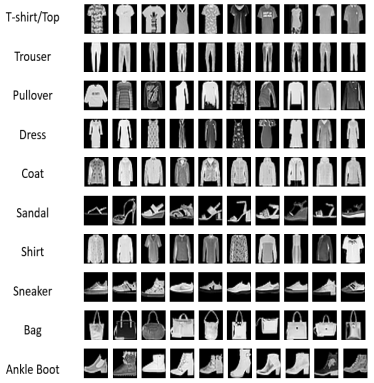
# Practice: Classifying Clothing Images Using a Convolutional Neural Network and the Python API

The Modified National Institute of Standards and Technology Fashion (**MNIST Fashion**) data set contains images of clothing from the Zalando company. Approximately 15,000 images have been sampled from the data set to train an image classification model using the image pixels. Each image is a 28x28 grayscale pixel bounding box saved as a separate PNG files. Below are 10 images of each type of clothing.




1. From the Jupyter Lab Home directory, select the plus symbol to open a new page and then select Python under Notebook to create a new notebook.

2. Load the os, sys, SWAT, numpy, pandas, and matplotlib packages. Use the %matplotlib inline statement to create graphics within the notebook and set the CAS option to print CAS messages in the notebook.

3. Connect to CAS and create a connection object called **conn**.

4. Use the addCasLib action from the table action set to add a CAS library connecting to the course data directory, where the images are located.

5. Load the images from the CAS library to the CAS server. To do so, load the image action set and use the loadImages action to load the images from the **mnist_fashion** folder on the server. Note, the mnist_fashion folder has ten sub folders. Each folder is named for a unique number, or target level, and holds the specified images. Therefore, use the recurse=TRUE and labelLevels=1 arguments to load all images in subdirectories and keeping the folder name as the image label. Save the CAS table with the name **fashion**.

6. Use the summarizeImages action to view summary information of the images and use the columnInfo action from the table action set to view the variable names in the **fashion** data set.

7. Load the simple action set and use the freq action to view the number of images in each level of the variable _label_ created by the loadImages action.

8. Use the srs action from the sampling action set to partition the data into 40% for training, 30% for validation, and 30% for testing. Set the partind argument to TRUE to add the partition indicator variable to the **fashion** data table.

9. Use the deepLearn action set to build a convolutional neural network with an input layer, a convolution layer, a pooling layer, a fully connected layer, and an output layer.
    - Use the buildModel action to initialize the CNN.
    - Add an input layer with 1 channel, set the width and height to 28 for the image size, scale the image to 0.004, and standardize the images.
    - Connect the input layer to a convolutional layer with 10 filters. Let each filter be 5x5 with a stride of 1 and set the activation to rectified linear and the initialization to Xavier.
    - Connect the convolutional layer to a pooling layer. Let the localized regions be 2x2 with a stride of 2 and let the pooling summary be the maximum.
    - Connect the pooling layer to a fulling connected layer with 100 neurons, a drop out rate of 0.4, an activation of rectified linear, and Xavier initialization.
    - Connect the fully connected layer to the output with softmax activation. Use the modelInfo action to ensure the model has the correct five layers.

10. Use the dlTrain action to train the convolutional neural network using the _image_ variable as input and the _label_ variable as the target. Train for 100 epochs, use the momentum optimizer algorithm, a learning rate of 0.01, minibatch size of 64, and save the weight estimates.

11. Use the dlScore action to score the test data set. Save the scored information and use the copyVars argument to save the target and input in the scored output CAS table as well.

12. Use the crossTab action from the simple action set to print the actual vs predicted matrix and use an assignment statement with the action to download the table to the client as a data frame.

13. Use the crossTab data frame to create a data frame of the proportion of correct and misclassified predictions for each target level using open source functionality to see which types of clothing are the easiest and hardest to classify.

14. Create a bar chart of the proportion of misclassified for each target level on the client.

15. End the CAS session.