**PROGRAMMING ASSIGNMENT 4**

**ENSEMBLE METHODS**

By

Thota, Sai Abhishek – SXT210056

Gauri, Rahul – RXG200002

Part A – We have executed the above bagging function to get confusion matrices of maximum depth d = 3,5 and bag size (k = 10,20).

The screenshots of the matrices are below

|  |  |
| --- | --- |
| Max Depth: 3 Bag Size: 10 | Table  Description automatically generated |
| Max Depth: 3 Bag Size: 20 | Table  Description automatically generated |
| Max Depth: 5 Bag Size: 10 |  |
| Max Depth: 5 Bag Size: 20 |  |

Part 2 - For Boosting - We have executed the above bagging function to get confusion matrices of maximum depth d = 1,2 and bag size (k = 20,40).

|  |  |
| --- | --- |
| Max Depth: 1 Bag Size: 20 |  |
| Max Depth: 1 Bag Size: 40 |  |
| Max Depth: 2  Bag Size: 20 |  |
| Max Depth: 2  Bag Size: 40 |  |

Part C – scikit implementation

We have used scikit library to implement bagging and adaboost implementation.

For bagging using Scikit implementation

|  |  |
| --- | --- |
| Max Depth: 3  Bag Size: 10 |  |
| Max Depth: 5  Bag Size: 10 |  |
| Max Depth: 3  Bag Size: 20 |  |
| Max Depth: 5  Bag Size: 20 |  |
|  |  |

|  |  |
| --- | --- |
|  |  |

Adaboost implementation

|  |  |
| --- | --- |
| Max Depth: 1  Bag Size: 20 |  |
| Max Depth: 2  Bag Size: 20 |  |
| Max Depth: 1  Bag Size: 40 |  |
| Max Depth: 2  Bag Size: 40 |  |

We can observe that scikit implementation is better because it can categorize positives and negatives with less error. It has higher TPR and TNR.