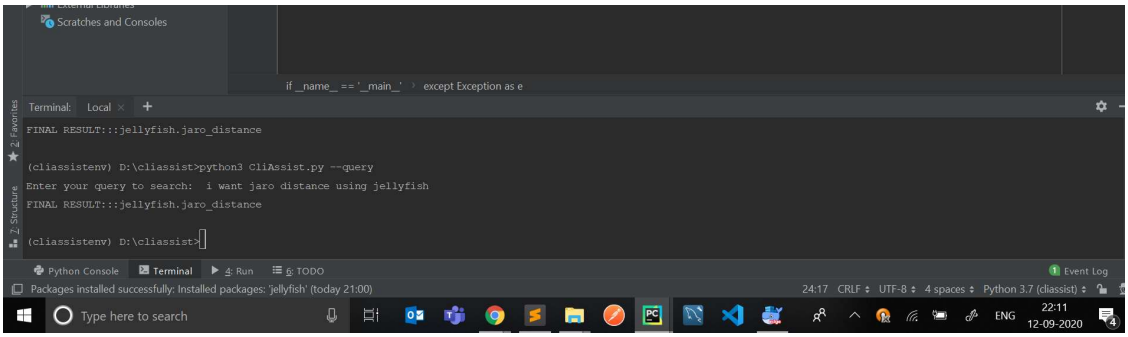
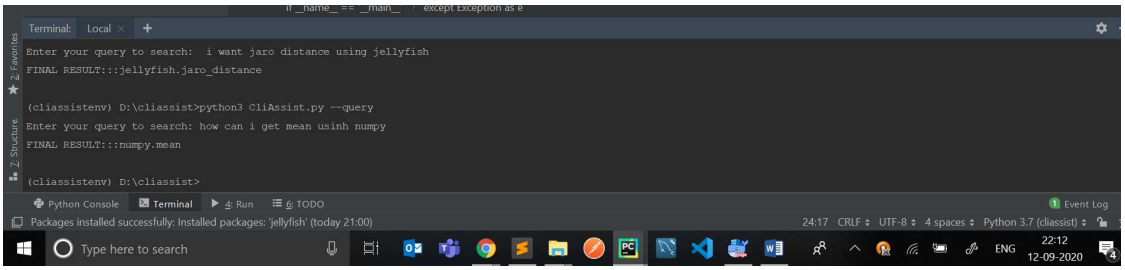
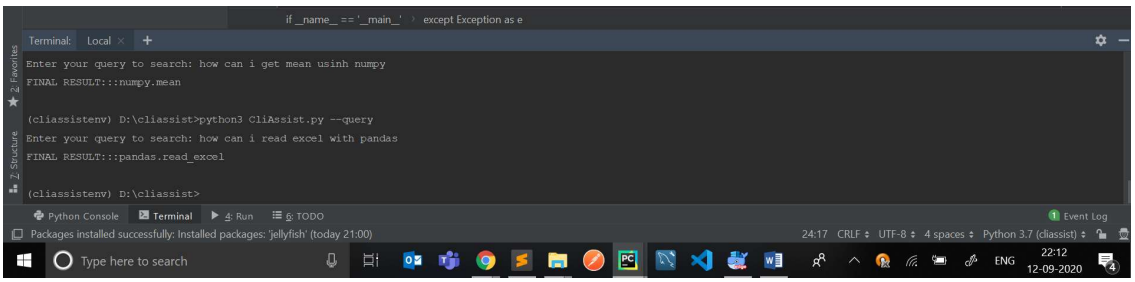
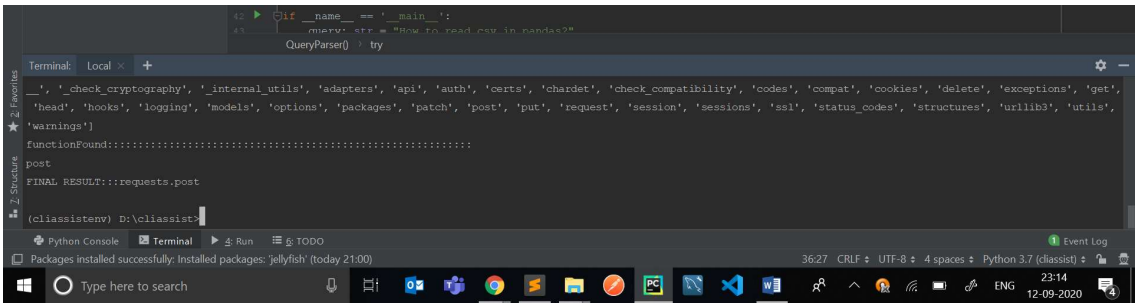
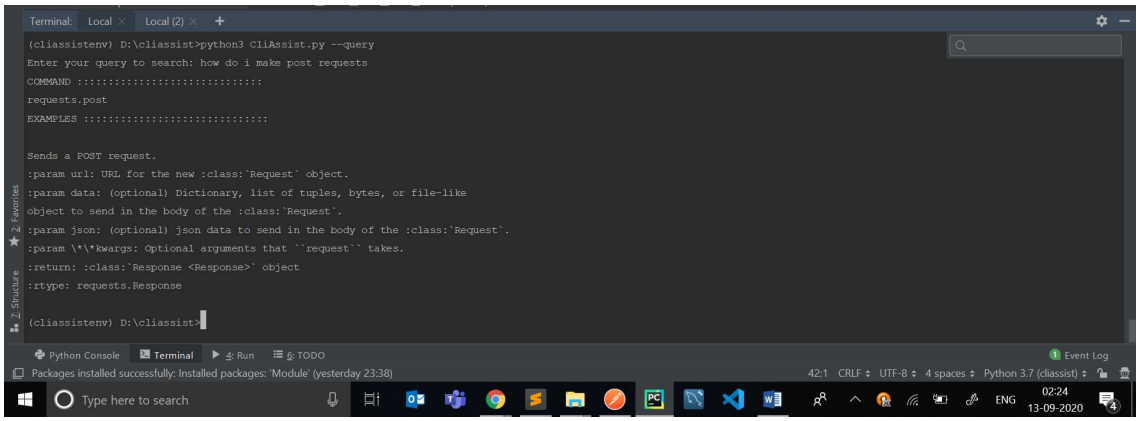
Cliassist test cases:

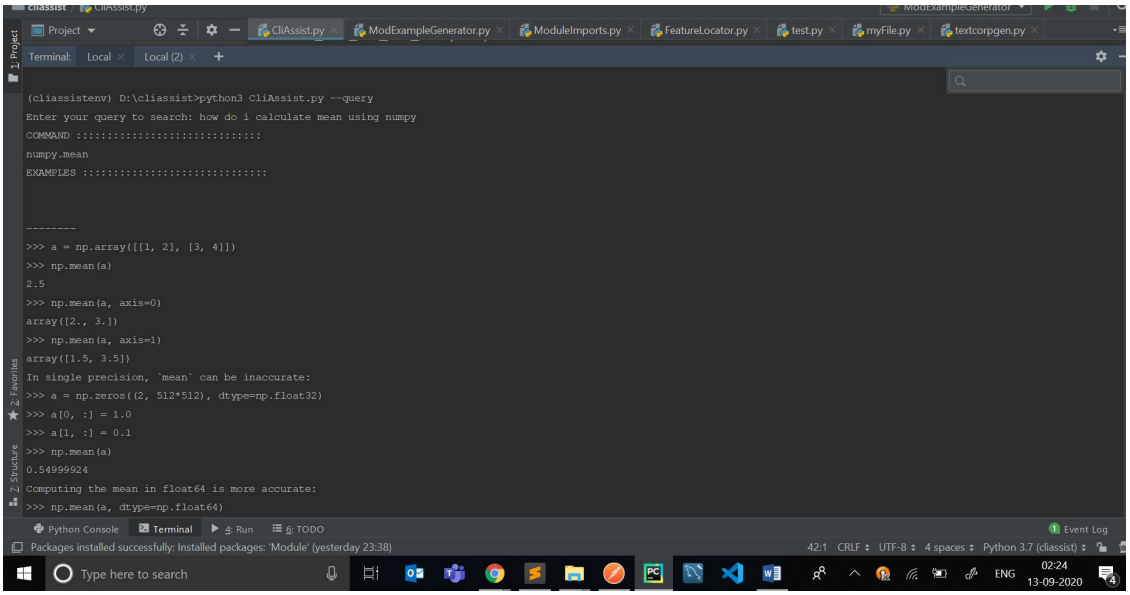




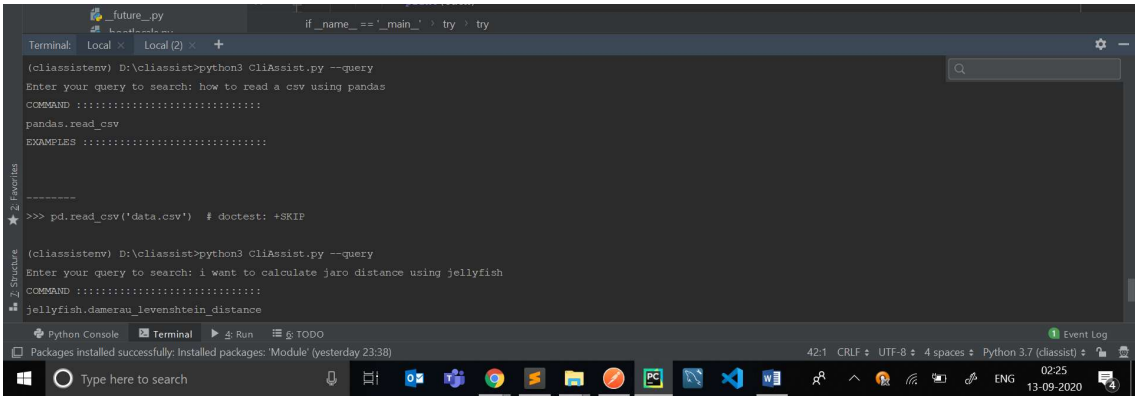



if __name__ == '__main__':
query: str = "How to read csv in pandas?"
QueryParser() › try
Terminal: Local
__', '_check_cryptography', '_internal_utils', 'adapters', 'api', 'auth', 'certs', 'chardet', 'check_compatibility', 'codes', 'compat', 'cookies', 'delete', 'exceptions', 'get', 'head', 'hooks', 'logging', 'models', 'options', 'packages', 'patch', 'post', 'put', 'request', 'session', 'sessions', 'ssl', 'status_codes', 'structures', 'urllib3', 'utils', 'warnings']
functionFound::
post
FINAL RESULT:::requests.post
(cliassistenv) D:\cliassist>
Python Console
Terminal
4: Run
6: TODO
Event Log
Packages installed successfully: Installed packages: 'jellyfish' (today 21:00)
36:27 CRLF UTF-8 4 spaces Python 3.7 (cliassist)
Type here to search
ENG
23:14
12-09-2020

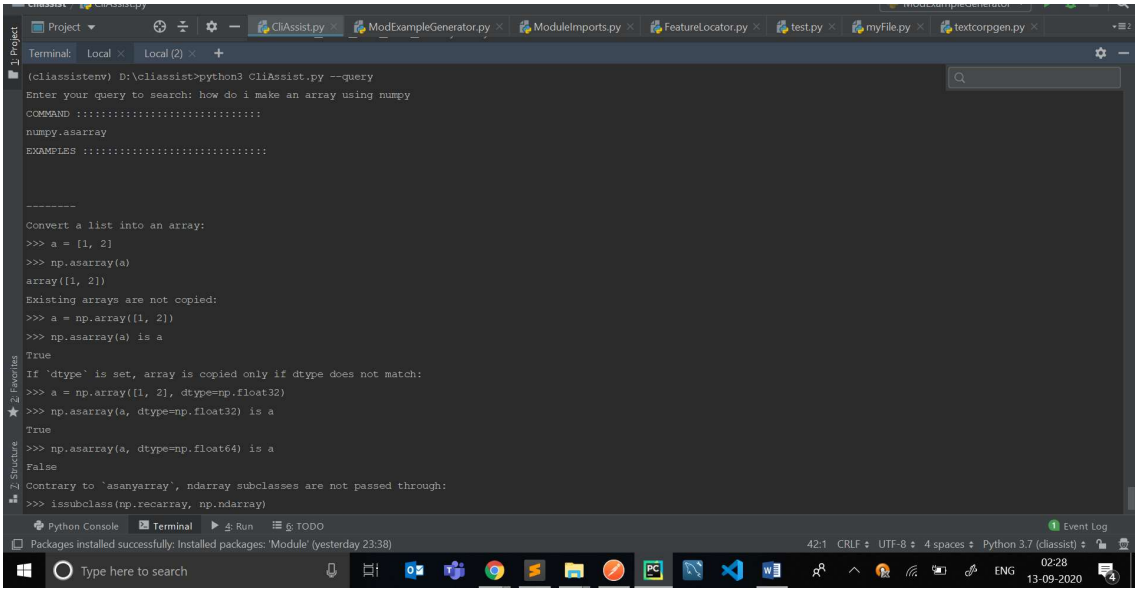

Terminal: Local Local (2)
(cliassistenv) D:\cliassist>python3 CliAssist.py --query
Enter your query to search: how do i make post requests
COMMAND ::::::::::::::::::::::::::::::::
requests.post
EXAMPLES ::::::::::::::::::::::::::::::::
Sends a POST request.
:param url: URL for the new :class:`Request` object.
:param data: (optional) Dictionary, list of tuples, bytes, or file-like
object to send in the body of the :class:`Request`.
:param json: (optional) json data to send in the body of the :class:`Request`.
:param **kwargs: Optional arguments that ``request`` takes.
:return: :class:`Response <Response>` object
:rtype: requests.Response
(cliassistenv) D:\cliassist>
Python Console
Terminal
4: Run
6: TODO
Event Log
Packages installed successfully: Installed packages: 'Module' (yesterday 23:38)
42:1 CRLF UTF-8 4 spaces Python 3.7 (cliassist)
Type here to search
ENG
02:24
13-09-2020


(cliassistenv) D:\cliassist>python3 CliAssist.py --query
Enter your query to search: how do i calculate mean using numpy
COMMAND ::::::::::::::::::::::::::::::
numpy.mean
EXAMPLES ::::::::::::::::::::::::::::::

>>> a = np.array([[1, 2], [3, 4]])
>>> np.mean(a)
2.5
>>> np.mean(a, axis=0)
array([2., 3.])
>>> np.mean(a, axis=1)
array([1.5, 3.5])
In single precision, `mean` can be inaccurate:
>>> a = np.zeros((2, 512*512), dtype=np.float32)
>>> a[0, :] = 1.0
>>> a[1, :] = 0.1
>>> np.mean(a)
0.54999924
Computing the mean in float64 is more accurate:
>>> np.mean(a, dtype=np.float64)


(cliassistenv) D:\cliassist>python3 CliAssist.py --query
Enter your query to search: how to read a csv using pandas
COMMAND ::::::::::::::::::::::::::::::
pandas.read_csv
EXAMPLES ::::::::::::::::::::::::::::::

>>> pd.read_csv('data.csv') # doctest: +SKIP
(cliassistenv) D:\cliassist>python3 CliAssist.py --query
Enter your query to search: i want to calculate jaro distance using jellyfish
COMMAND ::::::::::::::::::::::::::::::
jellyfish.damerau_levenshtein_distance


CliAssist.py
ModExampleGenerator.py
ModuleImports.py
FeatureLocator.py
test.py
myFile.py
textcorpgen.py
Terminal: Local Local (2)
(cliassistenv) D:\cliassist>python3 CliAssist.py --query
Enter your query to search: how do i make an array using numpy
COMMAND ::::::::::::::::::::::::::::::
numpy.asarray
EXAMPLES ::::::::::::::::::::::::::::::

Convert a list into an array:
>>> a = [1, 2]
>>> np.asarray(a)
array([1, 2])
Existing arrays are not copied:
>>> a = np.array([1, 2])
>>> np.asarray(a) is a
True
If `dtype` is set, array is copied only if dtype does not match:
>>> a = np.array([1, 2], dtype=np.float32)
>>> np.asarray(a, dtype=np.float32) is a
True
>>> np.asarray(a, dtype=np.float64) is a
False
Contrary to `asanyarray`, ndarray subclasses are not passed through:
>>> issubclass(np.recarray, np.ndarray)
Python Console
Terminal
Run
TODO
Event Log
Packages installed successfully: Installed packages: 'Module' (yesterday 23:38)
42:1 CRLF UTF-8 4 spaces Python 3.7 (cliassist)
Type here to search
ENG
02:28
13-09-2020