Intel Products Sentiment Analysis from Online Reviews

Intel Unnati Industrial Training Program 2024 - Project Report

**Name: Nidhi G**

**USN: 1BI21 CS082**

**Department of Computer Science and Engineering**

**Bangalore Institute of Technology**

**K.R. Road, V.V.Puram, Bangalore-560 004**

**Acknowledgement**

I would like to express my heartfelt gratitude to everyone who contributed to the successful completion of the project titled “INTEL PRODUCTS SENTIMENT ANALYSIS FROM ONLINE REVIEWS.” This endeavor would not have been possible without the unwavering support and guidance from individuals and institutions.

First and foremost, I extend my sincere thanks to Intel Unnati Industrial Training Program for providing me with this incredible opportunity. The exposure to real-world data and industry practices has been invaluable, and I am grateful for the chance to work on such a relevant and challenging project.

I am deeply indebted to my internal guide, Dr. Bhanushree K J, for her mentorship, encouragement, and insightful feedback throughout the project. Her expertise and dedication significantly enriched my learning experience.

I also extend my appreciation to the external guide, Mr. Debdyut Hazra, whose expertise in the field of sentiment analysis was instrumental in shaping the project. His mentoring and practical insights were invaluable.

Furthermore, I would like to thank Bangalore Institute of Technology for fostering an environment conducive to learning and innovation. The support from the Computer Science and Engineering Department, led by Dr. Girija J, has been commendable. Their encouragement and resources allowed me to explore new horizons and enhance my skills.

Lastly, I express my gratitude to Intel for their commitment to nurturing young talent and fostering innovation. This project has been a remarkable learning journey, and I look forward to applying the knowledge gained in my future endeavors.

Warm regards, Nidhi

**Table of Contents**

|  |  |  |
| --- | --- | --- |
| Chapter Number | Chapter name | Page number |
| 1 | Introduction | 4 |
| 2 | Literature Review | 6 |
| 3 | Data Collection | 8 |
| 4 | Data Pre-processing | 10 |
| 5 | Exploratory Data Analysis | 15 |
| 6 | Sentiment Analysis Methodology | 17 |
| 7 | Implementation | 19 |
| 8 | Conclusion | 25 |

**List of Figures**

|  |  |  |
| --- | --- | --- |
| Figure number | Figure Name | Page number |
| 1 | Features and a peak into dataset | 9 |
| 2 | Characteristics of the data | 9 |
| 3 | Checking for duplicate rows | 11 |
| 4 | Duplicated reviews | 11 |
| 5 | Dropping Empty reviews | 11 |
| 6 | Cleaning Column | 11 |
| 7 | Displaying the MRP | 11 |
| 8 | Cleaned Data | 12 |
| 9 | Removing ‘/n’ from review | 12 |
| 10 | Translation | 13 |
| 11 | Stop word removal | 13 |
| 12 | Removing Emojis | 13 |
| 13 | Cleaning data | 14 |
| 14 | Tokenization and lemmatization | 14 |
| 15 | Number of reviews for each product | 15 |
| 16 | Product vs MRP | 15 |
| 17 | MRP vs Average rating | 16 |
| 18 | Applying VADER sentiment analyser | 21 |
| 19 | Applying VADER to entire dataset | 21 |
| 20 | Compound Score vs Rating | 22 |
| 21 | Other scores vs Rating | 22 |
| 22 | Classified Sentiments | 23 |
| 23 | Applying RoBERTa on a text | 23 |
| 24 | Applying RoBERTa on entire dataset | 23 |
| 25 | RoBERTa classification | 24 |
| 26 | Rating vs scores of RoBERTa | 24 |
| 27 | Reviews from different countries | 26 |
| 28 | Product vs Sentiment of the review | 26 |
| 29 | Average review sentiment vs Price | 27 |

**Chapter 1: Introduction**

**1.1 Abstract**

In this study, we perform sentiment analysis on customer reviews of Intel desktop processors from generations 11 to 14. The data is collected by scraping Amazon reviews. After cleaning and preprocessing the data, we conduct exploratory data analysis (EDA) to gain insights into the review distribution and key features.

We then employ two pretrained models, VADER and RoBERTa, to classify the sentiment of each review. By extracting results and conclusions from these models, we analyze the overall sentiment trends across different processor generations. Our findings provide valuable insights for Intel’s product development and marketing strategies.

**1.2 Introduction**

NLP is an exciting field of artificial intelligence that bridges the gap between human communication and computational algorithms. By analyzing text data, NLP algorithms extract meaning, identify patterns, and perform various language-related tasks. One crucial subfield within NLP is sentiment analysis, which systematically quantifies emotions expressed in textual data—whether it’s customer reviews, social media comments, or news articles. Understanding sentiment provides valuable insights for businesses, helping them enhance products, services, and brand strategies.

**Background**

This project is concerned with sentiment analysis field of NLP. User reviews play a crucial role in assessing the quality, performance, and overall satisfaction with Intel processors.

They act as a valuable feedback mechanism for Intel. By collecting reviews directly from customers, Intel gains insights into their experiences, preferences, and pain points. This feedback helps identify areas for improvement and informs future product development.

Analyzing reviews from users with varying backgrounds—such as gamers, professionals, or casual users—provides a holistic view of how different segments perceive Intel processors. This diversity ensures that Intel considers a wide range of opinions and needs.

Users discuss aspects like performance, power efficiency, compatibility, and reliability. Sentiment analysis helps extract sentiments associated with these features, revealing which aspects resonate positively or negatively with users.

Armed with sentiment analysis, Intel can prioritize enhancements. For example, if users consistently praise a particular feature, Intel can invest in optimizing it further. Conversely, if negative sentiments arise around a specific aspect, Intel can address it in future iterations.

**Objective of this project**

The primary goal of this project is to analyze user reviews and classify their sentiments as positive, negative, or neutral. User reviews related to Intel processors from one or more sources. Apply machine learning techniques to automatically categorize each review into one of three sentiment classes: positive (praising), negative (critical), or neutral (neutral observations). By classifying sentiments, we gain insights into how users perceive Intel processors.

Once reviews are classified, we can understand what users are saying about Intel processors. Positive sentiments highlight features that users appreciate, while negative sentiments point out areas for improvement. Neutral sentiments provide context and balance.

Finally, we can draw conclusions from it by analysing the patterns and trends in sentiment over time. These insights can be used to understand about product strengths, weaknesses, and potential enhancements.

**Scope**

The scope of this project involves collecting approximately 1,500 user reviews from Amazon for Intel desktop processors spanning generations 11 to 14. The primary objectives are to classify these reviews based on sentiment (positive, negative, or neutral) and analyze the results. By doing so, we aim to gain insights into users’ perceptions of Intel processors.

**Chapter 2: Literature Review**

**2.1 Overview of Sentiment Analysis**

* Sentiment analysis, also known as opinion mining, has gained prominence within natural language processing (NLP) due to the exponential growth of digital data.The need to extract insights from textual information has become crucial across various domains, including business intelligence, social sciences, and disaster response.
* Sentiment analysis involves identifying, extracting, and analyzing subjective information from text to determine the overall sentiment expressed toward a specific entity, product, topic, or event.
* Text is classified into predefined categories such as positive, negative, or neutral sentiments, or more nuanced emotions and opinions.

**Traditional Approaches and Challenges**

Initially, sentiment analysis relied on machine learning algorithms like support vector machines (SVM), Naive Bayes, logistic regression, and random forests.

However, as the scope of sentiment analysis expanded, new gaps emerged within sentiment classifications. Researchers shifted toward more complex algorithms to address emerging challenges.

Deep learning algorithms, including architectures like convolutional neural networks (CNNs), long short-term memory networks (LSTMs), and recurrent neural networks (RNNs), demonstrated efficacy in handling complex sentiment analysis tasks.

**Challenges and Ethical Considerations**

* Sentiment analysis faces challenges such as deciphering sarcasm and irony, ensuring ethical use, and adapting to new domains.
* The dynamic nature of sentiment analysis necessitates further research to understand the nuances of human sentiment expression.
* Responsible and impactful applications across industries and languages are encouraged.

**2.2 Pretrained Models for Sentiment Analysis**

Pre-trained Language Models (PLMs) have revolutionized downstream tasks, including sentiment analysis. These models come with extensive pre-trained knowledge from large text corpora, enabling them to capture complex linguistic nuances, contextual comprehension, and semantic associations. By fine-tuning PLMs, we can apply them directly to sentiment analysis tasks without training from scratch.

**Popular Pre-trained Sentiment Analysis Models**:

**BERT (Bidirectional Encoder Representations from Transformers)**:

BERT, based on transformer architecture, has achieved state-of-the-art performance in various NLP tasks, including sentiment analysis.Its bidirectional context understanding allows it to capture rich contextual information.

**Google Natural Language**:

Google’s Natural Language API provides pre-trained models for sentiment analysis.It leverages deep learning techniques to analyze sentiment in text data.

**IBM Watson Natural Language Understanding**:

IBM’s NLU service offers pre-trained models for sentiment analysis. It combines rule-based and machine learning approaches to extract sentiment information.

**TextBlob Simplified Text Processing**:

TextBlob, a Python library, provides simple sentiment analysis tools.While not as sophisticated as deep learning models, it’s easy to use and suitable for basic tasks.

**VADER Sentiment Analysis**:

VADER (Valence Aware Dictionary and sEntiment Reasoner) is a lexicon-based approach. It assigns sentiment scores to words and aggregates them to determine overall sentiment.

**Chapter 3: Data Collection**

Data collection is fundamental for informed decision-making, research, and quality assurance. Researchers rely on data to analyze trends and patterns, while predictive modeling benefits from accurate data. Additionally, organizations use collected data to evaluate performance and identify areas for improvement.

**3.1 Data sources**

Data can be collected from various sources , some include survey, customer reviews, reviews from technical experts , social media , news articles, online shopping sites, and many more.

In this project the customer reviews has been scraped from Amazon.com which provides a rich dataset. Amazon’s global user base ensures a wide range of perspectives, making it a valuable resource. By analyzing these reviews, you can gain insights into Intel products’ real-world performance and customer satisfaction. The unbiased nature of customer feedback allows you to assess products without external influence. Trustworthy platforms like Amazon contribute significantly to accurate sentiment analysis.

**3.2 Data Acquisition**

In this project, Playwright library of python is utilized to extract data from Amazon. Specifically, leveraging the “async\_playwright” module, which integrates with “asyncio” for asynchronous web scraping. Playwright is a powerful tool for automating browser interactions, allowing to navigate web pages, interact with elements, and extract relevant information. By combining it with “asyncio”, one can efficiently handle concurrent tasks, making it ideal for scraping large amounts of data from dynamic websites like Amazon. This collected data is stored in JSON files

**3.3 Data Description**

The features of the collected data are following:

* Product name -has the complete name of the product reviews extracted from amazon. Example- “ Intel Core I9-14900K LGA 1700 New Gaming Desktop Processor 24 Cores (8 P-Cores + 16 E-Cores) with Integrated Graphics – Unlocked “
* Rate – The rating of the product given by users . This rating is out of 5. Example – “ 5.0 out of 5 stars ”
* Review Date – The date and location of the review. Example- “ Reviewed in India on 20 November 2023”
* Review Text -The actual review by the customer. Example – “ Fast shipping. Works as advertised. ”
* Patter Name- Name of the pattern of the processor. Only present in those products where categories are present under the same product name. Example – “Pattern Name: ProcessorStyle Name: Core™ i9-14900k ”
* Product MRP – The maximum retail price of the product . “ ₹82,350”

A group of text boxes

Description automatically generated

Figure 1 – Features and a peek into the dataset

**Description of the dataset**

A close up of a person

Description automatically generated

Figure 2 – Characteristics and description of the data

The dimension of this dataset is 1650 \* 6. All the features are of object type.

**Chapter 4 : Data Pre-processing**

Data pre-processing is a critical step in preparing raw data for analysis. It involves cleaning, transforming, and organizing the data to improve its quality and suitability for further processing. Common tasks include handling missing values, removing duplicates, standardizing formats, and scaling features. Proper data pre-processing ensures accurate and reliable results in subsequent analyses.

**4.1 Cleaning of data**

**Missing values**

The data doesn’t seem to have any missing values. But the missing values may be disguised as empty string. This need to be checked.

It also does not have any duplicate rows.

A screenshot of a computer

Description automatically generated

Figure 3 – Checking for duplicate rows

A screenshot of a computer

Description automatically generated

Figure 4 – Duplicate Reviews

The above command is used to check if same review is recorded twice while collecting it. It doesn’t seem like this has happened. Though there are some empty spaces and some common review text like “Very Good” which are seen as duplicate. So this doesn’t account for any actual duplicated review data.

The reviews with only rating and no text review has to be removed since doesn’t meet our criteria for sentiment analysis.

A screenshot of a computer code

Description automatically generated

Figure 5 - Dropping empty reviews

The rating of products is object type, it can be converted to integer type by having a value from 1 to 5 depending on the rating.

A screen shot of a computer

Description automatically generated

Figure 6 – Cleaning the rating column

Next, the prices of each product is checked to ensure that its right, some prices are wrong , so it is corrected. Later the MRP is changed to integer type.

A screenshot of a computer

Description automatically generated

Figure 7 - MRP of products

The review data column is made into date type and a separate column consisting of the country names from where the review was written is made. Following is how the data looks after all the cleaning.

A screenshot of a computer

Description automatically generated

Figure 8 – Cleaned data

**4.2 Text pre-processing**

To apply various models for training, text preprocessing one crucial step. It involves cleaning the text, tokenization, removing stop words, lemmatization and various other techniques are applied to do the same.

In this project, text preprocessing is done so that meaningful data can be extracted.

Firstly , the review text is cleaned by removing parts that are unnecessary.

A screenshot of a computer

Description automatically generated

Figure 9- Removing ‘/n’ from review

Many reviews are present in other languages, this has to be translated. This is done using Google’s Translator API.

A screen shot of a computer code

Description automatically generated

Figure 10 – Translating the text to English if it is not in English

Stop word removal is a preprocessing step in natural language processing (NLP) where common words (such as “the,” “and,” “is”) are filtered out from text data. These words don’t carry significant meaning and can be safely removed to improve the efficiency and accuracy of NLP tasks like sentiment analysis.

A screenshot of a computer

Description automatically generated

Figure 11- Stop words are removed from review text

The text has some emojis in it. Since it is not a lot, it can be removed .

A screenshot of a computer code

Description automatically generated

Figure 12 – Handling Emojis

The words that have numbers are not significant, hence they are removed.

A computer screen shot of a computer code

Description automatically generated

Figure 13 – Removing words containing Numbers

Next crucial step is to tokenize and then lemmatize the text.

A computer screen shot of a computer code

Description automatically generated

Figure 14 – Tokenizing and Lemmatizing

Thus, the data is prepared and can be used to perform sentiment classification

**Chapter 5: Exploratory Data Analysis**

Exploratory Data Analysis (EDA) plays a pivotal role in data science. It involves visualizing and summarizing data to uncover patterns, outliers, and relationships, helping analysts gain insights before diving into more complex modelling.

**Number of reviews for each product**

**A graph of blue bars with numbers and letters

Description automatically generated with medium confidence**

Figure 15 – Number of reviews for each product

As we can see all the products don’t have equal number of reviews. Data for some products is very less.

**MRP of different products**

**A graph of blue bars

Description automatically generated**

Figure 16 – MRP vs Product

**Average rating vs Price**

**A graph with blue lines and dots

Description automatically generated**

Figure 17 – MRP vs Average rating

We can see that product prices doesn’t seem to influence the ratings.

**Chapter 6 : Sentiment Analysis Methodology**

**6.1 Model and Approach**

For sentiment analysis in this project, VADER and RoBERTa models have been used.

The VADER (Valence Aware Dictionary and sentiment Reasoner) model is a rule-based approach for sentiment analysis. It relies on a pre-defined lexicon of words and their associated sentiment scores to determine the overall sentiment of a text.

On the other hand, the RoBERTa model, which is based on the Transformer architecture, falls under the category of deep learning approaches. It fine-tunes a large-scale language model on a specific sentiment analysis task, learning contextual representations from vast amounts of text data.

**VADER**

VADER relies on a pre-built sentiment lexicon, which contains words with associated sentiment scores.

Each word in the lexicon is assigned a polarity score (positive, negative, or neutral) and an intensity score.The lexicon includes common words, phrases, and emoticons, along with their sentiment values.By summing up the scores of individual words in a text, VADER determines the overall sentiment.

VADER incorporates specific rules (heuristics) to account for context and emphasize sentiment intensity. These rules go beyond what a typical bag-of-words model captures.

Some examples of these rules:

* Capitalization: Uppercase words may indicate strong sentiment.
* Punctuation: Exclamation marks increase intensity.
* Degree modifiers: Intensifiers (e.g., “very,” “extremely”) affect sentiment.
* Conjunctions: “But” can reverse sentiment.
* Negation: Words like “not” change polarity.

**RoBERTa**

RoBERTa (A Robustly Optimized BERT Pretraining Approach) is a powerful language model that builds upon the foundation of BERT (Bidirectional Encoder Representations from Transformers).

RoBERTa is based on the Transformer architecture, which excels at capturing contextual information from text.

Unlike BERT, RoBERTa pretrains on full sentences without the Next Sentence Prediction (NSP) loss. This change allows it to learn better sentence representations. It uses dynamic masking during pretraining, enhancing its ability to understand context.

RoBERTa employs large mini batches during training, which improves efficiency and generalization. The tokenization process involves a larger byte-level BPE (Byte-Pair Encoding), leading to richer subword representations.

**6.2 Model Selection**

K Means Clustering

K-means clustering is an iterative process that partitions a dataset into similar groups based on the distance between their centroids. It assigns each observation to the nearest centroid and updates the centroids until convergence. Widely used in customer segmentation, image compression, and anomaly detection, K-means minimizes the sum of squared distances within clusters.

This model is chosen, because it is an unsupervised learning approach and this can help to classify the data. The dataset is not labelled as positive, negative or neutral, hence unsupervised learning approach is the best way.

On applying K Means clustering it is found that the classification is done. But this classification doesn’t show the separation between positive, negative and neutral rather it just separates longer and shorter reviews. Hence, a requirement for pretrained model arises. This is when models like VADER and RoBERTa are useful.

Choosing RoBERTa and VADER

These models, will classify the data and help in gaining the required insights from the data.

Benefits of using VADER:

* Rule-Based Simplicity: Lexicon-based approach with context-aware rules.
* Social Media Focus: Ideal for short, social media-style content.
* Emoticon Handling: Considers emoticons and punctuation.
* Efficient Inference: Real-time analysis with low computational overhead.

Benefits of using RoBERTa:

* Deep Learning Powerhouse: Captures rich context using Transformers.
* Fine-Tuning Flexibility: Adaptable to specific tasks like sentiment analysis.
* Large-Scale Pretraining: Learns from vast text data for better generalization.
* Domain Generalization: Performs well across diverse domains and languages.

**Chapter 7: Implementation**

**7.1 Tools and Libraries**

**Python**

Python is a versatile, high-level programming language widely used for various applications.

Using Python is advantageous due to its extensive library support and simplicity, making it ideal for sentiment analysis and natural language processing with libraries like NLTK and spaCy, which facilitate efficient text analysis and manipulation. For web scraping, Python offers powerful tools like BeautifulSoup and Playwright, allowing easy extraction and automation of web data. Its clear syntax and active community enhance rapid development and problem-solving. Additionally, Python's adaptability across different domains and platforms makes it a go-to language for developers and data scientists alike.

**playwright.async\_api**

Playwright is a powerful library for browser automation and testing. It allows you to control browsers (such as Chromium, Firefox, and WebKit) programmatically.

**Asyncio**

Asyncio is a Python library used for concurrent programming, specifically asynchronous I/O. It allows you to write code that doesn’t block execution while waiting for I/O operations (such as reading from files, making network requests, or interacting with databases).

**Pandas**

The pandas library in Python is a powerful tool for data manipulation and analysis, providing data structures like DataFrame and Series for handling structured data efficiently. It simplifies data cleaning, transformation, and aggregation tasks, making it indispensable for data science and machine learning workflows.

**Numpy**

NumPy is a fundamental library for numerical computing in Python, providing support for arrays, matrices, and a vast collection of mathematical functions. Its efficient handling of large datasets and powerful operations on n-dimensional arrays make it essential for scientific computing and data analysis.

**Googletrans**

The googletrans library in Python provides a simple interface for translating text using Google Translate's API. It supports numerous languages and can detect the source language automatically, making it a convenient tool for multilingual text processing and analysis.

**matplotlib.pyplot**

The `matplotlib.pyplot` module in Python is a plotting library that provides a MATLAB-like interface for creating static, interactive, and animated visualizations. It's widely used for generating a variety of charts, including line plots, scatter plots, and histograms, making it essential for data visualization in scientific and analytical applications.

**Seaborn**

The `seaborn` library in Python is built on top of `matplotlib` and provides a high-level interface for creating attractive and informative statistical graphics. It simplifies complex visualizations such as heatmaps, violin plots, and pair plots, making it ideal for exploratory data analysis and presentation.

**nltk.sentiment**

The `nltk.sentiment` module in Python is part of the Natural Language Toolkit (NLTK) and provides tools for sentiment analysis. It includes pre-trained models like VADER (Valence Aware Dictionary and sEntiment Reasoner) for accurately assessing the sentiment polarity of text. This module simplifies the process of determining whether a piece of text expresses positive, negative, or neutral sentiments. It is widely used in applications like social media analysis, customer feedback evaluation, and opinion mining.

**AutoTokenizer**

The `transformers` library by Hugging Face includes the `AutoTokenizer` class, which provides an easy way to load pre-trained tokenizers for various transformer models. It automates the process of tokenizing text for models like BERT, GPT-3, and T5, facilitating seamless integration into natural language processing tasks.

**AutoModelForSequenceClassification**

The `AutoModelForSequenceClassification` class in the `transformers` library by Hugging Face allows for easy loading of pre-trained models specifically tailored for sequence classification tasks. This class simplifies implementing tasks such as sentiment analysis, text categorization, and binary or multi-class classification by leveraging powerful transformer models like BERT, RoBERTa, and DistilBERT.

**Softmax**

The `softmax` function from `scipy.special` is used to compute the softmax of an array, transforming it into a probability distribution. It is commonly applied in machine learning for multi-class classification tasks, ensuring that the output values are non-negative and sum to one.

**nltk.corpus.stopwords**

The `stopwords` module from `nltk.corpus` provides a list of common words in various languages that are typically filtered out in natural language processing tasks. By removing stopwords, such as "the," "is," and "in," it helps improve the efficiency and relevance of text analysis and model training.

**Regular expression**

The `re` module in Python provides support for regular expressions, allowing users to search, match, and manipulate strings based on specific patterns. It is widely used for tasks such as text parsing, validation, and complex string replacements in data processing and analysis.

**7.6 Sentiment classification**

**Using VADER model**

To use VADER, nltk.sentiments library provides a sentiment analyser. The sentiment intensity analyser object needs to be created first.

The model can then be applied on any sentence to test its polarity.

A screenshot of a phone

Description automatically generated

Figure 18 – Applying VADER sentiment analyser

Here, the negative score shows how negative the input sentences is , its value ranging from 0 to1. Similarly neutral and positive scores show the neutrality and positivity of the sentence respectively ranging from 0 to 1. 1 being more of the characteristic and 0 being does not show that characteristic.

Compound score the overall sentiment score for the data ranging from -1 to 1. If the score is between -1 and -0.1 , it shows negative sentiment, if it is above 0.1 , it depicts positive sentiment and every other score as neutral.

This is applied to the entire dataset.

A screenshot of a computer

Description automatically generated

Figure 19 – Applying VADER model to entire dataset

This returns a dataframe having negative, neutral, positive, and compound score for each review.

**Checking accuracy**

The accuracy of the model can be assessed by comparing the ratings given by each person with the model scores.

A graph of blue rectangles and black lines

Description automatically generated

Figure 20- Compound score vs rating

As we can see that , as the rating increases , the score does too except for 3 star rating.

Comparing with other scores

A graph of blue bars

Description automatically generated with medium confidence

Figure 21 – rating Vs other scores

These scores also makes sense. The positive scores increase with increase in rating , the negative scores decrease. While the neutral scores is high for rating 3.

Based on the compound score ,the sentiment is decided for the review

* Positive- compound score > 0.1
* Negative – compound score< -0.1
* Neutral – -0.1 < compound score <0.1

A screenshot of a computer

Description automatically generated

Figure 22 – Sentiments classified

**Using RoBERTa model**

The same process can be repeated with RoBERTa model. This will help in comparing both the models.

The text is initially tokenized using AutoTokenizer library. On this tokenized text , the model is applied. This gives the scores as output. On the scores , softmax is applied. Finally the scores contain positive , negative and neutral.

A screenshot of a computer

Description automatically generated

Figure 23 – Apply RoBERTa model on a Sentence

Next, the model is applied on the whole dataset to find the sentiment of the review.

A screenshot of a computer screen

Description automatically generated

Figure 24 – RoBERTa applied for the whole dataset

The scores are compared with the rating values to see their accuracy. Following results are obtained.

A graph of blue bars

Description automatically generated with medium confidence

Figure 25 – RoBERTa scores vs rating

The sentiment is estimated based on these scores. Whichever score out of positive, negative and neutral is highest, the review is classified to that class.

A screenshot of a computer

Description automatically generated

Figure 26 – RoBERTa classification

RoBERTa and VADER classifications are compared and it seems like both have done almost same classification.

**Chapter 8: Conclusion**

There are a few analysis that can be drawn from the data after classification.

**Comparing the reviews of people from different countries**

Looking at the reviews of people from different countries, we can see the acceptance and how they find INTEL products.

Here is the comparison:

A graph of blue and black lines

Description automatically generated with medium confidence

Figure 27 – Comparing the reviews of people from different countries

We can see that people from Poland have given the most positive reviews and Netherlands has the most negative reviews.

This does not imply much because of limitation of the available data. This process can be used to withdraw meaningful insights when the data is of large scale.

**Checking the reviews for each product**

**A graph with blue lines

Description automatically generated**

Figure 28 – Product vs their review sentiment

Most of the products have similar trend in positive reviews. The negative reviews are highest for the model 11900K.

**Sentiment of the product and price of the product**

**A graph with blue lines and numbers

Description automatically generated**

Figure 29 – Average sentiment vs price

As we can see, there is no trend seen here. So we can conclude that the price doesn’t influence the sentiment of a review. The average sentiment of the product priced 60,000 is more than the others.

**References**

[Natural language processing - Wikipedia](https://en.wikipedia.org/wiki/Natural_language_processing)

[A review of sentiment analysis: tasks, applications, and deep learning techniques | International Journal of Data Science and Analytics (springer.com)](https://link.springer.com/article/10.1007/s41060-024-00594-x)

[Getting Started with Sentiment Analysis using Python (huggingface.co)](https://huggingface.co/blog/sentiment-analysis-python)

[Sentiment Analysis: Exploring Pre-trained and Domain-dependent Models | by Alex Ianovski | Analytics Vidhya | Medium](https://medium.com/analytics-vidhya/sentiment-analysis-exploring-pre-trained-and-domain-dependent-models-9604b4f48561)