# Privacy Preserving Decision Tree Prediction

Reem Younis, Assia Khateeb, Atheer Abo Foul
*Lecturer : Dr. Adi Akavia, Laboratory in Privacy Preserving Machine Learning, University of Haifa*
*Email: reembyounis@gmail.com, assia.khteb@gmail.com, 19aether6@gmail.com*

***Abstract*—In machine learning, the decision tree is an algorithm for supervised learning for classification. The algorithm allows for learning, in that it processes elements in the training set one at a time.We implement decision tree algorithm on clear-text dataset. Then, we test accuracy of our decision tree algorithm classification results.**

## 1. Introduction

Decision Tree algorithm belongs to the family of supervised learning algorithms. Unlike other supervised learning algorithms, decision tree algorithm can be used for solving regression and classification problems too.

The general motive of using Decision Tree is to practice it .
Decision Tree represents a procedure for classifying data based on attributes or features. It is also an efficient way of processing data ,for this very reason it has wide application in data mining.
In Decision Tree structure each internal node represents a "test" on an attribute, each branch represents the outcome of the test, and each leaf node represents a class label. The paths from root to leaf represent classification rules.
This data structure is quite intuitive and easy to assimilate by humans.

## 2. Methodology

The working model of decision tree is quite easy to implement and can be very effective in most of the classification problems.
In our decision tree, for predicting a class label for a dataset We compare the values of the root attribute with dataset's attribute. On the basis of comparison, we follow the branch corresponding to that value and jump to the next node.

We continue comparing our dataset's attribute values with other internal nodes of the tree until we reach a leaf node with predicted class value.

### 2.1. Prediction And Samples

Assumptions we make while using Decision Tree :
*1.* At the beginning, we consider the whole training set as the root.
*2.* Attributes are assumed to be categorical for information gain and for gini index, attributes are assumed to be continuous.
*3.* On the basis of attribute values records are distributed recursively.
*4.* We use statistical methods for ordering attributes as root or internal node.

### 2.2. Attribute Selection Measures

Attribute selection measure is a heuristic for selecting the splitting criterion that partition data into the best possible manner. It is also known as splitting rules because it helps us to determine breakpoints for tuples on a given node. ASM provides a rank to each feature(or attribute) by explaining the given data set. Best score attribute will be selected as a splitting attribute (Source). In the case of a continuous-valued attribute, split points for branches also need to define. Most popular selection measures are Information Gain, Gain Ratio, and Gini Index.

### 2.3. Entropy

In physics and mathematics, entropy referred as the randomness or the impurity in the system. In information theory, it characterizes the impurity of an arbitrary collection of examples. Entropy is the measure of uncertainty of a random variable, The higher the entropy the more the information content.
The entropy can explicitly be written as:
H(X) = $\sum_{n=1}^{N} p(x_i) \log_2 p(x_i)$

By calculating entropy measure of each attribute we can calculate their information gain. Information Gain calculates the expected reduction in entropy due to sorting on the attribute. Information gain can be calculated.

### 2.4. Information Gain

Information gain is the decrease in entropy. Information gain computes the difference between entropy before split and average entropy after split of the data set based on given attribute values.
Definition: Suppose S is a set of instances, A is an attribute, $\S_v$ is the subset of S with A=v and Values(A) is the set of all possible of A, then
Gain(S,A) = Entropy(S) - $\sum_{v:val(A)} |S_v| Entropy(|S_v|)$
$|S|$ denotes the size of set S

### 2.5. Gini index

Another decision tree algorithm CART (Classification and Regression Tree) uses the Gini method to create split points.
Gini index and Information Gain both of these methods are used to select from the n attributes of the dataset which attribute would be placed at the root node or the internal node.
Gini index = 1 - $\sum_j P_j^2$

## 3. Algorithm

How the algorithm works?
*1.* Select the best attribute using Attribute Selection Measures(ASM) to split the records.
*2.* Make that attribute a decision node and breaks the dataset into smaller subsets.
*3.* Starts tree building by repeating this process recursively for each child until one of the condition will match:
–All the tuples belong to the same attribute value.
–There are no more remaining attributes.
–There are no more

### 3.1. Pruning Strategy

To prune each node one by one (except the root and the leaf nodes), and check weather pruning helps in increasing the accuracy, if the accuracy is increased, prune the node which gives the maximum accuracy at the end to construct the final tree (if the accuracy of 100% is achieved by pruning a node, stop the algorithm right there and do not check for further new nodes).

## 4. Decision Tree - Python

### 4.1. driver.py

This file gets input from online sources (for example IRIS data ”https://archive.ics.uci.edu/ml/machine-learning-databases/iris/iris.data”).
In addition, all the functions in DecisionTree.py are called in this file. For example : build_tree, getLeafNodes, getInnerNodes, computreAccuracy, print_tree.

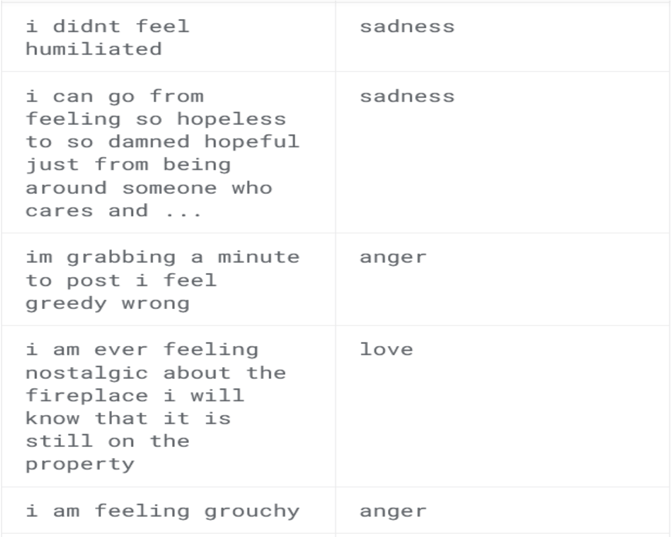
### 4.2. DecisionTree.py

function build_tree : a recursice function that returns the final tree.
rows - contains number of objects.
header - contains number of columns/features/labels.
function find_best_split : returns best question that could be asked so far, in addition to best information gain.
rows,header refer to the same as in build_tree.
function Leaf : initializes the leaf data.
function partition : checks if a question matches an object, if yes then add to true_rows, if no add to false_rows. returns two arrays, true_rows,false_rows.
function Decision_Node : This holds a reference to the question, and to the two child nodes.

## 5. Datasets

### 5.1. Emotions in text

In this dataset we identify the emotion behind the words written. There are two columns, Text and Emotions. Quite self explanatory right. The Emotions column has various categories ranging from happiness to sadness to love and fear. We have fun building models which can identify what words denote what emotion.



| | |
|---|---|
| i didnt feel humiliated | sadness |
| i can go from feeling so hopeless to so damned hopeful just from being around someone who cares and ... | sadness |
| im grabbing a minute to post i feel greedy wrong | anger |
| i am ever feeling nostalgic about the fireplace i will know that it is still on the property | love |
| i am feeling grouchy | anger |

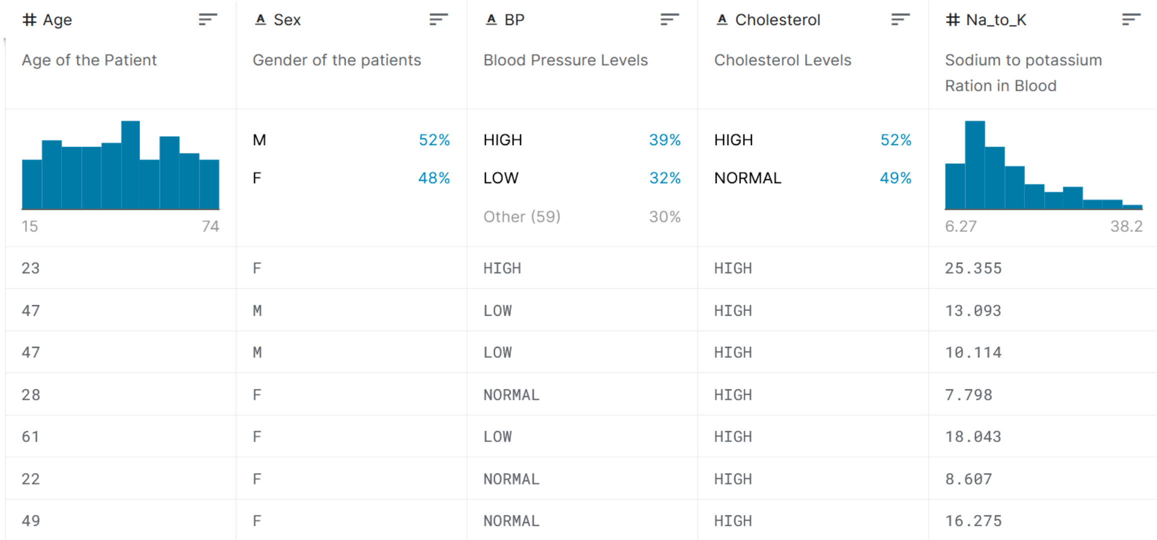
Figure 1. Emotions in text Dataset

### 5.2. Drug Classification

The dataset contains various information that effect the predictions like Age, Sex, BP, Cholesterol levels, Na to Potassium Ratio and finally the drug type.

The target feature is:
–Drug type

The feature sets are:
*1.* Age
*2.* Sex
*3.*Blood Pressure Levels (BP)
*4.* Cholesterol Levels
*5.* Na to Potassium Ration

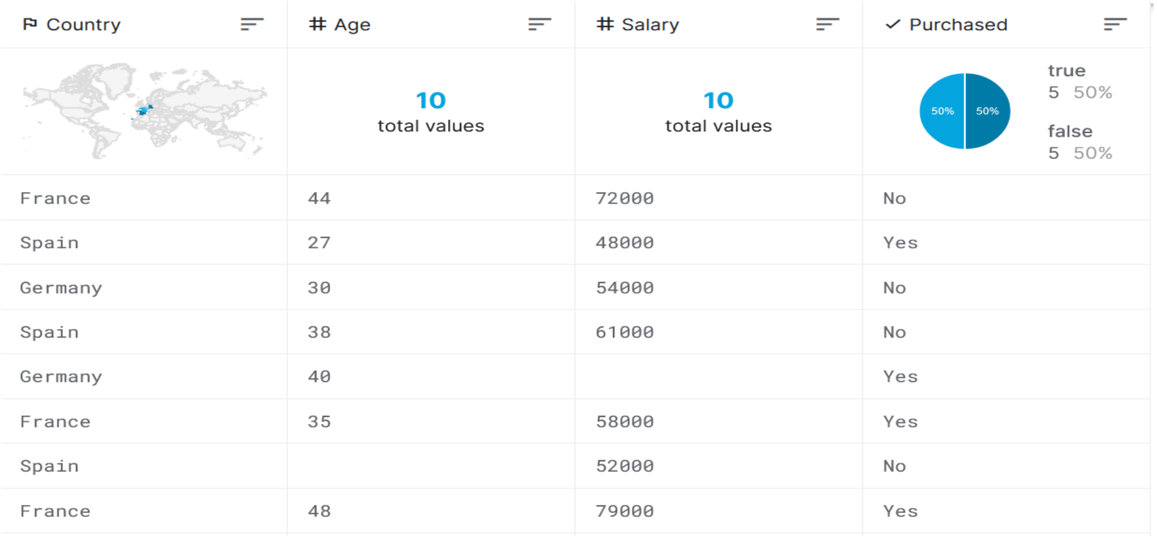


| # Age | Sex | BP | Cholesterol | # Na_to_K |
|---|---|---|---|---|
| Age of the Patient | Gender of the patients | Blood Pressure Levels | Cholesterol Levels | Sodium to potassium Ration in Blood |
| 15 74 | M 52%<br>F 48% | HIGH 39%<br>LOW 32%<br>Other (59) 30% | HIGH 52%<br>NORMAL 49% | 6.27 38.2 |
| 23 | F | HIGH | HIGH | 25.355 |
| 47 | M | LOW | HIGH | 13.093 |
| 47 | M | LOW | HIGH | 10.114 |
| 28 | F | NORMAL | HIGH | 7.798 |
| 61 | F | LOW | HIGH | 18.043 |
| 22 | F | NORMAL | HIGH | 8.607 |
| 49 | F | NORMAL | HIGH | 16.275 |

Figure 2. Drug Classification Dataset

### 5.3. Salary

This dataset contains age and salary information of people from a different region with conditions of them whether a buying a product or not.



| Country | # Age | # Salary | ✓ Purchased |
|---|---|---|---|
| | 10<br>total values | 10<br>total values | true<br>5 50%<br>false<br>5 50% |
| France | 44 | 72000 | No |
| Spain | 27 | 48000 | Yes |
| Germany | 30 | 54000 | No |
| Spain | 38 | 61000 | No |
| Germany | 40 | | Yes |
| France | 35 | 58000 | Yes |
| Spain | | 52000 | No |
| France | 48 | 79000 | Yes |

Figure 3. Salary Dataset

## 6. Results

### 6.1. Sample outputs (Emotions in text Dataset)

–Accuracy before pruning: 40.0%
–Accuracy after pruning: 40.0%
Pruning strategy did not increased accuracy
–Final Tree with accuracy: 40.0%

### 6.2. Sample outputs (Drug Classification Dataset)

–Accuracy before pruning: 97.0%
–Accuracy after pruning: 97.0%
Pruning strategy did not increased accuracy
–Final Tree with accuracy: 97.0%

### 6.3. Sample outputs (Salary Dataset)

–Accuracy before pruning: 0.0%
–Accuracy after pruning: 0.0%
Pruning strategy did not increased accuracy
–Final Tree with accuracy: 0.0%