# Designing and Building an Optimal Review Ranking System for Amazon

Mike Cai | Yilin Huang | Haoxiang Zhang | Ivan Wang

# 00

# Introduction

# Introduction

## 01

### Background & Motivation

- **User-generated reviews are crucial to shoppers.**
- **The sheer volume of user generated review** make it harder for shoppers to finding the right information
- An objective ranking method to show **the most relevant information would provide immense value**.
- Significant value in researching and developing a ranking mechanism that could **filter out inaccurate information** and **return the most objective** ones a customer would want.

## 02

### Research Questions

- What is a high quality review?
- What **factors** can help us determine the quality of a review?
- How does **reviewers' credibility** affects their reviews?
- What standards constitute a review as **helpful and valuable**?
- Why are those factors and standards important for a review to have?

# Data Set and Variables

- The dataset we chose is Amazon review dataset 2018. It includes all review data in Amazon from May, 1996 to October, 2018. The raw review data has 233.1 million reviews, which has a size of 34GB.

- The subset of the metadata we selected is the **per-category data on musical instruments department**
    - **Five-core data**
    - contains 231,392 reviews
    - relatively compact size


```
{
  "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"
}
```


## Data format is one-review per line in json

1. **reviewerID** - ID of the reviewer
2. **asin** - ID of the product
3. **reviewerName** - name of the reviewer
4. **vote** - helpful votes of the review
5. **style** - a dictionary of the product metadata
6. **reviewText** - text of the review
7. **overall** - rating of the product
8. **summary** - summary of the review
9. **unixReviewTime** - time of the review (UNIX time)
10. **reviewTime** - time of the review (raw)

# Challenges

## Feature Selection and Engineering

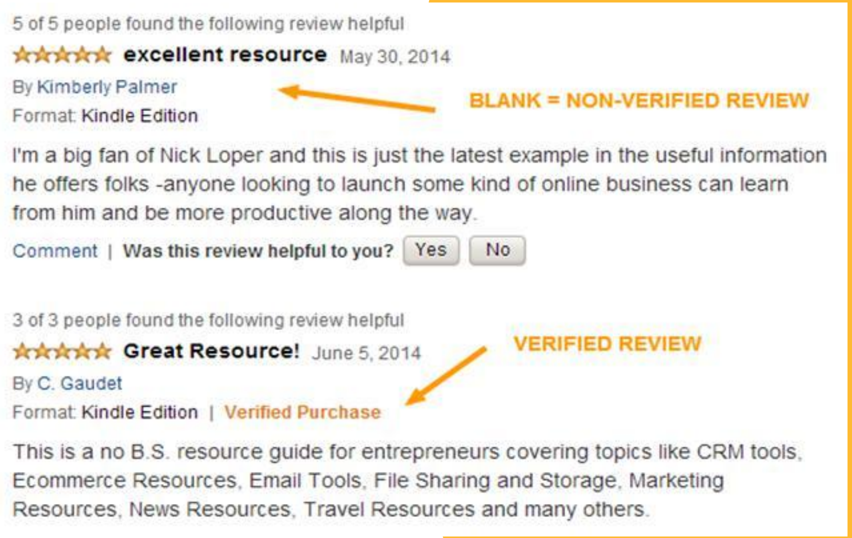
Need to determine what features are the most important for our ranking algorithm and what should be left out.

## Lack of User Information

Without using reviewers' private information, we will need to discover a method to **quantify their credibility** with the current data we have obtained, and create the **query information** to rank the reviews as documents

## Balancing Trade-offs

We would have to decide how to use the different metrics that we have determined to be important and how to **assign weights** to each one of the factors in our ranking algorithm together.



01

# Related Work

# Related Work

## Past Works

- two ranking mechanisms for user-generated product reviews: **a consumer-oriented mechanism** that would rank the reviews according to their expected helpfulness, and **a manufacturer-oriented mechanism** that would rank them according to their expected effect on sales.
- using a regression model that can predict their helpfulness scores. Features that are selected into the model are extracted from **review text, product description, and customer Q-and-A data**. The study **excluded the low quality reviews** out since they would have low predicted helpfulness scores and would not be included in the top k reviews list.

## How our design differs:

- Our mechanisms will be **consumer oriented mechanism**. When ranking a product review, we will we take **the quality of the review** and the **credibility of the review poster** into consideration. Taking such aspect into consideration allows us to better **filter out fake reviews and counterfeiters**.
- The features that we selected are more **objective** and **extracted from the reviews themselves**. Features in Saumya et al.'s work included information from customer Q-and-A data, which can be subjective.

# Method Roadmap

## Building Inverted Index

- Data Cleaning
- Build product indices
- Build user indices

## Parameter Extraction

- Product Selection
- Web Scraping
- Feature Engineering
- Exploratory Analysis

## Coefficient Determination

- Regression Techniques
- Importance Weighting


# 02

# Preliminary Work

# Building Inverted Index

## User Indices

- This is the index that we used to extract user information
- Built from the whole dataset
- Bag of words: user ID
- All parameters of a review document beside *reviewText* and *summary* were added to the index as fields.

## Review Document Indices

- These are the review documents that we are ranking.
- Built from the review data of a specific product
- Bag of words: Made by tokenize concatenation of *reviewText* and *summary*.
- All parameters of a review document were added to the index as fields.

## Code Snippet

```java
for(i=0; i<corpus.getCorpusSize(); i++) {
    ReviewDoc review = corpus.getDoc(i);

    Document doc = new Document();
    doc.add(new Field( name: "content",  value: review.getReviewText()+" "+review.getSummary(), _contentFieldType));/
    doc.add(new Field( name: "reviewerID", review.getReviewerID(), _contentFieldType));//the author field is search
    doc.add(new Field( name: "asin", review.getAsin(), _contentFieldType));//the author field is searchable
    doc.add(new Field( name: "reviewText", review.getReviewText(), _contentFieldType));//the content field is searc
    doc.add(new Field( name: "summary", review.getSummary(), _contentFieldType));//the content field is searchable
    doc.add(new Field( name: "image",  value: ""+review.getImage(), _contentFieldType));//the content field is search
    doc.add(new Field( name: "overall",  value: ""+review.getOverall(), _contentFieldType));//the content field is se
    doc.add(new Field( name: "vote", review.getVote(), _contentFieldType));//the content field is searchable
    doc.add(new Field( name: "verified",  value: ""+review.isVerified(), _contentFieldType));//the content field is s
    doc.add(new Field( name: "reviewTime",  value: ""+review.getUnixreviewtime(), _contentFieldType));//the content f
    doc.add(new Field( name: "reviewerName",  value: ""+review.getReviewerName(), _contentFieldType));//the content f

    writer.addDocument(doc);

    if (i % 1000 == 0)
        System.out.println(" -> indexed " + i + " docs...");
}
System.out.println(" -> indexed " + i + " total docs.");
```

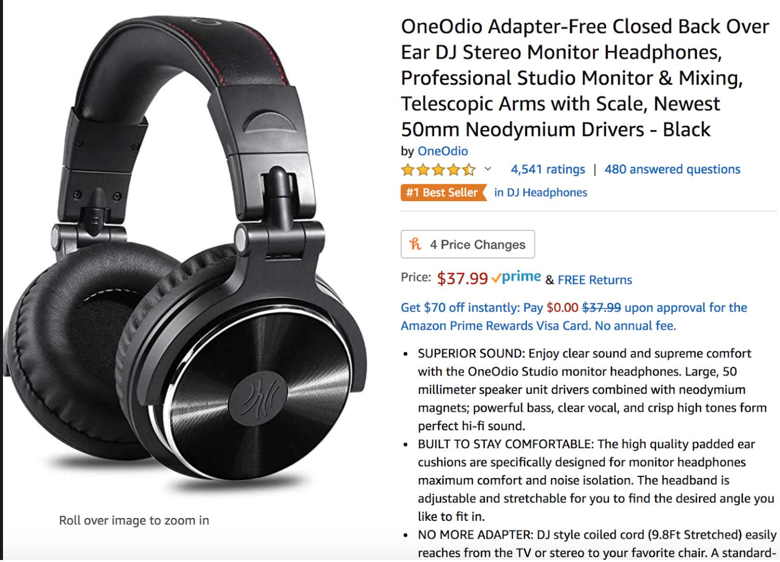
...

# Product Review Selection

We selected the **#1 best seller DJ headphone**

- more than **4000 shopper's reviews**
- **average rating of 4.7/5**
- abundance of its **UGR contents**

Product Webpage




# Web Scraping Data

- Crawled the **top 800 reviews**
- **Top 20 % of the total review**
- Reranked the order based
- Dropped unnecessary column
  - **Author, url, next,** etc...

| | web-scraper-order | web-scraper-start-url | author | title | date | content | rating | next | next-href | verified | vote |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **30** | 1587446039-1441 | https://www.amazon.com/OneOdio-Adapter-Free-He... | Jet Mech 1 | Super "bang-for-the buck" | Reviewed in the United States on June 4, 2018 | This is an initial out-of-the box review:\nFir... | 5.0 out of 5 stars | NaN | NaN | Verified Purchase | 940 people found this helpful |
| **731** | 1587446039-1442 | https://www.amazon.com/OneOdio-Adapter-Free-He... | Sean | Quality, High Fidelity, Comfortable, and a pri... | Reviewed in the United States on July 31, 2018 | I bought these last year and let them sit on m... | 5.0 out of 5 stars | NaN | NaN | Verified Purchase | 315 people found this helpful |
| **716** | 1587446039-1443 | https://www.amazon.com/OneOdio-Adapter-Free-He... | David Diamond | Clean, full sound with hearing compensation, a... | Reviewed in the United States on October 8, 2019 | The OneOdio headphones have a very clean sound... | 5.0 out of 5 stars | NaN | NaN | Verified Purchase | 176 people found this helpful |
| **643** | 1587446039-1444 | https://www.amazon.com/OneOdio-Adapter-Free-He... | Ricardo Mera | Nice Product Awesome. | Reviewed in the United States on December 8, 2017 | Beginning with the sound it is spectacular the... | 5.0 out of 5 stars | NaN | NaN | Verified Purchase | 137 people found this helpful |

# Feature Engineering

- Raw data from the scrapped *JSON* file
- Require **data cleaning**
- Perform **feature extraction** to provide analytical value

**1. Combining Review Text:**

- Add **review title** and **body**
- Integrate the query on each review text together

```
review = list()
for i in range(len(title)):
  review.append(title[i] + " " + content[i])
```

**2. Convert Date Format**

- Striping out the substring that contain the **date**
- Convert it to the **datetime format**

```
time = list()
for i in range(len(dates)):
time.append(dates[i].replace('Reviewed in the
United States on ', ''))
new_date = list()
input_format = "%B %d, %Y"
output_format = "%Y, %m, %d"
for t in time:
nd = datetime.strptime(t, input_format).
strftime(output_format)
new_date.append(nd)
# converting string to datetime object
dt = [datetime.strptime(x, output_format)
    for x in new_date]
```

- ***"Reviewed in the United States on June 4, 2018"*** **→ "2018, 06, 04"**
- Subtract from **current day**
- Get how many **days have passed** since the posting
- Apply in the scoring model with ***timepast*** (month)

**3. Review Related Features:**

Three features were extracted:

- ***reviewlength*: amount of information captured**
- ***reviewsentiment*: neutrality degree of the tone**
- ***reviewsubjectivity*: level of objectivity**

```
from textblob import TextBlob
review_polarity = list()
review_subjectivity = list()
for r in review:
  blob = TextBlob(r)
  sentiment = (blob.sentiment.polarity+1)/2
  subjectivity = blob.sentiment.subjectivity
  review_polarity.append(sentiment)
  review_subjectivity.append(subjectivity)
len(review_polarity), len(review_subjectivity)
```

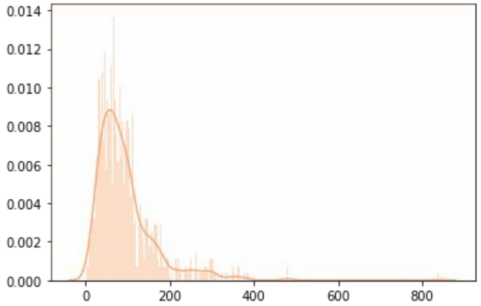
**4. Helpful Vote:**

- Apply the same string operation on ***date*** on ***vote***
- Only the numerical part of the **"6 people found this helpful"** → "6"

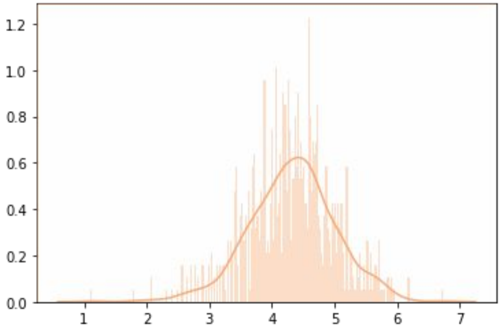
# Exploratory Analysis

## 1. Review Length

- distribution heavily **right-skewed**



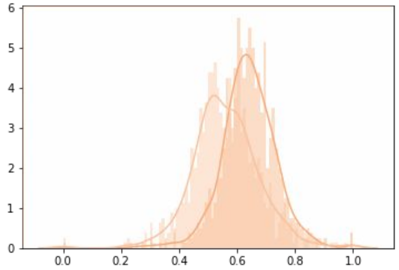

- diminishing **marginal return of value gain, normalized feature**




## 2. Sentiments

From the processed data, we have two review sentiment:

- **Polarity**:
  - Positive
  - Neutral
  - Negative
- **Subjectivity**: degree of objectiveness




We **re-scaled** this ***polarity*** index by shifted its value up 1 unit and then divided it by 2 so that the new index would **range from 0** (most negative) **to 1** (most positive), matching that of the ***subjectivity*** index.

## 3. Review Duration

- Many top ranked reviews were **posted a year or two ago**
- **Outdated reviews** cannot represent current condition

## 4. Helpful Votes

- Current algorithm heavily weighs on the number of *votes*
- will bury high quality reviews that were posted recently.
- The top 50 votes accounting for 77.108 percent of the total helpful vote counts.
- This indicates that Amazon place a high emphasis on it, but we think that it is an overkill.

# How Amazon Rank it?

**By looking into the top 50 reviews for this product, we discovered some potential patterns and trends of the Amazon ranking system.**

**The Longer the Better**

- Amazon place a heavy emphasis on **longer review than shorter ones**
- **10 out of the top 50 reviews** have abnormally short review length in comparison to their neighbors.

**Positive Ones Preferred**

- 45 of the top 50 reviews have **higher than average sentiment score** in terms of review polarity
- The **top 10 reviews** have **significantly higher** polarity

# Dive into an Example

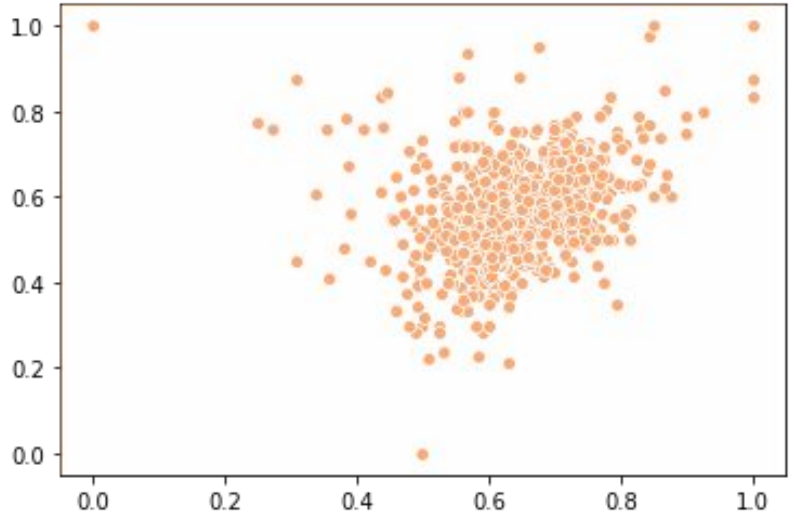
**Sample Review:** *"Nice Product Awesome. Beginning with the sound it is spectacular the levels of low and treble that allows you to use it is spectacular, its construction is of very good quality, it comes with two cables of very good aspect and quality. It is a product that gives a very good benefit for its coast"*

**Analysis:**

- ranked 4th overall
- only contains 55 words in length
- lacking in contextual information
- Over use of positive nouns and adj.
- Wordiness and Spelling Error
- Sentiment score of 0.82, which is 2 standard deviation above the mean

**Implication:**

1. Ranked higher than other reviews that are more comprehensive and informative
2. We hypothesize that this mechanism is intentionally designed to drive product sale
3. Correlation analysis of the rank score and polarity index validates our hypothesis



# 03

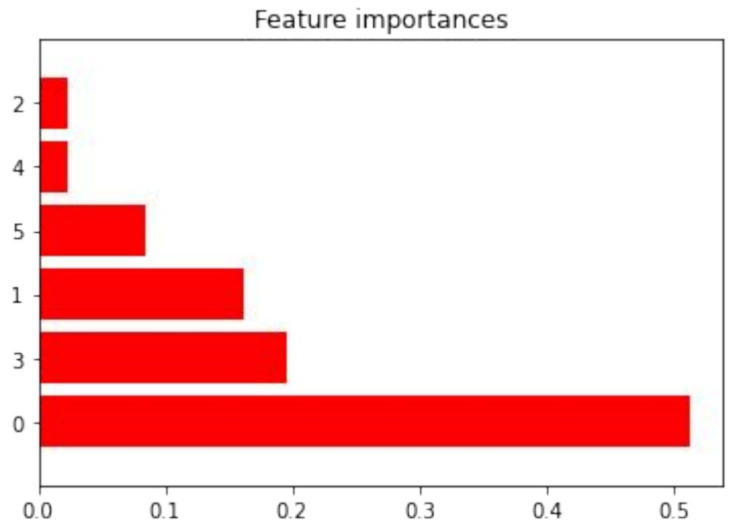
# Algorithm

# Benchmarking Parameter - OLS

| | coef | std err | t | P>\|t\| | [0.025 | 0.975] |
|---|---|---|---|---|---|---|
| **x1** | 0.0066 | 0.000 | 13.822 | 0.000 | 0.006 | 0.008 |
| **x2** | 0.7198 | 0.400 | 1.798 | 0.072 | -0.066 | 1.505 |
| **x3** | -0.1657 | 0.289 | -0.573 | 0.567 | -0.733 | 0.402 |
| **x4** | 0.0001 | 0.000 | 0.935 | 0.350 | -0.000 | 0.000 |
| **x5** | -0.0558 | 0.029 | -1.894 | 0.059 | -0.114 | 0.002 |
| **x6** | -0.0004 | 0.001 | -0.447 | 0.655 | -0.002 | 0.001 |
| **x7** | 4.8589 | 0.243 | 19.965 | 0.000 | 4.381 | 5.337 |

**x1: Review Length**
**x2: Polarity**
x3: Subjectivity
x4: Duration of Review
**x5: Rating**
x6: Helpful Votes
**x7: Verified**

# Benchmarking Parameter - Random Forest




**0: Review Length**
**1: Polarity**
2: Subjectivity
**3: Duration of Review**
4: Rating
5: Helpful Vote

# Building the scoring model

## Intrinsic Parameters

- **directly derived from a piece of review text itself.**
  - Review length
  - Number of Image
  - Sentiment
  - Helpful votes
  - Relevancy

## Extrinsic Parameters

- **affect the quality of a review base on factors outside of the review text or query matching.**
  - Image
  - Verification
  - User credibility

## Time Decay Parameter

- Duration of Review

# Scoring Model

$$Score = DoR * Vrf * (Crd * (w_1 * Rl + w_2 * St + w_3 * Vt + w_4 * Rel) + Img)$$

***DoR*** **(duration of review)** = $\frac{1}{log2(MonthsElapsed+2)}$

***Vrf*** **(verified purchase)** = $\begin{cases} 1, \text{ verified} \\ 0.1, \text{ not verified} \end{cases}$

***Crd*** **(credibility of user)** = 1+log(1+log(1+reviewerAverageVote))

***Rl*** **(relative review length)** = $\frac{log(reviewLength+1)}{log(averageReviewLength+2)}$

***St*** **(Sentiment score)** = $\frac{2}{1+|sentiment - AverageSentiment|}$

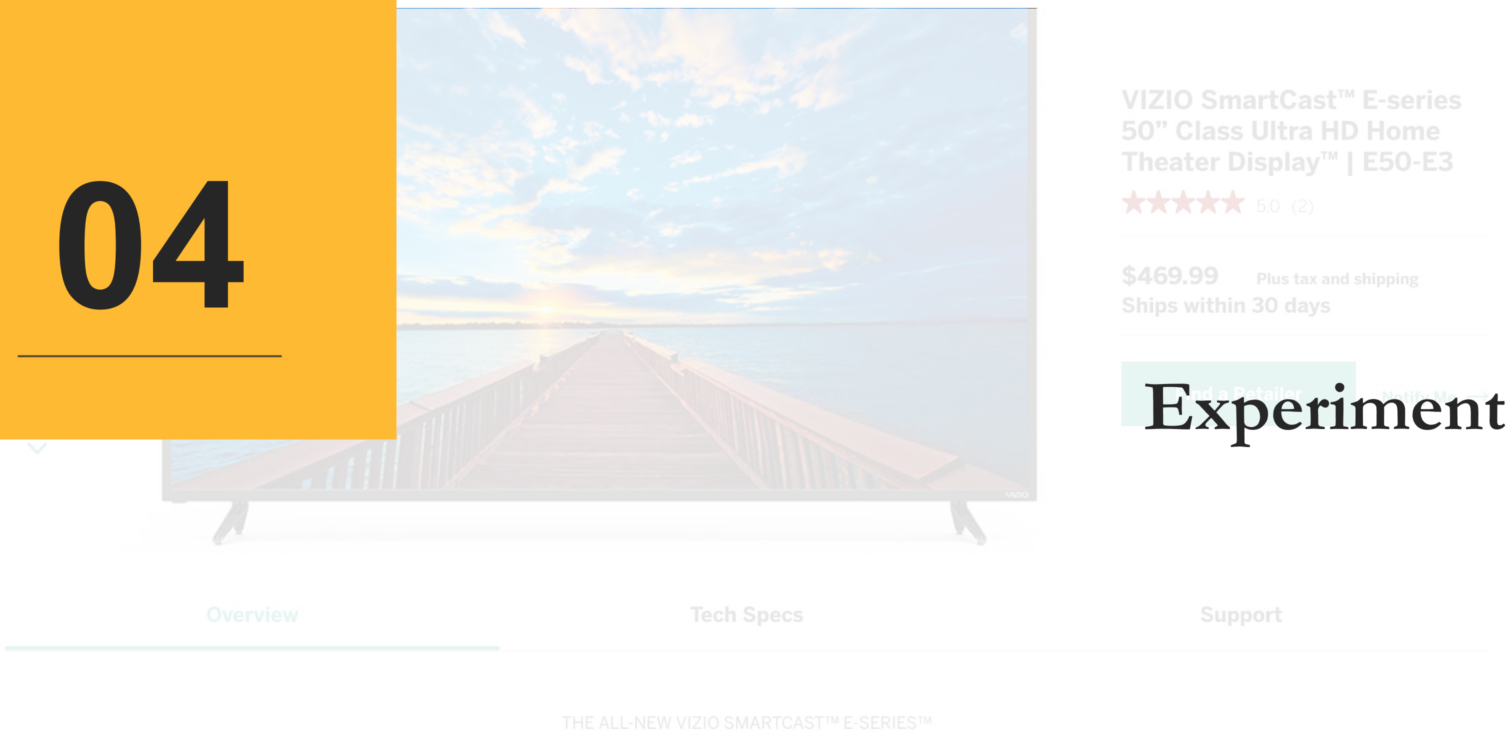
***Vt*** **(helpful votes)** = $\frac{Vote}{averageVote}$

***Rel*** **(relevancy score)** = Score received from input query

***Img*** **(image number)** = log(imageNumber+1)

# 04

# Experiment



# Query Generation

- Generating queries by **extracting keywords** from reviews that a user posted
  - Cannot guarantee every user has posted review.
  - Not realistic enough

- Simulating queries that we thought could be **representative of what a user is looking for**
  - Simple but effective
  - Could be close to query generated in real life

# Experiment Result

- Manual speculation
- Compare different queries with default ranking
- Holistic reviews will always stay on top
- Reviews that meet information need will be ranked higher, but more depend on its intrinsic value
- Example

**Query: sturdy product**

(1) Title: Took a chance, grateful I did.
Review Text: ...this pedal is very *sturdy* and does it's job exceptionally well, without any fluff to inflate the price. It's low cost, it's quality construction, and it does exactly what it says it does...a quality product at a very reasonable price.
Vrf: 1, St: 0.18685, Rel: 0.32344, Crd: 1.98356, Rl: 0.52802, Vt: 0.06614, Score: 1.29585

(2) Title: Great Stereo Volume/Expression pedal!
Review Text: ...it works flawlessly...as both a volume and expression...I use it mainly as a permanent fixture on my custom made digital only effects board for guitar...are still the same great quality...
Vrf: 1, St: 0.27959, Rel: 0.0, Crd: 1.88434, Rl: 0.49381, Vt: 0.26454, Score: 1.25040

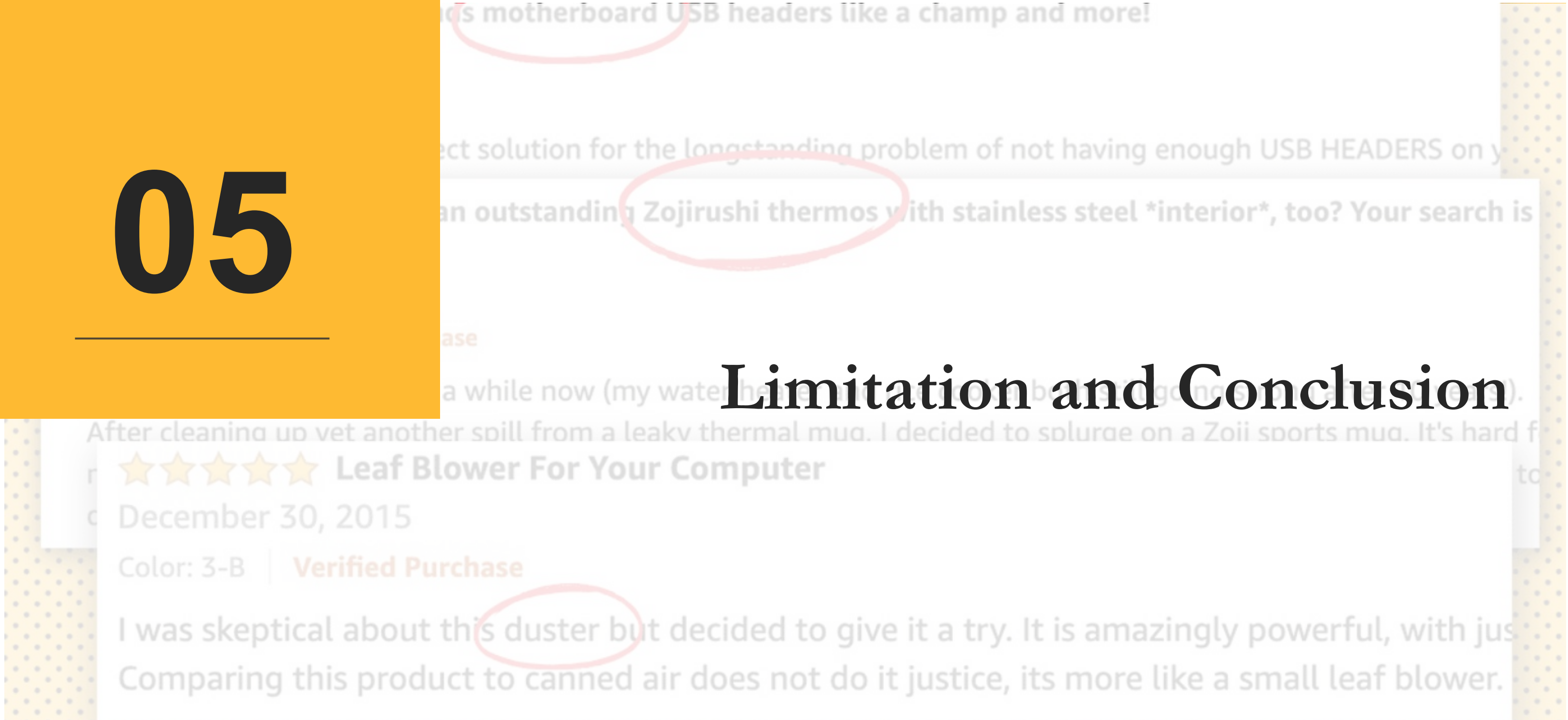
(3) Title: Works Great with my Yamaha Reface
Review Text: ...this is a great pedal...a *sturdy* large pedal, even wider than my guitar wah pedal. I bought this for use with my Yamaha Reface...
Vrf: 1, St: 0.26214, Rel: 0.27157, Crd: 1.63087, Rl: 0.44160, Vt: 0.06614, Score: 1.16369

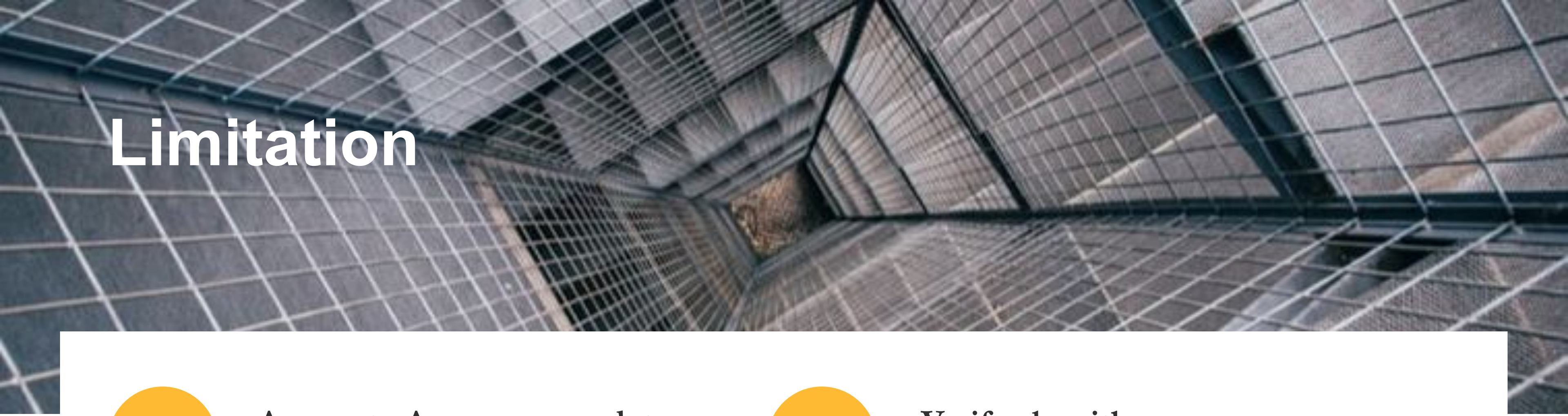
(4) Title: Great, takes some getting used to.
Review Text: ...works as advertised...able to use half-pedaling ...You have a bit of travel on the pedal before it actually registers... and the half-pedaling effect seems to go to a full pedal early...If there was a way to adjust the sensitivity on this pedal...well worth having one of these...
Vrf: 1, St: 0.25961, Rel: 0.0, Crd: 1.82459, Rl: 0.50147, Vt: 0.16534, Score: 1.16034

(5) Title: No right angle plug or removable cord
Review Text: ...very good quality...it has an attached cord instead of allowing you to use your own 1/4" cord...need an right angle connection at the end which this does not have...Would not recommend if...
Vrf: 1, St: 0.17089, Rel: 0.0, Crd: 2.41319, Rl: 0.44682, Vt: 0.13227, Score: 1.12881

05

# Limitation and Conclusion





# Limitation

## 01 Access to Amazon user data

**If we have access…**

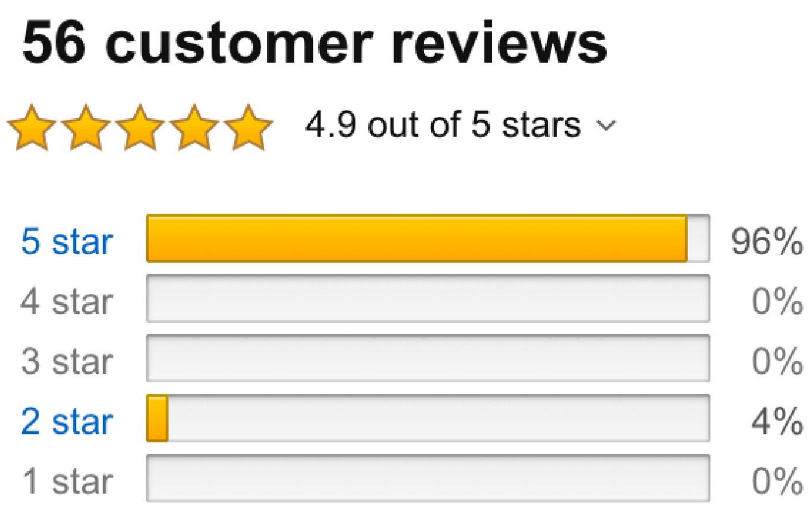
- Generate personalized queries
- Comprehensive credibility score

## 02 Verify algorithm

- We could only checking our ranking results manually
- Not able to justify if our model is optimal or not
- Maybe in the future we could have users to indicate whether a document is relevant.

# Conclusion

- Multi-feature
  - Author credibility
  - Recentness
  - Polarity
  - Relevance
- Weight
  - Regression
  - Random Forest
  - Experimental Fine Tuning



# Thank You!
# Q & A