Lab3_Ex2

June 30, 2020

[1]:
```
from sklearn import datasets #import standard data sets
from sklearn.linear_model import LogisticRegression #import the Logistic␣
↪regression model
from sklearn.model_selection import train_test_split #split data set into a␣
↪train and test set

#importing modules to measure classification performance
from sklearn.metrics import classification_report
from sklearn.metrics import confusion_matrix,accuracy_score
```

[2]:
```
dataset =datasets.load_digits() #load 'wine' data set from standard data sets
x = dataset["data"] #defining features values
y = dataset["target"] #defining target variable values
```

[3]:
```
#splitting data set into a train and test set with 80% and 20%
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.2,␣
↪random_state=1)
```

[4]:
```
log_reg = LogisticRegression() #creating an instance of the model
log_reg.fit(x_train,y_train) #fitting the relationship between data
```

```
c:\users\user\appdata\local\programs\python\python38\lib\site-
packages\sklearn\linear_model\_logistic.py:938: 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(
```

[4]:
```
LogisticRegression(C=1.0, class_weight=None, dual=False, fit_intercept=True,
                   intercept_scaling=1, l1_ratio=None, max_iter=100,
                   multi_class='auto', n_jobs=None, penalty='l2',
                   random_state=None, solver='lbfgs', tol=0.0001, verbose=0,
```

```
                   warm_start=False)
```

[5]:
```
predictions = log_reg.predict(x_test) #predict labels for test data
```

[8]:
```
print("Confusion Matrix = \n",confusion_matrix(y_test, predictions))
print("Accuracy = ",accuracy_score(y_test, predictions))
```

```
Confusion Matrix =
 [[42  0  0  0  1  0  0  0  0  0]
 [ 0 34  0  0  1  0  0  0  0  0]
 [ 0  0 36  0  0  0  0  0  0  0]
 [ 0  0  0 40  0  0  0  0  1  0]
 [ 0  0  0  0 38  0  0  0  0  0]
 [ 0  1  0  1  0 28  0  0  0  0]
 [ 0  0  0  0  0  0 37  0  0  0]
 [ 0  0  0  1  1  1  0 33  0  1]
 [ 0  0  0  0  0  1  0  0 28  0]
 [ 0  0  0  0  0  1  0  0  0 33]]
Accuracy =  0.9694444444444444
```

[ ]:
```

```