Massive Online Analysis (MOA) Lab Report

CHEN Hang

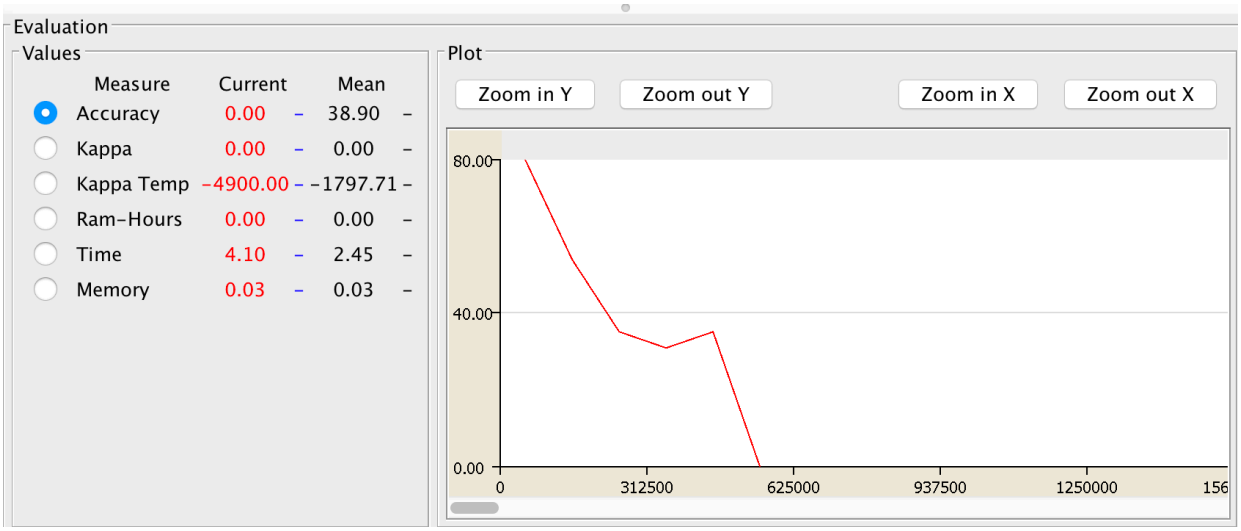
Classifiers selected for the experiments

1. MajorityClass
2. NaiveBayes
3. Perceptron
4. HoeffdingAdaptiveTree
5. Lazy KNN

The results of the experiments

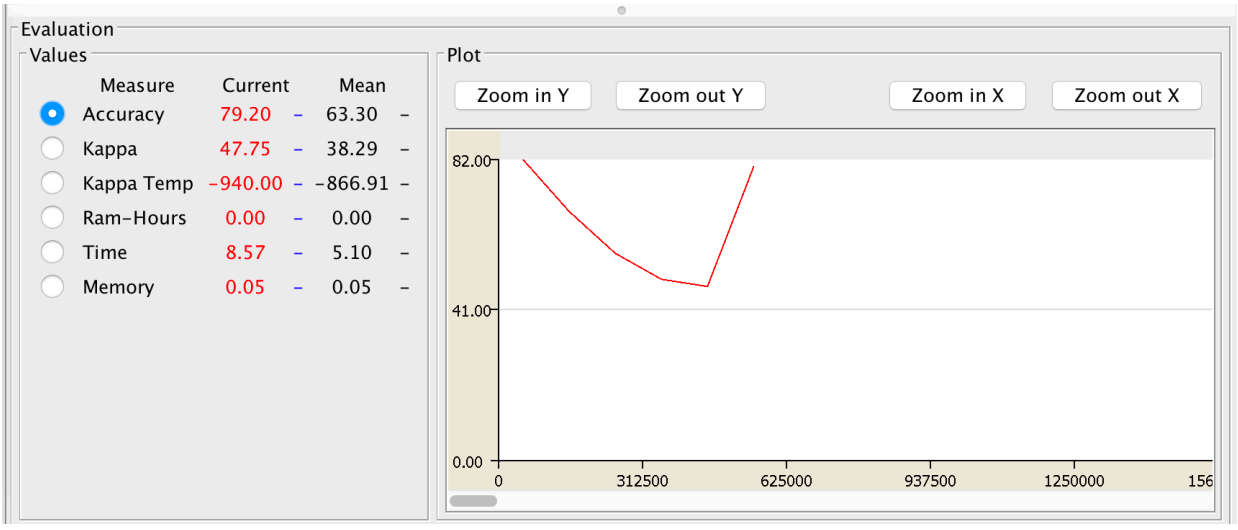
1. MajorityClassifier

From the graph below, we have the average accuracy = 38.90% and run time = 2.45s



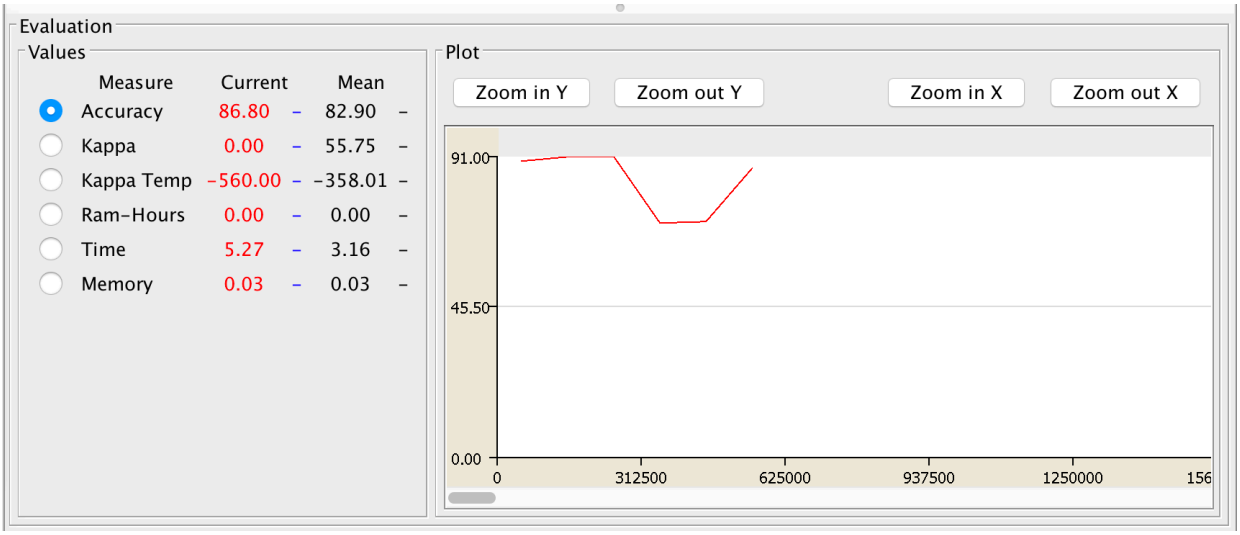
2. NaiveBayes

From the graph below, we have the average accuracy = 63.30% and run time = 5.10s



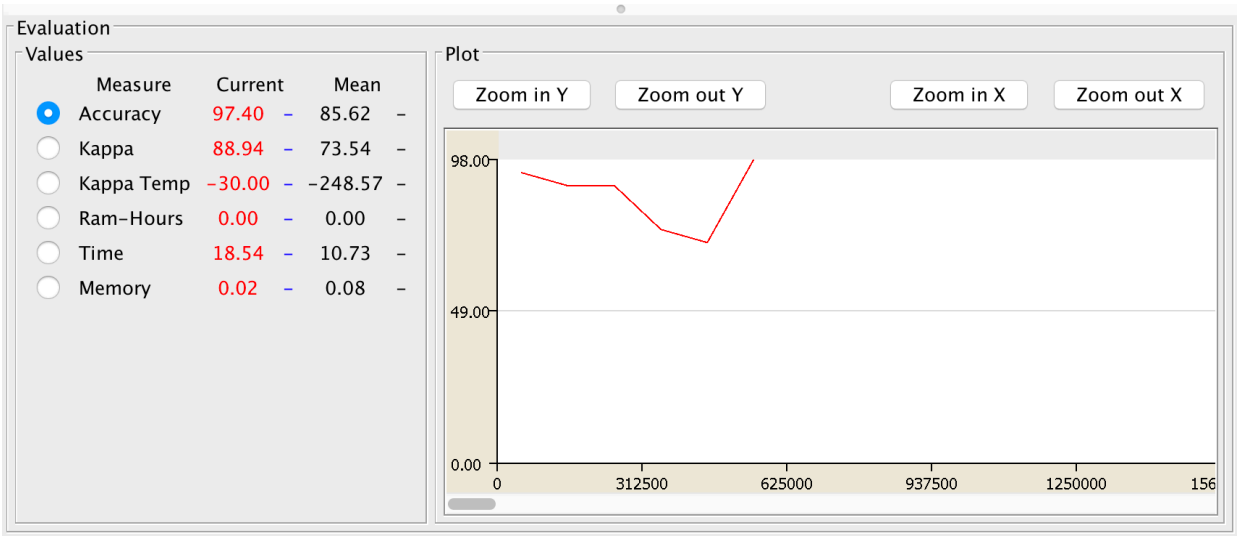
3. Perceptron

From the graph below, we have the average accuracy = 82.90% and run time = 3.16s



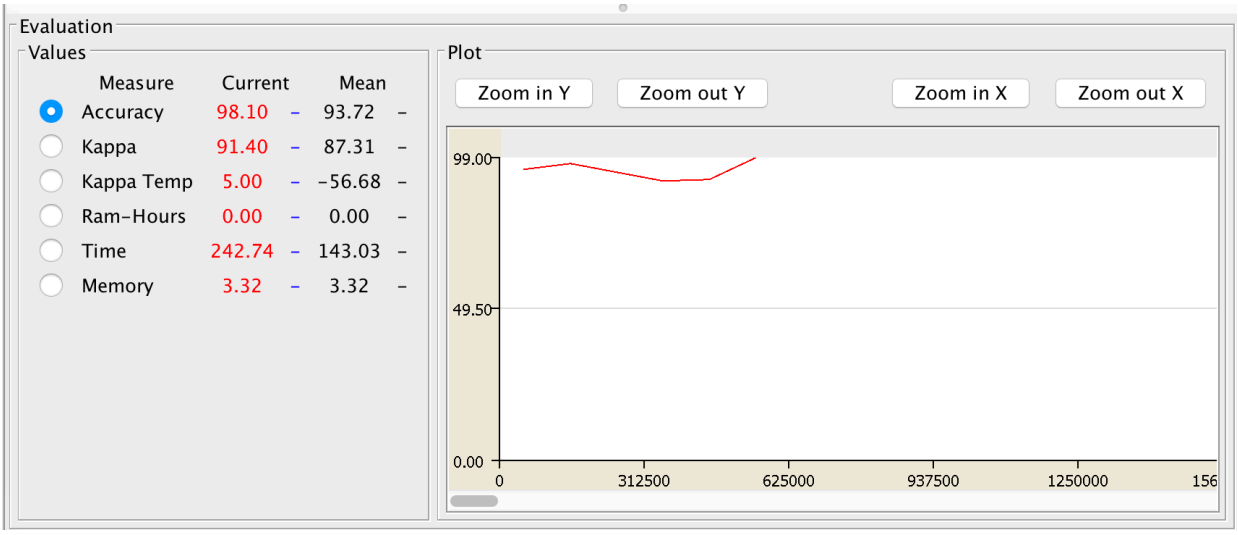
4. HoeffdingAdaptiveTree

From the graph below, we have the average accuracy = 85.62% and run time = 10.73s

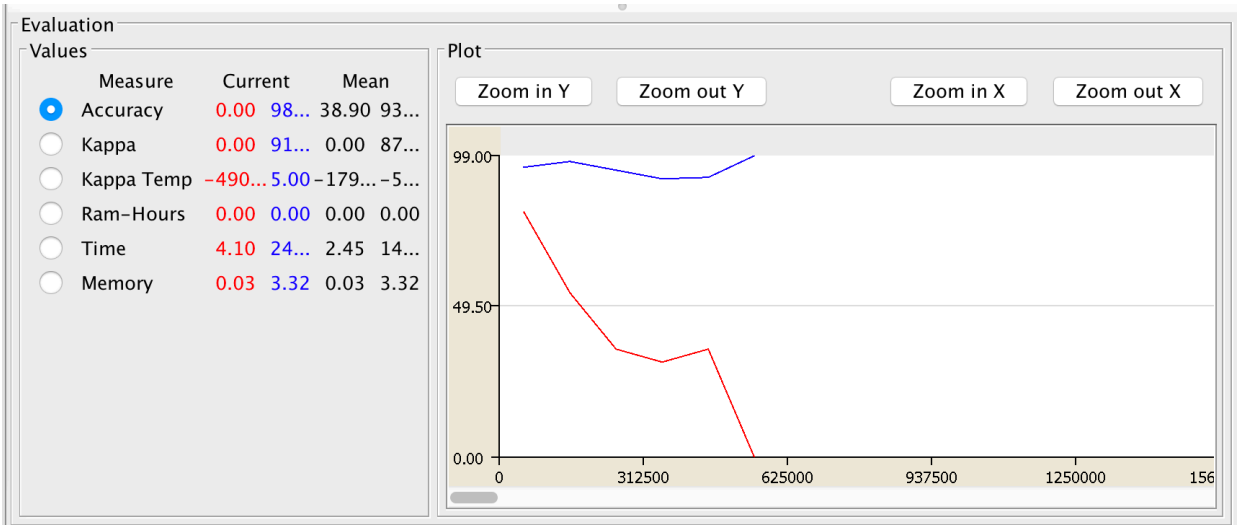


5. Lazy KNN

From the graph below, we have the average accuracy = 93.72% and run time = 143.03s

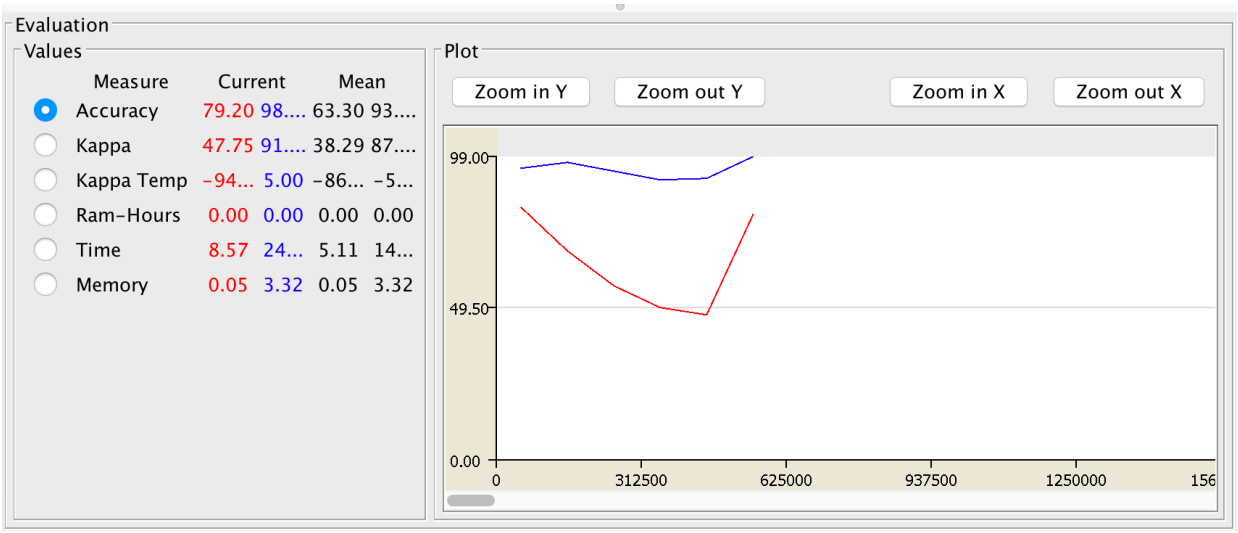


Discussion about the results

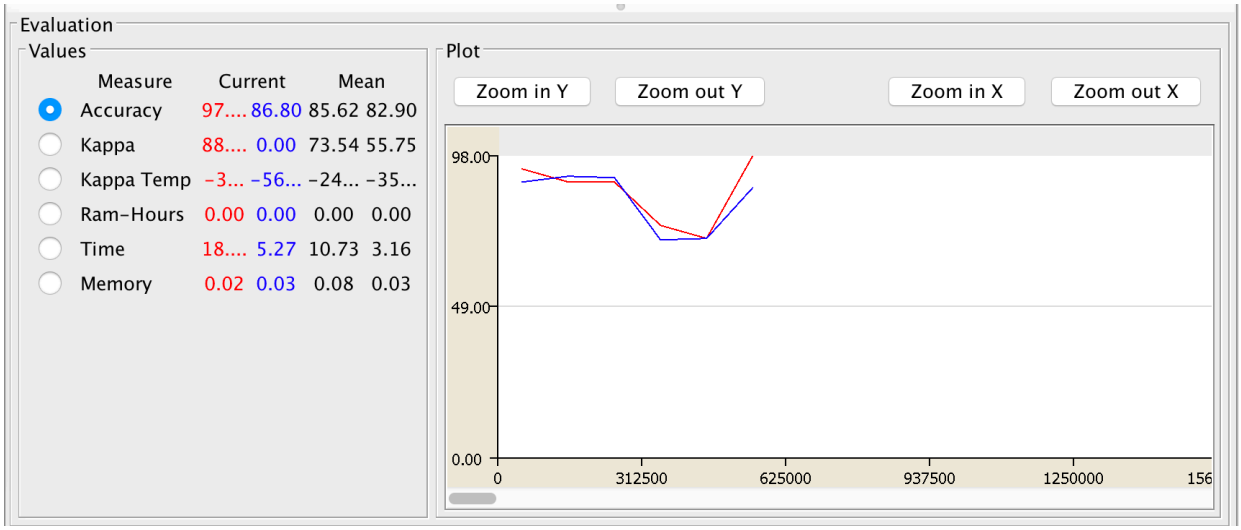


From these 5 experiments, we get the worst result is the Majority Classifier, it has only 38.90% average accuracy. In my opinion, its because the datasets of Covertype is too complex for the Majority Classifier, so it causes the bad result.

And the classifier with the highest average accuracy is the Lazy KNN, it has the 93.72% average accuracy. But its shortcoming is that the running time of Lazy KNN is longer than the others. The running time of these other 4 classifier are all less than 11s, the Lazy KNN has about 140s of running time.



Then, for the NaiveBayes Classifier, it has 63.30% average accuracy, its not as well as expected. Because the NaiveBayesClass requires the dataset with a high independence, but many attributes of the CoverType dataset are related, so the result is not very good.



The HoeffdingAdaptiveTree and the Perceptron have the similar result. The average accuracy of HoeffdingAdaptiveTree is 85.62% and the Perceptron is 82.90%. But the running time of HoeffdingAdaptiveTree is a little longer.

Recommeded classifier

From the results of the 5 experiments, if the dataset is not too big that it need to much time to run it, just like this CoverType dataset, I recommend the Lazy KNN because of its high accuracy. But if the dataset is very large, I will recommend the HoeffdingAdaptiveTree, because it has a high accuracy and an acceptable running time.