Decision Tree and Ensemble Learning

Task 1

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:

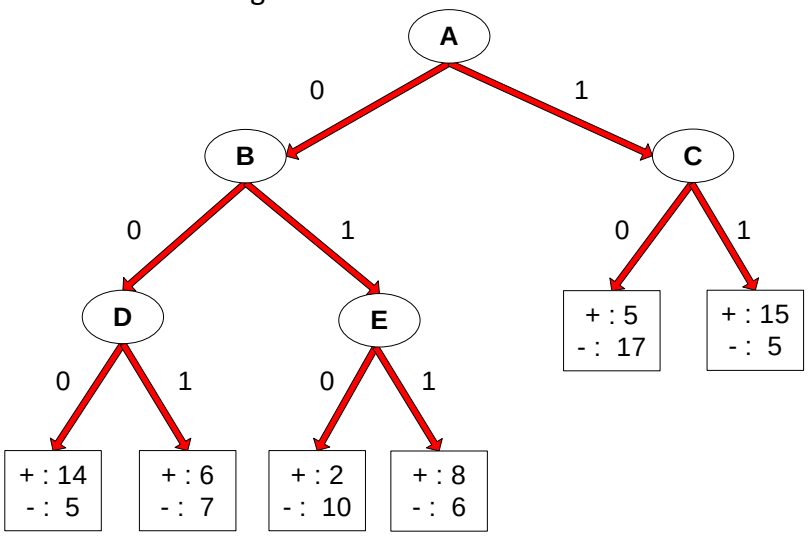
1) Preprocess your Titanic training data;
2) Select a set of important features. Please show your selected features and explain how you perform feature selection.
3) Learn and fine-tune a decision tree model with the Titanic training data, plot your decision tree;
4) Apply the five-fold cross validation of your fine-tuned decision tree learning model to the Titanic training data to extract average classification accuracy;
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; 6) Which algorithm is better, Decision Tree or Random Forest?
6) Which algorithm is better, Decision Tree or Random Forest?
7) What are your observations and conclusions from the algorithm comparison and analysis?

All the answer to the first question are provided in the jupyter notebook uploaded on the below GitHub link:

https://github.com/anujsarda3/ML-HW2/blob/main/HW2.ipynb

Task 2

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?

Answer)

Training Error rate = Fraction of mistakes made on training set

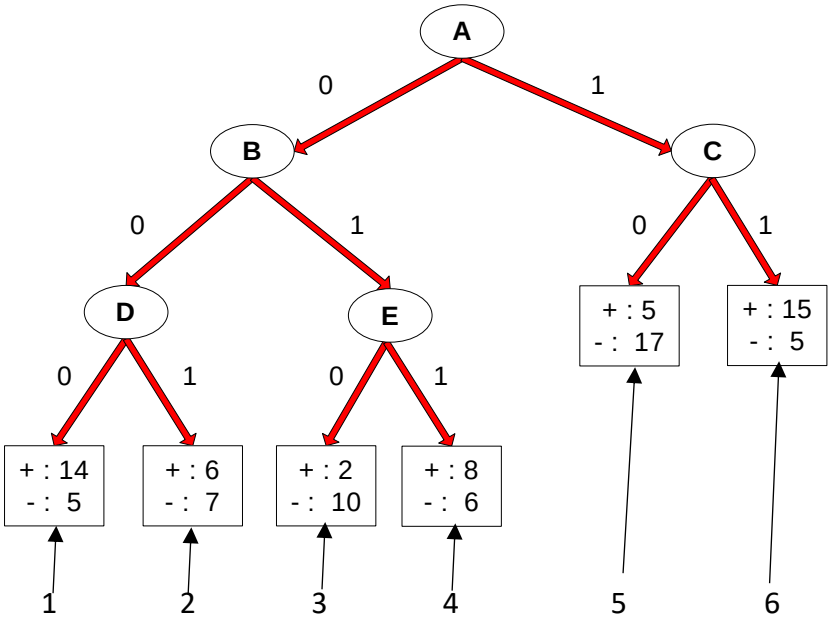
= number of misclassified records/ total number of records

Number of misclassified records can be calculated in the following way:

i) In a node, calculate the total number of positive and negative samples in that node.

ii) Number of misclassified records will be the one which are less in number as compared to the number records of other class.

Total number of records in this question = 100



Number of misclassified records are as follows:

1 -> 5(negative class)
2 -> 6(positive class)
3 -> 2(positive class)
4 -> 6(negative class)
5 -> 5(positive class)

6 -> 5(negative class)

Total number of misclassified records = 29

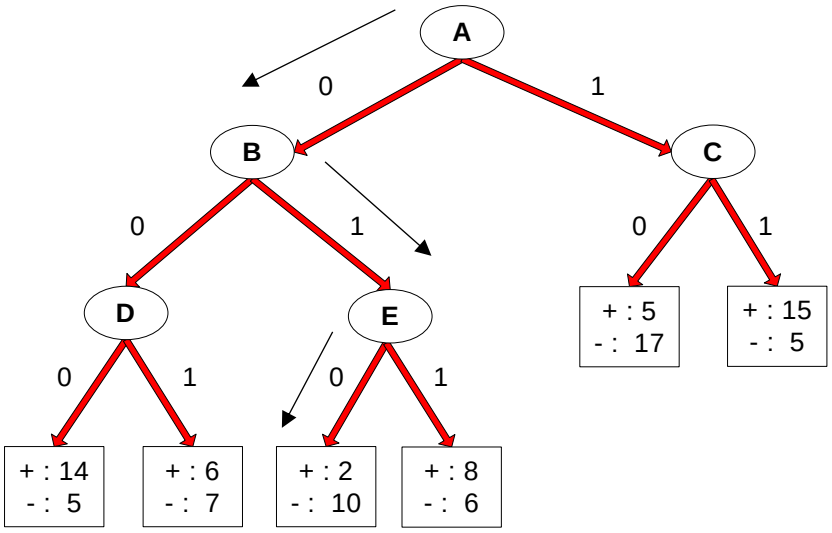
Training error rate = 29/100 = 0.29

(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?

Answer)

While assigning a class to a test instance, tree is always traversed starting with the root node. As root node is A and in the test instance A=0, therefore the next node it reaches is B. Same follows at B, and next node is E. E=0 in the test instance therefore it will follow the E=0 box. As it can be observed that majority of the records in E=0 box is negative, so the model will ascend negative class for the whole box and hence negative class would be assigned to T. It can also be seen in below diagram:

Test instance T= {A=0, B=1, C=1, D=1, E=0}



Therefore, Negative class would be assigned to test instance T.

Task 3

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	−

Q1: What is the overall entropy before splitting?

Answer)

$$\text{Entropy} = -\sum_j p\left(\frac{j}{t}\right) \log p\left(\frac{j}{t}\right)$$

Where $p\left(\frac{j}{t}\right)$ is the relative frequency of class j at node t.

Before splitting:

Class Label	Count of Class label
+	4
-	6

Entropy = - (p(+/t) log p(+/t) + p(-/t) log p(-/t))
= - (4/10 * log (4/10) + 6/10 * log (6/10))
= - (0.4 * -(1.32) + 0.6 * -(0.74))
= (0.528 + 0.443)
= 0.970

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

Answer)

Decision tree after splitting on A:

Class Label	Count of Class label
+	4
-	6

T F

1)

Class Label	Count of Class label
+	4
-	3

2)

Class Label	Count of Class label
+	0
-	3

Gain in entropy after splitting = Entropy before splitting – $\sum_{i=1}^{k}((n_i/n) * Entropy(i))$

Where n_i is number of records in partition i.

Entropy before splitting = 0.970 (as calculated in last question)

Entropy(1) = - ((4/7) * log (4/7) + (3/7) * log(3/7))
= 0.985
Entropy(2) = - ((0/3) * log (0/3) + (3/3) * log(3/3))
= 0

Gain in entropy = 0.970 – ((7/10) * 0.985 + (3/10) * 0)
= 0.970 – 0.689
= 0.281

Q3: What is the gain in entropy after splitting on B?

Answer)

Decision tree after splitting on B:

Class Label	Count of Class label
+	4
-	6

T F

1)

Class Label	Count of Class label
+	3
-	1

2)

Class Label	Count of Class label
+	1
-	5

Gain in entropy after splitting = Entropy before splitting – $\sum_{i=1}^{k}((n_i/n) * Entropy(i))$

Where n_i is number of records in partition i.

Entropy before splitting = 0.970 (as calculated in last question)

Entropy(1) = - ((3/4) * log (3/4) + (1/4) * log(1/4))
= 0.811
Entropy(2) = - ((1/6) * log (1/6) + (5/6) * log(5/6))
= 0.65

Gain in entropy = 0.970 – ((4/10) * 0.811 + (6/10) * 0.65)
= 0.970 – 0.7144
= 0.257

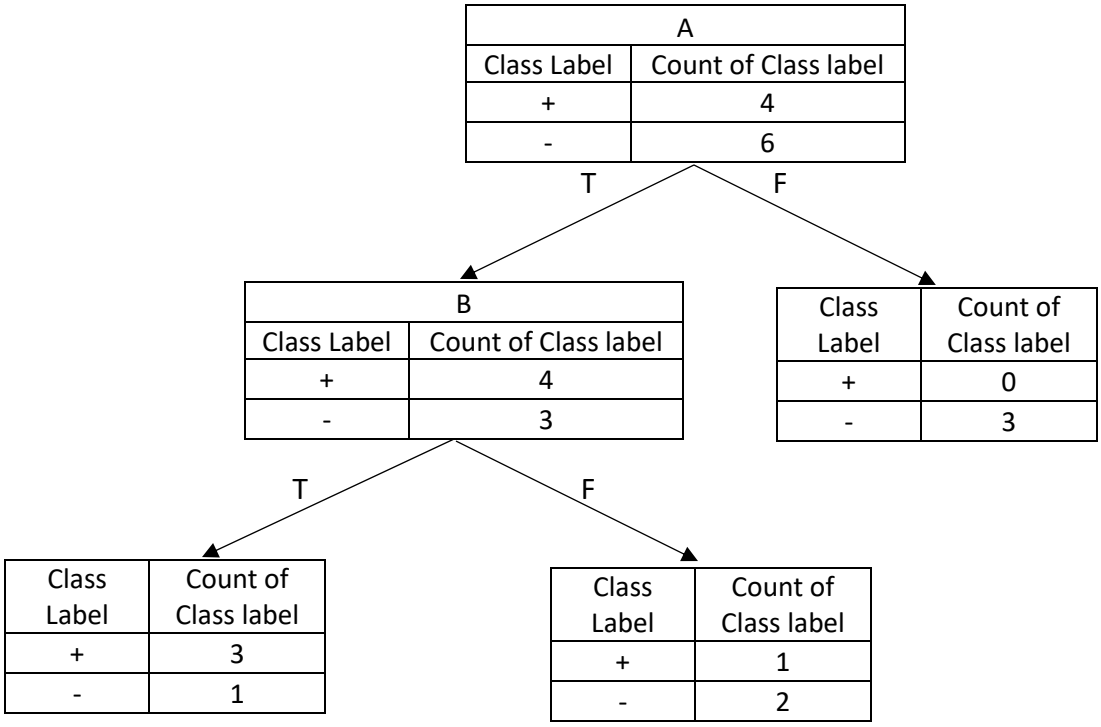
Q4: Which attribute would the decision tree choose?

Answer)

As the gain in entropy after splitting on A (i.e., 0.281) is more than the gain in entropy after splitting on B (i.e., 0.257), therefore attribute A would be chosen by the decision tree.

Q5: Draw the full decision tree that would be learned for this dataset. You do not need to show any calculations. (We want to split first on the variable which maximizes the information gain until there are no nodes with two class labels.)

Answer)



Task 4: Please answer and explain.

Q1: Are decision trees a linear classifier?

Answer)

Decision trees are used to fit linearly inseparable datasets. A linearly inseparable data is the one where the datapoints cannot be separated in a single line, as opposed to linearly separable data where single line is enough to separate.

Q2: What are the weaknesses of decision trees?

Answer)

Weaknesses of the decision tree are as follows:

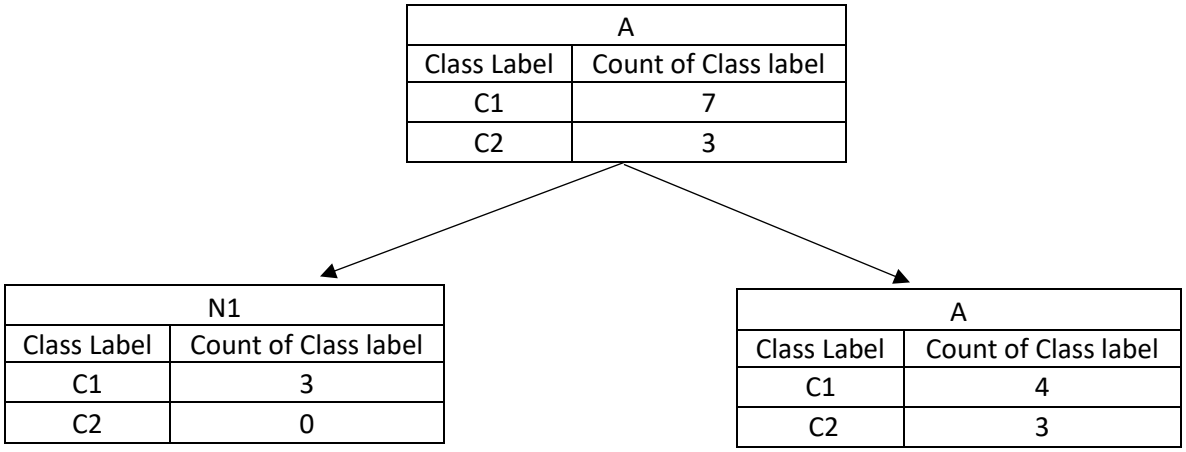
1) It may lead to overfitting of the data. It tends to split a node into may partitions, to make each partition pure.
2) Small change in the data can lead to large structural change in the tree. Addition of single data point can lead to complete change in the structure of the tree.
3) Large datasets may grow a complex tree and can lead to overfitting. Therefore, decision tree is not suitable for large datasets.

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

Answer)

No, Misclassification errors are not better than Gini index as the splitting for decision trees. A small change in the dataset will reflect better in Gini as compared to misclassification error as the impurity vs probability curve is smooth for Gini. Gini is more sensitive than misclassification.

Example:



A	
Class Label	Count of Class label
C1	7
C2	3

N1	
Class Label	Count of Class label
C1	3
C2	0

A	
Class Label	Count of Class label
C1	4
C2	3

	A	N1	N2
Gini Error	0.42	0	0.489
Misclassification Error	0.3	0	0.428

$Gain_{Gini} = 0.42 – (0.489*0.7) = 0.07$

$Gain_{Misclassification} = 0.3 – (0.428 * 0.7) = 0$

Using Misclassification, we can observe no gain after splitting as compared with Gini error, where gain is 0.07.

Please submit a PDF report. In your report, please answer each question with your explanations, plots, results in brief. DO NOT paste your code or snapshot into the PDF. At the end of your PDF, please include a website address (e.g., Github, Dropbox, OneDrive, GoogleDrive) that can allow the TA to read your code.