# 1 Zip Practical Work 1

In this problem we are working on the classification of handwritten digits. The data that we are using to train our model is a dataset of normalized handwritten digits automatically scanned from envelopes by the U.S Postal Service in 16 x 16 grayscale images (from -1 to 1). The dataset is composed of 7291 digits for training and 2007 for testing where we will be able to evaluate or accuracy. Below we can see an image of how the data looks like:

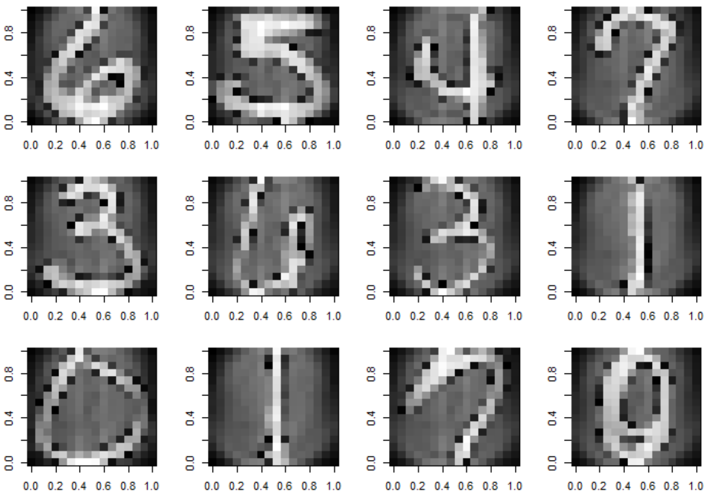

Figure 1

As specified in the statement, we are not going to use all our training set, we are going to take a 5% subset in a random fashion. This subset is the one we are going to use to perform our training on, which corresponds to $\approx 365$ observations (smaller than our training set).

Once we have created our training subset, we decided to define two matrices, the response matrix (Y) which corresponds to the first column of our dataset, and the predictor matrix (X) which corresponds to the remaining columns.

- Because our response matrix is a single column and we are approaching this problem as a classification one, we need to perform a one hot encoding where we create an additional column per possible result. Since we have 10 possible values, we are going to have a column for each possible digit $[0, 9]$.

- We need to center the predictor matrix. We are going to center it and not to normalize or standardize it because if we do the latter, we would not consider the position of the pixel and we think that it is relevant.

Note that the same we do for the training set is going to be also done for the test set, since we need both datasets to be in the same conditions in order to try our model in the test set.

Once we have defined the matrices, we are going to perform a multivariate regression with the training data and we are going to compute the average $R^2$. In order to compute the average $R^2$ we have added all multiple R-squared obtained from each response variable and we have divided it by 10. The final result is of $\approx 0.92$ by the seed with the see we chose. In the same fashion we have also calculated the $R^2$ by using the LOO which gave us $\approx -3.51$.

When we perform the multivariate regression to our test data, we obtain an error rate of $\approx 40\%$. In order to know which is the number that we decide that is predicted, we take the one which has the maximum response. Because of the high error, we are going to perform a PCR using the LOO validation and we are going to decide how many components do retain information for our prediction. In our particular case, we have seen that 20 components, seem to retain enough information where we have an error rate of $\approx 0.17$.

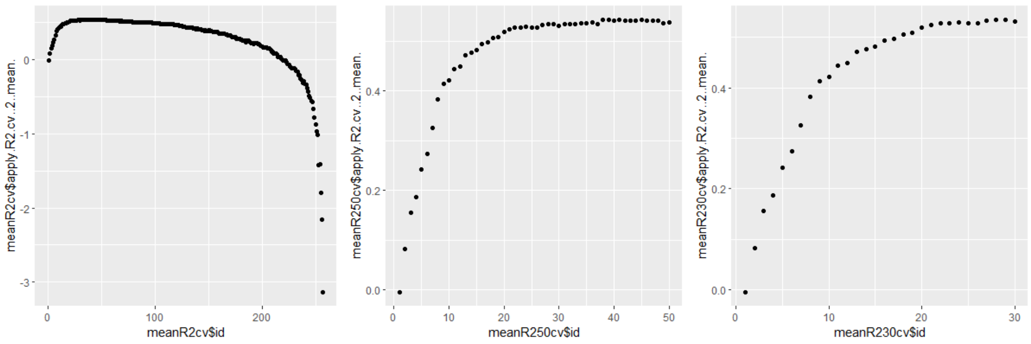

Figure 2: From left to right: 255, 50, 30 components

The obtained accuracy that we have is very minimal considering what other people can obtain in the MNIST data as we can see that the Tensorflow tutorial gets a 98% of accuracy. Why this happens?

Well first if we take a look at our data and MNIST and perform a t-SNE in both:

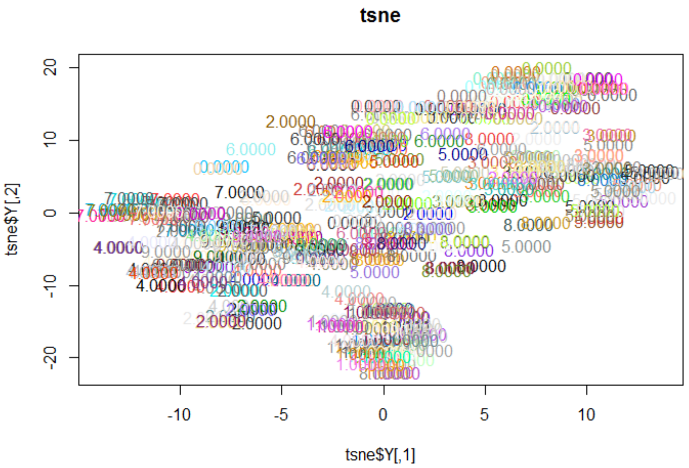

Figure 3: Our 5% data t-sne

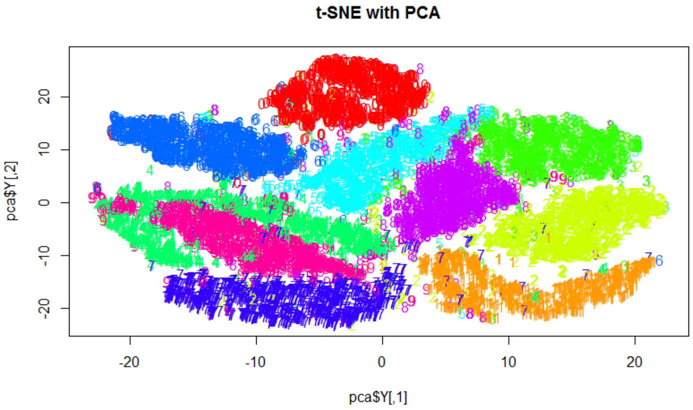

Figure 4: MNIST

We can observe that the data we are using is not enough and does not show a pattern as clear as in the MNIST t-SNE. Regardless, we have to say that the accuraccy would increase the more observations we use since our dataset is very limited multivariate regression gives us a poor accuracy. PCR performed better because it was able to capture latent interactions that multivariate could not. Both performances are far from what a CNN can obtain but the scope of this project was not to obtain a high accuracy rather to work with the aforementioned methods.