HELPFUL REVIEWS

According to the Machine

By Corey Wade, September 1, 2018

THE PROBLEM

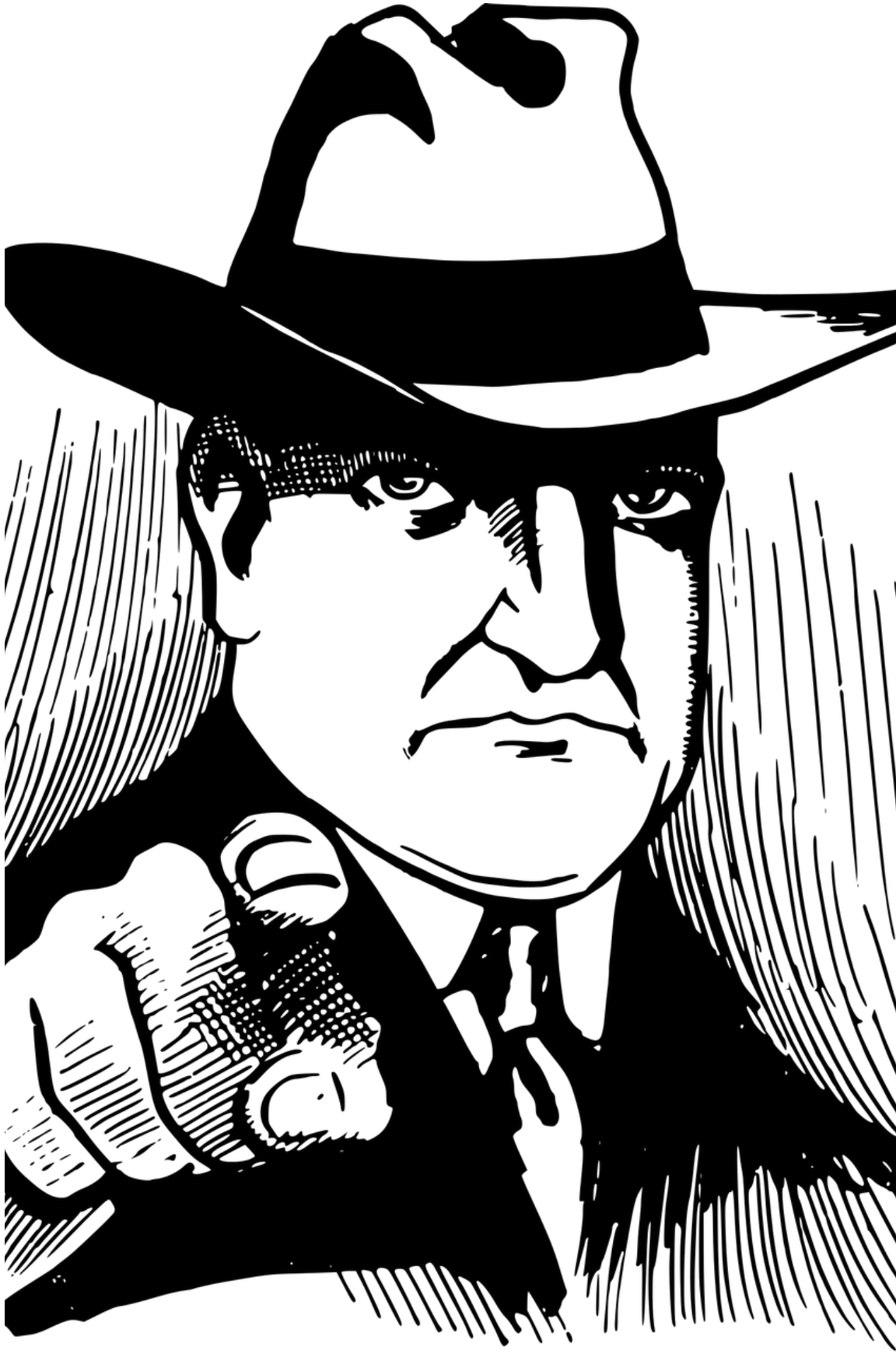
- Amazon's Fake Review Problem Is Now Worse Than Ever, Study Suggests
  - Forbes
- How merchants use Facebook to flood Amazon with fake reviews
  - Washington Post
- Scammers elude Amazon crackdown on fake reviews with new tricks
  - New York Post

THE MISSION

- The first step of the CJW Independent Review Team is to determine whether given reviews are helpful.
- The starting point will be Amazon book reviews.
- A new metric must be created to classify helpful reviews.
- The metric will be later used to determine helpful reviewers.
- Identifying helpful reviewers will be one small step toward eliminating fraud.

THE DATA

- Amazon Book Reviews Dataset.
- 8.9 million rows.
- May 1996 - July 2014.
- Limited to users and books with at least 5 reviews.
- Relevant columns include star rating, text of review, and helpful/unhelpful votes.
- See http://jmcauley.ucsd.edu/data/amazon/ for more info and datasets.



THE FACTS

- Median review is 5 stars.
- Mean review is 4.25 stars.
- 75% are 4 stars or higher.
- Reviews under 3 stars are outliers.
- Most reviews receive 0 or 1 helpful/unhelpful votes.
- 74.3% of votes are helpful.
- Top review has 23,311 helpful votes.
- Fun reviews with most helpful votes at coreyjwade.medium.com

NEW METRIC

- Goal: develop helpful rating metric.
- Strategy: use helpful votes and helpful percentage as primary columns.

CAVEATS

- Helpful Votes must be converted into numbers between 0 and 1.
- The Helpful Votes column is very right skewed.
- The final numbers should reflect meaningful percentages: 50% about average, 90% great.

SKEWED_TO_LINEAR FUNCTION

- Goal: transform skewed column into numbers between 0 and 1 that resemble a ranking.
- Requirements: straight line function; user inputs number of pivots.
- Pivots: points spread across the data using logspace.
- Percentiles: each pivot is joined with its percentile ranking.
- The range (y-value) are the percentiles from the first pivot to the last [0,1].
- Example: 50 pivots. 98th percentile is 716 votes. (716, 0.98) are joined.
- Each pivot is connected to the next via the straight line function. This creates a piecewise linear graph.
- All data points between pivots (x-values) receive a ranking from the corresponding y-values on the graph.

See full function and all relevant jupyter notebooks at https://github.com/coreyjwade/Helpful_Reviews.

HELPFUL RATING METRIC

- Use scaled helpful votes, helpful vote percentage, and percentage of helpful reviews per book (also scaled).
- Percentage of helpful reviews per book takes popularity into account. A user with 1,000 helpful votes of a popular book may have 5% of the votes, while a user with 25 helpful votes of an obscure book may have 75% of the helpful votes.
- Taking the standard deviation of each column, then multiplying by 100 and rounding down gives the following formula:

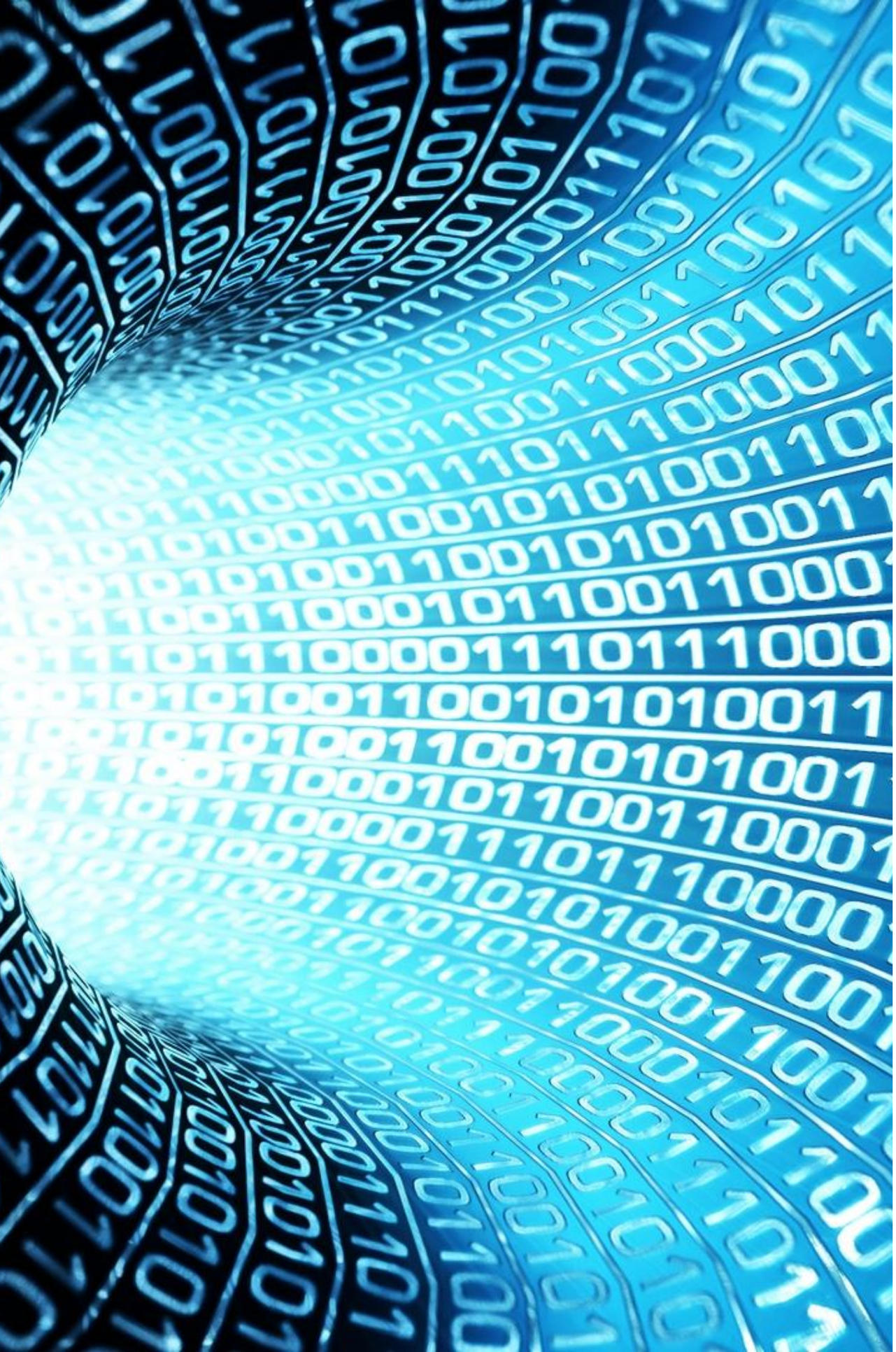
Helpful_Rating = 0.42 * Helpful_Votes_Scaled + 0.56 * Helpful_Percentage + 0.02 * Percentage_Helpful_Reviews

NATURAL LANGUAGE PROCESSING

- Before making predictions, book reviews must be converted into a corpus.
- Normalize corpus with lowercase letters; eliminate stop words and special characters.
- Use CountVectorizer and TfidfVectorizer to convert individual reviews into a sparse matrix.
- Iterate over ngrams. One word, is the default. Also try 2 word combinations and 3 word combinations.
- Best results consistently came from CountVectorizer(ngram_range=1,2), 1 and 2 word combinations.

PREDICTIONS

- Question: Is a particular review helpful?
- X column is the book review, converted into a sparse matrix with CountVectorizer.
- Y column is the helpful rating
- Instead of predicting an exact rating, the data is split into helpful and unhelpful scores.
- Reviews with a helpful rating of over 80% are helpful.
- Reviews with a helpful rating of under 50% are not helpful.



MACHINE LEARNING

- Naive Bayes, Random Forests, Decisions Trees and Logistic Regression were all attempted to make predictions.
- Logistic Regression consistently delivered the best results, followed by Naive Bayes.
- Logistic Regression Cross-Validation had AUC means of over 90%.
- Confusion Matrix precision of unhelpful ratings were over 80%, and precision of helpful ratings were over 90%.

RESULTS

- Helpful reviews were predicted with an accuracy between 80-90%.
- Star ratings were predicted with greater accuracy. This can be used to flag users who gave the wrong amount of stars.
- Deep learning did not initially outperform Logistic Regression, but more tests and reviews are warranted.
- The helpful rating metric can be applied to any product that sums helpful and unhelpful votes (traditionally thumbs up / thumbs down).
- The next step is to use the same pipeline to determine helpful reviewers.
- By highlighting helpful reviewers, companies can help protect consumers against fraud.

REFERENCES

All reports, data wrangling, data analysis, and machine learning test results can be found at Corey Wade's Helpful_Reviews github page.

https://github.com/coreyjwade/Helpful_Reviews

coreyjwade.com