```
        n = X_train.shape[0]

        count = 0
        for i in range(n):
            if self.predict(X_train[i]) == labels[i]:
                count += 1

        print("The decision tree is %d percent accurate on %d training data" % ((co
unt / n) * 100, n))

        n = X_test.shape[0]

        count = 0
        for i in range(n):
            if self.predict(X_test[i]) == y_test[i]:
                count += 1

        print("The decision tree is %d percent accurate on %d test data" % ((count
/ n) * 100, n))

        return count / n
```

Run the tree

Try different values of K (depth of tree a.k.a. number of features the tree will split on) and compare the performance. Which feature gives the largest information gain? Which feature is the least useful for the decision tree?

In [103]:

```
def featureList(node, features):
    if node is not None and node.feature is not None:
        features.append(node.feature)
        featureList(node.left,features)
        featureList(node.right,features)
```

In [104]:

```
X_train, y_train, X_test, y_test = import_data(split=0.8)

tree = DecisionTree(K=10, verbose=False)
tree.buildTree(X_train, y_train)
tree.homework_evaluate(X_train, y_train, X_test, y_test)
features = []
featureList(tree.root,features)
print(features)
```

```
data imported

//anaconda3/lib/python3.7/site-packages/ipykernel_launcher.py:130: Runt
imeWarning: divide by zero encountered in log
//anaconda3/lib/python3.7/site-packages/ipykernel_launcher.py:130: Runt
imeWarning: invalid value encountered in multiply

The decision tree is 99 percent accurate on 1097 training data
The decision tree is 98 percent accurate on 275 test data
[0, 1, 2, 0, 1, 0, 0, 0, 0, 0, 0, 2, 1, 0, 0, 3, 2, 0, 0, 0, 0, 1, 2,
0, 0, 0, 0]
```

In [105]:

```
k_list = range(1, 7)
for i in k_list:
    tree = DecisionTree(K=i, verbose=False)
    tree.buildTree(X_train, y_train)
    print('K={}:'.format(i))
    tree.homework_evaluate(X_train, y_train, X_test, y_test)
    features = []
    featureList(tree.root,features)
    #tree.printTree()
    print(features)
    print('The feature gives largest information gain is {}'.format(max(set(feature
s), key = features.count)))
    print('The feature gives least information gain is {}'.format(min(set(features
), key = features.count)))
```

```
//anaconda3/lib/python3.7/site-packages/ipykernel_launcher.py:130: Runt
imeWarning: divide by zero encountered in log
//anaconda3/lib/python3.7/site-packages/ipykernel_launcher.py:130: Runt
imeWarning: invalid value encountered in multiply

K=1:
The decision tree is 89 percent accurate on 1097 training data
The decision tree is 89 percent accurate on 275 test data
[0, 1, 0]
The feature gives largest information gain is 0
The feature gives least information gain is 1
K=2:
The decision tree is 95 percent accurate on 1097 training data
The decision tree is 93 percent accurate on 275 test data
[0, 1, 2, 0, 0, 2, 0]
The feature gives largest information gain is 0
The feature gives least information gain is 1
K=3:
The decision tree is 98 percent accurate on 1097 training data
The decision tree is 96 percent accurate on 275 test data
[0, 1, 2, 0, 1, 0, 0, 0, 0, 2, 1, 3, 0]
The feature gives largest information gain is 0
The feature gives least information gain is 3
K=4:
The decision tree is 98 percent accurate on 1097 training data
The decision tree is 97 percent accurate on 275 test data
[0, 1, 2, 0, 1, 0, 0, 0, 0, 0, 0, 2, 1, 0, 0, 3, 2, 1, 0]
The feature gives largest information gain is 0
The feature gives least information gain is 3
K=5:
The decision tree is 99 percent accurate on 1097 training data
The decision tree is 98 percent accurate on 275 test data
[0, 1, 2, 0, 1, 0, 0, 0, 0, 0, 0, 2, 1, 0, 0, 3, 2, 0, 0, 1, 2, 0, 0]
The feature gives largest information gain is 0
The feature gives least information gain is 3
K=6:
The decision tree is 99 percent accurate on 1097 training data
The decision tree is 98 percent accurate on 275 test data
[0, 1, 2, 0, 1, 0, 0, 0, 0, 0, 0, 2, 1, 0, 0, 3, 2, 0, 0, 0, 0, 1, 2,
0, 0, 0, 0]
The feature gives largest information gain is 0
The feature gives least information gain is 3
```

In [106]:

```
k_list = range(1, 7)
for i in k_list:
    tree = DecisionTree(K=i, verbose=False)
    tree.buildTree(X_train, y_train,metric = "gini")
    print('K={}:'.format(i))
    tree.homework_evaluate(X_train, y_train, X_test, y_test)
    features = []
    featureList(tree.root,features)
    print(features)
    print('The feature gives largest information gain is {}'.format(max(set(feature
s), key = features.count)))
    print('The feature gives least information gain is {}'.format(min(set(features
), key = features.count)))
```

```
//anaconda3/lib/python3.7/site-packages/ipykernel_launcher.py:130: Runt
imeWarning: divide by zero encountered in log
//anaconda3/lib/python3.7/site-packages/ipykernel_launcher.py:130: Runt
imeWarning: invalid value encountered in multiply

K=1:
The decision tree is 89 percent accurate on 1097 training data
The decision tree is 89 percent accurate on 275 test data
[0, 1, 0]
The feature gives largest information gain is 0
The feature gives least information gain is 1
K=2:
The decision tree is 95 percent accurate on 1097 training data
The decision tree is 93 percent accurate on 275 test data
[0, 1, 2, 0, 0, 2, 0]
The feature gives largest information gain is 0
The feature gives least information gain is 1
K=3:
The decision tree is 98 percent accurate on 1097 training data
The decision tree is 96 percent accurate on 275 test data
[0, 1, 2, 0, 1, 0, 0, 0, 0, 2, 1, 3, 0]
The feature gives largest information gain is 0
The feature gives least information gain is 3
K=4:
The decision tree is 98 percent accurate on 1097 training data
The decision tree is 97 percent accurate on 275 test data
[0, 1, 2, 0, 1, 0, 0, 0, 0, 0, 0, 2, 1, 0, 0, 3, 2, 1, 0]
The feature gives largest information gain is 0
The feature gives least information gain is 3
K=5:
The decision tree is 99 percent accurate on 1097 training data
The decision tree is 98 percent accurate on 275 test data
[0, 1, 2, 0, 1, 0, 0, 0, 0, 0, 0, 2, 1, 0, 0, 3, 2, 0, 0, 1, 2, 0, 0]
The feature gives largest information gain is 0
The feature gives least information gain is 3
K=6:
The decision tree is 99 percent accurate on 1097 training data
The decision tree is 98 percent accurate on 275 test data
[0, 1, 2, 0, 1, 0, 0, 0, 0, 0, 0, 2, 1, 0, 0, 3, 2, 0, 0, 0, 0, 1, 2,
0, 0, 0, 0]
The feature gives largest information gain is 0
The feature gives least information gain is 3
```

As we can see from the above result, the feature 0 has the largest information gain, and the feature 3 has the least. Using Gini and entropy does not have a sigfinicant difference. Increase the k wounld not cause overfit since the test error didn't increase much.