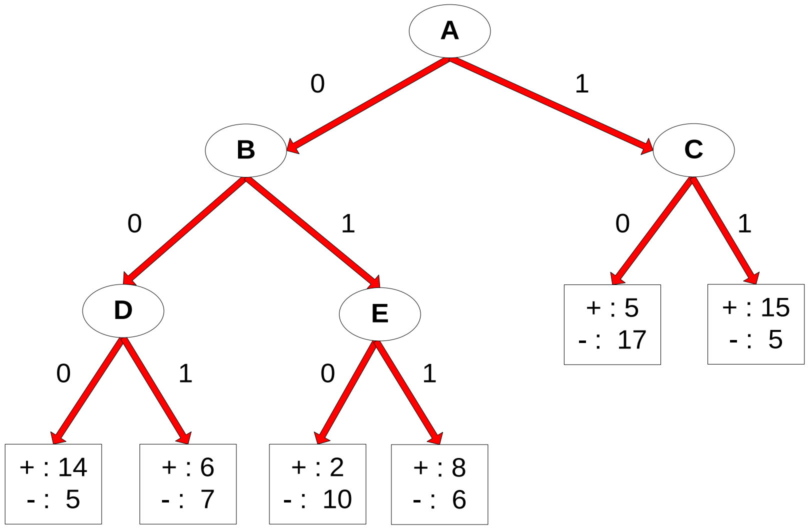
Colin Houde

CAP 5610: HW3: Decision Tree and Ensemble Learning

Question 1. (10 points) Understanding Training Error and Testing

Consider the decision tree shown in the diagram below. The counts shown in the leaf nodes correspond to the number of training records associated with the nodes.



(a) What is the training error rate for the tree? Explain how you get the answer?

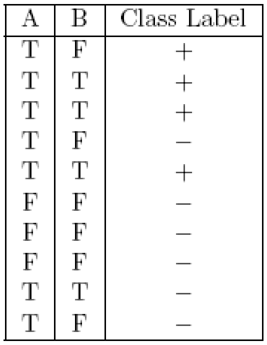
With an optimistic error rate, we find: 29/100 = 29%
With a pessimistic error rate, we find: 29/100 + 1*6/100 = 35/100 = 35%

(b) Given a test instance T={A=0, B=1, C=1, D=1, E=0}, what class would the decision tree above assign to T? Explain how you get the answer?

Since there are three 1's and 2 0's, the decision tree will assign a 1 to the test instance.

Question 2 (16 points) Understand Splitting Process

Consider the following data set for a binary class problem.



A	B	Class Label
T	F	+
T	T	+
T	T	+
T	F	−
T	T	+
F	F	−
F	F	−
F	F	−
T	T	−
T	F	−

Number of True: 4
Number of False: 6
True rate: 40%
False rate: 60%

Q1: What is the overall gini before splitting?

Original Gini = 1 – (0.4)^2 – (0.6)^2 = 0.48

This means that 48% of a randomly chosen variables will be wrongly classified

Q2: What is the gain in gini after splitting on A?

Number of True in A: 7
Number of False in A: 3

Gini(A)= T = 1 – (4/7)^2 – (3/7)^2 = 0.49
Gini(B = F = 1 – (3/3)^2 – (0/3)^2 = 0

Gain = GOrig(0.48) – (7/10* GTrue(0.49)) – (3/10 * GFalse(0)) = 0.48 – 0.343 – 0 = 0.137

Q3: What is the gain in gini after splitting on B:

Number of True in B: 4
Number of False in B: 6

Gini(B) = T = 1 - (3/4)^2 - (1/4)^2 = 0.375
Gini(B) = F = 1 – (1/6)^2 – (5/6)^2 = 0.2778

Gain = GOrig – (4/10 * GTrue(0.375)) – (6/10 * GFalse(0.2778)) = 0.48 – 0.15 – 0.16668 = 0.16332

Q4: Which attribute would the decision tree choose?

Since splitting B yields a higher gain in Gini, we would choose B to split the node

Colin Houde

Question 3: (15 points) Please answer and explain.

Q1: Are decision trees a linear classifier? Why?

- **Decision Trees are non-linear because there are no linear relationships between the independent and the dependent variables within the tree.**

Q2: What are the weaknesses of decision trees? Why?

- **Disadvantages to a decision tree can be :**
  - **The interacting attributes may be outweighing other attributes that are less discriminating. This is bad because one attribute can essentially decide the class of something without taking anything else into account.**
  - **The decisions can only take in a single attribute. This is bad because everything has to be either a yes or no answer to something and not a continuous gradient of answers.**

Q3: Is Misclassification error better than Gini index as the splitting criteria for decision trees? Why?

- **The Gini index is preferred over Misclassification because it is more sensitive to the data. It will allow for further splits than misclassification which means the tree will become more accurate.**

Question-4 (35 points) Build decision tree and random forest using Scikit Learn （https://scikit-learn.org/stable/）

For the Titanic challenge (https://www.kaggle.com/c/titanic), we need to guess whether the individuals from the test dataset had survived or not. Please:

1) Preprocess your Titanic training data; Please briefly describe what preprocess you have done.

   When looking at the data for preprocessing, I took a look at the following:

   Null Values

   - **I found there were 177 Null Values in the Age category and 687 Null values in the Cabin category.**
     - **For the age category, I set all the null values to the mean average of all the ages that weren't null.**
     - **I set the Cabin category to a numeric representation with 0 representing Null and 7-1 representing cabins A-G. I set cabin A as a higher value than cabin G due to the nature of cabin A generally having a higher priority than lower cabins.**

   - **I also found there were 2 Null Values in the Embarked categories, since there was such a miniscule amount in the embarked category, I decided to just drop those two rows from the training set all together.**

   Categorical Data:

   - **I changed 'Sex' from 'Male' and 'Female' to 1 and 2 respectively**

   Names

   - **I have decided to drop the names column entirely. There was a correlation between names with 'Dr.' as they had a higher chance of survival. But there were only 7 'Dr.' total in the whole training set so I just went and removed them.**

2) Select a set of important features. Please show your selected features and explain how you perform feature selection.

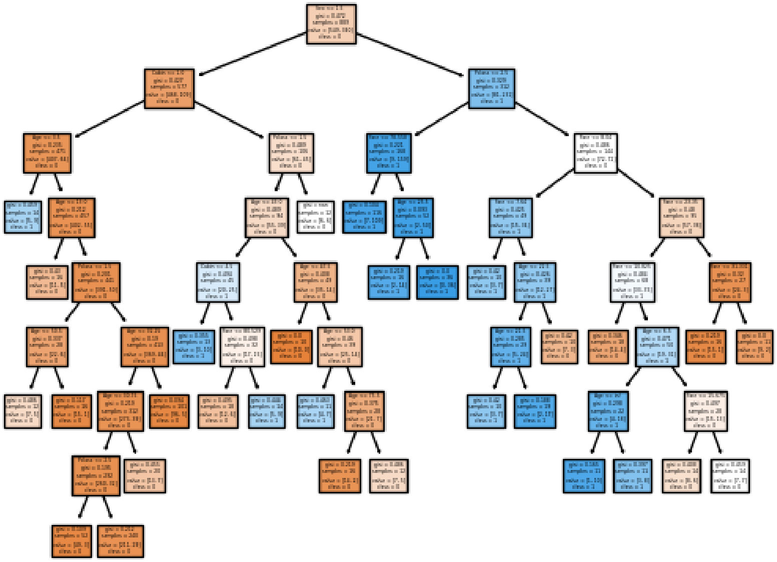
For the important features I used the selectKBest function from the sklearn library to return the top features.

The top 5 features according to the chi2 score are:

1) Fare
2) Cabin
3) Sex
4) Age
5) PClass

So, I will be using just these 5 features in the training.

3) Learn and fine-tune a decision tree model with the Titanic training data, plot your decision tree;



I used max_depth=8, min_samples_leaf=10, min_samples_split=10 for the tree above.

4) Apply the five-fold cross validation of your fine-tuned decision tree learning model to the Titanic training data to extract average classification accuracy;

   I got Decision Tree Classifier: 0.80409 accuracy with a standard deviation of 0.03

5) Apply the five-fold cross validation of your fine-tuned random forest learning model to the Titanic training data to extract average classification accuracy;

   I got Random Forest Classifier: 0.81307 accuracy with a standard deviation of 0.04

6) Which algorithm is better, Decision Tree or Random Forest?

   I found that in every case I ran, the Random Forest Classifier produced a higher accuracy percentage than the decision tree classifier with roughly the same standard deviation. So, in that case, I would conclude the random forest algorithm is a bit better with this model.

7) What are your observations and conclusions from the algorithm comparison and analysis?

   The Random Forest and Decision Tree classifiers are very similar with this data set and model. They both produce about 80% accuracy rate after my data preprocessing and feature selection. I am sure there is a better variation of what data preprocessing to use and what features to pick. I have also found that without proper hyperparameter tuning, the decision tree classifier is very prone to overfitting. Some of the decision trees produced were very large and had an unnecessary number of splits and depth. The trees itself were also extremely different each time I produced a new tree with the same data. The Random Forest, I found, had a better generalization of the data and a more consistent prediction rate.

Question-5 (20 points) Build a bagging classifier using Scikit Learn for the above Titanic challenge.

I used the decision tree I made for the above question as the base classifier in this bagging question. The bagging resulted in roughly about the same accuracy as the random forest classifier but a bit higher. I averaged about 82% with the bagging classifier when using parameters: n_estimators=10, random_state=42. I also found that the more you increase the test_size of the training data split, the lower accuracy the bagging became. I found a test_size of 0.2 produced the best results with the bagging method.

Bagging Accuracy: 0.8202247191011236

Question-6 (20 points) Build an Adaboost classifier using Scikit Learn for the above Titanic challenge.

For the AdaBoost classifier, I also used the same parameters of: n_estimators=10, random_state=42 to keep a baseline comparison relative to the bagging classifier. I found the Adaboost to be least accurate when compared to the decision tree, random forest, and the bagging classifiers. Not by much though. For example, here is one test run of all the classifiers and their respective accuracies:

Decision Tree Classifier: 0.79960 accuracy with a standard deviation of 0.03
Random Forest Classifier: 0.80969 accuracy with a standard deviation of 0.04
Bagging Accuracy: 0.8202247191011236
AdaBoost Accuracy: 0.797752808988764

As we can see they are all very close, but the AdaBoost Is the least accurate.