Data Warehousing and On-line Analytical Processing and Classification and Clustering Report

Background:

Data mining is a process to extract useful information from a vast amount of data. It is used to discover new, accurate and useful patterns in data, looking for meaning.

Data: refers to characteristics /numerical/categorical which are collected through observation.

Datasets: collections of items which are described by a set of attributes.

Data Mining starts with collection or sourcing of raw data. Raw data can have multiple issues like poor data quality such as noisy data, dirty data, missing values, inexact or incorrect values, inadequate data size and poor representation in data sampling.

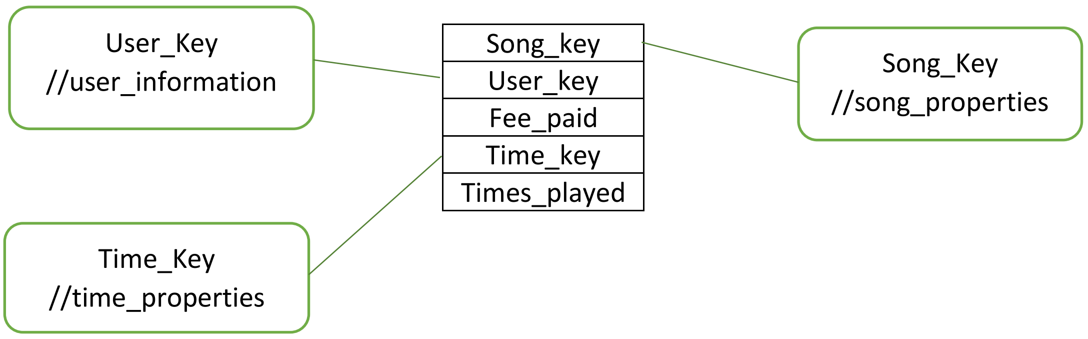
Data Warehousing and On-line Analytical Processing:

1. Data Warehousing: It is a process of constructing and using data in warehouses.
2. Data Warehouse: A data warehouse is constructed by integrating data from multiple heterogeneous sources that support analytical reporting, structured and/or ad hoc queries, and decision making. Data warehousing involves data cleaning, data integration, and data consolidations.
3. OLAP (for online analytical processing): It is a used to perform analysis of large volumes of multidimensional data at high speeds. The data can be from unified, centralized data store, like a data warehouse.
4. OLTP, or online transactional processing, is necessary for enabling the real-time execution of large numbers of database transactions by many people, more often over the internet.

   Example:

Q1: A data warehouse for a music streaming company consists of the dimensions song, user, time (time and date of when the user listened to a song), and the two measures count (how many times a user listened to the song) and fee (fee paid by the streaming company to the artist every time a user listens to that song).

1. Draw a schema diagram for the above data warehouse using a star schema.



Star schema representation

2. Starting with the base cuboid [time, user, song], what specific OLAP operations should be performed in order to list the total fee collected for a given song for a given month of a given year (e.g. October 2021)?

Starting with the base cuboid of [time, user, song]. These are the following operations that can be performed to calculate the total fee collected over a song for a month:

- Perform slicing of the year of interest on time dimension.
- Next implement the drill down operation from year to month on time dimension.
- Repeat the slice on the month of interest in time dimension. This gives us the User and Song dimensions of that month and year.
- Now slice the song of interest, help in getting the numbers of users played that particular song in a month and year and the fee that has to be paid for it.
- Calculate fee collected: fee paid*sum of users played a particular song.

3. Assume that the time dimension has 4 levels: day, month, quarter, year; and that the song and user dimensions both have 1 level (not including the virtual level 'all'). How many cuboids will this cube contain (including the base and apex cuboids)?

   Using the below formula for computing cuboids:

$$T = \prod_{i=1}^{n} (L_i + 1)$$

   Time = 6, song = 2, user = 2
   No. of cuboids:
   Time * song * user : 6*2*2 = 24

<u>Q2:</u> Suppose that a car rental company has a data warehouse that holds record ID lists of vehicles in terms of brands (Audi, Ford, Mini) and store branches (Tower Hamlets, Newham, Hackney). Each record consists of a combination of vehicle brand and branch, and records for all combinations exist. We would like to index the OLAP data using bitmap indices. Write down the base table for record IDs, and the corresponding bitmap index table for vehicle brand. [0.5 marks out of 5].

<u>*Base Table:*</u>

RID	Brands	Branch
R1	Audi	Tower Hamlets
R2	Audi	Newham
R3	Audi	Hackney
R4	Ford	Tower Hamlets
R5	Ford	Newham
R6	Ford	Hackney
R7	Mini	Tower Hamlets
R8	Mini	Newham
R9	Mini	Hackney

Bitmap indexing based on car brands:

RID	Audi	Ford	Mini
R1	1	0	0
R2	1	0	0
R3	1	0	0
R4	0	1	0
R5	0	1	0
R6	0	1	0
R7	0	0	1
R8	0	0	1
R9	0	0	1

Bitmap indexing as per places:

RID	Tower Hamlets	Newham	Hackney
R1	1	0	0
R2	1	0	0
R3	1	0	0
R4	0	1	0
R5	0	1	0
R6	0	1	0
R7	0	0	1
R8	0	0	1
R9	0	0	1

Q3: Using the same CSV file and data cube in the above lab tutorial, modify the "tutorial_model.json" file to include aggregate measures for the minimum and maximum amount in the data cube. Using these implemented aggregate measures, produce the values for the minimum and maximum amount in the data per year. Make sure to show your workings in the PDF report. You can read the Cubes package documentation for assistance in this task. [1 mark out of 5]

Q4: Using the CSV file "country-income.csv" (found in the week 5 supplementary lab documents), perform the following:

Load the CSV file using Cubes, create a JSON file for the data cube model, and create a data cube for the data. Use as dimensions the region, age, and online shopper fields. Use as measure the income. Define aggregate functions in the data cube model for the total, average, minimum, and maximum income. In your PDF report, show the relevant scripts and files created. [0.5 marks out of 5]

Using the created data cube and data cube model, produce aggregate results for: the whole data cube; results per region; results per online shopping activity; and results for all people aged between 40 and 50. [1 mark out of 5]

#CODE

Converting csv to json file (reference taken from web: https://pythonexamples.org/python-csv-to-json/#3)

```
import csv

import json

def csv_to_json(csvFilePath, jsonFilePath):

  jsonArray = []

  #read csv file

  with open(csvFilePath, encoding='utf-8') as csvf:

    #load csv file data using csv library's dictionary reader

    csvReader = csv.DictReader(csvf)
```

```
        #convert each csv row into python dict
        for row in csvReader:
            #add this python dict to json array
            jsonArray.append(row)

    #convert python jsonArray to JSON String and write to file
    with open(jsonFilePath, 'w', encoding='utf-8') as jsonf:
        jsonString = json.dumps(jsonArray, indent=4)
        jsonf.write(jsonString)

csvFilePath = r'country-income.csv'
jsonFilePath = r'country-income.json'
csv_to_json(csvFilePath, jsonFilePath)
```

Working with json file: Part 1 - Followed the same approach as provided in the tutorial

```
from cubes import Workspace
import cubes as cubes
from sqlalchemy import create_engine
from cubes.tutorial.sql import create_table_from_csv
engine = create_engine('sqlite:///data.sqlite')
create_table_from_csv(engine,
                      "country-income.csv",
                      table_name="country_income",
                      fields=[
                          ("region", "string"),
                          ("age", "integer"),
                          ("income", "integer"),
                          ("online_shopper", "string")],
                      create_id=True
                     )
```

Classification and clustering:

Q1: Consider a dataset D that contains only two observations x1=(1,−1) and x2=(−1,1) . Suppose that the class of the first observation is y1=0 and that the class of the second observation is y2=1 . How would a 1-nearest neighbour classifier based on the Euclidean distance classify the observation x=(−2,3) ? What are the distances between this new observation and each observation in the dataset?

For x1 = $\sqrt{(1+2)^2+(-1-3)^2}$ = 5

For x2 = $\sqrt{(-1+2)^2+(1-3)^2}$ = $\sqrt{5}$

The observation can be assigned to class 0 as it is closer to x2.

Q2: Consider a dataset D that only contains observations of two different classes. Explain why a k - nearest neighbour classifier does not need a tie-breaking policy when k is odd. [0.5 marks out of 5]

- When k is odd; the voting is in favor of a class. A separate tie breaking not needed.
- When k is even; chance that k-nearest neighbors have a 50-50 voting for the observations class. In this case a tie breaker is needed.

Q3: Explain why a classifier that obtains an accuracy of 99.9% can be terrible for some datasets. [0.5 marks out of 5]

In such datasets, the classifier tends to be in a class with more samples, starting at to predict everything for that class. Due to the large number of samples in this class, the accuracy of the looks high in the end. However, this model can adversely affect the less-represented class. Implicitly low actual performance creates the illusion of a well-trained generalized model in terms of accuracy.

Q4: Consider a classifier tasked with predicting whether an observation belongs to class y (positive class). Suppose that this classifier has precision 1.0 and recall 0.1 on a test dataset. If this classifier predicts that an observation does not belong to class y , should it be trusted? Should it be trusted if it predicts that the observation belongs to class y ? [0.5 marks out of 5]

The low recall of 0.1, shows the classifier has marked many positives as negatives which in turn misses many positive observations. The classifier cannot be trusted as it will correctly separate out the negative observations but will tend to miss out the positives. Hence, wrongly classifying the positives as negative class as well.

Q5: Based on the confusion matrix shown in this lab notebook, what is the pair of classes that is most confusing for the 1 -nearest neighbour classifier trained in the previous sections? [0.5 marks out of 5]

Most confusing classes: 4 and 9, most confused pair is 4 with 9.

Q6: Train a support vector machine classifier using the same (subsampled) training dataset used in the previous sections and compute its accuracy on the corresponding test dataset. You can use the default hyperparameters for the class SVC from sklearn.svm. Show the code in the report. [1.25 marks out of 5]

Q7: Using the same (subsampled) training dataset used in the previous sections, employ GridSearchCV to find the best hyperparameter settings based on 5-fold cross-validation for a RandomForestClassifier.

Consider n_estimators $\in\{50,100,200\}$ and max_features $\in\{0.1,0.25\}$. Use the default values for the remaining hyperparameters. Compute the accuracy of the best model on the corresponding test dataset. Show the code in the report. [1.25 marks out of 5]