Classification of Human Resources Data using AdaBoost Classifier in Python

Burak Koryan | burak@koryan.ca | http://koryan.ca | January 5 2019

Course resource : udemy.com/machinelearning

Objective:¶

The aim of this project is to investigate accuracy of AdaBoost classifier on a dataset when classifier parameters are changed and classifying chosen dataset with AdaBoost.

Introduction:

One of my favourite classifiers is AdaBoost and in order to investigate this classifier further,I chose to use it to classify an human resources dataset I found on kaggle.com [1]

The following sentences about the Adaboost algorithm are taken from Wikipedia.org: "AdaBoost is adaptive in the sense that subsequent weak learners are tweaked in favor of those instances misclassified by previous classifiers " [2].AdaBoost is sensitive to noisy data and outliers. In some problems it can be less susceptible to the overfitting problem than other learning algorithms[2].AdaBoost (with decision trees as the weak learners) is often referred to as the best out-of-the-box classifier[2]."

According to SciKit Learn,the Adaboost classifier is explained as follows: "An AdaBoost classifier is a meta-estimator that begins by fitting a classifier on the original dataset and then fits additional copies of the classifier on the same dataset but where the weights of incorrectly classified instances are adjusted such that subsequent classifiers focus more on difficult cases.[4]"

The dataset used in classification has multi-column data about employees of a company.The dataset has detailed information,such as pay rate,age of employees,state the employee is from.However,in order to make classification easier,only three parameters were selected from the dataset : age,pay rate,and sex of 310 employees in total.In line 3 of the source code below,what the dataset looks like can be seen.

In the next section below,the source code is shown and its output is printed step by step.Multiple plots of the classifier output can be seen when the number of trees changed from 1 to 50 step by step in order to examine the classifier output plots in detail.To support this,classifier accuracy and the confusion matrix of related plot is given as well.

Procedure:

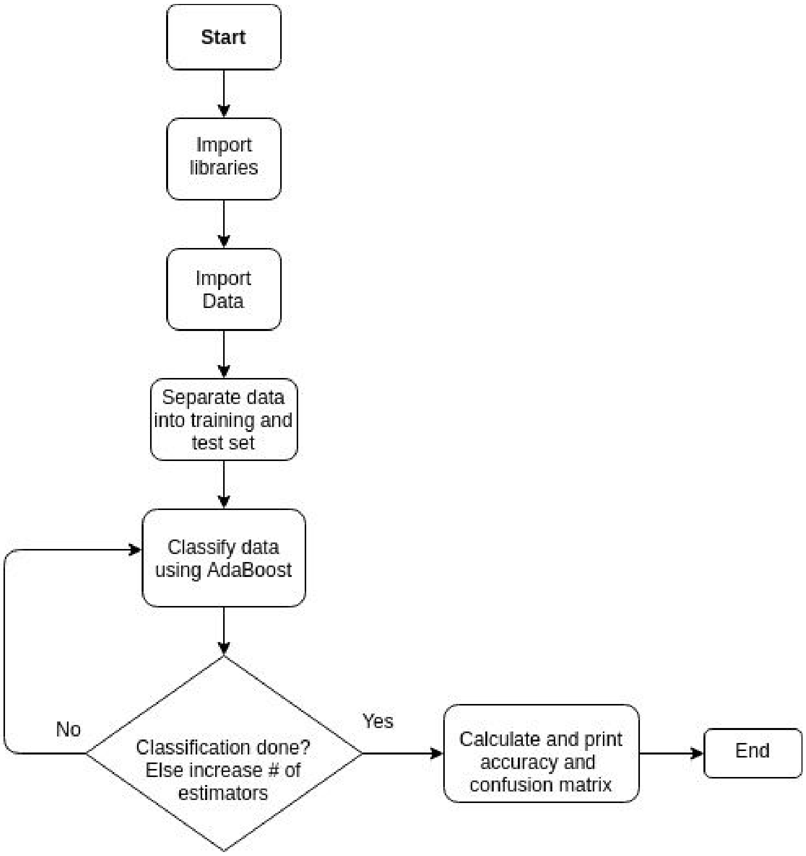
Start
Import libraries
Import Data
Separate data into training and test set
Classify data using AdaBoost
No
Classification done? Else increase # of estimators
Yes
Calculate and print accuracy and confusion matrix
End

Figure 1: Overall classifier algorithm

The procedure of the AdaBoost classification is as simple as shown in Figure 1.After the dataset was obtained from Kaggle.com,it was imported and selected columns in the dataset was separated as X and Y variables.In order to start classification,the training and test sets needed to be made.After the necessary test size has been selected,the test and training sets have been made.AdaBoost classifier parameters were chosen and set in the classifier function.Classification,as many times as needed,was done while changing the number of estimators in the classifier.As a result,the classifier accuracies and confusion matrices have been calculated and kept in a data array.

Results:

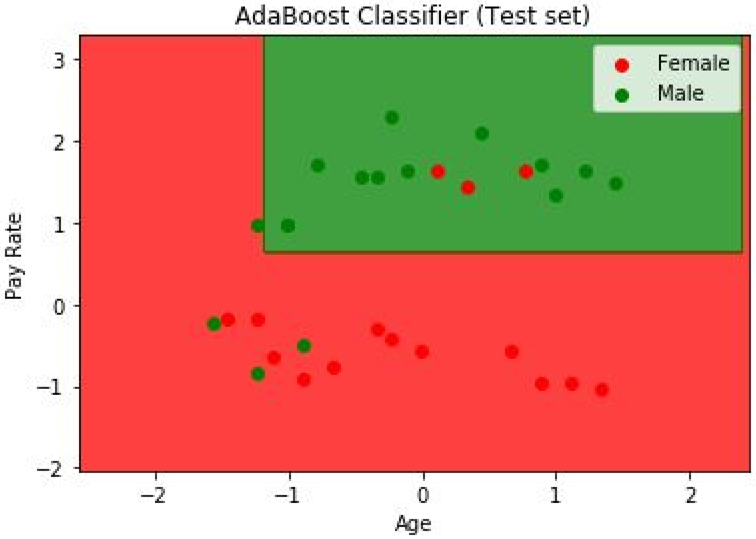

Figure 2 : Classifier output plot with maximum accuracy of 77% with number of estimators : 6 and test size: 10%

12	3
4	12

Table 1: Confusion matrix of AdaBoost when test size : 10% and accuracy : 77%

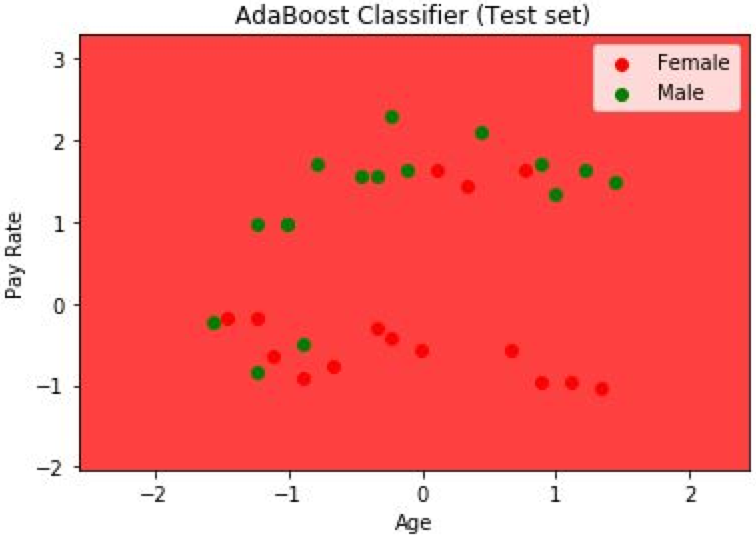

Figure 3 : Classifier Output plot with accuracy :48%,number of estimators = 1,and test size = 10%

15	0
16	0

Table 2:Confusion matrix of AdaBoost when test size : 10% ,accuracy : 48%,and number of estimators = 1

In order to see how the accuracy of AdaBoost changes with the given dataset,the test size and the number of estimators in the classifier have been changed.The test size was first set to be 10%,which is 31,of the total number of data elements,that is 310.As shown in Figure 2 and Table 1,the highest accuracy,77 %, from the classifier was obtained when the number of estimators is 6.In Table 2,the confusion matrix has been shown for these parameters,15 correct and 16 incorrect predictions have been made.Well,does this mean that when the training set is a lot more than the test set,the accuracy of the classifier is the highest? Definite answer should not be given without testing the classifier more.

The classifier output when the lowest accuracy was obtained can be seen in Figure 3.When the number of estimators was set to 1,the classifier accuracy was 48 %.From this,it cannot be certain that there is a direct relation between the number of estimators and the classifier accuracy without further investigation of classifier behavior for the dataset used.

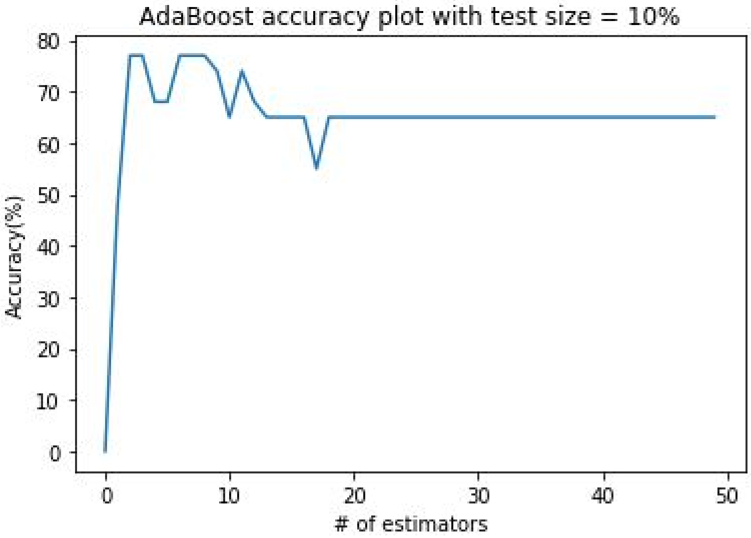

Figure 4: AdaBoost classifier plot with accuracy : 77% and test size : 10%

As shown in the figure above,the classifier accuracy is almost constant when the number of estimators increased by one from 20 to 50.The lowest accuracy was obtained when the number of estimators is 1,and the highest accuracy is obtained when the number of estimators is 6,8 and 9.

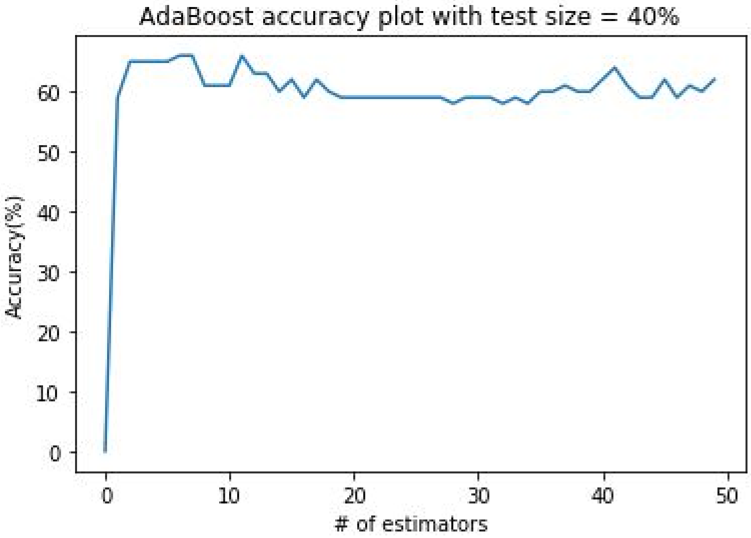

Figure 5: Adaboost classifier accuracy plot with test size 40%

When the test set size is changed to 40% and training set size to 60%,the overall accuracy of the classifier falls but still seem to be in the acceptable range,at least it is more than 50%.The maximum accuracy obtained from this change is about 66% when the number of estimators is 6 in the classifier.The overall accuracy fluctuates between 55% and 66% throughout the number of estimators change.In Figure 6,the classifier output can be seen.In table 3,the number of correct predictions have been shown.82 correct and 42 incorrect predictions have been made by AdaBoost.

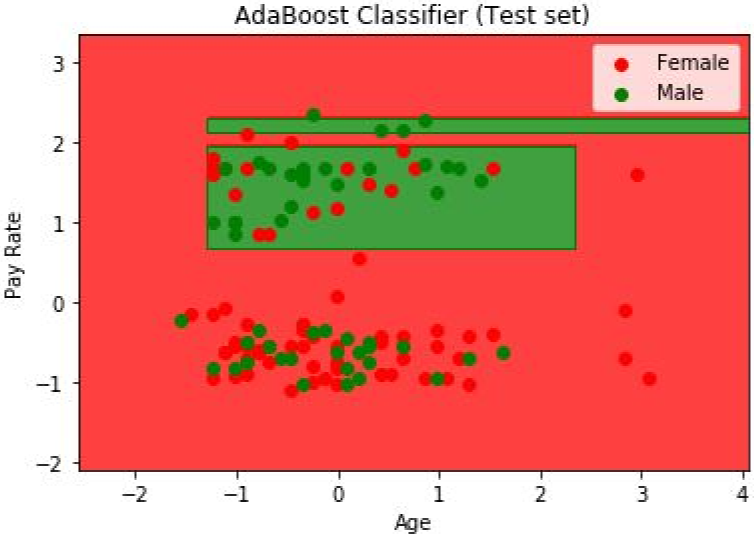

Figure 6 : Adaboost classifier output plot when test size:40% and accuracy : 66%

57	16
26	25

Table 3: Confusion matrix when test size : 40% and accuracy : 66%

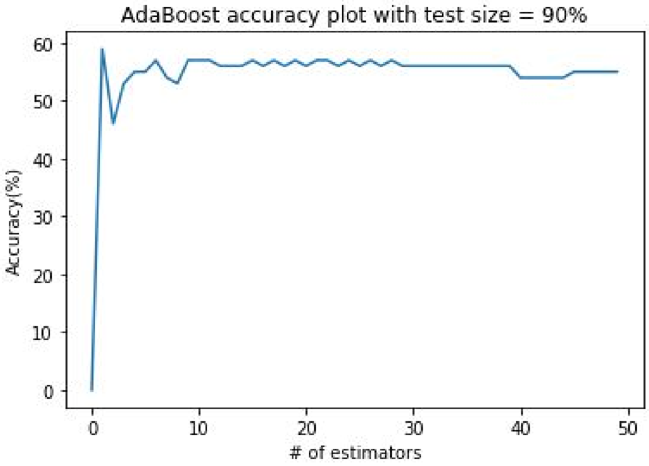

Figure 7: Adaboost classifier plot when test size:90%

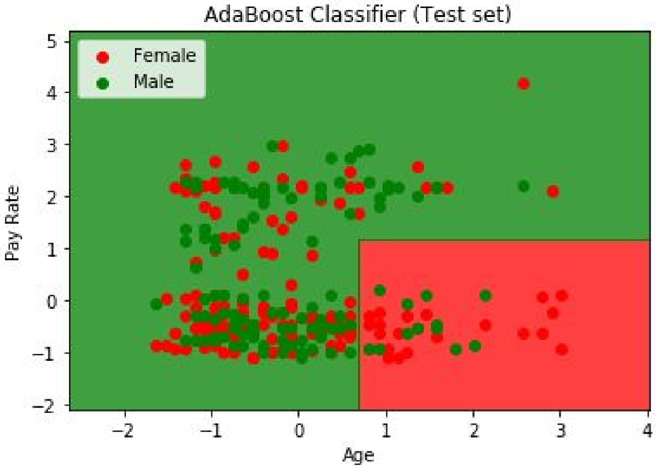

Figure 8: AdaBoost classifier output with test size : 90%,accuracy = 46% and number of estimators = 2

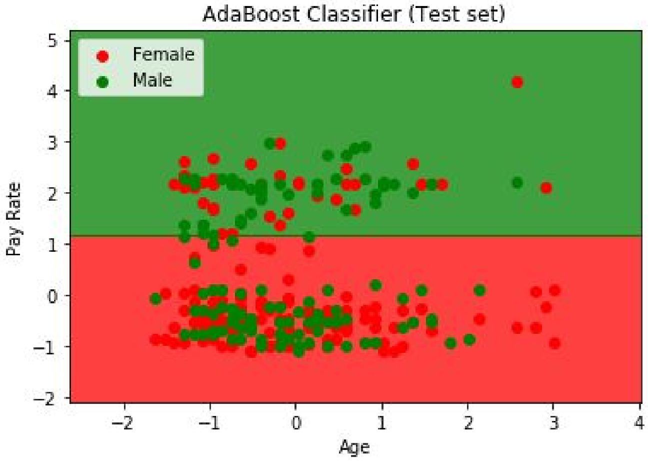

Figure 9 : AdaBoost classifier output with test size : 90%,accuracy = 59% and number of estimators = 1

23	138
12	106

Table 3: Confusion matrix of AdaBoost when test size : 90% accuracy : 46%,and number of estimators = 2

122	39
74	44

Table 4:Confusion matrix of AdaBoost when test size : 90%,accuracy : 59%,and number of estimators = 1

In Figure 7,AdaBoost accuracy plot is shown.The maximum accuracy obtained was 59%.As the number of estimators in the classifier increases,the classifier accuracy fluctuates between 50% and 60%.In Figures 8 and 9,the classifier outputs are shown in scatter plots.When the classifier accuracy is higher,the data points for either gender are classified in a wider range.In Table 3 and 4,confusion matrices are shown for the 46% and 59% accuracies.Out of 279 data points,129 are predicted correctly as shown in Table 3,which in other words translates into 46% accuracy.As shown in Table 4,out of 279 data points,166 data points are predicted correctly.

Conclusion:

Overall observation made in this project was that the accuracy rate of AdaBoost definitely relates to the test and training sizes of the classifier.As the test size gets smaller,the accuracy rate observed to be higher.The problem with that is though,in reality,it would be hard to have a 10% test-set all the time and assume that any classifier will give the highest accuracy rate.The accuracy of a classifier in classifying data also depends on the dataset used.Overall accuracy when the test set size is 40% or higher observed to be 46% or above which is acceptable for the sake of research,and compare/contrast of classifiers.Classifier parameters,such as number of estimators and classifier test/training set sizes definitely play a crucial role in classifier end-results.

References:

[1] **Dataset used from Kaggle.com :** https://www.kaggle.com/rhuebner/human-resources-data-set
[2] **AdaBoost on Wikipedia:** https://en.wikipedia.org/wiki/AdaBoost
[3] **AdaBoost classifier on SciKit:** https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.AdaBoostClassifier.html#sklearn.ensemble.AdaBoostClassifier