Bagging classifier

# example code

|  |
| --- |
| # Import list  from sklearn.tree import DecisionTreeClassifier  from sklearn.model\_selection import train\_test\_split  from sklearn.ensemble import BaggingClassifier  import numpy as np  import pandas as pd  # Load dataset  Data = pd.read\_csv("../[01]data\_set/diabetes.csv")  # Split dataset in features and target variable  X = Data[['Pregnancies','Insulin','BMI','Age','Glucose','BloodPressure','DiabetesPedigreeFunction']]  Y = Data.Outcome  # Split dataset into train set and val set  X\_train, X\_val, y\_train, y\_val = train\_test\_split(X, Y, test\_size=0.2, random\_state=1)  # Create Bagging Decision Tree classifier object.  model = BaggingClassifier(base\_estimator=DecisionTreeClassifier(), n\_estimators=10 , bootstrap=True)  # Model train.  model.fit(X\_train,y\_train)  # Create Decision Tree classifer object (default = gini)  decision\_tree = DecisionTreeClassifier()  # Train Decision Tree classifer  decision\_tree = decision\_tree.fit(X\_train,y\_train)  print("decision\_tree\_model score(train\_set): {}".format(decision\_tree.score(X\_train, y\_train)))  print("decision\_tree\_model score(val\_set): {}".format(decision\_tree.score(X\_val,y\_val)))  # Print Bagging\_model score.  print("Bagging\_model score(train\_set): {}".format(model.score(X\_train, y\_train)))  print("Bagging\_model score(val\_set): {}".format(model.score(X\_val,y\_val)))  # Make prediction test data.  test\_data = np.array([[5,183,50,175,30.1,0.398,32]])  # Prediction of test data.  bagging\_test\_result = model.predict(test\_data)  decision\_tree\_test\_result = decision\_tree.predict(test\_data)  # Test Result  print("test data result : \n bagging : {}\n decision\_tree : {}".format(bagging\_test\_result,decision\_tree\_test\_result)) |

# testing result

|  |
| --- |
| decision\_tree\_model score(train\_set): 1.0  decision\_tree\_model score(val\_set): 0.7272727272727273  Bagging\_model score(train\_set): 0.988599348534202  Bagging\_model score(val\_set): 0.7922077922077922  test data result :  bagging : [1]  decision\_tree : [1]  #result can be change. |