Labsheet - 4

Decision Trees

Machine Learning

BITS F464
I Semester 2021-22

Decision Trees

Decision tree is a graph to represent choices and their results in form of a tree. The nodes in the graph represent an event or choice and the edges of the graph represent the decision rules or conditions. They are very powerful algorithms, capable of fitting complex datasets

Important Terminology related to Decision Trees

1. Root Node: It represents entire population or sample and this further gets divided into two or more homogeneous sets.
2. Splitting: It is a process of dividing a node into two or more sub-nodes.
3. Decision Node: When a sub-node splits into further sub-nodes, then it is called decision node.
4. Leaf/ Terminal Node: Nodes with no children (no further split) is called Leaf or Terminal node.
5. Pruning: When we reduce the size of decision trees by removing nodes (opposite of Splitting), the process is called pruning.
6. Branch / Sub-Tree: A sub section of decision tree is called branch or sub-tree.
7. Parent and Child Node: A node, which is divided into sub-nodes is called parent node of sub-nodes where as sub-nodes are the child of parent node.

There are couple of algorithms to build a decision tree:

1. CART :- uses Gini Index (Classification) as metric.
2. ID3:- uses Entropy function and Information gain as metrics.
3. C4.5
4. Hunt’s algorithm

In order to grow decision tree please load the Carseats dataset using the rpart package in R. Mind that you need to install the ISLR and rpart packages in your R Studio environment first. Let's first load the Carseats dataframe from the ISLR package.

```
library(ISLR)
```

Let's also load the rpart package.

```
library(rpart)
```

The Carseats dataset is a dataframe with 400 observations on the following 11 variables:

- Sales: unit sales in thousands
- CompPrice: price charged by competitor at each location
- Income: community income level in 1000s of dollars
- Advertising: local ad budget at each location in 1000s of dollars
- Population: regional pop in thousands
- Price: price for car seats at each site
- ShelveLoc: Bad, Good or Medium indicates quality of shelving location
- Age: age level of the population
- Education: ed level at location
- Urban: Yes/No
- US: Yes/No

Let's take a look at the histogram of car sales:

```
hist(carseats$Sales)
```

Observe that Sales is a quantitative variable. You want to demonstrate it using trees with a binary response. To do so, you turn Sales into a binary variable, which will be called High. If the sales is less than 8, it will be not high. Otherwise, it will be high. Then you can put that new variable High back into the dataframe

```
High = ifelse(carseats$Sales<=8, "No", "Yes")

carseats = data.frame(carseats, High)
```

Now let's fill a model using decision trees. Of course, you can't have the Sales variable here because your response variable High was created from Sales. Thus, let's exclude it and fit the tree.

```
tree.carseats = rpart(High~.-Sales, data=carseats)
```

Let's see the summary of your classification tree:

```
summary(tree.carseats)
```

You can see the variables involved, the number of terminal nodes, the residual mean deviance, as well as the misclassification error rate. To make it more visual, let's plot the tree as well, then annotate it using the handy text function:

```
plot(tree.carseats, uniform = TRUE)

text(tree.carseats, pretty=1)

# Examine the complexity plot

printcp(tree.carseats)

plotcp(tree.carseats)
```

There are so many variables, making it very complicated to look at the tree. At least, you can see that at each of the terminal nodes, they're labeled Yes or No. At each splitting node, the variables and the value of the splitting choice are shown (for example, Price < 92.5 or Advertising < 13.5). For a detailed summary of the tree, simply print it. It'll be handy if you want to extact details from the tree for other purposes:

```
tree.carseats
```

There are two types of pruning: pre-pruning and post-pruning.

1. Pre-pruning

Prepruning is also known as early stopping criteria. As the name suggests, the criteria are set as parameter values while building the rpart model. Below are some of the pre-pruning criteria that can be used. The tree stops growing when it meets any of these pre-pruning criteria, or it discovers the pure classes.

- **maxdepth**: This parameter is used to set the maximum depth of a tree. Depth is the length of the longest path from a Root node to a Leaf node. Setting this parameter will stop growing the tree when the depth is equal the value set for maxdepth.

- **minsplit**: It is the minimum number of records that must exist in a node for a split to happen or be attempted. For example, we set minimum records in a split to be 5; then, a node can be further split for achieving purity when the number of records in each split node is more than 5.

```
Preprune_tree.carseatss = rpart(High~.-Sales, data=carseats, method=”class”, control
= rpart.control(cp = 0, maxdepth = 8,minsplit = 100))
```

2. **Post-pruning**

The idea here is to allow the decision tree to grow fully and observe the CP value. Next, we prune/cut the tree with the optimal CP value as the parameter as shown in below code:

```
Postprune_tree.carseats <- prune(tree.carseats, cp = 0.0026 )
```

Exercises

- What are the reasons for overfitting in decision tree?
- How do you deal with continuous attribute in terms of splitting?
- What is the accuracy of decision tree by randomly selecting 70% data for testing and 30% for training?
