**Assignment 2**

1. The ages of a group people are listed in an increasing order as 13, 15, 16, 17, 19, 20, 20, 21, 22, 23, 25, 25, 26, 30, 31, 32, 35, 35, 35, 36, 48, 50, 52, 55, 70.
2. Use clustering to smooth these data, the number of clusters is 3. Illustrate your steps. Comment on the effect of this technique for the given data.
3. How might you determine outliers in the data?
4. What other methods are there for data smoothing?
5. Use min-max normalization to transform the value $72000 for price onto the range [0.0, 1.0](income range $12800 to $100000).
6. Use z-score normalization to transform value $72000 for price( μ = 54,000, σ = 16,000).
7. Use normalization by decimal scaling to transform the value $72000 for price
8. Comment on which method you would prefer to use for the given data, giving reasons as to why.

# 3. Play MNIST with PCA(code)

## **Introduction**

The MNISTdatabase is an image dataset of handwritten digits (0~9). Each image in MNIST is represented by a pixel matrix of 28\*28, totally 784 elements. Since the images in MNIST are in gray scale, the value of each pixel ranges from 0 to 255, representing different gray degrees.

For this assignment, we have a ".csv" file, the file represents image samples. The first column indicates the handwritten digit of the image (label). Each of the rest columns denotes the value of each pixel axis (from 1 to 28 for both x-axis and y-axis) in an image. For example, the column "1x28" denotes the pixel value at x=1 (the first row) and y=28 (the 28th column). Now we want you to apply PCA to MNIST for data analytics.

Hints:

#### **Load "mnist.csv" with pandas**

#### **Print the shape of the data**

#### **Take the pixel values of the first sample and reshape it to (28, 28)**

#### **4. Plot the first sample with plt.imshow()**

#### **5. Split the data into X (the pixel values) and y (the digit label)**

#### **6.Standardize X by dividing 255.0**

#### **7. Apply PCA to X and obtain the eigenvalues as well as the eigenvectors (please implement it with numpy instead of other advanced libraries)**

#### **8. You may notice that the results contain the complex numbers of which the imaginary part is zero. Please remove the imaginary part with np.real()**

#### **9. Use the top-30 eigenvectors (with the largest thirty eigenvalues) to transform the training data to Z.**

#### **10. Reconstruct the training data from Z to**X~**, then multiply it by 255.0**

#### **11. Plot the sample of the reconstructed data**X~**. Compare it with the original one.**