Due: Feb. 13, 2023 @ 11:59 p.m.

General notes to keep in mind:

- *All deliverables for the assignment must be submitted as a **single ZIP file per group** via the Brightspace D2L course shell. Submissions containing multiple ZIP files per group or those with a file that is not in the ZIP format will NOT be graded.*
- *The code submitted must be written purely using the Python programming language and it should execute within the Python 3.11.1 interpreter running on the Windows operating system (version 10 or above). The submitted code should NOT require external python modules other than scikit-learn 1.2.1 , matlplotlib 3.6.3, pandas 1.5.3 and their dependencies.*
- *Read the "Assignment code submission requirements" carefully and prepare the code accordingly. It is your responsibility to ensure that the submitted code executes. If the grader is unable to execute your code and/or your code does NOT adhere to the submission requirements, your code may not be graded.*
- *The written responses required to the questions in the assignments must be compiled into **single PDF** file named as `report.pdf`. You are encouraged to use LaTeX for typesetting your written responses, but however, the use of Microsoft Word™ or any other such programs is also acceptable.*

Alzheimer's Disease (AD) diagnosis based on glucose metabolism changes in the brain

Alzheimer's Disease (AD) which is clinically referred to as the dementia of Alzheimer's type (DAT) is a deadly disease that gradually causes the death of neurons in the brain (see Figure 1). It is believed that, long before the neuronal death happens, the neurons start exhibiting glucose hypometabolism, i.e., they become "sick" and are unable to perform their regular activities which is reflected in the lesser and lesser amount of glucose they "drink" from the blood. Although, there is currently no cure for DAT, early detection of DAT by identifying the reduced glucose metabolism patterns in the brain maybe help the development of disease modifying treatments in the future.

The goal of this assignment is to compare the glucose metabolism measurements taken from the isthmuscingulate and precuneus regions of the brain between healthy individuals or stable normal controls (sNC) and individuals with advanced or stable DAT (sDAT). Specifically, **you will attempt to build a kNN classifier that can predict if an individual belongs to the sNC group or the sDAT group** based on their brain glucose metabolism signature.

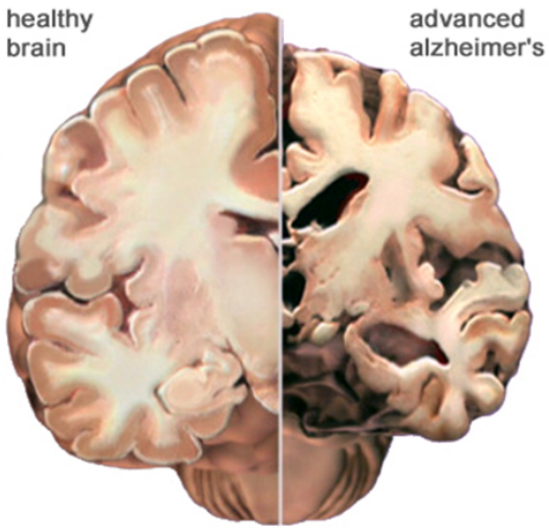

Figure 1: Alzheimer's disease changes the whole brain. The cortex shrinks while fluid-filled spaces called ventricles get larger. Image ©2013 Alzheimer's Association. www.alz.org. All rights reserved. Illustration by Stacy Jannis.

Data

The data for this assignment can be downloaded from here. The training dataset consists of glucose metabolism features taken from the above mentioned two brain regions across 200 sNC and 200 sDAT individuals given in the `train.sNC.csv` and `train.sDAT.csv` files respectively. The `test.sNC.csv` and `test.sDAT.csv` files correspond to a test dataset with the same brain glucose metabolism features taken from another 100 sNC and 100 sDAT individuals respectively. Further, brain glucose metabolism features from yet another 100 sNC and 100 sDAT individuals has been gathered and will be used as the "indpendent" test dataset to perform a "blinded" validation of your submitted kNN model.

A set points on the 2D Cartesian plane are provided in `2D_grid_points.csv` which are useful for generating the visualization of the classification boundaries as shown in Figure 2.

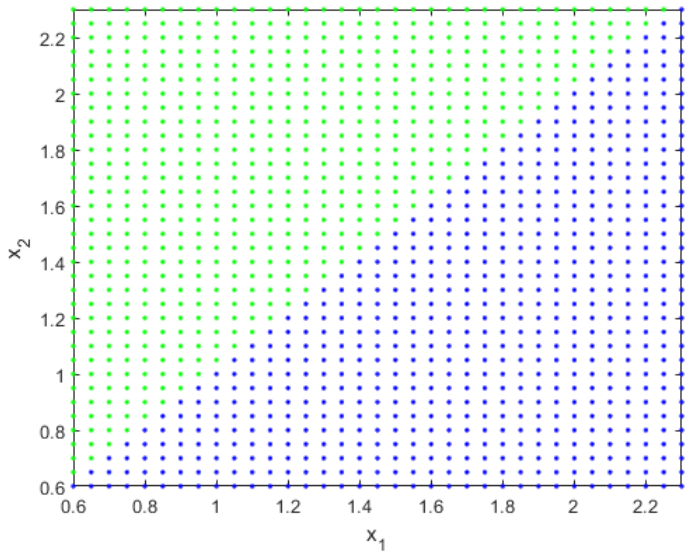

Figure 2: Visualization of the classification boundary corresponding to $\hat{y} = \hat{f}(x_1, x_2)$, where $\hat{f}(x_1, x_2) = 0$ when $x_1 \leq x_2$ and $\hat{f}(x_1, x_2) = 1$ when $x_1 > x_2$, where the "0" and '1" classes are represented by the green and blue dots respectively.

Class label and plotting convention

For consistency, all through your code, the sNC class should be assigned a label "0", whereas the sDAT class should be assigned a label "1". In the plots, green color should be used to represent the sNC class whereas blue color should be used to represent the sDAT class. Further, in the plots, the training data samples should be marked using the "o" marker whereas the "+" marker should be used to mark the test data samples. The "." marker as shown in Figure 2 is reserved for marking the Cartesian grid points.

Question 1 [25 marks]

Train kNN classifiers using the *Euclidean* distance metric and setting $k = 1, 3, 5, 10, 20, 30, 50, 100, 150, 200$ respectively. Generate classification boundary visualization plots similar to Figure 2 for each of the trained classifiers. However, your plots should also contain an overlay of the training and test dataset samples colored based on their *true class labels*. In the plot title, report the error rates achieved on the training and test datasets respectively. Discuss the classification performance of the classifiers trained for various "k" values in the context of over(under)fitting, bias and variance.

Question 2 [25 marks]

Select the classifier with the lowest test error rate from the experiment in Question 1. Using the "k" value from this classifier but changing the distance metric to *Manhattan* distance, train a new classifier. Again, generate the visualization plot for the classification boundary with the training and test dataset samples overlaid and colored based on their true labels. In the plot title report the training and test error rates. Discuss the performance of this classifier in comparison to the classifier with the lowest test error rate from Question 1.

Question 3 [25 marks]

Based on the experiments in Question 1 and Question 2, select the distance metric (i.e., Euclidean or Manhattan) that leads to a lower test error rate. Using this chosen distance metric generate the "Error rate versus Model capacity" plot discussed in *Lecture 4, Slide 3*. As shown in that plot, parameterize "Model capacity" as "$\frac{1}{k}$" and explore the parameter space from "0.01" to "1.00". The "x-axis" must be plotted using the "log-scale" and the training and test rate error curves shown. You need not plot the Bayes Classifier error line (is it possible to plot this using only the information you have?). Discuss the trend of the training and test error rate curves in the context of model capacity, bias and variance. Comment on the over(under)fitting zones in the plot.

Question 4 [25 marks]

Leveraging the experience you gained from the experiments thus far and/or conducting further experiments based on the "kNN improvement strategies" discussed in *Lecture 4, Slide 6*, design the "best" kNN classifier for discriminating between the sNC and sDAT groups using the glucose metabolism features derived from the two brain regions. You may also employ any other strategies that we have not discussed yet to train this "best" kNN classifier. The only restriction is that, you may not use datasets other than the ones provided as part of this assignment. Submit this "best" kNN classifier as the following method:

```
def ytest = diagnoseDAT(Xtest, data_dir):
"""Returns a vector of predictions with elements "0" for sNC and "1" for sDAT,
corresponding to each of the N_test features vectors in Xtest

Xtest     N_test x 2 matrix of test feature vectors

data_dir  full path to the folder containing the following files:
          train.sNC.csv, train.sDAT.csv, test.sNC.csv, test.sDAT.csv
"""
```

The above method will be evaluated on the "indpendent" test dataset by the grader to determine the classification error rate. See note below regarding the grading rubric for this question.

Note on grading

The grading for Question 1, Question 2 and Question 3 will be based on the appropriateness of the submitted code and the written responses. The grading for Question 4 will be based on the relative performance of your trained model. The submission(s) with the best performing model (referred below as 1^{st} ranked model) in terms of error rate (rounded to 4 decimal places) will receive full marks on Question 4 (i.e., 25 marks). All other submissions will receive marks that are proportional to the decrease in performance of their model with respect to the 1^{st} ranked model. For example, if the error rate of the model of a given submission is 10% higher than the 1^{st} ranked model, then that submission will receive 22.5 marks for Question 4.