CHAPTER_8_EXERCISES

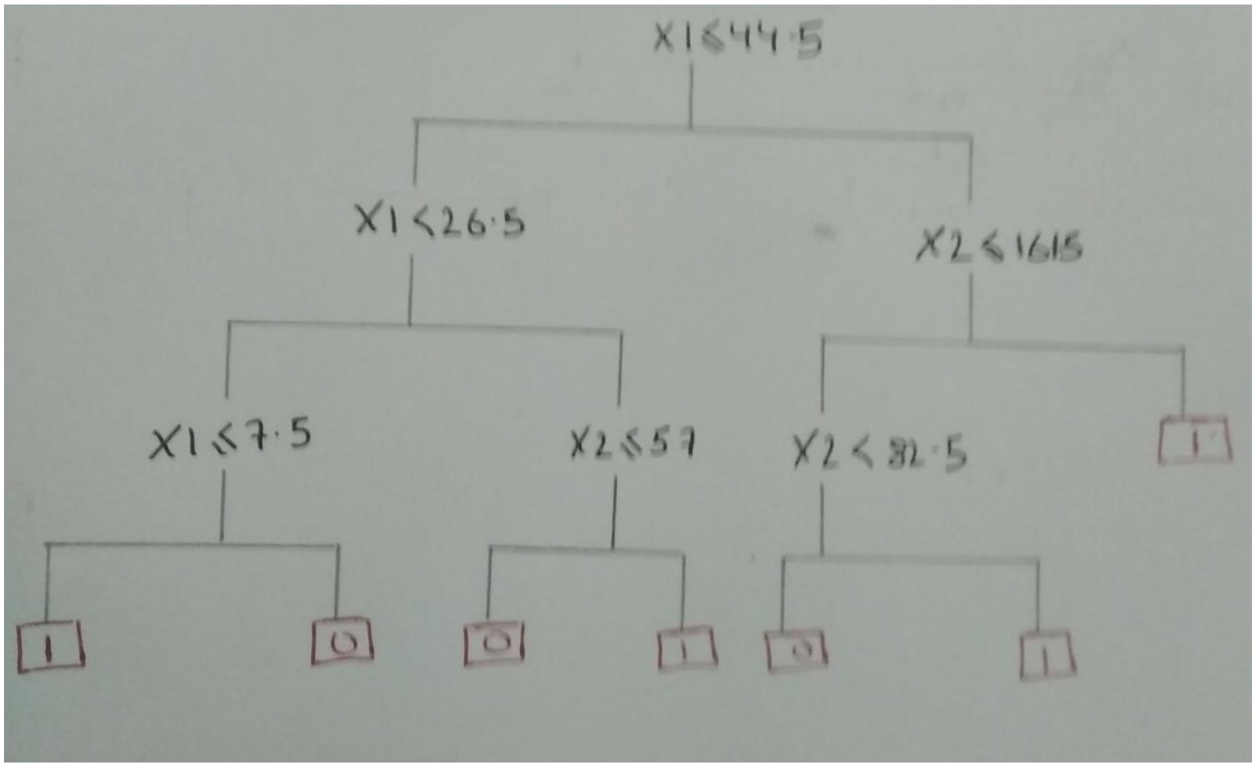
CONCEPTUAL

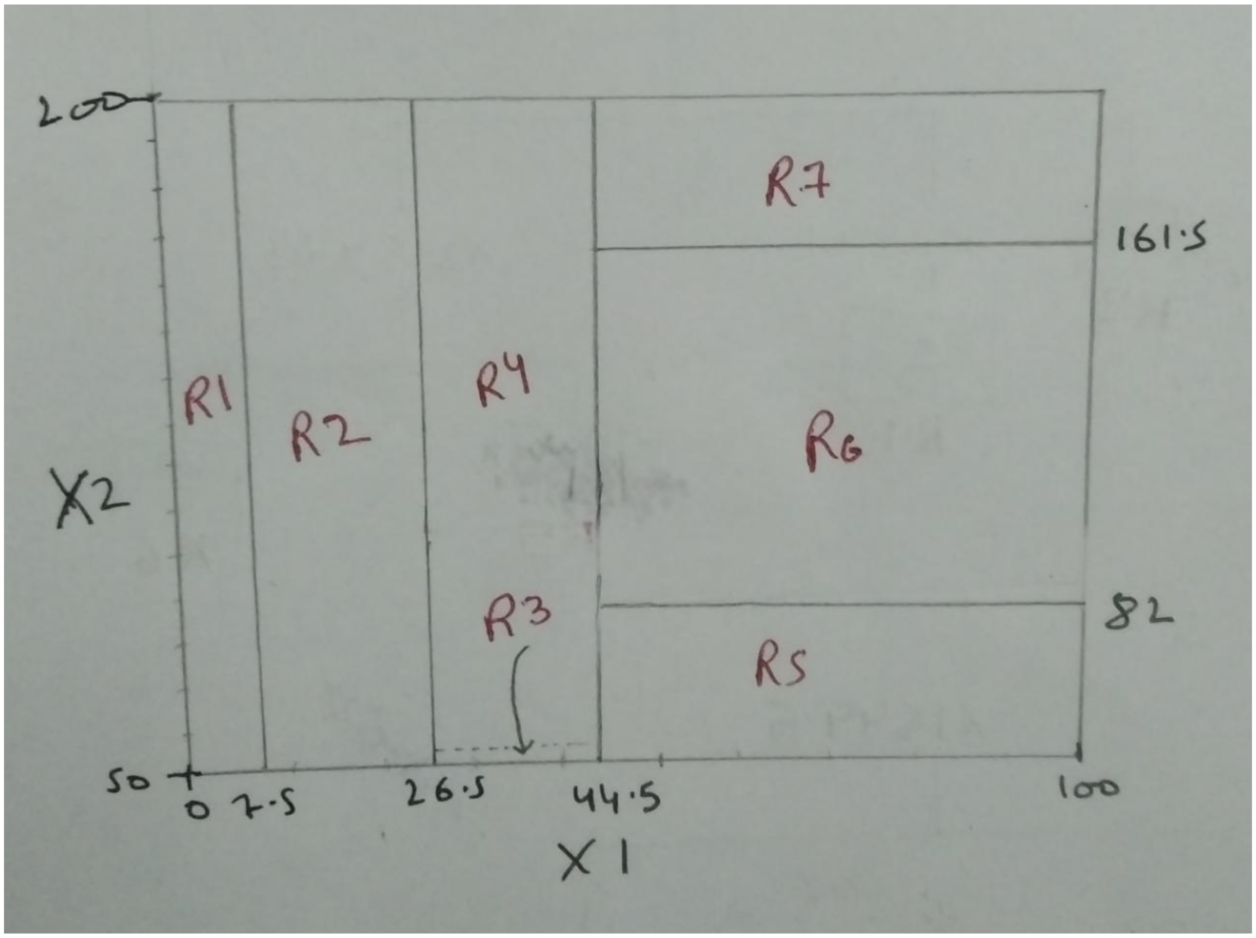
1. Draw an example (of your own invention) of a partition of two dimensional feature space that could result from recursive binary splitting. Your example should contain at least six regions. Draw a decision tree corresponding to this partition. Be sure to label all aspects of your figures, including the regions R1, R2,..., the cutpoints t1, t2,..., and so forth.

```
In [77]: np.random.seed(4)
         X1 = np.random.choice(np.arange(0,101),size = (100,1),replace = False)
         X2 = np.random.choice(np.arange(50,201),size = (100,1),replace = False)

         y = np.random.choice([0,1],size = 100,replace = True)
```

```
In [78]: X = np.concatenate((X1,X2),axis = 1)
         X.shape

Out[78]: (100, 2)
```

```
In [79]: clf = DecisionTreeClassifier(max_depth = 3)
         clf.fit(X,y)
         print(clf.score(X,y))

         0.72
```





2. It is mentioned in Section 8.2.3 that boosting using depth-one trees (or stumps) leads to an additive model: that is, a model of the form

f(X) = p j = 1 fj (Xj).

Explain why this is the case. You can begin with (8.12) in Algorithm 8.2.

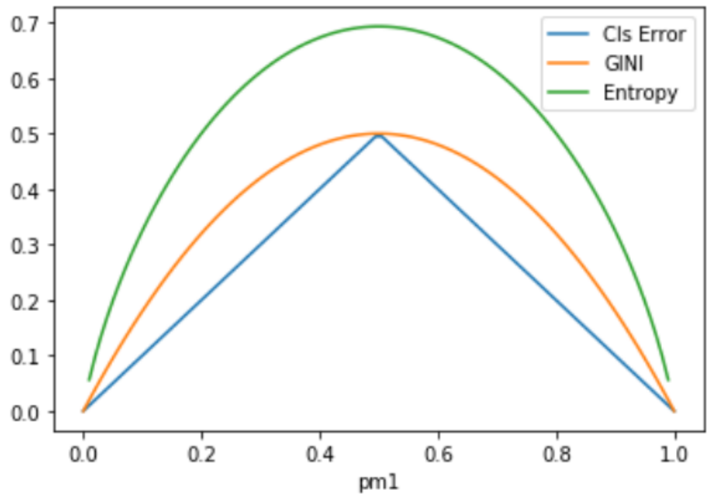
In algorithm 8.2, we describe the algorithm behind boosting. This case we are discussing about boosting when d = 1. When d = 1, we are fitting depth 1 tree in each iteration, since there are p different trees that can be fitted, the iteration will go from 1 to p.

In each iteration we will fit a stump, in the next step when we add a shrunked version of the tree, we are going to add the stump only, (because it cant be shrunk more), and hence the final model will be additive.

3. Consider the Gini index, classification error, and cross-entropy in a simple classification setting with two classes. Create a single plot that displays each of these quantities as a function of ˆpm1. The xaxis

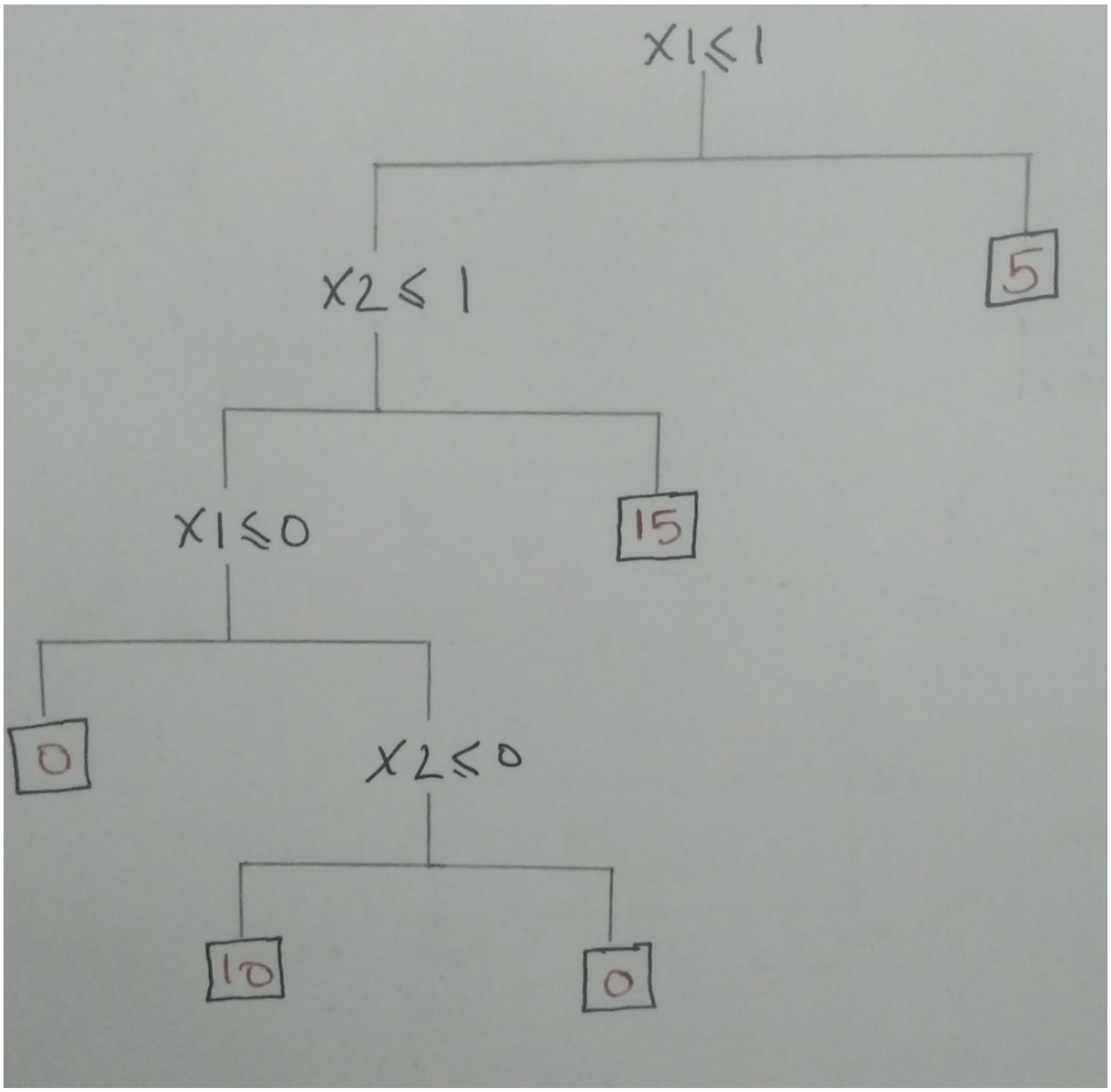
should display ˆpm1, ranging from 0 to 1, and the y-axis should display the value of the Gini index, classification error, and entropy

```
In [21]: def classification_error(x):
             return x if x < 0.5 else (1 - x)
         cls_error = list(map(classification_error,pm1))
         gini = pm1*(1 - pm1) + pm2*(1 - pm2)
         entropy = -( pm1*np.log(pm1) + pm2*np.log(pm2))

         plt.plot(pm1,cls_error,label = 'Cls Error')
         plt.plot(pm1,gini,label = 'GINI')
         plt.plot(pm1,entropy,label = 'Entropy')
         plt.xlabel('pm1')
         plt.legend()

Out[21]: <matplotlib.legend.Legend at 0x22ea2c5fba8>
```

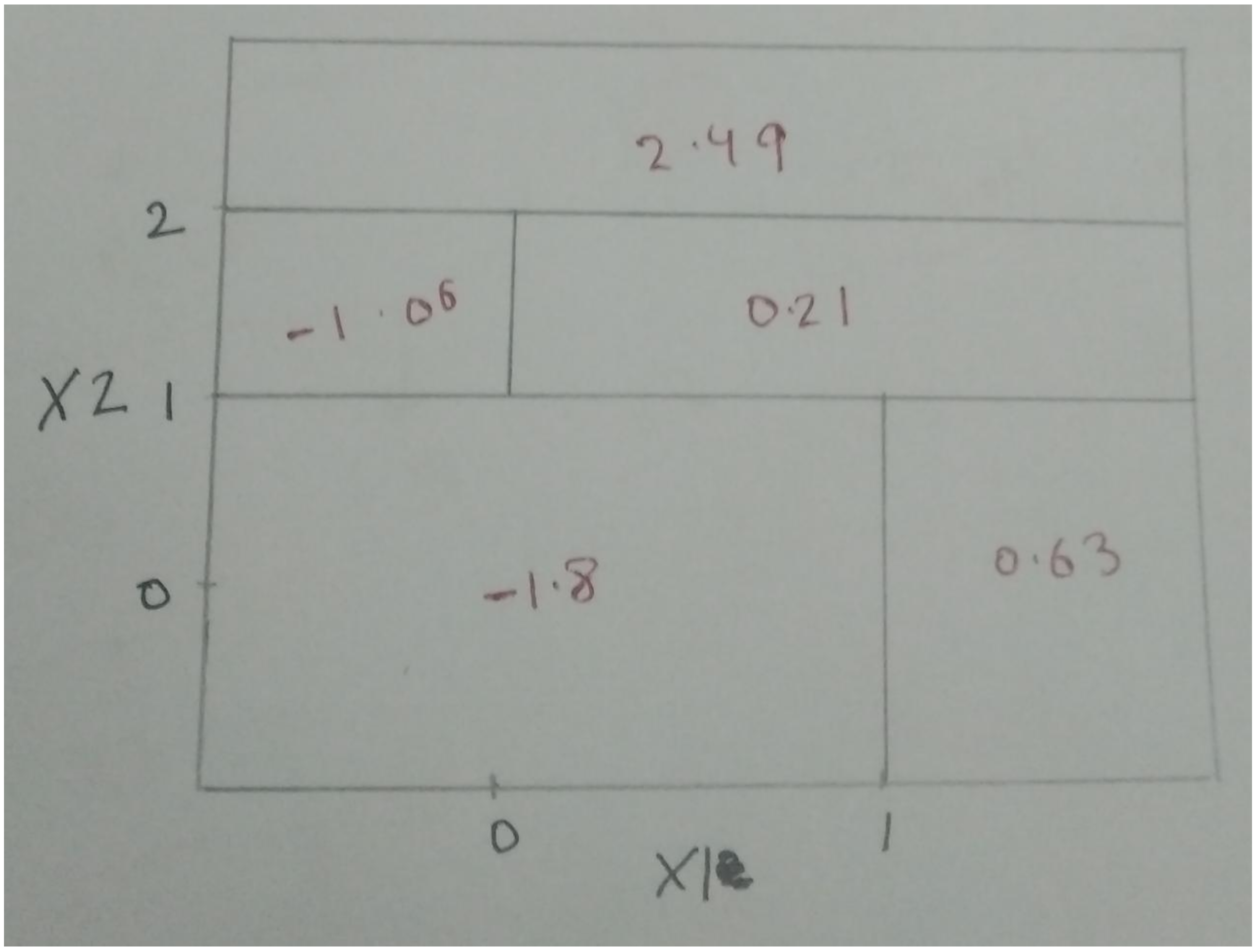


4. This question relates to the plots in Figure 8.12.

(a) Sketch the tree corresponding to the partition of the predictor space illustrated in the left-hand panel of Figure 8.12. The numbers inside the boxes indicate the mean of Y within each region.



(b) Create a diagram similar to the left-hand panel of Figure 8.12, using the tree illustrated in the right-hand panel of the same figure. You should divide up the predictor space into the correct regions, and indicate the mean for each region



5. Suppose we produce ten bootstrapped samples from a data set containing red and green classes. We then apply a classification tree to each bootstrapped sample and, for a specific value of X, produce 10 estimates of

P(Class is Red|X): 0.1, 0.15, 0.2, 0.2, 0.55, 0.6, 0.6, 0.65, 0.7, and 0.75.

There are two common ways to combine these results together into a single class prediction. One is the majority vote approach discussed in this chapter. The second approach is to classify based on the average probability. In this example, what is the final classification under each of these two approaches?

If we look at majority vote approach, then 6 out of 10 probabilities have P(Class is Red|X) > 0.5, hence in this case the value of P(Class is Red|X) will be 0.6. The class predicted is Red class.
For the average probability, we will take the average of all the probabilities here which will be 0.45, hence the class predicted in this case will be Green class.

6. Provide a detailed explanation of the algorithm that is used to fit a regression tree.

Please refer to section 8.1.1 of the book. ☺