logistic regression

# example code

|  |
| --- |
| import pandas as pd  import matplotlib.pyplot as plt  import numpy as np  from sklearn.model\_selection import train\_test\_split  from sklearn.linear\_model import LogisticRegression  from sklearn.preprocessing import StandardScaler  from sklearn.metrics import accuracy\_score  from sklearn.datasets import load\_breast\_cancer  # 데이터 셋 로드  cancer = load\_breast\_cancer()  # 데이터 프레임 만들기  df = pd.DataFrame(cancer['data'], columns=cancer['feature\_names'])  df['target'] = cancer['target']  df.head()  # standardization  scaler = StandardScaler()  scaled = scaler.fit\_transform(df.drop('target', axis=1))  # train\_test\_split  x\_train, x\_valid, y\_train, y\_valid = train\_test\_split(scaled, df['target'],  random\_state=10)  # logistic regression  lr\_clf = LogisticRegression()  lr\_clf.fit(x\_train, y\_train)  pred = lr\_clf.predict(x\_valid)  # 정확도 측정  accuracy\_score(y\_valid, pred)  from sklearn.metrics import confusion\_matrix  from IPython.display import Image  cm = confusion\_matrix(y\_valid, pred)  import seaborn as sns  sns.heatmap(cm, annot=True, annot\_kws={"size": 20}, cmap='YlOrBr')  plt.xlabel('Predicted', fontsize=20)  plt.ylabel('Actual', fontsize=20)  TN, FP, FN, TP = cm.ravel()  tn, fp, fn, tp = confusion\_matrix([0, 1, 0, 1], [1, 1, 1, 0]).ravel()  precision = TP / (FP + TP)  from sklearn.metrics import precision\_score  precision\_score(y\_valid, pred)  # 검증  recall = TP / (FN + TP)  2 \* (precision \* recall) / (precision + recall)  from sklearn.metrics import f1\_score  f1\_score(y\_valid, pred) |

# testing result

|  |
| --- |
| 0.9666666666666667 |