Lachlan Sinclair

1/26/20

CS_419_400

Week 3 assignment 2: 10-fold partitioning of a dataset

Reflections:

A. To combine the nodes labelled “Model x” and “Train/Test folds” we would simply need to edit the Train/Test folds node and delete the Model x node. The editing would entail changing the if statement in the Train/Test folds node, rather than checking if the fold equals the Model ID, we would code it to check the specific integer value that was previously assigned to the Model ID in the Model x node. Using Model 1 as an example the new if state would check if the Fold is equal to one. A partition node sets up a partition, but in this scenario, we are using a type node because the partitions have already been generated via the K-fold method. This type node is simply letting the flow know that the partition field created in the previous derive node is intended to be used as a partition. Therefor using a partition node here would not work.

B. I am assuming one stream per training-test split means that essentially each of the 10 training-test splits would have its own entire stream, I’m not sure about this assumption but I couldn’t think of a better interpretation of the question. Implementing it in this manner would be easily achievable, each model x node would be connected directly to the fold node. This would have to be done for all 10 splits, each having its own stream leading up to and including the Fold node. One issue I think may occur is having different Fold nodes for each stream, since unlike the partition node we can’t specify the same random assignment of instances but rather rely on a cache. Since each stream would have its own Fold node caching would not work and each of the ten streams would assign different Fold values. If there is a way to get around this fold generation issue or it is acceptable to ignore it the 10 splits paradigm naturally flows into using a single stream per split.