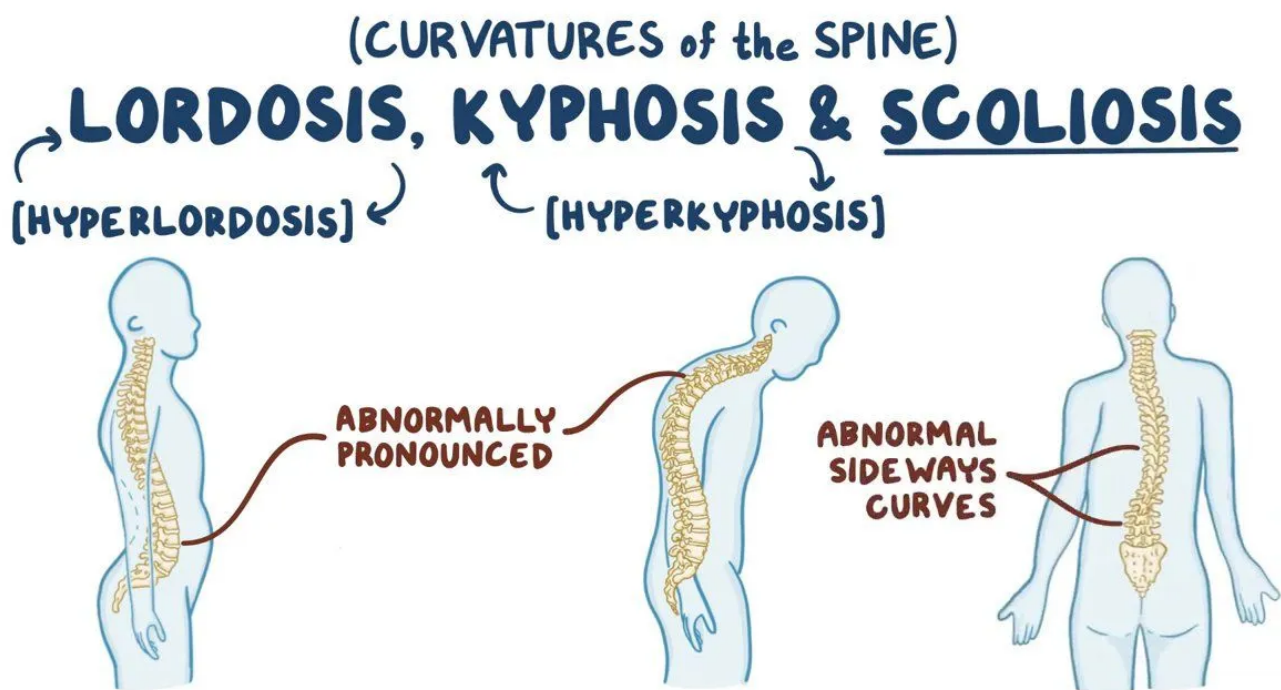


Double-click (or enter) to edit

# Decision Trees and Random Forests in Python

Reference: <http://net-informations.com/ds/mla/dtree.htm>, <https://www.kaggle.com/code>

Credit (Image) from [https://www.osmosis.org/learn/Lordosis,\\_kyphosis,\\_and\\_scoliosis](https://www.osmosis.org/learn/Lordosis,_kyphosis,_and_scoliosis)



- ✓ Install this package before starting this lab
- ✓ Import Libraries

```
!pip install --upgrade scikit-learn==1.0.2
!pip install --upgrade numpy==1.21.5
```

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

## ✓ Get the Data

```
!wget https://github.com/davidjohnnn/all\_datasets/raw/master/bay/kyphosis.csv
```

```
df = pd.read_csv('kyphosis.csv')
```

```
df.head()
```

## ✓ EDA

We'll just check out a simple pairplot for this small dataset.

```
sns.pairplot(df,hue='Kyphosis',palette='Set1')
```

## ✓ Train Test Split

Let's split up the data into a training set and a test set!

```
from sklearn.model_selection import train_test_split
```

```
X = df.drop('Kyphosis',axis=1)
y = df['Kyphosis']
```

```
X_train, X_test, y_train, y_test = train_test_split(X, y, stratify=y, test_size=
```

## ✓ Decision Trees

We'll start just by training a single decision tree.

<http://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeClassifier.html#sklearn.tree.DecisionTreeClassifier>

```
from sklearn.tree import DecisionTreeClassifier
```

```
dtree = DecisionTreeClassifier(min_samples_leaf=10, criterion='entropy')
```

```
dtree.fit(X_train,y_train)
```

```
import pickle
filename = 'model.sav'
pickle.dump(dtree, open(filename, 'wb'))
```

## ✓ Prediction and Evaluation

Let's evaluate our decision tree.

```
dtree = pickle.load(open(filename,'rb'))
dtree
```

```
predictions = dtree.predict(X_test)
```

```
from sklearn.metrics import classification_report,confusion_matrix
```

```
print(classification_report(y_test,predictions,digits=4))
```

```
print(confusion_matrix(y_test,predictions,labels=['absent','present']))
```

## ✓ Tree Visualization

Scikit learn actually has some built-in visualization capabilities for decision trees, you won't use this often and it requires you to install the pydot library, but here is an example of what it looks like and the code to execute this:

```
from sklearn import tree
tree.plot_tree(dtree)
```

```
print(X.columns) # feature names
print(y.unique().tolist()) # class names
```

```
fn=X.columns # feature names
cn=y.unique().tolist() # class names
fig, axes = plt.subplots(nrows = 1,ncols = 1,figsize = (4,6), dpi=100)
tree.plot_tree(dtree,
                feature_names = fn,
                class_names=cn,
                filled = True);
fig.savefig('imagenname.png')
```