A Comparative Study of Decision Tree and Random Forest using MAGIC Telescope Dataset.

INM431: Machine Learning
Nazia Ferdause Sodial

1. Description and motivation:

The focus of this study is to classify the images generated by MAGIC telescope using Random Forest and Decision Tree and detect which model out performs the other. The inputs of this dataset were collected from Monte Carlo data which were generated, approximately triggered and pre-processed for an imaging Major Atmospheric Gamma Imaging Cherenkov (MAGIC) telescope. The data belongs to two classes of images. They are either formed due to incident gamma rays or caused by hadronic showers [1].

2. Initial Analysis of Dataset:

- The dataset is MagicTelescope from OpenML whose source is UCI Machine Learning Repository.
- The dataset has 19020 rows and 12 columns.
- There are 10 predictors and 1 target which has two classes of images, one class is labelled as 'g' which implies the image originating due to gamma rays and the other class is 'h' which implies the image caused by hadronic showers.
- There are no missing values in the dataset.
- Figure 1 depicts that class 'g' has 12332 instances and class 'h' 6688 instances.
- The table 2 provides all the basic statistics of the data.
- The outliers were visualized using boxplot but they were not removed considering that the tree models are robust towards outliers.
- The skewness of the predictors can be observed in the distribution table 3 where the class 'g' is assigned to 1 and 'h' to 0.
- At this stage of initial analysis the data was not normalized, as it was crucial to experiment the importance of normalized data in creating tree models.
- Pair plot and correlation matrix were created to visualize the correlation of the features amongst each other.
- From the correlation heat map it is clearly visible that the dataset has high multicollinearity. There are highly correlated features like fConc and fConc1 having correlation as high as 0.98. However, none of the features were dropped as it requires a deep understanding of how images are created by Cherenkov Telescope and how significant is the contribution of these features.
- Further, the important predictors were selected to create and validate the tree models.

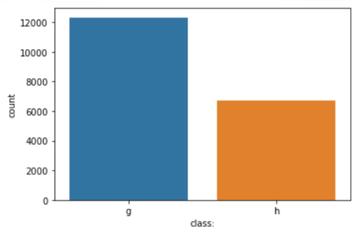

Figure 1. Counterplot of the classes

Attributes	Mean	Std	25%	50%	75%	Min	Max
fLength	53.250154	42.364855	24.336000	37.147700	70.122175	4.283500	334.177000
fWidth	22.180966	18.346056	11.863800	17.139900	24.739475	0.000000	256.382000
fSize	2.825017	0.472599	2.477100	2.739600	3.101600	1.941300	5.323300
fConc	0.380327	0.182813	0.235800	0.354150	0.503700	0.013100	0.893000
fConc1	0.214657	0.110511	0.128475	0.196500	0.285225	0.000300	0.675200
fAsym	-4.331745	59.206062	-20.586550	4.013050	24.063700	-457.916100	575.240700
fM3Long	10.545545	51.000118	-12.842775	15.314100	35.837800	-331.780000	238.321000
fM3Trans	0.249726	20.827439	-10.849375	0.666200	10.946425	-205.894700	179.851000
fAlpha	27.645707	26.103621	5.547925	17.679500	45.883550	0.000000	90.000000
fDist	193.818026	74.731787	142.492250	191.851450	240.563825	1.282600	495.561000

Figure 2. Basic statistics of the data

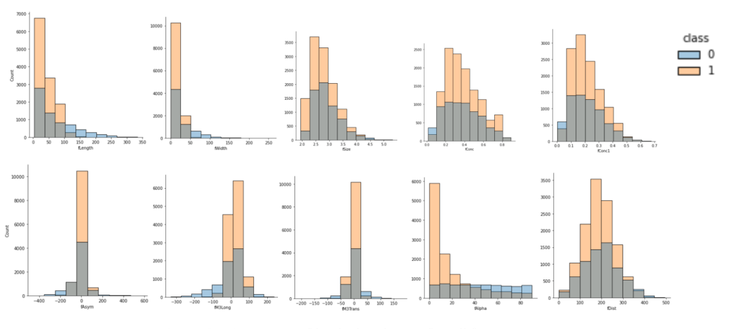

Figure 3. Histogram of both the classes for all features

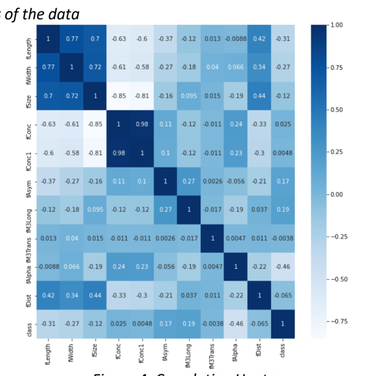

Figure 4. Correlation Heatmap

3. Model Selection:

3.1. Decision Tree:

- A decision tree algorithm starts at the root node, tests the attribute, and then moves down the tree branch corresponding to the value of the attribute. The process is repeated until a terminal node is reached where a final decision is made(Jegadeeshwaran and Sugumaran, 2013).

3.1.1. Pros:

- Easy to implement.
- Missing values and outliers can be easily handled by Decision Tree algorithm.
- Indifferent to normalization of data.

3.1.2. Cons:

- Decision Tree algorithm isn't efficient when it comes to predicting continuous values.
- It has higher possibility of overfitting.
- This algorithm is highly sensitive to noise.
- Some problems may only work with exponentially large trees [2].

3.2. Random Forest:

- Random Forest is an ensemble of separately trained binary decision trees(Ravì et al., 2016).
- In Breiman's approach, at each node a small group of features or variable are selected randomly to split forming a collection of and, secondly, by calculating the best split based on these features in the training set. These trees are grown using CART methodology (Breiman et al., 1984) to maximum size, without pruning. This subspace randomization scheme is blended with bagging (Breiman, 1996; Buhlmann and Yu, 2002; Buja and Stuetzle, 2006; Biau et al., 2010) to resample, with replacement, the training data set each time a new individual tree is grown [3].

3.2.1. Pros:

- Upon introduction of appropriate weak learners (i.e tress) it reduces the chance of overfitting.
- It adapts to sparsity, in the sense that its rate of convergence depends only on the number of strong features and not on how many noise variables are present [3].
- It makes possible for a tree to recover from a bad split by increasing the sample size. If a branch repeatedly splits on noise, the tree terminates the bad branch [5].

3.2.2. Cons:

- Due to the complexity of the algorithm, it is difficult to interpret.
- Takes higher computational time.
- Again, visualization is difficult to interpret.

4. Hypothesis Statement:

- Considering that Random Forest will outperform any other model as per the study by R.K. Bock(2004).
- A significant improvement of the accuracy of a single classification tree can be observed by constructing a forest or an ensemble of predictors) [1].
- However, Random Forest is likely to take longer computational time.
- According to the study of R.K. Bock (2004), in ensembled modeling the quality of the classification remains unchanged after 50 trees.

5. Description of the choice of training and evaluation methodology:

- The dataset is split in such a manner that 80% comprises of training data and the rest is test data.
- Techniques like feature selection, cross validation and optimization of hyperparameters were used to improve the quality of the classification of the images.
- A similar approach of the study by R.K. Bock(2004) was followed in terms of selection of hyperparameters.
- Based on how the models perform on the train dataset, two best performing models were picked from each of the machine learning algorithm to compare the results.
- The results include the generalization error, validation error, AUC, precision and other basic metrics that can measure the overall performance of the classification.

4. Choice of parameters and experimental results:

4.1. Decision Tree:

4.1.1. Experimental results:

- Normalizing the data didn't improve the accuracy. Hence, the experiment was continued using data without normalization.
- The accuracy obtained by manually tuning the hyperparameters and Bayesian optimization algorithm provided similar accuracy in terms of cross validation and prediction of test data. But the model with manual tuning of hyperparameters is considered for this study due to it's consistency.

4.1.2. Choice of parameters:

- The maximum split was considered by plotting a histogram of the splits of the model created using default options of MATLAB and then taking the mean of the splits.
- The minimum leaf size was estimated by
- For Split criterion, Gini's diversity index was considered as Gini's impurity is more efficient in terms of computational power. Also, impurity measures are quite consistent with each other.
- The importance of predictors were determined by summing changes in the risk due to splits on every predictor and dividing the sum by the number of branch nodes. However, the default selection seemed to be a better choice of parameters.

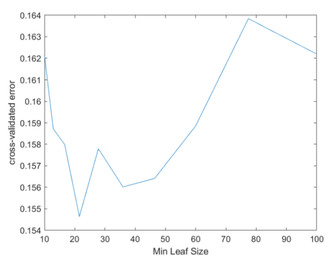

Figure 5. DT – Error vs Min Leaf size

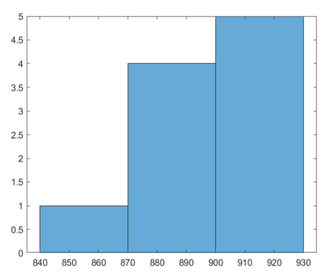

Figure 6. DT – Histogram of maximum split

4.2. Random Forest:

4.2.1. Experimental results:

- Again, normalizing the data didn't help in improving the accuracy of the model.
- Most of the default parameters provided similar results as the model which used Bayesian optimization algorithm to tune the hyperparameters.
- Of all the ensembled learning algorithm, Bagging (Bootstrapping aggregation) was used to classify the images and the model has performed well with good test accuracy.

4.2.2. Choice of parameters:

- Trees were selected as weak learners for the ensembled algorithm.
- As per the research paper [1], 50 trees could be the ideal parameter of the RF model for this dataset. And it was confirmed by the plot a graph of errors against several number of trees. In Figure , the out of bag error and classification error decreases significantly at 50 trees. However, with the introduction of 80 weak learners the error is almost zero which indicates high variance. And after that as we increase the weak learners, the error decreases which could possibly happen due to overfitting.
- The default hyperparameter measures seemed to work in favour of the accuracy of the model.

Metrics	RF	DT
AUC	0.927	0.889
Precision	0.8722	0.8306
Recall	0.848	0.8149
F1 Score	0.859	0.8227
Sensitivity	0.867	0.802
Specificity	0.877	0.859
Training Accuracy	87.900	84.543
Testing Accuracy	87.408	84.096
Classification error	0.126	0.1590
Resubstitution error	1.9716e-04	0.1172
KfoldLoss	0.121	0.155
Out of Bag Loss	0.128	NA

Figure 7. Metrics of DT and RF

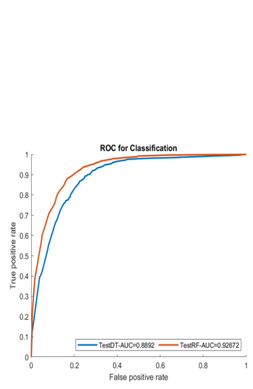

Figure 8. ROC curve of both the models with AUC values.

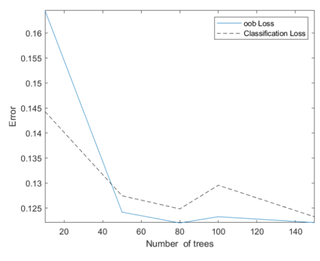

Figure 9. RF – Error vs Weak Learners

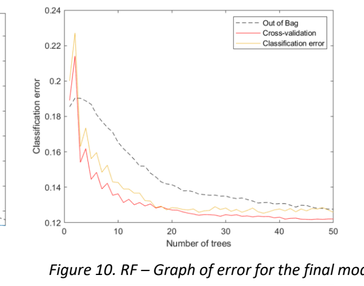

Figure 10. RF – Graph of error for the final model

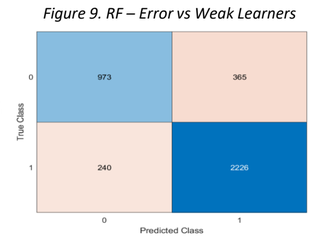

Figure 11. DT – Confusion Matrix

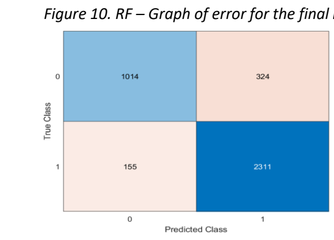

Figure 12. RF – Confusion Matrix

6. Analysis and critical evaluation of results:

- Random forest outperformed Decision Tree and it is well aligned with the hypothesis because of the bootstrapping procedure which reduces the variance without increasing the bias.
- Increase in accuracy is just by ~3% but the hypothesis of significant improvement of accuracy from a single classification tree to an ensembled algorithm get rejected. This could have happened because of careful choice of hyperparameter tuning in Decision tree.
- Bootstrapping aggregation method is an excellent methodology due to which the out of bag samples were used to predict the class of the images and the loss provided a fair picture of the model without even using cross validation for initial experimental trials.
- Initially, simple models were created with and without normalized data. And it was observed that both the algorithms were indifferent to normalization.
- fLength and fAlpha were determined as the two most important predictors. However, it could be clearly deduced from the scatter plot that there was no clear classification of the classes and most of them were overlapping each other. After that two models were created in both Decision Tree and Random Forest, the accuracy of the model went down and the generalised error increased. Again, this clearly proved that rest of the predictors have a major role to play in the models.
- In Decision Tree and Random Forest, Bayesian Optimization algorithm was used to tune the hyperparameters. But this algorithm is stochastic and returns different results of the hyperparameters with 30 iterations and no fixed time limit. In Random Forest, the Elapsed time was ~ 482.34 seconds to complete the processing of the model and for Decision Tree it took ~ 29.59 seconds.
- Clearly, Bayesian Optimization algorithm was taking longer computational time with inconsistent results without any significant increase in the accuracy. Hence, for both the algorithms this approach didn't seem to be ideal.
- Although, Random Forest has outperformed Decision Tree in terms of accuracy and generalized error but it was taking longer computational time. The average elapsed time was ~3.5 seconds. However, including 50 bins in the model helped to decrease the time. It almost reduced to ~1.7 seconds.
- The computational time for Decision Tree (elapsed time - ~0.06 seconds) is less than Random Forest. Hence, the hypothesis regarding computational time is accepted.
- The tuning of hyperparameters in Decision Tree was conducted precisely. The minimum leaf size was chosen with respect to minimum error and Gini's impurity was used to decide the number of split at each node. However, in Figure 6 the mean of the histogram was used to determine the maximum number of split.
- For Random Forest, as mentioned in the choice of parameter section, 50 weak learners proved to provide better results in agreement to the hypothesis. Default settings were used for rest of the hyperparameters.
- Maximum number of decision splits were considered as n-1, n is the number of observation and at each split 3 predictors i.e. square root of total predictors were randomly selected [4].
- As per the confusion matrix of decision tree, although the number of misclassified class is image originating due to gamma rays but if the ratio is considered then the image constructed due to hadronic shows was misclassified the most.

7. LESSONS LEARNED:

- Tuning hyperparameter and finding the right parameter is quite critical.
- Deep domain knowledge which is the mechanism behind the construction of the images originating from Cherenkov Telescope could have helped to explore the role of the features.
- Features shouldn't be dropped just because of multicollinearity.
- Accuracy of any Machine Learning algorithm cannot solely outperform other algorithms. Computational time or model complexity can overpower the accuracy of any model.

8. FUTURE WORK:

- Outliers can be removed to check if the assumption that these models are robust to outliers is absolute or not.
- Carefully noise can be added to check the authenticity of the model in .
- The default settings of pruning of the trees can be altered to explore it's impact on the model.
- Parallel Bayesian Optimization algorithm can be used to observe the change in accuracy and decrease in computational time.
- Different other Machine Learning or ensembled algorithms can be developed to validate if Random Forest outperform those models as well.

REFERENCES:

[1] Bock, R.K., Chilingarian, A., Gaug, M., Hakl, F., Hengstebeck, T., Jirina, M., Klaschka, J., Kotrc, E., Savicky, P., Towers, S., Vaicilius, A., Wittek W. (2004).
[2] INM431: Machine Learning lecture notes
[3] Gerard Biau , "Analysis of a Random Forests Model", Journal of Machine Learning Research 13 (2012) 1063-1095 (2012)
[4] Statistics and Machine Learning Toolbox
[5] Ishwaran, H. The effect of splitting on random forests. Mach Learn 99, 75–118 (2015)