CS6720 - Data Mining
Assignment I

Instructions:

1. The assignment consists of 8 questions.

2. You are required to submit a report in PDF (typeset in LaTex).

3. Some of the problem numbers referred in this assignment are based on version 1.2 of the book "**Mining Of Massive Datasets**" (Link for the soft copy is available on moodle).

4. For the questions of type `implementation', the report should contain implementations(code of map and reduce functions)

5. Efficient implementation will carry more marks. Remember that efficiency of a map reduce program depends both on the time complexity and on the amount of intermediate data to be sent to the reducer (Use of **Combiner**)

Questions:

1. **Implementation - Excercise 2.3.1 from the textbook.**
Design map-reduce algorithms to take a very large file of integers and produce as output:

(a) The largest integer.
(b) The same set of integers, but with each integer appearing only once.
(c) The count of the number of distinct integers in the input.

2. **Implementation - Average rating of the movies**
Write map, reduce and combine functions using Hadoop to find average rating of movies.
Dataset: MovieLens 100K
The full u data set, 100000 ratings by 943 users on 1682 items. Each user has rated at least 20 movies. Users and items are numbered consecutively from 1. The data is randomly ordered. This is a tab separated list of **user id | item id | rating | timestamp.**

3. **Implementation - Excercise 3.3.7 from the textbook**
Suppose we want to use a map-reduce framework to compute minhash signatures. If the matrix is stored in chunks that correspond to some columns, then it is quite easy to exploit parallelism. Each Map task gets some of the columns and all the hash functions, and computes the minhash

signature of its given columns. However, suppose the matrix were chunked by rows, so that a Map task is given the hash functions and a set of rows to work on. Design Map and Reduce functions to exploit map-reduce with data in this form.

4. **Implementation - Excercise 3.4.4 from the textbook**

Suppose we wish to implement LSH by map-reduce. Specifically, assume chunks of the signature matrix consist of columns, and elements are key-value pairs where the key is the column number and the value is the signature itself (i.e., a vector of values).

(a) Show how to produce the buckets for all the bands as output of a single map-reduce process. Hint : Remember that a Map function can produce several key-value pairs from a single element.

(b) Show how another map-reduce process can convert the output of (a) to a list of pairs that need to be compared. Specifically, for each column i, there should be a list of those columns j > i with which i needs to be compared.

Theory questions

Question No.	Problem no.
5	3.3.3
6	3.3.6
7	3.4.2
8	3.4.3