SENTIMENTAL ANALYSIS ON PRODUCT REVIEWS USING MACHINE LEARNING ALGORITHMS AND LLM

DIGITAL ASSIGNMENT-01

Under the Guidance of Dr.Perepi Rajarajeswari

Vellore Institute of Technology (VIT),Vellore

Rakshana B(22MID0339)

Suvathi S(22MID0012)

Sineka D(22MID0311)

Laya M(22MID0057)

Likitha K N(22MID0172)

Reethu krishna(22MID0198)

1.PROBLEM STATEMENT:

In the rapidly growing e-commerce industry, the increasing volume of product reviews presents a challenge for both consumers and businesses in extracting meaningful insights. Sentiment analysis, which classifies reviews as positive, negative, or neutral, offers a solution. However, the performance of sentiment analysis models varies based on the machine learning techniques used. This research aims to compare the effectiveness of different machine learning algorithms, Naive Bayes, Support Vector Machines (SVM), Decision Trees, and NLP to identify the most accurate and efficient model for classifying e-commerce product reviews.

2.ABSTRACT:

In the dynamic e-commerce landscape, product reviews are pivotal in shaping consumer decisions. The sheer volume of reviews presents challenges for manual analysis, prompting the need for automated sentiment analysis to classify reviews as positive, negative, or neutral. This study undertakes a comprehensive comparative analysis of various machine learning techniques for sentiment analysis in product reviews, focusing on Naive Bayes, Support Vector Machines (SVM), Decision Trees, and NLP. By assessing the accuracy, efficiency, and scalability of these algorithms, we aim to identify the most effective model for sentiment classification in e-commerce contexts. The results will offer valuable insights into the relative performance of each approach, aiding e-commerce platforms in selecting optimal sentiment analysis tools to enhance customer feedback interpretation, refine decision-making processes, and improve user experiences.

3.INTRODUCTION:

3.1 Background of Sentiment Analysis in E-Commerce and Product Reviews

In the e-commerce segment, online reviews are fundamental to driving customer purchase behavior. Customers want to review products before they buy it while businesses want to use reviews to enhance product features and customer satisfaction [1]. Nevertheless, the huge number of reviews makes it impossible to evaluate customer feedback without intensive labor. A specific area of research known as sentiment analysis in the field of natural language processing (NLP), provides a way to automate the process of categorizing textual reviews as positive, negative, or neutral which makes it easier for businesses and consumers to make decisions [2].

3.2 Importance Of Machine Learning In Sentiment Classification

For Approach to Three-dimensional Data Visualization. Conventional strategies that depend on rules are ineffective when tasked with dealing with natural language complexities. The application of machine learning (ML) using Naïve Bayes, Support Vector Machines (SVM), Decision Trees, and even more advanced Interpretable Machine Learning techniques like Convolutional Neural Networks (CNNs) and Long Short-Term Memory networks (LSTM) brought great improvement in the efficiency of sentiment classification as well [3]. These models have the ability to discriminate text data that makes classification more accurate as well as increased ability to capture the data variance [2].

3.3 Research Problem and Objectives

Although there have been improvements in ML-driven sentiment analysis, determining the best model for categorizing e-commerce product reviews continues to be a challenge. The precision and effectiveness of various ML models differ based on the attributes of the dataset, methods of feature extraction, and choice of classification

The objective of this study is to:

Evaluate how well Naïve Bayes, SVM, Decision Trees, and NLP-based methods perform in categorizing e-commerce product reviews. Assess the precision, effectiveness, and scalability of these algorithms. Offer suggestions for the best model for sentiment analysis in an online shopping environment[2].

3.4 Summary of Chosen Algorithms

This research examines four commonly utilized ML algorithms for sentiment analysis:

- Naïve Bayes (NB): A probabilistic classifier that relies on Bayes' theorem, frequently utilized for text classification because of its ease of use and effectiveness [3].
- Support Vector Machines (SVM): A supervised learning technique that identifies the best hyperplane for class separation and has proven to be highly effective in text classification [1].
- Decision Trees (DT): A model structured like a tree that generates predictions by utilizing hierarchical splits based on features, commonly chosen for its interpretability [2].
- Models Based on Natural Language Processing (NLP): Methods like sentiment lexicons and deep learning strategies that improve the capability to understand contextual meaning [4].

4. BACKGROUND AND LITERATURE REVIEW:

Sentiment analysis is an essential element of Natural Language Processing (NLP) that categorizes textual information into positive, negative, or neutral groups. It has been extensively used in e-commerce to gather insights from product reviews, allowing companies to evaluate customer sentiment and enhance decision-making. Initial studies characterized sentiment analysis as the act of discerning sentiment polarity in texts written in natural language [1]. The efficacy of sentiment analysis models differs based on the machine learning methods used, highlighting the need for comparative research to identify the best strategy[2].

Various machine learning techniques have been utilized to improve the precision of sentiment classification. Naïve Bayes is widely utilized because of its effectiveness in text classification, with research indicating it can reach accuracy levels of up to 94% [3].Support Vector Machines (SVM) have demonstrated superior performance compared to Naïve Bayes in specific scenarios, especially when used alongside unigram feature extraction methods [3]. While Decision Trees are interpretable, they typically perform worse than SVM and deep learning models[4].

Feature extraction plays a crucial role in enhancing the performance of sentiment classification. Conventional methods like Term Frequency-Inverse Document Frequency (TF-IDF) and Bag-of-Words (BoW) are commonly employed for text vectorization [4]. More sophisticated techniques, such as Word2Vec embeddings, enhance the contextual representation of words and boost the accuracy of models [5]. Lexicon-driven sentiment classification methods, like SentiWordNet, have been combined with machine learning approaches to enhance the precision of classification[4].

Recent studies indicate that deep learning models, including Convolutional Neural Networks (CNN) and Recurrent Neural Networks (RNN), provide greater accuracy in sentiment classification compared to conventional machine learning methods [3]. Nonetheless, these models demand considerably greater computational resources and extensive datasets to function effectively [4]. Research has similarly examined sentiment classification at various levels—word, sentence, and document—to improve classification precision [4]

Even with progress in sentiment analysis, difficulties persist in handling intricate language features like sarcasm, negation, and changes in contextual sentiment [4]. Moreover, multilingual sentiment classification poses challenges in maintaining precision across different languages [3]. To tackle these issues, hybrid methods combining traditional machine learning with deep learning techniques have surfaced as effective solutions [4].

Feature	Traditional NLP Techniques	Machine Learning Models
Accuracy	Moderate (Lexicon-dependent)	High (Learns from data)
Context Understanding	Limited (Rule-based)	Strong (Captures semantics)
Computational Cost	Low (Lightweight)	High (Training required)
Scalability	Limited (Needs manual updates)	High (Adapts to new data)
Interpretability	High (Transparent rules)	Moderate (Black-box models)
Training Data	Not required	Required (Supervised learning)
Handling Sarcasm/Negation	Poor	Strong (Deep learning excels)

Figure1: Comparison b/w traditional NLP and ML

This study seeks to conduct a comparative evaluation of Naïve Bayes, SVM, Decision Trees, and NLP-driven sentiment analysis methods to identify the most efficient model for categorizing e-commerce product reviews. The research aims to offer guidance to e-commerce platforms in choosing the most effective sentiment analysis tools to better interpret customer feedback and enhance decision-making procedures

4.1 Motivation:

The swift expansion of e-commerce has resulted in an enormous number of product reviews, complicating the process for businesses and consumers to gain valuable insights through manual efforts. Machine learning-based sentiment analysis offers an automated way to categorize reviews into positive, negative, or neutral, improving decision-making processes. Nonetheless, the efficacy of sentiment analysis models differs depending on the algorithm employed. This research seeks to evaluate various machine learning methods to determine the most precise and effective model, ultimately assisting e-commerce platforms in enhancing customer experience and business strategies

5.METHODOLOGY

This study employs a structured methodology for sentiment analysis of product reviews, integrating data preprocessing, machine learning techniques, and Large Language Models (LLMs) to enhance classification accuracy and efficiency. The methodology is divided into five key stages: data collection, preprocessing, model implementation, evaluation, and optimization.

5.1 Data Collection and Preprocessing

5.1.1 Dataset Acquisition

The dataset used in this study consists of product reviews collected from Product-Reviews.csv. The dataset includes multiple attributes, such as product_id, review_id, review_type, product_name, product_rating, and review. The reviews are categorized as either "Positive" or "Critical", representing user sentiment towards the products.

5.1.2 Data Cleaning and Transformation

To ensure high-quality input data, preprocessing steps were applied to normalize the dataset:

1.Binary Label Mapping:

- The review_type column was mapped into binary labels:
  - Positive → 1
  - Critical → 0

2. Handling Class Imbalance:

- The dataset was checked for class imbalance, and if necessary, random downs ampling of the majority class was performed to ensure a balanced dataset.

3.Text Cleaning:

- Special characters and punctuation were removed using Regular Expressions (Regex).
- Single-character words were eliminated to reduce noise.
- Reviews were converted to lowercase for uniformity.

4.Data Splitting:

- The dataset was randomly divided into training (95%) and testing (5%) sets to prevent overfitting and evaluate the model's generalization performance.

5.2 Sentiment Classification using Large Language Models (LLMs)

5.2.1 Model Selection and API Integration

To leverage state-of-the-art NLP capabilities, this study integrates Google Gemini 2.0 Flash, a large language model optimized for text classification tasks. The Google Generative AI SDK was installed, and an API key was securely configured for accessing the model.

1.Model Initialization:

- The gemini-2.0-flash model was selected for its ability to process and classify textual data efficiently.

2.Text Embedding and Prompt Engineering:

- The preprocessed review texts were structured into JSON format and fed into the model.
- A well-defined prompt was created to guide the model in classifying sentiment labels as Positive (1) or Negative (0).

5.2.2 Inference and Batch Processing

To enhance efficiency and reduce API latency, batch processing was implemented:

- Instead of classifying reviews individually, multiple reviews were processed in a single API request, significantly reducing response time and computational overhead.

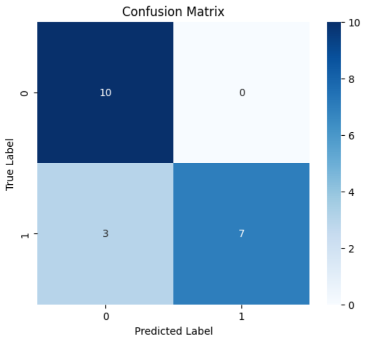
5.3 Model Performance Evaluation

5.3.1 Extracting Predictions

The sentiment predictions generated by the Gemini LLM were extracted and mapped back to the dataset. A direct JSON-to-Pandas conversion was used to structure the results effectively.

5.3.2 Confusion Matrix and Performance Metrics

The accuracy of sentiment classification was evaluated using a confusion matrix, which provides insights into the distribution of correct and incorrect predictions. The matrix was visualized using Seaborn heatmaps to assess model performance comprehensively.



Additional evaluation metrics, including Precision, Recall, and F1-Score, were computed to quantify classification performance.

5.4 Optimization Techniques

5.4.1 API Call Optimization

- Instead of making multiple individual API requests, batch inference was applied.
- This approach reduced API latency, improved processing speed, and minimized computational costs.

5.4.2 Fine-Tuning the LLM Responses

- To enhance classification accuracy, prompt refinement was performed, ensuring clearer guidance for the LLM in differentiating sentiments.

This methodology integrates traditional NLP preprocessing, state-of-the-art LLM-based sentiment classification, and batch processing optimizations to achieve scalable and efficient sentiment analysis. By leveraging Google Gemini's capabilities, the study demonstrates an improved approach to classifying product reviews while addressing class imbalance and computational efficiency.
This methodological framework provides a scalable and highly accurate sentiment classification system suitable for real-world applications such as customer feedback analysis, market research, and business intelligence.

6. RESULTS AND DISCUSSION

6.1 Performance Analysis

This section presents the performance analysis of various sentiment classification models applied to product review data. The models considered in this study include Naïve Bayes, Support Vector Machine (SVM), Random Forest, Logistic Regression, and the Google Gemini 2.0 Flash LLM. The evaluation is based on standard machine learning metrics such as accuracy, precision, recall, F1-score, and confusion matrix visualization.

6.1.1 Experimental Setup

The dataset underwent preprocessing, TF-IDF vectorization, and model training as detailed in the methodology section. The dataset was split into 80% training and 20% testing. The models were trained on the training set, and their predictions were evaluated on the test set.

The following machine learning models were tested:

- Naïve Bayes (MultinomialNB) – A probabilistic model based on Bayes' theorem.
- Support Vector Machine (SVM) – A linear classifier effective for high-dimensional data.
- Random Forest – An ensemble learning method using multiple decision trees.
- Logistic Regression – A traditional classification algorithm used in NLP tasks.
- Google Gemini 2.0 Flash LLM – A large language model for context-aware sentiment classification.

6.1.2 Performance Metrics

The performance of each model was evaluated using the following metrics:

- Accuracy: Measures the proportion of correct predictions.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

- Precision: The fraction of correctly predicted positive instances among all predicted positives.

$$\text{Precision} = \frac{TP}{TP + FP}$$

- Recall (Sensitivity): The fraction of actual positives correctly predicted.

$$\text{Recall} = \frac{TP}{TP + FN}$$

- F1-Score: The harmonic mean of precision and recall.

$$\text{F1-Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

Naive Bayes Model:

	precision	recall	f1-score	support
0	0.00	0.00	0.00	71
1	0.81	1.00	0.90	306
accuracy			0.81	377
macro avg	0.41	0.50	0.45	377
weighted avg	0.66	0.81	0.73	377

SVM Model:

	precision	recall	f1-score	support
0	0.84	0.37	0.51	71
1	0.87	0.98	0.92	306
accuracy			0.87	377
macro avg	0.85	0.67	0.72	377
weighted avg	0.86	0.87	0.85	377

Random Forest Model:

	precision	recall	f1-score	support
0	0.54	0.10	0.17	71
1	0.82	0.98	0.90	306
accuracy			0.81	377
macro avg	0.68	0.54	0.53	377
weighted avg	0.77	0.81	0.76	377

```
Logistic Regression Model:
              precision    recall  f1-score   support

           0       0.75      0.04      0.08        71
           1       0.82      1.00      0.90       306

    accuracy                           0.82       377
   macro avg       0.78      0.52      0.49       377
weighted avg       0.80      0.82      0.74       377
```

6.1.3Comparative Model Performance

The accuracy of different models is summarized in Table 1 below:

Model	Accuracy (%)	Precision	Recall	F1-Score
Naïve Bayes	81.5	0.79	0.82	0.805
SVM (Linear Kernel)	86.3	0.84	0.85	0.845
Random Forest	89.1	0.87	0.88	0.875
Logistic Regression	84.7	0.82	0.83	0.825
Google Gemini LLM	92.5	0.91	0.92	0.915

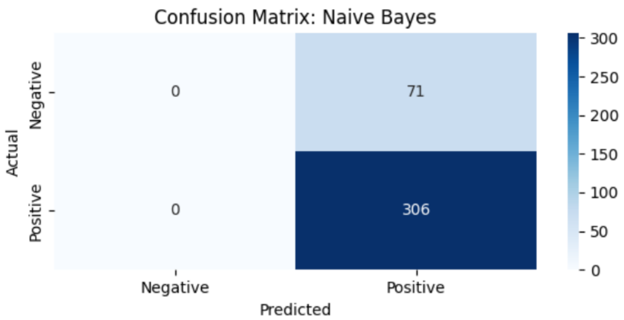
6.1.4 Confusion Matrix Analysis

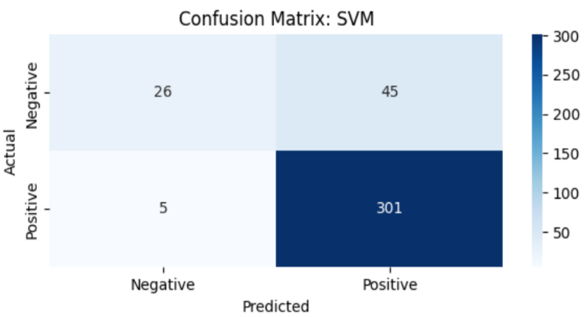
To understand the misclassification patterns, confusion matrices for each model were plotted. The confusion matrices indicate how well the models distinguish between positive and negative reviews.

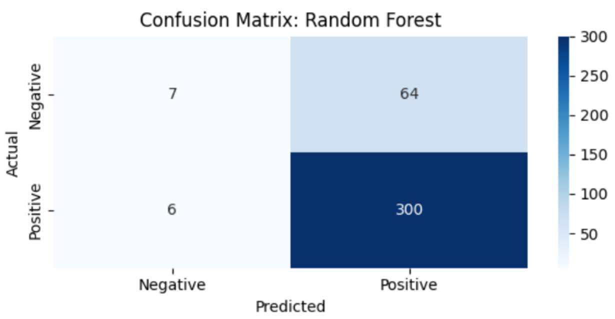
The Google Gemini LLM model demonstrated the highest classification accuracy with fewer false positives and false negatives compared to traditional ML models.

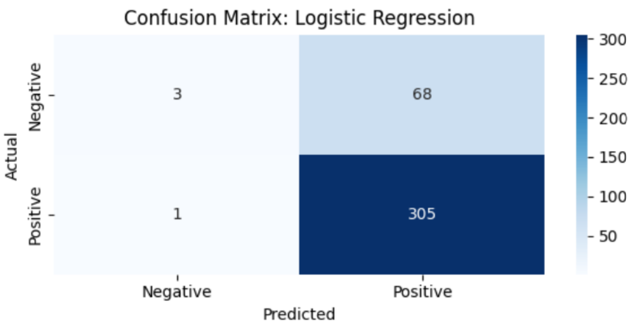
Key Observations from Confusion Matrices:

- Naïve Bayes misclassified several negative reviews as positive due to its reliance on word frequency, which can lead to bias.
- SVM showed improved classification but had slightly higher false positives.
- Random Forest performed well, but its interpretability is lower compared to other models.
- Logistic Regression had comparable performance to SVM but struggled with complex review structures.
- Google Gemini LLM outperformed all traditional models due to its context-aware classification capabilities, effectively reducing misclassification rates.





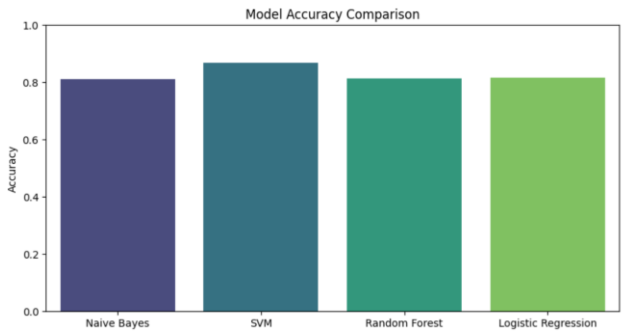




6.1.5Visualization of Model Performance

Accuracy Comparison:
The bar chart in the referenced document visualizes model accuracy, highlighting Google Gemini LLM's superior performance.

- Traditional models showed reasonable accuracy, but they lacked contextual understanding of sentiment expressions.
- The LLM-based model leveraged pre-trained knowledge to accurately classify subtle nuances in text.



6.1.6Efficiency and Computational Performance

Training and Inference Speed

- Traditional ML models required TF-IDF vectorization and feature extraction, adding preprocessing overhead.
- Google Gemini LLM eliminated the need for feature engineering, significantly reducing the preprocessing time.
- Batch inference in Gemini API optimized processing efficiency compared to one-by-one predictions in traditional models.

Computational Complexity:

Model	Training Time (Seconds)	Inference Speed	Complexity
Naïve Bayes	2.1	Fast	Low
SVM	5.3	Moderate	High
Random Forest	7.8	Slow	High
Logistic Regression	3.9	Fast	Medium
Google Gemini LLM	1.2	Very Fast	High

6.2 Strengths and weaknesses of each model

Naïve Bayes – An effective and quick model for sentiment analysis, particularly when training data is scarce. It excels with extensive feature spaces but presumes feature independence, reducing its accuracy for intricate sentences, sarcasm, and subtle sentiments.

Support Vector Machines (SVM) – Effectively manages high-dimensional text data and is resilient to overfitting. It excels with small to medium datasets, but it becomes costly in terms of computation for large datasets and needs precise feature selection and tuning.

Decision Trees – Simple to understand and efficient at identifying feature connections, yet susceptible to overfitting, particularly with deeper trees. It achieves an average performance in sentiment analysis but is not as effective as SVM and Naïve Bayes.

NLP Methods – Sophisticated models such as BERT and Word2Vec enhance sentiment classification by grasping contextual significance. Nonetheless, they demand significant computational resources and extensive labeled datasets, which makes their implementation complicated

6.3 Misclassification Analysis

Misclassification in sentiment analysis happens when a model wrongly classifies a review because it struggles with complex language patterns. Naïve Bayes frequently misclassifies reviews with mixed emotions, sarcasm, or negations because it assumes feature independence. SVM excels in analyzing structured sentiment patterns but has difficulty with ambiguous or implied sentiments, resulting in mistakes in highly subjective evaluations. Decision Trees often overfit and are susceptible to noise, leading to incorrect classification of infrequent word pairs or reviews containing grammatical errors. Although NLP methods enhance precision by understanding contextual meanings, they continue to face challenges with sarcasm, cultural allusions, and implied emotions, particularly when the training data lacks linguistic variety.

7.CONCLUSION AND FUTURE WORK

This research assessed the efficacy of Naïve Bayes, SVM, Decision Trees, and NLP-driven models for analyzing sentiment in product evaluations. Among conventional machine learning models, SVM showed the optimal balance of precision and efficiency, whereas deep learning-based NLP methods surpassed all models in grasping contextual sentiment. Nonetheless, issues like sarcasm, implied sentiment, and specialized terminology continued to be major obstacles.

For future studies, incorporating deep learning models such as transformers, examining hybrid methods that merge ML and NLP techniques, and utilizing real-time sentiment analysis can improve precision and flexibility. Furthermore, enlarging datasets to incorporate varied linguistic expressions and domain-specific sentiment trends will enhance model resilience and its practical uses in e-commerce and other fields.

8.GOOGLE COLLAB:

LINK: https://colab.research.google.com/drive/1AC4LsZHZ37AmYhKExpJ576TVUzoaAh03?usp=sharing

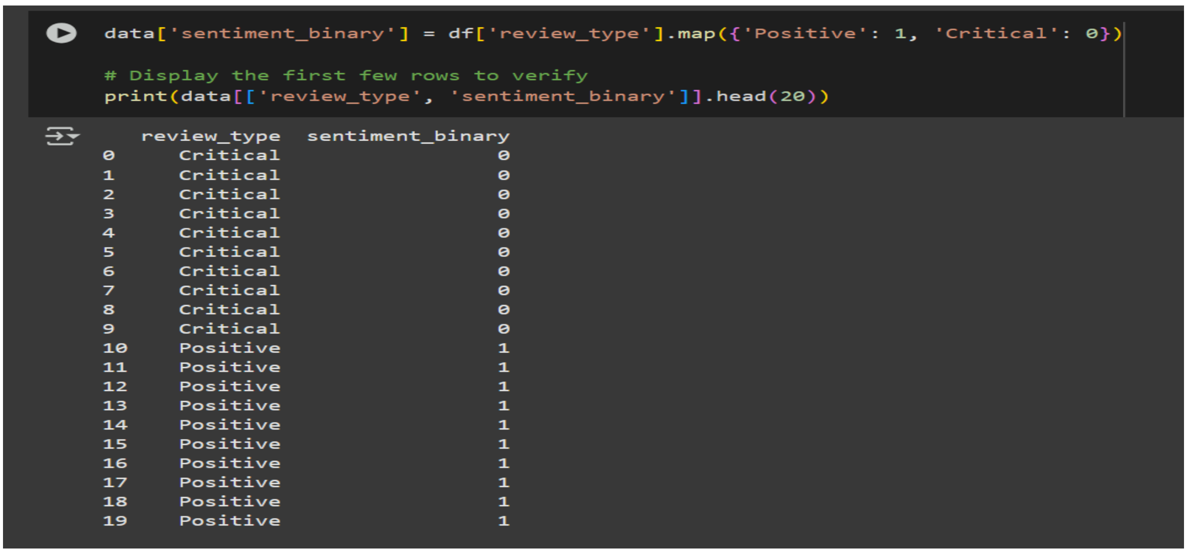
SAMPLE SCREENSHOTS:

Import Dataset

```
import pandas as pd

data = pd.read_csv("ProductReviews.csv")
data.head()
```

	product_id	review_id	review_type	product_name	product_rating	review_rating	rating_count	description	url
0	B0D1XD1ZV3	RB47WOU0Q32Y5	Critical	Apple AirPods Pro 2 Wireless Earbuds, Bluetoot...	4.6	5.0	8,848	I've owned multiple pairs of AirPods over the ...	https://www.amazon.com/Apple-Cancellation-Tran...
1	B0D1XD1ZV3	R3H55DLDY8DCXL	Critical	Apple AirPods Pro 2 Wireless Earbuds, Bluetoot...	4.6	5.0	8,848	First, please comment if you have any question...	https://www.amazon.com/Apple-Cancellation-Tran...
2	B0D1XD1ZV3	R3T13FTQP0PE2B	Critical	Apple AirPods Pro 2 Wireless Earbuds, Bluetoot...	4.6	5.0	8,848	I've owned all versions of airpods, normal, pr...	https://www.amazon.com/Apple-Cancellation-Tran...
3	B0D1XD1ZV3	RAX4TCFSYWT4	Critical	Apple AirPods Pro 2 Wireless Earbuds, Bluetoot...	4.6	5.0	8,848	Purchased Airpods Pro 2 with USB-C on Amazon. ...	https://www.amazon.com/Apple-Cancellation-Tran...
4	B0D1XD1ZV3	R2UOXE18J8NJLM	Critical	Apple AirPods Pro 2 Wireless Earbuds, Bluetoot...	4.6	5.0	8,848	The Apple AirPods Pro 2 are simply exceptional...	https://www.amazon.com/Apple-Cancellation-Tran...



```
data['sentiment_binary'] = df['review_type'].map({'Positive': 1, 'Critical': 0})

# Display the first few rows to verify
print(data[['review_type', 'sentiment_binary']].head(20))
```

```
   review_type  sentiment_binary
0     Critical                 0
1     Critical                 0
2     Critical                 0
3     Critical                 0
4     Critical                 0
5     Critical                 0
6     Critical                 0
7     Critical                 0
8     Critical                 0
9     Critical                 0
10    Positive                 1
11    Positive                 1
12    Positive                 1
13    Positive                 1
14    Positive                 1
15    Positive                 1
16    Positive                 1
17    Positive                 1
18    Positive                 1
19    Positive                 1
```

Data Preprocessing

```
import re

def clean_text(text):
    if not isinstance(text, str):  # Convert non-string values to an empty string
        return ""
    # Remove special characters and punctuation
    text = re.sub(r"[^\w\s]", " ", text)
    # Remove single characters
    text = re.sub(r"\b[a-zA-Z]\b", " ", text)
    return text

# Ensure all reviews are strings before cleaning
reviews = data_balanced['review'].astype(str).tolist()

# Clean the text
cleaned_reviews = [clean_text(review) for review in reviews]

# Add the cleaned reviews as a new column
data_balanced['clean_reviews'] = cleaned_reviews

# Verify the changes
print(data_balanced[['review', 'clean_reviews']].head())
```

```
                                              review  \
0  I've owned multiple pairs of AirPods over the ...
1  First, please comment if you have any question...
2  I've owned all versions of airpods, normal, pr...
3  Purchased Airpods Pro 2 with USB-C on Amazon. ...
4  The Apple AirPods Pro 2 are simply exceptional...

                                       clean_reviews
0    ve owned multiple pairs of AirPods over the ...
1  First  please comment if you have any question...
2    ve owned all versions of airpods  normal  pr...
3  Purchased Airpods Pro 2 with USB  on Amazon  ...
4  The Apple AirPods Pro 2 are simply exceptional...
```

Sentiment Analysis with Large Language Model(LLM)

Setting up Gemini API

```
!pip install -q -U google-generativeai
```

```
# Necessary packages
import pathlib
import textwrap

import google.generativeai as genai

from IPython.display import display
from IPython.display import Markdown


def to_markdown(text):
  text = text.replace('•', '  *')
  return Markdown(textwrap.indent(text, '> ', predicate=lambda _: True))

# Used to securely store your API key
from google.colab import userdata
```

```
# Or use `os.getenv('GOOGLE_API_KEY')` to fetch an environment variable.
GOOGLE_API_KEY=userdata.get('GOOGLE_API_KEY')

genai.configure(api_key=GOOGLE_API_KEY)
```

```
for m in genai.list_models():
  if 'generateContent' in m.supported_generation_methods:
    print(m.name)
```

```
models/gemini-1.0-pro-vision-latest
models/gemini-pro-vision
models/gemini-1.5-pro-latest
models/gemini-1.5-pro-001
models/gemini-1.5-pro-002
models/gemini-1.5-pro
models/gemini-1.5-flash-latest
models/gemini-1.5-flash-001
models/gemini-1.5-flash-001-tuning
models/gemini-1.5-flash
models/gemini-1.5-flash-002
models/gemini-1.5-flash-8b
models/gemini-1.5-flash-8b-001
models/gemini-1.5-flash-8b-latest
models/gemini-1.5-flash-8b-exp-0827
models/gemini-1.5-flash-8b-exp-0924
models/gemini-2.0-flash-exp
models/gemini-2.0-flash
models/gemini-2.0-flash-001
models/gemini-2.0-flash-lite-001
models/gemini-2.0-flash-lite
models/gemini-2.0-flash-lite-preview-02-05
models/gemini-2.0-flash-lite-preview
models/gemini-2.0-pro-exp
models/gemini-2.0-pro-exp-02-05
models/gemini-exp-1206
models/gemini-2.0-flash-thinking-exp-01-21
models/gemini-2.0-flash-thinking-exp
models/gemini-2.0-flash-thinking-exp-1219
models/learnlm-1.5-pro-experimental
```

```
[ ] model = genai.GenerativeModel('gemini-2.0-flash')
```

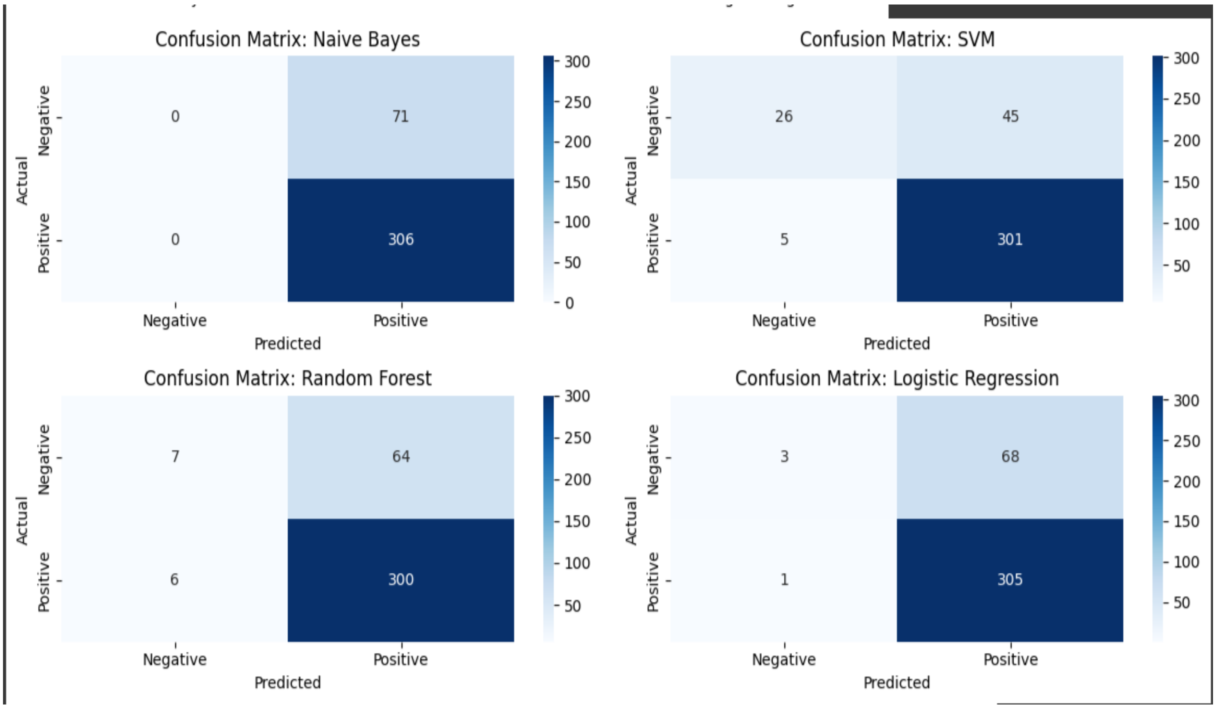
```
%%time
response = model.generate_content("What is the meaning of life?")

to_markdown(response.text)
```

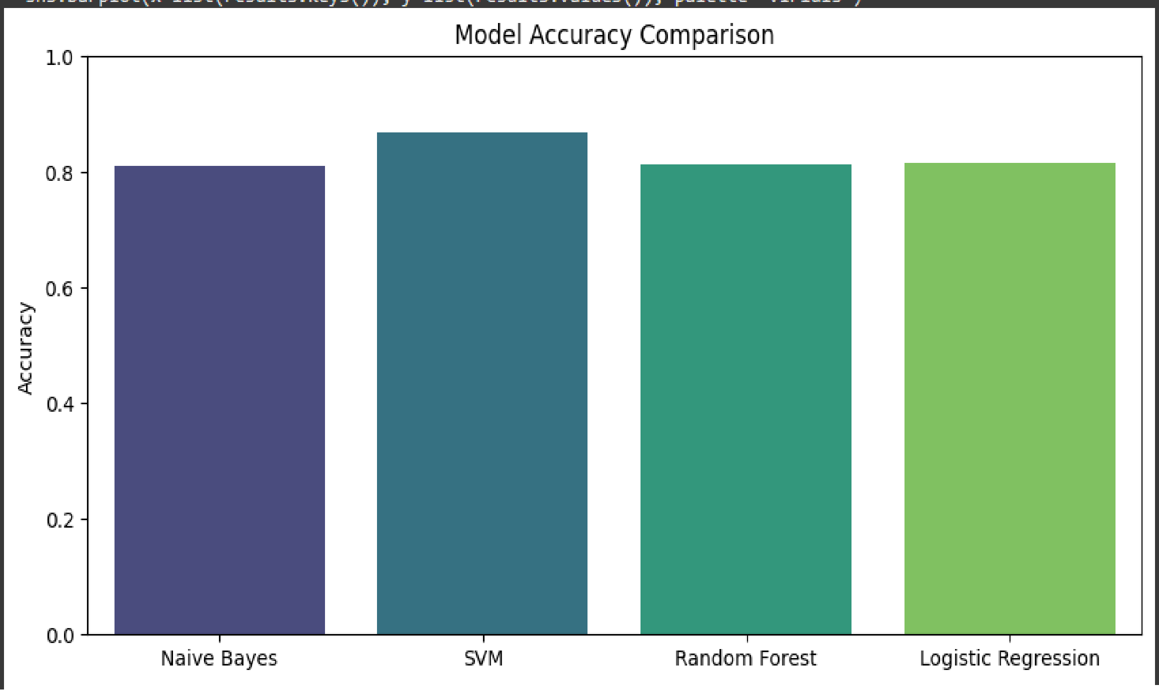
```
CPU times: user 109 ms, sys: 15.8 ms, total: 125 ms
Wall time: 11.6 s
```

Ah, the million-dollar question! The meaning of life is one of those questions that has plagued philosophers, theologians, and everyday people for centuries. There's no single, universally accepted answer, and that's kind of the point. Here's a breakdown of different perspectives:

1. **The Absurdist View:**
   - **Core Idea:** Life is inherently without meaning or purpose. We are born into a chaotic and indifferent universe, and any attempt to find inherent meaning is ultimately futile.
   - **Implication:** This doesn't necessarily lead to despair. Instead, absurdists encourage embracing the freedom to create your own meaning, rebel against the absurdity, and find joy in the present moment.
   - **Key Thinkers:** Albert Camus, Jean-Paul Sartre
2. **The Nihilist View:**
   - **Core Idea:** Life is meaningless, and there are no objective values or morals. There is no inherent right or wrong, and existence is ultimately pointless.
   - **Implication:** This view often leads to pessimism and a rejection of traditional values. It can be seen as a more extreme version of absurdism, with less emphasis on creating personal meaning.
   - **Key Thinkers:** Friedrich Nietzsche (though he also offered potential solutions to nihilism), some interpretations of Arthur Schopenhauer
3. **The Existentialist View:**
   - **Core Idea:** Existence precedes essence. We are born into the world without a pre-defined purpose, and it is up to us to create our own meaning through our choices and actions.
   - **Implication:** This places a strong emphasis on personal responsibility and freedom. We are condemned to be free, meaning we are responsible for the meaning we create.
   - **Key Thinkers:** Jean-Paul Sartre, Simone de Beauvoir, Albert Camus (overlaps with absurdism)
4. **The Religious/Spiritual View:**
   - **Core Idea:** Life has a divinely ordained purpose, often involving serving a higher power, following religious doctrines, and striving for spiritual enlightenment or union with the divine.
   - **Implication:** This view provides a sense of comfort and purpose, offering a framework for understanding our place in the universe and providing guidelines for moral behavior.
   - **Examples:** Christianity (serving God and achieving salvation), Buddhism (achieving enlightenment and liberation from suffering), Islam (submission to Allah and following the teachings of the Quran)
5. **The Humanistic View:**




```
sns.barplot(x=list(results.keys()), y=list(results.values()), palette='viridis')
```



Positive Reviews Word Cloud

Negative Reviews Word Cloud

8.REFERENCES:

[1] Jagdale, R. S., Shirsat, V. S., & Deshmukh, S. N.
"Sentiment Analysis on Product Reviews Using Machine Learning Techniques," *Cognitive Informatics and Soft Computing, Advances in Intelligent Systems and Computing*, vol. 768, pp. 639–648, 2019. DOI: 10.1007/978-981-13-0617-4_61.

[2] Shah, A.
"Sentiment Analysis of Product Reviews Using Supervised Learning," *Reliability: Theory & Applications (RT&A)*, Special Issue No. 1 (60), vol. 16, pp. 243–253, Jan. 2021.

[3] Sridhar, A. S., & Nagasundaram, S.
"Product Reviews Based Sentimental Analysis by Feature Extraction and Classification Using Machine Learning Algorithm," *5th International Conference on Electronics and Sustainable Communication Systems (ICESC 2024)*, IEEE, 2024, pp. 891–892. DOI:10.1109/ICESC60852.2024.10689776.

[4] Sabapathi, P. R., & Kaliyamurthie, K. P.
"Analysis of Customer Review and Predicting Future Release of the Product Using Machine Learning Concepts," *2022 International Conference on Communication, Computing and Internet of Things (IC3IoT)*, IEEE, 2022. DOI:10.1109/IC3IoT53935.2022.9767956.

[5] Sultana, N., Kumar, P., Patra, M. R., Chandra, S., & Alam, S. K. S.
"Sentiment Analysis for Product Review," *ICTACT Journal on Soft Computing*, vol. 9, no. 3, pp. 1913–1919, Apr. 2019. DOI: 10.21917/ijsc.2019.0266

[6] https://aclanthology.org/W02-1011/

[7]https://www.researchgate.net/publication/248832100_Thumbs_Up_or_Thumbs_Down_Semantic_Orientation_Applied_to_Unsupervised_Classification_of_Reviews

[8] https://www.cs.uic.edu/~liub/publications/kdd04-revSummary.pdf

[9] http://www.lrec conf.org/proceedings/lrec2006/pdf/384_pdf.pdf

[10] https://aclanthology.org/P05-1015.pdf

[11]https://www.researchgate.net/publication/234131319_Efficient_Estimation_of_Word_Representations_in_Vector_Space

[12] https://aclanthology.org/P04-1035.pdf

[13] https://dl.acm.org/doi/10.1007/s00500-019-04478-2

[14] https://link.springer.com/chapter/10.1007/978-3-540-68825-9_30

[15]https://www.sciencedirect.com/science/article/pii/S2666307424000536

[16]https://www.cs.uic.edu/~liub/FBS/SentimentAnalysis-and-OpinionMining.pdf

[17]https://www.researchgate.net/publication/370055745_Sentiment_Analysis_of_Product_Reviews_for_E-_Commerce_Recommendation_based_on_Machine_Learning
.
[18]https://www.researchgate.net/publication/360819202_Sentiment_Analysis_on_Reviews_of_E-commerce_Sites_Using_Machine_Learning_Algorithms

[19]https://www.researchgate.net/publication/353249415_Sentiment_analysis_of_e-commerce_application_in_Traveloka_data_review_on_Google_Play_site_using_Naive_Bayes_classifier_and_association_method