**Vivekanand Education Society’s**

**Institute of Technology**

**(Affiliated to University of Mumbai, Approved by AICTE & Recognized by Govt. of Maharashtra)**

**Department of Information Technology**

AIDS - 2 Lab

Experiment - 02

Aim: To build a Cognitive text based application to understand context for a Customer service application/ Insurance/ Healthcare Application/ Smarter Cities/ Government etc.

| Roll No. | 53 |
| --- | --- |
| Name | SAKSHI SANTOSH PATIL |
| Class | D20B |
| Subject | AIDS - 2 |
| Grade: |  |

**EXPERIMENT - 02**

**AIM:** To build a Cognitive text based application to understand context for a Customer service application/ Insurance/ Healthcare Application/ Smarter Cities/ Government etc.

**DATASET:** <https://drive.google.com/file/d/1bDuFy8mrQUOBTmZh-5KfIwsnHtxNqcnm/view?usp=drivesdk>

**THEORY**:

Sentiment Analysis is a use case of Natural Language Processing (NLP) and comes under the category of text classification. To put it simply, Sentiment Analysis involves classifying a text into various sentiments, such as positive or negative, Happy, Sad or Neutral, etc. Thus, the ultimate goal of sentiment analysis is to decipher the underlying mood, emotion, or sentiment of a text. This is also known as Opinion Mining.

The process involves several steps:

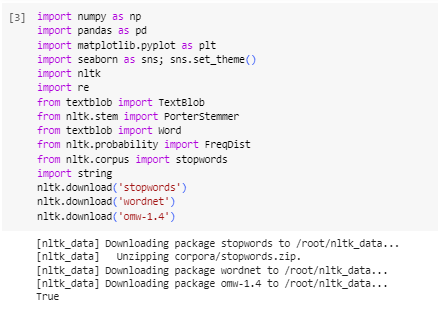
1. **Text Preprocessing:** The text data is cleaned by removing irrelevant information, such as special characters, punctuation, and stopwords.
2. **Tokenization:** The text is divided into individual words or tokens to facilitate analysis.
3. **Feature Extraction:** Relevant features are extracted from the text, such as words, n-grams, or even parts of speech.
4. **Sentiment Classification:** Machine learning algorithms or pre-trained models are used to classify the sentiment of each text instance. This can be achieved through supervised learning, where models are trained on labeled data, or through pre-trained models that have learned sentiment patterns from large datasets.
5. **Post-processing:** The sentiment analysis results may undergo additional processing, such as aggregating sentiment scores or applying threshold rules to classify sentiments as positive, negative, or neutral.
6. **Evaluation:** The performance of the sentiment analysis model is assessed using evaluation metrics, such as accuracy, precision, recall, or F1 score.

Various types of sentiment analysis can be performed, depending on the specific focus and objective of the analysis. Some common types include:

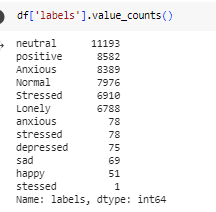
1. **Document-Level Sentiment Analysis:** This type of analysis determines the overall sentiment expressed in a document, such as a review or an article. It aims to classify the entire text as positive, negative, or neutral.
2. **Sentence-Level Sentiment Analysis:** Here, the sentiment of each sentence within a document is analyzed. This type provides a more granular understanding of the sentiment expressed in different text parts.
3. **Aspect-Based Sentiment Analysis:** This approach focuses on identifying and extracting the sentiment associated with specific aspects or entities mentioned in the text. For example, in a product review, the sentiment towards different features of the product (e.g., performance, design, usability) can be analyzed separately.
4. **Entity-Level Sentiment Analysis:** This type of analysis identifies the sentiment expressed towards specific entities or targets mentioned in the text, such as people, companies, or products. It helps understand the sentiment associated with different entities within the same document.
5. **Comparative Sentiment Analysis:** This approach involves comparing the sentiment between different entities or aspects mentioned in the text. It aims to identify the relative sentiment or preferences expressed towards various entities or features.

**IMPLEMENTATION**:

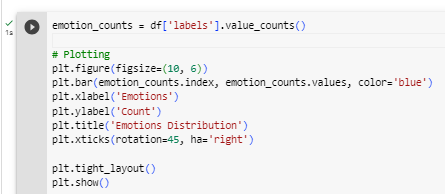
* Import required libraries

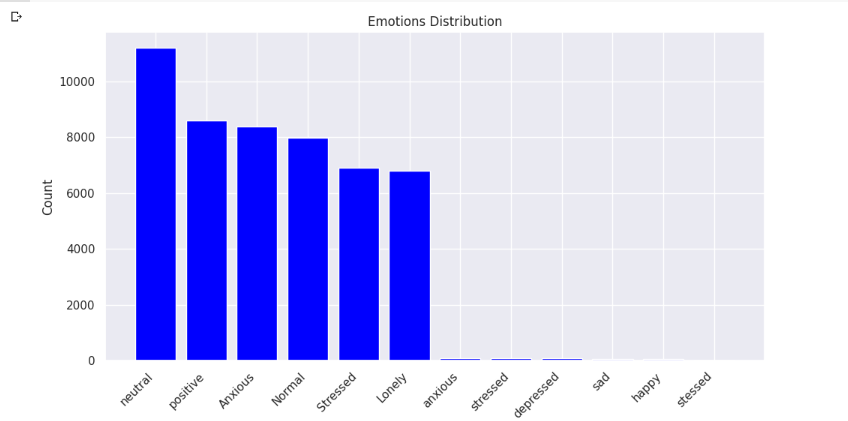


* This code snippet calculates and displays the count of each unique value in the 'labels' column of the DataFrame 'df'.

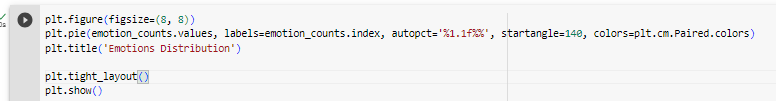


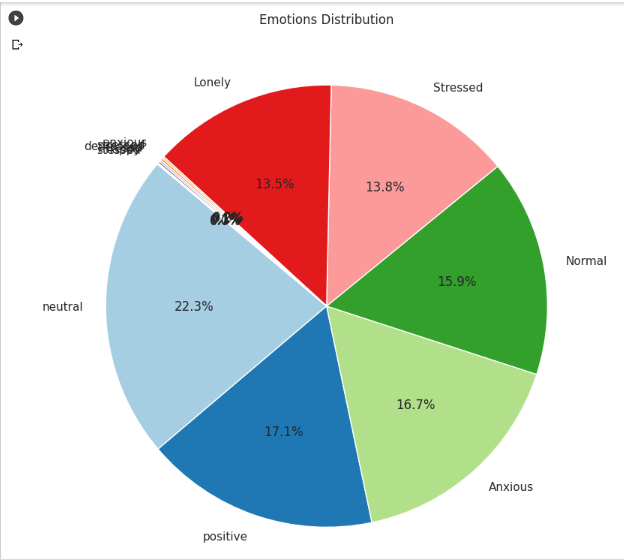
* This code calculates the count of each unique value in the 'labels' column of the DataFrame and then creates a bar plot to visualize the distribution of these counts for different emotions. The x-axis represents emotions, the y-axis represents the count.



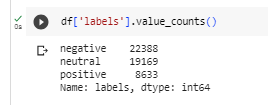


* Pie chart for the same:-

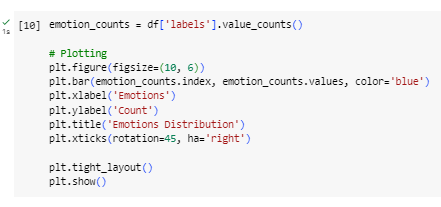


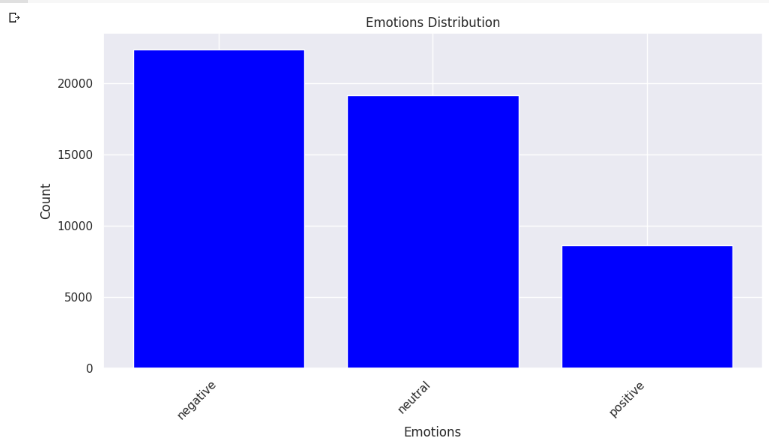


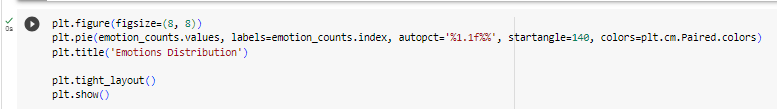
* Now we will divide our parameters into 3 parts: Negative, Positive and Neutral. The label count is:

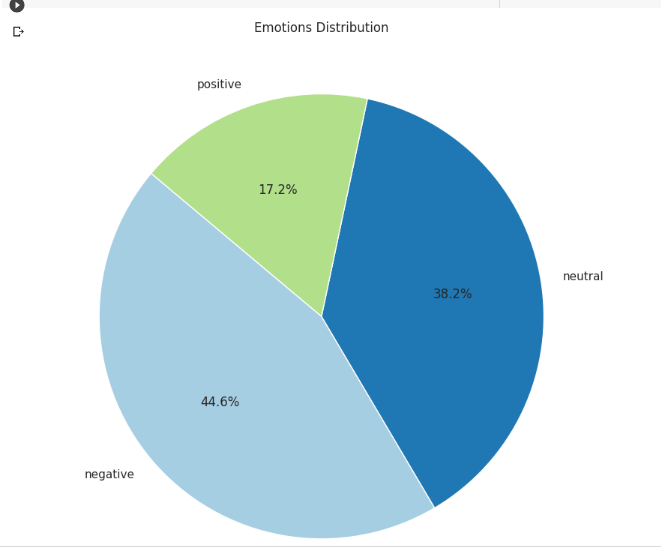


* Plotting a new bar graph and pie chart to show the bifurcation of positive , negative and neutral.









* This code defines a function called `preprocess\_text` that takes a DataFrame (`df`) and a column name (`column\_name`) as inputs. The function performs stemming, removal of stopwords on the specified column:

1. Converts the text to lowercase.

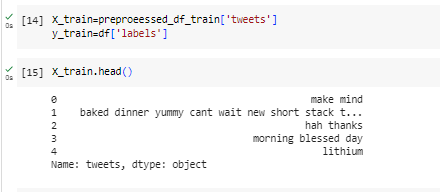
2. Removes punctuation and special characters.

3. Removes English stop words using NLTK's stopwords corpus.

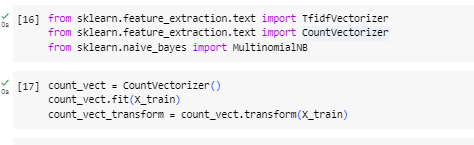
4. Lemmatizes the words using the WordNetLemmatizer from NLTK.

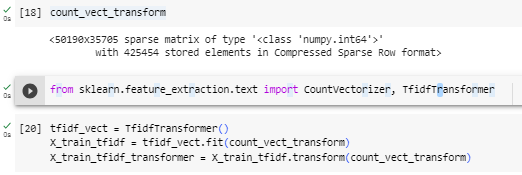


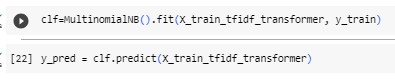
* In the provided code: `X\_train` holds the preprocessed text data from the 'tweets' column of a DataFrame called `preproeessed\_df\_train`. y\_train` contains the emotion or label information from the 'labels' column of the original DataFrame `df`. These variables are set up to be used for training a machine learning model where `X\_train` represents the input text data, and `y\_train` represents the corresponding output labels.



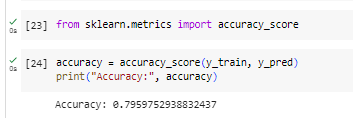
* Now we perform Vectorization and Also perform Classification using multinomial Naive Bayes Algorithm







* The accuracy of our model is 79.59%.

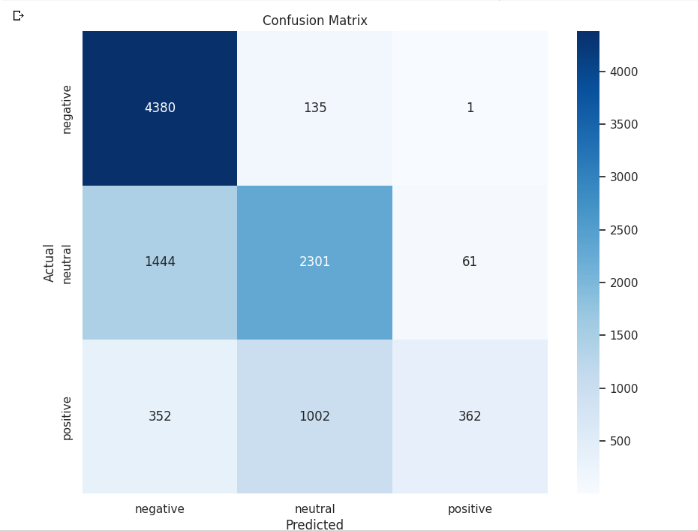


* Confusion matrix:

The output heatmap visually represents how well the classifier's predictions match the actual labels. Diagonal cells represent correct predictions, and off-diagonal cells indicate misclassifications. The x-axis represents predicted labels, while the y-axis represents actual labels. The intensity of color in each cell corresponds to the number of occurrences in the confusion matrix, with darker shades indicating higher counts.







**CONCLUSION:**

Sentiment analysis is an NLP task, which involves classifying texts or parts of texts into a predefined sentiment. We have done sentiment analysis for a tweet dataset. It determines the sentiment from a set of emotions like anxious, stressed, happy, sad, depressed, etc. categorized into Negative, positive and Neutral.