Support Vector Machine (SVM)

Goal : train SVM classifier on the raw pixel intensities and then classify unknown digits.

Step 1 – Structuring initial dataset

Step 2 – Splitting the dataset

Step 3 – Extracting features

Step 4 – Training classification model

Step 5 – Evaluating classifier

```
# import the necessary packages
from __future__ import print_function
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import classification_report
from sklearn import datasets
from skimage import exposure
import numpy as np
import imutils
import cv2
import sklearn

from sklearn.model_selection import train_test_split

# load the MNIST digits dataset
mnist = datasets.load_digits()

# take the MNIST data and construct the training and testing split, using 75% of the
# data for training and 25% for testing
(trainData, testData, trainLabels, testLabels) = train_test_split(np.array(mnist.data),
    mnist.target, test_size=0.25, random_state=42)

# now, let's take 10% of the training data and use that for validation
(trainData, valData, trainLabels, valLabels) = train_test_split(trainData, trainLabels,
    test_size=0.1, random_state=84)

# show the sizes of each data split
print("training data points: {}".format(len(trainLabels)))
print("validation data points: {}".format(len(valLabels)))
print("testing data points: {}".format(len(testLabels)))
```

```
training data points: 1212
validation data points: 135
testing data points: 450
```

Imp Parameters

1) Reularization parameter (C)

2) kernel

```
from sklearn.svm import SVC
model = SVC(C=0.5,kernel='linear')
model.fit(trainData, trainLabels)
score = model.score(valData, valLabels)
print (score*100)
```

```
98.51851851851852
```

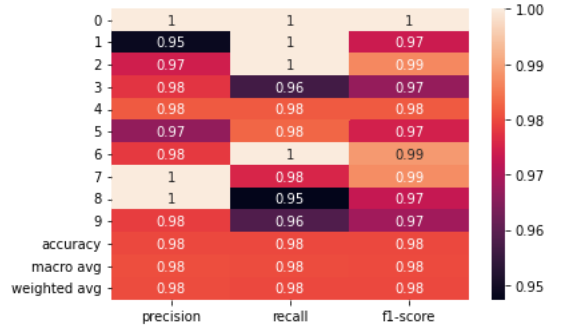
```
# test data
predictions = model.predict(testData)

# show a final classification report demonstrating the accuracy of the classifier
# for each of the digits
print("EVALUATION ON TESTING DATA")
report = (classification_report(testLabels, predictions,output_dict=True)) # made the output of report as dictionary
print (report)
```

```
EVALUATION ON TESTING DATA
{'0': {'precision': 1.0, 'recall': 1.0, 'f1-score': 1.0, 'support': 43}, '1': {'precision': 0.9487179487179487, 'recall': 1.0, 'f1-score
```

classfication report

```
import seaborn as sns
import pandas as pd
sns.heatmap(pd.DataFrame(report).iloc[:-1, :].T, annot=True)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fd71d6a56a0>
```




```
from google.colab.patches import cv2_imshow
for i in list(map(int, np.random.randint(0, high=len(testLabels), size=(5,)))):
  # grab the image and classify it
  image = testData[i]

  prediction = model.predict(image.reshape(1, -1))[0]

  # convert the image for a 64-dim array to an 8 x 8 image compatible with OpenCV,
  # then resize it to 32 x 32 pixels so we can see it better
  image = image.reshape((8, 8)).astype("uint8")
  image = exposure.rescale_intensity(image, out_range=(0, 255))
  image = imutils.resize(image, width=32, inter=cv2.INTER_CUBIC)

  # show the prediction
  print("I think that digit is: {}".format(prediction))
  cv2_imshow(image)

```

```
I think that digit is: 5
```

```
I think that digit is: 1
```

```
I think that digit is: 8
```

```
I think that digit is: 3
```

```
I think that digit is: 3
```

✓ 0s completed at 3:04 PM