Assignment 4

Overview

Methods: Logistic Regression, k-Neighbors, Bagging, Random Forests, AdaBoost

Packages Used: stats, class, adabag, randomForest

Finding best classifiers

Logistic Regression

Experiment #	Classifier	Cross-Validation Fold	Parameter 1, Distribution	Average Accuracy as RMSE
1	KNN	20	poisson	0.5024818
2	KNN	20	binomial	0.2276561
3	KNN	20	gaussian	0.4504893
4	KNN	20	Quasi	0.4504893
5	KNN	20	Quasibinomial	0.2276561

For logistic regression the best values were obtained when using binomial or quasibinomial distribution as a parameter.

kNN

Experiment #	Classifier	Cross-Validation Fold	Parameter 1, K	Average Accuracy
1	KNN	20	30	0.9712264
2	KNN	20	50	0.9693553
3	KNN	20	20	0.9721639
4	KNN	20	10	0.9718534
5	KNN	20	10	0.9718514

As we can see the best parameter for kNN is k=20.

Bagging

Experiment #	Classifier	Cross-Validation Fold	Parameter 1, rpart minSplit	Parameter 2, mfinal	Average Accuracy as error rate
1	Bagging	20	0	5	0.01912736
2	Bagging	20	0	30	0.01435183

3	Bagging	20	10	15	0.01527136
4	Bagging	20	20	25	0.03242207
5	Bagging	20	10	30	0.014625

As we can see the best parameter for Bagging is minsplit = 0, and mfinal =30. MinSplit is the minimum split for a tree and mfinal is the total number of trees.

Boosting

Experiment #	Classifier	Cross-Validation Fold	Parameter 1, rpart minSplit	Parameter 2, rpart mfinal	Parameter 3, Learning Coefficient	Average Accuracy as error rate
1	Boosting	20	0	30	Breiman	0.01373395
2	Boosting	20	10	30	Breiman	0.01805778
3	Boosting	20	0	50	Breiman	0.013534
4	Boosting	20	0	50	Freund	0.0222154
5	Boosting	20	10	30	Freund	0.06834558

As we can see the best parameter for boosting is when we use a Breiman learning coefficient and mfinal is 50 and minSplit is 0. mfinal is the number of trees and minSplit is the minimum number of classes to make a split.

RandomForest

Experiment #	Classifier	Cross-Validation Fold	Parameter 1, Number of trees	Parameter 2, Max. no. of nodes in tree	Parameter 3, predictors sampled for splitting	Average Accuracy as success rate
1	RandomForect	20	5	4	10	0.4633013
2	RandomForect	20	10	4	10	0.7583892
3	RandomForect	20	15	40	100	0.9796875
4	RandomForect	20	35	40	100	0.9746875
5	RandomForect	20	30	50	100	0.9759375

For RandomForest the best methodology is when we use Number of Trees = 15, Max no of ndes =40 and predictors sampled =100.

Training and Testing methodology

To test the performance of the classifiers, I used 20 fold cross validation. The average values are as reported in the earlier section

Pseudocode:

```
folds←cut(seq(1,nrow(training_values)),breaks=num_of_breaks,labels=FALSE)
for(i in num_of_breaks)
{
Do Algorithm
}
```

Besides accuracy as mean error rate and mean accuracy, I also implemented ROC and Area Under ROC Curve(AUC).

Final Report

Classifier	Technique	Accuracy	AUC for ROC
KNN	20-fold C.V. Avg	97.37166	0.5
Boosting	20-fold C.V. Avg	99.1375	0.8335
Bagging	20-fold C.V. Avg	98.57390	0.715
RandomForest	20-fold C.V. Avg	97.40625	0.7227273
Logistic Regression	20-fold C.V. Avg	78.92059	0.5384615

Analysis

The best classifiers were the ones that were based on decision trees. This is in agreement of the previous assignment that used the same data set. This is because the current data set is based on chess moves. Any winning strategy for chess follows a long if-else statement that can be easily learned by decision trees.

The highest performers were RandomForests, Bagging, Boosting and kNN. The relatively high performance of kNN is due to the fact that it favors data sets that are clustered, similar to this one.

The worst performer was the generalized Linear Model. This is because the data set is not at all linearly separable. When using binomial distribution, we got a better result. But overall it performed poorly in comparison to the other classifiers.

Receiver Operating Characteristics(ROC)graphs are useful for organizing and visually estimating classifiers. It was helpful in determining the trade-off between hit rates and false alarm rates. Although such an analysis would be in useful in many datasets, it wasn't so valid in ours. This was because,

1. We already had high accuracy rates.
2. There is a high trade-off between false positives and true positives.

In data in the medical field or other such analysis there isn't a high cost for a false positive diagnosis. But in chess, there is, as winning is the aim. In both however, true negatives are to be avoided. Hence, accuracy was a better metric in this case.