## BDS Assignment 2

A restaurant data set from the UK contains data records in following JSON format.

Sample record:

{

“URL":"http://www.just-eat.co.uk/restaurants-1-2-3-chinese-rowlands-gill/menu",

“\_id":{"$oid":"55f14312c7447c3da7051b2f"},

"address":"Unit 4 Spencer House”,

"address line 2”:”Swalwell",

“name”:"1 2 3 Chinese",

“outcode":"NE16",

“postcode":"3DS",

“rating":4.5,

“type\_of\_food":"Chinese"

}

Each record is for a restaurant with its name, unique id, address, cuisine/type of food, rating and zipcode in 2 parts (outcode and postcode).

**Part 1: Data loader application**

Write a data loader application to do the following:

1. Read the records and insert them into a MongoDB collection but with the transformations specified below. Choose appropriate data types.
2. There may be duplicate records in the data set. Delete duplicates during insertion.
3. Remove the URL field during insertion.
4. Combine the outcode and postcode into one zip code field in the DB, i.e. zip code = outcode + postcode.
5. Create any indices that you think may be important based on the query types specified in the next part.

**Part 2: Data query application**

Write an end user facing query application which can run the following query types after taking required user inputs. Only specified output fields need to be returned.

* 1. User provides a choice of cuisine and zipcode, the query returns a list of top 5 restaurant names with full address and rating with results sorted descending order based on rating.
  2. User provides a string to be searched in the address field and a minimum rating. The query returns all restaurant names and cuisine that match the inputs provided along with the match score. Sort output by descending score.
  3. Give a cuisine, output how many (i.e. count) matching restaurants are there per zip code. Sort descending the 2 column output (zipcode, count) by the count.
  4. Show the average rating per type of food and provide an ascending sorted output (type of food, rating) by rating.

You can create the data query application with the 4 types of query types encoded as functions. The user can run this program from the command line with parameters : type of query (1-4) and corresponding inputs. The query is executed on the DB and results returned on console.

Deliverables:

1. Working code (in Python, Java or any preferred language) for the 2 programs (a) data loader (b) data query.
2. At least 4 test cases (one for each query type) with the command line parameters that yield results to demonstrate.