University of Canberra

Faculty of Science and Technology

**Programming for Data Science G (11521)**

**Assignment 2**

**Classification in Data Science**

**Due dates: 23:59 Sunday 01/11/2020 (Week 13)**

**Type:** Individual assignment

**Mark for assessment:** 20

**Minimum mark to pass this assignment:** 4

**Submission:** Submit a .zip file containing either **Python (.py)** or **Jupyter Notebook (.ipynb)** files for your project via Canvas site.

**Late submission:** 5% of the total mark per day (1 mark per day). Information on how to apply for extension can be found in the unit outline on Canvas.

**Task**: Implement a Python graphical user interface (GUI) application for classification using cross validation.

4. input different values of parameter



5.

For each value of parameter:

Initialise classifier with

a parameter value

Set K to a fixed value (=5)

Run cross\_val\_score to get score

Output plot for score versus parameter

Find best score -> best parameter value

X\_train y\_train

8. Output accuracy and confusion matrix

7. y\_pred = classifier.predict(X\_test)

6. Initialise classifier with best parameter value -> run classifier. fit(X\_train, y\_train)

3. Input classifier

1. Input dataset

X\_testy\_test

y\_test

y\_pred

2. From dataset

-> X, y

-> train\_test\_split

-> X\_train, y\_train, X\_test, y\_test

* **GUI**: User can select a dataset and a classification technique then run the system. User does not need to enter filename and technique name, just use mouse to select them.
* **Python packages**: **NumPy**, **Pandas**, **Matplotlib**, **Scikit-learn**, and **Tkinter**.

GUI: Use 2 radio button groups to select classification techniques and data filename. A button to run the entire project.

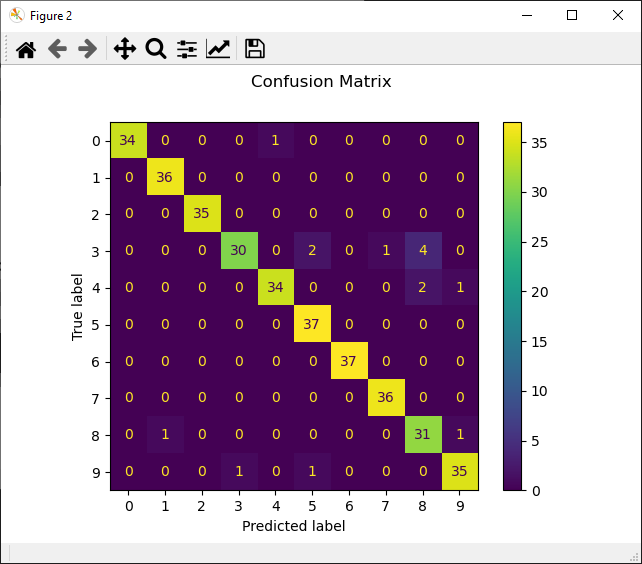
* **Data Set**: 3 sets in **Scikit-learn** (**iris**, **breast-cancer**, and **wine**). These aremulti-class datasets (**M** > 1) and multi-dimensional datasets (Dimensionality **D** > 2). For example, the **iris** dataset has 3 classes (**M = 3**), 50 samples per class, and each sample is a 4-dimensional sample. Each dataset will be partitioned into 2 sets that are **Training** set (80%) and **Testing** set (20%). These sets are separated that means no common sample between them.
* **K-fold cross validation**: the original data set is *randomly* partitioned into K equal sized subsets for all classes. The system will run cross validation **K** times, the first time it gets the first subset for validation and the remaining (K-1) subsets for training, the second time it gets the second subset for validation and the remaining (K-1) subsets for training, and so on. For example, if **K = 5**, the **iris** dataset is partitioned into 5 subsets, each subset has 3 classes and 10 samples per class. The cross-validation technique is used to select the best parameters (e.g., number of centroids) for a classification technique and use them to re-train the whole training set. You can build your own cross validation function or use existing functions in Scikit-learn package.

Recommended: cross validation function in Scikit-learn. See Week 12 Tutorial

* **Classification Techniques**: **K-Nearest Neighbours Classification**, **~~Gaussian Mixture Model~~ Classification**, and **Support Vector Classification**. These techniques are available in Scikit-learn package.

Only two classifiers (classification techniques and their models) are required.

* **Outputs**: accuracy, confusion matrix, and plots for parameters when running cross validation
  + **Accuracy**: accuracy (in %) = number of samples correctly classified \* 100% / total number of samples (both numbers are for the testing set).
  + **Confusion matrix**: Each row of the matrix shows number of samples in a predicted class while each column shows number of samples in an actual class. For example, below are 2 examples of confusion matrix for 10-digit classification



Output confusion matrix on either plot or Console

Classification report for classifier SVC (gamma=0.001):

accuracy 0.96 360

Confusion matrix:

[[34 0 0 0 1 0 0 0 0 0]

[ 0 36 0 0 0 0 0 0 0 0]

[ 0 0 35 0 0 0 0 0 0 0]

[ 0 0 0 30 0 2 0 1 4 0]

[ 0 0 0 0 34 0 0 0 2 1]

[ 0 0 0 0 0 37 0 0 0 0]

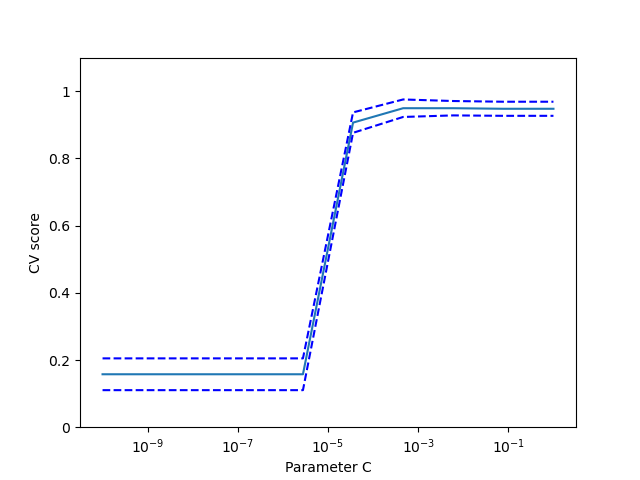
[ 0 0 0 0 0 0 37 0 0 0]

[ 0 0 0 0 0 0 0 36 0 0]

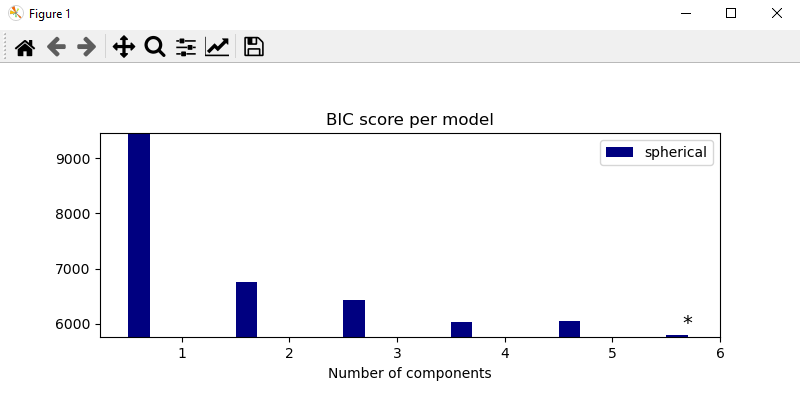
[ 0 1 0 0 0 0 0 0 31 1]

[ 0 0 0 1 0 1 0 0 0 35]]

* + Plots for parameter when running cross validation (for example, parameter K for K-nearest Neighbours Classification, number of Gaussian components in Gaussian Mixture Model Classification, and parameter C in Support Vector Classification). You can choose a fixed value for other parameters in these classification techniques. Below are examples for parameter C and number of Gaussian components.



Any plot style for this output is acceptable



No report is required.

Details and demos will be given in lectures and tutorials from Week 9 to Week 13.

**Marks (total: 20 marks)**

1. **[4 marks]** GUI
2. **[1 mark]** Read data set files and partition them to training and testing sets
3. **[5 marks]** Cross validation and best parameter selection
4. **[5 marks]** Training and testing
5. **[4 marks]** Outputs
6. **[1 mark]** Overall

-- END --