assignment7

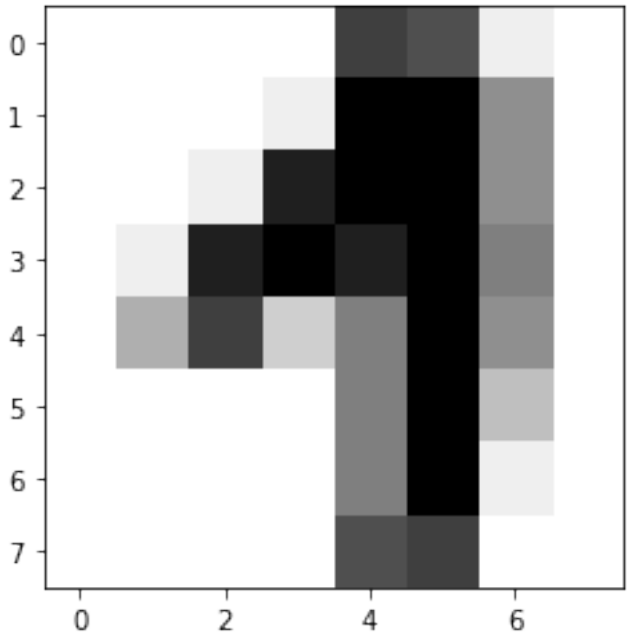
March 29, 2021

[1]:
```
#importing all the necesaary libraries

import numpy as np
import pandas as pd
from sklearn.datasets import load_digits
from sklearn.model_selection import train_test_split
from sklearn.neural_network import MLPClassifier
from sklearn.metrics import accuracy_score
import matplotlib.pyplot as plt
import seaborn as sb
from tqdm import tqdm
import warnings
warnings.filterwarnings("ignore")
```

[2]:
```
# loading the mnist dataset

digits = load_digits()
data = digits.data
target = digits.target
```

[13]:
```
toview = data.reshape(data.shape[0], 8, 8).astype('float32')
plt.imshow(toview[177], cmap=plt.cm.binary)
```

[13]:
```
<matplotlib.image.AxesImage at 0x7fc9eb470690>
```



1 Normalization and splitting dataset into train and test data

[5]:
```
X = data.astype("float32") / 255
y = target

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.2,␣
↪stratify = y)
```

2 Models

[6]:
```
model1 = MLPClassifier(random_state=10, max_iter=500)
model2 = MLPClassifier(hidden_layer_sizes=(400,150,50), activation = 'relu',␣
↪max_iter=500)
model3 = MLPClassifier(hidden_layer_sizes=(400,150,50), activation =␣
↪'logistic', max_iter=500)
model4 = MLPClassifier(hidden_layer_sizes=(64,32,8), activation = 'relu',␣
↪max_iter=500)
model5 = MLPClassifier(hidden_layer_sizes=(32,16), activation = 'relu',␣
↪max_iter=500)
model6 = MLPClassifier(hidden_layer_sizes=(120,64,16), activation = 'relu',␣
↪max_iter=500)
```

```
model7 = MLPClassifier(hidden_layer_sizes=(320,120,32), activation = 'relu',␣
↪max_iter=500)
```

[7]:
```
models = [model1, model2, model3, model4, model5, model6, model7]
```

3 Running cross validation

[9]:
```
from sklearn.model_selection import cross_val_score

cv_scores = []
test_scores = []

for model in tqdm(models):
    cv_scores.append(cross_val_score(model, X_train, y_train, cv = 5))
    model.fit(X_train, y_train) # training on whole train data
    test_scores.append(accuracy_score(y_test, model.predict(X_test))) #␣
↪evaluating on the test set
```

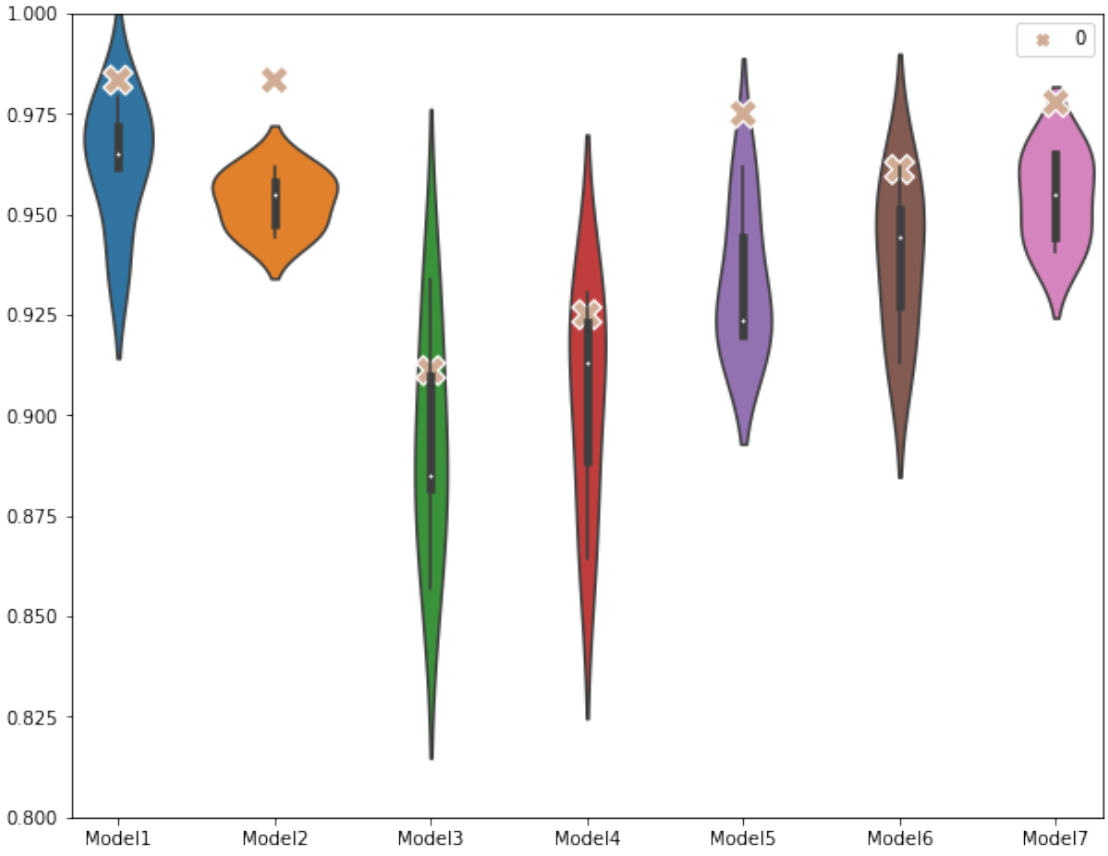
```
  0%|          | 0/7 [00:00<?, ?it/s]
 14%|          | 1/7 [00:23<02:23, 23.91s/it]
 29%|          | 2/7 [01:52<03:36, 43.34s/it]
 43%|          | 3/7 [04:28<05:07, 76.97s/it]
 57%|          | 4/7 [04:59<03:10, 63.35s/it]
 71%|          | 5/7 [05:20<01:41, 50.76s/it]
 86%|          | 6/7 [06:04<00:48, 48.57s/it]
100%|          | 7/7 [07:20<00:00, 62.97s/it]
```

4 Storing the accuracy scores in dataframes

[10]:
```
df_cv = pd.DataFrame(cv_scores, columns = ['fold1', 'fold2', 'fold3', 'fold4',␣
↪'fold5']).T
df_cv.columns = ['Model1', 'Model2', 'Model3', 'Model4', 'Model5', 'Model6',␣
↪'Model7']

df_test = pd.DataFrame(test_scores)
df_test.index = ['Model1', 'Model2', 'Model3', 'Model4', 'Model5', 'Model6',␣
↪'Model7']
```

[12]:
```
plt.figure(figsize=(10, 8))
plt.ylim(0.8, 1)
sb.violinplot(data=df_cv)
sb.scatterplot(data=df_test, markers = 'X', s = 250, palette = 'pink')
```

```
plt.show()
```


1.000
0.975
0.950
0.925
0.900
0.875
0.850
0.825
0.800
0
Model1
Model2
Model3
Model4
Model5
Model6
Model7