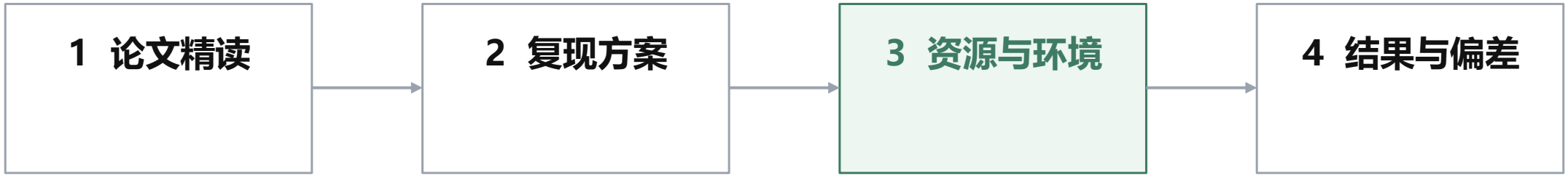
科研素养训练II

第3次课：复现资源管理与实验环境构建

宋庆禹

2026.09.21

本次课程在复现流程中的位置



本次任务：将既定方案落实为能够检查和交接的实验项目

本次课的输入与输出

输入：复现方案

复现目标及其证据位置

数据、方法与评价指标

必须保持一致和允许调整的条件

输出：最小实验包

来源明确的代码与数据

可核查的环境、配置与入口

有效实验运行或完整实验受阻记录

统一最低记录要求，具体工具与组织方式可以不同

真实案例： 新闻组文本的主题分类（20 Newsgroups）

厦門大學 XIAMEN UNIVERSITY

20 Newsgroups是一个英文新闻组帖子数据集，包含约2万篇文本，来自20个不同主题的新闻组。
任务是：给定一篇帖子，构建一个模型预测它属于哪个主题类别。

输入：一条新闻组文本

“I don’ t clearly understand ‘occlusion’ （遮挡） in computer graphics.”

输出：所属主题类别

分类依据：文本中的词语及其统计特征

四个类别示例（总共20个类别）

alt.atheism（无神论）
comp.graphics（计算机图形学）
sci.space（太空）
talk.religion.misc（宗教杂谈）

输入：“I don’t clearly understand ‘occlusion’ in computer graphics.”
真实类别：`comp.graphics`（计算机图形学）
模型任务：根据输入文本预测类别，再与真实标签比较。

S1：https://scikit-learn.org/1.7/auto_examples/text/plot_document_classification_20newsgroups.html

一条文本样本包含哪些信息

真实样本：test/comp.graphics/39075

Subject: Definition of Occlusion

Hi! Everyone,
I don’ t clearly understand “occlusion”
in computer graphics.
Would you please give me an explanation?

需要分析的文本组成

邮件头与主题行
正文
可能存在的签名与引用

NOTE：展示文本已隐去个人身份信息；实验数据仍按既定预处理规则使用。

复现考虑：原示例分别使用保留元信息和移除元信息的数据，复现对应哪一种?

S1：https://scikit-learn.org/1.7/auto_examples/text/plot_document_classification_20newsgroups.html

数据处理：将文本转换为数值矩阵

简化示例：用三条自拟文本说明词频表示
space orbit / graphics image / space image

文本	space	orbit	graphics	image
文本1	1	1	0	0
文本2	0	0	1	1
文本3	1	0	0	1

NOTE：本表仅用于解释表示方式

复现考虑：复现时需要同时核对输入文本、特征提取方法及其参数

S2: https://amueller.github.io/COMS4995-s20/slides/aml-15-text-data/

数据处理：特征矩阵的规模与存储方式

上一页用4个词说明文本的数值表示。
真实实验中的词表可能包含数千个词，但每条文本通常只使用其中一小部分。

文档	词1	词2	词3	词4	…
文档1	0	0.6	0	0	…
文档2	0	0	0	0.8	…
文档3	0.5	0	0	0	…

NOTE：
1. 示意数值，非实际运行输出。每行对应一条文本，每列对应一个词；0表示该文本中没有这个词，非零值表示其TF-IDF权重。
2. TF-IDF权重：衡量一个词对当前文本的重要程度。一个词在当前文本中出现得越多、在其他文本中越少见，其权重通常越高。

本例全量训练矩阵：2034行 × 7831列（保留元信息条件），对应保留元信息条件下的2034条训练文本、7831个词特征

复现考虑：复现时需要核对矩阵形状与存储方式，避免无意转换为稠密矩阵而增加内存开销

S1：https://scikit-learn.org/1.7/auto_examples/text/plot_document_classification_20newsgroups.html

数据处理：两种文本预处理条件

TF-IDF根据输入文本生成特征。
如果输入包含邮件头、签名和引用内容，其中的词语也可能进入特征矩阵。

两种不同数据预处理方式：左右对照

对照项	条件A：保留元信息	条件B：移除指定元信息
加载配置	remove=()	remove=("headers", "footers", "quotes")
输入文本	保留加载器返回的文本，包括正文及元信息	按加载器规则移除邮件头、签名尾部和引用内容
可能形成的特征	正文词语，以及元信息中的词语	主要来自剩余文本的词语，仍可能残留元信息
处理边界	不额外执行上述过滤	过滤并不完美，不能保证只剩纯正文

共同条件：四类别、原训练 / 测试划分、TF-IDF与RidgeClassifier

S1：https://scikit-learn.org/1.7/auto_examples/text/plot_document_classification_20newsgroups.html

RidgeClassifier

RidgeClassifier（岭分类器）是一种带 L2 正则化的线性分类模型，可以理解为：将岭回归用于分类任务。

它的基本思路是：

1. **将类别转成数值标签**：例如二分类中，将两个类别编码为 -1 和 $+1$。
2. **用岭回归拟合这些标签**：既要求预测接近标签，也限制模型权重过大。
3. **根据预测值判定类别**：预测值大于 0 判为 $+1$，否则判为 -1。

其训练目标为：

$$\min_{\mathbf{w},b} \sum_{i=1}^{n} \left(y_i - \mathbf{w}^\top \mathbf{x}_i - b\right)^2 + \alpha \|\mathbf{w}\|_2^2$$

其中，第一项是**平方误差**，第二项是 **L2 正则化**；α 越大，对权重的约束越强。

实验设置：实验流程与资源准备清单

根据原示例的执行流程，逐项确认所需的数据、代码、配置和输出位置

1 数据加载 → 2 特征处理 → 3 训练与预测 → 4 评价与保存

环节	本例执行什么操作?	运行前需要准备什么?
数据加载	读取四类别文本，执行指定的元信息处理	数据缓存、类别列表、训练 / 测试划分及remove配置
特征提取	在训练集上拟合TF-IDF，再转换测试集	特征提取代码、停用词与词频过滤等参数
训练与预测	训练Ridge分类器，预测测试文本类别	分类器实现及参数、所需软件依赖
评价与保存	计算指标，导出预测与运行记录	评价代码、输出目录及文件写入权限

资源准备应严格按照目标实验的执行流程

实验设置：在官方源码中定位实验设置

以scikit-learn 1.7.2官方示例为依据，沿执行流程定位关键代码

主体：四个环节与代码位置

实验环节	源码中的关键接口	重点核对什么?
数据加载	load_dataset(...)	类别、训练 / 测试子集、元信息处理条件
特征提取	TfidfVectorizer(...)	TF-IDF参数，以及在哪个数据集上拟合
训练与预测	RidgeClassifier(...)、fit(...)、predict(...)	分类器参数、训练输入与预测对象
评价	ConfusionMatrixDisplay.from_predictions(...)	使用哪些真实标签与预测结果

示例:

训练: X_train = vectorizer.fit_transform(data_train.data)

测试: X_test = vectorizer.transform(data_test.data)

把方案中的每项关键设置，对应到具体的函数、参数或代码位置

S1: https://scikit-learn.org/1.7/auto_examples/text/plot_document_classification_20newsgroups.html

实验设置：原始代码与本节课封装的对应关系

官方示例中的核心实现	本课新增的教学封装
四类别数据加载与元信息处理	用独立JSON文件选择实验条件
TF-IDF特征提取	保存实际生效的配置与环境信息
Ridge分类器训练与预测	导出逐条预测、指标和运行日志
原示例中的评价与可视化	增加输出完整性检查及故障演示入口

示例：
原代码： load_dataset(remove=("headers", "footers", "quotes"))
新代码： "remove": ["headers", "footers", "quotes"]

新增记录功能不等于更换实验方法；凡是改变数据、参数或执行范围的操作，都需要另行说明

本机核验：verification/upstream/checks.json

实验设置：数据范围与处理条件的记录

仅记录“使用20 Newsgroups”还不够
还需要说明选用了哪些文本、如何划分，以及经过了什么处理

记录项	本课实验的具体内容
数据来源	20news-bydate数据归档，获取路径与校验信息另行保存
类别范围	无神论、计算机图形学、太空、宗教杂谈，共4类
训练 / 测试划分	使用加载器提供的原有train与test子集，不重新划分
样本数量	完整实验：训练2034条，测试1353条
标签对应关系	以加载器返回的target_names为准，记录数字标签对应的类别
文本处理条件	保留元信息，或按规则移除邮件头、签名尾部与引用内容

NOTE：两次运行即使都是“训练2034条、测试1353条”，也可能使用不同的文本处理条件。
样本数量相同，不代表输入内容相同。

数据记录应让他人能够确认：用了哪些样本、怎样划分、怎样处理

来源：本机记录：full-with_metadata / full-without_metadata

实验设置：同名数据集中的不同实验设置对比

对照项	scikit-learn本课案例	CMU历史案例
类别范围	四类别	介绍20类任务及特定二分类
数据划分	既有train / test	50% / 50%随机划分
方法	RidgeClassifier	kNN
重复与汇总	按选定流程记录	课件说明10次平均
预处理细节	源码可定位	仅凭相关页面仍不完整

确定复现目标时，应对应具体实验设置，而不仅是数据集名称

课堂练习：检查数据准备记录

待检查的记录：

数据集：20 Newsgroups

类别：选取四类别，使用既有train / test

准确率：已报告

训练集：2034条；测试集：1353条

当前状态：数据加载成功

问题：

1. 找出其中**两处不够明确的描述。**
2. 对其中一处，写出**需要补充的信息或下一步核查动作。**

加载成功、数量一致，只能确认部分条件；还需要核对样本范围、划分与处理规则

实验设置：实验资源清单与核验状态

资源清单不仅记录“有什么”，还要说明它是否对应目标实验，以及哪些问题尚未解决

资源	身份或来源	状态 / 核查重点
原始代码	scikit-learn 1.7.2	已冻结：保留提交标识、原始脚本与许可
数据	MIT作者原站	已核验：归档校验值与上游记录一致；已检查加载后的类别与数量
实验配置	两份元信息处理配置	已核验；运行记录保存实际生效的参数
软件依赖	requirements-lock.txt	已冻结；已在本机环境完成运行验证
预训练权重	本例不使用	不适用

状态应区分：已核验、待核验、受阻、不适用
每项资源都应能回答：是哪一份、检查到哪一步、还有什么待解决

示例：数据下载受阻，核验替代数据来源

本次课前准备中，默认Figshare下载地址返回HTTP 403，未能取得数据。
随后查阅冻结版本的加载器源码，找到作者原始发布地址

步骤	本次实际处理	为什么需要这一步？
记录失败	保存默认地址及HTTP 403信息	说明资源在哪里受阻，避免误判为模型或算法问题
查找替代来源	根据加载器源码中的说明，访问MIT域名下的作者原站	替代地址具有可追溯的来源依据
核验下载文件	下载20news-bydate.tar.gz，比较文件版本	确认文件内容与预期版本一致，而不只是文件名相同
保存获取记录	保留下载地址、校验值、原始归档及缓存生成记录	让他人能够追查数据从哪里取得、如何进入实验

更换下载路径时核验文件身份；改变数据内容时重新确认实验条件

实验设置： 预处理规则的记录粒度

设置	本例记录	影响环节
remove	() 或 (headers, footers, quotes)	输入文本
stop_words	english	词项保留
min_df / max_df	5 / 0.5	词频过滤
sublinear_tf	True	词频权重
拟合范围	仅训练集fit_transform	避免测试信息进入词表

“已清洗数据”应展开为可定位的操作、参数与拟合范围

S1: https://scikit-learn.org/1.7/auto_examples/text/plot_document_classification_20newsgroups.html

Take away：实验资源的确认

代码是否对应目标实验?	来源与版本明确，能够定位关键实验设置，并区分原实现与本地改动
数据是否符合实验条件?	来源可追溯，类别范围、训练 / 测试划分及文本处理条件明确
尚未解决的问题是否有记录?	区分已核验、待核验和受阻事项，说明下一步检查动作

资源确认后，下一步是搭建运行环境，并检查实验能否按预定条件执行

运行实验需要哪些环境条件?

代码需要解释器执行，也会调用其他软件提供的功能，因此仅下载源码并不能保证实验可以运行

环境组成	本课案例需要什么?	提供什么能力?
硬件与操作系统	CPU、内存及操作系统	提供程序运行和数据存储所需的基础条件
Python解释器	Python	执行实验脚本
计算与建模依赖	NumPy、SciPy、scikit-learn	完成数值计算、稀疏矩阵处理、特征提取与分类
结果展示依赖	Matplotlib	绘制混淆矩阵等图表

示例:

```
from sklearn.feature_extraction.text import TfidfVectorizer
```

环境准备需要同时确认：用什么执行代码，以及代码依赖哪些软件

原代码要求的与实际运行环境的对照

复现时应区分已经明确的要求、尚未说明的信息，以及实际采用的环境

环境项目	原示例或冻结材料提供的信息	本课已验证的运行环境
scikit-learn	选用1.7.2版本的官方示例与源码	1.7.2
Python	所选示例未单独锁定具体小版本	3.12.14
数值计算依赖	需结合发行版依赖要求核对	NumPy 2.2.6、SciPy 1.15.3
绘图依赖	示例使用Matplotlib	Matplotlib 3.10.6
系统与硬件	所选示例未完整说明	macOS、ARM64、CPU运行

“显式”区分来源要求与实际条件，让环境选择有依据、变更可追踪

来源：本机记录：runs/*/environment.json

版本差异示例：旧接口无法调用

① 运行现象 历史课程中的一处具体接口变化

```
vect.get_feature_names()
```

② 原因判断 当前冻结环境中的实际检查：旧接口不可用

```
AttributeError: ... has no attribute  “get_feature_names”
```

③ 适配处理 适配后核对返回值含义，记录修改位置与原因

```
vect.get_feature_names_out()
```

遇到接口报错，先核对版本与文档，再决定恢复旧环境还是适配当前环境

环境配置：独立环境与依赖安装

不同实验可能依赖不同的软件版本
为项目建立独立环境，可以减少安装或升级软件时对其他项目的影响

步骤	示例命令	作用
创建独立环境	python3.12 -m venv .venv	在项目中建立使用Python 3.12的独立环境
安装项目依赖	python -m pip install -r requirements.txt	按依赖文件中的要求安装软件
使用该环境运行	python experiment.py	明确使用该环境的解释器执行脚本

scikit-learn==1.7.2
matplotlib==3.10.6

安装依赖和执行脚本，应使用同一个环境中的Python解释器

环境配置： 环境安装与导入检查

检查层次	检查什么?	能确认什么?
解释器与版本	实际使用的Python路径、Python及主要依赖版本	是否进入了预期环境，版本是否符合计划
依赖导入	尝试导入实验需要的模块	当前解释器能否找到并加载这些依赖
最小运行	用少量有效数据执行实验流程	相关功能能否协同工作，并生成基本有效的输出

示例:

```
import sys
import sklearn
print(sys.executable)
print(sys.version)
print(sklearn.__version__)
```

版本正确、导入成功和最小运行通过，是不同层次的证据，不能相互替代

来源：本机实际导入与环境记录

环境配置：环境记录的范围与工具边界

准备方式	能帮助解决什么？	仍需核对什么？
独立Python环境	隔离不同项目的软件包依赖	Python版本、操作系统及外部系统库
依赖版本文件	记录需要安装的软件包及版本	对应平台是否有可用安装包，依赖是否完整
容器镜像	打包应用及较完整的用户态运行环境	CPU架构、宿主驱动，以及挂载的数据和文件路径

容器可以理解为预先打包的一套应用运行环境，本课不展开其安装与使用

选择适合任务的环境管理方式，并记录它没有覆盖的运行条件

运行配置：用配置明确本次实验条件

Case1: with_metadata.json

```
"remove": []
```

Case2: without_metadata.json

```
"remove": [
  "headers", "footers", "quotes"
]
```

两份配置共同指定
四类别、打乱种子42、向量器参数、分类器参数与冒烟抽样规则

运行前核对配置内容，运行后记录实际生效的设置

运行配置：运行入口与实际生效的配置

```
.venv/bin/python experiment.py \
  --config configs/without_metadata.json \
  --run-id trial-01
```

命令部分	含义
.venv/bin/python	使用已准备的独立环境
experiment.py	指定实验执行入口
--config ...	选择本次实验配置
--run-id trial-01	为本次运行设置一个便于区分的编号

保存完整运行命令与实际生效的配置，才能说明这次实验究竟如何执行

运行配置：随机种子与数据操作的对应关系

数据操作	本课案例的实际设置	需要明确什么？
训练 / 测试划分	使用数据集原有划分	本例没有使用随机种子重新划分数据
数据顺序打乱	random_state=42	控制加载后样本的排列顺序，不改变训练 / 测试样本分布
小规模测试抽样	使用种子1729	控制从各类别抽取哪些样本，并保留样本标识

随机种子应与具体操作一起记录；相同种子不代表相同实验条件

实验记录： 一次运行需要保留哪些记录

运行编号：trial-01

记录内容	需要回答的问题
代码与数据信息	使用了哪个代码版本、哪些数据及处理条件?
实际环境与配置	在什么环境中运行，哪些参数实际生效?
执行过程与状态	使用了什么命令，是否完成，在哪里报错?
指标与实际输出	得到什么结果，对应哪些预测或输出文件?

建议使用不同的编号

每个结果都应能够追溯到产生它的那次运行

实验记录：冒烟测试与完整实验

冒烟测试是正式实验前的小规模运行，用于尽早发现数据加载、特征处理、训练预测和输出保存中的明显问题

对照项	冒烟测试	完整实验
目的	检查流程是否能够贯通，输出是否基本有效	获得目标实验条件下的正式结果
训练 / 测试	使用经过选择的少量样本使用方案规定的完整数据范围	使用经过选择的少量样本使用方案规定的完整数据范围
示例	每类80条训练、30条测试	使用指定四类别的全部样本
重点检查	各环节是否完成，预测与输出是否完整	按既定协议获得结果，供后续比较与分析
结果用途	不能据此判断正式实验效果或复现是否成功	仍需核对条件、分析结果与偏差

先用小规模运行检查流程，再按既定方案执行完整实验；两者的结果用途不能混淆

实验记录：冒烟测试的通过条件

检查环节	本课案例检查什么？
数据输入	样本非空，覆盖四个类别；数量与本次抽样设置一致
特征处理	在训练集上拟合，测试集使用同一转换规则；两者特征列数一致
模型预测	每条测试文本都有预测结果，预测类别属于预期范围
指标计算	例如二分类混淆矩阵计数之和等于测试样本数；准确率与预测记录一致
输出保存	文件能够读取，内容完整，并与本次配置和运行编号对应

通过条件应依据输入规模、实验流程和预期输出确定，而不只是“没有报错”

运行测试：最小运行与实际输出

smoke-without-metadata

实际输出

训练320条，测试120条
特征1050维，预测120条

真\预	A	G	S	R
A	13	1	3	13
G	2	24	1	3
S	1	5	17	7
R	8	3	3	16

A无神论 / G图形学
S太空 / R宗教杂谈

准确率0.5833，仅作流程检查

运行测试：预测完成，输出是否完整?

示列：程序已完成120条测试文本的预测，但检查导出的CSV文件时，发现其中只有列名，没有数据记录

```
run_id,sample_id,true_label,predicted_label

# predictions.csv：没有任何数据行
```

回答：应该先检查算法，还是输出写入环节?

文件存在不等于输出有效；检查应包括内容、数量及其与本次运行的对应关系

来源：教学故障注入：fault-empty-output

运行测试：运行失败的定位与复测

将数据缓存路径指向空目录，并禁止自动下载后，程序在加载数据时失败

```
FAILED stage=data-loading
OSError: 20Newsgroups dataset not found
```

步骤	本例检查与处理
定位失败阶段	根据报错位置确认失败发生在数据加载阶段，尚未进入训练
核对资源入口	检查程序实际使用的缓存路径，以及该位置是否存在预期数据
修正已确认的问题	将缓存路径恢复为已经核验的数据位置，保持其他实验设置不变
重新运行并检查	使用新的运行编号，确认数据加载成功，再继续检查预测与输出

修复是否有效，应由重新运行的证据确认，而不是以“修改了代码或配置”为准

多人合作：实验交接说明的必要信息

接手者的问题	说明中需要提供什么？
这个实验要运行什么？	目标实验、采用的条件，以及本次实现的范围
运行前需要准备什么？	代码与数据来源、环境要求、必要的获取或安装步骤
从哪里、怎样启动？	工作目录、执行入口、所用配置与完整命令
运行后检查什么？	输出位置、预期内容和检查标准
有哪些已知限制？	已验证的平台、尚未验证的条件，以及已知问题的处理线索

README文件

交接说明应减少接手者的猜测，而不只是记录自己做过什么

实验交接：模糊描述与明确描述

模糊描述	更明确的描述示例
“装好环境后运行xxx脚本。”	“在项目根目录创建独立环境，按依赖文件安装软件，并使用该环境中的Python执行下面列出的完整命令。”
“数据路径自己改一下。”	“将配置中的数据路径设为已核验的缓存目录； 启动后确认程序实际读取的位置，以及加载的类别和样本数。”
“没有报错就可以。”	“本次小规模测试应生成120条预测；检查文件可读、样本标识无遗漏或重复，并确认指标与预测记录一致。”

NOTE: 以上是写法示例。
实际交接时，应填入对应：文件名、路径、命令和检查标准

不在场解释时，另一位同学能否找到运行入口，并判断输出是否符合预期?

作业3：实验准备与最小运行

基于作业2的复现方案，准备代码、数据和运行环境，完成一次小规模流程检查，并整理他人可以使用的运行说明

提交项	具体要求
资源与环境记录	说明代码、数据的来源与版本，记录实际环境； 标明已核验内容及未解决问题
运行说明与配置	提供准备步骤、工作目录、完整命令、实际采用的参数及必要的本地修改说明
一次运行的证据	保存对应的运行编号、日志和实际输出，并列出检查结果
简短总结	说明本次完成了什么、仍有什么限制，以及下一步如何开展完整实验

两种可能结果：
运行完成：提交有效输出，说明哪些检查通过、依据是什么。
运行受阻：提交失败位置、错误信息、已尝试的处理及下一步计划，不以一句“跑不通”代替记录。

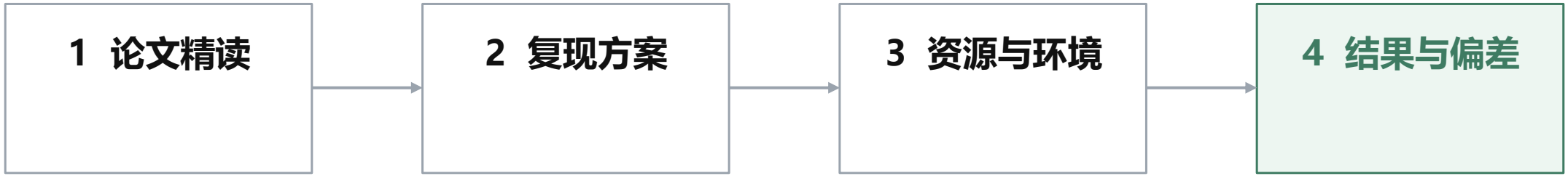
评价资源与目标实验的对应关系、记录的可追溯性及检查依据，不以准确率高低评分

Take-away：运行验证与结论验证

本节课建立的基础	后续仍需完成的判断
确认代码和数据的来源及对应关系	实际实验条件与目标条件是否一致
准备环境，明确配置与运行入口	条件差异是否影响结果的可比性
检查小规模运行及输出完整性	完整实验结果是否支持原有结论
保留运行记录与交接说明	结果偏差如何解释，结论适用于什么范围

本节课为结果分析准备可追踪的实验过程与输出证据

下一课：结果比较与偏差解释



追踪证据　　明确条件　　保存运行　　解释差异

结果一致需要证据，结果不同也需要解释