IML: Bonus task assignment

1 All hail the clusters!

Considering that students got various grades by the end of the semester, we decided to split the last BONUS assignment in the course into 3 tracks, by the level of difficulty. All of them are dedicated to clustering algorithms in different areas, but the harder tracks require more things to implement from scratch. The tracks have a description of how many points you may achieve for your submission. Note you can choose **one** track only.

The notebooks contain such bold comments:

- **Do not use any additional libraries** - you can use only the modules that are imported before *# Write your code here* comment line. In general, it means we expect you to implement some algorithm or function without ready-made solutions from specialized libraries.
- **You're allowed to use any libraries** - you may use any method or class that you prefer and import any module for that.

Deadline: May 8, 23:59
Submission format: Jupyter notebook to the corresponding task on Moodle
Cheating policy: Plagiarism in the code will result in failing. If you use code from the internet, cite it by adding the source of the code as a comment in the first line of the code cell
Contacts: Telegram @GRoman20; mail o.garaev@innopolis.unirersity
Link to the question Google Sheet: link

2 Easy track [5 points max]

The dataset for clustering in this track - is **credit card history**, and you're asked to group the customers according to this history. You should prepare the data, do a clustering by K-means++, write a function to calculate the optimum number of clusters (elbow method) and evaluate the quality of the clusters.

3 Medium track [15 points max]

In this track you'll use cluster algorithms for **image segmentation**. You'll do it by two algorithms: K-means++ and Fuzzy C-means. The former one is familiar to you, and you're allowed to use its realization from the standard libraries. Fuzzy C-means, a preferable clustering algorithm for the image segmentation, wasn't considered in the lectures and seminars, so it will take some time to understand how it works. Moreover, in this track, you will implement it by yourself.

Additional (yet minor) difficulty of this track is working with images. You're supposed to work heavily with OpenCV and numpy packages.

4 Hard track [20 points max]

In this track you're supposed to work with text data and cluster the **similar articles** into the groups. The main difficulty here is vectorization of the documents with different length and content. You're going to use the TF-IDF statistic to represent every document as a numerical vector with fixed length. To make it harder, we ask you to implement this algorithm from scratch.

Another difficulty in this track is the implementation of DBSCAN algorithm. While you have all the theory from the last lecture, it can take some time to carefully implement the formulas in code and configure the parameters.

In this track you will also get familiar with TruncatedSVD - dimension reduction algorithm for the sparse matrix. We're not asking you to write your own realization of it, but you should answer some theoretical questions to demonstrate your understanding how it works.