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.

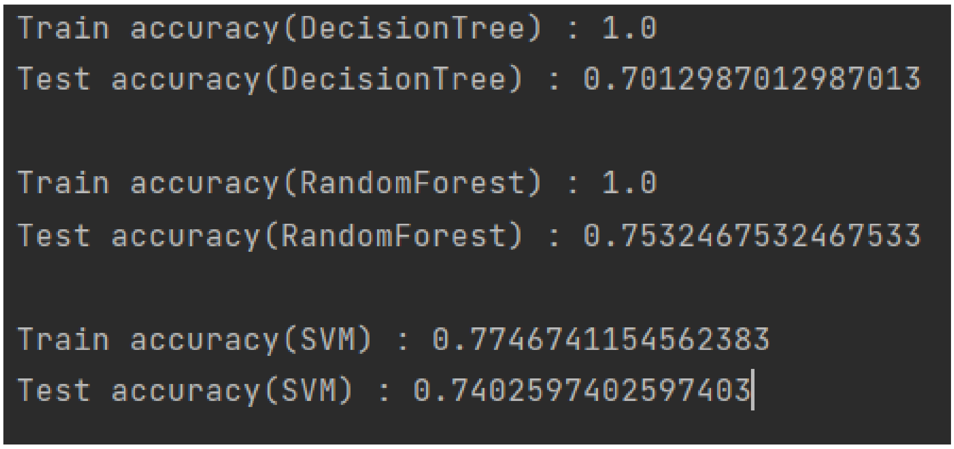
- 각 알고리즘의 hyper parameter 의 값은 default value를 이용한다.

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:



```
Train accuracy(DecisionTree) : 1.0
Test accuracy(DecisionTree) : 0.7012987012987013

Train accuracy(RandomForest) : 1.0
Test accuracy(RandomForest) : 0.7532467532467533

Train accuracy(SVM) : 0.7746741154562383
Test accuracy(SVM) : 0.7402597402597403
```

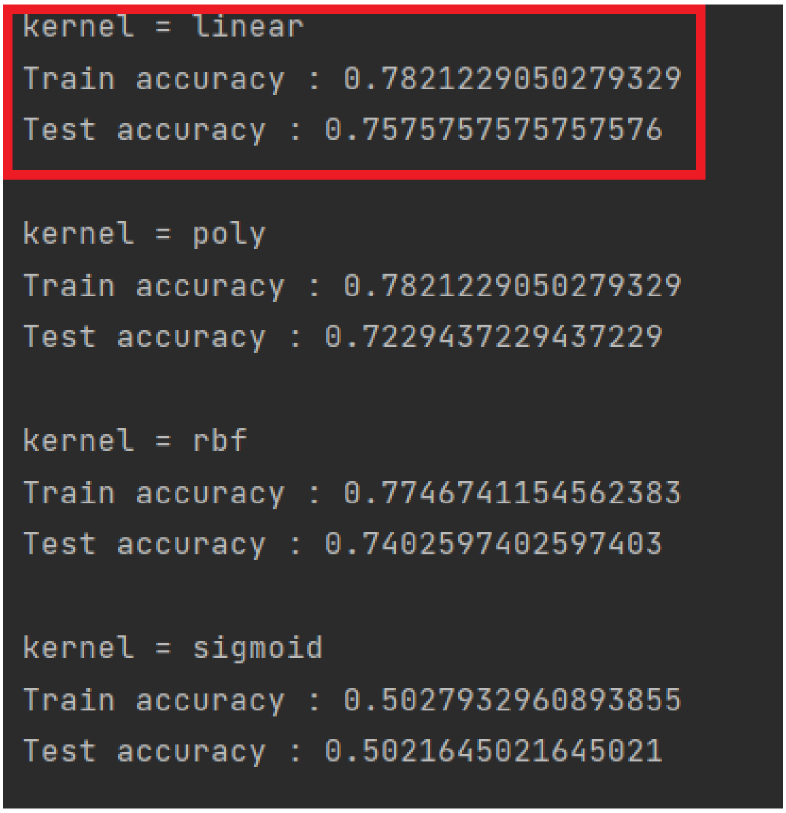
Q2. (3점) 다음의 조건에 따라 support vector machine 알고리즘를 이용하여 PimaIndiansDiabetes dataset에 대한 분류 모델을 생성하고 accuracy를 비교하시오.

- 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/PimaIn
diansDiabetes.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:



```
kernel = linear
Train accuracy : 0.7821229050279329
Test accuracy : 0.7575757575757576

kernel = poly
Train accuracy : 0.7821229050279329
Test accuracy : 0.7229437229437229

kernel = rbf
Train accuracy : 0.7746741154562383
Test accuracy : 0.7402597402597403

kernel = sigmoid
Train accuracy : 0.5027932960893855
Test accuracy : 0.5021645021645021
```

Q3. (3점) 다음의 조건에 따라 Random Forest 알고리즘를 이용하여 PimaIndiansDiabetes dataset에 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

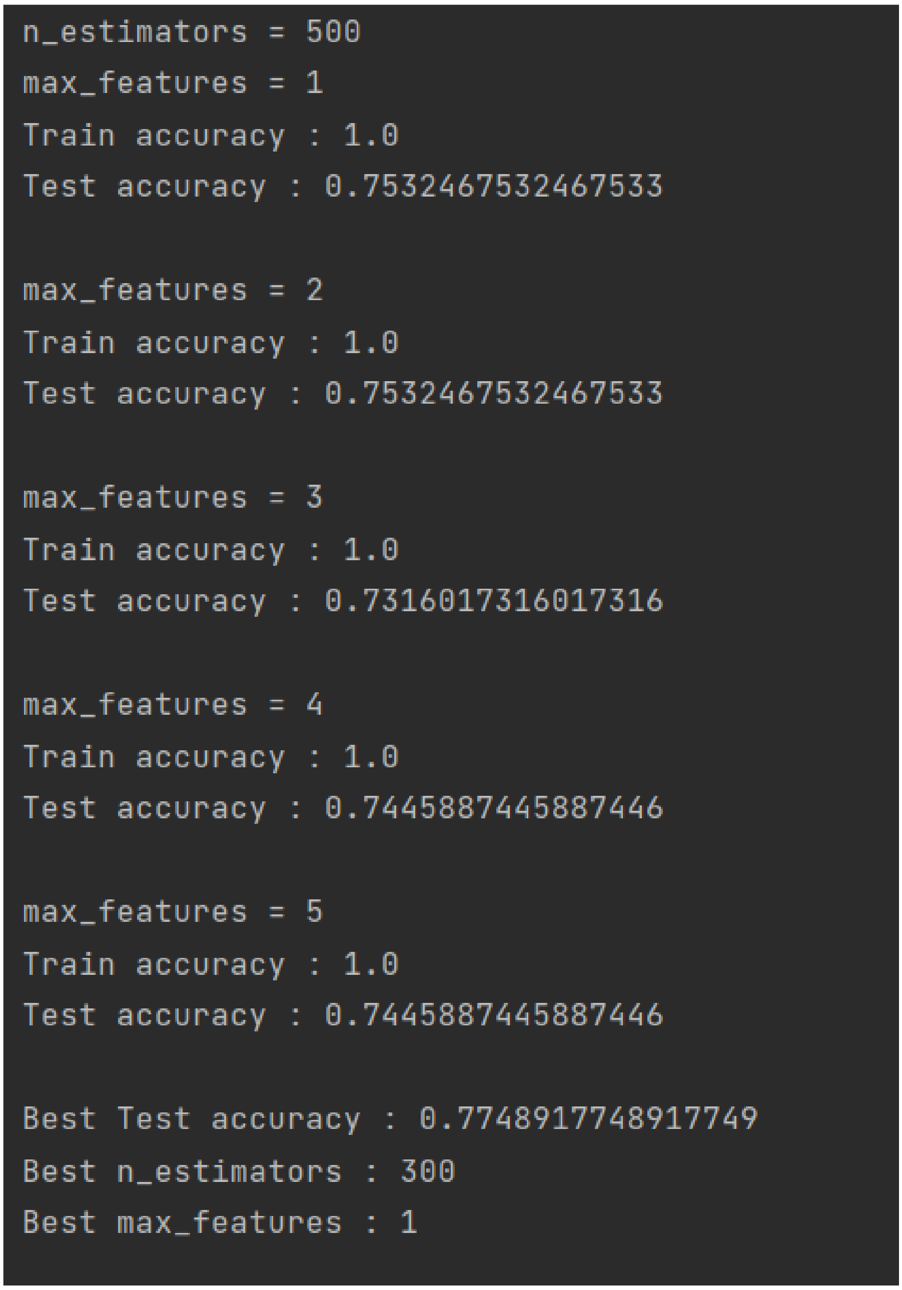
어떤 조합이 가장 높은 accuracy를 도출하는지 확인하시오.

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 캡처:



```
n_estimators = 500
max_features = 1
Train accuracy : 1.0
Test accuracy : 0.7532467532467533

max_features = 2
Train accuracy : 1.0
Test accuracy : 0.7532467532467533

max_features = 3
Train accuracy : 1.0
Test accuracy : 0.7316017316017316

max_features = 4
Train accuracy : 1.0
Test accuracy : 0.7445887445887446

max_features = 5
Train accuracy : 1.0
Test accuracy : 0.7445887445887446

Best Test accuracy : 0.7748917748917749
Best n_estimators : 300
Best max_features : 1
```