POLITECNICO
MILANO 1863

System and Methods for Unstructured Data Project: Amazon Reviews

Tommaso Capacci (10654230)

AY 2023/2024

Contents

1 Introduction

The dataset we've chosen to use represents a collection of reviews of products sold on Amazon. Datasets like this are usually used for *strategic purposes*, such as understanding the customers' needs and preferences, for *marketing purposes*, such as understanding the customers' opinions on a product, or *for sentiment analysis*, such as training a classifier to predict the sentiment of a review based on its content, since information about the review's text is also provided.

In our case, we will consider the first task, since it can be reduced to the problem of writing a suited set of queries and can be also efficiently supported by the database technology we decided to use, which is MongoDB. One of the main reasons for this technical choice is the fact that the dataset is in JSON format, which is natively supported by MongoDB. Moreover, the dataset is composed of a single collection of, possibly, nested documents, which is also a good fit for MongoDB, since it is a document-oriented database, meaning that it is specifically designed to store data as documents instead of relational tables.

We've decided to interpret the task of strategic analysis as the problem of extracting meaningful statistics from the dataset (such as the number of reviews per product, the average rating per product, the review with highest score, ...) which can be then included on the company's reports and used to make strategic decisions.

2 Data Wrangling

3 Dataset

The dataset of our choice is presented at *this link*. It is composed of multiple collections of documents, each one containing some reviews, registered between the 1996 and 2018, about a specific kind of products sold on Amazon. This fact makes the dataset suitable for our purposes, since it contains a lot of reviews but also makes it possible to filter the reviews by product category, allowing to eventually focus the analysis on a specific kind of products.

Among all the available collections, we've decided to use the one about **Digital Music** (can be retrieved at *this link*) since this file contains a suitable amount of reviews (169.781) while still weighting less then 100 MB, allowing us to upload it to a GitHub repository and parallelize the workload.

The documents inside the dataset have the following shape:


```
{
    "image": ["https://images-na.ssl-images-amazon.com/images/..."],
    "overall": 5.0,
    "vote": "2",
    "verified": true,
    "reviewTime": "01 8, 2015",
    "reviewerID": "A36GE53TK8V94L",
    "asin": "B000T1EJ0W",
    "style": {"Format:": "MP3 Music"},
    "reviewerName": "MysticWolf229",
    "reviewText": "the theme song to the one and only movie that ...",
    "unixReviewTime": 1420675200
}
```


where the fields have the following meaning:

- **image**: an array of URLs pointing to the images of the product review;
- **overall**: the rating of the product, a float number going from 1 to 5;
- **vote**: the number of votes the review received, saved as a string;
- **verified**: a boolean value indicating whether the review has been verified or not;

- **reviewTime**: the date of the review, saved in RAW date format as a string;
- **reviewerID**: the alphanumeric ID of the user who wrote the review;
- **asin**: acronym of Amazon Standard Identification Number, is the alphanumeric ID that represents a specific product;
- **style**: a subdocument containing additional data about the version of the product. In the case of this collection it just contains information about the format of the product, whose possible values will be retrieved with a suited query;
- **reviewerName**: a string containing the name of the user who wrote the review;
- **reviewText**: a string containing the text of the review;
- **summary**: a string containing a summary of the review;
- **unixReviewTime**: the date of the review in Unix epoch time format.

4 Queries

```
db.reviews.aggregate([
    {"$unwind": "$style"},
    {"$group": {"_id": "$style"}}
])
```