

## TERA KOLLECT ML PACK DEPLOYMENT INSTRUCTIONS – WINDOWS

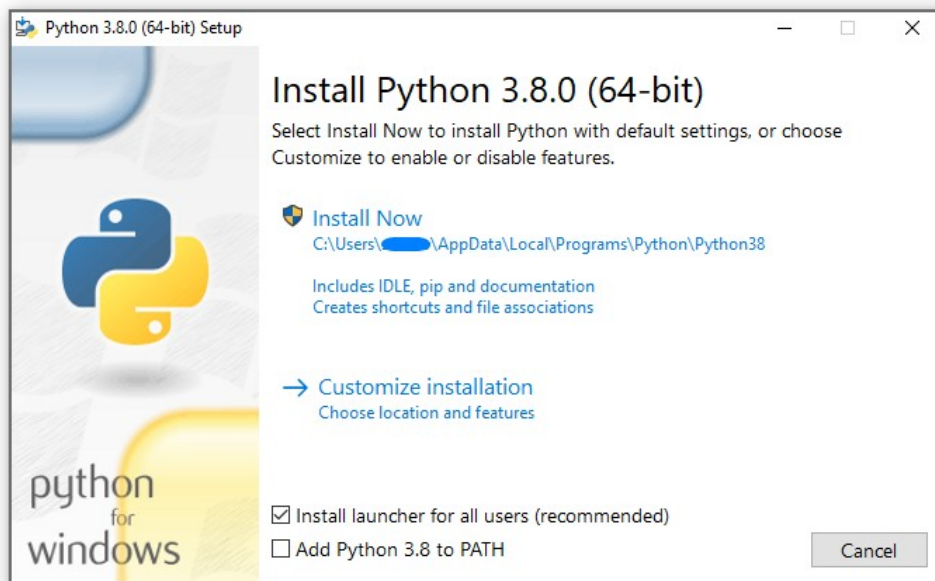
### STEP 1: PYTHON INSTALLATION

\* Python 3.8 should be installed initially. Download python3.8 (preferred 3..8.10) exe file from the official website. Added link in below and this may change based on their site maintenance

link : <https://www.python.org/downloads/release/python-3810/>

\* double tap to click install and check **add python 3.8 add to path and** click **Install Now**. Now click install for all users and continue steps further.

\* Note down the **path of installing** while installing. This path should be used during the environment variable setup



\* To check whether it is installed or not, open command prompt and type below to validate

~\$ **python3 --version**

```
Microsoft Windows [Version 10.0.14393]
(c) 2016 Microsoft Corporation. All rights reserved.

C:\Users\Terraoffline>python --version
Python 3.8.10

C:\Users\Terraoffline>_
```

\* If you get **errors**, try adding path's manually to windows environment variables. For this you've to find the python installation path in C drive.

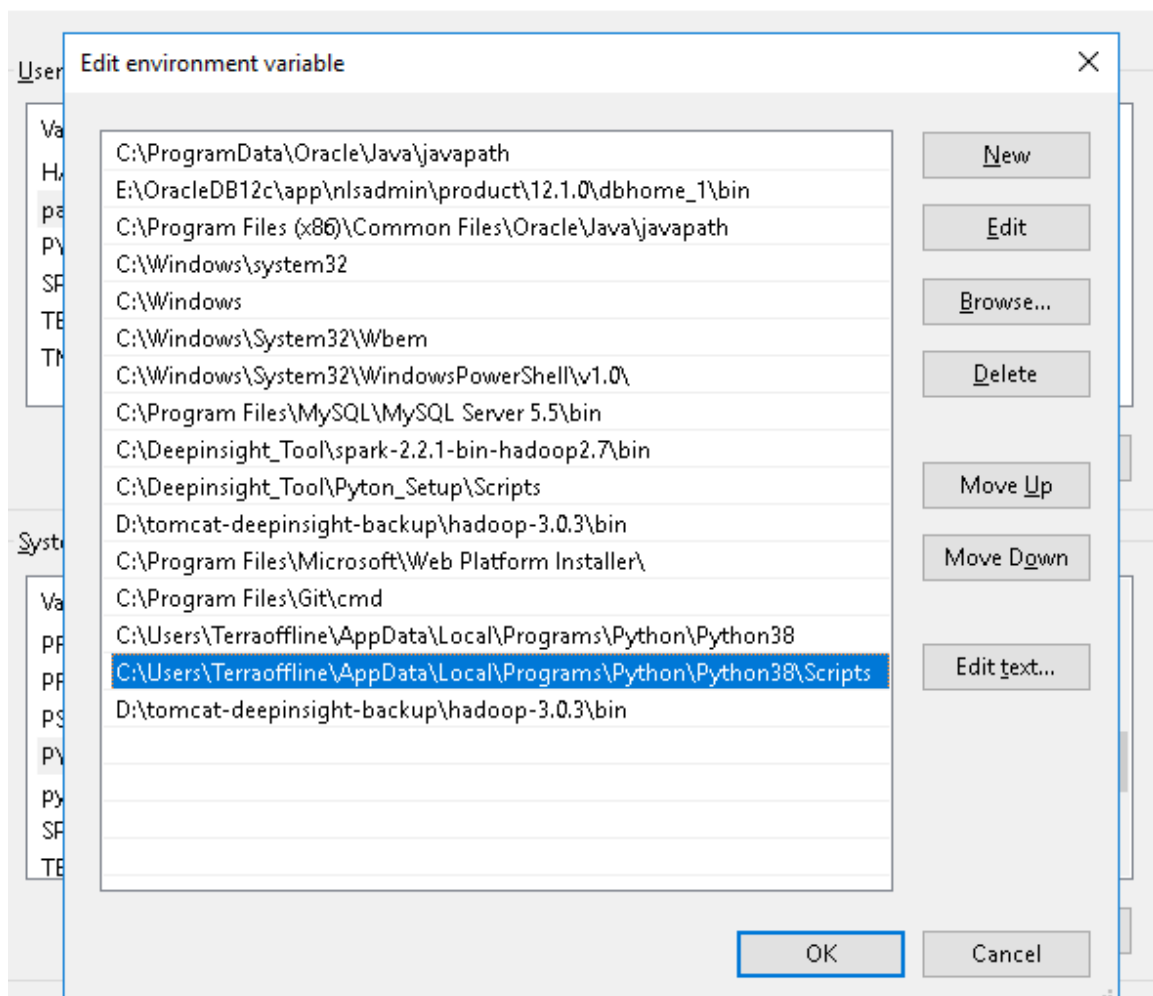
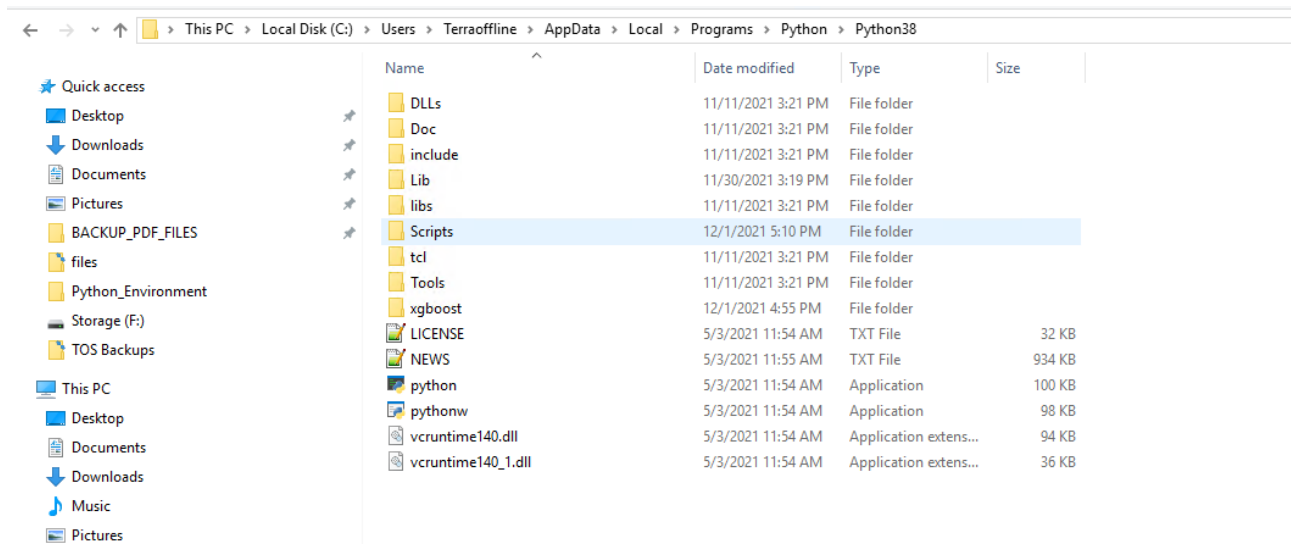
Go to --> mycomputer -right click --> properties --> advanced system settings --> environment variables --> system variables

## TERA KOLLECT ML PACK DEPLOYMENT INSTRUCTIONS – WINDOWS

1) In system variables click on **path** and add the path you've installed python & Scripts folder

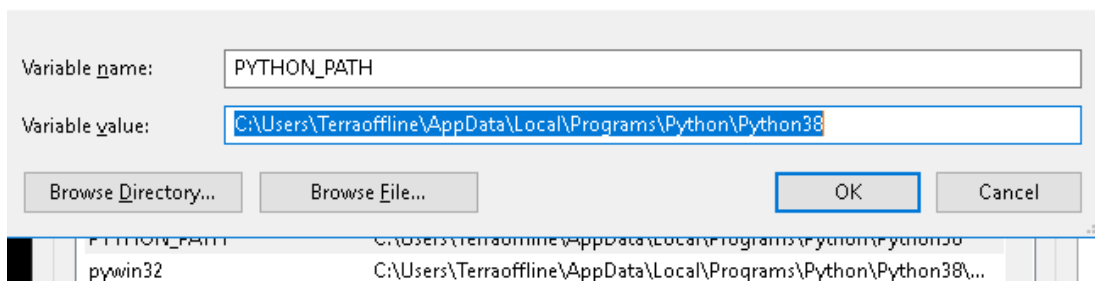
ex : C:\Users\Terraoffline\AppData\Local\Programs\Python\Python38

C:\Users\Terraoffline\AppData\Local\Programs\Python\Python38\Scripts

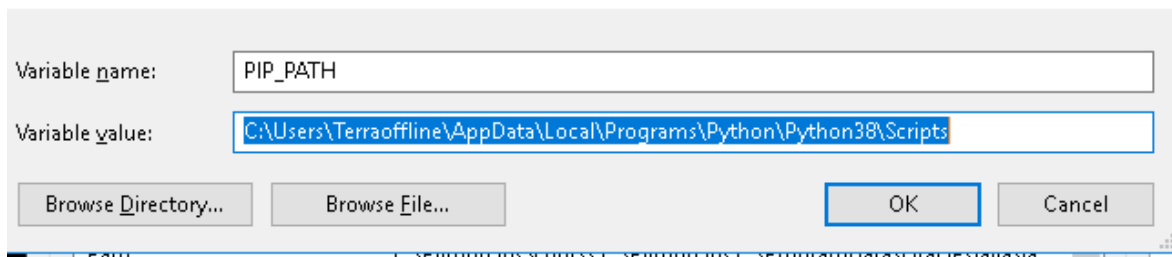


## TERA KOLLECT ML PACK DEPLOYMENT INSTRUCTIONS – WINDOWS

In environment variable click new and add python installed source path as python path



add one more variable, In python source path you may find Scripts folder add that folder as PIP\_PATH

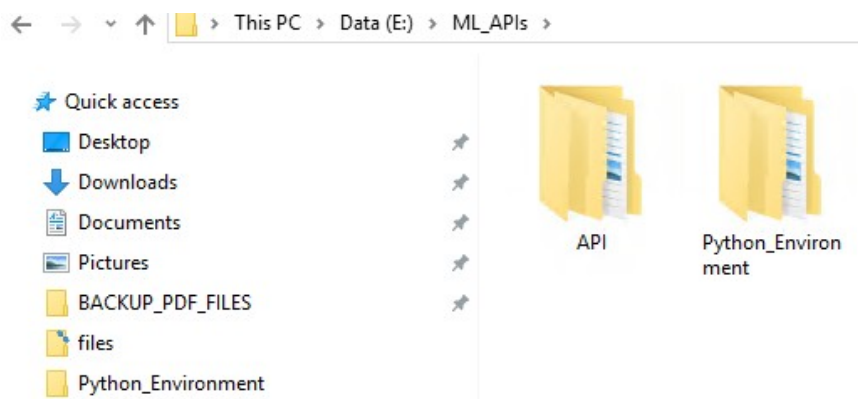


Save and do check now. You may able to get the python version in cmd.

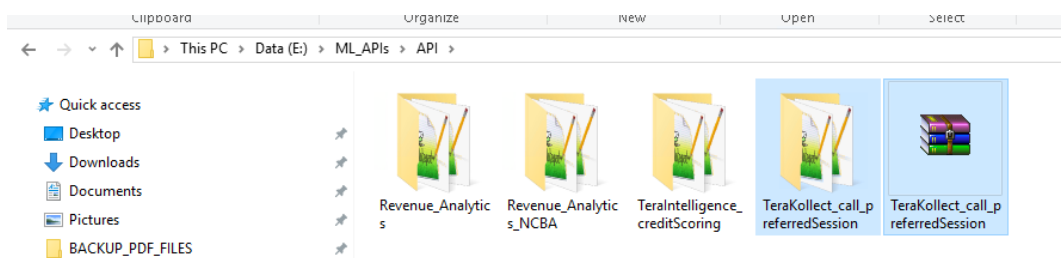
### STEP 2: FOLDER STRUCTURE

\* To create the folder structure go to the respective path you've allocated for installation and create a following directory

\* API                      \*Python\_Environments



In API folder paste the source file and extract the file

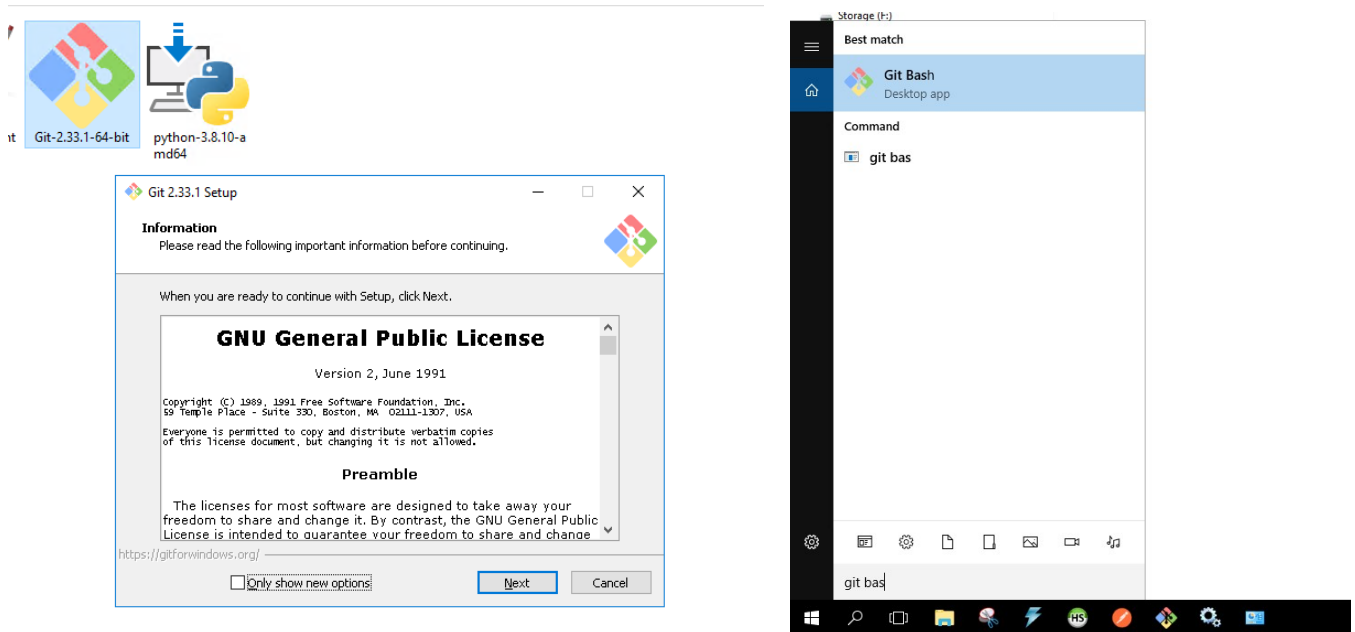


## TERA KOLLECT ML PACK DEPLOYMENT INSTRUCTIONS – WINDOWS

### STEP 3: GIT TERMINAL SETUP

\* Download the git terminal using below link and do install it. Do search for Git Bash and open the terminal


Link : <https://git-scm.com/downloads>

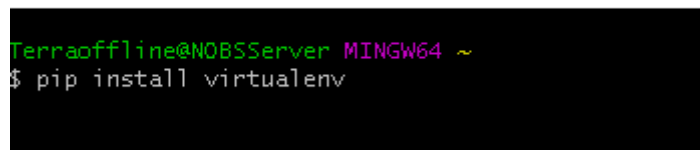


### STEP 4: ENVIRONMENT SETUP

\* Create the virtual environment and install necessary libraries to the respective environment. To create the virtual environment, first to install virtual environment setup

~\$ **pip install virtualenv**

 MINGW64:/c:/Users/Terraoffline



and followed by to create the virtual environment in Python\_Environment folder we've created. Navigate to python environment folder using below command

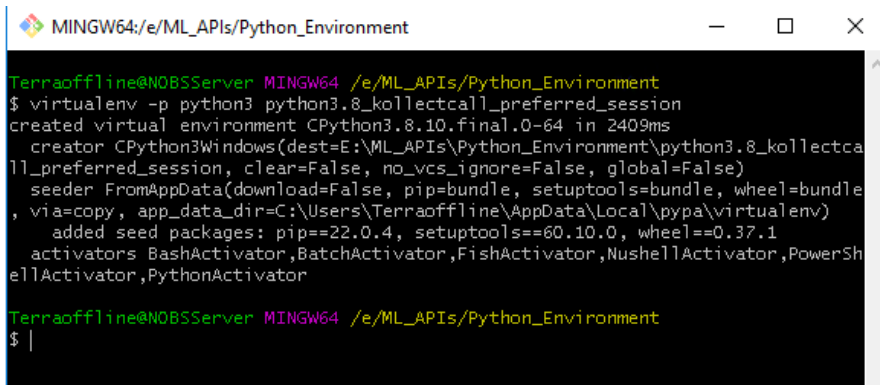
~\$ **cd E:/ML\_APIs/Python\_Environment**

and to install python environment

~\$ **virtualenv -p python3 python3.8\_kollectcall\_preferred\_session**

here navigated to Python\_Environments path for installing python environment and python3.8\_kollectcall\_preferred\_session is the environment name

## TERA KOLLECT ML PACK DEPLOYMENT INSTRUCTIONS – WINDOWS



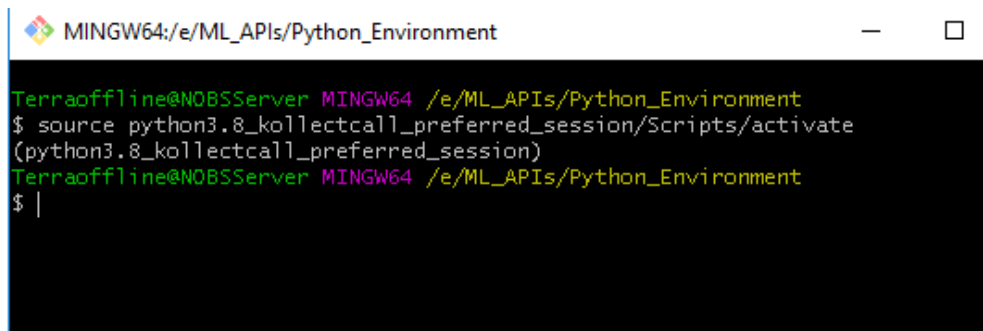
```
MINGW64:/e/ML_APIs/Python_Environment
Terraoffline@NOBSServer MINGW64 /e/ML_APIs/Python_Environment
$ virtualenv -p python3 python3.8_kollectcall_preferred_session
created virtual environment CPython3.8.10.final.0-64 in 2409ms
creator CPython3Windows(dest=E:\ML_APIs\Python_Environment\python3.8_kollectca
ll_preferred_session, clear=False, no_vcs_ignore=False, global=False)
seeder FromAppData(download=False, pip=bundle, setuptools=bundle, wheel=bundle
, via=copy, app_data_dir=C:\Users\Terraoffline\AppData\Local\pypa\virtualenv)
added seed packages: pip==22.0.4, setuptools==60.10.0, wheel==0.37.1
activators BashActivator,BatchActivator,FishActivator,NusHELLActivator,PowerSh
ellActivator,PythonActivator

Terraoffline@NOBSServer MINGW64 /e/ML_APIs/Python_Environment
$ |
```

once the environment installed you may find the folder in the respective path you've mentioned

\* To activate the created python environment

~\$ **source python3.8\_kollectcall\_preferred\_session/Scripts/activate**



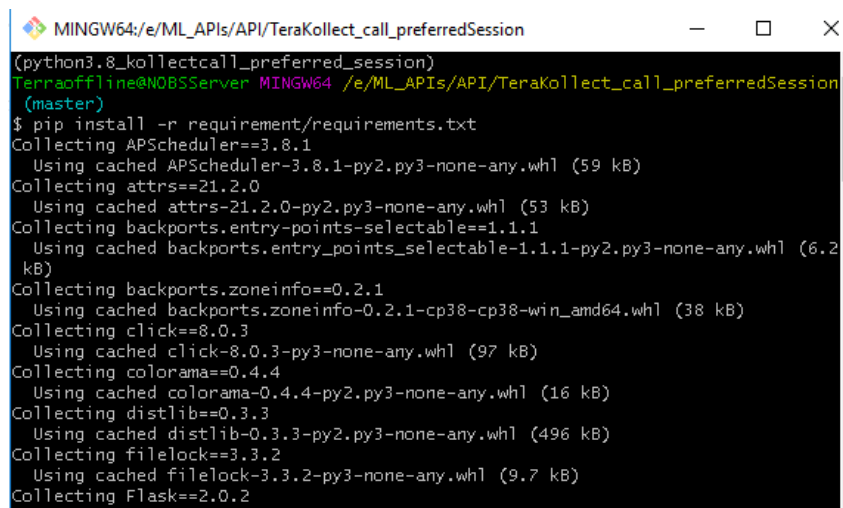
```
MINGW64:/e/ML_APIs/Python_Environment
Terraoffline@NOBSServer MINGW64 /e/ML_APIs/Python_Environment
$ source python3.8_kollectcall_preferred_session/Scripts/activate
(python3.8_kollectcall_preferred_session)
Terraoffline@NOBSServer MINGW64 /e/ML_APIs/Python_Environment
$ |
```

Here we navigated to python environment directory and then /Scripts/activate will activate your environment. you may find the environment name in terminal

\* Now install the necessary libraries created environment. first navigate to project directory we've extracted and in that find requirement folder & in that requirement.txt have the all the necessary libraries

~\$ **cd E:/ML\_APIs/API/TeraKollect\_call\_preferredSession**

~\$ **pip install -r requirement/requirements.txt**



```
MINGW64:/e/ML_APIs/API/TeraKollect_call_preferredSession
(python3.8_kollectcall_preferred_session)
Terraoffline@NOBSServer MINGW64 /e/ML_APIs/API/TeraKollect_call_preferredSession
(master)
$ pip install -r requirement/requirements.txt
Collecting APScheduler==3.8.1
  Using cached APScheduler-3.8.1-py2.py3-none-any.whl (59 kB)
Collecting attrs==21.2.0
  Using cached attrs-21.2.0-py2.py3-none-any.whl (53 kB)
Collecting backports.entry-points-selectable==1.1.1
  Using cached backports.entry_points_selectable-1.1.1-py2.py3-none-any.whl (6.2
kB)
Collecting backports.zoneinfo==0.2.1
  Using cached backports.zoneinfo-0.2.1-cp38-cp38-win_amd64.whl (38 kB)
Collecting click==8.0.3
  Using cached click-8.0.3-py3-none-any.whl (97 kB)
Collecting colorama==0.4.4
  Using cached colorama-0.4.4-py2.py3-none-any.whl (16 kB)
Collecting distlib==0.3.3
  Using cached distlib-0.3.3-py2.py3-none-any.whl (496 kB)
Collecting filelock==3.3.2
  Using cached filelock-3.3.2-py3-none-any.whl (9.7 kB)
Collecting Flask==2.0.2
  Using cached Flask-2.0.2-py3-none-any.whl (95 kB)
```

## TERA KOLLECT ML PACK DEPLOYMENT INSTRUCTIONS – WINDOWS

it will take some time based on the libraries we've used

### STEP 5: DATABASE PREREQUISITES

\* Fetching data from the customer\$information table and make sure below fields are exist in that table

```
~$ SELECT ID, GENDER, CUSTOMER_RISK_STATUS, INDUSTRY, RESIDENCE,  
CUSTOMER_STATUS, PROMISE_SUCCESS, TIME_STAMP, REPAY_CURRENCY, STAGE from  
customer$information
```

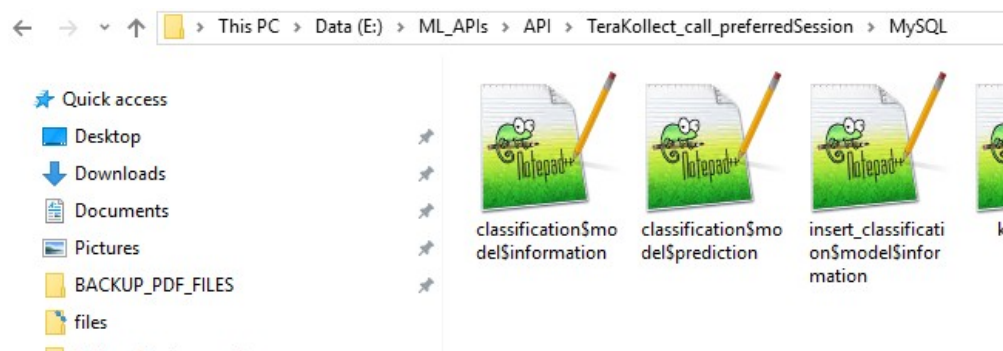
\* In further need two table's to store the model information and input/output details

1) **classification\$model\$information**

2) **classification\$model\$prediction**

you may find the DDL script in project file under MYSQL folder and do execute the script.

\* one last to execute the insert scripts in the **insert\_classification\$model\$information.sql** under MYSQL folder in project directory



### STEP 6: APPLICATION PREREQUISITES

\* we are almost done, on last final step to setup is to set up database credentials in config file. To do that in source file go to the config folder and open **config.sh** using notepad. In that change the values as per the requirement

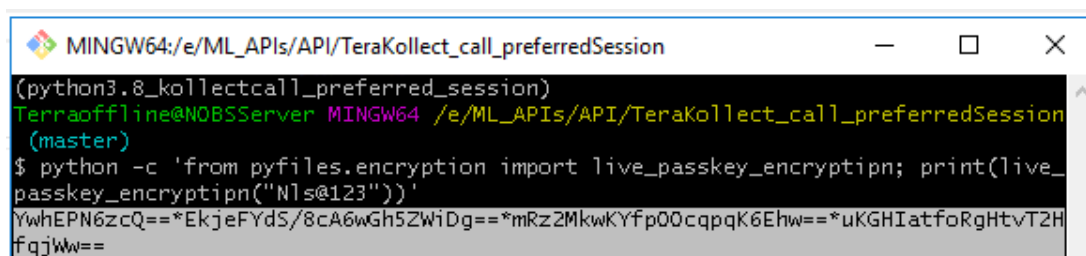
To encrypt the MYSQL password do the below

\* Activate the environment as per above commands

\* Nagavigate to project directory using above guidance and do the follow

```
~$ source config/config.sh
```

```
~$ python -c 'from pyfiles.encryption import live_passkey_encryptipn;  
print(live_passkey_encryptipn("PASSWORD_STRING"))'
```



## TERA KOLLECT ML PACK DEPLOYMENT INSTRUCTIONS – WINDOWS

Replace password string with the respective password and enter you will get the encrypted string . Replace the encrypted string with LIVE\_MYSQL\_PASSWORD string

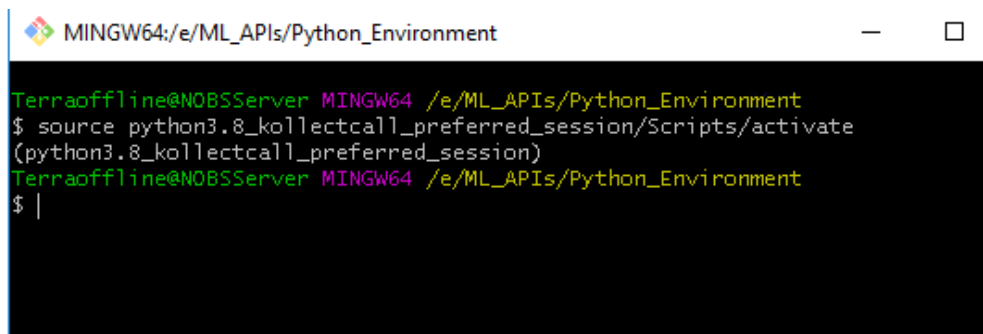
```
export LIVE_MYSQL_USER="root"
export LIVE_MYSQL_PASSWORD="Rs6wIA==*uuYEPmy6/9eshVbHbQ/U4A" # Password encrypted
export LIVE_MYSQL_HOST="127.0.0.1"
export LIVE_MYSQL_PORT=3306
export LIVE_MYSQL_DB="kollect_cba"
```

```
## UAT/LIVE ENVIRONMENT MYSQL CONFIG
export LIVE_MYSQL_USER="root"
export LIVE_MYSQL_PASSWORD="+JRa6wwUIA==*uuYEPmy6/9eshVbHbQ/U4A==*D7XI1h4lMMduhM635WMLgw==*1aha5rPdMuUFcWTl2sRKbQ==" # Password encrypted
export LIVE_MYSQL_HOST="127.0.0.1"
export LIVE_MYSQL_PORT=3306
export LIVE_MYSQL_DB="kollect_cba"
```

### STEP 7 : RUN OUR APPLICATION

\* To run our application first to active the python environment we've created initially. To active the environment navigate the respective directory and do activate

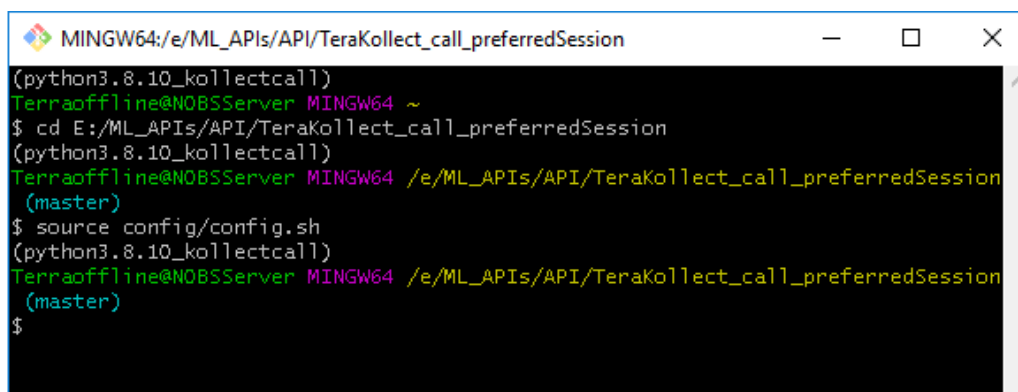
```
~$ cd E:/ML_APIs/Python_Environment
~$ source python3.8_kollectcall_preferred_session/Scripts/activate
```

A terminal window titled 'MINGW64:/e/ML\_APIs/Python\_Environment' shows the following commands and output:

```
Terraoffline@NOBSServer MINGW64 /e/ML_APIs/Python_Environment
$ source python3.8_kollectcall_preferred_session/Scripts/activate
(python3.8_kollectcall_preferred_session)
Terraoffline@NOBSServer MINGW64 /e/ML_APIs/Python_Environment
$ |
```

\* Next, to active the configuration files which has the DB related information. To active the config file navigate to project directory and execute below commands

```
~$ cd E:/ML_APIs/API/TeraKollet_call_preferredSession
~$ source config/config.sh
```

A terminal window titled 'MINGW64:/e/ML\_APIs/API/TeraKollet\_call\_preferredSession' shows the following commands and output:

```
(python3.8.10_kollectcall)
Terraoffline@NOBSServer MINGW64 ~
$ cd E:/ML_APIs/API/TeraKollet_call_preferredSession
(python3.8.10_kollectcall)
Terraoffline@NOBSServer MINGW64 /e/ML_APIs/API/TeraKollet_call_preferredSession
(master)
$ source config/config.sh
(python3.8.10_kollectcall)
Terraoffline@NOBSServer MINGW64 /e/ML_APIs/API/TeraKollet_call_preferredSession
(master)
$
```

\* To execute the application as background process

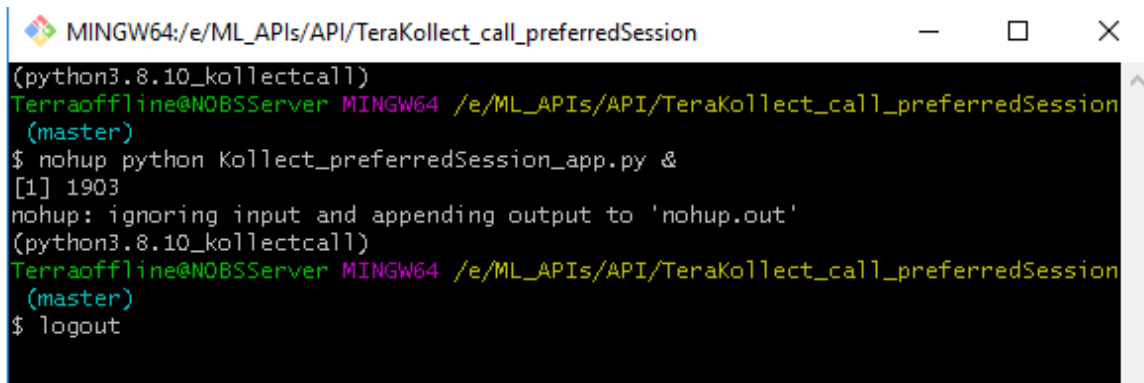
## TERA KOLLECT ML PACK DEPLOYMENT INSTRUCTIONS – WINDOWS

~\$ **nohup python Kollekt\_preferredSession\_app.py &**

and type

~\$ **logout**

and close terminal and execute the process in background



```
MINGW64:/e/ML_APIs/API/TeraKollekt_call_preferredSession
(python3.8.10_kollectcall)
Terraoffline@NOBSServer MINGW64 /e/ML_APIs/API/TeraKollekt_call_preferredSession
(master)
$ nohup python Kollekt_preferredSession_app.py &
[1] 1903
nohup: ignoring input and appending output to 'nohup.out'
(python3.8.10_kollectcall)
Terraoffline@NOBSServer MINGW64 /e/ML_APIs/API/TeraKollekt_call_preferredSession
(master)
$ logout
```

### STEP 8: VALIDATE APPLICATION RUNNING OR NOT

\* once application is up, the application creates the log files in source file directory under folder log named **errorlog.log**. You may open using notepad and see the latest time you've started the application.

```
1 |
2 | -----
3 | Datetime - 2022-04-20 11:44:24,785 & Filename - werkzeug & Type - WARNING
4 | -----
5 | * Running on all addresses.
```

\* In another way we may pass the respective url and input details in postman to get our respective output

METHOD TYPE : post

URL: [http://127.0.0.1:2012/kollectCall\\_preferredSession\\_xgbClassifier\\_modelPredictions](http://127.0.0.1:2012/kollectCall_preferredSession_xgbClassifier_modelPredictions)

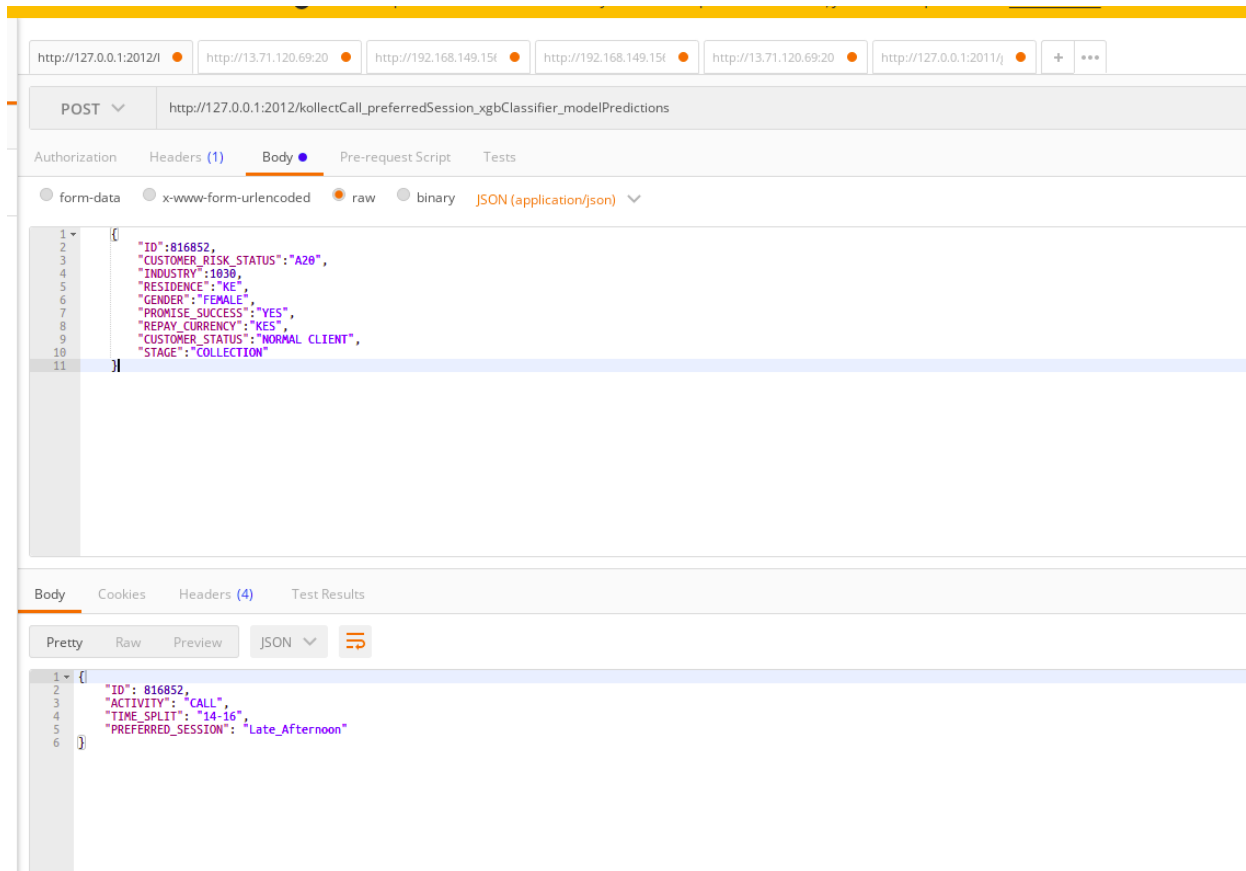
INPUT: {

```
"ID":816852,
"CUSTOMER_RISK_STATUS":"A20",
"INDUSTRY":1030,
"RESIDENCE":"KE",
"GENDER":"FEMALE",
"PROMISE_SUCCESS":"YES",
"REPAY_CURRENCY":"KES",
"CUSTOMER_STATUS":"NORMAL CLIENT",
"STAGE":"COLLECTION"
}
```

and you'll get the respective output as mentioned



## TERA KOLLECT ML PACK DEPLOYMENT INSTRUCTIONS – WINDOWS

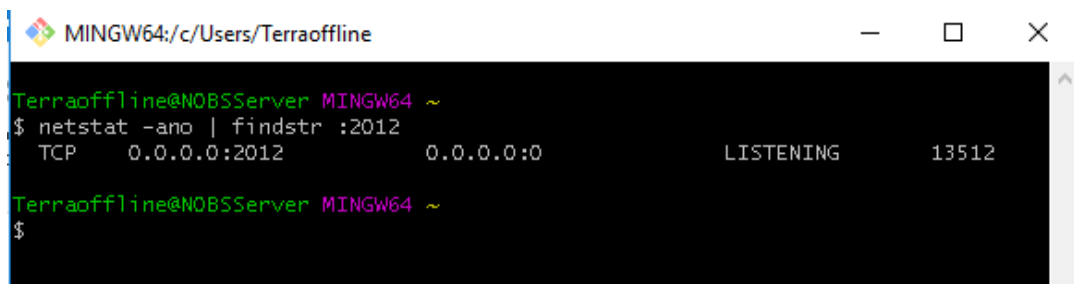


### \* FURTHER VERSION'S DEPLOYMENT

In the above we've seen how to deploy the fresh version. here quickly brief how to deploy the next and further releases

- \* Extract the latest file to the destination folder
- \* Activate the respective python environment using above commands
- \* navigate to the respective project directory in terminal
- \* First step to terminate the process we've executed already in past. We're running our application in port no 2012 and going to kill our process using port number. To do that in cmd

~\$ **netstat -ano | findstr :2012**

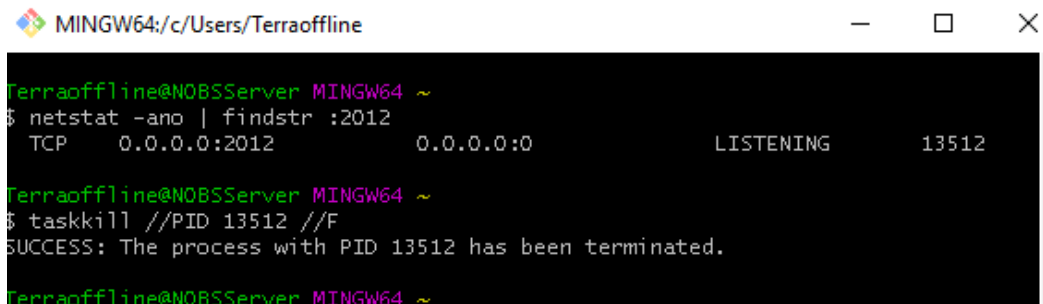


## TERA KOLLECT ML PACK DEPLOYMENT INSTRUCTIONS – WINDOWS

and using the above PID (**13512**) we have to kill the process. (Some times it may show multiple PID we have to kill all). To double check execute the cmd again and again till it is empty

To kill the PID

~\$ **taskkill //PID 13512 //F**

A screenshot of a Windows terminal window titled "MINGW64:/c/Users/Terraoffline". The terminal shows the following commands and output:

```
Terraoffline@NOBSServer MINGW64 ~  
$ netstat -ano | findstr :2012  
TCP 0.0.0.0:2012 0.0.0.0:0 LISTENING 13512  
  
Terraoffline@NOBSServer MINGW64 ~  
$ taskkill //PID 13512 //F  
SUCCESS: The process with PID 13512 has been terminated.  
  
Terraoffline@NOBSServer MINGW64 ~
```

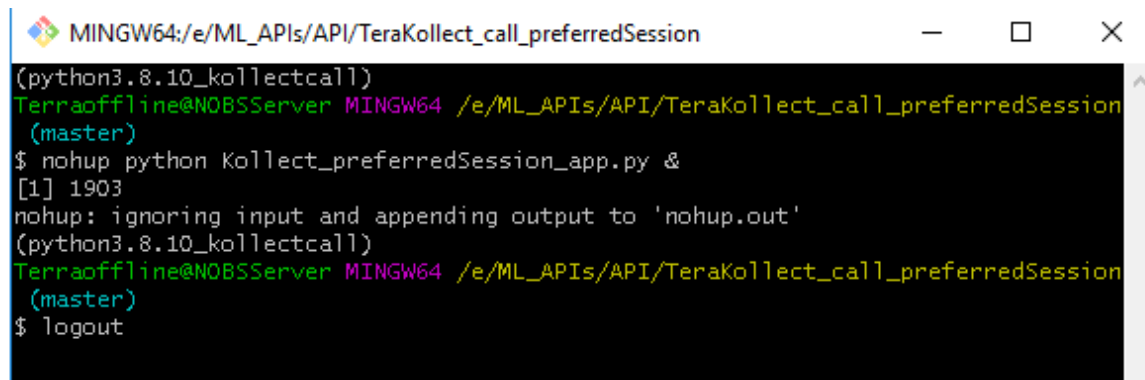
To Run our application again

~\$ **nohup python Kollekt\_preferredSession\_app.py &**

and type

~\$ **logout**

and close terminal and execute the process in background

A screenshot of a Windows terminal window titled "MINGW64:/e/ML\_APIs/API/TeraKollekt\_call\_preferredSession". The terminal shows the following commands and output:

```
(python3.8.10_kollektcall)  
Terraoffline@NOBSServer MINGW64 /e/ML_APIs/API/TeraKollekt_call_preferredSession  
(master)  
$ nohup python Kollekt_preferredSession_app.py &  
[1] 1903  
nohup: ignoring input and appending output to 'nohup.out'  
(python3.8.10_kollektcall)  
Terraoffline@NOBSServer MINGW64 /e/ML_APIs/API/TeraKollekt_call_preferredSession  
(master)  
$ logout
```

and this need's to be validated further, whether the application running or not using the above methods mentioned