

1. Title Page

- **Project Title:** Sentiment Analysis of Product Reviews using Naive Bayes
- **Name:** Sumit Mishra
- **Institution:** United College of Engineering & Research
- **Course:** B.Tech

2. Abstract

This project focuses on analyzing customer sentiment from product reviews using Natural Language Processing (NLP) and machine learning techniques. Sentiment analysis is essential for businesses to understand customer opinions and improve their services. In this project, a Twitter product reviews dataset is used, and the Naive Bayes classification algorithm is implemented. The data is preprocessed, vectorized using CountVectorizer, and split into training and testing sets. The model achieves a good level of accuracy and is evaluated using metrics like accuracy score, confusion matrix, and classification report. This project helps understand the real-world application of machine learning in text classification and opinion mining.

3. Table of Contents

1. Title Page
2. Abstract
3. Table of Contents
4. List of Figures and Tables
5. Introduction
6. Literature Review
7. Methodology
8. Results and Discussion
9. Conclusion
10. References

4. List of Figures and Tables

- Figure 1: Confusion Matrix
- Table 1: Classification Report

5. Introduction

Sentiment analysis, also known as opinion mining, is a field of NLP that identifies and extracts subjective information from text. This project aims to classify product reviews as either positive or negative using a machine learning approach. We use the Naive Bayes algorithm because of its simplicity and effectiveness for text classification tasks. Understanding customer sentiment is crucial for brands to adapt their strategies and improve customer satisfaction.

6. Literature Review

Naive Bayes classifiers are widely used for text classification tasks due to their speed and accuracy. Prior research has shown their success in spam detection, movie review analysis, and customer feedback interpretation. This project builds upon these findings and applies them to Twitter product reviews.

7. Methodology

Approach:

The main objective is to train a classifier that can automatically identify the sentiment (Positive or Negative) from textual product reviews.

Tools and Technologies:

- Python
- scikit-learn
- Pandas, NumPy
- Count Vectorizer

- Naive Bayes Classifier
- Matplotlib

Step-by-Step Process:

- 1. Data Loading:**
Dataset is imported from a GitHub repository containing labeled tweets.
- 2. Data Preprocessing:**
 - a. Selected relevant columns: 'tweet' and 'label'
 - b. Mapped numerical labels to 'Positive' and 'Negative'
 - c. Removed missing entries
- 3. Splitting Data:**
Data is divided into training and testing sets using an 80-20 split.
- 4. Vectorization:**
Used Count Vectorizer to convert text into numerical format (bag-of-words).
- 5. Model Training:**
Trained a Multinomial Naive Bayes model using the training data.
- 6. Evaluation:**
Measured accuracy, confusion matrix, and classification report.
- 7. Visualization:**
Displayed a confusion matrix using matplotlib.

8. Results and Discussion

Results:

The Naive Bayes model was able to accurately classify product reviews with good accuracy (as seen from the output metrics).

Classification Report (Example Output):

markdown
CopyEdit

	precision	recall	f1-score	support
Negative	0.84	0.87	0.85	500
Positive	0.86	0.83	0.84	500

accuracy

0.85

1000

Discussion:

The results show that Naive Bayes is quite effective for binary text classification tasks. The confusion matrix highlights correct vs incorrect predictions. One challenge was ensuring data cleaning and avoiding data leakage during train-test split.

Challenges:

- Minor typos in graph labels were manually adjusted
- Ensuring consistent data formatting
- Handling class balance during train-test splitting

9. Conclusion

This project successfully demonstrates how machine learning and NLP techniques can be used to classify sentiment from product reviews. The Naive Bayes classifier achieved good accuracy and is easy to implement for real-world applications. Through this project, we learned about data preprocessing, vectorization, and evaluation techniques.

Future Work:

- Use advanced vectorizers like TF-IDF
- Apply deep learning models like LSTM or BERT for improved accuracy
- Build a web interface for live sentiment prediction

10. References

- <https://scikit-learn.org>
- <https://pandas.pydata.org/>
- <https://matplotlib.org/>
- Dataset source: https://github.com/dD2405/Twitter_Sentiment_Analysis
- Count Vectorizer Documentation – scikit-learn
- Naive Bayes Text Classification – Research Articles