In [1]:
```
import pandas as pd
import pylab as pl
import numpy as np
import scipy.optimize as opt
from sklearn.tree import DecisionTreeClassifier
from sklearn.model_selection import train_test_split
from sklearn import metrics
import matplotlib.pyplot as plt
import warnings
from sklearn.metrics import confusion_matrix
from sklearn.metrics import classification_report
from sklearn import tree
import graphviz
import pydotplus
from PIL import Image
# Adicion de colores al arbol de decision
from six import StringIO
from IPython.display import Image
from sklearn.tree import export_graphviz
warnings.filterwarnings('ignore')
import os
os.chdir('/Users/Lenovo/Desktop/EBAC')
```

In [3]:
```
data = pd.read_csv('drugs1.csv')
data
```

Out[3]:

	Age	Sex	BP	Cholesterol	Na_to_K	Drug
0	23	F	HIGH	HIGH	25.355	drugY
1	47	M	LOW	HIGH	13.093	drugC
2	47	M	LOW	HIGH	10.114	drugC
3	28	F	NORMAL	HIGH	7.798	drugX
4	61	F	LOW	HIGH	18.043	drugY
...	...	...	...	...	...	...
195	56	F	LOW	HIGH	11.567	drugC
196	16	M	LOW	HIGH	12.006	drugC
197	52	M	NORMAL	HIGH	9.894	drugX
198	23	M	NORMAL	NORMAL	14.020	drugX
199	40	F	LOW	NORMAL	11.349	drugX

200 rows × 6 columns

In [5]:
```
features_cols = ['Age', 'Sex', 'BP', 'Cholesterol', 'Na_to_K']
X = data[features_cols].values
y = data.Drug
```

In [7]:
```
from sklearn import preprocessing
Cod_Sex = preprocessing.LabelEncoder()
Cod_Sex.fit(['F', 'M'])
X[:,1] = Cod_Sex.transform(X[:,1])

Cod_BP = preprocessing.LabelEncoder()
Cod_BP.fit(['HIGH', 'LOW', 'NORMAL'])
X[:,2] = Cod_BP.transform(X[:,2])

Cod_Cholesterol = preprocessing.LabelEncoder()
Cod_Cholesterol.fit(['HIGH', 'LOW', 'NORMAL'])
X[:,3] = Cod_Cholesterol.transform(X[:,3])
```

In [9]:
```
# Creacion de grupos de entrenamiento y prueba
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.3, random_state = 1)
```

Gini, Produndidad = 5

In [28]:
```
# Crear Objeto de clasificacion del arbol de decision
clf = DecisionTreeClassifier(criterion = 'gini', max_depth = 5)

# Apliacion del algoritmo de arboles de decision a los grupos de entrenamiento
clf = clf.fit(X_train, y_train)

# Prediccion de la respuesta para el grupo de prueba
```

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

In [30]:
```
# Matrix de confusion
from sklearn.metrics import confusion_matrix
cm = confusion_matrix(y_test, y_pred)
```

In [32]:
```
# estadisticas de desempe;o
from sklearn.metrics import classification_report
print(classification_report(y_test, y_pred))
```

```
              precision    recall  f1-score   support

       drugA       0.67      1.00      0.80         4
       drugB       1.00      0.67      0.80         6
       drugC       1.00      1.00      1.00         4
       drugX       1.00      1.00      1.00        19
       drugY       1.00      1.00      1.00        27

    accuracy                           0.97        60
   macro avg       0.93      0.93      0.92        60
weighted avg       0.98      0.97      0.97        60
```

In [34]:
```
# Creacion de informacion para el arbol (reglas de decision)
dot_data = tree.export_graphviz(clf, out_file = None, feature_names = features_cols, class_names = ['DrugA', 'D

# Creacion de la grafica del arbol
graph = pydotplus.graph_from_dot_data(dot_data)
```

In [36]:
```
from PIL import Image
# Creacion del rbol en formato PDF
graph.write_pdf("medicinas.pdf")

# Creacion del rbol en formato PNG
graph.write_png('medicinas.png')

image = Image.open('medicinas.png')
image.show()
```

Para el criterio de Gini con una profundidad de 5, vemos una acertividad del 97% que es muy buena para predecir los resultados.

Gini, Produndidad = 3

In [39]:
```
# Crear Objeto de clasificacion del arbol de decision
clf = DecisionTreeClassifier(criterion = 'gini', max_depth = 3)

# Apliacion del algoritmo de arboles de decision a los grupos de entrenamiento
clf = clf.fit(X_train, y_train)

# Prediccion de la respuesta para el grupo de prueba
y_pred = clf.predict(X_test)

cm = confusion_matrix(y_test, y_pred)

print(classification_report(y_test, y_pred))
```

```
              precision    recall  f1-score   support

       drugA       0.67      1.00      0.80         4
       drugB       1.00      0.67      0.80         6
       drugC       0.00      0.00      0.00         4
       drugX       0.83      1.00      0.90        19
       drugY       1.00      1.00      1.00        27

    accuracy                           0.90        60
   macro avg       0.70      0.73      0.70        60
weighted avg       0.86      0.90      0.87        60
```

In [41]:
```
# Creacion de informacion para el arbol (reglas de decision)
dot_data = tree.export_graphviz(clf, out_file = None, feature_names = features_cols, class_names = ['DrugA', 'D

# Creacion de la grafica del arbol
graph = pydotplus.graph_from_dot_data(dot_data)

from PIL import Image
# Creacion del rbol en formato PDF
graph.write_pdf("medicinas.pdf")

# Creacion del rbol en formato PNG
graph.write_png('medicinas.png')
```

```
image = Image.open('medicinas.png')
image.show()
```

Para el criterio de Gini con una profundidad de 3, vemos una acertividad del 90% que, aunque se pudiera decir que es alta, tiene un decremento en comparacion con la profundidad de 5, por lo que se recomienda usar mejor produndidad de 5.

Entropy, Profundidad = 5

In [44]:
```
# Crear Objeto de clasificacion del arbol de decision
clf = DecisionTreeClassifier(criterion = 'entropy', max_depth = 5)

# Apliacion del algoritmo de arboles de decision a los grupos de entrenamiento
clf = clf.fit(X_train, y_train)

# Prediccion de la respuesta para el grupo de prueba
y_pred = clf.predict(X_test)

cm = confusion_matrix(y_test, y_pred)

print(classification_report(y_test, y_pred))
```

```
              precision    recall  f1-score   support

       drugA       0.67      1.00      0.80         4
       drugB       1.00      0.67      0.80         6
       drugC       1.00      1.00      1.00         4
       drugX       1.00      1.00      1.00        19
       drugY       1.00      1.00      1.00        27

    accuracy                           0.97        60
   macro avg       0.93      0.93      0.92        60
weighted avg       0.98      0.97      0.97        60
```

In [46]:
```
# Creacion de informacion para el arbol (reglas de decision)
dot_data = tree.export_graphviz(clf, out_file = None, feature_names = features_cols, class_names = ['DrugA', 'D

# Creacion de la grafica del arbol
graph = pydotplus.graph_from_dot_data(dot_data)

from PIL import Image
# Creacion del rbol en formato PDF
graph.write_pdf("medicinas.pdf")

# Creacion del rbol en formato PNG
graph.write_png('medicinas.png')

image = Image.open('medicinas.png')
image.show()
```

Para el criterio de Entropy con una profundidad de 5, vemos una acertividad del 97% que es muy buena para predecir los resultados.

Entropy, Profundidad = 3

In [55]:
```
# Crear Objeto de clasificacion del arbol de decision
clf = DecisionTreeClassifier(criterion = 'entropy', max_depth = 3)

# Apliacion del algoritmo de arboles de decision a los grupos de entrenamiento
clf = clf.fit(X_train, y_train)

# Prediccion de la respuesta para el grupo de prueba
y_pred = clf.predict(X_test)

cm = confusion_matrix(y_test, y_pred)

print(classification_report(y_test, y_pred))
```

```
                precision    recall  f1-score   support

       drugA       0.67      1.00      0.80         4
       drugB       1.00      0.67      0.80         6
       drugC       0.00      0.00      0.00         4
       drugX       0.83      1.00      0.90        19
       drugY       1.00      1.00      1.00        27

    accuracy                           0.90        60
   macro avg       0.70      0.73      0.70        60
weighted avg       0.86      0.90      0.87        60
```

In [57]:
```
# Creacion de informacion para el arbol (reglas de decision)
dot_data = tree.export_graphviz(clf, out_file = None, feature_names = features_cols, class_names = ['DrugA', 'D

# Creacion de la grafica del arbol
graph = pydotplus.graph_from_dot_data(dot_data)

from PIL import Image
# Creacion del rbol en formato PDF
graph.write_pdf("medicinas.pdf")

# Creacion del rbol en formato PNG
graph.write_png('medicinas.png')

image = Image.open('medicinas.png')
image.show()
```

Para el criterio de Entropy con una profundidad de 3, vemos una acertividad del 90% que, aunque se pudiera decir que es alta, tiene un decremento en comparacion con la profundidad de 5, por lo que se recomienda usar mejor produndidad de 5.

Que medicamento recomendaria utilizar para un paciente con los siguientes datos?

In [82]:
```
X1_test = np.array([[50,0,0,2,15.302]])
```

In [84]:
```
# Crear Objeto de clasificacion del arbol de decision
clf = DecisionTreeClassifier(criterion = 'gini', max_depth = 5)

# Apliacion del algoritmo de arboles de decision a los grupos de entrenamiento
clf = clf.fit(X_train, y_train)

# Prediccion de la respuesta para el grupo de prueba
y_pred = clf.predict(X1_test)
```

In [92]:
```
y_pred
```

Out[92]:
```
array(['drugY'], dtype=object)
```

Para los parametros anteriormente mencionados, necesitaria utilizar el medicamento "DrugY"

In [ ]:
```

```