

# *A Recommendation System Based On Sentiment Analysis of User Reviews Using Machine Learning*

Sruthy Chandran  
Department of Computer Applications  
Amal Jyothi College of Engineering,  
Kanjirappally, India  
sruthychandran2023b@mca.ajce.in

Lisha Varghese  
Department of Computer Applications  
Amal Jyothi College of Engineering,  
Kanjirappally, India  
lishavarghese@amaljyothi.ac.in

**Abstract:** Sentiment analysis, also known as Opinion Mining, is an essential aspect of Natural Language Processing (NLP) that involves extracting people's emotions, attitudes, opinions, and sentiments from text based on polarity, i.e., positive or negative. It is a highly researched area in NLP and data mining, with widespread applications in businesses and society. The Single Value Decomposition (SVD) algorithm can be used for sentiment analysis in review systems. SVD breaks down a matrix into its constituent parts and helps to identify the most significant features in the data, which can be used to classify reviews based on their polarity. This technique is widely used in social media and other review-based systems to understand people's sentiments towards a particular product or service.

**Keywords —** Single Value Decomposition, Machine learning Algorithm, Sentiment Analysis, Natural Language Processing (NLP)

## I. INTRODUCTION

Sentiment analysis, also known as opinion mining, is a natural language processing technique used to determine the emotional tone or sentiment expressed in a piece of text, such as a tweet, review, or article. It involves using computational methods to identify and extract subjective information from the text, such as positive, negative, or neutral sentiment, as well as the intensity of the sentiment.

Sentiment analysis can be useful in a variety of applications, such as social media monitoring, market research, customer feedback analysis, and brand reputation management. It can help businesses and organizations to gain insights into customer sentiment and preferences, improve customer satisfaction, and make data-driven decisions.

Reviews are a written or spoken evaluation or assessment of a product, service, experience, or performance, typically provided by a customer, user, or consumer. Reviews are a form of feedback that provide information and insights about a particular product or service, including its strengths, weaknesses, and overall quality.

Reviews can be found on a variety of platforms, including e-commerce websites, social media, and review sites such as

Yelp, TripAdvisor, or Google Reviews. They can help other potential customers make informed decisions about whether or not to purchase a product or service, as well as provide valuable feedback to the business or organization about their performance.

Reviews can vary in length and level of detail, from a brief star rating to a more detailed written review. They can also include images or video content, as well as other contextual information such as the date of the review, the reviewer's location, and their purchase history.

## II. LITERATURE REVIEW

Sentiment analysis played a dominant role in the area of researches done by many researchers; there are many methods to carry out sentiment analysis. Many researches are going on to find out better alternatives due to its importance.

Soudamini Hota & Sudhir Pathak[1] compares the K-Nearest neighbour [KNN] algorithm & Support Vector Machine[SVM]. It shows that the analysis is improved further by using KNN algorithm to train the classifier than the SVM technology. It is improved further by employing distance weighted KNN algorithm that involves associating weights with the nearest neighbors based on their proximity to the data point.

Lopamudra Dey[2] compares the KNN algorithm with the Naive Bayes Classification. Their experimental results show that the classifiers yielded better results for the movie reviews with the Naïve Bayes, approach. It giving above 80% accuracies and outperforming than the K-NN approach. DOI : 10.5281/zenodo.7042432 National Conference in Emerging Computer Applications (NCECA2019) Emerging Trends in Engineering Research ISBN-978-93-5267-349-0 29

Surya Prakash Sharma[3] says that, first extracts the feature,

DOI: 10.5281/zenodo.7950006

ISBN: 978-93-5906-046-0@2023, Dept. of Computer Applications, Amal Jyothi College of Engineering Kanjirappally, Kottayam

modifier and opinion from the dataset and then using clustering mechanism divide them into discrete clusters by user's opinion. A feature wise opinion mining system to determine the polarity of the opinions in reviews documents using Senti-WordNet.

Devika M D[4] had made a comparative study of different approaches in sentiment analysis. She conclude that, in the internet world majority of people depends on social networking sites to get their valued information, analyse he reviews from these blogs will yield a better understanding and help in their decision-making.

Wararat Songpan[5] uses two methods to calculate the sentiment analysis called the Naive Bayes classification & Decision tree algorithm. The classifier model has calculated probability that shows value of trend to give the rating using naive bayes techniques, which gives correct classifier to 94.37% ~ compared with decision tree Techniques.

Vidisha M. Pradhan [6] had done a Survey on Sentiment Analysis Algorithms for Opinion Mining. Dictionary based technique takes less processing time even though the accuracy is not up to the mark. However, the supervised learning approach provides better accuracy. From the survey, it summarizes the supervised techniques provide better accuracy compared to dictionary-based approach.

[7] Chowanda A, [7] social media conversation" discusses the application of machine learning techniques to recognize emotions expressed in social media conversations. The authors highlight the importance of emotion recognition for various applications, such as customer service, mental health, and social media analysis. The study presents an evaluation of different machine learning algorithms, including Naive Bayes, Support Vector Machine, Random Forest, and Multilayer Perceptron, on a dataset of social media conversations.

[8] The proposed approach involves pre-processing the data using techniques such as tokenization, stemming, and stop- word removal. Word embeddings are then generated using the Skip-gram model, which is trained on a large unlabeled corpus of Arabic text. The authors use the pre-trained word embeddings to initialize the embedding layer of a CNN, which is trained on a labeled dataset of Arabic tweets. The evaluation is conducted using metrics such as accuracy, precision, recall, and F1 score.

[9] The authors provide a comprehensive survey of the literature on sentiment analysis in the education sector, highlighting the various applications of sentiment analysis in this field, such as student feedback analysis, teacher evaluation, social media monitoring, and sentiment-based recommendation systems. They also discuss the challenges and limitations of sentiment analysis in education, such as the

lack of labeled datasets, language variability, and ethical issues related to data privacy.

[10] Analysis, highlighting the challenges associated with negation in sentiment analysis. The authors present a dataset of English sentences annotated with negation tags and sentiment labels, which they use to evaluate the performance of several state-of-the-art sentiment analysis techniques, including rule-based methods, machine learning algorithms, and deep learning models.

### III. METHODOLOGY

A product recommendation system based on user-item interactions. It takes a request object as input, which contains information about the user who is currently logged in. The system uses this information to generate personalized product recommendations for the user.

First, the code retrieves all the reviews submitted by the user and creates a list of the products that the user has reviewed. These products will be used as a basis for generating recommendations.

Next, the code creates an interaction matrix that represents the interactions between users and products. The matrix is populated with 1s for each user-product interaction, indicating that the user has purchased or shown interest in the product. This matrix is then factorized using Singular Value Decomposition (SVD) to obtain user and item matrices that can be used to predict ratings for user- item pairs.

The SVD algorithm decomposes the interaction matrix into three matrices: U, S, and Vt. The U matrix represents the users, the Vt matrix represents the items, and the S matrix contains the singular values. The user and item matrices are calculated as follows:

- User matrix:  $U \times S$ , where U is the user matrix and S is the diagonal matrix containing the singular values.
- Item matrix: Vt, the transpose of the item matrix.

The code then uses the user and item matrices to predict ratings for each user-item pair. It calculates the dot product of the user matrix and the transpose of the item matrix to obtain a vector of predicted ratings for each user. The predicted ratings are used to identify the top recommended products for the user. Finally, the code returns the top recommended products to the recommendations.html template for rendering on the page.

This code is an implementation of a recommendation system in Python using Singular Value Decomposition (SVD). The goal of the recommendation system is to suggest products to users based on their past purchases or reviews.

The code starts by getting the ID of the current user from the request object. Then, it retrieves all the reviews submitted by the user using the ReviewRating model from the database. Next, the code creates an empty list called product\_list and populates it with the products that the user has reviewed.

After that, the code creates an empty dictionary called recommendations to store the recommended products and their

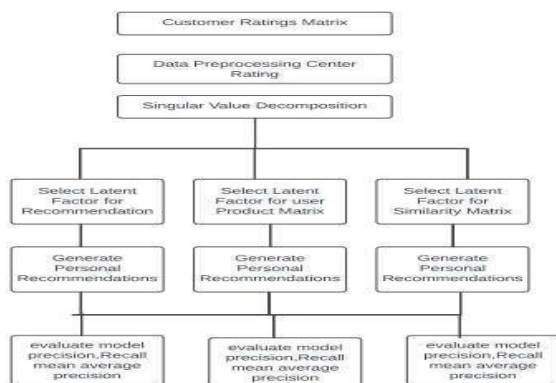
scores. It then retrieves all the orders placed by the user using the OrderPlaced model from the database and stores them in a list called order\_list.

To create a user-item interaction matrix, the code retrieves all the unique user IDs and product IDs from the order\_list. It then creates a 2D numpy array called interaction\_matrix with the number of rows equal to the number of unique users and the number of columns equal to the number of unique products. The values in the interaction\_matrix are initialized to zero.

Next, the code loops through each order in order\_list. For each order, it finds the row and column index in the interaction\_matrix corresponding to the user and product ID of the order, respectively, and sets the value at that position to 1. After creating the user-item interaction matrix, the code performs Singular Value Decomposition (SVD) on the matrix. SVD is a matrix factorization technique that breaks down a matrix into three matrices: a left singular matrix, a diagonal singular value matrix, and a right singular matrix.

The code uses the svds function from the scipy.sparse.linalg module to perform SVD on the interaction\_matrix. The k parameter specifies the number of singular values to compute. In this case, k is set to 2. The output of svds is three matrices: U, s, and Vt.

The code then creates a user matrix by multiplying the left singular matrix U with the diagonal singular value matrix S, and an item matrix by transposing the right singular matrix Vt. Next, the code predicts ratings for each user-item pair by taking the dot product of the user matrix and the transpose of the item matrix. The predicted ratings for the current user are then retrieved and stored in an array called predicted\_ratings. The code then finds the top three recommended products by sorting the predicted ratings array in descending order and retrieving the indexes of the top three products. The actual



product IDs are then retrieved from the products list using these indexes and stored in a list called top\_recommendations. Finally, the code renders a template called recommendations.html and passes the top\_recommendations list as a context variable. This template can be customized to display the recommended products in a user-friendly manner. In summary, this code implements a recommendation system

using Singular Value Decomposition (SVD) to suggest products to users based on their past purchases or reviews.

The system creates a user-item interaction matrix, performs SVD on the matrix, and uses the resulting matrices to predict ratings for each user-item pair. The top recommended products are then retrieved and displayed to the user. BUILD MODEL

1. All of the necessary packages should be imported.

```
import pandas as pd
from sklearn.preprocessing import MultiLabelBinarizer
from sklearn.metrics.pairwise import cosine_similarity
```

1. Load the previous purchase data into a panda's data frame

Transform the data using MultiLabelBinarizer to get a binary matrix. Calculate cosine similarity between each user. Define a function to recommend items to a user based on their previous purchases. Get the top 3 most similar users. Create an empty dictionary to store the count of items purchased by other users

```
# Define a function to recommend items to a user based on their previous purchases
def recommend_items(user):
    # Get the index of the user
    user_index = data[data['user'] == user].index[0]
```

2. Calculate cosine similarity between each user. Define a function to recommend items to a user based on their previous purchases. Get the top 3 most similar users. Create an empty dictionary to store the count of items purchased by other users

```
# Calculate the cosine similarity between the user and all other users
user_similarities = cosine_sim[user_index]
# Get the top 3 most similar users
top_users = user_similarities.argsort()[::-1][1:]

# Create an empty dictionary to store the count of items purchased by other users
item_count = {}
```

3. Sort the items by count in descending order. Return the top 3 items as recommendations

```
# Sort the items by count in descending order
sorted_items = sorted(item_count.items(), key=lambda x: x[1], reverse=True)

# Return the top 3 items as recommendations
recommendations = [item[0] for item in sorted_items][0:3]

return recommendations
```

4. Test the function by recommending items to a user

```
[ ]
# Test the function by recommending items to a user
user = 'user1'
print(f"Recommended items for {user}: {recommend_items(user)}")

Recommended items for user1: ['Grilles', 'protein puddings', 'quark']
```

#### IV. CONCLUSION

The Singular Value Decomposition (SVD) method has proven to be an effective approach for sentiment analysis.

We were able to identify hidden variables that contribute to positive, negative, or neutral sentiment by decomposing the review matrix into its constituent segments. This method was particularly useful for dealing with the sparsity and high-dimensionality of review data. This paper explores various methods of sentiment analysis and aims to develop a more effective approach for categorizing user reviews. Given the widespread use of social networking sites for obtaining information, analysing reviews from these platforms can provide valuable insights to aid decision-making.

#### VI. REFERENCES

- [1] Soudamini Hota, Sudhir Pathak, "KNN classifier based approach for multi-class sentiment analysis of twitter data"- International Journal of Engineering & Technology, 7 (3) (2018) 1372-1375
- [2] Lopamudra Dey, Sanjay Chakraborty, Anuraag Biswas, "Sentiment Analysis of Review Datasets Using Naïve Bayes, and K-NN Classifier," I.J. Information Engineering and Electronic Business, 2016.
- [3] Surya Prakash Sharma, Dr Rajdev Tiwari, Dr Rajesh Prasad, "Opinion Mining and Sentiment Analysis on Customer Review Documents"- A Survey, International Conference on Advances in Computational Techniques and Research Practices-Vol. 6, Special Issue 2, February 2017
- [4] Devika M D, Sunitha C, Amal Ganesha, "Sentiment Analysis: A Comparative Study On Different Approaches"- Procedia Computer Science 87 (2016).
- [5] Wararat Songpan, "The Analysis and Prediction of Customer Review Rating Using Opinion Mining"-, 2017 IEEE SERA 2017, June 7-9, 2017, London, UK.
- [6] Vidisha M. Pradhan, Jay Vala, Prem Balani, "A Survey on Sentiment Analysis Algorithms for Opinion Mining"-, International Journal of Computer Applications (0975 – 8887) Volume 133 – No.9, January 2016
- [7] Chowanda A, Sutoyo R, Tanachutiwat S et al (2021) Exploring textbased emotions recognition machine learning techniques on social media conversation. *Procedia Comput Sci* 179:821–828
- [8] Dahou A, Xiong S, Zhou J, Haddoud MH, Duan P (2016) Word embeddings and convolutional neural network for arabic sentiment classification. In: *Proceedings of COLING 2016, the 26th international conference on computational linguistics: Technical papers*, pp 2418–2427
- [9] Archana Rao PN, Baglodi K (2017) Role of sentiment analysis in education sector in the era of big data: a survey. *Int J Latest Trends Eng Technol* 22–24.
- [10] Mukherjee P, Badr Y, Doppalapudi S, Srinivasan SM, Sangwan RS, Sharma R (2021) Effect of negation in sentences on sentiment analysis