Decision Trees

ML Instruction Team, Fall 2022

CE Department
Sharif University of Technology

Supervised Learning

- Supervised Learning is the task of learning a mapping function from the input space to the output space using a training set comprising examples of input vectors along with their corresponding target vectors.

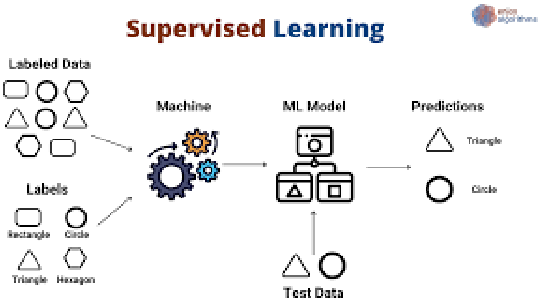


Figure: Supervised learning model to predict name of shape using it's picture, Source

Digit recognition example

- Digit Recognition is an example of Supervised learning in which the aim is to assign each input image to it’s corresponding digit.

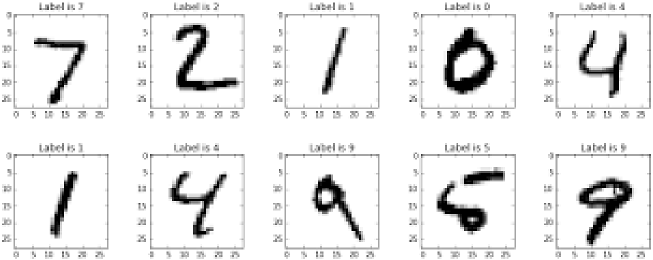


Figure: MNIST handwritten digit classification dataset consisting of handwritten digits and their corresponding labels, Source

Classification

- Machine learning tasks such as Digit Recognition are classification tasks, the goal in classification is to take and input vector x and assign it one of the K discrete classes $\mathcal{C}_k$ where $k = 1, 2, \ldots, K$.

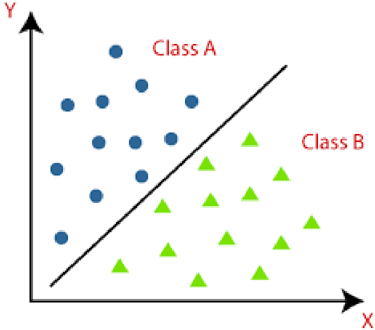


Figure: Classifying points in the planes into two classes using linear separator, Source

Spam filter

- Spam filter is an example of machine learning classification task where given an email we want to classify it into one of the categories ”spam” or ”ham”.

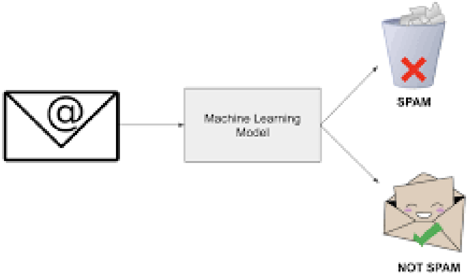


Figure: Classifying points in the planes into two classes using linear separator, Source

Classification algorithms

- Some of number of classification algorithms in machine learning include:
  - Logistic Regression
  - K-Nearest Neighbours
  - Support Vector Machines
  - Naïve Bayes
  - Decision Tree Classification

Regression

- If the desired output consists of one or more continuous variables, then the task is called regression.

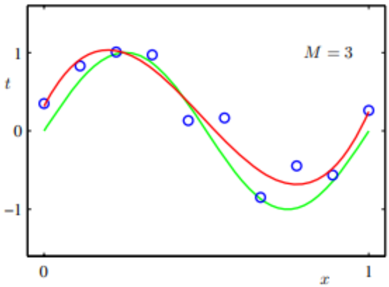


Figure: fitting a polynomial curve of degree 3 to points generated by a sinusoidal curve with gaussian noise

Linear regression

- Linear regression is an example of regression task where we want to find the best line fit to the data.

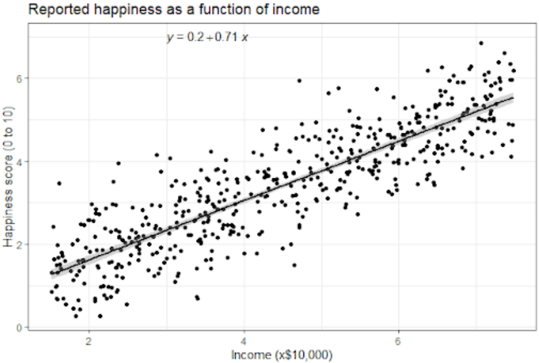


Figure: Linear fit for reported happiness in terms of income the more income usually causes more happiness, Source

Regression algorithms

- Some of number of regression algorithms in machine learning include:
  - Simple Linear Regression
  - Support Vector Regression
  - Decision Tree Regression

Decision Tree

- Decision Tree is a classification model which uses Series of yes/no questions after which the class label is inferred.

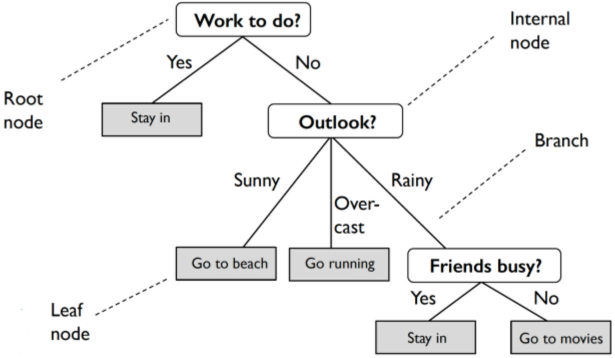


Figure: Decision tree for task determination

Tree representation

- All decision trees contain a single **root** node, the node from which decisions are started. The root node contains all the training examples.
- Each non-leaf node including the root node comes with some decision inside and each **branch** contains the answer to that question. Training data is divided according to the answer it responds to the yes/no question.

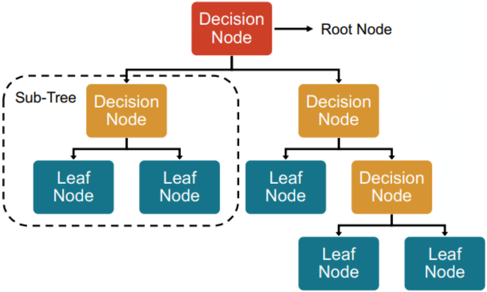


Figure: Notations for decision tree

Decision Tree for classification

- First we will consider Decision tree for classification where each attribute is a binary label. The tree will be then binary, each node either is a leaf or has exactly two branches.

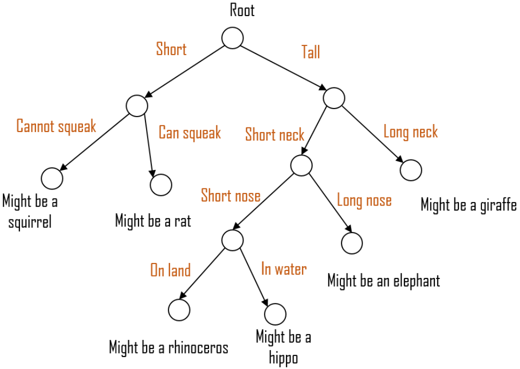


Figure: Animal classification using binary attributes, Source

Decision in the leaf nodes

- Assume using some algorithm we have built our decision tree and in each leaf node there are some training data according to which we will develop some algorithm for classification in that node.
- We will choose the simplest possible classifier for each leaf-node, a classifier that outputs always a single number although this number can be different in different leaf-nodes.

Accuracy as a metric

- The accuracy score for a classification task is defined as the portion of labels classified correctly.
- Classifier in each leaf node will be chosen in a way that maximizes the accuracy score.
- It’s easy to see that this leads to a classifier predicting the majority class in the leaf-node.

Finding the best Tree

- There are many possible decision trees for a dataset depending on the split it uses on each step.
- We prefer smaller decision trees where only few decision are considered when inferring the target value for the sample.
- We will use a greedy approach where in each step we find the ”best” attribute to split upon and then split each new node created recursively.
- We need the notion of Information gain in order to justify the meaning of ”best” split.

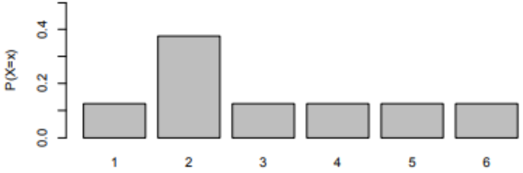
Entropy

- Entropy
  for some discrete random variable X taking values in some discrete space $\mathcal{X}$ is defined as:

$$H(X) := -\mathbb{E}[\mathit{log}_2(P(X=x))] = -\sum_{x\in\mathcal{X}} P(X=x)\,\mathit{log}_2(P(X=x))$$

Entropy example

- Entropy of a random variable following the distribution below is calculated as:



$$-0.4\,\mathit{Log}(0.4) - 5(0.12\,\mathit{Log}(0.12)) = 2.36$$

Entropy as measure of randomness

- More entropy indicates more randomness i.e not much predictable outcome. Low entropy means highly predictable.

$$\begin{bmatrix}1\\1\\1\\1\\1\\1\\1\\1\end{bmatrix}$$

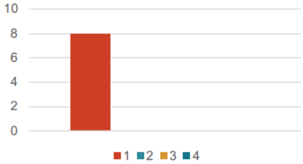


Figure: Low entropy means “very predictable”

$$\begin{bmatrix}1\\2\\4\\3\\1\\2\\1\\3\end{bmatrix}$$

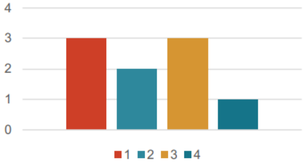


Figure: High entropy means “very random”

Entropy as measure of randomness

- More entropy means more randomness and hence more purity and less entropy means less purity.

$$\begin{bmatrix}1\\1\\1\\1\\1\\1\\1\\1\end{bmatrix}$$

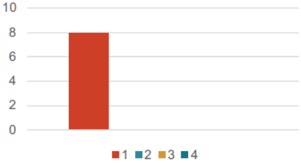


Figure: Low entropy means “high purity ”

$$\begin{bmatrix}1\\2\\4\\3\\1\\2\\1\\3\end{bmatrix}$$

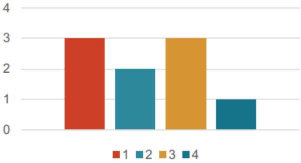


Figure: High entropy means “low purity”

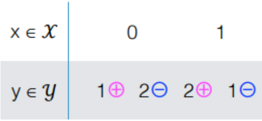
Conditional Entropy

- The conditional entropy of some random variable Y conditioned on some other random variable X each taking values in some discrete space $\mathcal{Y}, \mathcal{X}$ respectively is defined by:

$$H(Y|X) = -\sum_{x\in\mathcal{X}} P(X=x) \sum_{y\in\mathcal{Y}} P(Y=y|X=x)\, \mathit{log}_2(P(Y=y|X=x))$$

Conditional Entropy example

- Assume X, Y are both binary the condition entropy of distribution below will be calculated as:



$x \in \mathcal{X}$	0	1
$y \in \mathcal{Y}$	1⊕ 2⊖	2⊕ 1⊖

$$-\frac{3}{6}(0.33\,\mathit{Log}(0.33) + 0.67\,\mathit{Log}(0.67)) - \frac{3}{6}(0.67\,\mathit{Log}(0.67) + 0.33\,\mathit{Log}(0.33)) = 0.91$$

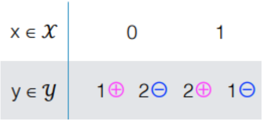
Information Gain

- Information Gain(aka mutual information) for some discrete random variables X, Y is defined as:

$$IG(X,Y) = H(X) - H(X|Y)$$

Information Gain example

- Information gain for joint distribution below is computed as:



$x \in \mathcal{X}$	0	1
$y \in \mathcal{Y}$	1⊕ 2⊖	2⊕ 1⊖

$$H(Y) - H(Y|X) = -2(\frac{3}{6}\, log(\frac{3}{6})) - 0.91 = 1 - 0.91 = 0.09$$

Some properties of information gain

- Information gain is always non-negative and equality occurs if and only if X, Y are independent.
- Information gain is symmetric $IG(X, Y) = IG(Y, X)$.
- It quantifies the amount of information obtained about random variable X by observing the random variable Y.
- Information gain for continuous random variables is zero differential entropy is defined instead.

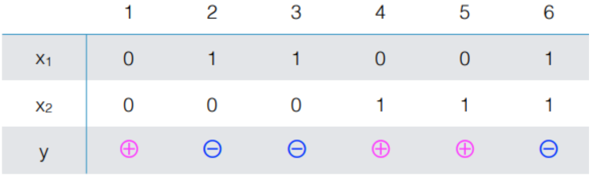
Finding the best split

- Let X be a random variable from the space of all samples to their corresponding target label and let Y_k denote a random variable from the space of all samples to their corresponding k'th feature. We will choose k in such a way that $IG(X, Y_k)$ is maximized. In other words knowing the value of k'th feature will tell us the most about the target label.

  index of the feature to split $= argmax_k IG(X, Y_K)$

Finding the best split example

- Assume we have two binary features X_1, X_2 and we want to decide which one to use for the first split:



	1	2	3	4	5	6
x_1	0	1	1	0	0	1
x_2	0	0	0	1	1	1
y	⊕	⊖	⊖	⊕	⊕	⊖

- $IG(Y, X_1) = 1$, $IG(Y, X_2) = 0.09$ so we will use X_1 feature for the first split.

Time complexity

- Assuming we have n samples in the training set and the tree is balanced. The time complexity for building a decision tree is equal to $m.n^2 \; \mathit{Log}(n)$

Gini impurity

- Instead of entropy one can use Gini impurity to decide between splits Gini impurity for a discrete random variable X taking values in discrete space $\mathcal{X}$ is equal to:

$$1 - \sum_{x \in \mathcal{X}} P(X = x)^2$$

Why Growing Decision Trees via Entropy or Gini Impurity instead of Misclassification Error?

- Zero gain may mislead to stop growing in the Misclassification Error case.

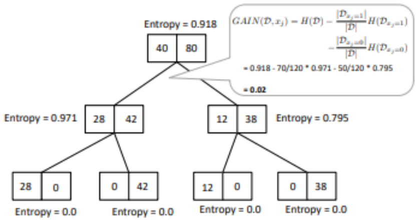

Entropy = 0.918
40
80
$GAIN(\mathcal{D}, x_j) = H(\mathcal{D}) - \frac{|\mathcal{D}_{x_j=1}|}{|\mathcal{D}|} H(\mathcal{D}_{x_j=1})$
$- \frac{|\mathcal{D}_{x_j=0}|}{|\mathcal{D}|} H(\mathcal{D}_{x_j=0})$
= 0.918 - 70/120 * 0.971 - 50/120 * 0.795
= 0.02
Entropy = 0.971
28
42
12
38
Entropy = 0.795
28
0
0
42
12
0
0
38
Entropy = 0.0
Entropy = 0.0
Entropy = 0.0
Entropy = 0.0

Figure: splitting using information gain metric

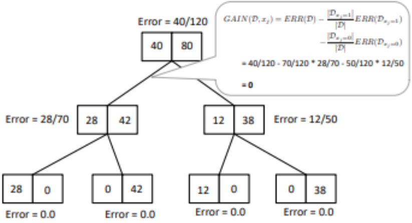

Error = 40/120
40
80
$GAIN(\mathcal{D}, x_j) = ERR(\mathcal{D}) - \frac{|\mathcal{D}_{x_j=1}|}{|\mathcal{D}|} ERR(\mathcal{D}_{x_j=1})$
$- \frac{|\mathcal{D}_{x_j=0}|}{|\mathcal{D}|} ERR(\mathcal{D}_{x_j=0})$
= 40/120 - 70/120 * 28/70 - 50/120 * 12/50
= 0
Error = 28/70
28
42
12
38
Error = 12/50
28
0
0
42
12
0
0
38
Error = 0.0
Error = 0.0
Error = 0.0
Error = 0.0

Figure: splitting using misclassification error metric

When to stop splitting?

- We will stop splitting the nodes until at least one of the following conditions are met:
  - There is no attribute to split upon.
  - Splitting upon any attribute will put all the samples in only one of the two nodes induced by this split.(In particular if all the samples in the current node have similar target labels)

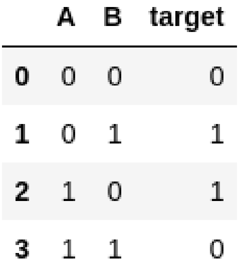
.

Zero information gain as a stopping criterion

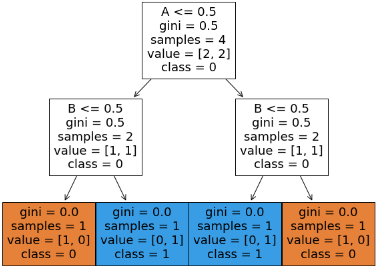
- Another option is to stop splitting when all the attributes are independent to target label(equivalently the information gain is zero). This doesn't work well since pairwise independence doesn't imply independence of several random variables the xor example fully illustrates this.

Xor example

- The task is to learn Xor function of two one-bit numbers and as features we use value of each number. Conditioning on the value of A and B both yield to information gain of zero but the function can be learnt exactly.



	A	B	target
0	0	0	0
1	0	1	1
2	1	0	1
3	1	1	0



Overfitting in Decision Trees

- Building a full Decision tree may cause splitting upon features that may not really have an effect in the label but had shown to have an effect just by chance in the training set. If you make your tree too deep it will overfit the data.

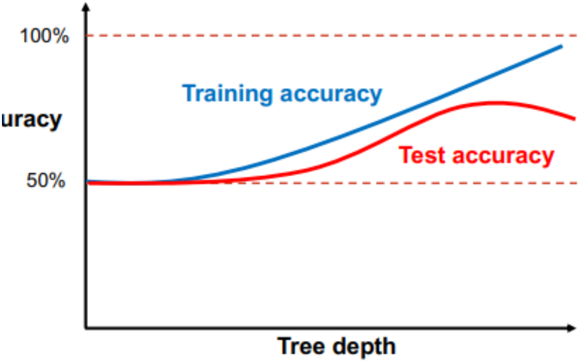


Figure: Train and Test accuracy in terms of depth

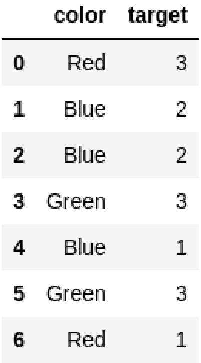
Overfitting Solutions

- Make an split only if it has a positive effect in the accuracy of validation set.
- Limit the depth of the resulting tree.
- Limit the minimum number of samples required to split a node.
- Grow the full tree then prune the nodes based on whether the feature is independent to the target in that node. To do so fix some significance level α and apply χ^2 independence test if p-value is less than α prune that edges connected to that node and make that node a leaf.

Discrete features with more than two values

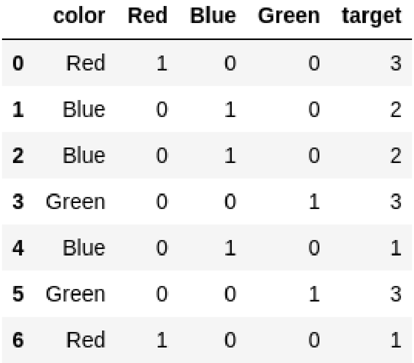
- When dealing with categorical features with $K > 2$ categories one there are several possibilities
  - Create K branches for the decision tree.
  - use one hot encoding for the feature.

One hot encoding for the feature



	color	target
0	Red	3
1	Blue	2
2	Blue	2
3	Green	3
4	Blue	1
5	Green	3
6	Red	1

Figure: initial dataframe



	color	Red	Blue	Green	target
0	Red	1	0	0	3
1	Blue	0	1	0	2
2	Blue	0	1	0	2
3	Green	0	0	1	3
4	Blue	0	1	0	1
5	Green	0	0	1	3
6	Red	1	0	0	1

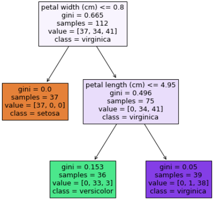
Figure: one-hot encoded dataframe

Continuous features

- When dealing with continuous features in order to make a binary split we will choose some threshold α and split the training examples in the node according to whether $X <= \alpha$ or $X > \alpha$ where X is the random variable indicating value of that feature.
- We will choose value of α maximizing information gain in the split(Only values of alpha which occur in the range of X in the training set should be considered.
- This feature along with other continuous and discrete features will be compared in order to choose the split with maximum information gain.

Decision Tree example on Iris dataset

- Iris dataset consists of 3 different types of irises' (Setosa, Versicolour, and Virginica). The features are Sepal Length, Sepal Width, Petal Length and Petal Width. We will try to predict the type of iris using this features. A decision tree with maximum depth of 2 is used to illustrate this.



petal width (cm) <= 0.8
gini = 0.665
samples = 112
value = [37, 34, 41]
class = virginica

gini = 0.0
samples = 37
value = [37, 0, 0]
class = setosa

petal length (cm) <= 4.95
gini = 0.496
samples = 75
value = [0, 34, 41]
class = virginica

gini = 0.153
samples = 36
value = [0, 33, 3]
class = versicolor

gini = 0.05
samples = 39
value = [0, 1, 38]
class = virginica

- As you can see when can get to quite good results even with maximum depth of 2 in the training set. The accuracy on the test set using this classifier is 89.4.

Decision Tree for regression

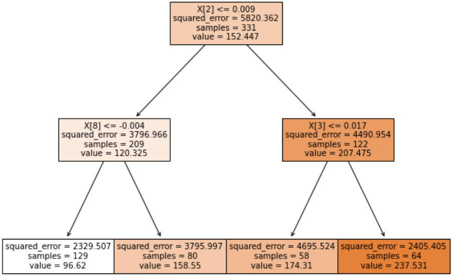
- So far we have discussed decision trees for classification. Decision Trees can also be used for regression. We will use a similar model for each leaf node as we did for classification we will use a single number as the prediction of all samples in the leaf node.
- The single number will be chosen in a way that the least squares error(or equivalently the sample variance) will be minimized. The resulting point will be mean of all target values in the leaf node:

$$\frac{d\sum_{n=1}^{N}(x_n - x)^2}{dx} = 2(\sum_{n=1}^{N}(x_n - x)) = 2(\sum_{n=1}^{N} x_n - nx)$$

  setting the derivative equal to zero yields the result.
- The splits will be chosen in a way to maximize the reduction in inner node least squares error(sum of squares of distances between each target to it's predicted value). The details are just the same as classification version.

Decision Tree for regression example

- We will use diabetes dataset to illustrate Decision Tree for regression. The dataset consists of 9 continuous features and one continuous target value. We will use a Decision tree of depth 2 to illustrate this.



Pros and Cons

- (+) Easy to interpret and communicate
- (+) Independent of feature scaling
- (-) Easy to overfit
- (-) Elaborate pruning required
- (-) Expensive to just fit a “diagonal line”
- (-) Output range is bounded (dep. on training examples) in regression trees

References

- Christopher M. Bishop, Pattern recognition and machine learning
- https://sebastianraschka.com/pdf/lecture-notes/stat479fs19/06-trees__slides.pdf
- https://astronomy.nju.edu.cn/DFS//file/2021/03/03/20210303092750678hv4fsz.pdf

Thank You!

Any Question?