Data Mining

Mohammed Almulla

College of I.T

Decision Tree Induction for Classification

In this section, I will talk about decision tree for classification. Before going through decision tree induction, I will talk about classification process. In classification process we must first divide our data into training set that will be used to make our initial model based on it and it will be divided into training records and validation records. Validation set or records and it will be applied with the initial model to get a refined model. At the end we will use test set to be applied it into the refined model so then we get our final model with measured accuracy. Figure 54 shows process of classification.

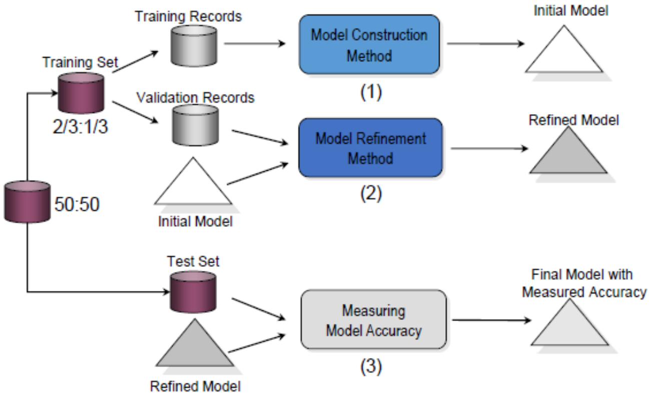


Figure 54: Process of Classification

There are some influential factors to be sure we got a good model. The first one is classification accuracy that is estimated accuracy during development stage vs. actual accuracy during practical use. Secondly, classification performance that is time taken for model construction and time taken for classification, also we need to make sure that if the model takes more time or a lot of memory. Finally, comprehensibility of the model that is ease of interpreting decisions by the classification model, or can we explain the results of the model.

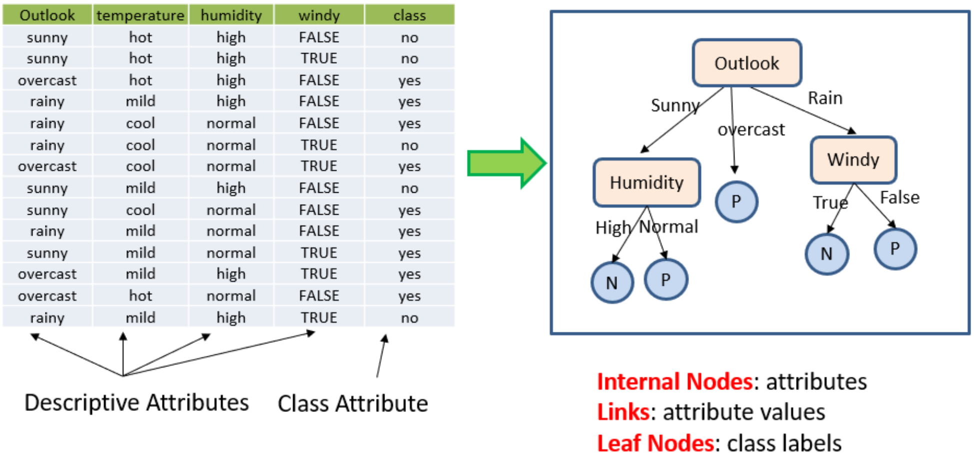
Now let us talk about decision tree induction. To build a decision tree we have principles of tree construction they are as follows (Figure 55 shows a decision tree):

1. If the training set is empty, create a leaf node and label it as NULL.
2. If all examples are of the same class, create a leaf node labelled by the class.
3. If examples in the training set are of different classes:
   a. Determine which attribute should be selected as the root of the current tree.

b. Partition the input examples into subsets according to the values of the selected root attribute.

c. Construct a decision tree recursively for each subset.

d. Connect the roots for the sub-trees to the root of the whole tree via labelled links.



Outlook	temperature	humidity	windy	class
sunny	hot	high	FALSE	no
sunny	hot	high	TRUE	no
overcast	hot	high	FALSE	yes
rainy	mild	high	FALSE	yes
rainy	cool	normal	FALSE	yes
rainy	cool	normal	TRUE	no
overcast	cool	normal	TRUE	yes
sunny	mild	high	FALSE	no
sunny	cool	normal	FALSE	yes
rainy	mild	normal	FALSE	yes
sunny	mild	normal	TRUE	yes
overcast	mild	high	TRUE	yes
overcast	hot	normal	FALSE	yes
rainy	mild	high	TRUE	no

Internal Nodes: attributes
Links: attribute values
Leaf Nodes: class labels

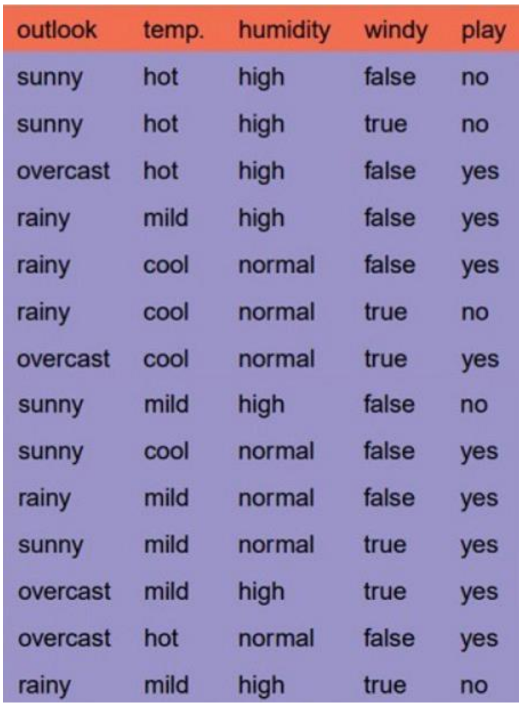
Figure 55: Decision Tree

There are multiple solutions to build a decision tree and many ways, and we need to choose the simpler solution, and this known as Occam's Razor law. As stated in William's words, Occam's Razor that is "Plurality must never be posited without necessity". Moreover, to choose the best attribute is to choose the attribute that produces the purest class nodes. Also, to know the purest attribute we need to calculate the information gain. Information gain is a criterion than increases with the average purity of the subsets, that means we need to choose the attribute that gives greatest information gain. Also, information gain is calculated by comparing the entropy of the dataset before and after a transformation. It measures the reduction in entropy or surprise by splitting a dataset according to a given value of a random variable. An entropy or surprise or also known as measure of uncertainty is the average of the information content that is the information you gained. It used to calculate the homogeneity of a sample. The algorithm that uses entropy and information gain and we will learn how it works it is called ID3 algorithm. The ID3 algorithm was invented by Ross Quinlan.

ID3 Algorithm

I will use a simple example and famous data set to work it on ID3 algorithm. I will explain step by step on how to build a decision tree using ID3 algorithm. So, let us start!

Our data set is shown in figure 56.



outlook	temp.	humidity	windy	play
sunny	hot	high	false	no
sunny	hot	high	true	no
overcast	hot	high	false	yes
rainy	mild	high	false	yes
rainy	cool	normal	false	yes
rainy	cool	normal	true	no
overcast	cool	normal	true	yes
sunny	mild	high	false	no
sunny	cool	normal	false	yes
rainy	mild	normal	false	yes
sunny	mild	normal	true	yes
overcast	mild	high	true	yes
overcast	hot	normal	false	yes
rainy	mild	high	true	no

Figure 56: Data set

1. Firstly, we must choose attribute to be the root of our tree. We must determine the attribute that best classifies the training data. So, to choose the best attribute we must choose the one that give us the highest information gain value. To define information gain, we will first define the entropy that characterizes the purity or impurity of the training examples.

$$Entropy = H(S) = \sum_{x \in X} -p(x) \log_2 p(x)$$

Where S is the current data set (It changes at each step of ID3 in case of splitting on an attribute), X is the set of classes in S, and $p(x)$ is the proportion of the number of elements in class x to the number of elements in set S. When $H(S) = 0$ means that all elements in S are of the same class.

$$Information\ Gain(S, A) = H(S) - \sum_{V \in Values(A)} \frac{|S_V|}{|S|} \times H(S_V)$$

Where V is possible values of A, S is set of examples, and S_V is subset where $X_A = V$. Also, $\frac{|S_V|}{|S|}$ is the probability of positive or negative classes.

2. As we said first, we must calculate the entropy for the whole data set by counting yeses and no's in the play attribute. Then, apply the entropy law to it.

$$H(S) = -\frac{9}{14} \times \log_2 \frac{9}{14} - \frac{5}{14} \times \log_2 \frac{5}{14} = 0.94$$

3. For every attribute we must calculate entropy for all categorical values, take average information entropy for the current attribute, and calculate gain for the current attribute. So, for every attribute calculate the entropy and information gain.

$$E(Outlook = sunny) = -\frac{2}{5}\log_2\frac{2}{5} - \frac{3}{5}\log_2\frac{3}{5} = 0.971$$

$$E(Outlook = overcast) = -\frac{4}{4}\log_2\frac{4}{4} - \frac{0}{4}\log_2\frac{0}{4} = 0$$

$$E(Outlook = rainy) = -\frac{3}{5}\log_2\frac{3}{5} - \frac{2}{5}\log_2\frac{2}{5} = 0.971$$

The average entropy information for outlook attribute will be as follows.

$$I(Outlook) = \frac{5}{14} \times 0.971 + \frac{4}{14} \times 0 + \frac{5}{14} \times 0.971 = 0.693$$

$$Information\ Gain(Outlook) = 0.94 - 0.693 = 0.247$$

We will do the same for Windy attribute.

$$E(windy = false) = -\frac{6}{8}\log_2\frac{6}{8} - \frac{2}{8}\log_2\frac{2}{8} = 0.811$$

$$E(windy = true) = -\frac{3}{6}\log_2\frac{3}{6} - \frac{3}{6}\log_2\frac{3}{6} = 1$$

$$I(Windy) = \frac{8}{14} \times 0.811 + \frac{6}{14} \times 1 = 0.892$$

$$Information\ Gain(Windy) = 0.94 - 0.892 = 0.048$$

Even for humidity attribute as shown below.

$$E(humidity = high) = -\frac{4}{7}\log_2\frac{4}{7} - \frac{3}{7}\log_2\frac{3}{7} = 0.985$$

$$E(humidity = normal) = -\frac{1}{7}\log_2\frac{1}{7} - \frac{6}{7}\log_2\frac{6}{7} = 0.591$$

$$I(humidity) = \frac{7}{14} \times 0.985 + \frac{7}{14} \times 0.591 = 0.788$$

$$Information\ Gain(humidity) = 0.94 - 0.788 = 0.152$$

Also, do it for temp attribute.

$$E(temp = hot) = -\frac{2}{4}\log_2\frac{2}{4} - \frac{2}{4}\log_2\frac{2}{4} = 1$$

$$E(temp = mild) = -\frac{2}{6}\log_2\frac{2}{6} - \frac{4}{6}\log_2\frac{4}{6} = 0.918$$

$$E(temp = cool) = -\frac{1}{4}\log_2\frac{1}{4} - \frac{3}{4}\log_2\frac{3}{4} = 0.811$$

$$I(temp) = \frac{4}{14} \times 1 + \frac{6}{14} \times 0.918 + \frac{4}{14} \times 0.811 = 0.911$$

$$Information\ Gain(temp) = 0.94 - 0.911 = 0.029$$

Figure 57 shows the trees for all four attributes.

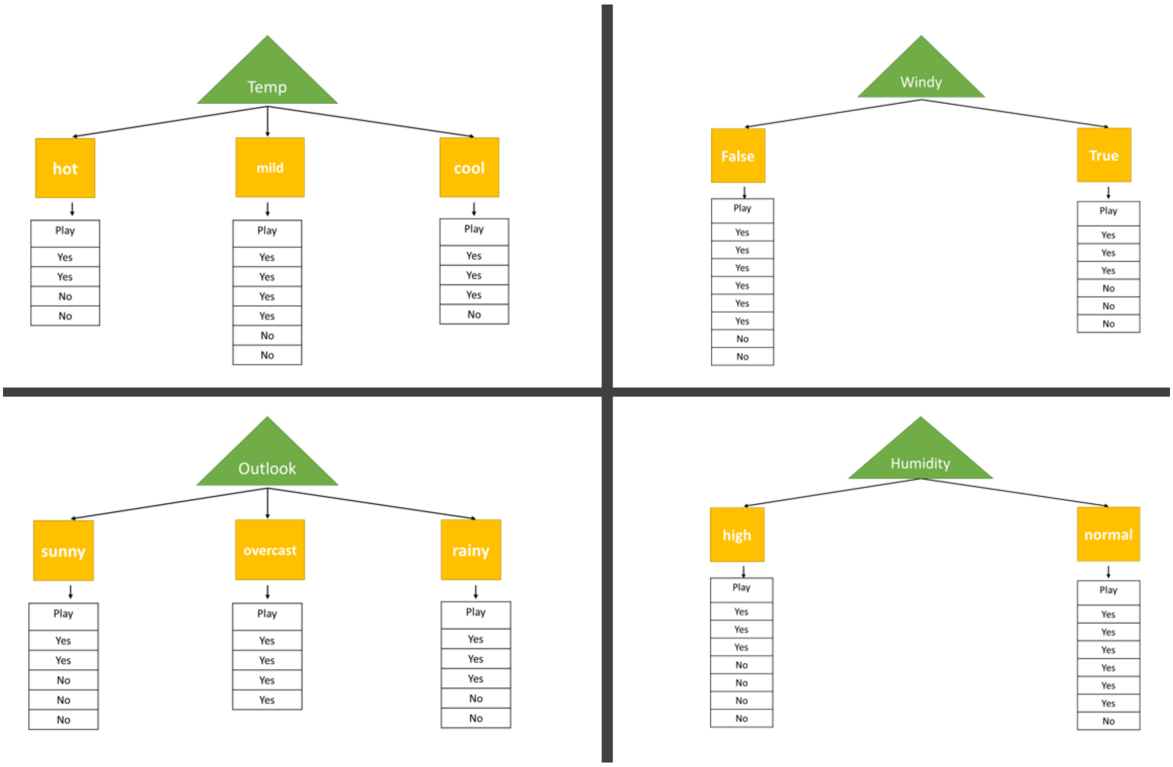


Figure 57: trees for the four attributes

4. Now, we will choose the highest information gain value that is outlook attribute, and we will make it to be the root of our tree. Then, we must calculate the entropy and information gain again to define the nodes for the branches.

$$H(S_{sunny}) = -\frac{3}{5}\log_2\frac{3}{5} - \frac{2}{5}\log_2\frac{2}{5} = 0.971$$

$$E(humidity = high) = -\frac{3}{3}\log_2\frac{3}{3} - \frac{0}{3}\log_2\frac{0}{3} = 0$$

$$E(humidity = normal) = -\frac{0}{2}\log_2\frac{0}{2} - \frac{2}{2}\log_2\frac{2}{2} = 0$$

$$I(humidity) = \frac{3}{5} \times 0 + \frac{2}{5} \times 0 = 0$$

$$Information\ Gain(S_{sunny}, humidity) = 0.971 - 0 = 0.971$$

Also, for windy attribute.

$$E(windy = true) = -\frac{1}{2}\log_2\frac{1}{2} - \frac{1}{2}\log_2\frac{1}{2} = 1$$

$$E(windy = false) = -\frac{2}{3}\log_2\frac{2}{3} - \frac{1}{3}\log_2\frac{1}{3} = 0.918$$

$$I(windy) = \frac{2}{5} \times 1 + \frac{3}{5} \times 0.918 = 0.9508$$

$$Information\ Gain(S_{sunny}, windy) = 0.971 - 0.9508 = 0.020$$

Even we will calculate for temp attribute based on outlook = sunny and it information gain will be equal to 0.571. So, we will choose humidity to be at the branch after sunny value of outlook attribute. As shown in figure 58.

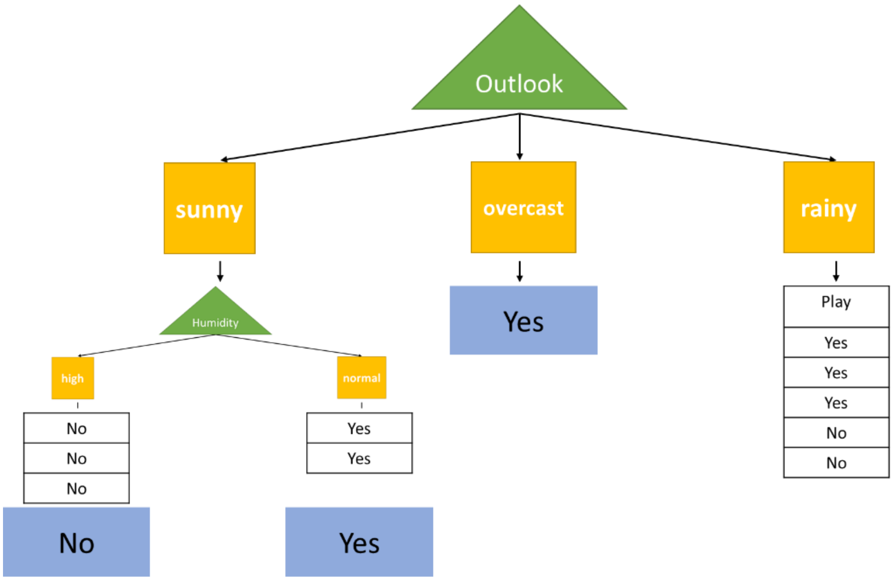


Figure 58: The tree after adding humidity attribute

5. Finally, we will repeat for the rest of attributes the same way. We will calculate entropy based on outlook = rainy. Then, calculate entropy and information gain for windy and temp attributes. And, from calculating it windy will give the highest information gain value, so it will be the next branch under rain value for outlook attribute. As shown in figure 59 our last decision tree.

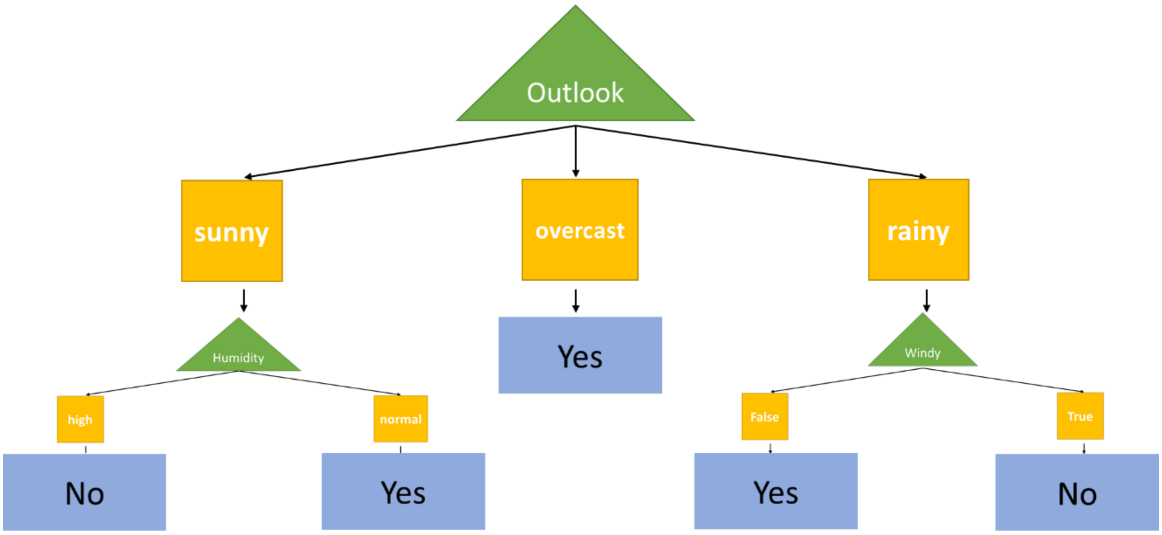


Figure 59: Last decision tree

There are many other algorithms for building a decision tree such as C4.5 and C5 that they are considered to be ID3 family due to the similarities between them and ID3. Another algorithm for building decision trees is CART that use GINI index, and CHAID that use chi-square statistics. The algorithms mainly differ in attribute selection measures adopted. Studies show that there are only marginal differences among the attribute selection measures with respect to model accuracy.

Overfitting

Any algorithms that build a classification model based on a finite set of training examples tend to have the problem of model overfitting. And overfitting is the model induced from the training examples fits the training examples too well. It reflects the specific features of the examples than features of actual data at large. It causes of the problem include presence of noise data, lack of representative training examples, ML technique choice, etc. Figure 60 shows overfitting and underfitting.

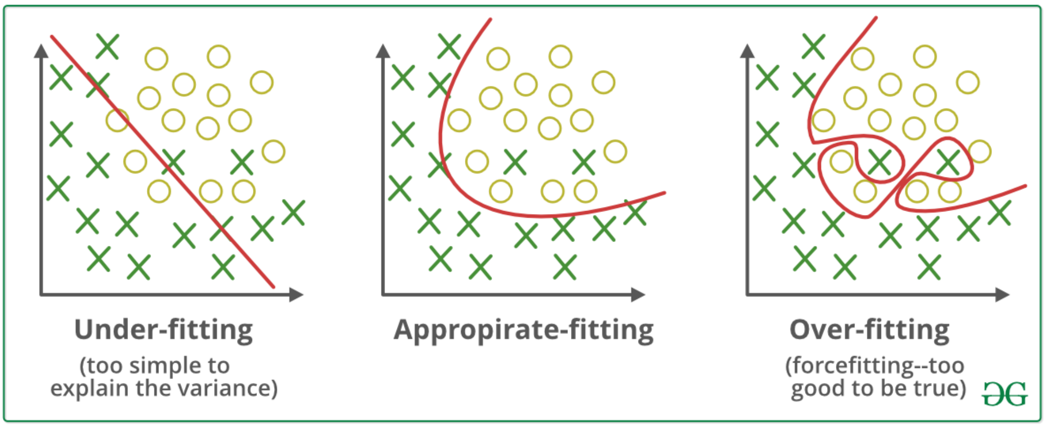

Under-fitting
(too simple to explain the variance)
Appropirate-fitting
Over-fitting
(forcefitting--too good to be true)

Figure 60: overfitting and underfitting

There are two approaches to tackle overfitting problem:

1. Early stopping rule, also known as pre-pruning, that is tree construction is halted at some stage. Stop growing a branch when information becomes unreliable.
2. Tree pruning, also known as post-pruning, that is to “cut back” certain subtrees and replace them with leaf nodes, making the tree smaller and more robust. Take a fully-grown decision tree and discard unreliable parts.
3. Use an independently sampled validation set to assist tree pruning.

The pruning methods are as follows:

1. Reduced error pruning.
2. Cost complexity pruning.
3. Pessimistic pruning and many others.

Figure 61 shows an example on pruning.

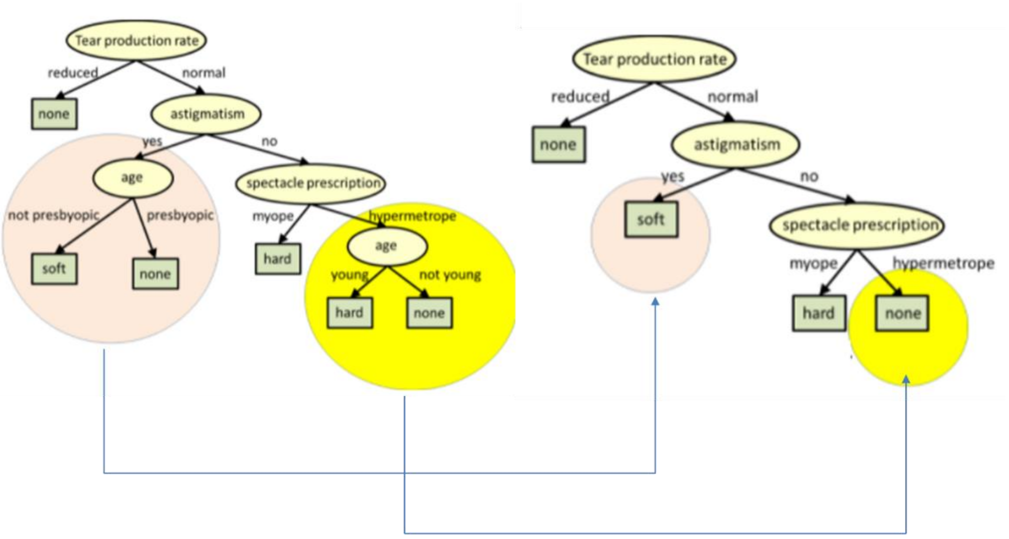
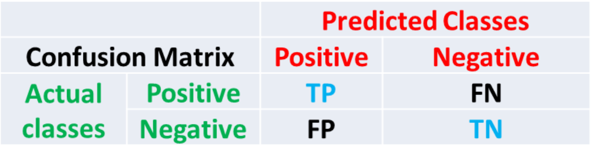


Figure 61: An example on pruning

Evaluating Tree Accuracy

Evaluation is performed on the validation set. And accuracy is measured in terms of error rate. Details of errors are shown in a confusion matrix. A confusion matrix is a specific table layout that allows visualization of the performance of an algorithm. Each row represents the instances in an actual class while each column represents the instances in a predicted class, or vice versa. Figure 62 shows a confusion matrix.



		Predicted Classes	
Confusion Matrix		Positive	Negative
Actual classes	Positive	TP	FN
	Negative	FP	TN

Figure 62: confusion matrix

Let me explain the terminologies above:

- TP or True Positive: that is when a model predicts the positive true value.
- TN or True Negative: that is when a model predicts the negative true value.
- FP or False Positive: that is when a model predicts yes or true while the real value is no or false.
- FN or False Negative: that is when a model predicts no or false while the real value is yes or true.
- Accuracy: is the closeness of the measurements to a specific value.

$$Accuracy = \frac{Correct\ predictions}{All\ predictions} = \frac{TP + TN}{TP + TN + FP + FN}$$

$$Error\ rate = \frac{Mismatch\ predictions}{All\ predictions} = 1 - Accuracy$$

- Recall or Sensitivity is the number of correctly classified positive examples divided by the total number of actual positive examples in the test set.

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

- Precision is the number of correctly classified positive examples divided by the total number of examples that are classified as positive.

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

Figure 63 shows recall and precision.

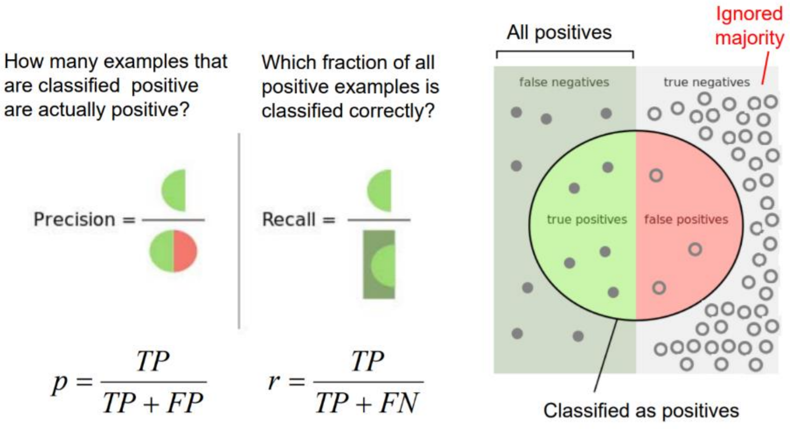


Figure 63: Recall and Precision

- F1-measure is the harmonic mean of precision and recall. And the harmonic mean of two numbers tends to be closer to the smaller of the two. Thus, for the F1-measure to be large, both precision and recall must be large. Figure 64 shows the difference between harmonic mean and arithmetic mean.

$$F1 - measure = \frac{2 \times Precision \times Recall}{Precision + Recall} = \frac{2 \times TP}{2 \times TP + FP + FN}$$

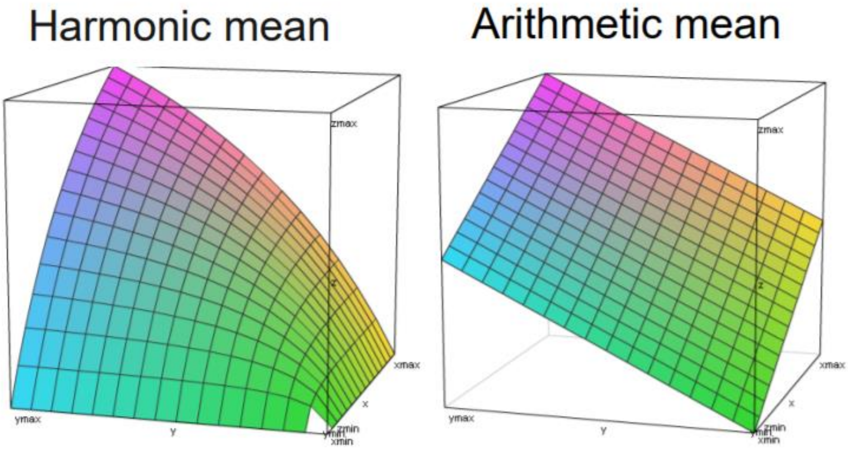


Figure 64: The difference between harmonic and arithmetic mean

There are many evaluation methods that will make sure we obtain a reliable estimate of the generalization performance. They are as follows:

1. Holdout method: It reserves a certain amount for testing and uses the remainder for training, usually one third for testing, the rest for training. The problem behind it is that the samples might not be representative e.g., class might be missing in the test data. Advanced version of holdout uses stratification that ensures that each class is represented with approximately equal proportions in both subsets.
2. Random Subsampling: use the holdout method several times and take the average of the accuracy.
3. Bootstrap: use sampling with replacement. Training examples are also test examples.
4. Cross Validation: the data set is divided into k equal-size partitions. For each round of decision tree induction, one partition is used for testing and the rest used for training. After k rounds, the average error rate is used.
5. Leave-One-Out: a particular form of cross-validation where fold size is equal to 1.

Let us see now the strengths of decision trees that they are capable of generating understandable rules, Efficiency in classifying unseen data, and Ability to indicate the most important attribute for classification. While the weaknesses are Error rate increases as the training set contains a small number of instances of a large variety of classes, The computationally expensive to build, and Single attribute splitting results in class boxes in rectangular shapes.

References & Useful Resources:

- CSBP 320: Data Mining, UAE University
- Textbook: Hongbo Du.*Data Mining Techniques and Applications: an introduction*, 1st Edition
- https://en.wikipedia.org/wiki/ID3_algorithm
- https://medium.com/deep-math-machine-learning-ai/chapter-4-decision-trees-algorithms-b93975f7a1f1
- https://paginas.fe.up.pt/~ec/files_1112/week_08-DecisionTrees.pdf
- https://www.unimannheim.de/media/Einrichtungen/dws/Files_Teaching/Data_Mining/FSS2020/DM02-Classification-2-FSS2020.pdf