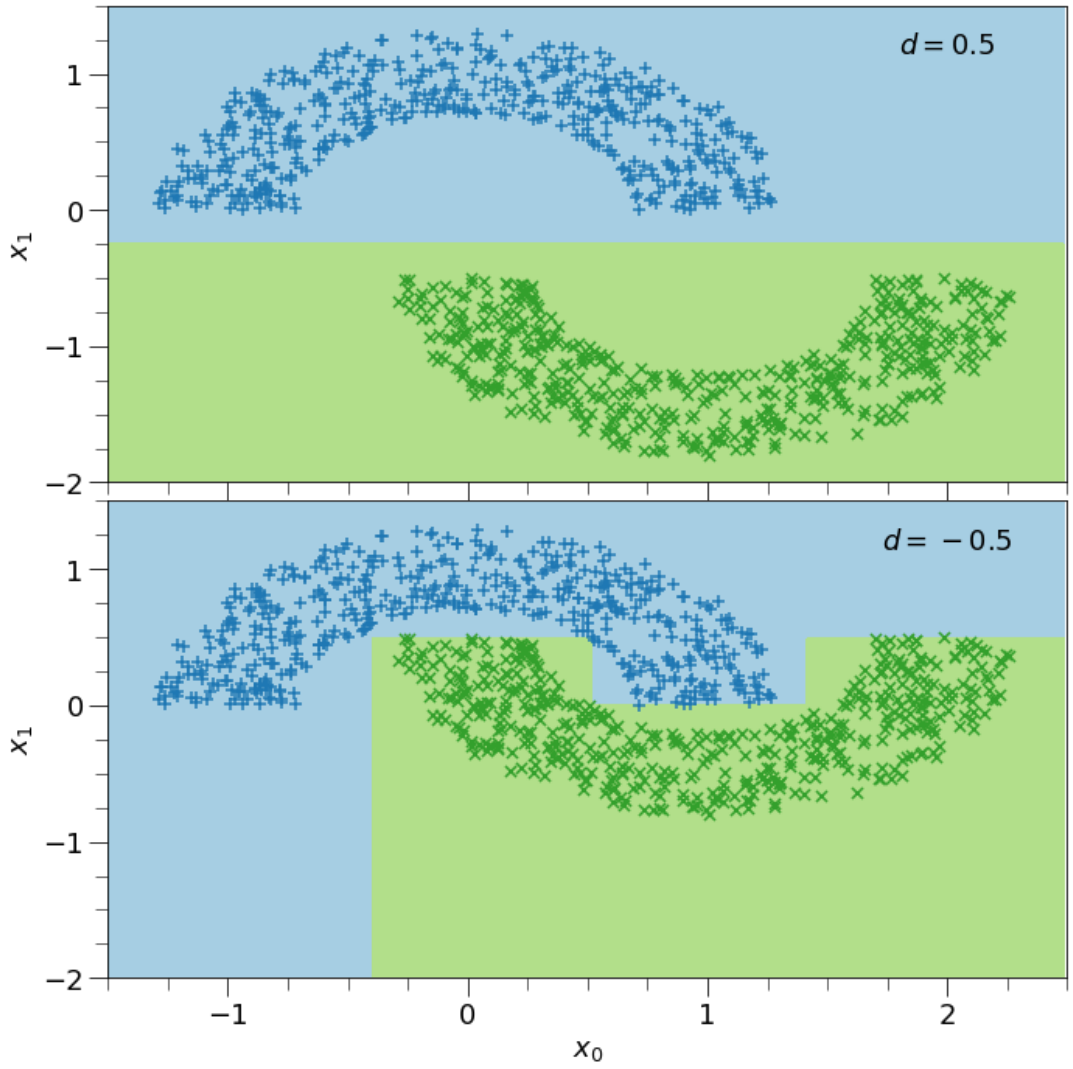
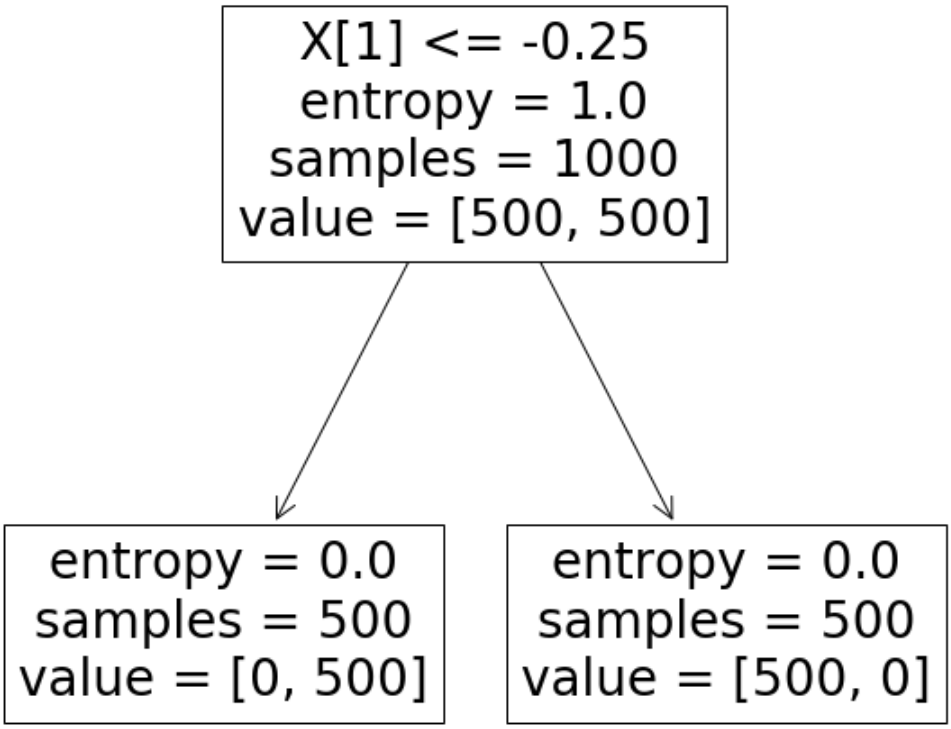
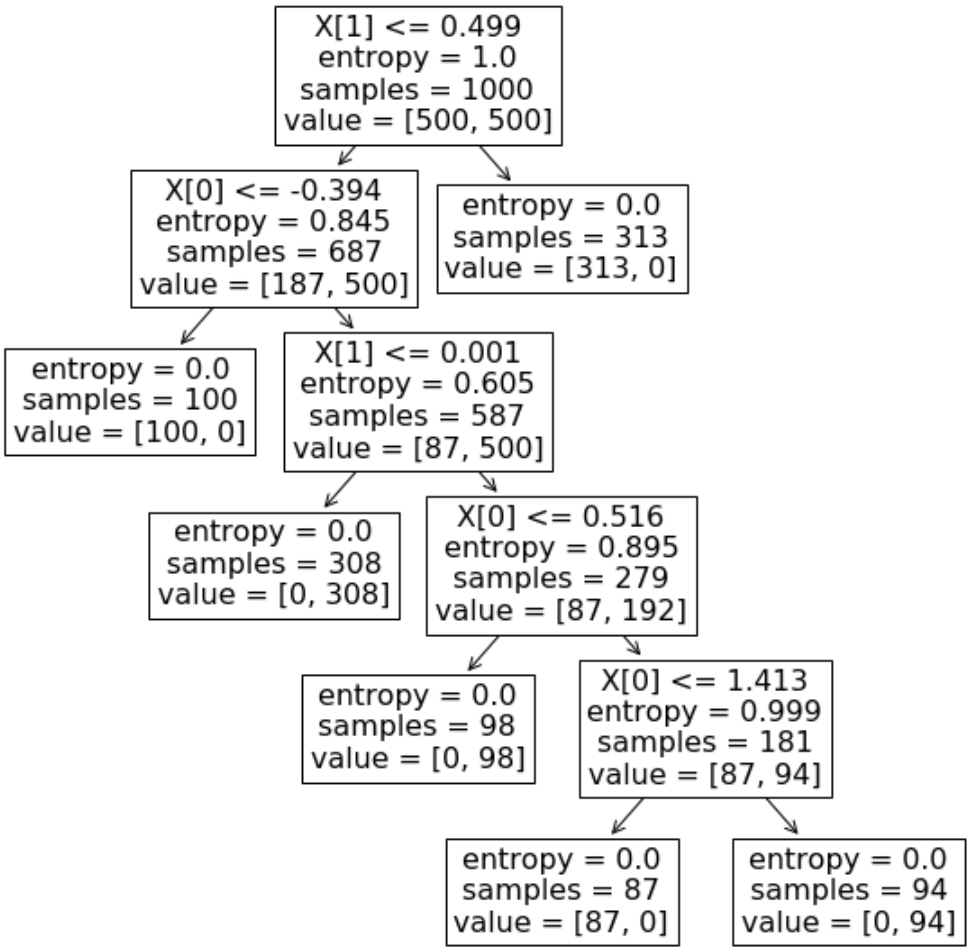
In [3]:
```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""
Created on Tue Dec  3 21:49:59 2019

@author: jorgeagr
"""

import numpy as np
from sklearn.tree import DecisionTreeClassifier, plot_tree
from requiredFunctions.doubleMoon import doubleMoon
import matplotlib as mpl
import matplotlib.ticker as mtick
import matplotlib.pyplot as plt

width = 10
height = 10

mpl.rcParams['figure.figsize'] = (width, height)
mpl.rcParams['font.size'] = 18
mpl.rcParams['figure.titlesize'] = 'small'
mpl.rcParams['legend.fontsize'] = 'small'
mpl.rcParams['xtick.major.size'] = 12
mpl.rcParams['xtick.minor.size'] = 8
mpl.rcParams['xtick.labelsize'] = 18
mpl.rcParams['ytick.major.size'] = 12
mpl.rcParams['ytick.minor.size'] = 8
mpl.rcParams['ytick.labelsize'] = 18

cmap = plt.get_cmap('Paired')
cmap_scatter = mpl.colors.ListedColormap(cmap((1, 3, 5)))
cmap_contour = mpl.colors.ListedColormap(cmap((0, 2)))

N = 1000
r = 1
w = 0.6
d_vals = [0.5, -0.5]
titles = ['b', 'c']
```

In [4]:
```
figds, axds = plt.subplots(nrows=2, sharex=True)
for i, d in enumerate(d_vals):
    # Part B
    data = doubleMoon(N, w, r, d, seed=0)
    x_train = data[:,:2]
    y_train = data[:,-1]

    tree = DecisionTreeClassifier(criterion='entropy', random_state=1)
    tree.fit(x_train, y_train)

    # Predict for the data points and assign indeces of what's right a
nd wrong
    y_pred = tree.predict(x_train)
    y_right = np.where(y_pred == y_train)[0]
    y_wrong = np.where(y_pred != y_train)[0]
    blue_ind = y_right[np.where(y_right<N//2)]
    green_ind = y_right[np.where(y_right>=N//2)]

    # Make mesh for decision regions and predict for the points within
    x0_min, x0_max = -1.5, 2.5
    x1_min, x1_max = -2, 1.5
    xx0, xx1 = np.meshgrid(np.arange(x0_min, x0_max, 0.01),
                           np.arange(x1_min, x1_max, 0.01))
    cc = tree.predict(np.c_[xx0.ravel(), xx1.ravel()]).reshape(xx0.sha
pe)

    # Plotting of decision regions and data points
    axds[i].contourf(xx0, xx1, cc, cmap=cmap_contour)
    axds[i].scatter(x_train[:,0][blue_ind], x_train[:,1][blue_ind], 5
0,
                    c=[cmap_scatter(0)], marker='+')
    axds[i].scatter(x_train[:,0][green_ind], x_train[:,1][green_ind],
50,
                    c=[cmap_scatter(1)], marker='x')
    axds[i].scatter(x_train[:,0][y_wrong], x_train[:,1][y_wrong], 50,
                    c=[cmap_scatter(2)], marker='*')
    axds[i].set_xlim(x0_min, x0_max)
    axds[i].set_ylim(x1_min, x1_max)
    axds[i].xaxis.set_major_locator(mtick.MultipleLocator(1))
    axds[i].xaxis.set_minor_locator(mtick.MultipleLocator(0.25))
    axds[i].yaxis.set_major_locator(mtick.MultipleLocator(1))
    axds[i].yaxis.set_minor_locator(mtick.MultipleLocator(0.25))
    axds[i].text(2, 1.15, r'$d = {}$'.format(d_vals[i]), horizontalali
gnment='center')
    axds[i].set_ylabel(r'$x_1$')

    fig, ax = plt.subplots()
    plot_tree(tree, ax=ax)
    #fig.savefig('../prob1' + titles[i] + '.eps')

axds[1].set_xlabel(r'$x_0$')
figds.tight_layout(h_pad=0)
#figds.savefig('../prob1ds.eps',dpi=500)
```


$d = 0.5$
$d = -0.5$
x_0
x_1
x_1
1
0
−1
−2
−1
0
1
2


X[1] <= -0.25
entropy = 1.0
samples = 1000
value = [500, 500]
entropy = 0.0
samples = 500
value = [0, 500]
entropy = 0.0
samples = 500
value = [500, 0]


X[1] <= 0.499
entropy = 1.0
samples = 1000
value = [500, 500]
X[0] <= -0.394
entropy = 0.845
samples = 687
value = [187, 500]
entropy = 0.0
samples = 313
value = [313, 0]
entropy = 0.0
samples = 100
value = [100, 0]
X[1] <= 0.001
entropy = 0.605
samples = 587
value = [87, 500]
entropy = 0.0
samples = 308
value = [0, 308]
X[0] <= 0.516
entropy = 0.895
samples = 279
value = [87, 192]
entropy = 0.0
samples = 98
value = [0, 98]
X[0] <= 1.413
entropy = 0.999
samples = 181
value = [87, 94]
entropy = 0.0
samples = 87
value = [87, 0]
entropy = 0.0
samples = 94
value = [0, 94]

In [ ]: