(一)假新闻检测（我的建议）

• 数据集：fakenews.csv

fake：<https://www.kaggle.com/datasets/iamrahulthorat/fakenews-csv/data>

包含4729个唯一值，标记为“real”或“fake”。

• 难度：中等

涉及文本处理和二分类问题。

• 优点：

• 数据集较小，易于加载和处理。

• 主题贴近生活，容易理解。

• 可以学习文本预处理、特征提取和简单的机器学习模型（如逻辑回归、朴素贝叶斯）。

技术点：

• 文本预处理（分词、去除停用词、词干提取）。

• 特征提取（TF-IDF、Word2Vec）。

• 分类算法（逻辑回归、朴素贝叶斯、简单的神经网络）。

• 适用人群：对自然语言处理（NLP）感兴趣，希望从简单的文本分类任务入手。

(二)财经新闻的情绪分析

• 数据集：Sentiment Analysis for Financial News

包含两列：“Sentiment”和“News Headline”，情绪分为消极、中性、积极。

• 难度：中等

与假新闻检测类似，但情绪分类是多分类问题（3类）。

• 优点：

• 数据集结构简单，易于理解。

• 可以学习多分类问题的处理方法。

• 主题与金融相关，适合对财经领域感兴趣的人。

• 技术点：

• 文本预处理和特征提取（与假新闻检测类似）。

• 多分类算法（逻辑回归、SVM、简单的神经网络）。

• 适用人群：对NLP和金融领域感兴趣，希望学习多分类问题。

(三)社交媒体情绪分析

• 数据集：Sentiment Analysis

具体内容不详，但通常包含社交媒体文本和情绪标签。

• 难度：中等

与前两个主题类似，但社交媒体文本可能包含更多口语化表达和噪声。

• 优点：

• 数据集可能较大，适合学习大规模数据处理。

• 主题贴近生活，容易理解。

• 技术点：

• 文本预处理（处理表情符号、缩写等）。

• 特征提取和分类算法。

• 适用人群：对社交媒体分析和NLP感兴趣，希望处理更复杂的文本数据。

(四)有害的大脑活动分类

• 数据集：HMS-Harmful Brain Activity Classification

包含脑电图（EEG）数据，用于分类癫痫发作等有害脑活动。

• 难度：较高

涉及时间序列数据和医学背景知识。

• 优点：

• 数据集可能包含丰富的特征，适合学习特征工程。

• 主题具有实际应用价值。

• 技术点：

• 时间序列处理（傅里叶变换、小波变换）。

• 特征提取和分类算法（SVM、深度学习中的CNN）。

• 适用人群：对信号处理、医学数据分析感兴趣，有一定数学基础。

(五)手写识别 PyTorch内部数据集：KMNIST

手写日文字母的数据集，类似于MNIST。

• 难度：中等

涉及图像分类问题。

• 优点：

• 数据集结构简单，与MNIST类似，易于上手。

• 可以学习图像预处理和卷积神经网络（CNN）。

• 技术点：

• 图像预处理（归一化、增强）。

• 卷积神经网络（CNN）。

• 适用人群：对计算机视觉和深度学习感兴趣，希望从简单的图像分类任务入手。

(六)图像分类

• 数据集：[CIFAR-100]()

包含100个类别的小图像数据集。

• 难度：较高

数据集较大，图像分类任务较为复杂。

• 优点：

• 数据集经典，资源丰富。

• 可以学习高级卷积神经网络架构（如ResNet）。

• 技术点：

• 图像预处理和增强。

• 卷积神经网络（CNN）和数据增强技术。

• 适用人群：对计算机视觉和深度学习感兴趣，有一定编程基础。

个人建议从(一)假新闻检测或(五)手写识别入手：

• 假新闻检测：

• 数据集简单，易于处理。

• 可以学习文本处理和分类算法。

• 适合对NLP感兴趣的人。

• 手写识别（KMNIST）：

数据集结构简单，与MNIST类似。

可以学习图像处理和CNN。

适合对计算机视觉感兴趣的人。

模型设计。对于您的项目，您需要考虑以下问题并编写代码来选择您的模型：

1. 模型选择：选择合适的深度学习模型。

2. 损失函数：选择合适的损失函数。

3. 优化器：选择合适的优化器。

4. 超参数：设置超参数，如批量大小（batch size）、学习率（learning rate）、训练轮数（epochs）等。

5. 微调与预训练：对于大型语言模型或Transformer模型，考虑是否使用预训练模型和微调。

性能可视化

训练与测试性能选择一个模型进行以下可视化：

1. 训练损失、训练准确率和测试准确率随训练轮数的变化（使用验证集或其他方法确定轮数）。

2. 使用不同损失函数时的性能变化。

3. 使用不同学习率（如0.1、0.01、0.001、0.0001）时的性能变化。

4. 使用不同批量大小（如8、16、32、64、128）时的性能变化。

5. 测试集前100个结果的预测标签、输入数据和真实标签的可视化。

项目规划与落地

1. 第1周：项目选题与数据集选择

• 确定项目主题和数据集。

• 下载并初步探索数据集。

2. 第2周：数据预处理

• 数据清洗、特征提取和数据增强。

• 编写数据预处理代码。

3. 第3周：模型设计

• 选择合适的深度学习模型。

• 编写模型代码，设置损失函数、优化器和超参数。

4. 第4周：模型训练与调优

• 训练模型，调整超参数。

• 记录训练过程中的性能指标。

5. 第5周：性能可视化

• 绘制训练损失、准确率等性能指标的变化曲线。

• 可视化测试集结果。

6. 第6周：撰写报告与准备演示

• 撰写项目报告。

• 准备演示PPT和视频。

项目落地

1. 代码实现

• 使用Python和PyTorch框架实现模型（代码结构清晰、注释详细）

2. 报告撰写（报告不少于2000字（3月初），包含以下内容：

• 引言：介绍项目背景、目标和当前流行技术。

• 设计与功能：说明系统的主要功能、使用的DL技术和模型及其原因。

• 演示与性能：描述数据集，展示模型功能与性能，演示实现与相关代码。

• 结论：总结系统，指出局限性和未来工作。

• 参考文献：引用相关文献。

演示视频：提交应用运行的演示视频链接。

GitHub链接：提交GitHub链接