Project Report for Classification on Haberman’s Survival Dataset

Sharon Saronian

In this project we will do classification on Haberman’s Survival dataset by using Decision Tree, Naïve Bayes and Ensemble techniques. We also will consider different evaluation techniques to test the model performance and also do hyperparameter tuning using 10-fold Cross Validation for the project.

1. Dataset information

This dataset contains information from a study at University of Chicago’s Billings Hospital on the survival of patients that underwent surgery for breast cancer.

There are 3 attributes, all numerical, in the following form:

1. Age of patient at the time of operation
2. Patient’s year of operation
3. Number of positive axillary nodes detected

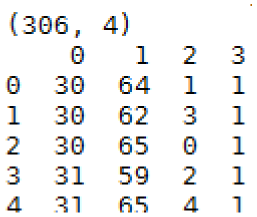
There is one class attribute, Survival Status, which is binary:

- 1 = The patient survived 5 years or older
- 0 = The patient died within 5 years

According to the metadata, there are 306 instances, and there are no missing values.

2. Classification

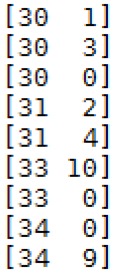
First let’s see what our dataset looks like:



```
(306, 4)
    0   1  2  3
0  30  64  1  1
1  30  62  3  1
2  30  65  0  1
3  31  59  2  1
4  31  65  4  1
```

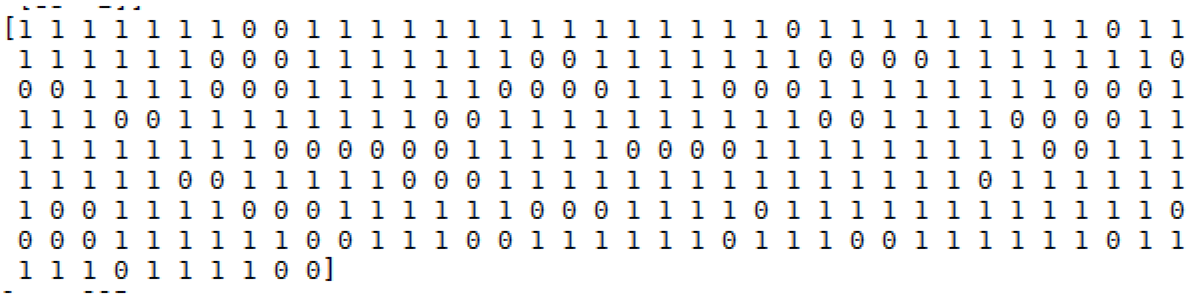
This confirms that we have 306 instances and 4 columns, and the first 5 instances are shown.

For our classification purposes, as we're going to use the DT classifier, we need to consider attributes Age and Number of nodes only, and the Year attribute should be omitted because Decision Trees give false results when we use attributes that have many, many different values with the same distribution, such as date or year. So, after omitting the second attribute and splitting the dataset to features and targets we have x in the form of:



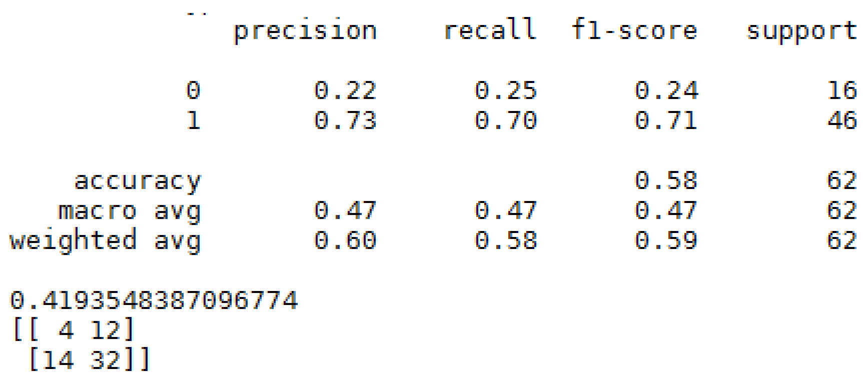
```
[30  1]
[30  3]
[30  0]
[31  2]
[31  4]
[33 10]
[33  0]
[34  0]
[34  9]
```

And y in the form of:



```
[1 1 1 1 1 1 1 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 0 1 1
 1 1 1 1 1 1 0 0 0 1 1 1 1 1 1 1 0 0 1 1 1 1 1 1 1 0 0 0 0 1 1 1 1 1 1 1 0
 0 0 1 1 1 1 0 0 0 1 1 1 1 1 1 0 0 0 0 1 1 1 0 0 0 1 1 1 1 1 1 1 1 0 0 0 1
 1 1 1 0 0 1 1 1 1 1 1 1 1 0 0 1 1 1 1 1 1 1 1 1 1 0 0 1 1 1 1 0 0 0 0 1 1
 1 1 1 1 1 1 1 1 0 0 0 0 0 0 1 1 1 1 1 0 0 0 0 1 1 1 1 1 1 1 1 1 0 0 1 1 1
 1 1 1 1 1 0 0 1 1 1 1 1 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1
 1 0 0 1 1 1 1 0 0 0 1 1 1 1 1 1 0 0 0 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 0
 0 0 0 1 1 1 1 1 1 0 0 1 1 1 0 0 1 1 1 1 1 1 0 1 1 1 0 0 1 1 1 1 1 1 0 1 1
 1 1 1 0 1 1 1 1 0 0]
```

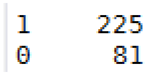
Now, we will split the dataset to train and test sets with test size of 0.2, build a Decision Tree classifier, fit the model on the train set, predict the model using the test set, then calculate the confusion matrix, accuracy, precision, recall, F1-score, error, sensitivity and specificity to gain an insight on the model performance:



```
              precision    recall  f1-score   support

           0       0.22      0.25      0.24        16
           1       0.73      0.70      0.71        46

    accuracy                           0.58        62
   macro avg       0.47      0.47      0.47        62
weighted avg       0.60      0.58      0.59        62

0.41935483870967 74
[[ 4 12]
 [14 32]]
```

As we can see, the accuracy is 0.58, the error is 0.41, the precision, recall, and F1-scores are 0.47, the sensitivity is 0.25, and the specificity is 0.70. We can see from the confusion matrix that the model has classified 36 samples correctly, but has misclassified 26 samples.

From the results we can see that the model’s performance is not so well. To find out why, we will count the labels:



```
1    225
0     81
```

We can see that 225 samples belong to the class 1, and only 81 samples belong to class 2. That means that are dataset is imbalanced which could be the reason for the poor performance of the model.

In order to increase the performance of the model, two things are done. First, we will use 10-fold cross validation to obtain the optimal depth of the tree. Then we will use three ensemble techniques (Random Forest, Bagging, and Adaboost) on the decision tree and compare the results.

Note that because we have imbalanced class distributions, F1 score is more reliable than accuracy in time of comparisons on different models. Therefore, although all performance measures are calculated, we will use F1 score to do the comparisons in this report.

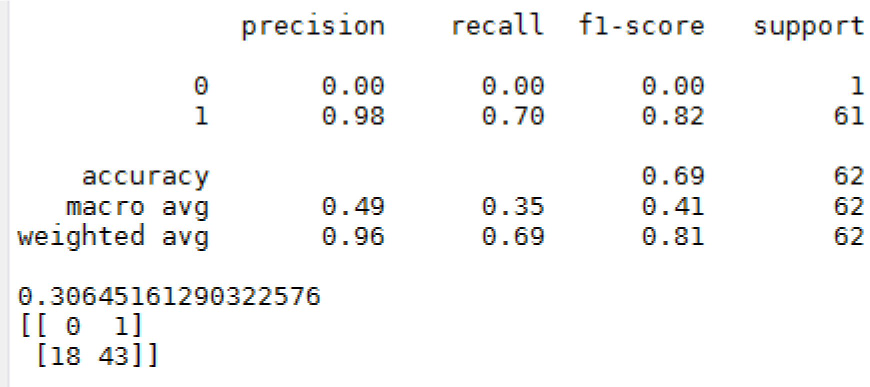
Another thing to note is that the Naïve Bayes classifier works well for imbalanced datasets as well, even without the ensemble techniques, therefore, along the comparisons we will give results for Naïve Bayes as well and compare it with the ensemble techniques.

The optimal depth obtained by cross validation is 2. So, that will be used in the ensemble algorithms.

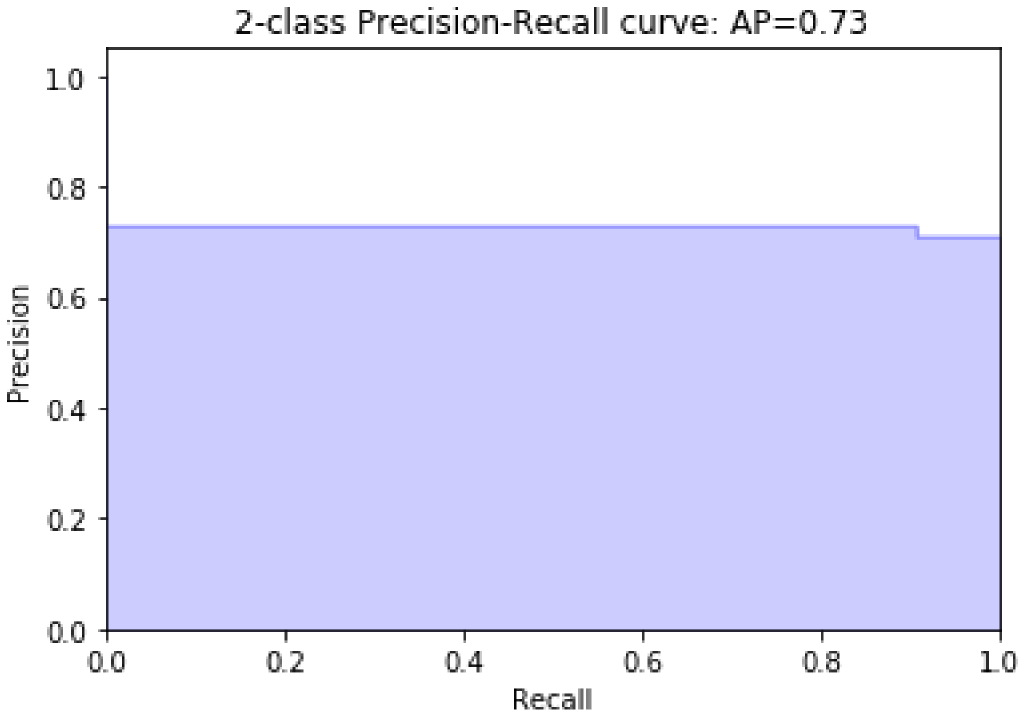
The base algorithm used in the ensemble models will be the decision tree classifier, and 50 estimators will be used to train the model. The results are shown below.

One reason to use Precision-Recall curves instead of ROC curves is that ROC curves are used for balanced datasets, whereas Precision-Recall curves are used in imbalanced ones.

Random Forest Classifier:

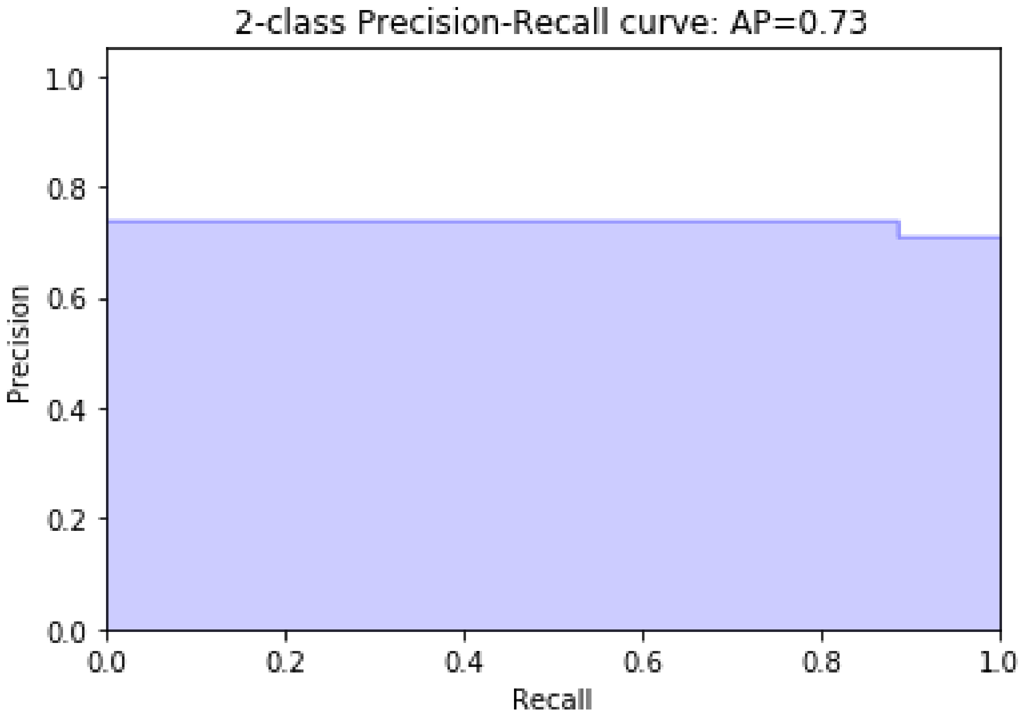


```
              precision    recall  f1-score   support

           0       0.00      0.00      0.00         1
           1       0.98      0.70      0.82        61

    accuracy                           0.69        62
   macro avg       0.49      0.35      0.41        62
weighted avg       0.96      0.69      0.81        62

0.30645161290322576
[[ 0  1]
 [18 43]]
```

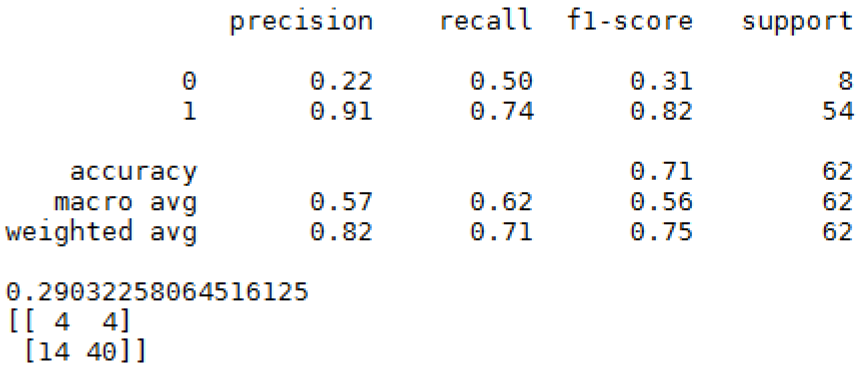


Bagging Classifier:

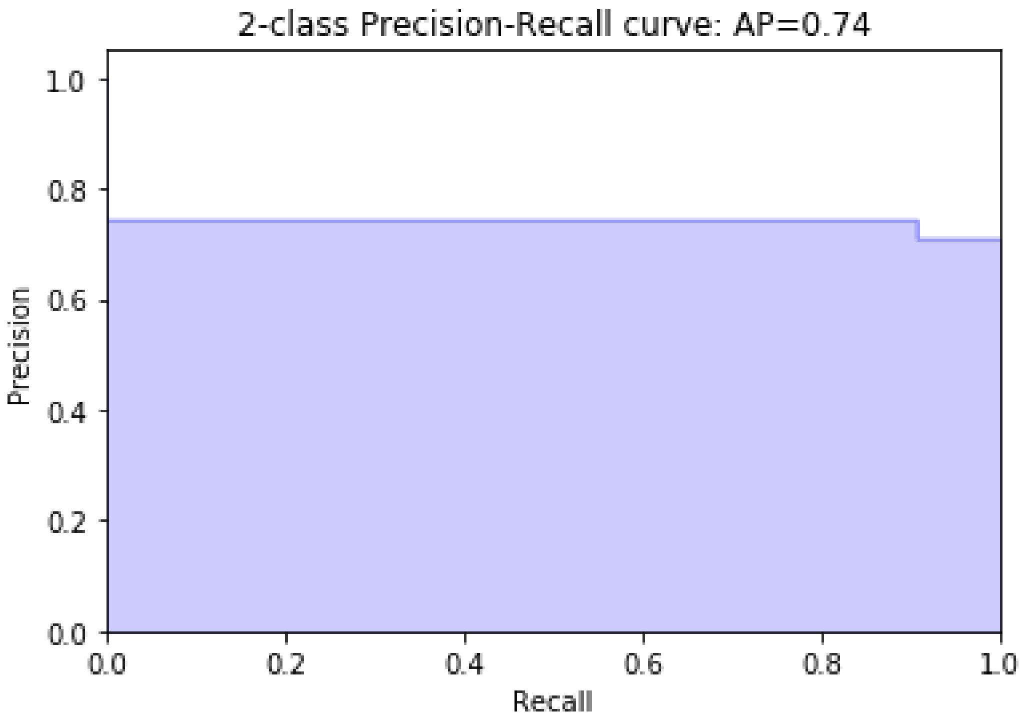
```
              precision    recall  f1-score   support

           0       0.17      0.43      0.24         7
           1       0.91      0.73      0.81        55

    accuracy                           0.69        62
   macro avg       0.54      0.58      0.52        62
weighted avg       0.83      0.69      0.74        62

0.30645161290322576
[[ 3  4]
 [15 40]]
```



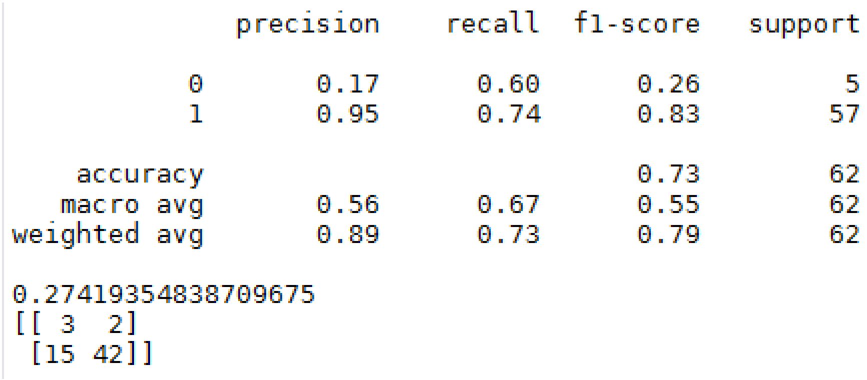
Naïve Bayes Classifier:



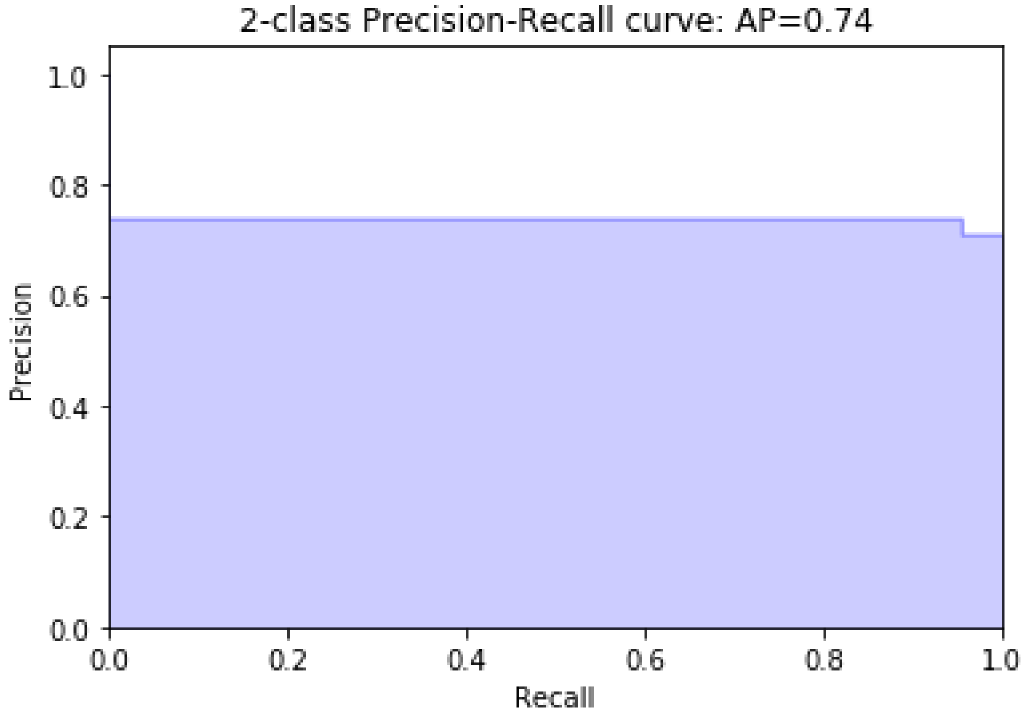
```
              precision    recall  f1-score   support

           0       0.22      0.50      0.31         8
           1       0.91      0.74      0.82        54

    accuracy                           0.71        62
   macro avg       0.57      0.62      0.56        62
weighted avg       0.82      0.71      0.75        62

0.29032258064516125
[[ 4  4]
 [14 40]]
```



AdaBoost Classifier:



```
              precision    recall  f1-score   support

           0       0.17      0.60      0.26         5
           1       0.95      0.74      0.83        57

    accuracy                           0.73        62
   macro avg       0.56      0.67      0.55        62
weighted avg       0.89      0.73      0.79        62

0.27419354838709675
[[ 3  2]
 [15 42]]
```



3. Comparisons and Conclusion

The Precision-Recall curves all show that the model performance has increased. Because we know that the higher the area under the curve, the better the model is at classifying samples correctly.

For comparing the F1 scores, we will consider the weighted average scores, since it gives us more accurate insights.

The scores are as followed:

- Random Forest: 0.81
- Bagging: 0.74
- AdaBoost: 0.79
- Naïve Bayes: 0.75

Out of all the models, Random Forest has the best score, then AdaBoost is the second best, then Naïve Bayes is the third best and lastly, we have Bagging.

Considering both accuracy and F1-scores, we can see that Cross validation combined with Ensemble techniques have increased the model performance and are better classifiers than the original Decision Tree classifier. The same can be said with Naïve Bayes, which proved to be a good classifier in case of an imbalanced dataset.

Overall, all 4 models are good for classifying the dataset, with Random Forest being the best of all.