# QBUS3820 Group Assignment (20 marks)

September 29, 2017

## 1 Background

Developing a predictive model for ATM cash demand is an important task for every bank. Suppose that you are employed by a bank, and your task is to optimise the bank's cash management by making smarter decisions about reloading its ATM network.

The variable `Withdraw` in the dataset `ATM_training.csv` is the total cash amount withdrawn per day from an ATM, recorded from the ATM network of a bank. The response variable and covariate variables are described in the following table.

| Variable | Description |
|---|---|
| Withdraw | The total cash withdrawn a day (in 1000 local currency) |
| Shops | Number of shops/restaurants within a walkable distance (in 100) |
| ATMs | Number of other ATMs within a walkable distance (in 10) |
| Downtown | =1 if the ATM is in downtown, 0 if not |
| Weekday | = 1 if the day is weekday, 0 if not |
| Center | =1 if the ATM is located in a center (shopping, airport, etc), 0 if not |
| High | =1 if the ATM has a high cash demand in the last month, 0 if not |

Your task is to develop a model for predicting the cash demand `Withdraw` based on the covariates.

The test dataset `ATM_test.csv` (not provided) has the same structure as the training data `ATM_training.csv`.

### 1.1 Test error

For the measure of prediction accuracy, we use mean squared error (MSE), computed on the test data. Let $\widehat{y}_i$ be the prediction of $y_i$ where $y_i$ is the i-th withdraw in the test data. The test error is computed as follows

$$\text{Test\_error} = \frac{1}{n_{\text{test}}} \sum_{y_i \in \text{test data}} (\widehat{y}_i - y_i)^2,$$

where $n_{\text{test}}$ is the number of observations in the test data.

## 2 Submission Instructions

1. Each group needs to submit three files (or more if necessary) via the link in the LMS site

   - A document file, named document.pdf, that explains the model, how the model is developed and estimated.
   - Two Python files: A Python code file, named `analysis.py`, that implements your data analysis procedure and another Python file, named `prediction_error.py` that produces the test error. You might submit additional files that are needed for your implementation.

2. The document file should describe your data analysis procedure: what model is used, how the model is trained, etc. The description should be detailed enough so that other data scientists, who are supposed to have background in your field, understand how to implement the task.
3. The Python files will be run using Anaconda Python, **with the files `ATM_training.csv` and `ATM_test.csv` in the same folder as the Python files**. **If you use deep learning models, then please assume that Keras (with Tensorflow backend) has been installed**.
   - If the training of your model involves generating random numbers, the random seed in `analysis.py` must be fixed, so that the marker expects to have the same results as you had.
   - The file `prediction_error.py` must have the following format

     ```
     import pandas as pd

     ATM_test = pd.read_csv('ATM_test.csv')

     # YOUR CODE HERE: code that produces the test error test_error

     print(test_error)
     ```

     The idea is that, when the marker runs `prediction_error.py`, he/she expects to see the same test error as you did.

## 3 Marking Criteria

This assignment weighs 20 marks in total. The content in document.pdf contributes 10 marks, and the Python implementation contributes 10 marks, but the priority is given to the prediction accuracy (see below). The marking is structured as follows.

1. The accuracy of your prediction: Your test error will be compared against a benchmark test error obtained by the teaching team (sure, you can beat us!). The marker first runs `prediction_error.py`
   - Given that this file runs smoothly and a test error is produced, 10 marks will be allocated proportionally to your prediction accuracy. **If your test error is as small as the benchmark error, you're awarded the total 20 marks (the document file is then no longer needed).**
   - If the marker cannot get `prediction_error.py` run, some partial marks (maximum 5) will be allocated based on `analysis.py`.
2. Appropriateness of your method as described in document.pdf (maximum 10 marks)
   - If your Python implementation part runs smoothly, you're likely to receive the total 10 out of these 10 marks.

- If your Python implementation fails, some partial marks (out of 10) will be given based on the appropriateness of your data analysis procedure.

## 4 Award

The team with the best prediction accuracy will be announced (unless they don't want to be named) and receive a prize! Their test error and the test error obtained by the teaching team, model, etc will also be made available.

## 5 Errors

If you believe there are any errors with this assignment please email the lecturer immediately at minh-ngoc.tran@sydney.edu.au.