

```
In [2]: import numpy as np
np.random.seed(0)
import mltools as ml
import matplotlib.pyplot as plt
import mltools.dtree as dt
%matplotlib inline
#reload(dt)
```

```
In [20]: X = np.genfromtxt("kaggle.X1.train.txt",delimiter=',')
Y = np.genfromtxt("kaggle.Y.train.txt",delimiter=',')
Xe = np.genfromtxt("kaggle.X1.test.txt",delimiter=',')

#curve= np.genfromtxt("data/iris.txt", delimiter=None)
#X = curve[:,0]
#X = X[:,np.newaxis]
#Y = curve[:,1]
```

```
In [4]: Xt, Xv, Yt, Yv = ml.splitData(X,Y,0.75)

lr = dt.treeRegress(Xt,Yt, maxDepth=20)
```

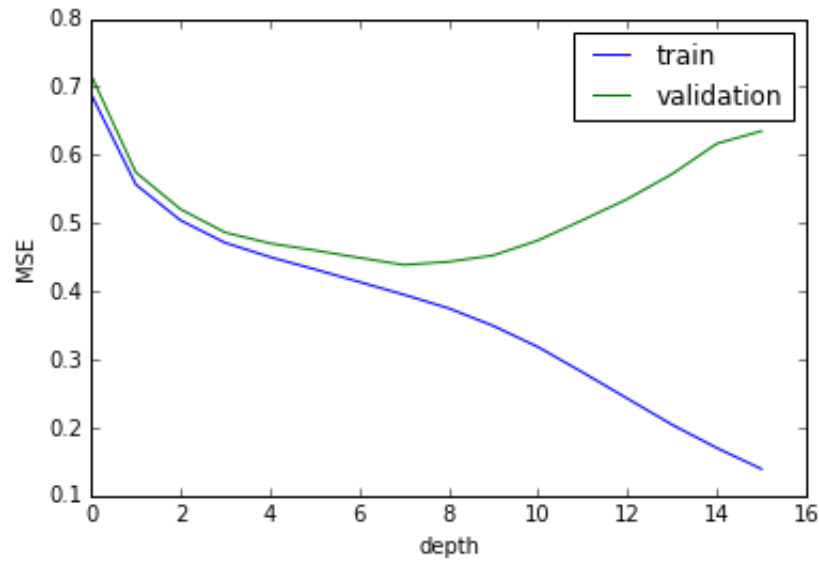
```
In [5]: train_error = np.zeros(shape = (16,1))
valid_error = np.zeros(shape = (16,1))
depth = np.zeros(shape = (16,1))

for i in range (0,16):
    lr = ml.dtree.treeRegress(Xt, Yt, maxDepth = i)
    train_error[i] = lr.mse(Xt, Yt)
    valid_error[i] = lr.mse(Xv, Yv)
    depth[i] = i
```

```
In [6]: plt.plot(depth, train_error, 'b', depth, valid_error, 'g')
plt.legend(['train', 'validation'])

plt.xlabel('depth')
plt.ylabel('MSE')
```

Out[6]: <matplotlib.text.Text at 0x12359ecc0>



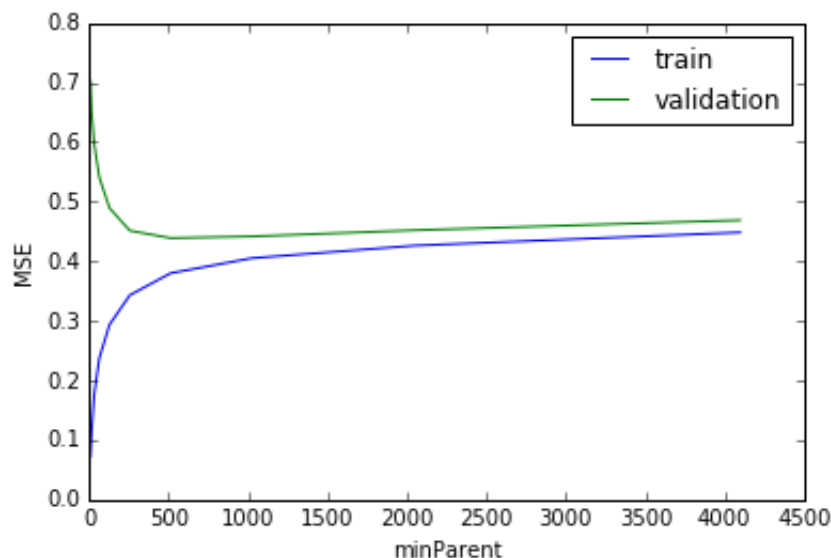
```
In [7]: #minParent
ertc = []
ervc = []
minparents = []

for i in range(3,13):
    lr = ml.dtree.treeRegress(Xt, Yt, maxDepth = 20, minParent = 2*
*i)
    ertc.append( lr.mse(Xt, Yt))
    ervc.append( lr.mse(Xv, Yv))
    minparents.append(2*i)

plt.plot(minparents, ertc , 'b', minparents, ervc , 'g')
plt.legend(['train', 'validation'])

plt.xlabel('minParent')
plt.ylabel('MSE')
```

Out[7]: <matplotlib.text.Text at 0x10a292f60>



```
In [18]: #minScore
ertc = []
ervc = []
minScore = []

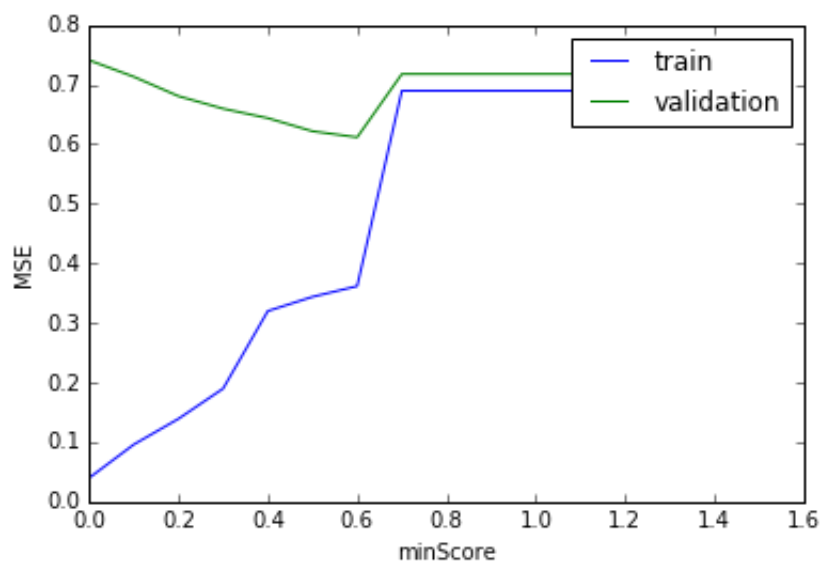
r = [0,.1,.2,.3,.4,.5,.6,.7,.8,.9,1,1.1,1.2,1.3,1.4,1.5]
#r = [.001,.002,.003,.004,.005,.006,.007,.008,.009,.010,.011,.012,.
013,.014,.015,.016,.017,.018
#      ,.019,.20,.021,.022,.023,.024,.025,.026,.027,.028,.029,.3,0.3
1,0.32,0.33,0.34,0.35,.36,.37,.38,.39,.4,
#      0.41,.42,.43,.44,.45,.46,.47,.48,.49,.5]

for i in r:
    lr = ml.dtree.treeRegress(Xt, Yt, maxDepth = 20, minScore= i)
    ertc.append( lr.mse(Xt, Yt))
    ervc.append( lr.mse(Xv, Yv))
    minScore.append(i)

plt.plot(minScore, ertc , 'b', minScore, ervc , 'g')
plt.legend(['train', 'validation'])

plt.xlabel('minScore')
plt.ylabel('MSE')
```

Out[18]: <matplotlib.text.Text at 0x10b82d898>



```
In [27]: lr.train(X,Y, minParent=2**9, maxDepth=7, minScore= 0.4)
         YeHat = lr.predict(Xe)
```

```
In [28]: # Output our predictions to a file:
         fh = open('predict_dtree.csv','w') # open file for upload
         fh.write('ID,Prediction\n') # output header line
         for i,yi in enumerate(YeHat.ravel()):
             fh.write('{}{}\n'.format(i+1,yi)) # output each prediction
         fh.close() # close the file
```

In []: