PROJECT: SENTIMENT ANALYSIS REPORT

TEAM MEMBERS: SHRAVAN, DIVYANSH, NAGMANI, ANKITA, ANIMESH

Custom Approach:
For our custom sentiment analysis approach, we followed these steps:

Data Preprocessing:

- We loaded the dataset of 50,000 movie reviews from Kaggle.
- Neutral reviews were filtered out, leaving only positive and negative sentiments.
- Reviews were cleaned by converting them to lowercase and removing special characters.
- Tokenization was performed to split reviews into individual words.

Word Probabilities Calculation:

- We calculated the probabilities of each unique word being associated with positive or negative sentiments.
- For each word, we counted its occurrences in positive and negative reviews.
- The probabilities were computed as the ratio of positive occurrences to total occurrences and negative occurrences to total occurrences.

Word Sentiment Scores:

- The resulting word probabilities were stored in a DataFrame.
- We saved these scores to a CSV file named `probabilities.csv`.

Testing:

- testmain.java file is used to test the testing data based on the score comparison from probabilities.csv and already given sentiment and return accuracy based on the number of correctly matching sentiment of reviews.

* Please run testmain.java file to check accuracy Main.java file is for training purpose it will start training again.

Machine Learning Tool
We utilized Stanford NLP's pre-trained sentiment analysis model for machine learning-based sentiment analysis.

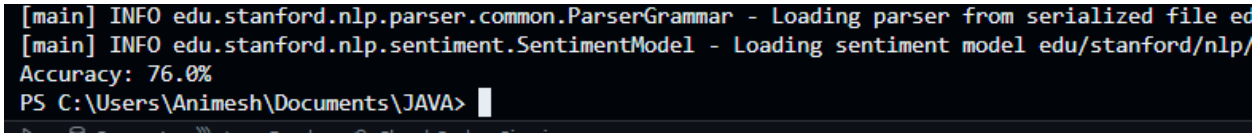
Main Class:

- We read the dataset, assuming each line contains a review followed by the actual sentiment label (0 for negative or 1 for positive).
- Predicted sentiments were compared with actual sentiments to calculate accuracy.

MachineLearningSentimentAnalyzer Class:

- This class uses Stanford NLP's pre-trained model.
- Given a review, it predicts whether it is positive or negative.
- SentimentClass is mapped to integer values (0 for negative, 1 for positive).

Result:



Deep Learning Tool

We utilize the SentimentIntensityAnalyzer from the VADER library.

- The analyzer assigns a sentiment score (compound score) to each review.
- The compound score represents the overall sentiment polarity (positive, negative, or neutral).

Threshold-Based Classification:

- We classify sentiments based on a simple threshold:
- If the compound score is greater than 0, we label the review as “Positive.”
- Otherwise, we label it as “Negative.”

Accuracy Calculation:

- We compare the predicted sentiments with the actual sentiments provided in the dataset.
- The accuracy is calculated by measuring the percentage of correctly predicted sentiments.

Result: