1. Do get the data file and take 100 data from the notepad file to csv. Next by taking different technique do calculate the efficiency of the model. Data should represent minimum 100 records and all classes should be present in the data set.

In [1]:

```
import numpy as np
import pandas as pd
from sklearn.impute import SimpleImputer
from sklearn.preprocessing import MinMaxScaler
from sklearn import model_selection
from sklearn.ensemble import BaggingClassifier
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import AdaBoostClassifier
```

In [7]:

```
df = pd.read_csv(r'C:\Users\Nirmalya Majhi\Desktop\Advanced IT Workshop\cancer.csv')
df.drop(['Sample Code Number', 'id'],axis = 1, inplace= True)
impute_value = df.values
imputer = SimpleImputer()
imputeData = imputer.fit_transform(impute_value)
scaler = MinMaxScaler(feature_range=(0,1))
normalizedData = scaler.fit_transform(impute_value)
X = normalizedData[:,0:9]
Y = normalizedData[:,9]
kfold = model_selection.KFold(n_splits=10, random_state=7, shuffle=True)
cart = DecisionTreeClassifier()
num_trees = 100
model = BaggingClassifier(base_estimator=cart, n_estimators=num_trees, random_state=7)
results = model_selection.cross_val_score(model, X, Y, cv = kfold)
print("For 100 records (%): ",round(results.mean(),2)*100)
seed = 7
num_trees = 70
kfold = model_selection.KFold(n_splits=10, random_state=7, shuffle=True)
model = AdaBoostClassifier(n_estimators=num_trees, random_state=seed)
results = model_selection.cross_val_score(model, X, Y, cv = kfold)
print("For 70 records (%): ",round(results.mean(),2)*100)
```

```
For 100 records (%):  97.0
For 70 records (%):  91.0
```

Discussion: Ensemble models combine the decisions from multiple models to improve the overall performance. We can see increasing the number of records, the accuracy of our model increases.

In [ ]: