# Assignment 1A

## CAB420, Machine Learning, Semester 1, 2020

This document sets out the four (4) questions you are to complete for CAB420 Assignment 1A. The assignment is worth 15% of the overall subject grade. All questions are weighted equally. Students are to work either individually, or in groups of two. Students should submit their answers in a single document (either a PDF or word document), and upload this to TurnItIn. If students work in a group of two, only one student should submit a copy of the report.

Further Instructions:

1. Data required for this assessment is available on blackboard alongside this document in *CAB420_Assessment_1A_Data.zip*. Please refer to individual questions regarding which data to use for which question.

2. Answers should be submitted via the TurnItIn submission system, linked to on Blackboard. In the event that TurnItIn is down, or you are unable to submit via TurnItIn, please email your responses to `cab420query@qut.edu.au`.

3. For each question, a short written response (approximately 2-4 pages depending on the nature of the question, approach taken, and number of figures included) is expected. This response should explain and justify the approach taken to address the question (including, if relevant, why the approach was selected over other possible methods), and include results, relevant figures, and analysis.

4. MATLAB or Python code, including live scripts or notebooks (or equivalent materials for other languages) should be included as appendices. Figures and outputs/results that are critical to question answers should be included in the main question response, and not appear only in an appendix. Note that MATLAB Live Scipts, Python Notebooks, or similar materials will not on their own constitute a valid submission and a written response per question is expected as noted above.

5. Students who require an extension should lodge their extension application with HiQ (see `http://external-apps.qut.edu.au/studentservices/concession/`). Please note that teaching staff (including the unit coordinator) cannot grant extensions.

**Problem 1. Regression.** The data in `Q1/communities.csv` contains socio-economic data from the 1990 US census for various US communities, and the number of violent crimes per capita (in the column `ViolentCrimesPerPop`). The purpose of the data is to explore the link between the various socio-economic factors and crime.

Given the provided data, you are two:

- Split the data into training, validation and testing sets.
- Train a linear regression model to predict the number of violent crimes per captia from the socio-economic data.
- Train a LASSO regression model to predict the number of violent crimes per captia from the socio-economic data.
- Train a Ridge regression model to predict the number of violent crimes per captia from the socio-economic data.

For your analysis, you should disregard the first five columns (`state`, `county`, `community`, `communityname string` and `fold`). Note that the provided data may also contain missing values, and may need to be sanitized in some way prior to model development.

For LASSO and Ridge models, the validation dataset should be used to select the optimal value of $\lambda$. The performance of all models should be compared on the separate test set, and should consider the predictive power of the model, the model complexity, and the model validity.

Your final response should include:

- Discussion of how the data is handled, including any data cleaning or removal, and how the data is split into training, validation and testing.
- Details of the three trained models, including details such as values for $\lambda$ for the LASSO and Ridge models, and a brief discussion of how these were selected.
- An evaluation comparing the three models, considering model accuracy and model validity.

**Problem 2. Classification.** Land use classification is an important task to understand our changing environment. One approach to this involves the use of data from aerial sensors that captures different spectral reflectance properties of the ground below. From this data, the land type can be classified.

You have been provided with training and testing data (`Q2/training.csv` and `Q2/testing.csv`) that include 27 spectral properties and an overall classification of land type, which can be one of:

- $s$: 'Sugi' forest;
- $h$: 'Hinoki' forest;
- $d$: 'Mixed deciduous' forest;
- $o$: 'Other' non-forest land.

Using this data you are to train two multi-class classifiers to classify land type from the spectral data. These classifiers are to be:

1. A K-Nearest Neighbours Classifier;
2. An ensemble of binary classifiers.

Model hyper-parameters (such as $K$, type of binary classifier and it's parameters, etc) should be evaluated using a validation set that you will need to create by splitting the data. Note that using automatic hyper-parameter optimization is not an acceptable way to choose hyper-parameters for this question. Instead you should perform a series of evaluations, with clear rationale, to evaluate and select appropriate parameters. The resultant models should be evaluated on a testing set.

Your answer to this question should include:

- Details on how the data was split into training, validation and testing sets.
- Details of hyper-parameter selection, including justification for the approach taken and any intermediate results that led to the final models.
- An evaluation and comparison of the final two models, including a discussion exploring any difference in performance between the models.

**Problem 3. Training and Adapting Deep Networks.** When training deep neural networks, the availability of data is a frequent challenge. Acquisition of additional data is often difficult, due to one or both of logistical or financial reasons. As such, methods such as fine tuning and data augmentation are common practices to address the challenge of limited data.

You have been provided with two portions of data from the Street View House Numbers (SVHN) dataset. SVHN can be seen as a 'real world' MNIST, and although the target classes are the same, the data within SVHN is far more diverse. The two data portions are:

1. A training set, `Q3/q3_train.mat`, containing 100 examples of each class ($1,000$ samples total).

2. A testing set, `Q3/q3_test.mat`, containing $1,000$ examples of each class ($10,000$ samples total).

These sets do no overlap, and have been extracted randomly from the original $SVHN$ testing dataset. Note that the training set being significantly smaller than the test set is by design for this question, and is not an error.

Using these datasets you are to:

1. Train a model from scratch, using no data augmentation, on the provided abridged SVHN training set.

2. Train a model from scratch, using the data augmentation of your choice, on the provided abridged SVHN training set.

3. Fine tune an existing model, trained on another dataset used in CAB420 (such as MNIST, KMINST or CIFAR), on the provided abridged SVHN training set. Data augmentation may also be used if you so choose.

All models should be evaluated on the provided SVHN test set, and their performance should be compared.

In addressing this question you should:

- Ensure that all choices (e.g. network design, type of augmentation) are explained and justified in your response.

- Consider computational constraints. You do not need to train the most complex model possible. It is acceptable to use a simpler architecture due to computational constraints, though efforts should still be made to achieve a good level of performance and details regarding these choices and the tradeoff between computational load and performance should be stated in the response.

- Include all relevant figures and/or tables to support the response.

**Problem 4. Domain Mismatch in Data.** Age estimation is a widely studied task related to facial recognition, with applications in domains such as biometrics, and human computer interaction. Estimating age from facial images suffers from many of same challenges as face recognition, such as variations in appearance caused by pose, lighting, and facial accessories (i.e. glasses) or facial hair; and has similar issues around the lack of diversity in training datasets.

The file `Q4/UTKFace.zip` within the data archive contains the aligned and cropped face images from the UTKFace dataset[1]. This archive contains $20,000+$ colour face images, all of which have been cropped and aligned in preparation for further processing. A selection of example raw images (i.e. uncropped images) are shown in Figure 1.

Figure 1: Example raw images from UTKFace. Note that the supplied cropped and aligned images contain only the face regions.

Faces in the archive are named as follows: `[age]_[gender]_[race]_[datestamp].jpg`, where:

- `[age]`: an integer from 0 to 116, indicating the age;
- `[gender]`: either 0 (male) or 1 (female);
- `[race]`: an integer from 0 to 4, denoting White, Black, Asian, Indian, and Others (like Hispanic, Latino, Middle Eastern);
- `[datestamp]`: date and time stamp, in the format of yyyymmddHHMMSSFFF, corresponding to the date and time an image was collected to UTKFace.

The `[age]` value is to be the primary response of your model. You may use or ignore the other variables as you choose.

Using this data you are two:

[1] see `https://susanqq.github.io/UTKFace/` for more details

1. Develop a method to determine that age of a person in an image. There are multiple ways to approach this, including:

    - As a regression task, where you regress from the image (or features from an image) to the age.
    - As a classification task, where an image is classified into it's age. Note that in particular for a classification approach, you may wish to classify the age into ranges (such as decades) rather than having one class for each individual age.

2. Create a hold-out evaluation protocol, where 70% of the data is used for training, 15% is used for validation, and 15% is used for testing. Evaluate the model using this protocol.

3. Create a cross-fold evaluation protocol based on the *race* annotation, i.e. your should create for folds of the data with each fold containing all instances of one *race*. The evaluation should then train the model on 4 of the folds (i.e. 4 races), and the test on the unseen race. This should be repeated such that all folds are the test set in turn.

4. Comment on the performance of the model observed in both the hold-out evaluation scenario, and the cross-fold evaluation scenario. Discuss how the model performs when encountering data that does not match that which is in the training set, and what could be done to improve model generalisation. Points to consider include:

    - Is model performance consistent across all folds, and the hold-out validation?
    - Is the response distribution consistent across all folds (and the hold-out validation)? And does this impact accuracy?
    - Are training sets for different evaluations of similar size? Does this have any impact on performance?

In addressing this problem you are free to select your own approach to determine age. Given the large size of the database, you are encouraged to down-sample the images to a lower resolution to expedite model training (I would suggest $32x32$). Though be aware that if you are too aggressive in your down-sampling you may lose the ability to estimate age at all. Note also that within the data archive there are two files (`Q4/faceparse.m` and `Q4/faceparse.py`) that will load the data and extract the age, gender and race annotations.

In summary, your answer should:

- Briefly explain the method you have chosen to use, and provide a brief justification your choice.
- Provide an analysis and discussion of your models performance on the two evaluations (hold-out and cross-fold), being sure to include all relevant figures and/or tables to support your discussion.