**인공지능**

Assignment 3

소프트웨어학부

20163231

신승은

**<문제의 답>** 1번 :

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **Parameter 1** | **Parameter 2** | **Precision** | **Recall** | **F1** |
| Decision Tree | criterion  =‘entropy’ | min\_impurity\_split=0.1 | 0.86 | 0.87 | 0.86 |
| min\_impurity\_split=0.2 | 0.86 | 0.92 | 0.87 |
| min\_impurity\_split=0.3 | 0.85 | 0.85 | 0.85 |
| min\_impurity\_split=0.4 | 0.90 | 0.92 | 0.90 |
| min\_impurity\_split=0.5 | 0.90 | 0.87 | 0.88 |
| criterion  =‘gini’ | min\_impurity\_split=0.1 | 0.92 | 0.91 | 0.91 |
| min\_impurity\_split=0.2 | 0.92 | 0.91 | 0.91 |
| min\_impurity\_split=0.3 | 0.86 | 0.87 | 0.86 |
| min\_impurity\_split=0.4 | 0.91 | 0.93 | 0.91 |
| min\_impurity\_split=0.5 | 0.35 | 0.5 | 0.41 |
| K Nearest Neighbor | weights  =‘uniform’ | k=1 | 0.80 | 0.79 | 0.79 |
| k=5 | 0.87 | 0.87 | 0.87 |
| k=10 | 0.88 | 0.88 | 0.88 |
| k=20 | 0.88 | 0.84 | 0.86 |
| k=30 | 0.92 | 0.92 | 0.92 |
| weights  =‘distance’ | k=1 | 0.81 | 0.84 | 0.82 |
| k=5 | 0.91 | 0.89 | 0.89 |
| k=10 | 0.86 | 0.84 | 0.85 |
| k=20 | 0.90 | 0,87 | 0,88 |
| k=30 | 0.93 | 0.91 | 0.91 |

**<문제의 답>**2번

1번의 결과를 해석한다면?

Decision tree를 이용한 방식이 비교했을 때 훨씬 더 정확도가 뛰어났습니다. decision트트리 만들었을 때 min\_impurity\_split이 0에 가까울수록 깊이가 싶었지만, 과적합이 일어날수있어서 0.1부터는 branch가 많아지고, 값도 커질수록 더 적은 branch를 가집니다. K Nearest Neibor에서 weight을 똑같이 해 줬을 경우와 weight을 가까울수록 더 주었을 경우는 그닥 차이가 없었습니다.

**<문제의 답>** 3번

Data X의 속성 중 가장 중요한 속성은?(Decision Tree 활용)

Detergents\_paper

**<문제의 답>**4번

3번의 결과를 해석한다면?

Detergents\_paper속성을 기준으로 호레카와 리테일을 적절한 데이터들을 활용할 수 있었습니다.

**<소스코드>**

>> Decision Tree

*'''  
인공지능 2019  
과제 3  
pandas, numpy, sklearn 등의 library는 PPT 자료 참고하여 설치  
'''*import pandas as pd  
import numpy as np  
from sklearn import tree  
from sklearn.model\_selection import train\_test\_split  
from sklearn.metrics import precision\_score, recall\_score, f1\_score  
import graphviz  
#import sklearn.neighbors import KNeighborsClassifier  
  
# prepare data  
data = pd.read\_csv('C:/Users/tmddms2292/Desktop/인공지능 (2019-1)/assignment3/as3.csv') # read csv file in pandas dataframe  
data1 = np.array(data) # change dataframe to numpy array  
class\_names = ['Horeca', 'Retail'] # python list of class name  
feature\_names = data.columns # get list of feature names from pandas dataframe  
  
X = list(map(lambda x: x[1:], data1))  
Y = list(map(lambda x: x[0], data1))  
  
# random split  
x\_train, x\_test, y\_train, y\_test = train\_test\_split(X, Y, test\_size=0.2)  
  
# 하단에 작성  
  
criterion = 'entropy'  
min\_impurity\_split = 0.5 #0.1~0.5  
clf = tree.DecisionTreeClassifier(criterion = criterion,  
 min\_impurity\_split = min\_impurity\_split)  
clf = clf.fit(x\_train,y\_train)  
  
#predict  
pred = clf.predict(x\_test)  
  
#calculate scores  
prec = precision\_score(y\_test,pred,average='macro')  
rec = recall\_score(y\_test,pred,average='macro')  
f1 = f1\_score(y\_test,pred,average='macro')  
print("criterion : {}".format(criterion))  
print("impurity split threshold : {}".format(min\_impurity\_split))  
print("Precision\_score : {:.2f}".format(prec))  
print("Recall\_score:",rec)  
print("F1\_score : ",f1)  
print("==================")  
print()  
  
print("Feature importances\n")  
for i in range(7):  
 print(feature\_names[i + 1])  
 print(clf.feature\_importances\_[i])  
  
# print graph  
dot\_data = tree.export\_graphviz(clf, out\_file=None,  
 feature\_names=feature\_names[1:],  
 class\_names=class\_names,  
 filled=True, rounded=True,  
 special\_characters=True)  
  
graph = graphviz.Source(dot\_data)  
graph.render("entropy\_decision\_tree")

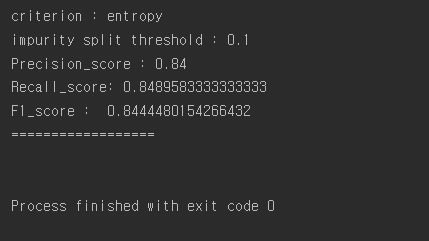
>> K Nearest Neighbor

*'''  
인공지능 2019  
과제 3  
pandas, numpy, sklearn 등의 library는 PPT 자료 참고하여 설치  
'''*import pandas as pd  
import numpy as np  
#from sklearn import tree  
from sklearn.model\_selection import train\_test\_split  
from sklearn.metrics import precision\_score, recall\_score, f1\_score  
from sklearn.neighbors import KNeighborsClassifier  
  
# prepare data  
data = pd.read\_csv('C:/Users/tmddms2292/Desktop/인공지능 (2019-1)/assignment3/as3.csv') # read csv file in pandas dataframe  
data1 = np.array(data) # change dataframe to numpy array  
class\_names = ['Horeca', 'Retail'] # python list of class name  
feature\_names = data.columns # get list of feature names from pandas dataframe  
  
X = list(map(lambda x: x[1:], data1))  
Y = list(map(lambda x: x[0], data1))  
  
# random split  
x\_train, x\_test, y\_train, y\_test = train\_test\_split(X, Y, test\_size=0.2)  
  
# 하단에 작성  
  
weights = 'distance'  
#weights = 'uniform'  
k=30 #k값을 바꿔줌 1,5,10,20,30  
clf = KNeighborsClassifier(n\_neighbors=k,weights = weights)  
clf = clf.fit(x\_train,y\_train)  
  
#predict  
pred = clf.predict(x\_test)  
  
#calculate scores  
prec = precision\_score(y\_test,pred,average='macro')  
rec = recall\_score(y\_test,pred,average='macro')  
f1 = f1\_score(y\_test,pred,average='macro')  
#print("criterion : {}".format(criterion))  
#print("impurity split threshold : {}".format(n\_neighbors))  
print("weights : {}".format(k))  
print("Precision\_score : {:.2f}".format(prec))  
print("Recall\_score:",rec)  
print("F1\_score : ",f1)  
print("==================")  
print()

**<Run 결과>**

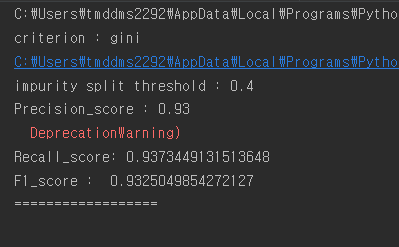
>>Decision Tree에서 Parameter1이 ‘entropy’일 때 결과화면

(min\_impurity\_split이 0.1~0.5 모두 같은 방식으로 나와 생략했습니다.)



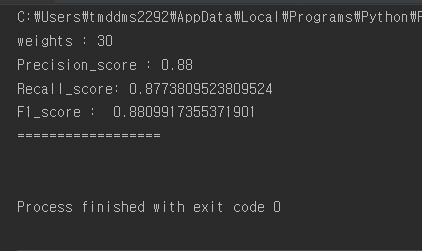
>>Decision Tree에서 Parameter1이 ‘gini’일 때 결과화면

(min\_impurity\_split이 0.1~0.5 모두 같은 방식으로 나와 생략했습니다.)



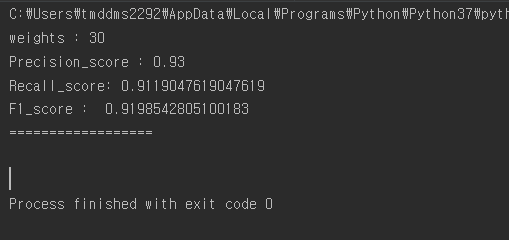
>>K Nearest Neighbor에서 Parameter1이 ‘uniform’일 때 결과화면

(k가 1,5,10,20,30 일 때 모두 같은 방식으로 나와서 생략했습니다)



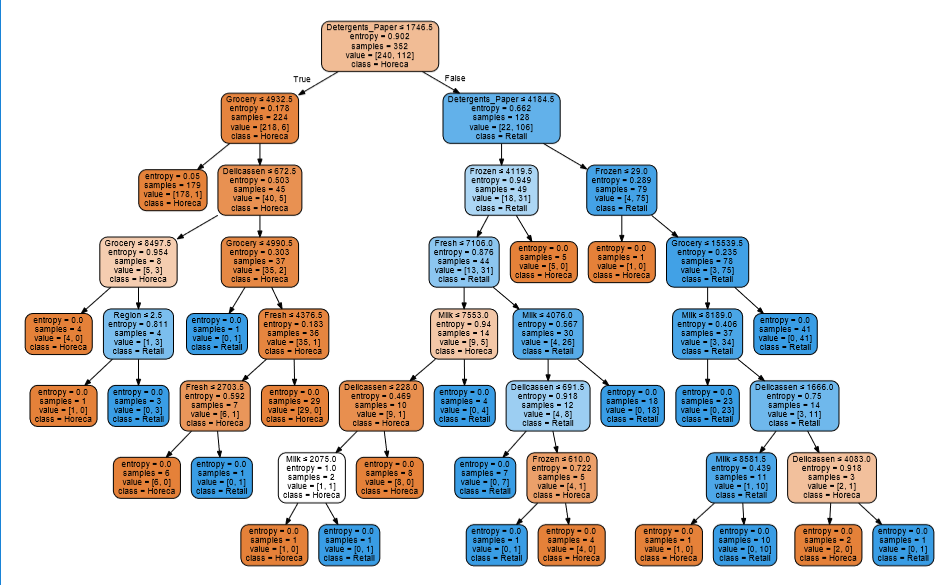
>>K Nearest Neighbor에서 Parameter1이 ‘distance’일 때 결과화면

(k가 1,5,10,20,30 일 때 모두 같은 방식으로 나와서 생략했습니다)

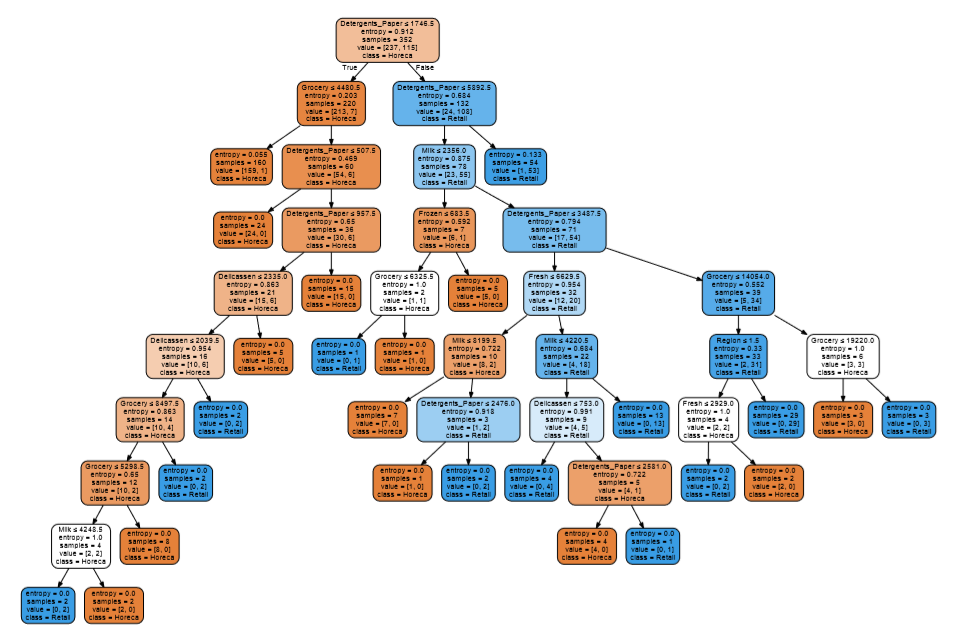


**<decixion\_tree\_entropy 트리모형>**

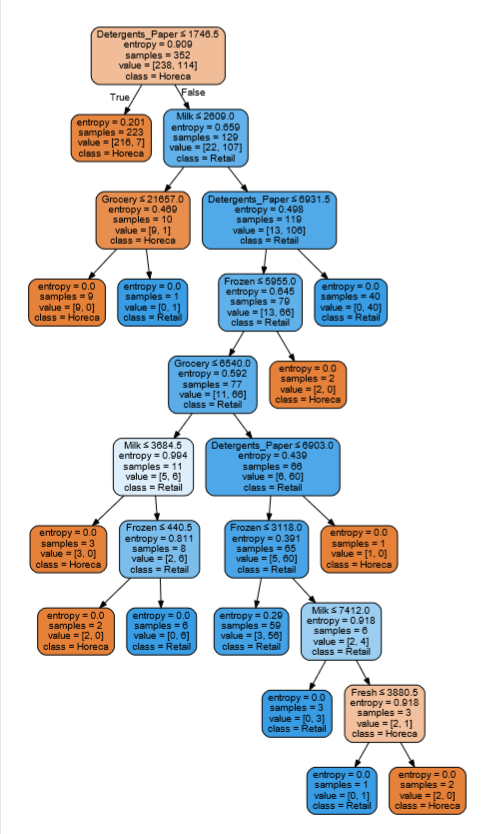
Min\_impurity\_split이 0.1일 때



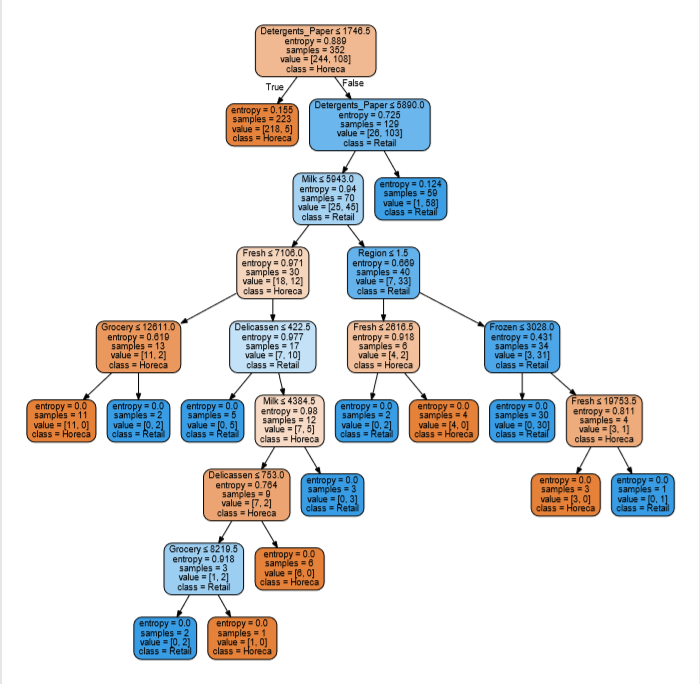
Min\_impurity\_split이 0.2일 때



Min\_impurity\_split이 0.3일 때



Min\_impurity\_split이 0.4일 때



Min\_impurity\_split이 0.5일 때

