In [6]:
```
import pandas as pd
from sklearn.tree import DecisionTreeClassifier # Import Decision Tree Classifier
from sklearn.model_selection import train_test_split # Import train_test_split function
from sklearn import metrics #Import scikit-learn metrics module for accuracy calculation
```

In [11]:
```
col_names = ['age', 'sex', 'cp', 'trestbps', 'chol',
             'fbs', 'restecg', 'MHR', 'exang',
             'oldpeak', 'slope', 'ca', 'thal', 'target']
# load dataset
pima = pd.read_csv("heart.csv", header=None, names=col_names)
```

In [12]:
```
pima.head()
```

Out[12]:

	age	sex	cp	trestbps	chol	fbs	restecg	MHR	exang	oldpeak	slope	ca	thal	target
0	63	1	3	145	233	1	0	150	0	2.3	0	0	1	1
1	37	1	2	130	250	0	1	187	0	3.5	0	0	2	1
2	41	0	1	130	204	0	0	172	0	1.4	2	0	2	1
3	56	1	1	120	236	0	1	178	0	0.8	2	0	2	1
4	57	0	0	120	354	0	1	163	1	0.6	2	0	2	1

In [16]:
```
#split dataset in features and target variable
feature_cols =['age', 'sex', 'cp', 'trestbps', 'chol',
             'fbs', 'restecg', 'MHR', 'exang',
             'oldpeak', 'slope', 'ca', 'thal']
X = pima[feature_cols] # Features
y = pima.target # Target variable
```

In [17]:
```
# Split dataset into training set and test set
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=1) # 70% training and 30% test
```

In [18]:
```
# Create Decision Tree classifer object
clf = DecisionTreeClassifier()

# Train Decision Tree Classifer
clf = clf.fit(X_train,y_train)

#Predict the response for test dataset
y_pred = clf.predict(X_test)
```

In [19]:
```
# Model Accuracy, how often is the classifier correct?
print("Accuracy:",metrics.accuracy_score(y_test, y_pred))
```

```
Accuracy: 0.725274725275
```

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [46]:
```
def print_decision_tree(tree, feature_names, offset_unit='    '):
    left      = tree.tree_.children_left
    right     = tree.tree_.children_right
threshold = tree.tree_.threshold
value = tree.tree_.value
if feature_names is None:
    features  = ['f%d'%i for i in tree.tree_.feature]
else:
    features  = [feature_names[i] for i in tree.tree_.feature]

def recurse(left, right, threshold, features, node, depth=0):
        offset = offset_unit*depth
        if (threshold[node] != -2):
                print(offset+"if ( " + features[node] + " <= " + str(threshold[node]) + " ) {")
                if left[node] != -1:
                        recurse (left, right, threshold, features,left[node],depth+1)
                print(offset+"} else {")
                if right[node] != -1:
                        recurse (left, right, threshold, features,right[node],depth+1)
                print(offset+"}")
        else:
                #print(offset,value[node])

                #To remove values from node
                temp=str(value[node])
                mid=len(temp)//2
                tempx=[]
                tempy=[]
                cnt=0
                for i in temp:
                    if cnt<=mid:
                        tempx.append(i)
                        cnt+=1
                    else:
                        tempy.append(i)
                        cnt+=1
                val_yes=[]
                val_no=[]
                res=[]
                for j in tempx:
                    if j=="[" or j=="]" or j=="." or j==" ":
                        res.append(j)
                    else:
                        val_no.append(j)
                for j in tempy:
                    if j=="[" or j=="]" or j=="." or j==" ":
                        res.append(j)
                    else:
                        val_yes.append(j)
                val_yes = int("".join(map(str, val_yes)))
                val_no = int("".join(map(str, val_no)))

                if val_yes>val_no:
                    print(offset,'\033[1m',"YES")
                    print('\033[0m')
                elif val_no>val_yes:
                    print(offset,'\033[1m',"NO")
                    print('\033[0m')
                else:
                    print(offset,'\033[1m',"Tie")
                    print('\033[0m')
recurse(left, right, threshold, features, 0,0)
```

In [49]:
```
print_decision_tree(clf,feature_cols)
```

```
if ( cp <= 0.5 ) {
    if ( exang <= 0.5 ) {
        if ( ca <= 0.5 ) {
            if ( thal <= 2.5 ) {
                if ( MHR <= 96.5 ) {
                      NO

                } else {
                      YES

                }
            } else {
                if ( restecg <= 0.5 ) {
                      NO

                } else {
                    if ( oldpeak <= 0.850000023842 ) {
                          YES

                    } else {
                          NO

                    }
                }
            }
        } else {
            if ( trestbps <= 109.0 ) {
                if ( ca <= 1.5 ) {
                      NO

                } else {
                      YES

                }
            } else {
                  NO

            }
        }
    } else {
        if ( MHR <= 166.5 ) {
            if ( MHR <= 106.5 ) {
                if ( oldpeak <= 1.0 ) {
                      YES

                } else {
                      NO

                }
            } else {
                  NO

            }
        } else {
            if ( chol <= 238.0 ) {
                  YES

            } else {
                  NO

            }
        }
    }
} else {
    if ( oldpeak <= 2.09999990463 ) {
        if ( trestbps <= 176.0 ) {
            if ( age <= 56.5 ) {
                if ( trestbps <= 119.0 ) {
                    if ( age <= 46.5 ) {
                          YES

                    } else {
                        if ( age <= 50.0 ) {
                              NO

                        } else {
                              YES

                        }
                    }
                } else {
                      YES

                }
            } else {
                if ( sex <= 0.5 ) {
                    if ( thal <= 2.5 ) {
                          YES

                    } else {
                          NO

                    }
                } else {
                    if ( chol <= 253.0 ) {
                        if ( ca <= 0.5 ) {
                            if ( age <= 65.5 ) {
                                  YES

                            } else {
                                if ( cp <= 1.5 ) {
                                      YES

                                } else {
                                      NO

                                }
                            }
                        } else {
                            if ( MHR <= 148.0 ) {
                                  YES

                            } else {
                                  NO

                            }
                        }
                    } else {
                        if ( age <= 66.5 ) {
                              NO

                        } else {
                              YES

                        }
                    }
                }
            }
        } else {
              NO
```

```
            }
        } else {
            if ( chol <= 239.5 ) {
                  NO

            } else {
                if ( ca <= 0.5 ) {
                      YES

                } else {
                      NO

                }
            }
        }
    }
```

In [ ]:

In [ ]: