Logistic Regression

68_Adnan Shaikh

```
In [3]: from sklearn.linear_model import LogisticRegression
        from sklearn.svm import SVC
        from sklearn.ensemble import RandomForestClassifier
        from sklearn.model_selection import train_test_split
        import numpy as np
        from sklearn.datasets import load_digits
        import matplotlib.pyplot as plt
        digits = load_digits()
```

```
In [4]: x_train,x_test,y_train,y_test=train_test_split(digits.data,digits.target,test_size=0.3)
```

```
In [5]: lr=LogisticRegression(solver='liblinear',multi_class='ovr')
        lr.fit(x_train,y_train)
        lr.score(x_test,y_test)
```

```
Out[5]: 0.9611111111111111
```

```
In [6]: svm= SVC(gamma='auto')
        svm.fit(x_train,y_train)
        svm.score(x_test,y_test)
```

```
Out[6]: 0.32222222222222224
```

```
In [7]: rf=RandomForestClassifier(n_estimators=40)
        rf.fit(x_train,y_train)
        rf.score(x_test,y_test)
```

```
Out[7]: 0.9648148148148148
```

K fold cross Validation

```
In [8]: from sklearn.model_selection import KFold
        kf = KFold(n_splits=3)
        kf
```

```
Out[8]: KFold(n_splits=3, random_state=None, shuffle=False)
```

```
In [9]: 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]
```

```
In [10]: 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)
```

In [11]:

```
from sklearn.model_selection import StratifiedKFold
fold = StratifiedKFold(n_splits=3)

score_logistic=[]
score_svm=[]
score_rf=[]

for train_index,test_index in fold.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]
  score_logistic.append(get_score(LogisticRegression(solver='liblinear',multi_class='ovr'),x_train,x_test,y_train,y_test))
  score_svm.append(get_score(SVC(gamma='auto'),x_train,x_test,y_train,y_test))
  score_rf.append(get_score(RandomForestClassifier(n_estimators=40),x_train,x_test,y_train,y_test))
```

In [12]:

```
score_logistic
```

Out[12]:

```
[0.8948247078464107, 0.9532554257095158, 0.9098497495826378]
```

In [13]:

```
score_rf
```

Out[13]:

```
[0.9332220367278798, 0.9599332220367279, 0.9315525876460768]
```

In [14]:

```
score_svm
```

Out[14]:

```
[0.3806343906510851, 0.41068447412353926, 0.5125208681135225]
```

Cross_val_Score Function

In [15]:

```
from sklearn.model_selection import cross_val_score
```

In [16]:

```
cross_val_score(LogisticRegression(solver='liblinear',multi_class='ovr'),digits.data,digits.target,cv=3)
```

Out[16]:

```
array([0.89482471, 0.95325543, 0.90984975])
```

In [17]:

```
score1=cross_val_score(RandomForestClassifier(n_estimators=5),digits.data,digits.target,cv=3)
np.average(score1)
```

Out[17]:

```
0.8503060656649972
```

In [18]:

```
score2=cross_val_score(RandomForestClassifier(n_estimators=20),digits.data,digits.target,cv=3)
np.average(score2)
```

Out[18]:

```
0.9254312743461325
```

In [19]:

```
score3=cross_val_score(RandomForestClassifier(n_estimators=30),digits.data,digits.target,cv=3)
np.average(score3)
```

Out[19]:

```
0.9304396215915415
```

In [20]:

```
score4=cross_val_score(RandomForestClassifier(n_estimators=40),digits.data,digits.target,cv=3)
np.average(score4)
```

Out[20]:

```
0.9304396215915415
```