**18CSC305J-Artificial Intelligence**

**LAB 12 – To develop an ensemble model for an application**

**Aim:** To develop an ensemble model for an application, namely

1. Bagged decision tree

2. Random forest

3. Extra trees

**Description:**

1. Bagging (Bootstrap Aggregation) is used when our goal is to reduce the  variance of a decision tree. Here idea is to create several subsets of data  from training sample chosen randomly with replacement. Now, each  collection of subset data is used to train their decision trees.

2. Random forest is a Supervised Machine Learning Algorithm that is used  widely in Classification and Regression problems. It builds decision trees on  different samples and takes their majority vote for classification and average  in case of regression.

3. Extra Trees is an ensemble machine learning algorithm that combines the  predictions from many decision trees. It is related to the widely used random  forest algorithm.

**Code:**

**Bagged decision tree:**

import pandas as pd

import numpy as np

from sklearn import model\_selection

from sklearn.ensemble import BaggingClassifier

from sklearn.tree import DecisionTreeClassifier

url = "diabetes.csv"

names = ['preg', 'plas', 'pres', 'skin', 'test', 'mass', 'pedi',  'age','class']

dataframe = pd.read\_csv(url, names=names)

dataframe = dataframe.iloc[1:]

array = dataframe.values

X = array[:,0:8]

Y = array[:,8]

seed = 7

kfold = model\_selection.KFold(n\_splits=10, random\_state=seed,  shuffle=True)

cart = DecisionTreeClassifier()

num\_trees = 100

dataframe = dataframe.replace(r'^\s\*$', np.nan, regex=True) model = BaggingClassifier(base\_estimator=cart, n\_estimators=num\_trees,  random\_state=seed)

results = model\_selection.cross\_val\_score(model, X, Y, cv=kfold,  error\_score='raise')

print(results)

print(results.mean())

**Random Forest:**

import pandas

from sklearn import model\_selection

from sklearn.ensemble import RandomForestClassifier  url = "diabetes.csv"

names = ['preg', 'plas', 'pres', 'skin', 'test', 'mass', 'pedi',  'age','class']

dataframe = pandas.read\_csv(url, names=names)

dataframe = dataframe.iloc[1:]

array = dataframe.values

X = array[:,0:8]

Y = array[:,8]

seed = 7

num\_trees = 100

max\_features = 3

kfold = model\_selection.KFold(n\_splits=10)

model = RandomForestClassifier(n\_estimators=num\_trees,  max\_features=max\_features)

results = model\_selection.cross\_val\_score(model, X, Y, cv=kfold) print(results)

print(results.mean())

**Extra Trees:**

import pandas

from sklearn import model\_selection

from sklearn.ensemble import RandomForestClassifier from sklearn.ensemble import ExtraTreesClassifier

url = "diabetes.csv"

names = ['preg', 'plas', 'pres', 'skin', 'test', 'mass', 'pedi', 'age','class']

dataframe = pandas.read\_csv(url, names=names)

dataframe = dataframe.iloc[1:]

array = dataframe.values

X = array[:,0:8]

Y = array[:,8]

seed = 7

num\_trees = 100

max\_features = 7

kfold = model\_selection.KFold(n\_splits=10)

model = ExtraTreesClassifier(n\_estimators=num\_trees,

max\_features=max\_features)

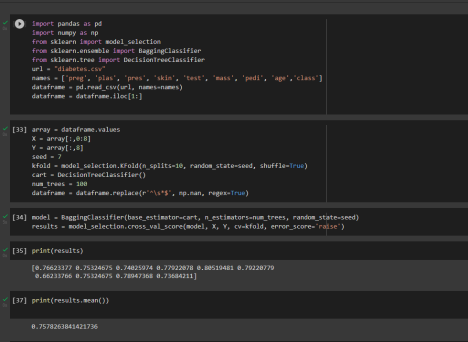
results = model\_selection.cross\_val\_score(model, X, Y,

cv=kfold) print(results)

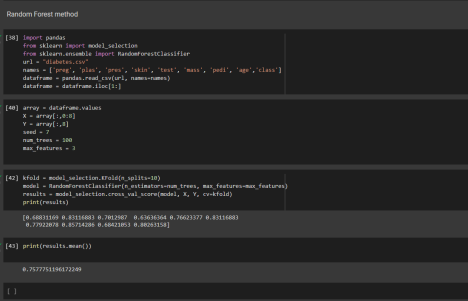
print(results.mean())

**OUTPUT :**

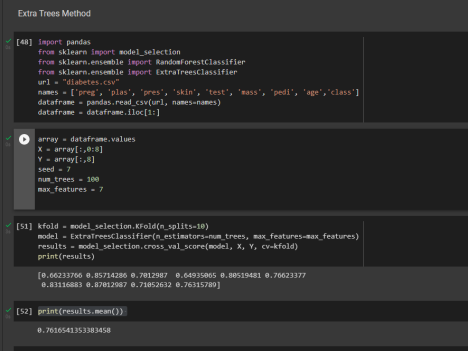
**Bagged Decision trees :**

****

**Random forest :**

****

**Extra trees:**

****

**Result:** Hence we have successfully developed an ensemble model for an application