# Advanced Machine Learning - Assignment 3 Report

**Submitted by:**   
Miryam Horwicz - 313246704, Yaniv Tal - 031431166

## Notebook Overview

Our submitted notebook consists of four sections:

1. **General section** – handles the logistics:
   1. Imports
   2. Mount Google drive
   3. Load data from disk and put in a dataframe etc.
2. **Task part 1 –** implementation of K most similar documents based on NCD distance
3. **Task part 2** – Similar documents finder implementation, partially based on part-1 function

## Part 1 – K Most Similar documents function

We implemented the K most similar instances method using Normalized compression distance (NCD) based on gzip compression.

NCD is defined as:

* x, y - Text documents
* len(g(x)) – length of the output of the gzip compression function applied on document x

Our function gets a reference document record and a corpus of articles, calculates the NCD score between the reference and each document in the corpus, and returns the K most similar items (K items with lowest NCD scores)

## Part 2 – relevant / similar documents finder implementation

### General

In this part, we decided to create a method that gets a text string and finds the most relevant items in the corpus, and use our NCD function as one step within this process.

NCD similarity is lower (More similar) when the documents contain exact identical segments, and is very sensitive to noise – letter case, spelling, word order, phrasing etc. For this reason, it is more suitable for application like Malware search, where entire binary segments can be exact copy of known references, and less suitable for human created texts.

We decided to tackle these issues as following:

1. Text preprocessing – remove as much “noise” as possible from the text while preserving the essence to improve similarity accuracy and reduce writer variance
2. Initial clustering of the article corpus using TF/IDF (“Bag of words” approach) with K-means to identify clusters of articles in the same area
3. Secondary filtering using the NCD within the relevant cluster to extract the most similar samples

In addition, based on the above, we decided to focus only on English articles (Which is the majority of the corpus), since both BOW and NCD will work poorly on multi-lingual corpus.

### Training and Inference flows overview

**The training flow consists of the following stages:**

1. Preprocess the text in the entire article corpus as described in the previous section, store for later use for Inference
2. Train a TF/IDF model on the entire corpus, store for later use for Inference
3. Select number of clusters
4. Train a K-means model on the corpus, store for later use for Inference
5. Cluster the entire corpus using the model, store for later use for Inference

**The Inference (“Search”) flow consists of the following stages:**

1. Get a reference string for similarity search
2. Preprocess it as described above
3. Vectorize it using the saved TF/IDF model
4. Find cluster using the saved K-means model
5. Run the NCD similarity between the reference string and each document in the relevant cluster and return the most similar N results

### Preprocessing step:

We apply the same preprocessing funnel both on the training corpus and on our search string.  
The preprocess flow consists of:

1. Concatenate the Title, Abstract and Full Text parts of each sample into one continuous string (Since we want to take all of them into account when searching similar docs)
2. Changing all text to lower case
3. Remove special characters and excess white space
4. Remove “Stop words” such as “The”, “What”, “a” etc. using the Spacy stop words dictionary
5. Remove word parts that contribute high variance but do not change the essence of the topics in the documents. For that we used the Snowball Stemmer.

**Example for preprocess input / output:**

**Input**: A ‘Multiomic’ Approach of Saliva Metabolomics, Microbiota, and Serum Biomarkers to Assess the Need of Hospitalization in Coronavirus Disease 2019

**Output**: multiom approach saliva metabolom microbiota serum biomark assess need hospit coronavirus diseas 2019

### TF/IDF step

TF/IDF - term frequency – inverse document frequency, is a numerical statistic that is intended to reflect how important a word is to a document in a collection or corpus. It is a “Bag of words” method – meaning it does not take into account word order or context, only the number of appearances in the document and in the corpus. In addition, it gives more weight to words that are more rare, which means they relatively unique to the specific sample.

We chose TF/IDF based clustering as a first step, because we wanted our search to prioritize relevant documents, i.e., documents with as many words which are highly relevant to the search, and not only documents which have exact copies of entire sentences.

Our decision to implement a hierarchical search with TF-IDF based clustering as a first step and NCD search only within the selected cluster was done because we decided to first search using the wider perspective that BOW approach provides, and then sort the results within that cluster using NCD, which looks for exact similarities.

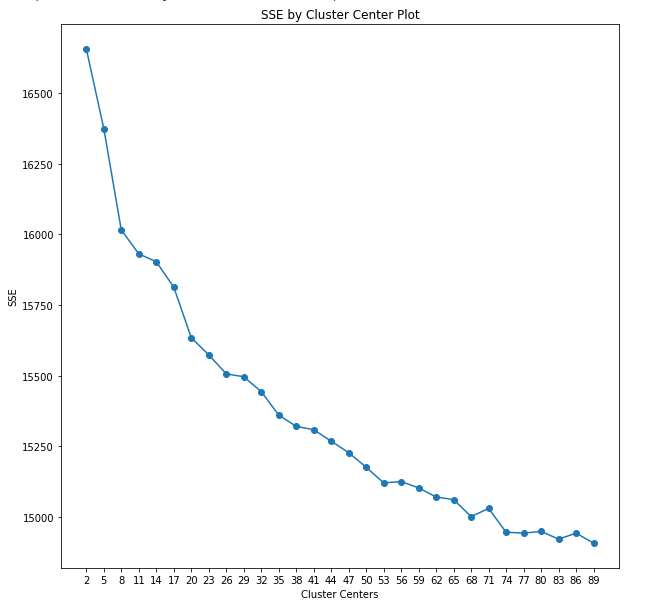
### K-Means clustering step

K-means is an unsupervised clustering algorithm which gets K as a hyperparameter and clusters the data by finding K points in space which are the means of the points closest to them.

In general, the choice of K is typically done by prior knowledge or by loss calculation. In our case, the choice of K defines the balance between the context search (TF/IDF) stage, and the NCD search:

* Small K values will produce less clusters with more samples in each cluster, moving more weight to the NCD search stage (Because NCD will have more samples to choose from)
* Large K values will produce more clusters with less samples in each, and the NCD stage will have less samples to “Choose from”, being constrained to work within the selected cluster.

The plot below shows the l2 error per K value. Since this is a first step in the funnel out of two steps, and since we wanted to allow some space for the NCD step, we chose a relatively low K value of K=16, leaving relatively high variation within the clusters.



The below plots show the TF/IDF vectorized data reduced to 2-D using PCA and TSNE, where the point color indicates the selected cluster:

