Task Description:

Your task is to generate a random binary classification dataset and train and evaluate three different algorithms using 10-fold cross-validation. For certain algorithms, an inner 5-fold cross-validation should be implemented to select the optimal hyperparameters. Report the classification performance metrics for each fold as well as the average across all folds, followed by a concise discussion of the results. Please upload your report (docx or pdf format) and your source code.

Bonus:

For the following dataset containing both training (adult.data) and test sets (adult.test), train and evaluate the same three algorithms. In your analysis, consider the following: Which features (attributes) are used by each algorithm? How are these features processed or transformed before training? How does using only a subset of features affect the model's predictive performance—does it improve or reduce accuracy?

任务报告

1. 概述

采用的三类算法：

Logistic Regression
Support Vector Machine（SVM）
Random Forest

实验分为两部分：

随机生成的二分类数据集

使用 10-fold 外层交叉验证评估模型性能
使用 5-fold 内层交叉验证（GridSearch）选择最佳超参数
输出每折指标（Accuracy, Precision, Recall, F1）及平均结果

UCI Adult 数据集（adult.data / adult.test）

对所有特征进行预处理（数值标准化 / 类别 One-Hot）后训练
使用子集特征进行对比
分析特征选择对模型性能的影响

2. 随机二分类数据集实验结果

2.1 Logistic Regression

总体表现
平均 Accuracy：0.8460　　平均 Precision：0.8228
平均 Recall：0.8816　　平均 F1：0.8504

Logistic Regression 的结果较为稳健，召回率（0.88）较高但精度略低，符合其线性模型简洁、鲁棒性高的特点。

2.2 SVM（RBF Kernel）
总体表现
平均 Accuracy：0.9310　　平均 Precision：0.9188
平均 Recall：0.9458　　平均 F1：0.9319
SVM 在该随机数据集表现最佳。其 RBF 核能有效捕获非线性结构，因此整体指标大幅优于线性模型 Logistic Regression。

2.3 Random Forest
总体表现
平均 Accuracy：0.9170　　平均 Precision：0.9131
平均 Recall：0.9217　　平均 F1：0.9171
Random Forest 的表现略逊于 SVM，但仍明显优于 Logistic Regression。其表现稳定、方差较低，说明树模型对噪声的鲁棒性良好。

2.4 三模型对比（随机数据集）

模型	Accuracy	Precision	Recall	F1	排名
SVM	0.9310	0.9188	0.9458	0.9319	1
Random Forest	0.9170	0.9131	0.9217	0.9171	2
Logistic Regression	0.8460	0.8228	0.8816	0.8504	3

SVM > Random Forest > Logistic Regression
SVM 核技巧赋予其强大的非线性建模能力，是该实验背景下的最佳选择。

3. Adult 数据集实验结果

Adult 数据集包含结构化的类别与数值混合特征，目标为预测收入是否超过 50K。
所有模型均在以下两种设置下比较：
使用全部 14 个原始输入特征（数值 6 +类别 8）
使用特征子集（手工挑选的 8 个特征）

3.1 Logistic Regression

实验	最佳 CV Accuracy	测试 Accuracy
全部特征	0.8523	0.8530
特征子集	0.8488	0.8495

全部特征表现更好。

Logistic Regression 属于线性模型，过度减少特征可能导致信息损失，因此略降。

3.2 SVM (RBF Kernel)

实验	最佳 CV Accuracy	测试 Accuracy
全部特征	0.8565	0.8597
特征子集	0.8557	0.8542

全部特征表现明显更好。
SVM 常受益于高维空间（经 One-Hot 后维度较高），故去掉特征会降低性能。

3.3 Random Forest

实验	最佳 CV Accuracy	测试 Accuracy
全部特征	0.8649	0.8652
特征子集	0.8625	0.8627

全部特征表现略优。
Random Forest 在 Adult 数据集中表现最好。
树模型天然支持非线性和不同类型的特征，也对噪声更加鲁棒。

4. 特征处理与模型使用的特征分析

4.1 数值特征处理方式

对数值特征使用 StandardScaler 标准化
标准化对 Logistic Regression、SVM 至关重要
对 Random Forest 则影响不大（树模型不依赖距离度量）

4.2 类别特征处理方式

使用 One-Hot Encoder 将类别变量展开为稀疏高维特征
影响：
Logistic Regression：线性可分性增强
SVM：高维空间有利于核函数分类
Random Forest：能自然处理 One-Hot，提升树分裂的表达能力

4.3 特征子集的影响

实验现象

在三个模型中，全部特征均优于子集特征，但差距非常小（0.003 到 0.006 之间）。

原因讨论

Adult 数据集的信息主要集中于几个核心特征：
education-num
capital-gain
hours-per-week

marital-status

子集选择已包含这些主特征，因此性能下降较小

对于 SVM 和 Logistic Regression，降维可能移除具有补充判别力的类别变量。对于 Random Forest，类别 One-Hot 扩展可帮助树模型构造更多有效分裂。

5. 综合对比与结论

5.1 随机数据集

SVM（RBF）表现最佳，Random Forest 次之，Logistic Regression 最弱。

说明随机生成的数据可能具有较强非线性结构。SVM 的核方法最能捕获该结构。RF 也能处理复杂关系但略逊。Logistic Regression 作为线性模型能力有限。

5.2 Adult 数据集

最佳模型：Random Forest（Test Accuracy = 0.8652）

Adult 数据结构复杂，有大量类别与非线性关系。Random Forest 天然适合这种 Tabular 数据。

第二名：SVM（0.8597）

虽然高维能力强，但 Adult 数据类别较多、样本大，计算成本高。

第三名：Logistic Regression（0.8530）

性能最稳定，但表达能力有限。

5.3 特征数量的影响

三个模型均为全特征更佳。

结论：更多特征能够略微提升性能，但提升幅度很小。说明特征子集已经覆盖了主要信息，但移除剩余特征仍有轻微损失。

6. 总结

本实验系统比较了三种经典分类算法在随机数据集与 Adult 数据集上的表现，并分析了特征工程与模型表现之间的关系。

最终结论如下：

随机数据集：SVM > Random Forest > Logistic Regression

Adult 数据集：Random Forest > SVM > Logistic Regression

特征工程与特征数量至关重要，全部特征略优于特征子集。

附录：程序运行结果

【Logistic Regression】

===== Part 1: 随机二分类数据集上的 Logistic Regression 嵌套交叉验证 =====

--- 外层 Fold 1 ---
最佳超参数: {'clf__C': 0.1, 'clf__penalty': 'l1'}
内层 CV 最佳平均 Accuracy: 0.8467
外层测试集: Accuracy=0.8400, Precision=0.8148, Recall=0.8800, F1=0.8462

--- 外层 Fold 2 ---
最佳超参数: {'clf__C': 0.1, 'clf__penalty': 'l1'}
内层 CV 最佳平均 Accuracy: 0.8478
外层测试集: Accuracy=0.8100, Precision=0.8298, Recall=0.7800, F1=0.8041

--- 外层 Fold 3 ---
最佳超参数: {'clf__C': 0.1, 'clf__penalty': 'l1'}
内层 CV 最佳平均 Accuracy: 0.8433
外层测试集: Accuracy=0.8800, Precision=0.8393, Recall=0.9400, F1=0.8868

--- 外层 Fold 4 ---
最佳超参数: {'clf__C': 0.1, 'clf__penalty': 'l1'}
内层 CV 最佳平均 Accuracy: 0.8544
外层测试集: Accuracy=0.7900, Precision=0.7736, Recall=0.8200, F1=0.7961

--- 外层 Fold 5 ---
最佳超参数: {'clf__C': 0.1, 'clf__penalty': 'l1'}
内层 CV 最佳平均 Accuracy: 0.8567
外层测试集: Accuracy=0.7900, Precision=0.7458, Recall=0.8800, F1=0.8073

--- 外层 Fold 6 ---
最佳超参数: {'clf__C': 0.1, 'clf__penalty': 'l1'}
内层 CV 最佳平均 Accuracy: 0.8389
外层测试集: Accuracy=0.8400, Precision=0.8148, Recall=0.8800, F1=0.8462

--- 外层 Fold 7 ---
最佳超参数: {'clf__C': 0.1, 'clf__penalty': 'l1'}
内层 CV 最佳平均 Accuracy: 0.8422
外层测试集: Accuracy=0.8900, Precision=0.8421, Recall=0.9600, F1=0.8972

--- 外层 Fold 8 ---
最佳超参数: {'clf__C': 0.1, 'clf__penalty': 'l1'}

内层 CV 最佳平均 Accuracy: 0.8456
外层测试集: Accuracy=0.8900, Precision=0.8824, Recall=0.9000, F1=0.8911

--- 外层 Fold 9 ---
最佳超参数: {'clf__C': 0.1, 'clf__penalty': 'l1'}
内层 CV 最佳平均 Accuracy: 0.8456
外层测试集: Accuracy=0.7900, Precision=0.7800, Recall=0.7959, F1=0.7879

--- 外层 Fold 10 ---
最佳超参数: {'clf__C': 0.1, 'clf__penalty': 'l1'}
内层 CV 最佳平均 Accuracy: 0.8356
外层测试集: Accuracy=0.9400, Precision=0.9057, Recall=0.9796, F1=0.9412

=== 每折结果 ===

Fold	accuracy		precision	recall	f1
1	0.8400	0.8148	0.8800	0.8462	
2	0.8100	0.8298	0.7800	0.8041	
3	0.8800	0.8393	0.9400	0.8868	
4	0.7900	0.7736	0.8200	0.7961	
5	0.7900	0.7458	0.8800	0.8073	
6	0.8400	0.8148	0.8800	0.8462	
7	0.8900	0.8421	0.9600	0.8972	
8	0.8900	0.8824	0.9000	0.8911	
9	0.7900	0.7800	0.7959	0.7879	
10	0.9400	0.9057	0.9796	0.9412	

=== 平均结果 ===
accuracy: mean=0.8460, std=0.0496
precision: mean=0.8228, std=0.0463
recall: mean=0.8816, std=0.0640
f1: mean=0.8504, std=0.0493

===== Part 2: Adult 数据集上的 Logistic Regression 实验 =====
训练集大小: (32561, 14), 测试集大小: (16281, 14)

--- 实验 A: 使用所有特征 ---
使用的数值特征: ['age', 'fnlwgt', 'education-num', 'capital-gain', 'capital-loss', 'hours-per-week']
使用的类别特征: ['workclass', 'education', 'marital-status', 'occupation', 'relationship', 'race', 'sex', 'native-country']
最佳超参数（所有特征）: {'clf__C': 0.1, 'clf__penalty': 'l2'}
训练集 5-fold CV 最佳平均 Accuracy（所有特征）: 0.8523
测试集 Accuracy（所有特征）: 0.8530

--- 实验 B: 使用特征子集 ---

特征子集: ['age', 'education-num', 'hours-per-week', 'capital-gain', 'capital-loss', 'marital-status', 'occupation', 'sex']
子集中数值特征: ['age', 'education-num', 'hours-per-week', 'capital-gain', 'capital-loss']
子集中类别特征: ['marital-status', 'occupation', 'sex']
最佳超参数（特征子集）: {'clf__C': 0.1, 'clf__penalty': 'l1'}
训练集 5-fold CV 最佳平均 Accuracy（特征子集）: 0.8488
测试集 Accuracy（特征子集）: 0.8495

=== 对比：全部特征 vs 特征子集 ===
测试集 Accuracy（全部特征）: 0.8530
测试集 Accuracy（特征子集）: 0.8495
使用全部特征在测试集上表现更好。

【SVM】

===== Part 1: 随机二分类数据集上的 SVM 嵌套交叉验证 =====

--- 外层 Fold 1 ---
最佳超参数: {'clf__C': 1.0, 'clf__gamma': 'scale', 'clf__kernel': 'rbf'}
内层 CV 最佳平均 Accuracy: 0.9211
外层测试集: Accuracy=0.9400, Precision=0.9400, Recall=0.9400, F1=0.9400

--- 外层 Fold 2 ---
最佳超参数: {'clf__C': 1.0, 'clf__gamma': 'scale', 'clf__kernel': 'rbf'}
内层 CV 最佳平均 Accuracy: 0.9222
外层测试集: Accuracy=0.9600, Precision=0.9600, Recall=0.9600, F1=0.9600

--- 外层 Fold 3 ---
最佳超参数: {'clf__C': 1.0, 'clf__gamma': 'scale', 'clf__kernel': 'rbf'}
内层 CV 最佳平均 Accuracy: 0.9278
外层测试集: Accuracy=0.9400, Precision=0.9074, Recall=0.9800, F1=0.9423

--- 外层 Fold 4 ---
最佳超参数: {'clf__C': 1.0, 'clf__gamma': 'scale', 'clf__kernel': 'rbf'}
内层 CV 最佳平均 Accuracy: 0.9256
外层测试集: Accuracy=0.8800, Precision=0.8519, Recall=0.9200, F1=0.8846

--- 外层 Fold 5 ---
最佳超参数: {'clf__C': 1.0, 'clf__gamma': 'scale', 'clf__kernel': 'rbf'}
内层 CV 最佳平均 Accuracy: 0.9333

外层测试集: Accuracy=0.9200, Precision=0.9038, Recall=0.9400, F1=0.9216

--- 外层 Fold 6 ---
最佳超参数: {'clf__C': 1.0, 'clf__gamma': 'scale', 'clf__kernel': 'rbf'}
内层 CV 最佳平均 Accuracy: 0.9289
外层测试集: Accuracy=0.9100, Precision=0.9184, Recall=0.9000, F1=0.9091

--- 外层 Fold 7 ---
最佳超参数: {'clf__C': 10.0, 'clf__gamma': 'scale', 'clf__kernel': 'rbf'}
内层 CV 最佳平均 Accuracy: 0.9200
外层测试集: Accuracy=0.9200, Precision=0.9038, Recall=0.9400, F1=0.9216

--- 外层 Fold 8 ---
最佳超参数: {'clf__C': 1.0, 'clf__gamma': 'scale', 'clf__kernel': 'rbf'}
内层 CV 最佳平均 Accuracy: 0.9222
外层测试集: Accuracy=0.9900, Precision=0.9804, Recall=1.0000, F1=0.9901

--- 外层 Fold 9 ---
最佳超参数: {'clf__C': 1.0, 'clf__gamma': 'scale', 'clf__kernel': 'rbf'}
内层 CV 最佳平均 Accuracy: 0.9333
外层测试集: Accuracy=0.9000, Precision=0.8824, Recall=0.9184, F1=0.9000

--- 外层 Fold 10 ---
最佳超参数: {'clf__C': 1.0, 'clf__gamma': 'scale', 'clf__kernel': 'rbf'}
内层 CV 最佳平均 Accuracy: 0.9211
外层测试集: Accuracy=0.9500, Precision=0.9400, Recall=0.9592, F1=0.9495

=== 每折结果 ===

Fold	accuracy		precision	recall f1
1	0.9400	0.9400	0.9400	0.9400
2	0.9600	0.9600	0.9600	0.9600
3	0.9400	0.9074	0.9800	0.9423
4	0.8800	0.8519	0.9200	0.8846
5	0.9200	0.9038	0.9400	0.9216
6	0.9100	0.9184	0.9000	0.9091
7	0.9200	0.9038	0.9400	0.9216
8	0.9900	0.9804	1.0000	0.9901
9	0.9000	0.8824	0.9184	0.9000
10	0.9500	0.9400	0.9592	0.9495

=== 平均结果 ===
accuracy: mean=0.9310, std=0.0301
precision: mean=0.9188, std=0.0357
recall: mean=0.9458, std=0.0285

f1: mean=0.9319, std=0.0294

===== Part 2: Adult 数据集上的 SVM 实验 =====
训练集大小: (32561, 14), 测试集大小: (16281, 14)

--- 实验 A: 使用所有特征 ---
使用的数值特征: ['age', 'fnlwgt', 'education-num', 'capital-gain', 'capital-loss', 'hours-per-week']
使用的类别特征: ['workclass', 'education', 'marital-status', 'occupation', 'relationship', 'race', 'sex', 'native-country']
最佳超参数（所有特征）: {'clf__C': 1.0, 'clf__gamma': 'scale', 'clf__kernel': 'rbf'}
训练集 5-fold CV 最佳平均 Accuracy（所有特征）: 0.8565
测试集 Accuracy（所有特征）: 0.8597

--- 实验 B: 使用特征子集 ---
特征子集: ['age', 'education-num', 'hours-per-week', 'capital-gain', 'capital-loss', 'marital-status', 'occupation', 'sex']
子集中数值特征: ['age', 'education-num', 'hours-per-week', 'capital-gain', 'capital-loss']
子集中类别特征: ['marital-status', 'occupation', 'sex']
最佳超参数（特征子集）: {'clf__C': 10.0, 'clf__gamma': 'scale', 'clf__kernel': 'rbf'}
训练集 5-fold CV 最佳平均 Accuracy（特征子集）: 0.8557
测试集 Accuracy（特征子集）: 0.8542

=== 对比：全部特征 vs 特征子集 ===
测试集 Accuracy（全部特征）: 0.8597
测试集 Accuracy（特征子集）: 0.8542
使用全部特征在测试集上表现更好。

【Random Forest】

===== Part 1: 随机二分类数据集上的 Random Forest 嵌套交叉验证 =====

--- 外层 Fold 1 ---
最佳超参数: {'clf__max_depth': 10, 'clf__max_features': 'sqrt', 'clf__min_samples_split': 2, 'clf__n_estimators': 200}
内层 CV 最佳平均 Accuracy: 0.9167
外层测试集: Accuracy=0.9100, Precision=0.9184, Recall=0.9000, F1=0.9091

--- 外层 Fold 2 ---
最佳超参数: {'clf__max_depth': 10, 'clf__max_features': 'sqrt', 'clf__min_samples_split': 5, 'clf__n_estimators': 200}

内层 CV 最佳平均 Accuracy: 0.9133
外层测试集: Accuracy=0.9200, Precision=0.9038, Recall=0.9400, F1=0.9216

--- 外层 Fold 3 ---
最佳超参数: {'clf__max_depth': None, 'clf__max_features': 'sqrt', 'clf__min_samples_split': 2, 'clf__n_estimators': 200}
内层 CV 最佳平均 Accuracy: 0.9256
外层测试集: Accuracy=0.9100, Precision=0.8868, Recall=0.9400, F1=0.9126

--- 外层 Fold 4 ---
最佳超参数: {'clf__max_depth': 10, 'clf__max_features': 'sqrt', 'clf__min_samples_split': 5, 'clf__n_estimators': 200}
内层 CV 最佳平均 Accuracy: 0.9233
外层测试集: Accuracy=0.9100, Precision=0.9020, Recall=0.9200, F1=0.9109

--- 外层 Fold 5 ---
最佳超参数: {'clf__max_depth': None, 'clf__max_features': 'sqrt', 'clf__min_samples_split': 2, 'clf__n_estimators': 200}
内层 CV 最佳平均 Accuracy: 0.9267
外层测试集: Accuracy=0.8900, Precision=0.8824, Recall=0.9000, F1=0.8911

--- 外层 Fold 6 ---
最佳超参数: {'clf__max_depth': None, 'clf__max_features': 'sqrt', 'clf__min_samples_split': 5, 'clf__n_estimators': 100}
内层 CV 最佳平均 Accuracy: 0.9144
外层测试集: Accuracy=0.9100, Precision=0.9184, Recall=0.9000, F1=0.9091

--- 外层 Fold 7 ---
最佳超参数: {'clf__max_depth': 10, 'clf__max_features': 'sqrt', 'clf__min_samples_split': 5, 'clf__n_estimators': 100}
内层 CV 最佳平均 Accuracy: 0.9100
外层测试集: Accuracy=0.9100, Precision=0.8868, Recall=0.9400, F1=0.9126

--- 外层 Fold 8 ---
最佳超参数: {'clf__max_depth': None, 'clf__max_features': 'sqrt', 'clf__min_samples_split': 5, 'clf__n_estimators': 100}
内层 CV 最佳平均 Accuracy: 0.9144
外层测试集: Accuracy=0.9400, Precision=0.9400, Recall=0.9400, F1=0.9400

--- 外层 Fold 9 ---
最佳超参数: {'clf__max_depth': None, 'clf__max_features': 'sqrt', 'clf__min_samples_split': 5, 'clf__n_estimators': 200}
内层 CV 最佳平均 Accuracy: 0.9111
外层测试集: Accuracy=0.8900, Precision=0.9130, Recall=0.8571, F1=0.8842

--- 外层 Fold 10 ---
最佳超参数: {'clf__max_depth': None, 'clf__max_features': 'sqrt', 'clf__min_samples_split': 2, 'clf__n_estimators': 200}
内层 CV 最佳平均 Accuracy: 0.9122
外层测试集: Accuracy=0.9800, Precision=0.9796, Recall=0.9796, F1=0.9796

=== 每折结果 ===

Fold	accuracy		precision	recall	f1
1	0.9100	0.9184	0.9000	0.9091	
2	0.9200	0.9038	0.9400	0.9216	
3	0.9100	0.8868	0.9400	0.9126	
4	0.9100	0.9020	0.9200	0.9109	
5	0.8900	0.8824	0.9000	0.8911	
6	0.9100	0.9184	0.9000	0.9091	
7	0.9100	0.8868	0.9400	0.9126	
8	0.9400	0.9400	0.9400	0.9400	
9	0.8900	0.9130	0.8571	0.8842	
10	0.9800	0.9796	0.9796	0.9796	

=== 平均结果 ===
accuracy: mean=0.9170, std=0.0249
precision: mean=0.9131, std=0.0278
recall: mean=0.9217, std=0.0321
f1: mean=0.9171, std=0.0253

===== Part 2: Adult 数据集上的 Random Forest 实验 =====
训练集大小: (32561, 14), 测试集大小: (16281, 14)

--- 实验 A: 使用所有特征 ---
使用的数值特征: ['age', 'fnlwgt', 'education-num', 'capital-gain', 'capital-loss', 'hours-per-week']
使用的类别特征: ['workclass', 'education', 'marital-status', 'occupation', 'relationship', 'race', 'sex', 'native-country']
最佳超参数（所有特征）: {'clf__max_depth': 20, 'clf__max_features': 'sqrt', 'clf__min_samples_split': 5, 'clf__n_estimators': 200}
训练集 5-fold CV 最佳平均 Accuracy（所有特征）: 0.8649
测试集 Accuracy（所有特征）: 0.8652

--- 实验 B: 使用特征子集 ---
特征子集: ['age', 'education-num', 'hours-per-week', 'capital-gain', 'capital-loss', 'marital-status', 'occupation', 'sex']
子集中数值特征: ['age', 'education-num', 'hours-per-week', 'capital-gain', 'capital-loss']
子集中类别特征: ['marital-status', 'occupation', 'sex']
最佳超参数（特征子集）: {'clf__max_depth': 20, 'clf__max_features': 'log2',

'clf__min_samples_split': 5, 'clf__n_estimators': 200}
训练集 5-fold CV 最佳平均 Accuracy（特征子集）: 0.8625
测试集 Accuracy（特征子集）: 0.8627

=== 对比：全部特征 vs 特征子集 ===
测试集 Accuracy（全部特征）: 0.8652
测试集 Accuracy（特征子集）: 0.8627
使用全部特征在测试集上表现更好。