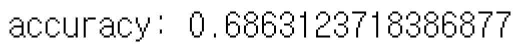
6주. Decision Tree, RF, SVM			
학번	32183164	이름	이석현

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.
- 10-fold cross validation을 실시하여 mean accuracy를 비교한다
- 각 알고리즘의 hyper parameter 의 값은 default value를 이용한다.

Source code :

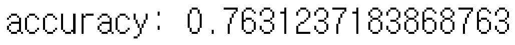
```
// source code 의 폰트는 Courier10 BT Bold으로 하시오
# DecisionTree

from sklearn.tree import DecisionTreeClassifier, export_graphviz
from sklearn.model_selection import train_test_split
from sklearn.metrics import confusion_matrix
from sklearn.metrics import accuracy_score
from sklearn.model_selection import KFold
import pandas as pd
import numpy as np
import pydot

df = pd.read_csv('D:/data/dataset_0914/PimaIndiansDiabetes.csv')
df_X = df.loc[:, df.columns != 'diabetes']
df_y = df[['diabetes']]
df_y[df_y['diabetes'] == 'pos'] = 1
df_y[df_y['diabetes'] == 'neg'] = 0

kf = KFold(n_splits=10, random_state=123, shuffle=True)
model =  DecisionTreeClassifier(random_state=1234)
acc = np.zeros(10)
i = 0

for train_index, test_index in kf.split(df_X):
   train_X, test_X = df_X.iloc[train_index].astype('int'),
df_X.iloc[test_index].astype('int')
   train_y, test_y = df_y.iloc[train_index].astype('int'),
df_y.iloc[test_index].astype('int')

   model.fit(train_X, train_y)

   pred_y = model.predict(test_X)

   acc[i] = accuracy_score(test_y, pred_y)
   i += 1

print('accuracy:', np.mean(acc))
```

실행화면 캡쳐: 

```
accuracy: 0.6863123718386877
```

```
// source code 의 폰트는 Courier10 BT Bold으로 하시오
# RandomForest

from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import train_test_split
from sklearn.metrics import confusion_matrix
from sklearn.metrics import accuracy_score
from sklearn.model_selection import KFold
import pandas as pd
import numpy as np
import pydot

df = pd.read_csv('D:/data/dataset_0914/PimaIndiansDiabetes.csv')
df_X = df.loc[:, df.columns != 'diabetes']
df_y = df[['diabetes']]
df_y[df_y['diabetes'] == 'pos'] = 1
df_y[df_y['diabetes'] == 'neg'] = 0

kf = KFold(n_splits=10, random_state=123, shuffle=True)
model = RandomForestClassifier(n_estimators=10, random_state=1234)
acc = np.zeros(10)
i = 0

for train_index, test_index in kf.split(df_X):
   train_X,    test_X    =    df_X.iloc[train_index].astype('int'),
df_X.iloc[test_index].astype('int')
   train_y,    test_y    =    df_y.iloc[train_index].astype('int'),
df_y.iloc[test_index].astype('int')

   model.fit(train_X, train_y)

   pred_y = model.predict(test_X)

   acc[i] = accuracy_score(test_y, pred_y)
   i += 1

print('accuracy:', np.mean(acc))
```

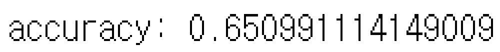
실행화면 캡쳐:



```
accuracy: 0.7631237183868763
```

```
// source code 의 폰트는 Courier10 BT Bold으로 하시오
# Support Vector Machine

from sklearn import svm
from sklearn.model_selection import train_test_split
from sklearn.metrics import confusion_matrix
from sklearn.metrics import accuracy_score
from sklearn.model_selection import KFold
import pandas as pd
import numpy as np
import pydot

df = pd.read_csv('D:/data/dataset_0914/PimaIndiansDiabetes.csv')
df_X = df.loc[:, df.columns != 'diabetes']
df_y = df[['diabetes']]
df_y[df_y['diabetes'] == 'pos'] = 1
df_y[df_y['diabetes'] == 'neg'] = 0

kf = KFold(n_splits=10, random_state=123, shuffle=True)

model = svm.SVC()

acc = np.zeros(10)
i = 0

for train_index, test_index in kf.split(df_X):
   train_X, test_X = df_X.iloc[train_index].astype('int'),
df_X.iloc[test_index].astype('int')
   train_y, test_y = df_y.iloc[train_index].astype('int'),
df_y.iloc[test_index].astype('int')

   model.fit(train_X, train_y)

   pred_y = model.predict(test_X)

   acc[i] = accuracy_score(test_y, pred_y)
   i += 1

print('accuracy:', np.mean(acc))
```

실행화면 캡처:

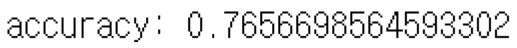


```
accuracy: 0.650991114149009
```

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, sigmoid, precomputed를 각각 테스트하여 어떤 kernel 이 가장 높은 accuracy를 도출하는지 확인하시오.
- 10-fold cross validation을 실시하여 mean accuracy를 비교한다

Source code :

```
// source code 의 폰트는 Courier10 BT Bold으로 하시오
# kernel: linear

from sklearn import svm
from sklearn.model_selection import train_test_split
from sklearn.metrics import confusion_matrix
from sklearn.metrics import accuracy_score
from sklearn.model_selection import KFold
import pandas as pd
import numpy as np
import pydot

df = pd.read_csv('D:/data/dataset_0914/PimaIndiansDiabetes.csv')
df_X = df.loc[:, df.columns != 'diabetes']
df_y = df[['diabetes']]
df_y[df_y['diabetes'] == 'pos'] = 1
df_y[df_y['diabetes'] == 'neg'] = 0

kf = KFold(n_splits=10, random_state=123, shuffle=True)
model = svm.SVC(kernel='linear')

acc = np.zeros(10)
i = 0

for train_index, test_index in kf.split(df_X):
    train_X, test_X = df_X.iloc[train_index].astype('int'),
df_X.iloc[test_index].astype('int')
    train_y, test_y = df_y.iloc[train_index].astype('int'),
df_y.iloc[test_index].astype('int')

    model.fit(train_X, train_y)
    pred_y = model.predict(test_X)
    acc[i] = accuracy_score(test_y, pred_y)
    i += 1

print('accuracy:', np.mean(acc))
```

실행화면 캡쳐: 

```
accuracy: 0.7656698564593302
```

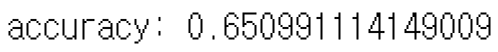
Source code :

```
// source code 의 폰트는 Courier10 BT Bold으로 하시오
#poly

from sklearn import svm
from sklearn.model_selection import train_test_split
from sklearn.metrics import confusion_matrix
from sklearn.metrics import accuracy_score
from sklearn.model_selection import KFold
import pandas as pd
import numpy as np
import pydot

df = pd.read_csv('D:/data/dataset_0914/PimaIndiansDiabetes.csv')
df_X = df.loc[:, df.columns != 'diabetes']
df_y = df[['diabetes']]
df_y[df_y['diabetes'] == 'pos'] = 1
df_y[df_y['diabetes'] == 'neg'] = 0

kf = KFold(n_splits=10, random_state=123, shuffle=True)
model = svm.SVC(kernel='poly', degree=2)

acc = np.zeros(10)
i = 0

for train_index, test_index in kf.split(df_X):
    train_X,    test_X    =    df_X.iloc[train_index].astype('int'),
df_X.iloc[test_index].astype('int')
    train_y,    test_y    =    df_y.iloc[train_index].astype('int'),
df_y.iloc[test_index].astype('int')

    model.fit(train_X, train_y)
    pred_y = model.predict(test_X)
    acc[i] = accuracy_score(test_y, pred_y)
    i += 1
print('accuracy:', np.mean(acc))
```

실행화면 캡쳐: 프로그램이 계속 실행되고 끝나지 않는데, 이유를 모르겠습니다.

Source code :

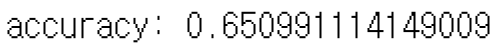
```
// source code 의 폰트는 Courier10 BT Bold으로 하시오
#rbf

from sklearn import svm
from sklearn.model_selection import train_test_split
from sklearn.metrics import confusion_matrix
from sklearn.metrics import accuracy_score
from sklearn.model_selection import KFold
import pandas as pd
import numpy as np
import pydot

df = pd.read_csv('D:/data/dataset_0914/PimaIndiansDiabetes.csv')
df_X = df.loc[:, df.columns != 'diabetes']
df_y = df[['diabetes']]
df_y[df_y['diabetes'] == 'pos'] = 1
df_y[df_y['diabetes'] == 'neg'] = 0

kf = KFold(n_splits=10, random_state=123, shuffle=True)

model = svm.SVC(kernel='rbf')

acc = np.zeros(10)
i = 0

for train_index, test_index in kf.split(df_X):
   train_X,    test_X    =    df_X.iloc[train_index].astype('int'),
df_X.iloc[test_index].astype('int')
   train_y,    test_y    =    df_y.iloc[train_index].astype('int'),
df_y.iloc[test_index].astype('int')
   model.fit(train_X, train_y)
   pred_y = model.predict(test_X)
   acc[i] = accuracy_score(test_y, pred_y)
   i += 1

print('accuracy:', np.mean(acc))
```

실행화면 캡처: 

```
accuracy: 0.650991114149009
```

Source code :

```
// source code 의 폰트는 Courier10 BT Bold으로 하시오
#sigmoid

from sklearn import svm
from sklearn.model_selection import train_test_split
from sklearn.metrics import confusion_matrix
from sklearn.metrics import accuracy_score
from sklearn.model_selection import KFold
import pandas as pd
import numpy as np
import pydot

df = pd.read_csv('D:/data/dataset_0914/PimaIndiansDiabetes.csv')
df_X = df.loc[:, df.columns != 'diabetes']
df_y = df[['diabetes']]
df_y[df_y['diabetes'] == 'pos'] = 1
df_y[df_y['diabetes'] == 'neg'] = 0

kf = KFold(n_splits=10, random_state=123, shuffle=True)

model = svm.SVC(kernel='sigmoid')

acc = np.zeros(10)
i = 0

for train_index, test_index in kf.split(df_X):
   train_X,    test_X    =    df_X.iloc[train_index].astype('int'),
df_X.iloc[test_index].astype('int')
   train_y,    test_y    =    df_y.iloc[train_index].astype('int'),
df_y.iloc[test_index].astype('int')

   model.fit(train_X, train_y)
   pred_y = model.predict(test_X)

   acc[i] = accuracy_score(test_y, pred_y)
   i += 1

print('accuracy:', np.mean(acc))
```

실행화면 캡쳐: 

```
accuracy: 0.650991114149009
```

Source code :

```
// source code 의 폰트는 Courier10 BT Bold으로 하시오
#precomputed

from sklearn import svm
from sklearn.model_selection import train_test_split
from sklearn.metrics import confusion_matrix
from sklearn.metrics import accuracy_score
from sklearn.model_selection import KFold
import pandas as pd
import numpy as np
import pydot

df = pd.read_csv('D:/data/dataset_0914/PimaIndiansDiabetes.csv')
df_X = df.loc[:, df.columns != 'diabetes']
df_y = df[['diabetes']]
df_y[df_y['diabetes'] == 'pos'] = 1
df_y[df_y['diabetes'] == 'neg'] = 0

kf = KFold(n_splits=10, random_state=123, shuffle=True)
model = svm.SVC(kernel='precomputed')

acc = np.zeros(10)
i = 0

for train_index, test_index in kf.split(df_X):
    train_X,     test_X     =     df_X.iloc[train_index].astype('int'),
df_X.iloc[test_index].astype('int')
    train_y,     test_y     =     df_y.iloc[train_index].astype('int'),
df_y.iloc[test_index].astype('int')

    kernel_train = np.dot(train_X, train_X.T)
    model.fit(kernel_train, train_y)
    kernel_test = np.dot(test_X, test_X.T)
    pred_y = model.predict(kernel_test)

    acc[i] = accuracy_score(test_y, pred_y)
    i += 1
print('accuracy:', np.mean(acc))
```

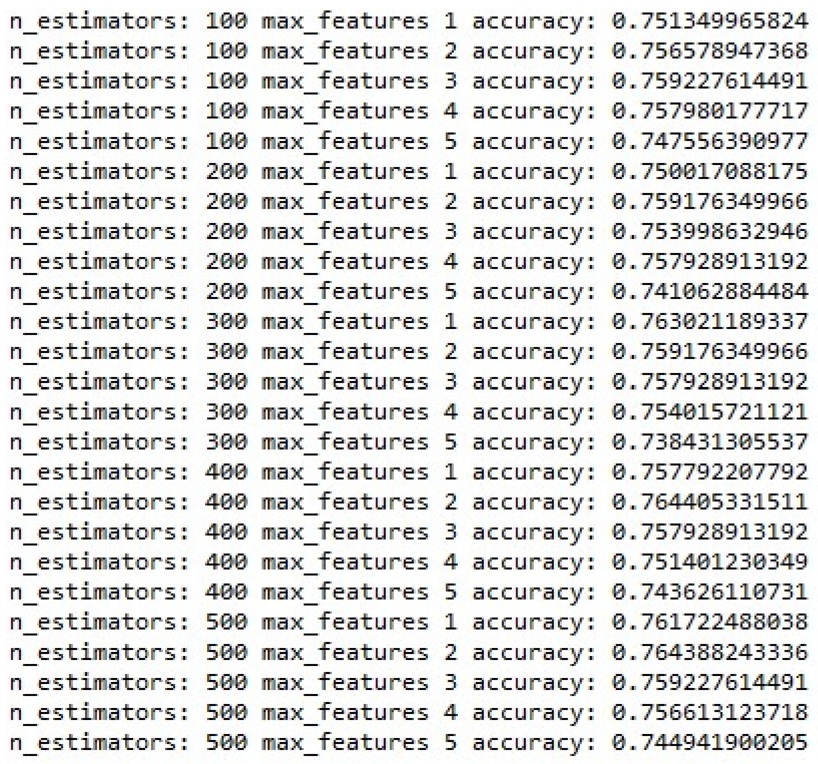
실행화면 캡처: 프로그램이 계속 실행되고 끝나지 않는데, 이유를 모르겠습니다.

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.
- 10-fold cross validation을 실시하여 mean accuracy를 비교한다

Source code :

```
// source code 의 폰트는 Courier10 BT Bold으로 하시오
from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import train_test_split
from sklearn.metrics import confusion_matrix
from sklearn.metrics import accuracy_score
from sklearn.model_selection import KFold
import pandas as pd
import numpy as np

df = pd.read_csv('D:/data/dataset_0914/PimaIndiansDiabetes.csv')
df_X = df.loc[:, df.columns != 'diabetes']
df_y = df[['diabetes']]
df_y[df_y['diabetes'] == 'pos'] = 1
df_y[df_y['diabetes'] == 'neg'] = 0
kf = KFold(n_splits=10, random_state=123, shuffle=True)
param_n_estimators = [100, 200, 300, 400, 500]
param_max_features = [1, 2, 3, 4, 5]
accuracy = []
for j in range(len(param_n_estimators)):
    for k in range(len(param_max_features)):
        model                                                  =
RandomForestClassifier(n_estimators=param_n_estimators[j],
max_features = param_max_features[k], random_state=1234)
        acc = np.zeros(10)
        i = 0
        for train_index, test_index in kf.split(df_X):
            train_X,  test_X  =  df_X.iloc[train_index].astype('int'),
df_X.iloc[test_index].astype('int')
            train_y,  test_y  =  df_y.iloc[train_index].astype('int'),
df_y.iloc[test_index].astype('int')
            model.fit(train_X, train_y)
            pred_y = model.predict(test_X)
            acc[i] = accuracy_score(test_y, pred_y)
            i += 1
        accuracy.append(np.mean(acc))
l = 0
for j in range(len(param_n_estimators)):
    for k in range(len(param_max_features)):
        print('n_estimators:', param_n_estimators[j], 'max_features',
param_max_features[k], 'accuracy:', accuracy[l])
        l += 1
```

실행화면 캡쳐:



```
n_estimators: 100 max_features 1 accuracy: 0.751349965824
n_estimators: 100 max_features 2 accuracy: 0.756578947368
n_estimators: 100 max_features 3 accuracy: 0.759227614491
n_estimators: 100 max_features 4 accuracy: 0.757980177717
n_estimators: 100 max_features 5 accuracy: 0.747556390977
n_estimators: 200 max_features 1 accuracy: 0.750017088175
n_estimators: 200 max_features 2 accuracy: 0.759176349966
n_estimators: 200 max_features 3 accuracy: 0.753998632946
n_estimators: 200 max_features 4 accuracy: 0.757928913192
n_estimators: 200 max_features 5 accuracy: 0.741062884484
n_estimators: 300 max_features 1 accuracy: 0.763021189337
n_estimators: 300 max_features 2 accuracy: 0.759176349966
n_estimators: 300 max_features 3 accuracy: 0.757928913192
n_estimators: 300 max_features 4 accuracy: 0.754015721121
n_estimators: 300 max_features 5 accuracy: 0.738431305537
n_estimators: 400 max_features 1 accuracy: 0.757792207792
n_estimators: 400 max_features 2 accuracy: 0.764405331511
n_estimators: 400 max_features 3 accuracy: 0.757928913192
n_estimators: 400 max_features 4 accuracy: 0.751401230349
n_estimators: 400 max_features 5 accuracy: 0.743626110731
n_estimators: 500 max_features 1 accuracy: 0.761722488038
n_estimators: 500 max_features 2 accuracy: 0.764388243336
n_estimators: 500 max_features 3 accuracy: 0.759227614491
n_estimators: 500 max_features 4 accuracy: 0.756613123718
n_estimators: 500 max_features 5 accuracy: 0.744941900205
```