```
# importing libraries
import numpy as nm
import matplotlib.pyplot as mtp
import pandas as pd
import sklearn

#importing datasets

data_set= pd.read_csv('data_banknote_authentication.txt')
data_set.head()
# data_set.shape
```

	3.6216	8.6661	-2.8073	-0.44699	0
0	4.54590	8.1674	-2.4586	-1.46210	0
1	3.86600	-2.6383	1.9242	0.10645	0
2	3.45660	9.5228	-4.0112	-3.59440	0
3	0.32924	-4.4552	4.5718	-0.98880	0
4	4.36840	9.6718	-3.9606	-3.16250	0

```
#Extracting Independent and dependent Variable
x= data_set.iloc[:, 0:4].values
y= data_set.iloc[:, -1].values
```

```
# Splitting the dataset into training and test set.
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=0)
```

```
#feature Scaling
from sklearn.preprocessing import StandardScaler
st_x= StandardScaler()
x_train= st_x.fit_transform(x_train)
x_test= st_x.transform(x_test)
```

```
print(x) # Features
```

```
[[  4.5459    8.1674   -2.4586   -1.4621 ]
 [  3.866    -2.6383    1.9242    0.10645]
 [  3.4566    9.5228   -4.0112   -3.5944 ]
 ...
 [ -3.7503  -13.4586   17.5932   -2.7771 ]
 [ -3.5637   -8.3827   12.393    -1.2823 ]
 [ -2.5419   -0.65804   2.6842    1.1952 ]]
```

```
print(y) # class labels
```

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

```
#Fitting Decision Tree classifier to the training set
from sklearn.tree import DecisionTreeClassifier
classifier= DecisionTreeClassifier(criterion='entropy', random_state=0)
classifier.fit(x_train, y_train)
```

```
▾                    DecisionTreeClassifier
DecisionTreeClassifier(criterion='entropy', random_state=0)
```

```
#Predicting the test set result
y_pred= classifier.predict(x_test)
```

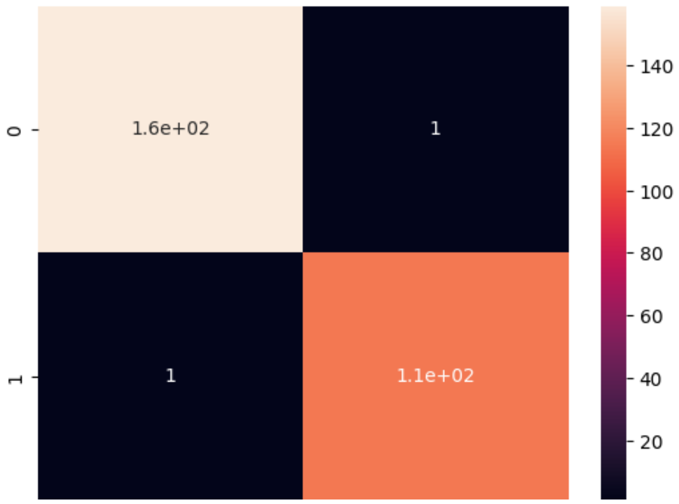
```
print(y)
```

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

```
#Creating the Confusion matrix
from sklearn.metrics import confusion_matrix
cm= confusion_matrix(y_test, y_pred)
```

```
import seaborn as sns
sns.heatmap(cm, annot=True)
```

```
<Axes: >
```



```
cm = cm.astype('float') / cm.sum(axis=1)[:, nm.newaxis]
cm.diagonal()  #Digonals element represent accuracy
```

```
array([0.99375   , 0.99130435])
```