Tree Classifiers

Tree Classifiers on **CNF Boolean Data**:

Best Parameters for the 15 Datasets trained on Train data and tuning on Validation data and the Accuracy and F1-Score on the Test data.

1. sklearn.tree.DecisionTreeClassifier

Index	c	d	criterion	max_features	splitter	Accuracy	F1-Score
0	300	100	gini	None	best	0.628141	0.647619
1	300	1000	entropy	None	best	0.671336	0.674591
2	300	5000	entropy	None	random	0.760976	0.759024
3	500	100	entropy	None	random	0.708543	0.697917
4	500	1000	entropy	None	random	0.683842	0.686819
5	500	5000	entropy	None	random	0.778078	0.77997
6	1000	100	gini	None	random	0.678392	0.703704
7	1000	1000	entropy	None	random	0.792896	0.805451
8	1000	5000	entropy	None	random	0.842084	0.845816
9	1500	100	gini	sqrt	random	0.753769	0.758621
10	1500	1000	entropy	None	random	0.912956	0.914956
11	1500	5000	entropy	None	random	0.954895	0.955377
12	1800	100	gini	None	best	0.944724	0.944724
13	1800	1000	entropy	None	best	0.972986	0.973346
14	1800	5000	entropy	None	random	0.983598	0.983717

2. sklearn.ensemble.BaggingClassifier with “DecisionTreeClassifier" as the base estimator.

Index	c	d	base_estimator	bootstrap	n_estimators	Accuracy	F1-Score
0	300	100	DecisionTreeClassifier()	TRUE	15	0.713568	0.724638
1	300	1000	DecisionTreeClassifier()	TRUE	25	0.826413	0.829651
2	300	5000	DecisionTreeClassifier()	TRUE	25	0.89669	0.90228
3	500	100	DecisionTreeClassifier()	TRUE	25	0.829146	0.83
4	500	1000	DecisionTreeClassifier()	TRUE	25	0.865933	0.866267
5	500	5000	DecisionTreeClassifier()	TRUE	25	0.914891	0.915851
6	1000	100	DecisionTreeClassifier()	TRUE	20	0.874372	0.870466
7	1000	1000	DecisionTreeClassifier()	TRUE	25	0.928464	0.929452
8	1000	5000	DecisionTreeClassifier()	TRUE	25	0.955896	0.956185
9	1500	100	DecisionTreeClassifier()	TRUE	20	0.979899	0.979798
10	1500	1000	DecisionTreeClassifier()	TRUE	25	0.977989	0.977956
11	1500	5000	DecisionTreeClassifier()	TRUE	25	0.988599	0.988593
12	1800	100	DecisionTreeClassifier()	TRUE	20	0.979899	0.979592
13	1800	1000	DecisionTreeClassifier()	TRUE	25	0.993997	0.993988
14	1800	5000	DecisionTreeClassifier()	TRUE	25	0.9972	0.997199

3. sklearn.ensemble.RandomForestClassifier.

Index	c	d	criterion	max_features	n_estimators	Accuracy	F1-Score
0	300	100	gini	sqrt	150	0.824121	0.814815
1	300	1000	gini	None	150	0.897449	0.899559
2	300	5000	gini	None	150	0.922692	0.927069
3	500	100	gini	sqrt	150	0.879397	0.881188
4	500	1000	entropy	sqrt	150	0.944472	0.944694
5	500	5000	gini	sqrt	150	0.954195	0.954382
6	1000	100	gini	sqrt	100	0.974874	0.974874
7	1000	1000	entropy	log2	150	0.993997	0.993994
8	1000	5000	gini	log2	150	0.9972	0.997201
9	1500	100	gini	sqrt	50	1	1
10	1500	1000	gini	sqrt	50	0.9995	0.999499
11	1500	5000	entropy	log2	150	1	1
12	1800	100	gini	sqrt	50	1	1
13	1800	1000	gini	sqrt	50	1	1
14	1800	5000	gini	sqrt	150	1	1

4. sklearn.ensemble.GradientBoostingClassifier

Index	c	d	learning_rate	max_features	n_estimators	Accuracy	F1-Score
0	300	100	0.1	sqrt	150	0.859296	0.858586
1	300	1000	0.1	log2	150	0.883442	0.883674
2	300	5000	1	sqrt	150	0.920792	0.921366
3	500	100	0.01	log2	150	0.904523	0.902564
4	500	1000	0.1	log2	150	0.947974	0.948617
5	500	5000	1	sqrt	150	0.963396	0.963553
6	1000	100	1	log2	150	0.98995	0.989899
7	1000	1000	1	sqrt	150	0.991996	0.991984
8	1000	5000	1	sqrt	150	0.9973	0.997302
9	1500	100	1	sqrt	50	1	1
10	1500	1000	1	sqrt	100	1	1
11	1500	5000	1	sqrt	150	1	1
12	1800	100	1	sqrt	50	1	1
13	1800	1000	1	sqrt	50	1	1
14	1800	5000	1	sqrt	50	1	1

5. Comparing the Four Tree Classifiers with the best tuned parameters.

Tabulated keeping the no. of clauses “c” as constant and varying no. of examples “d”

			Accuracy				F1-Score			
Index	**c**	**d**	**DTC**	**BC(DTC)**	**RFC**	**GBC**	**DTC**	**BC(DTC)**	**RFC**	**GBC**
0	**300**	**100**	0.6281	0.7136	0.8241	**0.8593**	0.6476	0.7246	0.8148	**0.8586**
1	**300**	**1000**	0.6713	0.8264	**0.8974**	0.8834	0.6746	0.8297	**0.8996**	0.8837
2	**300**	**5000**	0.7610	0.8967	**0.9227**	0.9208	0.7590	0.9023	**0.9271**	0.9214
3	**500**	**100**	0.7085	0.8291	0.8794	**0.9045**	0.6979	0.8300	0.8812	**0.9026**
4	**500**	**1000**	0.6838	0.8659	0.9445	**0.9480**	0.6868	0.8663	0.9447	**0.9486**
5	**500**	**5000**	0.7781	0.9149	0.9542	**0.9634**	0.7800	0.9159	0.9544	**0.9636**
6	**1000**	**100**	0.6784	0.8744	0.9749	**0.9900**	0.7037	0.8705	0.9749	**0.9899**
7	**1000**	**1000**	0.7929	0.9285	**0.9940**	0.9920	0.8055	0.9295	**0.9940**	0.9920
8	**1000**	**5000**	0.8421	0.9559	0.9972	**0.9973**	0.8458	0.9562	0.9972	**0.9973**
9	**1500**	**100**	0.7538	0.9799	**1.0000**	**1.0000**	0.7586	0.9798	**1.0000**	**1.0000**
10	**1500**	**1000**	0.9130	0.9780	0.9995	1.0000	0.9150	0.9780	0.9995	**1.0000**
11	**1500**	**5000**	0.9549	0.9886	**1.0000**	**1.0000**	0.9554	0.9886	**1.0000**	**1.0000**
12	**1800**	**100**	0.9447	0.9799	**1.0000**	**1.0000**	0.9447	0.9796	**1.0000**	**1.0000**
13	**1800**	**1000**	0.9730	0.9940	**1.0000**	**1.0000**	0.9733	0.9940	**1.0000**	**1.0000**
14	**1800**	**5000**	0.9836	0.9972	**1.0000**	**1.0000**	0.9837	0.9972	**1.0000**	**1.0000**

The GradientBoostingClassifier yields the best overall generalization accuracy/F1 score. It uses an ensemble model in a forward step wise manner where in each stage the error that occurred in the previous stage is compensated i.e., introduces a weak learner to compensate the shortcomings of existing weak learners. However, sometimes the RandomForestClassifier performs little better as it uses bagging to build independent decision trees and combine them in parallel. In case of **Binary classification** which is a special case in GBC where only a single regression tree is induced and performs better for this kind of CNF data which generally have less noise.

As we increase the amount of training data, each of the four classifiers yielded better accuracies and F1-Scores. As per the above table. But, for c=500 & d=1000 there is a decrease in accuracy and F1-Score for DTC. And, for c=1500 & d = 1000 there is a slight decrease in the Accuracy and F1-Score for BC(DTC) & RFC.

Tabulated keeping the no. of examples “d” as constant and varying no. of clauses “c”

			Accuracy				**F1-Score**			
Index	**c**	**d**	**DTC**	**BC(DTC)**	**RFC**	**GBC**	**DTC**	**BC(DTC)**	**RFC**	**GBC**
0	**300**	**100**	0.6281	0.7136	0.8241	**0.8593**	0.6476	0.7246	0.8148	**0.8586**
3	**500**	**100**	0.7085	0.8291	0.8794	**0.9045**	0.6979	0.8300	0.8812	**0.9026**
6	**1000**	**100**	0.6784	0.8744	0.9749	**0.9900**	0.7037	0.8705	0.9749	**0.9899**
9	**1500**	**100**	0.7538	0.9799	**1.0000**	**1.0000**	0.7586	0.9798	**1.0000**	**1.0000**
12	**1800**	**100**	0.9447	0.9799	**1.0000**	**1.0000**	0.9447	0.9796	**1.0000**	**1.0000**
1	**300**	**1000**	0.6713	0.8264	**0.8974**	0.8834	0.6746	0.8297	**0.8996**	0.8837
4	**500**	**1000**	0.6838	0.8659	0.9445	**0.9480**	0.6868	0.8663	0.9447	**0.9486**
7	**1000**	**1000**	0.7929	0.9285	**0.9940**	0.9920	0.8055	0.9295	**0.9940**	0.9920
10	**1500**	**1000**	0.9130	0.9780	0.9995	1.0000	0.9150	0.9780	0.9995	**1.0000**
13	**1800**	**1000**	0.9730	0.9940	**1.0000**	**1.0000**	0.9733	0.9940	**1.0000**	**1.0000**
2	**300**	**5000**	0.7610	0.8967	**0.9227**	0.9208	0.7590	0.9023	**0.9271**	0.9214
5	**500**	**5000**	0.7781	0.9149	0.9542	**0.9634**	0.7800	0.9159	0.9544	**0.9636**
8	**1000**	**5000**	0.8421	0.9559	0.9972	**0.9973**	0.8458	0.9562	0.9972	**0.9973**
11	**1500**	**5000**	0.9549	0.9886	**1.0000**	**1.0000**	0.9554	0.9886	**1.0000**	**1.0000**
14	**1800**	**5000**	0.9836	0.9972	**1.0000**	**1.0000**	0.9837	0.9972	**1.0000**	**1.0000**

As per the above table, as we increase the number of clauses, each of the four classifiers yielded better accuracies and F1-Scores. But, for c=1000 & d=100 there is a decrease in accuracy for DTC.

6. Tree Classifiers on **MNIST Data**:

1. sklearn.tree.DecisionTreeClassifier

Index	criterion	splitter	max_features	accuracy	run_time
0	gini	best	sqrt	0.8356	0.73
1	gini	best	log2	0.7964	0.344
2	gini	best	None	0.8751	17.74
3	gini	random	sqrt	0.8353	0.464
4	gini	random	log2	0.7899	0.273
5	gini	random	None	0.8763	7.357
6	entropy	best	sqrt	0.8507	1.182
7	entropy	best	log2	0.8086	0.541
8	**entropy**	**best**	**None**	**0.8856**	**23.262**
9	entropy	random	sqrt	0.822	0.461
10	entropy	random	log2	0.7758	0.282
11	entropy	random	None	0.8855	5.518

2. sklearn.ensemble.BaggingClassifier with “DecisionTreeClassifier" as the base estimator.

Index	base_estimator	n_estimators	bootstrap	accuracy	run_time
0	DecisionTreeClassifier()	5	TRUE	0.9187	34.274
1	DecisionTreeClassifier()	5	FALSE	0.8856	24.966
2	DecisionTreeClassifier()	10	TRUE	0.9391	48.252
3	DecisionTreeClassifier()	10	FALSE	0.8898	24.422
4	DecisionTreeClassifier()	15	TRUE	0.9517	61.017
5	DecisionTreeClassifier()	15	FALSE	0.8893	24.427
6	**DecisionTreeClassifier()**	**20**	**TRUE**	**0.9531**	**77.671**
7	DecisionTreeClassifier()	20	FALSE	0.8918	24.978
8	DecisionTreeClassifier()	25	TRUE	0.9517	95.559
9	DecisionTreeClassifier()	25	FALSE	0.8898	26.791

3. sklearn.ensemble.RandomForestClassifier.

Index	n_estimators	criterion	max_features	accuracy	run_time
0	50	gini	sqrt	0.9658	9.485
1	50	gini	log2	0.9642	11.907
2	50	gini	None	0.9518	30.716
3	50	entropy	sqrt	0.9673	3.601
4	50	entropy	log2	0.964	2.152
5	50	entropy	None	0.9599	44.248
6	100	gini	sqrt	0.9698	17.438
7	100	gini	log2	0.9666	21.558
8	100	gini	None	0.9573	55.305
9	100	entropy	sqrt	0.9694	4.756
10	100	entropy	log2	0.967	2.839
11	100	entropy	None	0.9613	72.877
12	150	gini	sqrt	0.9691	23.955
13	150	gini	log2	0.967	33.409
14	150	gini	None	0.9573	80.326
15	**150**	**entropy**	**sqrt**	**0.9703**	**6.117**
16	150	entropy	log2	0.968	3.668
17	150	entropy	None	0.9614	99.803

4. sklearn.ensemble.GradientBoostingClassifier

Index	learning_rate	n_estimators	max_features	accuracy	run_time
0	1	50	sqrt	0.9131	86.02
1	1	50	log2	0.9215	43.923
2	1	100	sqrt	0.8159	104.495
3	1	100	log2	0.8674	54.706
4	1	150	sqrt	0.2851	155.712
5	1	150	log2	0.6652	81.585
6	0.1	50	sqrt	0.9167	53.866
7	0.1	50	log2	0.9037	28.482
8	0.1	100	sqrt	0.94	106.765
9	0.1	100	log2	0.9291	56.578
10	**0.1**	**150**	**sqrt**	**0.9505**	**160.899**
11	0.1	150	log2	0.9399	84.336
12	0.01	50	sqrt	0.8529	54.012
13	0.01	50	log2	0.8352	28.448
14	0.01	100	sqrt	0.8724	108.013
15	0.01	100	log2	0.8642	56.464
16	0.01	150	sqrt	0.8829	162.013
17	0.01	150	log2	0.876	85.004

Tabulated by sorting the accuracies for comparison of Four Tree Classifiers.

	Sorted By Accuracies			
	DTC	**BC(DTC)**	**RFC**	**GBC**
	0.7758	0.8856	0.9518	0.2851
	0.7899	0.8893	0.9573	0.6652
	0.7964	0.8898	0.9573	0.8159
	0.8086	0.8898	0.9599	0.8352
	0.822	0.8918	0.9613	0.8529
	0.8353	0.9187	0.9614	0.8642
	0.8356	0.9391	0.964	0.8674
	0.8507	0.9517	0.9642	0.8724
	0.8751	0.9517	0.9658	0.876
	0.8763	0.9531	0.9666	0.8829
	0.8855		0.967	0.9037
	0.8856		0.967	0.9131
			0.9673	0.9167
			0.968	0.9215
			0.9691	0.9291
			0.9694	0.9399
			0.9698	0.94
			0.9703	0.9505
Average of Accuracy	0.8364	0.91606	**0.964305556**	0.846205556
Variance of Accuracy	0.001503845	0.000895758	**2.65429E-05**	0.023889661

Among the four tree classifiers the Random Forest Classifier yields the best generalization accuracy for the MNIST dataset. As the MNIST dataset is a multi-class object detection, which generally have a lot of noise and is not a good choice for Gradient Boosting Classifier as it is more prone to overfitting and it is evident from the above table where the accuracy ranges from 28.51% to 95.05% whereas for RFC the accuracy ranges from 95.18% to 97.03%