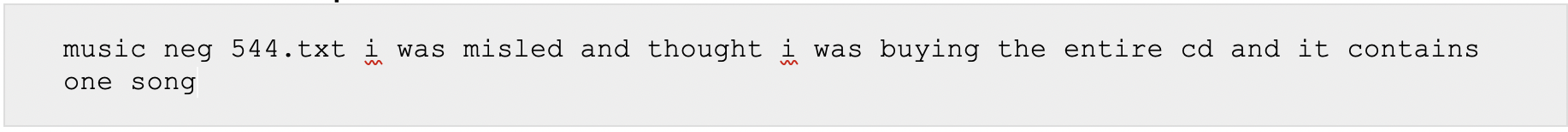
**INFO 5505 – APPLIED MACHINE LEARNING FOR DATA SCIETISTS**

**Assignment -5**

**Sentiment Analysis using Naïve Bayes**

**Data:**

The data is a collection of customer reviews from six of the review topics used in paper by Blitzer. The data has been formatted so that there is one review per line and the texts have been split into separated words (tokens) and lowercased as shown below.



A line in the form is organized in columns as shown below.

* 0: topic category label (books, camera, dvd, health, music, or software)
* 1: sentiment polarity label (pos or neg)
* 2: document identifier
* 3 and on: the words in the document

**Task Description:**

The main aim of this assignment is developing a Naïve Bayes classifier that can detect the sentiment in the reviews.

**Data Loading and Pre-Processing:**

As the data is in the form of text and the file is a text file the data (reviews text) is loaded using open function of codecs in Python. After opening the document each line of the document (each review) is split into words so that we can separate the sentiment labels and (negative and positive) and the actual text.

The actual text words and sentiment labels are separated into docs and labels.

Graphical user interface, text, application

Description automatically generated

**A picture containing graphical user interface

Description automatically generated**

The text and words and labels are separated. Now these two are put into a data frame so that further analysis could be easy. To form the data frame, I have converted all the words in each review into a text and transformed sentiment labels into numeric values.

**Converting docs into text:**

Background pattern

Description automatically generated with low confidence

After converting into text each review is like below.

Text

Description automatically generated

**Transforming sentiment labels into numeric values:**

In our data each review is already categorized with some sentiment whether it is negative or positive. So, there are two sentiments (Negative and Positive). I have transformed positive as 1 and negative as 0.

Table

Description automatically generated with low confidence

**Checking for class imbalance:**

When observations of one class outnumber the observations in another class of a target variable then we can say that there is a class imbalance in the data set. Class imbalance is a common problem in machine learning, especially in classification problems. Imbalance data can hamper the model accuracy big time.

In our data set to check the class imbalance I have used the count plot. By using the count plot, we can see the count of observations in each class of the diagnosis variable.

**Graphical user interface

Description automatically generated**

From the above plot it can be inferred that there is no class imbalance in the data.

**Splitting data into and train and test**

I have taken 80% of original data as training data set and 20% as test data set.

Table

Description automatically generated with low confidence

Graphical user interface

Description automatically generated with medium confidence

**Training the sentiment classifier:**

Here I have implemented Naïve bayes classifier to develop a model that can classify the text into a particular sentiment. Naïve bayes calculates the probability of each tag for our text sequences and outputs the tags with highest scores.

For example, knowing that the probabilities of appearance of the words “likes” and “good” in texts within the category “positive sentiment” are higher than the probabilities of appearance within the “negative” or “neutral” categories will help the Naive Bayes classifier predict how likely it is for an unknown text that contains those words to be associated with either category. Naive Bayes is commonly used in natural language processing.

Before developing a model, I have created a pipeline which contains two transformers and one Naïve bayes classifier. These two transformers are CountVectorizer and TfidfTransformer.

**CountVectorizer:** The vectorizer counts the number of words in each text sequence and creates the bag-of-word models.

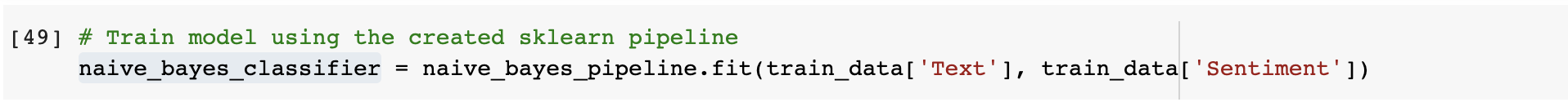
**TfidfTransformer:** The “Term Frequency Transformer” scales down the impact of words that occur very often in the training data and are thus less informative for the estimator than words that occur in a smaller fraction of the text samples. Examples are words such as “to” or “a”.

**MultinomialNB:** The multinomial Naive Bayes classifier is suitable for classification with discrete features (e.g., word counts for text classification). The multinomial distribution normally requires integer feature counts. However, in practice, fractional counts such as tf-idf may also work.

Graphical user interface

Description automatically generated with low confidence

Now fit our training data text variable and sentiment variable into the pipeline to develop the classifier.



**Predicting the labels for new documents:**

**Text

Description automatically generated with low confidence**

**Confusion Matrix:**

Text

Description automatically generated with low confidence

Graphical user interface, application

Description automatically generated

From the above confusion matrix below results can be inferred

In the test data set actually there are total 2383 reviews among which 1153 are actual negative reviews and 1230 are actual positive reviews

The Naïve bayes classifier we developed has predicted 69 reviews as positive from out of 2383 negative reviews and 139 reviews as negative out of 1230 positive reviews