Home Assignment Hand-out

Course:	Python: Introduction to Coding for Artificial Intelligence and Mechatronics
Assignment:	Day 6
Lecturer:	Dr. Jonathan Lesage jonathan.lesage@gmail.com
Handed out:	Tuesday, March 21, 2023
Submission deadline:	Monday, April 3, 2023, Midnight
To be submitted to:	SCFI at scfi@magna.com
Format:	python module
File name should be:	Assignment_Python_D6_studentlastname_firstname.ipynb
Subject of e-mail should be:	SCFI

INSTRUCTIONS:

Please answer all questions.

If an answer requires a calculation, show it. Don't just quote an answer.

If doubt exists as to the interpretation of any question, contact the lecturer for clarification. See table above for contact information. If no help can be obtained, for any question requiring a written answer, then submit a clear statement of any assumptions you make to get to an answer.

This assignment is open book. For the purpose of study, you may consult outside papers, books, and electronic sources of information, but cannot use text or figures from outside sources as part of your answers.

You can never represent as your own the work of anyone else. Everything you write must be your individual work and must be in your words. Absolutely no collaboration is allowed among students in activities used for academic performance evaluation. The word "collaboration" includes activities such as copying parts of assignments from each other (or from a third-party source), comparing answers, transferring templates or answer formats from one student to another, and discussions on methods to arrive at answers.

Please submit your Jupyter notebook file with the name Assignment_Python_D6_studentlastname_firstname.ipynb

Follow the file naming protocol as given in the information table above.

If there are attachments, print your name on the top left corner of the first page of every attachment you submit and follow the file naming protocol.

Always follow the file format requirements given in the assignment hand out. If, for example, the instructor asks for PDF files only, don't send Word documents, Excel files, or scanned JPG picture files as those will be rejected.

Submit separate files for separate assignments. Don't combine the assignments of different modules of the course into one file of zip folder, even if those are to be forwarded to the same lecturer who taught various modules of the course. SCFI can only track assignment submission if those files are received as separate files and can be saved in separate folders dedicated to each course module.

Magna's email system allows for attachments of 20MB or 25MB (check with your IT team) that should be enough for your file, so whenever possible, avoid submitting zipped files and folders. If any single file would exceed this limit, use only the zip format natively supported by Windows.

To get familiar with classification tasks it is very helpful to use the numpy.random methods to generate random to simulate random values. To this end, in this assignment you will be generating random data for training and testing a nearest neighbor classifier.

1. Create 2 classes of training data as pandas dataframes called “A” and “B”. Each dataframe should have 3 features as columns labeled “X1”, “X2” and “X3”, respectively. Use the numpy.random.normal function to simulate data points with the following parameters:

	Class A		**Class B**	
Features	**Mean**	**Standard Deviation**	**Mean**	**Standard Deviation**
X1	1	0.5	6	1
X2	2	1	12	2
X3	3	1.5	13	3

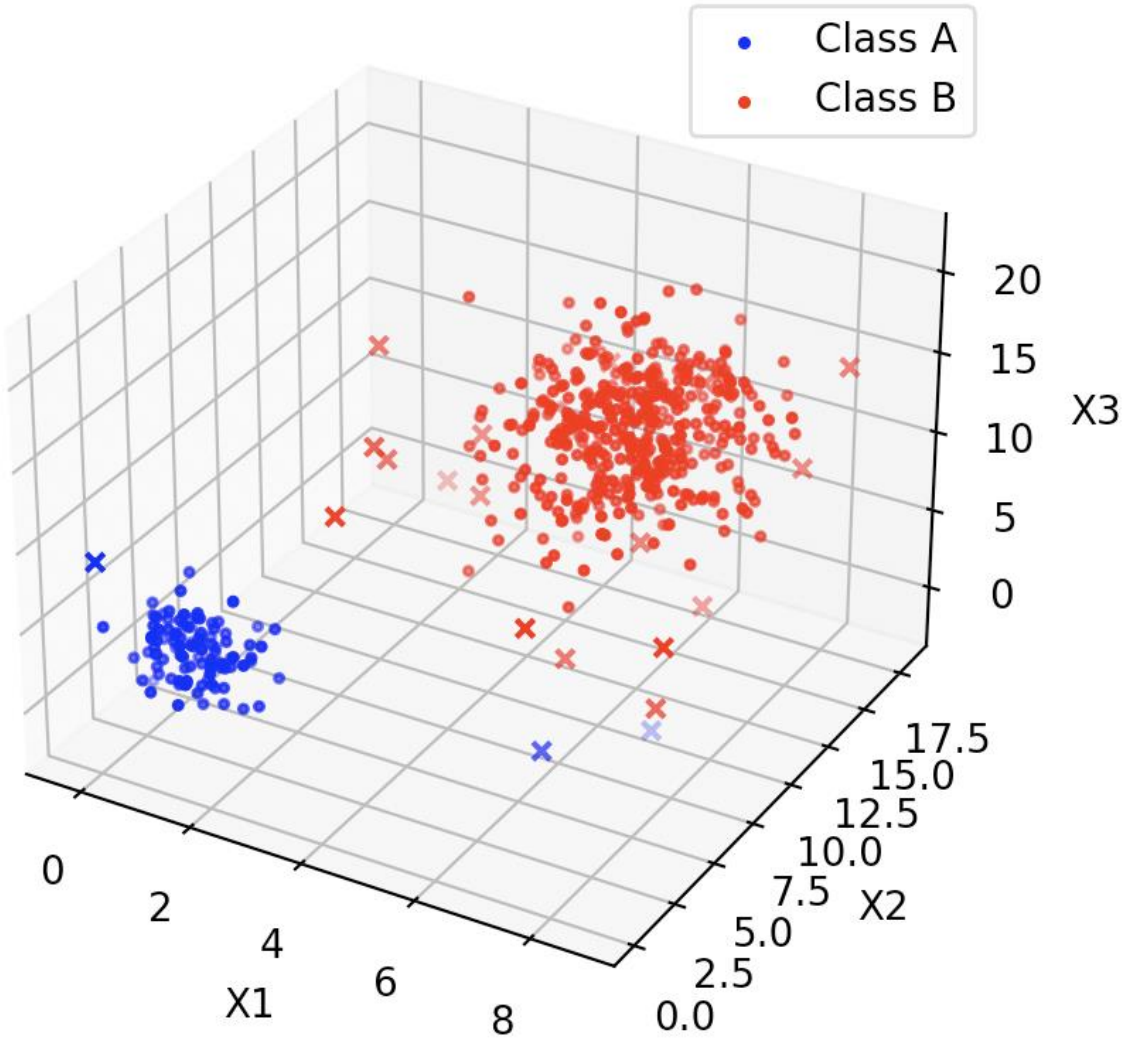
   Class A should contain 100 samples and class B 400 samples.

2. Save the training dataframes “A” and “B” into different sheets in an excel file named “TrainingData.xlsx” using the pandas ExcelWriter function.

3. Read the training data from the data frames for each class into arrays of shape (100,3) and (400,3) for classes A and B, respectively. Identify the minimum and maximum values of each feature and create an array of test data which spans the range of these features of shape (20, 3). Use the numpy.random.uniform(minimum, maximum, shape) to draw random samples spanning the range of each feature.

4. Create a new instance of the sklearn.neighbors KNeighborsClassifier class with the number of nearest neighbors set to n_neighbors = 21. Then use the .fit(features, labels) method to train the algorithm. With the .predict() method, get the predicted labels for the test data. Produce a scatter plot with the training data included as dots and the predicted categories from the testing data marked with x. The colors for class A samples should be blue and for class B samples should be blue. The resulting plot should look

similar to the one shown below. Use the ax.set_xlabel(), ax.set_ylabel() and ax.set_zlabel() methods to include the labels for the X1, X2 and X3 axes.



Provide only a Jupyter notebook file called "Assignment_Python_D6_studentlastname_firstname.ipynb" containing code required to produce all results requested.

Note: Although not part of your assignment, I encourage students to use the notebook created to modify the distributions of classes A and B (mean, standard deviations and sample sizes) to see what sort of influence this has on classification.