Building a decision Tree

Importing Necessary Libraries

In the following block of codes, We will import necessary libraries that include Pandas, Matplotlib and Sklearn

In [1]:

```
from pandas import Series, DataFrame
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split #This was cross_validat
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import classification_report
import sklearn
```

Loading the Dataset

The dataset is from *The National Longitudinal Study of Adolescent Health* (AddHealth) is a representative school-based survey of adolescents in grades 7-12 in the United States. The Wave 1 survey focuses on factors that may influence adolescents' health and risk behaviors, including personal traits, families, friendships, romantic relationships, peer groups, schools, neighborhoods, and communities.

In [2]:

```
AH_data = pd.read_csv("../datasets/datasetfortree.csv")
```

Checking first five rows of the data

In [3]:

```
AH_data.head()
```

Out[3]:

	BIO_SEX	HISPANIC	WHITE	BLACK	NAMERICAN	ASIAN	age	TREG1	ALCEVR1
0	2.0	0.0	0.0	1.0	0.0	0.0	NaN	0.0	1.0
1	2.0	0.0	0.0	1.0	0.0	0.0	19.427397	1.0	1.0
2	1.0	0.0	1.0	0.0	0.0	0.0	NaN	0.0	0.0
3	1.0	0.0	0.0	1.0	0.0	0.0	20.430137	1.0	0.0
4	2.0	0.0	0.0	1.0	0.0	0.0	NaN	0.0	1.0

5 rows × 25 columns

- **Creating a cleaned dataset by dropping missing values**

In [4]:

```
data_clean = AH_data.dropna()
```

- **Looking at the descriptive Statistics of the cleaned data**

In [5]:
```
data_clean.describe()
```

Out[5]:

	BIO_SEX	HISPANIC	WHITE	BLACK	NAMERICAN	ASIAN	
count	4575.000000	4575.000000	4575.000000	4575.000000	4575.000000	4575.000000	4
mean	1.521093	0.111038	0.683279	0.236066	0.036284	0.040437	
std	0.499609	0.314214	0.465249	0.424709	0.187017	0.197004	
min	1.000000	0.000000	0.000000	0.000000	0.000000	0.000000	
25%	1.000000	0.000000	0.000000	0.000000	0.000000	0.000000	
50%	2.000000	0.000000	1.000000	0.000000	0.000000	0.000000	
75%	2.000000	0.000000	1.000000	0.000000	0.000000	0.000000	
max	2.000000	1.000000	1.000000	1.000000	1.000000	1.000000	

8 rows × 25 columns

Modelling and Prediction

Spliting data into training and Testing Sets

Defining Predictors
* If the student ever had alcohol and if the student ever had marijuana

In [6]:
```
predictors = data_clean[['ALCEVR1','marever1']]
```

Defining Target varaible: 'Treg1'

In [7]:
```
targets = data_clean.TREG1
```

Predicting training test and target data with 40% of test size

In [8]:
```
pred_train, pred_test, tar_train, tar_test  =   train_test_split(predictors,
```

Checking the dimensions of the target and training data

In [9]:
```
pred_train.shape
```

Out[9]:
```
(2745, 2)
```

In [10]:
```
pred_test.shape
```

Out[10]:
```
(1830, 2)
```

In [11]:
```
tar_train.shape
```

Out[11]:
```
(2745,)
```

In [12]:
```
tar_test.shape
```

Out[12]:

```
(1830,)
```

Build model on training data

Confusion matrix

In [13]:

```
classifier=DecisionTreeClassifier()
classifier=classifier.fit(pred_train,tar_train)

predictions=classifier.predict(pred_test)

sklearn.metrics.confusion_matrix(tar_test,predictions)
```

Out[13]:

```
array([[1500,    0],
       [ 330,    0]])
```

The confusion matrix above shows that out of 1830, 330 observations are mispridected by the model, Now lets see the accuracy score

In [14]:

```
sklearn.metrics.accuracy_score(tar_test, predictions)
```

Out[14]:

```
0.819672131147541
```

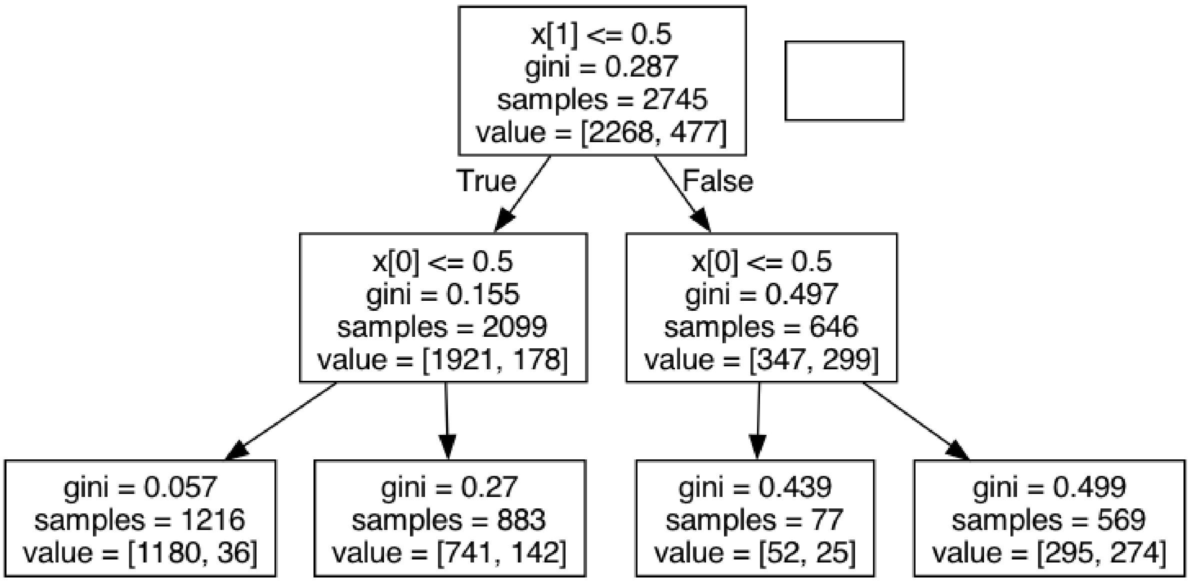
The accuracy score is 81.96% which is much better

Displaying the Decision Tree

In [15]:

```
from sklearn import tree
from io import StringIO
from IPython.display import Image
out = StringIO()
tree.export_graphviz(classifier, out_file=out)
import pydotplus
graph=pydotplus.graph_from_dot_data(out.getvalue())
Image(graph.create_png())
```

Out[15]:



In [ ]: