K-means

Group 04:

1. Kummer Lorenz, student ID – 00947188
2. Sidak Kevin, student ID – 01249373
3. Higgs Sam, student ID – 11844757
4. Setezhev Grigory, student ID – 11808684

# Introduction

K-means algorithm is used in the solution of the clustering problem. It aims to partition *n* given data points (or observations) into *k* different clusters. This algorithm dates back to 1957 by Stuart Lloyd and 1965 by E. W. Forgy.

# Programming setup

Python (version 3) was used as a programming language for this assignment. It was chosen for it’s ease of use and comprehensive list of available packages which could simplify the development of the program. Package-management system **pip** allowed us to use the following packages:

1. NumPy – library which supports multi-dimensional arrays and matrices and also provides the functions to operate on them. We also used it as a tool to import data from TXT and CSV files.
2. Scikit-learn – machine learning library which gave us the ability to calculate the Normalized Mutual Information score.
3. Matplotlib – plotting library which was used by us to visualize the clustering result.

# Project structure

The project has the following structure:

1. kmeans.py – main class, where the data is read, processed, metrics are calculated, execution time is measured, and the results are presented.
2. setup.py – class which is required for setuptools working.
3. init\_strategies.py – class which describes different techniques for the initial definition of clusters in k-means algorithm.
4. update\_strategies.py – class where different algorithms to update cluster’s centroids are described.
5. mq\_run\_script.py – separate script which is used for MacQueen update strategy.
6. experiment.py – benchmark tool which is used for calculation the average results among 100 runs.

# Initialization techniques

Our team has implemented three different algorithms for initial definition of clusters in k-means algorithm:

1. Random initialization – given *n* data points, we choose random *k* among them and define them as our cluster centroids.
2. Farthest point initialization – first point is selected randomly, second one is chosen with the largest distance to the first one, third point has the largest distance to two previous points and so on, until all *k* clusters are defined.

# Updating techniques

We have developed two algorithms