

```
1 package edu.caltech.cs2.lab03;
2 import java.util.*;
3
4 public class DecisionTree {
5     private final DecisionTreeNode root;
6
7     public DecisionTree(DecisionTreeNode root) {
8         this.root = root;
9     }
10
11     public String predict(Dataset.Datapoint point) {
12         return this.predictHelper(point, this.root);
13     }
14
15     private String predictHelper(Dataset.Datapoint point, DecisionTreeNode curr) {
16         // Node is an outcome node
17         if (curr.isLeaf()) {
18             return ((OutcomeNode) curr).outcome;
19         }
20
21         // Node is an attribute node
22         String treeAttribute = ((AttributeNode) curr).attribute;
23         String pointFeature = point.attributes.get(treeAttribute);
24         curr = ((AttributeNode) curr).children.get(pointFeature);
25
26         return predictHelper(point, curr);
27     }
28
29     public static DecisionTree id3(Dataset dataset, List<String> attributes) {
30         return new DecisionTree(id3helper(dataset, attributes));
31     }
32
33     private static DecisionTreeNode id3helper(Dataset data, List<String> attributes) {
34         // If all remaining data points have same outcome,
35         // return OutcomeNode with this outcome
36         String outcome = data.pointsHaveSameOutcome();
37         if (outcome.length() > 0) {
38             return new OutcomeNode(outcome);
39         }
40
41         // If no attributes left, return OutcomeNode with most common outcome
42         if (attributes.size() == 0) {
43             String mostCommonOutcome = data.getMostCommonOutcome();
44             return new OutcomeNode(mostCommonOutcome);
45         }
46
47         // Find the lowest entropy attribute, a.
48         String a = data.getAttributeWithMinEntropy(attributes);
49
50
51
52         // Create new list with all attributes, except a
53         List<String> newAttributes = new ArrayList<>();
```

```

54     for (String attribute : attributes) {
55         if (!attribute.equals(a)) {
56             newAttributes.add(attribute);
57         }
58     }
59
60     // For each feature, f, of a:
61     List<String> lowestEntropyFeatures = data.getFeaturesForAttribute(a);
62     Map<String, DecisionTreeNode> attributeToNode = new HashMap<>();
63     for (String f : lowestEntropyFeatures) {
64
65         // Find all the data points that have the feature f
66         Dataset newDSet = data.getPointsWithFeature(f);
67
68         // If there are none
69         if (newDSet.isEmpty()) {
70
71             // Guess! Make an outcome child with the most common outcome.
72             String mostCommonOutcome = newDSet.getMostCommonOutcome();
73             OutcomeNode newNode = new OutcomeNode(mostCommonOutcome);
74             attributeToNode.put(f, newNode);
75
76             // Otherwise:
77             } else {
78
79                 // Use the data! Recursively make a child out of the remaining data points,
80                 // using all attributes except for a.
81                 attributeToNode.put(f, id3helper(newDSet, newAttributes));
82             }
83         }
84
85         // Return the attribute node with the children generated above.
86         return new AttributeNode(a, attributeToNode);
87     }
88 }

```