Decision Tree for Classification

In this section we will predict whether a bank note is authentic or fake depending upon the four different attributes of the image of the note. The attributes are Variance of wavelet transformed image, curtosis of the image, entropy, and skewness of the image.

Importing Libraries

In [2]:

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

Loading Dataset

In [3]:

```
dataset = pd.read_csv("bill_authentication.csv")
```

In [4]:

```
dataset.head()
```

Out[4]:

	Variance	Skewness	Curtosis	Entropy	Class
0	3.62160	8.6661	-2.8073	-0.44699	0
1	4.54590	8.1674	-2.4586	-1.46210	0
2	3.86600	-2.6383	1.9242	0.10645	0
3	3.45660	9.5228	-4.0112	-3.59440	0
4	0.32924	-4.4552	4.5718	-0.98880	0

Data analysis

In [5]:

```
dataset.columns
```

Out[5]:

```
Index(['Variance', 'Skewness', 'Curtosis', 'Entropy', 'Class'], dtype='object')
```

In [6]:

```
#Checking for null values
dataset.isnull().sum()
#Summarizing---How many null/missing values by coloumn
```

Out[6]:

```
Variance    0
Skewness    0
Curtosis    0
Entropy     0
Class       0
dtype: int64
```

In [7]:

```
dataset.Class
```

Out[7]:

```
0       0
1       0
2       0
3       0
4       0
       ..
1367    1
1368    1
1369    1
1370    1
1371    1
Name: Class, Length: 1372, dtype: int64
```

Preparing the Data

In [8]:

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

In [9]:

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.20,random_state=
```

In [10]:

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

Out[10]:

```
DecisionTreeClassifier()
```

In [11]:

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

Evaluating Algorithm

In [12]:

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

```
[[149   3]
 [  3 120]]
              precision    recall  f1-score   support

           0       0.98      0.98      0.98       152
           1       0.98      0.98      0.98       123

    accuracy                           0.98       275
   macro avg       0.98      0.98      0.98       275
weighted avg       0.98      0.98      0.98       275
```

From the confusion matrix, we can see that out of 275 test instances, our algorithm misclassified only 6. This is 98% accuracy.

In [ ]:

```

```