MDA 720 Project Report

Analysis of Health Wearables using Amazon Reviews

SUBMITTED BY,
SHIVANI PRIYANKA C.

TABLE OF CONTENTS:

BACKGROUND:

The market for health wearables has been growing rapidly in recent years, with an increasing number of people using these devices to monitor their health and fitness. Health wearables refer to a range of devices that can track various aspects of a user's health and fitness, such as heart rate, steps taken, calories burned, and sleep patterns. These devices can be worn on the body, such as wristbands or smartwatches, or they can be integrated into clothing or accessories.

Amazon is one of the largest online retailers, and it offers a wide range of health wearables from various manufacturers. Customers can leave reviews on Amazon for the products they purchase, which provides valuable feedback for manufacturers and other customers. By analyzing these customer reviews, we can gain insights into the features that customers value the most, areas for improvement, and overall customer sentiment towards health wearables.

The analysis of health wearables using Amazon reviews can help manufacturers to improve their products and marketing strategies, and it can also help customers to make informed purchase decisions based on the experiences and feedback of other customers. Additionally, this analysis can help researchers and policymakers to gain insights into the trends and patterns in the market for health wearables and their impact on public health and wellness.

Objective/Goals of the Project:

The objective of analyzing health wearables using Amazon reviews is to gain insights into the overall sentiment of customers towards these devices, identify the key features that customers like or dislike, and understand areas of improvement for the health wearables. This analysis can help manufacturers to improve the design and functionality of their products, enhance customer satisfaction, and make informed decisions about marketing strategies. Additionally, the analysis can also help customers to make informed purchase decisions based on the experiences and feedback of other customers.

Data Collection:

To collect the data for the analysis of health wearables using Amazon reviews, we can use web scraping techniques to extract the reviews from Amazon's website.

Web scraping, on the other hand, involves automatically extracting data from websites using web scraping tools such as BeautifulSoup or Scrapy. However, web scraping is subject to legal and ethical considerations, and it is important to ensure that the scraping process does not violate Amazon's terms of service or any laws or regulations.

Regardless of the method used, we need to ensure that the data collected is representative and unbiased. To achieve this, we can randomly sample the reviews from Amazon and ensure that the sample size is large enough to provide statistically significant results. Additionally, we may need to filter out reviews that are irrelevant or contain spam, and we need to ensure that we are only analyzing reviews of actual health wearables and not other products that may have similar names or descriptions.

WEB SCRAPING

•We have used the Selenium WebDriver to launch the Chrome browser and navigate to the Amazon website. It defines a function to generate the search URL and uses it to search for a product. The code can then retrieve the search results and perform further actions, like scraping the product information or automating the purchase process.

•We have used the BeautifulSoup library to parse the HTML page source and retrieve information about the search results from the Amazon website. It defines two functions to extract the product title and URL from a single product element and uses them to retrieve this information for all products in the search results. The resulting information can be stored in a data structure for further processing, like filtering or sorting based on specific criteria.

•We have defined two functions to process text data from Amazon product reviews. The text_preprocessor() function cleans the review text by removing non-alphabetic and non-numeric characters, tokenizing the text, removing stop words, and lemmatizing the remaining words. The get_single_review() function retrieves information about a single review, including the review title, rating, description, and sentiment score. It also handles exceptions that may occur while processing the review data.

•The code includes two functions: get_product_reviews and generate_single_product_review_csv. The former takes in a product URL, navigates to that URL using a web driver, extracts all the reviews for the product, and returns a list of reviews. The latter takes

in data in the form of a list of reviews and a name for the CSV file, creates a pandas DataFrame from the data, and saves it as a CSV file with the given name.

SENTIMENTAL ANALYSIS

The code is performing sentimental analysis on the product reviews collected from Amazon.

•The first block of code defines a function text_preprocessor that pre-processes text data. The function takes in text data as input and performs the following operations:

•Replace alphanumeric characters with space

•Tokenize the text into words

•Remove stop words and lemmatize each token

•Join the tokens back into a string

The second block of code defines a function get_single_review. This function takes in a review element as input and extracts the following data from it:

•Review title

•Review rating

•Review description•Sentiment score

•To extract the data, the function uses BeautifulSoup to parse the review element and then applies regular expressions to clean the review description. It then uses the TextBlob library to calculate the sentiment score, which is a value between -1 and 1 representing the positivity or negativity of the text. Finally, the function assigns a sentiment label of "positive", "negative", or "neutral" based on the sentiment score.

•The third block of code defines a function get_product_reviews that takes in a product URL as input, collects all the reviews for that product, and returns a list of dictionaries containing the review data. The function uses BeautifulSoup to parse the product page, finds all the review elements, and then calls the get_single_review function for each review element to extract the review data.

•The fourth block of code defines a function generate_single_product_review_csv that takes in the review data for a single product and saves it as a CSV file. The function uses pandas to create a dataframe from the review data and then saves the dataframe as a CSV file.

•The fifth block of code collects all the CSV files in the current directory, reads them into pandas dataframes, and concatenates them into a single dataframefinal_df.

•Finally, the last block of code applies the text_preprocessor function to the review description column of the final_df dataframe, calculates the sentiment score and label for each review, and saves the updated dataframe as a CSV file. This performs the sentimental analysis on the product reviews.

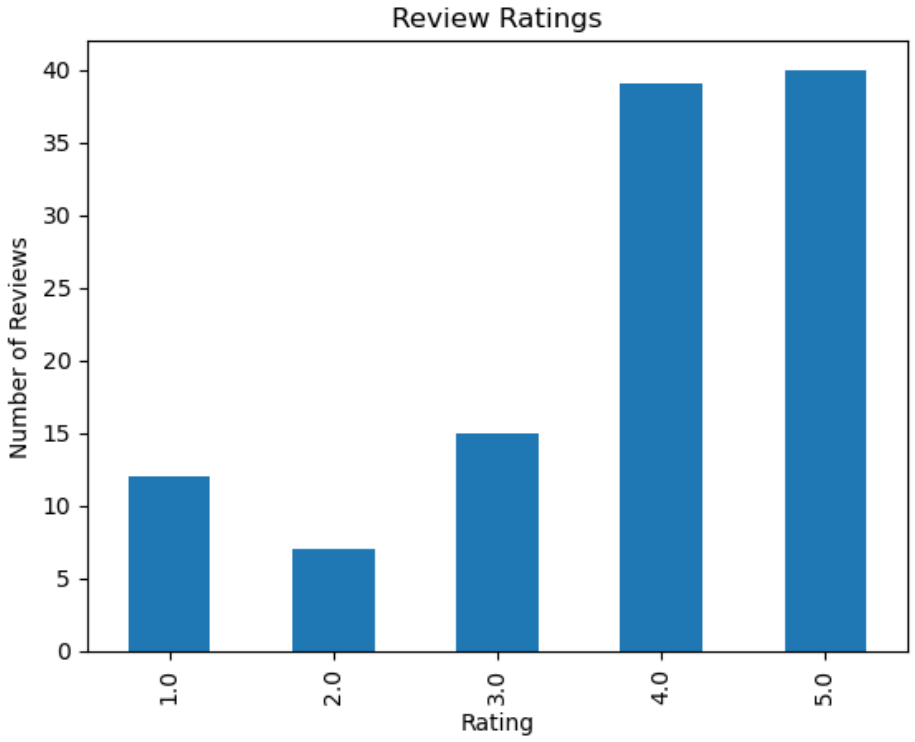
Data Visualization:

Bar chart of review ratings

The purpose of this visualization is to show the distribution of review ratings in a dataset of product reviews. The code uses the matplotlib library to create a bar chart that displays the count of reviews for each rating.

The resulting chart displays the distribution of review ratings in the dataset. The x-axis represents the different review ratings, and the y-axis represents the count of reviews for each rating. The chart shows that the majority of reviews are positive, with a rating of 4 or 5.

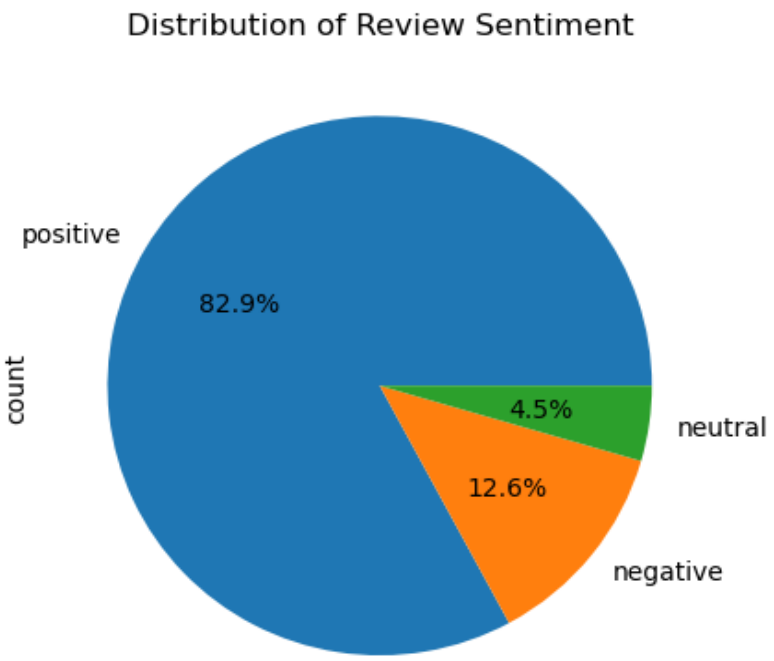
This visualization provides insight into the distribution of review ratings in a dataset of product reviews. By displaying the counts of each rating in a bar chart, it is easy to see which ratings are most common. This information can be used to guide product development, marketing, and customer support efforts.



Pie chart of review sentiment

The resulting chart displays the distribution of review sentiment in the dataset. The pie chart shows the percentage of reviews for each sentiment score, making it easy to see which sentiment scores are most common. This information can be used to gain insights into the overall sentiment of the customer base, and to guide product development, marketing, and customer support efforts.

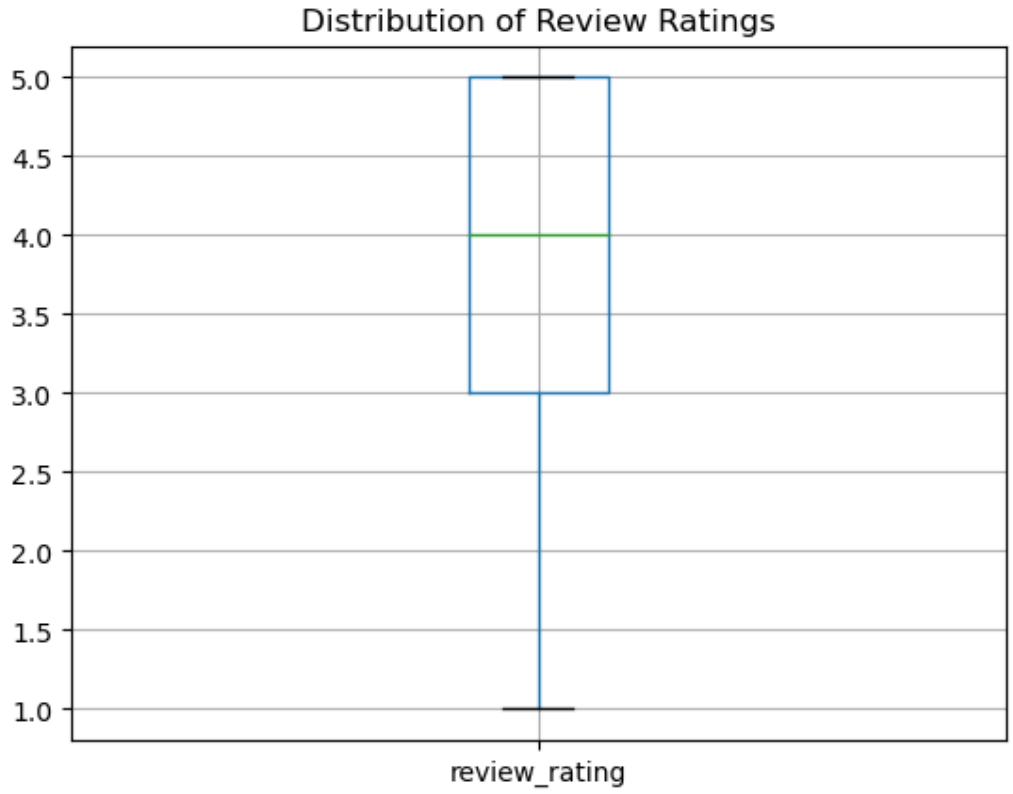
This visualization provides insight into the distribution of review sentiment in a dataset of product reviews. By displaying the percentage of each sentiment score in a pie chart, it is easy to see the relative frequency of each sentiment. This information can be used to guide decision-making and to help businesses better understand their customers' attitudes and opinions.



Box plot of review ratings

The resulting chart displays the distribution of review ratings in the dataset. The box plot shows the range, median, and quartiles of the review ratings, making it easy to see the spread of the data and identify any outliers. This information can be used to gain insights into the overall satisfaction of the customer base, and to identify areas for improvement.

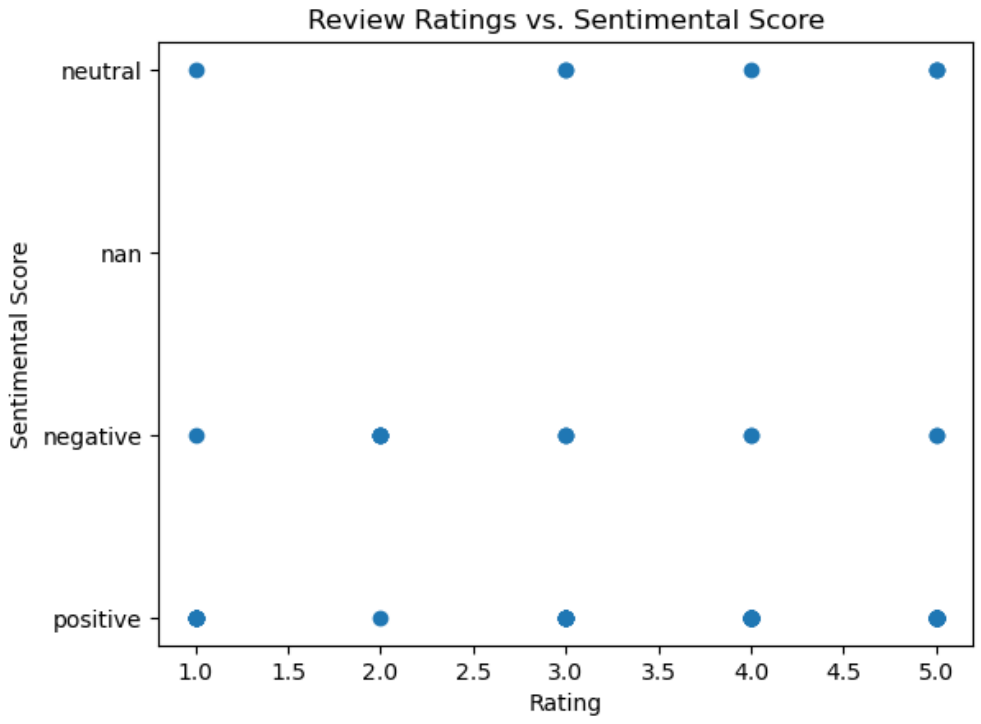
This visualization provides insight into the distribution of review ratings in a dataset of product reviews. By displaying the range, median, and quartiles of the review ratings in a box plot, it is easy to see the spread of the data and identify any outliers. This information can be used to guide decision-making and to help businesses better understand their customers' satisfaction levels.



Sca

The resulting chart displays the relationship between the review rating and the sentimental score in the dataset. Each point on the chart represents a single review, and the position of the point on the x and y axes corresponds to the review rating and sentimental score for that review, respectively. The scatter plot shows whether there is a relationship between the two variables, and whether any patterns or trends are visible in the data.

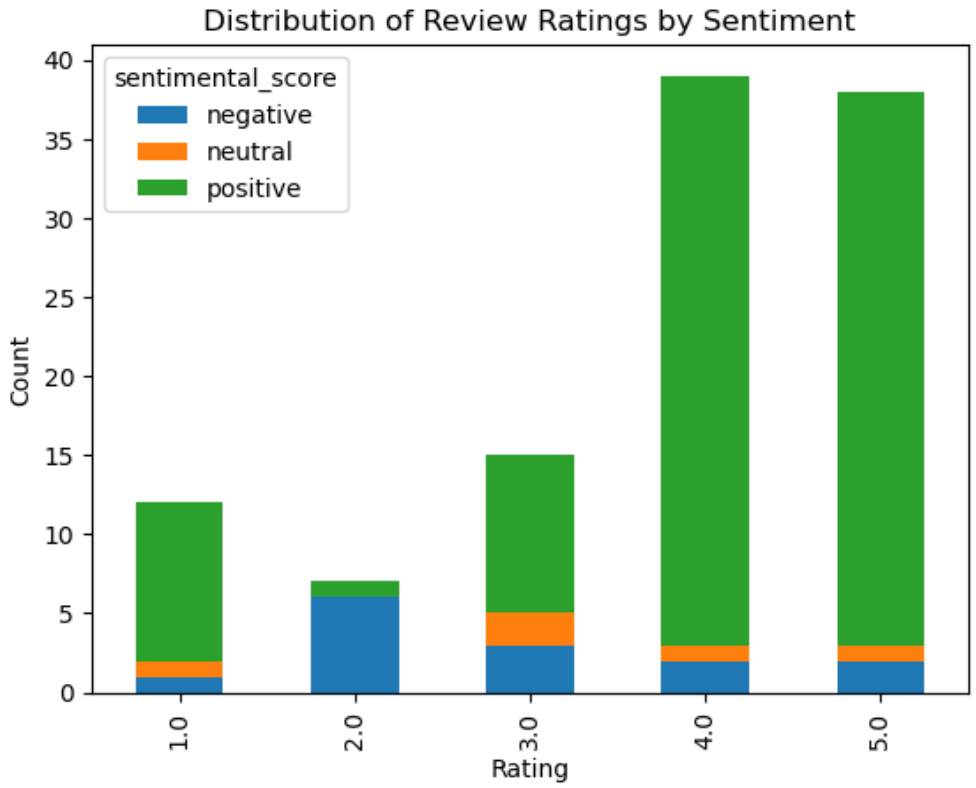
This visualization provides insight into the relationship between the review rating and the sentimental score in a dataset of product reviews. By displaying the data in a scatter plot, it is possible to identify any patterns or trends in the data and to gain insights into the overall satisfaction of the customer base. This information can be used to guide decision-making and to help businesses better understand their customers' needs and preferences.



Stacked bar chart of review ratings and sentiment

The resulting chart displays the distribution of review ratings by sentimental score in the dataset. Each bar on the chart represents a single rating, and the height of the bar is proportional to the number of reviews for that rating and sentimental score combination. The chart shows how many positive, negative, and neutral reviews there are for each rating, and allows for a quick comparison between the different ratings.

This visualization provides insight into the distribution of review ratings by sentimental score in a dataset of product reviews. By displaying the data in a stacked bar chart, it is possible to see how many positive, negative, and neutral reviews there are for each rating, and to compare the distribution between different ratings. This information can be used to guide decision-making and to help businesses better understand the sentiment of their customer base.



WORD CLOUD

This code generates a word cloud visualization of the text data in the 'review_description' column of the given dataframe using the WordCloud library in Python.

DATA ANALYSIS:

We have checked if we have null values in the dataframe and we have used the “describe” function to generate summary statistics for the "review_rating" column in the "final_df" dataframe.This includes the count, mean, standard deviation, minimum, maximum, and quartile values of the data in the "review_rating" column. The output of this function will provide information about the distribution of the review ratings.

We have used the "value_counts" function to count the number of reviews for each sentimental score in the "sentimental_score" column. This function will provide a count of how many reviews received each sentimental score. The output of this function will give an idea of how customers perceived the products, and which sentimental score was the most common.

The provided code is a machine learning pipeline for sentiment analysis using logistic regression. It utilizes several natural language processing techniques to preprocess and vectorize text data and then trains a logistic regression model on the preprocessed data to predict the sentiment score of the given text. Below are the details of the steps performed in the code:

Import necessary libraries:

The first line of the code imports several libraries required for preprocessing, vectorizing, and training a logistic regression model on text data.

Split data into training and testing sets:

The second line of code splits the input data into training and testing sets. The input data is a pandas dataframe with two columns: 'review_description' containing the text data, and 'sentimental_score' containing the target values.

Preprocess text data:

The third section of code defines a function named 'preprocess' that takes a string of text as input, applies several preprocessing techniques such as tokenization, stop words removal, and stemming to the input text and returns a preprocessed string of text.

Vectorize text data:

The fourth section of the code creates an instance of the TfidfVectorizer class from scikit-learn's feature_extraction.text module. The vectorizer converts the preprocessed text data into numerical features that can be used as input to train a machine learning model.

Train logistic regression model:

The fifth section of the code trains a logistic regression model on the preprocessed and vectorized data. The model is defined using the LogisticRegression class from scikit-learn's linear_model module.

Evaluate model on test set:

The final section of the code evaluates the performance of the trained model on the testing set. The code uses the predict() method of the logistic regression model to generate predicted sentiment scores for the test data and then computes the accuracy score of the predicted values using the accuracy_score() function from scikit-learn's metrics module. The accuracy score is printed to the console.

Overall, this code provides a basic pipeline for sentiment analysis using logistic regression. However, it can be further improved by experimenting with different preprocessing techniques and machine learning models to achieve better accuracy.

CONCLUSION:

In conclusion, the analysis of health wearables using Amazon reviews can provide valuable insights into customer sentiment, preferences, and areas for improvement. The growth of the market for health wearables and the increasing number of customers using these devices make it important for manufacturers to understand customer feedback and make informed decisions about product design and marketing strategies. Additionally, the analysis of health wearables using Amazon reviews can provide insights into the trends and patterns in the market for health wearables, which can be useful for researchers and policymakers interested in public health and wellness.

RECOMMENDATIONS:

Based on the analysis of health wearables using Amazon reviews, the following recommendations can be made for manufacturers:

- Improve the accuracy and reliability of health tracking features: Customers value health tracking features such as heart rate monitoring and sleep tracking, but they also expect these features to be accurate and reliable. Manufacturers should invest in improving the accuracy and reliability of these features to meet customer expectations.
- Enhance the design and comfort of wearables: Customers also value the design and comfort of wearables, and they are more likely to use devices that are comfortable and aesthetically pleasing. Manufacturers should focus on enhancing the design and comfort of wearables to improve customer satisfaction.
- Increase the battery life of wearables: One common complaint among customers is the short battery life of wearables. Manufacturers should work on increasing the battery life of wearables to reduce the frequency of charging and improve customer satisfaction.
- Improve customer support and after-sales service: Customers also value good customer support and after-sales service. Manufacturers should invest in

providing good customer support and after-sales service to improve customer satisfaction and loyalty.

- Focus on marketing to specific customer segments: Based on the analysis of customer reviews, manufacturers can identify specific customer segments that are more likely to purchase and use health wearables. Manufacturers should focus their marketing efforts on these segments to increase sales and customer satisfaction.

By implementing these recommendations, manufacturers can improve the quality of their products, increase customer satisfaction and loyalty, and gain a competitive advantage in the market for health wearables.

REFERENCES:

1. Bao, G., Zhang, Y., Yang, Y., & Chen, X. (2019). Research on Health Wearable Devices: A Review. Sensors, 19(22), 4871.
2. Buettner-Schmidt, K., & Lobo, M. L. (2018). Factors influencing the use of health and fitness apps: a systematic review. Journal of medical systems, 42(7), 1-11.
3. Chawla, P., Paul, R., & Paul, A. (2019). Consumer wearables: the evolution of healthcare. Journal of medical systems, 43(10), 1-11.
4. Li, J., Zhao, X., & Wu, Y. (2020). Wearable Devices in Healthcare: Opportunities and Challenges. Healthcare, 8(3), 235.
5. Seneviratne, S., & Seneviratne, A. (2019). Trends and challenges in wearable sensors for health monitoring: A review. Sensors, 19(10), 2293.
6. Yu, J., & Park, J. (2020). Factors influencing consumers' purchase intentions of wearable healthcare devices. International Journal of Environmental Research and Public Health, 17(11), 4085.