Performance comparison of two Decision tree algorithms based on splitting criteria for predicting child birth delivery type

S. Aliyu
Department of Computer Science
Ahmadu Bello University, Zaria.
aliyusalisu@abu.edu.ng

H. Musa
Institute of Computing and ICT
Ahmadu Bello University, Zaria.
musah@abu.edu.ng

F. Jauro
Department of Computer Science
Ahmadu Bello University, Zaria.
fatijauro@gmail.com

Abstract—*Caesarean sections, vaginal deliveries, obstetric forceps and vacuum extractions are common techniques used to perform child delivery in Maternity care. Predicting the type of delivery in advance envisages safety and high quality service. Decision tree is a data mining model for predicting by extracting hidden knowledge from large dataset. This paper aims at comparing the strength of two decision tree algorithms based on splitting criteria for predicting child birth delivery type. Data was collected from the obstetric and Gynea Department of the Ahmadu Bello University Teaching Hospital, Zaria-Nigeria. A total of 1673 distinct records with 14 variables representing patient specific information and the outcome of their childbirth delivery type was collected. A 70% to 30% train/test split model evaluation was used with five metrics measured to compare both algorithms which was implemented and tested on an ipython notebook. For the algorithm that uses the information Gain an accuracy of 68%, precision of 60%, sensitivity of 68%, classification error of 31% and F-Measure of 60% was recorded. However, the algorithm that uses the Gini Index criteria performs slightly better with an accuracy of 69%, precision of 63%, sensitivity of 69%, classification error of 30% and F-Measure of 62%.*

Keywords—Algorithm; Data mining; Decision Tree; Delivery Type

I. INTRODUCTION

It was estimated that about 830 women die from pregnancy or childbirth related complications around the world every day [1]. These complications largely determines the delivery type that should be employed. There are several types of child birth delivery methods some of which include, the spontaneous vaginal delivery (SVD), caesarean delivery (CS), forceps and vacuum extraction. vaginal delivery is the most common and safest type of child birth. The use of forceps which is an instrument resembling a large spoon may be used to cup the baby's head and help guide the baby through the birth canal. Vacuum delivery is another way similar to forceps delivery. In vacuum delivery, a plastic cup is applied to the baby's head by suction and the health care provider gently pulls the baby from the birth canal. Caesarean delivery is child delivery by surgery on the uterus.

The decision as to whether or not a particular birth requires assistance and the choice and timing of any intervention is still a major problem. Predicting the type of delivery in advance envisages safety and high quality service.

Data Mining is the process of selecting, exploring and modelling of large amount of data in order to discover unknown patterns or relationships which provides a clear and useful result to the data analyst [2]. Data mining has been applied with success to different fields of human endeavours (e.g. healthcare) for building either predictive or descriptive models. Predictive data mining modelling in healthcare uses patient specific information to predict the outcome of interest thereby supporting decision making. In the past decade, several predictive data mining algorithms have evolved such as Support Vector Machine (SVM), K-Nearest Neighbor (KNN), Naive Bayes and Decision Trees that seeks to foretell some response of interest. In recent years, several studies can be seen on the use of predictive data mining in maternal healthcare such as [3], [4], [5] and [6].

Decision Tree is a predictive data mining model or classifier expressed as a recursive partition of the instance space. The decision tree consists of nodes that form a rooted tree, meaning it is a directed tree with a node called a "root" that has no incoming edges. All other nodes have exactly one incoming edge. A node with outgoing edges is referred to as an "internal" or "test" node. All other nodes are called “leaves” (also known as "terminal" or "decision” nodes). In the decision tree, each internal node splits the instance space into two or more sub-spaces according to a certain discrete function of the input attribute values. Each leaf is assigned to one class representing the most appropriate target value [7].

The selection of the most appropriate attribute is made according to some splitting measures. The objective of the splitting algorithm is to find a variable threshold pair that maximises the homogeneity of the resulting two or more subgroups. The most commonly used mathematical splitting algorithm includes the Entropy based Information gain and the Gini Index used in the ID3 and CART decision tree algorithms respectively [7].

Decision tree algorithms automatically construct a decision tree from a given dataset. The goal is to find the optimal decision tree by minimizing the generalization error, the number of nodes or the average depth of the tree. Figure 1 is a top down decision tree generation algorithm used by both ID3 and CART with their major difference in the splitting criteria.

```
GenerateTree(S, A, y, SplitCriterion, StoppingCriterion)
    S     //Training Set
    A     //Input Feature Set
    y     // Target Feature
    SplitCriteria ($a_i$, S)    // The method for evaluating a certain split
    StoppingCriteria(S)   //The criteria to stop the growing process
    Create a new tree T with a single root node.
    IF StoppingCriteria(S) THEN
        Mark T as a leaf with the most common value of y
        in S as a label.
    ELSE
        $\forall a_i \in A$ find a that obtain the best SplitCriteria
        label t with a
        FOR each outcome $v_i$ of a:
            set $subtree_i$ = GenerateTree( $\delta_{a=vi}$  S, A, y)
            connect the root node of $t_T$ to $subtree_i$ with an
            edge that is labeled as $v_i$
        END FOR
    END IF
RETURN T
```

Fig. 1. Decision Tree Generation Algorithm adopted from [7]

Although several studies has shown the use of decision tree algorithm in health care decision support [3], [8], [5] and [9] , however, a study comparing the prediction performance of the decision tree splitting algorithms has not been commonly encountered in the literature. For this reason, a comparison of these two splitting approach forming two classes of decision tree algorithms is performed in this paper to estimate the prediction performance on child birth delivery type.

II. METHODOLOGY

A. *Data Source, Data Understanding and Preparation*

In order to perform the research reported in this paper, we used the data collected from the obstetric and Gynea Department of the Ahmadu Bello University Teaching Hospital, Zaria-Nigeria. A total of 1673 distinct records with 14 variables representing patient specific information and the outcome of their childbirth delivery type was used.

Each data instance consists of a set of variables: age, fetus presentation (breach, cephalic, face to pubis, footing breach or traverse), twin, sex, fetus weight and Estimated Blood Loss (EBL).

Statistical measures related to the numerical variables are represented in table 1.

Table 1. Statistical measure of the numerical variables used in the dataset.

Variables	**Minimum**	**Maximum**	**Average**	**Standard Deviation**
Age	15	53	28.19	7.057
Fetus Weight(Kg)	0.4	6.25	2.9815	0.677
EBL(mls)	1.1	300	279.60	258.52

The target variables represents the five delivery types: SVD, CS, forceps, vacuum and breach assisted. Figure 2 shows the data distribution.

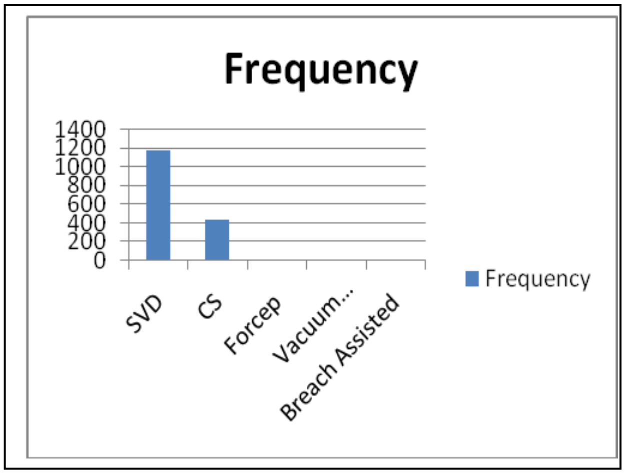

Fig. 2. Distribution of the target variable delivery type

In preparing the data, null or noise values were deleted. Some normalization was also required due to some inconsistencies in data capturing.

B. Decision Tree Algorithm Splitting Criteria

Decision tree is a popular data mining technique due to its intuitive representation, relatively fast to construct when compared to building other models and above all its high level of accuracy. Many variants of decision tree algorithms exist with the splitting criterion being the major differences between them. The split criterion is used to check which attribute to test at each node of the tree that is the attribute that is most useful for the classification of the dataset.

Information gain/Entropy and Gini Split/Index are two types of split criterion that is used in the ID3 and CART decision tree algorithm. The splitting operation is done until all samples for a given node belong to the same class, there are no remaining attributes for further partitioning and there are no samples left. This approach describes a measure of the degree of impurity while splitting the dataset.

1) Information Gain/Entropy

Given a data set S that has attributes A, B, C, ... and target classes x, y, z, ... We say a data set is pure if it contains only a single target class. If the data set contains several target classes then it is said to be impure. When data sets are split to grow decision trees, it is done to reduce impurity.

Entropy is one way to measure the degree of impurity of a split. It is computed in terms of the probability P_j of each class j in the dataset as given in equation 1.

$$Entropy = \sum_j P_j \log_2 P_j \qquad (1)$$

Entropy can be interpreted as the number of bits that allows us to overcome uncertainty about whether an item belongs to one class or another. If a set is completely pure you don't need any bit (0 bit) to determine whether an element picked from that set belongs to a class or not because all elements of the set are from one class. But suppose the set is made up of two equal number of classes, if an element is picked at random you will have no idea on which class it belongs, as a result you need a full bit (1 bit) to determine that. For different proportions of classes, you get decimals between 0 and 1bit as your entropy.

So, given the dataset S and subsets S_i as a result of splitting on attribute A, B and C, the difference of the degrees of impurity between S and S_i is called the Information gain. This measure is use to know what our gain is if we split the data set on either attribute A, B or C. The Information gain on splitting on attribute A is computed as the difference between the degree of impurity of the parent dataset S and the weighted summation of impurity degrees of the subset dataset S_i split on attribute A with values *v* mathematically given as:

$$Information\ Gain(S, A) = H(S) - \sum_v \frac{|S_v|}{|S|} H(S_v)$$

(2)

Where H(S) is the entropy of the parent dataset or node S, $H(S_v)$ is the entropy of the subset split base on values v of attribute A and |S| is the total number of entries in dataset S.

After computing the information gain for splitting base on each attribute, then the optimum attribute which is the attribute that produces the maximum information gain is selected. The dataset will then be splitted base on this optimum attribute. The process is repeated for each sub dataset until dataset with pure classes are assigned into the leaf nodes of the decision tree.

2) Gini Split/Index

Gini Index is another way of measuring the degree of impurity in a dataset. Given a training dataset S, the target attribute takes on j different values then the Gini index of S is defined as:

$$Gini(S) = 1 - \sum_{i=1}^{j} (P_i)^2 \qquad (3)$$

Where P_i is the probability of S belonging to class i . If a dataset has only one class, its gini index is 0 which signifies a pure dataset.

So Gini split, an impurity based splitting algorithm measures the divergences between the probability distribution of the target attributes values. This is achieve by selecting the attribute with the maximum gain. The gain by a Gini Split on dataset S and attribute A is given as:

$$Gini\ Split(S, A) = Gini(S) - \sum_{i=1}^{j} \frac{|S_i|}{|S|} Gini(S_i)$$

(4)

Where is the partition of S induced by the values of attribute A. For each partition base on the different attribute the gain is computed and the partition with the maximum gain is chosen.

B. Implementation

In this study, the algorithms was implemented on an ipython [10], jupyter notebook and pandas[11] library to import the dataset. A maximum height of three was used as the stopping criteria. Expository analysis were then performed and several estimators were obtained from scikit-learn[12] library.

C. Evaluation

In this study, we used the train/test split model evaluation procedure to estimate how well a model will generalize to out-of-sample data and five evaluation metrics to quantify the model's performance.

The experiment was performed on a 70% training set to 30% test data split. This was chosen as the train/test split for it is common in the literature. The train/test approach was also considered because of its simplicity, speed and flexibility.

For the evaluation metrics used to evaluate the algorithms, the following metrics where used:

- Accuracy: which measures the percentage of correct predictions.
- Classification Error: measures how often the classifier is incorrect. It is mathematically defined as:

$$\frac{(FP+FN)}{float(TP+TN+FP+FN)} \tag{5}$$

Where:

FP = False Positive (Incorrectly predicted positive)

FN=False Negative (Incorrectly predicted negative)

TN=True Negative (correctly predicted negative)

TP=True Positive (correctly predicted positive)

- Sensitivity (Recall): measures how often is the prediction correct.

$$\frac{TP}{float(FN+TP)} \tag{6}$$

- Precision: measures how often the prediction is correct when a positive value is predicted.

$$\frac{TP}{float(TP+FP)} \tag{7}$$

- F-Measure: is interpreted as a weighted average of the precision and recall.

$$2*\frac{(\Pr ecision*\operatorname{Re} call)}{(\Pr ecision+\operatorname{Re} call)} \tag{8}$$

III. RESULTS

In this study, the algorithms were evaluated based on the measures described above. For the algorithm that uses the information Gain an accuracy of 68%, precision of 60%, sensitivity of 68%, classification error of 31% and F-Measure of 60% was recorded. However, the algorithm that uses the Gini Index criteria performs slightly better with an accuracy of 69%, precision of 63%, sensitivity of 69%, classification error of 30% and F-Measure of 62%. Table 2 summarizes the results obtained.

Table 2: Algorithm Evaluation Metric

Evaluation Metrics	**Information Gain**	**Gini Index**
Accuracy	68%	69%
Precision	60%	63%
Sensitivity	68%	69%
Classification Error	31%	30%
F-Measure	60%	62%

Fig. 3 and Fig. 4 shows the decision trees generated by the two algorithms.

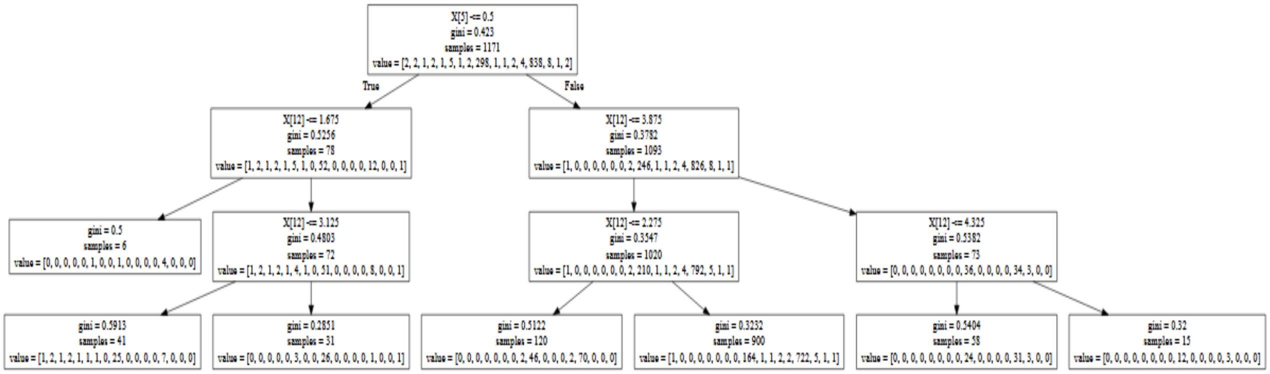
X[5] <= 0.5
gini = 0.423
samples = 1171
value = [2, 2, 1, 2, 1, 5, 1, 2, 298, 1, 1, 2, 4, 838, 8, 1, 2]
True
False
X[12] <= 1.675
gini = 0.5256
samples = 78
value = [1, 2, 1, 2, 1, 5, 1, 0, 52, 0, 0, 0, 0, 12, 0, 0, 1]
X[12] <= 3.875
gini = 0.3782
samples = 1093
value = [1, 0, 0, 0, 0, 0, 0, 2, 246, 1, 1, 2, 4, 826, 8, 1, 1]
gini = 0.5
samples = 6
value = [0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 4, 0, 0, 0]
X[12] <= 3.125
gini = 0.4803
samples = 72
value = [1, 2, 1, 2, 1, 4, 1, 0, 51, 0, 0, 0, 0, 8, 0, 0, 1]
X[12] <= 2.275
gini = 0.3547
samples = 1020
value = [1, 0, 0, 0, 0, 0, 0, 2, 210, 1, 1, 2, 4, 792, 5, 1, 1]
X[12] <= 4.325
gini = 0.5382
samples = 73
value = [0, 0, 0, 0, 0, 0, 0, 0, 36, 0, 0, 0, 0, 34, 3, 0, 0]
gini = 0.5913
samples = 41
value = [1, 2, 1, 2, 1, 1, 1, 0, 25, 0, 0, 0, 0, 7, 0, 0, 0]
gini = 0.2851
samples = 31
value = [0, 0, 0, 0, 0, 3, 0, 0, 26, 0, 0, 0, 0, 1, 0, 0, 1]
gini = 0.5122
samples = 120
value = [0, 0, 0, 0, 0, 0, 0, 2, 46, 0, 0, 0, 2, 70, 0, 0, 0]
gini = 0.3232
samples = 900
value = [1, 0, 0, 0, 0, 0, 0, 0, 164, 1, 1, 2, 2, 722, 5, 1, 1]
gini = 0.5404
samples = 58
value = [0, 0, 0, 0, 0, 0, 0, 0, 24, 0, 0, 0, 0, 31, 3, 0, 0]
gini = 0.32
samples = 15
value = [0, 0, 0, 0, 0, 0, 0, 0, 12, 0, 0, 0, 0, 3, 0, 0, 0]

Fig. 3. Decision Tree Generated by the Gini Split/Index

IV. DISCUSSION

As shown in the results above , advanced data mining methods can be used to develop models that possesses a high degree of predictive accuracy. However there are several issues related to these algorithms.

Information Gain measure tends to prefer attributes with large number of possible values while Gini Index tends to isolate the largest class from all other classes.

These predictive models can be valuable in diagnosis, developing successful treatment or avoidance of ailments.

V. CONCLUSION AND FUTURE WORK

This paper aimed at studying the performance of two decision tree algorithms based on different splitting criteria on the prediction of childbirth delivery type. Acceptable results were obtained with slight distinction between the two algorithms. Future work will be targeted towards improving the algorithms in other to achieve better performance of the prediction model evaluation metrics.

REFERENCES

[1] Shakibazadeh, E., et al., *Respectful care during childbirth in health facilities globally: a qualitative evidence synthesis.* BJOG: An International Journal of Obstetrics & Gynaecology, 2017.

[2] Giudici, P., *Applied data mining: Statistical methods for business and industry*. 2005: John Wiley & Sons.

[3] Morais, A., et al., *Predicting the need of Neonatal Resuscitation using Data Mining.* Procedia Computer Science, 2017. **113**: p. 571-576.

[4] Fonseca, F., et al., *Step Towards Prediction of Perineal Tear.* Procedia Computer Science, 2017. **113**: p. 565-570.

[5] Delen, D., G. Walker, and A. Kadam, *Predicting breast cancer survivability: a comparison of three data mining methods.* Artificial intelligence in medicine, 2005. **34**(2): p. 113-127.

[6] Tayefi, M., et al., *hs-CRP is strongly associated with coronary heart disease (CHD): A data mining approach using decision tree algorithm.* Computer methods and programs in biomedicine, 2017. **141**: p. 105-109.

[7] Rokach, L. and O. Maimon, *Data Mining with Decision Trees: Theory and Applications*. 2014: World Scientific.

[8] Pereira, S., et al., *Predicting type of delivery by identification of obstetric risk factors through data mining.* Procedia Computer Science, 2015. **64**: p. 601-609.

[9] Mehta, R., N. Bhatt, and A. Ganatra, *A survey on data mining technologies for decision support system of maternal care domain.* International Journal of Computers and Applications, 2016. **138**(10): p. 20-4.

[10] Pérez, F. and B.E. Granger, *IPython: a system for interactive scientific computing.* Computing in Science & Engineering, 2007. **9**(3).

[11] Anthony, F., *Mastering pandas*. 2015: Packt Publishing Ltd.

[12] *Scikit Learn. Documentation of scikit-learn*. 2014.