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 maintenence

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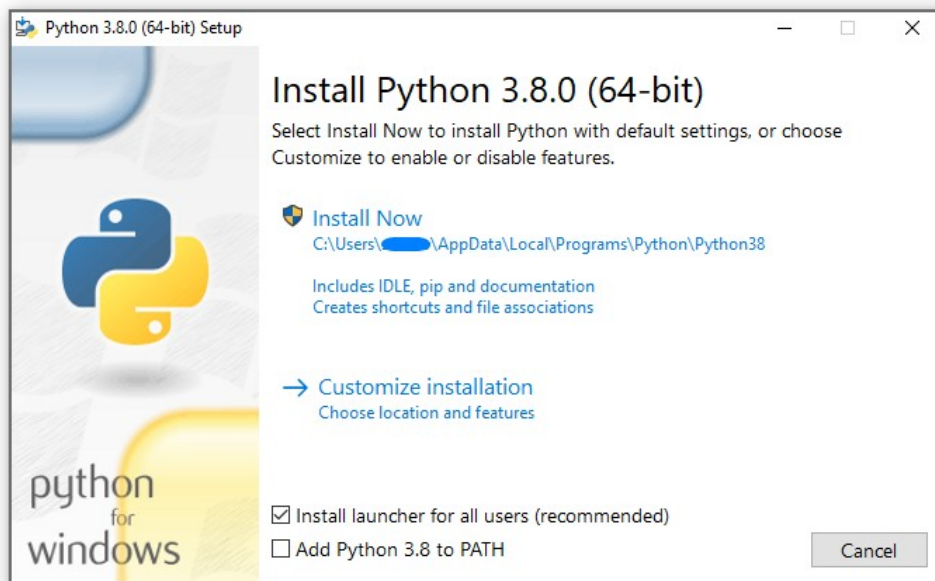
* To check whether it is installed or not, open command prompt and type below to validate

~$ **python3 –version**

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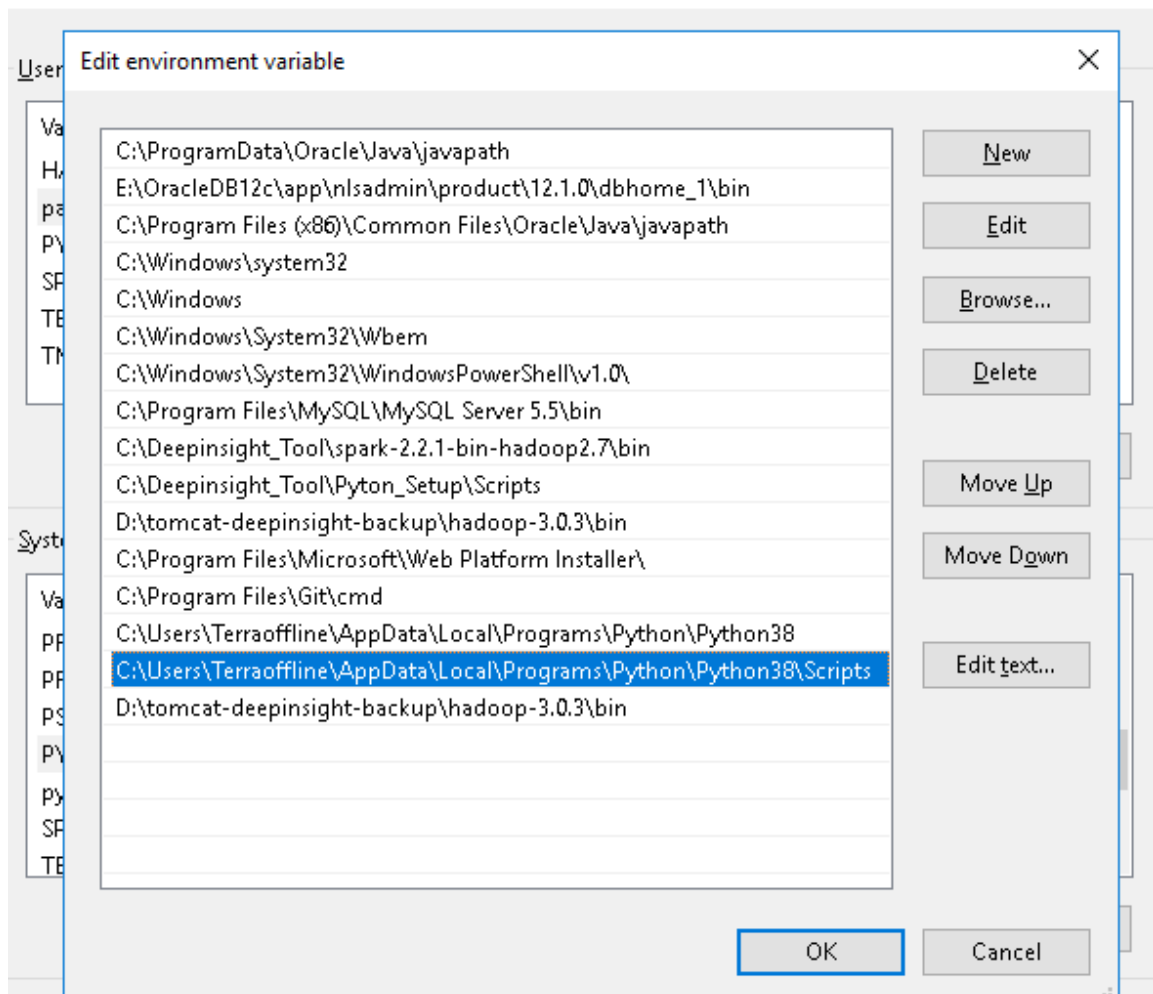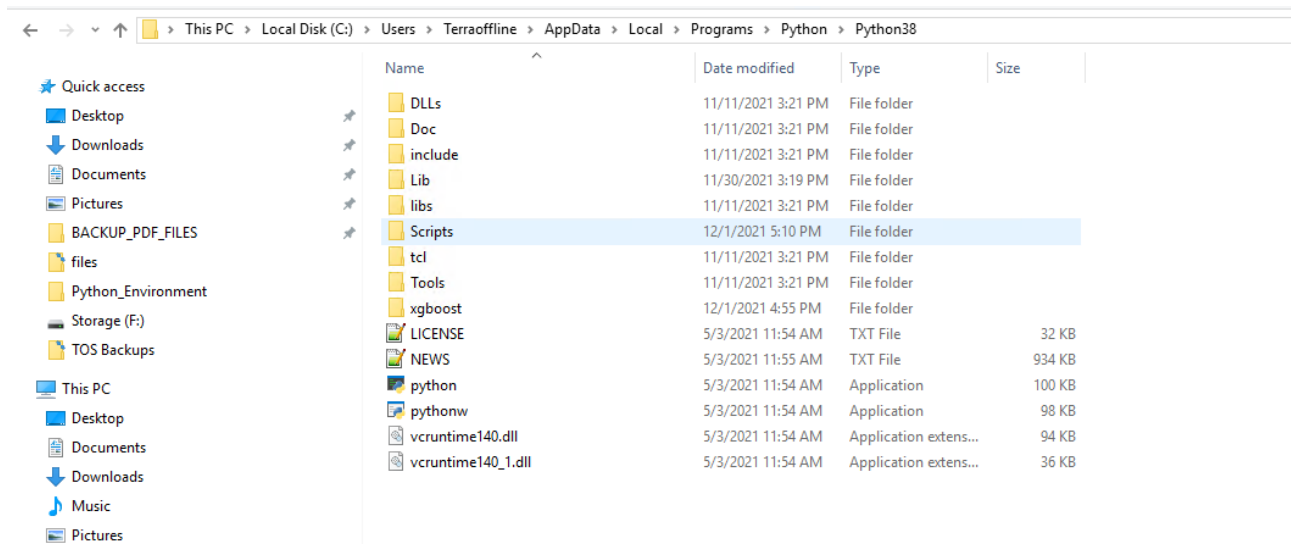
Go to --> mycomputer -right click --> properties --> advanced system settings --> environment variables --> system variables

TERA KOLLECT ML PACK DEPLOYMENT INSTRUCTIONS – WINDOWS

1) In system variables click on **path** and add the path you've installed python & Scripts folder
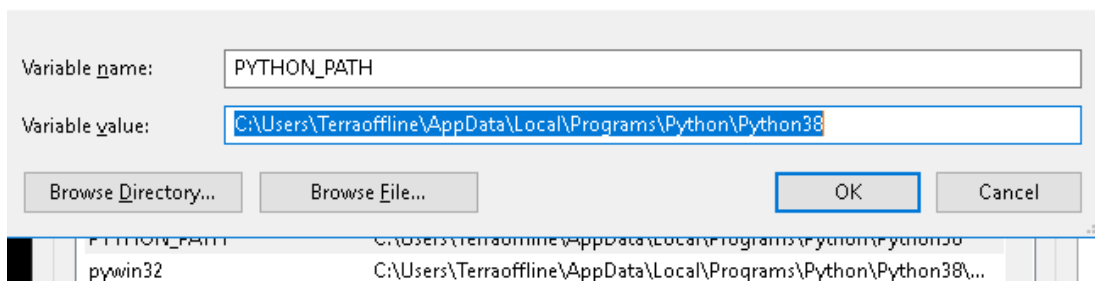
ex : C:\Users\Terraoffline\AppData\Local\Programs\Python\Python38
C:\Users\Terraoffline\AppData\Local\Programs\Python\Python38\Scripts
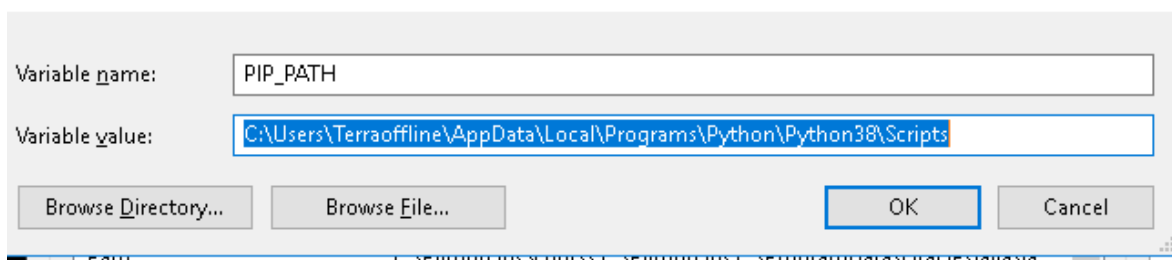
In environment variable click new and add python installed source path as python path

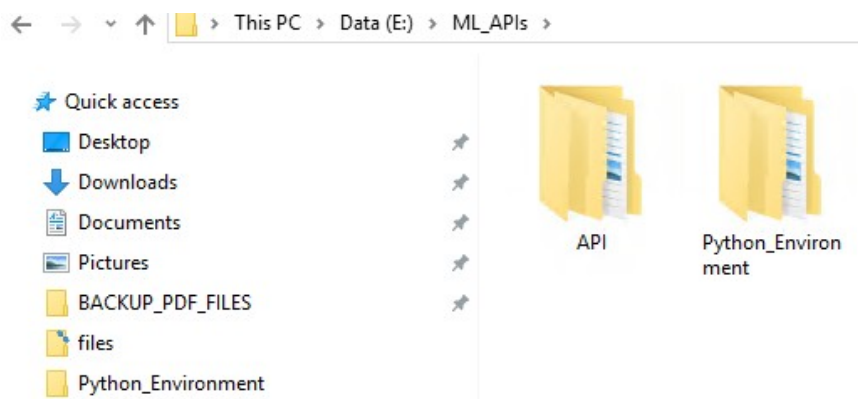add one more variable, In python source path you may find Scripts folder add that folder as PIP_PATH

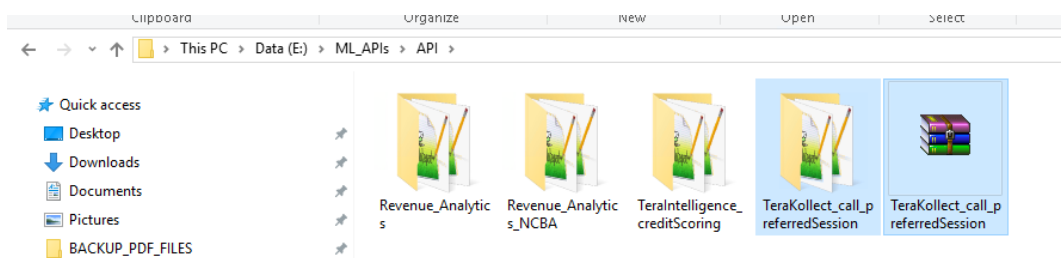Save and do check now. You may able to get the python version in cmd.

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
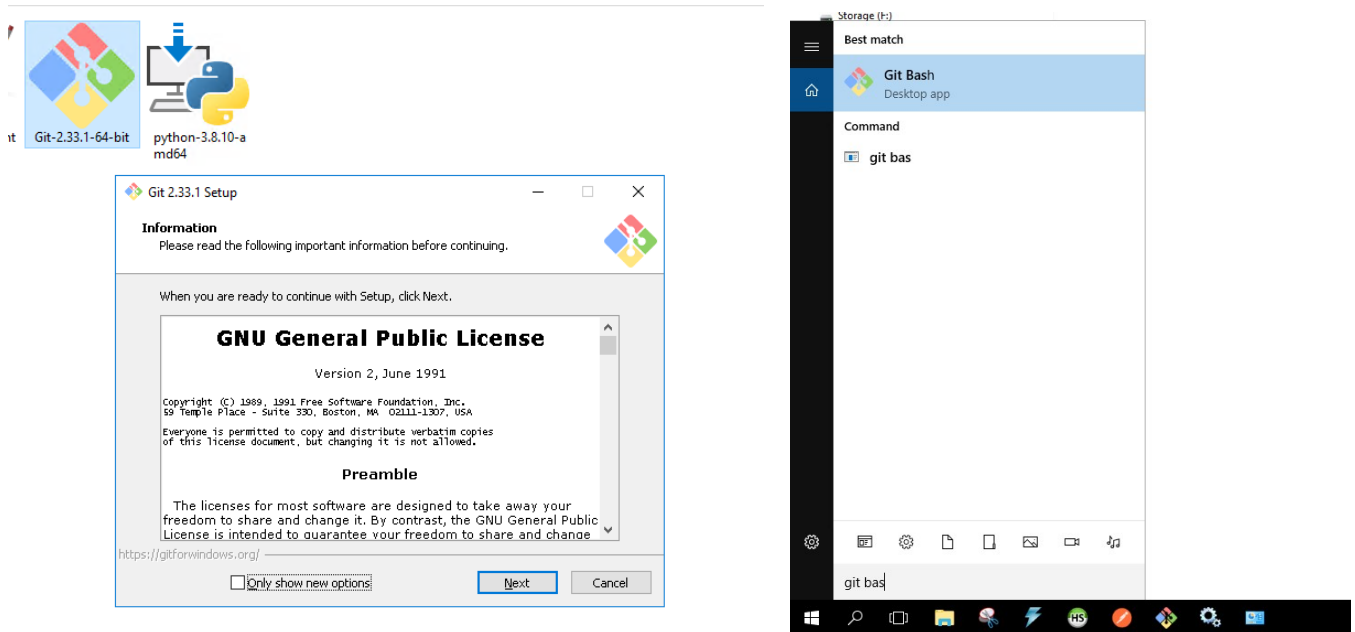

In API folder paste the source file and extract the file

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 necessory libraries to the respective enviroment. To create the virtual environmet,first to install virtual environment setup
* Go to project directory

~$ cd E:/ML _APIs/API/TeraKollect_call_preferredSession

* Move to Dependency directory

~$ cd Dependencies/dependencies

* To install the dependencies

~$ pip install * -f ./ --no-index

* To install virtual environment from local

~$ pip install Dependencies/dependencies/virtualenv-20.10.0-py2.py3-none-any.whl -f ./ --no-index

TERA KOLLECT ML PACK DEPLOYMENT INSTRUCTIONS – WINDOWS

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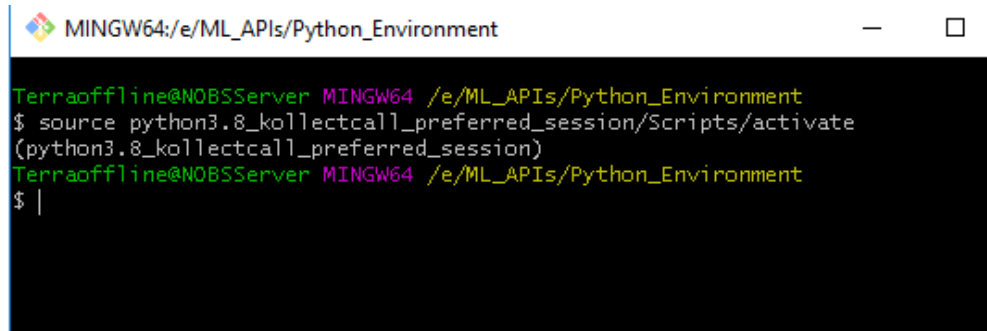
once the environment installed you may find the folder in the respective path you've mentioned

* To activate the created python environment

~$ **source python3.8_kollectcall_preferred_session/Scripts/activate**
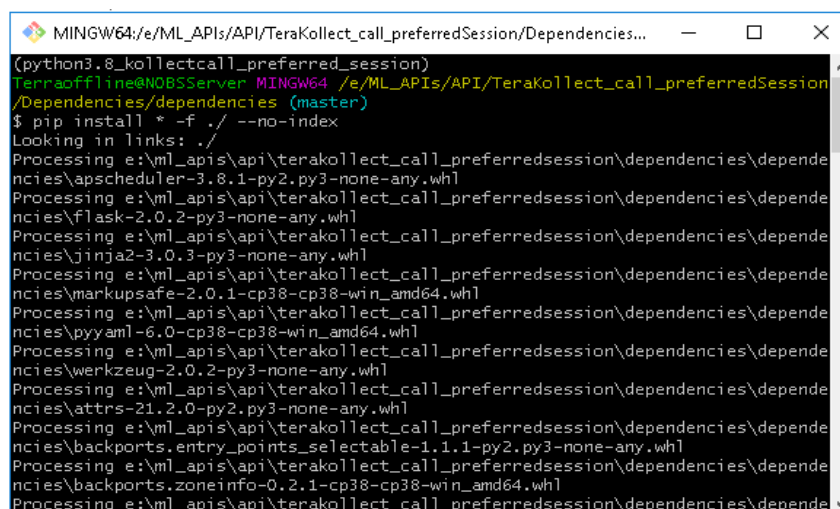
TERA KOLLECT ML PACK DEPLOYMENT INSTRUCTIONS – WINDOWS

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 necessory libraries

~$ **cd E:/ML_APIs/API/TeraKollect_call_preferredSession/Dependencies/dependencies**
~$ **pip install * -f ./ --no-index**

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
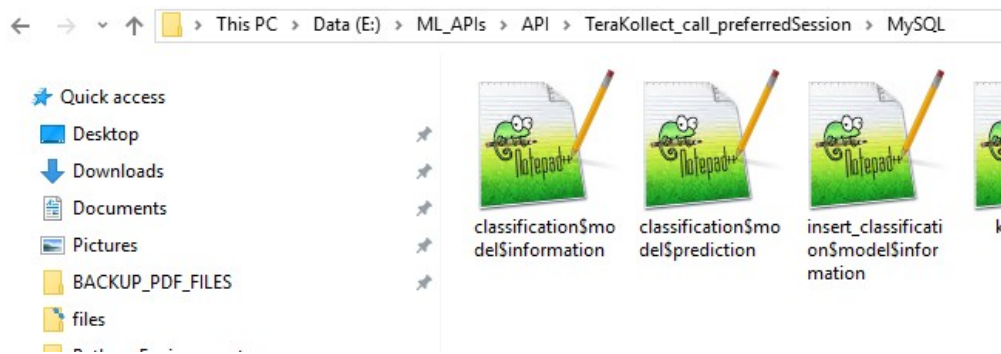
STEP 6: APPLICATION PREREQUISITES

* 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"))'**
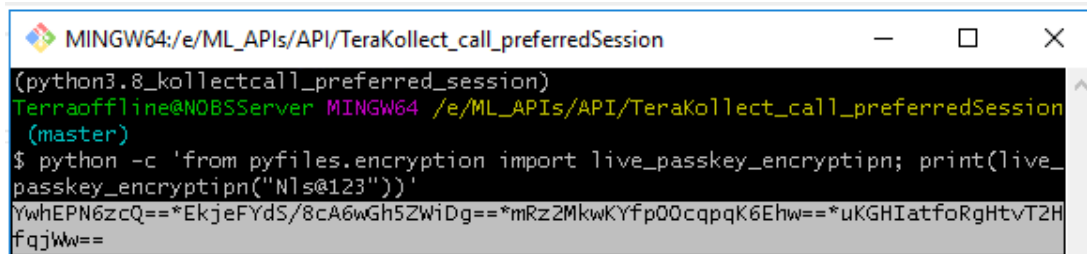
Replace password string with the respective password and enter you will get the encrypted string . Replace the encrypted string with LIVE_MYSQL_PASSWORD string

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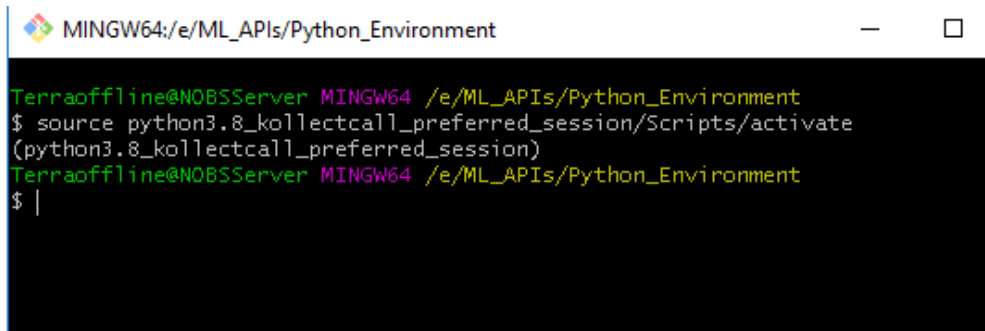export LIVE_MYSQL_PORT=3306
export LIVE_MYSQL_DB="kollect_cba"

STEP 7 : RUN OUR APPLICATION

TERA KOLLECT ML PACK DEPLOYMENT INSTRUCTIONS – WINDOWS

* 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**
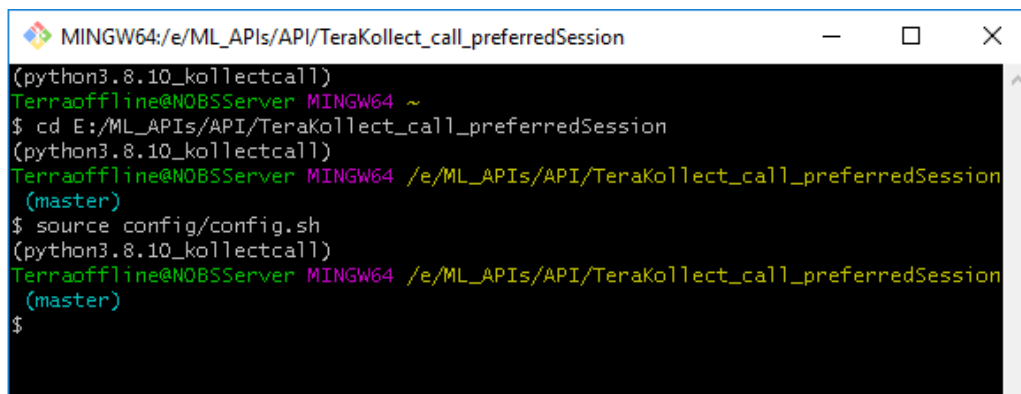~$ **source python3.8_kollectcall_preferred_session/Scripts/activate**

* 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/TeraKollect_call_preferredSession**
~$ **source config/config.sh**

* To exceute the application as background process

~$ **nohup python Kollect_preferredSession_app.py &**
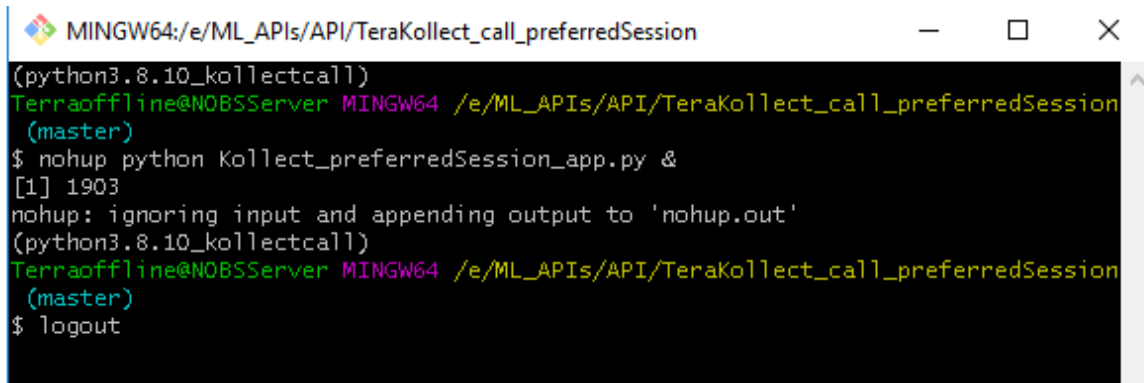
and type

~$ **logout**

and close terminal and execute the process in background

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.

* 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
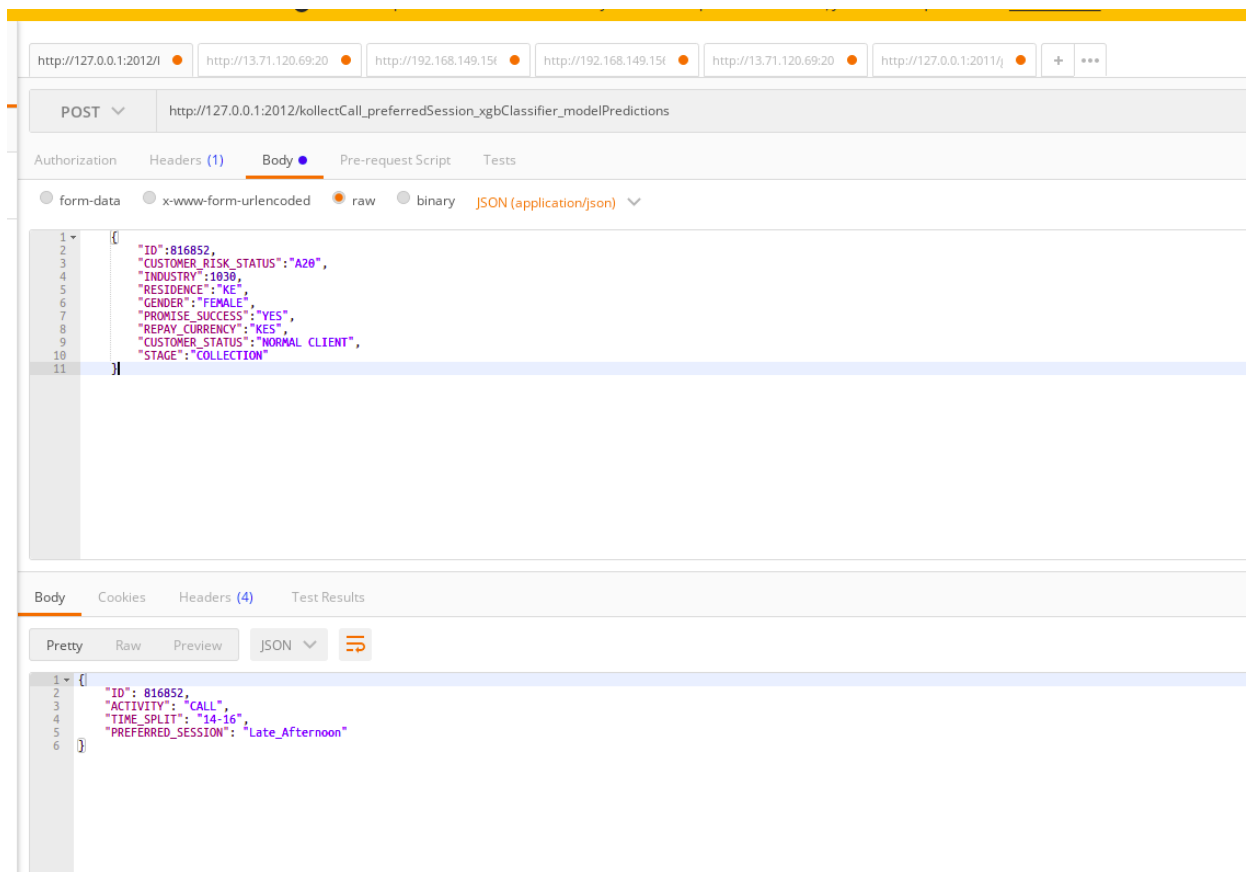INPUT: {

       "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

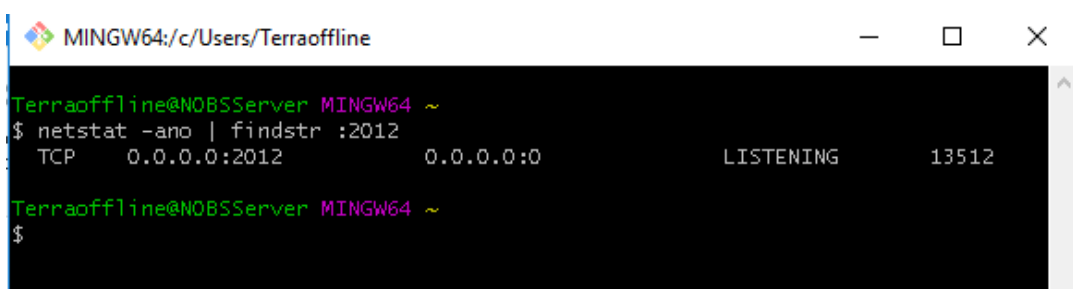**TERA KOLLECT ML PACK DEPLOYMENT INSTRUCTIONS – WINDOWS**

* 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 repective 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

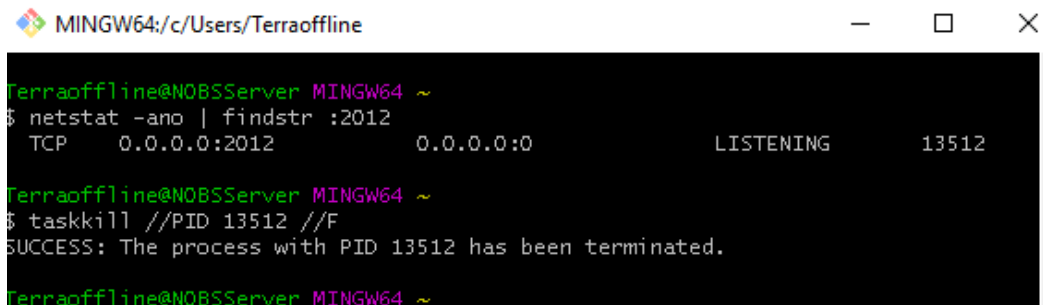~$ **netstat -ano | findstr :2012**

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
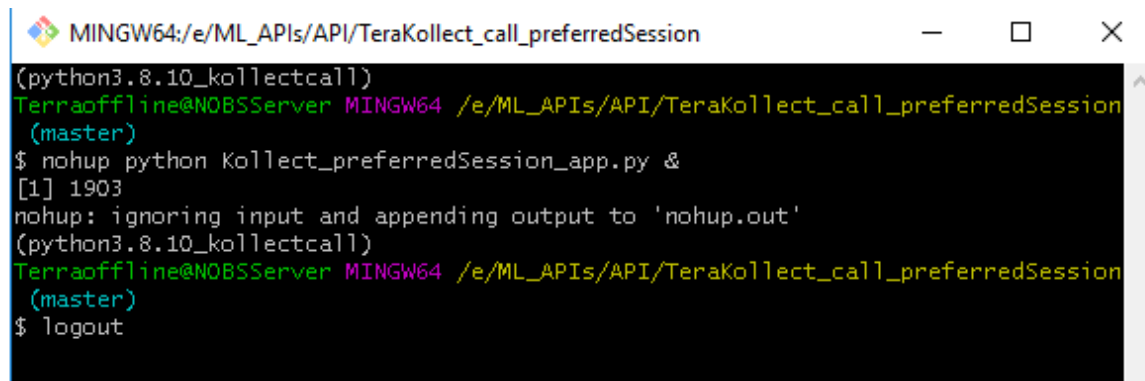
To kill the PID

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

To Run our application again

~$ **nohup python Kollect_preferredSession_app.py &**

and type

~$ **logout**

and close terminal and execute the process in background

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