Comparative Analysis of Decision Tree Based Classification Methods

Daiwei Lin, Navni Gupta, and Vaishali Singh
Group-30
University of Waterloo, Waterloo

***Abstract*— In data mining, classification of objects based on their features into predefined categories is a widely studied problem. It is one of the significant techniques with application in fraud detection, Medical diagnosis, Artificial Intelligence and many other fields[9]. There are many classification algorithms available but decision trees are most commonly used because of their ease of implementation and are easier to understand compared to other classification algorithms. In this paper, we have investigated three decision tree methods- CART, C4.5 and CHAID and compared them on the basis of different parameters which we have discussed later. For investigating these algorithms, we proposed to implement them from scratch in python. Further, we have taken the results from the previous researches in which they implemented these algorithms using WEKA data mining tool as benchmark and compared our results with it.The three data sets on which we have performed the comparative analysis are Spambase, Breast Cancer Wisconsin and Chronic Kidney Disease data set.**

***Index Terms*— decision tree; data mining; comparative analysis; CART; C4.5; CHAID; spam detection; medical diagnosis; splitting criterion; accuracy; memory; training time**

I. Introduction

In machine learning and statistics, classification is an instance of supervised learning approach in which the computer program learns from the data input given to it and then uses this learning to classify new observation. This data set may simply be bi-class (like identifying whether the person is male or female or that the mail is spam or non-spam) or it may be multi-class too. In classification, a given data set is divided into training and test data.The training data is used in building the classification model, while the test data is used in validating the model. The model is then used to classify and predict new set of data that is different from both the training and test data.[15]

There are many classification algorithms in the field of machine learning. Among them, decision tree algorithms are most commonly used because decision trees often mimic the human level thinking. Hence, making it easier for us to understand the data and make some good interpretations.

Decision tree builds classification models in the form of a tree structure. Classification tree models are those where the target variable can take a discrete set of values. In these tree structures, leaves represent class labels and branches represent conjunctions of features that lead to those class labels.

In this study, we will try to understand the nature of different decision tree methods by implementing them on multiple data sets. Then we will compare their performance in terms of accuracy, time, memory, size of tree and overfitting. We have taken multiple parameters to compare these methods in order to get a clear understanding about which among them performs better. These parameters have been discussed in detail later.

II. Literature Review

We studied various research papers regarding performance evaluation of various classification algorithms which were performed on WEKA. Some of them are described here.

In Quinlan's early work[1], ID3, is proposed to solve chess end game problem. It splits the data set at the feature with maximum information gain(IG). The attribute values are discrete, and ID3 couldn't handle missing values very well. Later, Quinlan proposed an extension of ID3, C4.5[2]. This algorithm supports both discrete and continuous feature values and can handle missing data. C4.5 also support pruning which reduces the probability of overfitting. CHAID is another tree building method by GV Kass[3]. It tries to find the best explanatory variable using Chi-square test and also the best merger of categories. Each node can have more than two child nodes. Different from CHAID, CART is a binary tree introduced by Breiman, Leo in 1984[4]. The dependent variable of CART can be categorical only.

Overall, all three methods, ID3, C4.5 and CART, use impurity measures to choose the splitting attribute and the split value/s. ID3 uses information gain while the successor, C4.5 uses gain ratio, and CART uses Gini coefficient for impurity measurements. Unlikely, CHAID uses Chi-square or F-statistic to choose the splitting variable[3].

Aman Kumar Sharma et al[5], in this research paper they have performed experiments in order to determine the classification accuracy of different decision tree algorithms. They calculated the classification accuracy in terms of which algorithm better determines whether an email is spam or not in WEKA. Four algorithms namely ID3, J48, Simple CART and ADTree were compared on the basis of different percentage of correctly classified instances. According to their results, C4.5(J48) has better classification accuracy than the other three algorithms.

Hiba Arsi et al.[7], in this reference they have employed four main algorithms: SVM(Support Vector Machine), NB(Naive bayes), k-NN(k-Nearest Neighbour) and C4.5 on the Wisconsin Breast Cancer (original) data sets. They have compared the efficiency and effectiveness of those algorithms in terms of accuracy, precision, sensitivity and specificity to find the best classification accuracy.

E.Avci et al.[8], in this study performance comparison was made using WEKA software with NB(Naive bayes), k-star, SVM(Support Vector Machine) and J48(C4.5) algorithms using data set named "Chronic_Kidney_Disease" extracted from UCI database. Accuracy, precision, sensitivity and F-measure parameters are used as the performance criteria. According to the results obtained from this performance comparison, the J48(C4.5) classifier had better results.

From the literature review, all the three data sets have been analyzed by implementing the classification algorithms using WEKA data mining tool. Among all the algorithms used for comparison

in these references, C4.5(J48) usually performs better than others. Therefore, we have taken the results of C4.5(J48) for these data sets as a benchmark to compare it with our implementation of CART, C4.5 and CHAID(in python).

III. Methods and material

For our implementation of decision tree methods we have taken three data sets. All these data sets have been taken from UCI Machine Learning Repository[18].

A. Data set

1) Spam-base Data set: Nowadays, email has become one of the quickest and most expensive means of communication. However, popularity of email has further increased spam mails during the past years. The first data set that we have taken is Spam-base data set. This data set consists of 4601 instances and 57 features. Out of these features, 56 are continuous and 1 is nominal(Class attribute). We have used decision tree methods to categorise the email as spam or non-spam.

2) Breast Cancer Wisconsin Data set: Breast Cancer is a malignant or benign tumour, inside breast, wherein cells divide and grow without control. Scientist have tried to know the exact reason behind breast cancer, as there are a few risk factors which increase the likelihood of a woman developing breast cancer. Age, genetic risk and family history are some such factors being considered for breast cancer. For the aforementioned problem, we have taken Breast Cancer Wisconsin data set which consists of 699 instances, 9 attributes, 1 class attribute and also has some missing values.

3) Chronic Kidney Disease Data set: Currently, kidney disease is one of the most common public health problem in the world. It can be very dangerous if not immediately treated on time, and may be fatal. If the doctors have a good tool that can identify patients who are likely to have kidney disease in advance, they can heal the patients in time. For this we have taken chronic kidney disease data set which has 400 instances, 11 numeric and 14 nominal features along with some missing values.

B. Methods

1) CART: In 1984, Leo Breiman, Jerome Friedman, Richard Olsen and Charles Stone jointly developed Classification and Regression Tree(CART)[4]. A CART tree is a binary decision tree that is constructed by splitting a node into two child nodes repeatedly, beginning with the root node that contains the whole learning sample. The basic idea of tree growing is to choose a split among all the possible splits at each node so that the resulting child nodes are the "purest". It deals with both numerical and categorical features.

Splitting Criterion: CART uses Gini impurity (Eq.1) which is a measure of how often a randomly chosen element from the set would be incorrectly labeled if it was randomly labeled according to the distribution of labels in the subset. The Gini impurity can be computed by summing the probability p_i of an item with label i being chosen times the probability $\sum_{k \neq i} p_k = 1 - p_i$ of a mistake in categorizing that item. It reaches its minimum (zero) when all cases in the node fall into a single target category.

$$I_G(p) = 1 - \sum_{i=1}^{J} p_i^2 \tag{1}$$

2) C4.5: C4.5 is an algorithm used to generate a decision tree developed by Ross Quinlan which is an extension of Quinlan's earlier ID3 algorithm. C4.5 builds decision trees from a set of training data in the same way as ID3, using the concept of information entropy[2]. The training data is a set $S = s_{1,s_2,...}$ of already classified samples. Each sample S_i consists of p-dimensional vector$(x_{1,i}, x_{2,i}, ..., x_{p,i})$, where the x_j represents attribute values or features of the sample, as well as the class in which s_i falls. At each node of the tree, C4.5 chooses the attribute of the data that most effectively splits its set of samples into subsets enriched in one class or the other.

Splitting Criterion: The splitting criterion is information gain(difference in entropy). Information Gain measures how much "information" a feature gives us about the class.
Entropy of a set with k classes is calculated as:

$$H = -\sum_{i=1}^{k} p_k \times log_2 p_k \tag{2}$$

Then the change in entropy , or Information Gain, is defined as the difference between the entropy of parent node and weighted sum (m_i) of all $n(n \geq 2)$ children nodes:

$$\Delta H = H_{parent} - \sum_{i=1}^{n} \frac{m_i}{m} H_i \tag{3}$$

3) CHAID: CHAID(Chi- square Automatic Interaction detector) analysis is an algorithm used for discovering relationship between a categorical response variable and other categorical predictor variable. It is useful when looking for patterns in data sets with lots of categorical variables and is a convenient way of summarizing the data as the relationship can be easily visualized. This algorithm accepts nominal or ordinal categorical predictors. When predictors are continuous, they are transformed into ordinal predictors before using the algorithm. Chi-square tests are applied at each of the stages in building the CHAID tree, to ensure that each branch is associated with a statistically significant predictor of the response variable[13].

Splitting Criterion: To choose the split of the predictor variable with the smallest (Bonferroni) adjusted p-value, i.e, the predictor variable that will yield the most significant split; if the smallest adjusted p-value for any predictor is greater than user-specified α_{split} value, then no further splits will be performed and the respective node is a terminal node.

IV. Implementation

We have implemented the above mentioned decision tree methods by following the pseudo codes described below.

1) Algorithm for CART: A tree is grown starting from the root node by repeatedly using the following step on each node in Algorithm 1. We referred to online resources[19] when implementing this algorithm.

Algorithm 1 CART

1: Find each feature's best split. For each feature with k different values there exist k-1 possible splits. Find the split, which maximizes the splitting criterion. The resulting set of splits contains best splits(one for each feature).
2: Find the node's best split. Among the best splits from step 1 find the one, which maximizes the splitting criterion.
3: Split the node using best node split from step 2 and repeat from step 1 until stopping criterion is satisfied.

2) Algorithm for C4.5: The implementation of C4.5 is almost similar to what we have done in CART 1. The differences between their implementation are:

- A C4.5 tree is not always a binary tree because for categorical features the number of branches for that feature is equal to its class labels.
- The splitting criterion used for C4.5 is Information Gain.

In Quinlan's work[2], the splitting criterion of C4.5 can be improved further by normalizing Information Gain with Splitting Information defined in Eq.4 and Eq.5. In Eq.4, $|T_i|$ is the size of ith partition and $|T|$ is the size of the complete set. The split info is a measure of how much information is generated by such partition.

$$SplitInfo(X) = -\sum_{i=1}^{n} \frac{|T_i|}{|T|} \times log_2(\frac{|T_i|}{|T|}) \tag{4}$$

$$GainRatio(X) = Gain(X)/SplitInfo(X) \tag{5}$$

We firstly implemented C4.5 using Gain Ratio, however, the accuracy result is very poor. Then we revert back to using Information Gain (Eq.1). The results shown in this paper are based on Information Gain only.

3) Algorithm for CHAID: This algorithm consists of three parts - Merging, Splitting and Termination.

Merging: For each predictor variable X, merge non-significant categories. Each final category of X will result in one child node if X is used to split the node. Merging steps are illustrated in Algorithm 2.

Algorithm 2 CHAID - Merge

if X has 1 category **then**
  set the adjusted p-value to be 1
  return adjusted p-value
end if
loop
  if X has 2 categories **then**
    break
  else
    loop through all allowable pair of categories of X. Find the pair $pair_{max} = (cat_i, cat_j)$ that is least significantly different, which gives the largest p-value with respect to the dependent variable Y.
  end if
  if p value of $pair_{max} \geq \alpha_{merge}$ **then**
    this pair is merged into a single compound category. Then a new set of categories of X is formed.
  end if
end loop
Compute adjusted p-value for the merged categories by applying Bonferroni adjustments
return adjusted p-value

Splitting: The best split for each predictor X_{best} is found in the merging step. The splitting step selects which predictor to be used to best split the node.
1. Select the predictor X_{best} that has the smallest adjusted p-value (i.e., most significant).
2. If adjusted p-value of X_{best} is less than or equal to a user-specified alpha-level α_{split} (alpha_split), split the node using this predictor. Else, do not split and the node is considered as a terminal node.

Stopping: The stopping step checks if the tree growing process should be stopped according to the following stopping rules.

- If a node becomes pure, stop splitting
- If all cases in a node have identical values for each predictor, the node will not be split.
- If the current tree depth reaches the user specified maximum tree depth limit value, the tree growing process will stop.

A general diagram of algorithm is illustrated in Algorithm 3.

Algorithm 3 CHAID - Construct Tree

Recursionloop
for all predictors X **do**
  Merge categories of X
end for
Select the best predictor X_{best}
if split **and** not stop **then**
  Create child nodes for all n merged categories in X_{best}
  for each child node **do**
    Recursionloop
  end for
else
  Terminate as leaf node
end if
end loop

Continuous Features: In the original paper presented by Kass[3], it only deals with categorical features. In our approach, we discretize continuous features into bins, in which equal number of samples are placed. The maximum number of bins k is specified by users. If samples have a large number of duplicate values and separating them into k bins is not possible, k is reduced to the maximum bins allowed. After pre-processing the continuous features, the algorithms treats all features as categorical.

V. Testing

While implementing the decision tree we will go through the following two phases:

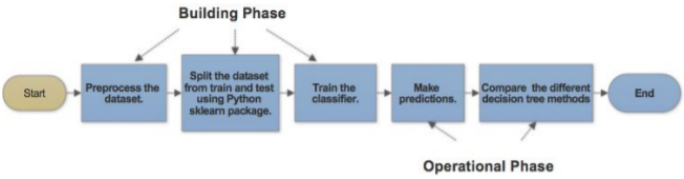


Fig. 1: Approach for testing our implementation

A. Building Phase

1) Preprocess the data set: We have all the data sets in the form of csv files. We loaded each data set in the form of a data

frame. Out of three data sets, two(Breast Cancer and Chronic Kidney disease) of them have missing values. We handled the missing values by replacing them with the mode of that attribute.

2) Split the data set: We have randomly splitted the data set into train and test using the train_test_split module of sklearn package.

3) Getting optimal Parameters: For getting the optimal parameters we have performed 5-fold-cross validation with GridSearchCV().

B. Operational Phase

After getting optimal parameters from the building phase, we will get best classifiers and train the test set. Then we calculate the test score(accuracy) for our test set.

We repeat step 2, 3 and operational phase for 10 times for getting stable results. After performing it for 10 times we calculate the average of statistics for each decision tree method, including training and testing accuracy, precision, recall, F1-score and speed.

To obtain the memory usage during the training, we use the Memory Profiler package from PyPi library. We put test cases in python files and run them from the console.

All tests are performed on the computer with AMD Ryzen 5 1600 Six-Core Processor 3.5GHz and 8GB memory.

VI. Results and Discussions

In this section, we have divided our results into three sections. Table I, II, and III demonstrate the comparison of our implementation of three decision tree methods(CART, C4.5 and CHAID) on the basis of precision, recall and F1-score.

In all the three data sets, we observed that C4.5 has higher precision, recall and F1-score than the other two methods.

Algorithms	CART	C4.5	CHAID
Precision(SPAM)	0.899	0.887	0.883
Precision(NO-SPAM)	0.916	0.936	0.896
Recall(SPAM)	0.875	0.904	0.823
Recall(NO-SPAM)	0.932	0.923	0.933
F1-score(SPAM)	0.887	0.896	0.914
F1-score(NO-SPAM)	0.924	0.929	0.852

TABLE I: Comparison of decision tree methods on Spambase dataset.

Algorithms	CART	C4.5	CHAID
Precision(Malignant)	0.901	0.928	0.972
Precision(Benign)	0.953	0.956	0.918
Recall(Malignant)	0.907	0.922	0.843
Recall(Benign)	0.948	0.959	0.986
F1-score(Malignant)	0.902	0.925	0.903
F1-score(Benign)	0.950	0.957	0.951

TABLE II: Comparison of decision tree methods on Breast Cancer Wisconsin dataset.

Algorithms	CART	C4.5	CHAID
Precision(CKD)	0.976	0.992	0.931
Precision(NOTCKD)	0.949	0.962	0.954
Recall(CKD)	0.972	0.979	0.955
Recall(NOTCKD)	0.958	0.985	0.930
F1-score(CKD)	0.974	0.985	0.954
F1-score(NOTCKD)	0.953	0.973	0.929

TABLE III: Comparison of decision tree methods on Chronic Kidney Disease dataset.

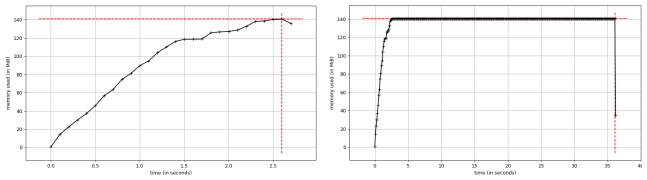
Now we have compared the accuracy, training time(speed), memory, size of the tree and over-fitting for three decision tree methods namely CART, C4.5 and CHAID. Table IV, V and VI demonstrate the comparison between three decision tree methods on the basis of above mentioned parameters for all the three data sets.

1. **Accuracy** - In all the three data sets we can see that accuracy of C4.5 is better than CART and CHAID. CHAID performs equally good in Breast Cancer data set, but worse than the other two in the rest two data sets. It should be noted that Breast Cancer data set only has categorical features, while Spambase dataset is fully numerical and Chronic Kidney Disease data set is a mixture of both. According to the nature of data sets, it is believed that discretizing continuous features will reduce the accuracy of the model. This might be improved if we adjust the binning method used by CHAID. We can allow different number of samples per bin when many duplicate values occur in the data, then potentially we could have more bins. Thus the number of categories increases, which could help improve the model accuracy.
It is one of the most important parameters for comparing the algorithms. The goal of the Machine Learning model is to learn patterns that generalize well for unseen data instead of just memorizing the data that it was shown during training. Once you have a model, it is important to check if your model is performing well on unseen examples that you have not used for training the model. To do this, you use the model to predict the answer on the evaluation data set (held out data) and then compare the predicted target to the actual answer (ground truth).
We use a number of metrics to measure the predictive accuracy of a model. The choice of accuracy metric depends on the Machine Learning task.

2. **Training Time/Speed** - In spambase data set, CHAID is comparatively faster in training the data set than C4.5 and CART where as in other two data sets(Breast Cancer Wisconsin and Chronic Kidney Disease) CART is comparatively faster than the other two. Also in these two data sets, CHAID is comparatively slower in training the data. We believe it is because in CHAID, every time before the data is trained, the continuous features are converted into categorical features. The time of conversion is more dominant when the size of dataset is small, for example Breast Cancer Wisconsin data set and Chronic Kidney Disease data set. However, discretizing continuous features into categorical features will reduce the complexity of data, thus reducing the training time. This factor becomes more dominant when the size of the data set is large, causing CART and C4.5 taking more time to train than CHAID.
The training time is a second important factor that affects the suitability of a classification algorithm regarding an unknown data set. With large amount of data, CHAID can be used to predict faster than CART and C4.5, with a cost in accuracy. If the dataset is small, CHAID predicts as good as CART and C4.5 but slower.

3. **Memory** - According to Table IV,V,VI, in all the data sets, we observe that the memory consumption for implementing CHAID is comparatively more than the other two decision tree methods. But the differences among algorithms are marginal.
We have also compared the memory consumption using sklearn packages with the memory consumption of our implementation. A comparison on Spam Base Dataset is shown in Fig.2. From the figure, we can see the difference in the shape of curves, where sklearn's implementation gradually increases the memory consumption while our implementation reaches maximum memory usage at the begin-

ning. We believe it is because our implementation is not optimized, thus taking more memory than standard library. In addition, sklearn's package is not fully consistent with CART and is an optimized variation.



(a) (b)

Fig. 2: Memory consumption of using CART on Spam Base Dataset. (a) DecisionTreeClassifier from sklearn (b) Our implementation

4.**Size of the tree**- We are calculating the size of the tree by its max-depth, number of internal nodes and leaf nodes.
In case of Breast Cancer Wisconsin data set, the size of C4.5 decision tree i.e. number of nodes in C4.5 decision tree is larger than that of decision tree constructed by CART and C4.5 methods. In Chronic kidney disease data set, size of decision tree constructed by CHAID algorithm has more number of nodes than the decision tree constructed by CART and C4.5. In Spambase data set, the size of C4.5 is largest, followed by CHAID and the smallest is CART.
C4.5 produces largest trees in two out of three data sets. The depths of trees given by three algorithms are similar. This means that C4.5 grows more horizontally than CHAID. CART is binary tree only, therefore we expect it to have less branches but grow deeper in order to get competitive accuracy to C4.5 and CHAID. But the deeper depth is only observed in the case of Chronic Kidney Disease data set.

5.**Overfitting**- A symptom of over fitting is that the classifier performance on the train set is better that the one on the test set. All the models that we implemented tends to overfit even after pre-pruning because in all data sets, the training accuracy are always higher than testing accuracy.

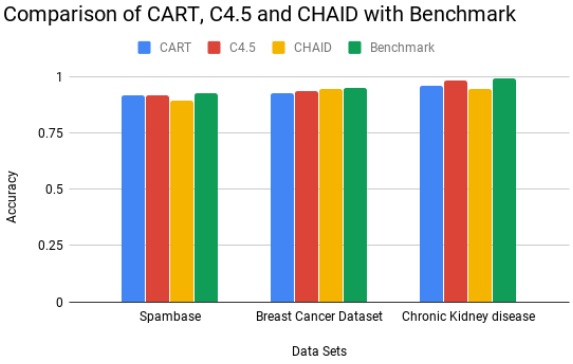


Fig. 3: Comparison Test and Train score(Spambase).

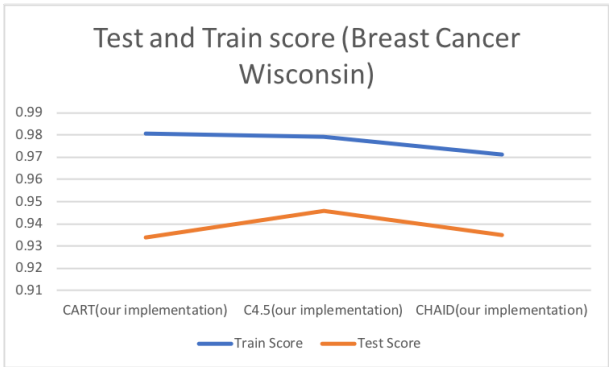


Fig. 4: Comparison Test and Train score(Breast Cancer wisconsin)

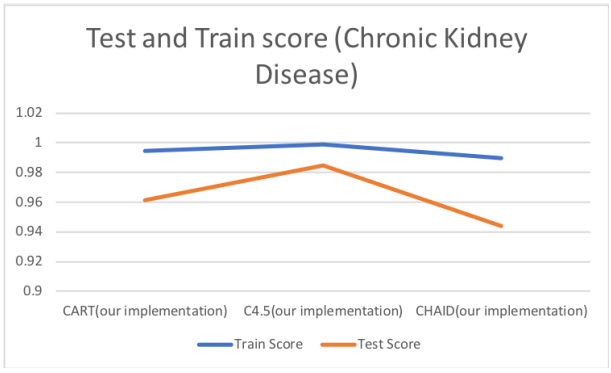


Fig. 5: Comparison Test and Train score(Chronic kidney disease)

The above graphs show that the train score for all the decision tree methods that we implemented is higher than the test score for all the data sets which clearly tells that these models overfit.

Parameters	CART	C4.5	CHAID
Speed(Training Time in secs)	502.262	462.102	236.526
Memory	143.2 MiB	144.8 MiB	146.3 MiB
Accuracy	0.916392	0.915734	0.891376
Max-depth	10	10	8
Number-of-internal-nodes	65	165	133
Number-of-leaf-nodes	66	166	75
Over-fitting	Yes	Yes	Yes

TABLE IV: Comparison of decision tree methods on Spambase

Parameters	CART	C4.5	CHAID
Speed(Training Time in secs)	1.05444	1.09395	5.91954
Memory	140.6 MiB	141.0 MiB	142.5 MiB
Accuracy	0.93376	0.94588	0.93506
Max-depth	5	5	5
Number-of-internal-nodes	10	46	15
Number-of-leaf-nodes	11	37	10
Over-fitting	Yes	Yes	Yes

TABLE V: Comparison of decision tree methods on Breast Cancer Wisconsin

Parameters	CART	C4.5	CHAID
Speed(Training Time in secs)	0.84501	0.89288	4.13153
Memory	140.3MiB	140.2 MiB	142.2 MiB
Accuracy	0.96136	0.98485	0.94394
Max-depth	8	5	5
Number-of-internal-nodes	8	4	17
Number-of-leaf-nodes	9	5	10
Over-fitting	Yes	Yes	Yes

TABLE VI: Comparison of decision tree methods on Chronic Kidney Disease

We lastly compared the accuracy of our implementation of decision tree methods with benchmark that we took as a reference. This benchmark is in fact C4.5 implemented using WEKA data mining tool. Note that the accuracy of algorithms is not affected by programming languages.

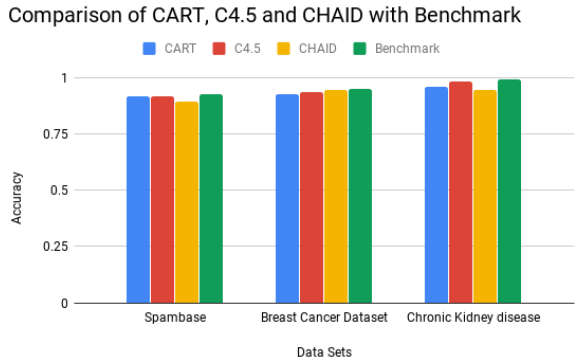


Fig. 6: Comparison of CART, C4.5 and CHAID with benchmark.

We observe that the results of the decision tree methods that we implemented are close enough to the benchmark.

VII. Conclusion

Several data mining classification techniques can be applied on the data sets mentioned above. In this paper we have used decision tree classification methods, mainly CART, C4.5 and CHAID on three problems. We have compared these methods on multiple parameters for prediction. With all our results we can conclude that C4.5 has better accuracy than the other two methods (CART and CHAID). In terms of training time, CART is comparatively faster than CART and CHAID for two of the data sets except the Spambase data set. In case of Spambase data set, CHAID performs comparatively faster than the other two methods. In terms of memory we cannot conclude which one of them is most space efficient as all of them are using almost same memory. All the decision tree models that we implemented are over-fitting.

VIII. Future Works

We can later do the analysis of decision tree algorithms by introducing post pruning methods. Also, we can include the feature selection to make our decision tree algorithms more optimized and can study their behaviour after applying feature selection methods.

References

[1] Quinlan, J. Ross. *"Induction of decision trees."* Machine learning 1.1 (1986): 81-106.
[2] Quinlan, J. Ross. *C4. 5: programs for machine learning.* Elsevier, 2014.
[3] Kass, Gordon V. *"An exploratory technique for investigating large quantities of categorical data."* Journal of the Royal Statistical Society: Series C (Applied Statistics) 29.2 (1980): 119-127.
[4] Breiman, Leo. *Classification and regression trees.* Routledge, 2017.
[5] Sharma, Aman Kumar, and Suruchi Sahni. *"A comparative study of classification algorithms for spam email data analysis."* International Journal on Computer Science and Engineering 3.5 (2011): 1890-1895.
[6] A. Subasi, S. Alzahrani, A. Aljuhani and M. Aljedani , *"Comparison of Decision Tree Algorithms for Spam E-mail Filtering."* IEEE, 2018.
[7] Hiba Asri, Hajar Mousannif, Hassan Al Moatassime and Thomas Noel, *"Using Machine Learning Algorithms for Breast Cancer Risk Prediction and Diagnosis."* Procedia Computer Science 83, pp. 1064-1069, 2016.
[8] Avci, Engin, et al. *"Performance comparison of some classifiers on Chronic Kidney Disease data."* 2018 6th International Symposium on Digital Forensic and Security (ISDFS). IEEE, 2018.
[9] Lavanya, D., and K. Usha Rani, *Performance evaluation of decision tree classifiers on medical datasets* Sri Padmavathi Mahila Visvavidyalayam Tirupati - 2 , Andhra Pradesh, 2011, pp. 1–4.
[10] Reif, Matthias, Faisal Shafait, and Andreas Dengel.*"Prediction of classifier training time including parameter optimization."* Annual Conference on Artificial Intelligence. Springer, Berlin, Heidelberg, 2011.
[11] *"How to choose algorithms for Azure Machine Learning Studio."*[Online]. Available: https://docs.microsoft.com/en-us/azure/machine-learning/studio/algorithm-choice
[12] *"Popular Decision Tree: CHAID Analysis, Automatic Interaction Detection."*[Online]. Available: http://www.statsoft.com/Textbook/CHAID-Analysis
[13] *"CHAID (Chi-square Automatic Interaction Detector)."*[Online]. Available: https://select-statistics.co.uk/blog/chaid-chi-square-automatic-interaction-detector
[14] Quinlan, John Ross. *"Decision trees and multivalued attributes."* Machine intelligence 11 (1988): 305-318.
[15] Anyanwu, Matthew N., and Sajjan G. Shiva. *"Comparative analysis of serial decision tree classification algorithms."* International Journal of Computer Science and Security 3.3 (2009): 230-240.
[16] *"CART Algorithms"*[Online]. Available: https://www.ibm.com/support/knowledgecenter/en
[17] Loh, WeiYin. *"Classification and regression trees."*Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery 1.1 (2011): 14-23.
[18] *"UCI Machine Learning Repository."*[Online]. Available: https://archive.ics.uci.edu/ml/datasets
[19] Brownlee, Jason (2016). *"How To Implement The Decision Tree Algorithm From Scratch In Python"*. Available: https://machinelearningmastery.com/implement-decision-tree-algorithm-scratch-python/