스마트시스템 이론과 응용 Report #6 기계학습 프로그래밍

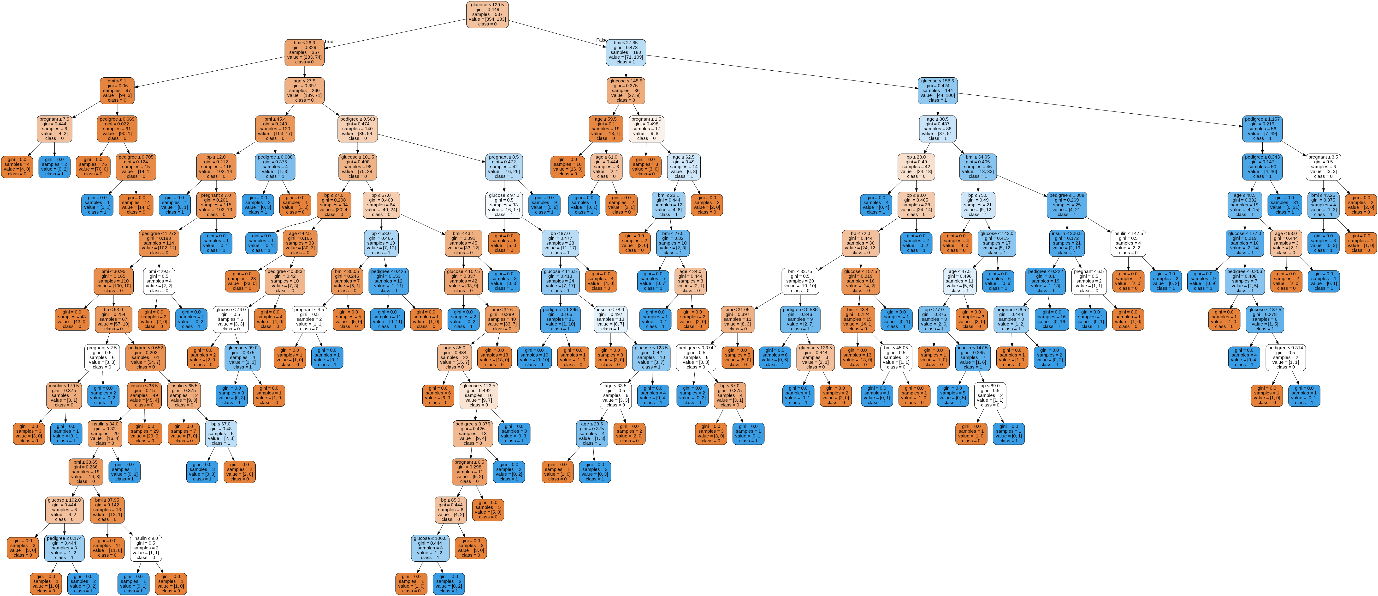
2019451170/ 조민근

1. 아래의 URL을 참고하여 피마 인디언 당뇨병에 대한 Decision Tree를 만들고 임의의 데이터를 입력하여 의사결정이 이루어진 결과를 보이시오.

\* 참고 URL: https://www.datacamp.com/community/tutorials/decision-tree-classification-python

Index(['6', '148', '72', '35', '0', '33.6', '0.627', '50', '1'], dtype='object')

Accuracy: 0.6796536796536796



2. 사이킷런(Scikit Learn)에 제공하는 Iris Dataset을 이용하여 k-Means 클러스터링을 수행하고 임의의 데이터를 입력하여 어떤 클러스터에 할당되는지 결과를 보이시오.

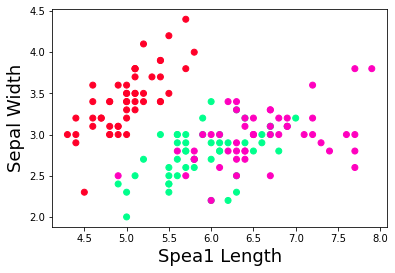
\* 참고: Google에서 Iris Dataset을 이용한 Scikit Learn의 k-Means 클러스터링을 검색하면 다양한 형태로 작성된 파이썬(Python) 코드를 확인할 수 있음.

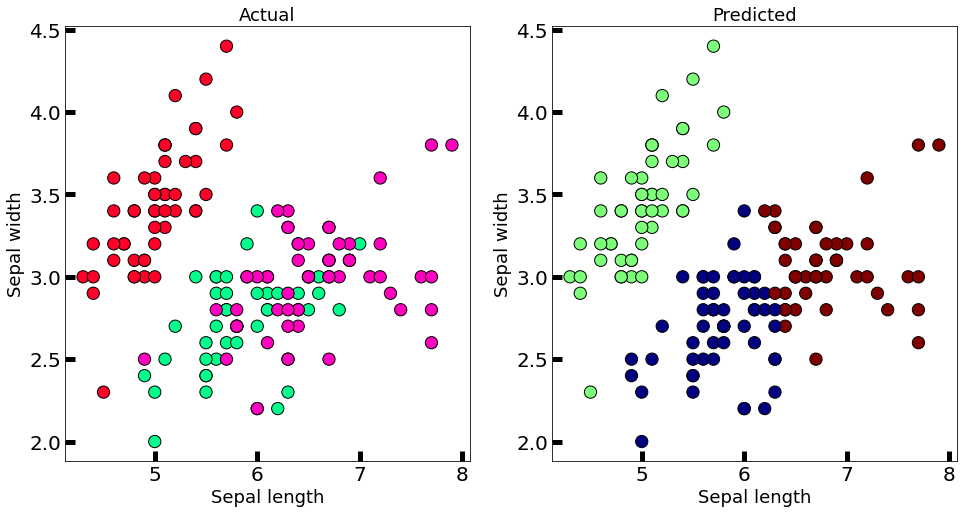
[[5.77358491 2.69245283]

[5.006 3.428 ]

[6.81276596 3.07446809]]

Text(0.5, 1.0, 'Predicted')





소스 코드

1.

# Load libraries

import pandas as pd

from sklearn.tree import DecisionTreeClassifier # Import Decision Tree Classifier

from sklearn.model\_selection import train\_test\_split # Import train\_test\_split function

from sklearn import metrics #Import scikit-learn metrics module for accuracy calculation

diabetes = pd.read\_csv('/content/gdrive/My Drive/Colab Notebooks/diabetes.csv')

print(diabetes.columns)

col\_names = ['pregnant', 'glucose', 'bp', 'skin', 'insulin', 'bmi', 'pedigree', 'age', 'label']

# load dataset

pima = pd.read\_csv('/content/gdrive/My Drive/Colab Notebooks/diabetes.csv', header=None, names=col\_names)

pima.head()

#split dataset in features and target variable

feature\_cols = ['pregnant', 'insulin', 'bmi', 'age','glucose','bp','pedigree']

X = pima[feature\_cols] # Features

y = pima.label # Target variable

# Split dataset into training set and test set

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.3, random\_state=1) # 70% training and 30% test

# Create Decision Tree classifer object

clf = DecisionTreeClassifier()

# Train Decision Tree Classifer

clf = clf.fit(X\_train,y\_train)

#Predict the response for test dataset

y\_pred = clf.predict(X\_test)

# Model Accuracy, how often is the classifier correct?

print("Accuracy:",metrics.accuracy\_score(y\_test, y\_pred))

from sklearn.tree import export\_graphviz

from sklearn.externals.six import StringIO

from IPython.display import Image

import pydotplus

dot\_data = StringIO()

export\_graphviz(clf, out\_file=dot\_data,

                filled=True, rounded=True,

                special\_characters=True,feature\_names = feature\_cols,class\_names=['0','1'])

graph = pydotplus.graph\_from\_dot\_data(dot\_data.getvalue())

graph.write\_png('diabetes.png')

Image(graph.create\_png())

2.

from sklearn import datasets

import matplotlib.pyplot as plt

import pandas as pd

from sklearn.cluster import KMeans

iris = datasets.load\_iris()

X = iris.data[:, :2]

y = iris.target

plt.scatter(X[:,0], X[:,1], c=y, cmap='gist\_rainbow')

plt.xlabel('Spea1 Length', fontsize=18)

plt.ylabel('Sepal Width', fontsize=18)

km = KMeans(n\_clusters = 3, n\_jobs = 4, random\_state=21)

km.fit(X)

centers = km.cluster\_centers\_

print(centers)

#this will tell us to which cluster does the data observations belong.

new\_labels = km.labels\_

# Plot the identified clusters and compare with the answers

fig, axes = plt.subplots(1, 2, figsize=(16,8))

axes[0].scatter(X[:, 0], X[:, 1], c=y, cmap='gist\_rainbow',

edgecolor='k', s=150)

axes[1].scatter(X[:, 0], X[:, 1], c=new\_labels, cmap='jet',

edgecolor='k', s=150)

axes[0].set\_xlabel('Sepal length', fontsize=18)

axes[0].set\_ylabel('Sepal width', fontsize=18)

axes[1].set\_xlabel('Sepal length', fontsize=18)

axes[1].set\_ylabel('Sepal width', fontsize=18)

axes[0].tick\_params(direction='in', length=10, width=5, colors='k', labelsize=20)

axes[1].tick\_params(direction='in', length=10, width=5, colors='k', labelsize=20)

axes[0].set\_title('Actual', fontsize=18)

axes[1].set\_title('Predicted', fontsize=18)