COMP30120 – Assignment 3

Student ID: 14210231

Student Name: Felipe Guth

1. Evaluate the performance of two individual classifiers on the data: KNN (k=2) and Decision Trees (J48).

Knn, k =2

Correctly Classified Instances %	67
Incorrectly Classified Instances	33
Precision	0.735
Recall	0.67
TP rate	0.67
FP rate	0.27

J48

Correctly Classified Instances %	70
Incorrectly Classified Instances	30
Precision	0.71
Recall	0.7
TP rate	0.7
FP rate	0.34

The classifiers had a similar performance with the Decision tree classifier having a slightly better accuracy, 3%, over the KNN.

2. Apply ensembles with bagging using each of the classifiers from Task 1. Investigate the performance of both classifiers as the ensemble size increases, in steps from 10 members to 100 members. Also investigate how changing the number of instances in the bootstrap samples affects classification performance (i.e. the "bag size").

N = ensemble size.

Knn, k = 2						
	10 N	**30 N**	**50 N**	**70 N**	**90 N**	**100 N**
Correctly Classified Instances %	75	76	77	77	76	78

J48						
	10 N	**30 N**	**50 N**	**70 N**	**90 N**	**100 N**
Correctly Classified Instances %	76	81	80	77	79	79

The increase of numbers of instance have not implied in better performance in all cases. The ensembles of 50, 30 and 40 instances had the better performances in the J48 while in the KNN the 100 instances had a slightly better accuracy than the 50 instances ensemble.

Bagsize >	**20**	**50**	**70**	**100**	
Correctly Classified Instances %	77	80	80	79	**100 N j48**
Correctly Classified Instances %	82	77	79	78	**100 N knn (k =2)**

The changing in the number of instance in the bootstrap samples generated better performances in the classifiers using 100 instance ensembles. Using 50% of bag size bootstrap the KNN was able to achieve 80% of accuracy. The J48 was able to achieve 82% of accuracy, as the bag size augmented, the accuracy was slightly reduced, having 79% and 78% accuracy using 70% and 100% of bag size from the data.

3. Apply ensembles with boosting using each of the classifiers from Task 1. Investigate the performance of both classifiers as the ensemble size increases, in steps from 10 members to 100 members.

N = ensemble size.

Knn, k = 2						
	10 N	**30 N**	**50 N**	**70 N**	**90 N**	**100 N**
Correctly Classified Instances %	69	69	69	69	69	69

J48						
	10 N	**30 N**	**50 N**	**70 N**	**90 N**	**100 N**
Correctly Classified Instances %	69	76	76	78	78	81

4. Apply ensembles with random subspacing using each of the classifiers from Task 1. Investigate the performance of both classifiers as the ensemble size increases, in steps from 10 members to 100 members. Also investigate how changing the number of features used when applying random subspacing affects classification performance (i.e. the “subspace size”).

N = ensemble size.

Knn, k = 2						
Subspace size	**10 N**	**30 N**	**50 N**	**70 N**	**90 N**	**100 N**
0.35	74	80	79	80	81	81
0.7	76	78	79	79	79	78
1	68	69	66	65	64	65

J48						
Subspace size	**10 N**	**30 N**	**50 N**	**70 N**	**90 N**	**100 N**
0.35	74	76	74	76	75	77
0.7	69	72	71	72	74	74
1	68	66	66	66	66	66

5. Summarise the differences in the classification performance results from Tasks 1-4, discussing the factors which might explain these differences.

The classification accuracy of the classifiers in a single instance were of 67 % for KNN (k=2) and 70% for Decision tree J48. These results were improved by using ensembles containing multi instances of the same classifiers, the performances were affected by parameters of ensemble size and techniques applied.

The bagging with bootstrap aggregation trains different classifiers on different subsets of the original data. This classifier had an accuracy of 75, 76, 77, 77, 76 and 78% for KNN and 76, 81, 80, 77, 79 and 79% for decision tree (J48). Ensembles size were equal to 10, 30, 50, 70, 90 and 100 instances. The alteration of bagging size, which configures the use of a portion of the data to construct the bootstrap, also was tested. KNN had an accuracy of 77, 80, 80 and 79% whereas decision tree (J48) had accuracy of 82, 77, 79 and 78%. The bag sizes were set to 20, 50, 70 and 100% of the data, respectively, using ensembles of 100 instances.

Boosting method had an accuracy of 69% for all tests using 10, 30, 50, 70, 90 and 100 instances on KNN (k=2). Using the same number instances of ensembles the decision tree (J48) had an accuracy of 69, 76, 76, 78, 78 and 81%. This difference might be explained due the fact that decision trees are more sensible to alteration in the data than KNN.

Finally, random subspacing method was tested, the results are shown in the next tables (as in question 4).

Knn, k = 2						
Subspace size	**10 N**	**30 N**	**50 N**	**70 N**	**90 N**	**100 N**
0.35	74	80	79	80	81	81
0.7	76	78	79	79	79	78
1	68	69	66	65	64	65

J48						
Subspace size	**10 N**	**30 N**	**50 N**	**70 N**	**90 N**	**100 N**
0.35	74	76	74	76	75	77
0.7	69	72	71	72	74	74
1	68	66	66	66	66	66

The best results were achieved using a high number of ensembles and low subspace size (number of features taken). The use of a higher number of features not necessarily helps to better classify the data. In fact, the more features are taken into account, the more the data tends to lose the variance.

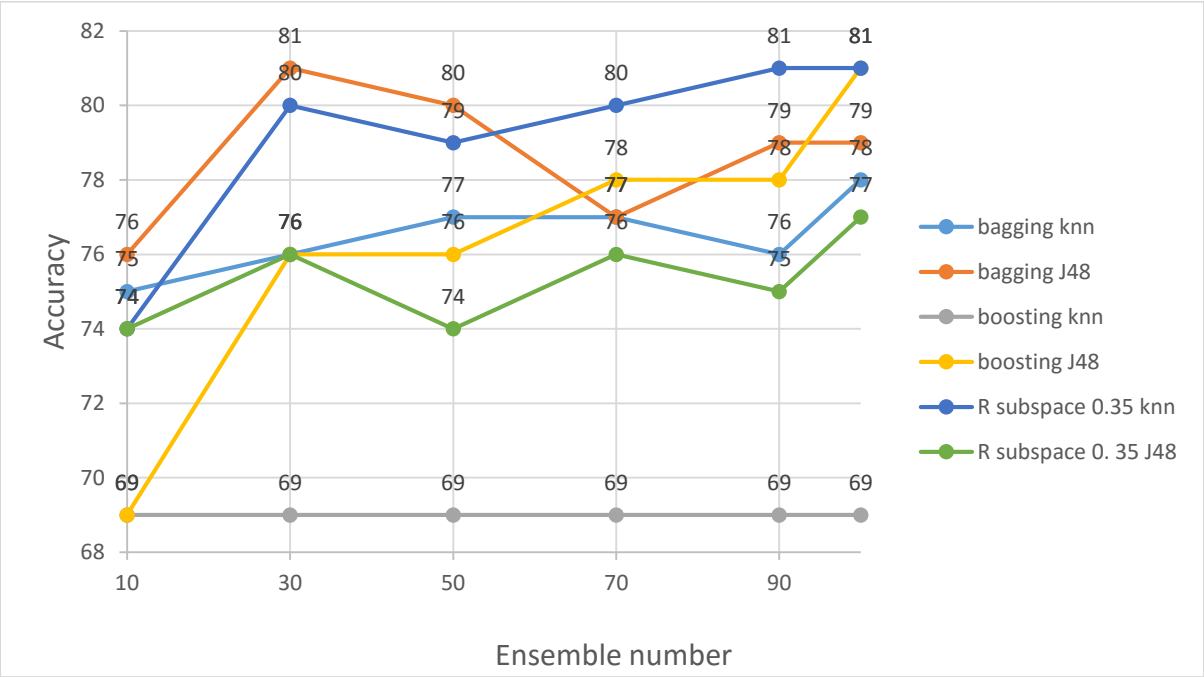

Figure 1 – Classifiers performance

The Figure 1, shows the comparison of performance on the classifiers. The results present significant variance according to the parameters. Bagging ensembles with random subsample tend to work better with sensible classifiers, sensible to data variance as decision trees, this is shown on the graph comparing bagging using KNN and J48 decision tree. Boosting focus on errors to improve the ensemble classification, using hard samples to adjust weights and improve the classification results. Using KNN as classification method, the results were constant while J48 decision trees varied and produced good results. This fact may also be attributed to the sensitivity of decision trees to the data variance compared to KNN, which requires greater variance to produce different classification results. Using random subspacing ensembles the better results were obtained using 35% of features and a large number of ensembles. Since lesser number of features produce more variance and a larger number of instances produces higher diversity, this combination was proved to achieve a satisfying performance.