In [21]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

In [22]:
```
df=pd.read_csv("leaf.csv")
```

In [23]:
```
df
```

Out[23]:

	id	species	margin1	margin2	margin3	margin4	margin5	margin6	margin7	margin8	...	texture55	texture56	texture57	texture58	texture59	texture60	texture61	texture6
0	1	Acer_Opalus	0.007812	0.023438	0.023438	0.003906	0.011719	0.009766	0.027344	0.0	...	0.007812	0.000000	0.002930	0.002930	0.035156	0.000000	0.000000	0.00488
1	2	Pterocarya_Stenoptera	0.005859	0.000000	0.031250	0.015625	0.025391	0.001953	0.019531	0.0	...	0.000977	0.000000	0.000000	0.000977	0.023438	0.000000	0.000000	0.00097
2	3	Quercus_Hartwissiana	0.005859	0.009766	0.019531	0.007812	0.003906	0.005859	0.068359	0.0	...	0.154300	0.000000	0.005859	0.000977	0.007812	0.000000	0.000000	0.00000
3	5	Tilia_Tomentosa	0.000000	0.003906	0.023438	0.005859	0.021484	0.019531	0.023438	0.0	...	0.000000	0.000977	0.000000	0.000000	0.020508	0.000000	0.000000	0.01757
4	6	Quercus_Variabilis	0.005859	0.003906	0.048828	0.009766	0.013672	0.015625	0.005859	0.0	...	0.096680	0.000000	0.021484	0.000000	0.000000	0.000000	0.000000	0.00000
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
985	1575	Magnolia_Salicifolia	0.060547	0.119140	0.007812	0.003906	0.000000	0.148440	0.017578	0.0	...	0.242190	0.000000	0.034180	0.000000	0.010742	0.000000	0.000000	0.00000
986	1578	Acer_Pictum	0.001953	0.003906	0.021484	0.107420	0.001953	0.000000	0.000000	0.0	...	0.170900	0.000000	0.018555	0.000000	0.011719	0.000000	0.000000	0.00097
987	1581	Alnus_Maximowiczii	0.001953	0.003906	0.000000	0.021484	0.078125	0.003906	0.007812	0.0	...	0.004883	0.000977	0.004883	0.027344	0.016602	0.007812	0.000000	0.02734
988	1582	Quercus_Rubra	0.000000	0.000000	0.046875	0.056641	0.009766	0.000000	0.000000	0.0	...	0.083008	0.030273	0.000977	0.002930	0.014648	0.000000	0.041992	0.00000
989	1584	Quercus_Afares	0.023438	0.019531	0.031250	0.015625	0.005859	0.019531	0.035156	0.0	...	0.000000	0.000000	0.002930	0.000000	0.012695	0.000000	0.000000	0.02343

990 rows × 194 columns

In [24]:
```
df.isnull().sum()
```

Out[24]:
```
id           0
species      0
margin1      0
margin2      0
margin3      0
           ..
texture60    0
texture61    0
texture62    0
texture63    0
texture64    0
Length: 194, dtype: int64
```

In [25]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 990 entries, 0 to 989
Columns: 194 entries, id to texture64
dtypes: float64(192), int64(1), object(1)
memory usage: 1.5+ MB
```

In [26]:
```
from sklearn.preprocessing import LabelEncoder
le=LabelEncoder()
df.iloc[ : , 1]=le.fit_transform(df.iloc[ : , 1])
```

In [27]:
```
df=df.drop('id',axis=1)
df
```

Out[27]:

	species	margin1	margin2	margin3	margin4	margin5	margin6	margin7	margin8	margin9	...	texture55	texture56	texture57	texture58	texture59	texture60	texture61	texture62	textur
0	3	0.007812	0.023438	0.023438	0.003906	0.011719	0.009766	0.027344	0.0	0.001953	...	0.007812	0.000000	0.002930	0.002930	0.035156	0.000000	0.000000	0.004883	0.000
1	49	0.005859	0.000000	0.031250	0.015625	0.025391	0.001953	0.019531	0.0	0.000000	...	0.000977	0.000000	0.000000	0.000977	0.023438	0.000000	0.000000	0.000977	0.039
2	65	0.005859	0.009766	0.019531	0.007812	0.003906	0.005859	0.068359	0.0	0.000000	...	0.154300	0.000000	0.005859	0.000977	0.007812	0.000000	0.000000	0.000000	0.020
3	94	0.000000	0.003906	0.023438	0.005859	0.021484	0.019531	0.023438	0.0	0.013672	...	0.000000	0.000977	0.000000	0.000000	0.020508	0.000000	0.000000	0.017578	0.000
4	84	0.005859	0.003906	0.048828	0.009766	0.013672	0.015625	0.005859	0.0	0.000000	...	0.096680	0.000000	0.021484	0.000000	0.000000	0.000000	0.000000	0.000000	0.000
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
985	40	0.060547	0.119140	0.007812	0.003906	0.000000	0.148440	0.017578	0.0	0.001953	...	0.242190	0.000000	0.034180	0.000000	0.010742	0.000000	0.000000	0.000000	0.000
986	5	0.001953	0.003906	0.021484	0.107420	0.001953	0.000000	0.000000	0.0	0.029297	...	0.170900	0.000000	0.018555	0.000000	0.011719	0.000000	0.000000	0.000977	0.000
987	11	0.001953	0.003906	0.000000	0.021484	0.078125	0.003906	0.007812	0.0	0.003906	...	0.004883	0.000977	0.004883	0.027344	0.016602	0.007812	0.000000	0.027344	0.000
988	78	0.000000	0.000000	0.046875	0.056641	0.009766	0.000000	0.000000	0.0	0.037109	...	0.083008	0.030273	0.000977	0.002930	0.014648	0.000000	0.041992	0.000000	0.001
989	50	0.023438	0.019531	0.031250	0.015625	0.005859	0.019531	0.035156	0.0	0.003906	...	0.000000	0.000000	0.002930	0.000000	0.012695	0.000000	0.000000	0.023438	0.025

990 rows × 193 columns

In [28]:
```
Y=df.iloc[: , 0]
X=df.drop('species',axis=1)
```

In [29]:
```
from sklearn.model_selection import train_test_split
xtrain, xtest, ytrain, ytest =train_test_split(X,Y, test_size=0.3,random_state=5)
```

In [30]:
```
from sklearn.tree import DecisionTreeClassifier
dtc=DecisionTreeClassifier(criterion='entropy',random_state=2)
dtc.fit(xtrain,ytrain)
```

Out[30]:
```
DecisionTreeClassifier(criterion='entropy', random_state=2)
```

In [31]:
```
pred2=dtc.predict(xtest)
dtc.score(xtest,ytest)
```

Out[31]:
```
0.6262626262626263
```

In [32]:
```
from sklearn.ensemble import RandomForestClassifier
rfc=RandomForestClassifier(n_estimators = 40)
rfc.fit(xtrain,ytrain)
```

Out[32]:
```
RandomForestClassifier(n_estimators=40)
```

In [33]:
```
pred3=rfc.predict(xtest)
```

In [34]:
```
rfc.score(xtest,ytest)
```

Out[34]:
```
0.9528619528619529
```

In [ ]:

In [ ]: