Bagged Decision Trees for Classification

In [3]:

```
from pandas import read_csv
from sklearn.model_selection import KFold
from sklearn.model_selection import cross_val_score
from sklearn.ensemble import BaggingClassifier
from sklearn.tree import DecisionTreeClassifier
```

In [5]:

```
filename = 'pima-indians-diabetes.data.csv'
names = ['preg','plas','pres','skin','test','mass','pedi','age','class']
dataframe = read_csv(filename,names=names)
```

In [6]:

```
array = dataframe.values
X = array[:,0:8]
Y = array[:,8]
seed = 7
```

In [9]:

```
kfold = KFold(n_splits=10)
cart = DecisionTreeClassifier()
num_trees = 100
model = BaggingClassifier(n_estimators=num_trees)
results = cross_val_score(model,X,Y,cv=kfold)
print(results.mean())
```

```
0.7720437457279563
```

Random Forest Classification

In [13]:

```
from sklearn.ensemble import RandomForestClassifier
X = array[:,0:8]
Y = array[:,8]
num_trees = 1000
max_features = 20
kfold = KFold(n_splits=10)
model = RandomForestClassifier(n_estimators=num_trees,max_features=max_feature
results = cross_val_score(model,X,Y,cv=kfold)
print(results.mean())
```

```
0.7629357484620644
```

AdaBoost Classification

In [16]:

```
from sklearn.ensemble import AdaBoostClassifier
filename = 'pima-indians-diabetes.data.csv'
names = ['preg','plas','pres','skin','test','mass','pedi','age','class']
dataframe = read_csv(filename,names=names)
array = dataframe.values

X = array[:,0:8]
Y = array[:,8]
num_trees = 30
seed = 7
kfold = KFold(n_splits=10)
model = AdaBoostClassifier(n_estimators=num_trees)
results = cross_val_score(model,X,Y,cv=kfold)
print(results.mean())
```

```
0.760457963089542
```

Stacking Ensemble for classification

In [17]:

```
from sklearn.linear_model import LogisticRegression
from sklearn.tree import DecisionTreeClassifier
from sklearn.svm import SVC
from sklearn.ensemble import VotingClassifier
```

In [18]:

```
filename = 'pima-indians-diabetes.data.csv'
names = ['preg','plas','pres','skin','test','mass','pedi','age','class']
dataframe = read_csv(filename,names=names)
array = dataframe.values

kfold = KFold(n_splits=10)
# create the sub models
estimators = []
model1 = LogisticRegression (max_iter=500)
estimators.append (('logistic', model1))
model2 = DecisionTreeClassifier()
estimators.append(('cart', model2))
model3 = SVC()
estimators.append(('svm', model3))
print(results.mean())
```

```
0.760457963089542
```