Study on effect of number of partitions in Regression Trees

by Joan Pareras Velasco

January 7, 2022

Abstract

Regression trees are the most intuitive machine learning models. Using a classification that fit the data the best, uses an iterative binary partition in order to classify the targets based on the given features. The main objective of the project is to study the effect of number of partitions on a data base of kangaroo species.

1 Introduction

First presented in 1963 by Morgan and Sonquist [JNM63] and later statistically formalized by Breiman, Friedman, Olshen and Stone, in 1984 [BFSO84], regression trees are used to model a response variable, defined as Y, using a set of predictive variables, defined as $X_1, X_2, ..., X_p$.

This method works by doing recursive binary partitions of the data. Two criterion are needed. In the one hand, the partitioning criterion, which will determine how to select the best predictive variable among those available variables and which point along the range of that predictor is selected to make the split. In the other hand, a stopping criterion is needed to decide when to stop partitioning.

2 Objectives

The main goal of delivery paper is to show the effect of the number of partitions in regression trees. This will be showed using the kanga data-frame from the open-library faraway. This data-frame shows three different species of kangaroo and the measurements of their skull. Using these skull measurement characteristics the decision tree model will have to be able to differentiate between each specie given the skull measurements of a single individual.

3 Introduction to Regression Trees

As previously explained regression trees are used to make decisions based on some features. These idea is probably the easiest and more intuitive way to proceed. In our daily life this procedure is the way we use to differentiate between things.

Lets present an example. Imagine you have to explain someone that has never seen neither a pen or a pencil, which is which. You will basically make a comparison between them so the other person can identify which name corresponds to each of the objects. Notice, that some characteristics may be more important than others. For instance, as the shape is more or less the same, it will be quicker to start with the difference between both mechanisms.

Taking into account this example, there are some targets that should be classified and to do that there are some features to look at. The model will choose the most relevant features and will split the range of that parameter in two parts. This will be done iteratively until we arrive to a certain criterion of convergence. This criterion can be understood as when its considered that the model is capable to make a good prediction.

Someone may ask why not doing infinite partitions until having each possibility associated to a target. In many cases this could be impossible as the parameters could have an infinite range of possibilities or because a huge computation time and data will be needed.

The model is able to achieve a certain precision based on the amount of data and its quality. So most of the times the model precision will not achieve a 100% when characterizing each input.

Another important concept to consider is the number of partitions the model have. Meaning how many times our data has been split in order to characterize each target. This is a very important characteristic as will let the model be more accurate. However, as explained before,

there are times where doing more partitions is counterproductive, as the model is splitting the data used to train the model, and could be not enough to characterize precisely each target given any features.

4 Mathematical strategy

As previously mentioned, the regression tree model has two criterion that need to be established. This will be a problem specific task, meaning that in each case the best option has to be found.

4.1 Classification criteria

In this particular, the function to measure the quality of a split will be the residual sum of squares (RSS) For a variable j and an observation i

$$RSS_{ij} = RSS\left(part_{X_j \leq x_{ij}}\right) + RSS\left(part_{X_j > x_{ij}}\right) = \sum_{i|X_j \leq x_{ij}} (y_i - \bar{y}_1)^2 + \sum_{i|X_j > x_{ij}} (y_i - \bar{y}_i)^2 \quad (1)$$

where

$$\bar{y}_1 = \frac{1}{\sum l(X_j > x_{ij})} \sum_{i|X_j \leq x_{ij}} y_i \quad (2)$$

and

$$\bar{y}_2 = \frac{1}{\sum l(X_j > x_{ij})} \sum_{i|X_j > x_{ij}} y_i \quad (3)$$

This way the model selects where to do the partitions such that the RSS is the lowest possible.

4.2 Condition of ending

As the main goal of this paper is to study the effect of increasing the number of partitions, there will not be any ending criterion a priory. It will be discussed on the conclusions.

5 Our case

At the end of this paper, kanga data-frame can be found. Notice that the data-frame contains some nan values. In order to make the model work, this columns have to be discarded, so at the end only the remaining columns persist: species, sex, nasal width, lacrymal width, orbital width, crest width, foramina length, mandible width, mandible depth, ramus height. Notice also that the species which are our target and sex are not numbers which brings incompatibility with the model, as it is a numerical based model, so an integer number is associated to each word.

To be able to prove that the model works, a split on the data has been done. The 70% of the data has been utilized for training the model and the other 30% has been used to study the results.

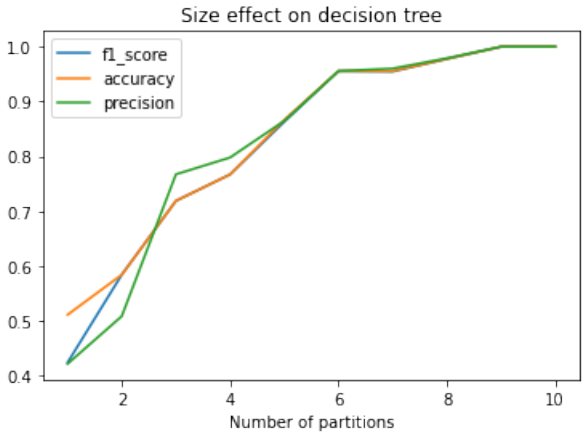


Figure 1: In this figure, the effect of having more partitions can be observed. The f1 score, accuracy and precision have been computed for each of the Regression Trees.

In the figure 1 the effect of size in different measures let us know the validity of the model. Notice that for 9 layers the model already predicts with a 100% accuracy the values. Notice also, that there are three different measures represented with different colours. The first measure to consider is the accuracy shows the percentage of predicted values that are actually the real ones. The second measure corresponds to $tp/(tp + fp)$ where tp is the number of true positives and fp the number of false positives. Intuitively its the ability of the classifier not to label as positive a sample that is negative. Finally, the f1 takes into account both, precision and recall, and it is defined as $2(precision * recall)/(precision + recall)$ where recall is defined as $tp/(tp + fn)$ where fn is the number of false negative. This measure is the most relevant as shows actually how good the model is on predicting both positive and negatives [PVG+11].

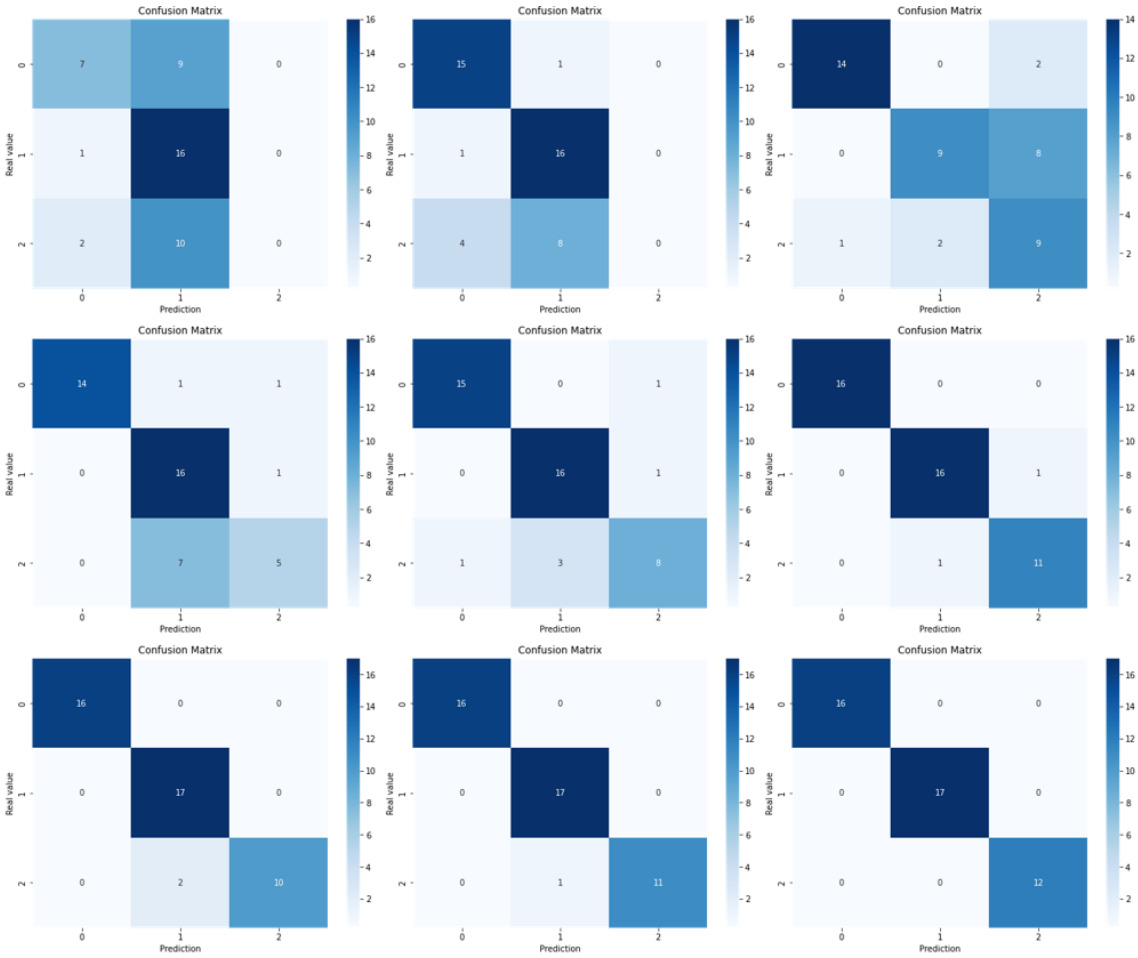


Figure 2: In this figure, the confusion matrices for each Regression Tree in ascendant order from left to right and from top to bottom.

An evolution of the confusion matrices are shown in figure 2. Notice that in both cases for number of layers 9 and 10, the confusion matrix is the same, as all predicted values are the actual ones. Also notice, that the first confusion matrices do not predict the possibility of having one of the species, this is a consequence of the data to train the model, as the best possible partitions fit better when considering features that do not differentiate between two of the species. However, when considering more partitions the number of good predicted values increase.

5.1 Results

Let now focus on how good is the model. As we can observe on the tree diagram on the annex, which has 10 partition levels, most of the end terms contain only one sample, giving a Gini entropy of almost 0, but also others that have not only one possible solution. This means that the actual accuracy of 100% is only an effect of the data considered to validate the model.

When considering 5 partition steps the RSS is when all the branches of the tree have approximately the same and low RSS. This result permits to assure that there is an over-fitting if

considering more partitions.

As the model is considering a small amount of data to be trained, will not perform as well when given more data as it is over-fitting the model. However, this diagram let us know also the previous partition steps also, so we can modified if needed so that new data fits better on the model.

6 conclusions

As we have seen our model predicts using 70% of the data-frame as training and a 30% for validation. The validation precision when considering this data, increases with the number of partitions the model have giving the best precision for 9 or 10 partitions. However, with more then 5 partitions there is an over-fitting. Noticing also that most of the features have been discarded because of Nan values, can possibly be affecting in how many layers are needed to have a good model.

In conclusion, the model predicts better data when increasing the number of partitions, but these may suppose an over-fitting. In order to improve the result, more data or not Nan data is needed.

References

[BFSO84] L. Breiman, J. Friedman, C.J. Stone, and R.A. Olshen. *Classification and Regression Trees*. Taylor & Francis, 1984.

[JNM63] John A. Sonquist James N. Morgan. Problems in the analysis of survey data, and a proposal. *American Statistical Association, JSTOR*, 58(302):415–434, 1963.

[PVG+11] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

7 Annex

7.1 kanga data-frame

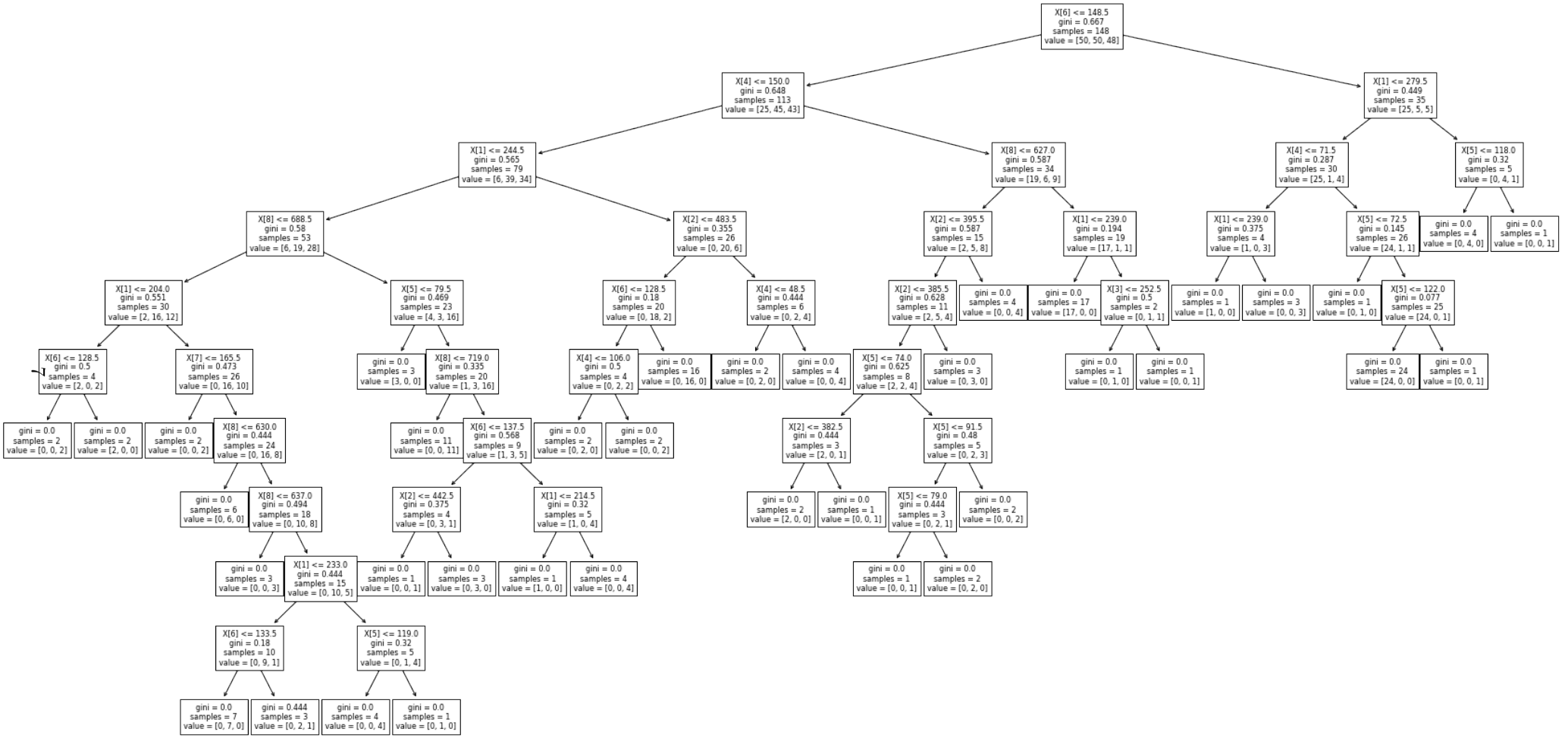
species	sex	basilar_length	occipitonasal_length	palate_length
giganteus	Male	1312.0	1445.0	882.0
giganteus	Male	1439.0	1503.0	985.0
giganteus	Male	1378.0	1464.0	934.0
giganteus	Male	1315.0	1367.0	895.0
giganteus	Male	1413.0	1500.0	969.0
...	...	...	...	...
fuliginosus	Female	1485.0	1500.0	1016.0
fuliginosus	Female	1468.0	1536.0	996.0
fuliginosus	Female	1510.0	1546.0	1043.0
fuliginosus	Female	1526.0	1512.0	1052.0
fuliginosus	Female	1570.0	1583.0	987.0

palate_width	nasal_length	nasal_width	squamosal_depth	lacrymal_width
NaN	609.0	241	180.0	394
230.0	629.0	222	150.0	416
NaN	620.0	233	135.0	403
230.0	564.0	207	158.0	394
NaN	645.0	247	161.0	426
...	...	...	...	...
277.0	552.0	205	203.0	454
264.0	667.0	222	190.0	431
264.0	656.0	218	197.0	423
281.0	625.0	250	201.0	470
285.0	646.0	244	198.0	482

zygomatic_width	orbital_width	rostral_width	occipital_depth	crest_width
782.0	249	227.0	531.0	153
824.0	233	248.0	632.0	141
778.0	244	240.0	575.0	144
801.0	224	242.0	568.0	116
823.0	241	252.0	607.0	120
...	...	...	...	...
919.0	225	278.0	676.0	122
951.0	217	305.0	650.0	178
891.0	190	270.0	651.0	78
934.0	236	289.0	680.0	145
984.0	253	291.0	699.0	188

foramina_length	mandible_length	mandible_width	mandible_depth	ramus_height
88	1086.0	131	179	591
100	1158.0	148	181	643
107	1131.0	116	169	610
79	1090.0	132	189	594
99	1175.0	131	197	654
...	...	...	...	...
74	1260.0	148	194	751
82	1287.0	141	199	736
87	1337.0	158	210	747
106	1334.0	153	211	739
103	1354.0	153	223	807

7.2 Regression tree


X[6] <= 148.5
gini = 0.667
samples = 148
value = [50, 50, 48]
X[4] <= 150.0
gini = 0.648
samples = 113
value = [25, 45, 43]
X[1] <= 279.5
gini = 0.449
samples = 35
value = [25, 5, 5]
X[1] <= 244.5
gini = 0.565
samples = 79
value = [6, 39, 34]
X[8] <= 627.0
gini = 0.587
samples = 34
value = [19, 6, 9]
X[4] <= 71.5
gini = 0.287
samples = 30
value = [25, 1, 4]
X[5] <= 118.0
gini = 0.32
samples = 5
value = [0, 4, 1]
X[8] <= 688.5
gini = 0.58
samples = 53
value = [6, 19, 28]
X[2] <= 483.5
gini = 0.355
samples = 26
value = [0, 20, 6]
X[2] <= 395.5
gini = 0.587
samples = 15
value = [2, 5, 8]
X[1] <= 239.0
gini = 0.194
samples = 19
value = [17, 1, 1]
X[1] <= 239.0
gini = 0.375
samples = 4
value = [1, 0, 3]
X[5] <= 72.5
gini = 0.145
samples = 26
value = [24, 1, 1]
gini = 0.0
samples = 4
value = [0, 4, 0]
gini = 0.0
samples = 1
value = [0, 0, 1]
X[1] <= 204.0
gini = 0.551
samples = 30
value = [2, 16, 12]
X[5] <= 79.5
gini = 0.469
samples = 23
value = [4, 3, 16]
X[6] <= 128.5
gini = 0.18
samples = 20
value = [0, 18, 2]
X[4] <= 48.5
gini = 0.444
samples = 6
value = [0, 2, 4]
X[2] <= 385.5
gini = 0.628
samples = 11
value = [2, 5, 4]
gini = 0.0
samples = 4
value = [0, 0, 4]
gini = 0.0
samples = 17
value = [17, 0, 0]
X[3] <= 252.5
gini = 0.5
samples = 2
value = [0, 1, 1]
gini = 0.0
samples = 1
value = [1, 0, 0]
gini = 0.0
samples = 3
value = [0, 0, 3]
gini = 0.0
samples = 1
value = [0, 1, 0]
X[5] <= 122.0
gini = 0.077
samples = 25
value = [24, 0, 1]
X[6] <= 128.5
gini = 0.5
samples = 4
value = [2, 0, 2]
X[7] <= 165.5
gini = 0.473
samples = 26
value = [0, 16, 10]
gini = 0.0
samples = 3
value = [3, 0, 0]
X[8] <= 719.0
gini = 0.335
samples = 20
value = [1, 3, 16]
X[4] <= 106.0
gini = 0.5
samples = 4
value = [0, 2, 2]
gini = 0.0
samples = 16
value = [0, 16, 0]
gini = 0.0
samples = 2
value = [0, 2, 0]
gini = 0.0
samples = 4
value = [0, 0, 4]
X[5] <= 74.0
gini = 0.625
samples = 8
value = [2, 2, 4]
gini = 0.0
samples = 3
value = [0, 3, 0]
gini = 0.0
samples = 1
value = [0, 1, 0]
gini = 0.0
samples = 1
value = [0, 0, 1]
gini = 0.0
samples = 24
value = [24, 0, 0]
gini = 0.0
samples = 1
value = [0, 0, 1]
gini = 0.0
samples = 2
value = [0, 0, 2]
gini = 0.0
samples = 2
value = [2, 0, 0]
X[8] <= 630.0
gini = 0.444
samples = 24
value = [0, 16, 8]
gini = 0.0
samples = 11
value = [0, 0, 11]
X[6] <= 137.5
gini = 0.568
samples = 9
value = [1, 3, 5]
gini = 0.0
samples = 2
value = [0, 2, 0]
gini = 0.0
samples = 2
value = [0, 0, 2]
X[2] <= 382.5
gini = 0.444
samples = 3
value = [2, 0, 1]
X[5] <= 91.5
gini = 0.48
samples = 5
value = [0, 2, 3]
gini = 0.0
samples = 2
value = [2, 0, 0]
gini = 0.0
samples = 1
value = [0, 0, 1]
X[5] <= 79.0
gini = 0.444
samples = 3
value = [0, 2, 1]
gini = 0.0
samples = 2
value = [0, 0, 2]
gini = 0.0
samples = 1
value = [0, 0, 1]
gini = 0.0
samples = 2
value = [0, 2, 0]
gini = 0.0
samples = 6
value = [0, 6, 0]
X[8] <= 637.0
gini = 0.494
samples = 18
value = [0, 10, 8]
X[2] <= 442.5
gini = 0.375
samples = 4
value = [0, 3, 1]
X[1] <= 214.5
gini = 0.32
samples = 5
value = [1, 0, 4]
gini = 0.0
samples = 3
value = [0, 0, 3]
X[1] <= 233.0
gini = 0.444
samples = 15
value = [0, 10, 5]
gini = 0.0
samples = 1
value = [0, 0, 1]
gini = 0.0
samples = 3
value = [0, 3, 0]
gini = 0.0
samples = 1
value = [1, 0, 0]
gini = 0.0
samples = 4
value = [0, 0, 4]
X[6] <= 133.5
gini = 0.18
samples = 10
value = [0, 9, 1]
X[5] <= 119.0
gini = 0.32
samples = 5
value = [0, 1, 4]
gini = 0.0
samples = 7
value = [0, 7, 0]
gini = 0.444
samples = 3
value = [0, 2, 1]
gini = 0.0
samples = 4
value = [0, 0, 4]
gini = 0.0
samples = 1
value = [0, 1, 0]