REPORT FOR SENTIMENT ANALYSIS OF CUSTOMER REVIEW

BY
UDOETTE, Ubong
Student ID: 21085920

1.0 Problem Statement and Data Description:

This project report covers my approach in developing a binary sentiment classifier (Naïve Bayes) that labels a restaurant customer reviews as positive (appreciative) or negative (critical) based on collected historical data. The report also shows the comparison in performance with Logistic Regression model on the same dataset using pre-built libraries. Further extension has been added to the Logistic Regression Model to capture Neutral sentiments and Aspect Based Sentiment Analysis.

1.2 Data Acquisition:

The data used in training Naïve Bayes classifier model was collected from Kaggle: https://www.kaggle.com/datasets/vigneshwarsofficial/reviews.

The dataset used for Aspect Based Sentiment Analysis for Logistic Model was collected from Github: https://github.com/yangheng95/ABSADatasets/tree/v2.0/datasets/acos_datasets/502.Restaurant14

1.3 Data Description:

The dataset used to train the Naïve Bayes classifier is a balanced dataset containing 500 positives and 500 negatives reviews. The plot below shows the count of these reviews in found in the dataset.

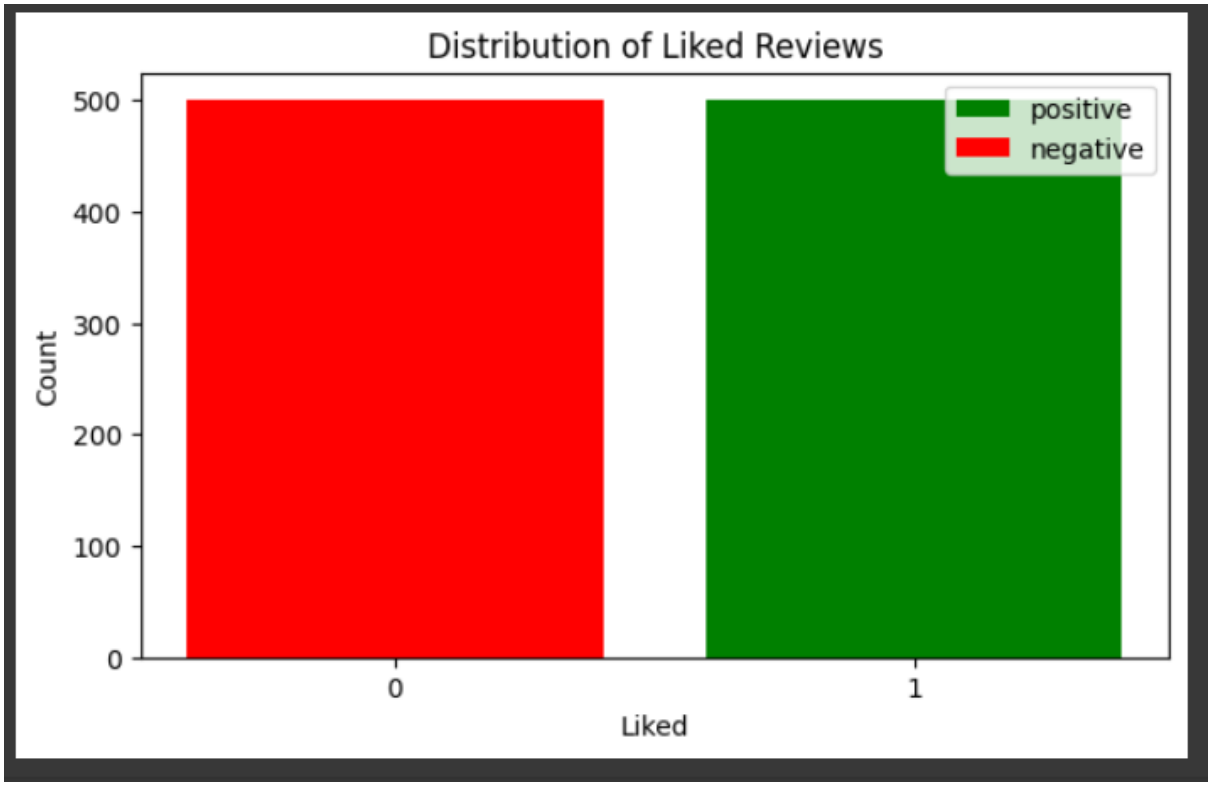

Figure 1 – showing the count distribution the features labels in the datase.

2.0 Data Preprocessing and Feature Extraction

I used pre-built preprocessing tools; NTLK, Spacy and also Python Regular Expression library to preprocess the reviews for Naïve Bayes Model.

My preporcessing steps included:

1. Converting text to lowercase
2. Removing punctuations, special characters, and numbers
3. Stop word removal
4. Lemmatization.

2.1 Feature Extraction

To convert the reviews into a numerical representation suitable for Naïve Bayes Classifier, I used the Bag of Word model (Term Frequency) to capture the frequency of each word in a document in the corpus after preprocessing. I used TF-IDF model for feature extraction transforming the reviews to sparse matrix when training the same dataset with Logistic Regression Model.

3.0 Model Implementation

3.1 Naïve Bayes Classifier

I implemented Naïve Bayes Classifier from Scratch using a statistical approach and improved its perfomance using Laplacian smoothing and Log-likelihood techniques.

The downloaded dataset was read into a suitable Python data object and splitted into 80% train and 20% test data randomly using a scikit-learn library. The train portion of the dataset was then used to train the model and the test portion of the dataset was used to test the performance of the model. The model evaluation shows **74%** accuracy**, 80%** precision**, 66%** Recall and **73%** F1-Score.

The confusion matrix is displayed below:

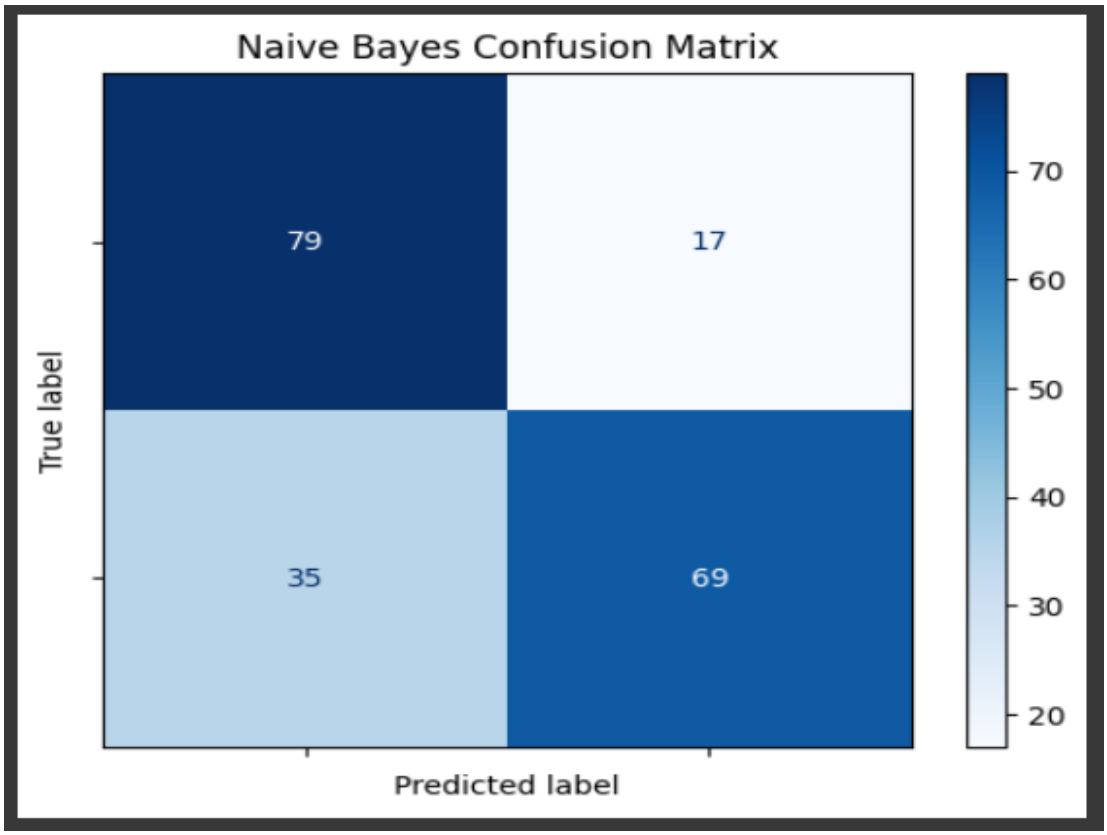

Fig2- shows confusion matrix for Naïve Bayes Classifier

3.2 Logistic Regression Classifier

To acertain the performance level of the Naïve Bayes Classifier as compared to most commonly used models used for similar task, I developed a Logistic Regression model with the same dataset to make comparsion. Result shows that the Logistic Regression peforms better than the Naïve Bayes Classifier. The evaluation results has **81%** accuracy, **88% precision**, **74%** Recall and **80%** F1-Score.
In developing the Logistic model, I used the pre-built libraries from scikit-learn. Then splitted the data into train (80%) and test (20%) randomly with the same seed number I used for developing Naïve Bayes Classifier. I used the TF-IDF for feature extraction to transform the data into numerical representation.

The confusion matrix for this model is presented below.

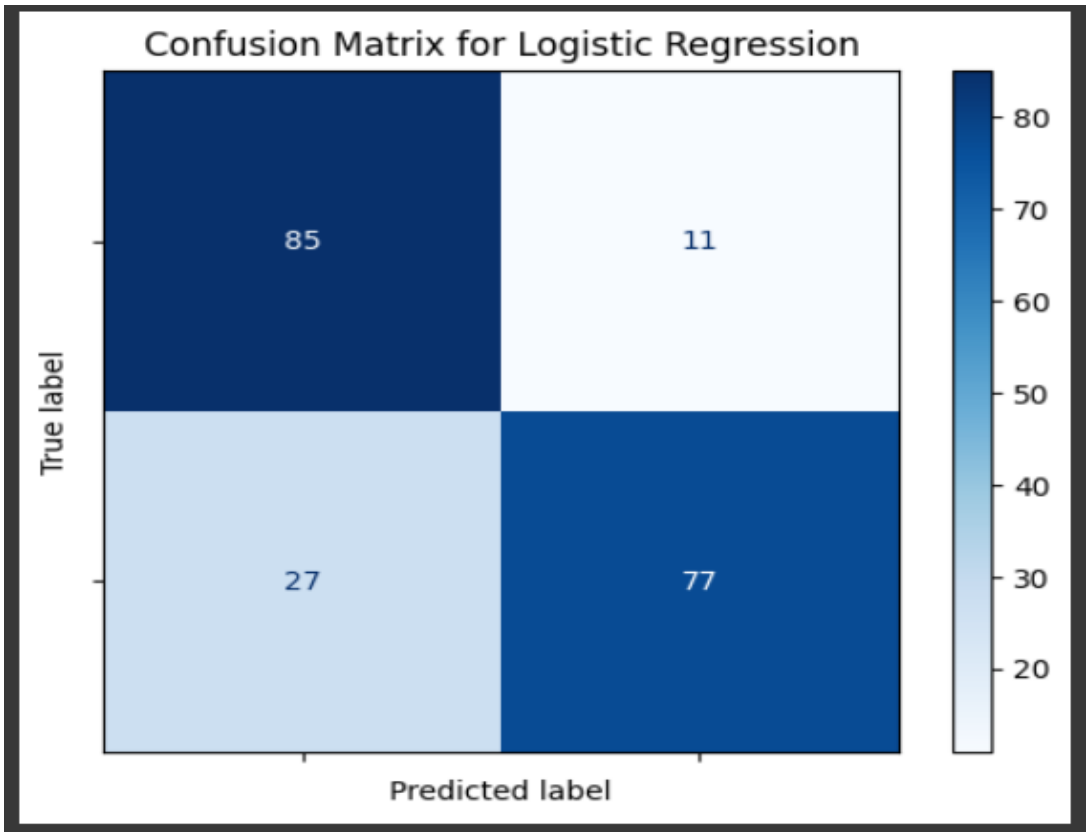

Fig2- shows confusion matrix for Logistic Regression Classifier

3.2.1 Advanced Sentiment Analysis

3.2.2 Neutral Sentiment Analysis
The Logistic Regression model was also extended to capture advanced sentiment analysis including predicting Neutral Sentiment. To classify a neutral sentiment, I used the confidence score, that is the prediction probabilities from the model and defined a threshold for a review to be either considered positive, negative or neutral. The motivation behind this method was that neutral senitment is a combination of positive and negative sentiments, the predicted probability of a neutral sentiment will be somewhat average, that is not high on the positive or negative side.

3.2.3 Aspect Based Sentiment Analysis
To capture a aspect based sentiment, I trained a new Logistic Regression model on a new but similar data set. The new dataset contained aspect lables which the former dataset used for Naïve Bayes classifier did not have. The dataset was in **jsonl** format which was parsed to panda dataframe and then cleaned (removing null values). Feature extraction was then done on the clean dataset and fed into the new model for training. In evaluating the model, the model has an overall accuracy of **88%**, **86%** Precision, **67%** Recall and **71%** F1-Score.

Challenges

I encountered some difficulties in the preprocessing stage. Some reviews from the data set contained some characters like ellipses, that were not removed after preprocessing using pre-built libraries. To overcome this challenge, I used Python Regular Expression to remove all characters and non-numerical characters. Secondly, I could only do Lemmatization with NTLK but with Spacy.
The Neutral sentiment analysis can be improved by training the model on a dataset set containing an additional label for the neutral class to capture neutral sentiment more efficiently.