
Making a DecisionTree Class

```
classdef DecisionTree
    properties
        node = NaN;
        left = NaN;
        right = NaN;
        isLeaf = false
        depth = 0;
    end
    methods

        function obj = train(obj,data,labels)
            obj.depth = obj.depth+1;
            if sum(labels) >= 0.99*length(labels)
                obj.isLeaf = true;
                obj.node = 1;
                obj.left = NaN;
                obj.right = NaN;
                return
            else if sum(labels) <= 0.01*length(labels)
                obj.isLeaf = true;
                obj.node = 0;
                obj.left = NaN;
                obj.right = NaN;
                return
            else
                [index,threshold] = Segmentor(data,labels);
                obj.node = [index,threshold];
                if obj.depth <= 25
                    obj.right = obj.train(data(data(:,index) >
threshold,:),labels(data(:,index) > threshold));
                    obj.left = obj.train(data(data(:,index) <=
threshold,:),labels(data(:,index) <= threshold));
                else
                    obj.isLeaf = true;
                    obj.node = mode(labels);
                    obj.left = NaN;
                    obj.right = NaN;
                end
            end
        end

        function label = predict(obj,data,column_names)
            if obj.isLeaf == true
                label = obj.node;
                display(label,'The assigned Label');
            else
                split_value = obj.node;
                index = split_value(1);
```

```
        threshold = split_value(2);
        formatSpec= '(%s) <= %d. \n';
        input  = column_names{index};
        dis = sprintf(formatSpec,input,threshold);
        disp(dis);
        if data(:,index) <= threshold
            obj = obj.left;
            label = obj.predict(data(data(:,index) <=
threshold,,:),column_names);
        else
            dt = obj.right;
            label = dt.predict(data(data(:,index) >
threshold,,:),column_names);
        end
    end
end

ans =

    DecisionTree with properties:

    node: NaN
    left: NaN
    right: NaN
    isLeaf: 0
    depth: 0
```

Published with MATLAB® R2015b