Mohammad Shahriar Hooshmand
Ph.D. Candidate
Department of Materials Science and Engineering
The Ohio State University

IMPORTANT: In this assignment, since we are dealing with the error/accuracy, I want to point that mean and standard deviations are reported as percentage. Also, I printed accuracy which is more conceivable for the reader/grader. It's obvious but worth mentioning that the error =100-accuracy in percentage.

a) Step 2:

$$P(v_j|C=c)=\frac{P(v_j and\ C=c)}{P(C=c)}$$

Here, we simply run the kmeans algorithm, having 0-6 (7 classes) and k=10 clusters, then we count the number of data points assigned to j cluster having the label c and then divide by the total number of data labeled as C=c. This lead us to a $j \times c$ probability table. The results for a particular run is as follows:

```
running k-means
+------+--------------------+--------------------+--------------------+--------------------+--------------------+--------------------+--------------------+
| V/C  |         C0         |         C1         |         C2         |         C3         |         C4         |         C5         |         C6         |
+------+--------------------+--------------------+--------------------+--------------------+--------------------+--------------------+--------------------+
| V:1  |        0.0         |        0.0         |        0.0         | 0.02531645569620253 |        0.0         |        0.0         | 0.9827586206896551 |
| V:2  | 0.47368421052631576 | 0.028169014084507043 |        0.0         |        0.0         |       0.0625       |        0.0         |        0.0         |
| V:3  |        0.0         |        0.0         |        0.0         | 0.9746835443037974 |        0.0         |        0.0         | 0.017241379310344827 |
| V:4  | 0.013157894736842105 |  0.7183098591549296  |        0.0         |        0.0         |       0.3125       | 0.014925373134328358 |        0.0         |
| V:5  |        0.0         |        0.0         |        0.0         |        0.0         |        0.0         | 0.29850746268656714 |        0.0         |
| V:6  |        0.0         |        0.0         | 0.9764705882352941 |        0.0         |        0.0         |        0.0         |        0.0         |
| V:7  |        0.0         | 0.22535211267605634 |        0.0         |        0.0         |       0.625        |        0.0         |        0.0         |
| V:8  |        0.0         | 0.028169014084507043 |        0.0         |        0.0         |        0.0         | 0.3582089552238806 |        0.0         |
| V:9  |  0.5131578947368421  |        0.0         | 0.023529411764705882 |        0.0         |        0.0         |        0.0         |        0.0         |
| V:10 |        0.0         |        0.0         |        0.0         |        0.0         |        0.0         | 0.3283582089552239 |        0.0         |
+------+--------------------+--------------------+--------------------+--------------------+--------------------+--------------------+--------------------+
```

b) Step 3:

Now that we have the probability table, we can calculate $P(C=c)$ by counting the numbers of classes assigned as c and divide by the total number of labels. Then using Bayes rule, $P(C|V)=\alpha P(V|C)P(C)$. Training for this problem is done in the step 2 by generating CPTs. Then we first classify the unseen test data based on the CPT in previous step and calculate the accuracy% (or similarly error=100-accuracy) for 10 iterations. The mean and standard deviations for these 10 runs are reported as follows:

```
Step 3 Begins:
Accuracy results over 10 iterations: [87.5, 87.5, 90.83333333333333,
84.16666666666667, 89.16666666666667, 87.5, 86.66666666666667,
88.33333333333333, 88.33333333333333, 83.33333333333334]

Mean = 87.33333333333334 pct and STD= 2.1015867021530785
```

c) Step 4:

Email: hooshmand.1@osu.edu
Website: http://u.osu.edu/hooshmand.1

Mohammad Shahriar Hooshmand
Ph.D. Candidate
Department of Materials Science and Engineering
The Ohio State University

Here we change the number of clusters to see what would be the impact on the accuracy. As the results show, very few number of clusters lead to a very low accuracy (large error). As the number of clusters increases, it is expected that data could be classified in a better trend as there are more center means available for point assignment until it reaches a plateau. It is also worth mentioning depending on the number of data points, number of clusters shouldn't exceed a limit otherwise the error would be expected to increase. So, number of cluster optimization is an important step in kmean clustering procedure.

```
Step 4 Begins:
K= 2 Mean = 34.416666666666664 pct and STD= 1.1814539065631555
K= 5 Mean = 72.66666666666666 pct and STD= 3.2659863237109064
K= 6 Mean = 82.16666666666666 pct and STD= 6.091888960832354
K= 8 Mean = 88.33333333333334 pct and STD= 2.2047927592204917
K= 12 Mean = 88.41666666666666 pct and STD= 1.1456439237389593
K= 15 Mean = 87.0 pct and STD= 2.793842435706701
K= 20 Mean = 88.24999999999999 pct and STD= 1.0172129679778077
K= 50 Mean = 89.41666666666666 pct and STD= 1.5388487760516154
```

d) Step 5:

In this part, the data generator script has been modified by imposing a for loop to study the effect of different gaussian widths (variances) of [3.2,4,6,8,10] on the classification accuracy. The results show that as the variance is increased, generated data would be more spread over the mean. Since the kmeans algorithm we are studying here is Euclidian based algorithm, as the distance between data points are increased, there would be a higher chance that algorithm misclassifies the points. Also, since kmeans is based on hard assignment of points, this adds to the limitation for cases dealing with spread data. So, this matches with my experiment that as variance increases, accuracy (error) would be reduced (increased).

```
Step 5 Begins:
Gauss = 3.2 Mean = 58.0 pct and STD= 1.1303883305208784
Gauss = 4 Mean = 37.83333333333333 pct and STD= 2.1147629234082546
Gauss = 6 Mean = 44.41666666666667 pct and STD= 1.8276426832884414
Gauss = 8 Mean = 33.25 pct and STD= 2.873489941594445
Gauss = 10 Mean = 37.666666666666664 pct and STD= 1.7795130420052188
```

Email: hooshmand.1@osu.edu
Website: http://u.osu.edu/hooshmand.1

Mohammad Shahriar Hooshmand
Ph.D. Candidate
Department of Materials Science and Engineering
The Ohio State University

e) Bonus 1:

To change the gaussian width in different dimensions, instead of single value, I imposed a list of variances which will be multiplied (elemental not matrix multiplication) with the random distribution of points to get train points (the corresponding edits are done in "generate_data_Bon1.py". This would cause more spread of data over the space rather than having a symmetric gaussian over all the dimensions. With the same analogy explained in step 5, kmeans is expected to result in lower accuracy when data is more spread. This is due to the Euclidian based criterion which outperforms as the distance between the data is enhanced. Here is the resulting accuracy (100-error) over 10 iterations for the case with asymmetric gaussian:

```
Bonus 1 Begins:
Mean = 52.91666666666667 pct and STD= 2.5617376914899
```

d) Bonus 2:

The modified code for this part is located in "generate_data_Bon2.py". Here a for loop is imposed to generate data in higher dimensions. As we go to higher dimensions, the means of generated data are more separated from each other. As a result, the gaussian generator data generated around each mean are also more clustered respectively (it can be also visualized by looking at the figures). This means that it would be easier for kmean classifier to classify the better initially separated data at higher dimensions compared to the lower dimensions. This analogy is supported by the results of my experiment as follows. As we see, upon going to the higher dimensions, the accuracy of classification is improved as well.

```
Bonus 2 Begins:
Dim = 2 Mean = 89.0 pct and STD= 1.105541596785135
Dim = 3 Mean = 100.0 pct and STD= 0.0
Dim = 4 Mean = 99.83333333333334 pct and STD= 0.5000000000000014
Dim = 5 Mean = 100.0 pct and STD= 0.0
```

Email: hooshmand.1@osu.edu
Website: http://u.osu.edu/hooshmand.1