1. Dataset: http://jmcauley.ucsd.edu/data/amazon/

Table

Description automatically generated

Figure 1

Download all 5-core json files

2. Python Files:

* Jaccard Distance Calculation: Machine learning algorithms
  + Jaccard.py
  + Similarilty.py
* All other modules: HTML support
  + Display.py

3. Image Files

* logo.ics
* top.gif

4. HTML RESOURCES

* green.png
* red.png
* yellow.png
* style.css

Dependency Packages:

* + matplotlib
  + mpld3
  + nltk
  + numpy
  + pandas
  + py2app
  + pycularity
  + rpy2
  + scipy

**file\_to\_database.py**

* Instead of parsing every line in a csv or json file directly, this script first exports all data from the data files and keeps it stored in a database (can be on a server or serverless).
* A single database called Review.db will hold multiple tables (see Figure 1 for 5-core names) and store relevant meta data ([row count x column count], table number) about each data in a dictionary.
* This way we can more easily query, parse, and process data.
* Note that this is not a dynamic process. Script is ran before web hosting.

**Jaccard.py**

Three methods for detecting fake reviews:

**(i) Detection of Duplicate Reviews**

(ii) Detection of Anomaly in Review Count and Rating Distribution

(iii) Detection of Incentivized Reviews

**Jaccard Similarity**

* We can use the Jaccard Similarity algorithm to work out the similarity between two things. We might then use the computed similarity as part of a recommendation query.
  + For example, you can use the Jaccard Similarity algorithm to show the products that were purchased by similar customers, in terms of previous products purchased.
* Score between 0 and 1
  + 0 – large distance – lesser similarity
  + 1 – short distance – higher similarity

**Major Functions**

**find\_bigram\_crcs(1)**

**Description:**

Three types of duplicate reviews:

1. Duplicates from different reviewer IDs on the same product
2. Duplicates from the same reviewer ID on different products
3. Duplicates from different reviewer IDs on different products.

This system uses duplicate reviews as one of the methodologies to score the credibility of the product.

Figure 4 shows the steps used in detecting the duplicate reviews.

1. Each review is converted into a set of bigram shingles, which are formed by combining two consecutive words together.
   1. <https://www.elastic.co/blog/searching-with-shingles>

Diagram

Description automatically generated

1. Shingles are given shingle IDs using a CRC32 hash function.
   1. Used for increasing computational efficiency in large datasets.
2. Min-Hash signatures were calculated for each review using a random hash function which prevents from having to explicitly compute random permutations of all the shingle IDs

Diagram

Description automatically generated

**Display.py**

* Amazon.com allowed the sellers to offer products for free or at high discounts in exchange for positive reviews about that product.
* While most of such reviews have a disclaimer that *“the customer received this product in exchange for an honest review”* stated in the reviews, there still exist many reviews written under this condition without including the disclaimer.
* **Incentivized reviewers**, though they claim to be unbiased, tend to give positive and less critical reviews for the products compared with the non-incentivized reviewers.
* The occurrence of these reviews has transformed the review panels into advertising forum. Detecting such incentivized or biased reviews is often more challenging.
* Examples of incentivized reviews:
  + *“I got these at no charge in exchange for an honest review*
  + *I received this product in exchange for a truthful review and I must say that I am overall satisfied with it*
  + *I received this product at a discounted rate in exchange for my fair and honest.”*
* Figure 10 shows the overall steps in the detection of incentivized reviews.

**How are incentivized reviews handled?**

**def detectKeywords()**

**Description:**

* A collection of synonym phrases for a set of key phrases derived from common incentivized reviews are generated using the **Natural Language Toolkit** from **WordNet**.
  + WordNet is a large lexical database which resembles the thesaurus.
  + It groups words together based on their meanings and the specific sense of the words into sets of cognitive synonyms (synsets).
* If a key phrase such as “honest review” is given as input to the function, equivalent phrases are such as “truthful review”, genuine review”, “genuine feedback”, and so on are searched for.
* A dictionary was made of both single and double paired words, for example, “discount” and “no charge”.
* The reviews with these synonyms were identified using the regular expression and the time intervals were also captured for analysis.