Agenda

1. Background
2. Data Preprocessing
3. Exploratory Data Analysis
4. Pipeline
5. Modeling
6. Conclusion and Future Work

Overview

Amazon is among the largest online marketplace in the world for various products. With its popularity, Amazon is the place where many people actually spend time and write detailed reviews. Data from customer reviews is critical in today's data-driven business environment.

Customer reviews reveal customers' experiences regarding the customer service, prices, quality, and ease of shopping. However, customer reviews are unstructured. Searching and comparing text reviews can be frustrating and time-consuming.

Business Use Case

Goal

The goal is to analyze and understand the sentiments expressed in the customer reviews more efficiently and cost-effectively

Proposed Solution

We built and trained machine learning models that has a high accuracy of predicting the sentiment from reviews. The solution will assist both consumers and manufacturers by:

Business	Assisting companies gain more insights into customer experiences and develop effective strategies to enhance the quality of their offerings
Customers	Helping customers make up their minds for better decision making on purchase

Data Source and Profile

The datasets contain customer review texts regarding products on Amazon with accompanying metadata.

Data Source	Format and Size	Rows/ Columns
Amazon Reviews on Books (Kaggle)	Structured TSV File, 3.24 GB	~3 million rows/ 15 cols
Amazon Reviews on Ebooks (Kaggle)	Structured TSV File, 3.22 GB	~5 million/ 15 cols

Data Source → **Big Data Infrastructure** → **Data Analysis**

kaggle → Google Cloud → Apache Spark

Load files into shared GCP storage bucket and created Dataproc cluster

Data Preprocessing

```
root
 |-- marketplace: string (nullable = true)
 |-- customer_id: integer (nullable = true)
 |-- review_id: string (nullable = true)
 |-- product_id: string (nullable = true)
 |-- product_parent: integer (nullable = true)
 |-- product_title: string (nullable = true)
 |-- product_category: string (nullable = true)
 |-- star_rating: integer (nullable = true)
 |-- helpful_votes: integer (nullable = true)
 |-- total_votes: integer (nullable = true)
 |-- vine: string (nullable = true)
 |-- verified_purchase: string (nullable = true)
 |-- review_headline: string (nullable = true)
 |-- review_body: string (nullable = true)
 |-- review_date: string (nullable = true)
```

1.Drop irrelevant columns

2. Deal with missing values and duplicates

3. Create label column

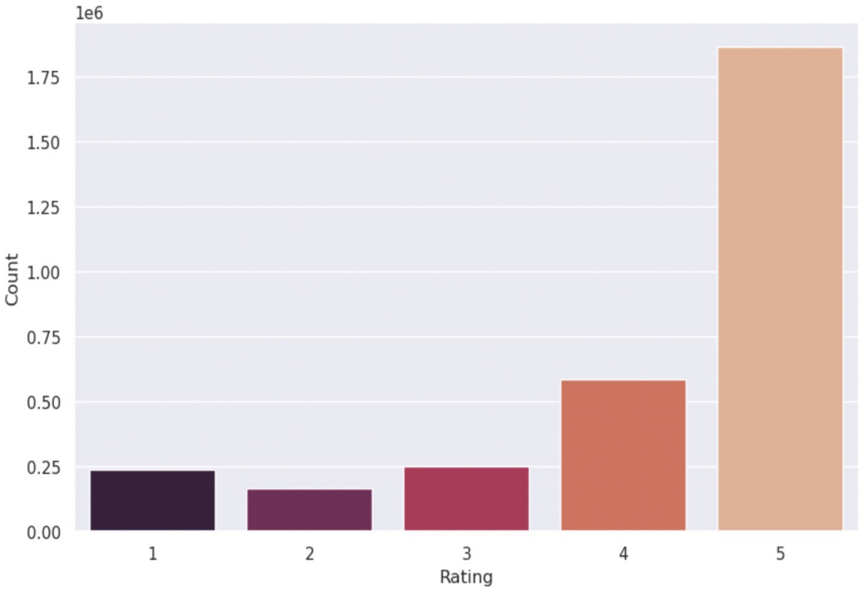
Generate Sentiment Label

"star_rating"		"label"
1, 2, 3	→ Sentiment label →	Negative 0
4, 5	→	Positive 1

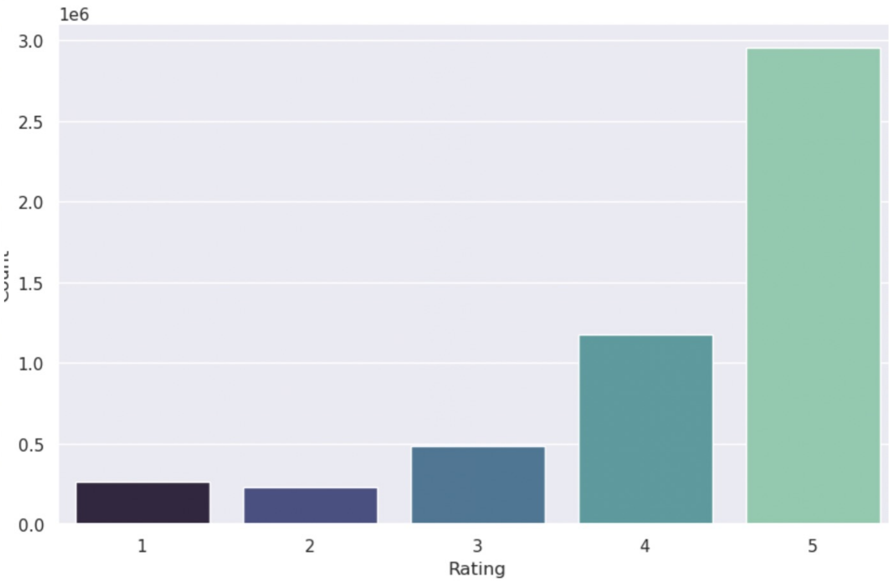
Exploratory Data Analysis

Star Rating

Books

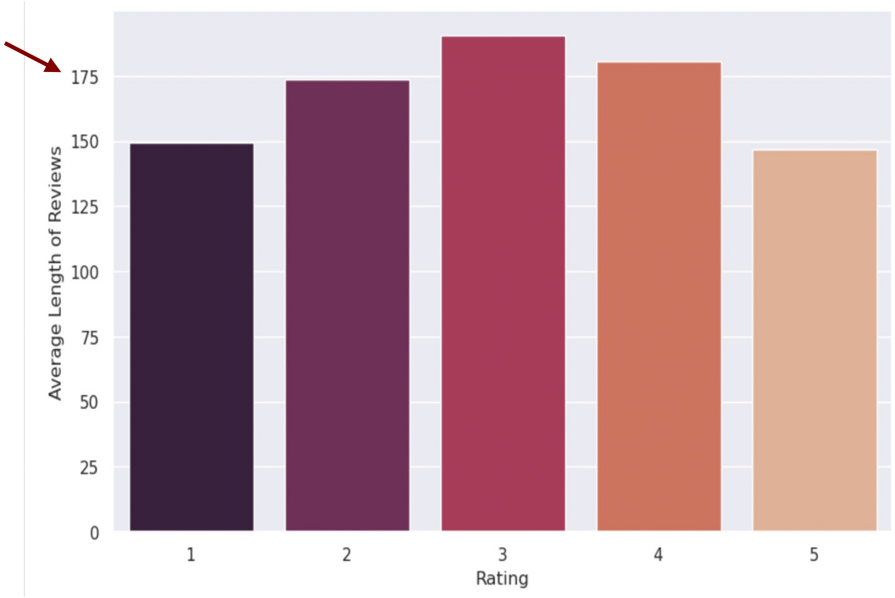


EBooks

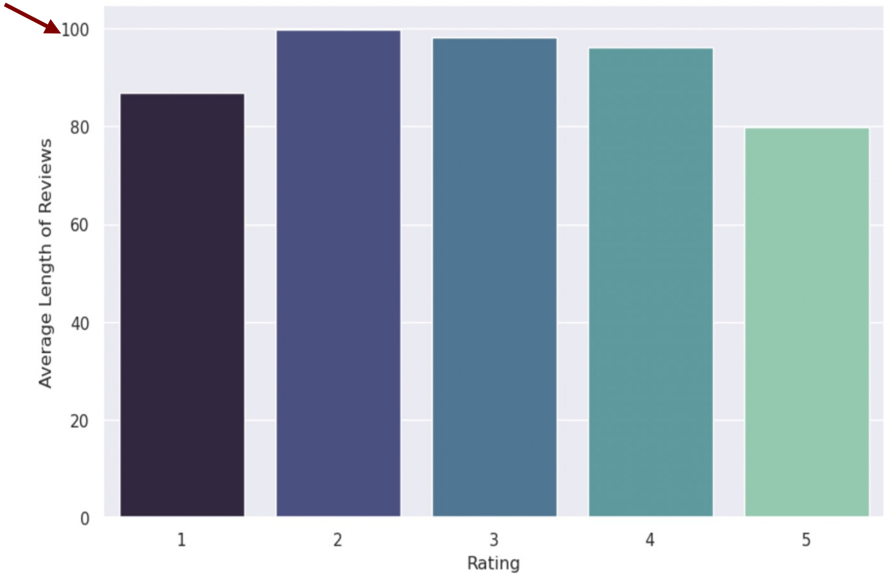


Length of Reviews

Books

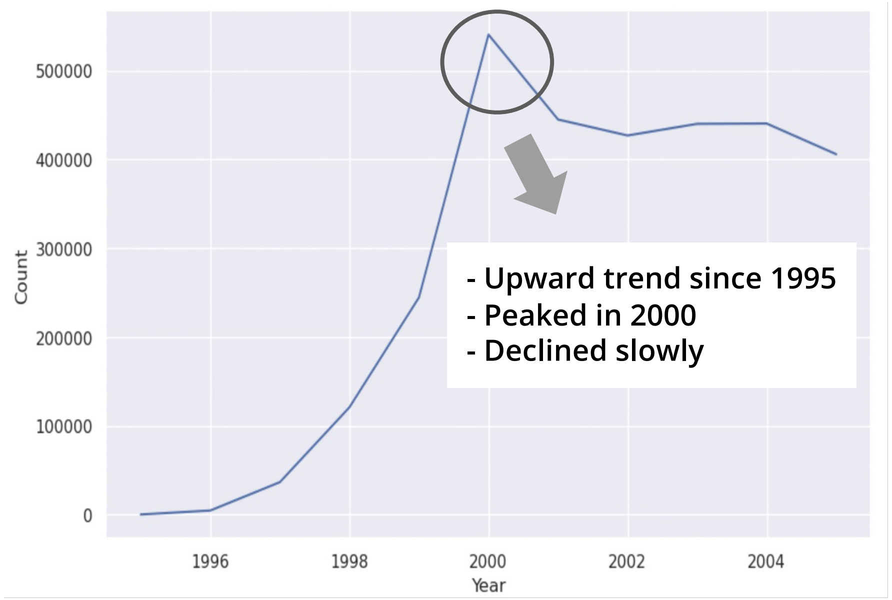


EBooks

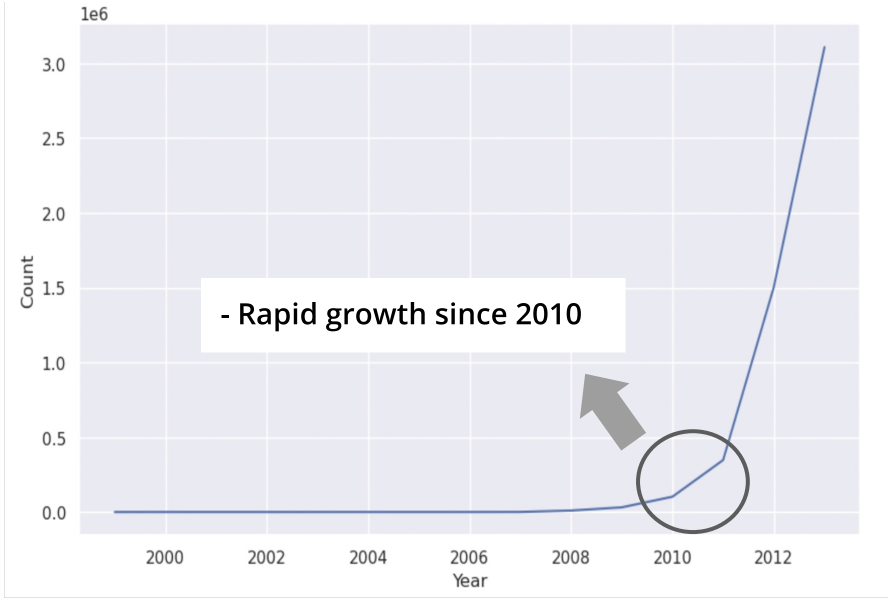


Year of Reviews

Books

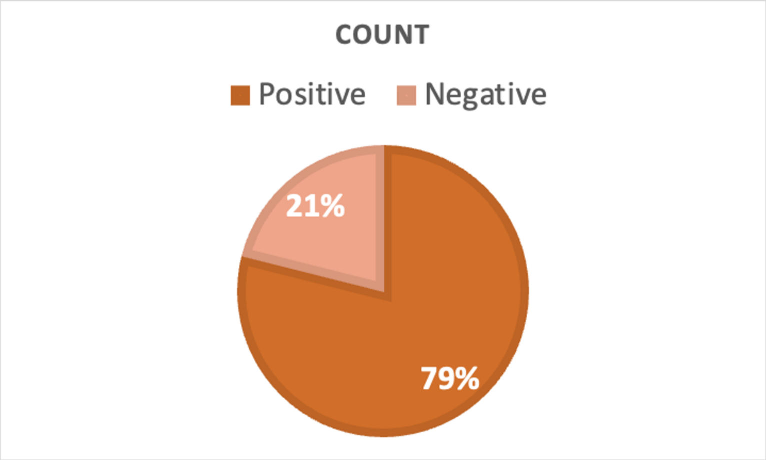


EBooks

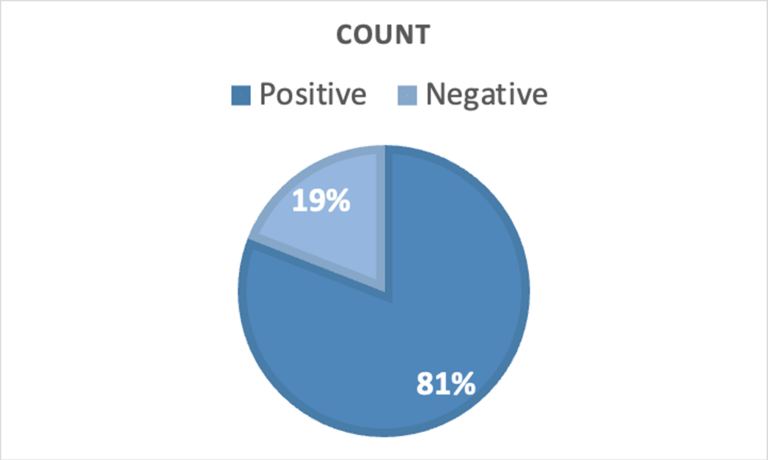


Sentiment

Books



EBooks



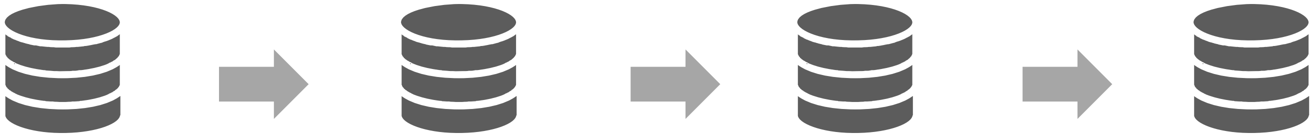
- Target variable
- Imbalanced dataset

NLP Pipeline

NLP Pipeline

Pipeline (Estimator) | RegexTokenizer | StopWordsRemover | CountVectorizer TF-IDF

Pipeline.fit()



Raw text | **Words** | **Cleaned Words** | **Feature vectors**

Modeling

Methodology

NLP Technique	Model
Count Vectorizer	**Logistic Regression**
Convert a collection of text documents to vectors of token counts	Baseline model to predict the sentiment of reviews - positive/negative
TF-IDF	**Naive Bayes**
TF-IDF (Term Frequency – Inverse Document Frequency) provides a normalized version of term frequencies	Bayes' Theorem based, used to compare with Logistic Regression on sentiment classification

Model Selection

Books

Model	Accuracy	F1	Training Time
Logistic Regression with Count Vectorizer	0.84	0.81	4min 31s
Logistic Regression with TF-IDF	0.84	0.81	4min 6s
Naive Bayes with Count Vectorizer	0.84	0.85	1min 35s
Naive Bayes with TF-IDF	0.80	0.81	2min 15s

Model Performance

Books & EBooks

There is a total of **8M product reviews** in the combined dataset.

Model	Accuracy	F1	Training Time
Naive Bayes with Count Vectorizer	0.86	0.87	3min 10s

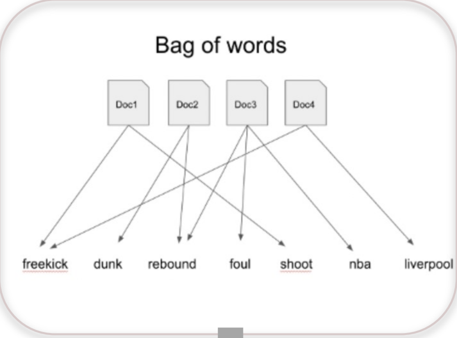
Topic Modeling with LDA

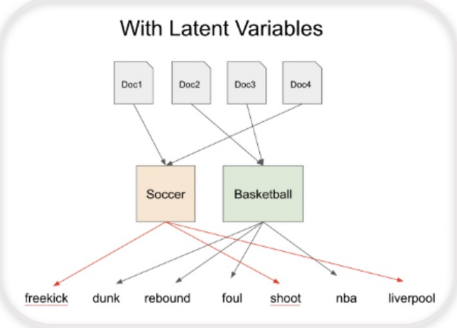
Topic Modeling

Topic modeling is a method for unsupervised classification of documents, similar to clustering on numeric data, which finds natural groups of items even when we're not sure what we're looking for.

LDA (Latent Dirichlet allocation)

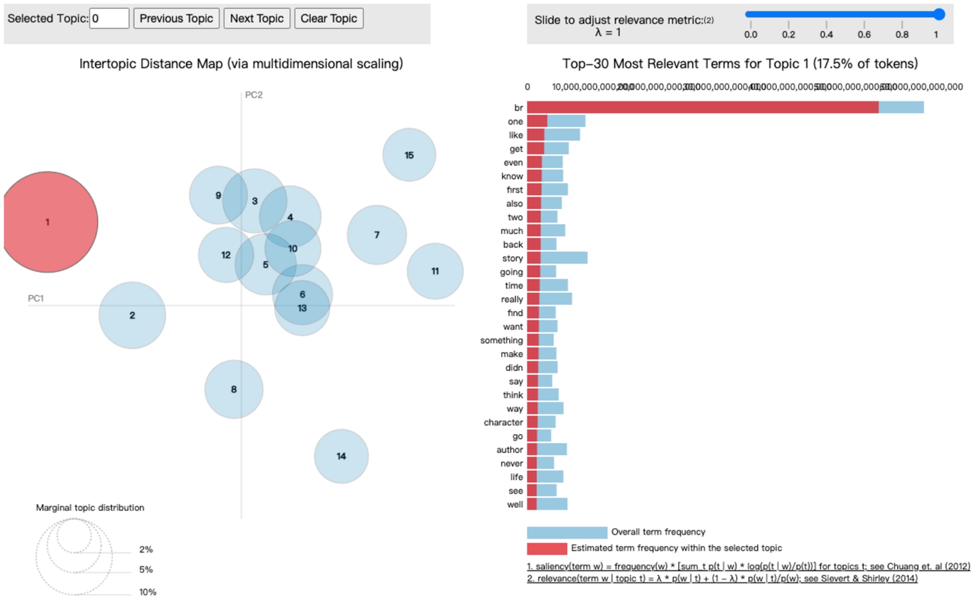
Latent Dirichlet allocation (LDA) is a particularly popular method for fitting a topic model. Each document is treated as a mixture of topics, and each topic is treated as a mixture of words. Rather than being separated into discrete groups, documents can "overlap" in terms of content, mimicking typical natural language use.





Topic Modeling - EBooks

EBooks



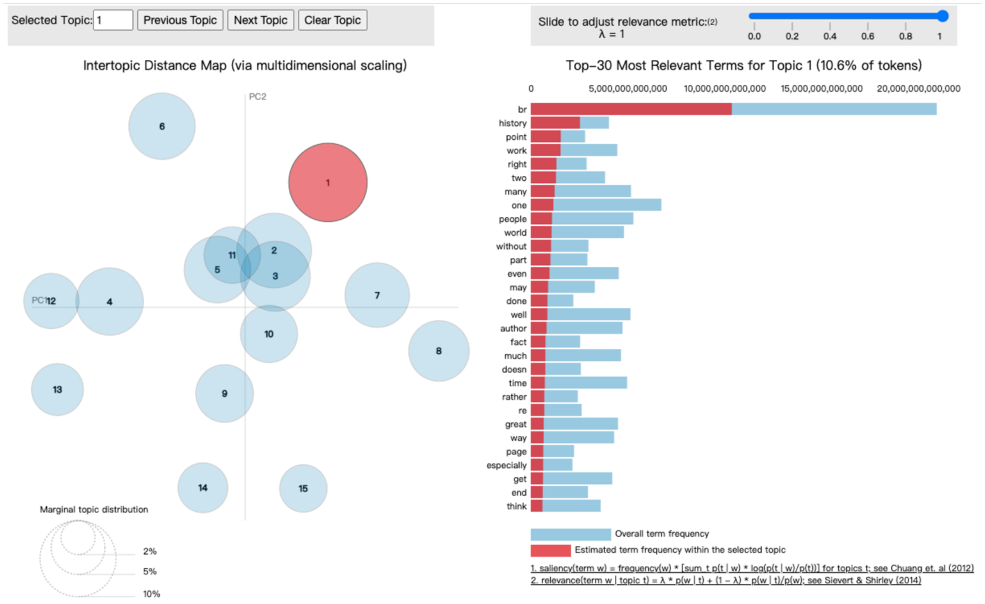
Specify k=15: generate 15 clusters

Top 5 words for each cluster:

1. br, one, like, get, even
2. br, things, makes, **life**, **world**
3. enjoyed, didn, something, back, really
4. m, love, another, time, going, well
5. series, lot, first, right, think
6. found, real, another, work, made
7. stories, interesting, enjoy, books, written
8. **kindle**, better, people, keep, many
9. New, **romance**, liked, see, little
10. easy, couldn, put, best, series
11. loved, every, story, really, thought
12. still, got, bit, though, story
13. wait, next, waiting, hard, world
14. **recommend**, anyone, life, always, **must**
15. **novel**, looking, forward, reading, plot

Topic Modeling - Books

Books



Specify k=15: generate 15 clusters

Top 5 words for each cluster:

1. br, **history**, point, work, right
2. worth, seems, br, nothing, far
3. us, people, world, help, going
4. **novel**, characters, series, plot, character
5. br, day, life, **family**, man
6. highly, reader, job, **recommend**, written
7. information, used, d, lot, br
8. quot, br, one, work, us
9. ll, want, pages, m, takes
10. easy, looking, busy, new, must
11. br, excellent, books, quite, understand
12. love, put, style, story, wonderful
13. children, old, year, young, enjoyed
14. quot, ve, last, didn, stories
15. quot, got, enjoy, thought, almost

Conclusion and Future Work

Conclusion

Key Findings

- Sentiment analysis was performed using supervised and unsupervised machine learning techniques
- Sentiment Prediction - Naive Bayes with count vectorizer performs the best
- Topic Modeling - Discover latent relationships in the corpus

Challenges

- Imbalanced dataset with more positive labels, will cause overfitting problem
- Huge computational power needed to extract and process reviews data

Future Work

- Utilize advanced sentiment analyzer to return sentiment labels
- Use higher-order n-gram methods (such as trigram) to have a deeper understanding of the review's context
- Apply other embedding techniques such as GloVe, BERT for vectorizing the words
- Explore more categories' reviews and combine larger dataset

Thank you.

Reference

- Doll, Tyler. “Lda Topic Modeling.” Medium. Towards Data Science, March 11, 2019. https://towardsdatascience.com/lda-topic-modeling-an-explanation-e184c90aadcd.
- Ipshita. “Topic Modelling Using LDA.” Medium. Analytics Vidhya, August 4, 2021. https://medium.com/analytics-vidhya/topic-modelling-using-lda-aa11ec9bec13.
- Robinson, Julia Silge and David. “6 Topic Modeling: Text Mining with R.” 6 Topic modeling | Text Mining with R. Accessed November 26, 2022. https://www.tidytextmining.com/topicmodeling.html.
- Seth, Neha. “Topic Modeling and Latent Dirichlet Allocation (LDA) Using Gensim.” Analytics Vidhya, August 26, 2021. https://www.analyticsvidhya.com/blog/2021/06/part-2-topic-modeling-and-latent-dirichlet-allocation-lda-using-gensim-and-sklearn/.