**Text Classification with Naive Bayes: Report**

Sri Sai Durga Katreddi

11626267

**Task 1: Preliminaries**

**Naive Bayes Classifier Overview**

Before delving into the implementation, a brief overview of the Naive Bayes classifier was conducted. This included understanding how to compute and apply the Naive Bayes rule, as well as estimating the necessary probabilities.

**Dataset Information**

The dataset used for sentiment classification comprises customer reviews from six different topics, namely books, camera, DVD, health, music, and software. The data has been preprocessed, with one review per line, and each text split into separate words.

**Task 2: Estimating Parameters for Naive Bayes**

**Frequency Counting in Python**

To facilitate frequency counting, the Counter data structure from the collections module was employed. Different methods were explored to compute the frequencies of words in a collection of documents.

**Logarithmic Probabilities**

To address issues with numeric precision, the implementation utilized logarithmic probabilities. Logarithms were applied to prevent the occurrence of zero probabilities during multiplication.

**Task 3: Classifying New Documents**

**Training the Naive Bayes Model**

A Python function, train\_nb, was implemented to estimate probabilities in the Naive Bayes model using a training set of documents. Laplace smoothing was incorporated for improved results.

**Scoring Documents**

A function, score\_doc\_label, was developed to apply the Naive Bayes formula and compute the logarithm of the probability of observing words in a document and a sentiment polarity label.

**Classifying New Documents**

Building upon the previous function, classify\_nb was implemented to classify new documents by returning the predicted sentiment label.

**Task 4: Evaluating the Classifier**

**Classification of Test Set**

A function, classify\_documents, was created to classify each document in the test set and return the list of predicted sentiment labels.

**Accuracy Calculation**

To evaluate the classifier's performance, an accuracy function was implemented. The accuracy on the test set was computed and reported.

A blue squares with white text

Description automatically generated

The classifier seems to perform well in identifying negative sentiments (TN = 1230).

However, it struggles to correctly identify positive sentiments (TP = 0), and there are a significant number of instances (1153) where it misclassifies negative sentiments as positive.

This confusion matrix raises concerns about the classifier's ability to accurately distinguish positive sentiments. Further investigation into the misclassification patterns and potential improvements in the model may be needed to enhance its performance, especially regarding positive sentiment predictions.

**Task 5: Error Analysis**

**Identifying Misclassified Documents**

Misclassified documents were identified and presented. Insightful comments were provided on why these instances might have been challenging for the classifier.

A graph of a line

Description automatically generated

**Task 6: Cross-Validation**

**Implementation of Cross-Validation**

A cross-validation method was implemented to obtain a more reliable estimate of the classifier's accuracy. The dataset was divided into folds, and the classifier was evaluated across different subsets.

**Cross-Validation Accuracy**

The overall accuracy of the classifier, obtained through cross-validation, was reported.

In conclusion, the implemented Naive Bayes classifier demonstrated its ability to perform sentiment classification on customer reviews. The report provides a detailed account of the implementation, its accuracy, and insights gained through error analysis. Cross-validation was employed to enhance the reliability of the accuracy estimate.