BIG DATA ANALYTICS USING PYTHON

ALGORITHM PRESENTATION WRITE-UP

Decision tree algorithm

BY

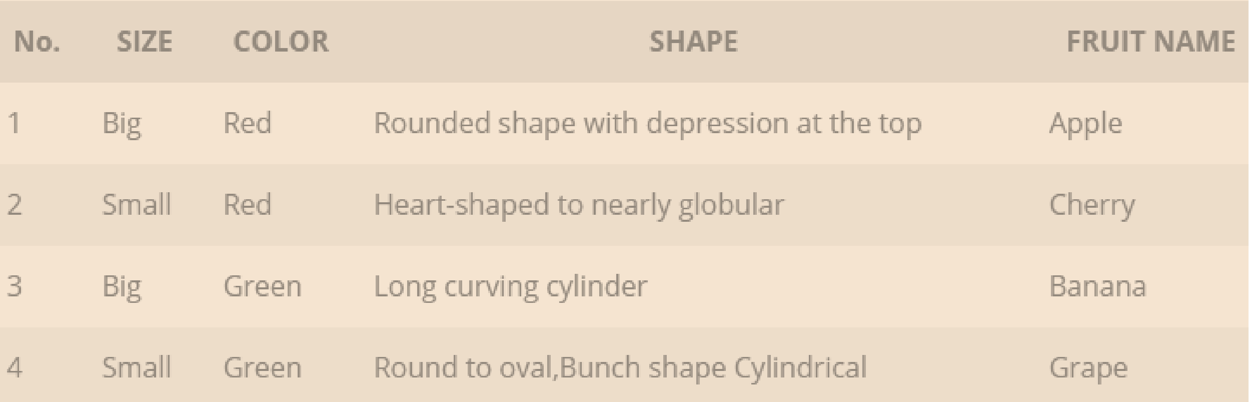
SHESHANK BANDAKUNTA

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 create a training model which can use to predict class or value of target variables by learning decision rules inferred from prior data(training data).
- The understanding level of Decision Trees algorithm is so easy compared with other classification algorithms. The decision tree algorithm tries to solve the problem, by using tree representation. Each internal node of the tree corresponds to an attribute, and each leaf node corresponds to a class label

supervised learning algorithms

- Supervised learning is the machine learning task of inferring a function from labeled training data. The training data consist of a set of training examples. In supervised learning, each example is a pair consisting of an input object (typically a vector) and a desired output value (also called the supervisory signal).



No.	SIZE	COLOR	SHAPE	FRUIT NAME
1	Big	Red	Rounded shape with depression at the top	Apple
2	Small	Red	Heart-shaped to nearly globular	Cherry
3	Big	Green	Long curving cylinder	Banana
4	Small	Green	Round to oval,Bunch shape Cylindrical	Grape

Note : The attribute FRUIT NAME is called class label. In supervised learning each row of different values is associated with a class label. Whereas in Unsupervised learning there is no class label associated. For example, If you find a big, Red and Rounded shape fruit you cannot name it as apple since you don’t have class label to it, In supervised learning you can say the fruit belongs to apple family.

Regression and Classification

- The main goal of classification is to predict the target class (Yes/ No). For example problems like approving a loan to a person. (yes or no type of problems)
- The main goal of regression algorithms is the predict the discrete or a continues value. For example problems like how much percentage a student would get in his final exams depending on his performance in prior exams.(a single value, In above example it is percentage such as 80% or 90%.)

Note : As mentioned earlier decision tree algorithm can solve both regression and classification problems.

Training data vs Test data

- A training set is a set of data used to discover potentially predictive relationships. A test set is a set of data used to assess the strength and utility of a predictive relationship. Test and training sets are used in intelligent systems, machine learning, genetic programming and statistics.

When to Consider Decision Trees

- Instances describable by attribute–value pairs
  - Going deeper into the algorithm, The dataset is divided with attribute–value pairs, Therefore the data must be describable by attribute–value pairs.
- Target function is discrete valued
  - Target value must be discrete or single valued that is It must either be yes or no. It cannot be (yes, no) for single class.
- Possibly noisy training data
  - Some data might be missing in the dataset, With such kind of dataset the algorithm works efficiently.

Application of big data to the algorithm

- **Medical Equipment**
  - For suppose if we have a data set consisting of number of patients coming to an hospital with various diseases and management wants to predict they need more equipment for and x disease, they can take decision based on this algorithm.
- **Credit risk analysis**
  - If a person is visiting for personal loan, Then bank takes all the details from the customer and can decide to give loan based on the previous data.
- **Scheduling preferences**
  - If a tennis match is scheduled on a particular date and place. If we want know if we can play or not based on the previous data of temperature in that area.

What to do next?

- Place the best attribute of the dataset at the **root** of the tree.
- Split the training set into **subsets**. Subsets should be made in such a way that each subset contains data with the same value for an attribute.
- Repeat step 1 and step 2 on each subset until you find **leaf nodes** in all the branches of the tree.

How to do step-1

- Order to placing attributes as root or internal node of the tree is done by using some statistical approach. If we follow a random approach, it may give us bad results with low accuracy.

  The popular attribute selection measures:

- ➢ Entropy
- ➢ Information gain

ENTROPY

Entropy(S) = expected number of bits needed to encode + or - for a randomly drawn member of S(under the optimal, shortest length code).

- Where S is a sample of training examples.
- P is the proportion of positive examples in S
- N is the proportion of negative examples in S
- Entropy measures the impurity of S

  Information theory:

- The optimal length code for a message having the probability p is − log2 p bits.
- Entropy(S) = - p(P)log2 p(P) – p(N)log2 p(N)

Information gain

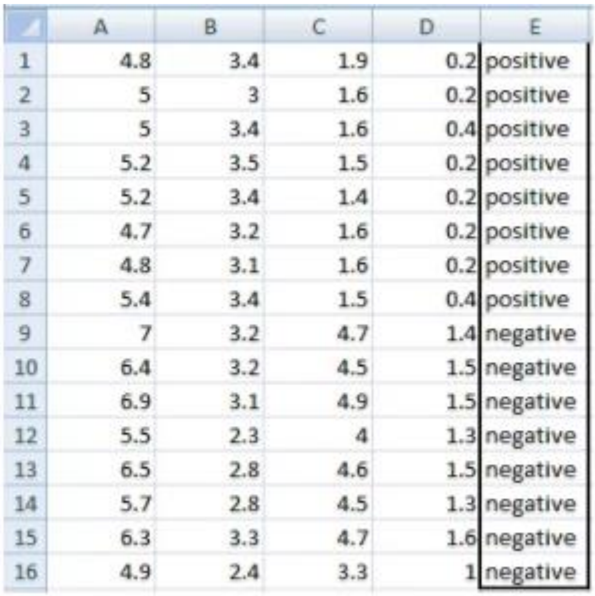
Information Gain calculates the expected reduction in entropy due to sorting on the attribute A.

For a binary classification problem with only two classes, positive and negative class.

- If all examples are positive or all are negative then entropy will be zero i.e, low.
- If half of the records are of positive class and half are of negative class then entropy is one i.e, high.
- Else

$$Gain(S, A) \equiv Entropy(S) \; - \sum_{v \in Values(A)} \frac{|S_v|}{|S|} Entropy(S_v)$$

Example: Constructing Decision Tree using "information gain"



	A	B	C	D	E
1	4.8	3.4	1.9	0.2	positive
2	5	3	1.6	0.2	positive
3	5	3.4	1.6	0.4	positive
4	5.2	3.5	1.5	0.2	positive
5	5.2	3.4	1.4	0.2	positive
6	4.7	3.2	1.6	0.2	positive
7	4.8	3.1	1.6	0.2	positive
8	5.4	3.4	1.5	0.4	positive
9	7	3.2	4.7	1.4	negative
10	6.4	3.2	4.5	1.5	negative
11	6.9	3.1	4.9	1.5	negative
12	5.5	2.3	4	1.3	negative
13	6.5	2.8	4.6	1.5	negative
14	5.7	2.8	4.5	1.3	negative
15	6.3	3.3	4.7	1.6	negative
16	4.9	2.4	3.3	1	negative

- A, B, C, D attributes can be considered as predictors and E column class labels can be considered as a target variable.
- For constructing a decision tree from this data, we have to convert continuous data into categorical data. Which means select a threshold value and group greater and lesser values and apply the algorithm.
- We have to choose some random values to categorize each attribute:

There are **2 steps for calculating information gain** for each attribute:

- Calculate entropy of Target.
- Entropy for every attribute A, B, C, D needs to be calculated. Using information gain formula we will subtract this entropy from the entropy of target. The result is Information Gain.

The entropy of Target

- We have 8 records with negative class and 8 records with positive class. So, we can directly estimate the entropy of target as 1.
- E(8,8) = -1*((p(P)*log(p(P)) + (p(N)*log(p(N)))
  = $-1*((8/16)*\log_2(8/16)) + (8/16) * \log_2(8/16))$
  = 1

Note: In our case entropy =1.

Information gain for Var A

- Var A has value >=5 for 12 records out of 16 and 4 records with value <5 value.
- For Var A >= 5 & class == positive: 5/12
- For Var A >= 5 & class == negative: 7/12

  - Entropy(5,7) = -1 * ((5/12)*log2(5/12) + (7/12)*log2(7/12)) = 0.9799
- For Var A <5 & class == positive: 3/4
- For Var A <5 & class == negative: 1/4
  - Entropy(3,1) = -1 * ((3/4)*log2(3/4) + (1/4)*log2(1/4)) = 0.81128
- Entropy(Target, A) = P(>=5) * E(5,7) + P(<5) * E(3,1)
  = (12/16) * 0.9799 + (4/16) * 0.81128 = 0.937745
- Information Gain(IG) = E(Target) – E(Target, A) = 1-0.937745 = 0.062255

Information gain for Var B

- Var B has value >=3 for 12 records out of 16 and 4 records with value <5 value.
- For Var B >= 3 & class == positive: 8/12
- For Var B >= 3 & class == negative: 4/12
  - Entropy(8,4) = -1 * ((8/12)*log2(8/12) + (4/12)*log2(4/12))
    = 0.39054
- For VarB <3 & class == positive: 0/4
- For Var B <3 & class == negative: 4/4
  - Entropy(0,4) = -1 * ((0/4)*log2(0/4) + (4/4)*log2(4/4)) = 0
- Entropy(Target, B) = P(>=3) * E(8,4) + P(<3) * E(0,4)
  = (12/16) * 0.39054 + (4/16) * 0 = 0.292905
- Information Gain(IG) = E(Target) – E(Target, B) = 1-0.292905 = 0.070795

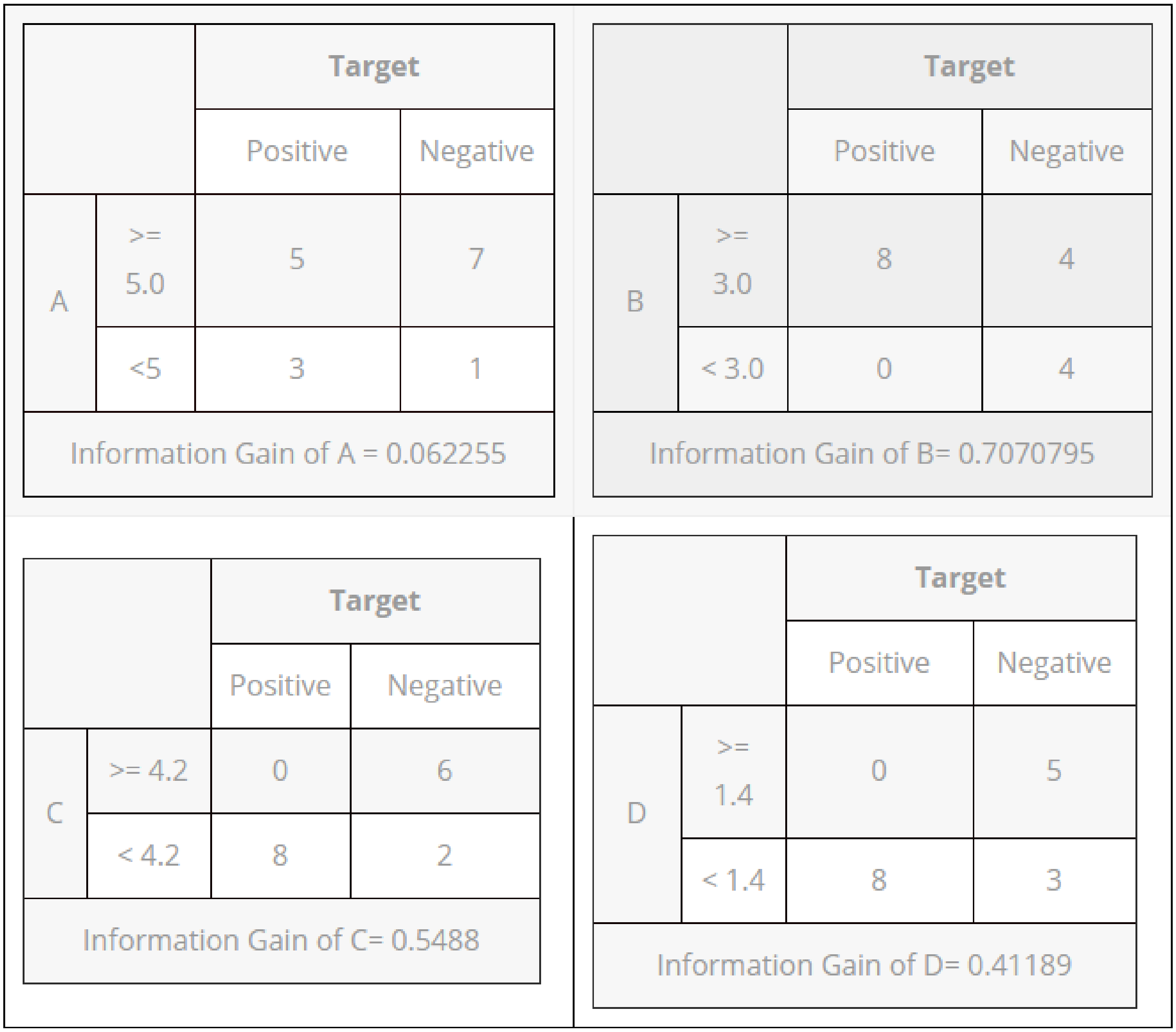
Information gain for Var C

- Var C has value >=4.2 for 6 records out of 16 and 10 records with value <4.2 value.
- For Var C >= 4.2 & class == positive: 0/6
- For Var C >= 4.2 & class == negative: 6/6
    - Entropy(0,6) = 0
- For VarC < 4.2 & class == positive: 8/10
- For Var C < 4.2 & class == negative: 2/10
    - Entropy(8,2) = 0.72193
- Entropy(Target, C) = P(>=4.2) * E(0,6) + P(< 4.2) * E(8,2)
  = (6/16) * 0 + (10/16) * 0.72193 = 0.4512
- Information Gain(IG) = E(Target) – E(Target, C) =1-0.4512 = 0.5488

Information gain for Var D

- Var D has value >=1.4 for 5 records out of 16 and 11 records with value <5 value.
- For Var D >= 1.4 & class == positive: 0/5
- For Var D >= 1.4 & class == negative: 5/5
    - Entropy(0,5) = 0
- For Var D < 1.4 & class == positive: 8/11
- For Var D < 14 & class == negative: 3/11
    - Entropy(8,3) = -1 * ((8/11)*log2(8/11) + (3/11)*log2(3/11))
      = 0.84532

- Entropy(Target, D) = P(>=1.4) * E(0,5) + P(< 1.4) * E(8,3)
  = 5/16 * 0 + (11/16) * 0.84532 = 0.5811575
- Information Gain(IG) = E(Target) – E(Target, D) = 1-0.5811575 = 0.41198



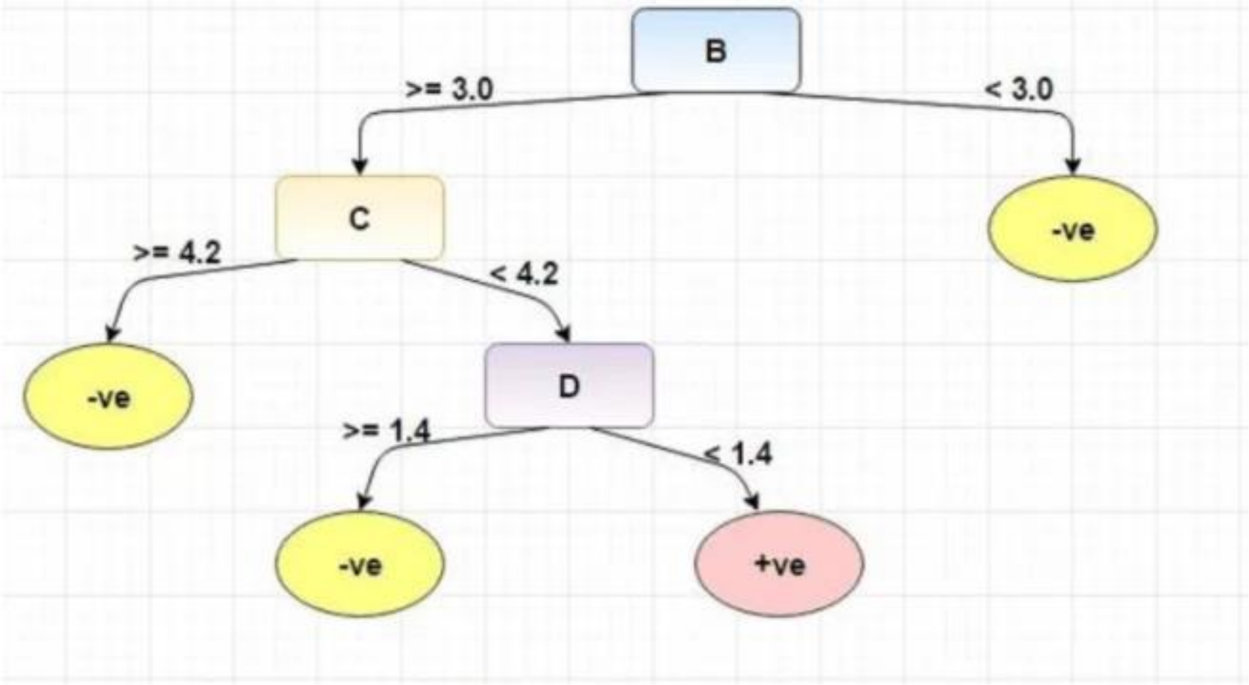
		Target	
		Positive	Negative
A	>= 5.0	5	7
	<5	3	1
Information Gain of A = 0.062255			

		Target	
		Positive	Negative
B	>= 3.0	8	4
	< 3.0	0	4
Information Gain of B= 0.7070795			

		Target	
		Positive	Negative
C	>= 4.2	0	6
	< 4.2	8	2
Information Gain of C= 0.5488			

		Target	
		Positive	Negative
D	>= 1.4	0	5
	< 1.4	8	3
Information Gain of D= 0.41189			

- From the above Information Gain calculations, we can build a decision tree. We should place the attributes on the tree according to their values.
- An Attribute with better value than other should position as root and A branch with entropy 0 should be converted to a leaf node. A branch with entropy more than 0 needs further splitting.



Overfitting

Overfitting is a practical problem while building a decision tree model. The model is having an issue of overfitting is considered when the algorithm continues to go deeper and deeper in the to reduce the training set error but results with an increased test set error i.e, Accuracy of prediction for our model goes down. It generally happens when it builds many branches due to outliers and irregularities in data.

Two approaches which we can use to avoid overfitting are:

- Pre-Pruning
- Post-Pruning

Pre-Pruning

- In pre-pruning, it stops the tree construction bit early. It is preferred not to split a node if its goodness measure is below a threshold value. But it’s difficult to choose an appropriate stopping point.

Post-Pruning

- In post-pruning first, it goes deeper and deeper in the tree to build a complete tree. If the tree shows the overfitting problem then pruning is done as a post-pruning step. We use a cross-validation data to check the effect of our pruning. Using cross-validation data, it tests whether expanding a node will make an improvement or not.
- If it shows an improvement, then we can continue by expanding that node. But if it shows a reduction in accuracy then it should not be expanded i.e, the node should be converted to a leaf node.

Decision Tree Algorithm Advantages

- Decision Trees are easy to explain. It results in a set of rules.
- It follows the same approach as humans generally follow while making decisions.
- Interpretation of a complex Decision Tree model can be simplified by its visualizations. Even a naive person can understand logic.
- The Number of hyper-parameters to be tuned is almost null.

Disadvantages

- There is a high probability of overfitting in Decision Tree.
- Information gain in a decision tree with categorical variables gives a biased response for attributes with greater no. of categories.
- Calculations can become complex when there are many class labels.

REFERENCES

- https://dataaspirant.com/2017/01/30/how-decision-tree-algorithm-works/
- https://dataaspirant.com/2014/09/27/classification-and-prediction/
- http://www4.stat.ncsu.edu/~dickey/Analytics/Datamine/Reference%20Papers/machine%20learning.pdf
- Image credits - http://dataaspirant.com/2017/01/30/how-decision-tree-algorithm-works/