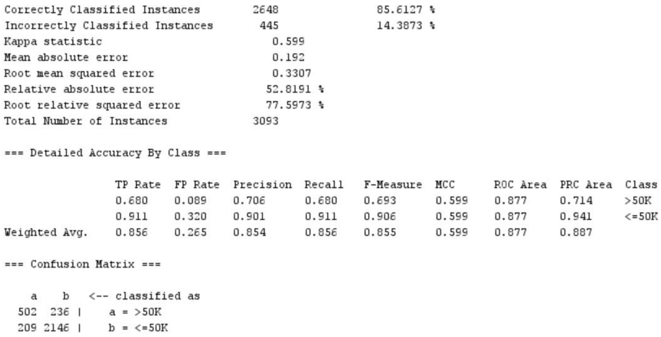
BINGBING LI	Machine Learning COMP47490	27.11.2019
18210705	Assignment 2

**Question 1**

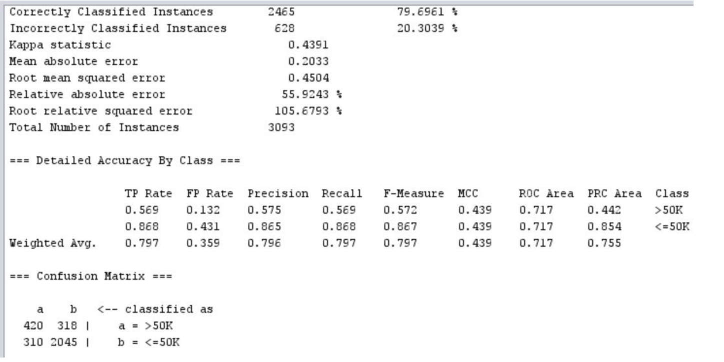
**(a) Evaluate the performance of two basic classifiers on your dataset: Decision Trees (J48) and 1-NN. Carefully consider the evaluation measure(s) that you use for this exercise and justify why you selected the particular evaluation measure(s).**

Decision Trees(J48):



```
Correctly Classified Instances        2648               85.6127 %
Incorrectly Classified Instances       445               14.3873 %
Kappa statistic                          0.599
Mean absolute error                      0.192
Root mean squared error                  0.3307
Relative absolute error                 52.8191 %
Root relative squared error             77.5973 %
Total Number of Instances             3093

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.680    0.089    0.706      0.680    0.693      0.599    0.877     0.714     >50K
                 0.911    0.320    0.901      0.911    0.906      0.599    0.877     0.941     <=50K
Weighted Avg.    0.856    0.265    0.854      0.856    0.855      0.599    0.877     0.887

=== Confusion Matrix ===

    a    b   <-- classified as
  502  236 |    a = >50K
  209 2146 |    b = <=50K
```

1-NN:



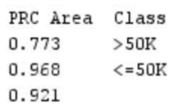
```
Correctly Classified Instances        2465               79.6961 %
Incorrectly Classified Instances       628               20.3039 %
Kappa statistic                          0.4391
Mean absolute error                      0.2033
Root mean squared error                  0.4504
Relative absolute error                 55.9243 %
Root relative squared error            105.6793 %
Total Number of Instances             3093

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.569    0.132    0.575      0.569    0.572      0.439    0.717     0.442     >50K
                 0.868    0.431    0.865      0.868    0.867      0.439    0.717     0.854     <=50K
Weighted Avg.    0.797    0.359    0.796      0.797    0.797      0.439    0.717     0.755

=== Confusion Matrix ===

    a    b   <-- classified as
  420  318 |    a = >50K
  310 2045 |    b = <=50K
```

Because the dataset is imbalanced, when dealing with imbalanced classes, precision and recall are better metrics than accuracy, in the same way, for imbalanced datasets a Precision-Recall curve is more suitable than a ROC curve. So we can use the PRC Area as the evaluate method. As shown in the figure above, the PRC Area in decision Trees is 0.714 and 0.941, comparing with the 1-NN method which is 0.442 and 0.854, we can see that the 1-NN method can't handle the attributes which is >50 k well. So the decision trees is a better method.

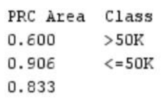
**(b) Apply ensembles with bagging using both classifiers from Task (a). Investigate the performance of both classifiers as the ensemble size increases (e.g., in steps of 20 from 20 to 100 members). Using the best performing ensemble size, investigate how changing the number of instances in the bootstrap samples affects classification performance (i.e. the "bag size").**

Decision trees:	1-NN:

Ensemble size =20:

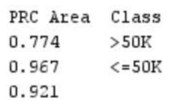


```
PRC Area  Class
0.773     >50K
0.968     <=50K
0.921
```

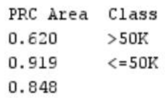


```
PRC Area  Class
0.600     >50K
0.906     <=50K
0.833
```

Ensemble size =40:

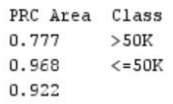


```
PRC Area  Class
0.774     >50K
0.967     <=50K
0.921
```

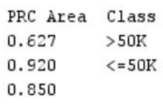


```
PRC Area  Class
0.620     >50K
0.919     <=50K
0.848
```

Ensemble size =60:

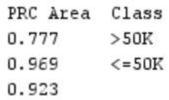


```
PRC Area  Class
0.777     >50K
0.968     <=50K
0.922
```

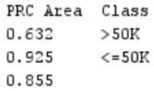


```
PRC Area  Class
0.627     >50K
0.920     <=50K
0.850
```

Ensemble size =80:

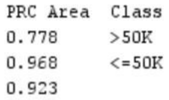


```
PRC Area  Class
0.777     >50K
0.969     <=50K
0.923
```

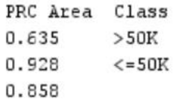


```
PRC Area  Class
0.632     >50K
0.925     <=50K
0.855
```

Ensemble size =100:



```
PRC Area  Class
0.778     >50K
0.968     <=50K
0.923
```

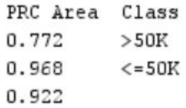


```
PRC Area  Class
0.635     >50K
0.928     <=50K
0.858
```

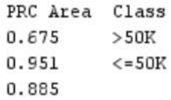
From the results we get, for both classifiers, we can know that as the ensemble size increases, the better the classifier performs. So I use the last one to research later:
I change the bag size percent in steps of 20 from 20 to 100 members, the results are as follows:

Decision trees: 1-NN:

Bag size percent=20：

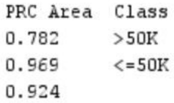


```
PRC Area  Class
0.772     >50K
0.968     <=50K
0.922
```

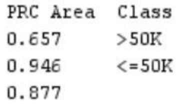


```
PRC Area  Class
0.675     >50K
0.951     <=50K
0.885
```

Bag size percent=40：

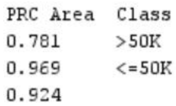


```
PRC Area  Class
0.782     >50K
0.969     <=50K
0.924
```

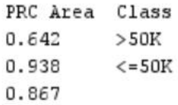


```
PRC Area  Class
0.657     >50K
0.946     <=50K
0.877
```

Bag size percent=60：

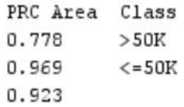


```
PRC Area  Class
0.781     >50K
0.969     <=50K
0.924
```

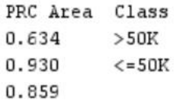


```
PRC Area  Class
0.642     >50K
0.938     <=50K
0.867
```

Bag size percent=80：

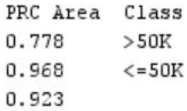


```
PRC Area  Class
0.778     >50K
0.969     <=50K
0.923
```

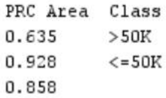


```
PRC Area  Class
0.634     >50K
0.930     <=50K
0.859
```

Bag size percent=100：



```
PRC Area  Class
0.778     >50K
0.968     <=50K
0.923
```



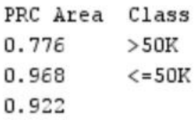
```
PRC Area  Class
0.635     >50K
0.928     <=50K
0.858
```

From the results we can see that as the bagsizepercent increases there is not much difference in decision tree , but the performance of classifier 1-NN gets worse.

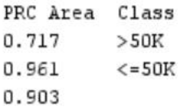
**(c) Apply ensembles with random subspacing using both classifiers from Task (a). Investigate the performance of both classifiers as the ensemble size increases (e.g., in steps of 20 from 20 to 100 members). Using the best performing ensemble size, investigate how changing the number of features used when applying random subspacing affects classification performance (i.e. the "subspace size").**

Decision trees: 1-NN:

Ensemble size =20:

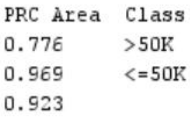


```
PRC Area  Class
0.776     >50K
0.968     <=50K
0.922
```

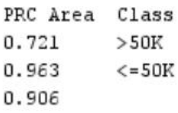


```
PRC Area  Class
0.717     >50K
0.961     <=50K
0.903
```

Ensemble size =40:

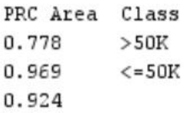


```
PRC Area  Class
0.776     >50K
0.969     <=50K
0.923
```

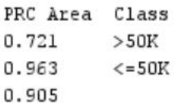


```
PRC Area  Class
0.721     >50K
0.963     <=50K
0.906
```

Ensemble size =60:

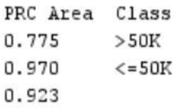


```
PRC Area  Class
0.778     >50K
0.969     <=50K
0.924
```

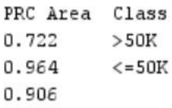


```
PRC Area  Class
0.721     >50K
0.963     <=50K
0.905
```

Ensemble size =80:

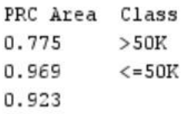


```
PRC Area  Class
0.775     >50K
0.970     <=50K
0.923
```

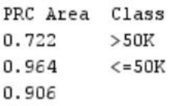


```
PRC Area  Class
0.722     >50K
0.964     <=50K
0.906
```

Ensemble size =100:



```
PRC Area  Class
0.775     >50K
0.969     <=50K
0.923
```

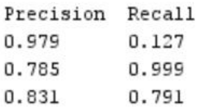


```
PRC Area  Class
0.722     >50K
0.964     <=50K
0.906
```

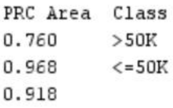
From the result, we can see that as the ensemble size increase, the performance of these two did not change obviously, so I use the ensemble size=100 to do the later research:
In this question, for decision tree, because the PRC area in the results are all equal to 0.923, so I use the precision and recall as the evaluation method, and for 1-NN, I use the PRC area as the evaluation method：

Decision trees: 1-NN:

subspace size=0.2:

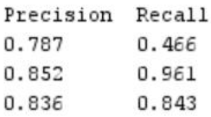


```
Precision  Recall
0.979      0.127
0.785      0.999
0.831      0.791
```

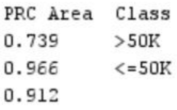


```
PRC Area  Class
0.760     >50K
0.968     <=50K
0.918
```

subspace size=0.4:

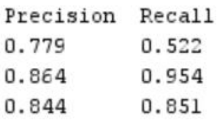


```
Precision  Recall
0.787      0.466
0.852      0.961
0.836      0.843
```

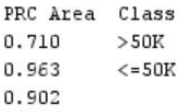


```
PRC Area  Class
0.739     >50K
0.966     <=50K
0.912
```

subspace size=0.6:

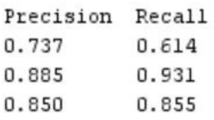


```
Precision  Recall
0.779      0.522
0.864      0.954
0.844      0.851
```

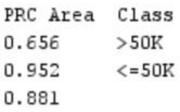


```
PRC Area  Class
0.710     >50K
0.963     <=50K
0.902
```

subspace size=0.8:



```
Precision  Recall
0.737      0.614
0.885      0.931
0.850      0.855
```



```
PRC Area  Class
0.656     >50K
0.952     <=50K
0.881
```

From the results we can see that as the subspace size increases, the performance of decision tree gets better but the 1-NN gets worse.

**(d) Which set of classifiers is expected to benefit from bagging techniques more and which set of classifiers is expected to benefit from random subspacing techniques more? For your dataset, determine the best ensemble strategy for each of the two classifiers.**

The decision tree classifier is suitable for both techniques, while the 1-NN classifier benefit from subspacing techniques. From the results above, I think the best ensemble strategy for decision tree is with subspacing technique, when ensemble size=100, subspace size=0.8.
And the best ensemble strategy for 1-NN is with subspacing technique, when ensemble size=100, subspace size=0.2.

**Question 2**

**(a) Explain the precision-recall tradeoff.**

Because as the precision increase, the recall will be worse. In some circumstance, the most search strategies can be adjusted to increase the value of precision at the expense of recall, or need the recall to be greater at the expense of precision. So choosing a better strategy should take the precision-recall tradeoff into consideration. A Precision-Recall Curve is an evaluation method which illustrates the Precision-Recall Tradeoff for a particular search method.

**(b) Explain the bias-variance tradeoff.**

A classifier can make errors, we can view the errors of a classifier on s given dataset in some ways: bias and variance are the two of them. Bias explains how close the classifier’s predictions are from the correct values, and variance explains error from sensitivity to small changes in the training set. Low bias but high variance classifiers tend to overfit, while high bias but low variance classifiers tend to underfit.
If our model is too simple and has very few parameters then the bias may be high and variance would be low. On the other hand if our model has large number of parameters then the bias may be low and variance may be low. So we need to find the right balance of bias-variance tradeoff without overfitting and underfitting the data.

**(c) Why do we need activation function in a perceptron?**

The activation function does the non-linear transformation to the input to make it to be able to learn and perform more complex tasks. In a perception, it is used to map the input between the required values like (0, 1) or (-1, 1).

**(d) What does R2 measure represent in the case of linear regression?**

R2 is the proportion of variation in the outcome Y, explained by the covariates X, is commonly described as a measure of goodness of fit. It measures how close the observed Y values are to the predicted (fitted) values from the model.