Evaluating the Impact of GINI Index and Information Gain on Classification using Decision Tree Classifier Algorithm*

Suryakanthi Tangirala
Faculty of Business, University of Botswana
Gaborone, Botswana

***Abstract*—Decision tree is a supervised machine learning algorithm suitable for solving classification and regression problems. Decision trees are recursively built by applying split conditions at each node that divides the training records into subsets with output variable of same class. The process starts from the root node of the decision tree and progresses by applying split conditions at each non-leaf node resulting into homogenous subsets. However, achieving pure homogenous subsets is not possible. Therefore, the goal at each node is to identify an attribute and a split condition on that attribute that minimizes the mixing of class labels, thus resulting into nearly pure subsets. Several splitting indices were proposed to evaluate the goodness of the split, common ones being GINI index and Information gain. The aim of this study is to conduct an empirical comparison of GINI index and information gain. Classification models are built using decision tree classifier algorithm by applying GINI index and Information gain individually. The classification accuracy of the models is estimated using different metrics such as Confusion matrix, Overall accuracy, Per-class accuracy, Recall and Precision. The results of the study show that, regardless of whether the dataset is balanced or imbalanced, the classification models built by applying the two different splitting indices GINI index and information gain give same accuracy. In other words, choice of splitting indices has no impact on performance of the decision tree classifier algorithm.**

Keywords*—Supervised learning; classification; decision tree; information gain; GINI index

I. Introduction

Machine learning problems can be broadly classified into two categories viz. supervised learning and unsupervised learning as shown in Fig. 1. With supervised learning techniques, the training data is labeled. It means each observation in the data set has both descriptive variables (i.e., independent variables or decision variables) and a labeled outcome variable. Labels can be either categories or continuous values [1]. With supervised learning, a labeled data set is used to train the model in making predictions. A learning model maps the input variables to the output variable, with the aim of accurately predicting the output for future input variables.

Unlike supervised learning, with unsupervised learning the data is not labeled. This means that the training data has descriptive variables only and no outcome variable. The model has to determine the patterns and interesting structures in the data that are not known beforehand [2].

Classification is a supervised learning problem, where the objective is to analyse the training data and develop a model that can predict the future behavior, here the training dataset is labeled. Decision tree algorithm is commonly used for classification tasks. Decision trees classify data into finite number of classes based on the values of input variables. It is most appropriate for categorical data [3].

Decision tree is a simple flowchart that selects class labels of an output variable using the values of one or more input variables. The classification process starts at the root node of the decision tree and recursively progresses until it reaches the leaf node with class labels. At each node a split condition is applied to decide whether the input value should continue towards left or right sub tree until it reaches the leaf nodes [4]. The split condition applied at each node should result in homogenous subsets. Homogenous subsets have records with same class label. However, it is impossible to achieve pure homogenous subsets with real time data. Some kind of mixing will always be there. Therefore, while building the decision tree, the goal at each node is to select split conditions that best divide the dataset into homogenous subsets. The "goodness of split criterion" was introduced, which is derived from the notion of impurity [5]. Impurity is measured mathematically for each split condition and split condition with lowest impurity value is chosen.

To measure the impurity value of a split condition several indices are proposed viz., GINI index, Information gain, gain ratio and misclassification rate. This paper empirically examines the effect of GINI index and Information gain on classification task. The classification accuracy is measured to check the suitability of the models in making good predictions.

Rest of the paper is organised as follows: Section II introduces the theoretical notions of Information gain and GINI index. Section III is literature review. Sections IV and V gives the details of data and experimental procedure to compare Information gain and GINI index on balanced and imbalanced data set along with results obtained, and Section VI summarizes the results of the study.

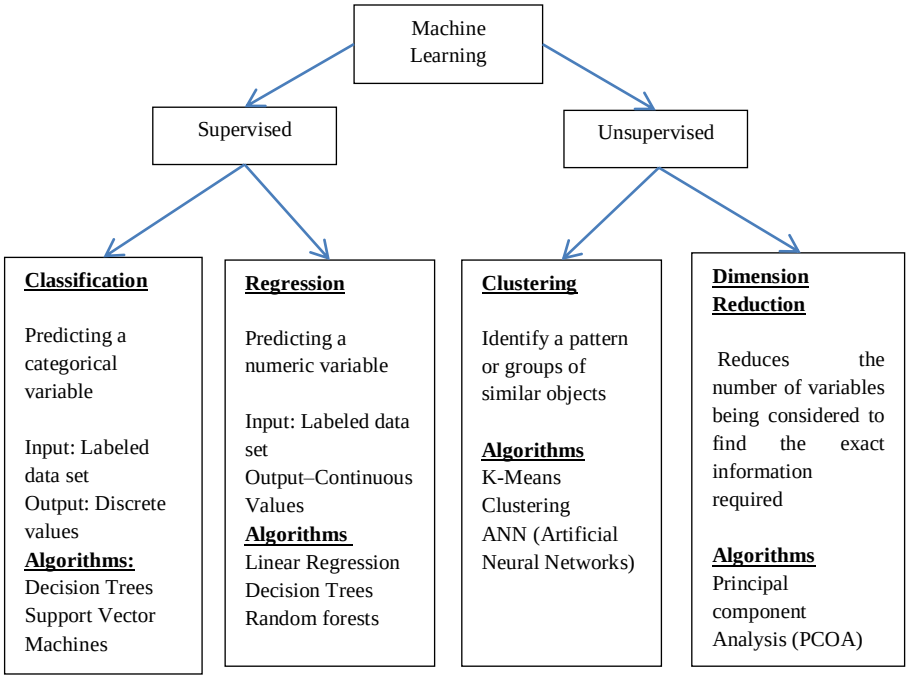

Fig. 1. Broad Classification of Machine Learning Techniques.

II. Theoritical Notation

This section briefly discusses theoretical notions of Information gain and GINI index. Raileanu and Stoffel [6] presented theoretical comparison of GINI index and Information gain.

Let L be a learning sample, L= $\{(x_1, c_1), (x_2, c_2) \dots (x_i, c_j)\}$; Where $x_1, x_2 \dots x_i$ is a measurement vector and $c_1, c_2 \dots c_j$ are class labels. x_i can be viewed as a vector of input variables, and split conditions are based on one of these variables. If p_i is probability that an arbitrary tuple belongs to class c_i, p_i can be measured as

$$p_i = \frac{C_i}{L}$$

A. Entropy

Information gain is based on Entropy. Entropy measures the extent of impurity or randomness in a dataset [7]. If the observations of subsets of a dataset are homogenous, then there is no impurity or randomness in the dataset. If all the observations of subsets belong to one class, the entropy of that dataset would be 0. Entropy is defined as the sum of the probability of each label times the log probability of that same label.

$$Entropy(L) = \langle C_1|L\rangle \log_2\langle C_1|L\rangle + \langle C_2|L\rangle \log_2\langle C_2|L\rangle + \dots + \langle C_j|L\rangle \log_2\langle C_j|L\rangle$$

$$Entropy(L) = p_1 \log_2 p_1 + p_2 \log_2 p_2 + \dots + p_j \log_2 p_j$$

$$Entropy(L) = -\sum_{i=1}^{j} p_i \log_2(p_i)$$

For a dataset with one class label, p_i will be 1 and $\log_2(p_i)$ is 0. Hence the Entropy of homogenous data set is zero [8]. If the entropy is higher the uncertainty/impurity/mixing is higher [9].

B. Information Gain

Information gain is based on Entropy. Information gain is the difference between Entropy of a class and conditional entropy of the class and the selected feature. It measures the usefulness of a feature f in classification [10] i.e., the difference in Entropy from before to after the split of set L on a feature f. In other words, it measures the reduction of uncertainty after splitting the set on a feature. If information gain value increases, it means the feature f is more useful for classification. The feature with highest information gain is the best feature to be selected for split. Assuming that there are V different values for a feature f, $|L^v|$ represents the subset of L with f=v, Information gain after splitting L on a feature f is measured as [8].

$$IG\ (L, f) = Entropy\ (L)\ - \sum_{v=1}^{V} \frac{|L^V|}{|L|}\ (Entropy\ (L^V))$$

C. GINI Index

GINI index determines the purity of a specific class after splitting along a particular attribute. The best split increases the purity of the sets resulting from the split. If L is a dataset with j different class labels, GINI is defined [3] as

$$\mathrm{GINI(L)} = 1 - \sum_{i=1}^{j} p_i^2$$

Where pi is relative frequency if class i in L. If the dataset is split on attribute A into two subsets L1 and L2 with sizes N1 and N2 respectively, GINI is calculated as

$$\mathrm{GINI_A(L)} = \frac{\mathrm{N_1}}{\mathrm{N}}\mathrm{GINI(L_1)} + \frac{\mathrm{N_2}}{\mathrm{N}}\mathrm{GINI(L_2)}$$

Reduction in impurity is calculated as

$$\Delta\mathrm{GINI(A)} = \mathrm{GINI(L)} - \mathrm{GINI_A(L)}$$

III. Literature Review

This section briefly presents some of the empirical studies that compared the performance of decision tree algorithms which use different impurity metrics for feature selection at non-leaf nodes. An attempt is made to find out if the choice of these feature selection metrics has any impact on the accuracy of the model from past studies.

Mingers [11] tested different feature selection measures empirically, and reported that choice of the feature selection measure affects the size of the tree but not its accuracy. The accuracy remained the same even when attributes are randomly selected. Patil [12] studied the two decision tree based classification algorithms C5.0 and CART. C5.0 uses information gain and CART algorithm uses GINI index to select the features for split conditions. Their study was an experiment to compare C5.0 and CART classification algorithms to classify if a customer qualifies for membership card or not. The study revealed that C5.0 gives higher classification accuracy of 99.6% than CART algorithm with 94.8% accuracy.

A study empirically compared different feature selection measures and proposed a variant of GINI index which uses GINI index ratios for feature selection. In this study they compared the classification accuracy of modified GINI with other classification algorithms ID3, C4.5 and GINI. The results show that ID3 and C4.5 based on Information gain have low classification and prediction accuracy than GINI index and modified GINI index. Modified GINI index is reported to obtain the highest accuracy among all algorithms that were compared [13]. Adhatrao et.al [14] present experiments to compare the performance of two decision tree algorithms, ID3 and C4.5 in predicting the performance of first year engineering students based on the performance achieved by old students who are now in second year engineering. The results show that both the algorithms give same accuracy. In a study Hssina, et.al [15] compared different decision tree algorithms viz. ID3, C4.5, C5, CART and the results reported show that C4.5 has achieved the highest classification accuracy. C4.5 uses information gain to evaluate goodness of split.

Above discussed studies give varied results on the performance of Information gain and GINI index. Moreover, the empirical studies compared the models that were built using different tree based algorithms. These algorithms differ in splitting attribute selection, number of splits (binary /ternary), order of splitting attribute (splitting the same attribute only once or multiple times), stopping criteria and pruning technique (pre/post) [14]. All these factors contribute to the performance of the models built using these algorithms.

The present study is unique as it focuses only on finding the impact of GINI index and Information gain on classification. Therefore, unlike other studies, this study develops classification models using single algorithm called decision tree classifier on which GINI index and information gain are applied individually. This neutralizes the impact of all other factors on models.

IV. Experimental Setup

This section gives the details of data and experimental procedure.

A. Dataset Description

The experiment is conducted using real data provided by UCI Machine Learning repository [16]. The data was collected by Portuguese banking institution by making phone calls to customers. The dataset is relatively a large dataset with 41187 rows and 21 columns. One input variable, 'duration' is discarded, as it is highly multi valued and should be avoided for good prediction. Details of the remaining variables are given in Table I. The classification goal is to predict whether customer will subscribe for a term deposit (y) based on remaining 19 input variables. The dataset is clean; it doesn't have Null values. Term deposit (y) is the outcome variable with two class labels (yes or no). Therefore, it is a binary classification problem.

TABLE. I. Description of the Dataset

Variable	***Description***	***Type***
age	Age of the customer	numeric
job	Type of job of customer	categorical
marital	marital status	categorical
education	Educational qualification	categorical
default	Has credit in default	categorical
housing	Has housing loan	categorical
loan	Has personal loan	categorical
contact	Contact communication type (cell, telephone)	categorical
month	Last contact month of the year	categorical
day_of_ week	Last contact day of the week	categorical
campaign	number of contacts performed during this campaign and for this client	numeric
pdays	number of days that passed by after the client was last contacted from a previous campaign	numeric
previous	number of contacts performed before this campaign and for this client	numeric
poutcome	outcome of the previous marketing campaign	categorical
emp.var.rate	employment variation rate - quarterly indicator	numeric
cons.price.index	consumer price index - monthly indicator	numeric
cons.conf.index	consumer confidence index - monthly indicator	numeric
euribor3m	euribor 3 month rate - daily indicator	numeric
nr.employed	number of employees - quarterly indicator	numeric
y (outcome variable)	has the client subscribed a term deposit? (binary: 'yes','no') (Yes=1, No=0)	categorical

When developing a decision tree, the goal at each node is to identify the attribute and a split condition of the attribute that best divides the training set into pure subsets at that node [17].

Given a dataset with input variables and an outcome variable with a class label, the decision tree algorithm recursively divides the training set until each division contains examples of same class label. If all the observations of the division belong to one class, then it is homogenous subset and if they belong to multiple classes it is impure or heterogeneous [18]. To evaluate the goodness of the split, two splitting indices, GINI index and Information gain are used. Both GINI index and Information gain are applied on Decision tree classifier algorithm and models are developed.

The dataset is split into two parts, training and test. The general practice is to divide the dataset into 80:20 ratios, 80 % training data and 20% test data (unseen data). Using the decision tree classifier algorithm, a classification model built recursively from the training data, dividing the data until each division is pure (homogenous class) and then its prediction accuracy is tested on the unseen test data. In this experiment, the classification model is trained to predict whether customers would subscribe for a term deposit (Yes or No) using the 19 input variables.

A k-fold cross validation method minimizes the bias associated with random sampling of the training and hold out of data samples while comparing the predictive accuracy of two or more methods [3]. In our experiment classification model is trained and tested 10 times where the training set is split into 10 exclusive subsets of equal size and each time, the model is trained on all 9 leaving 1 subset which will be used for testing. Overall accuracy is simply average of the 10 individual accuracies obtained.

B. Decision Tree Classifier

Many algorithms have been proposed for creating decision trees. In this experiment, Decision tree classifier, a supervised learning algorithm is used. It is based on CART and can be used for creating both classification and regression trees [19]. *rpart* is a package in *R* programming, which implements many of the ideas found in CART model. Different splitting criterions can be applied while splitting the nodes of the tree using rpart function [20]. The classification models built by applying Information gain and GINI index are shown in Fig. 2 and Fig. 3, respectively.

It is noted that both the splitting measures select the same feature, 'Number of employees' with same split condition at the root node. 'Number of employees' which is a numeric attribute is selected with split condition nr.employees >=5088.

C. Performance Evaluation Metrics

Classification is technique where the model is developed using a labeled dataset. It means each record in the training dataset has a class label associated with it. The model is later used to predict the class labels of new/unseen data. Predictive accuracy of classification model is its ability to correctly predict the class label of an unseen data. The common metrics for measuring the accuracy of classification models are confusion matrix, overall accuracy, per-class accuracy, recall and precision [3] [21]. First confusion matrix is created using which all other metrics are easily calculated.

- Confusion matrix

Confusion matrix gives detailed view of the performance with breakdown of correct and incorrect predictions for each class. The performance is measured by comparing the predicted outcome values with actual values. The information is tabulated in the form of a confusion matrix as shown in Table II.

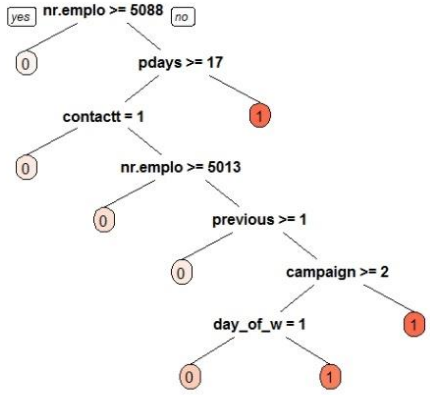

Fig. 2. Decision Tree Visualization using Information Gain.

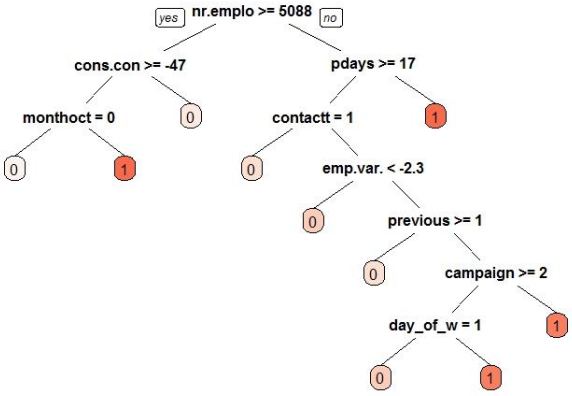

Fig. 3. Decision Tree Visualization using GINI Index.

TABLE. II. CONFUSION MATRIX

Actual			
Predicted class		Positive	Negative
	Positive	True positive count (TP)	False Positive count (FP)
	Negative	False Negative count (FN)	True Negative count (TN)

where True positives (TP) corresponds to the number of positive examples correctly predicted by the model, False negatives(FN) represents number of positive examples wrongly predicted as negative, False positive(FP) refers to number of negative examples wrongly predicted as positive and True negative (TN) is number of negative examples correctly predicted [22]

- Overall Accuracy

Overall classifier accuracy is the rate at which the model makes accurate predictions. It is the ratio of number of correct predictions to total number of predictions made.

$$\text{Accuracy} = \frac{\text{Number of correct predictions}}{\text{Total number of predictions}}$$

$$= \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}$$

- Per-class accuracy

Per class accuracy gives the average of accuracy of prediction of each class. It is particularly useful when the data sets are imbalanced. Overall accuracy is micro average and per-class accuracy is macro average.

$$\text{Per class accuracy} = \frac{\text{Number of correct predictions of that class}}{\text{Total count of predictions of that class}}$$

$$\text{Majority (positive) class accuracy} = \frac{\text{TP}}{(\text{Sensitivity})\text{TP} + \text{FN}}$$

$$\text{Majority (Negative) class accuracy} = \frac{\text{TN}}{(\text{Sensitivity})\text{TN} + \text{FP}}$$

- Precision is defined as the ratio of correctly classified majority class values (True positives) divided by sum of correctly classified majority class values (True positives) and incorrectly classified majority class values (False positive). It should be high.

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$

- Recall is defined as the ratio of correctly classified majority class values (True positives) divided by sum of correctly classified majority class values (True positives) and incorrectly classified minority class values (False Negatives). Recall estimates the classifiers accuracy in predicting the majority class. It should be high.

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

D. Performance Evaluation on the Test Set

The test set has a total of 8237 observations. Confusion matrix of Decision tree classifier with Information gain and GINI Index are shown in Table III and Table IV. Positive/majority class is represented as 0 negative/minority class is represented using 1.

TABLE. III. CONFUSION MATRIX OF CLASSIFICATION RESULTS OBTAINED BY DECISION TREE CLASSIFIER WITH INFORMATION GAIN

	Actual			Total
Predicted		0	1	
	0	7198	718	7916
	1	119	202	321
Total		7317	920	8237

TABLE. IV. CONFUSION MATRIX OF CLASSIFICATION RESULTS OBTAINED BY DECISION TREE CLASSIFIER WITH GINI INDEX

	Actual			Total
Predicted		0	1	
	0	7190	709	7899
	1	127	211	338
Total		7317	920	8237

TABLE. V. RESULTS OF OTHER PERFORMANCE EVALUATION METHODS

Methods	Overall classifier Accuracy	Majority class accuracy	Minority class accuracy	Recall (sensitivity)	Precision
Information gain	89.84	98.3	21.9	98.3	90.9
GINI Index	89.85	98.2	22.9	98.2	91.0

Accuracy, recall, precision and F1 score values are shown in Table V.

Results in Table V, quite clearly show that there is no significant difference between the classification accuracy obtained by the two feature selection measures. Overall accuracy as well as per class accuracy values remain approximately the same. Other observations are in line with literature which says, classifiers trained on low dimensional, imbalanced data classify most of the samples to majority class [23]. Therefore, it is deceivingly simple to achieve high overall accuracy, although it is difficult to classify the data reliably. This is evident from the results obtained, where the majority class accuracy is too high (98.3%) when compared to minority class accuracy (22% approx.). With imbalanced data set, even when the minority class accuracy is very low, the overall accuracy would be high because of high True positive count as in our case. Hence, kappa statistic is measured which takes in to account the chance agreement.

- Kappa Coefficient:

Kappa coefficient is an interesting alternative to measure the accuracy of classifier models. It is particularly useful when the data sets are imbalanced [24]. It is used to quantify the reproducibility of discrete variable.

Originally Cohen's Kappa(κ) coefficient was introduced to measure the level of inter-observer agreement, its value ranging from 0 to 1 [25]. If κ is 0 then the agreement between observed and expected is only by chance; if it 1, it is a perfect agreement. κ value between 0 and 0.2 indicates slight agreement, 0.2 to 0.4 says fair agreement, 0.6 to 0.8 is substantial agreement. [26]. The Kappa (κ) statistic takes into account the chance agreement and is defined as.

$$\text{Kappa } (\kappa) = \frac{\text{Observed Agreement} - \text{Expected Agreement}}{1 - - \text{Expected Agreement}}$$

Kappa coefficient is used to evaluate the accuracy of models by measuring agreement between predicted values and true values. Using the confusion matrix in Table III and Table IV, kappa values for the classifiers are generated as

Kappa value of the classifier model based on Information gain, Kappa (κ) =

$$\frac{\left((7198) + (202)\middle|(8237)\right) - \left((7916)(7317) + (321)(920)\middle|(8237)^2\right)}{1 - \left((7916)(7317) + (321)(920)\middle|(8237)^2\right)}$$

=0.284

Kappa value 0.28 indicates that observed agreement is 28% of the way between chance and perfect agreement.

Kappa value of the classifier model based on GINI index,

Kappa (κ) =

$$\frac{\left((7190) + (211)\middle|(8237)\right) - \left((7899)(7317) + (338)(920)\middle|(8237)^2\right)}{1 - \left((7899)(7317) + (338)(920)\middle|(8237)^2\right)}$$

=0.293

Kappa value 0.29 indicates that observed agreement is 29% of the way between chance and perfect agreement.

It is clearly evident from the results obtained that both the classifier models obtained near to equal results. In other words, the results clearly show that the classification accuracy of decision trees is not sensitive to choice of feature selection measures.

High overall accuracy (89% approx.) and very low minority class accuracy (22%) show that the data is not classified reliably. This could be because the dataset used in the experiment is highly imbalanced with 29231 positive (majority) samples and 3719 negative (minority) samples. In next section we provide the details of methods for balancing the dataset and discuss the results of the experiment conducted after balancing the dataset.

V. Balancing the Dataset

Imbalanced datasets have imbalanced class distribution; where by more observations belong to one class than other. Classification algorithms suffer from the problem of imbalanced dataset which leads to biases and poor generalizations. Sometimes, in real world applications, minority class would be of most interest and classifying them correctly should be given high importance, allowing small error rate in classification of majority class since the cost of misclassifying them could be relatively very [27].

For a binary classification problem, if S is the training data, y is the response variable, [28] defines imbalanced classification problem as follows:

$S = \{(x_1, y_1) \ldots (x_m, y_m)\}$, where $y_i \in \{-1, 1\}$ will be data labels.

$S^+ = \{(x, y) \in S: y = 1\}$ be the positive or minority instances.

$S^- = \{(x, y) \in S: y = -1\}$ be the negative or majority instances.

In the test set if, $|S^+| > |S^-|$, the performance of classification algorithm will be very poor, and misclassification rate will be high especially when it comes to the minority class. Therefore, to improve the performance, resampling methods are applied on the training dataset to generate a new set E with synthetic instances of minority class, transforming the training dataset into, $S = (S^+ \cup E) \cup S^-$

A. Resampling

Imbalanced datasets have imbalanced class distribution. The dataset used for the study is imbalanced with 29231 positive samples and 3719 negative samples. In such situations, it is difficult to classify the data reliably, although it is simple to attain high accuracy. It is quite essential to balance the dataset to classify reliably. Distribution of classes can be balanced by random oversampling minority class observations or random under sampling majority class observations or by combining both over and under in a systematic manner [29]. Random oversampling creates the problem of over fitting the classifiers and under sampling suffers from loss of useful observations. Another heuristic method, SMOTE (Synthetic Minority Oversampling Technique) based on oversampling is widely used which reduces the over fitting to certain extent and performs better than random over sampling. SMOTE generates synthetic observations of minority class [27] [23].

Before applying any of the resampling techniques training and test data must be split to avoid over fitting and poor generalizations. After resampling we have nearly equal ratio of observations for each class in the training set. The number of observations after applying the resampling methods on the training set can be seen in Table VI.

B. Results: Performance Evaluation after Resampling

After balancing the dataset with resampling techniques, the experiment described in section IV is repeated and accuracy is measured. Confusion matrix created after applying resampling techniques is shown in Table VII.

TABLE. VI. Number of Observations after Applying Resampling Techniques

Dataset	Number of features	Training set size	Number of positive samples	Number of Negative samples	Imbalance ratio
Original	20	32950	29231	3719	89:11
Over	20	58462	29231	29231	Equal
Under	20	7438	3719	3719	Equal
Both	20	32950	16556	16394	50.2 : 49.8
SMOTE	20	26033	14876	11157	57 : 43

TABLE. VII. CONFUSION MATRIX WITH DIFFERENT RESAMPLING TECHNIQUES

	OVER			UNDER			BOTH			SMOTE		
Information gain		0	1		0	1		0	1		0	1
	0	5858	311	0	6122	332	0	6041	322	0	6720	459
	1	1459	609	1	1195	588	1	1276	598	1	597	461
GINI index		0	1		0	1		0	1		0	1
	0	5858	311	0	6139	339	0	6064	330	0	6720	459
	1	1459	609	1	1178	581	1	1253	590	1	597	461

Tables VIII and IX summarizes the results obtained by the classification models after applying different resampling techniques. The results in the tables show that balancing the data set has decreased the majority class accuracy but improved the minority class accuracy. Balancing the data set has improved the minority class accuracy by increasing the count of true negative. As discussed earlier it is relatively simple to achieve high overall accuracy with imbalanced data sets, but classifying data reliably is difficult. Thus, after balancing the dataset the objective of classifying data reliably is achieved as the minority class accuracy has improved.

TABLE. VIII. RESULTS OBTAINED WITH DIFFERENT RESAMPLING TECHNIQUES USING INFORMATION GAIN

Metric	Overall Accuracy	Majority class accuracy	Minority class accuracy	Recall	Precision	Kappa
Over	78.5	80.0	66.2	80.0	94.9	29.9
Under	81.4	83.6	63.9	83.6	94.8	33.7
Both	80.6	82.5	65	82.5	94.9	32.7
SMOTE	87.18	91.8	50.1	91.8	93.6	39.3

TABLE. IX. RESULTS OBTAINED WITH DIFFERENT RESAMPLING TECHNIQUES USING INFORMATION GAIN

Metric	Overall Accuracy	Majority class accuracy	Minority class accuracy	Recall	Precision	Kappa
Over	78.5	80.0	66.2	80.0	94.9	29.9
Under	81.5	83.9	63.1	83.9	94.7	33.6
Both	80.7	82.8	64.1	82.8	94.8	32.6
SMOTE	87.18	91.8	50.1	91.8	93.6	39.3

Further analysis of results show that, SMOTE has achieved highest overall accuracy among all the resampling methods. Also, with Smote technique kappa value is 39%. It shows that SMOTE technique is relatively more reliable technique for balancing the dataset than other three methods studied.

VI. CONCLUSIONS

The empirical results reported in this paper show that both Information gain and GINI index produce the same accuracy for classification problems. The experiment is conducted before and after the data set is balanced. The results obtained prove that there is no significant difference in the performance of models using GINI index and Information gain before and after the data set balanced. The results are in line as stated by Mingers [11] that splitting indices have no impact on accuracy. In summary, the results obtained in this paper show that classification accuracy of decision trees for both balanced and imbalanced data sets, is not sensitive to the choice feature selection metrics that were studied.

Another interesting observation is balancing the dataset has lowered the majority class accuracy with decrease in count of true positives and minority class accuracy has improved with increase in the true negative count. In other words, the sensitivity decreased and specificity improved after the data set is balanced. Despite the fact that there is a decrease in overall accuracy, there is clearly a significant rise in the minority class accuracy. This proves that classification accuracy is sensitive to number of positive and negative samples in the data set and type of data, balanced or imbalanced.

REFERENCES

[1] James, G., Witten, D., Hastie, T., and Tibshirani, R.: 'Tree-based methods': 'An introduction to statistical learning' (Springer, 2013), pp. 303-335.

[2] Doherty, C., Camina, S., White, K., and Orenstein, G.: 'The path to predictive analytics and machine learning' (O'Reilly Media, 2017. 2017).

[3] Turban, E., Sharda, R., and Delen, D.: 'Business intelligence and analytics: systems for decision support' (Pearson Higher Ed, 2014. 2014).

[4] Loh, W.-Y., and Shih, Y.-S.: 'Split selection methods for classification trees', Statistica sinica, 1997, pp. 815-840.

[5] Breiman, L., Friedman, J.H., Olshen, R.A., and Stone, C.J.: 'Classification and regression trees. Belmont, CA: Wadsworth', International Group, 1984, 432, pp. 151-166.

[6] Raileanu, L.E., and Stoffel, K.: 'Theoretical comparison between the gini index and information gain criteria', Annals of Mathematics and Artificial Intelligence, 2004, 41, (1), pp. 77-93.

[7] Shannon, C.E.: 'A note on the concept of entropy', Bell System Tech. J, 1948, 27, (3), pp. 379-423.

[8] Wang, Y., Li, Y., Song, Y., Rong, X., and Zhang, S.: 'Improvement of ID3 algorithm based on simplified information entropy and coordination degree', Algorithms, 2017, 10, (4), pp. 124.

[9] Fan, R., Zhong, M., Wang, S., Zhang, Y., Andrew, A., Karagas, M., Chen, H., Amos, C., Xiong, M., and Moore, J.: 'Entropy - based information gain approaches to detect and to characterize gene - gene and gene - environment interactions/correlations of complex diseases', Genetic epidemiology, 2011, 35, (7), pp. 706-721.

[10] Lefkovits, S., and Lefkovits, L.: 'Gabor feature selection based on information gain', Procedia Engineering, 2017, 181, pp. 892-898.

[11] Mingers, J.: 'An empirical comparison of selection measures for decision-tree induction', Machine learning, 1989, 3, (4), pp. 319-342.

[12] Patil, N., Lathi, R., and Chitre, V.: 'Comparison of C5. 0 & CART classification algorithms using pruning technique', Int. J. Eng. Res. Technol, 2012, 1, (4), pp. 1-5.

[13] Suneetha, N., Hari, V., and Kumar, V.S.: 'Modified gini index classification: a case study of heart disease dataset', International Journal on Computer Science and Engineering, 2010, 2, (06), pp. 1959-1965.

[14] Adhatrao, K., Gaykar, A., Dhawan, A., Jha, R., and Honrao, V.: 'Predicting students' performance using ID3 and C4. 5 classification algorithms', arXiv preprint arXiv:1310.2071, 2013.

[15] Hssina, B., Merbouha, A., Ezzikouri, H., and Erritali, M.: 'A comparative study of decision tree ID3 and C4. 5', International Journal of Advanced Computer Science and Applications, 2014, 4, (2), pp. 13-19.

[16] Moro, S., Cortez, P., and Rita, P.: 'A data-driven approach to predict the success of bank telemarketing', Decis Support Syst, 2014, 62, pp. 22-31.

[17] SHARDA, R.D.: 'BUSINESS INTELLIGENCE AND ANALYTICS: Systems for Decision Support' (PRENTICE HALL, 2016. 2016).

[18] https://people.revoledu.com/kardi/tutorial/DecisionTree.

[19] https://dataaspirant.com/2017/02/03/decision-tree-classifier-implementation-in-r/.

[20] Therneau, T., Atkinson, B., Ripley, B., and Ripley, M.B.: 'Package 'rpart'', Available online: cran. ma. ic. ac. uk/web/packages/rpart/rpart. pdf (accessed on 20 April 2016), 2015.

[21] Zheng, A.: 'Evaluating machine learning models: a beginner's guide to key concepts and pitfalls', 2015.

[22] Tan, P.-N., Steinbach, M., and Kumar, V.: 'Introduction to data mining' (Pearson Education India, 2016. 2016).

[23] Blagus, R., and Lusa, L.: 'Improved shrunken centroid classifiers for high-dimensional class-imbalanced data', BMC bioinformatics, 2013, 14, pp. 64-64.

[24] McHugh, M.L.: 'Interrater reliability: the kappa statistic', Biochemia medica: Biochemia medica, 2012, 22, (3), pp. 276-282.

[25] McGee, S.: 'Evidence-based physical diagnosis e-book' (Elsevier Health Sciences, 2012. 2012).

[26] Ensrud, K.E., and Taylor, B.C.: 'Epidemiologic Methods in Studies of Osteoporosis': 'Osteoporosis' (Elsevier, 2013), pp. 539-561.

[27] Zheng, Z., Cai, Y., and Li, Y.: 'Oversampling method for imbalanced classification', Computing and Informatics, 2016, 34, (5), pp. 1017-1037.

[28] Cordón, I.: 'Working with imbalanced datasets'.

[29] Wasikowski, M.: 'Combating the class imbalance problem in small sample data sets', University of Kansas, 2009.