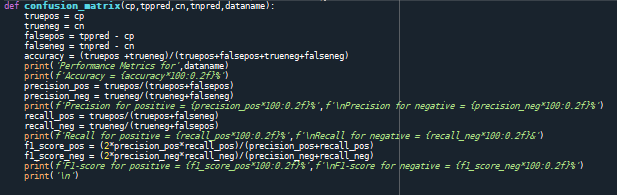
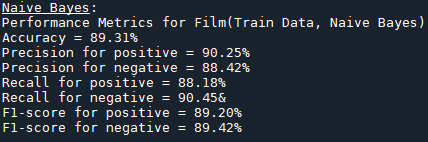
STEP 1:

The datasets used for this assignment are based on reviews from Rotten Tomatoes and Nokia. From the Blackboard website, it has been downloaded.

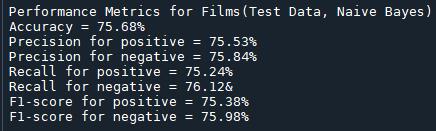
STEP 2:

Files containing the dataset are read, and positive/negative sentences are segmented into words and stored in a sentiment dictionary. The Naive-Bayes model is then trained and tested on separate datasets, created by partitioning sentiment sentences. Classification results are obtained by calculating scores from the confusion matrix using a specific function:

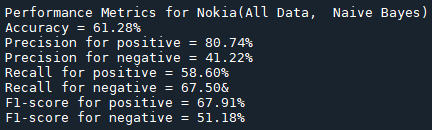
**Fig1: Confusion Matrix Python Function**

In this context, True Positive, False Positive, True Negative, and False Negative values are derived from the parameters, and various metrics such as Accuracy, Precision, Recall, and F1-Score are computed. The model underwent testing on the train data, test data and Nokia Data, yielding the following results:

**Fig2: Results for Naive-Bayes Classification on train data**

 The Naive-Bayes model performs well on training data with high accuracy and balanced metrics for positive and negative sentiments. However, robust testing on unseen data is essential to validate its predictive capabilities and assess generalizability.

**Fig3: Results for Naive-Bayes Classification on test data**

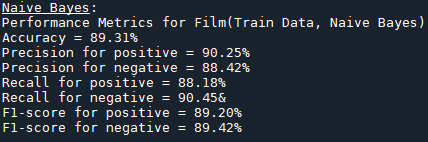
Observing the results, the testing accuracy is approximately 76%. Notably, Precision, Recall, and F1-score values for both positive and negative sentence predictions exhibit similarity. This suggests that the model is demonstrating a balanced performance, achieving a satisfactory level of accuracy.

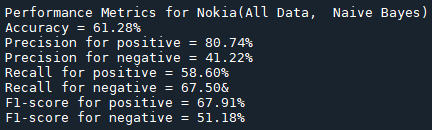
**Fig4: Results for Naive-Bayes Classification on Nokia data**

The displayed accuracy is around 61%, with high precision for positive sentiment but notably low precision for negative sentiments. Conversely, recall and F1-scores for positive sentiments are low, while for negative sentiments, the pattern is reversed. In summary, the Naive-Bayes model applied to the Nokia dataset shows mixed performance across various metrics, indicating varied efficacy in sentiment prediction.

STEP 3:

Here, the testBayes function has been invoked with the training dataset and data sourced from reviews of Nokia products. The outcomes are reported as follows:



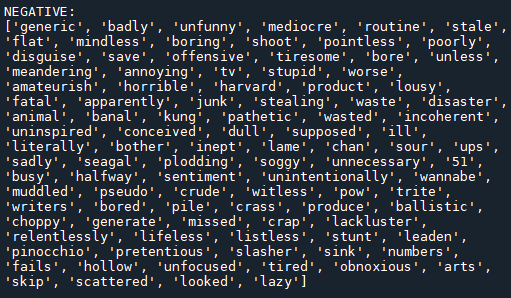


**Fig5: Results for Classification on train data and Nokia Data**

The Naive-Bayes Classifier performed significantly better on the training data, with higher metrics, primarily due to its training on film-related data. However, when applied to the Nokia dataset, specifically centered around reviews of Nokia products, its effectiveness diminished, resulting in comparatively lower metric values. The discrepancy in performance is likely due to the distinct nature of the two datasets, where the classifier excelled in a film-related context but faced challenges in predicting sentiments for Nokia product reviews.

STEP 4:

The negative sentiment highly predictive words which are included in the sentiment dictionary are given below:



**Fig6: Negative highly predictive words included in Sentiment Dictionary**

The positive sentiment highly predictive words which are included in the sentiment dictionary are given below:

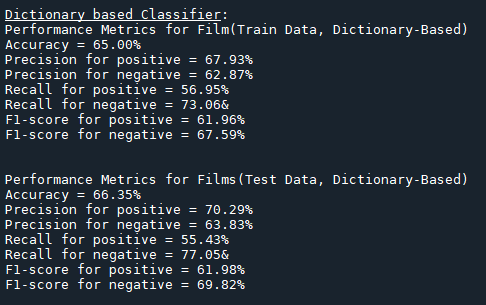
**Fig7: Positive highly predictive words included in Sentiment Dictionary**

Reviewing the positive and negative word list exposes inaccuracies in the Sentiment Dictionary, where neutral terms such as 'routine,' 'disguise,' and 'shoot' are mislabeled as negative. Conversely, neutral words like 'physical,' 'portrait,' and 'record' are incorrectly placed in the positive category. Notably, the term 'sadness,' typically associated with negativity, is found in the positive terms list. Hence, we can conclude that the words selected by the model are not so much in good sentiment terms.

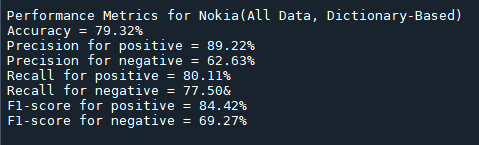
Counting the words in both lists reveals a total of 94 words that the model considers highly predictive and includes in the Sentiment Dictionary.

**Fig8: Calculated Number of words in the Sentiment Dictionary**

STEP 5:

 The classification results for function testDictionary() are obtained using the same Confusion Matrix mentioned above, which is called within it. The attained results are provided below:

**Fig9: Results for Dictionary-based Classification on train and test dataset from film reviews**



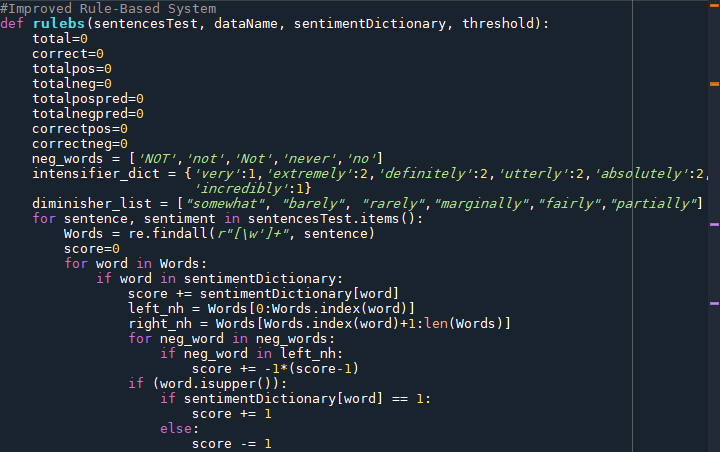
**Fig10: Results for Dictionary-based Classification on Nokia dataset**

Comparing the Dictionary-based method to the Naive-Bayes model:

* Naive-Bayes consistently outperforms across all datasets and metrics.
* In film reviews, Naive-Bayes exhibits superior accuracy, precision, recall, and F1-scores.
* The Dictionary-Based Classifier performs better on the Nokia dataset in accuracy and positive sentiment precision, but Naive-Bayes achieves a more balanced F1-score.
* Naive-Bayes effectively handles both positive and negative classes, while the Dictionary-Based Classifier faces challenges, particularly in Recall for the Film dataset.

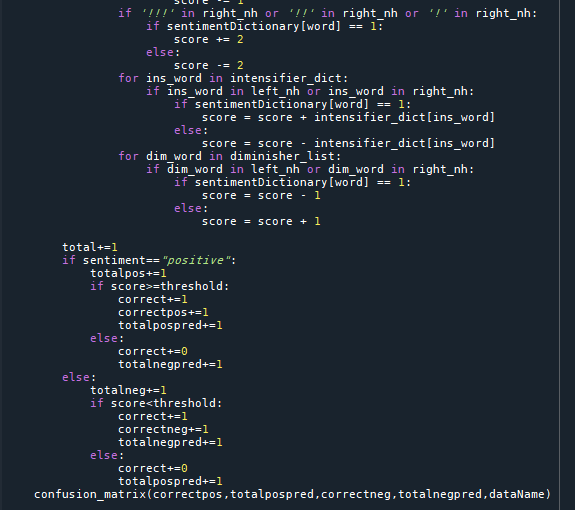
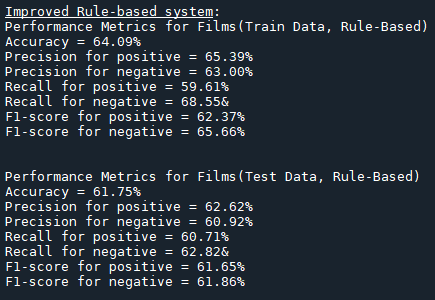
Key conclusions on statistical and rule-based approaches:

* Statistical methods, like Naive-Bayes, consistently perform well across diverse scenarios, showcasing balanced efficacy.
* Rule-based approaches, such as the Dictionary-Based method, may excel in specific contexts but require more customization for effective generalization.

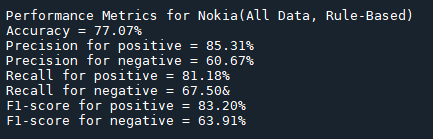
An improved rule-based approach has been implemented in code using the function below:

**Fig11: Rule-Based function (part 1)**

**Fig12: Rule-Based function (part 2)**

After applying and calling the function for both the film review and Nokia dataset, the following classification results are obtained:

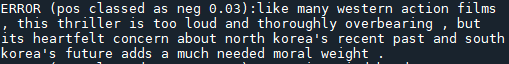
**Fig13: Results for improved Rule-based Classification on train and test dataset from film reviews**

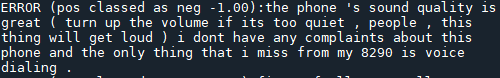


**Fig14: Results for improved Rule-based Classification on Nokia dataset**

The enhanced rule-based system performs moderately well, achieving the highest accuracy in the Nokia dataset. F1-scores reflect a balanced precision-recall trade-off, indicating overall well-rounded performance. This system excels in situations requiring specific domain knowledge.

STEP 6:

An error analysis has been executed by one of the cases of testBayes and testDictionary function calls each and setting PRINT\_ERRORS=1. Some mistakes obtained for both cases are listed below:

**Fig15: Mistakes for Naive-Bayes model**

**Fig16: Mistakes for Dictionary-based approach**

The model's errors can be attributed to several factors:

* Errors occur when the model predicts a probability less than 0.5 for genuinely positive sentiments (False Negative) and vice versa for genuinely negative sentiments (False Positive), influenced by the direct proportionality between the calculated probability and the probability of a word being positive (pPosW).
* Similar mistakes are observed in the Dictionary-based approach, where errors stem from scores falling below the threshold for positive predictions and above the threshold for negative predictions.
* Some errors arise from sentences conveying sentiment through their 'look and feel,' which may not align with calculated scores or probabilities based on the words they contain. For instance,

Despite a sentence seeming negative, the calculated probability exceeds 0.5 due to the inclusion of strongly positive words like 'masterpiece,' leading to a positive prediction.

*  In mixed sentiment sentences, stronger sentiment words often dominate weaker ones. For example,

Here, "great" and "perfect" being positive overpower the smaller negative impact of "tid bit," leading to the prediction as positive sentiment instead of negative.