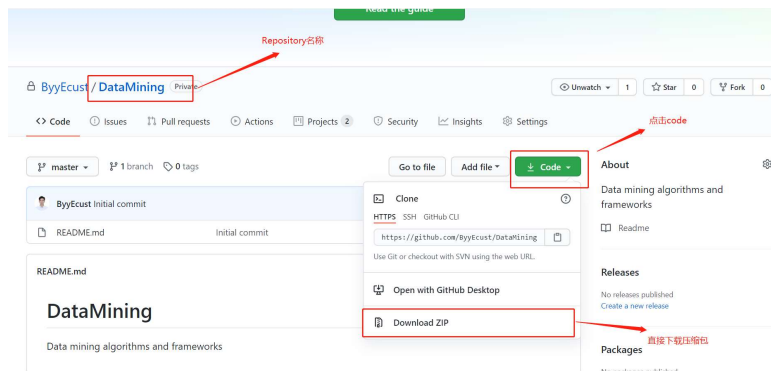
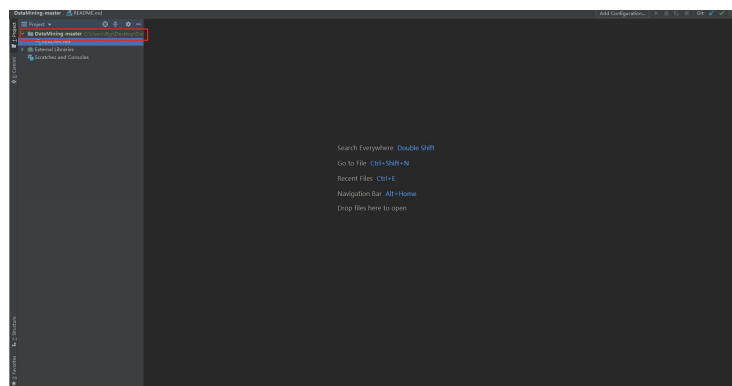


1. 在 github 中找到需要开发的 repository

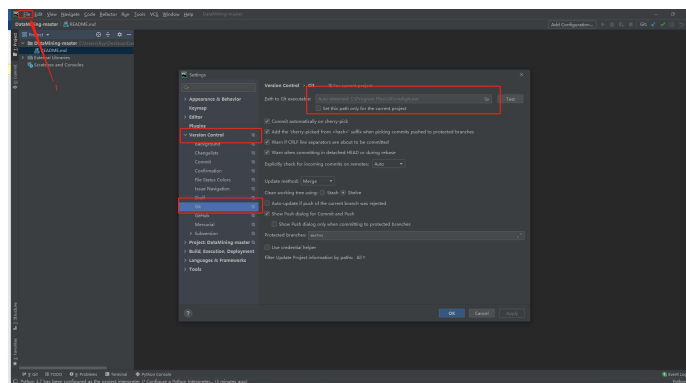


2. 解压文件之后用 pycharm 打开该文件



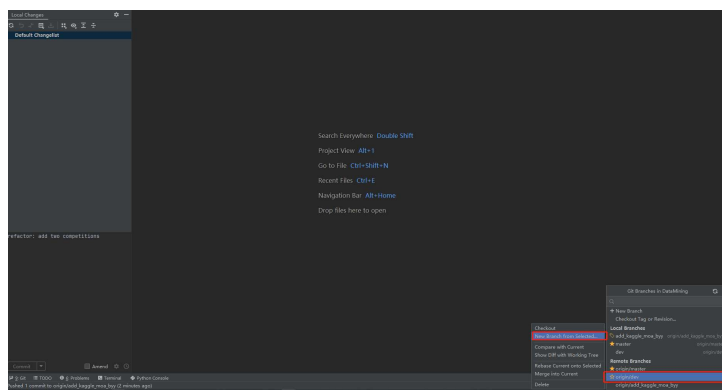
3. 配置 git 工具和绑定 github 的账号

- File → setting → version control → git → 选择本地安装的 git.exe

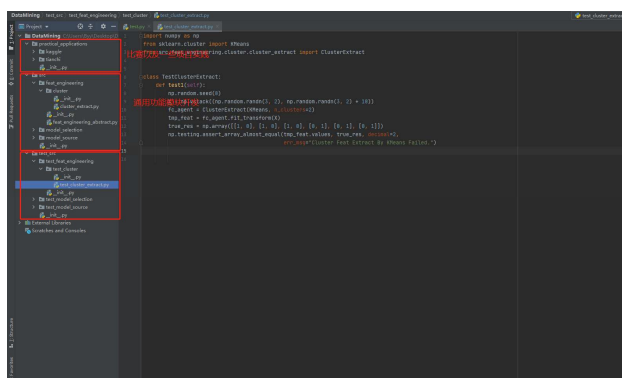


- File → setting → version control → github

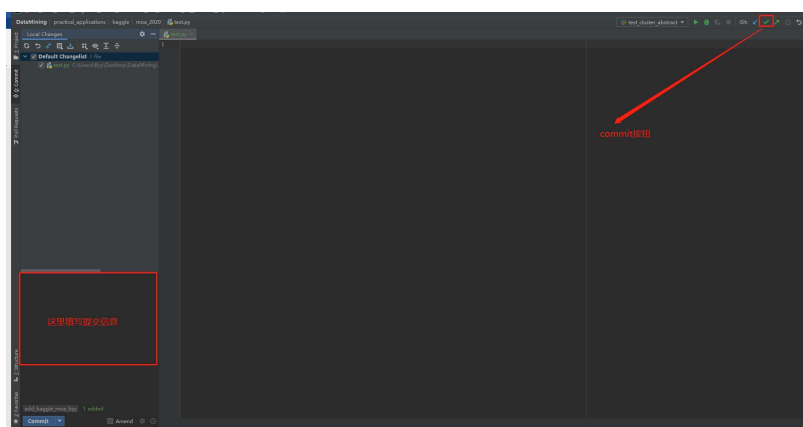
缩写为标准，如 `add_kaggle_moa_byy`，新建分支后默认会切换（checkout）到新建分支。



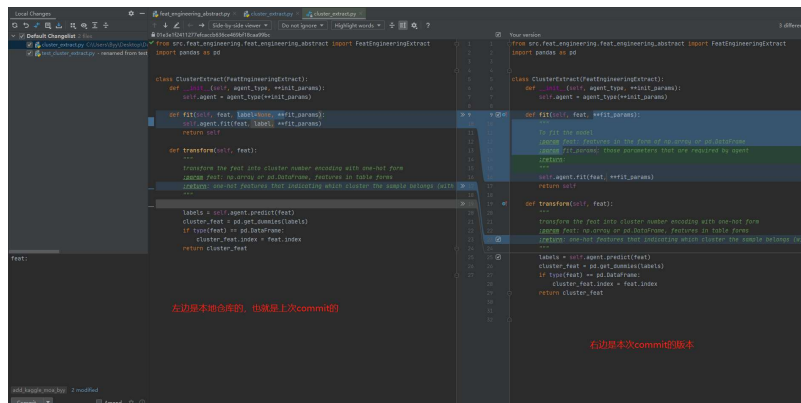
8. 目前开发目录包括两部分，一个是 `practical applications`，用于存放比赛的具体实现，一个是 `src` 目录，用来存放可复用的功能块。前者如果用到数据，可传至网盘，在 `readme` 中加以说明；后者的开发要保证可靠：1、代码规范，写明注释，出入参形式；2、在 `test_src` 目录相应位置尽量添加测试文件，测试文件格式如 `test_src/test_feat_engineering/test_cluster/test_cluster_extract.py` 所示（类和方法都以 `test` 开头）



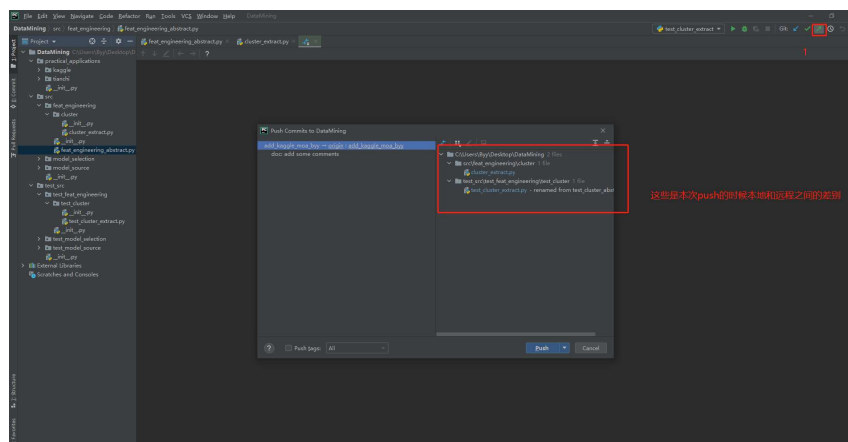
9. 开发完成后，先 `commit` 到本地仓库，再 `commit message` 框中简要填写本次开发的功能或修复的 `bug`（不超过 30 字），标准为“类型+内容”，类型包括 `<feat, test, refactor, fic, doc>`，如“`feat: create the cluster feat extraction class`”



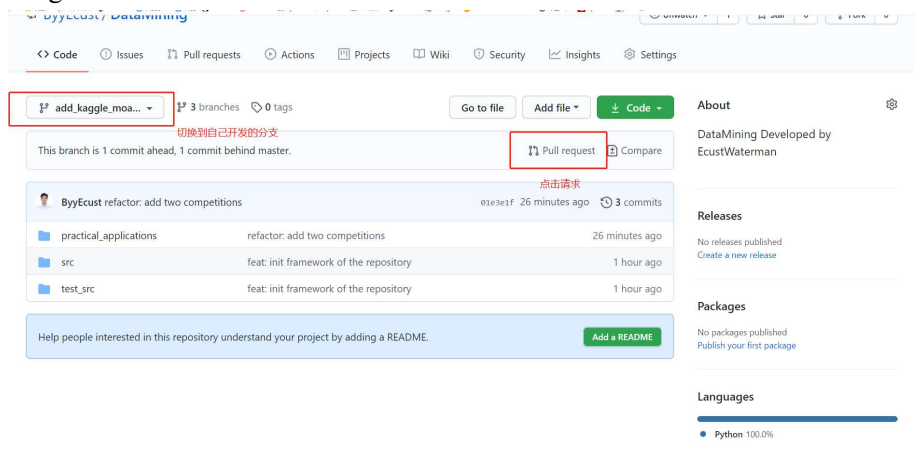
10. 每次 `commit` 的时候可以查看本次提交有那些改动，确保每次提交的都是你想要提交的



- 通过 push 按钮将本次仓库的代码提交到远程仓库，每次到 remote 之前，请先从远程 pull 下来后再 commit、push




- 如果本次拉取的分支功能已经开发完全，那么进到 github 页面，切到自己的分支，点急 pull request，选择需要合并的分支（一般为 dev）分支，点击 create pull request，在右侧的 assignees 中选择评审人 dev 至少两人评审通过才能合并，master 设置的是 3 个，评审人通过后，会 merge 到 dev 分支。



Open a pull request

Create a new pull request by comparing changes across two branches. If you need to, you can also [compare across forks](#).



The screenshot displays the GitHub pull request interface. At the top, a header bar shows the branch 'base dev' and a green checkmark indicating the pull request is mergeable. Below this, the pull request title 'Add kaggle moa by #2' is visible, along with a description 'doc: add some comments on files'. A green button labeled 'View pull request' is on the right. The main content area shows the pull request details, including a list of assignees. The 'Assignees' section is highlighted with a red box, showing a list of users including 'waterman-zrq' who is assigned.