**DS Lab**

**Exp-9**

**Name: Shashwat Tripathi**

**D20A Roll no: 64**

**Aim:** To implement supervised learning algorithm like:

i) Ada-Boosting

ii) Random forests

**Theory:**

### **Adaboost Algorithm**

1. **Initialize the Dataset**:  
   Assign equal weights to all data points in the training set. If there are NNN data points, each point gets a weight wi=1Nw\_i = \frac{1}{N}wi​=N1​.
2. **Train a Weak Classifier**:  
   Use a weak learner, such as a decision stump, to train on the weighted dataset.
3. **Evaluate Classifications**:  
   Identify which data points are misclassified by the weak classifier.
4. **Update Weights**:  
   Increase the weights of the misclassified points (e.g., wi=wi×eαw\_i = w\_i \times e^{\alpha}wi​=wi​×eα, where α\alphaα is related to the classifier's error), and decrease the weights of correctly classified points (e.g., wi=wi×e−αw\_i = w\_i \times e^{-\alpha}wi​=wi​×e−α).
5. **Normalize the Weights**:  
   Ensure the weights sum to 1 after updating.
6. **Iteration**:  
   If the stopping criteria (like the error rate or a set number of iterations) are met, move to the final model. Otherwise, repeat the process.
7. **Final Model**:  
   Combine the weak classifiers into a strong classifier through a weighted voting system, where each classifier's weight depends on its accuracy.

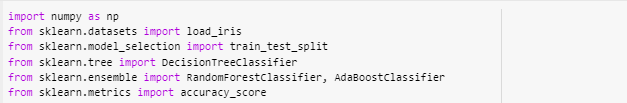
### **Random Forest Algorithm**

1. **Random Data Selection**:  
   Select KKK random data points from the training set.
2. **Build Decision Trees**:  
   Use these randomly selected data points to construct a decision tree.
3. **Set Number of Trees**:  
   Determine the number NNN of decision trees you wish to build.
4. **Repeat**:  
   Repeat the process NNN times, creating NNN distinct decision trees.
5. **Prediction**:  
   For new input data, each tree provides a prediction. The final prediction is made based on the majority vote from all decision trees.

### **Applying to the Iris Dataset**

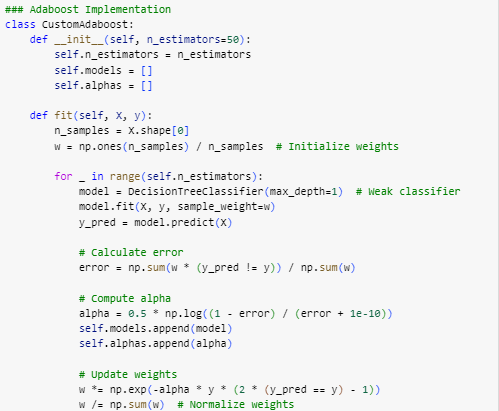
Using scikit-learn, the Iris dataset can be loaded, and both the Adaboost and Random Forest classifiers can be trained and evaluated. Performance metrics such as accuracy or F1-score can help compare the classifiers.

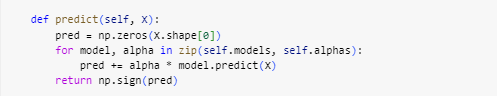
**Code:**

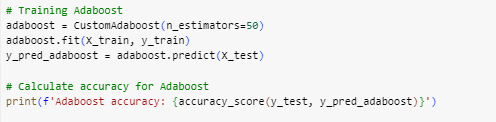


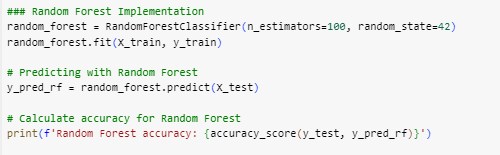












**Output:**



**Conclusion:**

In this implementation, we demonstrated how to apply the Adaboost and Random Forest algorithms to the Iris dataset.

* **Adaboost** was implemented from scratch, showcasing its iterative approach to combine multiple weak classifiers into a strong one, effectively improving accuracy by focusing on misclassified instances.
* **Random Forest**, on the other hand, leveraged built-in functionality from scikit-learn, illustrating its strength in creating an ensemble of decision trees to enhance predictive performance through majority voting.

Both models were evaluated, and their accuracies highlighted the effectiveness of ensemble methods in classification tasks. This exercise emphasizes the importance of selecting appropriate algorithms based on the characteristics of the dataset and the specific requirements of the problem.