
Lab Assignment: 2

Full name: Gardyan Priangga Akbar

Student number: 7695883

Email address: 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.

Answers to questions:

1. - It assumes class-condition independence, which means features are not dependent of another feature. For example, in the phrase "Dear Friend", the probability of "Friend" given a particular class does not depend on the probability of "Dear" given the same class.
   - It assumes that the position of a word in a sentence does not matter. For example, the phrase "Dear Friend" and "Friend Dear" is exactly the same according to Naïve Bayes.
2. The basic idea of the multinomial Naïve Bayes Classifier is determining the highest probability that a particular phrase or sentence belongs to a particular class. This is determined by multiplying the probability of a particular class over all other class with the probabilities of each individual feature given that it belongs to that class. For example, if the probability of the phrase "The soccer team won their match" belonging to class A is higher than the probability of it belonging to class B, the multinomial Naïve Bayes Classifier would determine that the phrase should be classified as class A.
3. a. 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.

b. 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.

	precision	recall	f1-score	support
alt.atheism	0.61	0.73	0.66	233
comp.graphics	0.60	0.66	0.63	253
comp.os.ms-windows.misc	0.73	0.65	0.69	249
comp.sys.ibm.pc.hardware	0.66	0.72	0.69	240
comp.sys.mac.hardware	0.69	0.78	0.73	236
comp.windows.x	0.78	0.72	0.75	240
misc.forsale	0.80	0.76	0.78	261
rec.autos	0.81	0.81	0.81	269
rec.motorcycles	0.82	0.90	0.86	284
rec.sport.baseball	0.91	0.90	0.91	248
rec.sport.hockey	0.87	0.96	0.91	231
sci.crypt	0.93	0.86	0.89	233
sci.electronics	0.77	0.70	0.74	244
sci.med	0.90	0.86	0.88	256
sci.space	0.88	0.83	0.85	246
soc.religion.christian	0.77	0.83	0.80	252
talk.politics.guns	0.68	0.83	0.75	249
talk.politics.mideast	0.90	0.83	0.86	281
talk.politics.misc	0.63	0.61	0.62	259
talk.religion.misc	0.57	0.35	0.43	236
accuracy			0.77	5000
macro avg	0.77	0.76	0.76	5000
weighted avg	0.77	0.77	0.76	5000

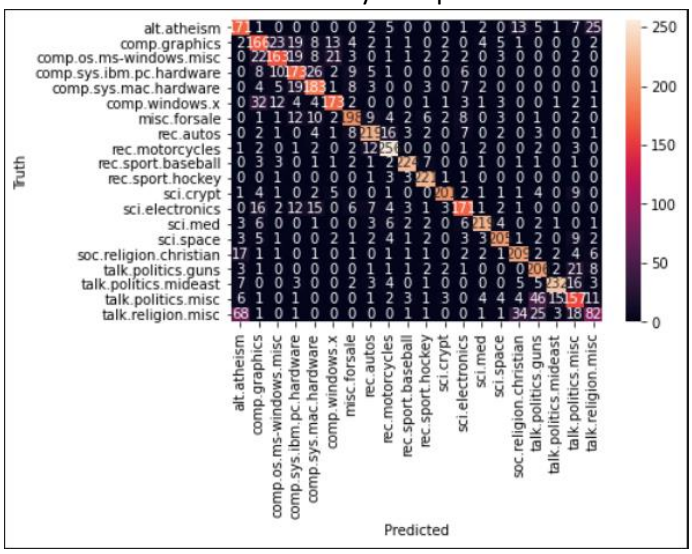
Figure 1 - Classification report (naive bayes implemented using sklearn)

	precision	recall	f1-score	support
alt.atheism	0.65	0.64	0.64	233
comp.graphics	0.51	0.57	0.54	253
comp.os.ms-windows.misc	0.85	0.26	0.40	249
comp.sys.ibm.pc.hardware	0.63	0.57	0.60	240
comp.sys.mac.hardware	0.92	0.37	0.53	236
comp.windows.x	0.52	0.80	0.63	240
misc.forsale	0.83	0.30	0.44	261
rec.autos	0.76	0.35	0.48	269
rec.motorcycles	0.98	0.31	0.47	284
rec.sport.baseball	0.98	0.61	0.75	248
rec.sport.hockey	0.86	0.84	0.85	231
sci.crypt	0.53	0.85	0.65	233
sci.electronics	0.77	0.32	0.46	244
sci.med	0.89	0.61	0.73	256
sci.space	0.81	0.63	0.71	246
soc.religion.christian	0.62	0.88	0.73	252
talk.politics.guns	0.75	0.42	0.54	249
talk.politics.mideast	0.51	0.93	0.66	281
talk.politics.misc	0.19	0.83	0.31	259
talk.religion.misc	0.51	0.20	0.29	236
accuracy			0.56	5000
macro avg	0.70	0.56	0.57	5000
weighted avg	0.70	0.56	0.57	5000

Figure 2 - Classification report (custom naive bayes implementation)

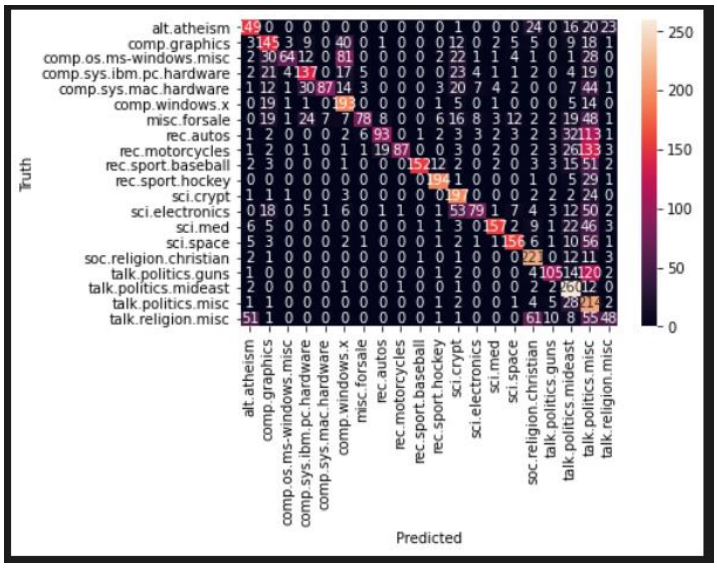
Precision is the ratio between the amount of correctly predicted classes and the total correct classes. In other words, a high precision score means there is a low amount of false positives. Recall is the ratio of correctly classified classes over to observation in the actual class. For example, from 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).

c. 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:



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.