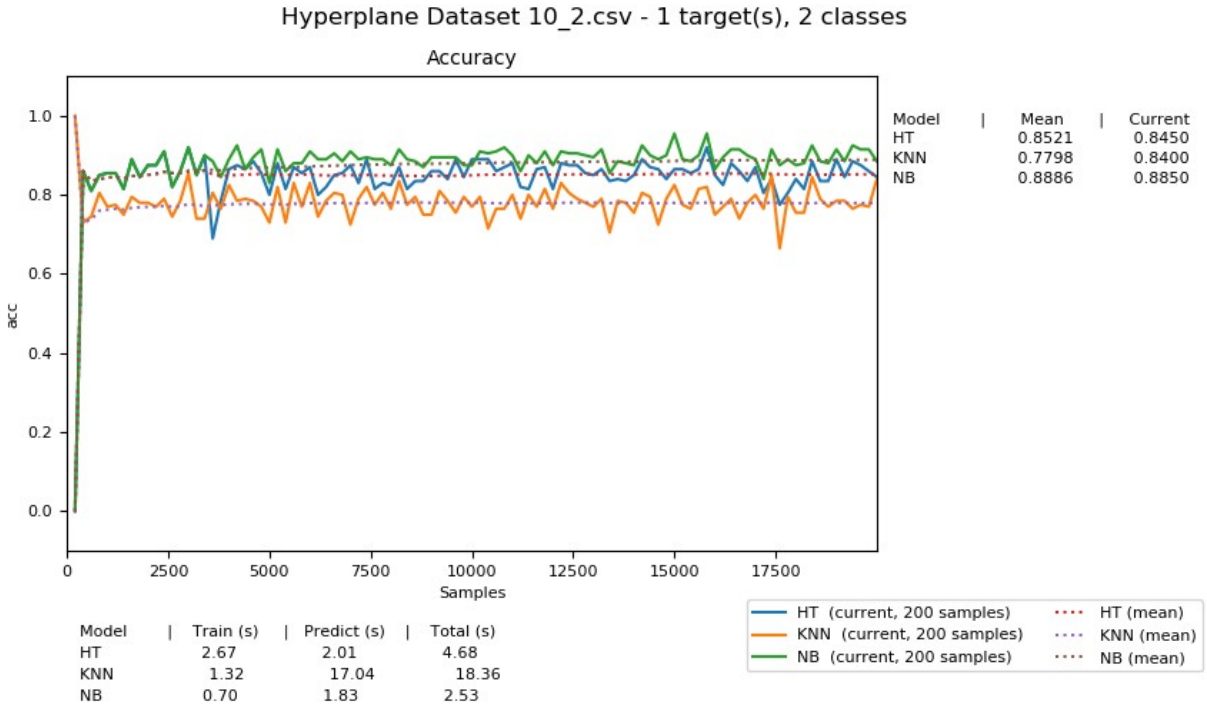
Hakan Gülcü
21702275
GE461-Project 5
13.05.2022

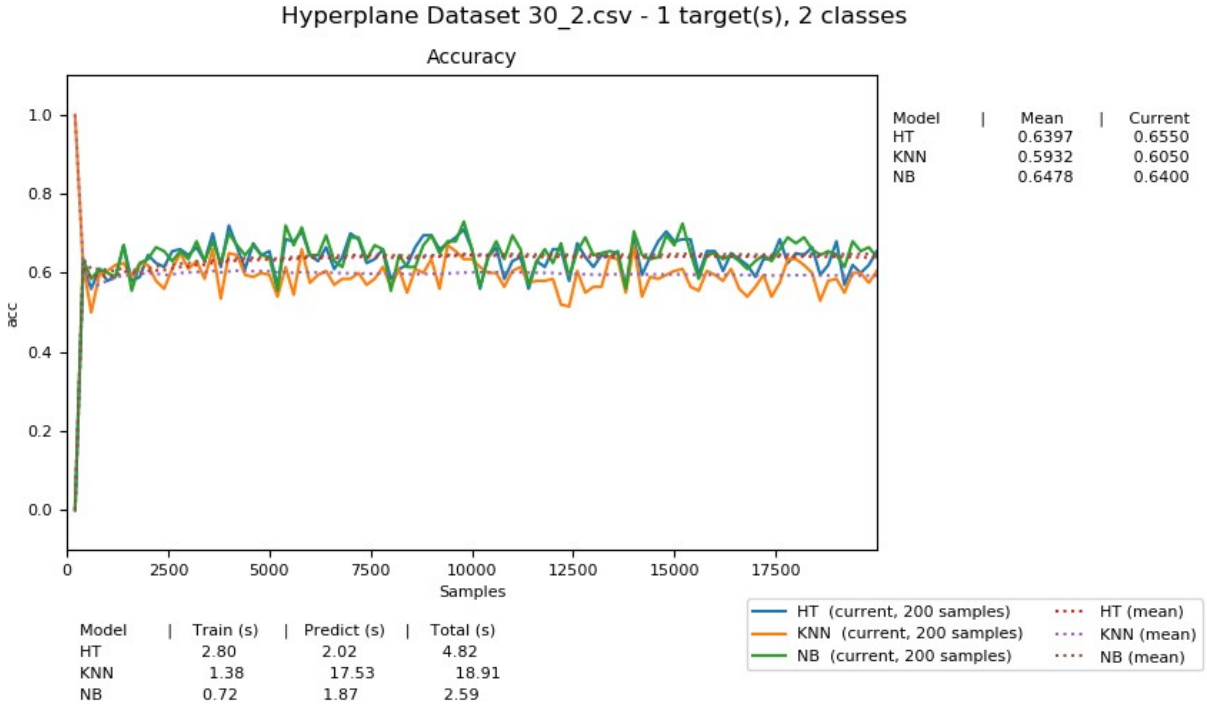
//Evaluation 10_2



[27.87 seconds]
Processed samples: 20000
Mean performance:
HT - Accuracy : 0.8521
HT - Training time (s) : 2.67
HT - Testing time (s) : 2.01
HT - Total time (s) : 4.68
KNN - Accuracy : 0.7798
KNN - Training time (s) : 1.32
KNN - Testing time (s) : 17.04
KNN - Total time (s) : 18.36
NB - Accuracy : 0.8886
NB - Training time (s) : 0.70
NB - Testing time (s) : 1.83
NB - Total time (s) : 2.53

With 10 percent noise and 2 number of drifting features, Naive Bayes has the best accuracy and minimum total time. KNN is the worst and HT is the average one.

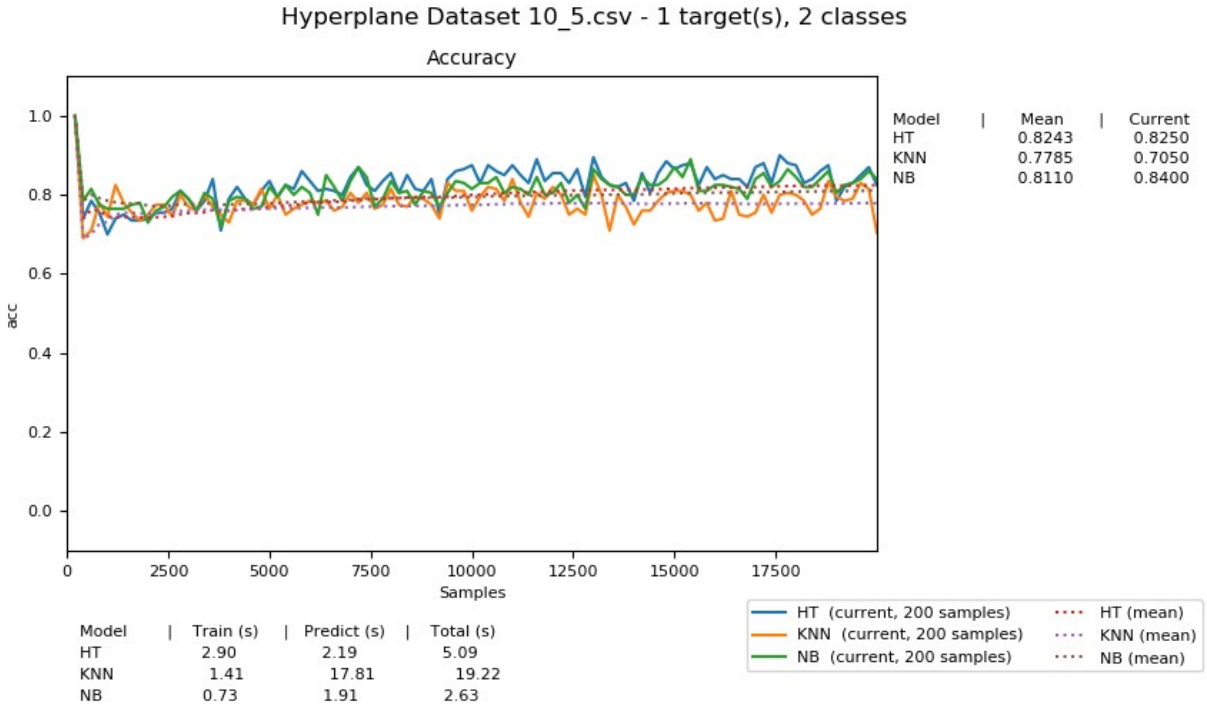
//Evaluation 30_2



[28.68 seconds]
Processed samples: 20000
Mean performance:
HT - Accuracy : 0.6397
HT - Training time (s) : 2.80
HT - Testing time (s) : 2.02
HT - Total time (s) : 4.82
KNN - Accuracy : 0.5932
KNN - Training time (s) : 1.38
KNN - Testing time (s) : 17.53
KNN - Total time (s) : 18.91
NB - Accuracy : 0.6478
NB - Training time (s) : 0.72
NB - Testing time (s) : 1.87
NB - Total time (s) : 2.59

With 30 percent noise and 2 number of drifting features, Naive Bayes has the best accuracy and minimum total time. KNN is the worst and HT is the average one. Therefore, changing noise does not affect the order. However, having more noise makes accuracy less and total time more in general.

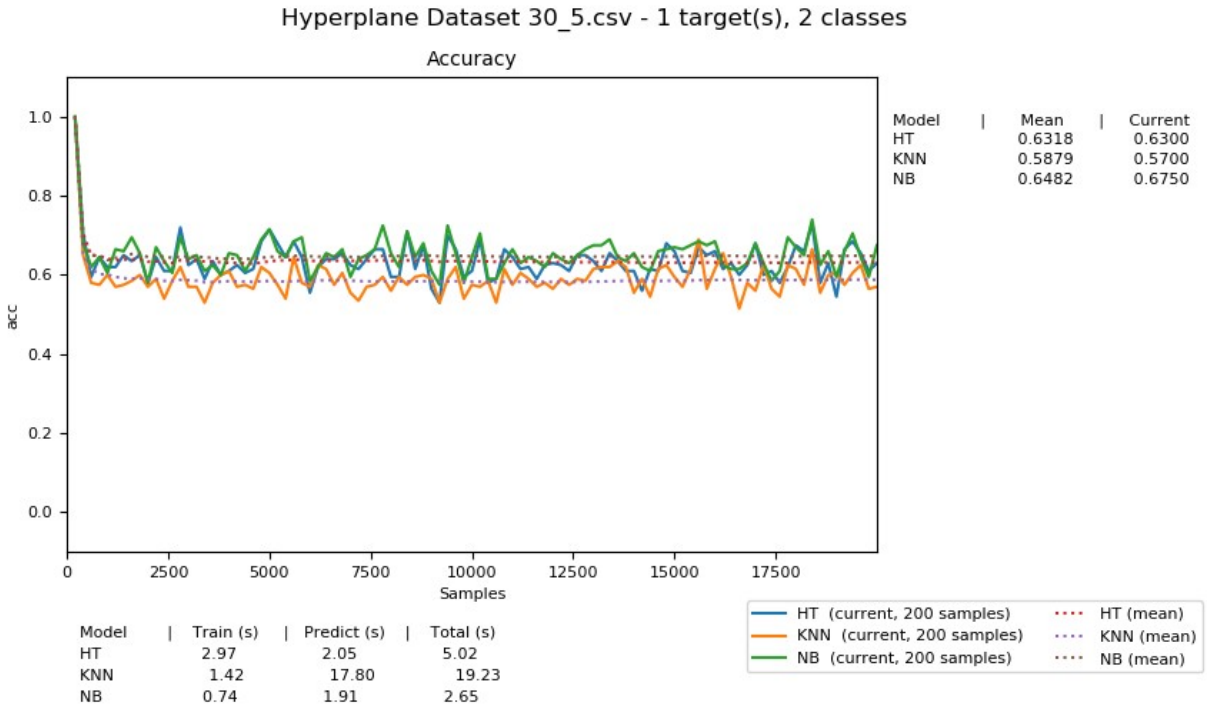
//Evaluation 10_5



[29.34 seconds]
Processed samples: 20000
Mean performance:
HT - Accuracy : 0.8243
HT - Training time (s) : 2.90
HT - Testing time (s) : 2.19
HT - Total time (s) : 5.09
KNN - Accuracy : 0.7785
KNN - Training time (s) : 1.41
KNN - Testing time (s) : 17.81
KNN - Total time (s) : 19.22
NB - Accuracy : 0.8110
NB - Training time (s) : 0.73
NB - Testing time (s) : 1.91
NB - Total time (s) : 2.63

With 10 percent noise and 5 number of drifting features, HT has the best accuracy and minimum total time. KNN is the worst and NB is the average one. However, NB is again finishing earlier. Also, having more drifting features causes less accuracy and more time according to 10 and 5 case.

//Evaluation 30_5



[29.25 seconds]
Processed samples: 20000
Mean performance:
HT - Accuracy : 0.6318
HT - Training time (s) : 2.97
HT - Testing time (s) : 2.05
HT - Total time (s) : 5.02
KNN - Accuracy : 0.5879
KNN - Training time (s) : 1.42
KNN - Testing time (s) : 17.80
KNN - Total time (s) : 19.23
NB - Accuracy : 0.6482
NB - Training time (s) : 0.74
NB - Testing time (s) : 1.91
NB - Total time (s) : 2.65

Same case with the above ones.

//Evaluation 10_2 Batch1
[29.48 seconds]
Processed samples: 20000
Mean performance:
HT - Accuracy : 0.8133
HT - Training time (s) : 2.99
HT - Testing time (s) : 2.05
HT - Total time (s) : 5.04
KNN - Accuracy : 0.7779
KNN - Training time (s) : 1.43
KNN - Testing time (s) : 17.89
KNN - Total time (s) : 19.32
NB - Accuracy : 0.8370
NB - Training time (s) : 0.74
NB - Testing time (s) : 1.92
NB - Total time (s) : 2.67

//Evaluation 10_2 Batch100

[8.58 seconds]
Processed samples: 20000
Mean performance:
HT - Accuracy : 0.8578
HT - Training time (s) : 2.57
HT - Testing time (s) : 1.36
HT - Total time (s) : 3.93
KNN - Accuracy : 0.7781
KNN - Training time (s) : 0.82
KNN - Testing time (s) : 0.77
KNN - Total time (s) : 1.59
NB - Accuracy : 0.8580
NB - Training time (s) : 0.47
NB - Testing time (s) : 1.19
NB - Total time (s) : 1.65
,
//Evaluation 10_2 Batch 1000
[6.84 seconds]
Processed samples: 20200
Mean performance:
HT - Accuracy : 0.8587
HT - Training time (s) : 2.29
HT - Testing time (s) : 1.22
HT - Total time (s) : 3.51
KNN - Accuracy : 0.7782
KNN - Training time (s) : 0.72
KNN - Testing time (s) : 0.55
KNN - Total time (s) : 1.28
NB - Accuracy : 0.8700
NB - Training time (s) : 0.42
NB - Testing time (s) : 1.14
NB - Total time (s) : 1.57

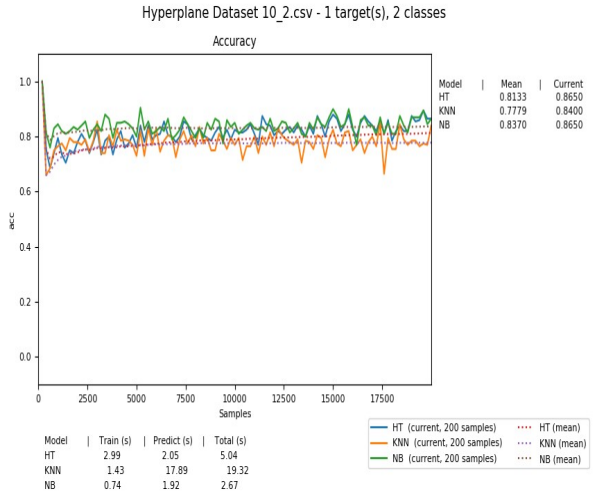
//Evaluation 30_2 Batch 1
[29.96 seconds]
Processed samples: 20000
Mean performance:
HT - Accuracy : 0.6332
HT - Training time (s) : 3.09
HT - Testing time (s) : 2.05
HT - Total time (s) : 5.15
KNN - Accuracy : 0.5932
KNN - Training time (s) : 1.46
KNN - Testing time (s) : 18.13
KNN - Total time (s) : 19.59
NB - Accuracy : 0.6551
NB - Training time (s) : 0.76
NB - Testing time (s) : 1.95
NB - Total time (s) : 2.71
/
/Evaluation 30_2 Batch 100
[8.07 seconds]
Processed samples: 20000
Mean performance:
HT - Accuracy : 0.6504
HT - Training time (s) : 2.46
HT - Testing time (s) : 1.23
HT - Total time (s) : 3.69
KNN - Accuracy : 0.5891
KNN - Training time (s) : 0.76
KNN - Testing time (s) : 0.73
KNN - Total time (s) : 1.49
NB - Accuracy : 0.6587
NB - Training time (s) : 0.43
NB - Testing time (s) : 1.13
NB - Total time (s) : 1.56

//Evaluation 30_2 Batch 1000
[7.08 seconds]
Processed samples: 20200
Mean performance:
HT - Accuracy : 0.6637
HT - Training time (s) : 2.43
HT - Testing time (s) : 1.28
HT - Total time (s) : 3.71
KNN - Accuracy : 0.5949
KNN - Training time (s) : 0.75
KNN - Testing time (s) : 0.56
KNN - Total time (s) : 1.31
NB - Accuracy : 0.6628
NB - Training time (s) : 0.44
NB - Testing time (s) : 1.12
NB - Total time (s) : 1.55

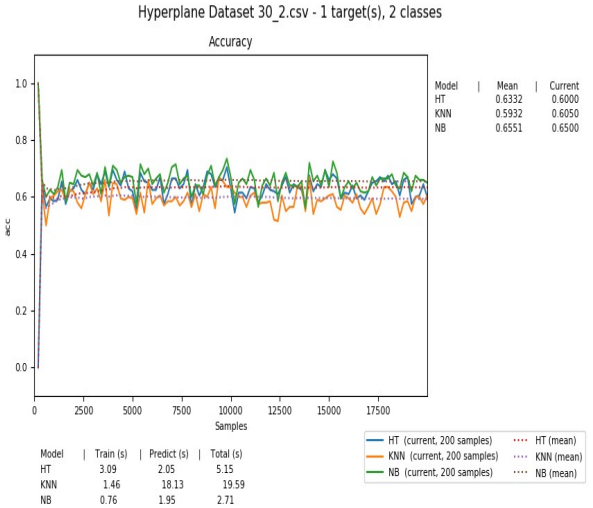
Firstly, it is clear that having more batch make classifier faster. However, it does not change the accuracy a lot. There are just slight changes. Also, having more batch affect KNN very good and its total time becomes very less. Even the best in some cases. However, it still has the less accuracy in any case.

Secondly, having more noise affects accuracy in a bad way. Other things are similar to the first case.

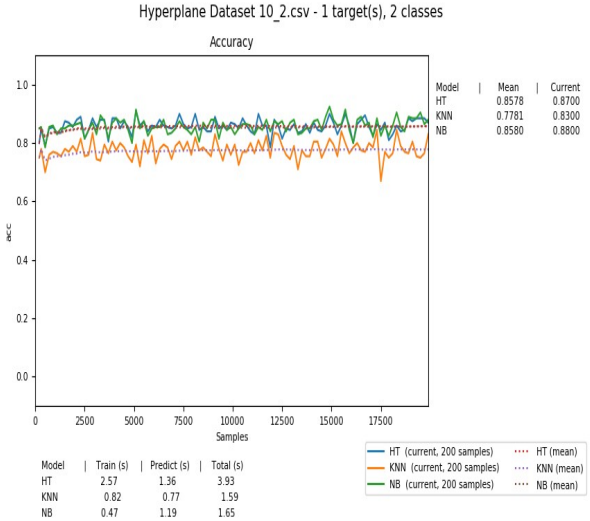
Batch 1



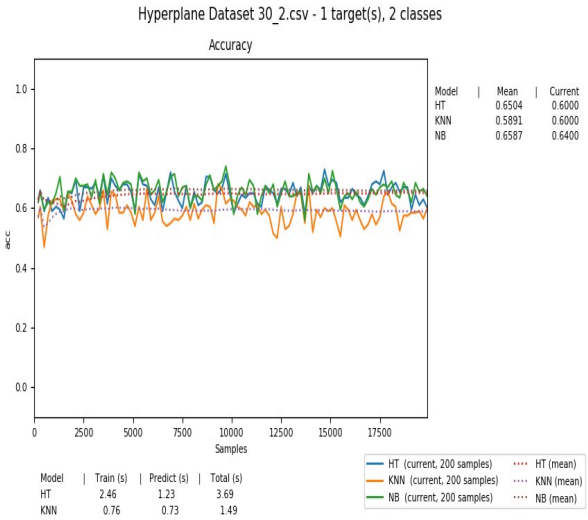
Batch 1



Batch 100



Batch 100



//Evaluation 30_2 Batch 1
[29.14 seconds]
Processed samples: 20000
Mean performance:
HT - Accuracy : 0.7624
HT - Training time (s) : 2.96
HT - Testing time (s) : 2.04
HT - Total time (s) : 5.01
KNN - Accuracy : 0.7785
KNN - Training time (s) : 1.42
KNN - Testing time (s) : 17.69
KNN - Total time (s) : 19.11
NB - Accuracy : 0.8086
NB - Training time (s) : 0.74
NB - Testing time (s) : 1.90
NB - Total time (s) : 2.64

//Evaluation 30_2 Batch 100
[8.83 seconds]
Processed samples: 20000
Mean performance:
HT - Accuracy : 0.8132
HT - Training time (s) : 2.65
HT - Testing time (s) : 1.51
HT - Total time (s) : 4.17
KNN - Accuracy : 0.7795
KNN - Training time (s) : 0.82
KNN - Testing time (s) : 0.76
KNN - Total time (s) : 1.58
NB - Accuracy : 0.8203
NB - Training time (s) : 0.47
NB - Testing time (s) : 1.19
NB - Total time (s) : 1.65
//Evaluation 30_2 Batch 1000
[7.37 seconds]
Processed samples: 20200
Mean performance:
HT - Accuracy : 0.8286
HT - Training time (s) : 2.43
HT - Testing time (s) : 1.48
HT - Total time (s) : 3.91
KNN - Accuracy : 0.7777
KNN - Training time (s) : 0.76
KNN - Testing time (s) : 0.58
KNN - Total time (s) : 1.34
NB - Accuracy : 0.8293
NB - Training time (s) : 0.43
NB - Testing time (s) : 1.12
NB - Total time (s) : 1.55

//Evaluation 30_5 Batch 1
[29.70 seconds]
Processed samples: 20000
Mean performance:
HT - Accuracy : 0.6358
HT - Training time (s) : 3.05
HT - Testing time (s) : 2.24
HT - Total time (s) : 5.29
KNN - Accuracy : 0.5879
KNN - Training time (s) : 1.43
KNN - Testing time (s) : 17.89
KNN - Total time (s) : 19.32
NB - Accuracy : 0.6474
NB - Training time (s) : 0.75
NB - Testing time (s) : 1.92
NB - Total time (s) : 2.67

//Evaluation 30_5 Batch 100
[8.32 seconds]
Processed samples: 20000
Mean performance:
HT - Accuracy : 0.6440
HT - Training time (s) : 2.52
HT - Testing time (s) : 1.45
HT - Total time (s) : 3.97
KNN - Accuracy : 0.5901
KNN - Training time (s) : 0.78
KNN - Testing time (s) : 0.72
KNN - Total time (s) : 1.50
NB - Accuracy : 0.6498
NB - Training time (s) : 0.44
NB - Testing time (s) : 1.11
NB - Total time (s) : 1.55

//Evaluation 30_5 Batch 1000
[7.74 seconds]
Processed samples: 20200
Mean performance:
HT - Accuracy : 0.6557
HT - Training time (s) : 2.60
HT - Testing time (s) : 1.54
HT - Total time (s) : 4.15
KNN - Accuracy : 0.5869
KNN - Training time (s) : 0.82
KNN - Testing time (s) : 0.56
KNN - Total time (s) : 1.38
NB - Accuracy : 0.6523
NB - Training time (s) : 0.48
NB - Testing time (s) : 1.18
NB - Total time (s) : 1.66

For the 5 drifting features case, having more batch affect time very good for each classifiers. However, having more noise affect accuracy in a bad way. Also, KNN becomes better with more batches where it was the worst at the very beginning at any case. All other things are similar to the previous cases.

Batch Hoeffding Tree Accuracy: 0.829 for dataset Hyperplane Dataset 10_2
Batch KNN Accuracy: 0.7718 for dataset Hyperplane Dataset 10_2
Batch Naive Bayes Accuracy: 0.8926 for dataset Hyperplane Dataset 10_2

Batch Hoeffding Tree Accuracy: 0.6722 for dataset Hyperplane Dataset 30_2
Batch KNN Accuracy: 0.5824 for dataset Hyperplane Dataset 30_2
Batch Naive Bayes Accuracy: 0.7026 for dataset Hyperplane Dataset 30_2

Batch Hoeffding Tree Accuracy: 0.8156 for dataset Hyperplane Dataset 10_5
Batch KNN Accuracy: 0.7628 for dataset Hyperplane Dataset 10_5
Batch Naive Bayes Accuracy: 0.8994 for dataset Hyperplane Dataset 10_5

Batch Hoeffding Tree Accuracy: 0.6412 for dataset Hyperplane Dataset 30_5
Batch KNN Accuracy: 0.5958 for dataset Hyperplane Dataset 30_5
Batch Naive Bayes Accuracy: 0.6932 for dataset Hyperplane Dataset 30_5

When we look at the ac curacies of each Classifiers with different datasets, Naive Bayes performs best. Hoeffding Tree is always the average, and KNN is the worst. However, there is not very big difference between Hoeffding Tree and KNN. Having more drifting features affects KNN in a good way, however, for the others it affects in a bad way or almost not affects.

Voting Accuracy: 0.8736 for dataset Hyperplane Dataset 10_2
Weighted Voting Accuracy: 0.8736 for dataset Hyperplane Dataset 10_2

Voting Accuracy: 0.691 for dataset Hyperplane Dataset 30_2
Weighted Voting Accuracy: 0.691 for dataset Hyperplane Dataset 30_2

Voting Accuracy: 0.8676 for dataset Hyperplane Dataset 10_5
Weighted Voting Accuracy: 0.8676 for dataset Hyperplane Dataset 10_5

Voting Accuracy: 0.6732 for dataset Hyperplane Dataset 30_5
Weighted Voting Accuracy: 0.6732 for dataset Hyperplane Dataset 30_5

For voting accuracies, having less noise affects positively the classifier. However, having more drifting features decreases the accuracy.
For weighted voting accuracy, it is exactly the same situation. It is because both have the same results.

To improve the model, there are different classifiers. For example, KNNADWIN is an option to KNN with better results. When we apply it to KNN and compare, it generates better results. Also, we can implement them manually to make performance better because those kind of modules are implemented for common cases.

Finally, individual models are better then ensemble methods when we look both accuracy and time. Also, for time time issue, ensemble models are good enough but not more than individual models. Having more batch sizes affects both positively.