**DAT Project 3: Anomaly Detection**

Spring 2023

**Introduction**

This project will focus on building anomaly detectors, using an SVM and a Neural net (Auto encoder)

The data we will use is from a review paper by Guansong Pang

See:

<https://towardsdatascience.com/adrepository-anomaly-detection-datasets-with-real-anomalies-2ee218f76292>

<https://github.com/GuansongPang/ADRepository-Anomaly-detection-datasets>

<https://arxiv.org/abs/2007.02500>

**Data**

There are 4 data sets that I have downloaded and attached below:

Bank-additional\_full\_normalized

Census income full mixed binarized

Celeba\_baldvsnobald\_normalized

KDD2014\_donors\_10\_nomissing\_normalized

For each dataset, the variable *class* indicates whether the data is normal or anomalous.

Pick one of the four sets

**Requirements**

1.) Try to locate some background on the data, ideally a data dictionary.

2.) Some of the data sets are very large, particularly the census set. You may have to reduce the number of variables you used in the detection to just those that seem particularly relevant or interesting. You may also need to work with a limited number of rows, at least for the SVM based encoder.

3.) Determine how many normal data rows you have (class=0) and how many anomalies. Split your data into an anomalous set, and split the anomalous set into two pieces (test and validation) of roughly equal size. Split the regular data (class 0) into three sets, (train, test and validation) with the test and evaluation data sets the same size as the anomalous test and validation sets. Use all the remaining “regular” data as your training set.

Remember this is “unsupervised” learning to create an anomaly detector.

4.) Do some exploratory data analysis to understand what you are working with. Do a range of various types of summary or descriptive analysis.

Include a heat map of the correlation of the variables.

5.) Create an SVM based anomaly detector, set it with a variable cutoff boundary, set this to 5% to start with, but be sure it is adjustable in code.

SVM gives you various options of operation, try several and find the optimum.

6.) Produce a confusion matrix of correct and incorrect anomaly detections, using your validation data. Show this at several different boundary levels.

7.) Now build an anomaly detector (autoencoder) using Tensorflow and Keras. Optimize its operation at least a bit.

Determine how to set up the boundary to detect anomalous and non-anomalous results.

8.) Decide which of the two methods seems to work best, using your validation data. Can you figure out how to make simultaneous use of both methods to improve overall performance? Do the two methods make the same mistakes in anomaly detection or are there differences in the two?

Once you have figured out what method, or combination of methods, works best, set it up and run it on your test data.

Report the rate of correct assignment of both anomalies and regular (normal) data.