K_Fold_Method

April 4, 2023

The Problem Statement In Holdout method , we have seen a problem arising , with repeated train and test data splitiing , overlapping of test and training and testing data might and will occur. The K fold cross validation method comes to rescue here. We can provide any number here to replace K here. Suppose we are using a 3 fold cross validation where k is being replaced by 3. The workflow will be as below -

- The entire dataset will be divided into 3 equal parts
- Suppose the parts are names as p1,p2,p3
- For training/testing the data , there will be 3 iteration here
- 1st iteration - p1 is the testing data / p2 and p3 is the training data - error estimate -
- 2nd iteration - p2 is the testing data / p1 and p3 is the training data - error estimate -
- 3rd iteration - p3 is the testing data / p1 and p2 is the training data - error estimate -
- Final error estimate - (e1+e2+e3)/k - k is 3 here
- The advantage here is that every set of data is being used for both training and testing.
- Primarily used for classification models but also used for regression problems for finding

Here is a basic example of 3 fold cross validation

[1]:
```
# importing the libraries
from sklearn.model_selection import KFold
''' We are defining here that number of splits of k will be 3'''
kf = KFold(n_splits=3)
kf
```

[1]:
```
KFold(n_splits=3, random_state=None, shuffle=False)
```

[2]:
```
''' The kf.split section here is taking a random numpy array data into␣
 ↪consideration and splitting it into 3 subsets as per k value being 3'''
''' The for loop is helping us to print train and test split data in each␣
 ↪iteraion as output'''
for train_index, test_index in kf.split([1,2,3,4,5,6,7,8,9]):
    print(train_index, test_index)
```

```
[3 4 5 6 7 8] [0 1 2]
[0 1 2 6 7 8] [3 4 5]
[0 1 2 3 4 5] [6 7 8]
```

Digit Classifier Data Now we will work on a data available in sklearn library , which is basically a image classifier , different images of blurred data is present in it

[7]:
```
# importing libraries
from sklearn.linear_model import LogisticRegression
from sklearn.svm import SVC
from sklearn.ensemble import RandomForestClassifier
import numpy as np
from sklearn.datasets import load_digits
from sklearn.model_selection import train_test_split
```

[4]:
```
digits = load_digits()
```

[5]:
```
print(digits.data)
```

```
[[ 0.  0.  5. … 0.  0.  0.]
 [ 0.  0.  0. … 10.  0.  0.]
 [ 0.  0.  0. … 16.  9.  0.]
 …
 [ 0.  0.  1. …  6.  0.  0.]
 [ 0.  0.  2. … 12.  0.  0.]
 [ 0.  0. 10. … 12.  1.  0.]]
```

[6]:
```
print(digits.target)
```

```
[0 1 2 … 8 9 8]
```

[8]:
```
# train test splitting without cross validation
xtrain, xtest, ytrain, ytest = train_test_split(digits.data,digits.
 ↪target,test_size=0.3)
```

[15]:
```
# now we will run a loop here to fit our training data into all imported models␣
 ↪, Logistic regression , random forest and support vector
lr = LogisticRegression(max_iter = 20)
rf = RandomForestClassifier(n_estimators= 40)
svm = SVC()
models = [lr,rf,svm]
```

[16]:
```
for i in models:
    i.fit(xtrain,ytrain)
    print(f'the score for model {i} is {i.score(xtest,ytest)}')
```

```
the score for model LogisticRegression(max_iter=20) is 0.9555555555555556
the score for model RandomForestClassifier(n_estimators=40) is
0.9722222222222222
the score for model SVC() is 0.9907407407407407

d:\Anaconda\envs\github1\lib\site-
packages\sklearn\linear_model\_logistic.py:458: ConvergenceWarning: lbfgs failed
to converge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.
```

```
Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-
regression
  n_iter_i = _check_optimize_result(
```

[18]:
```
# Now let's have a look how we can apply kfold cross validation here with a for␣
 ↪loop
from sklearn.model_selection import StratifiedKFold
''' Function to get the score for each model'''
def get_score(model, X_train, X_test, y_train, y_test):
    model.fit(X_train, y_train)
    return model.score(X_test, y_test)
''' Defining the splits'''
folds = StratifiedKFold(n_splits=3)
''' Blank lists where scores of each model will be stored'''
scores_logistic = []
scores_svm = []
scores_rf = []
''' for loop to iterate through our digit data and finding out the scores based␣
 ↪on 3 splits of all training and testing sets'''
for train_index, test_index in folds.split(digits.data,digits.target):
    X_train, X_test, y_train, y_test = digits.data[train_index], digits.
 ↪data[test_index], digits.target[train_index], digits.target[test_index]
    scores_logistic.append(get_score(LogisticRegression(), X_train, X_test,␣
 ↪y_train, y_test))
    scores_svm.append(get_score(SVC(), X_train, X_test, y_train, y_test))
    scores_rf.append(get_score(RandomForestClassifier(), X_train, X_test,␣
 ↪y_train, y_test))
```

```
d:\Anaconda\envs\github1\lib\site-
packages\sklearn\linear_model\_logistic.py:458: ConvergenceWarning: lbfgs failed
to converge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-
regression
  n_iter_i = _check_optimize_result(
d:\Anaconda\envs\github1\lib\site-
packages\sklearn\linear_model\_logistic.py:458: ConvergenceWarning: lbfgs failed
to converge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
```

```
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-
regression
  n_iter_i = _check_optimize_result(
d:\Anaconda\envs\github1\lib\site-
packages\sklearn\linear_model\_logistic.py:458: ConvergenceWarning: lbfgs failed
to converge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-
regression
  n_iter_i = _check_optimize_result(
```

[23]:
```
print(f'the logistic regression score set is {scores_logistic}')
```

```
the logistic regression score set is [0.9215358931552587, 0.9415692821368948,
0.9165275459098498]
```

[24]:
```
print(f'the support vector score set is {scores_svm}')
```

```
the support vector score set is [0.9649415692821369, 0.9799666110183639,
0.9649415692821369]
```

[25]:
```
print(f'the random forest score set is {scores_rf}')
```

```
the random forest score set is [0.9382303839732888, 0.9565943238731218,
0.9198664440734557]
```

[27]:
```
# using the sklearn inbuilt module finding the cross val score as above
from sklearn.model_selection import cross_val_score
logreg_cross_val = cross_val_score(LogisticRegression(), digits.data, digits.
 ↪target,cv=3)
svm_cross_val = cross_val_score(SVC(), digits.data, digits.target,cv=3)
rf_cross_val = cross_val_score(RandomForestClassifier(),digits.data, digits.
 ↪target,cv=3)
```

```
d:\Anaconda\envs\github1\lib\site-
packages\sklearn\linear_model\_logistic.py:458: ConvergenceWarning: lbfgs failed
to converge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-
regression
```

```
  n_iter_i = _check_optimize_result(
d:\Anaconda\envs\github1\lib\site-
packages\sklearn\linear_model\_logistic.py:458: ConvergenceWarning: lbfgs failed
to converge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-
regression
  n_iter_i = _check_optimize_result(
d:\Anaconda\envs\github1\lib\site-
packages\sklearn\linear_model\_logistic.py:458: ConvergenceWarning: lbfgs failed
to converge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-
regression
  n_iter_i = _check_optimize_result(
```

[29]:
```
print(f'logistic regression cross validation score set is {logreg_cross_val}')
print(f'support vector cross validation score set is {svm_cross_val}')
print(f'random forest cross validation score set is {rf_cross_val}')
```

```
logistic regression cross validation score set is [0.92153589 0.94156928
0.91652755]
support vector cross validation score set is [0.96494157 0.97996661 0.96494157]
random forest cross validation score set is [0.93989983 0.95993322 0.93656093]
```

[ ]:
```

```