Random forest(Gini,Entropy)

# example code

|  |
| --- |
| # Import list  import pandas as pd  import numpy as np  from sklearn.model\_selection import train\_test\_split  from sklearn.ensemble import RandomForestClassifier  from sklearn.metrics import accuracy\_score # accuracy\_check function  # load data  data = pd.read\_csv("../[01]data\_set/Housing.csv")  # Check data.head  print(data.head(2))  # Converting some string-type data, such as yes no inside the data, into an integer type.  mapping\_dict\_bool = {'yes' : True,'no' : False,}  mapping\_dict\_furnished = {'furnished' : 1, 'semi-furnished' : 0, 'unfurnished' : -1}  bool\_list = ['mainroad','guestroom','basement','hotwaterheating','airconditioning']  # Change yes or no string data to 1 or 0  for x in bool\_list:      data[x] = data[x].apply(lambda x : mapping\_dict\_bool[x])  # Change furnishingstatus data to 1 , 0 , -1  data['furnishingstatus'] = data['furnishingstatus'].apply(lambda x : mapping\_dict\_furnished[x])  # Set X, Y data.  X = data[['area', 'price', 'bathrooms', 'stories','mainroad','guestroom','basement','hotwaterheating','airconditioning','furnishingstatus']]  Y = data['bedrooms']  # Split dataset into train set and val set.  train\_x, test\_x, train\_y, test\_y = train\_test\_split(X,Y,test\_size = 0.2,random\_state = 42)  '''  Make clf\_20,clf\_100, clf\_20\_e,clf\_100\_e RandomForestClassifier      ex):      clf\_20 : RandomForestClassifier(criterion = 'gini', n\_estimators = 20 , max\_depth = 4)      clf\_100\_e : RandomForestClassifier(criterion = 'entropy', n\_estimators = 100 , max\_depth = 4)  '''  # Make RandomForestClassifier and Train model with Train dataset.  clf\_20 = RandomForestClassifier(criterion='gini', n\_estimators=20, max\_depth=4,random\_state=0)  clf\_20.fit(train\_x,train\_y)  # Make predict set of model with val dataset. and check the Acccuracy of the predict set.  predict20 = clf\_20.predict(test\_x)  print("gini : estimators = 10 : {}".format(accuracy\_score(test\_y,predict20)))  # Make RandomForestClassifier and Train model with Train dataset.  clf\_100 = RandomForestClassifier(criterion='gini', n\_estimators=100, max\_depth=4,random\_state=0)  clf\_100.fit(train\_x,train\_y)  # Make predict set of model with val dataset. and check the Acccuracy of the predict set.  predict100 = clf\_100.predict(test\_x)  print("gini : estimators = 100 : {}".format(accuracy\_score(test\_y,predict100)))  # Make RandomForestClassifier and Train model with Train dataset.  clf\_20\_e = RandomForestClassifier(criterion='entropy', n\_estimators=20, max\_depth=4,random\_state=0)  clf\_20\_e.fit(train\_x,train\_y)  # Make predict set of model with val dataset. and check the Acccuracy of the predict set.  predict20\_e = clf\_20\_e.predict(test\_x)  print("entropy : estimators = 10 : {}".format(accuracy\_score(test\_y,predict20\_e)))  # Make RandomForestClassifier and Train model with Train dataset.  clf\_100\_e = RandomForestClassifier(criterion='entropy', n\_estimators=100, max\_depth=4,random\_state=0)  clf\_100\_e.fit(train\_x,train\_y)  # Make predict set of model with val dataset. and check the Acccuracy of the predict set.  predict100\_e = clf\_100\_e.predict(test\_x)  print("entropy : estimators = 100 : {}".format(accuracy\_score(test\_y,predict100\_e))) |

# testing result

|  |
| --- |
|  |