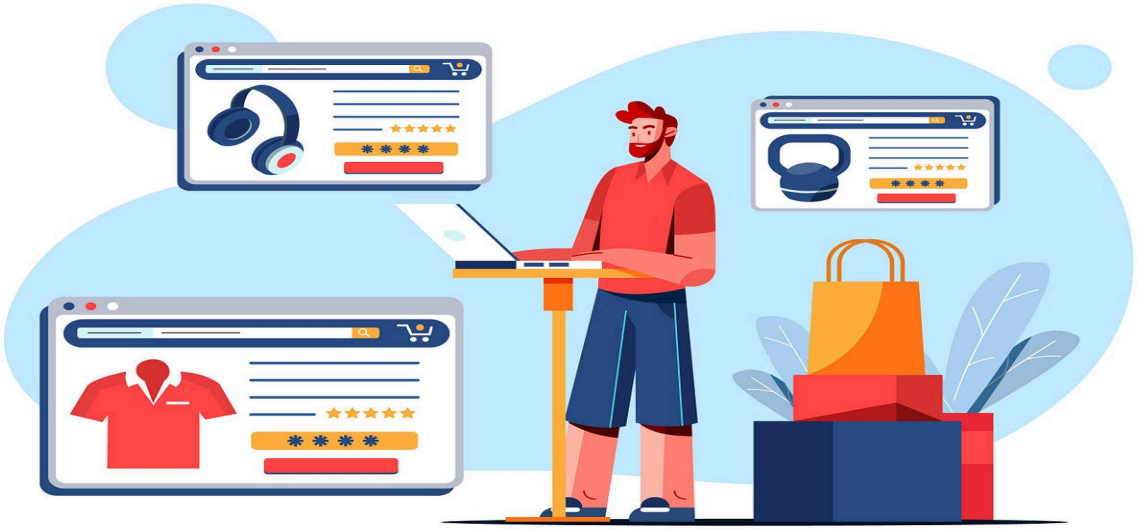
PRODUCT RECOMMENDATION SYSTEM USING USER REVIEWS



1. Introduction

Our project's aim is to deeply comprehend and evaluate user feedback in order to develop a strong recommendation system. By exploring the abundance of data in these reviews, our goal is to reveal important findings about user likes, feelings, and interactions with different products. This includes utilizing NLP techniques to analyze and understand text data, identifying important features and patterns to improve our recommendation systems.

The main goal is to improve the customization of product suggestions so that each user is provided with recommendations that closely match their unique preferences and requirements. In order to accomplish this, we will evaluate the opinions conveyed in the reviews, recognize common topics and keywords, and determine the general levels of satisfaction among users. Combining this analysis with user profiles and past behavior data will allow our recommendation system to forecast and propose products that users will probably enjoy and deem valuable.

Our method will consist of various phases such as collecting data, processing, analyzing sentiment, extracting features, and creating specialized machine learning models for recommending purposes. We will take into account a range of factors, including the variety of user viewpoints, the circumstances of review writing, and any potential biases in the data. Our goal is to create a thorough recommendation system that enhances user satisfaction and improves the shopping experience through relevant and timely product suggestions by blending these elements.

2. Literature review

A comprehensive literature review is essential for understanding the current state of research in review-based recommender systems and identifying areas for improvement. This field has seen significant advancements in recent years, leveraging the rich information contained in user reviews to enhance recommendation accuracy and interpretability.

Review-Based Recommender Systems: An Overview

Review-based recommender systems have emerged as a significant sub-field within the broader domain of recommendation systems. These systems leverage the wealth of information contained in user reviews to offer more personalized, accurate, and explainable recommendations[1]. Unlike traditional recommender systems that primarily rely on numerical ratings or browsing history, review-based systems explore the rich textual feedback provided by users[1].

3. Feature Extraction from Reviews

One of the key challenges in review-based recommender systems is extracting meaningful features from textual data. Several approaches have been developed to address this:

3.1 Sentiment Analysis: Sentiment analysis is widely used to extract user opinions from reviews. This technique helps in understanding the overall sentiment towards a product or specific aspects of it[2]. Advanced methods incorporate aspect-based sentiment analysis to provide more fine-grained insights into user preferences[3].

3.2 Topic Modeling: Topic modeling techniques, such as Latent Dirichlet Allocation (LDA), are employed to automatically discover latent topics in reviews. These topics can represent product aspects or user concerns, providing valuable information for recommendation algorithms[4].

3.1.1 Deep Learning Approaches: Recent advancements in deep learning have led to more sophisticated feature extraction methods. Techniques such as word embeddings and neural networks have shown promising results in capturing semantic information from reviews[5].

3.1.2 Integration with Recommendation Algorithms: Researchers have explored various ways to integrate review-based features into recommendation algorithms:

3.1.3 Matrix Factorization: Extended matrix factorization models that incorporate review-based features have shown improved performance over traditional collaborative filtering approaches[6].

(a) Deep Learning Models: Deep learning models, such as neural collaborative filtering enhanced with review information, have demonstrated state-of-the-art performance in several recommendation tasks[7].

(b) Hybrid Approaches: Hybrid models that combine collaborative filtering with content-based methods using review information have been proposed to leverage the strengths of both approaches[8].

4. DATASET

It is a Amazon Product Review Dataset of various consumer products and user feedbacks which can be useful for text

analysis, recommendation system and sentiment analysis

The dataset consists of 10 columns and 568,454 entries, each of which represents a distinct feature of product reviews.

The column are :

- ➢ Id- Unique identifier for each review
- ➢ ProductId-
  - ○ 74258 unique values
  - ○ Identifier for each product being reviewed.
- ➢ UserId-
  - ○ 256059 unique values
  - ○ Identifier for each user who wrote a review.
- ➢ ProfileName- 218418 unique values
- ➢ HelpfulnessNumerator- defines the number of users who found the reviews helpful
- ➢ Helpfulness Denominator- defines the number of users and whether they found the reviews useful or not
- ➢ Score- Ratings of the products given by the user
- ➢ Time- Timestamp of the review
- ➢ Summary-
  - ○ 295744 unique values
  - ○ Summary of the Review
- ➢ Text-
  - ○ 393579 unique values
  - ○ Full text of the review

"Amazon Reviews" that perfectly aligns with the requirements of our project. This dataset is comprehensive, circumscribing 568,454 entries and 10 distinct attributes, each providing valuable information about the product reviews. Despite the extensive range of data available, our focus will be primarily on two specific columns: 'Score' and 'Text'.

The 'Score' column, which is integral to our analysis, represents the numerical ratings given by users to the products. This numerical score is critical as it will serve as a benchmark for validating the sentiment analysis results derived from the 'Text' column. The 'Text' column, containing the detailed reviews written by users, is the cornerstone of our sentiment analysis. These reviews provide high level textual data, offering understanding into users' opinions, experiences and satisfaction levels with the products they have purchased and reviewed.

Our objective is to grasp the textual data in the 'Text' column to perform sentiment analysis and therefore understand the underlying sentiments expressed by users. By analyzing this text, we aim to categorize reviews into positive, negative, or neutral sentiments, which can then be correlated with the corresponding numerical scores for further validation and insights.In summary, the "Amazon Reviews" dataset not only meets our project requirements but also offers a robust foundation for conducting sentiment analysis. By focusing on the 'Score' and 'Text' columns, we can delve deeply into the sentiments expressed in the reviews, providing a nuanced understanding of customer feedback. This approach ensures that our analysis is both focused and relevant, ultimately contributing to the overarching goals of our project.

```
RangeIndex: 568454 entries, 0 to 568453
Data columns (total 10 columns):
 #   Column                  Non-Null Count   Dtype
---  ------                  --------------   -----
 0   Id                      568454 non-null  int64
 1   ProductId               568454 non-null  object
 2   UserId                  568454 non-null  object
 3   ProfileName             568428 non-null  object
 4   HelpfulnessNumerator    568454 non-null  int64
 5   HelpfulnessDenominator  568454 non-null  int64
 6   Score                   568454 non-null  int64
 7   Time                    568454 non-null  int64
 8   Summary                 568427 non-null  object
 9   Text                    568454 non-null  object
dtypes: int64(5), object(5)
memory usage: 43.4+ MB
```

Fig 1 Data Loading

5. Exploratory Data Analysis

We start the EDA by plotting the length of the reviews

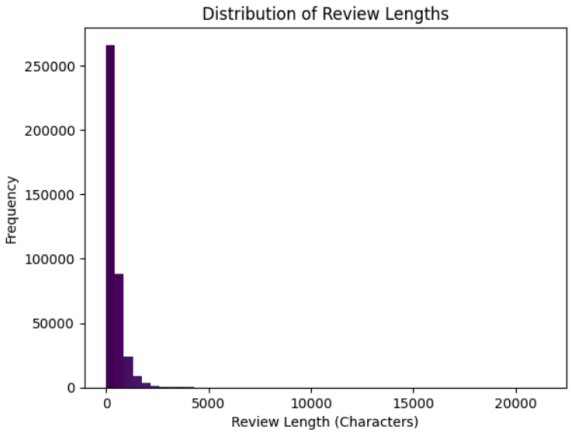


Fig 1 - Length of Reviews

5.1 Finding the univariate score

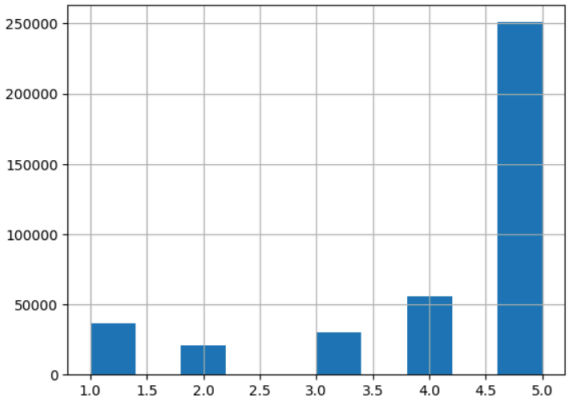


Fig 2 Univariate Score

5.2 Finding the bivariate score

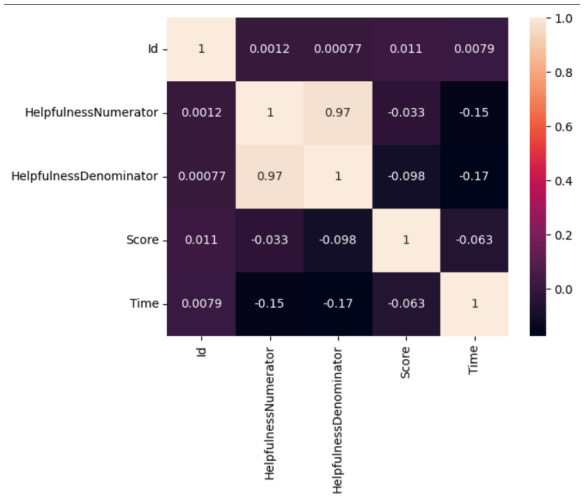


Fig 3 Bivariate Score

5.3 Finding the multivariate score

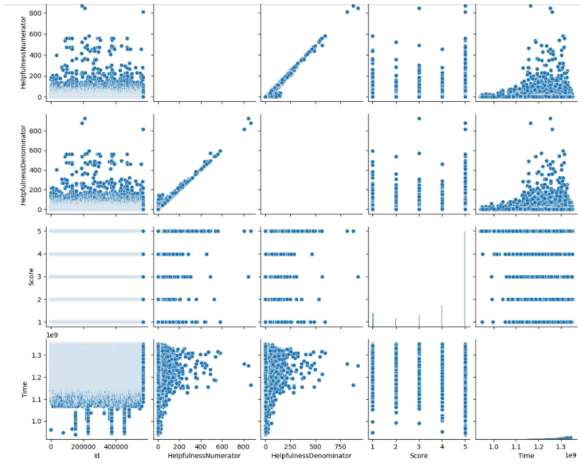


Fig 4 Multivariate Score

5.4 Extracting the information that is useful

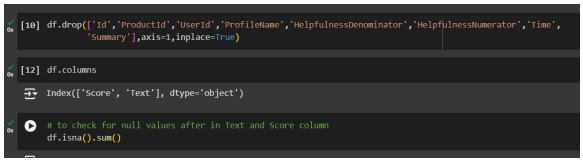


Fig 5

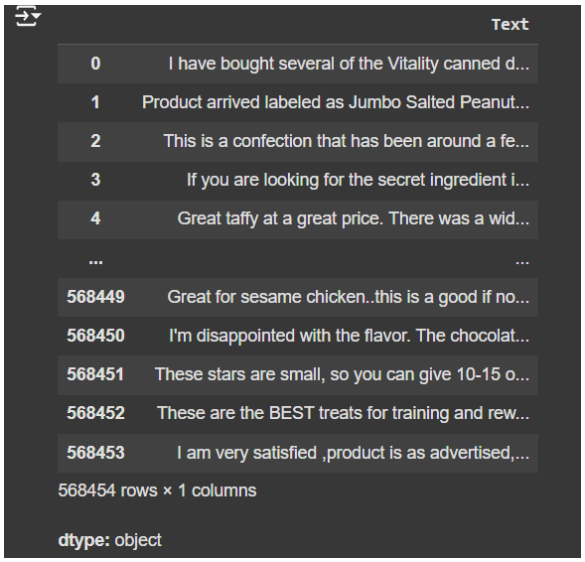


Fig 6

5.5 From this the score will be

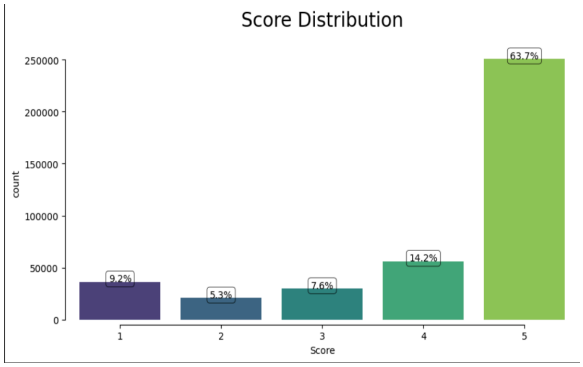


Fig 7

5.6 Next is removing all the duplicates,

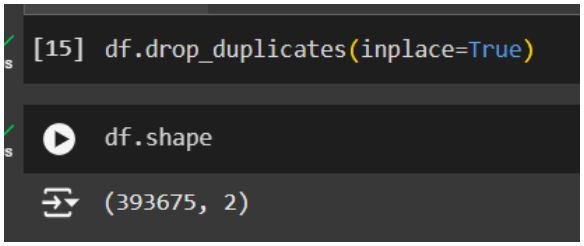


Fig 8

5.7 Now the updated score will be

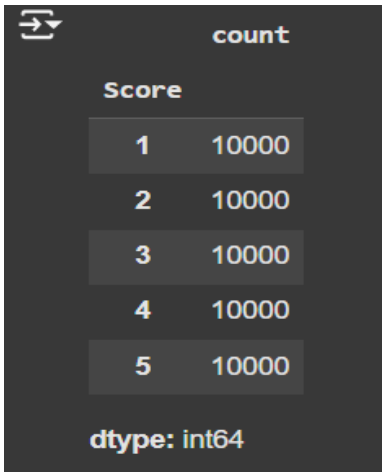


Fig 9

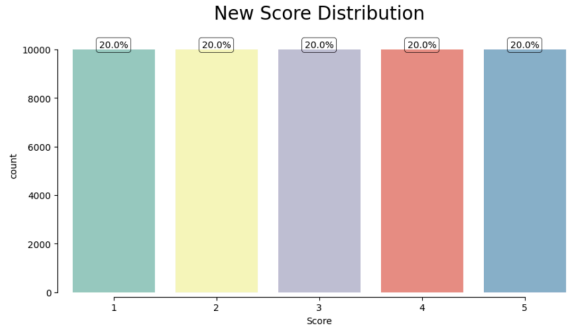


Fig 10

After this we started with the text preprocessing where we clean the text.

```python
import nltk
import string
from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize
from nltk.stem import PorterStemmer
# Download the 'punkt' resource
nltk.download('punkt')
def clean_text(text):
    # 1. Convert to lower
    txt=text.lower()

    # 1. split to words
    tokens=word_tokenize(text)

    # 3. remove punctuation
    tokens=[word for word in tokens if word not in string.punctuation]

    # 4. Remove stopwords
    tokens=[word for word in tokens if word not in stop_words]

    # 5. Remove numbers
    tokens=[word for word in tokens if not word.isdigit()]

    # 6. Apply Stemming
    tokens=[stemming.stem(word) for word in tokens]

    # To return these single words back into one string
    return ' '.join(tokens)
```

Fig 11

The dataset on which we will work on consists of data in this manner:

```
[39] new_df['cleaned_text'] = new_df['Text'].apply(clean_text)
```

new_df

	Score	Text	cleaned_text
0	1	I really LOVED this water. I've been ordering ...	i realli love water i 've order sinc februari ...
1	1	I placed the order on 12/6/09 and here it is 1...	i place order 12/6/09 12/31/09 i never receiv ...
2	1	yuck, yuck, and yuck. I ate about three of th...	yuck yuck yuck i ate three toss rest do n't ev...
3	1	We use this type of oil in a chicken marinade ...	we use type oil chicken marinad love the oil s...
4	1	Too pricy for what you get. I was expecting so...	too pricl get i expect someth delici unfortun ...
...	...	...	...
49995	5	What can I say? No more lugging coffee from th...	what i say no lug coffe store thi roast brew d...
49996	5	I did not know that I liked sardines.A friend ...	i know i like sardines.a friend mine share i t...
49997	5	I try to take this product on a regular basis,...	i tri take product regular basi everi day ever...
49998	5	Feb 3, 2012 *Simply put, De Cecco p...	feb br br '' simpli put de cecco pasta best ``...
49999	5	This was the first time we tried wild rice by ...	thi first time tri wild rice rice mix the cook...

50000 rows × 3 columns

Fig 12

5.8 Descriptive Analysis

The descriptive statistics for the numerical variables were obtained which are given below:

```
count    568401.000000
mean          1.743903
std           7.636845
min           0.000000
25%           0.000000
50%           0.000000
75%           2.000000
max         866.000000
Name: HelpfulnessNumerator, dtype: float64

count    568401.000000
mean          2.227911
std           8.288820
min           0.000000
25%           0.000000
50%           1.000000
75%           2.000000
max         923.000000
Name: HelpfulnessDenominator, dtype: float64

count    568401.000000
mean          4.183297
std           1.310376
min           1.000000
25%           4.000000
50%           5.000000
75%           5.000000
max           5.000000
Name: Score, dtype: float64
```

Fig 13

HelpfulnessNumerator is defined as the number of users who found the review helpful. The mean value of around 2 and Maximum value is 866. The minimum value is 0. This seems logically correct. The maximum value can be considered an outlier because 77th percentile is 2.

HelpfulnessDenominator is the total number of people who either find the review useful and not useful. Here also, the mean value is around 2. Maximum value is 923 which can be regarded as an outlier.

The mean value for score is around 4 which shows leaning towards positive reviews.

The standard deviation for all these attributes is low.

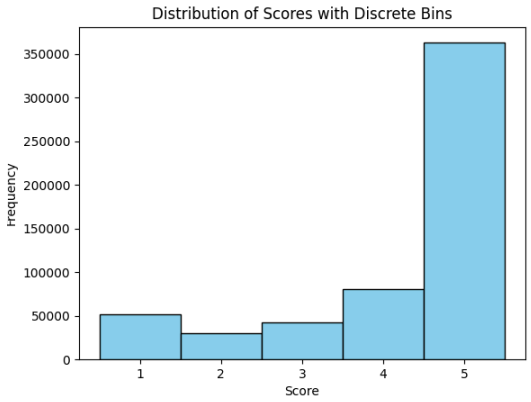


Fig 14

The distribution of the scores is highly negative which shows the higher proportion of positive reviews.

Let's observe the top ten rated products as well as low rated products.

```
Out[11]: (ProductId
          B009WVB40S    5.0
          B009PIAFTE    5.0
          B009PG8MVO    5.0
          B009PFJUF2    5.0
          B009PCDDO4    5.0
          B009OY38SY    5.0
          B009OM66IU    5.0
          B009OM65H2    5.0
          B009OM65GI    5.0
          B009O7DGEW    5.0
          Name: Score, dtype: float64,
          ProductId
          B005ZVBYM2    1.0
          B009O7B1I0    1.0
          B009P4KMZA    1.0
          B009UOFTUI    1.0
          B00006IDJO    1.0
          B00316WKAS    1.0
          B0030LELS8    1.0
          B0030MKP8C    1.0
          B0030E9EMS    1.0
          B009KPU6LO    1.0
          Name: Score, dtype: float64)
```

Fig 15

A new variable helpfulnessRatio is defined which shows the proportion of the number of people who found the reviews useful. The maximum value of Helpfulness Ratio should be one which shows that all of the people who rated it useful or not.

```
count    298371.000000
mean          0.777036
std           0.346273
min           0.000000
25%           0.600000
50%           1.000000
75%           1.000000
max           3.000000
Name: HelpfulnessRatio, dtype: float64
```

Fig 16

The maximum value for the Helpfulness ratio should be one because it shows all of the reviews.

```
count    298371.000000
mean          0.777036
std           0.346273
min           0.000000
25%           0.600000
50%           1.000000
75%           1.000000
max           3.000000
Name: HelpfulnessRatio, dtype: float64
```

Fig 17

	ProductId	Count
0	B007JFMH8M	913
1	B002QWP89S	632
2	B002QWP8H0	632
3	B002QWHJOU	632
4	B0026RQTGE	632
5	B003B3OOPA	623
6	B001EO5Q64	567
7	B006HYLW32	564
8	B001RVFEP2	564
9	B0013NUGDE	564

Fig 18

We can see there are 913 reviews for 1st product and 564 reviews for 10th highest product.

The correlation between HelpfulnessNumerator and Score is

-0.03259 which indicates that lower scores (negative reviews) receive more helpfulness votes. This is logically true because people try to avoid those products which have high lower scores.

6. Research Questions

1. How will specific product features mentioned in reviews affect user ratings?

Answer: By using Natural Language Processing (NLP) technique topic modeling or aspect-based sentiment analysis, we can identify key features like product quality, durability or price in user reviews and analyze their influence on overall ratings.

2. How will sentiment in the review text correlate with rating scores?

Answer: Sentiment analysis can be performed on the review text to find the positive, neutral, or negative labels.

3. How does review length impact the rating score?

Answer: By analyzing the length of reviews and comparing them to rating scores, we may find that longer reviews are either highly positive or highly negative.

4. Can we predict a product's rating based solely on the text of its reviews?

Answer: Using machine learning models like logistic regression, SVC, Naive Bayes, Recurrent Neural Network, LSTM on the review text, we will attempt to predict highest accuracy.

7. Challenges and Future Directions

Despite the progress in review-based recommender systems, several challenges remain:

1. Handling the vast and varied nature of textual data
2. Addressing issues like bias and privacy concerns in user-generated content
3. Improving the interpretability of recommendations derived from complex models
4. Incorporating multimodal data (e.g., images, videos) alongside textual reviews[1]

The 'Id' field is an integer that uniquely identifies each entry. Unique identifiers for users and goods are represented by the 'ProductId' and 'UserId' columns, which are both of type object. Although it has 26 missing values, the user names are contained in the object type "ProfileName." The helpfulness of reviews is measured by the integers in the 'HelpfulnessNumerator' and 'HelpfulnessDenominator' columns. The rating provided by users is represented by an integer in the 'Score' column. The timestamp of the review is probably represented by the 'Time' column, which is another integer. The brief summaries are contained in the object types 'Summary' and 'Text' columns.

Overall, the dataset utilizes 43.4 MB of memory, indicating a substantial volume of text data within the reviews.

8. Future research directions may include:

1. Exploring the integration of multi-criteria rating information
2. Investigating ethical considerations in review-based recommendations
3. Developing more robust and transparent systems that can explain their recommendations to users[1]

9. References:

[1] Li, C., Chen, G., & Wang, F. (2015). Recommender systems based on user reviews: the state of the art. User Modeling and User-Adapted Interaction, 25(2), 99-154.

[2] Chen, C., Zhang, M., Liu, Y., & Ma, S. (2018). Neural attentional rating regression with review-level explanations. In Proceedings of the 2018 World Wide Web Conference (pp. 1583-1592).

[3] Yin, Y., Huang, L., & Zhang, L. (2017). Aspect-based sentiment analysis using deep learning. In Proceedings of the 2017 ACM on Conference on Information and Knowledge Management (pp. 2267-2270).

[4] McAuley, J., & Leskovec, J. (2013). Hidden factors and hidden topics: understanding rating dimensions with review text. In Proceedings of the 7th ACM conference on Recommender systems (pp. 165-172).

[5] Zheng, L., Noroozi, V., & Yu, P. S. (2017). Joint deep modeling of users and items using reviews for recommendation. In Proceedings of the Tenth ACM International Conference on Web Search and Data Mining (pp. 425-434).

[6] Zhang, Y., Lai, G., Zhang, M., Zhang, Y., Liu, Y., & Ma, S. (2014). Explicit factor models for explainable recommendation based on phrase-level sentiment analysis. In Proceedings of the 37th international ACM SIGIR conference on Research & development in information retrieval (pp. 83-92).

[7] Seo, S., Huang, J., Yang, H., & Liu, Y. (2017). Interpretable convolutional neural networks with dual local and global attention for review rating prediction. In Proceedings of the eleventh ACM conference on recommender systems (pp. 297-305).

[8] Catherine, R., & Cohen, W. (2017). Transnets: Learning to transform for recommendation. In Proceedings of the eleventh ACM conference on recommender systems (pp. 288-296).

10. Citations:

[1] https://arxiv.org/html/2405.05562v2

[2] http://ijrsset.org/pdfs/v5-i12/6.pdf

[3] https://dl.acm.org/doi/10.1007/s11257-015-9155-5

[4] https://insights.axtria.com/hubfs/thought-leadership-whitepapers/Axtria-Insights-White-Paper-The-Use-of-Natural-Language-Processing-in-Literature-Reviews.pdf