

Lab-08

Implement Boosting ensemble method on a given dataset.

→ considering AdaBoost Algorithm. For the following sample data, show the decision stump calculation steps for the attribute CGPA.

CGPA	Interaction	Practical Knowledge	Communication Skill	Job Profile
≥ 9	Yes	Good	Good	Yes
< 9	No	Good	Moderate	Yes
≥ 9	No	Average	Moderate	No
< 9	No	Average	Good	No
≥ 9	Yes	Good	Moderate	Yes
≥ 9	Yes	Good	Moderate	Yes

Sol

Step 1: Initialize weights

Initially, all 6 data instances have equal weights i.e. $1/6 = 0.167$ for each instance.

Step 2: Create decision stump for CGPA

The decision rule is:

- * If CGPA ≥ 9 : Predict "Yes" for Job profile
- * If CGPA < 9 : Predict "No" for Job profile

Step 3: Evaluate predictions - 10.4

1. $CAPA \geq 9$; Predicted: Yes; Actual: Yes \rightarrow Correct
2. $CAPA < 9$; Predicted: No; Actual: Yes \rightarrow Incorrect
3. $CAPA \geq 9$; Predicted: Yes; Actual: No \rightarrow Incorrect
4. $CAPA < 9$; Predicted: No; Actual: No \rightarrow Correct
5. $CAPA \geq 9$; Predicted: Yes; Actual: Yes \rightarrow Correct
6. $CAPA \geq 9$; Predicted: Yes; Actual: Yes \rightarrow Correct.

Step 4: Calculate weighted Error

$$\begin{aligned} \text{Weighted Error} &= \text{Misclassified instances} \\ &= 2^{\text{nd}} + 3^{\text{rd}} \text{ instance} \\ &= 0.167 + 0.167 = 0.333 \end{aligned}$$

Step 5: Calculate stump influence (α)

$$\alpha = \frac{1}{2} \ln \left(\frac{1 - \text{Error}}{\text{Error}} \right)$$

$$= 0.5 \ln \left(\frac{1 - 0.333}{0.333} \right)$$

$$\alpha = 0.347$$

→ write python code to implement the following
Consider "income.csv" dataset.

Build a adaboost classifier to classify income dataset

- i) Measure prediction score using n -estimators (10).
- ii) Now, fine tune your model by changing number of trees in your classifier and identify what best score can be get using how many trees.

Sol.

i) Accuracy with 10 estimators = 0.8277

ii) Considering number of trees in classifier.

Estimator : 1	Accuracy: 0.8034
5	: 0.8120
10	: 0.8202
20	: 0.8245
50	: 0.8300
75	: 0.8318
100	: 0.8321
150	: 0.8323
200	: 0.8326

Best performance : 0.8326 (83.26%) with 200 trees

SS
5/5