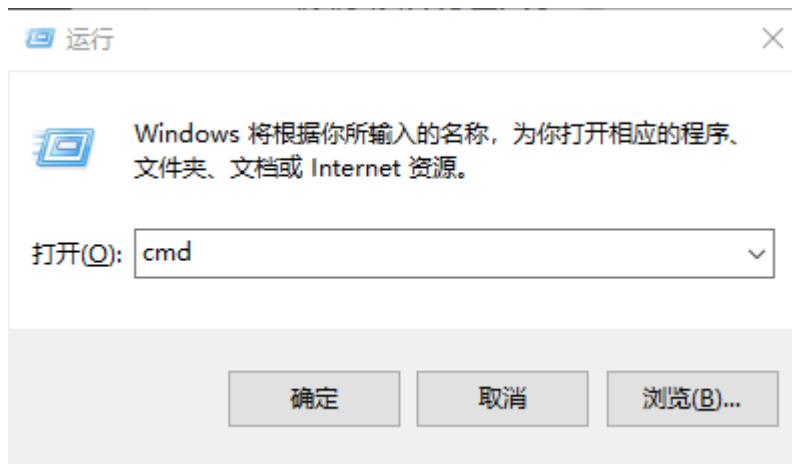


一 创建虚拟环境

1 Windows+R 输入 cmd 打开控制台



2 控制台输出 conda create --name 环境名称 python==3.8 创建环境 当弹出如下提示时, 输入 y 表示确定安装

```
C:\>conda create --name myTest python==3.8
Collecting package metadata (current_repodata.json): done
Solving environment: failed with repodata from current_repodata.json, will retry with next repodata source.
Collecting package metadata (repodata.json): done
Solving environment: done

## Package Plan ##

  environment location: D:\Anaconda\envs\myTest

added / updated specs:
- python==3.8

The following NEW packages will be INSTALLED:

ca-certificates    anaconda/pkgs/main/win-64::ca-certificates-2021.10.26-haa95532_2
certifi            anaconda/pkgs/main/win-64::certifi-2021.10.8-py38haa95532_0
openssl            anaconda/pkgs/main/win-64::openssl-1.1.1l-h2bbff1b_0
pip                anaconda/pkgs/main/win-64::pip-21.2.2-py38haa95532_0
python             anaconda/pkgs/main/win-64::python-3.8.0-hff0d562_2
setuptools         anaconda/pkgs/main/win-64::setuptools-58.0.4-py38haa95532_0
sqlite             anaconda/pkgs/main/win-64::sqlite-3.36.0-h2bbff1b_0
vc                 anaconda/pkgs/main/win-64::vc-14.2-h21ff451_1
vs2015_runtime     anaconda/pkgs/main/win-64::vs2015_runtime-14.27.29016-h5e58377_2
wheel              anaconda/pkgs/main/noarch::wheel-0.37.0-pyhd3eb1b0_1
wincertstore       anaconda/pkgs/main/win-64::wincertstore-0.2-py38haa95532_2

Proceed ([y]/n)? y
```

安装成功是显示如下界面

```
done
#
# To activate this environment, use
#
#     $ conda activate myTest
#
# To deactivate an active environment, use
#
#     $ conda deactivate
```

二 在虚拟环境中安装相关 Python 库

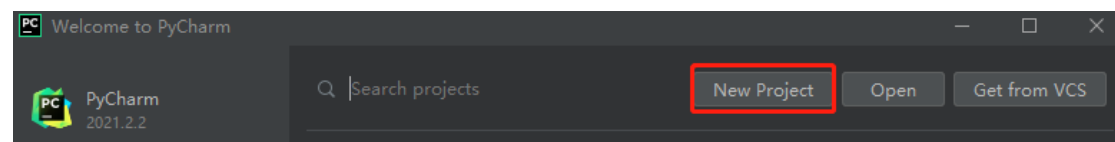
1 在控制台输入 activate 环境名称 进入虚拟环境

```
C:\Users\>activate myTest  
(myTest) C:\Users\>
```

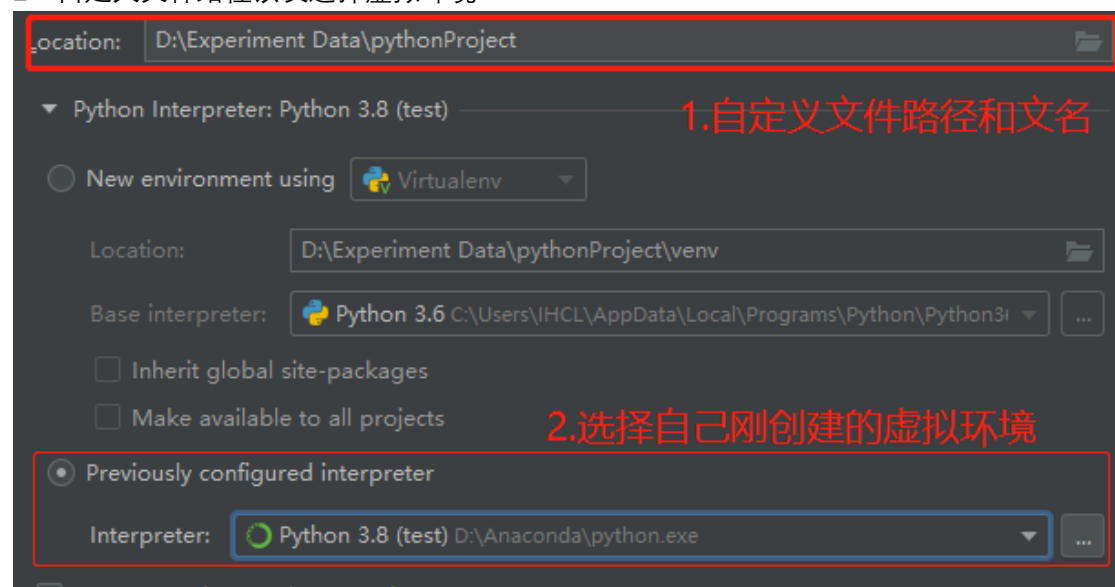
- 2 安装多标签学习库 `pip install scikit-multilearn`
安装 numpy 库 `pip install numpy==1.19.0`
安装 scipy 库 `pip install scipy==1.6.2`
安装 sklearn 库 `conda install scikit-learn==0.24.2`
安装 pandas 库 `conda install pandas==1.2.4`

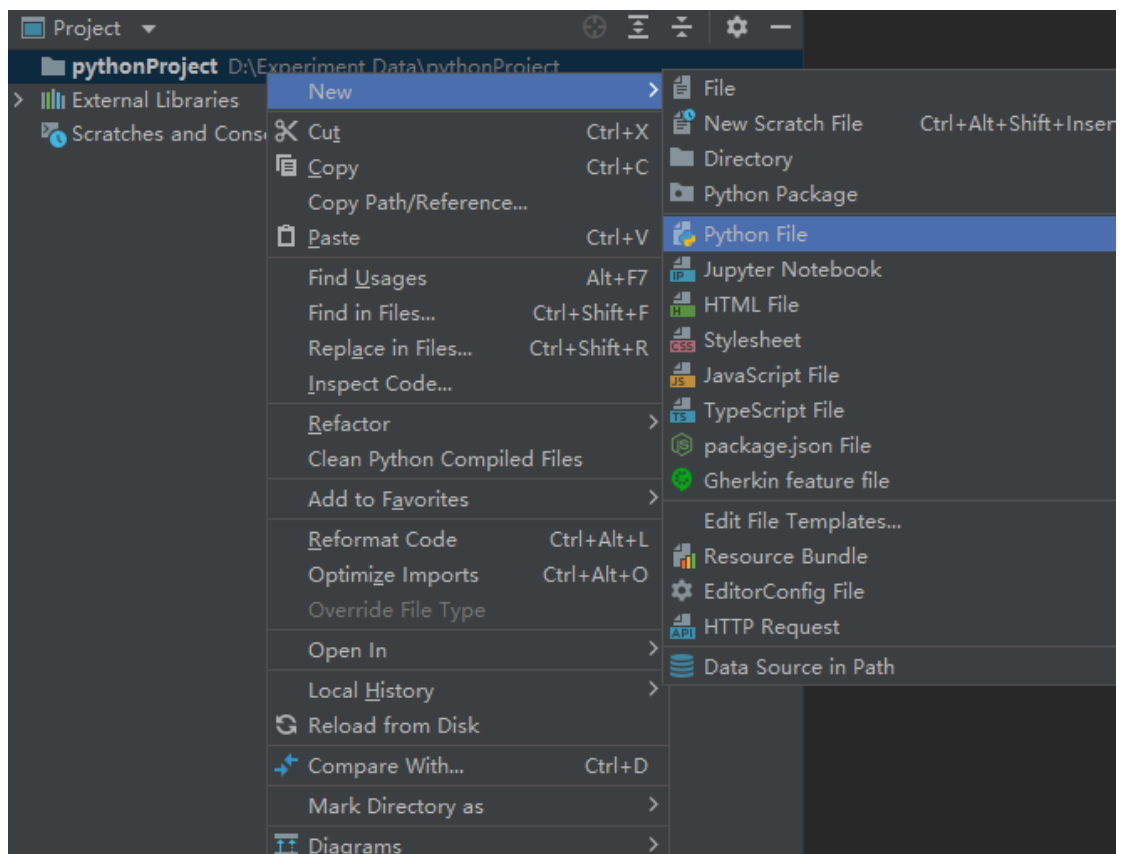
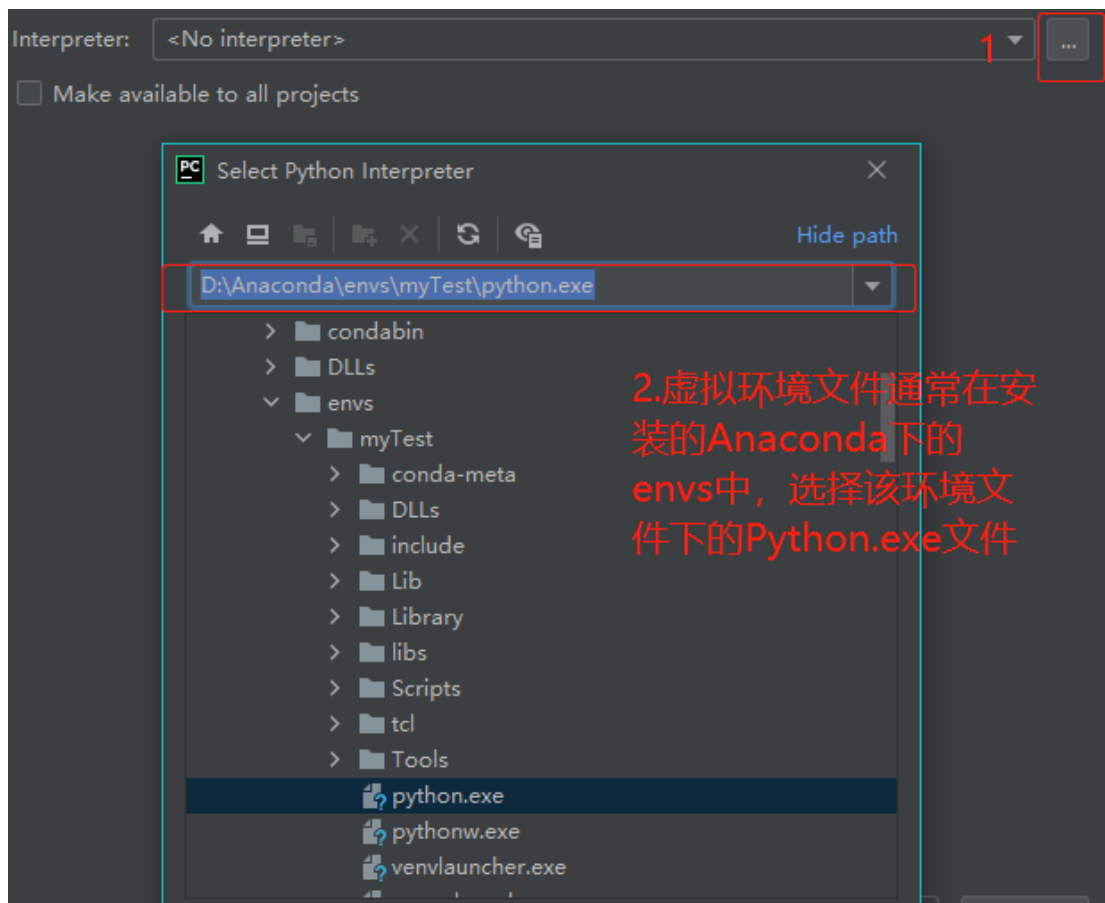
三 实验

- 1 打开 pycharm, New Project



- 2 自定义文件路径以及选择虚拟环境





3 数据是一份益生菌的数据集，数据集可以从这里 [multilabel datasets \(sourceforge.net\)](https://sourceforge.net/projects/multilabel-datasets/) 下载。数据样式如下图所示：共有 117 列，其中前 103 列是数据的 feature，后 14 列是数据的 label（都是 0 或者 1）。

	Att1	Att2	Att3	Att4	...	Class11	Class12	Class13	Class14
0	0.093700	0.139771	0.062774	0.007698	...	b'0'	b'0'	b'0'	b'0'
1	-0.022711	-0.050504	-0.035691	-0.065434	...	b'0'	b'1'	b'1'	b'0'
2	-0.090407	0.021198	0.208712	0.102752	...	b'0'	b'1'	b'1'	b'0'
3	-0.085235	0.009540	-0.013228	0.094063	...	b'0'	b'1'	b'1'	b'1'
4	-0.088765	-0.026743	0.002075	-0.043819	...	b'0'	b'0'	b'0'	b'0'

加载数据集

```
import pandas
from scipy.io import arff

data, meta = scipy.io.arff.loadarff('yeast/yeast-train.arff') # 加载数据集
data = pandas.DataFrame(data) # 将数据集转换成 dataframe，方便处理
print(data.head()) # 查看数据集的前五行
```

对数据做处理

```
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler

X = data.iloc[:, 0:103].values # 取出训练数据
y = data.iloc[:, 103:117].values # 取出标签
# 数据归一化
std = StandardScaler()
X = std.fit_transform(X)
# 划分训练集和测试集 比例为 4:1
trainx, testx, trainy, testy = train_test_split(X, y, test_size=0.2, random_state=0)
trainy = trainy.astype(np.float64)
testy = testy.astype(np.float64)
```

加载模型，训练，预测，查看结果

```
from skmultilearn.adapt import BRkNNaClassifier
import sklearn.metrics as metrics

classifier = BRkNNaClassifier()
# train
classifier.fit(trainx, trainy)
# predict
predictions = classifier.predict(testx)
predictions = predictions.toarray()
# 以下值越小越好
print(metrics.hamming_loss(testy, predictions))
print(metrics.zero_one_loss(testy, predictions))
print(metrics.coverage_error(testy, predictions))
```

```
print(metrics.label_ranking_loss(testy, predictions))
# 该值越大越好
print(metrics.average_precision_score(testy, predictions))
```

实验结果：

```
0.20214285714285715
0.8233333333333334
11.676666666666666
0.4791242091242091
0.374719574691436
```

可以自行尝试通过调节分类器参数，提高各项指标。

任务 1：使用下载的数据集，yeast-traint.arff 全部数据做训练集，使用 yeast-test.arff 全部数据做测试集，查看分类结果

任务 2：尝试使用 [scikit-multilearn: Multi-Label Classification in Python — Multi-Label Classification for Python](#) 下的不同多标签分类算法进行分类，对比不同算法性能

任务 3：尝试 dataset.csv 数据集，运行出结果。其中标签为 imageDiseaseCode

```
commonTzPhz  commonTzQxz  commonTzYxz  commonTzYixz
commonTzTsz  commonTzSrz  commonTzXyz  commonTzQyz
commonTzTbz, 思考如何标签和数据提取整合，然后输出结果
```