Introduction to Data Science

מטלה 4

סמסטר אביב 2021

מרצה:

ד"ר גייל גלבוע פרידמן

מתרגל:

עידן לופו

אפי פקאני

## **Ex4 - Decision Trees Verbal Answers**

**Part 1**

4. Before we start the preprocessing phase we can tell a few things about our data according our discreptional statistics:

By looking at columns ‘**harsh\_braking\_ratio’, ‘near\_miss\_bicycle\_ratio’, ‘near\_miss\_pedestrian\_ratio’** we understand that most of the values of those features are zero or very close to zero, this can imply those features might be less effective for our model. Moreover, ‘**harsh\_braking\_ratio’** has only 5162 observations opposed to most of the features which have 9,266 observations. That fact combined with the high amount of zero values implies we probably cannot gain interesting information from the‘**harsh\_braking\_ratio’** feature and we better drop it.

5. The proportions of dangerous sections is 24% vs. 76% of safe ones according to the “collision\_likelihood” column.

6. a. We can see that the column with the most missing (NAN) values was "harsh\_breaking\_ratio" with 4104 blank spaces.

b. The most secure way to handle this type of NAN values is to place either the median or the mean/average value of the column. By operating in this methodology in order to fill a relatively small amount of missing observations (5%-7% tops) we can rest assured that we haven't affected our values and that didn’t create a significant statistical deviation or a bias in our data. Note that the average values of “near\_miss\_pedestrian” & “near\_miss\_bicycle” are in a biased state upwords, 75% of the instances contain small values and close to zero, thus it’s better to fill them in with the median.

On the other hand, we can see that the “harsh\_braking\_count” column is half empty and contains NANs in almost 50% of its cells, together with many zeros, making it the least informative column, therefore we should decide to remove it.

**Part 3**

14. Accuracy – Tree Depth

Chart, histogram

Description automatically generated

15) a)We can note that we get an up trend in the training accuracy graph respectively along to the maximum tree depth. The deeper the tree goes the greater the accuracy of the train-model becomes.

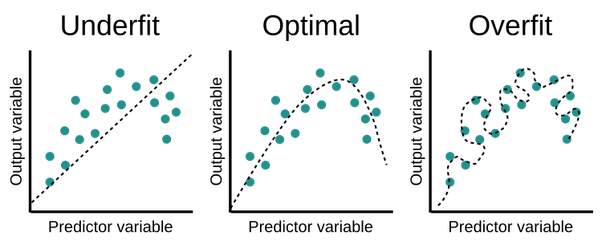
The tendency of the graph to grow as a function of the depth of the tree happens due to the improvement in the accuracy in each division of the tree.

In each split our model "learns" how to better predict the following values. This means that the deeper our tree will go, we will get a more accurate prediction for the train value. From our model we can see that after 20 splits of the tree we will reach the final accuracy of 1 which can imply that our model has reached overfitting.

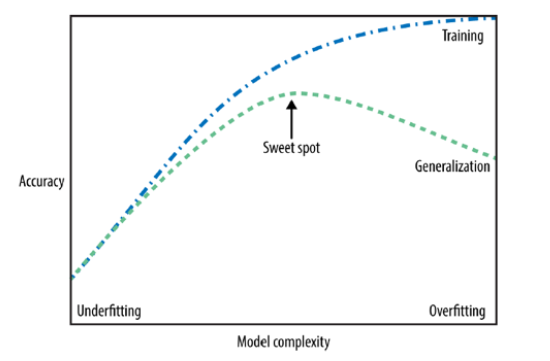
b) We can see from the graph above that the deeper the tree goes until the value of ~6 the accuracy of the model on the test-set becomes greater and from 6 and beyond the accuracy diminishes. We can infer that the max point of our test-accuracy should be around 6.

This occurs since values greater than 6 can cause an over-fitting of our model to our train data.

To avoid this from happening we should consider the trade-off between the accuracy of our prediction and its margin of error against our model from being too specific.



c) As a result we should recommend the use of a 6/7 maximial tree depth .   
Using a 6/7 max tree depth will assure us the most optimal accuracy for our testing measurements with the minimal validation loss thus preventing a case of the model overfitting our data.



19) Gini index is the criteria for calculating information gain.   
Decision tree algorithms use information gain to split a node.   
Gini measures the impurity of a node, so the algorithm would calculate the gini index for each feature, and choose them one by one in an increasing order.

20) In general, we will prefer to have a lower value for "Gini" in order to minimize the inequality of our data.

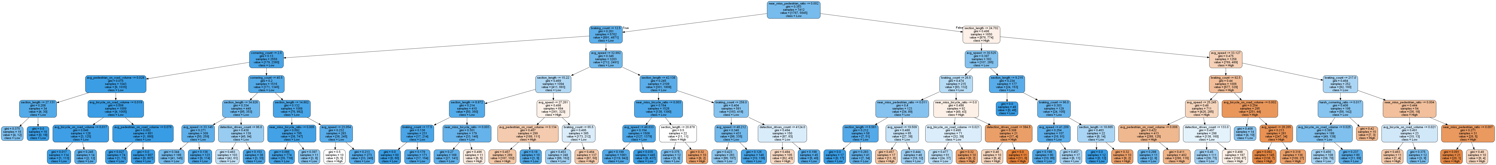
In our example (with random values for each feature) we can see that the "gini" for the "section\_length" parameter is 0.3361 and that the "gini" for "near\_miss\_pedestrian\_ratio" parameter is 0.3143. Therefore for this example the algorithm will choose the ‘near\_miss\_pedestrian\_ratio’ feature as its first split.

21) Feature importance graph:

Chart

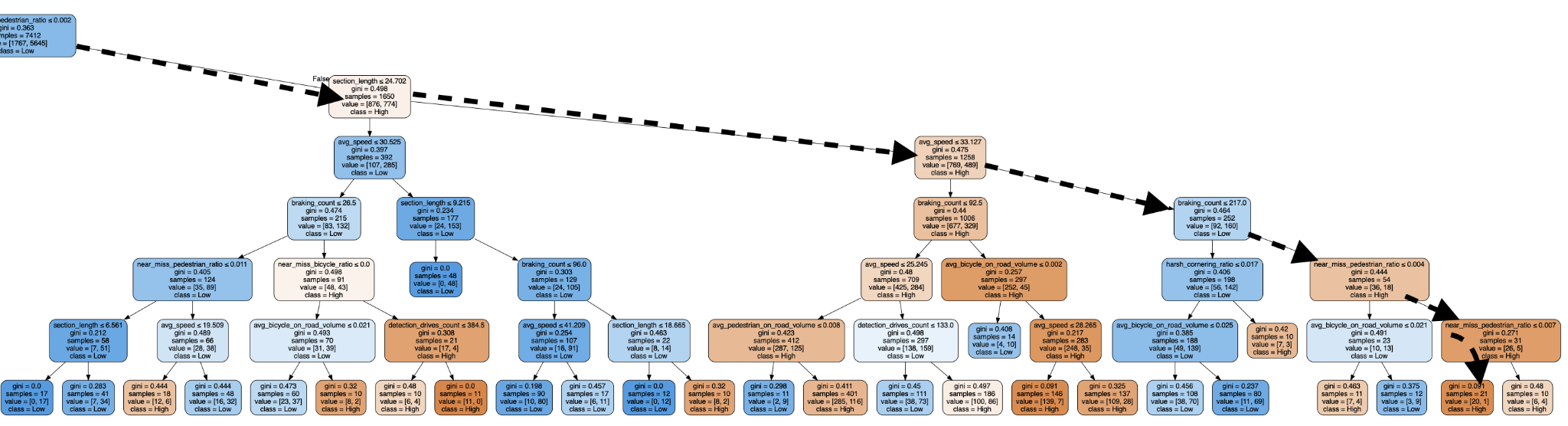
Description automatically generated

22) Graphing the tree:



23) When we look at the decision tree we’ve got, we can easily note that as the probability to predict correctly increases, the node’s color gets darker. The darker the blue color gets, the probability to predict ‘High’ increases. Same thing occurs when we check the orange color and ‘Low’ predictions.   
We can spot that we have a few nodes with a gini index of 0.   
this means that each that will get there will be classified as ‘High’ (gini = 0 means the instances are purely classified). By looking at the tree, we can tell that on the left side most of the nodes are blue. So, we can assume that sections classified as ‘High’ have ‘near\_miss\_pedestrian\_ratio’ less than 0.002, and the left side of this subtree is even darker (blue). Meaning that these sections have ‘braking\_count’ less than 12.5.

1st path:



1. Near\_miss\_pedestrain\_ration <=0.002
2. Section\_length < = 24.702
3. Avg\_speed <=33.127
4. Braking\_count<=217
5. Near\_miss\_pedestrain\_ration <= 0.004
6. Near\_miss\_pedestrain\_ration <= 0.007

2nd path:

1. Near\_miss\_pedestrain\_ration <=0.002
2. Section\_length < = 24.702
3. Avg\_speed <=30.525
4. Braking\_count<=26.5
5. Near\_miss\_bicycle\_ratio <= 0.0
6. Detection\_drives\_count <= 384.5

