------------------------------------------------------------------------------------------------------------------------------------------

Lab Assignment: 2

Full name: Gardyan Priangga Akbar

Student number: 7695883

Email address: [gardyanakbar99@gmail.com](mailto:gardyanakbar99@gmail.com)

------------------------------------------------------------------------------------------------------------------------------------------

**Text Classification with Naïve Bayes Classifier**

This lab assignment aims to explore the use of Naïve Bayes classifier for the purpose of text classification. This includes developing an understanding of Bayes’ formula and the preprocessing steps needed to train a Naïve Bayes model for text classification. The dataset used for this assignment is the 20-newsgroup dataset which consists of 20,000 texts divided into 20 classes. This assignment was conducted using a machine with the following specifications:

Processor: Intel i7-9700K

Graphics Card: Nvidia RTX 2070

RAM: 32 GB

Answers to questions:

1. Dd
2. D
   1. The first step is to load the dataset and split them into training and test sets. Next is the configuration of stopwords, which are words that are either too common or too rare and carries no significant weight or meaning in providing context to classify the texts. A helper function is then created to remove punctuations from the text. This is done because punctuations such as full stop or an indent is not needed to classify a text. Additionally, numerical strings, words with only 1 or 2 characters, and empty strings are to be removed as well for the same reason. The step after that is creating a helper function to tokenize the text. Tokenizing means taking the sentences and breaking them down into a list of words. Removing any metadata such as \n is the next step as, again, they do not mean anything for the classification task. A flatten function is also created in order to convert the text into an input the model can better understand. Afterwards, a dictionary needs to be created to store the vocabulary of each document as well as their frequency. This is then appended as a feature for training. Finally, the Naïve Bayes classifier is called.
   2. The classification accuracy on the test set was 0.7658 and 0.5632 for when using sklearn and the custom naïve bayes implementation respectively. Below is the full classification report on each of the classes for both cases.  
      Table

      Description automatically generated

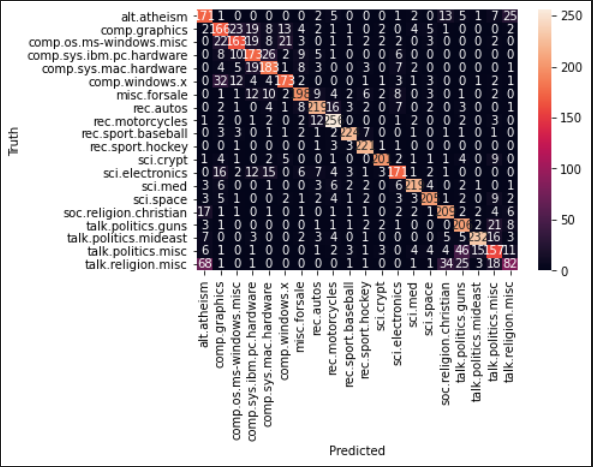
Figure - Classification report (naive bayes implemented using sklearn)

Table

Description automatically generated

Figure - Classification report (custom naive bayes implementation)

Precision is the ratio between the amount of correctly predicted classes and the total correct classes. In other words, a high precision score means there is a low amount of false positives. Recall is the ratio of correctly classified classes over to observation in the actual class. For example, from all the text that falls under the alt.atheism class, recall shows how much of the text that we classified it as such. F1-score is the weighted average of precision and recall. Support is simply the amount of text in each class (actual, not predicted).

* 1. Confusion matrix for Naïve Bayes implementation from sklearn:  
       
     The classifier confuses the most between alt.atheism and talk.religion.misc. In my opinion, this is understandable as the topic of atheism is usually closely related to religion.  
       
     Confusion matrix for custom Naïve Bayes implementation:  
     Graphical user interface

     Description automatically generated with medium confidence  
     This classifier confuses the most between talk.politics.misc and rec.motorcycles. This is quite unexpected as the topics on motorcycles don’t usually relate or revolve around politics often.