**DBScans Algorithm**

**Melvin Ismanto**

**9018720198**

**Algorithm Summary**

My program uses the concept of DBScans algorithm to cluster the data from the database (text input files). The provided data will only be in 2-dimension, so it should not be too difficult to do the clustering process. The main step of the algorithm are as follows:

1. Scan the dataset from the input file.
2. Create a kd-tree of the dataset, which will be used to find the neighbor points within some specific range.
3. Initialize cluster-id to 0.
4. Choose an UNCHECKED seed (i.e. seed point which hasn’t been traversed yet) from the list of points.
5. Find all the neighbor points of selected seed within *eps* range.
6. If the number of neighbors is below the minimum required points, set classification of current seed to OUTLIER, then go back to step 4.
7. Otherwise, set classification of the seed points and its neighbors to current cluster-id.
8. For each seed’s neighbor:
   1. Find all sub\_neighbor of the current seed’s neighbor.
   2. If the number of sub\_neighbor sufficient (equal or more than minimum required points), then for each sub\_neighbor:
      1. If the classification of current sub\_neighbor is either UNCHECKED or OUTLIER, set its classification as current cluster-id.
      2. If the classification of current sub\_neighbor is UNCHECKED, then also add this sub\_neighbor point to the seeds neighbor list (thus increasing the main loop count).
9. Increment the cluster-id, then go back to step 4 until all point is no longer UNCHECKED.
10. Let *m* be the number of cluster found, and *n* be the max number of cluster wanted. If m>n, then trim *m-n* cluster with the lowest number of points (setting it as OUTLIER).

For more in-depth (line-by-line) explanation of the algorithm, please refer to the added comments provided on the dbscans.py file.

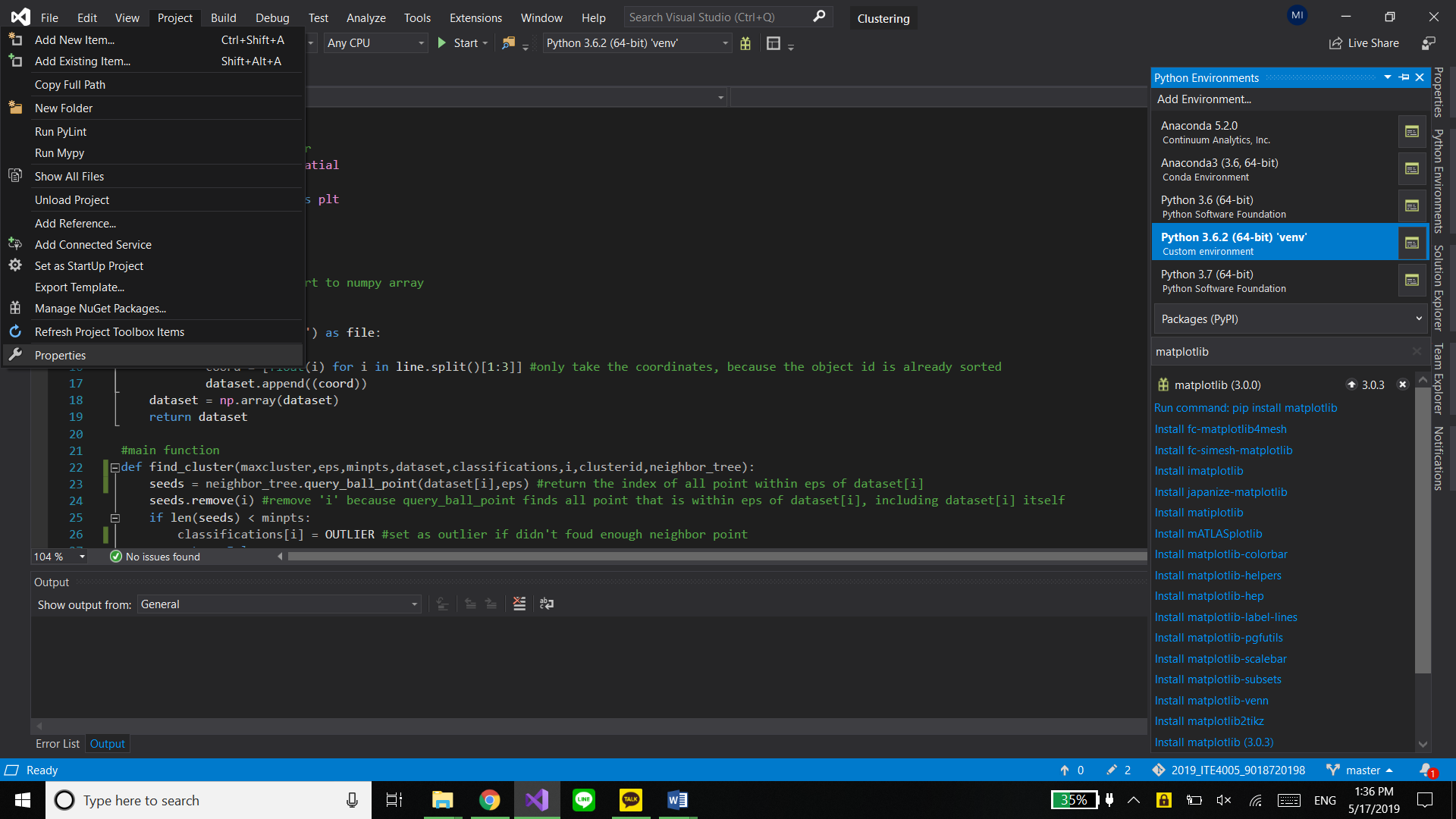
/////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////

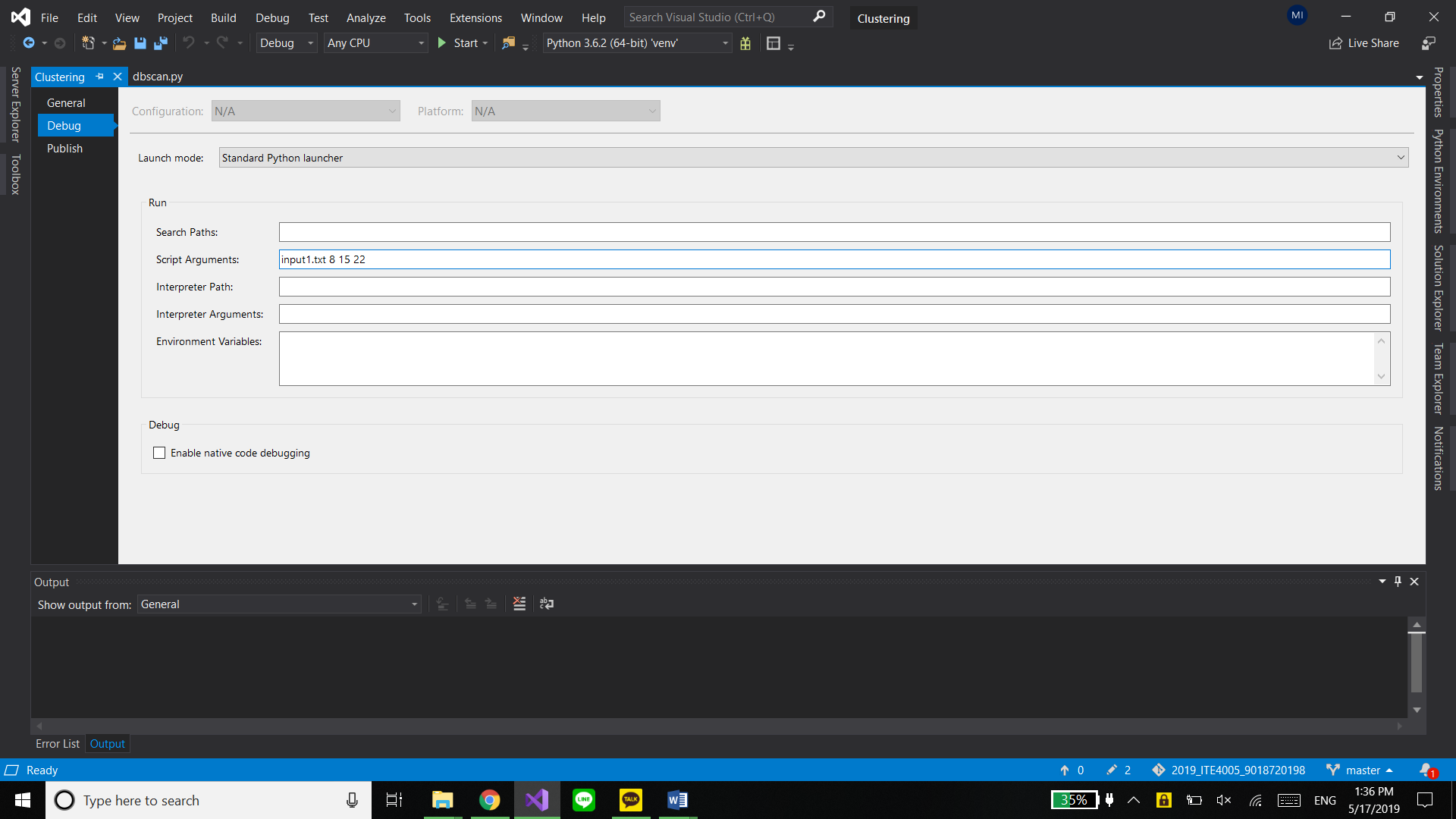
I uses the Microsoft Visual Studio Community **2019** (Version 16.0.1) IDE to write my code, using the Python version 3.6.2 language. So you can also use that to make compiling easier. When running the file, you should run the .sln file (project solution), not the .py file itself.

Download it here:

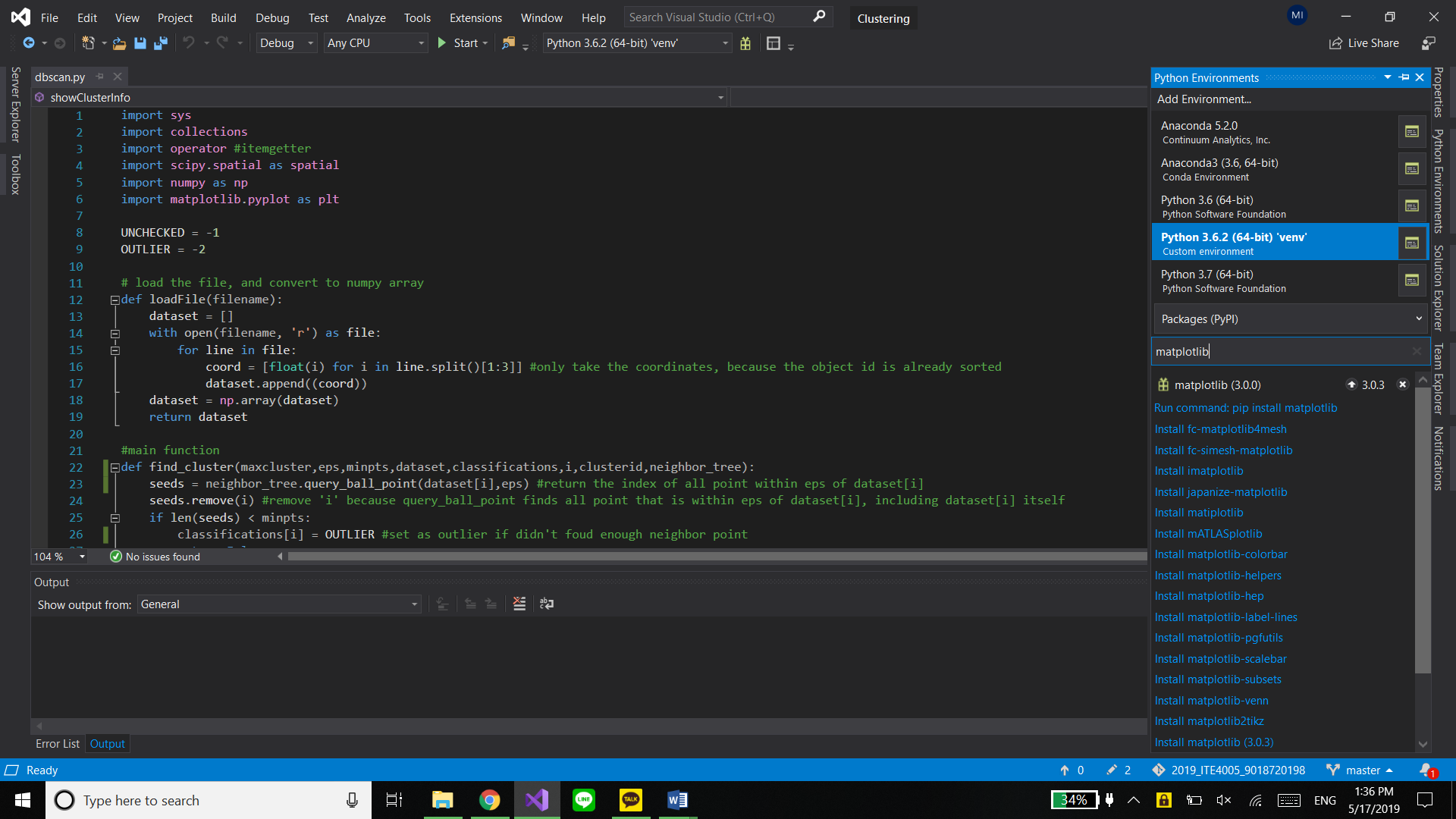
<https://visualstudio.microsoft.com/thank-you-downloading-visual-studio/?sku=Community&rel=16>

To change the argument parameter inside Visual Studio itself:





To add any package to the python environment, inside Visual Studio itself:



/////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////

Then basically just press F5 to start/debug the python program, or use it from the console command

I code this on a Windows 10 Pro 1809 (64 bit), with Intel i7-7500 CPU, and 16GB of RAM on my Laptop. It takes about 1 second for my program to analyze the provided text input file, and output the result to a text file.