DeMON: Deceptive Mining of Opinions & Notions

Munaf Arshad Qazi

Abstract

DeMON studies patterns of 1.6 million reviews for 63000 electronics provided by Amazon, visualizes and tries to mine them for fake and genuine reviews. This goes ahead hand in hand with estimating how reliable amazon reviews can be before making buying decisions. DeMON using a python and R duo, successfully pinpoints 49 products which are highly likely to be fake ones using a combination of Cosine Similarity for reviews and studying anomalies in posting times per product.

1 Introduction

As access to internet has increased exponentially in the past decade, the internet has indeed become a web; a web of misinformation and false pretenses. Many popular websites today are filled with cases of deceptive opinion spams. To combat this, very recently (October 3, 2016), Amazon decided to prohibit incentivized reviews unless they're facilitated through the Amazon Vine Program [1]. I, being an avid-Eshopper, needed to see how effective Amazon's current methods are for handling fake reviews and how reliable the reviews actually are. Amazon, currently, merges identical product reviews together and cross references them with each other for duplicates. Then passes reviews into it's own detection system (the details of which aren't disclosed). In the following sections I explain my approach on identifying fake product reviews using Natural Language Processing and gauge how effective Amazon has been in its efforts considering the data set it released.

2 Approach & Reasoning

"Sentiment Analysis is one of the most active research areas in natural language processing and is also widely studied in data mining, web mining, and text mining. In fact, this research has spread outside of computer science to the management sciences and social sciences due to its importance to business and society as a whole." (Liu, 2012) [3]

In his book, Sentiment Analysis and Opinion Mining, Bing Liu from UIC writes why this is true and explains that mining opinions, although a difficult task, is achievable. He explains this is possible due to the highly active research topic of sentiment analysis these days. The works of et. al. Mukherjee [5] have shown me fake reviews for a product have a lot in common. They are usually written by one agency and follow a same trend in all lexical, semantic and syntax ways and temporal ones as well (posting times).

Apart from Liu's book and Mukherjee's work, I used Introduction to Information Retrieval by Christopher Manning (Stanford) [2] to come up with a few important notions about information retrieval for deceptive opinion mining and sentiment analysis in general. For any type of text analysis, the first step (after preprocessing) is to convert text into numbers. This is achieved by creating tfidf vectors and sparse matrices. This matrix can then be used for different sentiment analysis.

My initial research indicated that to mine opinions properly and extract information from them we need to take advantage of both fields: psychology and data science. Studies show that organizations which hire individuals to write fake reviews try to keep it as discrete as possible. This is why usually it is the same group of people writing reviews. A psycho-analysis of the human thought process dictates that a person's writing style is something he/she cultivated it over a life long process. Therefore when writing about something, his basic writing style, his semantics, his structure and his syntax is very difficult to mutate. Apart from this, when writing professionally, writers often start plagiarizing other reviews instead of writing creatively. I took these notions to my advantage and using the power of natural language processing harvested cosine similarity between different reviews.

Cosine Similarity is a powerful technique used abundantly these days to identify plagiarism and document similarity. It begins by converting documents into a sparse matrix and then using term frequency – inverse document frequency (TF-IDF) scores to decipher how important different words are in a document. These TF-IDF scores for various documents can be used to calculate similarity. Just like taking a cos-product gives us the angle between two vectors, a cosine similarity score is the cos product of all TF-IDF scores normalized, for documents under analysis.

After implementing my cosine similarity analysis, the next metric I considered was analyzing anomalies in posting dates. A reviewing agency has to meet deadlines for review postings therefore reviews see a sudden spike during a small time. I took a window of 7 days to analyze if similar reviews were posted nearby temporally.

Using both these metrics made sense because subconsciously, a person cannot change his writing style, nor does an agency want to make it's deadlines for postings flexible. Thus following Bing Liu's advice, Christopher Manning's teachings, Mukherjee's work and my own research on human psychology, I created my own mathematical model to solve the problem at hand.

2.1 Mathematical Model

Cosine Similarity is the mathematical evaluation of the angle between two vectors. For documents, in information retrieval, it is the measure of similarity of documents. Just like for angles, it is commonly evaluated for documents as:

$$Cosine\,Similarity = cos(\theta) = \frac{A \cdot B}{|A|\,|B|} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2}\sqrt{\sum_{i=1}^{n} B_i^2}}$$

where A and be are different documents. Each term is notionally assigned a different dimension and a document is characterized by a vector where the value of each dimension corresponds to the number of times that term appears in the document (the sparse matrix). Cosine similarity then gives a useful measure of how similar two documents are likely to be in terms of their subject matter. Similarly, Cosine distance gives a measure of how distant two documents are, commonly calculated as:

$$Distance = 1 - Cosine\,Similarity$$

For each product the average cosine similarity can be written as:

$$AvgCosSim = \frac{\sum_{i=1}^{n} cos(\theta)}{n}$$

Where n is the total number of reviews.

Cosine similarity is just one measure of deduction. The next step is calculating reviews based on their times stamps. To mathematically write it, consider the indicator function:

$$I_A(x) = \begin{cases} 1 & if\, x \in A(x) \\ 0 & otherwise \end{cases}$$

Where x is a review time and

$$A(x) = \Big\{ |x - S(X)| < D \Big\}$$

Where S(X) is a set of all times for a product reviews except x and D is the number of days we want our review window to be of. I chose it to be of 7 days for my implementation.

Therefore, for all reviews, the time score evaluates to:

$$Time\,Score = \frac{\sum_{i=1}^{n} I_A(i)}{n}$$

Using these two measures, DeMON creates a final equation to calculate fakeness of the product:

$$Fakeness(P) \begin{cases} 1 & if AvgCosSim > 0.7 \\ W\sum_{i=1}^{n} I_{PA}(i) + (1-W)AvgCosSim(P) & otherwise \end{cases}$$

Where Asim(P) is average cosine similarity for product P, W is an assigned weight for the time component (I chose it to be 0.75) and I_{PA} is the indicator function for product P. If $Fakeness(P)$ evaluates to be greater than 0.5 for a product, it is marked as a potential fake reviewed one.

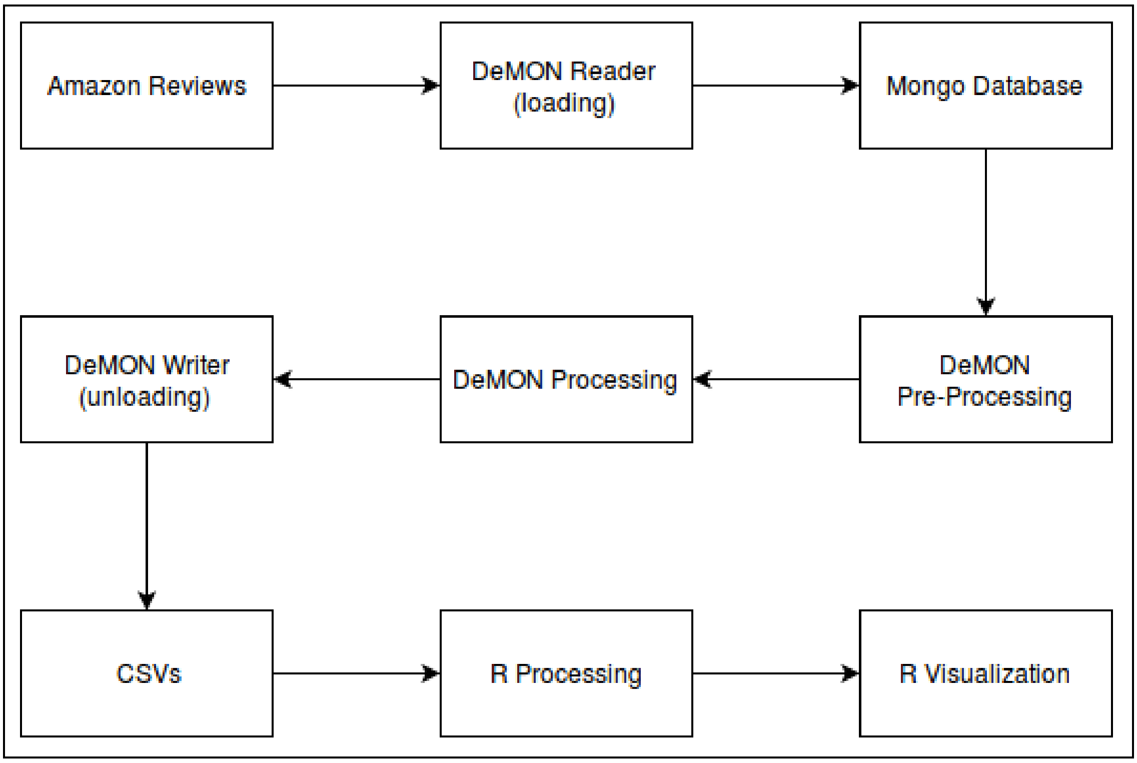


Figure 1: System Flow diagram for DeMON

3 System Architecture

DeMON uses a python engine at its back end for all major computations. The engine does takes data from the Mongo Database storage and does all the pre-processing and processing and outputs CSVs which are later read into R for further processing and visualizations. Figure 1 shows the System Architecture for the DeMON. All modules are explained in detail in Sections 4, 5 and 6.

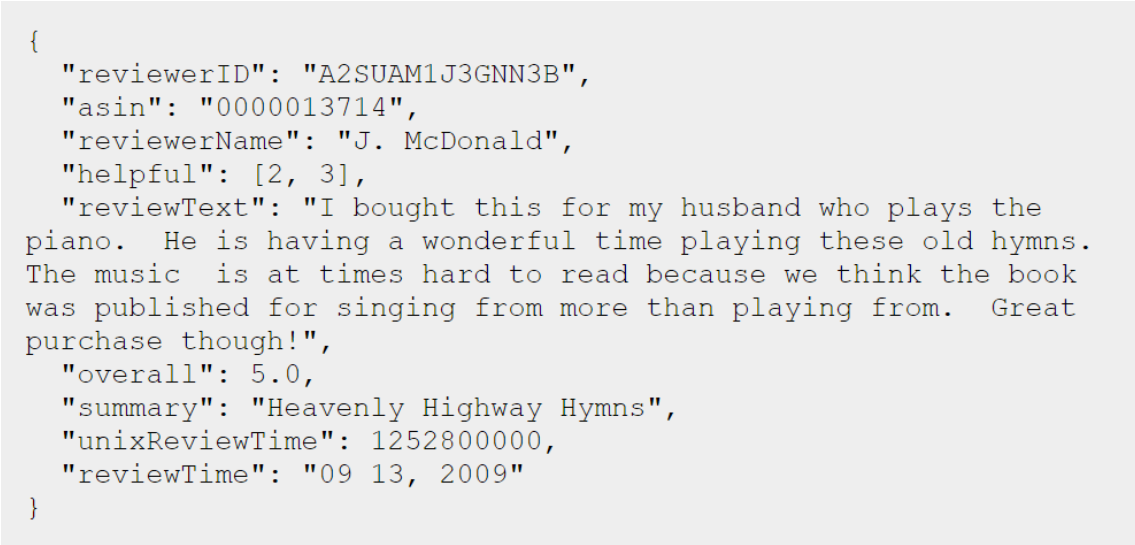
4 Data Architecture

DeMON uses a NoSQL approach while dealing with data. This is because of the uncertainty of presence of some fields in the dataset alongside the practicality of using NoSQL for a large amount of unstructured data. Because all these issues were addressed in Mongo and because of the flexibility and power mongo offers of manipulating documents and collections directly and not compromising on future extensibility, DeMON was built incorporating MongoDB.

DataSet

I used Amazon Datasets containing more than **1.6 million reviews** for different electronics. The dataset contains reviews since 1999. Because the format of the datasets is pretty similar, I started working with a subset dataset for cellphones (about 200k reviews) for the prototype and expanded it to other electronics. In this huge set of data, I looked at review text, review date and times. The dataset was collected by a research colleague at UCSD (Julian McAuley) who has made it freely available[6] after removing all duplicates from the data. After numerous interactions with him, he was generous enough to grant me access to use the metadata for all products as well. The meta data contained product information for **9.4 million products** [7].
Sample Review and Metadata objects are shown in Figure 2 and 3 where the fields in the review are:

- reviewerID - ID of the reviewer, e.g. A2SUAM1J3GNN3B
- asin - ID of the product, e.g. 0000013714



```
{
  "reviewerID": "A2SUAM1J3GNN3B",
  "asin": "0000013714",
  "reviewerName": "J. McDonald",
  "helpful": [2, 3],
  "reviewText": "I bought this for my husband who plays the
piano.  He is having a wonderful time playing these old hymns.
The music  is at times hard to read because we think the book
was published for singing from more than playing from.  Great
purchase though!",
  "overall": 5.0,
  "summary": "Heavenly Highway Hymns",
  "unixReviewTime": 1252800000,
  "reviewTime": "09 13, 2009"
}
```

Figure 2: Sample Review JSON by Amazon

- reviewerName - name of the reviewer
- helpful - helpfulness rating of the review, e.g. 2/3
- reviewText - text of the review
- overall - rating of the product
- summary - summary of the review
- unixReviewTime - time of the review (unix time)
- reviewTime - time of the review (raw)

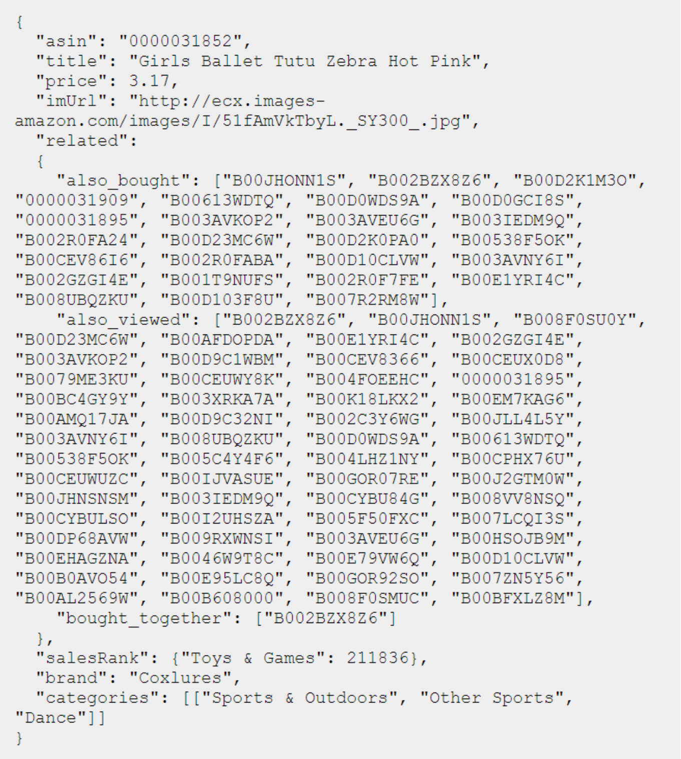
while the ones in metadata correspond to:

- asin - ID of the product, e.g. 0000031852
- title - name of the product
- price - price in US dollars (at time of crawl)
- imUrl - url of the product image
- related - related products (also bought, also viewed, bought together, buy after viewing)
- salesRank - sales rank information
- brand - brand name
- categories - list of categories the product belongs to

DeMON only needs a few fields from this JSON such as review text, review time, asin, price, categories and they will be filtered out when reading data.

5 Integration Architecture

Connecting Mongo to the DeMON engine and then porting data out for R is a task in itself. The flexibility that Python brings with itself though helped overcome these bridges with ease. For inserting data into a Mongo Collection from a JSON the DeMON reader was written. All scripts of this module were collected into one file called loading.py.



```
{
  "asin": "0000031852",
  "title": "Girls Ballet Tutu Zebra Hot Pink",
  "price": 3.17,
  "imUrl": "http://ecx.images-
amazon.com/images/I/51fAmVkTbyL._SY300_.jpg",
  "related":
  {
    "also_bought": ["B00JHONN1S", "B002BZX8Z6", "B00D2K1M3O",
"0000031909", "B00613WDTQ", "B00D0WDS9A", "B00D0GCI8S",
"0000031895", "B003AVKOP2", "B003AVEU6G", "B003IEDM9Q",
"B002R0FA24", "B00D23MC6W", "B00D2K0PA0", "B00538F5OK",
"B00CEV86I6", "B002R0FABA", "B00D10CLVW", "B003AVNY6I",
"B002GZGI4E", "B001T9NUFS", "B002R0F7FE", "B00E1YRI4C",
"B008UBQZKU", "B00D103F8U", "B007R2RM8W"],
    "also_viewed": ["B002BZX8Z6", "B00JHONN1S", "B008F0SU0Y",
"B00D23MC6W", "B00AFDOPDA", "B00E1YRI4C", "B002GZGI4E",
"B003AVKOP2", "B00D9C1WBM", "B00CEV8366", "B00CEUX0D8",
"B0079ME3KU", "B00CEUWY8K", "B004FOEEHC", "0000031895",
"B00BC4GY9Y", "B003XRKA7A", "B00K18LKX2", "B00EM7KAG6",
"B00AMQ17JA", "B00D9C32NI", "B002C3Y6WG", "B00JLL4L5Y",
"B003AVNY6I", "B008UBQZKU", "B00D0WDS9A", "B00613WDTQ",
"B00538F5OK", "B005C4Y4F6", "B004LHZ1NY", "B00CPHX76U",
"B00CEUWUZC", "B00IJVASUE", "B00GOR07RE", "B00J2GTM0W",
"B00JHNSNSM", "B003IEDM9Q", "B00CYBU84G", "B008VV8NSQ",
"B00CYBULSO", "B00I2UHSZA", "B005F50FXC", "B007LCQI3S",
"B00DP68AVW", "B009RXWNSI", "B003AVEU6G", "B00HSOJB9M",
"B00EHAGZNA", "B0046W9T8C", "B00E79VW6Q", "B00D10CLVW",
"B00B0AVO54", "B00E95LC8Q", "B00GOR92SO", "B007ZN5Y56",
"B00AL2569W", "B00B608000", "B008F0SMUC", "B00BFXLZ8M"],
    "bought_together": ["B002BZX8Z6"]
  },
  "salesRank": {"Toys & Games": 211836},
  "brand": "Coxlures",
  "categories": [["Sports & Outdoors", "Other Sports",
"Dance"]]
}
```

Figure 3: Sample Metadata JSON by Amazon

DeMON Reader/Loading

The implemented reader was very carefully written. The reason for this being, the data needed to be prepared for future processing and normal strings wouldn't work properly for DeMON's time analysis. Also not too much processing should've been done while reading itself because that would've just made the reader really slow. For these reasons the DeMON reader achieved the following two tasks:

Result Writer From the large number of fields, the fields relatively important for DeMON were Product ID to identify each product, the Review Time for time analysis, the Review Text for similarity analysis and userId, review score and review summary for later metadata joins.

Time Converter Converting string time (unix review time) to a date time object was a really important task. The DeMON reader fixed all times and constructed a proper mongo collection which was then used for further preprocessing.

Once into a mongo collection, I was able to get a complete count of all the filtered and fixed (data prepared) reviews. The total number of reviews for all electronics in my dataset quantified to a grand sum of 1.69 million.
Apart from loading data, the reader also has methods to read in CSVs.

DeMON Writer/Unloading

For preparing results to output into a CSV file which would be readable by R, another integration module was implemented, called the DeMON writer. All methods used for unloading were docked together in unloadingResults.py. It did the following two major tasks:

Result Writer This submodule created CSVs for the results given to it. It was used to create a collection of Cosine Similarity Results, a collection of Fake Reviews, as well as a separate collection of products with a majority of fake reviews. For the time reviews it, wrote out time scores calculated individually into CSVs.

Dumper This submodule created a dump for all the reviews in a folder form subdirectory. With each product having separate folders and each review inside of them in txt files. For the time dumps, this part created one folder with txt files, with one file per product and times inside of each txt file.

6 Design

6.1 DeMON preprocessing, processing and unloading

DeMON follows a series of steps to generate results. Before any processing can be achieved, the dataset needs to be compressed and all reviews for the same products need to be merged together. This creates a final dataset which can be processed on. The complete process flow for DeMON is shown in the process flow diagram in Figure 1. Figure 4 is an elaboration of the DeMON preprocessing, processing and writing steps from Figure 1.

After merging the 1.6 million reviews, there are a total of **63001 products** in the aggregated Mongo collection. There were 2 aggregations in total, creating one collection for reviews, and another collection for posting times.

Cosine Similarity

The next step is Cosine Similarity Analysis, which follows the following process:

1. Read the aggregated collection from MongoDB (DataSet Filtering has already taken place during loading).
   MongoDB provides extensive functionality for manipulating collections and documents directly. Thus integrating Mongo functionality into Python (using pymongo) a mongo script was created to aggregate the master collection based on product Ids and create two subcollections one for product reviews and the other one for the fixed time stamps. The following steps were then applied to all reviews on a per-product basis.

2. Preprocess (Change to lowercase, remove all punctuation marks and tokenize).
   Preprocessing is the first step for any Natural Language Processing System. All reviews for a product must be uniformly formatted because the goal is for a computer system to analyze text. Because of a computer's inability to comprehend the significance of punctuation marks and capitalizations and because of their low importance in opinionated texts, we remove them. Tokenizing ensures all sentences are processed as one row of a matrix with each word an element (and an assigned score in part 6).

3. Remove all stop words (as per the punkt corpus).
   Using one of the most developed stop words corpus in the world, DeMON removes all unnecessary tokens not adding any significance to the review. Common stop words include 'the', 'a', 'and', 'he' etc.

4. Use stem words for all remaining tokens
   All tokens are then modified to their root words. This is important for a better measure as instead of counting 'lovely' and 'loved' as different tokens, the TF-IDF will just count them both as 'love'.

5. Create a Term Frequency – Inverse Document Frequency vectorizer.
   A TF-IDF is a numerical statistic that is intended to reflect how important a word is to a document in a collection or corpus. It is often used as a weighting factor in information retrieval. The tf-idf value increases proportionally to the number of times a word appears in the document, but is offset by the frequency of the word in the corpus, which helps to adjust for the fact that some words appear more frequently in general like 'the','at' etc. With stop words removed, the TF-IDF vectorizer can focus more on important words.

6. Evaluate a sparse matrix with the TFS Scores using the vectorizer.
   The sparse matrix will contain score for each individual token depending upon it's importance in the English language corpus and it's frequency in each document. This matrix is the heart of DeMON. Not only can this matrix be used for any type of sentiment analyis, it can also

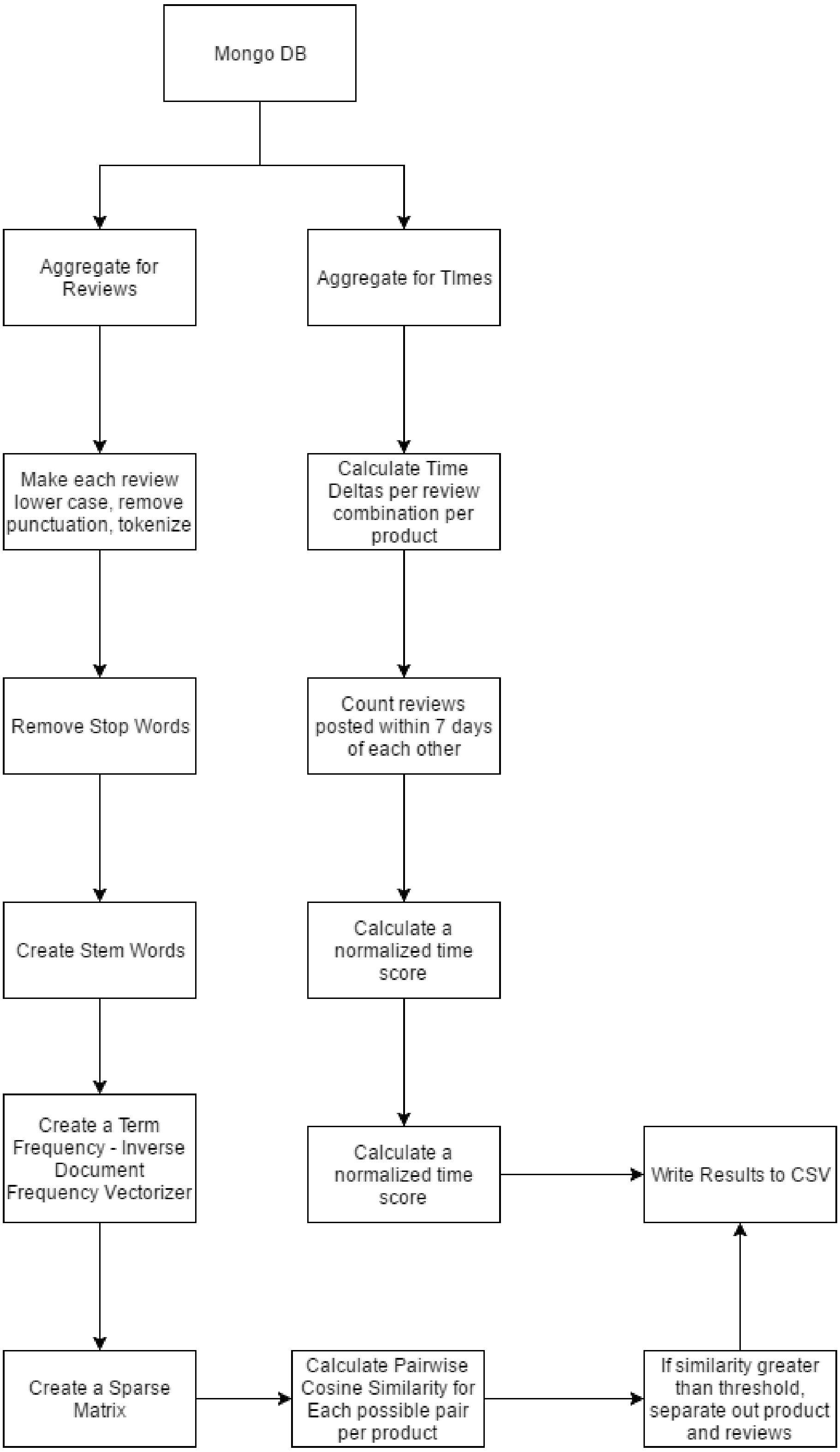

Mongo DB
Aggregate for Reviews
Aggregate for TImes
Make each review lower case, remove punctuation, tokenize
Calculate Time Deltas per review combination per product
Remove Stop Words
Count reviews posted within 7 days of each other
Create Stem Words
Calculate a normalized time score
Create a Term Frequency - Inverse Document Frequency Vectorizer
Calculate a normalized time score
Write Results to CSV
Create a Sparse Matrix
Calculate Pairwise Cosine Similarity for Each possible pair per product
If similarity greater than threshold, separate out product and reviews

Figure 4: Pre-Processing, Processing and Unloading flow diagram for Python

be used by a number of machine learning algorithms to learn a number of things, reviewer behavior amongst some of the interesting things.

7. Calculate Pairwise Cosine Similarity for the matrix.
   Using scores from the sparse matrix, review similarity is calculated for all reviews per product. Why cosine similarity helps evaluate similarity between documents, is mentioned in the mathematical model section.

8. Calculate Average Cosine Similarity of all reviews for each product.
   Once pairwise cosine similarity has been calculated, average cosine similarity for all reviews is calculated to get a complete picture of the cosine score of a product.

9. Evaluate if average and pairwise cosine similarity of any review is greater than a threshold and classify them.
   Both pairwise and average similarities are used in parallel at this step. If the average cosine similarity is greater than a threshold, it is marked as a 'fake' product in the first step. If pairwise similarity for some reviews is greater than a threshold, they are marked separately as potentials for fakes.

10. Write potential fake reviews, products with a majority of fake reviews, and average cosine score for each product in CSVs.
    These results are written out to a CSV for further processing.

Time Scoring

For the time analysis, a rather simple approach is followed:

1. Read the aggregated collection from MongoDB (DataSet Filtering has already taken place during loading).
   Using mongo's extensive functionality for manipulating collections and documents directly into Python (using pymongo) a mongo script was created to aggregate the master collection based on product Ids and create two subcollections one for product reviews and the other one for the fixed time stamps. The time stamps had been converted into Date Time objects from unix review time strings by the DeMON reader while loading the JSON into a mongo collection.

2. For all review date times, calculate time deltas.
   Per product, for all dates calculate a time delta matrix, to give an idea of when its reviews were posted.

3. Analyze time deltas to evaluate if there have been any peaks in review postings.
   For each time delta, look for anomalies and count all reviews posted within 7 days of each other.

4. Calculate a normalized score

$$Normalized\,Score = \frac{count\,from\,step\,3}{total\,number\,of\,reviews}$$

5. Output Score
   Write normalized score into a CSV for further/later processing.

6.2 R Processing and Visualization

This was just the first phase of processing. The output CSVs are next read into R for mathematical flexibility to use both time and cosine results in parallel to each other and output a list of products which are highly likely to be using fake reviewing services. Figure 5 shows the complete process the R model follows.

1. Generate a cos distance vs time plot
   The first step for R is to look at how reviews look like compared to time plots. For my dataset, DeMON created a Time Score vs Cos Distance plot as shown in Figure 6.

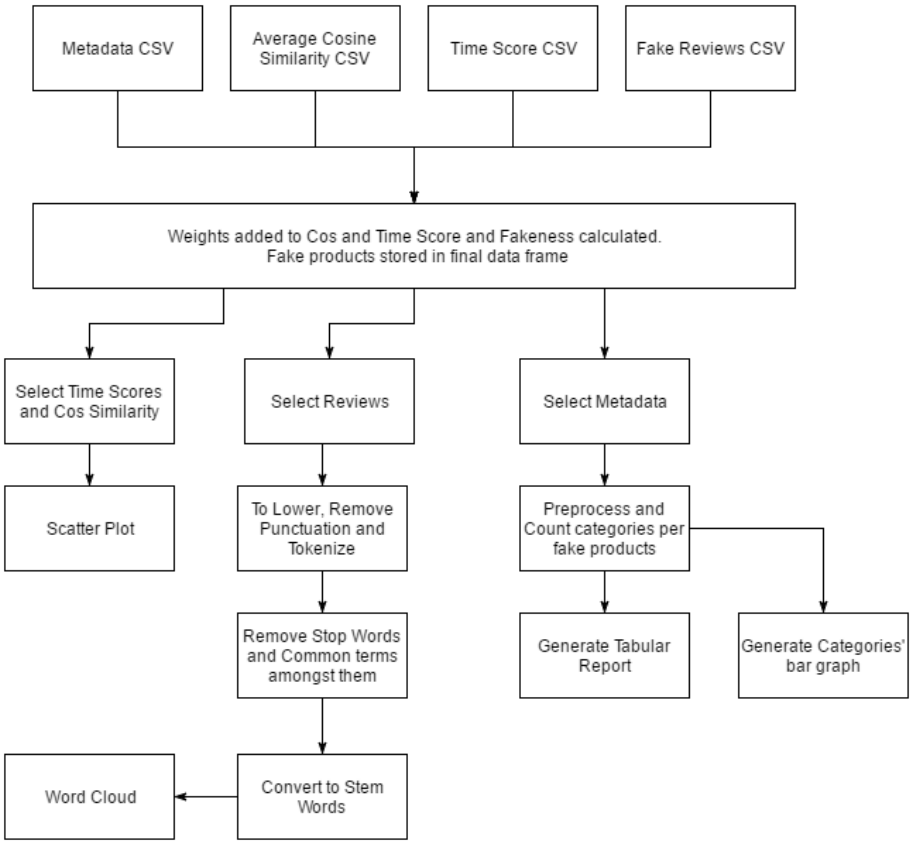

Metadata CSV
Average Cosine Similarity CSV
Time Score CSV
Fake Reviews CSV
Weights added to Cos and Time Score and Fakeness calculated. Fake products stored in final data frame
Select Time Scores and Cos Similarity
Select Reviews
Select Metadata
Scatter Plot
To Lower, Remove Punctuation and Tokenize
Preprocess and Count categories per fake products
Remove Stop Words and Common terms amongst them
Generate Tabular Report
Generate Categories' bar graph
Word Cloud
Convert to Stem Words

Figure 5: R Processing flow diagram

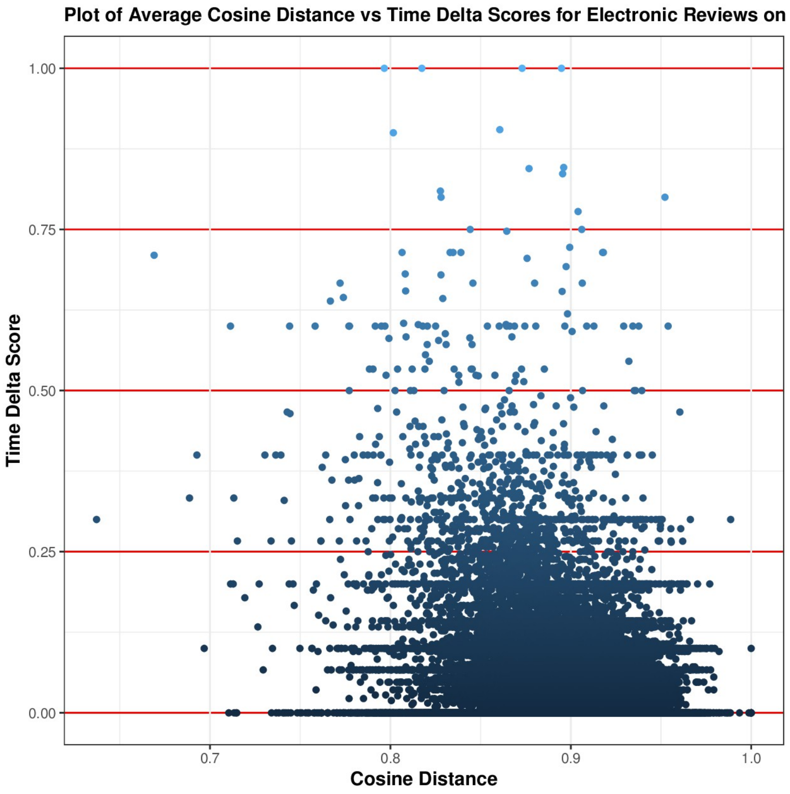


Figure 6: Cos Distance vs Time Score plot

Although a lot of products appear to be near 1 – meaning the reviews there are distant, some of the products especially near the 0.7 mark for cos sim and a higher time score than average are candidates for potential fakes. So the next phase is to identify these products.

2. Metadata Reading, Filtering and Preprocessing
   Reading Metadata from the generated CSV has a lot of noise in its various fields and is not usable for clustering based on product type. Therefore, it is first filtered to keep only product id, categories and product name. Next the noise in categories is reduced using preprocessing techniques again like removing stop words and common words such as 'electronics' from product categories.

3. Python Results into R
   Fake reviews separated out by the DeMON python engine are read in and time scores and cosine similarities for those reviews are concatenated to create a new data frame. This gives a bigger picture of each review.

4. Metadata merging with scores data frames
   The finalized data frame from Step 3 is merged with metadata based on product ids to create a final set of potential fake reviews, their cos scores, time scores, product name and product category.

5. Cos weight applied
   A weight is applied to the cos score as mentioned in the mathematical model. I used 0.25 as my default number.

6. Time weight applied
   A weight is applied to the time score. I chose 0.75 as my default weight.

7. Final fakeness calculated
   The two weights are added to give a measure of final fakeness and if it is greater than a threshold, the reviews are marked fake.

8. Report and bar graph generated
   Using the final data frame evaluated of fake products, a tabular report is generated along with a plotted graph for fake products based on their categories.

Addendum

9. After doing project 2 and learning more about R, I decided to add an extra feature into DeMON. The products which were marked as fake, their reviews can be used to study how fake reviews usually look like and can be used to generate a list of stop words to be removed for future sparse calculations. I created a word cloud for common terms in fake reviews to visualize them.

7 Validation

Once the csv has been generated from Cosine Results, the reviews can be manually inspected to validate whether they look similar to each other or not. My inspection showed me reviews such as the ones shown below.

> Two small similar reviews pointed out by DeMON:
>
> * works great with my cheapo 18-55 canon kit lens for shooting incredible macro photography.great build quality.
>
> * this mount adapter works great with my cheapo 18-55 canon kit lens for shooting incredible macro photography.great build quality

Now consider this large example, i have highlighted the similar parts:

> i bought this on sale at another store. this model is not a "smart" tv; the "smart" model is the vizio e320i-a0. i use a roku so don't need "smart". overall, i'm happy with the tv. has a good picture, ok sound & the connections i need. i use this as a bedroom tv & it works great for that. i do not have cable or satellite so i use antenna; i can see the city tower from my house so i have wonderful reception.i own 3 additional vizio tvs & they have been very good/dependable.**pros:-very good picture; the black levels are very good & the colors are accurate.-has digital audio out-3.5 mm analog audio out (side) - headphone/earphone jack*note, some consumers are saying this tv no longer has a headphone jack. mine does, but maybe vizio made a change & the tv no longer has the jack. go to vizio's website for specific spec information. there is a cheap & easy work-around if you only have rca jacks. buy this:2 x rca male, 1 x 3.5mm stereo female, y-cable 6-inchit will allow you to plug in your 3.5 mm headphones.-2 hdmi (1 side, 1 back)-1 usb (side)-slim frame design looks really nice & takes up less space horizontally/vertically. the tv is still kind of "thick" but that's because it has the leds mounted behind the screen not along the edge.-direct-lit/full-array led; not the ccfl edge-lit.-matte screen is pretty good in a bright room but not excellent; some minor reflections.-the 32-inch model has a plastic rectangular stand not the strange triangle/leaning version on the 29-inch tv.-audio is better than most tvs in this size/price. it has 2 -10w speakers. it's not audiophile quality but acceptable & better than many.-srs truvolume which evens out the spikes in volume. this used to be an issue with commercials but with the passage & implementation of the calm act (law passed by congress that makes it illegal for commercials to be broadcast at louder volume) that's less of an issue. it can still be helpful when streaming movies, etc.-ambient light sensor - i'm kind of indifferent to this feature.-simple remotecons:-unlit remote, small buttons-stand doesn't swivel-no "smart dimming" (vizio's name for local dimming)-60hz-720p-no dedicated pc connection but you can use hdmi with the proper cords.-no ethernet-guide - my old vizio's have the guide that shows what is currently on & what is coming up. this new vizio only shows what is currently on the channel. i really liked the old way because i could see what tv programs will be shown throughout the day.**fyi: the base/stand measurement is 15" x 7.5"vesa mount standard: 100mm x 100mm
>
> pros:-very good picture; the black levels are very good & the colors are accurate.-has digital audio out-3.5 mm analog audio out (side) - headphone/earphone jack*note, some consumers are saying this tv no longer has a headphone jack. mine does, but maybe vizio made a change & the tv no longer has the jack. go to vizio's website for specific spec information. there is a cheap & easy work-around if you only have rca jacks. buy this:2 x rca male, 1 x 3.5mm stereo female, y-cable 6-inch it will allow you to plug in your 3.5 mm headphones.-2 hdmi (1 side, 1 back)-1 usb (side)-slim frame design looks really nice & takes up less space horizontally/vertically. the tv is still kind of "thick" but that's because it has the leds mounted behind the screen not along the edge.-direct-lit/full-array led; not the ccfl edge-lit.-matte screen is pretty good in a bright room but not excellent; some minor reflections.-the 32-inch model has a plastic rectangular stand not the strange triangle/leaning version on the 29-inch tv.-audio is better than most tvs in this size/price. it has 2 -10w speakers. it's not audiophile quality but acceptable & better than many.-srs truvolume which evens out the spikes in volume. this used to be an issue with commercials but with the passage & implementation of the calm act (law passed by congress that makes it illegal for commercials to be broadcast at louder volume) that's less of an issue. it can still be helpful when streaming movies, etc.-ambient light sensor - i'm kind of indifferent to this feature.-simple remotecons:-unlit remote, small buttons-stand

Detector	Number of fakes identified
DeMON	35
ReviewSkeptic	39

Table 1: Validating DeMON using ReviewSkeptic using a sample set of 50 reviews

> doesn't swivel-no "smart dimming" (vizio's name for local dimming)-60hz-720p-no dedicated pc connection but you can use hdmi with the proper cords.-no ethernet-guide - my old vizio's have the guide that shows what is currently on & what is coming up. this new vizio only shows what is currently on the channel. i really liked the old way because i could see what tv programs will be shown throughout the day.i ordered it here at amazon, price is just amazing: http://amzn.to/13errkw

These are some perfect examples of potential fake reviews because of the obvious plagiarism. Their time stamp will be the second litmus test for them but for the second example for example it is pretty obvious that DeMON has indeed pinpointed a reviewer plagiarism other reviews.

DeMON's effectiveness was also validated and compared to an off the shelf Review Analyzer: ReviewSkeptic. A sample set of the same 50 reviews was given to both of them. The results are shown in Table 1.

8 Results

CSVs DeMON generates 4 CSVs: for time score, for average cosine similarity, for fake products and separated out potential fake reviews. 'Fake products' CSV is the one with products having an average cosine similarity greater than 0.7 as mentioned in the mathematical model's final fakeness equation. The average cosine and timescores are mathematical evaluations of each product while the potential fake reviews CSV contains all products and reviews where 2 reviews' pairwise cosine similarity > 0.5. This was added in DeMON to ensure that even if products don't have a majority of fake reviews (thus a low average) we still are able to identify individual mal-reviews.
For the Amazon Dataset, DeMON generated a total of 349 products with one or more similar reviews based on pairwise cosine similarity to be passed on for time analysis in R. A sigh of relief though, no product appeared as a fake out the dataset based on its average cosine similarity which would've been classified as availing fake reviewing services without a second test. All CSVs can be found in the DeMON>output folder.

Review and Time Dumps DeMON can also be used to create Review and Time Dumps for all products on Amazon. All reviews are saved in separate txt files in an individual folder per product (named after the Product ID). The Time Dump contains a series of txt files, one per product, with review times inside.
This was important to add, because for any analysis of Amazon products in DeMON or outside DeMON, unwrapping 1.6 million JSONs, which aren't even in any sequential order, everytime is time, effort and power consuming. These time and review dumps are created in the DeMON>data folder. I didn't upload them because of the extra space they consume, but the script can be run easily (see operating instructions).

Report Once R processing was complete, a total of 49 products were finally classified as availing fake services based on both their review similarity and the time scores. For them, a Report is generated presenting average cosine similarity, timescore and the fakeness score. The report also contains metadata for the products, their name and category on Amazon. A sample of the report generated is shown in Figure 7. The complete report can be found in DeMON>output folder.

Plots The identified products are also shown in a bar chart versus their categories on Amazon. The final bar chart plot obtained is presented in the Figure 8. In total, after all processing, a total

of 49 products were identified as having fake reviews (mentioned in the report), the bar char shows their breakdown. I must clarify here, that because Amazon discriminates between all three categories, audio, speakers and headphones, DeMON plotted them under different headings.
A Time Score vs Cos Distance plot was constructed in Figure 6 to show how the scores and review distances looked like. After initial visualization, it was used to set DeMON's R processing engine's set points for various thresholds. An analysis for it is also presented in the next section.

Word Cloud (Addendum) After completing project 2 and exploring R in depth, I decided to include another special feature in DeMON. A word cloud was generated for reviews of all products marked with a high fakeness score. This can help understand patterns in fake reviews and can be used to identify many others which were written more carefully and remained obscured. The word cloud can also be used to study terms which are commonly used by reviewers and thereby can be added as stop words for improving the efficiency and results of any sentiment analysis run on reviews. This is some future work I'm planning to undertake, because of the unavailability of a proper stop word corpus for product reviews. I want to create one now and publish it for other researchers in the same domain. The word cloud is shown in Figure 9.

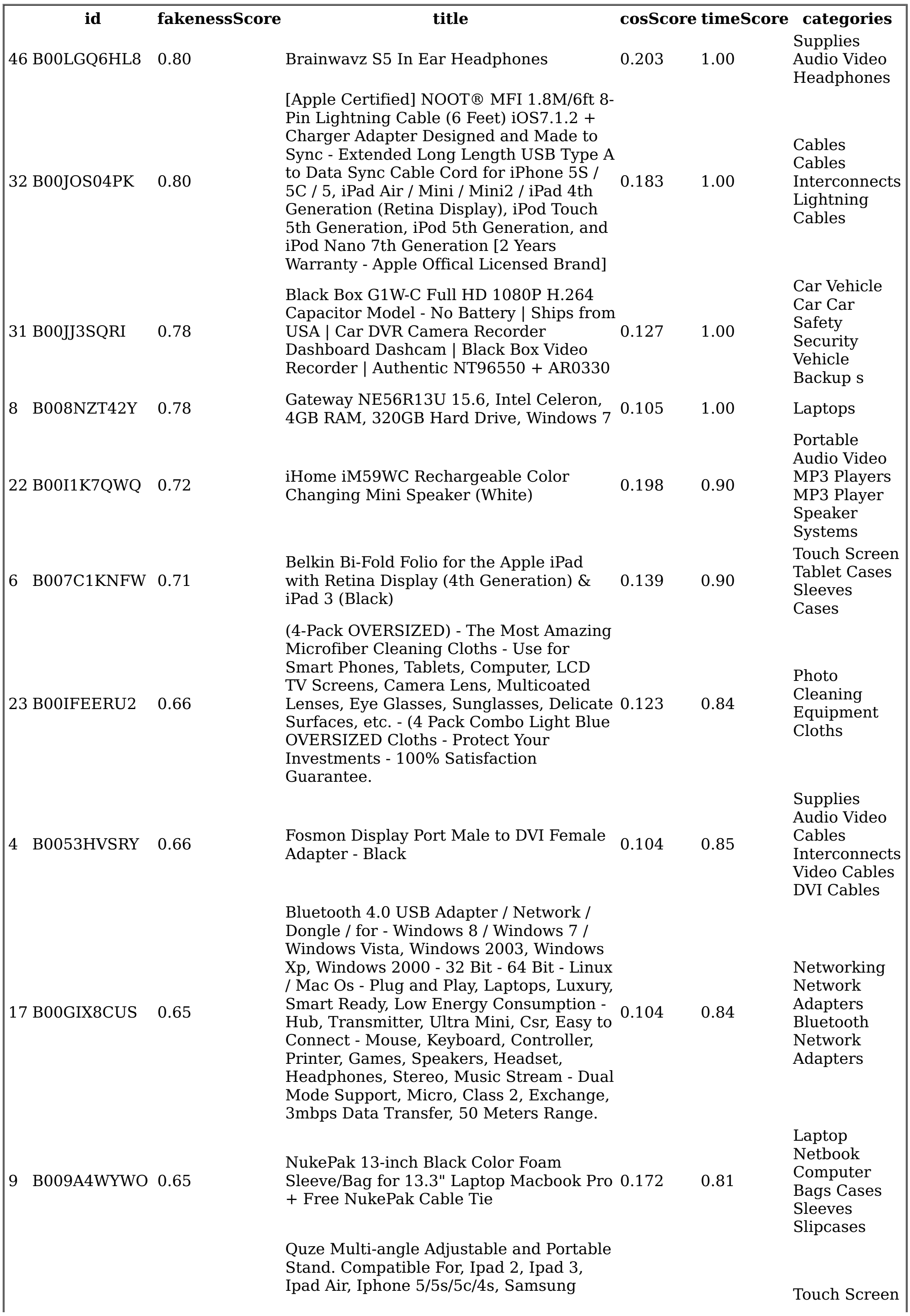
9 Analysis

Some very interesting observations came out from the generated CSVs, reports and charts. Lets consider each one individually.

CSVs It was a sigh of relief when I opened the Fake-Products CSV (should contain products with an average cosine similarity > 0.7) and it was empty. This ensures one can put faith in a majority of Amazon's reviews if not all, and that Amazon and it's mechanisms for dealing with reviews and reviewers are pretty potent. This is commented upon in further detail using evidence from the Time Scores and Cosine Similarity Scores CSVs under the Plots Analysis.
The 349 products in Fake-Reviews CSV were inspected, and because the duplicate reviews had already been removed in the data set by Julian McAuley [6][7], some plagiarism could be seen through the naked eye. I have given two examples from the CSV in the Validation section.

Report Figure 7 shows a sample of the output report of products which were identified as having faked reviews. It should be noted here that the cosScore shown is the average cos score, and so even though as a whole these products don't appear malicious, they're in the list because of their pairwise cosine similarity results. This means that even though a large number of their reviews weren't similar the ones that were, were definitely plagiarized.
It is interesting to note that almost all these products are of not very highly recognizable companies. Some of them like Quze or NukePak don't even have their own websites. So a lot of companies that do take part in faking reviews do so knowing their isn't much to lose.
Another interesting thing to note is that, out of 1.6 million reviews for 63001 products, only a handful of 49 products were pinpointed with fake reviews. This speaks volumes on behalf of Amazon's claims about their mechanism for handling fake reviews effectively.

Plots The bar chart in Figure 8 is a good depiction of which categories are most targeting by reviewing agencies. A lot of the fake reviews detected we part of accessories. This goes hand in hand with the observation from the report that companies with a good reputation, don't need to pay for reviews. Looking at the complete picture here, accessories are where small companies looking for profit operate. This is because people go for electronics from established companies, but because of the variety of accessories provided by smaller companies, customers tend to shop for accessories here.
Another item with a huge market in America are audio devices: be it speakers or headphones or just sound systems. The number of fake reviews in this category is proportional to market's size, and it makes sense because of companies competing day and night and by hook or by crook to create a name for themselves.
Finally, lets revisit the Cos Distance vs Time Score plot in Figure 6. This is probably one of the most important plots in DeMON. It shows all the current 63001 products in the Dataset Map on a scatter plot. A lot of the reviews fall in the fourth quadrant of the plot, which is extremely good.



	id	fakenessScore	title	cosScore	timeScore	categories
46	B00LGQ6HL8	0.80	Brainwavz S5 In Ear Headphones	0.203	1.00	Supplies Audio Video Headphones
32	B00JOS04PK	0.80	[Apple Certified] NOOT® MFI 1.8M/6ft 8-Pin Lightning Cable (6 Feet) iOS7.1.2 + Charger Adapter Designed and Made to Sync - Extended Long Length USB Type A to Data Sync Cable Cord for iPhone 5S / 5C / 5, iPad Air / Mini / Mini2 / iPad 4th Generation (Retina Display), iPod Touch 5th Generation, iPod 5th Generation, and iPod Nano 7th Generation [2 Years Warranty - Apple Offical Licensed Brand]	0.183	1.00	Cables Cables Interconnects Lightning Cables
31	B00JJ3SQRI	0.78	Black Box G1W-C Full HD 1080P H.264 Capacitor Model - No Battery | Ships from USA | Car DVR Camera Recorder Dashboard Dashcam | Black Box Video Recorder | Authentic NT96550 + AR0330	0.127	1.00	Car Vehicle Car Car Safety Security Vehicle Backup s
8	B008NZT42Y	0.78	Gateway NE56R13U 15.6, Intel Celeron, 4GB RAM, 320GB Hard Drive, Windows 7	0.105	1.00	Laptops
22	B00I1K7QWQ	0.72	iHome iM59WC Rechargeable Color Changing Mini Speaker (White)	0.198	0.90	Portable Audio Video MP3 Players MP3 Player Speaker Systems
6	B007C1KNFW	0.71	Belkin Bi-Fold Folio for the Apple iPad with Retina Display (4th Generation) & iPad 3 (Black)	0.139	0.90	Touch Screen Tablet Cases Sleeves Cases
23	B00IFEERU2	0.66	(4-Pack OVERSIZED) - The Most Amazing Microfiber Cleaning Cloths - Use for Smart Phones, Tablets, Computer, LCD TV Screens, Camera Lens, Multicoated Lenses, Eye Glasses, Sunglasses, Delicate Surfaces, etc. - (4 Pack Combo Light Blue OVERSIZED Cloths - Protect Your Investments - 100% Satisfaction Guarantee.	0.123	0.84	Photo Cleaning Equipment Cloths
4	B0053HVSRY	0.66	Fosmon Display Port Male to DVI Female Adapter - Black	0.104	0.85	Supplies Audio Video Cables Interconnects Video Cables DVI Cables
17	B00GIX8CUS	0.65	Bluetooth 4.0 USB Adapter / Network / Dongle / for - Windows 8 / Windows 7 / Windows Vista, Windows 2003, Windows Xp, Windows 2000 - 32 Bit - 64 Bit - Linux / Mac Os - Plug and Play, Laptops, Luxury, Smart Ready, Low Energy Consumption - Hub, Transmitter, Ultra Mini, Csr, Easy to Connect - Mouse, Keyboard, Controller, Printer, Games, Speakers, Headset, Headphones, Stereo, Music Stream - Dual Mode Support, Micro, Class 2, Exchange, 3mbps Data Transfer, 50 Meters Range.	0.104	0.84	Networking Network Adapters Bluetooth Network Adapters
9	B009A4WYWO	0.65	NukePak 13-inch Black Color Foam Sleeve/Bag for 13.3" Laptop Macbook Pro + Free NukePak Cable Tie	0.172	0.81	Laptop Netbook Computer Bags Cases Sleeves Slipcases
			Quze Multi-angle Adjustable and Portable Stand. Compatible For, Ipad 2, Ipad 3, Ipad Air, Iphone 5/5s/5c/4s, Samsung			Touch Screen

Figure 7: Sample of Report generated containing fake product's details

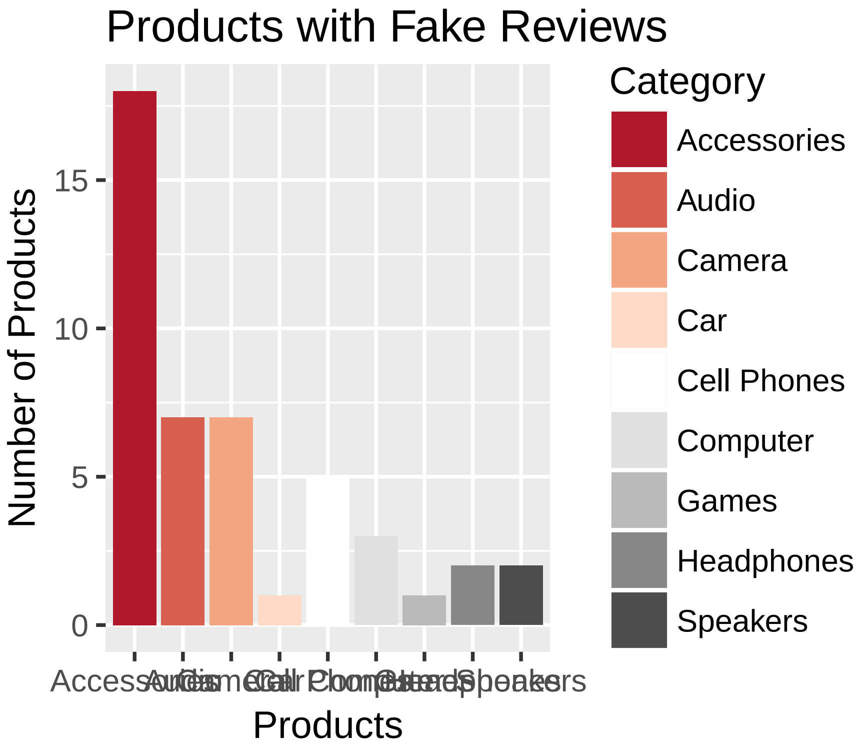


Figure 8: Bar Graph of no. of fake reviews per category

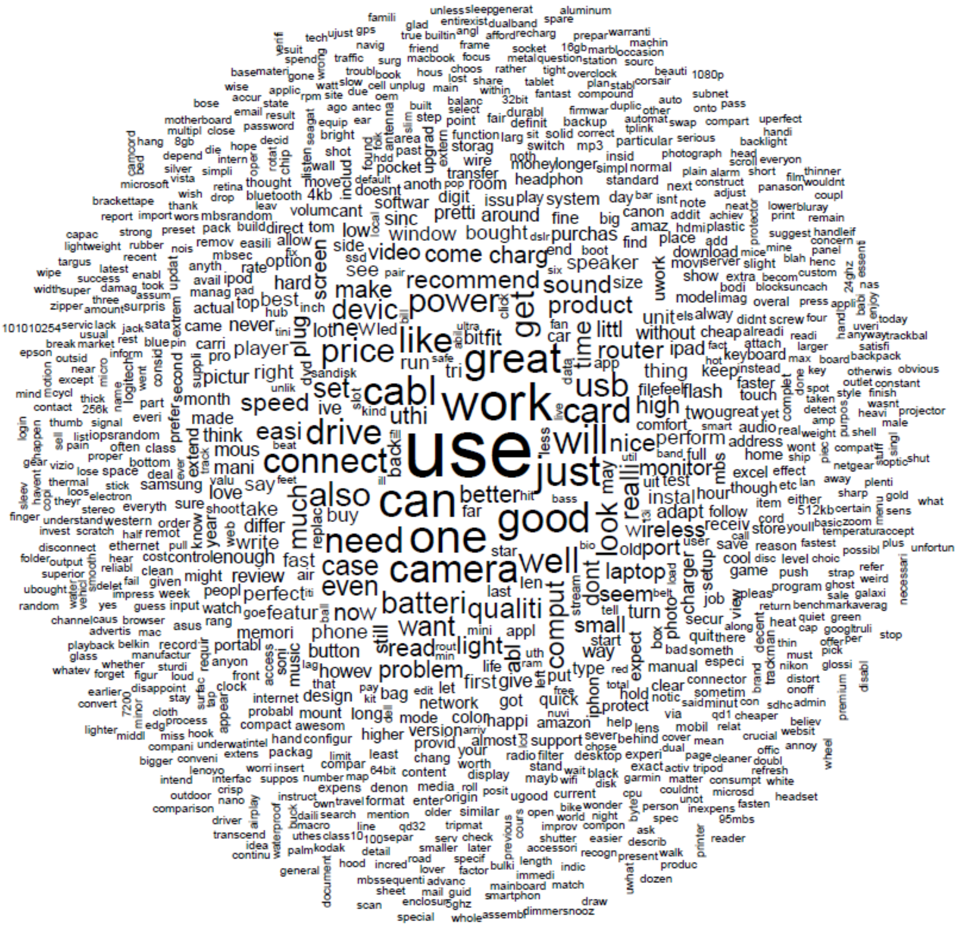


Figure 9: Wordcloud of common terms in fake reviews

Those products have evenly distributed in time and low in similarity reviews and thus are reliable sources for making decisions. The products of concern are in the second and third quadrants. This is because these either are highly similar to each other or there are abrupt peaks in their postings. These were the 349 products considered for further analysis to determine Fakeness. Then again, considering the large number of reviews not lying in these quadrants, Amazon maintains it's credibility.

Word Cloud (Addendum) Observing the word cloud, we can see a lot of common English terms with the word 'use' as the highest used term. Others include 'work', 'price', 'one' etc. Such terms can be used as stop words for review processing for any sentiment analysis. However there are some terms in there like 'good' and 'well' which can't be avoided because of their use in polarity analysis. Therefore it is a separate task filtering out terms passing out little value judgments. These terms however were also created using the final fake reviews of 49 products. Therefore, they can be used to study plans that fake reviewing agencies follow. For e.g a majority of these terms are positive. This says that companies try boosting up their own products rather than bringing other companies' products down - and this is an important observation describing various work patterns of these reviewing agencies and companies hiring them.

10 Operating Instructions

A complete set of step by step operating instructions have been provided in the project directory inside Instructions.txt.

11 Performance Objectives

The performance objectives for DeMON were to successfully categorize products with fake reviews alongside giving an overview of how reliable Amazon product reviews are. This goes hand in hand with analyzing the effectiveness of Amazon's current techniques of spotting, deleting and merging reviews. My research has indicated (Amazon doesn't clearly disclose their fake review detection system unlike Yelp!) that Amazon merges reviews for similar looking products automatically and cross-references them to check for any anomalies. Also, because it keeps an active check on how users are interacting with products, it becomes easier to identify which accounts are active reviewers and it tries to spot irregularities amongst them.

12 Performance Benchmarks & Metrics

Because professional fake review detectors boast a performance of 60% accuracy, I would consider DeMON to be anywhere close to that number, victorious. I used ReviewSkeptic to validate DeMON and the with it successfully classifying 78% of the reviews as fake, and DeMON not far behind at 70%, I consider DeMON to be a first leap in the right direction.

13 Future Work

As mentioned, DeMON is the first leap in the right direction. There are a number of more metrics added to the mathematical model to improve this base one. Like analysis on postings on reviewers, anomalies between reviews by the same reviewer (e.g. the same reviewer using multiple genders) or looking at products and their company profiles and analyzing the trends there. The boundaries to explore are limitless and any can be used in the future to further improve this Opinion Miner.
Another potential thing coming out of DeMON, is the development of a proper stop words corpus for reviews. Work is already underway on this because it is extremely essential for any sentiment analysis on user product reviews in the future.
Another aspect to look to is adding Machine Learning to DeMON and making it think for itself for every new review. This is something I am definitely going to explore during my Machine Learning Course.

14 Conclusion

This is the generation of E-Commerce, and with Amazon being the second biggest electronics retailer in the world [4]. For any one who shops online, user reviews are the first thing he/she looks at to judge a product. Thus these reviews have the power to mould anyone's thinking and either turning that into profit for a company or not. Companies, as a result, have to take user feedbacks really seriously. Establishing the importance of these reviews, and low-end companies going by hook or by crook to increasing their sales, it is valuable to have a check of whether the opinion being read has a genuinity to it or not.
DeMon analysed Amazon's product reviews for electronics to evaluate whether Amazon's claims for removing fake reviews on their websites have some substance to them. Out of the 630001 product with 1.69 million reviews, DeMON could short list 49 products as having fake reviews. This is not just a win for Amazon for having such a small number of them but is a win for DeMON as well as it was able to go into such fineness and extract out the smallest of gray areas left out by Amazon's handlers. Validating DeMON's model with one of the best off the shelf review detectors, DeMON is just 8% behind it and already 10% better than the average market.

I would like to acknowledge the efforts of Dr. Raman Kannan (NYU) for his constant motivation, as well as Dr. Julian McAuley (UCDavis) for wholeheartedly giving access to his datasets. I also acknowledge Chen Wen (Columbia) for his help in pointing me in the right direction when I needed it at the very start and IBM for providing a platform which kept my interest in Big Data and Cloud Computing invigorated.

References

[1] Amazon Press Release. 2016. *https://www.amazon.com/p/feature/abpto3jt7fhb5oc*

[2] Manning, Christopher. 2009. *Introduction to Information Retrieval. http://nlp.stanford.edu/IR-book/pdf/irbookonlinereading.pdf.*

[3] Liu, Bing. 2012. *Sentiment Analysis and Opinion Mining. https://www.cs.uic.edu/ liub/FBS/SentimentAnalysis-and-OpinionMining.html.*

[4] Amazon Passes Walmart As No. 2 In Electronics, *http://www.twice.com/news/top-100twice-research/amazon-passes-walmart-no-2-electronics/61632*

[5] Mukherjee, Arjun, 2011. *http://www2.cs.uh.edu/ arjun/papers/op_ spam_ acl_ 15_ tutorial.pdf.*

[6] Image-based recommendations on styles and substitutes. *J. McAuley, C. Targett, J. Shi, A. van den Hengel.* SIGIR, 2015.

[7] Inferring networks of substitutable and complementary products. *J. McAuley, R. Pandey, J. Leskovec.* Knowledge Discovery and Data Mining, 2015.