Solution Report: Sentiment Analysis Model for Reviews

1. Introduction

This report outlines the implementation of a **Sentiment Analysis Model** using Natural Language Processing (NLP) techniques. The model classifies product reviews into **positive, neutral, or negative sentiments** based on the provided dataset.

2. Dataset Overview

- **Dataset Name:** Product Reviews.csv
- **Key Columns:**
  - **Review Paragraph:** The review text.
  - **Rating (out of 5):** The star rating given by users.
  - **True Sentiment:** Derived from the rating column (Positive: 4-5, Neutral: 3, Negative: 1-2).

3. Data Preprocessing

To ensure clean and structured data, the following preprocessing steps were applied:

- **Text Cleaning:** Removal of special characters, HTML tags, and punctuations.
- **Lowercasing:** Standardizing text to lowercase.
- **Tokenization:** Breaking sentences into words.
- **Stopword Removal:** Eliminating common words (e.g., "the", "is").
- **Lemmatization:** Converting words to their base form (e.g., "running" → "run").

4. Sentiment Analysis Approaches

Multiple techniques were tested to predict sentiment:

4.1 Opinion Lexicon-Based Sentiment Scoring (NLTK)

- Uses NLTK's `opinion_lexicon`.
- Scores text based on positive and negative words.

4.2 VADER Sentiment Scoring

- Pre-trained lexicon specifically tuned for social media and short texts.
- Calculates polarity scores.

4.3 RoBERTa Sentiment Scoring

- Transformer-based deep learning model.
- Uses a pre-trained RoBERTa model fine-tuned on sentiment datasets.

4.4 Transformers Pipeline (Hugging Face)

- Implements a **zero-shot classification** method.
- Uses pre-trained Transformer models for contextual sentiment understanding.

5. Model Evaluation Metrics

To assess performance, we used the following metrics:

Metric	Description
Accuracy	Overall correctness of sentiment predictions
Precision	Correctness of predicted positive sentiments
Recall	Ability to identify actual positive sentiments
F1-Score	Balance between precision and recall

The confusion matrix was used to visualize classification errors.

"ACCURACY OF THE TRANSFORMER PIPELINE METHOD COMES OUT TO BE THE HIGHEST(more than 99%)"

6. Results and Findings

- **Lexicon-based models (NLTK & VADER)** provided fast but less accurate results.
- **Deep learning models (RoBERTa & Transformers)** achieved **higher accuracy (>85%)**.
- **Best Performing Model:** Transformer-based RoBERTa model, showing the highest **F1-score and recall**.

7. Conclusion

- The **deep learning-based models outperformed traditional lexicon-based models**.
- Preprocessing significantly improved model accuracy.
- Future improvements may include fine-tuning a Transformer model with domain-specific data.