|  |  |  |  |
| --- | --- | --- | --- |
| **5주. Decision Tree, RF, SVM** | | | |
| 학번 | 32200327 | 이름 | 김경민 |

**PimaIndiansDiabetes dataset을 가지고 Classification 을 하고자 한다.** (마지막의 diabetes 컬럼이 class label 임)

|  |
| --- |
| Q1 (4점) scikit-learn에서 제공하는 DecisionTree, RandonForest, support vector machine 알고리즘를 이용하여 **PimaIndiansDiabetes dataset**에 대한 분류 모델을 생성하고 accuracy를 비교하시오.  - 각 알고리즘의 hyper parameter 의 값은 default value를 이용한다. |

**Source code :**

|  |
| --- |
| // source code 의 폰트는 Courier10 BT Bold으로 하시오  from sklearn.tree import DecisionTreeClassifier  from sklearn.ensemble import RandomForestClassifier  from sklearn import svm  from sklearn.model\_selection import train\_test\_split  import pandas as pd  df = pd.read\_csv('C:/Users/user/PycharmProjects/deepLearning/data/PimaIndiansDiabetes.csv')  df\_X= df.loc[:, df.columns!= 'diabetes']  df\_y= df['diabetes']  # Split the data into training/testing sets  train\_X, test\_X, train\_y, test\_y= \  train\_test\_split(df\_X, df\_y, test\_size=0.3,\  random\_state=1234)  model\_DecisionTree = DecisionTreeClassifier(random\_state=1234)  model\_RandomForest = RandomForestClassifier(random\_state=1234)  model\_SVM = svm.SVC()  model\_DecisionTree.fit(train\_X, train\_y)  model\_RandomForest.fit(train\_X, train\_y)  model\_SVM.fit(train\_X, train\_y)  print('Train accuracy(DecisionTree) :', model\_DecisionTree.score(train\_X, train\_y))  print('Test accuracy(DecisionTree) :', model\_DecisionTree.score(test\_X, test\_y))  print()  print('Train accuracy(RandomForest) :', model\_RandomForest.score(train\_X, train\_y))  print('Test accuracy(RandomForest) :', model\_RandomForest.score(test\_X, test\_y))  print()  print('Train accuracy(SVM) :', model\_SVM.score(train\_X, train\_y))  print('Test accuracy(SVM) :', model\_SVM.score(test\_X, test\_y))  print() |

**실행화면 캡쳐:**

****

|  |
| --- |
| Q2. (3점) 다음의 조건에 따라 support vector machine 알고리즘를 이용하여 **PimaIndiansDiabetes dataset**에 대한 분류 모델을 생성하고 accuracy를 비교하시오.  - hyper parameter 중 kernel 에 대해 linear, poly, rbf, sigmoid를 각각 테스트하여 어떤 kernel 이 가장 높은 accuracy를 도출하는지 확인하시오. |

**Source code :**

|  |
| --- |
| // source code 의 폰트는 Courier10 BT Bold으로 하시오  from sklearn import svm  from sklearn.model\_selection import train\_test\_split  import pandas as pd  df = pd.read\_csv('C:/Users/user/PycharmProjects/deepLearning/data/PimaIndiansDiabetes.csv')  df\_X= df.loc[:, df.columns!= 'diabetes']  df\_y= df['diabetes']  # Split the data into training/testing sets  train\_X, test\_X, train\_y, test\_y= \  train\_test\_split(df\_X, df\_y, test\_size=0.3,\  random\_state=1234)  # Define learning model (kernel) ############  for i in ('linear','poly' , 'rbf' , 'sigmoid'):  print(f'kernel = {i}')  model = svm.SVC(kernel = i)  # Train the model using the training sets  model.fit(train\_X, train\_y)  # performance evaluation  print('Train accuracy :', model.score(train\_X, train\_y))  print('Test accuracy :', model.score(test\_X, test\_y))  print() |

**실행화면 캡쳐:**



|  |
| --- |
| Q3. (3점) 다음의 조건에 따라 Random Forest 알고리즘를 이용하여 **PimaIndiansDiabetes dataset**에 대한 분류 모델을 생성하고 accuracy를 비교하시오.  -다음의 hyper parameter를 테스트 하시오  . n\_estimators : 100, 200, 300, 400, 500  . max\_features : 1, 2, 3, 4, 5  어떤 조합이 가장 높은 accuracy를 도출하는지 확인하시오. |

**Source code :**

|  |
| --- |
| // source code 의 폰트는 Courier10 BT Bold으로 하시오  from sklearn.ensemble import RandomForestClassifier  from sklearn.model\_selection import train\_test\_split  import pandas as pd  df = pd.read\_csv('C:/Users/user/PycharmProjects/deepLearning/data/PimaIndiansDiabetes.csv')  print(df.head())  print(df.columns)  # column names  df\_X= df.loc[:, df.columns!= 'diabetes']  df\_y= df['diabetes']  # Split the data into training/testing sets  train\_X, test\_X, train\_y, test\_y= \  train\_test\_split(df\_X, df\_y, test\_size=0.3,\  random\_state=1234)  best\_acc=0  best\_n\_estimators = 0  best\_max\_features = 0  for i in (100,200,300,400,500):  print(f'n\_estimators = {i}')  for j in (1,2,3,4,5):  print(f'max\_features = {j}')  model = RandomForestClassifier(n\_estimators=i, max\_features=j, random\_state=1234)  # Train the model using the training sets  model.fit(train\_X, train\_y)  train\_acc = model.score(train\_X, train\_y)  test\_acc = model.score(test\_X, test\_y)  # performance evaluation  print('Train accuracy :', train\_acc)  print('Test accuracy :', test\_acc)  if (test\_acc > best\_acc):  best\_n\_estimators = i  best\_max\_features = j  best\_acc = test\_acc  print()  print(f'Best Test accuracy : {best\_acc}\nBest n\_estimators : {best\_n\_estimators}'  f'\nBest max\_features : {best\_max\_features}') |

**실행화면 캡쳐:**

