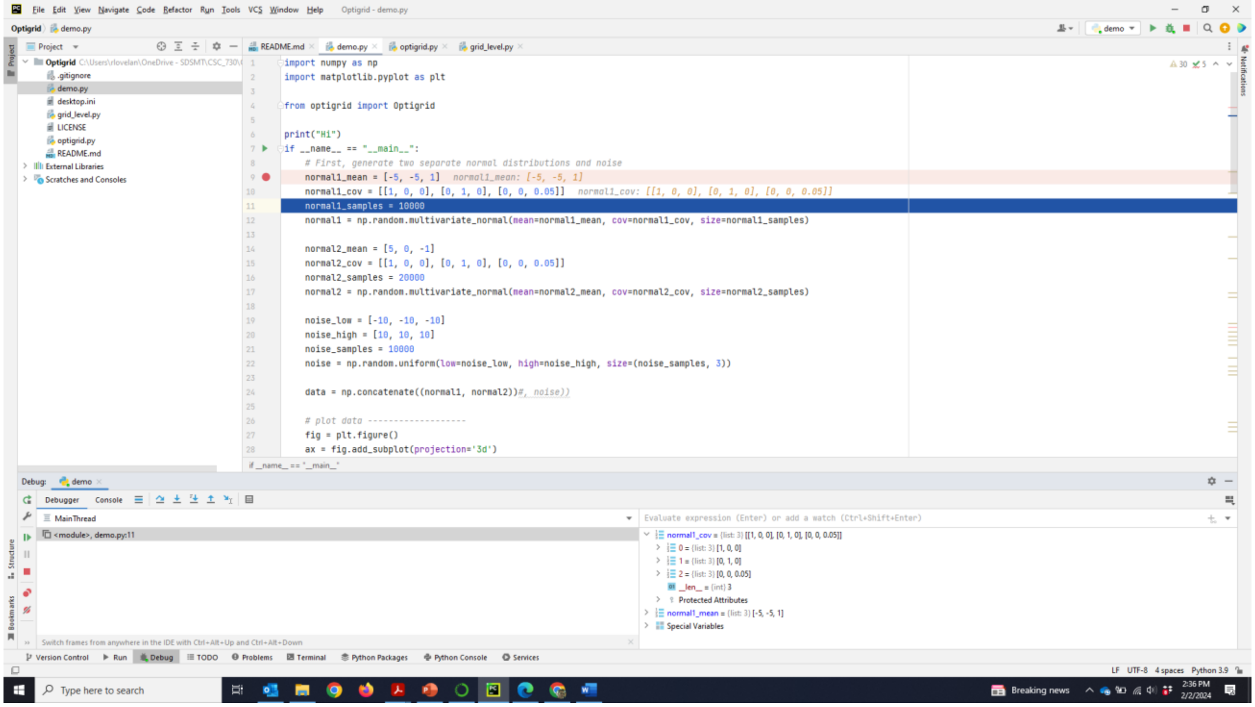
Assignment 3

OptiGrid – Cursing Dimensionality


File Edit View Navigate Code Refactor Run Tools VCS Window Help Optigrid - demo.py
Optigrid demo.py
Project
Optigrid C:\Users\rlovelan\OneDrive - SDSMT\CSC_730\
.gitignore
demo.py
desktop.ini
grid_level.py
LICENSE
optigrid.py
README.md
External Libraries
Scratches and Consoles
README.md demo.py optigrid.py grid_level.py
Debug: demo
Debugger Console
MainThread
<module>, demo.py:11
Switch frames from anywhere in the IDE with Ctrl+Alt+Up and Ctrl+Alt+Down
Version Control Run Debug TODO Problems Terminal Python Packages Python Console Services
LF UTF-8 4 spaces Python 3.9
Type here to search
Breaking news
2:36 PM
2/2/2024

```python
import numpy as np
import matplotlib.pyplot as plt

from optigrid import Optigrid

print("Hi")
if __name__ == "__main__":
    # First, generate two separate normal distributions and noise
    normal1_mean = [-5, -5, 1]
    normal1_cov = [[1, 0, 0], [0, 1, 0], [0, 0, 0.05]]
    normal1_samples = 10000
    normal1 = np.random.multivariate_normal(mean=normal1_mean, cov=normal1_cov, size=normal1_samples)

    normal2_mean = [5, 0, -1]
    normal2_cov = [[1, 0, 0], [0, 1, 0], [0, 0, 0.05]]
    normal2_samples = 20000
    normal2 = np.random.multivariate_normal(mean=normal2_mean, cov=normal2_cov, size=normal2_samples)

    noise_low = [-10, -10, -10]
    noise_high = [10, 10, 10]
    noise_samples = 10000
    noise = np.random.uniform(low=noise_low, high=noise_high, size=(noise_samples, 3))

    data = np.concatenate((normal1, normal2))#, noise))

    # plot data ------------------
    fig = plt.figure()
    ax = fig.add_subplot(projection='3d')
```

```
Evaluate expression (Enter) or add a watch (Ctrl+Shift+Enter)
normal1_cov = {list: 3} [[1, 0, 0], [0, 1, 0], [0, 0, 0.05]]
    0 = {list: 3} [1, 0, 0]
    1 = {list: 3} [0, 1, 0]
    2 = {list: 3} [0, 0, 0.05]
    __len__ = {int} 3
    Protected Attributes
normal1_mean = {list: 3} [-5, -5, 1]
Special Variables
```

In this assignment you will examine the mihailescum's OptiGrid implementation, at https://github.com/mihailescum/Optigrid, in order to fully understand the algorithm.

For this assignment you will:
1) You will write, in your own words, a description of how the OptiGrid clustering algorithm works.

2) You will then download the code, and work through it, writing an extremely detailed description of what it's doing. Specifically, you will write a paragraph explaining each function.

3) You will then modify the code demo as necessary to create a 2-D example, and create a figure showing data points and cutting planes the algorithm arrives at. (You may reduce use a smaller number of data points than the 30,000 in their example.)

You will turn in both the code and a short report, discussing what you've implemented, how well it worked, what you've learned, etc.

Remember, you must understand what you turn in – you may be asked to explain it to me and/or the class.