Programming Assignment 3: A Comparative Analysis of Reduced-error Pruning in Decision Tree

Wanyi (Julia) Dai WDAI17@JH.EDU
3400 North Charles St
Baltimore, MD 21218, USA

Abstract

This paper provides a comprehensive overview of Programming Assignment 3 from the course "Introduction to Machine Learning." The primary objective of this project is to implement decision tree models for both regression and classification tasks, employing the Classification and Regression Trees (CART) algorithm for regression and the Iterative Dichotomiser 3 (ID3) algorithm for classification. Upon project completion, we will apply k×2 cross-validation on all the datasets with the models, accommodating hyperparameter tunings based on the best parameters (whether to use reduced-error pruning) for each dataset.

Keywords: Programming Assignment 3, decision tree, CART, ID3, reduced-error prune, hyperparameter tuning, k×2 cross-validation

1 Introduction

Machine learning is an ever-evolving field that empowers us to extract valuable insights and predictions from data. Decision trees are a fundamental and versatile tool capable of addressing classification and regression tasks within this multifaceted domain. This paper delves into the heart of the decision tree paradigm, focusing on two pivotal algorithms: Classification and Regression Trees (CART) for regression tasks and the Iterative Dichotomiser 3 (ID3) algorithm for classification. This exploration offers a comprehensive understanding of these algorithms, their applicability as tools for our experiments, and their adaptability to a wide range of datasets and real-world scenarios. This paper demonstrates how these models can be effectively harnessed to predict numerical values (regression) or categorize data into distinct classes (classification). A key aspect of our exploration lies in hyperparameter optimization, focusing on the critical decision of using reduced-error pruning. By conducting k×2 cross-validation, we fine-tune the models to identify the optimal parameters for each dataset. This approach ensures that our decision tree models are tuned accordingly and capable of delivering accurate and reliable predictions optimized to each dataset's best performance. In the following sections, we delve into the details of the CART and ID3 algorithms, their application across diverse datasets, and the results of our hyperparameter tuning. Through this exploration, we endeavor to shed light on the intricate world of decision trees and their potential to address real-world challenges.

2 Problem Statement

This paper aims to address this need by investigating the practical application of two significant decision tree algorithms, Classification and Regression Trees (CART) for regression tasks and the Iterative Dichotomiser 3 (ID3) algorithm for classification, with a focus on their adaptability to diverse datasets and their ability to improve predictions through hyperparameter tuning.

We hypothesize that decision tree models, precisely CART for regression tasks and the ID3 algorithm for classification, will exhibit varying behavior but better performance when applied to diverse datasets compared to the null model results from Project 1. With its ability to segment and model numerical data effectively, we anticipate that the CART algorithm will perform well in regression tasks. It is expected to capture complex relationships within datasets, resulting in accurate predictions. For classification tasks using the ID3 algorithm, we expect a high level of

adaptability to different datasets, including those with categorical features. ID3's capability to discern and partition categorical data into decision paths is well-suited for diverse scenarios. However, the concrete performance could vary depending on the dataset's size, outliers' presence, and relationships' complexity. The behavior of whether the dataset chooses reduced-error pruning would differ when different datasets are applied. We expect the choice of pruning is more likely to happen on larger datasets, as larger datasets could result in more enormous trees that could be unnecessary.

The categorization of the datasets we will be using for the experiment is as follows:
Regression Datasets: abalone, forest fires, machine
Classification Datasets: breast cancer Wisconsin, car, house votes

3 Datasets

Before conducting experiments or applying any algorithm to the dataset, conducting a thorough preliminary analysis of each dataset is essential. This analysis should unveil the dataset's inherent characteristics, as they can significantly influence the success or failure of the algorithm in question. Understanding these dataset attributes is foundational to making informed decisions and achieving optimal results in the subsequent data analysis or modeling processes.

3.1 Classification Data

Classification datasets hold a critical role in the study of machine learning, offering a fertile ground for understanding the intricate interplay under the algorithm of ID3. These datasets challenge us to categorize data points into discrete classes or categories, a task that decision trees excel at. Hence, we expect fewer complications in the decision tree generation, and the decision tree should perform well on the following data:

- **Breast Cancer Wisconsin**: This breast cancer data set was obtained from the University of Wisconsin, aiming at distinguishing between benign and malignant breast tumors.
    - **Row number:** 699
    - **Features:** [sample code, clump thickness, uniformity of cell size, uniformity of cell shape, marginal adhesion, single epithelial cell size, bare nuclei, bland chromatin, normal nucleoli, motoses].
        - Each feature is constructed by integers ranging from 1 to 10. The numerical features could be problematic when passing in an ID3 decision tree. However, given the nature of integers ranging from 1 to 10, they could be treated as categorical variables. This will be further addressed in the preprocessing section.
    - **Target range:** [2, 4]
- **Car**: The data is on evaluations of car acceptability based on price, comfort, and technical specifications.
    - **Row number:** 1728
    - **Features:** [buying, maint, doors, persons, lug boot, safety].
        - The car's features are all categorical, so it is expected to do well in the decision train fit. Many data points exist, but the reduced-error pruning might be unnecessary given the simple feature settings.
    - **Target range:** [unacc, acc, vgood, good]
- **House Votes**: This data set includes votes for each U.S. House of Representatives Congressmen on the 16 key votes identified by the Congressional Quarterly Almanac.
    - **Row number:** 435
    - **Features:** [handicapped, water project cost sharing, adoption of the budget resolution, physician fee freeze, El Salvador aid, religious groups in schools, anti-satellite test ban, aid to Nicaraguan contras, mx missile, immigration, synfuels corporation cutback,

education spending, superfund right to sue, crime, duty-free exports, export administration act South Africa]
        - There are a lot of features in this data, which could result in a vast tree, but the categories in each feature are simply [yes, no, ?(unsure)]. Thus, the depth of the tree could be controlled, but pruning could still be applied, considering the number of features is extensive.
    - **Target range:** [republican, democrat]

3.2 Regression Data

Regression datasets serve as a pivotal element in machine learning, allowing us to explore the intricate connection between data characteristics and the application of algorithm CART. While classification data categorizes information into discrete classes, regression data embarks on predicting continuous numerical values. However, as the process might include generalizations of the data, the improvement might not be as impressive as the classification data on the decision tree.

- **Abalone**: Predicting the age of abalone from physical measurements.
    - **Row number:** 4177
        - There are a lot of data points in this dataset. With most of the features being numeric, we could end up with a very deep decision tree that might need to be pruned to maximize the performance.
    - **Features:** [sex, length, diameter, height, whole weight, shucked weight, viscera weight, shell weight]
    - **Target range:** [1 to 29]
        - The relatively small target range could make the generalization easier and the leaf node closer to the actual value.
- **Machine**: The authors evaluated the estimated relative performance values using a linear regression method.
    - **Row number:** 209
    - **Features:** [vendor, model, MYCT, MMIN, MMAX, CACH, CHMIN, CHMAX, ERP]
        - The dataset machine has complicated features, primarily numerical, but the feature of model is categorical with many categories. This could result in a complicated tree.
    - **Target range:** [6 to 1150]
        - The massive range could make the prediction hard to be accurate.
- **Forest Fires**: This dataset uses meteorological and other data to predict the burned area of forest fires in the northeast region of Portugal.
    - **Row number:** 517
    - **Features:** [X, Y, month, day, FFMC, DMC, DC, ISI, temp, R.H., wind, rain, area]
    - **Target range:** [0 to 1090.84]
        - The massive range could make the prediction hard to be accurate. The data from the target variable is also very skewed (Figure 1). Hence, we could result in a skewedly deep tree. Pruning might help to cut off some unnecessary branches and generalize the tree more.

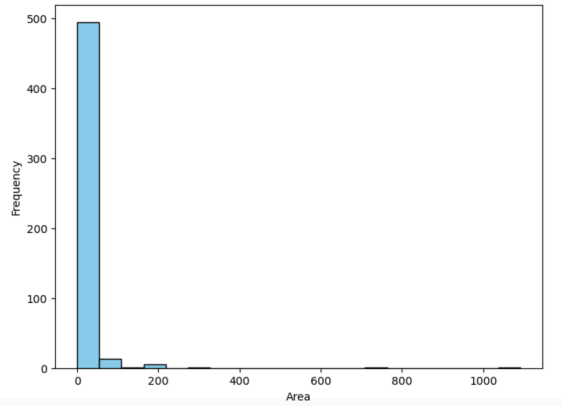

Figure 1. Distribution of Forest Fire Area

4 Preprocessing Steps

For us to experiment, several preprocessing steps exist on the datasets.

4.1 Loading Data

All datasets will be stored as Pandas DataFrames. These datasets are typically stored in .data files, and in most cases, they lack predefined headers. To address this, we will manually generate headers using information from .name files and apply them during the data-loading process. Additionally, the data loading function offers the flexibility to perform log transformations on specific columns, which can benefit various datasets. Decisions regarding header generation and log transformations are made based on each dataset's particular characteristics and requirements. This approach ensures that data is loaded and prepared in a manner that optimizes future performance and analysis based on the unique nature and context of each dataset.

4.2 Handling Missing Values

We first must handle the missing data before giving a dataset to a machine learning algorithm. Most of our datasets are complete, except the breast cancer Wisconsin data. The attribute Bare Nuclei has missing data denoted by "?". In this case, the missing value is credited with the column mean.

Note: house vote data also has "?", but it is denoted as a category of neither "yes" nor "no" instead of missing data.

4.3 Handling Categorical Data

Categorical data consists of discrete, distinct categories or labels representing different groups or classes in the dataset. They are often described as strings in the dataset, which could be hard to use in computations but not decision trees. Hence, all the categorical data in our datasets will remain what they are.

4.4 Handling Numerical Data

In the conventional ID3 algorithm, designed to work exclusively with categorical attributes, the treatment of numerical features can be cumbersome. Numerical data does not readily fit the paradigm of binary decisions, leading to concerns about the efficiency and adaptability of ID3 in handling such attributes. Consequently, incorporating numerical features into the ID3 framework necessitates careful consideration and, at times, specialized techniques.

There are generally two ways of handling numerical data in a decision tree:

1. Discretizing or binning: group the numerical data into bins to construct it as categorical.

2. Sort the data on the attribute and consider possible binary splits at midpoints between adjacent data points.

As required, we will be using the second approach in this project to applicable attributes. Among the classification datasets, the only one that contains numerical attributes is the breast cancer data. However, the numerical variables in this dataset have a discrete scale ranging from 1 to 10. With the discrete nature and the low range, they could be treated as each with ten categories, each represented by a number from 1 to 10. The primary benefit of doing so is to preserve information loss. Binary splits typically create a clear division between two categories based on a threshold. Converting a 1-10 scale into binary partitions would require selecting an arbitrary threshold, which may result in information loss. Even the project requirements suggest dividing the midpoints between data where class changes; saving the original numbers could preserve more information while the number of categories does not make too many difficulties for the decision tree generation process.

Hence, as explained, the original breast cancer feature data is kept and will be applied to the decision tree directly.

5 Experimental Approach

There are generally two steps in our experiment. First, build the decision tree models (ID3 and CART) we will use on our datasets. Then, apply the k×2 cross-validation to tune the hyperparameters.

5.1 ID3 and CART Decision Tree Models

Our experiments will use the ID3 decision tree model for classification and CART decision tree models for regression data.

The ID3 algorithm is as in *Ethem Alpaydin (2014),* **using the gain ratio as splitting criteria***:*

```
GenerateTree(X)
      If NodeEntropy(X)< θi
            Create a leaf labeled by the majority class in X
            Return
      i ← SplitAttribute(X)
      For each branch of X:
            Find Xi falling in branch
            GenerateTree(Xi)

SplitAttribute(X)
      gainRatio ← MIN
      For all attributes i ← 1 to d
            If xi is discrete with n values
                  Split X into X1, …, Xn by xi
                  e ← GainRatio(X1, …, Xn)
                  if e > gainRatio
                        gainRatio ← e; bestf ← i
            Else
                  For all possible split
                        Split X into X1, X2, on xi
                        e ← GainRatio(X1, X2)
                        if e > gainRatio
                              gainRatio ← e; bestf ← i
      Return bestf
```

Assumptions made for the algorithm:

1. The number of splits is based on the number of discrete values.
2. When splitting attributes, we choose the feature that maximizes the gain ratio:

$$gratio(f_i) = \frac{gain(f_i)}{IV(f_i)}$$

Where *gain(fi)* is the information gain:

$$gain(f_i) = \underbrace{\left(-\sum_{\ell=1}^{k} \frac{c_{\pi,\ell}}{|\mathcal{D}_\pi|} \lg \frac{c_{\pi,\ell}}{|\mathcal{D}_\pi|}\right)}_{\mathcal{H}(\mathcal{D}_\pi)} - \underbrace{\left(\sum_{j=1}^{m_i} \frac{|\mathcal{D}_\pi^j|}{|\mathcal{D}_\pi|} \mathcal{H}(\mathcal{D}_\pi^j)\right)}_{E_\pi(f_i)}$$

And *IV(fi)* is the split info:

$$IV(f_i) = -\sum_{j=1}^{m_i} \frac{|\mathcal{D}_\pi^j|}{|\mathcal{D}_\pi|} \lg \frac{|\mathcal{D}_\pi^j|}{|\mathcal{D}_\pi|}$$

With D_π being the subset of data in partition π, $c_{\pi,l}$ denoting the number of examples in partition π that occur in class l, and H, and m_i being the number of values in feature f_i.

The CART algorithm is as in *Greeksforgeeks*:

```
d = 0, endtree = 0
Note(0) = 1, Node(1) = 0, Node(2) = 0
while endtree < 1
    if Node(2^d-1) + Node(2^d) + .... + Node(2^d+1-2) = 2 - 2^d+1
        endtree = 1
    else
        do i = 2^d-1, 2^d, .... , 2^d+1-2
            if Node(i) > -1
                Split tree
            else
                Node(2^i+1) = -1
                Node(2^i-1) = -1
            end if
        end do
    end if
d = d + 1
end while
```

Assumptions made for the algorithm:

1. The split is chosen to minimize the squared error:

$$Err'_\pi = \frac{1}{|\mathcal{D}_\pi|} \sum_j \sum_t (r^t - \hat{r}^t_{\pi j})^2 b_{\pi j}(\mathbf{x}^t)$$

where r^t is the ground truth response value for $\mathbf{x}^t$, $\hat{r}^t_{\pi j}$ is the predicted response value for $\mathbf{x}^t$ in partition π after following branch j, and

$$b_{\pi j}(\mathbf{x}^t) = \begin{cases} 1 & \text{if } \mathbf{x}^t \in \mathcal{D}_\pi, \text{ i.e. } \mathbf{x}^t \text{ reaches node } \pi \text{ and takes branch } j \\ 0 & \text{otherwise.} \end{cases}$$

2. The leaf node of the target value is calculated by the mean of the target values in the corresponding partition.

5.2 K×2 Cross Validation

We will use k×2 cross-validation with k = 5 for hyperparameter tuning on each dataset's KNN models. The 5×2 cross-validation includes the following steps *(Quoting from the Project 1 description, modified according to Project 3 requirements)*:

1. Divide data into two parts: 80% will be used for training, and 20% will be used for tuning, pruning, or other types of "validation." Stratify so that the class distributions are the same in the partitions if the dataset type is classification.
2. Do the following five times:
    a. Hold out 20% of the training set for pruning.
    b. Divide the 80% into two equally sized partitions.
    c. Construct the decision tree model of our choice using a candidate set of parameters. Use the held-out in (a) if it needs to be pruned. Each model will be trained with one of the halves and tested on the held-out 20% in (1).
3. Average the results of these ten experiments for each parameter setting and pick the parameter settings with the best performance.
4. Do the following five times:
    a. Divide the 80% again into two equally sized partitions (stratified).
    b. Hold out 20% of the training set for pruning.
    c. Train a model using the tuned hyperparameters on the first half and test on the second half.
    d. Train a model using the tuned hyperparameters in the second half and test in the first.
5. Average the results of these ten experiments and report the average and the chosen hyperparameter setting.

(*End of Quote*)

Assumptions made for the algorithm:

1. The passed-in data and target are NumPy arrays with the features separated by columns.
2. The mean square error denotes the regression performance.
3. The accuracy denotes the classification performance.
4. To determine the choice of parameters, the function of regression datasets selects the parameters with minimum mean square error; the classification datasets select the parameters with the maximum accuracy.

6 Results

To compare the performance and best hyperparameters chosen for each model, we will use a result table to summarize our results from running the 5×2 cross-validation. We will not record the steps of the 5×2 cross-validation but compare the results directly. The attributes of our result tables are as follows:

1. Dataset Type: whether the dataset is regression or classification, we will combine datasets of the same type for easy comparison.
2. Dataset: name of the dataset.
3. Reduced-error Prune (Best Hyperparameters): whether to reduced-error pruning for the best performance.
4. Performance D.T.: the average performance of the ten folds using the best hyperparameter using the decision tree model. Note we have mean square error (MSE) for regression data and accuracy (A) for classification data.

5. Performance Null: The performance of the datasets using the null model from Project 1.

Our result table presents the following:

Dataset Type	Dataset	# of Rows	Reduced-error prune	Performance D.T.	Performance Null
Regression	Abalone	4177	True	MSE = 1.9547	MSE = 2.3382
	Forest Fire	517	True	MSE = 21.3171	MSE = 16.2358
	Machine	209	False	MSE = 49.6923	MSE = 102.1223
Classification	Breast Cancer	699	False	Accuracy = 0.9219	Accuracy = 0.6559
	Car	1728	False	Accuracy = 0.8467	Accuracy = 0.7004
	House Votes	435	True	Accuracy = 0.9550	Accuracy = 0.6138

Table 1. Result table of decision tree models using 5×2 cross-validation comparing with null models.

7 Discussion of the Behavior

For the regression datasets (abalone, forest fire, machine), reduced-error pruning is applied for abalone and forest fire but not for machine, which adheres with our hypothesis that the reduced-error pruning is more likely to happen to larger datasets. The observed MSE values of abalone and machine are lower for the decision tree models than the null model, but the MSE of the decision tree on forest fire is higher than the null model. This indicates that the decision tree models are not always more effective at predicting continuous numerical values (lower MSE) than the null model. The results partially fail our hypothesis that the decision tree will do better on regression datasets than the null model.

In the classification datasets (breast cancer, car, house votes), reduced-error pruning is employed for house votes but not for breast cancer and car, which fails our hypothesis that the reduced-error pruning is more likely to happen to larger datasets. Indeed, it was selected to use on the datasets with the lowest number of rows. The accuracy of the decision tree models, particularly for house votes, is notably higher than that of the null model. This implies that the decision tree models outperform the null model in correctly classifying instances, indicating their value for classification tasks.

Then, we will look at each dataset separately to see if our observed characteristics impact the performance of each dataset.

7.1 Abalone

We claimed that we might need to prune the abalone data to maximize the performance, which has been proven true. With the pruned abalone tree, we improved the performance (MSE) of the null model by around 0.3835, which is about 16% more improvement than the null model. We could conclude that a decision tree model is more appropriate to abalone data than the null model.

7.2 Forest Fires

For the forest fires data, we presumed that it may be hard for the prediction to be accurate considering its massive range and extremely skewed nature in the target variable. Pruning is helping to cut off unnecessary branches, but even the improved performance does not exceed the null model's MSE. For future investigations, we might want to consider transforming the forest fires

data, but for now, we could conclude that the decision tree is not the best model for the forest fires data.

7.3 Machine

We were worried that the complicating feature structure would harm the performance, but the result of applying the decision tree to machine is rather satisfactory. The MSE = 102.1223 of the null model is improved by 52.43, more than 50% improvement for the MSE. We are confident that the decision tree is a better model for machine than the null model.

7.4 Breast Cancer

The use of the feature values directly was our concern. Even if they were discrete values, they appear to be numeric. However, the decision tree model, with an accuracy of 92.19%, significantly outperformed the null model, which had an accuracy of 65.59%. With over 40% improvement over the null model, this highlights the added value of using a decision tree model for this classification task.

7.5 Car

Car has the most significant number of rows in the classification datasets, but it did not perform better with the pruned tree. Thus, the "more data, more likely to prune" theory we made initially is not always actual. Indeed, we considered that well-constructed features could contribute to avoiding prune, and the decision tree accuracy is better than the null model. However, it is not as good as the other classification datasets, with only 20% improvements, but the decision tree is still a better option for the car data than the null model.

7.6 House Votes

The house votes dataset shows the most substantial improvement in accuracy among the classification datasets, with about a 55% improvement rate than the null model. We anticipated that prune should be preferred as many features could result in a vast tree. Our result proves that the decision tree for house votes can yield a better performance through reduced-error pruning. Hence, we could conclude that the decision tree is one of the best options for the house votes dataset.

8 Conclusions

In conclusion, this paper has presented a comprehensive exploration of decision tree models in the domains of regression and classification. Through experimentation and analysis, we have gained valuable insights into the behavior and adaptability of decision tree models across a diverse range of datasets.

The results of our experiments revealed that the effectiveness of decision tree models in regression tasks depends on the dataset's characteristics. For abalone, pruning improved performance, demonstrating the decision tree's suitability. Forest fires posed challenges, possibly due to their skewed nature, and the decision tree did not outperform the null model. Surprisingly, the complex feature structure of the machine dataset was effectively handled by the decision tree, making it a preferred model. Breast cancer showcased the decision tree's exceptional performance in adapting discrete numerical features, significantly improving accuracy compared to the null model. Despite its large size, the car dataset did not require pruning, and the decision tree offered optimized accuracy. House votes benefited greatly from reduced-error pruning, emphasizing the importance of pruning in broad decision trees.

The overall findings of the paper highlight the flexibility and adaptability of decision tree models in handling both regression and classification tasks. However, the choice to apply the model or reduced-error pruning varies depending on dataset-specific characteristics, and its impact on

model performance can be significant if on the suitable dataset. The paper's results reinforce that the decision tree model can be a valuable tool for accurate prediction, especially in classification.

In practice, selecting the suitable model for a given dataset remains a data-driven and nuanced decision. The paper underscores the importance of considering the dataset's unique properties and the potential benefits of reduced-error pruning. These findings can inform future applications of decision tree models in real-world scenarios, where accurate predictions and classifications are crucial.

Ultimately, this paper contributes to the ongoing exploration of machine learning models in this class and their adaptability to a wide range of datasets, furthering our understanding of the intricate relationship between decision tree models and data characteristics.

Reference

Ethem Alpaydin. *Introduction to Machine Learning*. MIT Press, Boston, Massachusetts, 2014

Greeksforgreeks. *CART (Classification And Regression Tree) in Machine Learning.* Available online at Greeksforgreeks: https://www.geeksforgeeks.org/cart-classification-and-regression-tree-in-machine-learning/#