# 后端Spring Boot2

以下是常用的 Maven 命令及其用途：

1. mvn clean

- 清理项目，删除 target 目录及其内容。

2. mvn compile

- 编译项目源代码。

3. mvn test

- 运行项目的单元测试。

4. mvn package

- 打包项目，生成 JAR 或 WAR 文件。

5. mvn install

- 将项目打包并安装到本地 Maven 仓库。

6. mvn deploy

- 将项目打包并部署到远程 Maven 仓库。

7. mvn verify

- 运行所有检查，验证项目是否有效且符合质量标准。

8. mvn site

- 生成项目站点文档。

9. mvn dependency:tree

- 显示项目的依赖树。

10. mvn spring-boot:run

- 运行 Spring Boot 应用。

11. mvn help:effective-pom

- 显示项目的有效 POM（包括继承和插件的默认配置）。

12. mvn versions:set -DnewVersion=1.0.1

- 更新项目的版本号。

13. mvn archetype:generate

- 使用 Maven 原型生成新项目。

14. mvn clean install -DskipTests

- 跳过测试阶段进行清理和安装。

15. mvn clean package -Pprod

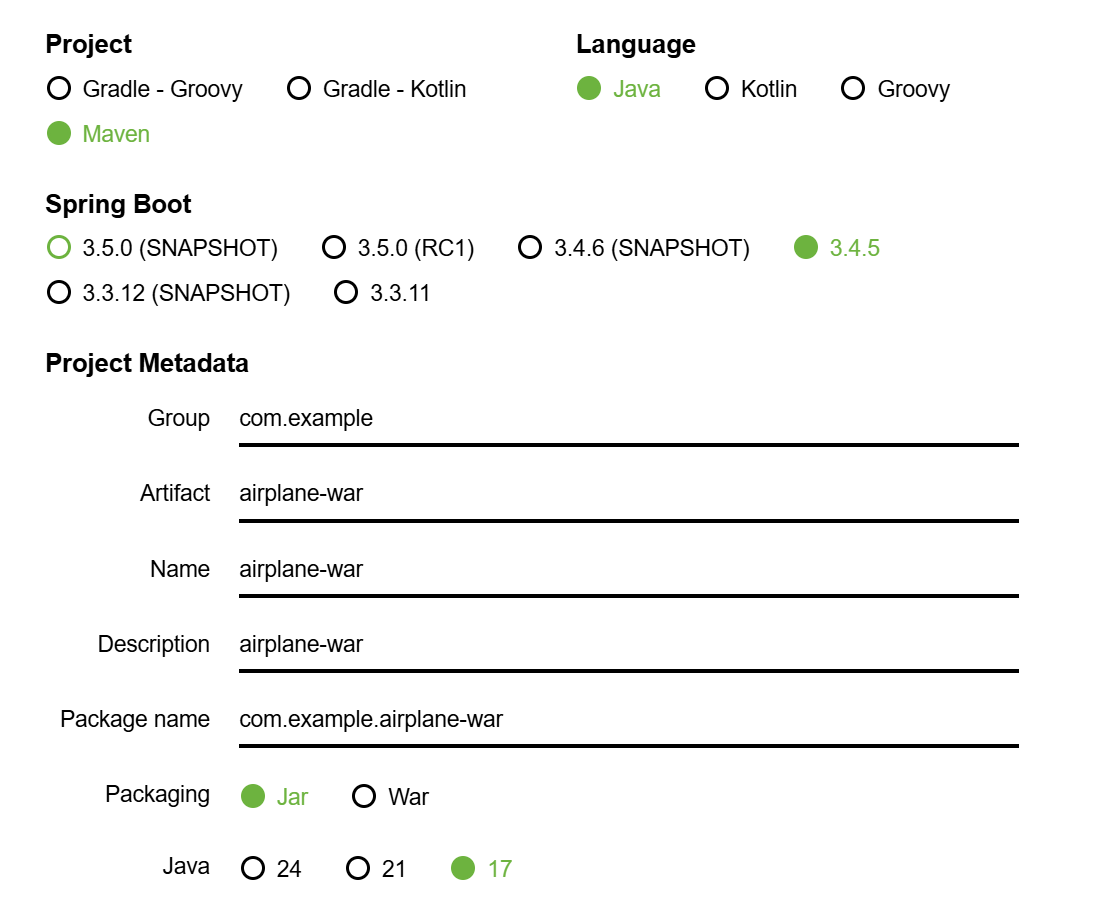
- 使用 prod 配置文件进行清理和打包。

要在执行 mvn package 时跳过测试，可以使用以下命令：

mvn package -DskipTests

如果你希望完全跳过测试代码的编译，可以使用：

mvn package -Dmaven.test.skip=true



在Windows系统下，运行Spring Boot项目的常用命令如下：

1. 直接运行项目 ：

mvn spring-boot:run

2. 打包后运行 ：

mvn clean package

java -jar target/airplanewar-0.0.1-SNAPSHOT.jar

3. 跳过测试运行 ：

mvn spring-boot:run -DskipTests

4.这些命令应该在项目根目录（包含pom.xml的目录）下执行。如果遇到问题，可以尝试清理并重新构建项目：

mvn clean install

Maven的生命周期（Lifecycle）是构建项目的核心概念，它定义了项目构建过程中的各个阶段。以下是Maven生命周期的主要环节及其功能：

1. clean生命周期 ：

- pre-clean ：执行清理前的准备工作

- clean ：删除target目录，即清理项目构建生成的文件

- post-clean ：执行清理后的收尾工作

2. default生命周期 （核心生命周期）：

- validate ：验证项目是否正确且所有必要信息是否可用

- compile ：编译项目的源代码

- test ：使用合适的单元测试框架测试编译后的代码

- package ：将编译后的代码打包成可分发的格式，如JAR、WAR等

- verify ：对集成测试的结果进行检查，确保质量达标

- install ：将打包的文件安装到本地仓库，供其他项目使用

- deploy ：将最终的包复制到远程仓库，供其他开发者和项目使用

3. site生命周期 ：

- pre-site ：执行生成站点文档前的准备工作

- site ：生成项目的站点文档

- post-site ：执行生成站点文档后的收尾工作

- site-deploy ：将生成的站点文档部署到指定的服务器

每个生命周期都包含一系列阶段（phase），这些阶段按顺序执行。当执行某个阶段时，Maven会自动执行该阶段之前的所有阶段。例如，执行 mvn package 时，Maven会依次执行 validate 、compile 、 test，最后执行 package 。

要使用Postman测试本项目的 /register 接口，请按照以下步骤操作：

1. 启动项目：

确保项目已成功启动，可以在终端运行以下命令：

2. 打开Postman并创建新请求：

- 请求方法：POST

- URL： http://localhost:8080/register

3. 设置请求头：

- 点击"Headers"标签

- 添加键值对：

4. 设置请求体：

- 点击"Body"标签

- 选择"raw"选项

- 选择"JSON"格式

- 输入以下JSON数据：

json

Open Folder

{

"username" : "testuser" ,

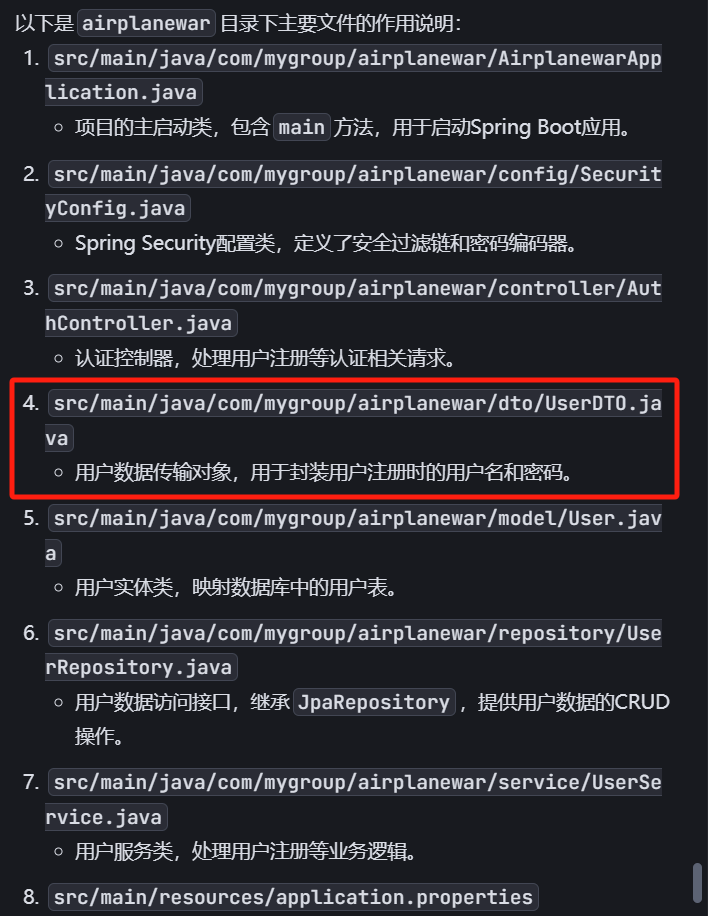
"password" : "testpassword"

}

5. 发送请求并查看响应：

- 点击"Send"按钮

- 在下方查看响应状态码和响应体



DTO的全称是 数据传输对象 （Data Transfer Object）。它是一种设计模式，主要用于在不同层或系统之间传输数据，通常用于封装和传递多个属性值。在您的项目中， UserDTO 类就是一个典型的DTO实现，用于封装用户注册时的用户名和密码信息。

curl -X POST -d "username=testuser" -d "password=testpass" http://localhost:8080/register

# 前端VUE

1.安装vue-router依赖：

npm install vue-router@4

2.创建路由配置文件

3.修改main.js文件以使用路由

4.创建组件：

5.安装项目依赖：

npm install

6.启动开发服务器：

npm run serve

以下是 npm 常用命令及其作用的整理：

1. npm init

npm init

初始化项目，创建 package.json 文件（交互式问答形式）

2. npm install

npm install < packag e > # 安装特定包（自动保存到 dependencies）

npm install < packag e > --save-dev # 安装开发依赖（保存到 devDependencies）

npm install -g < packag e > # 全局安装

npm install # 安装所有依赖（根据 package.json）

3. npm run

npm run < scrip t > # 执行 package.json 中scripts 配置的脚本

npm start # 快捷命令，等同于 npm run start

npm test # 快捷命令，等同于 npm run test

4. npm update

npm update < packag e > # 更新指定包

npm update # 更新所有包

5. npm uninstall

npm uninstall < packag e > # 移除本地包

npm uninstall -g < packag e > # 移除全局包

6. npm list

npm list # 查看当前项目安装的包

npm list -g --depth=0 # 查看全局安装的包（显示一级目录）

7. npm audit

npm audit # 检查依赖项中的安全漏洞

npm audit fix # 自动修复可修复的漏洞

8. npm config

npm config set registry https://registry.npmmirror.com # 设置镜像源

npm config list # 查看当前配置

9. npm outdated

npm outdated # 检查过时的依赖包

10. npm publish

npm publish # 发布包到 npm 仓库

# CentOS

## 常见操作命令

使用自带的vi编辑器（安装后执行）

sudo vi /etc/nginx/sites-available/default

vi编辑器基本操作：

- 按 i 进入编辑模式

- 按 ESC 退出编辑模式

- 输入 :wq 保存并退出

- 输入 :q! 不保存强制退出

实时搜索（推荐）

# 全盘搜索（需要sudo权限）

sudo find / -name "target\_file.txt" 2>/dev/null

# 当前目录递归搜索

find . -name "\*.log"

# 模糊搜索（忽略大小写）

find /var -iname "config.\*"

## 部署前端代码到远程虚拟机

1.首先在本地构建生产环境代码

npm run build //构建的代码放在dist文件下

2.上传构建文件到虚拟机（假设虚拟机IP为192.168.1.100）

scp -r ./dist/\* username@192.168.1.100:/var/www/html/

3.在虚拟机上安装nginx作为web服务器

sudo apt update //或者yum，取决安装了哪个包管理工具

sudo apt install nginx -y

sudo systemctl start nginx

4. 配置nginx指向上传的代码：

sudo vi /etc/nginx/nginx.conf

修改root路径为：

root /var/www/html;

5. 重启nginx服务：

sudo systemctl restart nginx

注意事项：

1. 确保虚拟机开放80端口

2. 如果使用其他web服务器，可全局安装serve工具：

如果报错：[root@localhost frontend]# sudo yum install nginx -y

Loaded plugins: fastestmirror

Loading mirror speeds from cached hostfile

No package nginx available.

Error: Nothing to do

这个错误是因为CentOS默认仓库中没有nginx包

首先安装EPEL仓库：

sudo yum install epel-release -y

再次尝试安装nginx：

sudo yum install nginx -y

如果仍然报错，可以手动添加Nginx官方仓库：

sudo yum install yum-utils -y

sudo yum-config-manager --add-repo https://nginx.org/packages/centos/$(rpm -E %{rhel}).0/nginx.repo

sudo yum update

sudo yum install nginx -y

安装完成后验证版本：

nginx -v

## 运行后端代码

nohup java -jar airplanewar-0.0.1-SNAPSHOT.jar > app.log 2>&1 &

1.查看所有系统服务状态

systemctl list-units --type=service --state=running

2. 查看具体服务进程

ps aux | grep [ 服务名 ]

# 示例（查看MySQL进程）：

ps aux | grep mysqld

## Windows系统安装OpenSSH客户端

1. 通过Windows功能启用OpenSSH客户端

# 以管理员身份运行PowerShell

Add-WindowsCapability - Online - Name OpenSSH.Client~~~~ 0.0 . 1.0

2. 验证安装状态

Get-WindowsCapability - Online | Where-ObjectName -like 'OpenSSH.Client\*'

成功安装会显示：

Name  : OpenSSH.Client~~~~0.0.1.0

State : Installed

3. 启动SSH服务

Start-Service ssh - agent

Set-Service - Name ssh - agent – StartupType Automatic

4. 测试连接功能

ssh - V

成功会显示：

OpenSSH\_for\_Windows\_8.6p1, LibreSSL 3.4.3

安装完成后，即可在CMD/PowerShell中使用 scp 命令传输文件到Linux虚拟机：

scp .\target /\* .jar username @虚拟机IP : / home /user /

## CentOS系统中解压zip文件

1.首先安装unzip工具（如未安装）：

sudo yum install unzip -y

2.解压zip文件到当前目录：

unzip 文件名.zip

3.解压到指定目录：

unzip 文件名.zip -d /目标路径

常用参数说明：

-q：静默模式（不显示解压过程）

-o：覆盖已存在文件

-P 密码：解压加密文件（需替换"密码"为实际密码）

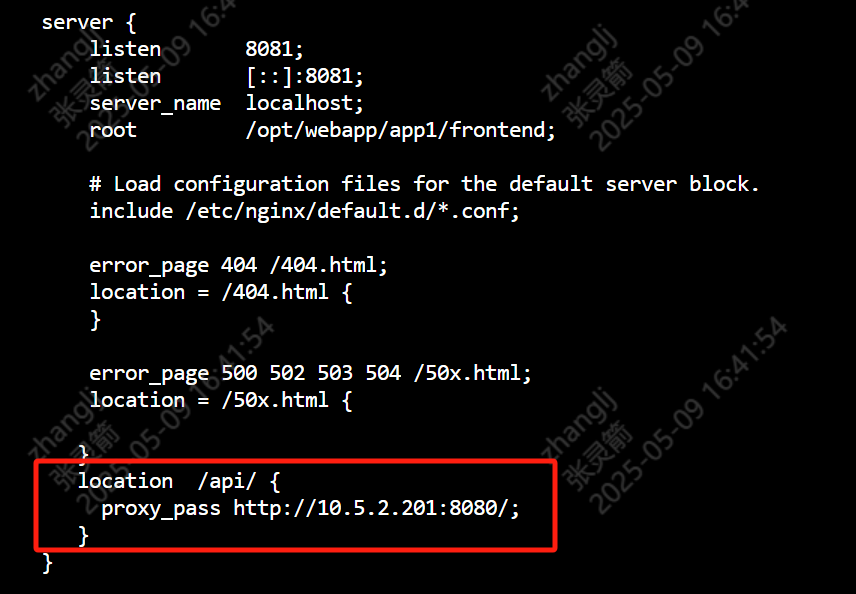
示例：静默解压加密文件到/home目录

unzip -q -P 123456 example.zip -d /home

## 配置Nginx路径转发内容

sudo yum install -y nginx

sudo vi /etc/nginx/nginx.conf



## Centos主要目录描述

核心系统目录

/bin → 存放基础命令（ls、cp等）

/boot → 内核与启动文件

/dev → 设备文件（磁盘、终端等）

/etc → 系统配置文件

/lib → 32位库文件

/lib64 → 64位库文件

/proc → 进程和内核信息（虚拟文件系统）

/sbin → 系统管理命令

/sys → 内核设备树（虚拟文件系统）

/usr → 用户程序资源

/var → 可变数据（日志、数据库等）

用户相关目录

/home → 用户家目录

/root → 管理员家目录

/tmp → 临时文件（自动清理）

特殊用途目录

/mnt → 临时挂载点

/media → 可移动设备挂载点

/opt → 第三方软件安装目录

/srv → 服务相关数据

## 创建虚拟环境

# 创建含特定Python版本的环境

conda create -n ml-env python=3.8

# 安装包含C扩展的包

conda install numpy=1.21 -c intel

# 导出环境配置

conda env export > environment.yml

# 激活base环境

conda activate base

# 退出当前环境返回base环境

conda deactivate

# 完全退出conda环境栈（当存在嵌套激活时）

conda deactivate --all

# 查看当前环境前缀（退出后应显示base或无前缀）

conda env list | grep '\*'

# 检查Python解释器路径（应指向系统或base环境）

which python  # Linux/macOS

# 列出所有conda环境

conda env list

# MySQL

## 创建数据库和表

CREATE DATABASE `airplane\_war` CHARACTER SET utf8mb4 COLLATE utf8mb4\_unicode\_ci;

SHOW DATABASES LIKE 'airplane\_war';

USE airplane\_war;

CREATE TABLE `user` (

`id` bigint NOT NULL AUTO\_INCREMENT,

`username` varchar(255) DEFAULT NULL,

`password` varchar(255) DEFAULT NULL,

PRIMARY KEY (`id`),

UNIQUE KEY `username` (`username`)

) ENGINE=InnoDB AUTO\_INCREMENT=26 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_0900\_ai\_ci;

DESCRIBE user;

# Trae

请基于本项目框架，新增商品信息管理模块，管理的商品信息主要名称、价格、京东商品链接URL等字段，用户登录后可以新增、删除和修改商品信息，请帮忙生成spring boot2纯后端代码，包括controller、dto、model、repository和service五个类代码

# 算法

## 分类、回归、聚类和降维算法

**分类**主要目标预测离散标签（如垃圾邮件/图像类别），评估指标准确率、精确率、召回率，算法SVM、贝叶斯、决策树、逻辑回归、随机森林，场景用于图像识别、垃圾邮件检测；

**回归**主要目标预测连续数值（如价格/温度/评分），评估指标均方误差(MSE)、R²分数，算法线性回归、多项式回归、岭回归、SVR等，场景用于房价预测、销量预测等；

**聚类**主要目标发现数据内在分组结构（无监督学习），输出离散的簇标签（如0/1/2），应用于客户分群、异常检测，评估方法轮廓系数、Calinski-Harabasz指数，常用K-Means、密度聚类（DBSCAN）、层次聚类（Hierarchical Clustering）、距离和相似度度量方法；K-Means：适合规则分布的大规模数据，快速聚类。

层次聚类：适合小规模数据和需要层次结构的场景，如基因分析或市场细分。

DBSCAN：适合处理复杂形状簇和含有噪声的数据，如地理空间数据或异常检测。

距离和相似度度量：用于选择合适的距离度量方式，根据数据特征进行优化选择。

**降维**主要目标减少数据维度同时保留关键信息，输出连续的低维特征矩阵（如二维坐标），应用于数据可视化、特征工程，评估方法保留方差比例、重构误差，常用PCA、t-SNE、LDA（示例使用PCA）

# Python

## Python常用核心库的分类介绍

一、数据处理与科学计算

import pandas as pd  # 表格数据处理（CSV/Excel）

import numpy as np   # 数值计算（多维数组）

import xlrd         # Excel文件读取

import openpyxl     # 现代Excel文件处理

二、数据可视化

import matplotlib.pyplot as plt  # 基础绘图库

import seaborn as sns            # 统计图表

import plotly.express as px      # 交互式可视化

三、Web开发

from flask import Flask         # 轻量级Web框架

import django                   # 全功能Web框架

import requests                 # HTTP请求库

from bs4 import BeautifulSoup   # 网页解析

四、机器学习/人工智能

import sklearn          # 机器学习算法库

import tensorflow as tf # 深度学习框架

import torch            # 深度学习框架

import nltk             # 自然语言处理

五、系统与文件操作

import os               # 操作系统接口

import shutil           # 文件高级操作

import pathlib          # 面向对象路径操作

六、图形界面开发

import tkinter as tk    # 标准GUI工具包

from PyQt5 import QtWidgets  # 跨平台GUI框架

安装方法（Windows）：

pip install pandas numpy matplotlib seaborn flask requests

特殊用途库：

- 异步编程：asyncio

- 图像处理：Pillow

- 数据库操作：sqlalchemy

- 时间处理：arrow

- 命令行工具：click

## Python常用库功能介绍及官网地址

1. NumPy

• 功能：提供高性能的多维数组对象和相关操作，是科学计算的基础库，支持大量的数学运算和线性代数操作。

• 官网地址：https://numpy.org/（https://numpy.org/）

2. Pandas

• 功能：用于数据处理和分析，提供高效的数据结构（如 DataFrame 和 Series）和数据分析工具，方便进行数据清洗、转换、统计分析和可视化。

• 官网地址：https://pandas.pydata.org/（https://pandas.pydata.org/）

3. Matplotlib

• 功能：用于创建各种类型的可视化图表，如折线图、柱状图、散点图等，支持多种格式输出。

• 官网地址：https://matplotlib.org/（https://matplotlib.org/）

4. Seaborn

• 功能：基于 Matplotlib 的高级可视化库，提供更美观的统计图表，简化复杂可视化的创建过程。

• 官网地址：https://seaborn.pydata.org/（https://seaborn.pydata.org/）

5. TensorFlow

• 功能：由谷歌（https://baike.baidu.com/item/Google/86964）开发的深度学习框架，支持构建和训练神经网络，广泛应用于机器学习和深度学习任务。

• 官网地址：https://www.tensorflow.org/（https://www.tensorflow.org/）

6. PyTorch

• 功能：由 Facebook 开发的深度学习框架，强调动态计算图，提供灵活性和速度，适合研究和开发。

• 官网地址：https://pytorch.org/（https://pytorch.org/）

7. Requests

• 功能：用于发送 HTTP 请求的库，简化网络请求过程，支持多种请求方法。

• 官网地址：https://docs.python-requests.org/（https://docs.python-requests.org/）

8. Beautiful Soup

• 功能：用于解析和处理 HTML 和 XML 文档，方便网页数据抽取。

• 官网地址：https://www.crummy.com/software/BeautifulSoup/（https://www.crummy.com/software/BeautifulSoup/）

9. SQLAlchemy

• 功能：提供数据库操作接口和 ORM（对象关系映射）功能，支持多种数据库。

• 官网地址：https://www.sqlalchemy.org/（https://www.sqlalchemy.org/）

10. OpenCV

• 功能：计算机视觉库，提供图像处理和计算机视觉功能，广泛应用于图像和视频处理。

• 官网地址：https://opencv.org/（https://opencv.org/）

11. NLTK

• 功能：自然语言处理库，提供文本处理工具，支持多种自然语言处理任务。

• 官网地址：https://www.nltk.org/（https://www.nltk.org/）

12. SpaCy

• 功能：高性能自然语言处理库，支持快速高效的文本分析。

• 官网地址：https://spacy.io/（https://spacy.io/）

13. Flask

• 功能：轻量级 Web 框架，易于扩展，适合小型项目和原型开发。

• 官网地址：https://flask.palletsprojects.com/（https://flask.palletsprojects.com/）

14. Django

• 功能：全功能 Web 框架，提供 ORM、模板引擎等，适合企业级 Web 应用开发。

• 官网地址：https://www.djangoproject.com/（https://www.djangoproject.com/）

15. FastAPI

• 功能：现代、快速的 Web 框架，强调速度和开发人员体验，支持异步编程。

• 官网地址：https://fastapi.tiangolo.com/（https://fastapi.tiangolo.com/）

## 基础数据结构

# 1.列表（有序可变）

numbers = [1, 2, 3]

# 2.元组（有序不可变）

coordinates = (40.7128, 74.0060)

# 3.集合（无序唯一）

unique\_nums = {1, 2, 3}

# 4.字典（键值对）

user = {"name": "Alice", "age": 30}

## scikit-learn 介绍

**统一的API设计**

所有算法遵循一致的接口规范：

fit(): 训练模型

predict(): 预测新数据

transform(): 数据预处理（如特征缩放）

**数据预处理工具**

特征工程：标准化（StandardScaler）、归一化（MinMaxScaler）、缺失值填充（SimpleImputer）、特征选择（SelectKBest）。

数据分割：train\_test\_split划分训练集与测试集。

编码转换：独热编码（OneHotEncoder）、标签编码（LabelEncoder）。

**模型评估与可视化**

提供分类（准确率、F1分数、ROC曲线）、回归（均方误差、R²分数）等评估指标。

支持混淆矩阵、学习曲线等可视化工具，辅助分析模型性能。

| 模块 | 功能概述 |
| --- | --- |
| **数据集** | 内置鸢尾花（Iris）、波士顿房价（Boston）、手写数字（Digits）等经典数据集。 |
| **预处理** | 特征缩放、缺失值处理、类别编码、数据标准化。 |
| **模型训练** | 分类（SVM、随机森林）、回归（线性回归、Lasso）、聚类（K-means、DBSCAN）。 |
| **降维与特征提取** | PCA、t-SNE、核主成分分析（KernelPCA）。 |
| **模型选择** | 交叉验证（cross\_val\_score）、网格搜索（GridSearchCV）、超参数调优。 |
| **管道（Pipeline）** | 串联预处理、模型训练步骤，简化复杂流程（如Pipeline([('scaler', StandardScaler()), ('clf', RandomForestClassifier())])）。 |

获取当前环境安装的模块列表

pip list --format=freeze

# Git

## 上传和拉取代码

1.初始化本地仓库

# 在项目根目录执行

git init

git add .

git commit -m "初始化项目"

2.配置远程仓库关联

git remote add origin https://github.com/williamscheung/leakproof.git

3.强制推送代码（首次）

git push -f origin main

4.若远程仓库已有内容，建议先拉取合并：

git pull origin main --allow-unrelated-histories

1）添加变更文件（排除大文件）

git add api\_service.py model\_training.py templates/\*

2）提交变更说明

git commit -m "优化分类器性能｜更新模板样式｜v1.2.0"

3）推送至远程main分支

git push origin main

Windows虚拟机配置

1.安装必要组件

# 以管理员身份运行

Set-ExecutionPolicy RemoteSigned -Scope Process

Install-PackageProvider NuGet -Force

Install-Module -Name PowerShellGet -Force

Install-Module -Name posh-git -Force

2.克隆仓库

cd D:\Projects

git clone git@github.com:williamscheung/leakproof.git