

```
In [4]: import pandas as pd
import numpy as np
from sklearn.svm import SVC
from sklearn.datasets import load_digits
from sklearn.linear_model import LogisticRegression
from sklearn.ensemble import RandomForestClassifier

In [6]: from sklearn.model_selection import train_test_split

In [11]: digit=load_digits()

In [12]: dir(digit)

Out[12]: ['DESCR', 'data', 'feature_names', 'frame', 'images', 'target', 'target_names']

In [13]: digit.data

Out[13]: array([[ 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.]])

In [15]: x_train,x_test,y_train,y_test=train_test_split(digit.data,digit.target,test_size=0.3)
# we dont need this if we use the cross_val_score

In [28]: la=LogisticRegression()
la.fit(x_train,y_train)
la.score(x_test,y_test)

C:\Users\Asus\AppData\Local\Programs\Python\Python311\Lib\site-packages\sklearn\linear_model\_logistic.py:460: 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(

Out[28]: 0.9574074074074074
```

cross validation is the technique to answer which model is best.

and kfold and stratikfold are the technique to train the model by divide the data set into folds for training.

```
In [29]: ra=SVC()
ra.fit(x_train,y_train)
ra.score(x_test,y_test) # the is the score of logistic regression model

Out[29]: 0.9777777777777777

In [32]: ma=RandomForestClassifier()
ma.fit(x_train,y_train)
ma.score(x_test,y_test)# this is the accuracy of SVC

Out[32]: 0.9611111111111111

In [34]: from sklearn.model_selection import KFold

In [35]: kf=KFold() # kfold divide the dataset into folds and train it

In [36]: kf

Out[36]: KFold(n_splits=5, random_state=None, shuffle=False)

In [37]: for train_index, test_index in kf.split([1,2,3,4,5,6,7,8,9]):
print(train_index,test_index) # here we learn to split the data into folds using kfold

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

In [49]: from sklearn.model_selection import cross_val_score
# with this we dont need to use kfold for splitting the dataset and training

In [51]: cross_val_score(LogisticRegression(),digit.data,digit.target)

'''this model train our data within by creating three different fold and give scores'''

C:\Users\Asus\AppData\Local\Programs\Python\Python311\Lib\site-packages\sklearn\linear_model\_logistic.py:460: 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(
C:\Users\Asus\AppData\Local\Programs\Python\Python311\Lib\site-packages\sklearn\linear_model\_logistic.py:460: 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(
C:\Users\Asus\AppData\Local\Programs\Python\Python311\Lib\site-packages\sklearn\linear_model\_logistic.py:460: 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(
C:\Users\Asus\AppData\Local\Programs\Python\Python311\Lib\site-packages\sklearn\linear_model\_logistic.py:460: 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(

Out[51]: 'this model train our data within by creating three different fold and give scores'

In [52]: cross_val_score(RandomForestClassifier(),digit.data,digit.target,cv=3)
# with the help of cv we can manually create the number of folds

Out[52]: array([0.94156928, 0.95325543, 0.93489149])

In [53]: cross_val_score(SVC(),digit.data,digit.target,cv=3)

Out[53]: array([0.96494157, 0.97996661, 0.96494157])
```

the `cross_val_score` divide the dataset into three folds by default but then we can change as well.