Week-9

Write a program to implement AdaBoost algorithm to classify theiris data set. Print both correct and wrong predictions.

**Algorithm:**

It works in the following steps:

1. Initially, Adaboost selects a training subset randomly.

2. It iteratively trains the AdaBoost machine learning model by selecting the training set based on the accurate prediction of the last training.

3. It assigns the higher weight to wrong classified observations so that in the next iteration these observations will get the high probability for classification.

4. Also, It assigns the weight to the trained classifier in each iteration according to the accuracy of the classifier. The more accurate classifier will get high weight.

5. This process iterate until the complete training data fits without any error or until reached to the specified maximum number of estimators.

6. To classify, we perform a "vote" across all of the learning algorithms we have built.

Program:

Importing Necessary Libraries:

import pandas as pd

import numpy as np

import sklearn

from sklearn import datasets

Loading the Dataset:

iris = datasets.load\_iris()

X = iris.data

Y = iris.target

Visualizing the Dataset:

iris.target\_names

iris.feature\_names

iris.data[0:5]

Splitting the Dataset:

from sklearn.model\_selection import train\_test\_split

X\_train, X\_test, Y\_train, Y\_test = train\_test\_split(X, Y, test\_size=0.3)

Building the AdaBoost Model:

from sklearn.ensemble import AdaBoostClassifier

adb = AdaBoostClassifier()

model = adb.fit(X\_train, Y\_train)

y\_pred = model.predict(X\_test)

y\_pred

Evaluating the Model

from sklearn.metrics import confusion\_matrix, classification\_report, accuracy\_score

print(confusion\_matrix(Y\_test, y\_pred))

print(classification\_report(Y\_test, y\_pred))

print(accuracy\_score(Y\_test, y\_pred))

Parameters Tuning: The most important parameters of AdaBoost are: ..

->base\_estimator ->

n\_estimators ->

-> learning\_rate

"Using SVM as a base estimator"""

from sklearn.svm import SVC

from sklearn.ensemble import AdaBoostClassifier

svc = SVC(probability = True, kernel='linear')

adb = AdaBoostClassifier(n\_estimators=100,base\_estimator = svc, learning\_rate= 0.01)

Training and Predicting the AdaBoost Classifier

model = adb.fit(X\_train, Y\_train)

y\_pred = model.predict(X\_test)

y\_pred

Evaluating the Model:

print(confusion\_matrix(Y\_test, y\_pred))

print(classification\_report(Y\_test, y\_pred))

print(accuracy\_score(Y\_test, y\_pred))

Output:

[[16 0 0]

[ 0 15 1]

[ 0 0 13]]

precision recall f1-score support

0 1.00 1.00 1.00 16

1 1.00 0.94 0.97 16

2 0.93 1.00 0.96 13

accuracy 0.98 45

macro avg 0.98 0.98 0.98 45

weighted avg 0.98 0.98 0.98 45

0.9777777777777777

**Discussion Topic**

**1. How AdaBoost work?**

It works on the principle of learners growing sequentially. Except for the first, each subsequent learner is grown from previously grown learners. In simple words, weak learners are converted into strong ones. The AdaBoost algorithm works on the same principle as boosting with a slight difference.

**2. What are advantages of AdaBoost?**

Adaboost is less prone to overfitting as the input parameters are not jointly optimized. The accuracy of weak classifiers can be improved by using Adaboost. Nowadays, Adaboost is being used to classify text and images rather than binary classification problems.

**3. What are the disadvantages of the AdaBoost algorithm?**

AdaBoost uses a progressively learning boosting technique. Hence high-quality data is needed in examples of AdaBoost vs Random Forest. It is also very sensitive to outliers and noise in data requiring the elimination of these factors before using the data.