

```

import numpy as np
import pandas as pd
from calculate import *
from choose_axis import *

def createTree(dataSet, labels, data_full, labels_full):
    classList = [example[-1] for example in dataSet]
    if classList.count(classList[0]) == len(classList):
        return classList[0]
    if len(dataSet[0]) == 1:
        return extraChoose(classList)
    bestFeat = chooseAxis(dataSet, labels)
    bestFeatLabel = labels[bestFeat]
    myTree = {bestFeatLabel: {}}
    featValues = [example[bestFeat] for example in dataSet]
    uniqueVals = set(featValues)
    if type(dataSet[0][bestFeat]).__name__ == 'str':
        currentlabel = labels_full.index(labels[bestFeat])
        featValuesFull = [example[currentlabel] for example in data_full]
        uniqueValsFull = set(featValuesFull)
    del(labels[bestFeat])
    for value in uniqueVals:
        subLabels = labels[:]
        if type(dataSet[0][bestFeat]).__name__ == 'str':
            uniqueValsFull.remove(value)
        myTree[bestFeatLabel][value] = createTree(normalData(
            dataSet, bestFeat, value), subLabels, data_full, labels_full)
    if type(dataSet[0][bestFeat]).__name__ == 'str':
        for value in uniqueValsFull:
            myTree[bestFeatLabel][value] = extraChoose(classList)
    return myTree

```

```

def main():
    df = pd.read_csv('input.csv')
    data = df.values[:, 1:].tolist()
    data_full = data[:]
    labels = df.columns.values[1:-1].tolist()
    labels_full = labels[:]
    tree = createTree(data, labels, data_full, labels_full)
    print(tree)
    return tree

```