https://stackabuse.com/decision-trees-in-python-with-scikit-learn/ (https://stackabuse.com/decision-trees-in-python-with-scikit-learn/)

In [2]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
```

In [3]:

```
dataset = pd.read_csv('รวมไม่มีเอม.csv', index_col=0)
```

In [4]:

```
dataset.shape
```

Out[4]:

```
(104, 35)
```

In [5]:

```
dataset.tail()
```

Out[5]:

	Label	Gx: Average	Gy: Average	Gz: Average	Ax: Average	Ay: Average	Az: Averag
Time window							
21:15:27	Sleep	43.728972	-29.813084	29.682243	-13280.85981	6957.925234	-4573.8411
21:15:28	Sleep	25.991150	-49.053097	-15.752212	-13388.69027	6811.380531	-4577.4867
21:15:29	Sleep	37.194690	58.159292	26.716814	-13403.79646	6799.884956	-4599.3097
21:15:30	Sleep	154.405405	-31.315315	230.549550	-13423.95495	6837.603604	-4649.6216
21:15:31	Sleep	-581.027027	1320.315315	867.333333	-12189.36937	9137.720721	-4198.9090

5 rows × 35 columns

In [6]:

```
X = dataset.drop('Label', axis=1)
y = dataset['Label']
```

In [7]:

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3)

#test30%train70%
```

In [8]:

```
from sklearn.tree import DecisionTreeClassifier
classifier = DecisionTreeClassifier()
classifier.fit(X_train, y_train)
```

Out[8]:

```
DecisionTreeClassifier(class_weight=None, criterion='gini', max_depth=None,
                       max_features=None, max_leaf_nodes=None,
                       min_impurity_decrease=0.0, min_impurity_split=None,
                       min_samples_leaf=1, min_samples_split=2,
                       min_weight_fraction_leaf=0.0, presort=False,
                       random_state=None, splitter='best')
```

In [15]:

```
y_pred = classifier.predict(X_test)
```

In [16]:

```
X_train
```

Out[16]:

	Gx: Average	Gy: Average	Gz: Average	Ax: Average	Ay: Average	Az: Average
Time window						
4:46:12 PM	-63.920400	-64.548700	-11.929200	11515.86000	-6716.110000	-6370.910000
4:46:10 PM	455.526800	-271.714000	149.839300	11437.73000	-6413.040000	-6707.310000
4:46:13 PM	-237.766000	77.747750	-45.144100	11385.45000	-6237.860000	-6856.590000
7:45:08 AM	109.495500	17.585590	-89.045000	1596.38700	-7381.180000	-12532.100000
21:15:26	28.596491	-195.833333	18.219298	-13275.77193	7236.201754	-4391.192982
...	...	...	...	...	...	...
21:15:09	56.883929	-62.258929	-322.633929	-13332.41964	6968.089286	-4519.651786
7:45:01 AM	-248.894000	34.946900	185.283200	2366.46000	-7559.670000	-12366.000000
7:46:56 AM	67.482760	-9.137930	54.793100	4595.73300	-13526.200000	1489.759000
21:14:45	81.230089	39.681416	49.938053	-14415.68142	6065.858407	-2768.442478
21:14:53	19.017699	-21.849558	94.300885	-14291.87611	6094.433628	-2962.902655

72 rows × 34 columns

In [17]:

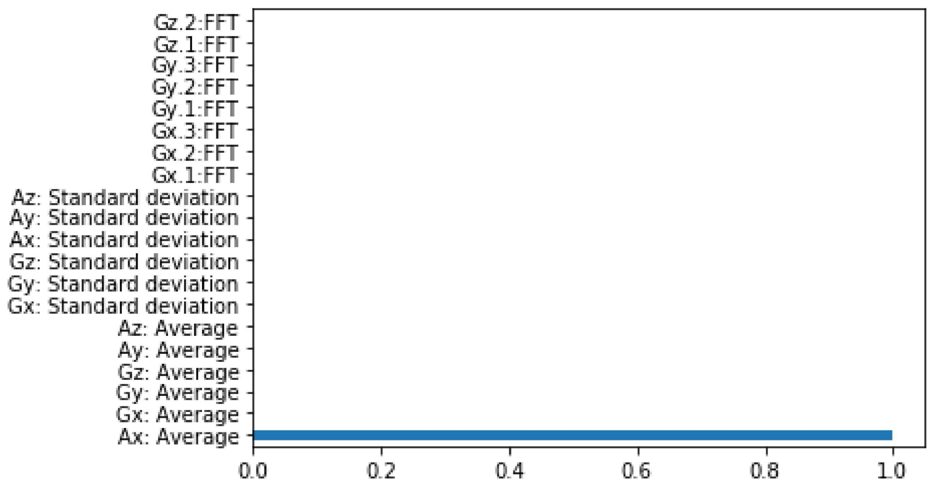
```
from sklearn.metrics import classification_report, confusion_matrix
print(confusion_matrix(y_test, y_pred))
print(classification_report(y_test, y_pred))
```

```
[[11  0]
 [ 0 21]]
              precision    recall  f1-score   support

         Eat       1.00      1.00      1.00        11
       Sleep       1.00      1.00      1.00        21

    accuracy                           1.00        32
   macro avg       1.00      1.00      1.00        32
weighted avg       1.00      1.00      1.00        32
```

In [18]:

```
print(classifier.feature_importances_) #use inbuilt class feature_importances of tree based classifiers

#plot graph of feature importances for better visualization
feat_importances = pd.Series(classifier.feature_importances_, index=X.columns)
feat_importances.nlargest(20).plot(kind='barh')
plt.show()
```

```
[0. 0. 0. 1. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
```



In [19]:

```
#from sklearn.tree.export import export_txt
from sklearn.tree import export_text

tree_rules = export_text(classifier, feature_names=list(X_train))
print(tree_rules)
```

```
|--- Ax: Average <= -2996.61
|   |--- class: Sleep
|--- Ax: Average >  -2996.61
|   |--- class: Eat
```

In [ ]:

```
```

In [ ]:

```
```