**EMOTION DETECTION FROM TEXT USING SOCIAL MEDIA ANALYSIS**

**NLP Based Emotion Detection Model using Neattext and Scikit-learn:**

* Natural language processing helps computers understand speech and written text like a human being. This allows machines to compute necessary responses. One of the many NLP applications is emotion detection in text.
* The emotion detection model is a type of model that is used to detect the type of feeling and attitude in a given text. It may be a feeling of joy, sadness, fear, anger, surprise, disgust, or shame.
* An emotion detection model can classify a text into the following categories. By using emotion detection in text, businesses can know how customers feel about their brand and products. This helps businesses improve product quality and service delivery.
* In this tutorial, we will use Neattext and Scikit-learn to build our model. Neattext is a Python library used to preprocess our dataset. Neattext will clean the text dataset by removing stop words and other noise.
* This makes it easier for the model to use the dataset during training. We’ll use Scikit-learn to build our model.

**TABLE OF CONTENTS**:

* [Exploring our dataset](https://www.section.io/engineering-education/nlp-based-detection-model-using-neattext-and-scikit-learn/#exploring-our-dataset)
* [Loading dataset](https://www.section.io/engineering-education/nlp-based-detection-model-using-neattext-and-scikit-learn/#loading-dataset)
* [Getting started with Neattext](https://www.section.io/engineering-education/nlp-based-detection-model-using-neattext-and-scikit-learn/#getting-started-with-neattext)
* [Importing machine learning packages](https://www.section.io/engineering-education/nlp-based-detection-model-using-neattext-and-scikit-learn/#importing-machine-learning-packages)
* [Model features and labels](https://www.section.io/engineering-education/nlp-based-detection-model-using-neattext-and-scikit-learn/#model-features-and-labels)
* [Dataset splitting](https://www.section.io/engineering-education/nlp-based-detection-model-using-neattext-and-scikit-learn/#dataset-splitting)
* [Pipeline approach](https://www.section.io/engineering-education/nlp-based-detection-model-using-neattext-and-scikit-learn/#pipeline-approach)
* [Model fitting](https://www.section.io/engineering-education/nlp-based-detection-model-using-neattext-and-scikit-learn/#model-fitting)
* [Calculating the accuracy score](https://www.section.io/engineering-education/nlp-based-detection-model-using-neattext-and-scikit-learn/#calculating-the-accuracy-score)
* [Making a single prediction](https://www.section.io/engineering-education/nlp-based-detection-model-using-neattext-and-scikit-learn/#making-a-single-prediction)
* [Conclusion](https://www.section.io/engineering-education/nlp-based-detection-model-using-neattext-and-scikit-learn/#conclusion)
* [Reference](https://www.section.io/engineering-education/nlp-based-detection-model-using-neattext-and-scikit-learn/#references)

**NEATTEXT TOOL:**

➢Neattext is a software tool designed to help users manipulate and process text data. It uses natural language processing (NLP) techniques to automate tasks such as text cleaning, normalization, tokenization, and sentiment analysis.

➢With Neattext , users can easily preprocess text data and extract meaningful insights from it. It can be used to analyze customer feedback, social media posts, news articles, and other types of unstructured text data.

➢Neattext supports multiple programming languages such as Python, R, and Java, and it can be integrated with other data science tools like Jupyter Notebook, RStudio, and Tableau . Neatext can also be used to perform advanced text analysis tasks, such as topic modeling, text summarization, and entity recognition.

➢Its powerful algorithms and user-friendly interface make it a valuable tool for anyone working with text data, from data scientists and analysts to marketers and business owners.

**INTRODUCTION TO NLP ALGORITHM**

There are several NLP algorithms that are commonly used in text analysis. Here are some of the most important ones:

**1. Tokenization:** This algorithm breaks down text into individual words or phrases, which are called tokens. Tokenization is the first step in many text analysis tasks.

**2.Part-of-Speech (POS) tagging:** This algorithm tags each word in a text with its corresponding part of speech, such as noun, verb, or adjective. POS tagging helps to identify the syntactic structure of a sentence and can be used for tasks such as text classification and named entity recognition.

**3.Named Entity Recognition (NER):** This algorithm identifies and extracts named entities from text data, such as names of people, organizations, and locations. NER is often used in information extraction and text classification tasks

4. **Sentiment Analysis**: This algorithm determines the overall sentiment of a text, whether it is positive, negative, or neutral. Sentiment analysis is commonly used in social media monitoring, customer feedback analysis, and brand reputation management

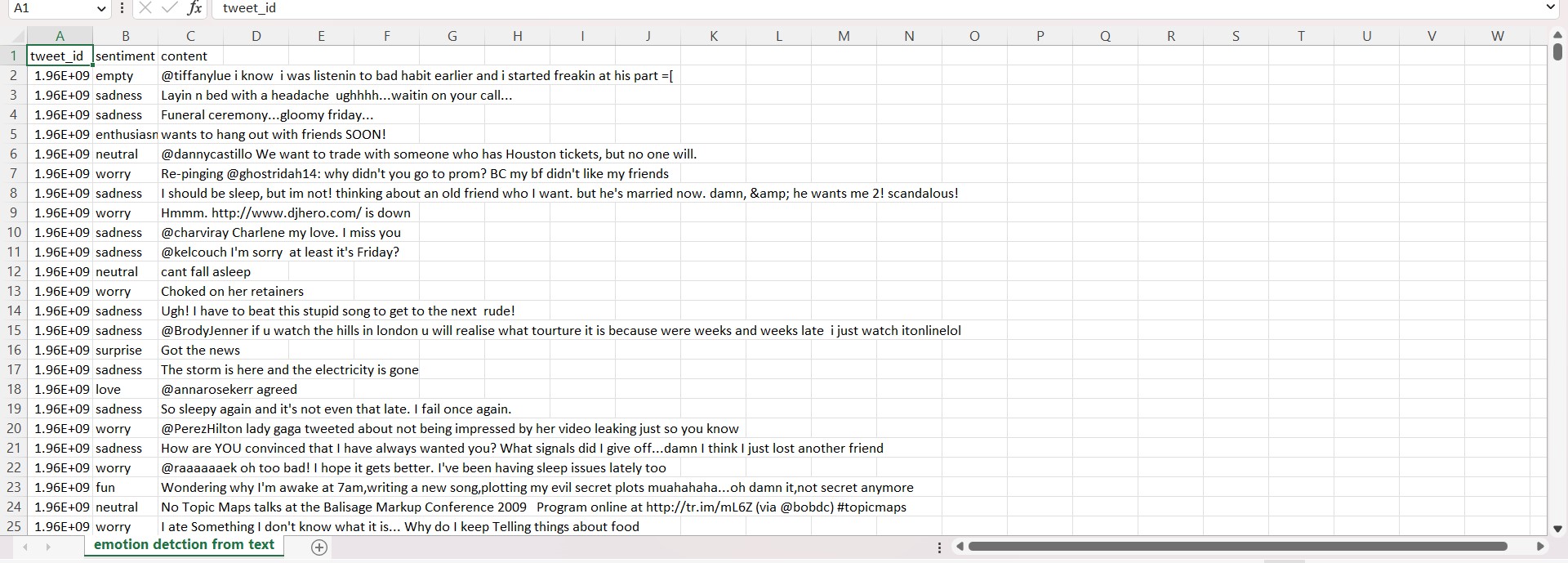
**5. Topic Modeling:** This algorithm discovers underlying topics in a text corpus by identifying cooccurring words and phrases. Topic modeling is useful for text summarization, document clustering, and content recommendation.

**6.Text Summarization:** This algorithm generates a concise summary of a longer text by extracting its most important information. Text summarization is commonly used in news article summarization, automatic document summarization, and social media monitoring.

**Exploring our dataset:**

The dataset has thirteen emotion labels that are named as follows:

* Neutral
* Worry
* Happiness
* Sadness
* Love
* Surprise
* Fun
* Relief
* Hate
* Empty
* Enthusiasm
* Boredom
* Anger

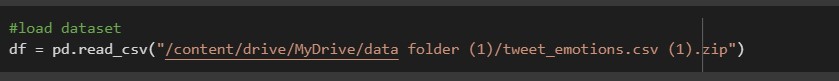
** Snip of the dataset**

**Coding part:**

* Import libraries:



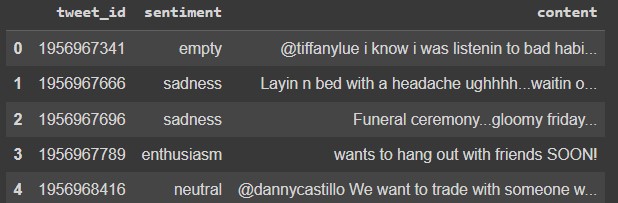
* Loading dataset:



* To see how the dataset is structured, using this commands:



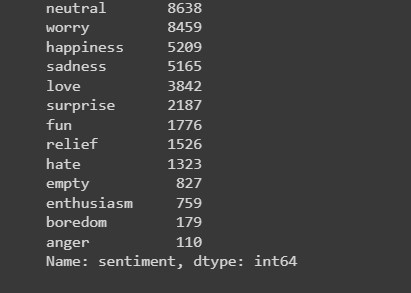
* The output is shown below:



* To display the total number of texts for each sentiment label:



* The output is shown below:



•Getting start with Neattext:

* To install Neattext, run this command



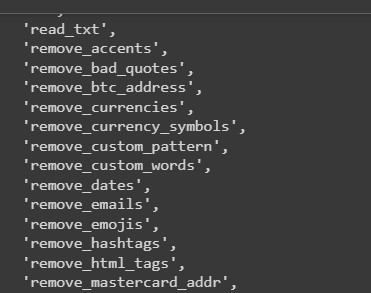
* We import the neattext as follows:



* To use neattext, we list all the methods and attributes used by neattext for data cleaning.



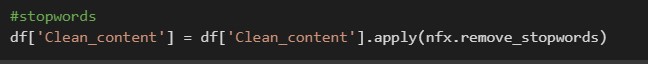
* The output for the methods and attributes is shown below:



* We use the remove\_userhandles method to remove them from our dataset.



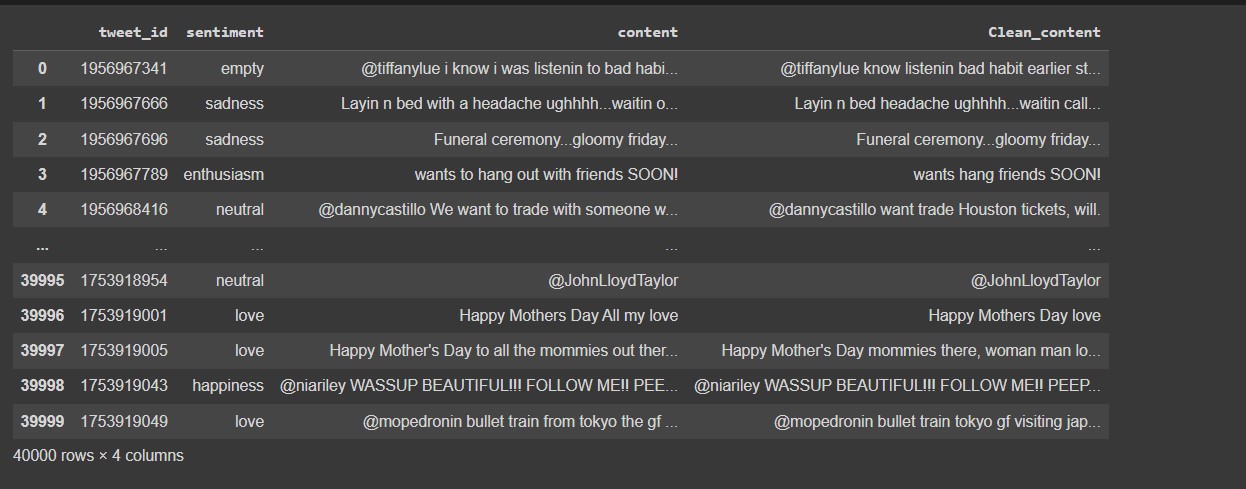
* Removing stopwords:

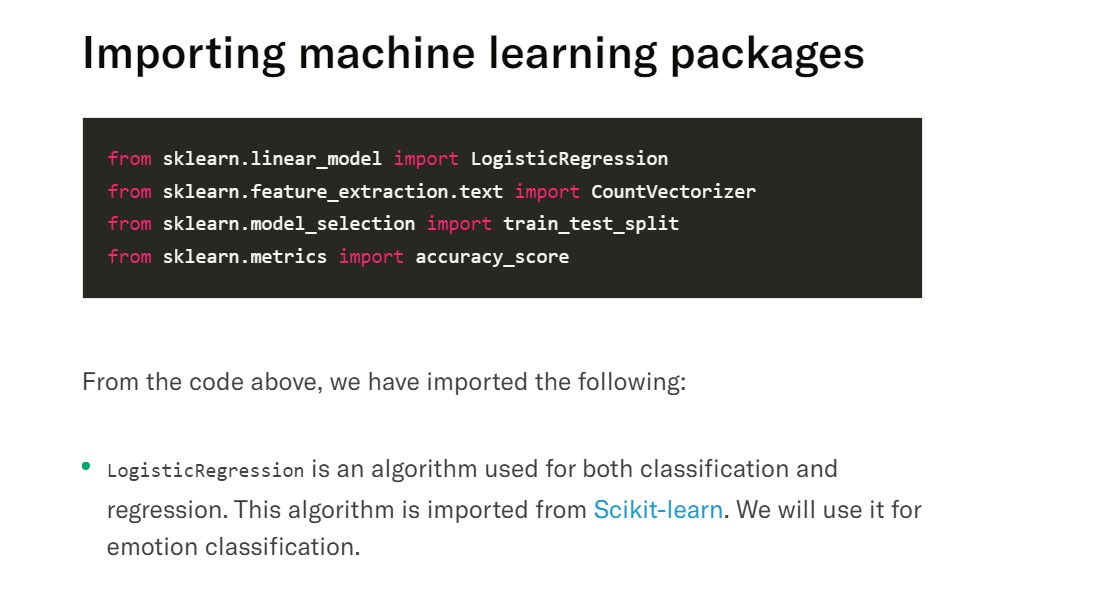


* To get the output of the clean dataset, run this command:



* The output of the dataset after removing user handles and stopwords is shown below:



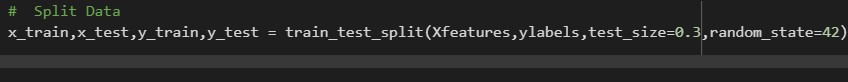


T

* Model features and labels:



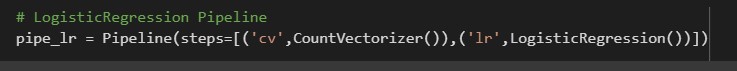
* Dataset splitting:

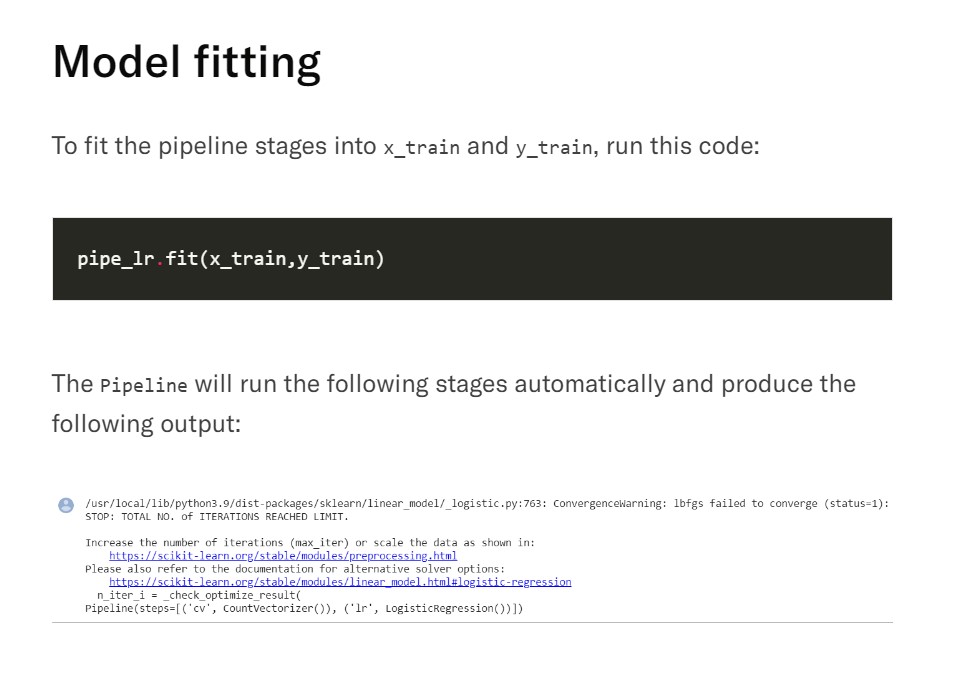


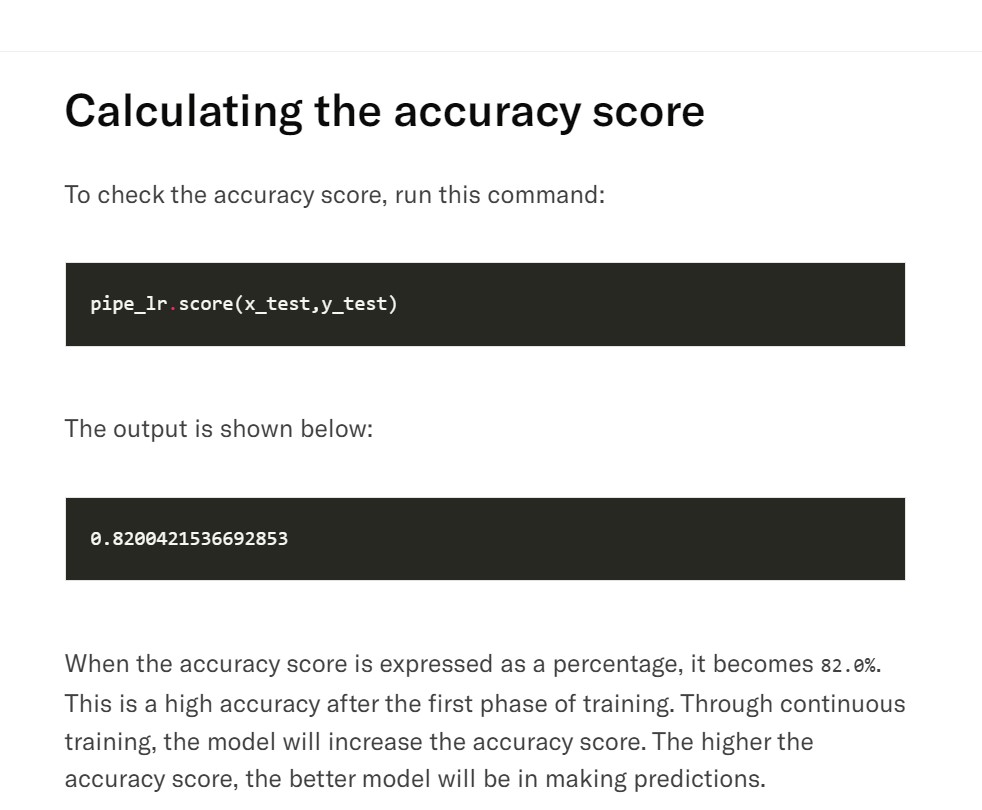
* Pipeline approach:
* The first stage is the CountVectorizer process.
* The second stage is the model training process using the

LogisticRegression algorithm.

The two pipeline stages are initialized as follows:







* The prediction outcome is “joy”. This is the correct prediction. This shows that our model can accurately predict.
* **CONCLUSION**
* Here,we have build an emotion detection model using Neattext and Scikit-learn. We started by cleaning our dataset using Neattext. The dataset has to be in the right shape before it is used for training.
* We used the logistic regression algorithm to build our emotion detection model.we also introduced a concept known as machine learning pipeline.The pipeline approach made our work easier.It automates the countervectorizer process and model building.
* Finally, we predicted the accurate emotion from text by using above mentioned model.