

Windows 部署Python节点

本章简要介绍如何在Windows环境上单独部署Python节点。

Python节点主要用于机器学习的DBSACN算法和自定义模块的Python脚本扩展。如果没有用到这两个功能模块，可以不用部署Python节点。



Windows 部署python节点目前支持Smartbi V10版本。

一、安装前准备

1、先确认本机有没有安装Anaconda其他版本或Python的运行环境，如果有则要先将其卸载。

```
cmdpython
python --version
```

```
C:\Users\Administrator>python --version
'python' 不是内部或外部命令，也不是可运行的程序
或批处理文件。
```

表示未安装python,若有版本信息则表示已安装

2、安装介质：（向Smartbi官方获取或自行去Anaconda3官网获取对应的版本）

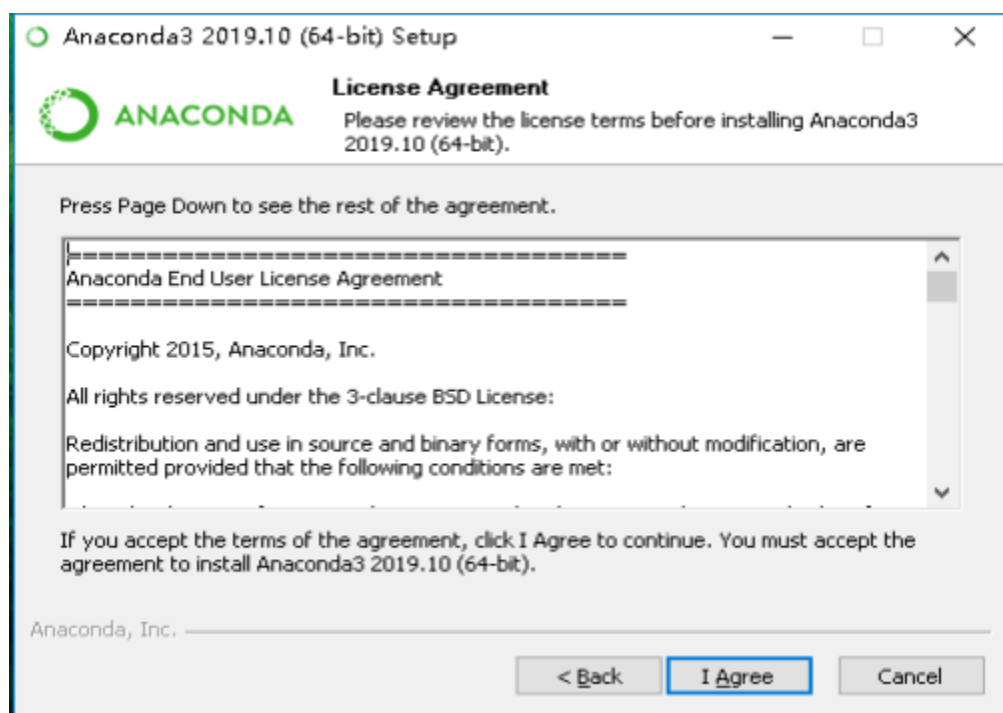
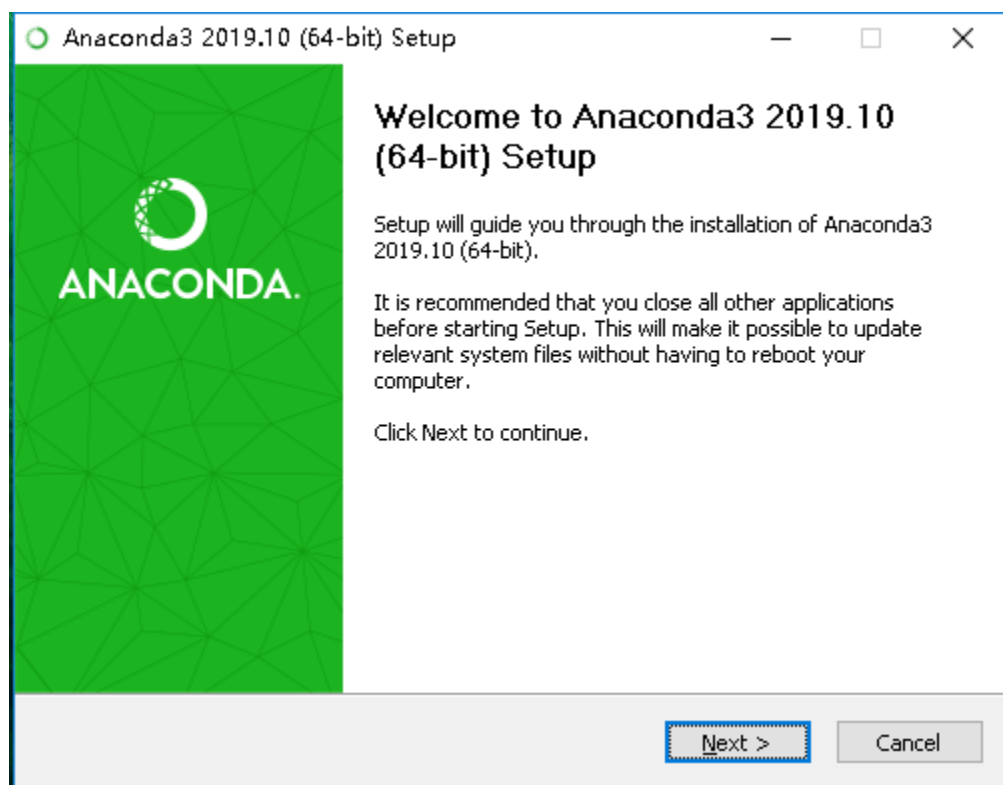
Anaconda3-2019.10-Windows-x86_64.exe

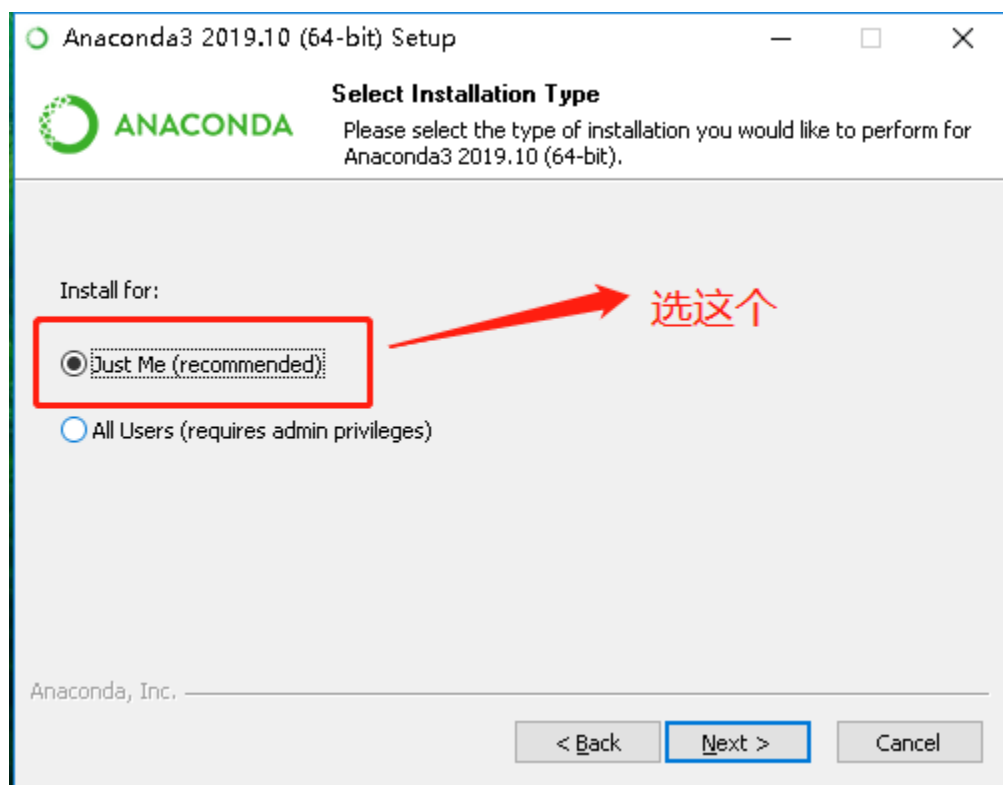
plugin_win.zip

二、安装python计算节点

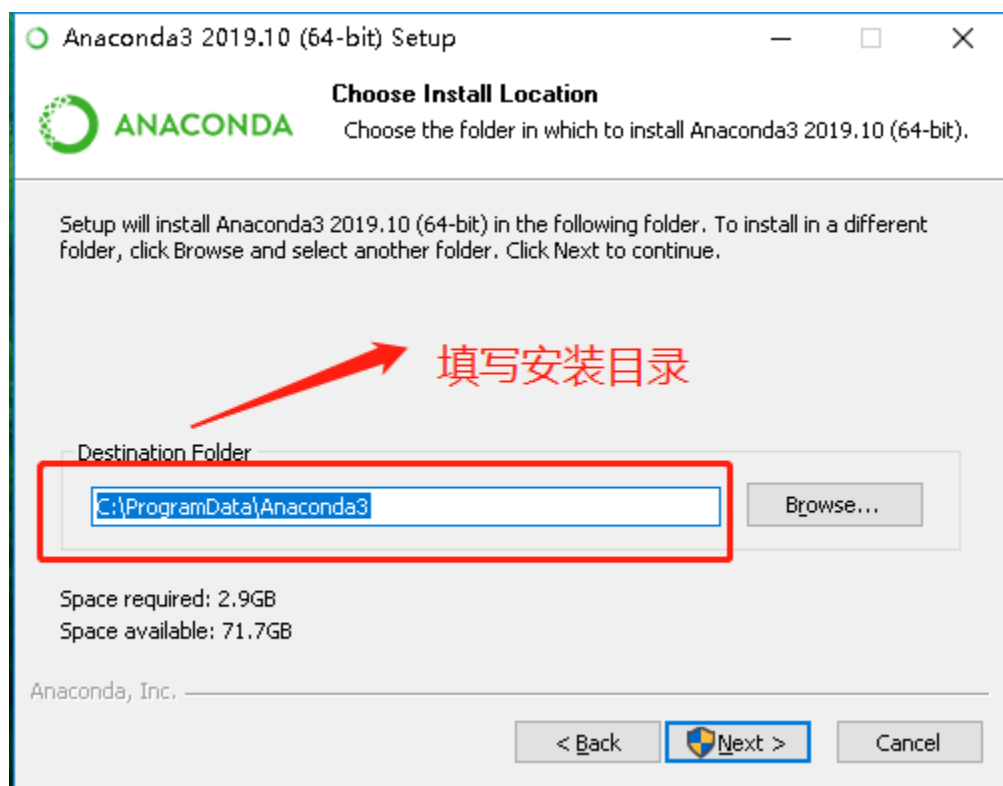
1、进入安装界面

点击 Anaconda3-2019.10-Windows-x86_64.exe 安装包开始安装

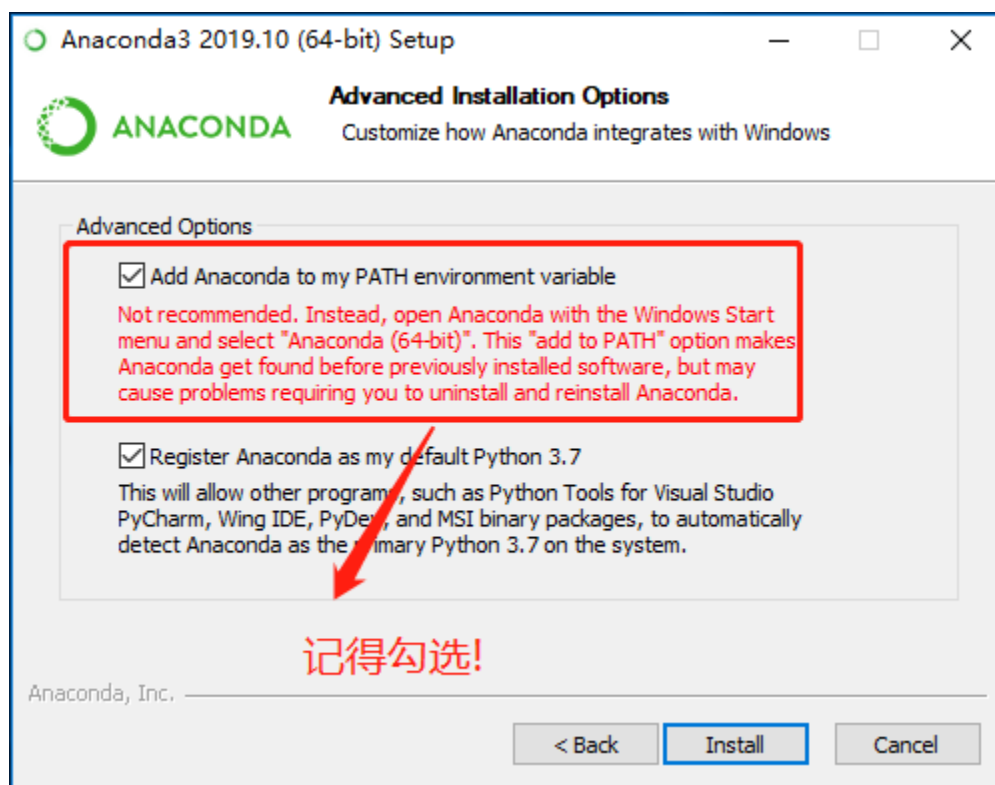




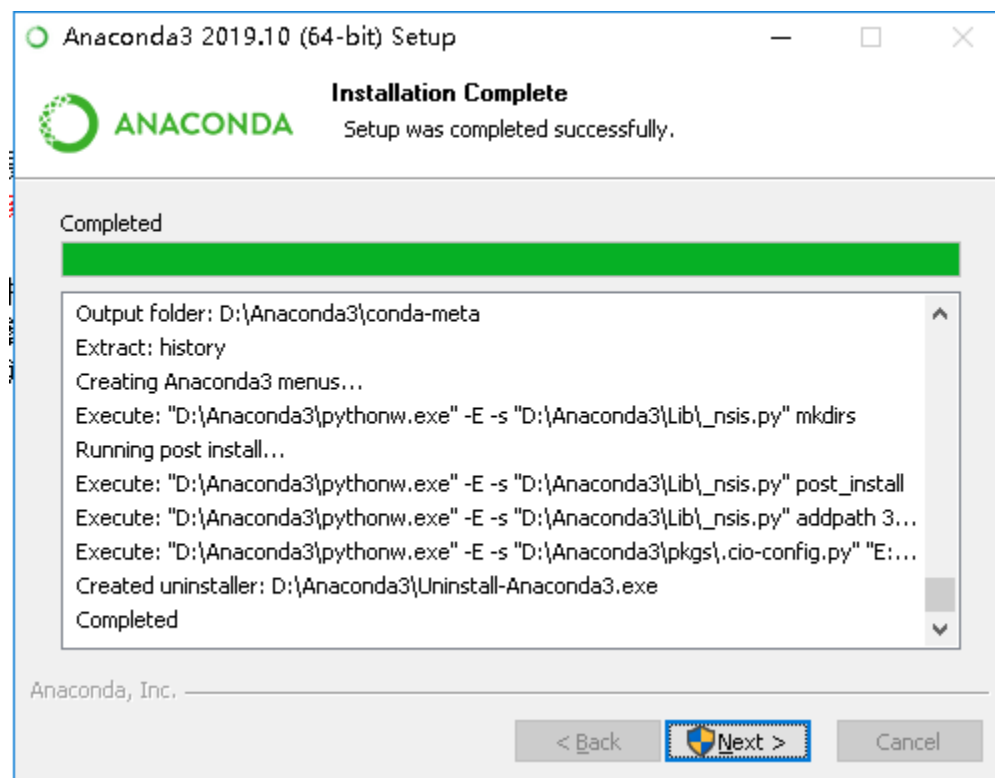
填写安装目录，填写完点击下一步

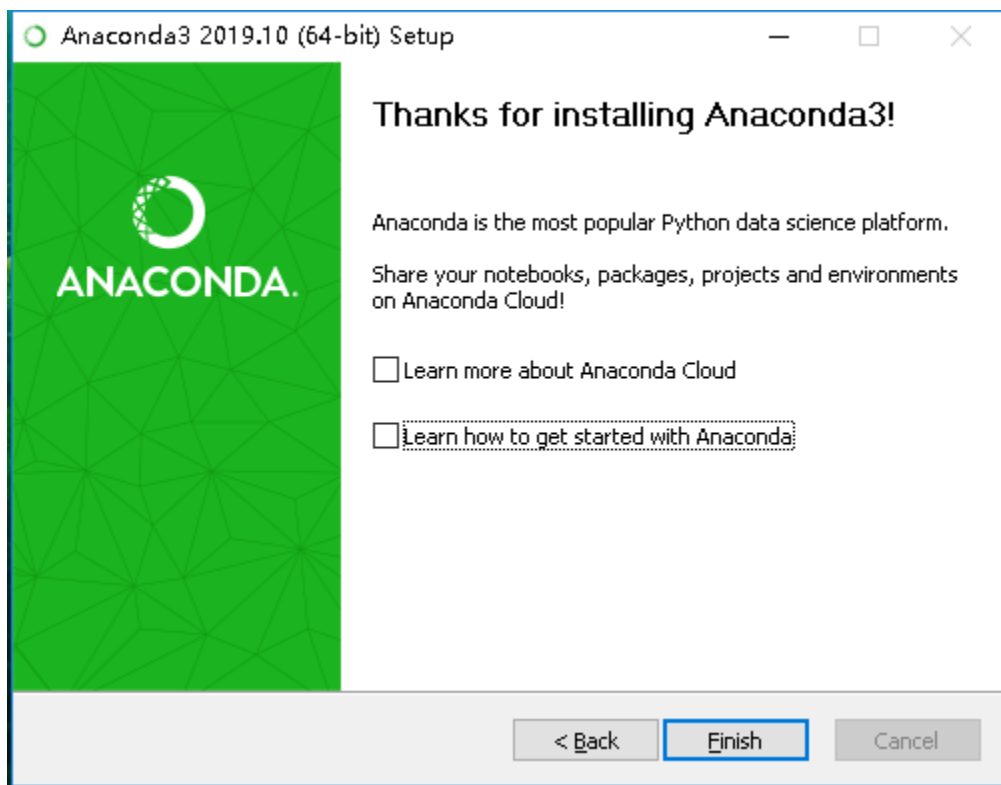


环境变量设置，记得勾选！



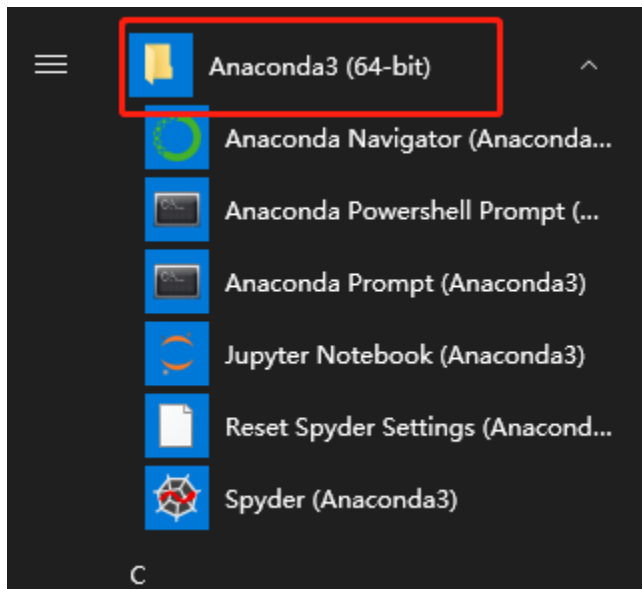
一路下一步完成安装





2、验证是否安装成功

查看系统开始菜单栏下面是否有这个目录



在cmd命令行运行“python --version”得到版本信息

```
C:\Users\Administrator>python --version
Python 3.7.4
```

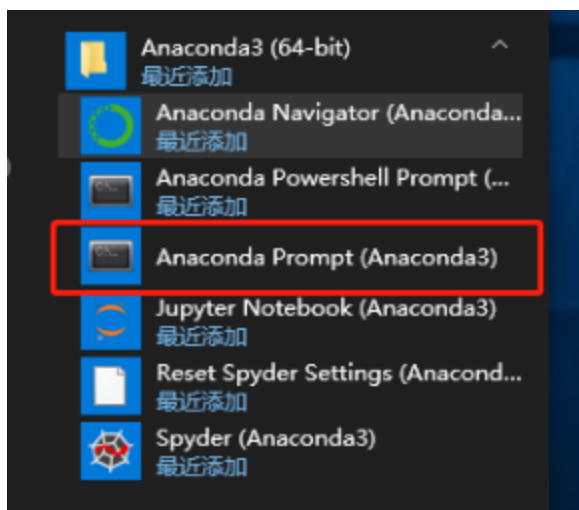
三、安装插件

1、解压插件包到D盘data目录

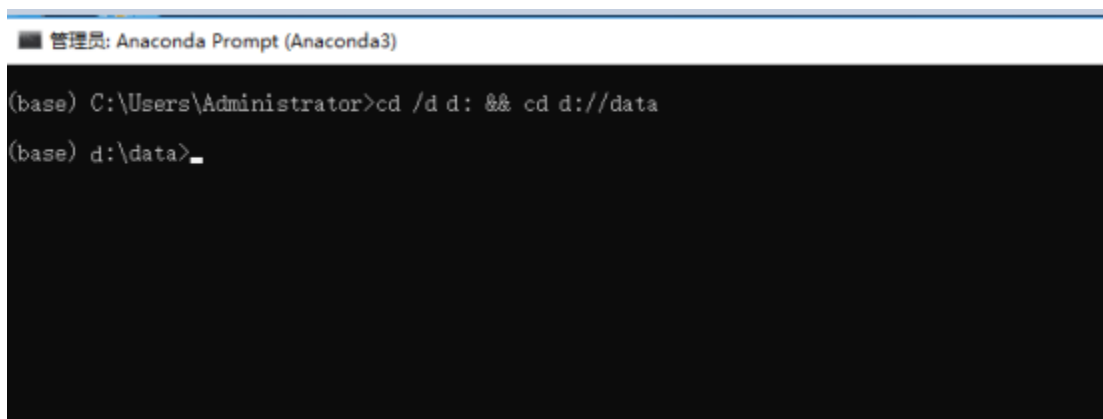
此电脑 > 软件 (D:) > data				
名称	修改日期	类型	大小	
plugin	2022/9/22 17:33	文件夹		
requirements.txt	2021/9/8 14:43	TXT 文件	1 KB	

2、批量离线安装所需插件包

(1) 点击 "Anaconda Prompt (Anaconda3)" 进入命令行界面



(2) 切换目录 `cd /d d://data`



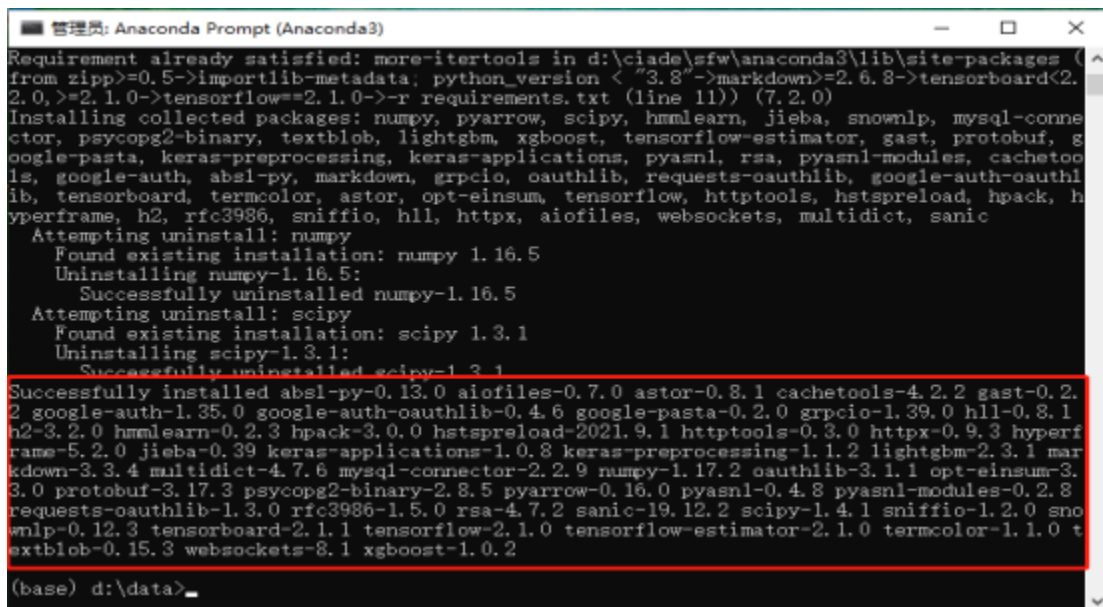
(3) 更新pip版本

```
python ./plugin/pip-20.0.2-py2.py3-none-any.whl/pip install --upgrade ./plugin/pip-20.0.2-py2.py3-none-any.whl
```

```
(base) d:\data\python ./plugin/pip-20.0.2-py2.py3-none-any.whl/pip install --upgrade ./plugin/pip-20.0.2-py2.py3-none-any.whl
Looking in indexes: https://pypi.tuna.tsinghua.edu.cn/simple
Processing ./plugin/pip-20.0.2-py2.py3-none-any.whl
Installing collected packages: pip
  Attempting uninstall: pip
    Found existing installation: pip 19.2.3
    Uninstalling pip-19.2.3:
      Successfully uninstalled pip-19.2.3
Successfully installed pip-20.0.2
```

(4) 批量安装

```
pip install --no-index --find-links=./plugin/ -r requirements.txt
```



```

管理员: Anaconda Prompt (Anaconda3)
Requirement already satisfied: more-itertools in d:\ciade\sfw\anaconda3\lib\site-packages (from zipp>=0.5->importlib-metadata; python_version < "3.8"->markdown)=2.6.8->tensorboard<2.2.0,>=2.1.0->tensorflow==2.1.0->-r requirements.txt (line 11)) (7.2.0)
Installing collected packages: numpy, pyarrow, scipy, hmmlearn, jieba, snownlp, mysql-connector, psycpg2-binary, textblob, lightgbm, xgboost, tensorflow-estimator, gast, protobuf, google-pasta, keras-preprocessing, keras-applications, pyasn1, rsa, pyasn1-modules, cachetools, google-auth, absl-py, markdown, grpcio, oauthlib, requests-oauthlib, google-auth-oauthlib, tensorboard, termcolor, astor, opt-einsum, tensorflow, httptools, hstspreload, hpack, hyperframe, h2, rfc3986, sniffio, hll, httpx, aiofiles, websockets, multidict, sanic
  Attempting uninstall: numpy
    Found existing installation: numpy 1.16.5
    Uninstalling numpy-1.16.5:
      Successfully uninstalled numpy-1.16.5
  Attempting uninstall: scipy
    Found existing installation: scipy 1.3.1
    Uninstalling scipy-1.3.1:
      Successfully uninstalled scipy-1.3.1
Successfully installed absl-py-0.13.0 aiofiles-0.7.0 astor-0.8.1 cachetools-4.2.2 gast-0.2.2 google-auth-1.35.0 google-auth-oauthlib-0.4.6 google-pasta-0.2.0 grpcio-1.39.0 hll-0.8.1 h2-3.2.0 hmmlearn-0.2.3 hpack-3.0.0 hstspreload-2021.9.1 httptools-0.3.0 httpx-0.9.3 hyperframe-5.2.0 jieba-0.39 keras-applications-1.0.8 keras-preprocessing-1.1.2 lightgbm-2.3.1 markdown-3.3.4 multidict-4.7.6 mysql-connector-2.2.9 numpy-1.17.2 oauthlib-3.1.1 opt-einsum-3.3.0 protobuf-3.17.3 psycpg2-binary-2.8.5 pyarrow-0.16.0 pyasn1-0.4.8 pyasn1-modules-0.2.8 requests-oauthlib-1.3.0 rfc3986-1.5.0 rsa-4.7.2 sanic-19.12.2 scipy-1.4.1 sniffio-1.2.0 snownlp-0.12.3 tensorboard-2.1.1 tensorflow-2.1.0 tensorflow-estimator-2.1.0 termcolor-1.1.0 textblob-0.15.3 websockets-8.1 xgboost-1.0.2
(base) d:\data>

```

(5) 验证是否安装成功

```
conda list
```

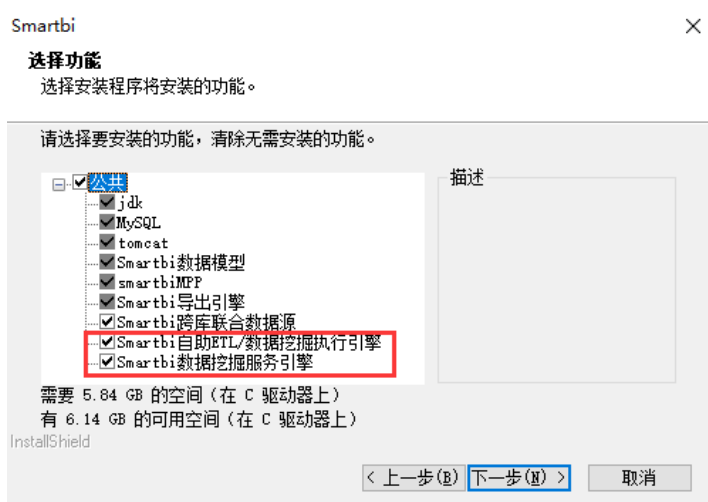
```

(base) d:\data>conda list
# packages in environment at C:\Anaconda3:
#
# Name                    Version                   Build  Channel
_ipyw_jlab_nb_ext_conf    0.1.0                     py37_0
alabaster                  0.7.12                    py37_0
anaconda                   2019.10                   py37_0
anaconda-client            1.7.2                     py37_0
anaconda-navigator         1.9.7                     py37_0
anaconda-project           0.8.3                     py_0
asn1crypto                 1.0.1                     py37_0
astroid                    2.3.1                     py37_0
astropy                    3.2.1                     py37he774522_0
atomicwrites               1.3.0                     py37_1
attrs                      19.2.0                    py_0
babel                      2.7.0                     py_0
backcall                   0.1.0                     py37_0
backports                  1.0                        py_2
backports.functools_lru_cache 1.5                       py_2
backports.os               0.1.1                     py37_0
backports.shutil_get_terminal_size 1.0.0                    py37_2
backports.tempfile         1.0                        py_1
backports.weakref          1.0.post1                 py_1
beautifulsoup4             4.8.0                     py37_0
bitarray                   1.0.1                     py37he774522_0
bkcharts                   0.2                       py37_0
blas                       1.0                        mkl
bleach                     3.1.0                     py37_0
blosc                      1.10.3                    h7bd577a_0
bokeh                      1.3.4                     py37_0

```

四、部署挖掘引擎

在图形化部署时在勾选 **Smartbi自助ETL/数据挖掘执行引擎**和**Smartbi数据挖掘服务引擎**

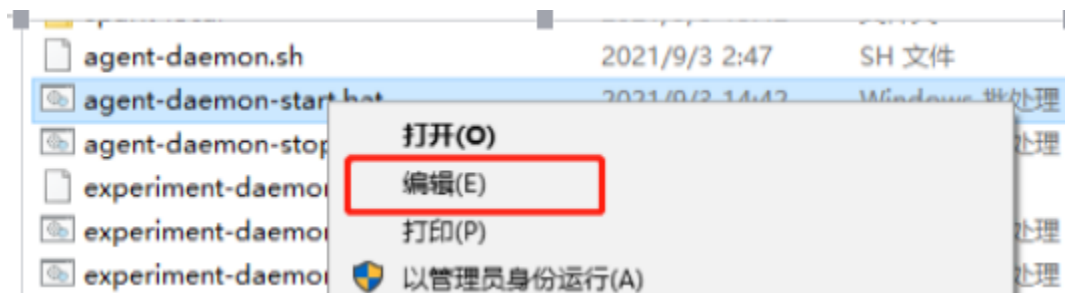


若在此处无勾选，则可参考：[Windows 部署挖掘引擎](#) 进行部署。

五、启动python代理器

启动Anaconda之前要保证挖掘引擎已经完成启动。

1、以文本编辑打开启动脚本文件(agent-daemon-start.bat)并设置JAVA_HOME(修改成自己的jdk目录)




```
agent-daemon-start.bat - 记事本
文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)
set JAVA_HOME=C:\Smartbi\jdk
set PATH=%JAVA_HOME%\bin;%PATH%
java -version
java -Dfile.encoding=UTF8 -Dsun.jnu.encoding=UTF8 -Xmx1g -Xms1g -cp ...
```

修改成自己的目录

2、cmd命令行下运行启动脚本 agent-daemon-start.bat

```
C:\Smartbi\SmartbiEngine\engine\sbin>agent-daemon-start.bat
C:\Smartbi\SmartbiEngine\engine\sbin>set JAVA_HOME=C:\Smartbi\jdk
C:\Smartbi\SmartbiEngine\engine\sbin>set PATH=C:\Smartbi\jdk\bin;C:\Windows\system32;C:\Windows;C:\Windows\System32\Wbem;C:\Windows\System32\WindowsPowerShell\v1.0\;C:\Windows\System32\OpenSSH\;D:\workplace\Git\bin;C:\Anaconda3;C:\Anaconda3\Library\mingw-w64\bin;C:\Anaconda3\Library\usr\bin;C:\Anaconda3\Library\bin;C:\Anaconda3\Scripts;C:\Users\Administrator\AppData\Local\Microsoft\WindowsApps
C:\Smartbi\SmartbiEngine\engine\sbin>java -version
java version "1.8.0_202-ea"
Java(TM) SE Runtime Environment (build 1.8.0_202-ea-b03)
Java HotSpot(TM) 64-Bit Server VM (build 25.202-b03, mixed mode)
C:\Smartbi\SmartbiEngine\engine\sbin>java -Dfile.encoding=UTF8 -Dsun.jnu.encoding=UTF8 -Xmx1g -Xms1g -cp .../lib/* smartbi.datamining.engine.agent.AgentApplication --master http://localhost:8899 --env python
2021-09-03 16:09:03.997 [1] INFO util.log.initialized:169 - Logging initialized @1866ms to org.eclipse.jetty.util.log.Slf4jLog
```

```
da:1388
2021-09-03 16:09:20.669 [50] INFO scheduler.DAGScheduler.logInfo:57 - Submitting 1 missing tasks from ResultStage 1 (MapPartitionsINFO[5] at count at SparkSQLExample.java:79) (first 15 tasks are for partitions Vector(0))
2021-09-03 16:09:20.671 [50] INFO scheduler.TaskSchedulerImpl.logInfo:57 - Adding task set 1.0 with 1 tasks resource profile 0
2021-09-03 16:09:20.696 [23] INFO scheduler.TaskSetManager.logInfo:57 - Starting task 0.0 in stage 1.0 (TID 1) (DESKTOP-5DJRILL, executor driver, partition 0, NODE_LOCAL, 4453 bytes) taskResourceAssignments Map()
2021-09-03 16:09:20.698 [60] INFO executor.Executor.logInfo:57 - Running task 0.0 in stage 1.0 (TID 1)
2021-09-03 16:09:20.903 [60] INFO storage.ShuffleBlockFetcherIterator.logInfo:57 - Getting 1 (60.0 B) non-empty blocks including 1 (60.0 B) local and 0 (0.0 B) host-local and 0 (0.0 B) remote blocks
2021-09-03 16:09:20.908 [60] INFO storage.ShuffleBlockFetcherIterator.logInfo:57 - Started 0 remote fetches in 18 ms
2021-09-03 16:09:20.956 [60] INFO executor.Executor.logInfo:57 - Finished task 0.0 in stage 1.0 (TID 1). 2691 bytes result sent to driver
2021-09-03 16:09:20.968 [62] INFO scheduler.TaskSetManager.logInfo:57 - Finished task 0.0 in stage 1.0 (TID 1) in 280 ms on DESKTOP-5DJRILL (executor driver) (1/1)
2021-09-03 16:09:20.972 [50] INFO scheduler.DAGScheduler.logInfo:57 - ResultStage 1 (count at SparkSQLExample.java:79) finished in 0.349 s
2021-09-03 16:09:21.000 [50] INFO scheduler.DAGScheduler.logInfo:57 - Job 0 is finished. Cancelling potential speculative or zombie tasks for this job
2021-09-03 16:09:20.975 [62] INFO scheduler.TaskSchedulerImpl.logInfo:57 - Removed TaskSet 1.0, whose tasks have all completed, from pool
2021-09-03 16:09:21.006 [50] INFO scheduler.TaskSchedulerImpl.logInfo:57 - Killing all running tasks in stage 1: Stage finished
2021-09-03 16:09:21.015 [1] INFO scheduler.DAGScheduler.logInfo:57 - Job 0 finished: count at SparkSQLExample.java:79, took 1.882065 s
2021-09-03 16:09:21.028 [1] INFO util.SparkSQLExample.runExample:80 - Number of dataset:1
2021-09-03 16:09:21.062 [1] INFO util.SparkSQLExample.runExample:81 - SparkSQL example executed successfully
2021-09-03 16:09:24.441 [51] WARN executor.ProcfsMetricsGetter.logWarning:69 - Exception when trying to compute pagesize as a result reporting of ProcessTree metrics is stopped
```

六、测试python节点

1、打开数据挖掘 - 新建实验



2、拖拽“示例数据源”，选择数据源 - 保存 - 运行 - 运行成功 - 保存



3、拖拽PYTHON脚本，与示例数据源连线，点击运行，如果显示运行成功则表示Python计算节点正常

Python

运行成功: 9秒

数据源

- 文本数据源
- Kafka数据源
- 关系数据源
- 示例数据源
- 数据集
- Excel文件
- 读取Excel sheet

目标源

- 已训练模型

数据预处理

- 特征工程
- 统计分析
- 评分卡分析
- 文本分析

机器学习

- 脚本模块
 - SQL脚本
 - PYTHON脚本
- 自定义模块
- 服务

示例数据源

PYTHON脚本

1.拖拽Python脚本

2.点击运行

运行成功: 9秒