In [1]:

import pandas as pd

import matplotlib.pyplot as plt

import numpy as np

from sklearn.tree import DecisionTreeClassifier

from sklearn.tree import DecisionTreeRegressor

from sklearn.model\_selection import train\_test\_split

from sklearn.metrics import accuracy\_score

from sklearn.ensemble import RandomForestRegressor

from sklearn import metrics

from sklearn import tree

from io import StringIO

from IPython.display import Image

from sklearn.tree import export\_graphviz

import pydotplus

import pydot

import sys

from sklearn import svm

from sklearn import model\_selection

from statsmodels.tools.eval\_measures import mse

from sklearn.metrics import mean\_squared\_error

from sklearn.linear\_model import Ridge

import statsmodels.api as sm

In [ ]:

data=pd.read\_csv(r'C:\Users\SILA\OneDrive\Desktop\Life Expectancy Data\dataproject.csv')

In [25]:

data.columns

In [26]:

dummies=pd.get\_dummies(data[['county', 'year', 'life\_expectancy', 'population', 'GDP\_mill',

'county.expenditure', 'adult.mort', 'poverty\_rate', 'under\_five.mort',

'HIV\_prev', 'BCG', 'diptheria', 'hepatitis', 'polio', 'expectancy1']])

X2=(dummies-dummies.min())/(dummies.max()-dummies.min())

X=X2.drop(columns=["life\_expectancy"])

Y=X2["life\_expectancy"]

X\_train,X\_test,Y\_train,Y\_test=model\_selection.train\_test\_split(X,Y,test\_size=.3,random\_state=1)

In [27]:

h =svm.SVR(kernel='rbf')

h.fit(X\_train,Y\_train)

Y\_pred=h.predict(X\_test)

print(mse(Y\_pred,Y\_test))

h.score(X\_test,Y\_test)

metrics.explained\_variance\_score(Y\_test,Y\_pred)

In [28]:

model=Ridge()

model.fit(X\_train,Y\_train)

model.score(X\_test,Y\_test)

predicts=model.predict(X\_test)

print(mse(predicts,Y\_test))

metrics.explained\_variance\_score(Y\_test,predicts)

In [53]:

model1=DecisionTreeRegressor()

model1.fit(X\_train,Y\_train)

predictions=model1.predict(X\_test)

print(mse(predictions,Y\_test))

metrics.explained\_variance\_score(Y\_test,predictions)

In [13]:

fig, axes = plt.subplots(nrows = 1,ncols = 1,figsize = (4,4), dpi=300)

tree.plot\_tree(model1,filled=True,rounded=True)

plt.savefig('C:/ProgramData/jupyter/svm.png')

In [113]:

rf= RandomForestRegressor()

rf.fit(X\_train,Y\_train)

rfpredict=rf.predict(X\_test)

print(mse(rfpredict,Y\_test))

metrics.explained\_variance\_score(Y\_test,rfpredict)

In [78]:

In [91]:

dummies1=pd.get\_dummies(data[[ 'life\_expectancy', 'population', 'GDP\_mill',

'county.expenditure', 'adult.mort', 'poverty\_rate', 'under\_five.mort',

'HIV\_prev', 'BCG', 'diptheria', 'hepatitis', 'polio']])

X3=(dummies1-dummies1.min())/(dummies1.max()-dummies1.min())

X2=X3.drop(columns=["life\_expectancy"])

Y2=X3["life\_expectancy"]

X2\_train,X2\_test,Y2\_train,Y2\_test=model\_selection.train\_test\_split(X2,Y2,test\_size=.3,random\_state=1)

In [92]:

X2\_train\_sm=sm.add\_constant(X2\_train)

lm=sm.OLS(Y2\_train,X2\_train\_sm).fit()

lm.params

lm.summary()