In [16]:
```
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier
from sklearn import metrics
```

In [17]:
```
#reading the data file into df
df = pd.read_csv("wdbc.data", delimiter = ",", header = None)
display(df)
```

	0	1	2	3	4	5	6	7	8	9	...	22	
0	842302	M	17.99	10.38	122.80	1001.0	0.11840	0.27760	0.30010	0.14710	...	25.380	17
1	842517	M	20.57	17.77	132.90	1326.0	0.08474	0.07864	0.08690	0.07017	...	24.990	23
2	84300903	M	19.69	21.25	130.00	1203.0	0.10960	0.15990	0.19740	0.12790	...	23.570	25
3	84348301	M	11.42	20.38	77.58	386.1	0.14250	0.28390	0.24140	0.10520	...	14.910	26
4	84358402	M	20.29	14.34	135.10	1297.0	0.10030	0.13280	0.19800	0.10430	...	22.540	16
...	...	...	...	...	...	...	...	...	...	...	...	...	
564	926424	M	21.56	22.39	142.00	1479.0	0.11100	0.11590	0.24390	0.13890	...	25.450	26
565	926682	M	20.13	28.25	131.20	1261.0	0.09780	0.10340	0.14400	0.09791	...	23.690	38
566	926954	M	16.60	28.08	108.30	858.1	0.08455	0.10230	0.09251	0.05302	...	18.980	34
567	927241	M	20.60	29.33	140.10	1265.0	0.11780	0.27700	0.35140	0.15200	...	25.740	39
568	92751	B	7.76	24.54	47.92	181.0	0.05263	0.04362	0.00000	0.00000	...	9.456	30

569 rows × 32 columns

In [18]:
```
y = df[1].values
x = df.iloc[:, 2:32]
x = np.array(x)

#print(y)
#print(type(y))
#print(x)
#print(type(x))

rs_cols = ["train accuracy", "test accuracy", "precision", "recall"] #column name
rs_rows = ["DT1", "DT2"] #row names for the result table
rs_table = pd.DataFrame(columns=rs_cols, index=rs_rows) #creating the result tabl

display(rs_table)
```

	train accuracy	test accuracy	precision	recall
DT1	NaN	NaN	NaN	NaN
DT2	NaN	NaN	NaN	NaN

In [19]:
```
clf = DecisionTreeClassifier(criterion = "entropy")

train_acc = []
test_acc = []
prec = []
recall = []

for i in range(20):

    (x_train, x_test, y_train, y_test) = train_test_split(x, y, test_size = 0.3)
    clf = clf.fit(x_train, y_train)
    print(clf.tree_.max_depth) #max_depth of the trees

    #for train dataset
    y_train_pred = clf.predict(x_train)
    train_acc.append(metrics.accuracy_score(y_train, y_train_pred))

    #for test dataset
    y_test_pred = clf.predict(x_test)

    test_acc.append(metrics.accuracy_score(y_test, y_test_pred))
    prec.append(metrics.precision_score(y_test, y_test_pred, pos_label = 'M'))
    recall.append(metrics.recall_score(y_test, y_test_pred, pos_label = 'M'))

#compute average of 20 performance
mean_train_acc = np.mean(train_acc)
mean_test_acc = np.mean(test_acc)
mean_prec = np.mean(prec)
mean_recall = np.mean(recall)

rs_table.loc['DT1']['train accuracy'] = mean_train_acc
rs_table.loc['DT1']['test accuracy'] = mean_test_acc
rs_table.loc['DT1']['precision'] = mean_prec
rs_table.loc['DT1']['recall'] = mean_recall

print(rs_table)
```

```
7
6
5
6
6
6
5
7
6
6
7
7
7
6
6
6
6
6
5
```

```
7
    train accuracy test accuracy precision    recall
DT1              1      0.926023  0.895843  0.909528
DT2            NaN           NaN       NaN       NaN
```

In [20]:
```
#we limit the depth of the tree
clf = DecisionTreeClassifier(criterion = "entropy", max_depth = 4)

print(clf)

#reinitialise clean list
train_acc = []
test_acc = []
prec = []
recall = []

for i in range(20):
    (x_train, x_test, y_train, y_test) = train_test_split(x, y, test_size = 0.3)
    clf = clf.fit(x_train, y_train)

    #for train dataset
    y_train_pred = clf.predict(x_train)
    train_acc.append(metrics.accuracy_score(y_train, y_train_pred))

    #for test dataset
    y_test_pred = clf.predict(x_test)

    test_acc.append(metrics.accuracy_score(y_test, y_test_pred))
    prec.append(metrics.precision_score(y_test, y_test_pred, pos_label = 'M'))
    recall.append(metrics.recall_score(y_test, y_test_pred, pos_label = 'M'))

#compute average of 20 performance
mean_train_acc = np.mean(train_acc)
mean_test_acc = np.mean(test_acc)
mean_prec = np.mean(prec)
mean_recall = np.mean(recall)

rs_table.loc['DT2']['train accuracy'] = mean_train_acc
rs_table.loc['DT2']['test accuracy'] = mean_test_acc
rs_table.loc['DT2']['precision'] = mean_prec
rs_table.loc['DT2']['recall'] = mean_recall

print(rs_table)
```

```
DecisionTreeClassifier(ccp_alpha=0.0, class_weight=None, criterion='entropy',
                       max_depth=4, max_features=None, max_leaf_nodes=None,
                       min_impurity_decrease=0.0, min_impurity_split=None,
                       min_samples_leaf=1, min_samples_split=2,
                       min_weight_fraction_leaf=0.0, presort='deprecated',
                       random_state=None, splitter='best')
    train accuracy test accuracy precision    recall
DT1              1      0.926023  0.895843  0.909528
DT2       0.982915      0.938012  0.913976  0.923981
```