**the overall accuracy on the test dataset for each dataset are as follows.**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Led | Nursery | Balance | Synthetic |
| Decision Tree | 0.852 | 0.972 | 0.613 | 0.474 |
| Random Forest | 0.859 | 0.973 | 0.667 | 0.648 |

**Classification method:**

I use the Decision Tree with gini Index. For each value in each attribute, I split the current dataset and calculate the gini Index to find a best split. If all samples for a given node belongs to the same class, or there are no remaining attributes for split, or there are no samples left, then the tree can stop splitting. To classify a leaf which samples belong to more than one class, the majority voting is employed for prediction. For each testing sample, the tree will give a predicted value.

In Random Forest, I choose to use several Decision Tree to form a forest. I train each tree model using a subset of training set and a subset of attribute set. Models learn independently. and then use the vote majority to be the predicted value.

**All model evaluation measures:**

Decision Tree:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Led | | Nursery | | Balance | | Synthetic | |
|  | train | test | train | test | train | test | train | test |
| accuracy | 0.854 | 0.852 | 0.997 | 0.972 | 0.847 | 0.613 | 1.000 | 0.474 |
| F1 score per class | 0.767 | 0.765 | 0.995 | 0.959 | 0.448 | 0.0 | 1.000 | 0.472 |
| 0.894 | 0.891 | 0.977 | 0.697 | 0.877 | 0.744 | 1.000 | 0.429 |
|  |  | 0.997 | 0.982 | 0.869 | 0.644 | 1.000 | 0.495 |
|  |  | 1.000 | 1.000 |  |  | 1.000 | 0.495 |
|  |  | 0.666 | 0.000 |  |  |  |  |

Random Forest:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Led | | Nursery | | Balance | | Synthetic | |
|  | train | test | train | test | train | test | train | test |
| accuracy | 0.859 | 0.859 | 0.998 | 0.973 | 0.872 | 0.667 | 1.000 | 0.648 |
| F1 score per class | 0.766 | 0.774 | 0.997 | 0.962 | 0.341 | 0.000 | 1.000 | 0.633 |
| 0.899 | 0.860 | 0.978 | 0.719 | 0.896 | 0.728 | 1.000 | 0.607 |
|  |  | 0.997 | 0.983 | 0.906 | 0.714 | 1.000 | 0.676 |
|  |  | 1.000 | 1.000 |  |  | 1.000 | 0.670 |
|  |  | 0.667 | 0.000 |  |  |  |  |

**Parameters:**

The parameter I choose for Decision Tree:

No parameters.

The parameter I choose for Random Forest:

For the dataset ‘synthetic.social’, I choose to generate 20 trees, each tree will have a randomly sampled input which size is 90% of the original input. And each tree is randomly chosen 90 attributes for constructing. For other three datasets ‘led’, ‘nursery’, ‘balance.scale’, I choose to generate 10 trees, and each tree will have a randomly sampled input which size is 90% of the original input.

Reason: I think for larger dataset like ‘synthetic.social’, tree number should be more than the other three small dataset. And in order to save time, I choose to use 10 trees for three small datasets and 20 trees for synthetic dataset.

**Conclusion:**

The ensemble methods do improve the performance of the basic method I choose. Because ensemble methods add randomness, which can avoid the risk of overfitting compared with a single classification model. And with majority voting, multiple models together can to some extent rectify some prediction faults if one or some models have wrong prediction.