1. **Purity function:**

I wrote my purity function with outliers in mind and from the help of a source listed below.  
  
Text

Description automatically generated

1. After running Kmeans 20 times for k = 9 and k = 13 on the Complex9\_gn8 dataset the visualization I obtained for the two clusterings is shown below:

**K=9**Chart, scatter chart

Description automatically generated

**K=13**Chart, scatter chart

Description automatically generated

**WCSS  
Chart, line chart

Description automatically generated**

To get the purity of my two clusterings I used the same purity function from part (a) and entered the clustering information and compared with the original clusters given. The purity I got was: for k = 9 -> .71 and for k = 13 -> .68. I believe that these purity scores are relatively good but can be improved with a little bit of tweaking of the k value we are running kmeans with. It is not exactly a bad purity score as we are predicting the natural clusters of the original Complex 9 dataset correctly roughly 71% of the time but there is room for improvement.

1. For this part I first worked to obtain the ZSHUT dataset from the original Shuttle22 dataset. To do so we were given the instructions to zscore all 9 attributes from the Shuttle22 dataset which would result in the ZSHUT dataset that we will use for our computations in this part.  
   Table

   Description automatically generated  
     
     
   Next using my Kmeans function, I ran for k=3 on the ZSHUT dataset. Since this time for kmeans we are comparing nine attributes using a clustering diagram like the ones from part a would result in useless information as we would need a three-dimensional graph to interpret this information, doing the same approach resulted in this weird graph below:  
   Chart, scatter chart

   Description automatically generated  
   As you can see this is useless as we have so many points overlapping our clustering would have no good visualization. Rather than the previous approach mentioned, for this part it was much more efficient to show our nine attributes through boxplots for the whole dataset so we can have a way of interpreting our data.  
     
     
     
     
     
     
     
     
     
     
     
     
     
     
     
     
     
     
     
     
     
     
     
     
     
     
   **Original Whole Dataset**Chart, box and whisker chart

   Description automatically generated **Cluster 0**Chart, box and whisker chart

   Description automatically generated **Cluster 1**Chart, box and whisker chart

   Description automatically generated **Cluster 2**Chart, box and whisker chart

   Description automatically generated  
   As for the purity of this clustering I got was .87 which is very high. I am pretty satisfied with this purity score since it is high enough that it is reliable. When looking at the four graphs with 9 boxplots each and evaluating them the best way to do so is by comparing centroids. Centroids are the mean/ middle of the cluster for each attribute. The general theme I am seeing when comparing the original boxplots with the clustered ones is that the whiskers drastically decreased in length in attributes 1, 3, and 9. The centroids of these clusters seems to be lower than the boxplots of the original also. For attribute 1: the attribute is positive for the cluster 1 while for clusters 2,3, and the whole dataset it is negative; for attribute 2: it seems to be consistent throughout the clusters and the whole dataset; for attribute 3: is 0 for the cluster 1 while for clusters 2,3, and the whole dataset it is negative; for attribute 4: it also seems to be consistent throughout our diagrams above; for attribute 5: It is positive for all clusters and the whole dataset; for attribute 6: it is consistently on 0; for attribute 7: it is positive in all scenarios other than cluster 1; for attribute 8: it is positive in clusters 1,2 and negative in the whole dataset and cluster 3, this attribute also shows the most change throughout the diagrams; for attribute 9: it is only positive in cluster 1 while clusters 2,3 and the whole dataset’s diagrams all show a negative.
2. For this subtask I obtained a DBSCAN clustering for the ZSHUT dataset. The parameters I had to follow were: between 2 and 15 clusters and must have less than 20% outliers. This task took a while to run on my pc, but I was able to run DBSCAN on my computer to find a valid eps and minpoints value to input for the given parameters. I concluded that the best one I could find was eps = .66 and minpoints = 10. These values were able to compute a purity score of .84 for my clustering through DBSCAN. I obtained an outlier percentage of only 4% and the clusters in total in this run and the cluster information is shown below:  
     
   **Function to return clusters\_df**Text

   Description automatically generated **Function to return the cluster information**Text

   Description automatically generated **Cluster information:**  
   Text

   Description automatically generated  
   \*-1 is the outliers\*
3. To develop my search procedure to look for the best clustering by changes in minpoints and eps values I reused many of the previous functions I defined for the parts above with some tweaks. The two parameters for this search were that the number of clusters must be between 2 and 15 and the number of outliers should be 10% or less as well. I ended up with eps = .46 and minpoints = 100 which resulted in a purity score of .93 which is the highest I was able to compute using my search procedure. The outlier percentage also was within the parameters given as it was only 6%. The cluster information for my best clustering is shown below as well as the search procedure I came up with:  
     
   **Cluster information, outlier percentage: 6%**  
     
   **Search Procedure**Text

   Description automatically generated **How it works:**My search procedure has the following steps:  
   1) Run dbscanplot1 with dataset,eps,minpoints to get the number of clusters and the outlier percentage  
   2) Once satisfied with the clustering information, run dbscanplot2 with same parameters to get clusters\_df which holds the labels for the clusters  
   3) add clusters\_df to original df and then filter out outliers by removing rows that have the cluster label of ‘-1’ which indicates that row is an outlier  
   4) set dataframe[9] to y\_true and dataframe[10] to y\_pred and run purity\_score with those two as parameters to get purity score of your clustering
4. **Extra Credit:**For this extra credit I ran my DBSCAN search procedure on the Shuttle dataset and the best clustering results I got are shown below. I ended up using eps = 10 and minpoints = 10 which resulted in a purity score of .85 which is pretty high. I found that it took a way larger eps for this dataset since the values are not zscored so we need a larger radius to evaluate and cluster the datapoints. I played around with the eps and minpoints for a while then finally computed something that was acceptable with the clusters shown below:  
     
     
     
   Text

   Description automatically generated  
     
   A picture containing calendar

   Description automatically generated  
     
   Text

   Description automatically generated