Project V

Data Stream Mining on Hyperplane Dataset

Caner Canlıer 21702121

caner.canlier@ug.bilkent.edu.tr

Bilkent University

GE 461 Data Science

10th of May, 2022

Question 1

I generated 20.000 instance with 10 features by using HyperplaneGenerator() function provided by skmultiflow.data and write them on a csv file. I changed noise percentage and number of drifting features based on the instructions in part (a-d). I had 4 different hyperplane dataset and named them according to their noise and number of drifting features.

Question 2

For this question, I construct and train three online classifiers for four Hyperplane datasets. Those classfiers were HoeffdingTree (HT), K-Nearest Neighbor (KNN), Naïve Bayes (NB). I benefited from Interleaved-Test-Then-Train approach in order to calculate temporal accuracy of each classifier on four different datasets. Individually, each classifier's accuracy can be seen in Figure 1-12, and comparison of all classifiers can be seen in Figure 13-16.

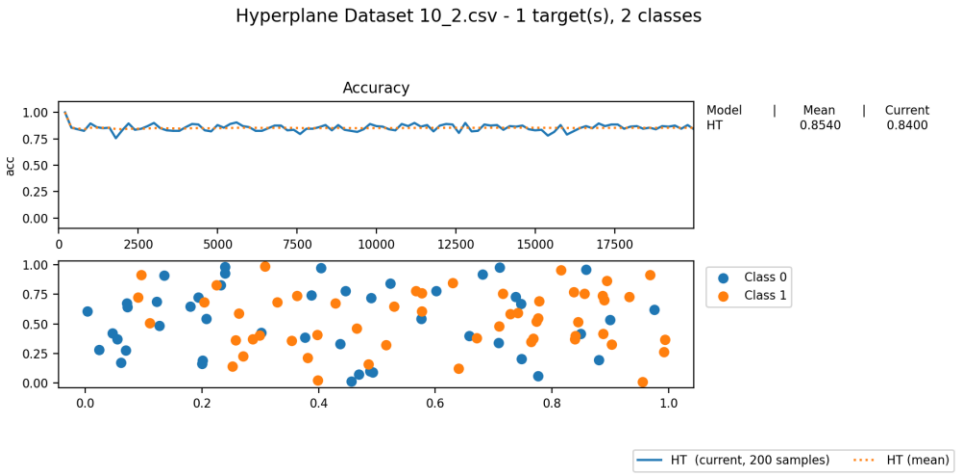

Figure 1: HoeffdingTree Temporal Accuracy on Hyperplane Dataset 10_2

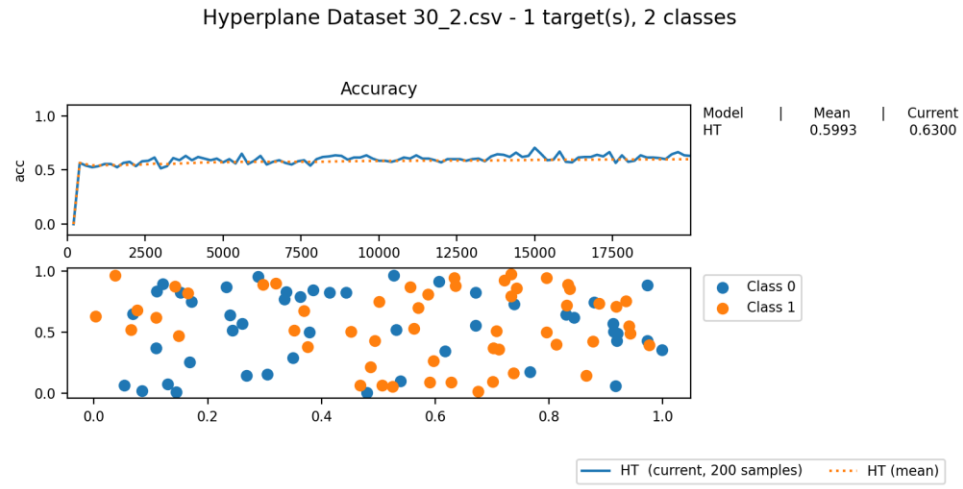

Figure 2: HoeffdingTree Temporal Accuracy on Hyperplane Dataset 30_2

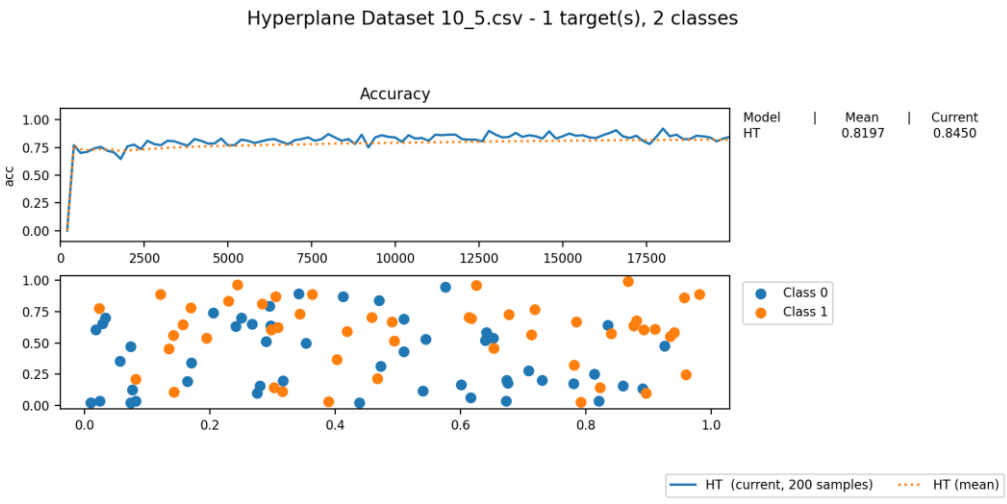

Figure 3: HoeffdingTree Temporal Accuracy on Hyperplane Dataset 10_5

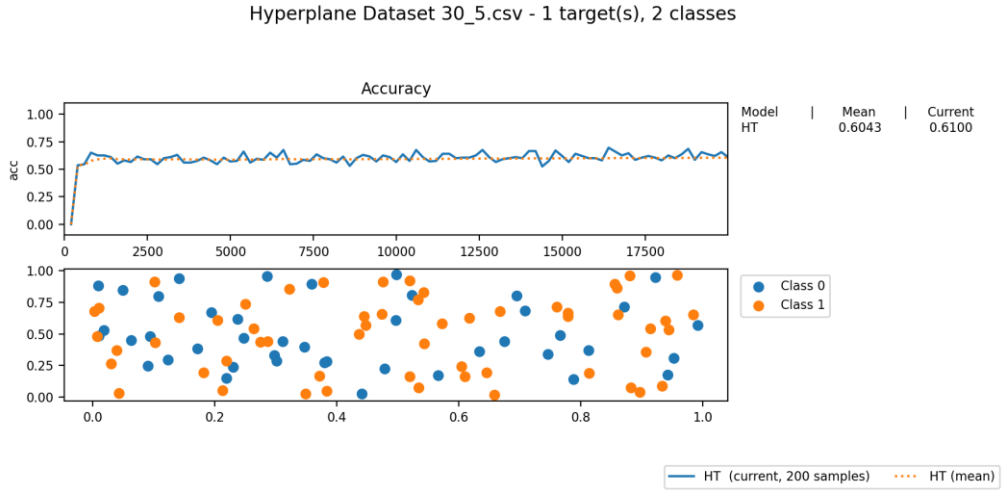

Figure 4: HoeffdingTree Temporal Accuracy on Hyperplane Dataset 30_5

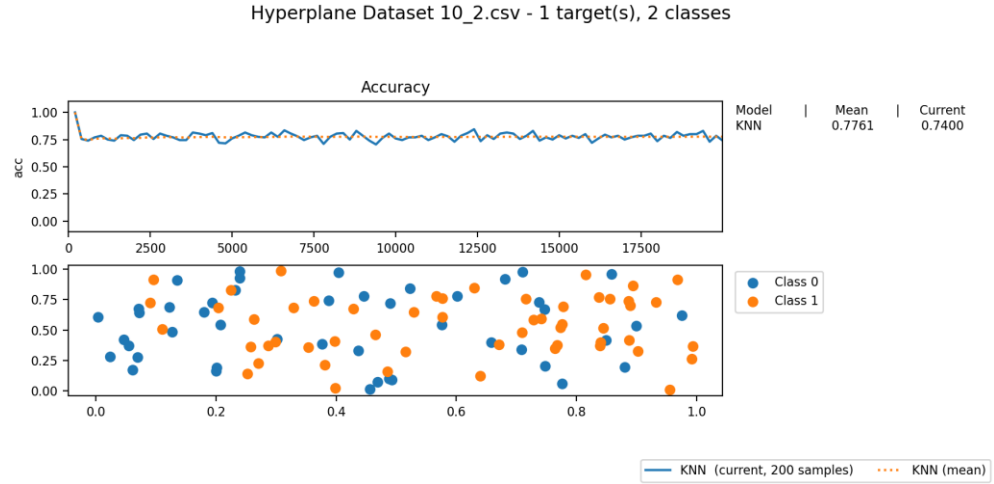

Figure 5: K-Nearest Neighbor Temporal Accuracy on Hyperplane Dataset 10_2

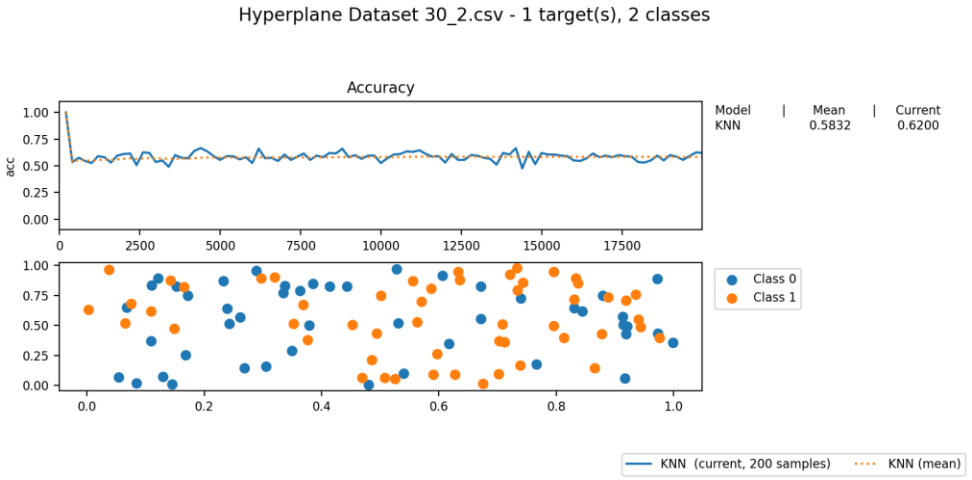

Figure 6: K-Nearest Neighbor Temporal Accuracy on Hyperplane Dataset 30_2

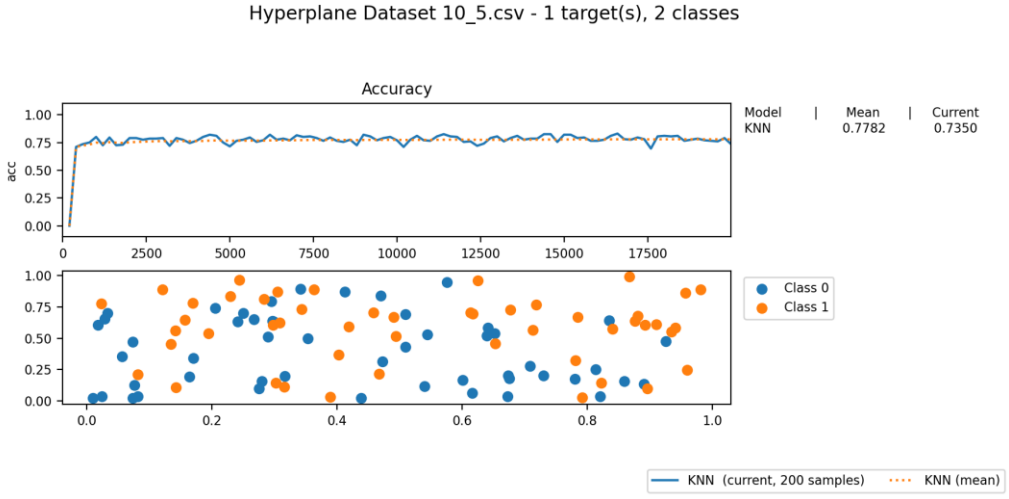

Figure 7: K-Nearest Neighbor Temporal Accuracy on Hyperplane Dataset 10_5

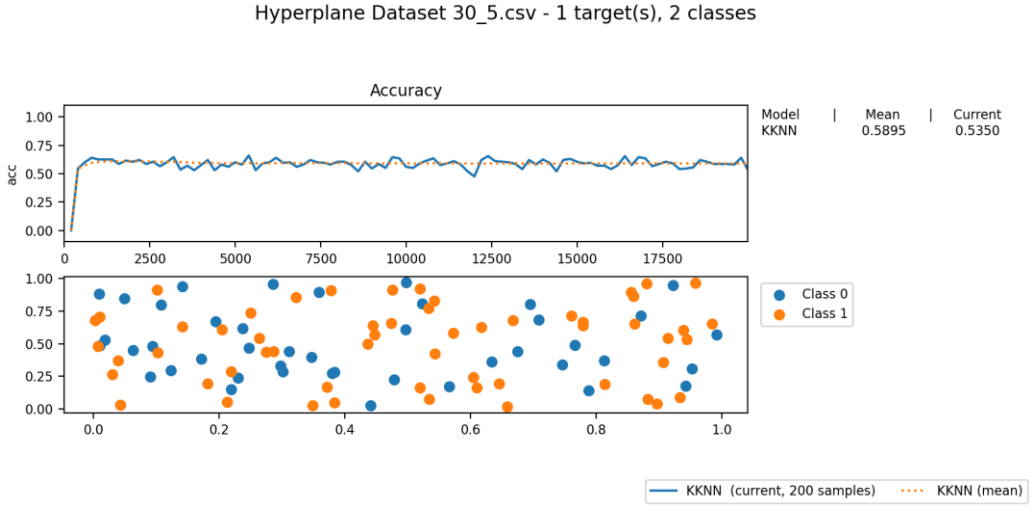

Figure 8: K-Nearest Neighbor Temporal Accuracy on Hyperplane Dataset 30_5

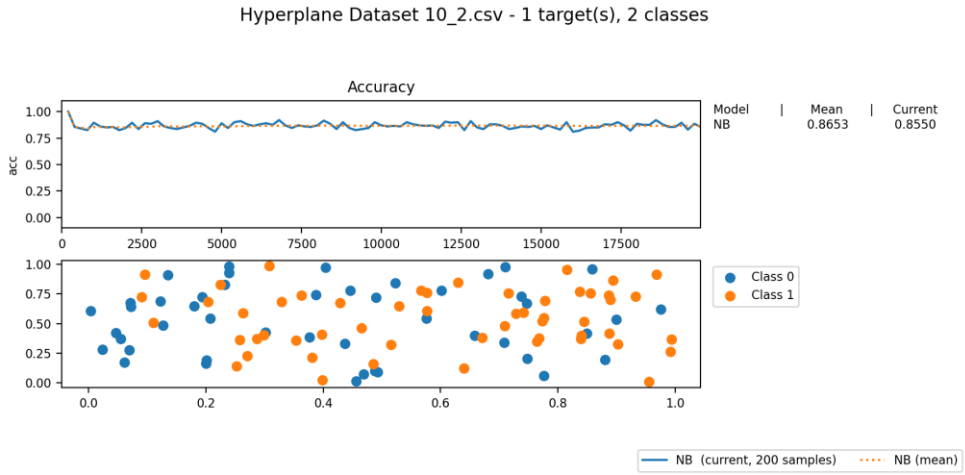

Figure 9: Naïve Bayes Temporal Accuracy on Hyperplane Dataset 10_2

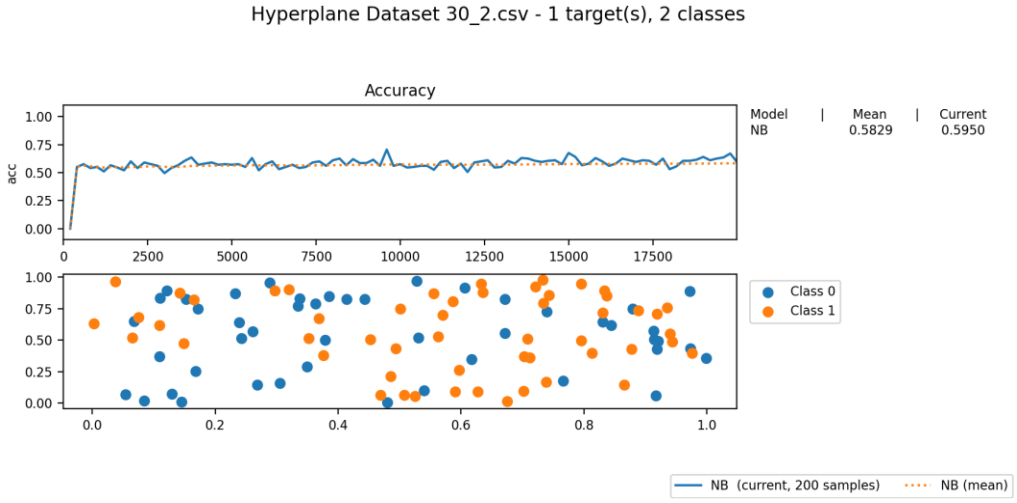

Figure 10: Naïve Bayes Temporal Accuracy on Hyperplane Dataset 30_2

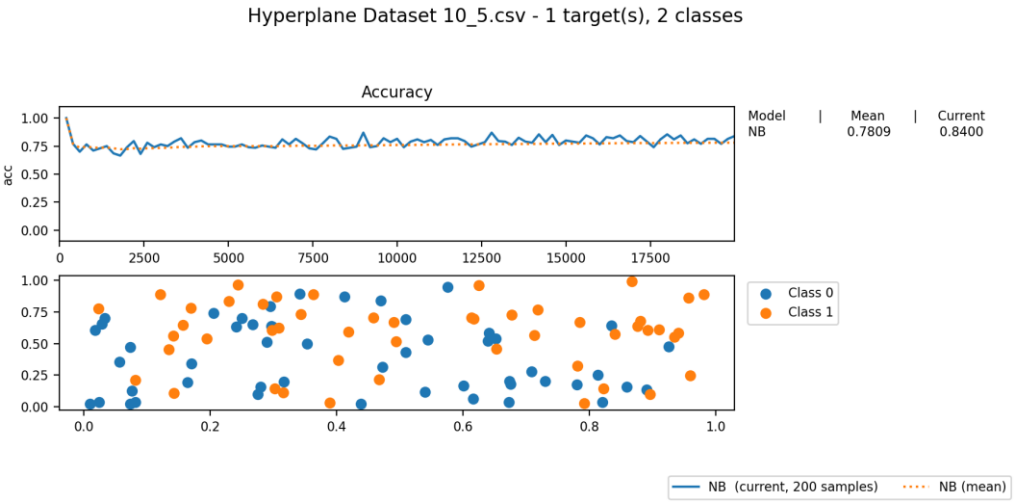

Figure 11: Naïve Bayes Temporal Accuracy on Hyperplane Dataset 10_5

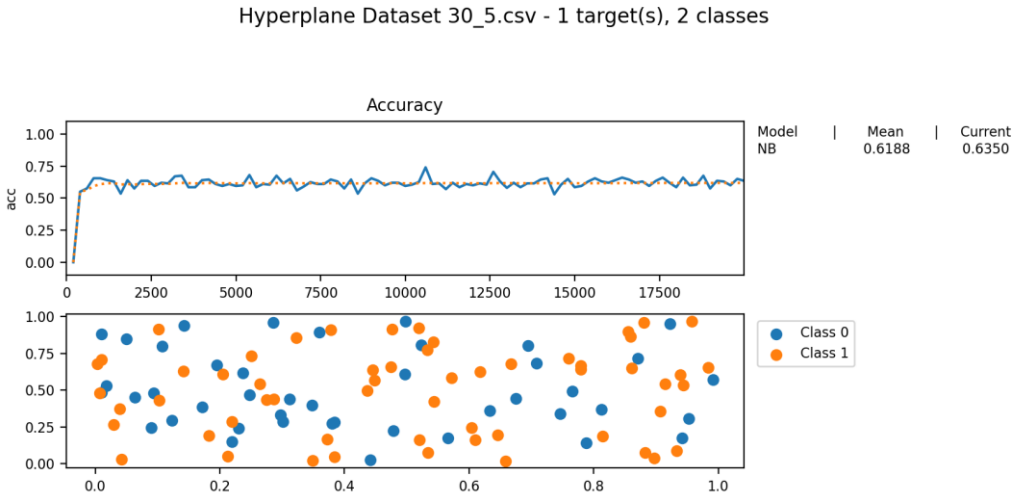

Figure 12: Naïve Bayes Temporal Accuracy on Hyperplane Dataset 30_5

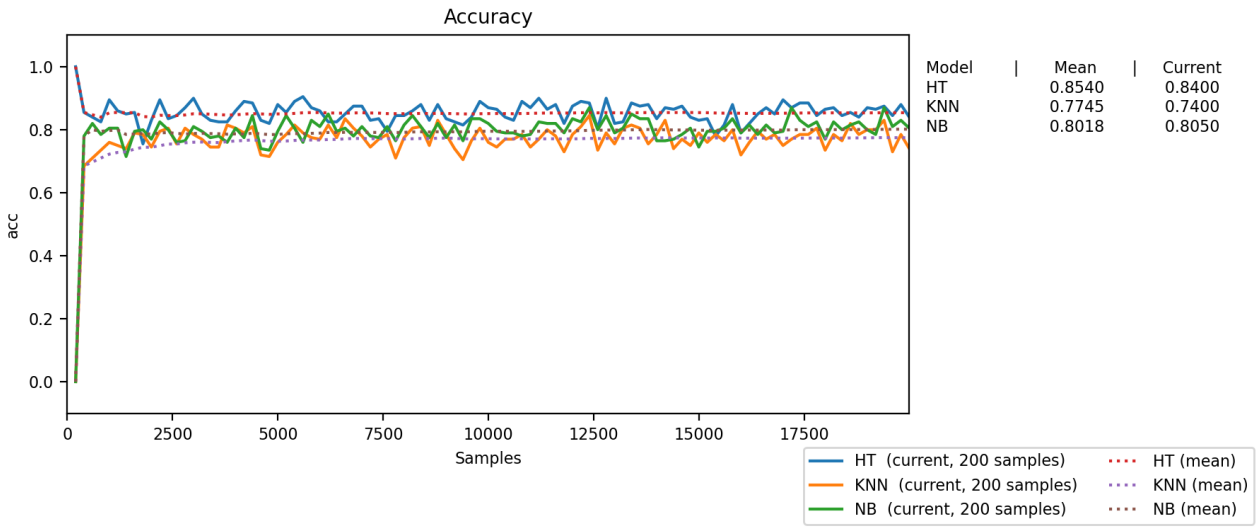

Figure 13: All Classifier's Temporal Accuracy on Hyperplane Dataset 10_2

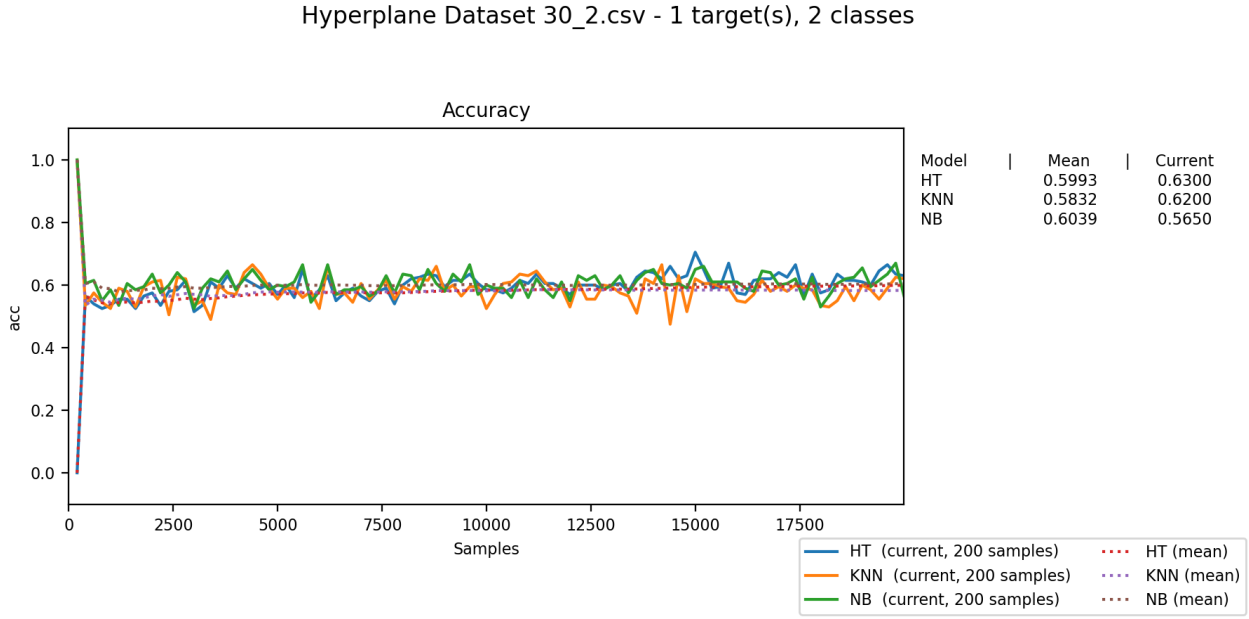

Figure 14: All Classifier's Temporal Accuracy on Hyperplane Dataset 30_2

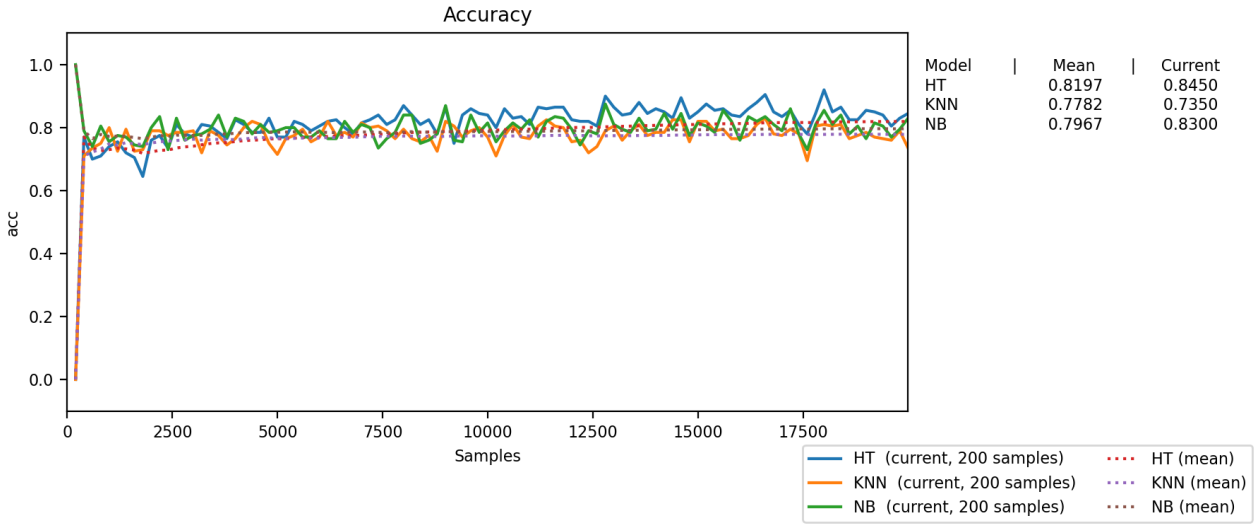

Figure 15: All Classifier's Temporal Accuracy on Hyperplane Dataset 10_5

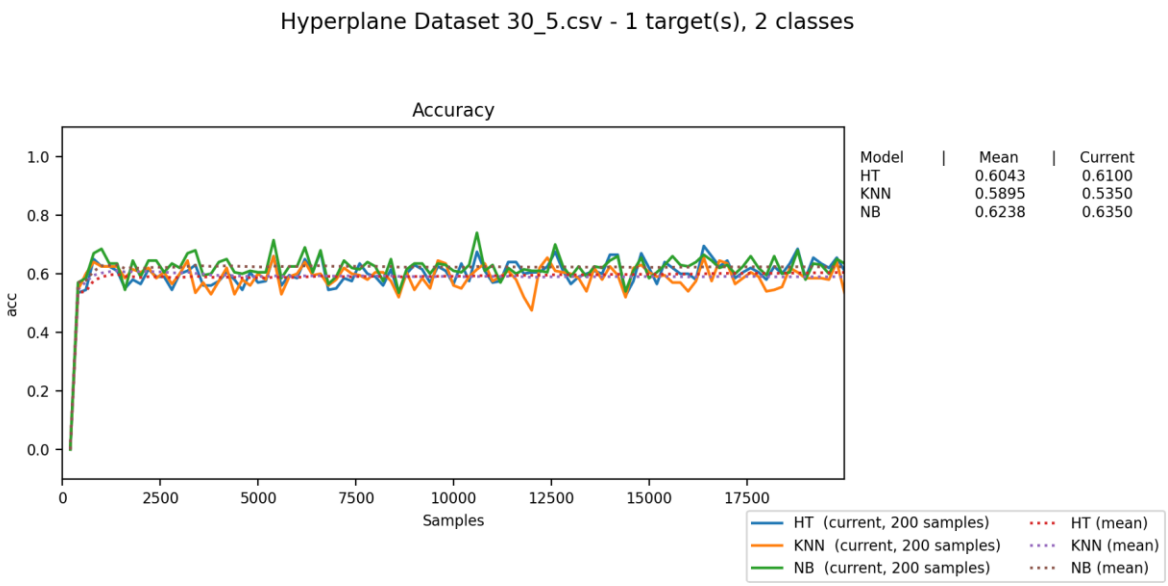

Figure 16: All Classifier's Temporal Accuracy on Hyperplane Dataset 30_5

	Hyperplane Dataset 10_2	**Hyperplane Dataset 30_2**	**Hyperplane Dataset 10_5**	**Hyperplane Dataset 30_5**
HoeffdingTree	0.85	0.59	0.81	0.60
K nearest Neighbour	0.77	0.58	0.77	0.58
Naïve Bayes	0.80	0.60	0.79	0.62

Table 1: Accuracy for different classifiers on different data sets

From the figure 13 it can be seen that for Hyperplane Dataset 10_2, accuracy of HT started from 1 and goes down with each sample, while KNN and NB started from 0. Also in the figure 16, all classifiers' accuracy starts from 0 and goes up. Until 500th sample, temporal accuracy of each classifier dramatically increase and then fluctuates.

Since Table 1 is the summary of all the figures, let's examine the results of the table. It can be observed that increasing the noise percentage decrease the temporal accuracy quickly. On the other hand, increasing the number of drifting features doesn't seem have an effect that much on the overall accuracy. We can claim that HoeffdingTree performs better than other online classifiers when noise percentage is low. However, when noise percentage is increased from 0.10 to 0.30, Naïve Bayes starts to perform slightly better than HoeffdingTree. It can be seen that K nearest Neighbour has lowest accuracy rate for all datasets. Also, to compare the accuracies of four dataset, we can observe that the all online classification algorithms achieve higher accuracy rate in HyperplaneDataset 10_2 which has the least noise percentage and number of drifting features.

Question 3

In this part, two different ensemble classifiers that combines HT, KNN and NB has been constructed for the four Hyperplane dataset. One of them was, Majority Voting Rule (MV) and other one was Weighted Majority Voting Rule (WMV). Interleaved-Test-Then-Train technique has been used to calculate temporal accuracy of each data set. Graph of the accuracy rates can be seen in Figure 17, Figure 18, Figure 19, Figure 20.

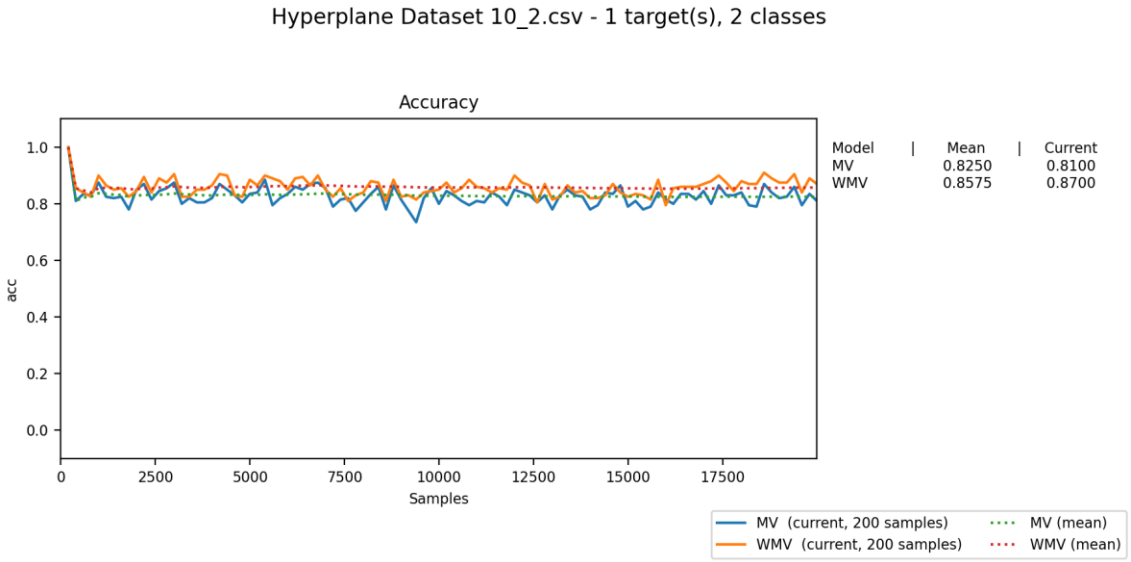

Figure 17: Ensemble Classifiers' Temporal Accuracy on Hyperplane Dataset 10_2

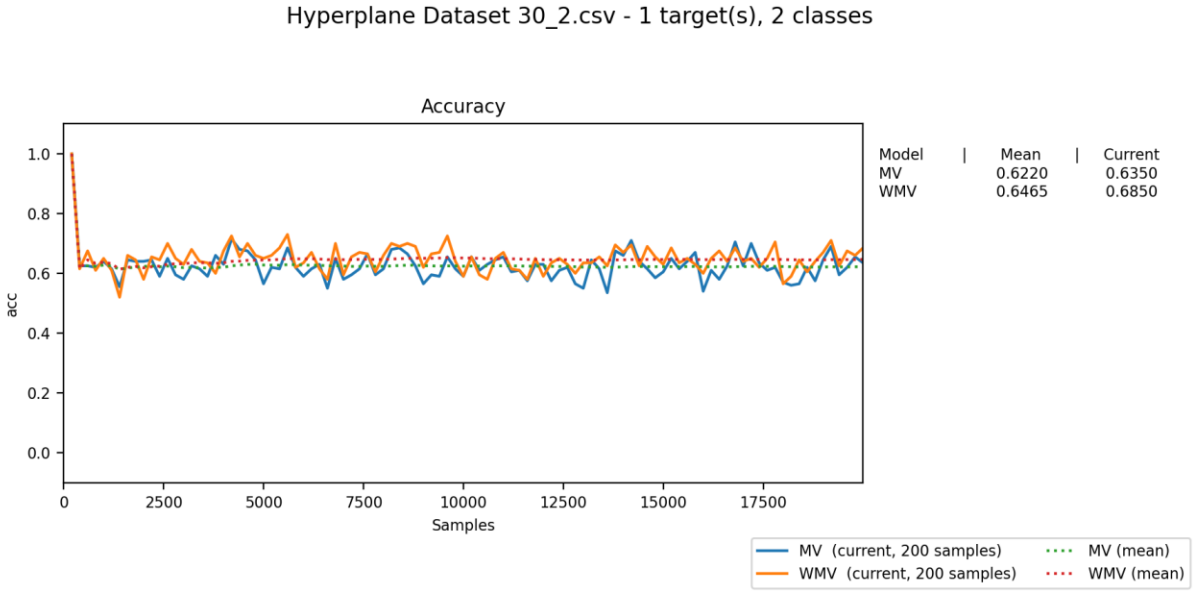

Figure 17: Ensemble Classifiers' Temporal Accuracy on Hyperplane Dataset 30_2

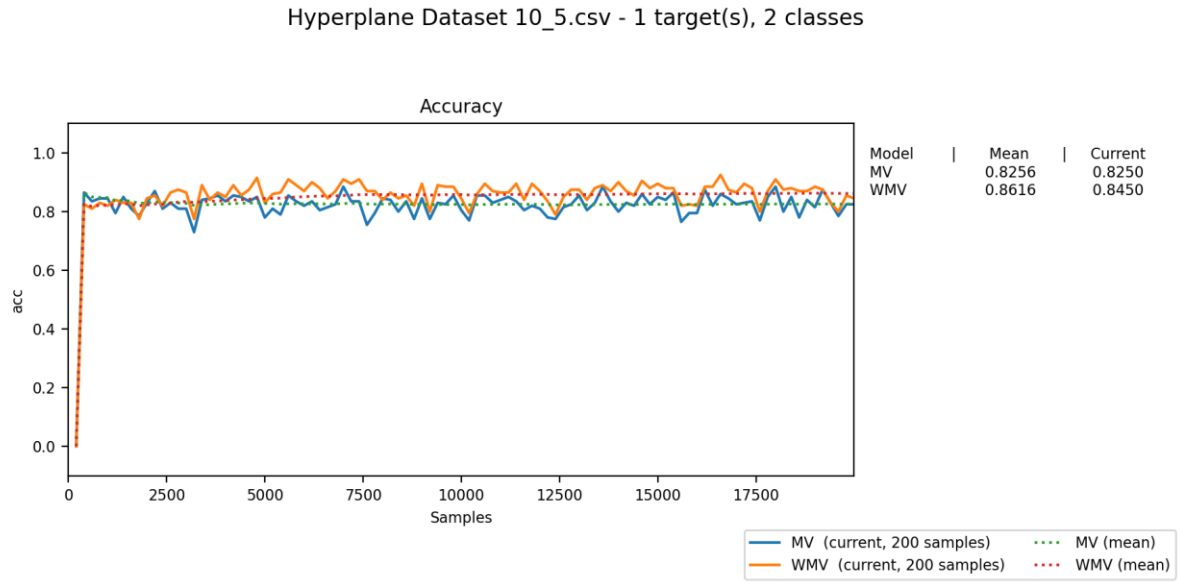

Figure 19: Ensemble Classifiers' Temporal Accuracy on Hyperplane Dataset 10_5

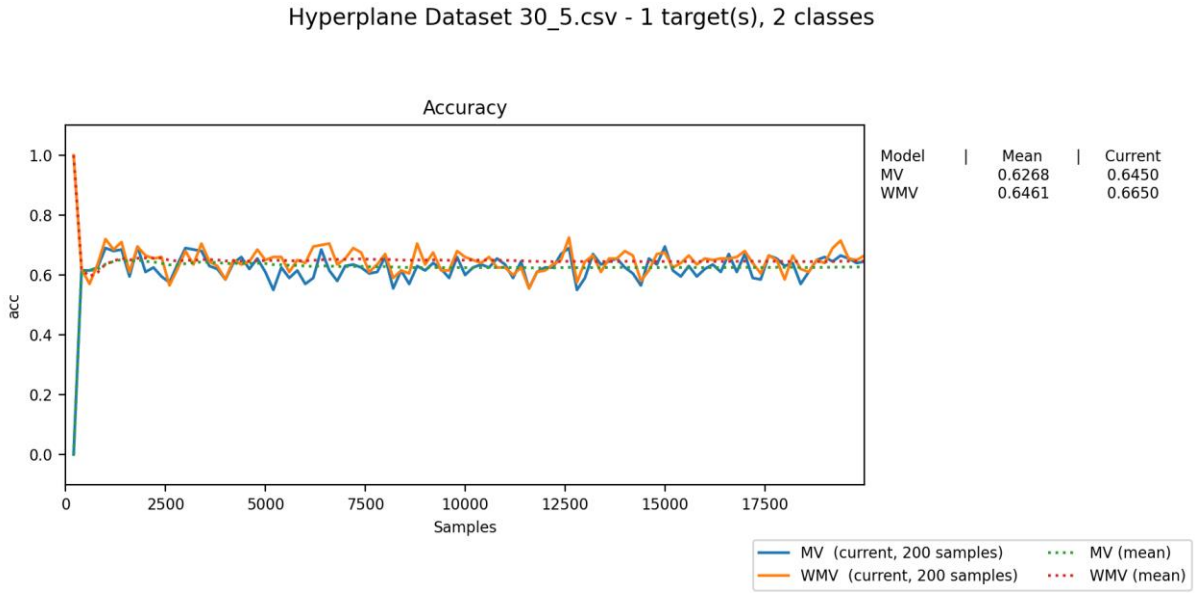

Figure 20: Ensemble Classifiers' Temporal Accuracy on Hyperplane Dataset 30_5

	MV	**WMV**
Hyperplane Dataset 10_2	0.82	0.85
Hyperplane Dataset 30_2	0.62	0.64
Hyperplane Dataset 10_5	0.82	0.86
Hyperplane Dataset 30_5	0.62	0.64

Table 2: Accuracy for different ensemble classifiers on different data sets

From the figures we can say that, for all datasets until approximately 500th sample, accuracy experience a drastic change. However, after that point it starts to fluctuate rather than showing a trend.

Table 2 is the summary for all the figures from 17 to 20. According to our observations, both MV and WMV show the same pattern like the other classifiers. For instance, when the noise increases, accuracy ration decreases for all ensemble classifiers, as well. On the other, the general performance of the ensemble classifiers is higher than single online classifiers. Also, when we compare the two ensemble classifier, we can claim that WMV outperform MV in each datasets. When the noise percentage goes up from 10 to 30, accuracy for both ensemble classifiers, go down approximately 20%.

Question 4

b) For this part, Temporal accuracies of online classifiers are compared by using Interleaved-Test-Then-Train approach. To split the datasets into two I used train_test_split() function and use 70% of data for train and 30% for test. In table 3, the temporal accuracies can be seen for four different datasets and three different online classifiers.

	HT	**KNN**	**NB**
Hyperplane Dataset 10_2	0.86	0.77	0.88
Hyperplane Dataset 30_2	0.67	0.59	0.65
Hyperplane Dataset 10_5	0.85	0.78	0.82
Hyperplane Dataset 30_5	0.63	0.56	0.63

Table 3: Accuracy for different classifiers on test data

The results of table 3 is similar to table 1. Since we didn't change the noise percentage or drifting features, we didn't obtain highly different results than table 1. According to table 3, it can be seen that when drifting features and noise percentage are at their low level, Naïve Bayes classifier performs better than others. However, if we increase the drifting features from 2 to 5, accuracy rate of the NB becomes lower than HoeffdingTree classifier. Although K nearest neighbour performs worse among the other classifiers, it can be seen that increasing drifting feature cause an increase in only KNN by 0.77 to 0.78.

As we interpreted from the figure 1, it can be claimed that when the noise percentage goes up, accuracy rate of the classifiers goes down. Also, it can be seen that HT gives better result when the noise percentage is high.

c)

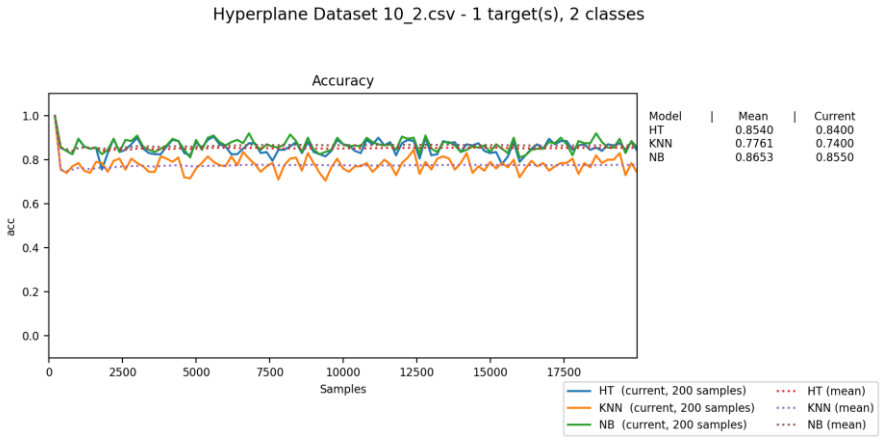

Figure 21: 1 Batch Classifiers Temporal Accuracy on Hyperplane Dataset 10_2

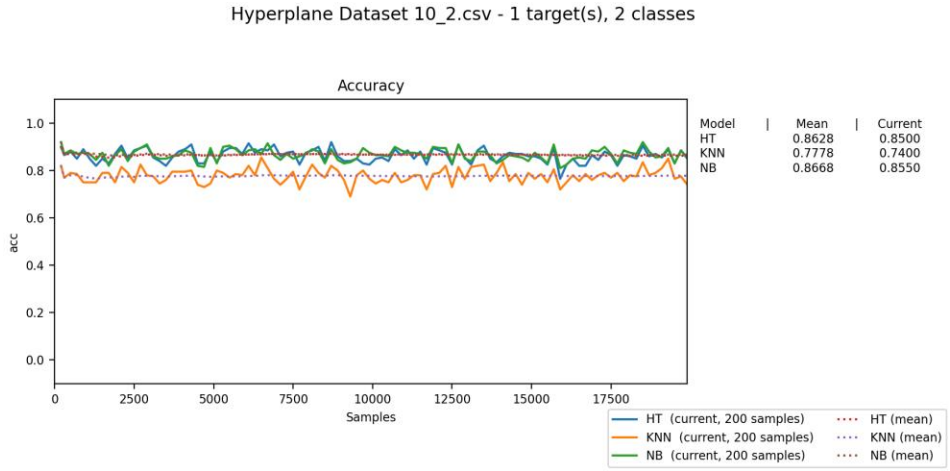

Figure 22: 100 Batch Classifier Temporal Accuracy on Hyperplane Dataset 10_2

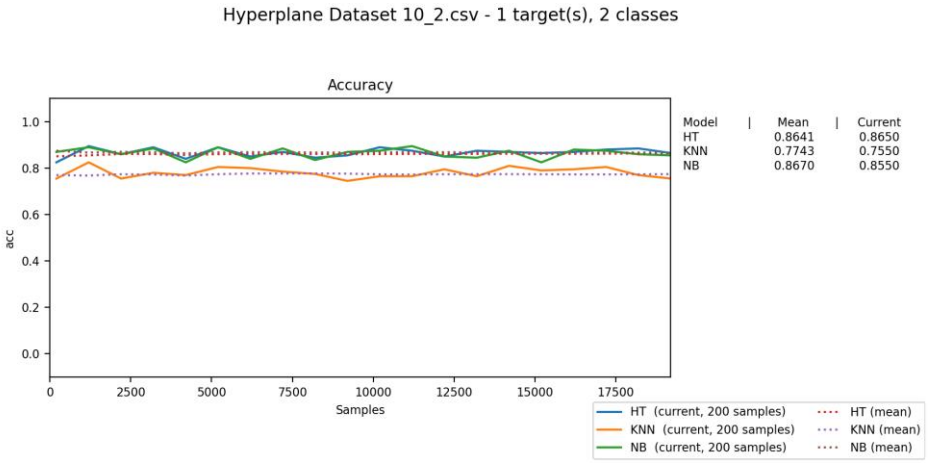

Figure 23: 1000 Batch Classifier Temporal Accuracy on Hyperplane Dataset 10_2

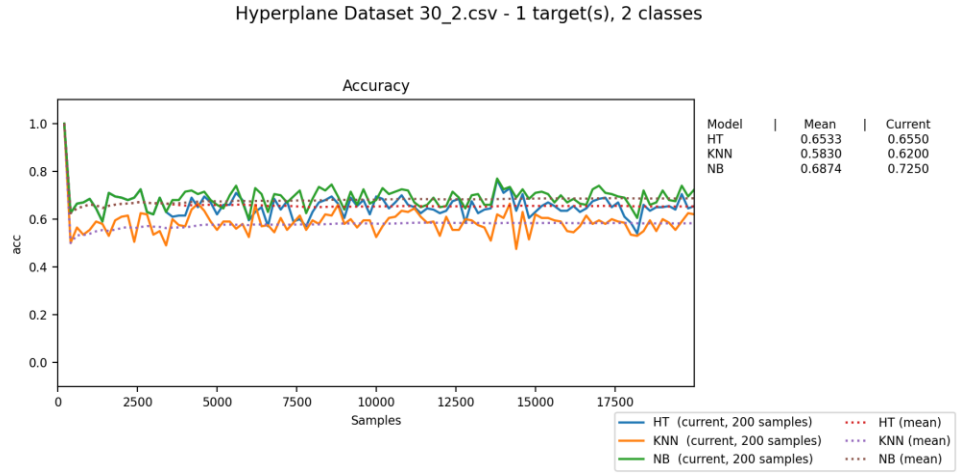

Figure 24: 1 Batch Classifier Temporal Accuracy on Hyperplane Dataset 30_2

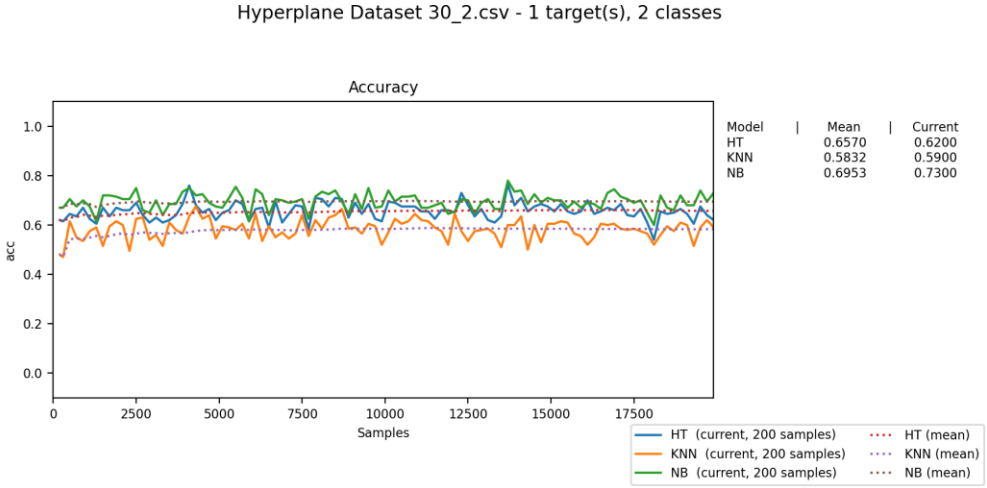

Figure 25: 100 Batch Classifier Temporal Accuracy on Hyperplane Dataset 30_2

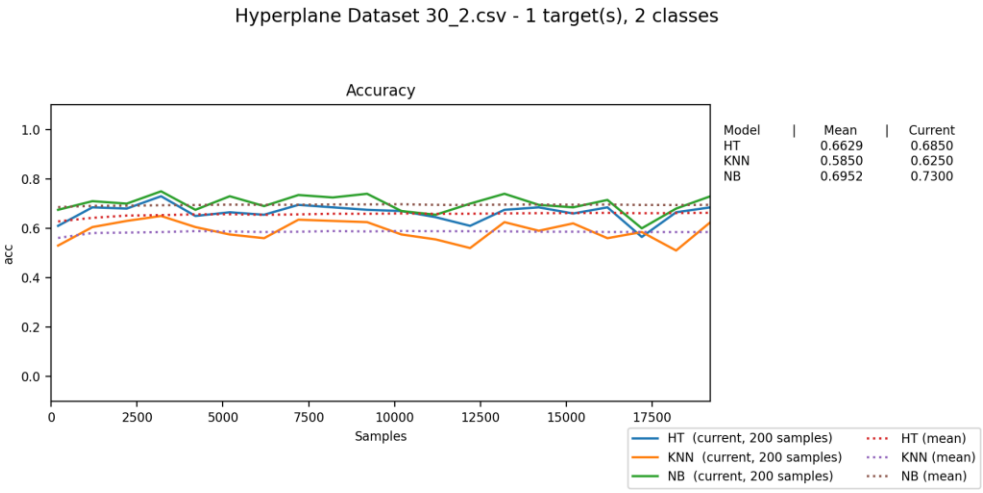

Figure 26: 1000 Batch Classifier Temporal Accuracy on Hyperplane Dataset 30_2

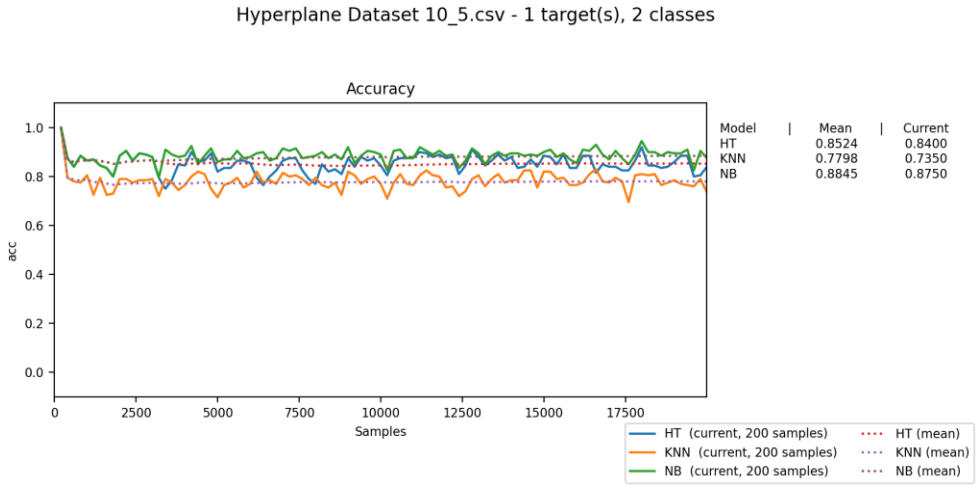

Figure 27: 1 Batch Classifier Temporal Accuracy on Hyperplane Dataset 10_5

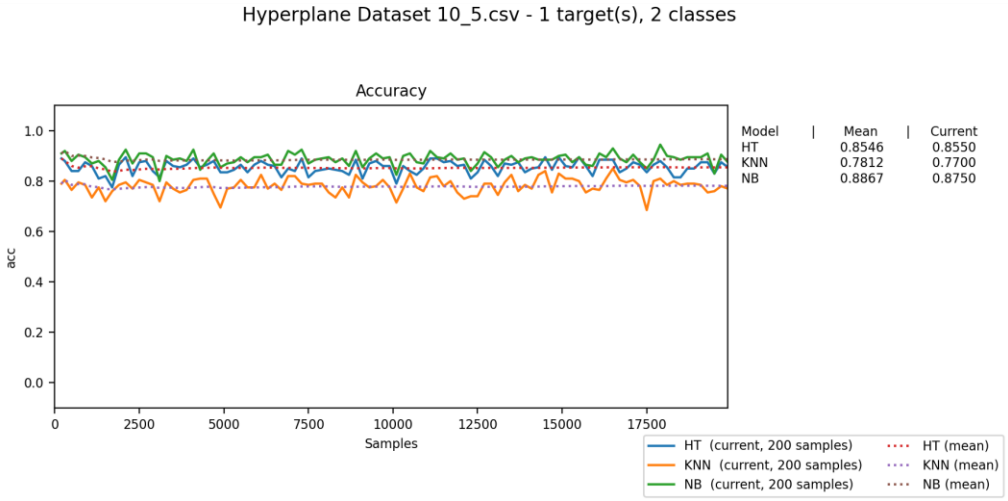

Figure 28: 100 Batch Classifier Temporal Accuracy on Hyperplane Dataset 10_5

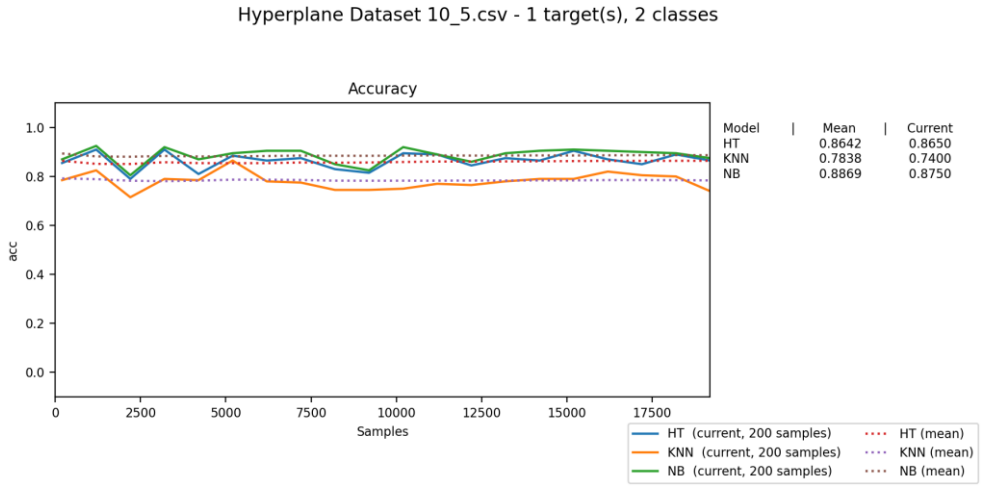

Figure 29: 1000 Batch Classifier Temporal Accuracy on Hyperplane Dataset 10_5

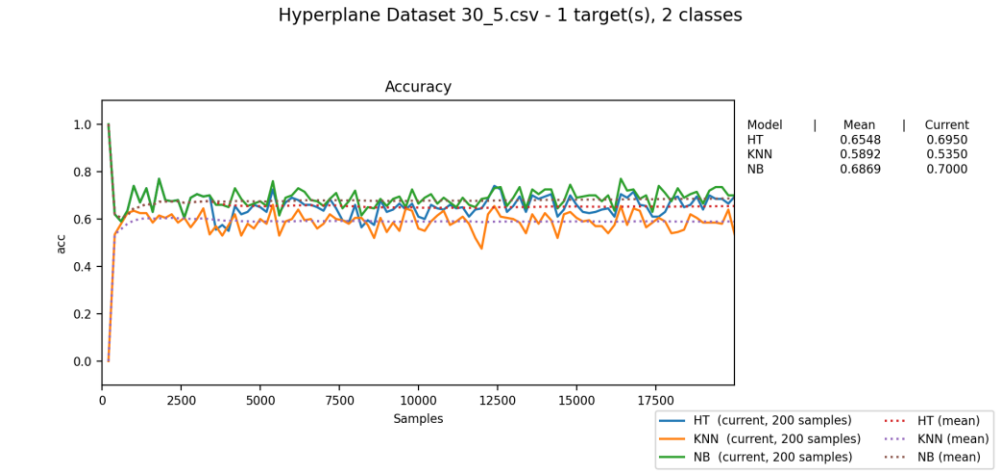

Figure 30: 1 Batch Classifier Temporal Accuracy on Hyperplane Dataset 30_5

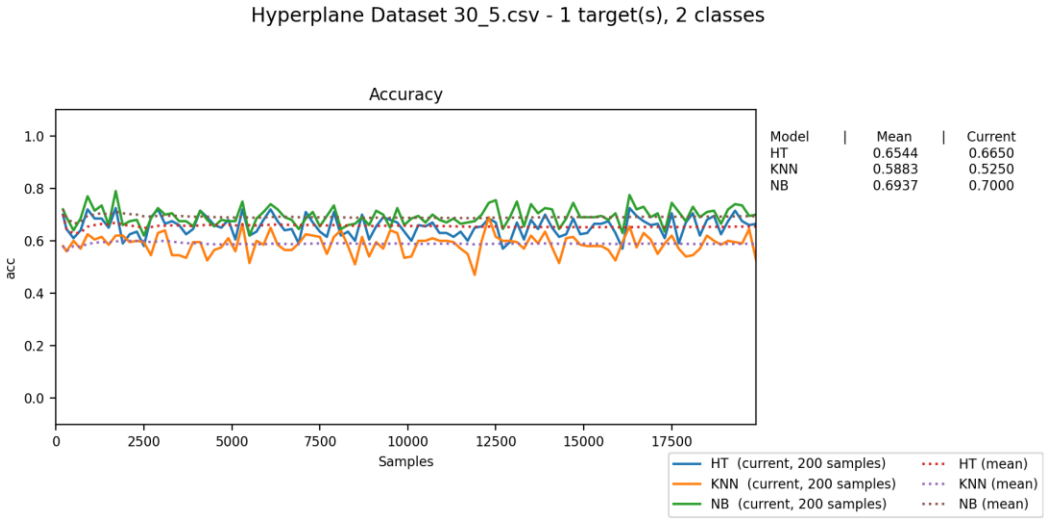

Figure 31: 100 Batch Classifier Temporal Accuracy on Hyperplane Dataset 30_5

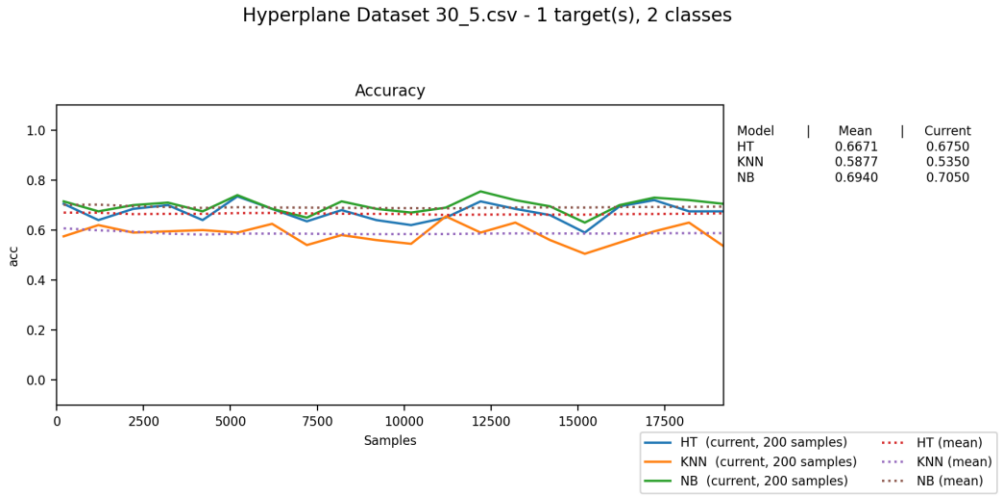

Figure 32: 1000 Batch Classifier Temporal Accuracy on Hyperplane Dataset 30_5

HT	Batch Size	Accuracy
Hyperplane Dataset 10_2	1	0.854
Hyperplane Dataset 10_2	100	0.862
Hyperplane Dataset 10_2	1000	0.864
Hyperplane Dataset 30_2	1	0.653
Hyperplane Dataset 30_2	100	0.657
Hyperplane Dataset 30_2	1000	0.662
Hyperplane Dataset 10_5	1	0.852
Hyperplane Dataset 10_5	100	0.854
Hyperplane Dataset 10_5	1000	0.864
Hyperplane Dataset 30_5	1	0.654
Hyperplane Dataset 30_5	100	0.654
Hyperplane Dataset 30_5	1000	0.667

Table 4: Accuracy of HT for different batch sizes on different datasets

As it can be seen from table 4, the accuracy rate generally increases for HT classifier when batch size increases. However, this increase may not be significant because it is approximately 0.2-0.3%.

KNN	Batch Size	Accurcy
Hyperplane Dataset 10_2	1	0.776
Hyperplane Dataset 10_2	100	0.777
Hyperplane Dataset 10_2	1000	0.774
Hyperplane Dataset 30_2	1	0.583
Hyperplane Dataset 30_2	100	0.583
Hyperplane Dataset 30_2	1000	0.585
Hyperplane Dataset 10_5	1	0.779
Hyperplane Dataset 10_5	100	0.781
Hyperplane Dataset 10_5	1000	0.783
Hyperplane Dataset 30_5	1	0.589
Hyperplane Dataset 30_5	100	0.588
Hyperplane Dataset 30_5	1000	0.587

Table 5: Accuracy of KNN for different batch sizes on different datasets

From the table 5, it can be said that increasing the batch size doesn't mean an increasing in accuracy for KNN classifier. In Hyperplane Dataset 10_2, increasing batch size from 100 to 1000 cause a decrease in accuracy. Same argument can be claimed for Hyperplane Dataset 30_5.

NB	Batch Size	Accurcy
Hyperplane Dataset 10_2	1	0.865
Hyperplane Dataset 10_2	100	0.866
Hyperplane Dataset 10_2	1000	0.867
Hyperplane Dataset 30_2	1	0.687
Hyperplane Dataset 30_2	100	0.695
Hyperplane Dataset 30_2	1000	0.695
Hyperplane Dataset 10_5	1	0.884
Hyperplane Dataset 10_5	100	0.886
Hyperplane Dataset 10_5	1000	0.886
Hyperplane Dataset 30_5	1	0.686
Hyperplane Dataset 30_5	100	0.693
Hyperplane Dataset 30_5	1000	0.694

Table 6: Accuracy of NB for different batch sizes on different datasets

As it can be seen from table 6, the accuracy rate generally increases for NB classifier when batch size increases. However, this increase may not be significant because it is approximately 0.2-0.3%. Also, it can be observed that at some points, despite of an increasing in batch size, accuracy stays same but never goes down.

From all these tables and figures, it can be concluded that batch size doesn't affect accuracy rate of the classifiers significantly. Although for different classifiers, different results can be observed, increasing and decreasing rate is so small, it can be neglected. So, it can be said that there is no correlation between batch size and classifiers.

d)

	HT	KNN	NB	MV	WMV
Hyperplane Dataset 10_2	0.86	0.77	0.88	0.82	0.85
Hyperplane Dataset 30_2	0.67	0.59	0.65	0.62	0.64
Hyperplane Dataset 10_5	0.85	0.78	0.82	0.82	0.86
Hyperplane Dataset 30_5	0.63	0.56	0.63	0.62	0.64

Table 7: Accuracy of Classifiers for different batch sizes on different datasets

From table 6, it can be seen that online ensemble classifiers do not perform better than single classifiers in terms of maximum accuracies. For hyperplane dataset 10_5 and 30_5, WMV slightly perform better than others, however, since the rate is small which is 1%, it can be consider insignificant. From this point of view, for higher drifting features, using WMV gives the better result while calculating the accuracy.

e) As a conclusion of given all tables, it can be claimed that batch size models with high batch size generally performs better than online classifiers. However, it is not the always the case. For instance, for Hyperplane Dataset 30_2, HT classifier with 1000 batch size doesn't perform better than online HT classifier. Also from the above table, it can be concluded that when the number of drifting feature increases, WMV which is an ensemble classifier starts to perform better than individual classifiers. As a result, the performance of different classifiers differs from dataset to dataset.

f) Combining models is one technique to increase the accuracy of predictive modeling. Also, if it is possible, including more data sample to the dataset can increase the accuracies. Besides, getting rid of outlier data might increase the accuracy rate since they might influence the overall performance. I used Stacking function to build a developed combining model. I tested with Hyperplane Dataset 10_2 and found out that, while, single online classifier HT had 86.11% accuracy rate and stacked model had 88.48%. This shows that prediction accuracy of the online classifiers can be improved.

Run-Time Check

	Online	Batch size 1	Batch size 100	Batch size 1000
Hyperplane Dataset 10_2	**37.44**	**46.53**	**7.36**	**6.70**
Hyperplane Dataset 30_2	**39.22**	**46.94**	**7.80**	**7.10**
Hyperplane Dataset 10_5	**42.78**	**52.42**	**7.34**	**7.14**
Hyperplane Dataset 30_5	**45.03**	**45.35**	**7.51**	**7.24**

Table 8: Efficiencies of Online and Batch size Classifiers on different datasets

According to this table, we can claim that batching process takes more time than working time of the online classifiers. However, after batching one time, using different batch sizes decreases the running time dramatically. Also, it can be said that if the dataset becomes more complicated, then the time spent for online classifiers increases. So, we can say that batch size affect the runtime.