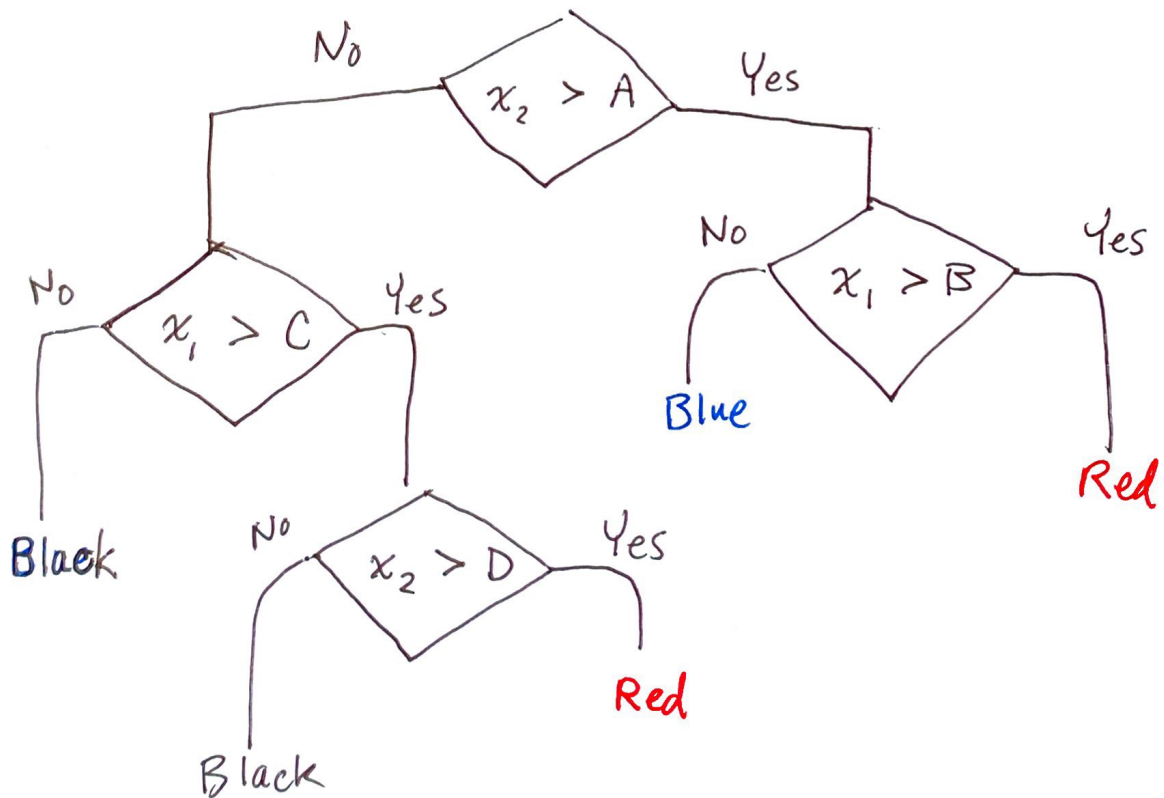
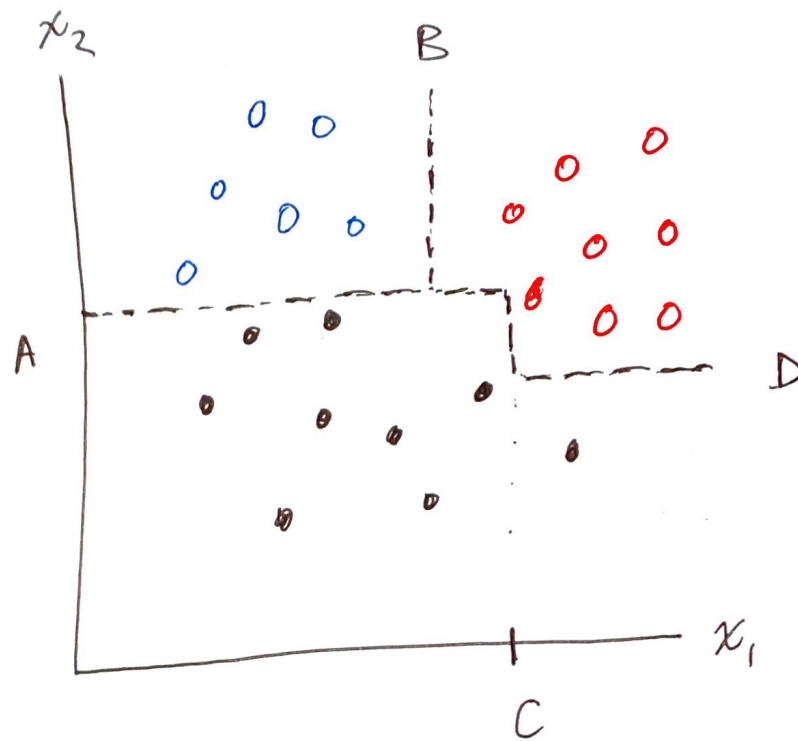


Decision trees



Read in group Data :

```
>> groupData = readtable ("groupData.csv")
```

<u>x</u>	<u>y</u>	<u>group</u>
0.604	7.111	{ 'A' }
	⋮	
1.45	2.093	{ 'B' }
	⋮	
4.978	9.221	{ 'C' }

Convert the text labels for the groups
into a categorical array:

```
>> groupData.group = categorical (...  
    groupData.group)
```

Partition the data into training:
and test sets

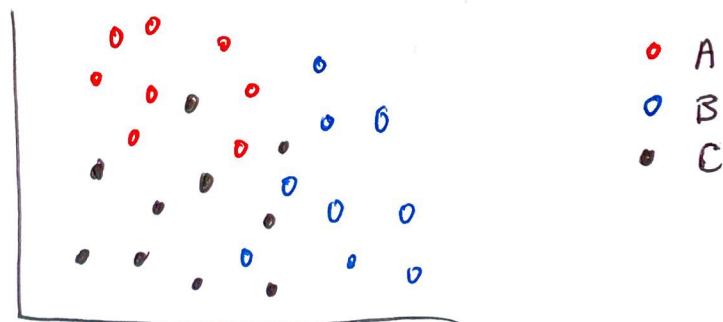
```
>> rng(0) % seed random number generator  
>> cvpt = cvpartition(groupData.group,  
    "Holdout", 0.35);
```

```
>> dataTrain = groupData(training(cvpt),:)
```

```
>> dataTest = groupData(test(cvpt),:)
```

Plot the grouped data:

```
>> gscatter(groupData.x, groupData.y, ...  
    groupData.group)
```



Use the training data to construct
a classification tree named mdl

where the response variable is named "group"

```
>> mdl = fitctree(dataTrain, "group")
```

Calculate loss (error) against test data:

```
>> errTree = loss(mdl, dataTest)
```

```
>> disp("Classification Tree Loss: " + errTree)
```

Classification
Tree Loss: 0.12593

You can change the level of branching of the classification tree using the `prune` function. Prune the tree to level one and calculate the loss:

```
>> mdlPruned = prune(mdl, "Level", 1)
>> errPruned = loss(mdlPruned, dataTest)
```

`errPruned = 0.2421` % worse

Find the predicted groups for the test data.

```
>> label = predict (mdl, dataTest)
```

```
>> gscatter (dataTest.x, dataTest.y, label)
```

 ↗ ↗ ↗
 predictor predictor response