## For KNN graph:

1. NN score is the indegree of a data point Xi. It can be said to be equal to number of Xj such that Xi belongs to the neighbourhood of Xj.
2. Xi belongs to neighbourhood of Xj when there is edge Xj to Xi. That means in the KNN graph Xi is a neighbour of Xj.
3. MNN score is more strict, it is the number of Xi such that Xi is part of neighbourhood of Xj and at the same time Xj is part of the neighbourhood of Xi.

Important changes:

* Don’t use compute MNN\_scores for all as it makes the complexity high for no reason
* Error in line: train sample = train sample ∪ X[index]
* Until loop is wrong in logic: (NN − score(X) = 0) ∨ (| train sample |≤ k); the second condition should be |X| <= k, as per the text in research paper -> highlighted for same.
* Another fault in algorithm: when we delete index and its mutual neighbors, we also need to delete all the mutual neighbor’s mutual neighbors too and so on till no longer bcoz if we don’t, then we need to do exception handling of index not present while drop of new ele if one of its mutual neighbor is this one that got deleted as mutual neighbor of someone else. (current solution I used is simple try and except block to ignore the indices that give key\_errors since they are already deleted)
* There isn’t enough clearity about how nn\_score are calculated for dynamic and static, basically the difference between them, hence I assume, all vertices (even deleted) are considered in static and for dynamic all vertices (that are not deleted) in the new graph are considered.
* I doubt the computational complexity is O(N\*N), it appears to be more
* Values of t, m, s for nets thing were not provided in the paper