**Practical-4**

**Aim**: **There is a need to advice the 12th standard students that which college**

**he/she should choose for engineering education. Decide the features to**

**use for grading the engineering college. Prepare the data set. Write a**

**program to apply random forest algorithm and suggest the best suited**

**college for 12th standard students.**

**Program:**

# To import libraries

import pandas as pd

from sklearn.metrics import mean\_absolute\_error

from sklearn.model\_selection import train\_test\_split

# splitting data into labels and features

labels = list(college\_data['Grad\_Rate'])

features = list(zip(college\_data['Apps'], college\_data['Accept'], college\_data['Enroll'],

college\_data['Top10perc'], college\_data['Top25perc'], college\_data['F.Undergrad'],

college\_data['P.Undergrad'], college\_data['Outstate'], college\_data['Room.Board'], college\_data['Books'], college\_data['Personal'], college\_data['PhD'], college\_data['Terminal'], college\_data['S.F.Ratio'], college\_data['perc.alumni'], college\_data['Expend']))

**#** training the model of Random Forest

train\_X, val\_X, train\_y, val\_y = train\_test\_split(features, labels, random\_state=1)

from sklearn.ensemble import RandomForestRegressor

rf\_model = RandomForestRegressor(random\_state=3)

rf\_model.fit(train\_X,train\_y)

predictions=rf\_model.predict(val\_X)

# Calculating mean absolute error and printing it

rf\_val\_mae = mean\_absolute\_error(predictions,val\_y)

print("Validation MAE for Random Forest Model: {}".format(rf\_val\_mae))



# predicting the best grade rate college for random values

print(rf\_model.predict([[1500, 1200, 500, 30, 60, 1855, 640, 12800, 4590, 780, 1390, 70, 55, 13.5, 25, 9549]]))

