Error Rate Table for dataset - au1_1000.arff and Iterations 30

| Classifier | Vannila | Bagging | Boosting |
|---|---|---|---|
| J48 | 0.2989 | 0.2975 | 0.2837 |
| Decision Stump | 0.3828 | 0.3776 | 0.3536 |
| Naive Bayes | 0.263 | 0.266 | 0.3553 |

Error Rate Table for dataset - au4_2500.arff and Iterations 30

| Classifier | Vannila | Bagging | Boosting |
|---|---|---|---|
| Naive Bayes | 0.347 | 0.3469 | 0.4017 |
| Decision Stump | 0.3617 | 0.2221 | 0.4263 |
| J48 | 0.2221 | 0.2456 | 0.2195 |

Error Rate Table for dataset - au6_250_drift_au6_cd1_500.arff and Iterations 30

| Classifier | Vannila | Bagging | Boosting |
|---|---|---|---|
| NAive Bayes | 0.213 | 0.2108 | 0.2115 |
| Decision Stump | 0.2099 | 0.2129 | 0.2122 |
| J48 | 0.2033 | 0.2101 | 0.1996 |

Error Rate Table for dataset - au1_1000.arff.arff and Iterations 100

| Classifier | Vannila | Bagging | Boosting |
|---|---|---|---|
| J48 | 0.2989 | 0.2995 | 0.2442 |
| Decision Stump | 0.3828 | 0.3781 | 0.4263 |
| Naive Bayes | 0.263 | 0.3567 | 0.3578 |

Error Rate Table for dataset - au4_2500.arff and Iterations 100

| Classifier | Vannila | Bagging | Boosting |
|---|---|---|---|
| Naive Bayes | 0.347 | 0.347 | 0.4017 |
| Decision Stump | 0.3617 | 0.3616 | 0.4263 |
| J48 | 0.2221 | 0.2496 | 0.185 |

Error Rate Table for dataset - au6_250_drift_au6_cd1_500.arff and Iterations 100

| Classifier | Vannila | Bagging | Boosting |
|---|---|---|---|
| NAive Bayes | 0.213 | 0.213 | 0.2115 |
| Decision Stump | 0.2099 | 0.2111 | 0.2122 |
| J48 | 0.2033 | 0.211 | 0.1932 |

Error Rate Table for dataset - au1_1000.arff.arff and Iterations 150

| Classifier | Vannila | Bagging | Boosting |
|---|---|---|---|
| J48 | 0.2989 | 0.3784 | 0.2377 |
| Decision Stump | 0.3828 | 0.3784 | 0.3535 |
| Naive Bayes | 0.263 | 0.3569 | 0.3578 |

Error Rate Table for dataset - au4_2500.arff and Iterations 150

| Classifier | Vannila | Bagging | Boosting |
|---|---|---|---|
| Naive Bayes | 0.347 | 0.3471 | 0.4017 |
| Decision Stump | 0.3617 | 0.3617 | 0.4263 |
| J48 | 0.2221 | 0.2497 | 0.1792 |

Error Rate Table for dataset - au6_250_drift_au6_cd1_500.arff and Iterations 150

| Classifier | Vannila | Bagging | Boosting |
|---|---|---|---|
| NAive Bayes | 0.213 | 0.2112 | 0.2115 |
| Decision Stump | 0.2099 | 0.2131 | 0.2122 |
| J48 | 0.2033 | 0.211 | 0.1992 |

# Bagging and Boosting

## Bagging:

It is a technique of learning many classifiers each with only a portion of data and combining them through model averaging technique.

In this project, Bagging Applied to datasets : au1_1000.arff, au4_2500.arff, au6_250_drift_au6_cd1_500.arff

The datasets were downloaded from Auto Union - UCI repository.

These datasets were loaded into WEKA explorer and the error rates for Bagging is noted.

From the observation of error rates of Bagging,

The best Learning Algorithm is j48 - au6_250_drift_au6_cd1_500.arff (numIterations = 30)

Assumption on Bagging role, for learning algorithm J48 to get a error rate - 0.2101

- Generally Bagging reduces overfitting by allowing the model to learn only a portion of original data.
  By doing so, the model cannot learn complete information about the original data set and fit the perfect decision boundary to classify the samples. More the Reduction in the Overfitting problem, less the error rate.
- Bagging works wells on high complex dataset. And the dataset - au6_250_drift_au6_cd1_500.arff has more than 100 features. This reduces the high variance and the bias remain unchanged.

# Boosting:

Boosting is a collection of regression predictors. It converts the early learners or a sequence of weak learners into a very complex predictor to increase the complexity of the particular model.

Ada Boost = Adaptive Boosting

Adaboost is efficient boosting method which can sample data from data distribution and improve the performance of learning algorithm.

From the observation of error rates of AdaBoost,

The best Learning Algorithm is j48 - au4_2500.arff (numIterations = 150)

Assumption on AdaBoost role, for learning algorithm J48 to get a error rate - 0.1792

- Generally Boosting reduces underfitting by sampling data distribution
- Bagging works wells on simple dataset. And the dataset - au6_250_drift_au6_cd1_500.arff has more than 20 features. This reduces the large bias and the variance remains unchanged.

Change in Bias, Variance of the learning algorithms:

If the model is too simple and has very low few parameters then underfitting problem occurs, which means large bias and small variance.

If the model is too complex and has very high parameters then overfitting problem occurs, which means large variance and small bias.

Inference from the Error Rates:

- The error rates of vanilla classifiers is greater than Bagging in au4_2500.arff dataset. Therefore, the bagging with 10-fold cross validation, reduces the variance and improves the efficiency of learning algorithm. Hence the learning model has low variance and low bias.

- The error rates of Vanilla classifiers and Bagging are quite similar in au1_1000.arff data set. Therefore the bagging doesn't improve the efficiency of the learning algorithm. This might be due to use of only one classifier in the bagging. Moreover, the au4_2500.arff dataset has less

features and bagging works well only in complex dataset. Generally more the number of classifiers, lesser the error rate is. Here, the learning model has low bias and high variance.

- Similarly, The error rates of Boosting model is lesser than vanilla classifiers in au6_250_drift_au6_cd1_500.arff dataset. This shows that, adaptive boosting improves the efficiency of the learning algorithm by sampling the data from data distribution. Hence, the outcome of the learning model is low bias and low variance.
- There are certain cases where bagging and boosting have high error rates than vanilla classifiers. This concludes certain domains are unbiased and bagging/boosting doesn't play any role in improving the efficiency of the learning model.