Programming Assignment 3: Association Analysis and Deep learning (10 points)

Shen-Shyang Ho (Dr.)

October 27, 2022

- In this assignment, you will be using the dataset assigned to you in Assignment 1 for a Deep Learning Task and a newly assigned dataset for an association analysis task.
- For the association analysis task, you will use Mlxtend (http://rasbt.github.io/mlxtend/)
- For the deep learning task, you will use Tensorflow (https://www.tensorflow.org/) via Keras (https://keras.io/) or PyTorch (https://pytorch.org/)

1. **[Association Rule Generation from Transaction Data]**
   (a) Download transaction dataset to your local drive.
      i. **Go to the following Google Drive link (Students must be logged in to their Rowan accounts)**:
      https://drive.google.com/drive/folders/1LuFEbgq3IvisEXT1jOZ-H4jWeqzqEH3m?usp=sharing
   (b) Download the 'Grocery_Items_{DATASET_NUMBER}.csv' file from the Google Drive Link.
   (c) Using minimum support = 0.01 for the Apriori Algorithm, extract the frequent itemsets from your dataset. (1 point) (see http://rasbt.github.io/mlxtend/user_guide/frequent_patterns/apriori/)
   (d) Using minimum support = 0.01 and minimum confidence threshold = 0.1, what are the association rules you can extract from your dataset? (1 point)
   (see http://rasbt.github.io/mlxtend/user_guide/frequent_patterns/association_rules/)
   (e) List the association rule(s) (i.e., one or more rules depending on your dataset) that have the highest confidence. What is that confidence value? (1 point)

2. **[Image Classification using CNN]** A 4-class classification (ignore negative class) using a convolutional neural network with the following simple architecture (2 point)
   (a) 1 Convolutional Layer with 8 3×3 filters.
   (b) 1 max pooling with 2×2 pool size
   (c) **Flatten** the Tensor
   (d) 1 hidden layer with 16 nodes for fully connected neural network
   (e) Output layer has 4 nodes (since 4 classes) using 'softmax' activation function.

   (Use 'Relu' for all layers except the output layer.) for 5 epochs using 'adam' optimizer and 'categorical cross entropy' loss function. For validation split, you will use 20%. For batch size, you can pick a size that will not slow down the training process on your machine. (see https://keras.io/examples/vision/mnist_convnet/)

- Plot a graph to show the learning curves (i.e., x-axis: number of epochs; y-axis: training and validation accuracy - 2 curves) (1 points)
- Do **ONE** of the following (1 point):
  (a) Train the CNN using 2 other filter size: 5×5 and 7×7 with all other parameters unchanged
  (b) Train the CNN using 2 other number of filters: 4 and 16 with all other parameters unchanged
  (c) Train the CNN using 2 other number of nodes in the hidden layer: 8 and 32 with all other parameters unchanged
- Plot the learning curves (i.e., x-axis: number of epochs; y-axis: training and validation accuracy - 2 curves) for the above 2 configurations (1 points)
- Describe and discuss what you observe by comparing the model performances. Do the models overfit or underfit or just right? Are the models performing better than the classifiers you used in Assignment 2? Compare their performances. (2 point)