1 What we learned in class

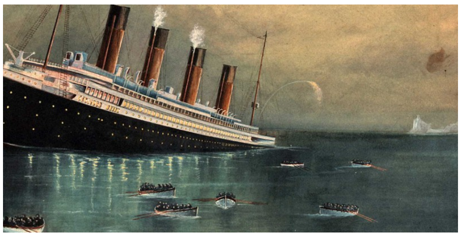
- What is a Decision Tree.
- How to creating a decision tree.
- How to use Gini Index to split.
- Implementation of CART on ”iris” data in R.

2 Take Home Assignment

Decision tree is one of the most powerful yet simplest supervised machine learning algorithm, it is used for both classification and regression problems. CART (Breiman *et al.*,1984) is one of the approaches to create such a decision tree.

In this assignment, we will use CART to create a decision tree to classify the two class labels ”survived” or ”not survived” in ”ptitanic” data sets. And report the classification accuracy just like what we did in class.

2.1 Data



The sinking of the Titanic

In class, we used ”iris” data sets to feed our CART model. In this assignment, we will use another built-in R data sets called ”ptitanic”. This dataset consists of details of passengers who were aboard on Titanic with two class labels ”survived” or ”not survived” (In ”iris” dataset, we have three different class label: ”setosa”, ”versicolor”, and ”virginica”). This data frame has 1046 observations on 6 variables/attributes. I listed an explanation of each variable.

- pclass: passenger class
- survived: died or survived
- sex: male or female
- age: age in years
- sibsp: number of siblings or spouses aboard
- parch: number of parents or children aboard

In R, you can use:

```
summary(ptitanic)
```

to see a summary of the ”ptitanic” data set.

2.2 Instructions

Recall that we used ”rpart” package in R to run a classification tree if the response variable is discrete, and a regression tree if the response variable is numerical. ”rpart.plot” package tailor the plot for the model’s response type. ”naniar” package is used to handle missing values.
Here is a step by step instructions for this assignment.

1. Import all necessary R packages and ”ptitanic” data set.

2. Check the missing values in this data sets.

3. Randomly sample 70% observations as your training data. The rest 30% will be your test data.

4. Implement CART on the training data with ”survived” or ”not survived” as your label. The variable name for this label is ”survived” (we use ”species” as our label in ”iris” dataset).

5. Prune the decision tree to find the optimal stopping point. Please use the following code for ”iris” as a reference:

   ```
   printcp(tree)
   pruned <- prune(tree,cp = tree$cptable[which.min(tree$cptable[,"xerror"]),"CP"])
   ```

6. Plot the pruned decision tree.

7. Make a prediction on the test data. Don’t forget to transfer your predicted label data into the ”data frame” form.

8. Combine the predicted label with the truth label, then create a confusion matrix to compare these two.

9. the last thing is to report the prediction accuracy.

3 Just for your interest

Tree based algorithms are considered to be one of the best and most used supervised learning methods. Tree based algorithms empower predictive models with high accuracy, stability and ease of interpretation. Unlike linear models, they map non-linear relationships quite well. They are adaptable at solving any kind of problem at hand (classification or regression). Methods like decision trees, random forest, gradient boosting are being popularly used in all kinds of data science problems. We introduced CART approach of decision tree in class, and we also implemented two examples by using ”iris” and ”ptitanic” data sets. I will summarize the advantages of the CART:

- **Easy to Understand:** Decision tree output is very easy to understand even for people from non-analytical background. It does not require any statistical knowledge to read and interpret them. Its graphical representation is very intuitive and users can easily relate their hypothesis.

- **Useful in Data Exploration:** Decision tree is one of the fastest ways to identify most significant variables and relation between two or more variables. With the help of decision trees, we can create new variables/attributes that has better power to predict target variable.

- **Non Parametric Method:** Decision tree is considered to be a non-parametric method. This means that decision trees have no assumptions about the space distribution and the classifier structure.

- **Less Data Cleaning:** It requires less data cleaning compared to some other modeling techniques. One of my friends is working at Cleveland Clinic as a data scientist, and he told me that he spends more than half of his time cleaning/pre-processing the raw data. CART usually requires less work in cleaning the data, which is really good news for data scientists.

There are definitely some disadvantages of CART/ decision trees. The first thing is about overfitting, it is one of the most practical difficulties for decision tree models. Small changes in the data can cause a large change in the structure of the decision tree that in turn leads to instability. This problem gets solved by setting constraints on model parameters and pruning. Another problem is about the time complexity, the calculations of creating a decision tree can go complex compare to the other traditional algorithms. There is still a lot more to learn, our in-class lecture and this assignment give you a quick start to explore. If you like CART model and want to dig deeper on this topic, here are some useful resources:

1. A book chapter written by Rokach and Maimon. It presents an updated survey of current methods for constructing decision tree (including CART). They suggest a unified algorithmic framework for presenting these algorithms, and describe various splitting criteria, like Information Gain, DKM, Gain Ratio... Also, there is a section that discusses various techniques for pruning decision trees.
   `https://www.ise.bgu.ac.il/faculty/liorr/hbchap9.pdf`

2. We saw Gini Index function in class, here is a great example to explain Gini Index in detail: `https://blog.quantinsti.com/gini-index/`

3. If you would like to implement CART in Python, here is a blog article that provides you a step by step instructions for implementing CART in Python. `https://blog.paperspace.com/decision-trees/`