I pledge my honor that I have abdied by the Stevens Honors System. *Kristin Kim*

1. After running the k-means clustering (Links to an external site.) (Read in the data set from https://raw.githubusercontent.com/tirthajyoti/Machine-Learning-with-Python/master/Datasets/College_Data (Links to an external site.)), include a snapshot of the confusion matrix. Which data cleaning step was performed on the data?

Read in the data by drive.mount() method,

```
from google.colab import drive
drive.mount('/content/drive')

df = pd.read_csv('/content/drive/MyDrive/Colab Notebooks/College_Data.txt',
```

Snapshot of the confusion matrix:

```
[[138  74]
 [531  34]]
              precision    recall  f1-score   support

           0       0.21      0.65      0.31       212
           1       0.31      0.06      0.10       565

    accuracy                           0.22       777
   macro avg       0.26      0.36      0.21       777
weighted avg       0.29      0.22      0.16       777
```

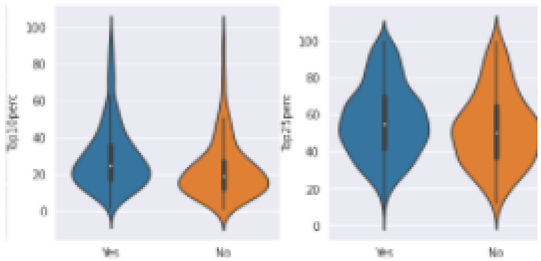
Data cleaning:

- For the data row with a graduation percentage more than 100%, 118%, set its rate to 100 so that the value is reasonable. Then to verify, confirm with df[df['Grad.Rate'] > 100] and histogram visualization to make sure the data has been cleaned.

2. Append the following code at the end and include a snapshot of the results. Which features seem to be the least distinct between private and public universities?

Based on the interpretation of violin plots,

These two attributes,



,

Top10perc Pct. new students from top 10% of H.S. class
Top25perc Pct. new students from top 25% of H.S. class

will be the least distinct attribute to private vs. public consideration, because mean/median, interquartile ranges, and the full distribution are very similar and almost identical for both private and public sectors, while other attributes show some distinctions.

3. Drop the non-discriminative attributes (at least three) you identified in the previous question using the following code (replace attribute1 with attribute title) and run the K Means Cluster Creation. Note that, you need to copy cluster center analysis and confusion matrix code blocks after this code. How did the evaluation performance change?

The attributes removed are:

- Top10perc Pct. new students from top 10% of H.S. class
- Top25perc Pct. new students from top 25% of H.S. class
- Terminal Pct. of faculty with terminal degree

Previously, the confusion matrix contains very high false negative values as below, along with a low accuracy

```
[[138  74]
 [531  34]]
              precision    recall  f1-score   support

           0       0.21      0.65      0.31       212
           1       0.31      0.06      0.10       565

    accuracy                           0.22       777
   macro avg       0.26      0.36      0.21       777
weighted avg       0.29      0.22      0.16       777
```

	K-means cluster centroid-distance	Mean of corresponding entity (private)	Mean of corresponding entity (public)
Apps	8549.904210	1977.929204	5729.919811
Accept	5263.732229	1305.702655	3919.287736
Enroll	2078.677379	456.945133	1640.872642
Top10perc	16.181324	29.330973	22.834906
Top25perc	16.732852	56.957522	52.702830
F.Undergrad	10873.386605	1872.168142	8571.004717
P.Undergrad	1869.402217	433.966372	1978.188679
Outstate	323.467406	11801.693805	6813.410377
Room.Board	332.107499	4586.143363	3748.240566
Books	53.230900	547.506195	554.377358
Personal	433.867381	1214.440708	1676.981132
PhD	15.955697	71.093805	76.834906
Terminal	13.508221	78.534513	82.816038
S.F.Ratio	-0.071923	12.945487	17.139151
perc.alumni	-3.100814	25.890265	14.358491
Expend	5238.453662	10486.353982	7458.316038
Grad.Rate	2.499917	68.966372	56.042453
Cluster	-0.478907	1.000000	0.000000

However, after removal, the precision and recall scores improved by a lot as expected. There are clearly less number of type I and type 2 errors after removal, therefore a high accuracy in predicit the correct values.

```
[[ 74 138]
 [ 34 531]]
              precision    recall  f1-score   support

           0       0.69      0.35      0.46       212
           1       0.79      0.94      0.86       565

    accuracy                           0.78       777
   macro avg       0.74      0.64      0.66       777
weighted avg       0.76      0.78      0.75       777
```

	K-means cluster centroid-distance	Mean of corresponding entity (private)	Mean of corresponding entity (public)
Apps	8549.904210	1977.929204	5729.919811
Accept	5263.732229	1305.702655	3919.287736
Enroll	2078.677379	456.945133	1640.872642
F.Undergrad	10873.386605	1872.168142	8571.004717
P.Undergrad	1869.402217	433.966372	1978.188679
Outstate	323.467406	11801.693805	6813.410377
Room.Board	332.107499	4586.143363	3748.240566
Books	53.230900	547.506195	554.377358
Personal	433.867381	1214.440708	1676.981132
PhD	15.955697	71.093805	76.834906
S.F.Ratio	-0.071923	12.945487	17.139151
perc.alumni	-3.100814	25.890265	14.358491
Expend	5238.453662	10486.353982	7458.316038
Grad.Rate	2.499917	68.966372	56.042453
Cluster	-0.478907	1.000000	0.000000

4. Normalize the data by transforming the values using the StandardScaler. How did the evaluation performance change? Briefly explain.

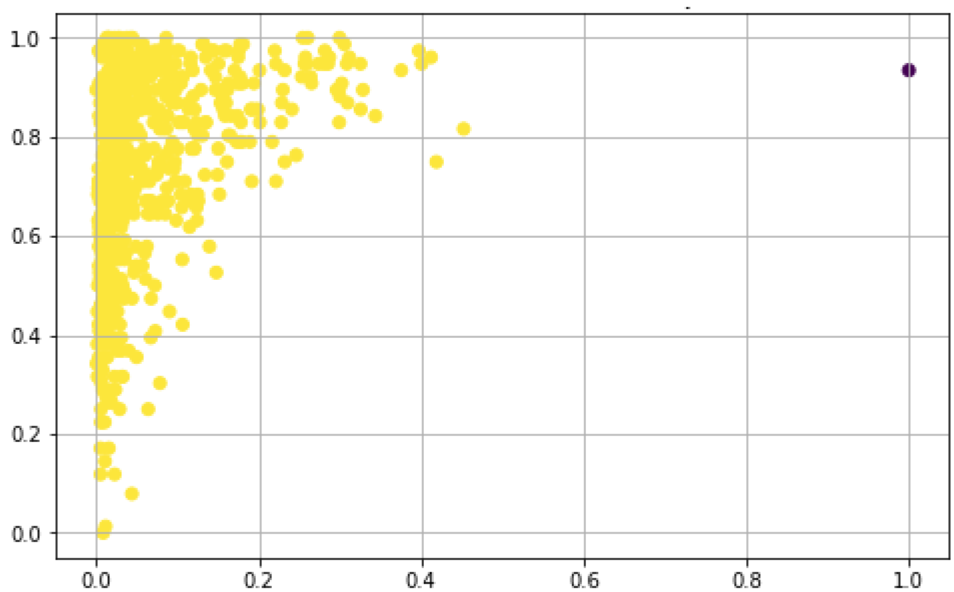
After normalization, the scores improved by a lot as shown below, and this is expected because the standard scaler removes the mean and scales each variable to unit variance.A Also, a heavy change in distance can be observed since the values have been normalized.

	precision	recall	f1-score	support
0	0.94	0.94	0.94	212
1	0.98	0.98	0.98	565
accuracy			0.97	777
macro avg	0.96	0.96	0.96	777
weighted avg	0.97	0.97	0.97	777

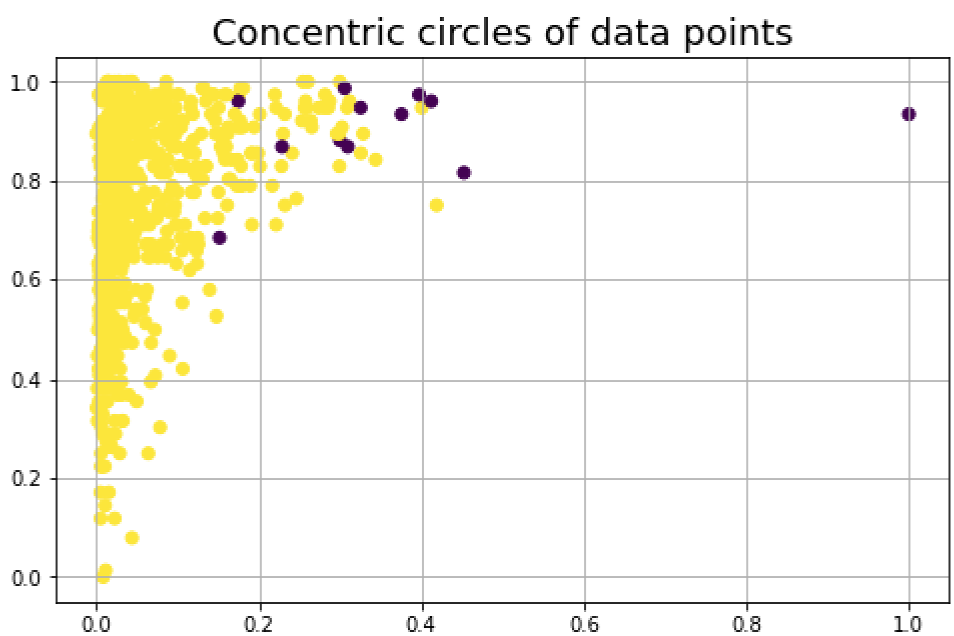
	K-means cluster centroid-distance	Mean of corresponding entity (private)	Mean of corresponding entity (public)
Apps	-1.111013	1977.929204	5729.919811
Accept	-1.224416	1305.702655	3919.287736
Enroll	-1.439052	456.945133	1640.872642
Top10perc	0.360647	29.330973	22.834906
Top25perc	0.173854	56.957522	52.702830
F.Undergrad	-1.522933	1872.168142	8571.004717
P.Undergrad	-1.140715	433.966372	1978.188679
Outstate	1.211212	11801.693805	6813.410377
Room.Board	0.737673	4586.143363	3748.240566
Books	-0.085545	547.506195	554.377358
Personal	-0.750798	1214.440708	1676.981132
PhD	-0.363658	71.093805	76.834906
Terminal	-0.300017	78.534513	82.816038
S.F.Ratio	-1.122505	12.945487	17.139151
perc.alumni	0.948085	25.890265	14.358491
Expend	0.542997	10486.353982	7458.316038
Grad.Rate	0.803475	68.966372	56.042453
Cluster	2.061342	1.000000	0.000000

6. After running the DBScan Clustering (Links to an external site.) append the following code block. Then modify the eps and min_samples parameters to obtain at least two clusters (note that a label of -1 indicates noise). Append the results and briefly explain.

While considering min samples of 100, epsilon of 0.9 gives no noise for ‘App’ and ‘Terminal’ columns

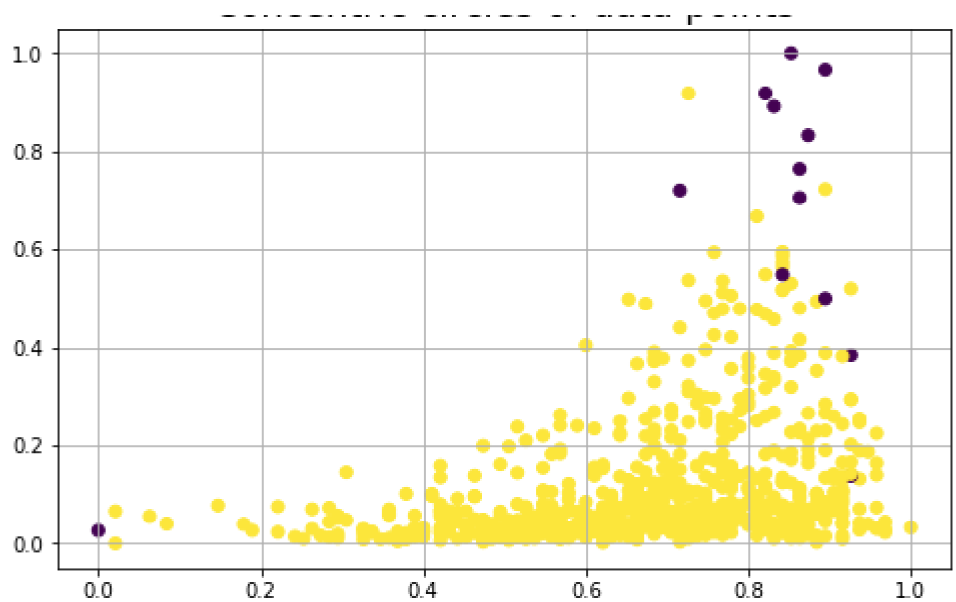


When I chose epsilon of 0.8, some noise was detected, and for epsilon of 0.7,



It was pretty clear that separation of clusters is not too explicit. However, here, I have to mention that the data itself doesn’t well represent any clusters, since the majority of data points are located on the left.

So, I've plotted for 'PhD' and 'Enroll' columns.

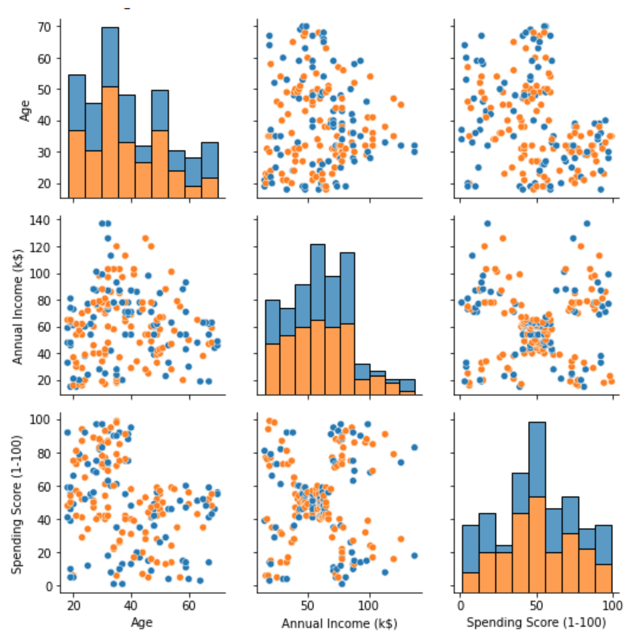


7. Use the following code block to find classification performance. Note that you may modify the converter function to decide which clusters are assumed to be private universities. Append the results and briefly explain.

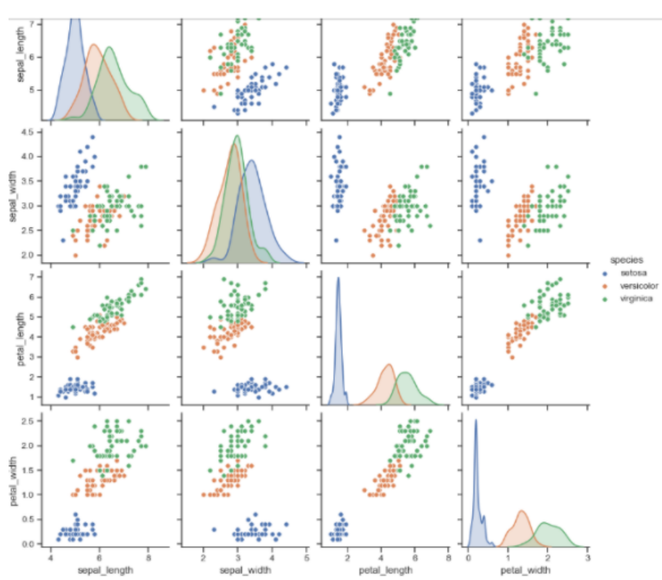
	precision	recall	f1-score	support
No	1.00	0.00	0.01	212
Yes	0.73	1.00	0.84	565
accuracy			0.73	777
macro avg	0.86	0.50	0.43	777
weighted avg	0.80	0.73	0.62	777

Here, based on the performanc reports, specifically from the precision and recall score, we know that it returns very few results, but most of its predicted labels are correct when comparing since recall is defined as the number of true positive/ (number of true positives + number of false negativies) I tried to modify the converter function, however, received the error for getting the performance report

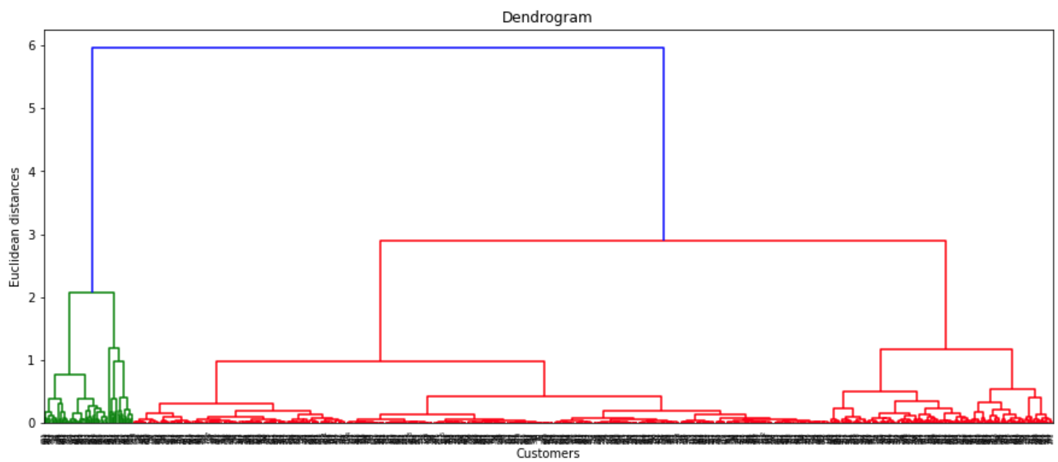
8. After running the Hierarchical Clustering (Links to an external site.) (Read in the data set from https://raw.githubusercontent.com/tirthajyoti/Machine-Learning-with-Python/master/Datasets/Mall_Customers.csv (Links to an external site.)) append the following code block to analyze attribute pairs that produce clear clusters. Comment on whether the identified pairs are reasonable.



I don’t think these identified paris are reasonable since they are pretty much mixed in all the scattors plots unlike this

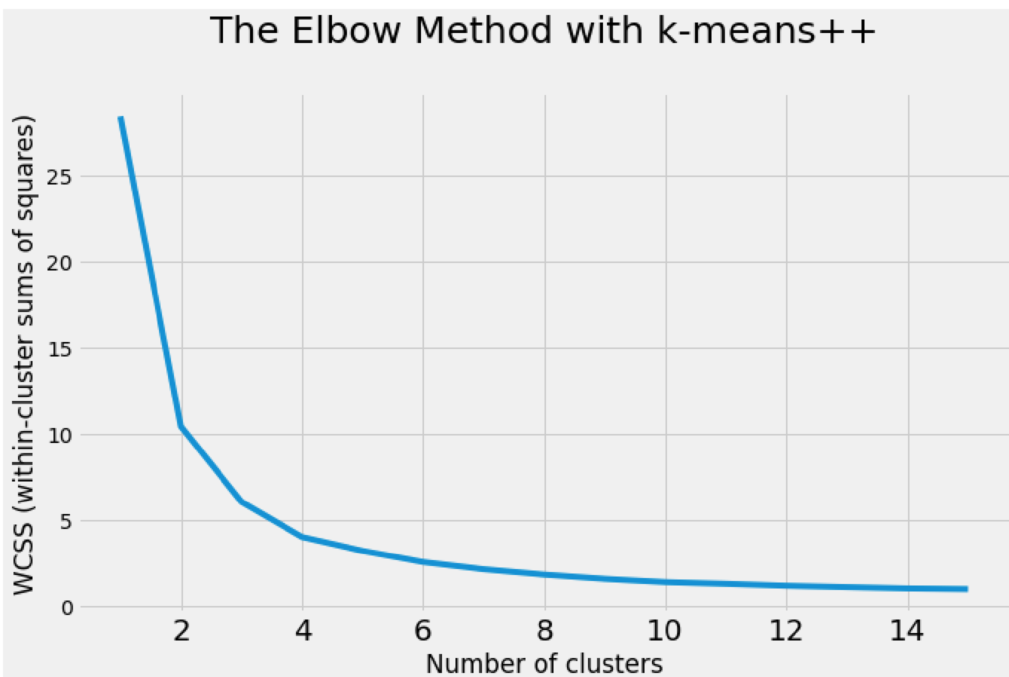


9. Let’s add age attribute and scale the values using the following code. Include a snapshot of the dendogram.



Add a second code block to plot errors and comment on the optimal number of clusters in this case.

The optimal number of clusters woulb be 4 because it seems there is no more drastic decrease in the value after 4.



10. Let’s analyze university data using the following code.

Clusters are generated using only the selected rows mentioned, and the performance is extremely good as shown on the reports below, not much of type I error and type II errors have been detected, with a good precision/ recall scores.

```
[[554  11]
 [ 87 125]]
              precision    recall  f1-score   support

          No       0.92      0.59      0.72       212
         Yes       0.86      0.98      0.92       565

    accuracy                           0.87       777
   macro avg       0.89      0.79      0.82       777
weighted avg       0.88      0.87      0.86       777
```

11. [2 bonus points] Show how you can identify the best set of attributes to classify private universities. Provide the classification performance.

To find the best set attributes, I will write a function that can iterate through different combinations of the attributes, and check the accuracy of it based on the confusion matrix and classification, then it will return two attributes that result with the highest accuracy.