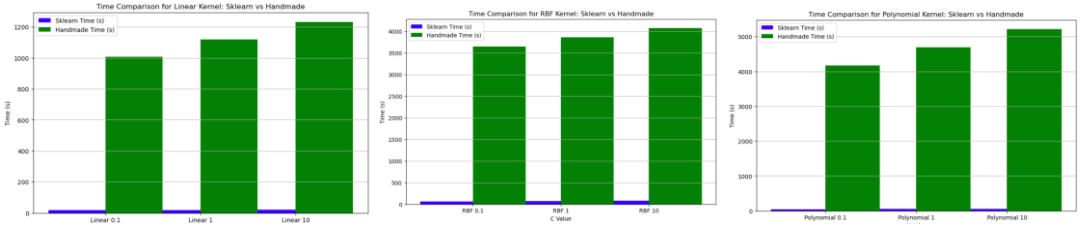
2.5.2. SVM

Kernel	C	Sklearn Accuracy	Handmade Accuracy	Sklearn Time (s)	Handmade Time (s)	Handmade Support Vectors
Linear	0.1	0.971	0.9691	18	1008	243
Linear	1	0.971	0.9671	20	1120	250
Linear	10	0.9691	0.9691	23	1232	246
Polynomial	0.1	0.912	0.8511	49	4176	87
Polynomial	1	0.9284	0.8511	55	4700	87
Polynomial	10	0.94	0.8511	60	5220	87
RBF	0.1	0.8781	0.8665	66	3644	4102
RBF	1	0.971	0.8733	74	3860	4104
RBF	10	0.97	0.8791	77	4072	4102

基向量（支持向量）的数量在不同的核函数和 C 值下差异显著。尤其在 RBF 核的情况下，支持向量的数量较多，这表明 RBF 核对数据的拟合更加精细，但同时也可能导致模型的复杂度和过拟合风险增加。



基向量（支持向量）的数量在不同的核函数和 C 值下差异显著。尤其在 RBF 核的情况下，支持向量的数量较多，这表明 RBF 核对数据的拟合更加精细，但同时也可能导致模型的复杂度和过拟合风险增加。



从测试结果可以看出，scikit-learn 的实现精度略高于手动实现，尤其是在多项式核和 RBF 核的情况下。同时，scikit-learn 的运行时间普遍低于手动实现，这可能是由于 scikit-learn 使用了更高效的优化算法和硬件加速。

线性核的优势：在高维数据上，线性核的表现可能会优于多项式核和 RBF 核，尤其是当数据本身接近线性可分时。

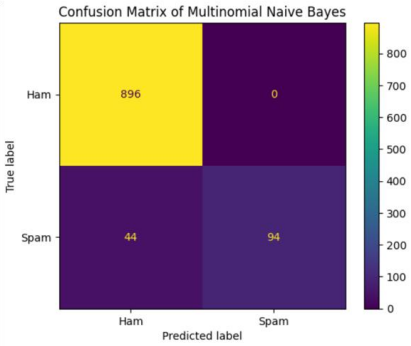
手写 SVM 的劣势：手写实现由于缺少数值优化和对复杂核函数的处理能力，通常在准确率上略逊于调包版本。

调包 SVM 的优势：通过使用高度优化的库（如 scikit-learn），调包 SVM 在训练时间和准确率上都有显著优势，特别是在处理高维和复杂数据时。

2.5.3 朴素贝叶斯模型

（1）多项式 Naive Bayes (Multinomial Naive Bayes)

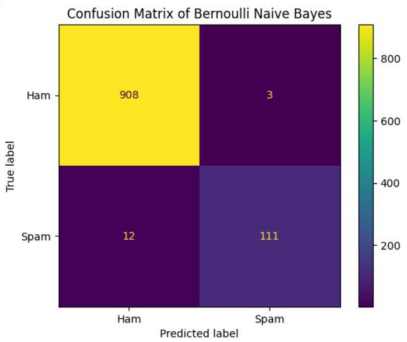
```
Handwritten Naive Bayes Classification Report:
              precision    recall  f1-score   support

           0       0.95      1.00      0.98       896
           1       1.00      0.68      0.81       138

    accuracy                           0.96      1034
   macro avg       0.98      0.84      0.89      1034
weighted avg       0.96      0.96      0.95      1034

accuracy: 0.9574468085106383
f1: 0.810344827586207
precision: 1.0
recall: 0.6811594202898551
```



由上方结果可知，多项式 Naive Bayes 表现尽管精确率很高，但召回率较低，表明该模型在识别正类时存在较高的漏检率。由于数据集中正例的样本明显多与反例，多项式 Naive Bayes 在处理此类存在类别不平衡或特征频率显著差异的数据时可能会出现偏差。

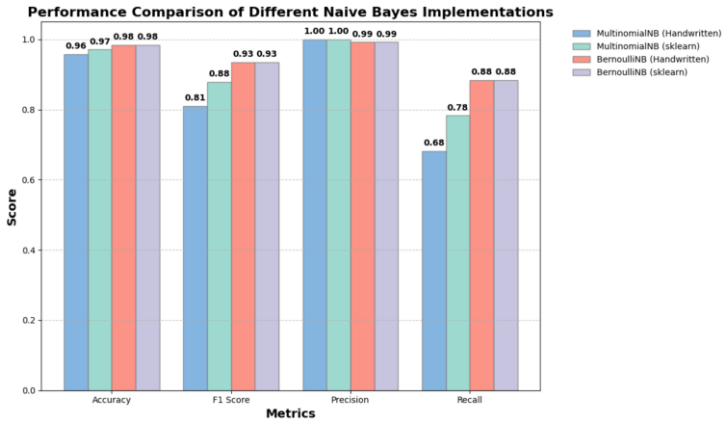
（2）伯努利 Naive Bayes (Bernoulli Naive Bayes)

```
Bernoulli Naive Bayes Classification Report:
              precision    recall  f1-score   support

           0       0.98      1.00      0.99       896
           1       0.99      0.88      0.93       138

    accuracy                           0.98      1034
   macro avg       0.99      0.94      0.96      1034
weighted avg       0.98      0.98      0.98      1034

accuracy: 0.9835589941972921
f1 Score: 0.9348659003831418
precision: 0.991869918699187
recall: 0.8840579710144928
```



由上方结果可知，伯努利 Naive Bayes 表现出较高的精确率和召回率，具有较好的鲁棒性。原因可能是因为数据集大小较小，数据类别差异较大，存在稀疏数据或关注特征的存在/缺失的情况。这种情况下，伯努利 Naive Bayes 依赖关键词“触发”会更加有效适用。

（3）与 sklearn 库中模型对比

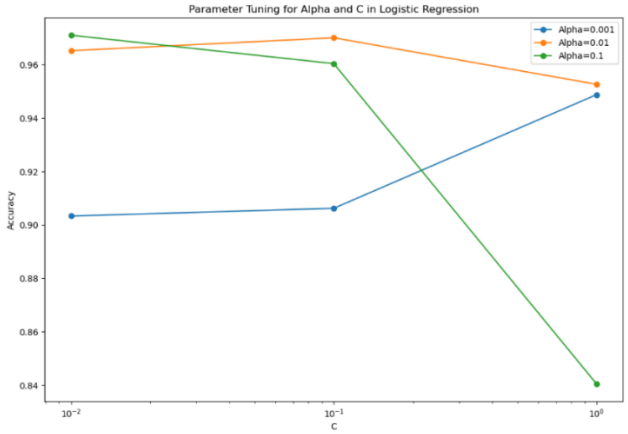


可以发现，手写实现的多项式 Naive Bayes 与库中相应模型差别较大，可能是因为没有能够良好的处理类别不平衡问题所致；而手写的伯努利 Naive Bayes 与相应的库中的模型实现了几乎一致的优秀表现。

2.6. 分析和优化（要包含对两类模型的结果的比较讨论）

2.6.1 逻辑回归模型分析与优化

通过调整逻辑回归模型的学习率参数（alpha）和正则化强度参数（C），以优化模型在垃圾邮件分类任务上的表现。我们希望通过实验找到一组参数，使模型能够在测试集上达到更高的准确率，从而提高模型的泛化能力。

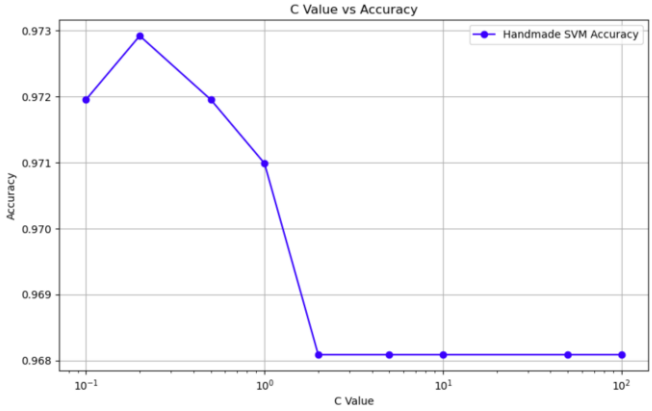


通过调整逻辑回归模型的学习率（alpha）和正则化参数（C），我们发现当 alpha=0.01 且 C=0.1 时，模型在测试集上达到了最高准确率（约为 0.97）。较低的学习率（如 alpha=0.001）表现稳定，但效果略逊，而较高的学习率（如 alpha=0.1）在正则化参数增大时表现不佳，可能导致模型不稳定。因此，alpha=0.01 和 C=0.1 是该实验中的最优参数组合，使模型在准确率和稳定性之间达到了良好平衡。

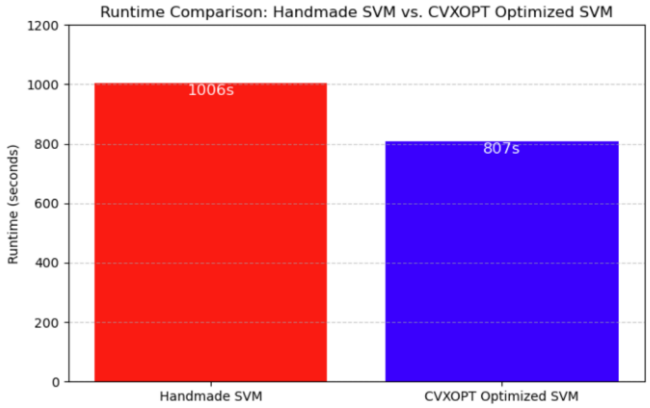
2.6.2 SVM 模型分析与优化

通过调整 SVM 模型的正则化强度参数 C，以优化模型在分类任务上的表现。我们希望通过实验找到一组合适的 C 值，使模型能够在测试集上达到更高的准确率，从而提高模型的泛化能力。

在选择线性核的情况下，我们对 C 值进行了多组测试。实验结果显示，当 C=0.2 时，手写 SVM 模型在测试集上达到了最高准确率（约为 0.973）。较低的 C 值（如 C=0.1）虽然表现稳定，但效果略逊，而较高的 C 值（如 C=10 及以上）会导致模型在测试集上表现较差，可能是由于模型过拟合。综合考虑，C=0.2 是该实验中的最优参数，使模型在准确率和稳定性之间达到了良好平衡。



同时，我们引入 CVXOPT 库，对 SVM 的二次规划求解部分进行了替代优化，再次训练模型并记录优化后的运行时间。实验环境和参数设置保持一致，避免了因环境因素导致的性能偏差。



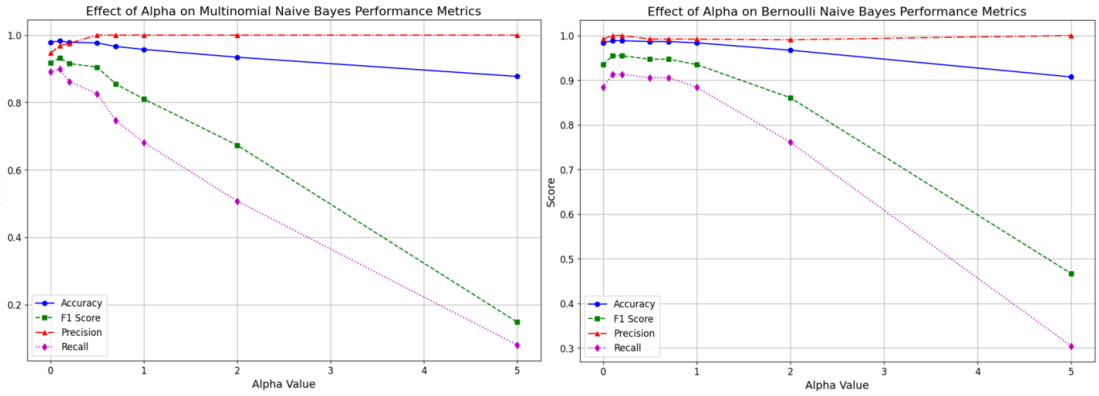
为了更直观地展示性能提升，我们绘制了运行时间的对比条形图。结果表明，引入 CVXOPT 后，运行时间减少了约 **20.5%**，大幅缩短了模型训练时间，验证了使用优化求解器的有效性。

引入 CVXOPT 优化后，SVM 模型的运行时间明显减少，原因在于 CVXOPT 库内部使用了高效的凸优化算法和底层 C 实现，能够更快速地求解二次规划问题。相比于手写实现，CVXOPT 的求解器在处理大规模优化问题时表现更为出色，尤其是在数据量较大的情况下，提升尤为明显。

2.6.3 朴素贝叶斯模型分析与优化

通过上方的实验结果和数据集的情况可知，当前任务的数据集类别不平衡，部分类的特征出现频率较低，模型很容易忽略这些特征。那么通过适当调小 Lidstone 平滑系数 α 可以减少未见特征的权重，使模型对少数类的特征更加敏感，进而提升召回率（Recall）和 F1 分数。下面一组实验验证了上述论述：下面一组实验中 α 的取值均为[0.001, 0.1, 0.2, 0.5, 0.7, 1.0, 2.0, 5.0]，其中 0.001 为了表示不加平滑处理时的情况（避免出现“零概率”问

题）。



2.6.4 不同模型间的比较分析

1、模型结果评估

我们使用 accuracy 准确率来评估模型的性能，准确率为测试集中预测正确的样本数与测试集总样本数的比值，其值越大说明模型性能越好。

各种传统分类方法在 accuracy 上的比较

	Accuracy
逻辑回归(GD)	0.97
逻辑回归(SGD)	0.97
SVM(Linear)	0.969
SVM(Polynomial)	0.851
SVM(RBF)	0.879
朴素贝叶斯(多项式)	0.957
朴素贝叶斯(伯努利)	0.983

从准确率的角度来看，朴素贝叶斯是此任务中表现最好的方法，特别是伯努利朴素贝叶斯的表现尤为突出。逻辑回归的两种方法准确率均为 **0.97**，表现非常一致，表明逻辑回归在这组数据上的表现相当稳定且优异。对于 SVM，线性核效果较好，而多项式核则相对较差，RBF 核的表现有所提升，但仍不如线性核。

2、模型结果分析

分别从数据的线性特性、模型的假设和特性以及模型复杂度三个方面分析模型结果。

1. **数据的线性特性：**

   TF-IDF 的作用在于将文本转换为稀疏的高维向量，这种转换可以增强数据的可分性，尤其在高维空间中，文本样本的边界更为明显。文本数据经过 TF-IDF 特征提取后，在高维空间中接近线性可分，适合逻辑回归、SVM 线性核这些线性模型。

2. **模型的假设和特性：**

   伯努利朴素贝叶斯模型假设特征是相互独立的二值特征，这种假设非常符合文本数据的特性，尤其是在经过二值化处理的 TF-IDF 特征上。

   对于垃圾邮件检测任务，词汇的出现与否可以很好地用二值特征表示，伯努利朴素贝叶斯利用这种特征来评估词汇对于类别的影响，从而实现较高的分类性能。

3. **模型复杂度：**

   SVM 的多项式核和 RBF 核具有较高的模型复杂度，适合处理复杂的非线性关系。在我们的数据集中，特征经过 TF-IDF 提取后是线性可分的，因此使用多项式

核和 RBF 核反而增加了模型的复杂度，导致模型容易过拟合，从而表现不如线性核和逻辑回归。

3. 总结

这次实验围绕分类任务展开，目标是利用传统的机器学习方法对电子邮件进行分类。实验共手写构建了三种分类模型，包括逻辑回归、SVM 和朴素贝叶斯模型，以对比它们在二分类任务中的表现。

1. 数据预处理：对数据进行了探索性统计分析、数据清洗、数据标准化以及二分化处理得到 TF-IDF 特征。

2. 模型实现与调优：

逻辑回归：适合处理线性可分的数据，模型简单且易于解释，能够很好地拟合线性决策边界。

SVM：通过核函数可以处理复杂的非线性问题，线性核适合线性可分数据，多项式核和 RBF 核适合处理非线性数据。

朴素贝叶斯：基于特征独立性的假设，适用于文本分类等高维稀疏数据，尤其在类别分布不均衡的情况下表现优异。

分别对三种模型基于数据特征进行调优，均取得了更好的模型性能。

3.实验结果与分析：通过 accuracy 值评估不同模型的性能，并对得到的模型结果进行对比分析。

此次实验不仅展示了不同回归模型的实现和调优方法，也为团队成员们在数据分析和建模方面提供了宝贵经验。