In [16]:
```
import pandas as pd
import numpy as np
from sklearn import tree
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import accuracy_score
from sklearn.metrics import confusion_matrix
import graphviz
```

In [17]:
```
data = pd.read_csv('../../Desktop/HW1/vertebral_column_data/column_3C.dat', sep=' ', header=None)
data.columns = ['pelvic_incidence numeric', 'pelvic_tilt numeric', 'lumbar_lordosis_angle numeric', 'sacral_slope numeric', 'pelvic_radius numeric','degree_spondylolisthesis numeric','class']
features = ['pelvic_incidence numeric', 'pelvic_tilt numeric', 'lumbar_lordosis_angle numeric', 'sacral_slope numeric', 'pelvic_radius numeric','degree_spondylolisthesis numeric']
classes = ['disk hernia (DH)', 'spondylolisthesis (SL)', 'normal (NO)']
```

In [18]:
```
X=data.iloc[:,0:6].values
Y=data.iloc[:,6].values
#print (len(data))
#print (data.shape)
```

In [ ]:

In [19]:
```
# Tree 1
```

In [20]:
```
X_train, X_test, Y_train, Y_test = train_test_split(X,Y,test_size=0.3,random_state=10,shuffle=True)
```

In [21]:
```
c = tree.DecisionTreeClassifier(criterion = "gini", random_state = 10, max_depth=3)
c = c.fit(X_train, Y_train)
```

In [22]:
```
dot_data = tree.export_graphviz(decision_tree=c, out_file=None, feature_names=features, class_names=classes, filled=True, rounded=True, special_characters=True)
graph = graphviz.Source(dot_data)
graph.render('plot1', view=True)
```

Out[22]:
```
'plot1.pdf'
```

In [23]:

```
prediction = c.predict(X_test)
print(accuracy_score(Y_test, prediction))
```

```
0.8387096774193549
```

In [24]:

```
print(confusion_matrix(Y_test, prediction))
```

```
[[13  3  0]
 [ 8 20  0]
 [ 1  3 45]]
```

In [ ]:

In [25]:

```
# Tree 2
```

In [26]:

```
X_train, X_test, Y_train, Y_test = train_test_split(X,Y,test_size=0.5,random_state=10,shuffle=True)
```

In [27]:

```
c = tree.DecisionTreeClassifier(criterion = "gini", random_state = 10, max_depth=6)
c = c.fit(X_train, Y_train)
```

In [28]:

```
dot_data = tree.export_graphviz(decision_tree=c, out_file=None, feature_names=features, class_names=classes, filled=True, rounded=True, special_characters=True)
graph = graphviz.Source(dot_data)
graph.render('plot2', view=True)
```

Out[28]:

```
'plot2.pdf'
```

In [29]:

```
prediction = c.predict(X_test)
print(accuracy_score(Y_test, prediction))
```

```
0.8774193548387097
```

In [30]:

```
print(confusion_matrix(Y_test, prediction))
```

```
[[24  4  1]
 [ 8 39  2]
 [ 1  3 73]]
```

In [ ]:

In [31]:

```
# Tree 3
```

In [32]:

```
X_train, X_test, Y_train, Y_test = train_test_split(X,Y,test_size=0.7,random_state=10,shuffle=True)
```

In [33]:
```
c = tree.DecisionTreeClassifier(criterion = "gini", random_state =
10, max_depth=4)
c = c.fit(X_train, Y_train)
```

In [34]:
```
dot_data = tree.export_graphviz(decision_tree=c, out_file=None, fea
ture_names=features, class_names=classes, filled=True, rounded=True
, special_characters=True)
graph = graphviz.Source(dot_data)
graph.render('plot3', view=True)
```

Out[34]:
```
'plot3.pdf'
```

In [35]:
```
prediction = c.predict(X_test)
print(accuracy_score(Y_test, prediction))
```

```
0.8202764976958525
```

In [36]:
```
print(confusion_matrix(Y_test, prediction))
```

```
[[ 25  10   1]
 [ 20  49   3]
 [  4   1 104]]
```

In [ ]:
```

```