

# WuBenjaminAssignment7

April 11, 2022

## 1 CS156 (Introduction to AI), Spring 2022

## 2 Homework 7 submission

2.0.1 Roster Name: Benjamin Wu

2.0.2 Student ID: 013607880

2.0.3 Email address: benjamin.wu01@sjsu.edu

### 2.1 Solution

### 2.2 Import libraries, setup random seed

```
[ ]: import numpy as np
      from sklearn import datasets
      import matplotlib.pyplot as plt
      from sklearn.model_selection import train_test_split
      from sklearn.neural_network import MLPClassifier
      from sklearn.model_selection import cross_val_score
      import seaborn as sns
      import pandas as pd
```

```
[ ]: np.random.seed(42)
```

### 2.3 References and sources

List all your references and sources here. This includes all sites/discussion boards/blogs/posts/etc. where you grabbed some code examples.

### 2.4 Code the solution

```
[ ]: stratifiedDF = pd.DataFrame({'Stratified': [], 'Model': []})
      accuracyDF = pd.DataFrame({'Accuracy': [], 'Model': []})
```

```
[ ]: mnist = datasets.load_digits()

      X = mnist.data
      Y = mnist.target
```

```
for i in range(len(X)):
    X[i] = X[i].astype("float32") / 255
```

```
[ ]: X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.2,
    ↪random_state=0, stratify=Y)
```

```
[ ]: model1 = MLPClassifier(random_state=1, max_iter=10000).fit(X_train, Y_train)
model2 = MLPClassifier(random_state=1, max_iter=10000, hidden_layer_sizes=(400,
    ↪150, 50)).fit(X_train, Y_train)
model3 = MLPClassifier(random_state=1, max_iter=10000, hidden_layer_sizes=(64,
    ↪32, 8)).fit(X_train, Y_train)
model4 = MLPClassifier(random_state=1, max_iter=10000, hidden_layer_sizes=(32,
    ↪16)).fit(X_train, Y_train)
model5 = MLPClassifier(random_state=1, max_iter=10000, hidden_layer_sizes=(120,
    ↪64, 16)).fit(X_train, Y_train)
model6 = MLPClassifier(random_state=1, max_iter=10000, hidden_layer_sizes=(320,
    ↪120, 32)).fit(X_train, Y_train)
```

```
[ ]: models = (model1, model2, model3, model4, model5, model6)
for i in range(len(models)):
    results = cross_val_score(models[i], X_train, Y_train, cv=5)
    accuracy = models[i].score(X_test, Y_test)
    for j in results:
        stratifiedDF.loc[len(stratifiedDF.index)] = [j, "Model" + str(i + 1)]
        accuracyDF.loc[len(accuracyDF.index)] = [accuracy, "Model" + str(i + 1)]
```

```
[ ]: sns.violinplot(x="Model", y="Stratified", data=stratifiedDF, inner="quartile")
sns.swarmplot(x="Model", y="Accuracy", data=accuracyDF, size=10, color="black")
```

```
[ ]: <AxesSubplot:xlabel='Model', ylabel='Accuracy'>
```

