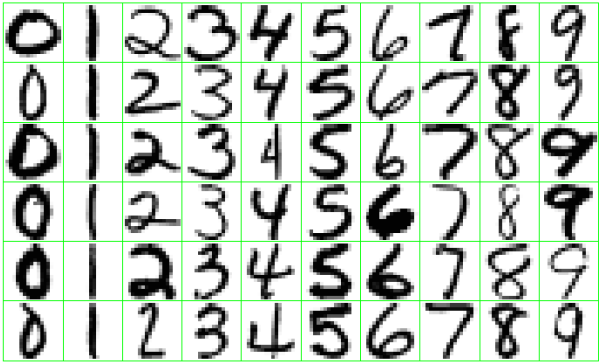
**Bayesian Learning and Support Vector Machines**

In this exercise we are going to classify hand-written number (Hand Written Digit Recognition). The file ZipData 2019.csv contains the data table. This figure shows what it contains. Each image is an array of 16 by 16 pixels:



These digitalized images were obtained by scanning hand-written zip code numbers from enveloped processed by the US Postal Service. These are gray-scale images with a 16 x 16 resolution. Each pixel’s intensity goes from -1 to 1 (White to black). Images have been processed so they have approximately the same size and orientation. You task us to determine from the digitized image the number that was written (0; 1; : : : ; 9) as precisely as possible. If it’s your algorithm is precise enough, it will be used in an automatic envelop selection algorithm.

This is naturally a classification problem and it is very important to keep the error rate very low to avoid mail misdirection. Column one contains the variable to classify coded in the following manner: 0=‘cero’; 1=‘uno’; 2=‘dos’; 3=‘tres’; 4=‘cuatro’; 5=‘cinco’;6=‘seis’; 7=‘siete’; 8=‘ocho’ y 9=‘nueve’ (0=‘zero’; 1=‘one’; 2=‘two; 3=‘three’; 4=‘four’; 5=‘five’;6=‘six’; 7=‘seven’; 8=‘eight’ y 9=‘nine’). The other columns are the predictive attributes. Each row of attributes represents a 16x16 image, thus each row has 256 predictive variables.

Carry out the following tasks:

1. Load file ZipData 202.csv using Pandas
2. Carry an exploratory data analysis with the variable to predict (the class). Compute correlation matrix (to learn which pixels are more definitive). Answer the following question: Is this a class balanced problem? Justify your answer.
3. Use Bayesian Learning (Naïve Bayes) and Support Vector Machines from Scikit-Learn methods to generate predictive classification models for dataset ZipData 2020.csv using 80 % of data for training and 20% for testing
4. Generate a confusion table and calculate accuracy
5. Can you compute recall and sensitivity?

As you can see, this is a multiclass classification problem, not binary classification. Thus variables tp, tn, fp, fn are not longer avialable because now we have a cxc matrix, where c is the number of classes, going from class “zero” to class “nine”.

To calculate a class recall number, you must divide the main diagonal element from that class (TP) by the sum of the entire row. Thus for class I we would have (better to create a for cycle for each class):

recall=cm[i,i]/sum(cm[i,:])

Also, the “Numero” column, that is the variable of interest, is not numeric, thus we have to recode it:

Instead of simply:

y=zipdata['Numero']

Add:

from sklearn.preprocessing import LabelEncoder

le=preprocessing.LabelEncoder()

zipdata['Numero\_enc']=le.fit\_transform(y)

y=zipdata['Numero\_enc']