Assignment 2: Data Analytics with the Fisher Iris Dataset

Submission Deadline: February 22, 2020 (11:59 PM)

Deliverables: Project Report in BMVC format (template can be downloaded from https://www.overleaf.com/project/5e301ad33c2f38000171a776, Ensure you insert your name and Entry Number at the top of your solution) and **submit your code in Python/Matlab/C/C++**. **Make a zip file containing the code and report, and upload the zip file with your name as title on Google classroom.**

This assignment will involve working on the **Fisher Iris** dataset, which is an extremely popular and useful dataset for introductory machine learning owing to the following reasons.

(1) The Fisher-Iris dataset involves multiple classes, which necessitates multi-class classification and may require specialized handling.
(2) It only has 4 attribute and 150 rows, meaning it is small and allows us to explore as well as experiment with multiple classifiers.

About Fisher Iris: The Iris flower data set or Fisher's Iris data set is a multivariate data set introduced by the British statistician and biologist Ronald Fisher in his 1936 paper. This is a very famous and widely used dataset for machine learning and statistics. The data set consists of 50 samples from each of three species of Iris (Iris setosa, Iris virginica and Iris versicolor). Four features were measured from each sample: the *length* and *width* of the *sepal* and *petal* respectively, in centimetres. The fifth column is the *species of the flower* observed (class label).

To load the dataset using *sklearn*, type:

```
from sklearn.datasets import load_iris
data = load_iris()
```

We will split the data into *training* and *test* sets.

TRAINING SET: Is the SEEN DATA which is used to build and train the model. In classification problems such as this, we train the model using the classification error rate: the percentage of incorrectly/correctly classified instances. We use the training data set to help us understand the data, select the appropriate model and determine model parameters.

TESTING SET: This is the UNSEEN DATA. We build a model because we want to classify new data. We are also chiefly interested in the model performance (error rate) on this new data as it is more realistic estimate of the model fit in the real world. Evidently, the training and test sets are mutually exclusive.

Randomly split the Iris data to keep 80% of the data for *training* and 20% data for *testing*. Repeating this process five times results in ***five-fold cross validation*** (wherein you split the dataset into five parts, and use four parts for training and one for testing; repeat the process over five runs with one of the parts as the *test* set during each run).

DATA EXPLORATION: We explore the data to:

- Understand the data
- Summarize the data
- Clean and Prune the data
- Understand relationships between attributes
- Get a preliminary feel for the types of models we think would best fit the data

To explore and visualize the data, for the *training* set (you could either consider ***all data*** and ***class-specific data*****)**

(1) Visualize the distribution of **Sepal Width** (*y*) by grouping **Sepal Length** (*x*) into 5 or 10 bins.
(2) Likewise, make a scatter plot with **Petal Length** (*x*) and **Petal Width** (*y*).
(3) Make boxplots for each of the four attributes: **Sepal Width, Sepal Length, Petal Width** and **Petal Length**.

Summarize your inferences from these plots. Insightful analyses will elicit higher scores.

DATA CLASSIFICATION: Employ supervised learning methods ***Gaussian Naïve Bayes*** and ***logistic regression*** to classify the data. You can look at the following resources for applying these algorithms to multi-class problems.

https://www.quora.com/Can-you-do-multiclass-classification-with-logistic-regression

https://www.youtube.com/watch?v=-EIfb6vFJzc

https://stats.stackexchange.com/questions/142505/how-to-use-naive-bayes-for-multi-class-problems

Employ the ***k*-means** unsupervised learning algorithm to classify the data. For each of these algorithms, tabulate the confusion matrix and also generate the classification report (see https://scikit-learn.org/stable/modules/generated/sklearn.metrics.confusion_matrix.html and https://scikit-learn.org/stable/modules/generated/sklearn.metrics.classification_report.html).

Also, where necessary, attempt to plot the misclassified data points via a 2D or 3D plot (you may choose to use a subset of the features or use principal component analysis for this purpose), and explain the misclassifications.

Based on the results obtained with each classification algorithm, comment on their performance and briefly summarize your learnings from this exercise.