Decision Tree (Gini,Entropy)

# example code

|  |
| --- |
| # import list  import pandas as pd  from sklearn.tree import DecisionTreeClassifier # Import Decision Tree  from sklearn.model\_selection import train\_test\_split # Import train\_test\_split function  from sklearn import metrics #Import scikit-learn metrics module  from sklearn.tree import export\_graphviz  from sklearn import tree  from sklearn.metrics import accuracy\_score  import graphviz  # First you need to download data set file. check [01]data\_set/diabetes.csv download.txt  # 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 Decision Tree classifer object (default = gini)  decision\_tree = DecisionTreeClassifier()  # Train Decision Tree classifer  decision\_tree = decision\_tree.fit(X\_train,y\_train)  # Predict of val data.  y\_pred\_gini = decision\_tree.predict(X\_val)  # Create Decision Tree classifer object ( criterion="entropy" )  decision\_tree\_entropy = DecisionTreeClassifier( criterion="entropy" , max\_depth = 4)  # Train Decision Tree classifer  decision\_tree\_entropy = decision\_tree\_entropy.fit(X\_train,y\_train)  # Predict of val data.  y\_pred\_entropy = decision\_tree\_entropy.predict(X\_val)  # Model Accuracy check  print("Accuracy(Gini):",metrics.accuracy\_score(y\_val, y\_pred\_gini))  print("Accuracy(entropy):",metrics.accuracy\_score(y\_val, y\_pred\_entropy))  # def for draw decision\_tree  def show\_trees(tree):      dot\_data = export\_graphviz(tree, out\_file=None, class\_names=["1", "0"],                      feature\_names = ['Pregnancies','Insulin','BMI','Age','Glucose','BloodPressure','DiabetesPedigreeFunction'],                      precision=3, filled=True, rounded=True, special\_characters=True)      pred = tree.predict(X\_val)      print('Accuracy : {:.2f} %'.format(accuracy\_score(y\_val, pred) \* 100))      graph = graphviz.Source(dot\_data)      return graph  gini\_graph = show\_trees(decision\_tree)  entropy\_graph = show\_trees(decision\_tree\_entropy)  # Show decision\_tree in JUPYTER  # gini\_graph  entropy\_graph  # make decision\_tree pdf  gini\_graph.render("Gini")  entropy\_graph.render("Entropy") |

# testing result

|  |
| --- |
| Accuracy(Gini): 0.7272727272727273  Accuracy(entropy): 0.8051948051948052  Accuracy : 72.73 %  Accuracy : 80.52 % |