Fall 2016
CSCE 666 Pattern Analysis
Homework #3

Due date: 10/31/2016

In recognition of the Texas A&M University policies of academic integrity, I certify that I have neither given nor received dishonest aid in this homework assignment.

Name: Caio Duarte Diniz Monteiro Signature:

Problem 1 (20%)

Download the dataset 'hw3p1_data.mat', which contains frontal images of senior faculty in the TAMU Computer Science Department; each image is represented by a row vector in data matrix 'x'. You are to generate a PCA decomposition of these faces using the 'snapshot' approach.
(a) Generate an image of the average face
(b) Generate images of the first six eigenvectors (i.e., "eigenfaces")
(c) Generate a 2D PCA scatter plots of the corresponding principal components
(d) DISCUSS YOUR RESULTS.

As observable in Figure 1 below, the result of averaging all faces is still a 'face', since all the faculty photos were taken with the same angle and distance, the averaged image is a smoothed face, with eyes, nose, and mouth clearly represented.

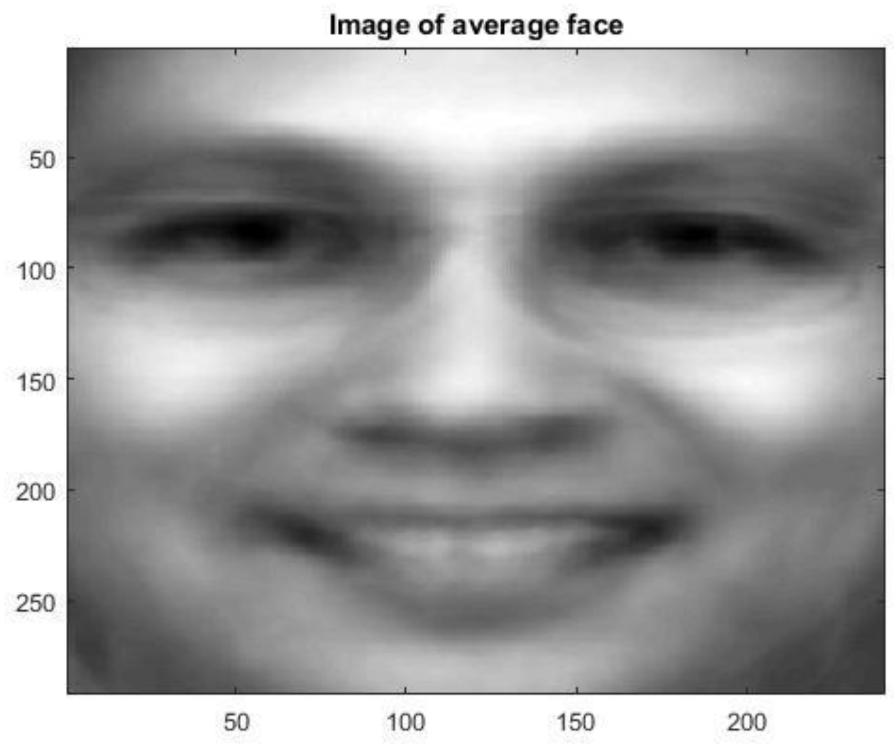

Figure 1. Average face of faculty images.

Figure 2 presents the first six eigenvectors extracted using the Snapshot approach. We can make the point that these six eigenfaces form a basis in which all the faculty members images can be generated with a certain level of detail as a combination of these eigenfaces.

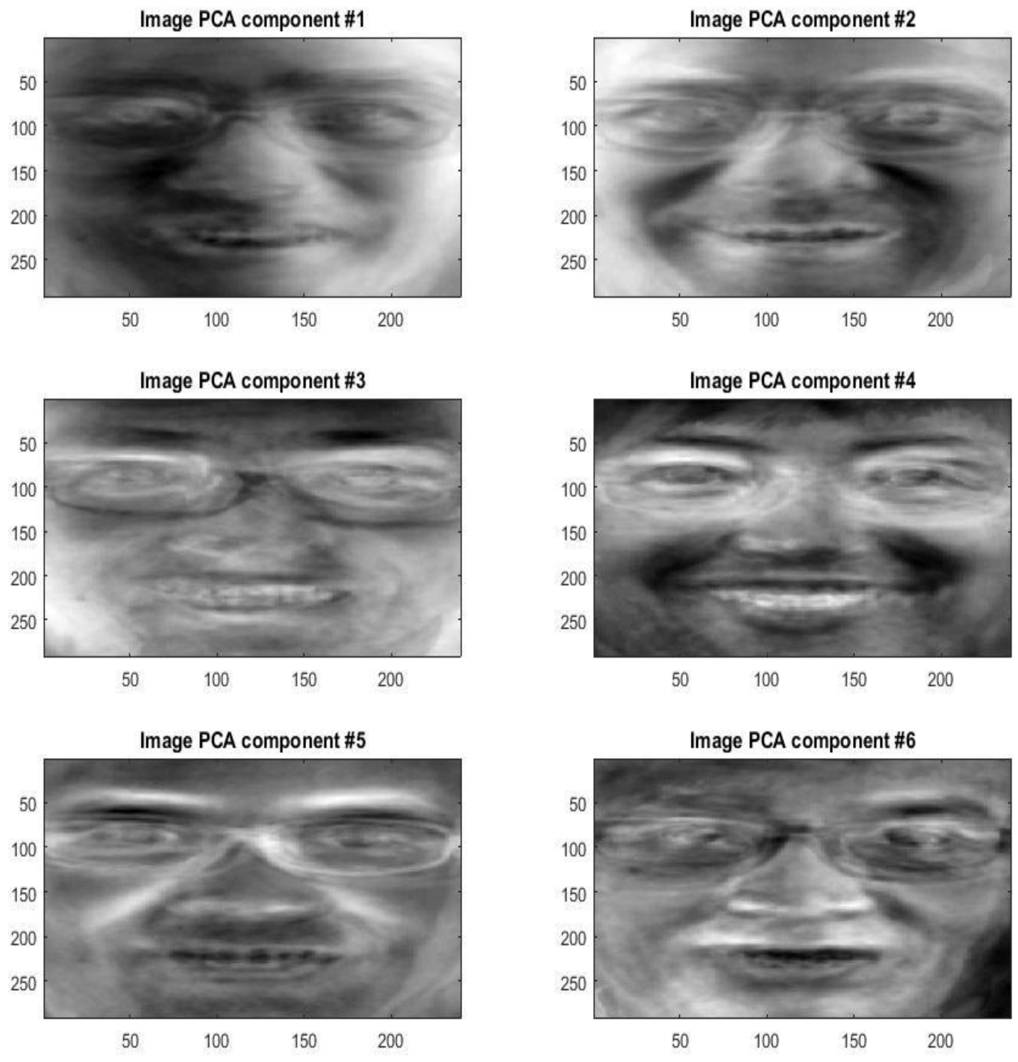

Figure 2. *Eigenfaces* from each of the first six principal components.

For the last part of the exercise, we had create a scatter plot with each faculty image being represented by its first two principal components. Figure 3 presents the scatter plot. As one can observe, majority of the women faculty concentrate on the bottom left of the plot, and also professors with beards can be seen on the upper half of it.

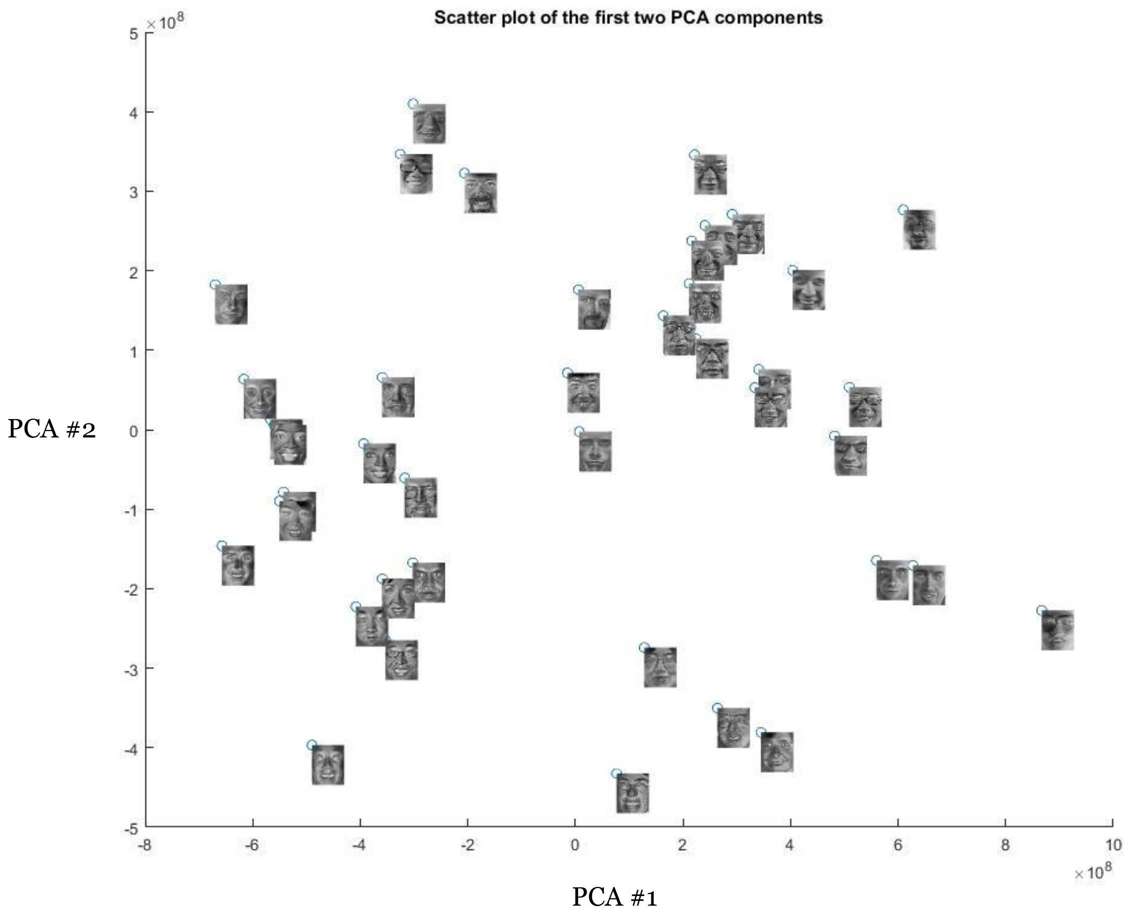

Figure 3. Scatter plot of faculty images using the first two principal components.

Problem 2 (20%)

(a) Download the dataset 'us_city_distance.mat', which contains the shortest geodesic distance (in meters) between pairs of cities in the United States. You are to compute the embedding manifold, and generate 2D and 3D scatter plots for these cities. How many dimensions are needed to capture the manifold?
(b) Repeat (a) on the dataset 'world_city_distance.mat', which contains the distance between pairs of cities worldwide. How many dimensions are needed to capture the manifold?
(c) DISCUSS YOUR RESULTS.

Figure 4 and 5 present the 2D and 3D scatter plots representing the learned manifolds for US cities and world cities, respectively. It is interesting to note that for the US cities, just two dimensions are enough to properly represent the distances, and this makes perfect sense since the US territory surface on the globe can be approximated by a plane with little to no distortion. The same is not true for the world cities, where the cities are spread all over the globe surface, and thus can't be properly represented on a plane without distortion. This fact is exactly what leads to the different map projections used to represent the earth. So, for the world cities, 3 dimensions are necessary to properly represent the distances.

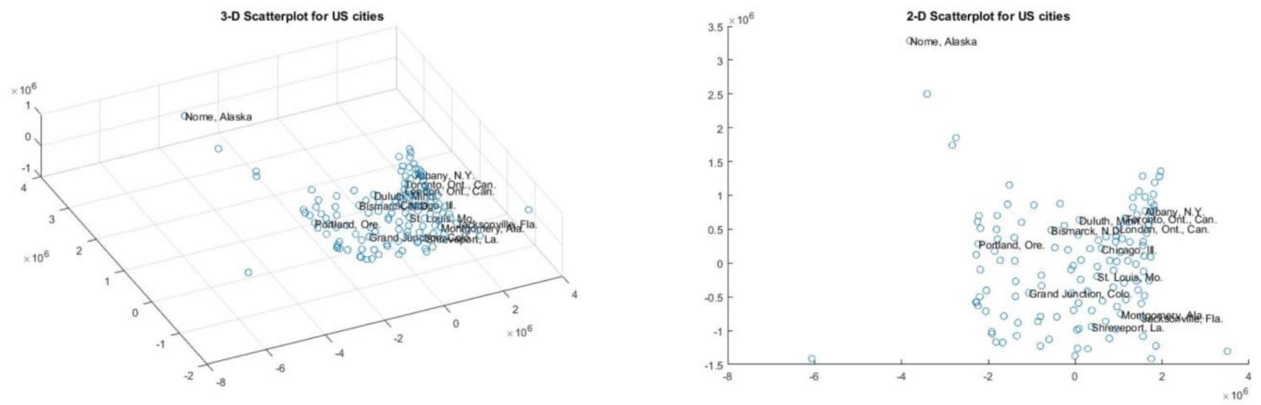

Figure 4. 2D and 3D manifold scatter plots for the US cities.

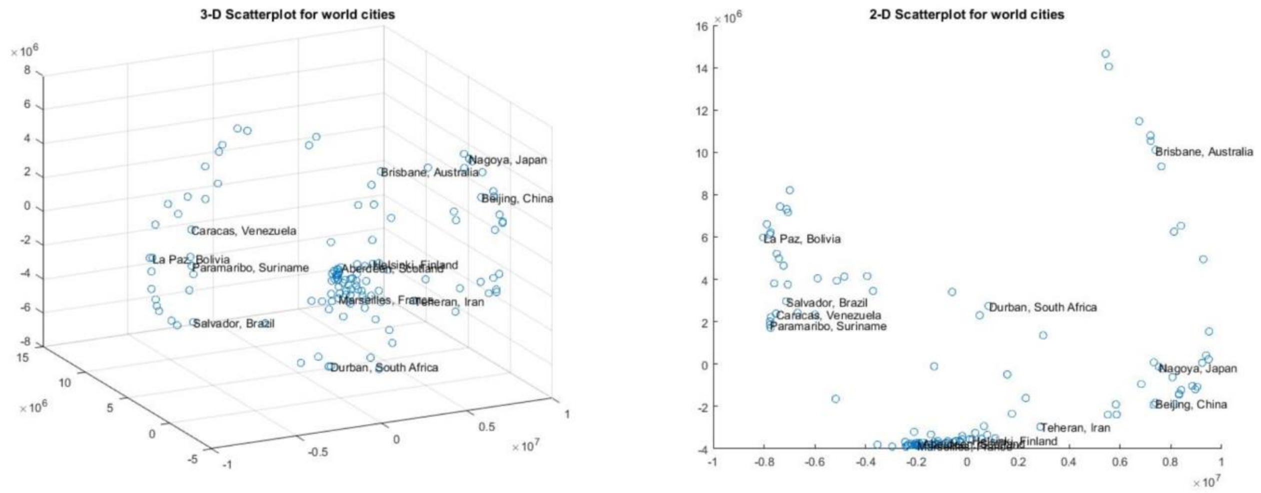

Figure 5. 2D and 3D manifold scatter plots for the world cities.

Problem 3 (20%)

Download the image 'hw3p3_im.jpg', and perform k-means clustering to vector-quantize pixels according to their RGB color.
(a) Reconstruct the image using the colors in the codebook for values of *k=1,2,…,10*.
(b) Generate a color JPEG image for each of the reconstructed images, and save it with the filename *'c1.jpg', c2.jpg', c3.jpg', … 'hw3p2c10.jpg'*.
(c) Can you explain which codewords emerge from the image as the codebook length increases?
(d) Generate a plot that shows the sum-squared-error (SSE) between the reconstructed image and the original image as a function of *k*, the number of clusters. To do so, repeat part (a) several times and, for each *k*, record the clustering that gives the minimum SSE. Can you make sense of this plot? NOTE: You may sub-sample the image (say, 2:1) in order to speed up computations in this part.
(e) DISCUSS YOUR FINDINGS.

Figure 6 presents each reconstructed image using 1 to 10 clusters as the codebook. It is interesting to note the increasing level of detail captured on each subsequent image. For k=1, the image is simply a green background, due to the high amount of pixels with green shades representing the soccer field in the original picture. For k=2, along with green, white shades are also represented, capturing Real Madrid players (mostly clothes), and the field demarcation lines. One could argue that Barcelona players should be represented instead of the Real Madrid ones (there is 6 Barcelona players against 5 from Real), but there is a higher contrast between white and green in comparison to the dark colors of Barcelona's jersey and the soccer field.

For k=3 Barcelona players are also represented on the image (with a single color jersey), and for k=4 players skin are represented. At this point, the picture contains all the basic structures of the original image (the field and its demarcation lines, each team jersey, and the players' skin) and each subsequent image add a new level of detail to it, like for k=5 where Barcelona's jersey becomes bicolor. At the final stage k=10, we have a fairly accurate reproduction of the original image, even partially including some jersey details like sponsors and club emblems along with lightning shadows and multiple shades of green to represent the grass.

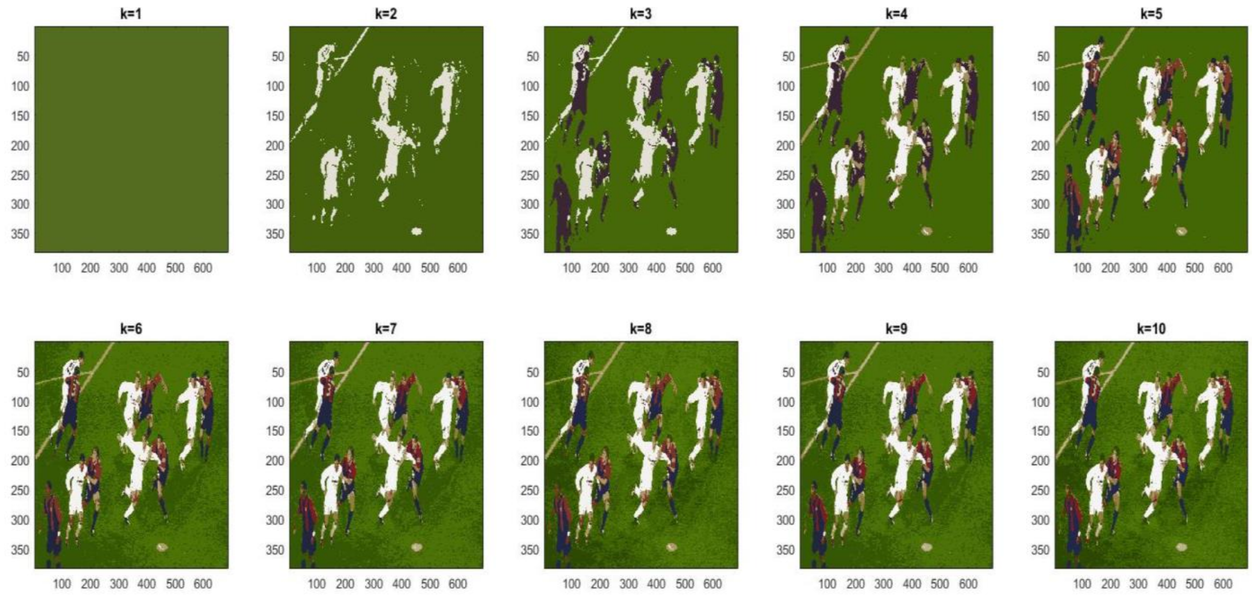

Figure 6. Reconstructed images using codebooks with size 1 to 10.

Figure 7 presents the sum squared error for each codebook length used. As explained before, using a single codeword to reconstruct the image yields to a really high error, since the only thing from the original image that is being represented is the soccer field grass. Adding another codeword to the image drastically improve the results, since now Real jerseys and field demarcation lines are also being represented. Analogous to what was discussed in the previous paragraph, after k=4, each new color represented have a much lower impact on the computed SSE, indicating that all the main structures of the original image are being represented on the first four colors.

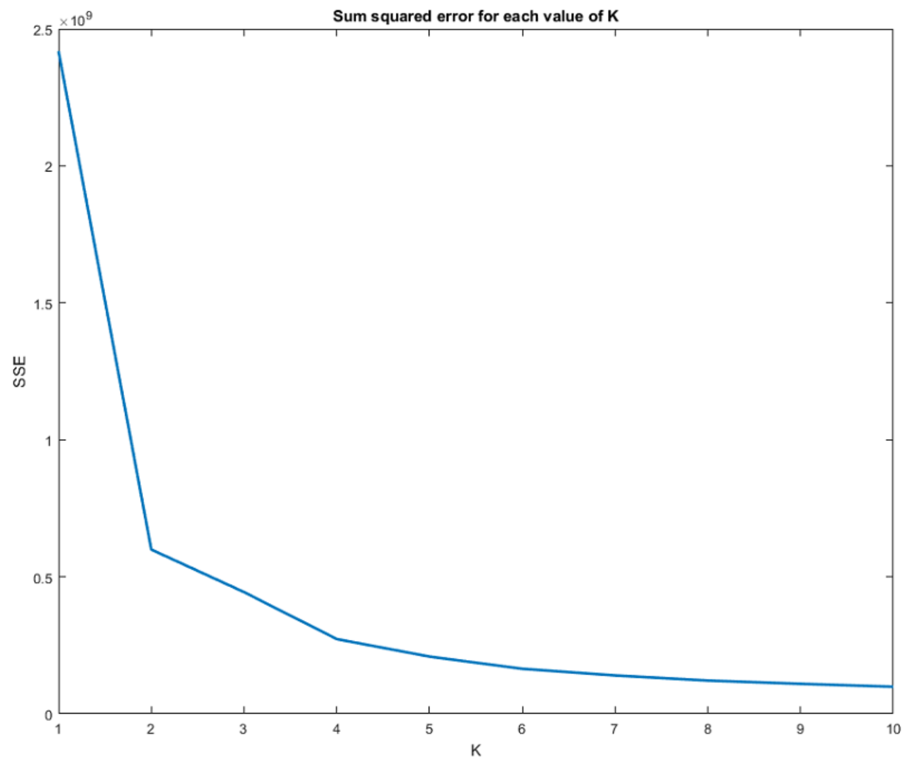

Figure 7. Sum-squared-error using different codebook lengths to reconstruct the original image.

Problem 4 (40%)

You are given a dataset containing nutritional content for a large number of food products from the 10 different groups shown in Table 1. Table 2 shows the nutritional information contained on each of the 46 features in the dataset. When you load the data, you will notice that a significant number of entries have a value of -1; this value indicates that the corresponding nutritional feature was not available for the sample in question.
As in Homework 2, you are given the following datasets:

- Dataset '*hw3p4_train.mat*' containing the following matrices:
  x1: training set (row vectors)
  clab1: training set labels
- Dataset '*hw3p4_test.mat*' contains the following matrices:
  x2: test set (row vectors)
  clab2: test set labels

You are to:
(a) Perform feature subset selection to determine a reduced number of features (less than or equal to 10) that provide good discriminatory information. You may employ any feature subset selection technique, and any of the classifiers that you have developed in previous homework assignments.
(b) To evaluate your final classifier, I will use a similar approach as in Homework #2. Prepare a MATLAB program called '*hw3p4.m*' that will load '*hw3p4_test.mat*' and classify each of the examples in the dataset x2. Once you submit your code through email, I will run your '*hw3p4.m*' program with a separate 'hw3p4_test.mat' file containing my own test data. Your grade will be based on the performance of your classifier on my test data, which will contain a very large

number of examples so I can approximate the true error rate. All datasets will obviously be generated from the same distribution.
(c) DESCRIBE YOUR APPROACH AND DISCUSS YOUR RESULTS. What search technique(s) did you try? What objective function(s) did you try? What classifier(s) did you employ? Which features were selected? How did you determine how many features to settle for? …

For problem 4 I had to perform feature subset selection on the training data and use the selected subset with one of the previously implemented classifiers to solve the 10 class classification problem. In order to evaluate a feature subset, there are two general approaches, filters and wrappers. To better estimate the performance of the feature subset, I used the wrapper approach, directly evaluating each feature subset on the particular classifier that will be used to solve the problem.

Since I was using the wrapper approach, the first step was to decide on a classifier to use. For that task a simple experiment was conducted. I set the feature subset as the first 10 features of each sample on the training set and performed a two-way split cross-validation on both KNN (k=1) and Quadratic classifiers. KNN performed much better, indicating that the data probably is not Gaussian. After deciding on the classifier, I made a similar experiment to define the neighborhood size for KNN, and after the experiments I set k=5.

With the classifier set it was performed a Plus-2-minus-1 feature subset selection. Each feature subset was evaluated using the same cross-validation approach of the last step. I recorded the accuracies for each experimented subset and settled for the one with the highest accuracy, which was the full 10 features subset obtained at the end of the process. One important detail of the dataset is that there are missing features for some entries, so, one last experiment was performed to assess the impact of this on the classifier. I repeated the whole Plus-2-minus-1 feature selection process, but this time replacing the missing features with the average of that feature on the dataset. The results for both experiments were very similar (less than 1% difference), so I decided to keep the dataset intact.

Table 1 presents the final 10 features subset selected. This feature subset obtained an average accuracy of %77.7 on the training set cross-validation and a very similar %76.5 on the never seen test set.

Feature index	**Feature name**
1	Water (g/100 g)
6	Carbohydrate (g/100 g)
8	Total sugars (g/100 g)
10	Iron (mg/100 g)
18	Selenium (µg/100 g)
25	Total Folate (µg/100 g)
27	Food Folate (µg/100 g)
29	Vitamin B12 (µg/100 g)
32	Retinol (µg/100 g)
33	Vitamin E (alpha-tocopherol) (mg/100 g)