Big Data Analysis
Assignment 1
Report

Earass Ahmad 19I-1254
Asfand Yar Ali 19I-1212

19 Apr 2021

1 Introduction

This assignment is about using the Pokec social network data to generate meaningful insights from it and to implement Locality Sensitive Hashing to generate similar items. We create a thorough data pipeline to implement these tasks. The core components of the pipeline are discussed in the later sections.

2 Data Pipeline

Following are the core modules in the data pipeline:

2.1 Dataset

The online available Pokec data set is a large dataset having more than 1.6 million records and more than 50 columns. The primary language used in the dataset is Slovakian. The records are details of the Pokec social media users. Some of the columns had plenty of null values, whereas, some of the records had numerous missing values. The dataset is in the form of a tab-separated text file which is read into memory using the Pandas package of Python. The Pandas library reads it in the form of tabular data making it easier and faster to perform column operations on the dataset. The dataset is then filtered, cleaned, and explored before passing it to the LSH engine.

2.2 Data exploration

In order to obtain information about the dataset at hand, we need to perform some sort of exploratory analysis on it. Using different data analysis methods and visualization techniques will ensure we have a richer understanding of the data. Once data exploration has uncovered connections within the data, and then are formed into different variables, it is much easier to prepare the data into charts or visualizations. Data exploration can help cut down the massive data set to a manageable size where we can focus our efforts on analyzing the most relevant data. It is both an art and a science.

In our work, we have explored the data on multiple dimensions and built visualizations on them. Some of them include:

1. Age distribution analysis
2. Marital status, smoking status, and drinking status based on gender
3. Marital status, smoking status, and drinking status based on age groups
4. Number of years since people are registered on Pokec
5. BMI analysis of the users
6. Percentage of filled values in rows
7. Percentage of missing values in columns

The above analysis is provided in the form of charts in the visualizations report.

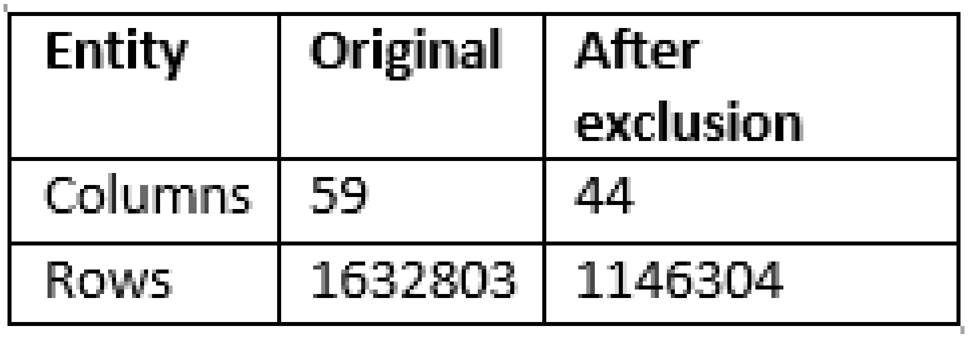
2.3 Data Cleaning

Data cleaning is an imperative task in any data science project. In order to achieve good results, a thorough preprocessing is performed on the raw dataset. After the data cleaning, the exploratory analysis is run again to visualize the cleaned data.

Following data cleaning operations are performed on the data:

2.3.1 Data Reshaping

The Raw data contained 1632803 records and 59 columns. A lot of these columns and records have missing values. In order to shorten our dataset without discarding the relevant information, we obtain an approach based on completion percentage in rows and columns. We calculate the percentage of the missing values for each column. Using that, we create a threshold at the 75th percentile. Any column that has missing values percentage greater than the threshold is excluded. A similar approach is obtained for the rows i.e. any row having less than the 75th percentile of filled values is excluded.



Entity	Original	After exclusion
Columns	59	44
Rows	1632803	1146304

Figure 1: Data shape before and after exclusion

2.3.2 Value Mapping

Some of the columns values are mapped in order to be more clear while visualizing and further data processing. These columns include:

1. Gender: the values are mapped from 1, 0 to Male and Female
2. Marital status: the values are converted from Slovakian language to English. Top 5 occurred words are selected and translated manually to English
3. Smoking status: the values are converted from Slovakian language to English. Top 5 occurred words are selected and translated manually to English
4. Drinking status: the values are converted from Slovakian language to English. Top 5 occurred words are selected and translated manually to English

2.3.3 Derived Fields

Some of the new fields are derived from the existing fields to have a better meaning and visualization. These fields include:

1. Weight and Height: The body field is split into Weight and Height columns
2. BMI: The BMI field is calculated off the Weight and Height fields
3. AgeGroup: The Age values are stratified into the Age groups

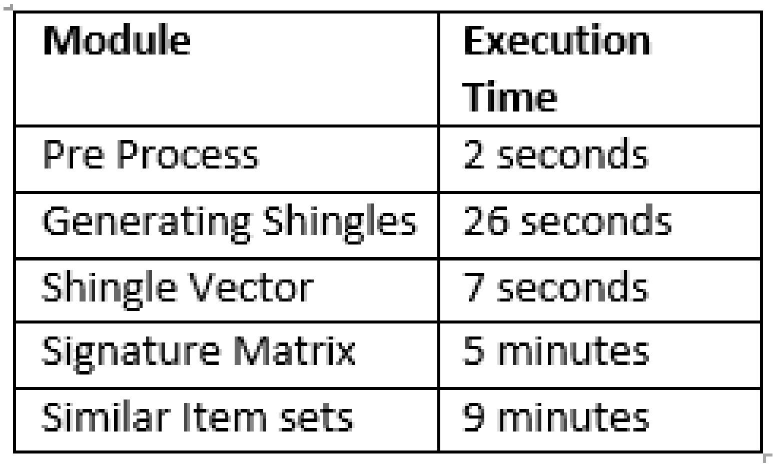
2.4 Locality Sensitive Hashing

Locality Sensitive Hashing (LSH) is a procedure for finding similar pairs in a large dataset. With LSH, similar items can be found with precision. We can create buckets having similar items with less computation and without having to compare each and every item with each and every other item.

In our implementation, we have used the cleaned Pokec dataset to find similar pair of records. Following is the breakdown of the LSH implementation:

1. Preprocessing: Includes excluding irrelevant columns like user id

2. Create Shingles: Each record in the dataset is coverted into a set of shingles
3. Create Shingle Vector: Union of all the sets of shingles to create a large set. Using that set, we create a shingle vector in the form of a dictionary that has a unique numeric Id for each shingle. The total number of unique shingles generated is 6623093
4. Generate Signature Matric: A signature matrix is generated with 10 hash functions. The number of hash functions can be passed as an argument to the function. So, the final matrix shape is 10 * 1146304
5. Compute Similarity: Finally, possible similar items are grouped together. This is implemented using the bands approach.
6. Output: There are two outputs from the LSH pipeline. One is the list of all the similar users. User Id is used as the label to refer to the user. The other is the index for referring to the hash buckets where the pairs are stored. Both the outputs are exported as csv files



Module	Execution Time
Pre Process	2 seconds
Generating Shingles	26 seconds
Shingle Vector	7 seconds
Signature Matrix	5 minutes
Similar Item sets	9 minutes

Figure 2: Time consumed in executing each module

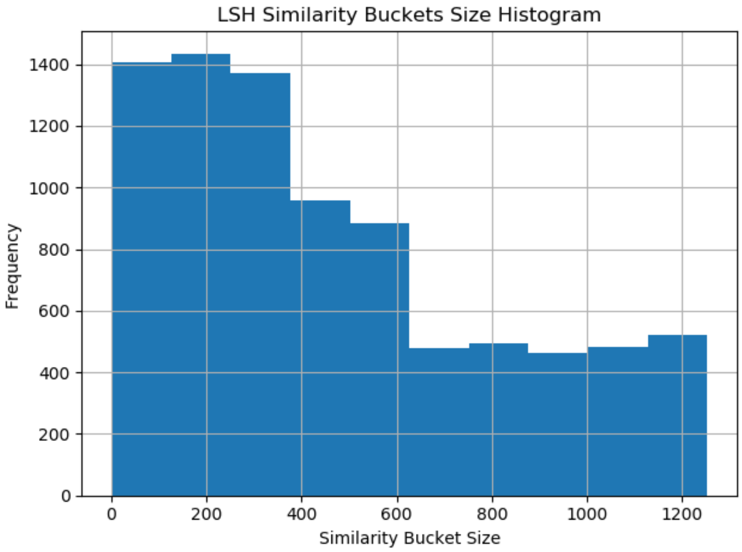


Figure 3: LSH Similarity Buckets Size Histogram

The Histogram above displays the frequency of size of buckets generated by the LSH. It could be seen that buckets with a greater number of similar users are lesser in size as compared with buckets that have a smaller number of similar users. This means that smaller-sized buckets are more frequent in our output.

2.5 Process Flow Diagram

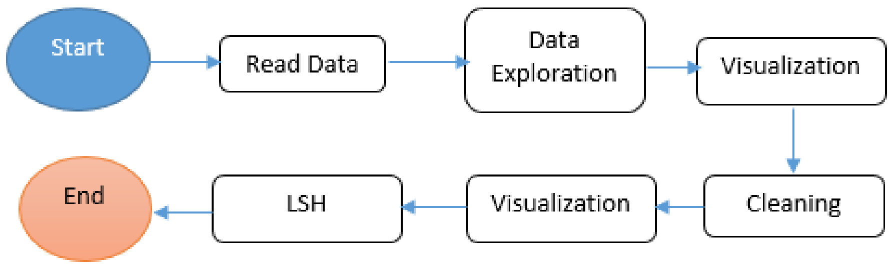


Figure 4: Process Flow Diagram

3 Tools

In this work, we have primarily used Python programming language. Following packages of Python are used:

1. Pandas: Used for data reading, wrangling and manipulation
2. Numpy: Used for performing data operations of large arrays
3. Matplotlib: Used for generating visualizations
4. Logging: To log the process flow

4 Conclusion and Future Work

This implementation was quite a learning opportunity. The cleaning and visualization of such a large dataset was a hard but indeed an interesting task. Writing the LSH code helped us understand the technique in depth.

In the future, we can clean the dataset even further to generate even more meaningful insights. Also, a more cleaned dataset would yield better results on LSH.