Problem Set - 5

Please read all of the guidelines carefully before submitting the problem set. (Unless specified) each question is ***20 points*** and there are ***100 points*** in total.

Due date: Sunday, February 27, 11:59 PM. Late submissions will be accepted with a penalty! (10% reduction per day – no submissions accepted two days after the deadline.)

Grading

Your answers will be evaluated based on the following criteria:

- **Completeness:** Your answers will be checked for completeness. Specifically, for a question that requires several steps of thinking / writing / coding, we expect you to complete the full range of steps to answer the question. The range of steps that needs to be completed will be determined by the course material and the specific nature of the question.
- **Correctness:** Your answers will be checked for correctness. For your answer to be correct, you need to have the correct answer, correct implementation, and the correct result. 'Correct' means that you follow the steps suggested by the assignment and your instructor and obtain the expected result without making any theoretical / mathematical / coding mistakes. 'Correctness' is not a binary term, there may be varying degrees of correctness; and, your grade will be evaluated based on how different your answer is from the expected result.
- **Format:** An indispensable part of every assignment is the format. To make sure that your assignment can be read and processed easily, we expect you to follow the guidelines set by the instructor. These guidelines may include specific requirements about text-based answers, code files, and datasets.
- **Academic Honesty:** We assignment that your submission fulfills the academic honesty expectations set by the instructor and put forward in the syllabus. Specifically, when expected, you need to produce work within the limits defined in the syllabus – some of the assignments may require you to work individually, and some others in a team. For more information, please read the syllabus.

Guidelines – Before You Start

1) **You should complete the problem set on your own.** Discussing ideas is fine; but, sharing answers and sharing code will be considered as plagiarism.
2) You will be using the `Python` programming language. You need to write your codes in an empty **.ipynb** file.
3) Make sure that you provide many comments to describe your code and the variables that you created.
4) Please use **LaTeX** or **MS Word** to submit your written responses (hand-written responses will not be graded).
5) For some of the coding exercises, you may need to do a little bit of "**Googling**" or review the documentation.

Deliverables:

1) The code of the problem set in **.ipynb** format (one file)
2) Short answers written with ***LaTeX*** or **MS Word** and exported in **.pdf** format (one file)

Questions

1) **[20 points]** Download the dataset called *'country_information.xlsx'* that can be found under the 'Data' tab on *BlackBoard*. Do the following:
   a. **[10 points]** Provide a summary of what the dataset is about (around 100 words) by checking the variable names.
   b. **[10 points]** Excluding the '**country**' column, apply 0-1 normalization on the numeric columns. Save the resulting dataset as:
   'country_information_normalized.xlsx' [Note: Do not forget to add the '***country***' column to the normalized dataset. For normalization, you can use a package.]

2) **[20 points]** Code the ***kmeans++*** algorithm from scratch. For more information about the individual steps of the algorithm, please check here:
   https://en.wikipedia.org/wiki/K-means%2B%2B.

   As input, your algorithm should take a `numpy` matrix or a `pandas` dataframe and a ***k*** value that denotes the expected number of clusters. The output needs to be the labels associated with feature vectors coming from your dataset.

   Note: You are welcome to use pre-packaged algorithms to calculate distances and means. If you need to pick a point randomly, please do the following:
   i. Import the `random` package of `Python`.
   ii. Set seed to 265 by running the following line: `random.seed(265)` **[This should be done at the very beginning of your code file, after importing the packages.]**
   iii. Run the following line: `randrange(0,len(name_of_your_dataset),1)`. Use the resulting the number as the index number for the data point that should be randomly picked in different stages of the kmeans++ algorithm.

For the remainder of the analysis, use the *'country_information_normalized.xlsx'* dataset you created in **Q1**.

3) **[20 points]** Now, we will test the code we have written in **Q2** and apply dimension reduction: Specifically, do the following:

   a. [**10 points**]. Set the random seed to *265* again (to (re-)guarantee the same initialization). Set ***k* = *6***. Run your ***kmeans++*** code on the *'country_information_normalized.xlsx'* dataset by excluding the '***country***' column.
   Record the labels. Attach the labels as a new column to your dataset by naming your new variable as ***kmeans_label***.

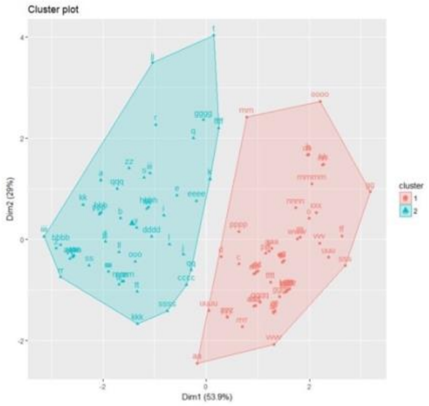
   b. [**10 points**] Excluding the '***country***' and '***kmeans_label***' columns, run dimension reduction (specifically PCA) on your dataset by using sklearn's `PCA` function: https://scikit-learn.org/stable/modules/generated/sklearn.decomposition.PCA.html.

[Note: set `n_components = 2` and `random_state = 265`. Other parameters should be left as 'default'.]. Add the new variables in your dataset as `pca_dim_1` and `pca_dim_2`.

For the next question, use the attached *'visualization_code.py'* file.

4) [**20 points**] Now, let's visualize the results, use the clustering labels to color our data points, and present them in convex hulls. Run the code provided to you in the '***visualization_code.py'*** file. Change the name of the dataset where it says […]. Add the visual to your **.pdf** submission.

   Note: For this exercise, you will need to find and explore the required packages that will need to be imported. The resulting plot should look (somewhat) similar to what is below (but, you will have ***k = 6***).



5) [**20 points**] Interpret the results (in around 300 words) by answering the following:
   a. [**5 points**] Which countries seem to be similar? Why do you think these countries are clustered together?
   b. [**5 points**] If you run the ***kmeans++*** algorithm more than once, do you think the results will change?
   c. [**5 points**] (Subjectively speaking) Do you think this is an accurate clustering of the countries? Would the results change greatly if we had different social/economic variables?
   d. [**5 points**] Do you think PCA may have affected the results at all? In other words, if we had a different number of principle components, would our visual interpretation be different?