**University of Waterloo**

# ECE 657A: Data and Knowledge Modeling and Analysis

**Winter 2021**

**Assignment 1:** Basic Environment Set-up and Classification

**Due:** February 3, 2021 at 11:59pm EST

## Overview

**Collaboration:** You may do your work individually or in pairs. You can collaborate with other on the right tools to use and setting up your programming environment. Both members of the pair must sign up for a PairGroup in LEARN.

**Hand in:** One report per person or pair, via the CROWDMARK website in PDF format. You will need to divide the PDF up into one file for each `[CMX]` question. So both the answers for Q1.1 and Q1.2 can be in a single pdf, it can be multiple pages, which will be assigned to question `[CM1]` on Crowdmark. Some "questions" in this assignment have no output so no pdf if needed. It is best to start each of these on a new page and then drag and drop each onto the relevant question in Crowdmark. You should receive an invite to Crowdmark by email. Overlap and duplicated text between questions is alright, as long as the *entire answer* for each question fully contained within that question's pdf file. Also submit the code/scripts needed to reproduce your work as a python jupyter notebook to the LEARN dropbox.

**Specific objectives:**

- Establish your software stack to carry out data analysis assignments for the rest of the course.
- Load datasets and perform some exploratory plots.
- Study how to apply some of the methods discussed in class and gain experience on the use of moments, kNN classification and evaluation methods.

**Tools:** You can use libraries available in python. You need to mention which libraries you are using, any blogs or papers you used to figure out how to carry out your calculations.

## Dataset 1

Use the Iris dataset (file uploaded on learn dropbox). This dataset is a classic and fairly simple benchmark for basic machine learning algorithms. It includes different features (attributes) of three Iris flower species (setosa, versicolor, virginica).

## Dataset 2

Another dataset is the heart disease dataset (file uploaded on learn dropbox). The dataset contains 14 features (attributes) and 303 instances. The features are multivariate with types - categorical, numeric, ordinal, binary. The target is a binary variable indicating the presence and absence of heart disease using 1 and 0 respectively.

### Answer Question 1, 2, and 3 for both the datasets 1 and 2.

# Question 1: Data Exploration

1. `[CM1]` To begin understanding the dataset, generate a "pairs plot" (also called a scatter plot matrix, `seaborn.pairplot` is one method to do this) of the data. Note that the pairs plot includes the scatter plots of every dimension versus every other dimension. From the pair plot, identify the subplots corresponding to the pairs of features where you see correlation.
   - **For Iris:** Make a single pair plot of all the features and data.
   - **For Heart Disease:** Choose your own subset of 3-5 features for the plot which highlight some *interesting* pattern. You will need to explore different subsets of features or their correlation, distribution etc, in order to choose a set of features.
2. `[CM1]` Justify why you chose those features.
3. `[CM2]` **Question: Calculate and report** the correlation coefficient for the pair of features. To what extent are the features correlated? Do you find any interesting or significant relationships?
4. `[CM2]` Calculate the mean, variance, skew, kurtosis for the datasets and **explain your observation** about the nature of data and the relationships between the features of the dataset.
5. `[CM3]` Are there any notable outliers in the data that should be removed? **Provide a short justification** for your answer in plots and/or words.
6. `[CM4]` **For the Heart Disease dataset only:** Group the features by their variable types and **plot a histogram** of the features to determine the number of present and absent heart disease cases.
7. `[CM5]` **Data Cleaning:** deal with any missing values in the data (use any of the methods discussed in class: dropping data, interpolating, replacing with approximations,. . . ). You can also remove any noise from the data by applying smoothing on some features. **Report any changes you make and justify them.** You can make comparisons if any of these approaches have an impact on classification performance using your validation set.

## Question 2: KNN

Classify the data using a KNN classifier. You will tune the parameter of the KNN classifier using sklearn functions, plot the different validation accuracies against the values of the parameter, select the best parameter to fit the model and Report the resulting accuracy. Carry out the following activities and reporting:

**Basic Model:** The intent for the steps 1-4 is to confirm your numerical answer, so follow the steps exactly.

1. Divide the data into train, validation, and test sets (60%, 20%, 20%) **Note:** set the random seed for splitting, use random_state=275 in the sci-kit learn `train_test_split` function to get the same split every time you run the program.

2. Train the model with the classifier's default parameters. Use the train set and test the model on the test set. Store the accuracy of the model.

3. Then, you should find the best parameters for the classifier, in this case, $k$ for KNN. To find the best parameter you should:

   (a) Pick a value of parameter. Test the following values for validation: $k$: {1, 5, 10, 15, 20, 25, 30, 35}

   (b) Fit the model using the train set.

   (c) Test the model with the validation set. Store the accuracy.

   (d) `[CM6]` When you finish trying all the possible parameter, **plot a figure** that shows the validation relationship between the accuracy and the parameter. Report the best k in terms of classification accuracy.

4. `[CM7]` Now, using the best found parameters, fit the model using the training set and predict the target on the test set. **Report the accuracy, AUC, f-score of your kNN classifier.**

**Your Improved Model:** Try to improve your classification results using any of the performance metrics we have discussed by exploring different ways to improve using your validation set.

5. **Normalization:** Normalize the data using methods we discussed and explain what you used and explain briefly what worked best.

6. **Weighted KNN:** The `KNeighborsClassifier` class has an option for *weighted* KNN where points that are nearby to the query point are more important for the classification than others. Try using different weighting schemes (default, manhatten, eculidean) to see the effect. You can also define your own distance metric to try to improve performance further (testing on validation only of course).

7. `[CM7]` After making these improvements compute your new classification results on the test set and **report the accuracy, AUC and f-score**.

8. Your result from this stage, for each dataset, is also the one you will submit as the solution to kaggle for class comparison, see Question 4.

## Question 3: Analysis

1. [CM8] Explain why you had to split the dataset into train, validation and test sets?

2. [CM8] Explain why you didn't evaluate directly on the test set and had to use a validation test when finding the best parameters for KNN?

3. [CM8] What was the effect of changing $k$ for KNN. Was the accuracy always affected the same way with an increase of $k$? Why do you think this happened?

## Question 4: Kaggle Submission

For both the datasets (heart and iris), submit the kNN predictions on kaggle, where you will be ranked amongst the classmates based on your submission score. There is a separate submission for both the datasets. The prediction submission file format is as follows: For iris, the submission file format is,

| id | species |
|---|---|
| 0 | iris-setosa |
| 1 | iris-setosa |
| 2 | iris-virginica |

For heart dataset, the format is as follows:

| id | target |
|---|---|
| 0 | 0 |
| 1 | 1 |
| 2 | 0 |

In the table, first column is the id of the test set instance, and the second column is the predicted value of the target.
Use the train and test set partition on the kaggle for model training and prediction. You may create a notebook from within the kaggle page and load the test/train dataset to obtain the solution, or you can use framework of your choice but do not forget to include the python code for both the challenge with your assignment submission on learn. The link for kaggle competition pages for the two datasets are as follows:

- Iris competion link: **https://www.kaggle.com/t/e8ef6af29d9745508425066b220feb76**
- Heart competition link: **https://www.kaggle.com/t/632d3b07b18c4b44904fc44439693692**

## Notes

You might find the following links are useful to solve this assignment:

- https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KNeighborsClassifier.html
- https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html
- https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html
- https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html
- https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.GradientBoostingClassifier.html