Five

April 16, 2023

Initialization

[ ]:

```
from numpy import *
from sklearn.datasets import load_digits
from sklearn.naive_bayes import GaussianNB
G = GaussianNB()
X,y = load_digits(return_X_y=True)
correct = zeros(10)
incorrect = zeros(10)
```

5 fold cross validation

[ ]:

```
n = X.shape[0]
for i in range(5):
    s = (i*n)//5
    X_train = X[r_[0:s,s+n//5:n]]
    X_test = X[s:s+n//5]
    y_train = sign(y[r_[0:s,s+n//5:n]])
    y_test = y[s:s+n//5]
    G.fit(X_train,y_train)
    y_pred = G.predict(X_test)
    for j in range(len(y_test)):
        num = y_test[j]
        if y_pred[j] == sign(num):
            correct[num] += 1
        else :
            incorrect[num] += 1
```

Average error

[ ]:

```
print("number accuracy(%)")
for i in range(10):
    print(" ",i,"   ",round(100*(correct[i]/(incorrect[i]+correct[i])),2))
```

```
number accuracy(%)
  0      98.31
  1      100.0
  2      99.44
  3      100.0
```

```
4     94.48
5     100.0
6     98.9
7     100.0
8     99.42
9     98.33
```

Thus we make the most error for digit 4.

For total average error :

[ ]:

```python
correct = sum(correct)
incorrect = sum(incorrect)
print("Total average error :",100*incorrect/(incorrect+correct),"%")
```

```
Total average error : 1.1142061281337048 %
```