实验报告（3）

学号：202202206341 姓名：王子俊

1. **实验名称： 神经网络**
2. **实验目的**
3. 理解人工神经网络
4. 了解sklearn库
5. 了解jieba库
6. 了解pandas库
7. **实验内容**

在给定的数据集上，训练神经网络模型，输出预测结果及其准确度。

1. **实验步骤**
2. **数据集准备。**

数据集“产品评价.xlsx”收集了某电商平台上销售的iPhone XR的1080条评价，其中有600条好评，480条差评。

1. **中文分词。**  
    提示：该数据集中的评论为文本数据，不能直接拿来训练，需要将文本分词，构建词频矩阵，再用来拟合模型。可以利用jieba库[[1]](#footnote-1)的cut()可以进行中文分词。
2. **特征提取。**

提示：使用sklearn.feature\_extraction.text中CountVectorizer的fit\_transform()将分词结果进行文本向量化，以构造特征变量。

1. **模型训练。**

提示：使用train\_test\_split()将数据拆分为训练集和测试集，使用sklearn.neural\_network.MLPClassifier的fit()进行模型训练。

1. **模型预测。**

提示：使用predict()进行预测，使用sklearn.metrics.accuracy\_score对预测结果进行评估。

1. **结果分析。**
2. **实验结果与分析**

import pandas as pd #基础库导入

import jieba

from sklearn.feature\_extraction.text import CountVectorizer

from sklearn.model\_selection import train\_test\_split

from sklearn.neural\_network import MLPClassifier

from sklearn.metrics import accuracy\_score, classification\_report, confusion\_matrix

import matplotlib.pyplot as plt

import seaborn as sns

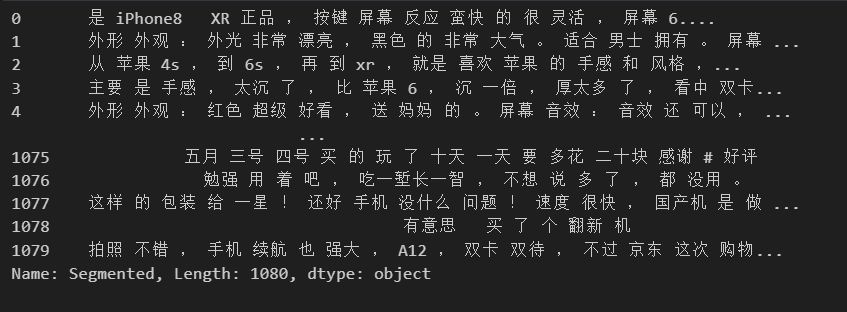
# 数据集准备

data = pd.read\_excel('产品评价.xlsx')  # 读取数据集

data.columns = ['ID', 'Comment', 'Label']  # 假设列名为 ID, Comment, Label

# 中文分词

data['Segmented'] = data['Comment'].apply(lambda x: " ".join(jieba.cut(x)))



# 特征提取

vectorizer = CountVectorizer()

X = vectorizer.fit\_transform(data['Segmented'])  # 分词后的评论向量化

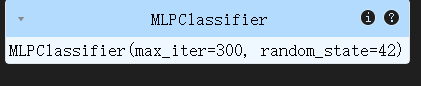
y = data['Label']  # 评价标签

# 模型训练

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)  # 拆分训练集和测试集

clf = MLPClassifier(hidden\_layer\_sizes=(100,), max\_iter=300, random\_state=42)  # 创建并配置MLP分类器

clf.fit(X\_train, y\_train)  # 训练模型



# 模型预测

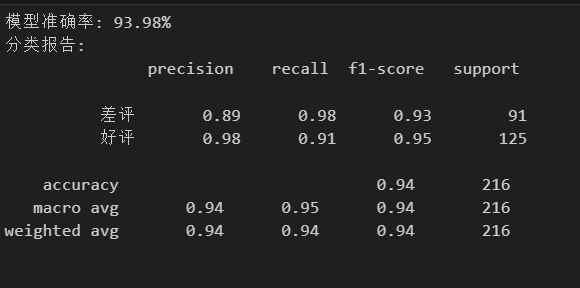
y\_pred = clf.predict(X\_test)

accuracy = accuracy\_score(y\_test, y\_pred)  # 计算准确率

# 结果分析

print("模型准确率: {:.2f}%".format(accuracy \* 100))

print("分类报告:\n", classification\_report(y\_test, y\_pred, target\_names=['差评', '好评']))



import matplotlib.pyplot as plt

import seaborn as sns

# 设置字体为SimHei，避免中文乱码

plt.rcParams['font.sans-serif'] = ['SimHei']

plt.rcParams['axes.unicode\_minus'] = False  # 避免负号显示为方块

#绘制混淆矩阵

conf\_matrix = confusion\_matrix(y\_test, y\_pred)

plt.figure(figsize=(6, 5))

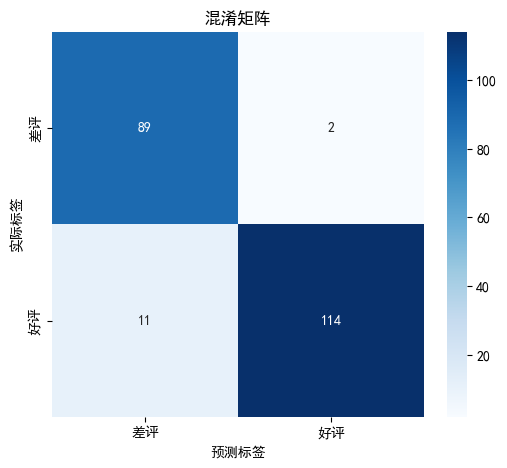
sns.heatmap(conf\_matrix, annot=True, fmt="d", cmap="Blues", xticklabels=['差评', '好评'], yticklabels=['差评', '好评'])

plt.xlabel('预测标签')

plt.ylabel('实际标签')

plt.title('混淆矩阵')

plt.show()



1. 使用了 MLPClassifier 进行 iPhone XR 产品评价的情感分类任务。数据集包含 1080 条评价，其中 600 条为好评，480 条为差评。在模型训练中，样本被拆分为训练集和测试集，使用 80% 数据进行训练，20% 数据用于测试，最终测试集样本量为 216 条。模型使用 CountVectorizer 提取特征，使用中文分词工具进行分词后，再进行特征向量化。

2. 结果分析

模型准确率：模型的整体准确率达到了 93.98%，表明模型对评论的情感分类表现较为准确。

分类报告分析：

差评的分类：精确率（Precision）：0.89 表示模型预测为“差评”的样本中，89% 是实际的差评。召回率（Recall）：0.98 表示实际为“差评”的样本中有 98% 被正确预测。F1 得分：0.93，反映了精确率和召回率的良好平衡。

好评的分类：精确率：0.98 表示模型预测为“好评”的样本中，98% 是实际的好评。召回率：0.91 表示实际为“好评”的样本中有 91% 被正确预测。F1 得分：0.95，表明在“好评”类别中模型表现非常稳定。

宏平均 (Macro Avg)：宏平均精确率和召回率分别为 0.94 和 0.95，进一步表明了模型在两个类别上的总体均衡性能。

加权平均 (Weighted Avg)：加权平均精确率和召回率均为 0.94，体现了模型在数据集中类别不平衡下的良好适应性。

3. 混淆矩阵分析

混淆矩阵表明：真实标签为差评的 91 个样本中，89 个被正确分类为差评，仅 2 个被误分类为好评。真实标签为好评的 125 个样本中，114 个被正确分类为好评，11 个被误分类为差评。

4. 总结

优点：模型在情感分类的精度和召回率上表现非常出色，特别是在好评的识别上，精度达到了 98%，整体准确率接近 94%，适用于实际场景的情感分类需求。

1. **实验总结**

算法特点：MLPClassifier 是一种前馈神经网络，适合处理复杂、非线性特征的分类问题。通过多层神经元连接，MLP 能够自动提取高阶特征。结合词频矩阵的特征提取方法，算法能有效分类带有情感倾向的文本。其表现优良，对情感分类准确度高，但在数据量增加时，计算资源需求和训练时间也显著提高。

实验难点：本实验的难点主要在于中文文本数据的预处理与特征提取，尤其是分词质量会直接影响分类效果。此外，MLP 的超参数设置较为敏感，如隐藏层数、神经元数量和学习率的调整需要通过反复实验进行优化。针对类别不平衡问题，也需要在训练过程中确保模型对好评和差评有良好的分类能力，而不过度偏向多数类。

实现效率：在实现效率上，MLP 适合小规模数据，因其非线性结构和多层网络需要较高的计算资源。在大数据集上，训练时间较长，收敛速度较慢（本次训练时长7.4s）。实验中使用 CountVectorizer 特征提取结合 MLP 分类器，使得文本分类任务可以快速实现，但在实际场景中的高效应用仍需考虑更多优化和资源。

1. https://pypi.org/project/jieba/ [↑](#footnote-ref-1)