Assignment 4 Version 2: A mathematical essay on Decision Trees

S, Karthik
ME18B149 - IDDD Data Science
Indian Institute of Technology Madras
Chennai, India
me18b149@smail.iitm.ac.in

***Abstract*—This document is an overview of the concept of Decision Trees. It gives a general idea of the topic, along with the mathematical aspects, as well as an application involving it based on data from a real-world problem.**

***Index Terms*—Classification, Decision Trees, Machine Learning, Data Science**

I. Introduction

Decision tree is one of the most powerful supervised learning tools for classification and regression. Decision tree has a flowchart like tree structure used to visualize the decision making process by mapping out different courses of action, as well as their potential outcomes. As they can be used for both classification and regression, they are collectively called Classification and Regression Trees (CART).

The specific problem discussed is a classification problem which determines the level of safety of the cars based on certain features using the Car Evaluation Dataset. Decision Trees will be used to analyze the data and build a model which will then be able to predict the category for unknown data. Various insights and conclusions will be made based on the trends followed by the dataset.

This paper is a case study through which the principles of Decision Trees are implemented. It aims to examine which parameters influence the level of safety of the car by automotive standards.

II. Decision Trees

Decision Trees (DTs) are a non-parametric supervised learning method used for classification and regression. The goal is to create a model that predicts the value of a target variable by learning simple decision rules inferred from the data features. A tree can be seen as a piecewise constant approximation.

A. Model Structure

Fig 2 shows the model structure of a decision tree. The components of a decision tree are the following:

1) Nodes: Represents a condition on a feature based on which tree splits into branches. Following are the types of nodes

- Root Node: Top-level node in the decision tree where the first conditional split happens.
- Internal Node: Node with incoming and outgoing branches having conditional splits.

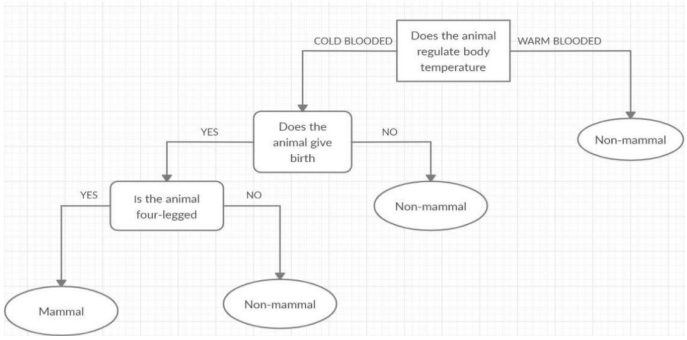

Fig. 1. Example: A simple decision tree to classify an animal as mammal or non-mammal

- Leaf Node: Terminal of a branch that doesn't split anymore and represents a target label or target variable.

2) Branches: Represents different options that are available based on the nodal condition

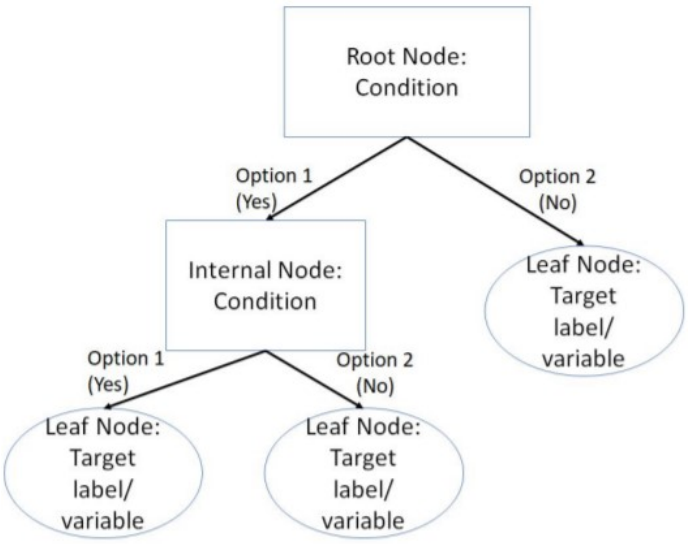

Fig. 2. General structure of a decision tree

B. Types

1) Regression Tree: A regression tree is used when the dependent variable is continuous. The value obtained by leaf nodes in the training data is the mean response of observation falling in that region. Thus, if an unseen data observation falls in that region, its prediction is made with the mean value. This means that even if the dependent variable in training data was continuous, it will only take discrete values in the test set. A regression tree follows a top-down greedy approach.

2) Classification Tree: A classification tree is used when the dependent variable is categorical. The value obtained by leaf nodes in the training data is the mode response of observation falling in that region It follows a top-down greedy approach.

C. Splitting criteria

The following are the types of criteria used for deciding which variable takes the node.

1) Gini impurity: Gini impurity can be understood as a criterion to minimize the probability of misclassification. If we select two items from a population at random then they must be of the same class and the probability for this is 1 if the population is pure. Equation 1 gives the expression for calculating the Gini impurity for a particular parameter in a particular node.

$$I_G = 1 - \Sigma_{i=1}^{k} p(i|n)^2 \tag{1}$$

where $p(i|n)$ is the proportion of samples that belong to a class k for a particular node n. The overall Gini score for split using the weighted Gini score of each node of that split is calculated for all the parameters. The parameter with the least impurity score is chosen for the node.

2) Chi-squared: It is an algorithm to find out the statistical significance between the differences between sub-nodes and parent node. We measure it by the sum of squares of standardized differences between observed and expected frequencies of the target variable.

First, Chi-square for an individual node is calculated using the deviation for both Success and Failure. Chi-square of Split is calculated using Sum of all Chi-square of success and Failure of each node of the split. Then, the feature with the max chi-squared value is selected

3) Information Gain: A less impure node requires less information to describe it and, a more impure node requires more information. Information theory is a measure to define this degree of disorganization in a system known as Entropy. If the sample is completely homogeneous, then the entropy is zero and if the sample is equally divided (50% — 50%), it has an entropy of one. Steps to calculate entropy for a split:

$$Entropy = -p \log_2 p - q \log_2 q \tag{2}$$

$$InformationGain = 1 - Entropy$$

- Calculate the entropy of the parent node
- Calculate entropy of each individual node of split and calculate the weighted average of all sub-nodes available in the split. The lesser the entropy, the better it is.
- Calculate information gain as follows and choose the node with the highest information gain for splitting

D. Advantages

- Easy to visualize and interpret: Its graphical representation is very intuitive to understand and it does not require any knowledge of statistics to interpret it.
- Useful in data exploration: We can easily identify the most significant variable and the relation between variables with a decision tree. It can help us create new variables or put some features in one bucket.
- Less data cleaning required: It is fairly immune to outliers and missing data, hence less data cleaning is
- needed. The data type is not a constraint: It can handle both categorical and numerical data.

E. Disadvantages

- Overfitting: single decision tree tends to overfit the data which is solved by setting constraints on model parameters i.e. height of the tree and pruning(which we will discuss in detail later in this article)
- Not exact fit for continuous data: It losses some of the information associated with numerical variables when it classifies them into different categories.

F. The problem of overfitting

Overfitting is one of the key challenges in a tree-based algorithm. If no limit is set, it will give 100% fitting, because, in the worst-case scenario, it will end up making a leaf node for each observation. Hence we need to take some precautions to avoid overfitting. It is mostly done in two ways:

- Setting constraints on tree size such as maximum depth, minimum samples for a node split, minimum samples for a leaf node, maximum features to consider for a split, etc.
- Tree pruning - this is a technique to optimize the size of the decision tree by removing the redundant and non-critical sections.

III. The Problem

A. Overview and objectives

The Car Evaluation Database was derived from a simple hierarchical decision model. The prediction task is to classify a car based on its safety.

In this assignment, we attempt to build a predictive Decision Tree model that answers the question: "what factors of people are more likely to influence the outcome?" using the above data (i.e doors, price, persons, etc).

B. Reading the data

We will be using python throughout this assignment. A single dataset titled "car_evaluation.csv" is provided which contains the details of various parameters of come cars which are shown in detail in Fig 3. Since we are creating a model,

we would be splitting the dataset into train and validation sets with a fraction of 10% for validation.

IV. Data Cleaning

After reading the data, the next task will be to filter out the attributes or entries that are not required or redundant in this analysis. There are no unfilled values in this dataset which means no entry is being eliminated. Also, all the features are different enough and no feature is redundant in this case which means the dataset will be used completely as it is without any alterations.

V. Exploratory Data Analysis

The complete details of all the attributes are shown in Fig 3.

Variable	Definition	Key
buying	buying price	vhigh, high, med, low
maint	Price of the maintenance	vhigh, high, med, low
doors	Number of doors	2, 3, 4, 5, more
persons	Capacity in terms of persons to carry	2, 4, more
lug_boot	The size of luggage boot	small, med, big
safety	Estimated safety of the car	low, med, high
Target	Target variable to predict	unacc, acc, good, vgood

Fig. 3. Exhaustive list of all the attributes in the given dataset

Our first task is to analyze the data and form visible conclusions and trends from the same with the help of plots.

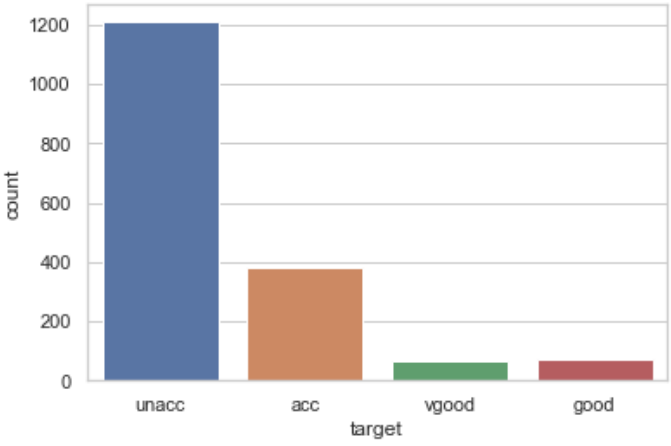

Fig. 4. Bar chart - Distribution among outcomes

From Fig 4, we can see that the dataset is a highly imbalanced and a skewed one as the number of unacceptable cars are much higher than others.

Fig 5 to Fig 10 shows the classification of the outcomes based on all the features individually. We can see that due to the skewness, every possible value of a feature has maximum fraction of unacceptable cars only for buying and maintenance costs, number of doors and lug boot size. Hence, we can conclude that these are the features with less weightage.

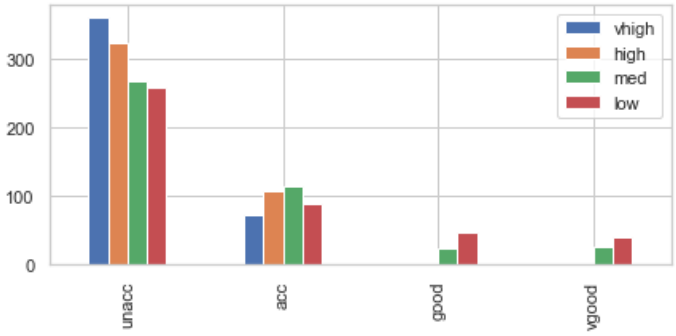

Fig. 5. Bar chart - Classification by price

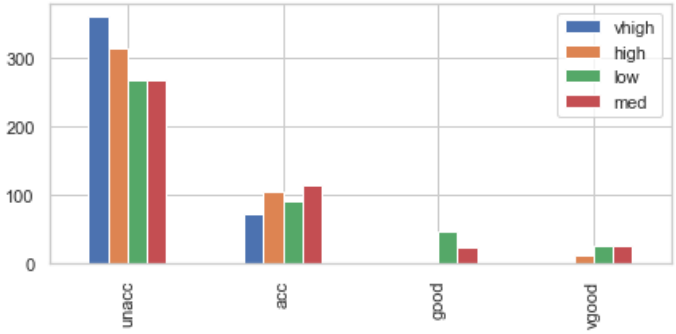

Fig. 6. Bar chart - Classification by maintenance cost

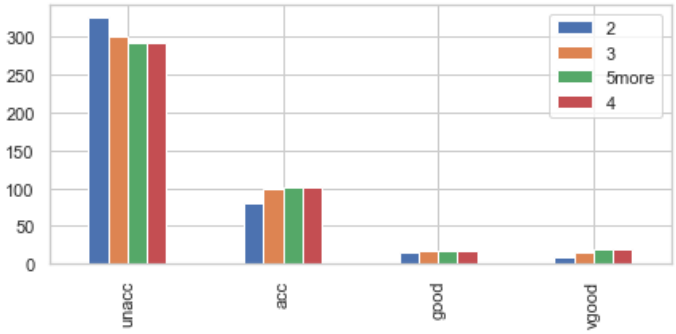

Fig. 7. Bar chart - Classification by number of doors

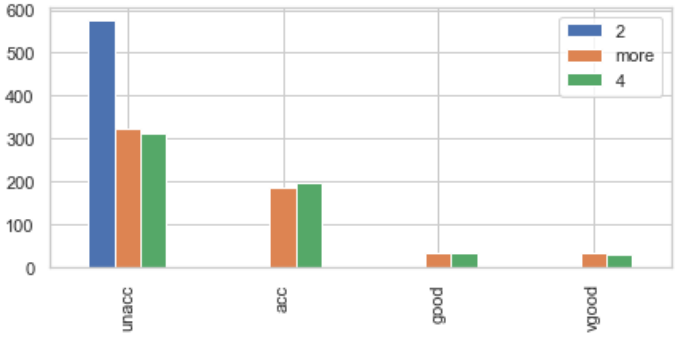

Fig. 8. Bar chart - Classification by capacity

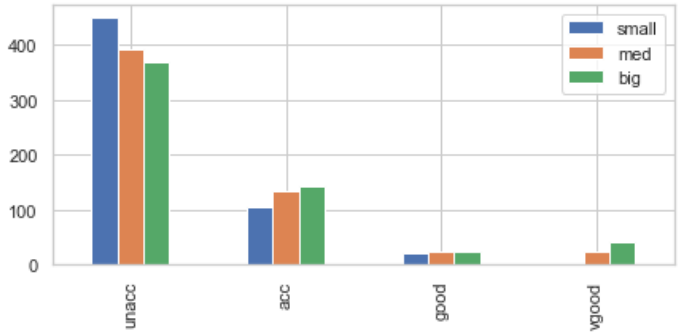

Fig. 9. Bar chart - Classification by lug boot size

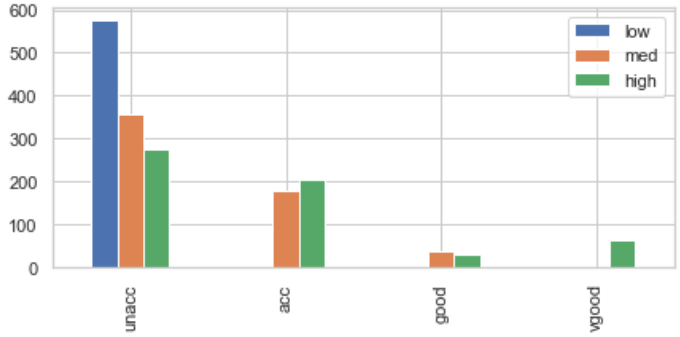

Fig. 10. Bar chart - Classification by safety

On the other hand, if we see the safety feature (Fig 10), it is clear that the majority of the unacceptable cars are of low safety which is reasonable as the car has to be safe enough by the automotive standards.

VI. Data preparation

The given dataset is clean in terms of missing data. All the variables are categorical. Hence, label encoding was done for every variable starting with the least possible outcome assigned to 0 and increased subsequently.

TABLE I
SAMPLE ENTRIES OF MODEL COMPATIBLE DATASET

S.No	buying	maint	doors	persons	lug_boot	safety	target
0	3	3	0	0	0	0	unacc
1	3	3	0	0	0	1	unacc
2	3	3	0	0	0	2	unacc
3	3	3	0	0	1	0	unacc
4	3	3	0	0	1	1	unacc

VII. Model preparation, training and validation

Since the datasets are now model compatible, the next step is to create the model. The entire dataset was split into training and validation datasets with a fraction of 10% kept for validation. Validation accuracy, i.e, the fraction of entries for which the predicted and the actual value are the same will be used as the metric to compare between models. 3 parameters (min_samples_leaf, min_samples_split, max_depth) were tuned for a range and graphs plotted between the parameter values and the accuracy. From Fig 11 to Fig 13, the values for the final model was determined as shown in Table II.

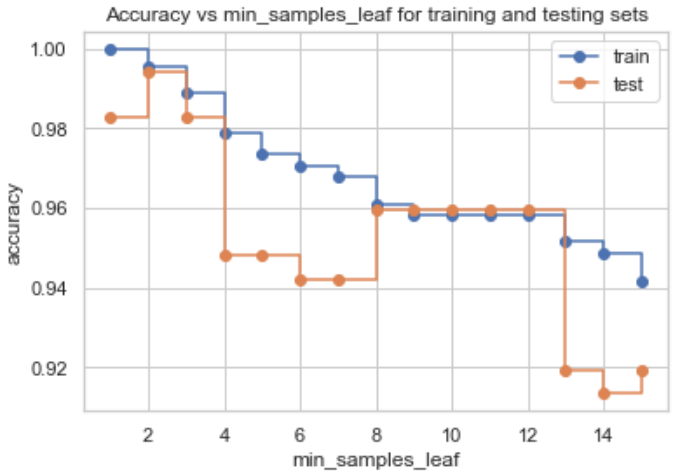

Fig. 11. Plot between accuracy and min_samples_leaf

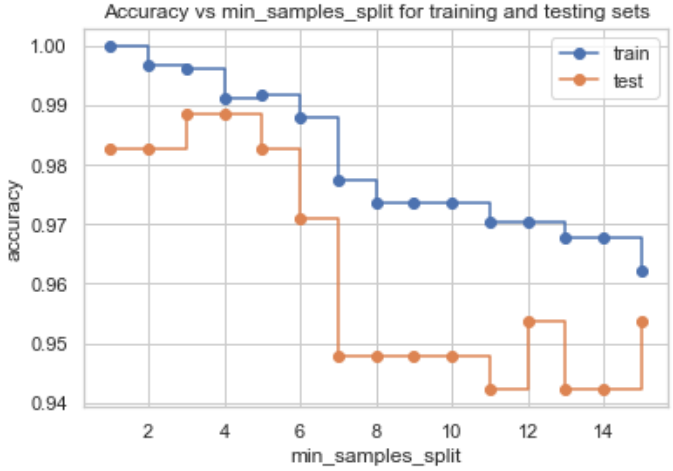

Fig. 12. Plot between accuracy and min_samples_split

TABLE II
HYPERPARAMETER TUNING RESULTS

S. No.	Parameter	Value
1	min_samples_leaf	2
2	min_samples_split	4
3	max_depth	10

From Fig 13, one very interesting observation which is atypical of any dataset is that even though the test data WAS NOT used in training, the accuracy proportionally increases therefore unable to determine overfitting nature.

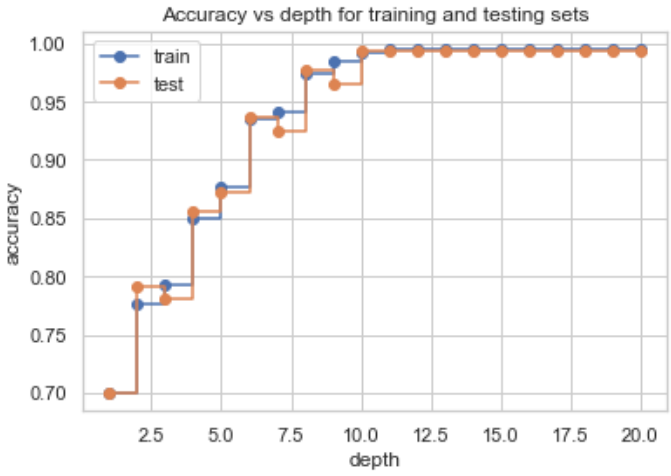

Fig. 13. Plot between accuracy and depth

This can be possible due to the test data being very close/identical entries to training data.

A. Pruning

When the dataset was roughly trained, the accuracy was pretty high in the ranges of 96%. Hence, intuitively speaking, pruning is not required in this case. Still, various values of the cost complexity parameter were used and plotted which complimented the intuitive reasoning of not requiring pruning.(Fig 14)

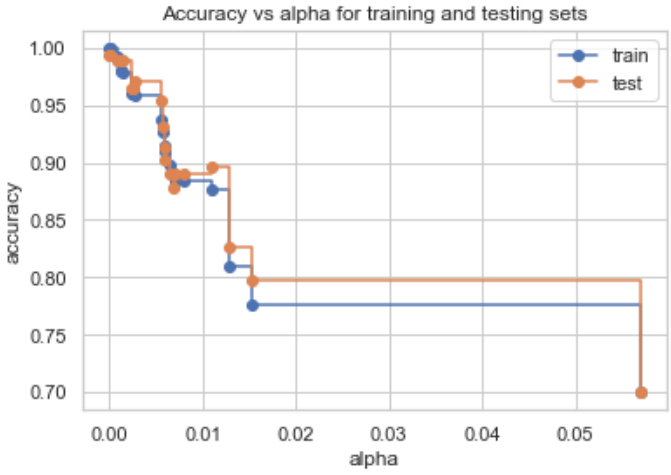

Fig. 14. Plot between accuracy and pruning cost complexity parameter

B. Final Model

Using the above parameters, a Decision tree classifier was created and the trained using the training data. The validation accuracy is determined to be **99.42%**.

Also, the confusion matrix which is the plot between the actual outcome and the predicted outcome is plotted for the model and is shown in Fig 15. Only one entry is incorrectly predicted from the test dataset which shows that this is a highly accurate model.

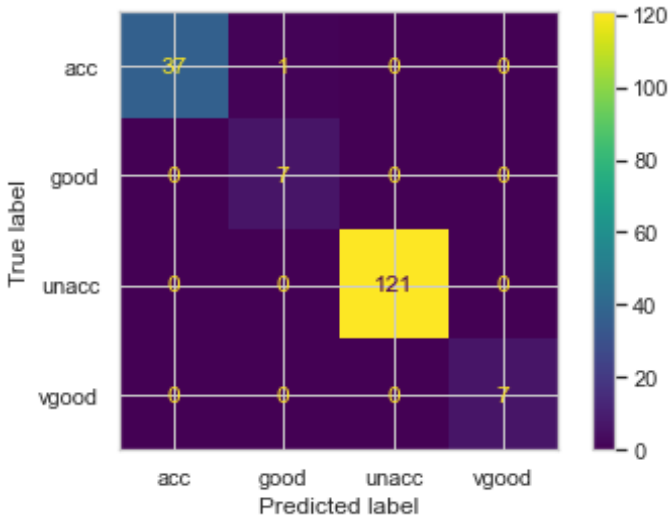

Fig. 15. Confusion matrix - test data

VIII. Conclusion

The effects of various factors which could have affected the income of the classification of the car in the Car Evaluation Database were analyzed in great detail. Various factors greatly affected the outcome like safety, number of doors, price, number of passengers.

Various decision tree models were constructed on the basis of the known data to determine the right set of hyperparameters which with the best possible model was created with a validation accuracy of 99.42%. Using this model, the confusion matrix was plotted which gave the exact numbers predicted correctly and incorrectly.

Further avenues could be explored using this data where they can be classified using various other classifiers available to study how the parameters might possibly affect each other and also how they affect the income. Also, implementing techniques more advanced than Decision Trees might give a model with better accuracy and predictions which would improve on the current model.

IX. Addendum for End Semester

A. Entropy Criterion and Grid Search

Since we used Gini criterion in the previous analysis and found the most accurate model among them, we have the option to use Grid Search around these parameters to check if the accuracy can be optimized further. Table III gives us the parameters given for the grid search.

Upon Grid Search, it was found that the most accurate model is still the same as the one done manually with a classifciation accuracy of **99.42%**. Hence, the final model is still unchanged

TABLE III
HYPERPARAMETER TUNING - GRID SEARCH

S.No	Parameter	Values
1	ccp$_alpha$	0.1, 0.01, 0.001
2	criterion	'gini', 'entropy'
3	L1-ratio	5, 6, 7, 8, 9,10,11,12
4	class$_weight$	'auto', 'sqrt', 'log2'

B. Attributes from the confusion matrix

We will try to find some useful parameters for the best model from the results.

1) Precision: Precision can be defined as the percentage of correctly predicted positive outcomes out of all the predicted positive outcomes. It can be given as the ratio of true positives (TP) to the sum of true and false positives (TP + FP).

So, Precision identifies the proportion of correctly predicted positive outcome. It is more concerned with the positive class than the negative class.

Mathematically, precision can be defined as the ratio of TP to (TP + FP). In the best model, it comes out to be 96.55%.

2) Recall/Sensitivity: Recall can be defined as the percentage of correctly predicted positive outcomes out of all the actual positive outcomes. It can be given as the ratio of true positives (TP) to the sum of true positives and false negatives (TP + FN). Recall is also called Sensitivity.

Recall identifies the proportion of correctly predicted actual positives.

Mathematically, recall can be given as the ratio of TP to (TP + FN). In the best model, it comes out to be 100%. This means it is very close to a perfect model.

3) True and False Positive Rate: True positive rate is same as recall (TP/(TP+FN)). False positive rate is given as (FP/(FP+TN)). In this model, True Positive rate is 100% and False Positive Rate is 12.50%.

TABLE IV
BEST MODEL ATTRIBUTES

S.No.	Attribute	Value
1	Classification accuracy	0.9942
2	Classification error	0.0278
3	Precision	0.9655
4	Recall or Sensitivity	1
5	True Positive Rate	1
6	False Positive Rate	0.1250
7	Specificity	0.8750

REFERENCES

[1] Dr. Kaushik Mitra, Dr. Ramkrishna Pasumarthy, "Decision Trees", EE4708: Data Analytics Laboratory - Week 6

[2] Decision Trees — Detailed Overview, Towards Data Science.

[3] Decision Trees - https://scikit-learn.org/stable/modules/tree.htmltips-on-practical-use