CHAT ANALYSIS

A data science project report of the course "Independent Study in Data Science – (DSC 3263)" presented by

BALASOORIYA, B.P.N. (S/17/317)

HISHAM, M.H.M. (S/17/379)

PUBUDIKA, L.W.P. (S/17/452)

With the supervision of the company,

ZONE24X7 (PVT) LTD
460 NAWALA RD, SRI JAYAWARDENEPURA KOTTE 10107

DEPARTMENT OF STATISTICS & COMPUTER SCIENCE
FACULTY OF SCIENCE
UNIVERSITY OF PERADENIYA
SRI LANKA
2023

DECLARATION

I hereby declare that the Project Summary Report entitled **"Chat Analysis"** is an authentic record of my own work as a requirement of the three-months project under the course of '**Independent study in Data Science (DSC3263)**' during the period from 28/11/2022 to 17/03/2023 for the award of the degree of B.Sc. Honours Study in Data Science from Department of Statistics and Computer Science Faculty of Science University of Peradeniya, under the guidance of Dr. S.P. Abeysundara (**Head of the Department**), Prof. Y.P.R.D. Yapa, Miss. B.R.P.M. Basnayake (**Internal Supervisors)** and Mr. Umair Ramzan (**External Supervisor**).

(Signature of student)

(Name of Student)

(Registration Number)

Date: ____________________

Certified by:

1. **Supervisor (Name):…………………………. Date: …………………..**

   (Signature):………………………..

2. **Head of the Department (Name): ………………………… Date : …………………..**

   (Signature):………………………..

Department Stamp:

ABSTRACT

Sentiment analysis is important because it allows us to analyze and understand people's opinions, emotions, and attitudes towards various products, services, or events. It provides valuable insights into customer preferences, areas for improvement, and more effective marketing strategies. The project will focus on identifying common themes and trends in the feedback and will be limited to a specific time period, determined by the availability of chat data. The project will involve training and fine-tuning a NLTK model using a large dataset of chat data, as well as designing and implementing a user-friendly interface for accessing and visualizing the analysis results. Collecting and importing chat data, cleaning and preprocessing the chat data, identifying important topics and themes in the chat data, performing sentiment analysis and finally visualizing and interpreting the results of the analysis are the methods we used in this project. The objective of this project was to evaluate the effectiveness of the Natural Language Toolkit (NLTK) model in sentiment analysis. Our results showed that the NLTK model achieved an accuracy of over 85% on our test dataset, which is a good indicator of the model's suitability for performing sentiment analysis. Moreover, we ensured that there was no overfitting or underfitting in the model, making it more reliable and robust. Overall, the NLTK model proved to be a highly effective and accurate tool for sentiment analysis. The findings of this project will help to inform the development of more accurate and effective sentiment analysis tools in the future.

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude to all those who have helped me in completing my chat analysis project. First and foremost, I am immensely grateful to my lecturers and supervisors, who have provided me with invaluable guidance and support throughout this project. Their insights and expertise have been critical in shaping the direction of my research and in ensuring that I remained on track throughout the process.

I am also deeply appreciative of my colleagues who have generously shared their time and expertise with me. Their feedback and insights have been instrumental in helping me to refine my research questions and to develop more effective analysis techniques.

Finally, I would like to extend my heartfelt thanks to all the individuals who participated in my research. Without their willingness to share their experiences and insights, this project would not have been possible.

Once again, I would like to express my sincere gratitude to everyone who has helped me in this project. Your support and encouragement have been instrumental in helping me to complete this project, and I am deeply grateful for everything you have done.

Table of Contents

List of Figure

List of Tables

CHAPTER 01

This section consists of the background of the company, organizational structure and overview of the project.

1.1: Background of the Company

Zone24x7 is a technology innovation and development company based in Sri Lanka, with additional offices in the United States and Australia. The company was founded in 2003 by a group of entrepreneurs who aimed to build a technology-focused business that could provide innovative solutions to businesses around the world.

Initially, Zone24x7 focused on providing software development services to customers in the retail and apparel industries. However, over time, the company expanded its offerings to include research and development in emerging technologies such as artificial intelligence, robotics, and Internet of Things (IoT) devices. Today, Zone24x7 specializes in developing innovative solutions in the areas of automation, robotics, and intelligent systems.

One of the key factors that has contributed to Zone24x7's success is its focus on research and development. The company has a dedicated research team that is constantly exploring new technologies and developing innovative solutions that can help businesses improve their operations and increase their efficiency. This focus on innovation has allowed Zone24x7 to stay ahead of the curve and deliver cutting-edge solutions to its customers.

In recent years, Zone24x7 has received several accolades for its innovative solutions and commitment to excellence. In 2017, the company was recognized as one of the Top 20 IoT Solution Providers by CIOReview. In 2018, Zone24x7 was named one of the Top 10 Robotics Solution Providers by Manufacturing Technology Insights.

Overall, Zone24x7 has a strong background in technology innovation and development, and a commitment to quality and customer satisfaction. These factors have helped the company build a strong reputation in the industry and position itself as a leader in emerging technologies such as AI, robotics, and IoT.

1.2: Organizational Structure

Zone24x7 has a functional organizational structure with a focus on cross-functional collaboration. The company is divided into several departments, each responsible for a specific area of expertise, such as software development, research and development, and quality assurance. The key departments at Zone24x7 include:

- Engineering: This department is responsible for software development, testing, and deployment of the company's solutions.

- Research and Development: This department is responsible for exploring and developing emerging technologies such as artificial intelligence, machine learning, and robotics.

- Quality Assurance: This department ensures that all products and solutions meet the company's high standards of quality and reliability.

- Sales and Marketing: This department is responsible for promoting and selling the company's products and services to customers.

- Human Resources: This department is responsible for managing the company's talent acquisition, employee engagement, and professional development programs.

In addition to these key departments, Zone24x7 has a project management office (PMO) that oversees the delivery of projects and ensures that they are completed on time, within budget, and to the client's satisfaction. The company also has a dedicated innovation lab that focuses on exploring new technologies and developing innovative solutions for clients.

Overall, Zone24x7's functional structure allows the company to efficiently and effectively deliver high-quality products and solutions to its clients. The focus on cross-functional collaboration and innovation ensures that the company stays at the forefront of emerging technologies and is able to meet the evolving needs of its clients.

1.3: Overview of the Project

The sentiment analysis project using the NLTK library aims to develop a model that can accurately classify the sentiment of textual data as positive, negative, or neutral. NLTK is a popular Python library for NLP, which provides various tools and resources for text analysis.

The project will involve various stages, including data collection and preprocessing, followed by the application of a lexicon-based approach and the use of a prebuilt Vader model, and finally the evaluation of performance of the model.

In the data collection and preprocessing stage, we will gather a large corpus of textual data from different sources such as social media, customer feedback, and product reviews. We will preprocess the text data using NLTK tools such as tokenization, stop-word removal, and part-of-speech tagging to convert the raw text into a suitable format for analysis.

In the application of a lexicon-based approach and the use of a prebuilt Vader model, the text data is analyzed to determine the sentiment expressed in the text. The lexicon-based approach involves identifying the sentiment of the words in the text by referring to a pre-existing sentiment lexicon. This is typically done by comparing the words in the text to the words in the lexicon and assigning a sentiment score to each word based on the score associated with that word in the lexicon. The sentiment scores of the individual words are then aggregated to provide an overall sentiment score for the text.

The Vader model, on the other hand, uses a set of rules to determine the sentiment expressed in the text. These rules take into account a variety of linguistic features such as the use of punctuation, capitalization, and emoticons, to identify the sentiment of the text. The Vader model is specifically designed to handle texts that contain sarcasm, irony, and other forms of figurative language.

In the evaluation stage, we will assess the performance of the sentiment analysis model using NLTK's evaluation tools such as accuracy, precision, recall, and F1-score. We will also perform error analysis using NLTK's visualization tools to identify the strengths and weaknesses of the model and make improvements as necessary.

Ultimately, the sentiment analysis model developed using the NLTK library will be useful for various applications such as social media monitoring, customer feedback analysis, and product review analysis. The project aims to develop a robust and accurate sentiment analysis model that can provide valuable insights into the opinions, attitudes, and emotions of customers, users, and audience.

CHAPTER 02

This section consists of introduction to the project with a focus on understanding the importance of sentiment analysis and its potential applications.

2.1: Introduction to the project

Sentiment analysis is a process of analyzing textual data to identify the emotions, opinions, and attitudes expressed in the text. It has become increasingly important for businesses and organizations to understand the sentiment of their customers, users, and audience to make data-driven decisions. The NLTK (Natural Language Toolkit) is a widely used Python library for NLP that provides various tools and resources for text analysis.

In this sentiment analysis project, we will be using the NLTK library to develop a model that can accurately classify text data as positive, negative, or neutral based on the sentiment expressed in the text. We will be exploring various NLP techniques such as tokenization, stop-word removal, and part-of-speech tagging to preprocess the text data and feature engineering techniques such as frequency distribution and n-grams to represent the text data as numerical features.

We have used various evaluation metrics such as confusion matrices, accuracy, precision, recall, F1-score and support to assess the performance of the model. Ultimately, the goal of this sentiment analysis project using the NLTK library is to develop a robust and accurate model that can classify text data based on the sentiment expressed in the text, which can be used for various applications such as social media monitoring, customer feedback analysis, and product review analysis.

2.2: Literature Review

This literature review aims to explore the current state of sentiment analysis research related to Sri Lanka and other countries, with a focus on the use of the NLTK library.

3.2.1: Related to Sri Lanka

The research article "Sentiment Analysis of Tweets Regarding the Sri Lankan Crisis using Automatic Coding in ATLAS.ti 22" (Musfira, 2022) explores the use of automatic coding in ATLAS.ti 22 for sentiment analysis of tweets related to the Sri Lankan crisis. The study focuses on the sentiment of tweets related to the Sri Lankan crisis that is considered the worst financial crisis in its history. The researchers collected tweets containing specific keywords related to the “Sri Lankan crisis” and used ATLAS.ti 22 to analyze the sentiment of the tweets. The findings of the study indicate that the automatic coding method using ATLAS.ti 22 is effective in analyzing sentiment in tweets related to the Sri Lankan crisis. The study found that the majority of the tweets (57%) had a negative sentiment, while 33.3% of the tweets had a neutral sentiment, and 9.6% of the tweets had a positive sentiment. The study also identified the most frequent keywords used in the tweets and the dominant themes discussed. Overall, the study demonstrates the effectiveness of ATLAS.ti 22 for sentiment analysis and highlights its potential use in understanding public opinion during crises.

In the research article "Sentiment Analysis of Tweets to predict Sri Lankan Election Results using Supervised Learning Techniques" (Gunasiri, 2021) the author explores the application of sentiment analysis to predict the results of the Sri Lankan presidential election. The study uses a dataset of tweets related to the 2019 presidential election in Sri Lanka and analyzes the sentiment of the tweets using supervised learning techniques. The author applied three classification algorithms, Naive Bayes, Decision Tree, and Support Vector Machine, to classify the tweets into positive, negative, and neutral sentiment categories. The study also compared the performance of the three algorithms and evaluated their accuracy in predicting the election results. The findings of the study suggest that the Naive Bayes algorithm performed the best. The study also identified the most common topics discussed in the tweets related to the election and their sentiment. The study demonstrates the potential of sentiment analysis in predicting election outcomes and highlights the usefulness of social media data in analyzing public opinion during political campaigns.

3.2.2: Related to other countries

The research article "Study of Sentiment of Governor's Election Opinion in 2018" (Agung Eddy Suryo Saputro, 2018) discusses the sentiment analysis of public opinion regarding the 2018 governor's election in Indonesia. The study utilizes the Natural Language Toolkit (NLTK) and a machine learning algorithm to analyze Twitter data. The authors collected 21,372 tweets containing the keyword "pilkada" (local election) and filtered out irrelevant data. The preprocessed data was then classified using a Naive Bayes algorithm and analyzed for sentiment. The results of the study show that the majority of the tweets (35%) were positive, while 34% were neutral, and 31% were negative. The study also found that the majority of the positive tweets were related to the candidates' achievements and capabilities, while negative tweets were related to issues such as corruption and inefficiency. Overall, the study demonstrates the effectiveness of sentiment analysis in understanding public opinion during an election and highlights the potential use of Twitter data for political campaigns and decision-making processes.

Kaur et al. (2022) conducted a study on sentiment analysis of electricity-related tweets on Twitter from the United Kingdom and India. The purpose of the study was to analyze people's reactions to increases in electricity bills expressed on Twitter's social media platform. The authors collected a total of 8,731 tweets using Python programming in Jupyter Notebook. The authors utilized several machine learning algorithms, including Naive Bayes, Decision Tree, Random Forest, and Logistic Regression, to analyze the tweets. The performance of these methods was measured using F-Score, Accuracy, Precision, and Recall. The authors found that the Random Forest model had the best accuracy level of 0.84 compared to the other methods. The study provides important insights into how people from different cultures react to increases in electricity bills and highlights the potential of sentiment analysis for examining public opinions and attitudes towards important issues. These findings can be useful for policymakers and utility companies in understanding public sentiment towards electricity and for improving communication strategies to address concerns and grievances.

During the COVID-19 pandemic, Pano et al. (2020) conducted a study on the sentiment analysis of Bitcoin tweets. The main aim was to analyze how sentiment scores of Twitter text correlate with Bitcoin prices during the pandemic. The researchers collected a total of 4,169,709 tweets between 8:47 AM, 22 May and 11:59 PM, 10 July. The study used various methods to preprocess text for VADER scoring and tested them on truncated and full-length tweets. The results revealed that splitting sentences, removing Twitter-specific tags, or their combination generally improve the correlation of sentiment scores and volume polarity scores with Bitcoin prices. However, the correlation between prices and sentiment scores was stronger over shorter periods of time. In summary, this study provides crucial insights into the relationship between Twitter sentiment and Bitcoin prices during a crisis period, emphasizing the significance of text preprocessing techniques for accurate sentiment analysis.

The study conducted by Bello et al. (2023) aimed to compare the performances of various natural language processing (NLP) techniques in text classification, with a specific focus on sentiment analysis of tweets using a BERT framework. To achieve this, the researchers used six datasets comprising of tweets collected from Kaggle, which had a total of 212,661 rows and 2 columns. The study observed the model's learning rates of BERT with different NLP variants such as CNN, RNN, and BiLSTM. The findings of the study indicated that a combination of BERT with CNN, RNN, and BiLSTM performed exceptionally well in terms of accuracy rate, precision rate, recall rate, and F1-score. This performance was significantly better than when BERT was used with Word2vec or with no variant at all. Overall, the study demonstrates the effectiveness of using a BERT framework with multiple NLP variants for sentiment analysis of tweets.

In the study (Rakhmanov, 2020) a large dataset comprising of over 52,000 comments was utilized to perform a comparative analysis of various vectorization and classification techniques. The results of the experiment revealed that the Random Forest classifier was the most efficient and optimal model for classification, achieving a state-of-the-art prediction accuracy of 97% for 3-class classification. Furthermore, to enhance the diversity of the comments, a 5-class dataset was created, and the experiment yielded an efficient classification model with an accuracy of 92%. The Tf-Idf vectorization technique outperformed the Count (Binary) vectorization approach. The study's findings provide valuable insights into the performance of different vectorization and classification techniques in sentiment analysis. The high accuracy rates obtained by the Random Forest classifier indicate its potential as a powerful tool for sentiment analysis in large datasets. Additionally, the experiment's ability to achieve high accuracy rates with a 5-class dataset is a promising result, as it shows that sentiment analysis can be applied effectively to a wide range of comment classifications. Overall, the study's outcomes highlight the importance of selecting appropriate techniques and approaches for specific applications in sentiment analysis.

CHAPTER 03

This section includes a description of the methodology used in the sentiment analysis project, which involves several steps such as data preprocessing, feature extraction using NLTK, machine learning model training, and model evaluation.

3.1: Project problem

Analyze the sentiment of customer support data on twitter. Here we used a Vader pre built model to analyze the sentiment of tweets.

The problem being addressed in this project is the analysis of customer support data on Twitter using a pre-built sentiment analysis model called Vader. With the increasing prevalence of social media as a means of communication between customers and businesses, it is important to be able to quickly and accurately analyze the sentiment of customer support interactions on these platforms. By utilizing a pre-built model such as Vader, this project aims to provide a solution that is both efficient and effective in analyzing customer support data on Twitter, allowing businesses to better understand and respond to their customers' needs and concerns.

3.2: Collect the data

We collected Customer Support on Twitter dataset from Kaggle.

Kaggle is an online platform for data scientists, machine learning engineers, and data enthusiasts to participate in data science competitions, collaborate on projects, and develop their data science skills. It was founded in 2010 and was later acquired by Google in 2017.On Kaggle, users can access a wide range of datasets.

Customer Support on Twitter dataset is a large, modern corpus of tweets and replies to aid innovation in natural language understanding and conversational models, and for study of modern customer support practices and impact.

3.2.1: Description of data set

The Customer Support on Twitter dataset contains information about 2.8 million tweets from the biggest brands on Twitter..The dataset covers the period from January 2013 to December 2017. Each tweet is described by several variables

3.2.2: Description of variables

tweet_id : A unique, anonymized ID for the Tweet. Referenced by response_tweet_id and in_response_to_tweet_id.

Author_id : A unique, anonymized user ID. @s in the dataset have been replaced with their associated anonymized user ID.

Inbound : Whether the tweet is "inbound" to a company doing customer support on Twitter. This feature is useful when re-organizing data for training conversational models.

Created_at : Date and time when the tweet was sent.

Text : Tweet content. Sensitive information like phone numbers and email addresses are replaced with mask values like email.

Response_tweet_id : IDs of tweets that are responses to this tweet, comma-separated.

In_response_to_tweet_id : ID of the tweet this tweet is in response to,

3.3: Preprocess the data

3.5.1: Text mining

Text mining is the process of using computational and statistical techniques to extract useful insights and knowledge from large amounts of unstructured textual data. Here we derived meaningful information from text data.

- Before text normalization, the data set was checked for missing data. Pandas is.na() function was used to detect missing values.
- Unwanted columns were dropped.
- Time column was separated into Date, Time and DayName columns to do exploratory analysis. Pandas to_datetime() function was used to Convert argument to datetime.
- The columns were renamed as below.
  - **tweet_id** was renamed as **Tweet_ID**.
  - **created_at** was separated to three columns and renamed as Date and Time and DayName.
  - **author_id** was renamed as **Author_ID**.
  - Text was renamed as **Messege**.
  - **Response_tweet_id** was renamed as **Response_Id.**
- The data set was sorted according to Date and Time.
- The date range of Data set was found. min() and max() python built in functions were used.

3.5.2: Text Normalization

Text Normalization is the process of transforming text into a canonical, standardized form that can be easily processed and analyzed by computer programs. It involves converting text data into a consistent and uniform format by removing or replacing any variations, inconsistencies, or redundancies in the text.

3.3.2.1: Text cleaning

Refers to the process of removing or correcting unwanted or irrelevant information from text. Here are the things that we have done under text cleaning.

- URLs, stop words, punctuation marks, numbers were removed.
- Case normalization: All the texts were Standardized by making it all lower case.
- Replace emojis with words.

3.3.2.2: Tokenization

Tokenization is the process of splitting text into individual words or tokens.

In this project word_tokenize(x) in the NLTK library was used to tokenize the words in x. This is the method that is applied to each element of the 'Message' column.

3.4: Perform EDA

In the exploratory data analysis (EDA) phase, we aimed to gain a better understanding of the preprocessed message data and identify any patterns or insights that could inform our analysis. To begin, we compared the message column before and after preprocessing to assess the impact of our preprocessing techniques on the data. We used visualizations and summary statistics to compare the length and content of the messages before and after preprocessing. Next, we created two bar charts to visualize the number of messages that were sent and received by customer services. These charts allowed us to identify any imbalances or anomalies in the data, and to gain an overall understanding of the distribution of messages. Also, we identified the most active dates by tabulating the number of messages by date. By doing so, we were able to identify any peaks or trends in message activity and gain insights into the overall volume of messages received over time.

We aimed to identify which days of the week were more active in terms of message volume. To achieve this, we tabulated the DayName column, which contains the name of the day of the week for each message, and the number of messages. This tabulation allowed us to identify the most active days of the week for customer service message volume, and to gain insights into any trends or patterns in message activity over the course of the week. To further visualize this data, we created a pie chart to obtain insights as a percentage. The pie chart allowed us to easily compare the relative proportions of message volume on each day of the week and to identify any significant differences in activity levels.

In the final step of our analysis, we aimed to identify the most frequently used words by each author in the message dataset. To achieve this, we employed a text analysis technique to extract and count the frequency of individual words from the preprocessed message data. By tabulating the most frequently used words for each author, we were able to gain insights into the specific language and communication style of each author, and to identify any differences or similarities in the language used by different authors.

By conducting this EDA phase, we were able to gain a better understanding of the preprocessed message data and identify patterns or insights that could inform our analysis. These insights will be used to inform the next phase of our analysis, which is the sentiment analysis part where we will explore more specific research questions and hypotheses.

3.5: Sentiment Analysis: Using NLTK Library

NLTK (Natural Language Toolkit) is a Python library for natural language processing that provides a wide range of tools and resources for working with human language data.

NLTK includes various pre-built models. VADER (Valence Aware Dictionary and sEntiment Reasoner) is considered a prebuilt or pre-defined model. A prebuilt model like VADER is designed to be used as is, without any additional training or customization.

The VADER (Valence Aware Dictionary and sEntiment Reasoner) model represents text using a lexicon-based approach, which means it relies on pre-built dictionaries of words and their associated sentiment scores.

That was specifically built for analyzing sentiment from social media resources with more than 9000 lexical features(words) and their intensities to score the sentiment of a piece of text. It provides a sentiment score for each sentence in the text, ranging from -1 (negative) to +1 (positive). It also takes into account the grammatical and syntactical structure of the text.

In VADER, each word in a piece of text is assigned a polarity score ranging from -1 to +1, where -1 indicates the most negative sentiment, +1 indicates the most positive sentiment, and 0 indicates a neutral sentiment. VADER also considers the context of the text by taking into account the effect of words' position and grammatical structure.

After assigning polarity scores to each word and bigram in the text, VADER calculates a compound score by taking the sum of all the polarity scores and normalizing it to a range between -1 and +1. The compound score is an overall measure of the sentiment of the text, where values closer to +1 indicate more positive sentiment and values closer to -1 indicate more negative sentiment.

3.5.1: Separating data set

The data set was separated according to the year 2013, 2014 and 2015.

Year 2013: **86 tweets.** Dataset name: **pasindu**
Year 2014: **199 tweets.** Dataset name: **hisham**
Year 2015: **100 tweets.** Dataset name: **pawani**

We extracted another 100 tweets for the testing part.

Test data set Year 2012: 100 tweets**.** Dataset name: **testdata**

3.5.2: Adding sentiments manually

All the tweets were read manually and added their sentiments. Similarly, added sentiments for the test dataset.

New column was added to the data sets.

Column name: manual_tag

3.5.3: Generating sentiments using NLTK

3.5.3.1: Generating sentiment score

Then we generated sentiment scores for all tweets in the above data sets using nltk library.

1. Import the SentimentIntensityAnalyzer class from the nltk.sentiment.vader module

2. Use the polarity_scores() method of the SentimentIntensityAnalyzer class to obtain the sentiment scores for a piece of text. The method returns a dictionary with four keys: 'neg' (negative score), 'neu' (neutral score), 'pos' (positive score), and 'compound' (overall score).

3. The 'compound' score is often used as a measure of overall sentiment. It ranges from -1 (extremely negative) to +1 (extremely positive).

3.5.3.2: Classify sentiments (positive, negative and neutral) according to sentiment score

- We used range to classify the sentiment of the tweets as positive, negative, or neutral.
- We selected different ranges for classifying sentiments according to the compound value that was generated by the polarity_score() method.
- Different ranges were compared in order to find the accurate range.

- Below table displays the different ranges that we used to classify sentiments of each tweet according to their compound value.

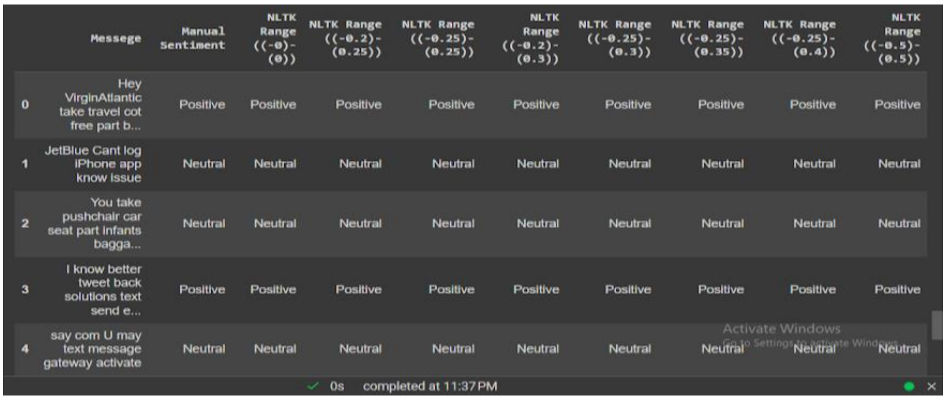
Table 3.5.3.2.1: Different ranges to classify sentiment according to compound value

Rangename	**Negative**	**Neutral**	**Positive**	**Column name**
Range	Compound<0	compound=0	Compound>0	**NLTK Range (0)**
Range 0.5	Compound<-0.5	-0.5 <=compound<=0.5	Compound>0.5	**NLTK Range (0.5)**
Range 0.25	Compound<-0.25	-0.25 <=compound<=0.25	Compound>0.25	**NLTK Range (0.25)**
Range A	Compound<-0.25	-0.25 <=compound<=0.3	Compound>0.3	**NLTK_A**
Range B	Compound<-0.25	-0.25 <=compound<=0.35	Compound>0.35	**NLTK_B**
Range C	Compound<-0.25	-0.25 <=compound<=0.4	Compound>0.4	**NLTK_C**
Range D	Compound<-0.20	-0.20 <=compound<=0.25	Compound>0.25	**NLTK_D**
Range E	Compound<-0.20	-0.20 <=compound<=0.3	Compound>0.3	**NLTK_E**

Sentiments were generated for all three datasets according to the given seven different ranges. New columns were added to all three datasets to store sentiments according to the above ranges.

Column Names: **NLTK Range (0), NLTK Range (0.5), NLTK Range (0.25), NLTK_A, NLTK_B, NLTK_C, NLTK_D, NLTK_E**

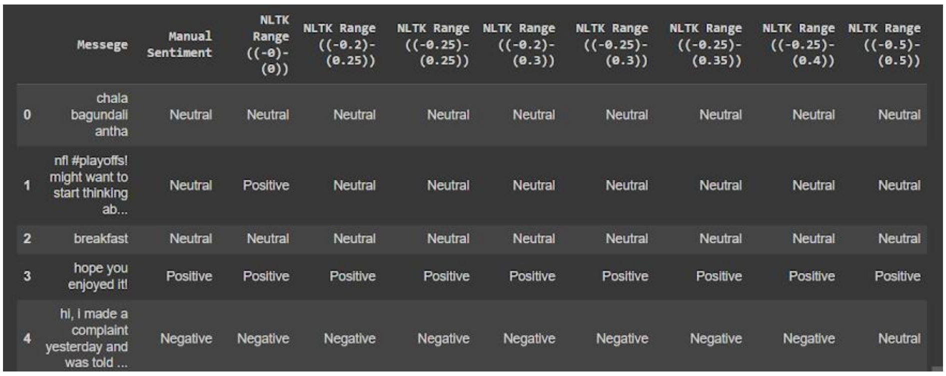
Now we have three different data sets(2013 data set, 2014 data set ,2015 data set) with seven new columns which contain sentiment according to the given range.



	Messege	Manual Sentiment	NLTK Range ((-0)-(0))	NLTK Range ((-0.2)-(0.25))	NLTK Range ((-0.25)-(0.25))	NLTK Range ((-0.2)-(0.3))	NLTK Range ((-0.25)-(0.3))	NLTK Range ((-0.25)-(0.35))	NLTK Range ((-0.25)-(0.4))	NLTK Range ((-0.5)-(0.5))
0	Hey VirginAtlantic take travel cot free part b...	Positive	Positive	Positive	Positive	Positive	Positive	Positive	Positive	Positive
1	JetBlue Cant log iPhone app know issue	Neutral	Neutral	Neutral	Neutral	Neutral	Neutral	Neutral	Neutral	Neutral
2	You take pushchair car seat part infants bagga...	Neutral	Neutral	Neutral	Neutral	Neutral	Neutral	Neutral	Neutral	Neutral
3	I know better tweet back solutions text send e...	Positive	Positive	Positive	Positive	Positive	Positive	Positive	Positive	Positive
4	say com U may text message gateway activate	Neutral	Neutral	Neutral	Neutral	Neutral	Neutral	Neutral	Neutral	Neutral

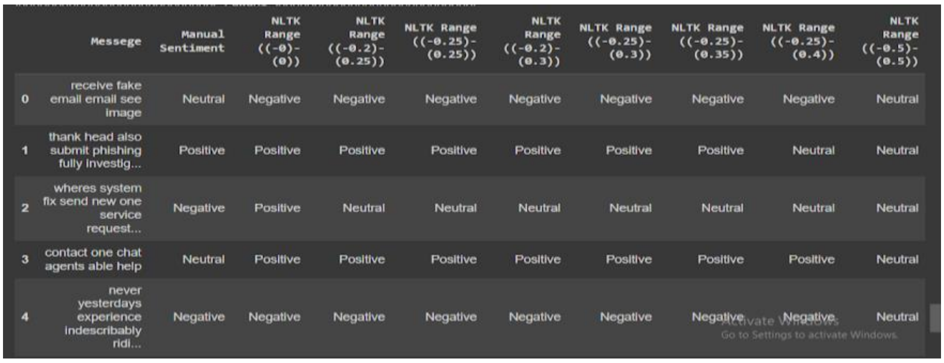
✓ 0s completed at 11:37 PM

Figure 3.5.3.2.1: First 5 data of 2013 dataset



	Messege	Manual Sentiment	NLTK Range ((-0)-(0))	NLTK Range ((-0.2)-(0.25))	NLTK Range ((-0.25)-(0.25))	NLTK Range ((-0.2)-(0.3))	NLTK Range ((-0.25)-(0.3))	NLTK Range ((-0.25)-(0.35))	NLTK Range ((-0.25)-(0.4))	NLTK Range ((-0.5)-(0.5))
0	chala bagundali antha	Neutral	Neutral	Neutral	Neutral	Neutral	Neutral	Neutral	Neutral	Neutral
1	nfl #playoffs! might want to start thinking ab...	Neutral	Positive	Neutral	Neutral	Neutral	Neutral	Neutral	Neutral	Neutral
2	breakfast	Neutral	Neutral	Neutral	Neutral	Neutral	Neutral	Neutral	Neutral	Neutral
3	hope you enjoyed it!	Positive	Positive	Positive	Positive	Positive	Positive	Positive	Positive	Positive
4	hi, i made a complaint yesterday and was told ...	Negative	Negative	Negative	Negative	Negative	Negative	Negative	Negative	Neutral

Figure 3.5.3.2.2: First 5 data of 2014 dataset



	Messege	Manual Sentiment	NLTK Range ((-0)-(0))	NLTK Range ((-0.2)-(0.25))	NLTK Range ((-0.25)-(0.25))	NLTK Range ((-0.2)-(0.3))	NLTK Range ((-0.25)-(0.3))	NLTK Range ((-0.25)-(0.35))	NLTK Range ((-0.25)-(0.4))	NLTK Range ((-0.5)-(0.5))
0	receive fake email email see image	Neutral	Negative	Negative	Negative	Negative	Negative	Negative	Negative	Neutral
1	thank head also submit phishing fully investig...	Positive	Positive	Positive	Positive	Positive	Positive	Positive	Neutral	Neutral
2	wheres system fix send new one service request...	Negative	Positive	Neutral	Neutral	Neutral	Neutral	Neutral	Neutral	Neutral
3	contact one chat agents able help	Neutral	Positive	Positive	Positive	Positive	Positive	Positive	Positive	Neutral
4	never yesterdays experience indescribably ridi...	Negative	Negative	Negative	Negative	Negative	Negative	Negative	Negative	Neutral

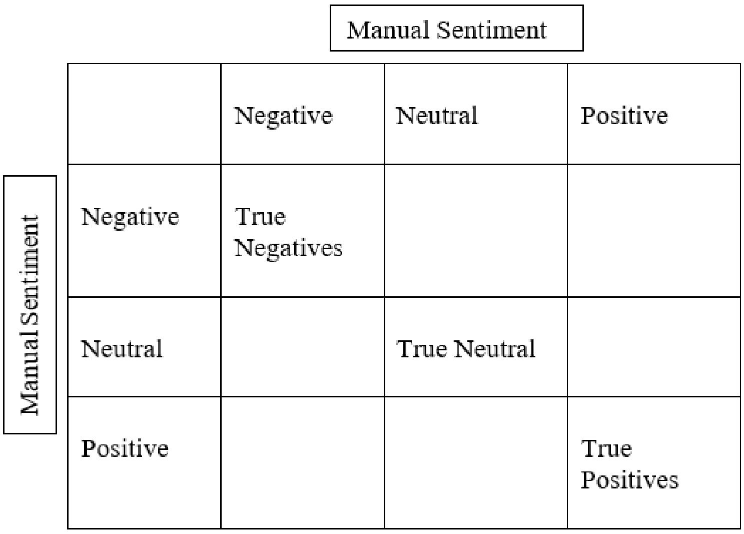
Figure 3.5.3.2.3: First 5 data of 2015 dataset

3.6: Evaluate the model

After generating sentiments according to the given ranges to all three data sets, we used a confusion matrix and precision, recall, f1 scores to evaluate the model.

3.6.1: Confusion Matrix

A confusion matrix can be used to evaluate the accuracy of sentiment analysis models that generate positive, negative, and neutral sentiments using the Natural Language Toolkit (NLTK). The matrix compares the predicted labels with the actual labels of the data and computes the number of true positives, false positives, true negatives, and false negatives for each sentiment category. By summing up the diagonal values of the matrix, which represent the correct predictions, and dividing them by the total number of samples, we can compute the overall accuracy of the model. The accuracy score obtained from the confusion matrix can be used to assess the performance of the sentiment analysis model and identify areas for improvement.



		Manual Sentiment		
		Negative	Neutral	Positive
Manual Sentiment	Negative	True Negatives		
	Neutral		True Neutral	
	Positive			True Positives

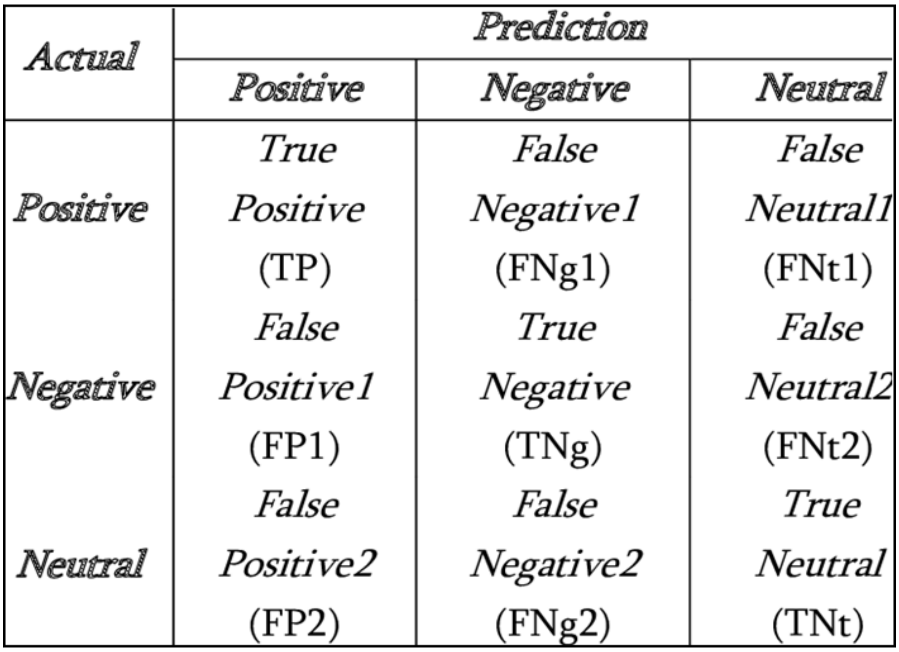
Figure 3.6.1.4: Confusion matrix for sentiments

To calculate the accuracy from a confusion matrix for three sentiment classes (positive, negative, and neutral), you can use the following formula:

Accuracy = (TP_pos + TP_neg + TP_neu) / (TP_pos + TP_neg + TP_neu + FP_pos + FP_neg + FP_neu + FN_pos + FN_neg + FN_neu)

3.6.2: Precision, Recall, and F1 scores

Precision, Recall, and F1 are metrics used to evaluate the performance of a classification model, such as sentiment analysis, that assigns labels to input data points. In sentiment analysis, the labels are usually positive, neutral, or negative



Actual	Prediction		
	Positive	Negative	Neutral
Positive	True Positive (TP)	False Negative1 (FNg1)	False Neutral1 (FNt1)
Negative	False Positive1 (FP1)	True Negative (TNg)	False Neutral2 (FNt2)
Neutral	False Positive2 (FP2)	False Negative2 (FNg2)	True Neutral (TNt)

Figure 3.6.2.5: Confusion matrix for three sentiment classes

To our labeled datasets with sentiment labels, we calculated the precision, recall, and F1 score for each sentiment label as follows:
For Positive Sentiments,

- True Positives (TP): the number of instances that were correctly classified as positive.·
- False Positives (FP): the number of instances that were incorrectly classified as positive.·
- False Negatives (FN): the number of instances that were incorrectly classified as neutral or negative but should have been classified as positive.

According to the above table,

Precision (P) for positive sentiment: TP / (TP + FP1+FP2)
Recall (R) for positive sentiment: TP / (TP + FNg1+FNt1)
F1 score (F1) for positive sentiment: 2 * P * R / (P + R)

For Neutral Sentiments,

- True Positives (TP): the number of instances that were correctly classified as neutral.
- False Positives (FP): the number of instances that were incorrectly classified as neutral.
- False Negatives (FN): the number of instances that were incorrectly classified as positive or negative, but should have been classified as neutral.

According to the above table,

Precision (P) for neutral sentiment: TNt / (TNt + FNt1+FNt2)
Recall (R) for neutral sentiment: TNt / (TNt + FNg2+FP2)
F1 score (F1) for neutral sentiment: 2 * P * R / (P + R)

For Negative Sentiments,

- True Positives (TP): the number of instances that were correctly classified as negative.
- False Positives (FP): the number of instances that were incorrectly classified as negative.
- False Negatives (FN): the number of instances that were incorrectly classified as positive or neutral, but should have been classified as negative.

According to the above table,

Precision (P) for negative sentiment: TNg / (TNg + FNg1+ FNg2)
Recall (R) for negative sentiment: TNg / (TNg + FNt2+FP1)
F1 score (F1) for negative sentiment: 2 * P * R / (P + R)

3.6.3: Using evaluation metrics

We made a confusion matrix according to the result of seven different ranges for three data sets separately.

By using the confusion matrix, we calculated the accuracy for the nltk library according to seven different ranges for three data sets.(R)

Then we combined all three data sets together and found accuracy for that combined dataset according to seven different ranges.

Combined data set name: **merged_df**

Table 3.6.3.2: The accuracy for nltk library using seven different ranges for three data sets

	NLTK **Range(0)**	**NLTK** **Range(0.5)**	**NLTK** **Range(0.25)**	**NLTK_A** **--0.25<c<0.3**	**NLTK_B** **--0.25<c<0.35**	**NLTK_C** **--0.25<c<0.4**	**NLTK_D** **--0.20<c<0.25**	**NLTK_E** **--0.20<c<0.3**
2013 dataset **Pasindu**	**84.88%**	**68.34%**	**81.4%**	**81.4%**	**80.23%**	**79.07%**	**81.4%**	**81.4%**
2014 dataset **Hisham**	**70.35%**	**78%**	**83.42%**	**84.92%**	**87.94%**	**87.44%**	**80.9%**	**82.41%**
2015 dataset **Pawani**	**73.0%**	**65.12%**	**82.0%**	**82.0%**	**86.0%**	**86.0%**	**81.0%**	**81.0%**
Combined **data set**	**74.29%**	**70.13%**	**82.6%**	**83.38%**	**85.71%**	**85.19%**	**81.04%**	**81.82%**

According to the above accuracy we selected the most accurate two ranges among all seven ranges. We selected range B and range C according to their accuracy.

We calculated precision, recall, and F1 score for each sentiment label (positive, negative, neutral) for selected two different ranges in all three data sets.

Table 3.6.3.3: Calculated precision, recall, F1 score for years separately for ranges B and C

Range B

2013 DATA SET

```
--------------- Precision/Recall/F1 Score ---------------
              precision          recall  f1-score   support

              neg       0.80       0.73       0.76       11
              neu       0.68       0.90       0.78       31
              pos       0.94      0.75       0.84       44

              accuracy                        0.80       86
   macro avg       0.81       0.79       0.79       86
weighted avg       0.83       0.80       0.81       86
```

Accuracy of NLTK library Range ((-0.25)-(0.35)): **80.23%**

2014 DATA SET

```
--------------- Precision/Recall/F1 Score ---------------
                precision          recall  f1-score   support

              neg       0.93       0.82       0.87       51
              neu       0.89       0.89       0.89       96
              pos       0.81       0.92       0.86       52

              accuracy                        0.88        199
   macro avg       0.88       0.88       0.88       199
weighted avg       0.88       0.88       0.88       199
```

Accuracy of NLTK library Range ((-0.25)-(0.35)): **87.94%**

2015 DATA SET

```
--------------- Precision/Recall/F1 Score ---------------
              precision          recall  f1-score   support

              neg       0.95       0.86       0.90       21
              neu       0.93       0.85       0.88       59
              pos       0.67       0.90       0.77       20

              accuracy                        0.86       100
   macro avg       0.85       0.87       0.85       100
weighted avg       0.88       0.86       0.86       100
```

Accuracy of NLTK library Range ((-0.25)-(0.35)): **86.0%**

Range C

2013 DATA SET

```
--------------- Precision/Recall/F1 Score ---------------
              precision          recall  f1-score   support

              neg       0.80       0.73       0.76       11
              neu       0.64       0.94       0.76       31
              pos       1.00       0.70       0.83       44

              accuracy                        0.79       86
   macro avg       0.81       0.79       0.78       86
weighted avg       0.85       0.79       0.80       86
```

Accuracy of NLTK library Range ((-0.25)-(0.4)): **79.07%**

2014 DATA SET

```
--------------- Precision/Recall/F1 Score ---------------
              precision          recall  f1-score   support

              neg       0.93       0.82       0.87       51
              neu        0.86       0.91       0.88       96
              pos       0.85       0.87       0.86       52

              accuracy                        0.87       199
   macro avg       0.88       0.87       0.87       199
weighted avg       0.88       0.87       0.87       199
```

Accuracy of NLTK library Range ((-0.25)-(0.4)): **87.44%**

2015 DATA SET

```
--------------- Precision/Recall/F1 Score ---------------
              precision          recall  f1-score   support

              neg         0.95       0.86       0.90       21
              neu       0.87       0.92       0.89       59
              pos       0.74       0.70       0.72       20

              accuracy                        0.86       100
   macro avg       0.85       0.82       0.84       100
weighted avg       0.86       0.86       0.86       100
```

Accuracy of NLTK library Range ((-0.25)-(0.4)): **86.0%**

Similarly, we calculated precision, recall, and F1 score for each sentiment label for the combined dataset according to the above selected four different ranges.

Table 3.6.3.4: Different ranges to classify sentiment according to compound value.

	Range B	**Range C**
Positive.precision	0.82	0.87
Positive.recall	0.85	0.78
Positive.F1	0.84	0.82
Negative.precision	0.92	0.92
Negative.recall	0.82	0.82
Negative.F1	0.87	0.87
Neutral.precision	0.86	0.82
Neutral.recall	0.88	0.91
Neutral.F1	0.87	0.86

Evaluated the performance of our sentiment analysis models for different ranges and found the best range.

Here we selected most accurate range as, **Range B**

Negative Compound<-0.25	Neutral -0.25 <=compound<=0.35	Positive Compound>0.35	**Range B** **NLTK_B**

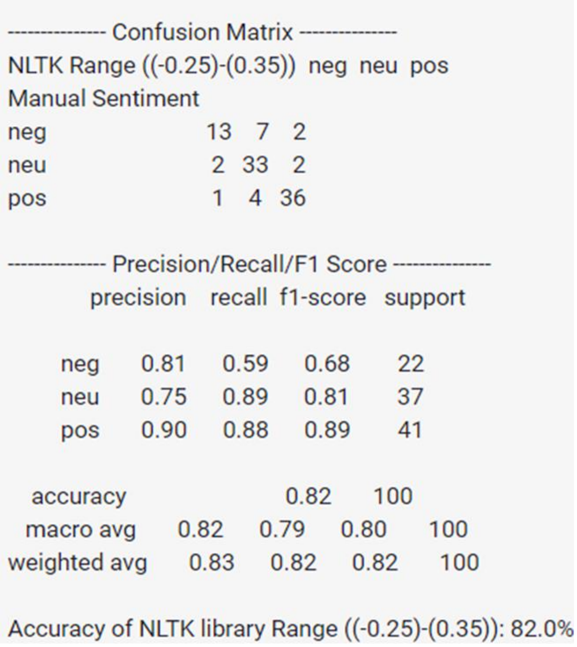
3.7: Apply the model

A test data set was used to apply the model.

According to 3.5.3.1: Generating sentiment score section, here we generated sentiment score for the test data set.

Sentiments were classified according to the selected most accurate range, -0.25 ≤ compound ≤ 0.35 Range B.

Accuracy was found for the test data set using a confusion matrix.



```
-------------- Confusion Matrix --------------
NLTK Range ((-0.25)-(0.35))  neg  neu  pos
Manual Sentiment
neg                          13    7    2
neu                           2   33    2
pos                           1    4   36

-------------- Precision/Recall/F1 Score --------------
          precision    recall  f1-score   support

       neg      0.81      0.59      0.68        22
       neu      0.75      0.89      0.81        37
       pos      0.90      0.88      0.89        41

  accuracy                          0.82       100
 macro avg      0.82      0.79      0.80       100
weighted avg      0.83      0.82      0.82       100

Accuracy of NLTK library Range ((-0.25)-(0.35)): 82.0%
```

Figure 3.7.7: Confusion matrix and classification report for sentiments

Overall, the test data set is performing relatively well for sentiment analysis, with an accuracy of 82%.

3.8: Visualizing the results

3.8.1: Designing the dashboard

The dashboard was designed according to the problem and objectives of our project. Then we sketched an interface/output and decided to present our results in a Streamlit dashboard.

3.8.2: Installing Streamlit

Streamlit is a Python library that makes it easy to create interactive web apps.

pip install streamlit

3.8.3: Importing necessary libraries

Streamlit, pandas, nltk, string, StringIO, session_state, WordCloud, STOPWORDS, matplotlib.pyplot, plotly.express, emoji, calendar, streamlit_option_menu

3.8.4: Defining our data

load_data() function was created for load data from a CSV file.

3.8.5: Defining the page layout

Three pages and a sidebar were added to our dashboard.

- Home
- EDA
- Sentiment Generator

```
selected = option_menu( menu_title=None,
options=["Home","EDA","Sentiment Generator"],
icons=['house','bar-chart','emoji-heart-eyes'],
menu_icon = 'cast',
orientation='horizontal',
styles={ "icon":{"color":"black","font-size":"25px"},
"nav-link":{"font-size":"25px","--hover-color":"#052f2b",},
"nav-link-selected":{"background-color":"#0c7c72"}
},
 )
```

The given code used the *option_menu()* function from the Streamlit library to create a horizontal menu bar with three options: "Home", "EDA", and "Sentiment Generator". The icons parameter was used to specify icons for each option, with the "house" icon representing "Home", the "bar-chart" icon representing "EDA", and the "emoji-heart-eyes" icon representing "Sentiment Generator". The menu_icon parameter was used to specify an icon for the entire menu bar, with the "cast" icon being used in this case.The styles parameter is used to specify the styling for the icons and menu links, with a black color and font size of 25 pixels for the icons, a font size of 25 pixels and a custom hover color for the links, and a custom background color for the selected link. Overall, this code creates a simple and visually appealing menu bar that can be used for navigation within a Streamlit app.

3.8.5.1: Home Page

The home page of a data set serves as an introduction to the data . Data set description will include information such as the name of the data set, the source of the data, the date the data was collected, the number of observations or variables included.The home page of a data set serves as an introduction to the data . Data set description will include information such as the name of the data set, the source of the data, the date the data was collected, the number of observations or variables included. Also included percentages of the three sentiments (positive, negative, and neutral) in our data set.

To implement this, we used the st.write function to display your data set description and sentiment percentages in streamlit library in Python. The st.write function is a simple and versatile way to display text, data, and visualizations in a Streamlit app.

3.8.5.2: EDA Page

All the charts and world clouds were added to exploratory data analysis

i. **Data was extracted according to the given date range.**

ii. **“Download extracted data” option was added as a button.**

```
elif selected == "EDA":
```

```
data = load_data()
start,end = date_range()

extract = st.button('Extract data')
if st.session_state.get('button') != True:
        st.session_state['button'] = extract

if st.session_state['button'] == True:
        date_range_df = data.loc[data["Date"].between(str(start), str(end))]
  st.header("Extracted Data set")
        st.dataframe(date_range_df)
        ExtractedData = date_range_df
        fileName = 'Date Range ([%s] - [%s]).csv'%(str(start), str(end))
        DownloadDataFrame(ExtractedData,fileName)
```

option_menu() function. If the user selects "EDA", the load_data() function is called to load the data. The date_range() function is then called to get the start and end dates of the date range the user wants to extract data from.

The code creates an "Extract data" button using st.button(). If the button is clicked, the code uses st.session_state to keep track of whether the button has been clicked before. If it is the first time the button is clicked, the code sets st.session_state['button'] to True.

If the button has been clicked at least once, the code extracts the data from the loaded dataset between the start and end dates using pandas' loc function. The extracted data is displayed using st.dataframe() and is stored in a variable called ExtractedData. The function DownloadDataFrame() is called to download the extracted data as a CSV file. Finally, the extracted data is stored in a variable named ExtractedData.

iii. Data was filtered according to the specific author.

```
if author != 'All':
        date_range_df = date_range_df[date_range_df['Author_ID']==author]
        st.header("Filterd Data set")
        st.subheader(f"Author :- {author}")
        st.dataframe(date_range_df)
        fileName = '%s ([%s] - [%s]).csv'%(author,str(start), str(end))
        DownloadDataFrame(date_range_df,fileName)
        else:
   author = "All Author"
```

This code filters the extracted dataset based on the selected author in the sidebar. If the selected author is not 'All', then the extracted dataset is filtered to include only the rows where the 'Author_ID' column matches the selected author. The filtered dataset is displayed in a header and subheader, showing the selected author, and a dataframe. The filtered dataset can also be downloaded as a CSV file, with the filename including the selected author's name and the selected date range. If the selected author is 'All', then the author variable is set to "All Author".

iv. According to the above condition,

- World clouds were drawn for selected sentiment. Radio buttons were to select sentiment.

```
def wordcloud(text,title):
st.set_option('deprecation.showPyplotGlobalUse', False)
text = WordCloud().generate(str(text))
plt.imshow(text)
plt.axis('off')
plt.title(title)
st.pyplot()
```

The wordcloud() function generates a word cloud visualization using the WordCloud class from the wordcloud library. This function takes in two arguments: text and title. text is the input text data, which is converted to a string format using the str() function.

The WordCloud() function is then used to generate a word cloud object from the text data, which is passed to the generate() method of the WordCloud object. This method generates a word cloud based on the frequency of words in the input text.

- Visualized number of sentiments using bar chart or pie chart. Radio buttons were added to the sidebar to select charts.

```
st.sidebar.header("Bar Chart/Pie Chart")
    select = st.sidebar.radio('What visualization type do you want to display
number of sentiments ?', (None,'Bar Chart', 'Pie Chart'))
```

The given code creates a sidebar in a Streamlit app that allows users to select between a bar chart or a pie chart for displaying the number of tweets by sentiment. The sidebar is created using the st.sidebar function and the header is set to "Bar Chart/Pie Chart" using the header() method.

The radio() method is used to create a radio button for selecting the visualization type. The options for the radio button are "Bar Chart" and "Pie Chart", and a default option of None is included to prevent the chart from being displayed until an option is selected.

```
    if select != None:
    sentiment = date_range_df['NLTK_Tag'].value_counts().index.tolist()
    sentiment_count = date_range_df['NLTK_Tag'].value_counts().tolist()
      percentage = [i*100/sum(sentiment_count) for i in sentiment_count]
    percentage = [str(round(i,2))+'%' for i in percentage]
    sentiment_count = pd.DataFrame({'Sentiment':sentiment,
'Tweets':sentiment_count})

    st.markdown("### Number of tweets by sentiment")
    st.subheader(f"Author :- {author}")
    if select == 'Bar Chart':
    fig = px.bar(sentiment_count, x='Sentiment', y='Tweets',text = percentage,
color='Sentiment')
    st.plotly_chart(fig)
    else:
    fig = px.pie(sentiment_count, values='Tweets', names='Sentiment')
    st.plotly_chart(fig)
```

If an option other than None is selected, the code calculates the number of tweets for each sentiment using the value_counts() method of the date_range_df['NLTK_Tag'] column. The resulting sentiment and tweet count values are stored in a panda DataFrame, sentiment_count.

The percentage of tweets for each sentiment is also calculated using list comprehension, and a new column is added to the sentiment_count DataFrame containing the percentage values.

The code then displays the header "Number of tweets by sentiment" and the author name in the app using st.markdown() and st.subheader(), respectively.

Finally, if the user selects "Bar Chart", the px.bar() function from plotly.express is used to create a bar chart of the tweet counts by sentiment. The chart is colored by sentiment using the color parameter and the percentage values are displayed on the chart using the text parameter.

If the user selects "Pie Chart", the px.pie() function from plotly.express is used to create a pie chart of the tweet counts by sentiment. The chart displays the percentage of tweets for each sentiment. The resulting chart is displayed in the app using st.plotly_chart().

Overall, this code provides an easy way to display the number of tweets by sentiment using either a bar chart or a pie chart in a Streamlit app.

v. **Visualized changes of sentiments of data over year using line charts according to the selected year. Drop down menu was added to the sidebar to select the year.**

```
st.sidebar.header("Line Chart")
   year = st.sidebar.selectbox("What year do you want to see the sentiment changes monthly
?",[None]+list(range(2008, 2018)))
f year != None:
          df = data[['Date','NLTK_Tag']]
          df['Date'] = pd.to_datetime(df['Date'],errors='coerce')
          df['Year'] = df['Date'].dt.year
          df['Month'] = df['Date'].dt.month
          df = df[df['Year']==year]

          pos,neg,neu=[],[],[]
          for month in range(1,13):
          df0 = df[df['Month']==month]
          pos_cnt,neu_cnt,neg_cnt = 0,0,0
          for sntmnt in df0['NLTK_Tag'].tolist():
          if sntmnt == 'Positive':
          pos_cnt += 1
          elif sntmnt == 'Neutral':
          neu_cnt += 1
          else:
          neg_cnt += 1
          pos.append(pos_cnt)
          neu.append(neu_cnt)
          neg.append(neg_cnt)

          line_chart_data = pd.DataFrame({'Positive':pos,'Neutral':neu,'Negative':neg},index =
list(calendar.month_name)[1:])
          st.markdown("### Monthly Changes of Sentiments Customer Support Data over Year")
```

```
        fig = px.line(line_chart_data,color_discrete_map={"Positive": "green","Neutral":
"white","Negative": "red"}).update_layout(
        title = {'text':f"Year - {year}",'x':0.5}, xaxis_title="Month", yaxis_title="Number of
Tweets",legend_title="Sentiment")

    st.plotly_chart(fig, use_container_width=True)
```

The chart shows the monthly changes of sentiment (positive, negative, and neutral) in customer support data for the selected year. The code starts by creating a selection box for the user to choose the year. Then it extracts the relevant data from the dataframe based on the year selected by the user. The code then calculates the number of tweets for each sentiment (positive, negative, and neutral) for each month of the selected year. Finally, it creates a Line Chart using Plotly that displays the monthly changes in sentiment for the selected year. The Line Chart is color-coded based on sentiment (green for positive, white for neutral, and red for negative).

3.8.5.3: Sentiment Generator Page

Sentiment Generator was added.

i. **Sentiment was generated for a given text or .txt file.**

- Input field was added to enter text or added to upload .txt file.

```
option = st.radio("Select your input option :",("Type a text",".txt"))
 if option == "Type a text":
text = st.text_input("Please enter your text in ***english*** for analysis :")
else:
file = st.file_uploader("Choose a file : ")
if file != None:
stringio = StringIO(file.getvalue().decode("utf-8"))
text = stringio.read()
```

This code allows the user to select their input option either by typing a text or by uploading a .txt file. If the user selects "Type a text" option, they can enter their text in English for analysis using st.text_input() method. If the user selects ".txt" option, they can upload his/her .txt file using st.file_uploader() method. Once the file is uploaded, it is decoded using the "utf-8" decoding format and then read using stringio.read() method to convert it into a string. The extracted text is then stored in the text variable for further analysis.

ii. **Generate button was added to click after adding text or upload file.**

iii. **Checkboxes were added two display two options as for Perform Sentiment Analysis and draw the word clouds.**

```
 generate = st.button('Generate')
if st.session_state.get('generate') != True:
        st.session_state['generate'] = generate
if st.session_state['generate']:
        check1 = st.checkbox("Perform Sentiment Analysis")
        check2 = st.checkbox("Draw a wordcloud")
        text = preprocess(text)
```

```
if check1:
sentimentGenerator(text)
if check2:
title = "WordCloud for generated text"
wordcloud(text,title)
```

The above code snippet creates a "Generate" button and checks if the button has been clicked. If the button is clicked, it creates two checkboxes: "Perform Sentiment Analysis" and "Draw a Wordcloud". It then preprocesses the input text by calling the "preprocess" function, and if the "Perform Sentiment Analysis" checkbox is checked, it calls the "sentimentGenerator" function to perform sentiment analysis on the preprocessed text. If the "Draw a Wordcloud" checkbox is checked, it calls the "wordcloud" function to generate a wordcloud for the preprocessed text. Finally, it passes the preprocessed text to both functions if their respective checkboxes are checked.

```
def sentimentGenerator(text):
    analyzer = SentimentIntensityAnalyzer()
    result = analyzer.polarity_scores(text)

    if result['compound'] > 0.35:
      st.success("Sentiment is Positive")
    elif result['compound'] < -0.25:
      st.error("Sentiment is Negative")
    else:
      st.info("Sentiment is Neutral")

    st.write(f'Positive - {result["pos"]*100}%')
    st.write(f'Neutral - {result["neu"]*100}%')
    st.write(f'Negative - {result["neg"]*100}%')
```

This is a Python function that performs sentiment analysis on a given text using the Vader pre-built model from the Natural Language Toolkit (NLTK) library. The function takes a single input parameter, text, which is the text to be analyzed for sentiment.

If the compound sentiment score is greater than 0.35, the function returns a success message indicating a positive sentiment. If the compound sentiment score is less than -0.25, the function returns an error message indicating a negative sentiment. Otherwise, the function returns an info message indicating a neutral sentiment.

Here we used the most accurate range(Range B) that we found from our evaluation.

CHAPTER 04

This section consists of the results and discussions of the sentiment analysis project, which includes the analysis of the performance of various machine learning models, the evaluation of feature selection techniques, and a discussion on the insights obtained from the sentiment analysis of the data.

4.1: Results and Discussion of EDA

The Customer Support on Twitter dataset used in this study is a comprehensive and extensive corpus of tweets and replies that spans almost a decade, from May 2008 to December 2017. The large volume of data in this dataset, consisting of nearly 2.8 million tweets and replies from major brands on Twitter, provides a unique opportunity to conduct exploratory data analysis and uncover insights into the patterns and trends in customer support interactions over time. The dataset's duration of 3496 days ensures that the analysis will cover a wide range of customer support experiences across a variety of industries.

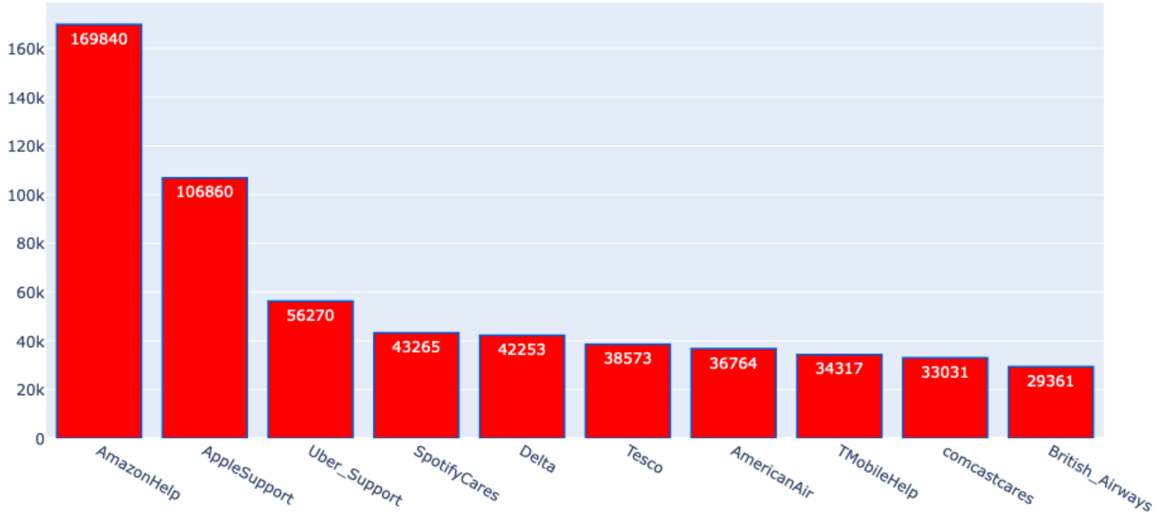


Figure 4.1.8: Frequency of sent messages vs. the corresponding customer service

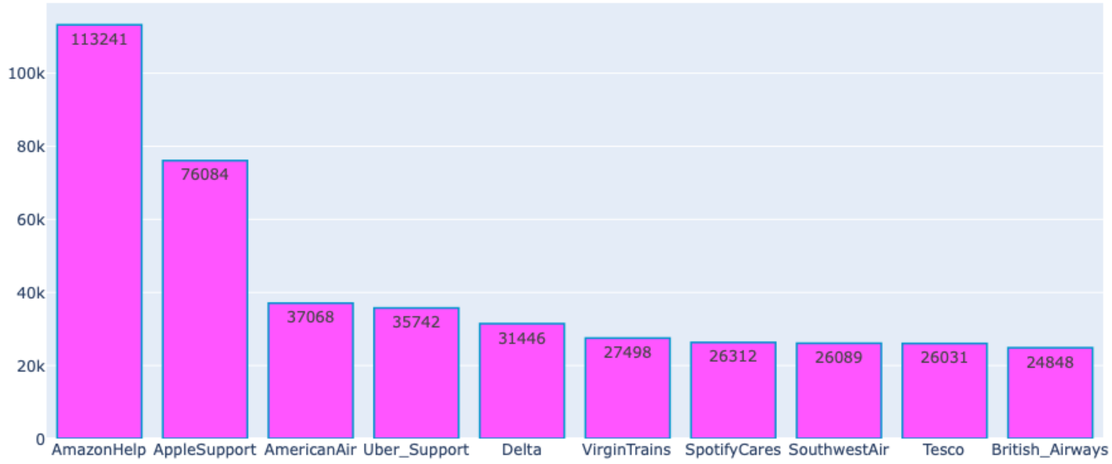
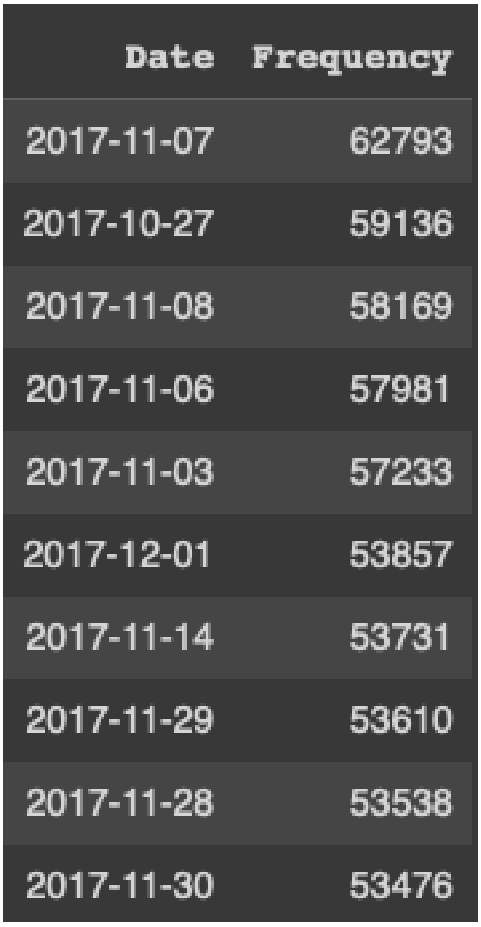


Figure 4.1.9: Frequency of received messages vs. the corresponding customer service

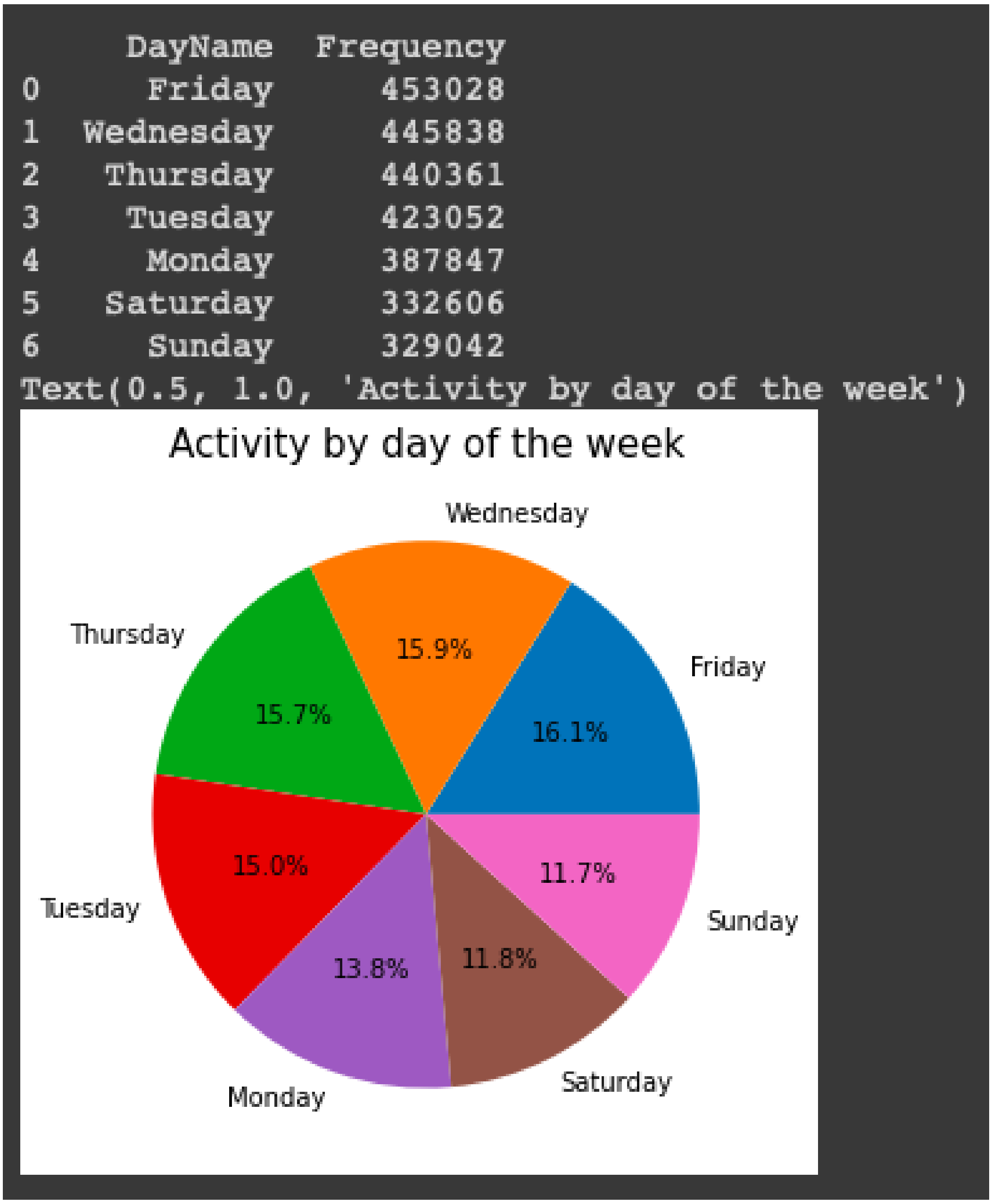
The results of our analysis show that AmazonHelp received the most messages out of all the customer support services included in the dataset, with a total of 113,241 messages. Additionally, AmazonHelp also sent the most messages, with a total of 169,840 messages. AppleSupport, on the other hand, received and sent the second most messages out of all the customer support services, with a total of 76,084 and 106,860 messages respectively. It is worth noting that all the other customer support services included in the dataset received at least 24,000 messages and sent more than 29,000 messages during the study period, indicating a significant level of customer engagement and communication across the board. These findings highlight the importance of effective customer support in maintaining customer satisfaction and brand loyalty, as well as the need for businesses to invest in robust customer support systems and strategies to ensure positive customer experiences.

Table 4.1.5: Top 10 most active dates and their frequencies



Date	Frequency
2017-11-07	62793
2017-10-27	59136
2017-11-08	58169
2017-11-06	57981
2017-11-03	57233
2017-12-01	53857
2017-11-14	53731
2017-11-29	53610
2017-11-28	53538
2017-11-30	53476

Our analysis of the Customer Support on Twitter dataset revealed that November 7, 2017, had the highest number of tweets, with a total of 62,793 messages. This suggests that this particular day was a particularly active one in terms of customer support interactions on Twitter. Furthermore, when we examined the dataset over time, we found that 2017 was the most active year. Within 2017, the month of November emerged as the most active month. The high level of activity in November 2017 may be attributed to the holiday season and associated shopping events, during which customers may have had more queries and issues that required the attention of customer support services. These findings suggest that businesses need to be particularly prepared and responsive during peak periods of customer engagement to maintain customer satisfaction and loyalty.

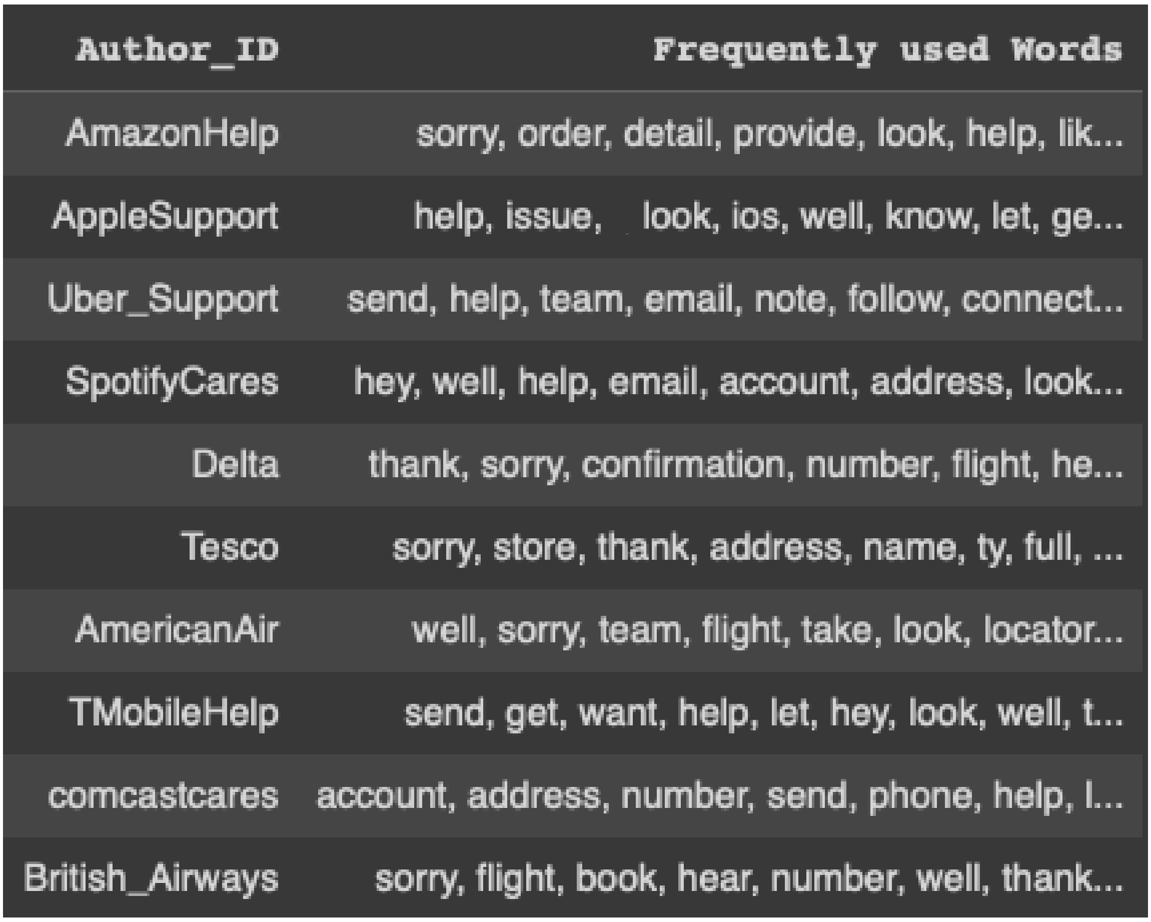


```
     DayName  Frequency
0     Friday     453028
1  Wednesday     445838
2   Thursday     440361
3    Tuesday     423052
4     Monday     387847
5   Saturday     332606
6     Sunday     329042
Text(0.5, 1.0, 'Activity by day of the week')
```

Figure 4.1.10: Pie chart for activity by day of the week

Through our examination of the Customer Support on Twitter dataset, we discovered noteworthy trends in how often tweets were sent and received on various days of the week. The results show that Friday was the most active day of the week, with a total of 453,028 tweets sent and received, accounting for 16.1% of all tweets in the dataset. On the other hand, Sunday was the least active day of the week, with a total of 329,042 tweets sent and received, accounting for only 11.7% of all tweets in the dataset. These findings suggest that there may be a correlation between the day of the week and the level of customer engagement with customer support services on Twitter. It is possible that customers are more likely to seek support during the weekdays, when they are likely to be at work and have access to their devices, and less active during the weekends when they may be engaging in other activities. The insights gained from this analysis may be useful for businesses in planning their customer support strategies, such as increasing staffing levels and resources during peak activity periods and adjusting schedules accordingly.

Table 4.1.8: Most frequently used words by customer service



Author_ID	Frequently used Words
AmazonHelp	sorry, order, detail, provide, look, help, lik...
AppleSupport	help, issue, look, ios, well, know, let, ge...
Uber_Support	send, help, team, email, note, follow, connect...
SpotifyCares	hey, well, help, email, account, address, look...
Delta	thank, sorry, confirmation, number, flight, he...
Tesco	sorry, store, thank, address, name, ty, full, ...
AmericanAir	well, sorry, team, flight, take, look, locator...
TMobileHelp	send, get, want, help, let, hey, look, well, t...
comcastcares	account, address, number, send, phone, help, l...
British_Airways	sorry, flight, book, hear, number, well, thank...

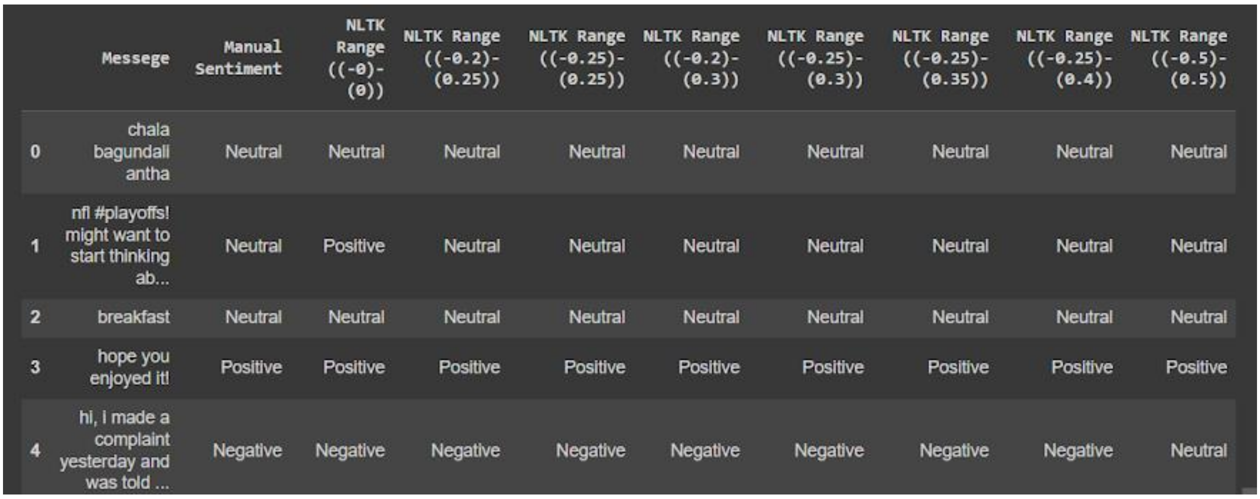
Through our investigation of the Customer Support on Twitter dataset, we identified the words that were most frequently utilized by all customer support services included in the dataset. The word "sorry" was used most frequently, indicating that customer support services place great emphasis on expressing empathy and acknowledging customers' concerns. Additionally, the words "help" and "thank" were also used frequently, suggesting that aiding and expressing gratitude are key components of effective customer support communication.

Identifying these frequently used words can provide valuable insights into the customer support strategies employed by businesses. For example, by analyzing the frequency and context of these words, businesses can better understand their customers' needs and preferences and adjust their customer support strategies accordingly. Furthermore, businesses can also use these insights to train their customer support staff on effective communication strategies and best practices. Overall, understanding the language and communication patterns of customer support interactions can help businesses build stronger relationships with their customers and improve overall customer satisfaction.

4.2: Results and Discussion of the model

Evaluated the performance of our sentiment analysis models for different ranges and found the best range.

Table 4.2.10: Three different datasets after generating sentiments according to different ranges.



	Messege	Manual Sentiment	NLTK Range ((-0)-(0))	NLTK Range ((-0.2)-(0.25))	NLTK Range ((-0.25)-(0.25))	NLTK Range ((-0.2)-(0.3))	NLTK Range ((-0.25)-(0.3))	NLTK Range ((-0.25)-(0.35))	NLTK Range ((-0.25)-(0.4))	NLTK Range ((-0.5)-(0.5))
0	chala bagundali antha	Neutral	Neutral	Neutral	Neutral	Neutral	Neutral	Neutral	Neutral	Neutral
1	nfl #playoffs! might want to start thinking ab...	Neutral	Positive	Neutral	Neutral	Neutral	Neutral	Neutral	Neutral	Neutral
2	breakfast	Neutral	Neutral	Neutral	Neutral	Neutral	Neutral	Neutral	Neutral	Neutral
3	hope you enjoyed it!	Positive	Positive	Positive	Positive	Positive	Positive	Positive	Positive	Positive
4	hi, i made a complaint yesterday and was told ...	Negative	Negative	Negative	Negative	Negative	Negative	Negative	Negative	Neutral

Used a confusion matrix to find the accuracy for nltk library using seven different ranges for three data sets and combined data set.

Table 4.2.13: The accuracy for nltk library using seven different ranges for three data sets.

	Range 0 NLTK Range(0)	Range 0.5 NLTK Range(0.5)	Range 0.25 NLTK Range(0.25)	Range A NLTK_A --0.25<c<0.3	Range B NLTK_B --0.25<c<0.35	Range C NLTK_C --0.25<c<0.4	Range D NLTK_D --0.20<c<0.25	Range E NLTK_E --0.20<c<0.3
2013 dataset Pasindu	84.88%	68.34%	81.4%	81.4%	80.23%	79.07%	81.4%	81.4%
2014 dataset Hisham	70.35%	78%	83.42%	84.92%	87.94%	87.44%	80.9%	82.41%
2015 dataset Pawani	73.0%	65.12%	82.0%	82.0%	86.0%	86.0%	81.0%	81.0%

combined data set	**74.29%**	**70.13%**	**82.6%**	**83.38%**	**85.71%**	**85.19%**	**81.04%**	**81.82%**

Selected the most accurate two ranges:

According to the above results of the confusion matrix, here we can see Range B and Range C have more accuracy compared to other ranges. We dropped the other three ranges because of low accuracy.

Calculated precision, recall, and F1 score for each sentiment label for selected two different ranges: Range ,Range ,Range and Range

For three data sets,

Range B

2013 DATA SET

```
--------------- Precision/Recall/F1 Score ---------------
             precision          recall  f1-score   support

             neg       0.80      0.73      0.76      11
             neu       0.68      0.90      0.78      31
             pos       0.94     0.75       0.84      44

             accuracy                      0.80      86
   macro avg         0.81      0.79      0.79      86
weighted avg         0.83      0.80      0.81      86
```

Accuracy of NLTK library Range ((-0.25)-(0.35)): **80.23%**

2014 DATA SET

```
--------------- Precision/Recall/F1 Score ---------------
               precision        recall  f1-score   support

             neg       0.93      0.82      0.87      51
             neu       0.89      0.89      0.89      96
             pos       0.81      0.92      0.86      52

             accuracy                      0.88       199
   macro avg         0.88      0.88      0.88      199
weighted avg         0.88      0.88      0.88      199
```

Accuracy of NLTK library Range ((-0.25)-(0.35)): **87.94%**

Range C

2013 DATA SET

```
--------------- Precision/Recall/F1 Score ---------------
             precision          recall  f1-score   support

             neg       0.80      0.73      0.76      11
             neu       0.64      0.94      0.76      31
             pos       1.00      0.70      0.83      44

             accuracy                      0.79      86
   macro avg         0.81      0.79      0.78      86
weighted avg         0.85      0.79      0.80      86
```

Accuracy of NLTK library Range ((-0.25)-(0.4)): **79.07%**

2014 DATA SET

```
--------------- Precision/Recall/F1 Score ---------------
             precision          recall  f1-score   support

             neg       0.93      0.82      0.87      51
             neu        0.86     0.91      0.88      96
             pos       0.85      0.87      0.86      52

             accuracy                      0.87      199
   macro avg         0.88      0.87      0.87      199
weighted avg         0.88      0.87      0.87      199
```

Accuracy of NLTK library Range ((-0.25)-(0.4)): **87.44%**

```
2015 DATA SET

--------------- Precision/Recall/F1 Score ---------------
             precision          recall  f1-score   support

             neg      0.95     0.86      0.90      21
             neu      0.93     0.85      0.88      59
             pos      0.67     0.90      0.77      20

             accuracy                    0.86      100
   macro avg          0.85     0.87      0.85      100
weighted avg          0.88     0.86      0.86      100

Accuracy of NLTK library Range ((-0.25)-(0.35)):
86.0%
```

```
2015 DATA SET

--------------- Precision/Recall/F1 Score ---------------
             precision          recall  f1-score   support

             neg        0.95   0.86      0.90      21
             neu      0.87     0.92      0.89      59
             pos      0.74     0.70      0.72      20

             accuracy                    0.86      100
   macro avg          0.85     0.82      0.84      100
weighted avg          0.86     0.86      0.86      100

Accuracy of NLTK library Range ((-0.25)-(0.4)):
86.0%
```

By looking at above results for range B and range C, we can evaluate performance clearly for each year.

For Range B, 2013 dataset

- For the "neg" sentiment class, the precision is 0.80, the recall is 0.73, and the F1-score is 0.76. This indicates that the model is correctly identifying most instances of "neg" sentiment, but there is some room for improvement in terms of recall.
- For the "neu" sentiment class, the precision is 0.68, the recall is 0.90, and the F1-score is 0.78. This suggests that the model is performing well for "neu" sentiment in terms of recall, but precision is relatively low.
- For the "pos" sentiment class, the precision is 0.94, the recall is 0.75, and the F1-score is 0.84. This indicates that the model is performing well for "pos" sentiment in terms of precision, but there is some room for improvement in terms of recall.
- Overall, the model's performance is relatively mixed, with varying levels of precision and recall for different sentiment classes. The accuracy of the model is 80.23%, but looking at the precision, recall, and F1-score for each sentiment class can give us a better understanding of where the model is performing well and where there is room for improvement

For Range B, 2014 dataset

- For the "neg" sentiment class, the precision is 0.93, the recall is 0.82, and the F1-score is 0.87. This indicates that the model is correctly identifying most instances of "neg" sentiment, but there is some room for improvement in terms of recall.
- For the "neu" sentiment class, the precision is 0.89, the recall is 0.89, and the F1-score is 0.89. This suggests that the model is performing relatively well for "neu" sentiment, with both precision and recall at a similar level.
- For the "pos" sentiment class, the precision is 0.81, the recall is 0.92, and the F1-score is 0.86. This indicates that the model is performing well for "pos" sentiment, but there is some room for improvement in terms of precision.

- Overall, the model is performing relatively well for sentiment analysis, with an accuracy of 87.94%. However, there is some room for improvement in terms of precision and recall for some of the sentiment classes.

For Range B, 2015 dataset

- For the neg sentiment class, the precision is 0.95, which means that out of all the predicted neg sentiment labels, 95% of them are correct. The recall is 0.86, which means that out of all the actual neg sentiment labels, 86% of them were correctly predicted as neg. The f1-score, which is the harmonic mean of precision and recall, is 0.90. The support for this class is 21.
- Similarly, for the neu sentiment class, the precision is 0.93, the recall is 0.85, and the f1-score is 0.88. The support for this class is 59.
- For the pos sentiment class, the precision is 0.67, the recall is 0.90, and the f1-score is 0.77. The support for this class is 20.
- The accuracy of the model is 0.86, which means that 86% of the labels were predicted correctly. The macro average of precision, recall, and f1-score is 0.85, and the weighted average is 0.88. The accuracy falls within the range of NLTK library accuracy.

For Range C, 2013 dataset

- For neg class, the precision is 0.8, recall is 0.73, and F1 score is 0.76. This means that when the model predicts a negative sentiment, it is correct 80% of the time. However, it only identifies 73% of the actual negative sentiments correctly. The F1 score is the harmonic mean of precision and recall and is a balanced measure between the two.
- For neu class, the precision is 0.64, recall is 0.94, and F1 score is 0.76. This means that the model is not very good at identifying neutral sentiments correctly, with a precision of only 64%. However, when there is a neutral sentiment, the model correctly identifies it 94% of the time.
- For pos class, the precision is 1.0, recall is 0.70, and F1 score is 0.83. This means that the model is very good at identifying positive sentiments, with a precision of 100%. However, it only identifies 70% of the actual positive sentiments correctly.
- The accuracy of the model is 79.07%, which is the percentage of correctly classified sentiments overall.

For Range C, 2014 dataset

- For the neg sentiment label, the precision is 0.93, which means that out of all the predictions that were classified as neg, 93% were actually neg. The recall is 0.82, which means that out of all the actual neg instances, the model correctly classified 82% as neg. The F1 score is 0.87, which is the harmonic mean of precision and recall.
- Similarly, for the neu sentiment label, the precision is 0.86, recall is 0.91, and F1 score is 0.88. For the pos sentiment label, the precision is 0.85, recall is 0.87, and F1 score is 0.86.
- The weighted average precision, recall, and F1 score are 0.88, 0.87, and 0.87, respectively. This indicates that the model has overall good performance across all sentiment labels.

For Range C, 2015 dataset

- For the neg sentiment label, precision of 0.95,which means that out of all the predictions that were classified as neg, 95% were actually neg. recall of 0.86,which means that out of all the actual neg instances, the model correctly classified 86% as neg. and F1-score of 0.90,which is the harmonic mean of precision and recall
- Similarly, for the neu sentiment label, precision of 0.87, recall of 0.92, and F1-score of 0.89.For the Pos sentiment label the precision of 0.74, recall of 0.70, and F1-score is 0.72

- Overall, the model achieved an accuracy of 86%, which indicates that it performs reasonably well on this dataset. However, the precision, recall, and F1-score values for the positive sentiment category are lower than those for the other categories, which suggests that the model might have more difficulty correctly identifying positive sentiment expressions.

For combined data set,

Table 4.2.16: Precision, recall and f1 score combined data for each sentiment using two selected different ranges.

	Range B	**Range C**
Positive.precision	0.82	0.87
Positive.recall	0.85	0.78
Positive.F1	0.84	0.82
Negative.precision	0.92	0.92
Negative.recall	0.82	0.82
Negative.F1	0.87	0.87
Neutral.precision	0.86	0.82
Neutral.recall	0.88	0.91
Neutral.F1	0.87	0.86

Selected the most accurate range:

According to the above result here we can compare precision and recall of each sentiment when we are selecting the most accurate range.

For Positive Sentiments,

We got higher values for precision and recall of Range B than Range C. That means the model is good at identifying positive instances when Range is B.

For Negative Sentiments,

We got higher and similar values for precision at Range B and Range C. We got higher and similar values for recall at Range B and Range C. That means the model is good at identifying negative instances when range is Range B or C.

For Neutral Sentiments,

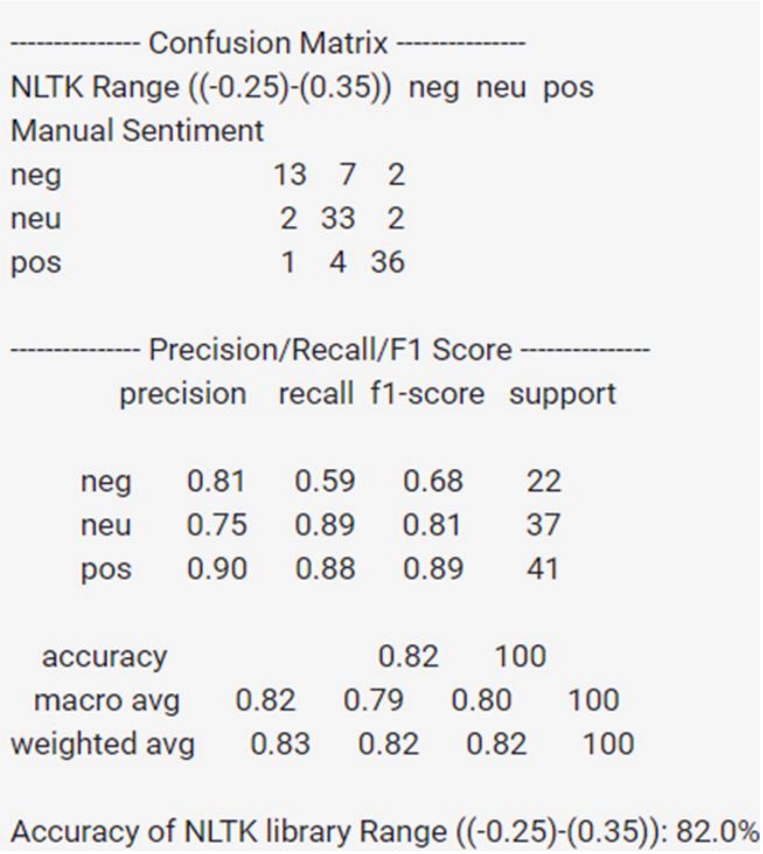
We got higher values for precision and recall of Range B than Range C. That means the model is good at identifying neutral instances when Range is B.

Considering all the metrics used above to evaluate the performance and 85.71% overall accuracy which was calculated from confusion matrix, here we selected most accurate range as, **Range B**

Negative Compound<-0.25	Neutral -0.25 <=compound<=0.35	Positive Compound>0.35	**Range B** **NLTK_B**

Apply for model

Selected range used to generate sentiments in test data(**testdata**) and find accuracy using confusion matrix below.



```
-------------- Confusion Matrix --------------
NLTK Range ((-0.25)-(0.35))  neg  neu  pos
Manual Sentiment
neg                          13    7    2
neu                           2   33    2
pos                           1    4   36

-------------- Precision/Recall/F1 Score --------------
          precision    recall  f1-score   support

       neg       0.81      0.59      0.68        22
       neu       0.75      0.89      0.81        37
       pos       0.90      0.88      0.89        41

  accuracy                           0.82       100
 macro avg       0.82      0.79      0.80       100
weighted avg       0.83      0.82      0.82       100

Accuracy of NLTK library Range ((-0.25)-(0.35)): 82.0%
```

Figure 4.2.12: Confusion matrix and classification report for sentiments

For the negative sentiment, the precision is 0.81, which means that out of all the instances that were predicted as negative, 81% were actually negative. The recall is 0.59, which means that out of all the instances that are actually negative, 59% were correctly predicted as negative. The F1 score is 0.68, which is the harmonic mean of precision and recall, and provides a combined measure of the two.

For the neutral sentiment, the precision is 0.75, which means that out of all the instances that were predicted as neutral, 75% were actually neutral. The recall is 0.89, which means that out of all the instances that are actually neutral, 89% were correctly predicted as neutral. The F1 score is 0.81. For the positive sentiment, the precision is 0.90, which means that out of all the instances that were predicted as positive, 90% were actually positive. The recall is 0.88, which means that out of all the instances that are actually positive, 88% were correctly predicted as positive. The F1 score is 0.89.

Overall, the performance of the model seems to be quite good, with high precision, recall, and F1 scores for all three sentiment categories.

4.3: Result and discussion for Final Output

Design of output dashboard

Below figures show the sketched design of the final output dashboard interface.

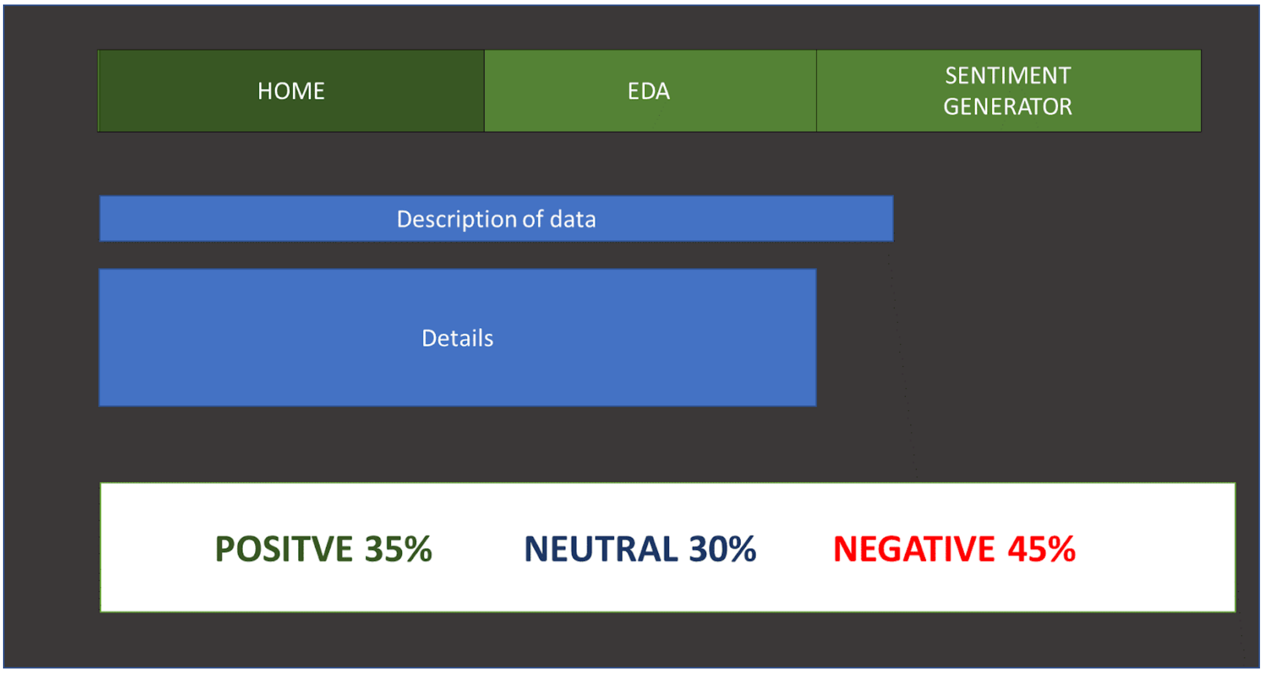


Figure 4.3.13: Design of Home tab

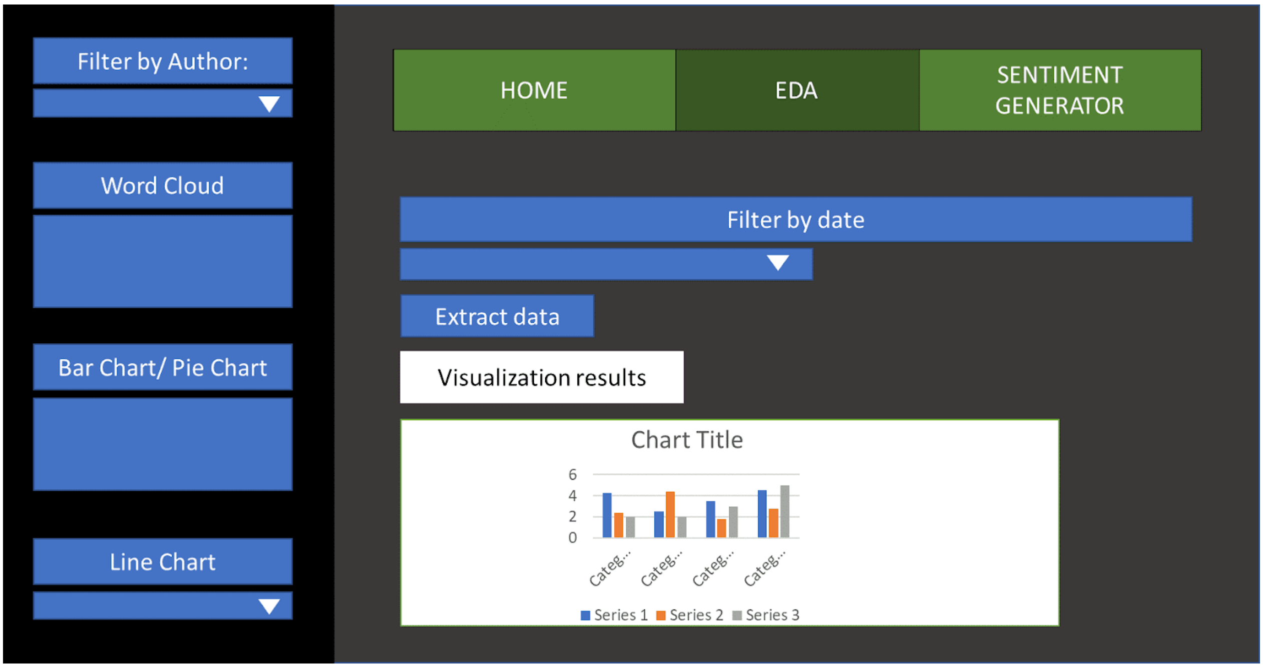


Figure 4.3.14: Design of EDA tab

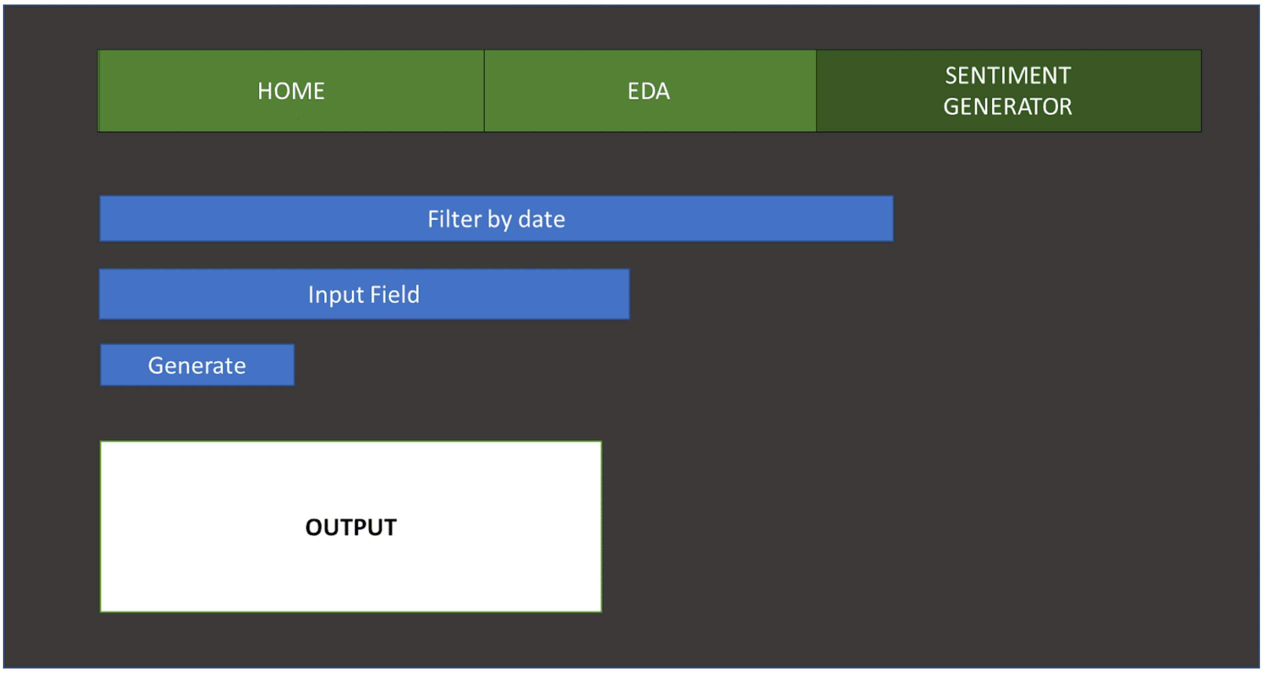
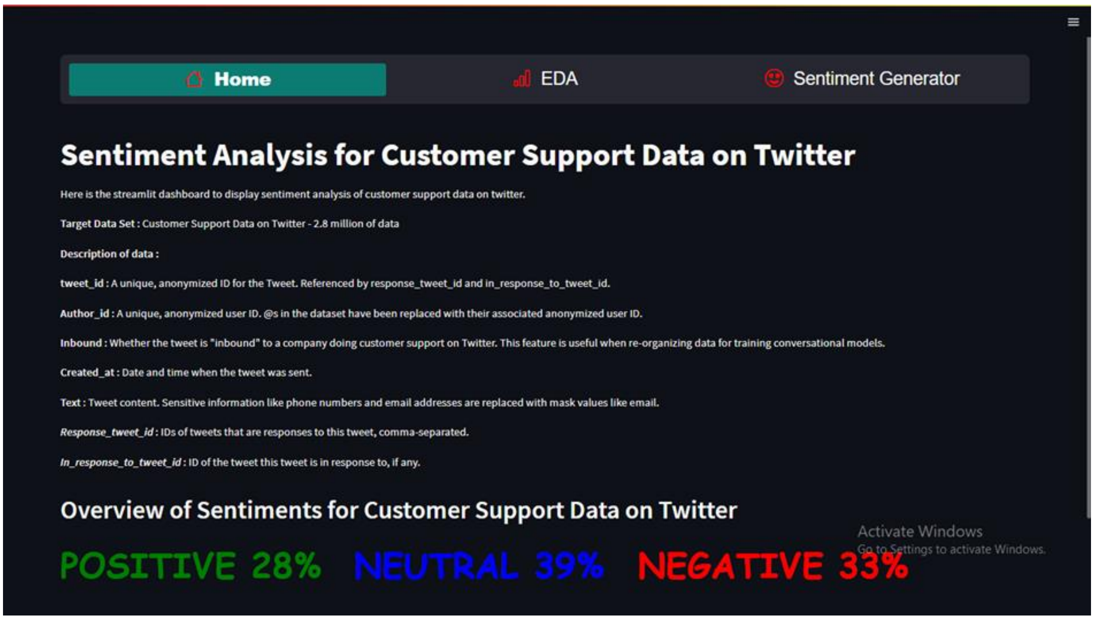


Figure 4.3.15: Design of sentiment generator tab

Final output dashboard

Below figures show the final output dashboard interface. It mainly consists of two parts. They are Side bar and the content layout. Content layout mainly consists of three tabs.

Home tab, EDA tab and Sentiment Generator tab.



Sentiment Analysis for Customer Support Data on Twitter

Here is the streamlit dashboard to display sentiment analysis of customer support data on twitter.

Target Data Set : Customer Support Data on Twitter - 2.8 million of data

Description of data :

tweet_id : A unique, anonymized ID for the Tweet. Referenced by response_tweet_id and in_response_to_tweet_id.

Author_id : A unique, anonymized user ID. @s in the dataset have been replaced with their associated anonymized user ID.

Inbound : Whether the tweet is "inbound" to a company doing customer support on Twitter. This feature is useful when re-organizing data for training conversational models.

Created_at : Date and time when the tweet was sent.

Text : Tweet content. Sensitive information like phone numbers and email addresses are replaced with mask values like email.

***Response_tweet_id* :** IDs of tweets that are responses to this tweet, comma-separated.

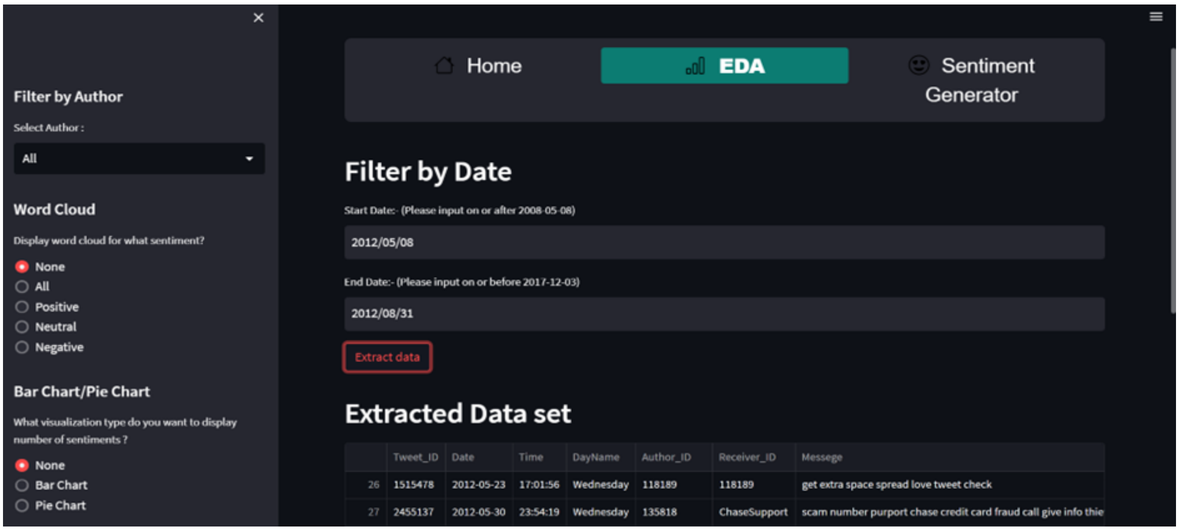
***In_response_to_tweet_id* :** ID of the tweet this tweet is in response to, if any.

Overview of Sentiments for Customer Support Data on Twitter

POSITIVE 28% NEUTRAL 39% NEGATIVE 33%

Figure 4.3.4: Home tab

The Figure displays the Home tab, we also consider it as the welcome page of our dashboard. To this page we added a description of our customer support data on twitter dataset. Also, we added an overview of the overall count of sentiments of our dataset.



Filter by Author

Select Author :

All

Word Cloud

Display word cloud for what sentiment?

● None
○ All
○ Positive
○ Neutral
○ Negative

Bar Chart/Pie Chart

What visualization type do you want to display number of sentiments ?

● None
○ Bar Chart
○ Pie Chart

Filter by Date

Start Date:- (Please input on or after 2008-05-08)

2012/05/08

End Date:- (Please input on or before 2017-12-03)

2012/08/31

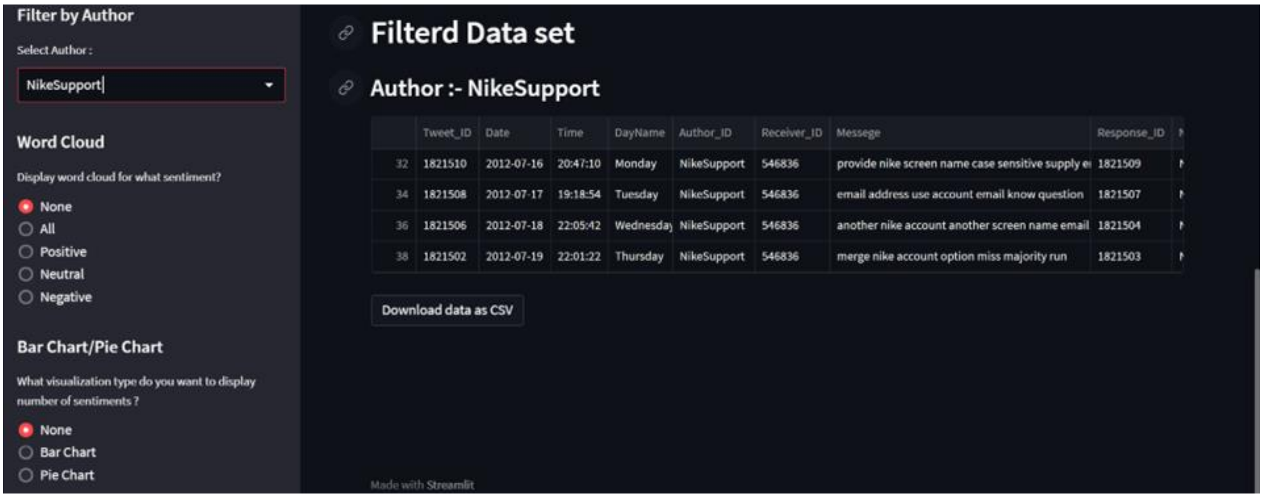
Extracted Data set

	Tweet_ID	Date	Time	DayName	Author_ID	Receiver_ID	Messege
26	1515478	2012-05-23	17:01:56	Wednesday	118189	118189	get extra space spread love tweet check
27	2455137	2012-05-30	23:54:19	Wednesday	135818	ChaseSupport	scam number purport chase credit card fraud call give info thie

Figure 4.3.5: EDA tab

EDA tab, this tab contained the most important analytics parts of our dataset.

At the top of the layout, we can see the option to select date range. By using that option, we can give a start date and the end date to extract some specific data from our dataset. Also, we added download csv option as a button to download the extracted data for more study.



Filter by Author

Select Author :

NikeSupport

Word Cloud

Display word cloud for what sentiment?

● None
○ All
○ Positive
○ Neutral
○ Negative

Bar Chart/Pie Chart

What visualization type do you want to display number of sentiments ?

● None
○ Bar Chart
○ Pie Chart

Filterd Data set

Author :- NikeSupport

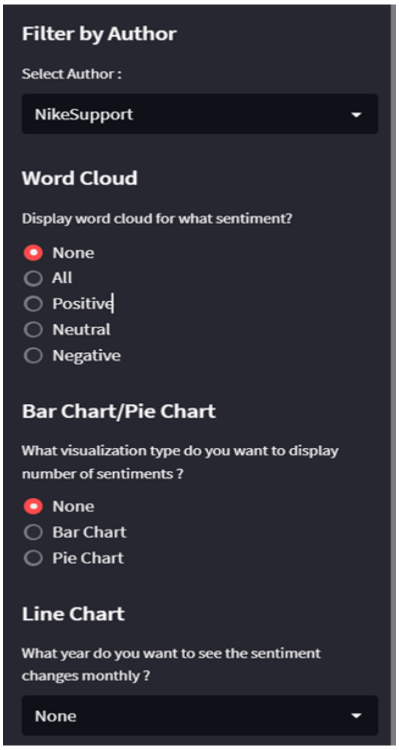
	Tweet_ID	Date	Time	DayName	Author_ID	Receiver_ID	Messege	Response_ID
32	1821510	2012-07-16	20:47:10	Monday	NikeSupport	546836	provide nike screen name case sensitive supply e	1821509
34	1821508	2012-07-17	19:18:54	Tuesday	NikeSupport	546836	email address use account email know question	1821507
36	1821506	2012-07-18	22:05:42	Wednesday	NikeSupport	546836	another nike account another screen name email	1821504
38	1821502	2012-07-19	22:01:22	Thursday	NikeSupport	546836	merge nike account option miss majority run	1821503

Download data as CSV

Made with Streamlit

Figure 4.3.6: Filtered data according to the author.

At the top of the side bar you can see the drop down menu to select specific author. Once you select the author the specific tweets related to that author will be filtered. Output will display in the side layout as table. Also, we added download data as csv option as a button to download the extracted data for more study.



Filter by Author

Select Author :

NikeSupport

Word Cloud

Display word cloud for what sentiment?

● None
○ All
○ Positive
○ Neutral
○ Negative

Bar Chart/Pie Chart

What visualization type do you want to display number of sentiments ?

● None
○ Bar Chart
○ Pie Chart

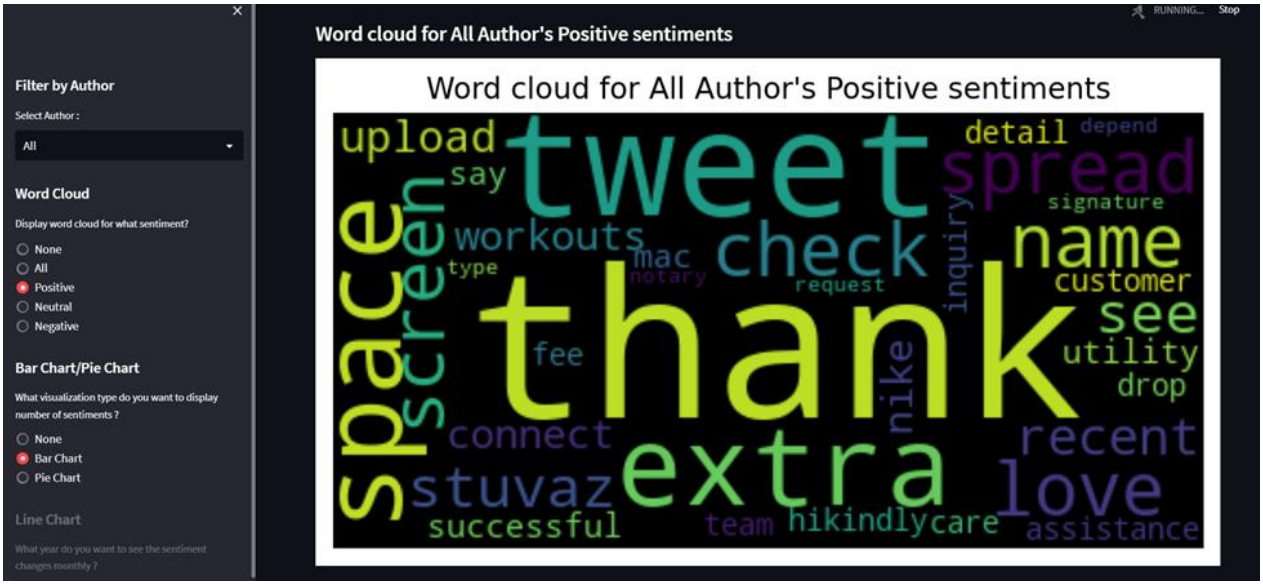
Line Chart

What year do you want to see the sentiment changes monthly ?

None

Figure 4.3.7: Side bar

Figure displays the side bar of the output dashboard. Mainly here we can see three types of visualization. Word Cloud , BarChart/PieChart and Line Chart



Filter by Author

Select Author : All

Word Cloud

Display word cloud for what sentiment?

○ None
○ All
● Positive
○ Neutral
○ Negative

Bar Chart/Pie Chart

What visualization type do you want to display number of sentiments ?

○ None
● Bar Chart
○ Pie Chart

Line Chart

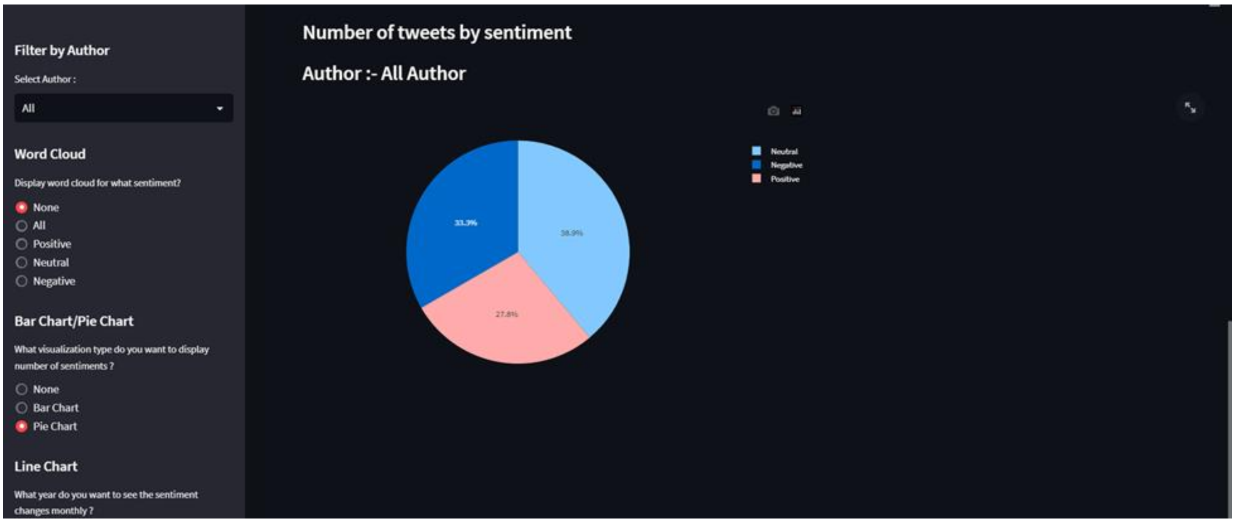
What year do you want to see the sentiment changes monthly ?

Word cloud for All Author's Positive sentiments

Word cloud for All Author's Positive sentiments

Figure 4.3.8: Word cloud

Figure displays example output of word Cloud: This option allows you to display a word cloud for a selected sentiment category (positive, negative, or neutral). A word cloud is a visual representation of the most frequently occurring words in a text, where the size of each word represents its frequency.



Filter by Author

Select Author : All

Word Cloud

Display word cloud for what sentiment?

● None
○ All
○ Positive
○ Neutral
○ Negative

Bar Chart/Pie Chart

What visualization type do you want to display number of sentiments ?

○ None
○ Bar Chart
● Pie Chart

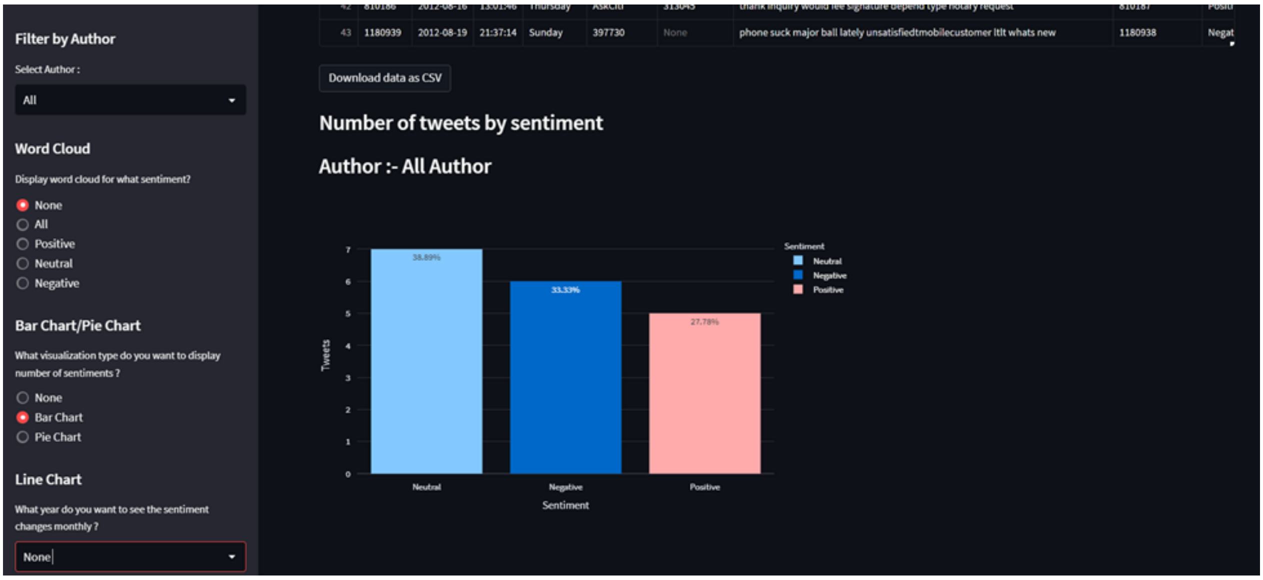
Line Chart

What year do you want to see the sentiment changes monthly ?

Number of tweets by sentiment

Author :- All Author

Figure 4.3.9: Pie Chart



Filter by Author

Select Author :

All

Word Cloud

Display word cloud for what sentiment?

● None
○ All
○ Positive
○ Neutral
○ Negative

Bar Chart/Pie Chart

What visualization type do you want to display number of sentiments ?

○ None
● Bar Chart
○ Pie Chart

Line Chart

What year do you want to see the sentiment changes monthly ?

None

42	810186	2012-08-16	13:01:46	Thursday	AskCiti	313045	thank inquiry would fee signature depend type notary request	810187	Positi
43	1180939	2012-08-19	21:37:14	Sunday	397730	None	phone suck major ball lately unsatisfiedtmobilecustomer ltlt whats new	1180938	Negat

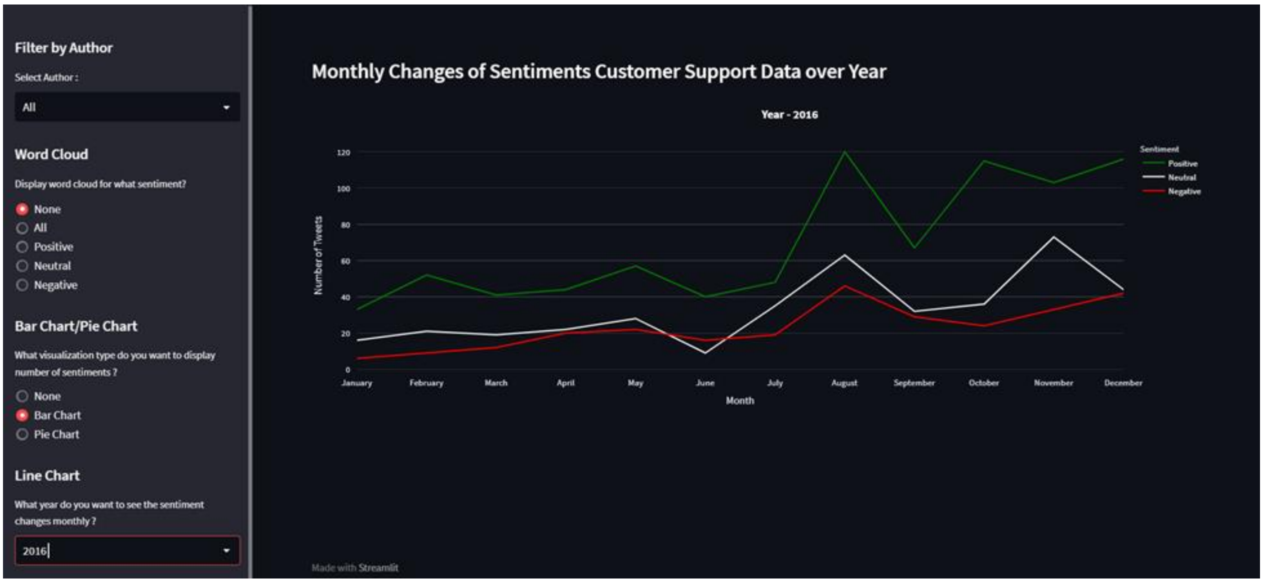
Download data as CSV

Number of tweets by sentiment

Author :- All Author

Figure 4.3.10: Bar Chart

Figure and figure display example output for Number of Sentiments: This option allows you to choose between displaying the number of sentiments as a bar chart or a pie chart. The bar chart shows the number of sentiments for each sentiment category (positive, negative, or neutral) as a bar graph. The pie chart shows the proportion of sentiments for each sentiment category as a pie chart.



Filter by Author

Select Author :

All

Word Cloud

Display word cloud for what sentiment?

● None
○ All
○ Positive
○ Neutral
○ Negative

Bar Chart/Pie Chart

What visualization type do you want to display number of sentiments ?

○ None
● Bar Chart
○ Pie Chart

Line Chart

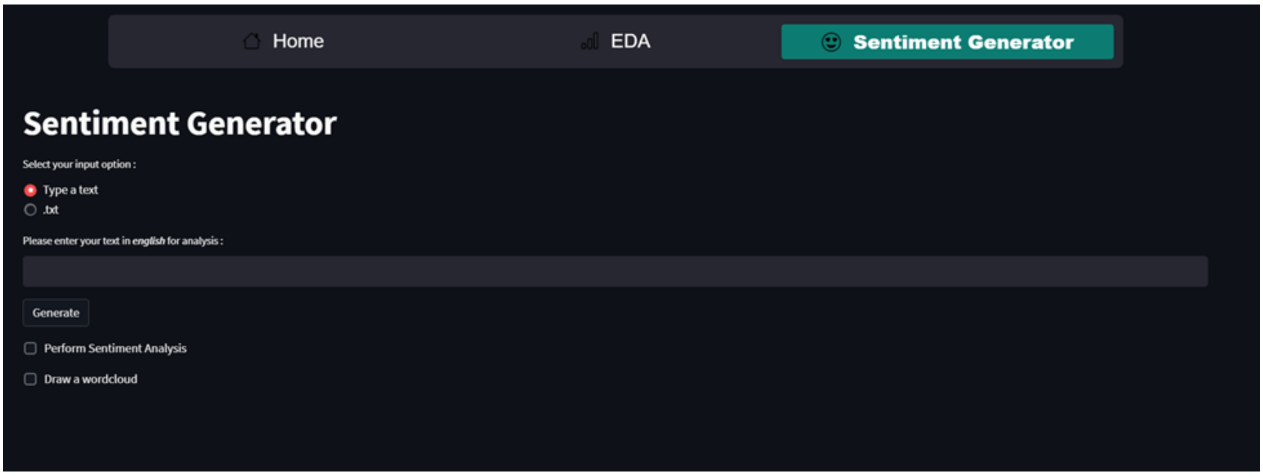
What year do you want to see the sentiment changes monthly ?

2016

Monthly Changes of Sentiments Customer Support Data over Year

Figure 4.3.11: Line Chart

Figure displays example output for line chart: This option allows you to select a year to see the changes in sentiment over time. The sentiment changes are displayed as a line chart, with the sentiment category (positive, negative, or neutral) on the y-axis and the month on the x-axis.



Sentiment Generator

Select your input option :

● Type a text
○ .txt

Please enter your text in *english* for analysis :

□ Perform Sentiment Analysis

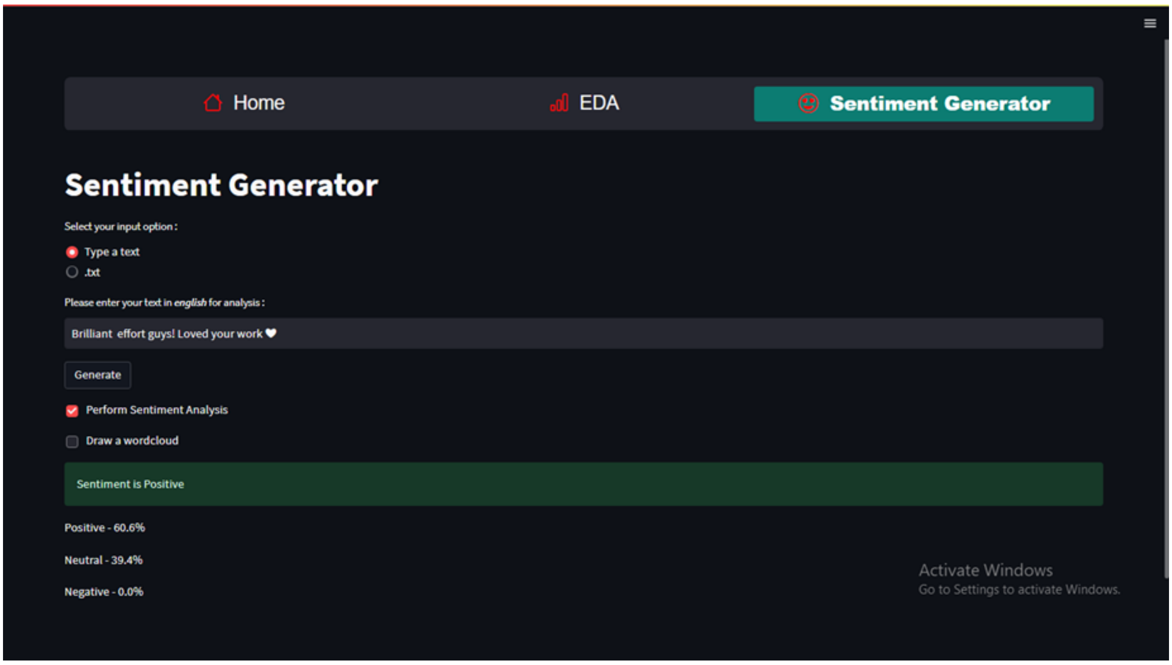
□ Draw a wordcloud

Figure 4.3.12: Sentiment Generator tab

The above figure displays the interface of sentiment generator tab. It is asking you to input text in English for sentiment analysis. You can input the text in the text box provided.

Alternatively, you can also upload a .txt file that contains the text you want to analyze for sentiment. To do this, click on the ".txt" button and select the file from your computer.

Once you have entered the text, click on the "Generate" button and it will ask give you two option as above figure to generate the sentiment analysis result or draw word cloud for given text or txt file. If you click the "Perform sentiment analysis" option, then the sentiment analysis results will provide an evaluation of the sentiment of the input text, categorizing it as positive, negative, or neutral. Shows in Figure If you click Draw a world cloud option, then world cloud will display for given text.



Sentiment Generator

Select your input option :

● Type a text
○ .txt

Please enter your text in *english* for analysis :

Brilliant effort guys! Loved your work ❤

☑ Perform Sentiment Analysis

□ Draw a wordcloud

Sentiment is Positive

Positive - 60.6%

Neutral - 39.4%

Negative - 0.0%

Figure 4.3.13: Example Output for Perform Sentiment Analysis.

CHAPTER 05

This section consists of the conclusions and the future directions of the project.

5.1: Conclusion

Based on our project, it can be concluded that sentiment analysis using the VADER pre-built model in the NLTK library is a useful approach for classifying sentiment in text data. By finding the most accurate range to classify sentiment, we have successfully identified a threshold for determining whether a piece of text expresses a positive, negative, or neutral sentiment.

Using this range to generate sentiment for a new dataset is a valuable application of sentiment analysis, as it allows us to quickly and effectively process large volumes of text data and extract meaningful insights. These insights can be used to inform decision-making in a variety of contexts, from market research to customer service to social media monitoring.

Overall, our project demonstrates the power of machine learning and natural language processing techniques for analyzing text data and extracting meaningful insights. By leveraging pre-built models and developing our own thresholds, we have shown how sentiment analysis can be applied to real-world problems and generate valuable insights for businesses and organizations.

5.2: Future directions

There are several potential future directions for our project on sentiment analysis using the VADER pre-built model in the NLTK library. Here are a some of them:

1. Improving accuracy on specific domains: While we have identified the most accurate range to classify sentiment in text data, there may be opportunities to improve the accuracy of sentiment classification for specific domains or types of text data. For example, we could explore ways to improve the accuracy of sentiment classification for text data related to financial news, or for text data related to customer feedback in specific industries.

2. Exploring other sentiment analysis models: While the VADER pre-built model has demonstrated high accuracy in our project, there are other sentiment analysis models and techniques that we could explore. For example, we could explore the use of deep learning models for sentiment analysis, or investigate alternative pre-built models for sentiment analysis in the NLTK library or other machine learning libraries.

3. Developing customized sentiment analysis models: Depending on our specific use case, it may be valuable to develop a customized sentiment analysis model rather than relying on a pre-built model. This could involve using machine learning techniques to train a model on labeled data specific to our domain or industry, or incorporating external data sources to improve the accuracy of sentiment classification.

4. Developing a custom emotional analysis model: One approach to improving our model's emotional analysis capabilities is to train a custom machine learning model specifically for emotional analysis. This would require a larger dataset with labeled emotional categories but could ultimately provide more accurate emotional analysis results.

CHAPTER 06

This section consists how the group members have contributed as individuals and as a whole.

6.1: Contribution to preprocessing and EDA	
Preprocess the data and perform EDA for customer support data on twitter & final EDA.	Hisham, Pawani, Pasindu.
Separated data between team members according to year.	Pasindu 2013 Hisham 2014 Pawani 2015
Drawn world cloud for each week for selected year and analysis what they are talking about and their sentiments.	Hisham, Pawani, Pasindu.
6.2: Contribution by using nltk library to generate sentiments.	
Manually read tweets in selected year and add sentiments manually for each tweets. Read 86 tweets manually in 2013 Read 100 tweets manually in 2014 Read 100 tweets manually in 2015	2013 Pasindu 2014 Hisham 2015 Pawani
Generated sentiment using NLTK library for different ranges.	Pawani, Hisham, Pasindu

6.3: Contribution to evaluate performance of model.	
Confusion matrix Comparing manual sentiments with nltk generated sentiments for different ranges. Found accuracy using confusion matrix for nltk library for different ranges.	Pasindu -2013 Pawani -2015 Hisham-2014
Precision, recall and f1. Evaluate models for different ranges using precision, recall and f1. find the most accurate range.	Hisham, Pawani
Combine all data together and found accuracy, precision, recall and f1 and find the most accurate range for generate sentiments using nltk.	Hisham, Pawani
6.4: Contribution to Streamlit output dashboard	
Designing phase of streamlit dasboard	
Designed streamlit dashboard	Pawani
Coding phase of streamlit dashboard	
Home page:	Hisham, Pawani

EDA page:	
Extracted data according to date range	Hisham
Extracted data according to specific author	Hisham
For extracted data generate world clouds for given sentiments	Pawani
For extracted data generate pie chart/bar chart to visualized count of sentiments	Pawani
Visualization of Monthly Changes of Sentiments Customer Support Data over Year for selected year by using line chart-	Hisham
Sentiment Generator page	Hisham
6.5: Contribution to the final report	Pawani, Pasindu

APPENDIX

This section include the codes used to implement the project described in this project. https://drive.google.com/drive/folders/13saLKJYdfkakMffE_XA3XlL5lgdlk4_7?usp=share_link

REFERENCES

1. Agung Eddy Suryo Saputro, K. A. (2018). Study of Sentiment of Governor's Election Opinion in 2018 . *International Journal of Scientific Research in Science, Engineering and Technology* , p. 231-238 .
2. Bello, A. N.-C.-F. (2023). A BERT Framework to Sentiment Analysis of Tweets . *Sensors*, 506.
3. Gunasiri, W. (2021). Sentiment Analysis of Tweets to predict Sri Lankan Election Results using Supervised Learning Techniques.
4. Musfira, A. a. (2022). Sentiment Analysis of Tweets Regarding the Sri Lankan Crisis using Automatic Coding in ATLAS.ti 22.
5. Pano, T. a. (2020). A Complete VADER-Based Sentiment Analysis of Bitcoin (BTC) Tweets during the Era of COVID-19 . *Big Data and Cognitive Computing*, 33.
6. Pardeep Kaur, M. E. (2022). Sentimeny Analysis on Electricity Twitter Posts .
7. Rakhmanov, O. (2020). A Comparative Study on Vectorization and Classification Techniques in Sentiment Analysis to Classify Student-Lecturer Comments. *Procedia Computer Science*, 194-204 .