INDIAN INSTITUTE OF INFORMATION TECHNOLOGY

DESIGN & MANUFACTURING, JABALPUR

Machine Learning Project:

Sentiment analysis on Amazon reviews using NLP

Ank Soni - 2019204

Hritik Dhoke - 2019069

Under the guidance of our instructor:

Dr. Kusum Kumar Bharti
Dept. of Computer Science and Engineering,IIITDM Jabalpur

Abstract

This is the age of information explosion. Numbers, text, video, and audio are all information. Extracting the instructive information we are interested in from a large amount of data is crucial. In this article, we built an evaluative model based on amazon product reviews. The review content is divided into good, medium, and bad to express the positive, negative, and neutral emotions. From the basic information about the data, we find that the number of reviews is increasing exponentially. More and more consumers choose to shop online, among which beauty and baby products are more popular among consumers. To understand consumers' preferences, we use TextBlob, a natural language processing library in Python, to do word frequency statistics and sentiment analysis on product reviews. It is found from the word cloud that consumers are most concerned about the quality and price of products.

INTRODUCTION

With the growth of e-commerce and the improvement of logistics service levels, more and more people choose online shopping as a fast and convenient way of shopping. With the development of big data and the massive increase of data, data plays an essential role in business decision-making and company development

This article will discuss the impact of reviews on online product sales in an e-commerce environment. Judge how much customers like the product through product ratings and review content

DATASET DESCRIPTION

In this study, the Amazon review dataset is used from Amazon Customer Reviews Dataset - Amazon Music instruments reviews.

This file has reviewer ID , User ID, Reviewer Name, Reviewer text, helpful,

Summary(obtained fro Reviewer text),Overall Rating on a scale 5, Review time Description of columns in the file:

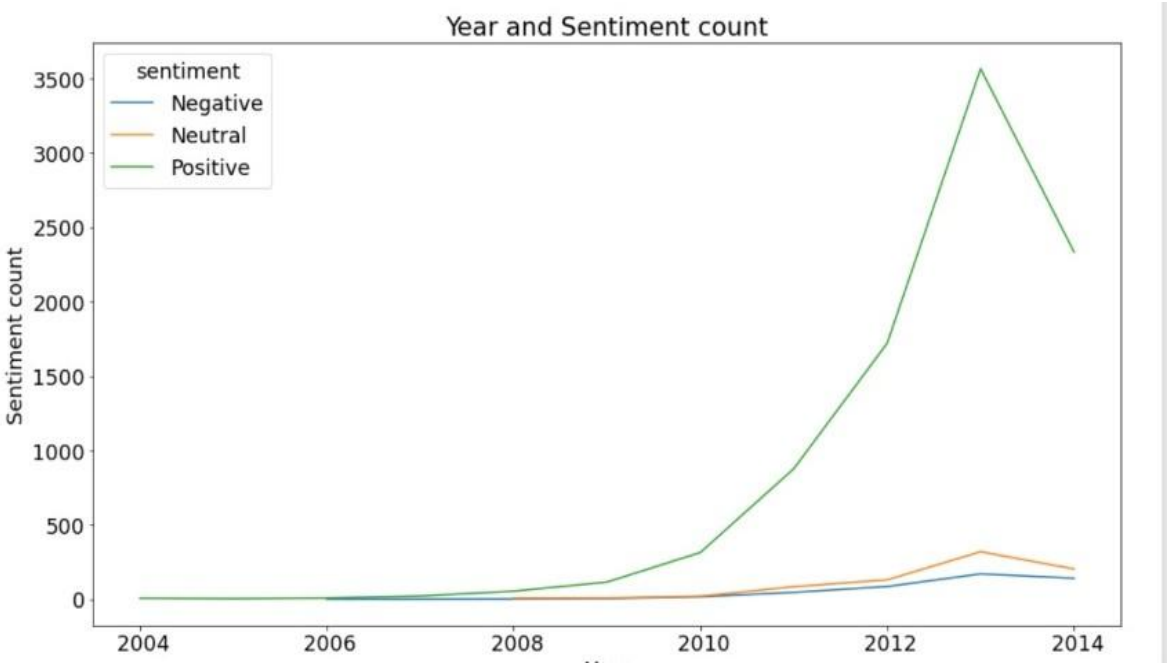
- reviewerID - ID of the reviewer, e.g. A2SUAM1J3GNN3B
- asin - ID of the product, e.g. 0000013714

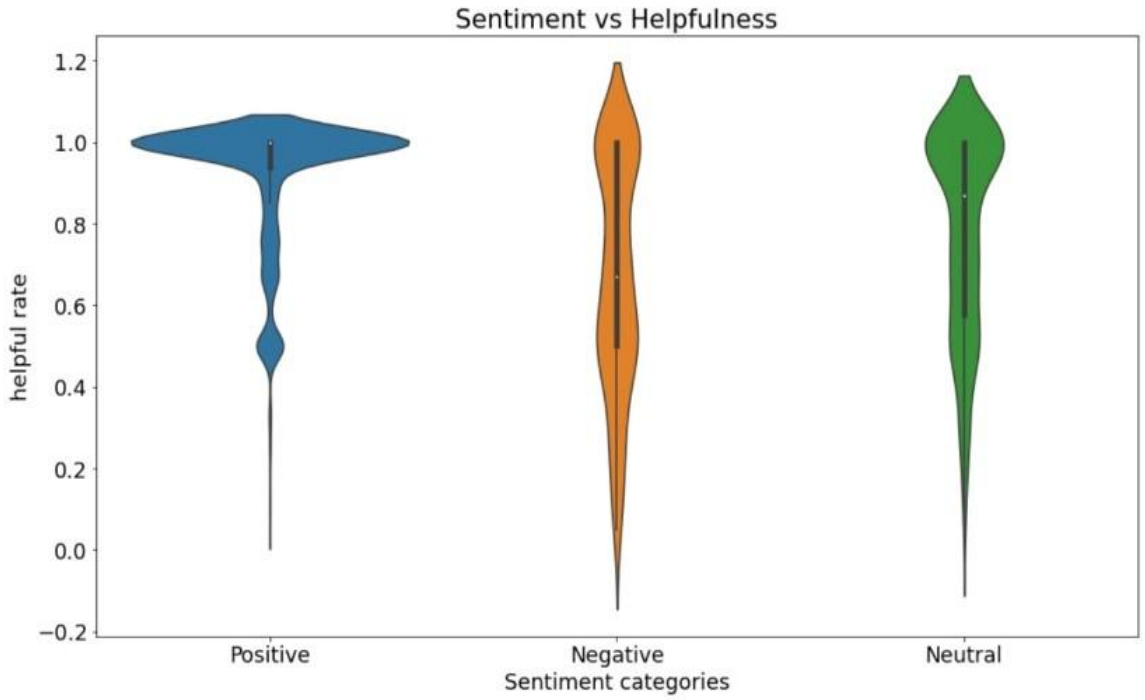
- reviewerName - name of the reviewer
- helpful - helpfulness rating of the review, e.g. 2/3
- reviewText - text of the review
- overall - rating of the product
- summary - summary of the review
- unixReviewTime - time of the review (unix time)
- reviewTime - time of the review (raw)

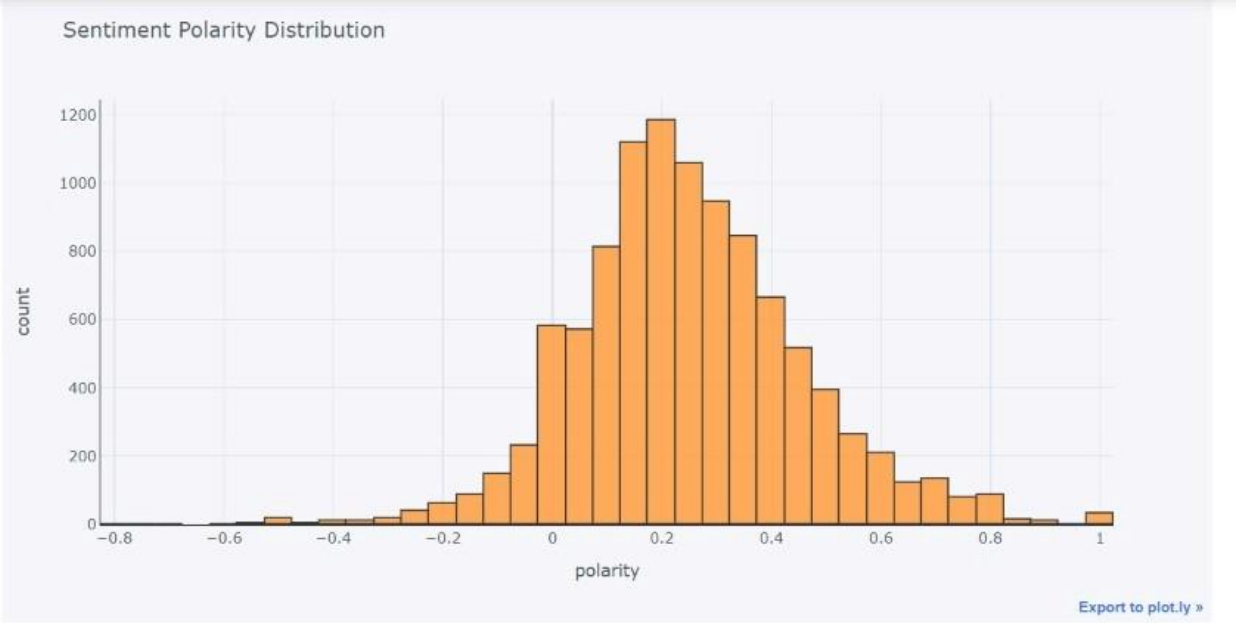
Data Pre-Processing steps

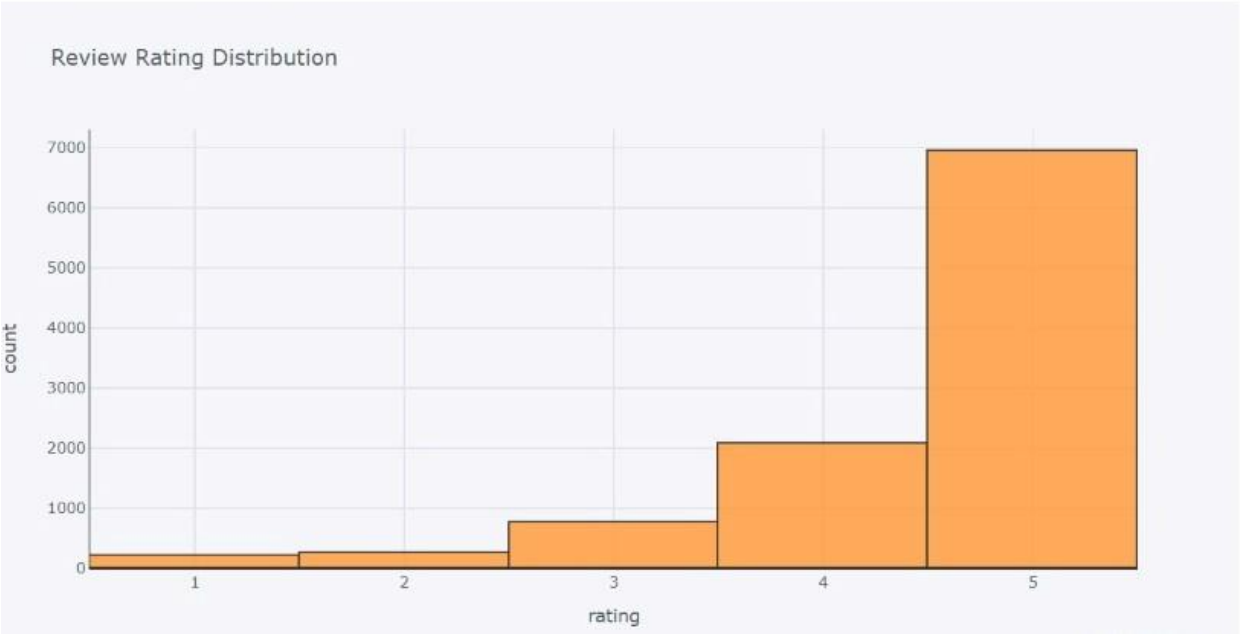
1. Concatenating review text and summary
2. Cleaning the data
3. Review text-Punctuation Cleaning
4. Review text-Stop words

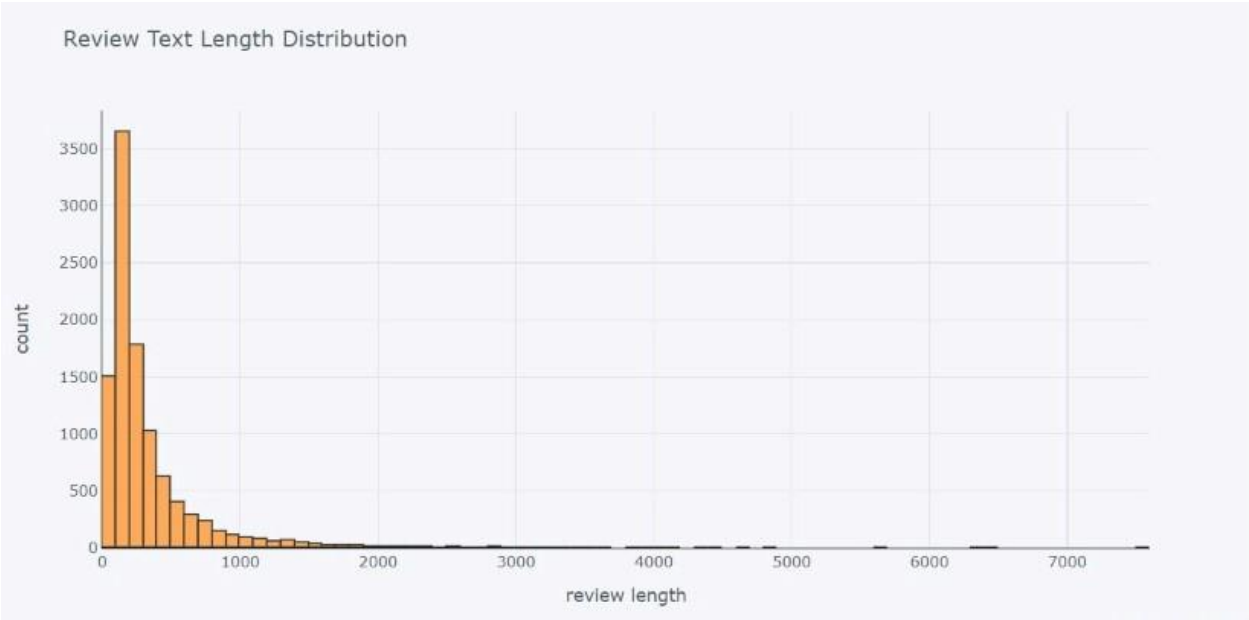
Data Visualization

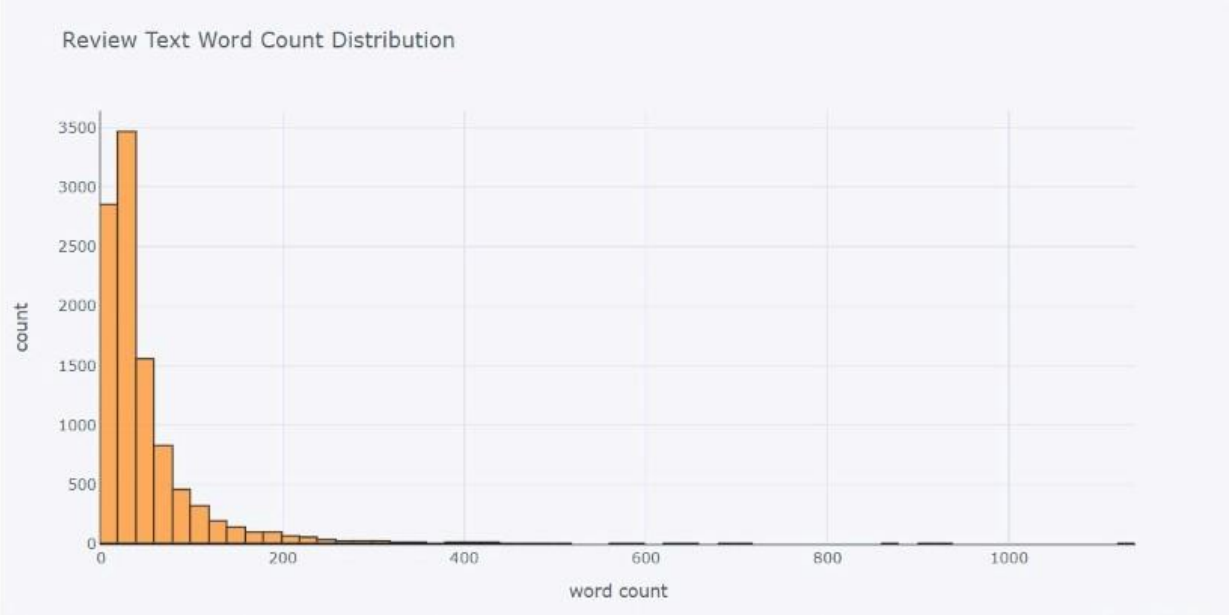


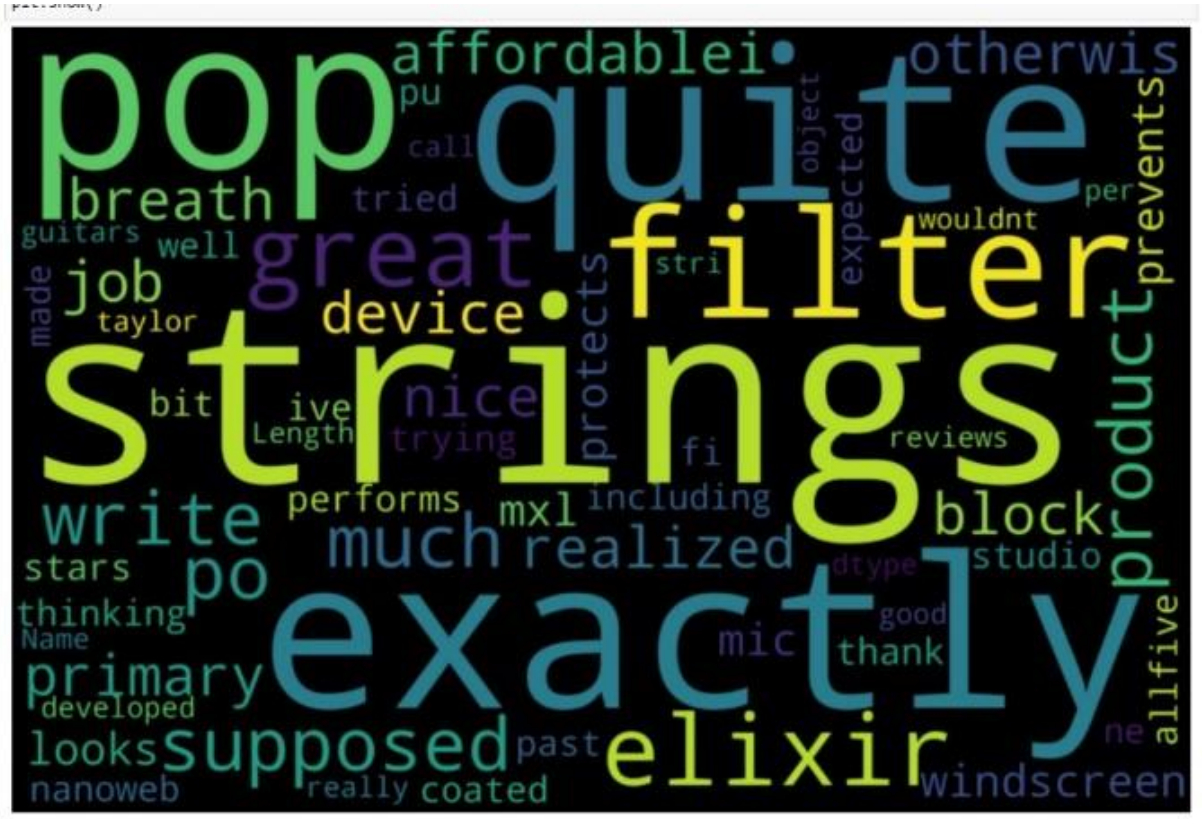

Sentiment vs Helpfulness
helpful rate
1.2
1.0
0.8
0.6
0.4
0.2
0.0
−0.2
Positive
Negative
Neutral
Sentiment categories

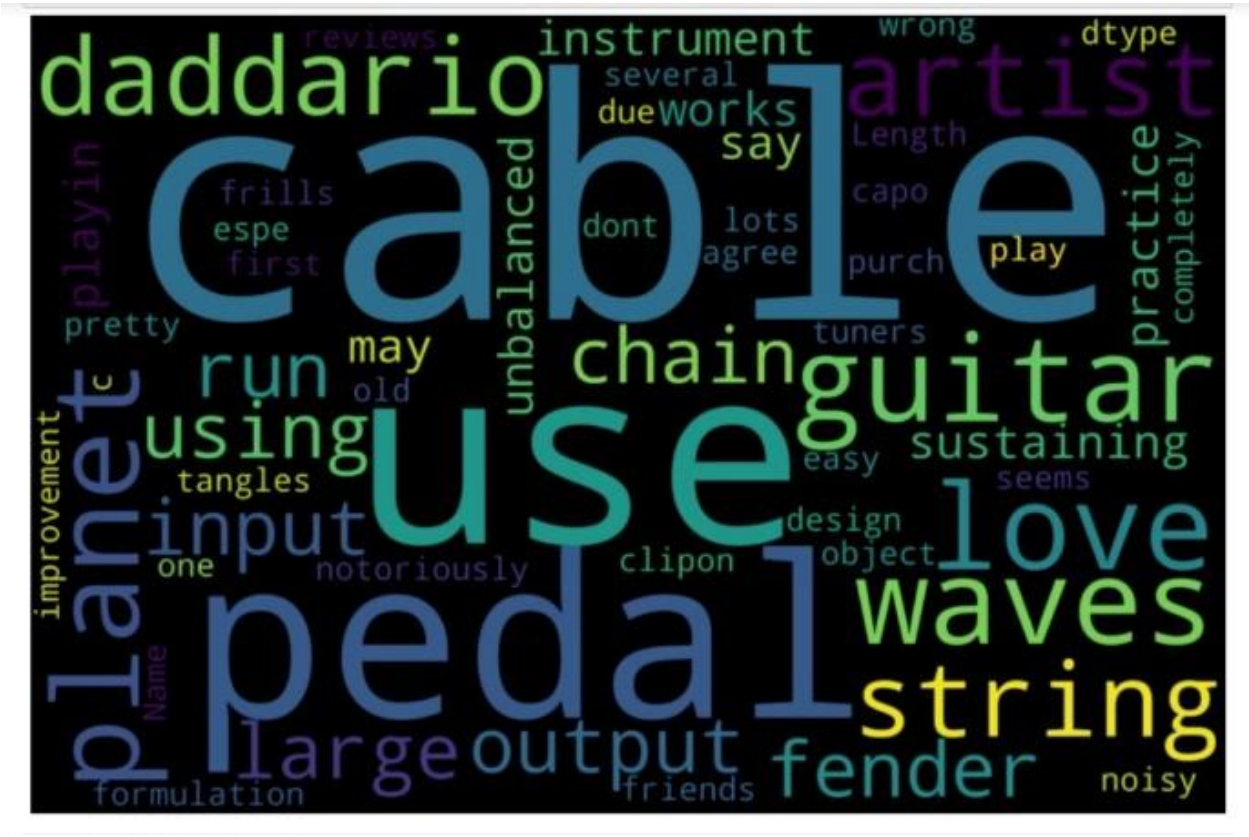

Sentiment Polarity Distribution
count
1200
1000
800
600
400
200
0
−0.8
−0.6
−0.4
−0.2
0
0.2
0.4
0.6
0.8
1
polarity
Export to plot.ly »


Review Rating Distribution
count
7000
6000
5000
4000
3000
2000
1000
0
1
2
3
4
5
rating


Review Text Length Distribution
count
3500
3000
2500
2000
1500
1000
500
0
0
1000
2000
3000
4000
5000
6000
7000
review length


Review Text Word Count Distribution
count
word count
0
500
1000
1500
2000
2500
3000
3500
0
200
400
600
800
1000


pop
quite
affordablei
otherwis
pu
call
object
expected
breath
tried
per
wouldnt
guitars
well
great
filter
prevents
made
job
stri
taylor
device
product
strings
bit
ive
nice
protects
Length
trying
fi
reviews
write
performs
mxl
including
block
much
realized
stars
po
dtype
studio
thinking
good
Name
mic
thank
exactly
primary
developed
looks
supposed
past
elixir
allfive
ne
nanoweb
really
coated
windscreen



We ‘ve used textblob to get and plot this image of words.

Feature engineering

We have engineered essential features like :

Polarity: We use Textblob for for figuring out the rate of sentiment . It is between [-1,1]where -1 is negative and 1 is positive polarity

Review length: length of the review which includes each letters and spaces

Word length: This measures how many words are there in review

Extracting Features from Cleaned reviews

Before we build the model for our sentiment analysis, it is required to convert the review texts into vector formation as the computer cannot understand words and their sentiment. In this project, we are going to use the TF-TDF method to convert the texts in vectors.

1. Label encoding
2. Stemming

Handling Imbalance target feature-SMOTE

In our target feature, we noticed that we got a lot of positive sentiments compared to negative and neutral. So it is crucial to balanced the classes in such situatio. Here I use SMOTE(Synthetic Minority Oversampling Technique) to balance out the imbalanced dataset problem.It aims to balance class distribution by randomly increasing minority class examples by replicating them.

Model Building: Sentiment Analysis

As we have successfully processed the text data, not it is just a normal machine learning problem. Wherefrom the sparse matrix we predict the classes in the target feature.

```
Logistic Regression Test Accuracy: 0.8809084541929313

Decision Tree Test Accuracy: 0.8193160874706511

KNN Test Accuracy: 0.8716498592581203

SVC Test Accuracy: 0.879641302759224

Naive Bayes Test Accuracy: 0.8034287682855306
```

From the results, we can see logistic regression outdone the rest of the algorithms and all the accuracies from the results are more than 80%. We choose Logistic Regression and SVC for hyperparameter tuning.

Hyperparameter tuning:

We use regularization parameters and penalty for parameter tuning for logistic regression and , C val and gamma for Support Vector Classification using GridSearchCV.

Also we have used Gray Wolf Optimization Algorithm for Hyperparameter tuning for SVC

Results:

After HyperParameter Tuning:

Classifier	Initaial Accuracy	After GridCV	After GWO
Logistic Regression	0.88	0.94	-
SVC	0.87	0.97	0.98

Conclusion and Future work

Online comments have received widespread attention because of their importance. In this research, we used TF-IDF to count the word frequency, accurately analyzed the

Amazon review data set, studied the products, and obtained people's preferences for online shopping products.

When discussing the product's future trend, we got points for the future development of the product. In this article, only part of the data is displayed. A linear model should be established between the number of reviews and the score of future product development. A correlation test should be conducted to find the corresponding relationship between the number of reviews and the number of studies, and the future product development score.

References

1. Sentiment analysis of Amazon product reviews based on NLP - https://doi.org/10.1109/AEMCSE51986.2021.00249

2. Hybridizing Gray Wolf Optimization (GWO) with Grasshopper Optimization Algorithm (GOA) for text feature selection and clustering - https://doi.org/10.1016/j.asoc.2020.10665