Classification using Decision Tree

Importing necessary libraries

In [1]:
```
import pandas as pd
import numpy as np
from sklearn import tree
```

In [ ]:
```
df = pd.read_csv("input.csv")
df.head()
```

Out[ ]:

	ID	Age	Income	Student	Credit Rating	Buy Car
0	1	Young	High	No	Fair	No
1	2	Young	High	No	Good	No
2	3	Middle	High	No	Fair	Yes
3	4	Old	Medium	No	Fair	Yes
4	5	Old	Low	Yes	Fair	Yes

Converting to numerical data

In [ ]:
```
age = df['Age'].factorize();
income = df['Income'].factorize();
student = df['Student'].factorize();
credit_rating = df['Credit Rating'].factorize();
buy_car = df['Buy Car'].factorize();
```

In [ ]:
```
factorized_data = {'Age':age[0], 'Income':income[0], 'Student':student[0], 'Credit Rating':credit_rating[0], 'Buy Car':buy_car[0]}

df1 = pd.DataFrame(factorized_data)
df1.head()
```

Out[ ]:

	Age	Income	Student	Credit Rating	Buy Car
0	0	0	0	0	0
1	0	0	0	1	0
2	1	0	0	0	1
3	2	1	0	0	1
4	2	2	1	0	1

Dividing the data into training set and testing set

In [ ]:
```
X = df1[['Age','Income','Student','Credit Rating']]
y = df1['Buy Car']
```

In [ ]:
```
from sklearn.model_selection import train_test_split
X_train,X_test,y_train,y_test = train_test_split(X,y,test_size = 0.2, random_state=1)

print(X_train.shape)
print(y_train.shape)

print(X_test.shape)
print(y_test.shape)
```

```
(11, 4)
(11,)
(3, 4)
(3,)
```

Training Classifier

In [ ]:
```
clf = tree.DecisionTreeClassifier(criterion='entropy',max_depth=3, random_state=1)
```

In [ ]:
```
clf.fit(X_train,y_train)
prediction = clf.predict(X_test)

prediction
```

Out[ ]:
```
array([1, 0, 1])
```

Testing our classifier

In [ ]:
```
from sklearn import metrics
print("Accuracy = {}%".format(metrics.accuracy_score(y_test,prediction) * 100))
```

```
Accuracy = 100.0%
```

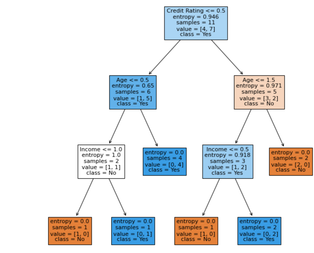
Finding root node

In [ ]:
```
import matplotlib.pyplot as plt

fn=['Age','Income','Student','Credit Rating']
cn=['No', 'Yes']

plt.figure(figsize = (10,10))

tree.plot_tree(clf,
               feature_names = fn,
               class_names=cn,
               filled = True);
```



Classification on Diabeties data

In [5]:
```
from sklearn import datasets
from sklearn.model_selection import train_test_split
from sklearn import metrics
```

In [13]:
```
df = pd.read_csv("diabetes.csv", header=None)
df.head()
```

Out[13]:

	0	1	2	3	4	5	6	7	8
0	6	148	72	35	0	33.6	0.627	50	1
1	1	85	66	29	0	26.6	0.351	31	0
2	8	183	64	0	0	23.3	0.672	32	1
3	1	89	66	23	94	28.1	0.167	21	0
4	0	137	40	35	168	43.1	2.288	33	1

In [14]:
```
X = df[[0,1,2,3,4,5,6,7]]
y = df[[8]]
```

In [15]:
```
X_train,X_test,y_train,y_test = train_test_split(X,y,test_size = 0.2, random_state=1)

print(X_train.shape)
print(y_train.shape)

print(X_test.shape)
print(y_test.shape)
```

```
(614, 8)
(614, 1)
(154, 8)
(154, 1)
```

In [29]:
```
clf = tree.DecisionTreeClassifier(criterion='entropy',max_depth=3, random_state=1)
```

In [30]:
```
clf.fit(X_train,y_train)
prediction = clf.predict(X_test)

prediction
```

Out[30]:
```
array([0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 1, 0, 1, 0, 0, 0, 0, 0,
       1, 0, 0, 1, 0, 1, 0, 1, 1, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 1,
       0, 0, 1, 1, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 1, 0, 1, 0, 0, 0,
       0, 0, 1, 1, 1, 1, 0, 0, 1, 0, 1, 0, 0, 1, 1, 0, 0, 1, 0, 1, 1, 0,
       1, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 1, 0, 0, 1, 1, 0, 0, 0, 0, 0,
       1, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0,
       1, 1, 0, 0, 0, 1, 0, 0, 1, 0, 0, 1, 1, 0, 0, 0, 0, 0, 1, 0, 0])
```

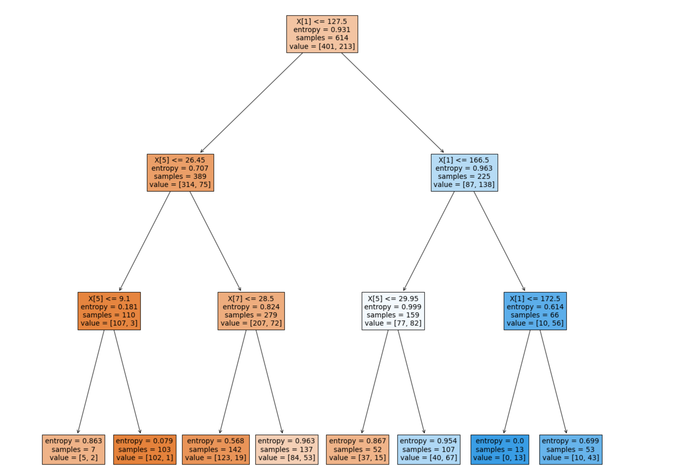
In [31]:
```
from sklearn import metrics
print("Accuracy = {}%".format(metrics.accuracy_score(y_test,prediction) * 100))
```

```
Accuracy = 79.87012987012987%
```

In [32]:
```
import matplotlib.pyplot as plt

plt.figure(figsize = (20,20))

tree.plot_tree(clf,
               filled = True);
```



In [32]:
```

```