Fall 2021: CSCI 597 Machine Learning

Assignment #1 To be completed in Python

As a simplified example of character recognition, we will investigate the potential for using the very simple K Nearest Neighbour (KNN) classifier to predict whether an input image represents the handwritten digit 9 (nine) or 8 (eight). These digits were selected as they are quite similar to each other, both with a rounded upper part of the digit, but with very similar yet different bottom parts (the nine is like an almost complete eight, with a small gap missing on the left side).

Download (from moodle), NumberRecognition.mat. Note the data downloaded is already divided into training and testing datasets. It also includes data samples for all handwritten digits 0 to 9, but we will be using only 8 and 9 for this assignment.

Python
You may wish to use functions from any of the following packages. Please see Derek's accompanying tutorial on moodle to help you get started, it is much more thorough than could be squeezed into a lecture!

```
import matplotlib.pyplot as plt
import numpy as np
import seaborn as sbn  # better plotting and aesthetics

from pathlib import Path  # just a utility for better cross-platform file-loading
from scipy.io import loadmat
from sklearn.neighbors import KNeighborsClassifier
```

specific functions of interest:
loading a .mat data file: loadmat
reshape to reshape a 3D matrix of 2D images into a decomposed 2D matrix
KNeighborsClassifier for training
predict for prediction

```
auc = sklearn.metrics.roc_auc_score(y_true=y1, y_pred=y2) for ROC area (AUC)
```

Note that as graduate students, you are expected to read code documentation online and explore the use of these techniques. Once you've tried something out, have looked online, and are still having a problem, you can consult the course TA for help (see the course TA Teams group for a question and answer forum). The course TA handles all coding questions. If you don't understand the underlying lecture material, come to the instructor's office hour to ask about your concerns.

Question 1: Build 20 KNN models with varying K=1,2,3,…..,20 in a loop. Provide a plot of testing error rate (as a percentage on the y axis) vs. K (x axis). Provide your python code. Provide the resultant plot. Answer the following questions:

a) Why does testing error rise at high values of K?
b) What is the error rate at the lowest K? Do you expect this to be a reliable performance estimate? Why?

It was previously announced on multiple occasions that each student is required to assemble their own dataset compatible with supervised learning based classification (i.e. a collection of measurements across many samples/instances/subjects that include a group of interest distinct from the rest of the samples). Note you are not required to stick to your chosen dataset all term across all assignments and the course project, you are permitted to switch datasets to tackle a more interesting/challenging problem or to gain more experience. Many students benefit from getting started with an easy dataset (a simple data frame, like a spreadsheet with samples in the rows and measurements/features in the columns. Many such simple examples are available here: https://archive.ics.uci.edu/ml/datasets.php*).*

Question 2: Describe the dataset you have collected: total number of samples, total number of measurements, brief description of the measurements included, nature of the group of interest and what differentiates it from the other samples, sample counts for your group of interest and sample count for the group not of interest. Write a program that analyzes each measurement individually. For each measurement, compute the area under the receiver operating characteristic curve (AUC). Provide an output of the 10 leading measurements (highest AUC), making it clear what those measurements represent in your dataset (these are the measurements with the most obvious potential to inform prediction in any given machine learning algorithm), and what the corresponding AUC values are. Provide this code.

Question 3: Adapt your code from Question 1 to be applied to the dataset that you've organized for yourself. You will need to first randomize your samples into training and testing subsets, so that you can train your machine learning model as you did in Question 1 – this only needs to be done once for this question (no repeat validation is required at this time, just a single randomization of your samples into training and testing groups). Provide the resultant plot and your code. Answer the following question: is the profile of K vs. test error rate similar or quite different to the digit recognition example of Question 1? Elaborate on those similarities/differences – what about your dataset may have contributed to what you observe in this plot?

Deadline: October 1st, 2020, online moodle submission system *in accordance with The TA's detailed submission instructions*.