



Islington college
(इस्लिङ्टन कलेज)

Module Code & Module Title
CU6051NI Artificial Intelligence

Assessment Weightage & Type
75% Individual Coursework

Year and Semester
2022-23 Autumn

Student Name: Subriti Aryal
London Met ID: 20049062
College ID: np01cp4s210044
Group: C13

Assignment Due Date: 11th January 2023
Assignment Submission Date: 11th January 2023

*I confirm that I understand my coursework needs to be submitted online via Google Classroom under the relevant module page before the deadline in order for my assignment to be accepted and marked.
I am fully aware that late submissions will be treated as non-submission and a marks of zero will be awarded.*

Acknowledgement

The completion of this coursework required a lot of guidance and assistance and I feel extremely fortunate to have got this all. I would like to express my deepest appreciation to all those who provided me with the possibility to complete this assignment.

Firstly, A special gratitude to our module leader, Mr. Bibek Khanal whose help, suggestions and encouragement helped me in co-ordination and completion of this report.

Secondly, I would like to sincerely thank our tutor Mr. Sarun Dahal for sparing his time to review and help correct our mistakes. I am always extremely grateful for his efforts to clarify our doubts and monitor us throughout.

Also, I owe my gratitude to my friends for supporting me through any situation of doubt and guiding me accordingly. Lastly, I am also grateful to my parents for their continuous support, and to everyone who contributed directly or indirectly towards a successful completion of this report.

Abstract

Artificial intelligence is a broad term for machines that can mimic human intelligence. While, Natural Language Processing comprehend what users say and what they mean to do, Machine Learning delivers more accurate responses by remembering previous learning interactions (Roldós, 2020).

With the help of these categories of AI, a problem domain of online movie reviews was explored, and a system was built through Supervised Learning model. The “IMDB Dataset of 50K Movie Reviews” was found from Kaggle. The analysis of sentiments in the user reviews was done using the Naïve Bayes Classifier method. Final performance metrics yielded an accuracy of 88.4%, precision of 88.7%, recall of 88.1% and F1 score of 88.45%.

Table of Contents

1. Introduction	1
1.1. Explanation of the AI Concept Used	1
1.2. Explanation of the chosen topic/ problem domain.....	3
2. Background.....	5
2.1. Research work done in coursework 1	5
2.1.1. Sentiment Analysis	5
2.1.2. Review and analysis of existing work in the problem domain	7
3. Solution	9
3.1. Approach used for solving the problem.....	9
3.1.1. Elaboration of the AI algorithm used (Naïve Bayes Classifier)	9
3.2. Pseudocode of the solution.....	11
3.3. Diagrammatic representation of the solution.....	12
3.4. Development process	14
3.4.1. Explanation of used tools and technologies.....	14
3.4.2. Explanation of used libraries.....	15
3.4.3. Explanation of the Development Process	16
3.5. Achieved Results	17
3.5.1. Importing Required Libraries	17
3.5.2. Importing the Dataset	17
3.5.3. Data pre-processing	18
3.5.4. Separating Train and Test Values	22
3.5.5. Creating and training the model.....	22
3.5.6. Performance metrics	22
4. Conclusion	29
4.1. Analysis of the work done	29
4.2. How the solution addresses real world problems.....	30
4.3. Further work.....	30

References.....	31
-----------------	----

Table of Figures

Figure 1: Subsets of Artificial Intelligence.....	1
Figure 2: Statistics for adoption of sentiment analysis technology	4
Figure 3: Sentiment Analysis.....	5
Figure 4: Snippet of the Dataset.....	7
Figure 5: Steps involved in data cleaning.....	10
Figure 6: Processes involved in building a sentiment analysis system.....	11
Figure 7: Flowchart.....	12
Figure 8: Flowchart- continued.....	13
Figure 9: Tools and Technologies Used.....	14
Figure 10: Importing Required Libraries	17
Figure 11: Importing the dataset	17
Figure 12: Checking for missing values.....	18
Figure 13: Converting Positive to 1 and Negative sentiment to -1	18
Figure 14: Removal of HTML tags	19
Figure 15: Removal of special characters	19
Figure 16: Converting text to lowercase	20
Figure 17: Removing stopwords.....	20
Figure 18: Stemming words to its root.....	21
Figure 19: Separating Training and Test Values with default vectorizer	21
Figure 20: Creating, fitting, and evaluating the model with default vectorizer.....	21
Figure 21: Separating Training and Test Values with parameterized vectorizer	22
Figure 22: Creating and fitting the model	22
Figure 23: Testing and Evaluating the model	23
Figure 24: Confusion Matrix	24
Figure 25: Plotting confusion matrix using Matplotlib	25
Figure 26: Installing wordcloud.....	26
Figure 27: Negative review word cloud	27
Figure 28: Positive review word cloud.....	28

Table of Tables

Table 1: Confusion Matrix	24
---------------------------------	----

Table of Equations

Equation 1: Bayes Theorem	9
---------------------------------	---

1. Introduction

1.1. Explanation of the AI Concept Used

A wide-ranging branch of computer science known as **Artificial Intelligence** is concerned with building smart machines that can carry out tasks that typically require human intelligence. The capability of the human mind is modelled and even improved upon by machines thanks to advancements in machine learning and deep learning. In this light, artificial intelligence (AI) is defined as "the study of agents that receive perceptions from the environment and perform actions." AI is becoming more and more prevalent in daily life, from the emergence of self-driving cars to the proliferation of smart assistants like Siri and Alexa (Glover, 2022).

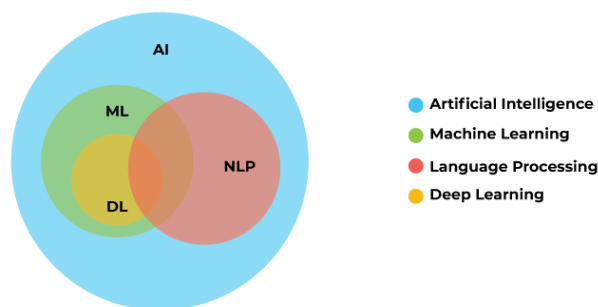


Figure 1: Subsets of Artificial Intelligence
(Source: encora, 2018)

A branch of artificial intelligence known as **Machine Learning (ML)** enables computers to acquire new skills and improve performance over time without having to be explicitly programmed. Numerous crucial algorithms enable machines to compare data, look for patterns, or learn by making mistakes before eventually making accurate predictions without human intervention (Mesevage, 2020).

The branch of machine learning known as **Natural Language Processing (NLP)** assists computers in understanding natural human language by combining both linguistics and computer science. NLP aims to bridge the gap between human language and a computer's command line interface. Since, humans cannot understand machine code consisting of millions of zeros and ones, NLP is very essential to human-computer interactions. Virtual assistants, Spell-checkers (Autocorrections), Autocompletions, Language Translators and many more are all powered by the NLP (Shivanandhan, 2020).

Therefore, systems like AI-powered chatbots learn to carry out tasks on their own and improve with practice when combined with NLP and Machine Learning algorithms (Roldós, 2020). The machine can either learn from a set of labeled datasets supplied in a Supervised Learning model or from a set of unlabeled datasets and identify patterns on its own (Mesevage, 2020).

Supervised Learning:

Supervised Learning refers to the process where the machine learns under supervision. It includes a model that utilizes a labeled dataset to make predictions. Labeled datasets indicate that the right answer has already been assigned to the data. The machine receives this known data and processes it to analyze and learn the association of the features based on certain patterns. Later, using the historical data, the machine can accurately predict the results.

- **Classification**

When the output variable is categorical, i.e., has number of classes and groups, classification is used. For instance, true or false, negative, or positive, etc. These classes can be called as labels/targets. One such example of classification algorithm is Email Spam Detector.

It consists of two types of learners: lazy learners and eager learners. Algorithms like KNN and case-based reasoning fall on the category of lazy learners. They take less time in training but significantly more time for predictions.

Other algorithms like Naïve Bayes, Decision Trees and ANN fall under eager learners and yield less time in prediction as they take more time in learning (JavaTpoint, 2021).

- **Regression**

When the output variable has a real or continuous value, regression is used. A change in one variable is related to a change in the other in this situation because there is a relationship between the two or more variables. For instance, humidity is a dependent variable while temperature is independent. The humidity decreases as the temperature rises.

The model is fed these two variables, and as a result, the computer learns how they relate to one another. Once trained, the machine can accurately predict the humidity based on the temperature.

Unsupervised Learning

Unsupervised learning involves the computer learning on its own while using unlabeled data. In the unlabeled data, the machine looks for patterns and responds.

- **Clustering**

The process of clustering involves grouping objects into clusters that are distinct from objects of another cluster while being similar to one another. Identifying the customers who purchased comparable products, for instance.

- **Association**

Discovering the likelihood that items in a collection will appear together is done using association, a type of rule-based machine learning. Identifying the products that were purchased alongside, for instance (Banoula, 2022).

1.2. Explanation of the chosen topic/ problem domain

In today's digital world, the internet provides a comparatively complete and comprehensive environment for people all over the world to communicate and share information. The way that movie reviews are written is also changing. Cyberspace provides an innovative platform for all movie fans, from traditional critics of experts to general audiences to share their opinions. These audience comments contain sentiment, which serves as a form of movie feedback and are very crucial for the film industry. Positive reviews for a film may help it draw a larger audience. However, positive reviews do not always translate into high box office success, and vice-versa (UKEssays, 2018).

Based on their reviews, sentiment analysis can infer the attitude of the critics. A movie review's sentiment can be analyzed to determine whether it is positive or negative, and this affects the movie's overall score. With this analyzed score, it is easier to conclude which type of content is more enjoyed and consumed by the audience in comparison to the negatively reviewed ones. Additionally for instance, if a movie teaser is released and gains a very negative response, with quick sentiment analysis can the necessary actions be taken as soon as possible.

This project will therefore attempt in analyzing the sentiments in the reviews of the movies in IMDB using the Naïve Bayes Classifier method.

Some Facts and Figures

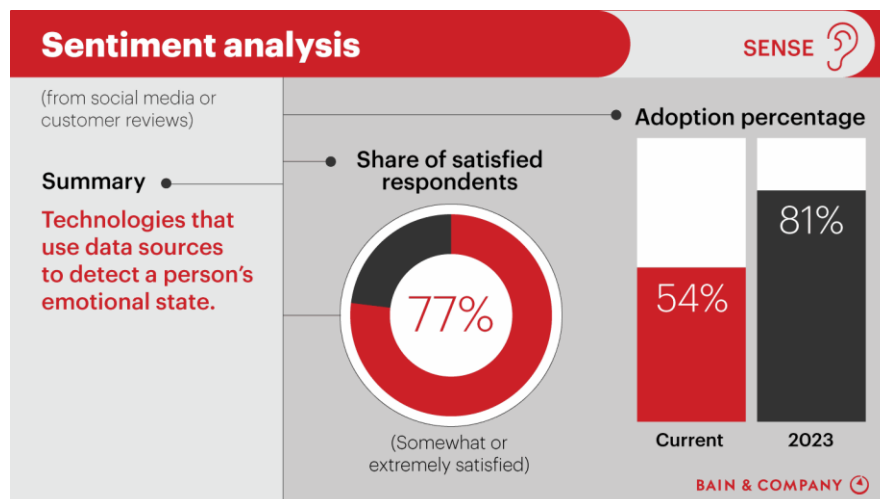


Figure 2: Statistics for adoption of sentiment analysis technology
(Source: Bain & Company, 2020)

- In 2020, 54% of businesses claimed to have implemented technologies for analyzing customer sentiment from online reviews or social media, with that number expected to rise to 80% by 2023.
- In correlation to the entertainment industry, customers are more satisfied (91%) with suggestions based on their sentiments from prior experiences than with random recommendation tools, which only achieve 65% of the customer satisfaction.
- The region with the largest revenue share in the global market for emotion detection is North America.
- Sentiment analysis uses machine learning algorithms that can identify fake reviews with 85% accuracy (Yilmaz, 2022).

2. Background

2.1. Research work done in coursework 1

2.1.1. Sentiment Analysis

Sentiment analysis also referred to as Opinion mining is a method for determining the polarity of a text using natural language processing. It is an automated process for categorizing data into positive, negative, and neutral sentiments. Some sentiment analysis goes further by classifying a text with more specific sentiment markers like disappointment, excitement, or disgust.

Online reviews, emails, chats with customer service representatives, survey responses, blogs, and news articles are a few examples of the text data collected from social media that is frequently subjected to sentiment analytics (Bredava, 2022).

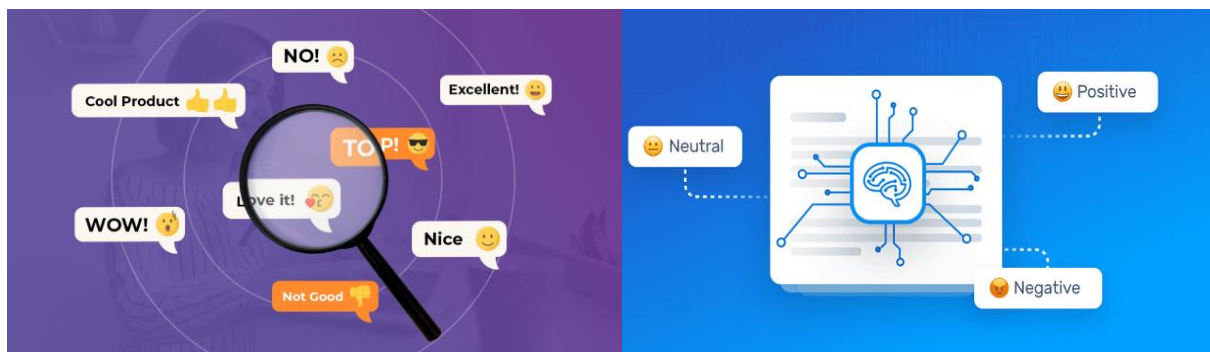


Figure 3: Sentiment Analysis
(Source: fireflies.ai, 2022 and MonkeyLearn, 2020)

Sentiment analysis enables large-scale, real-time data processing. It is also used to automate business processes, gain insights for data-driven decisions, and automatically understand how people are talking about a particular topic (Pascual, 2022). Numerous commercial processes, including brand monitoring, product analytics, customer service, and market research, can also benefit from sentiment analysis. Leading companies can work more quickly, more accurately, and toward more beneficial goals by integrating it into their current systems and analytics.

Sentiment analysis is now more than just a cool, high-tech trend; it is quickly becoming a vital tool for all contemporary businesses. Ultimately, sentiment analysis helps us gain fresh perspectives, better understand our clients, and more effectively empower related teams to produce better work (MonkeyLearn, 2022).

Advantages of the Topic/Problem Domain

There are several advantages to using sentiment analysis, some of which include:

- Analysis on the engagement of people with certain type of contents can be found easily by examining the user reviews.
- Analysis on the feedback and reviews of people can help get insights on their likes and dislikes and help find a group of people with similar tastes.
- Real-time support ticket analysis can be used to spot unhappy customers and take appropriate action to reduce customer complaints (Pascual, 2022).
- Based on analyzed previous reviews, a list of movie recommendations can be created for the viewers (Goyal & Parulekar, 2015).

Challenges for Sentiment Analysis

One of the most challenging tasks in natural language processing is sentiment analysis because even humans have difficulty doing it correctly. Some of the main difficulties with machine-based sentiment analysis are:

- The subjectivity and tone of statements make it challenging to classify.
- Also, the context and polarity of the domain may change the sentiment analysis altogether.
- People express sarcasm and ironies for negative sentiments with positive words. Hence, making the learning process difficult for machines.
- Additionally, the use of western (e.g., :D) and eastern emojis (e.g., 🙄, 😏, 😊) greatly influence the sentiments of the text (MonkeyLearn, 2022).

Chosen Dataset

The dataset for the proposed work on sentiment analysis was searched on multiple platforms, including Kaggle and Hugging Face. The chosen data set was found on Kaggle and is called IMDB Dataset of 50K Movie Reviews. This data set was developed by Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang and Andrew Y. Ng and Christopher Potts. It only consists of one feature attribute consisting of the user review and its corresponding sentiment label (Maas, et al., 2011).

	A	B
1	review	sentiment
2	One of the other reviewers has mentioned that after watching	positive
3	A wonderful little production. The filming techniqu	positive
4	I thought this was a wonderful way to spend time on a too hot s	positive
5	Basically there's a family where a little boy (Jake) thinks there's	negative
6	Petter Mattei's "Love in the Time of Money" is a visually stunnin	positive
7	Probably my all-time favorite movie, a story of selflessness, sac	positive
8	I sure would like to see a resurrection of a up dated Seahunt se	positive
9	This show was an amazing, fresh & innovative idea in the 70's w	negative
10	Encouraged by the positive comments about this film on here I	negative

Figure 4: Snippet of the Dataset

2.1.2. Review and analysis of existing work in the problem domain

There are more than 55,700 academic articles, papers, theses, books, and abstracts on the topic of sentiment analysis (MonkeyLearn, 2022).

1) Sentiment Analysis of IMDb Movie Reviews

Author: Avi Ajmera

Journal: International Journal for Research in Applied Science & Engineering Technology (IJRASET)

This journal article works with the Stanford University-provided IMDB Movie Review dataset. It uses a standardized N-gram approach and a comparison between various methods to find the best classifier. The approach based on BagOfWords yielded a higher accuracy of 84.4% for the Support vector machine classifier while TFIDF yielded a higher accuracy score of 75.1% for the Stochastic Gradient Descent (SGD). Similarly, the Word2Vec yielded a higher accuracy of 82.2% for SGD (Ajmera, 2022).

2) Sentiment Analysis of Movie Reviews Using Logistic Regression

Author: Furqan Lodhi

This article explains the finding on the sentiment analysis of movie reviews using feature extraction, and logistic regression. It includes steps like collection of data, preprocessing and feature extraction of the data. Implementation of logistic regression is done by training and testing the model. The visualization is done with respect to the number of epochs. Finally, comparison of results is done with Python's scikit Learn Library. The article uses the IMDB Dataset of 50K Movie Reviews. The outcome of the research shows similar accuracy of 72% and 73%

for Logistic Regression classifier without the use of Scikit Learn in comparison to the use of Scikit Learn for Logistic Regression classifier (Lodhi, 2021).

3) Sentiment Analysis of Movie Review using data Analytics Techniques

Authors: H. SWATHI, S.S. ARAVINTH, V. NIVETHITHA, T. SARANYA, R. NIVETHANANDHINI

Journal: MAR 2019 | IRE Journals | Volume 2 Issue 9 | ISSN: 2456-8880

This journal article works with the IMDB Dataset of 50K Movie Reviews. It classifies the polarity of the movie on a scale of 0 to 4 by following a lexical approach using the SentiWordNet. It performs feature extraction and uses those features to train the multi-label classifier to predict the correct labels for the movie reviews. Many algorithms like KNN, Naïve Bayes and Random Forest were used. The approach based on structured N-grams and classification techniques (Naïve Bayes and SVM) yielded an accuracy of 88.96% (H., et al., 2019).

4) Improved Accuracy of Sentiment Analysis Movie Review Using Support Vector Machine Based Information Gain

Authors: Reza Maulana, Panny Agustia Rahayuningsih, Windi Irmayani, Dedi Saputra, Wanty Eka Jayanti

Journal: Journal of Physics: Conference Series | 1742-6596 | 1641 | 012060

This journal article shows the findings on sentiment analysis using Support Vector Machines. It uses two different datasets namely Cornell and Stanford datasets. SVM based on Information Gain ultimately boosted the accuracy of the model on the Cornell dataset from 83.05% to 85.65% while only a slight increase of 86.46% to 86.62% accuracy was observed on the Stanford dataset (Reza, et al., 2020).

5) Movies Reviews Sentiment Analysis and Classification

Authors: Mais Yasen, Sara Tedmori

This research paper works with the IMDB reviews dataset containing 42926 reviews. It explores various classification techniques with 5 evaluation metrics to find the one with best performance. The report concluded that Random Forest yielded the highest accuracy score of 96.01% while Ripper Rule Learning performed the worst with only 79.51% accuracy according to the evaluation metric's results (Yasen & Tedmori, 2019).

Summarization of analysis

These studies covered the use of Naïve Bayes, Logistic Regressions, KNN, Random Forest and other classifiers for categorizing reviews, with the use of various feature extraction methods. Few papers used multiple algorithms and feature extraction methods to compare the accuracy scores while a few stucked to one. Multiple approaches like lexical, n-grams and negation handling were also used.

One significant finding in these papers was the exclusion of a neutral category in classification. This is on the grounds that neutral texts are disproportionately challenging to categorize because they are close to the boundary of the binary classifiers (Goyal & Parulekar, 2015).

3. Solution

3.1. Approach used for solving the problem

A supervised learning approach was followed for solving of the problem. Since, sentiment analysis is a very good example for classification in text analysis, the algorithm used here was the Naïve Bayes Classifier. Development works were carried out using Anaconda Navigator's Environment with Jupyter Notebook as the IDE. Various open-source libraries like sci-kit learn, NumPy, pandas, NLTK and Regular expression (re) and matplotlib were used for preparing the solution.

3.1.1. Elaboration of the AI algorithm used (Naïve Bayes Classifier)

The sentiment analysis system was built using the Naïve Bayes Classifier. It is a classification algorithm that follows a probabilistic approach. Because they are probabilistic, they determine the probabilities of each tag for a given text and output the tag with the highest probability. They calculate these probabilities by applying the Bayes' Theorem, which estimates a feature's likelihood based on prior knowledge of circumstances that might be related to a feature. Despite the significant advancements in machine learning over the past few years, Naïve Bayes has proven to be quick, accurate, and dependable. It has been successfully applied to a variety of challenges, but it excels at resolving natural language processing (NLP) problems (Stecanella, 2017). The algorithm we'll be using is called **Multinomial Naive Bayes**.

$$P(y/X) = \frac{P(X/y) * P(y)}{P(X)}$$

Equation 1: Bayes Theorem

Processes involved in building a sentiment analysis system:

I. Text Pre-Processing

This process includes the analyzation and cleaning of the data before fitting into the ML models. Specifically, HTML tags and special characters are removed, all the reviews are converted to lowercase and stop words like (is, for the) are also removed. Finally, stemming is performed to remove words with different forms like (review-ing, review-ed) to its stem word “review”.

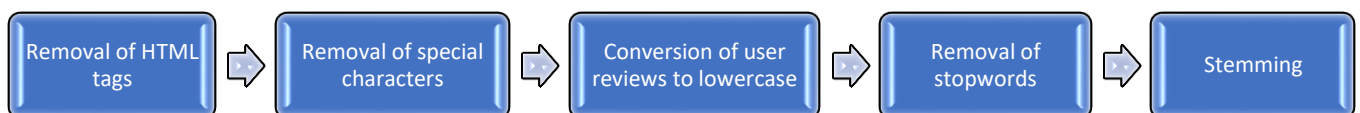


Figure 5: Steps involved in data cleaning

II. Feature engineering/ Vectorize

It is the choosing of appropriate features for building a machine learning model. The data points extracted from the text and provided to the algorithm for its learning are known as features (Stecanella, 2017). This conversion of textual reviews into numerical data/features is done using a **Bag of words** approach. It assigns a column name to each unique word in the dataset and stores the frequency of each word for each row of a user review (Chaudhury, 2020). During development, the data was vectorized using the library **TfidfVectorizer**. Next, data is split into the training and test set.

III. Creating a Machine Learning Model

The library *sklearn* is imported for model creation. The Multinomial NB model is then created, and the data is fit into them.

Model fitting refers to the process of running an algorithm on data with labeled variables resulting in a machine learning model. The model's ability to generalize similar data that it was trained on is measured by how well it fits a model.

Fitting is the process of changing a model's parameters to increase accuracy. The accuracy of the model is evaluated by comparing its results to the target variable's actual, observed values. Also, when given unknown inputs, a model with a good model fit accurately predicts the output (Educative Answers Team, 2022).

Hence, based on accuracy, precision, recall, F1 score, and confusion matrix will the performance of the model be evaluated.

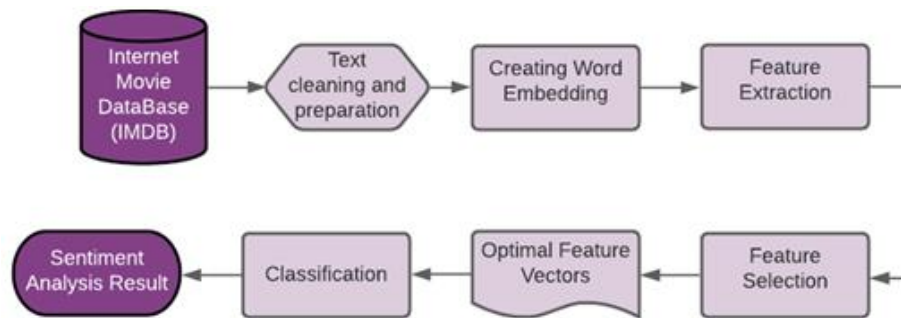


Figure 6: Processes involved in building a sentiment analysis system

3.2. Pseudocode of the solution

START

IMPORT libraries

IMPORT dataset

ANALYSE dataset

CONVERT dataset labels to 1(positive) and -1(negative)

PRE-PROCESS dataset

REMOVE HTML tags

REMOVE special characters

CONVERT text to lowercase

REMOVE stop words

STEM words

VECTORIZE dataset

SPLIT dataset into train and test data

CREATE MultinomialNB model

TRAIN the model by fitting the train dataset

TEST the model with test dataset

CALCULATE accuracy, precision, F1 and recall score of the model

CREATE confusion matrix

CREATE positive and negative word clouds

END

3.3. Diagrammatic representation of the solution

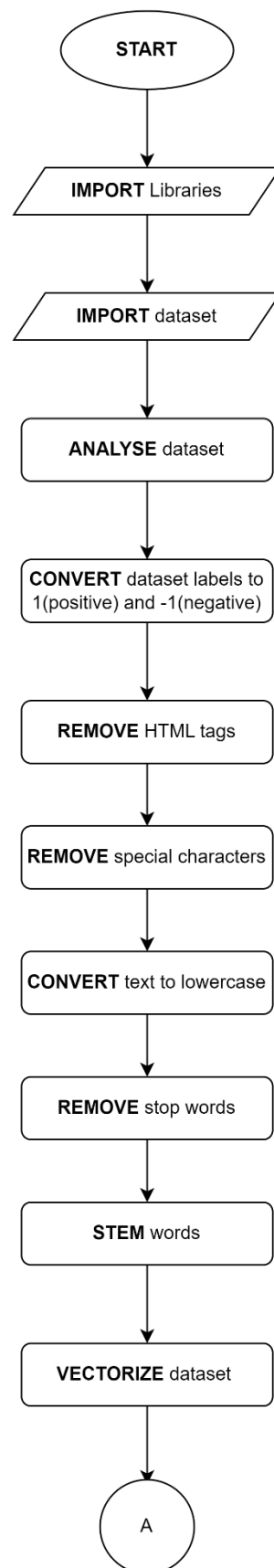


Figure 7: Flowchart

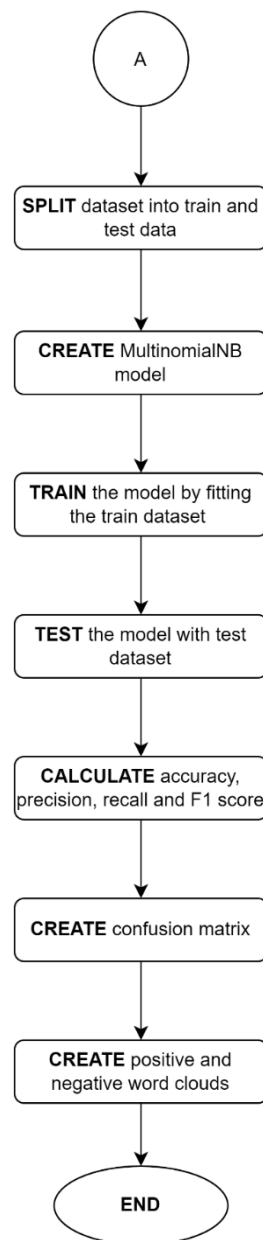


Figure 8: Flowchart- continued

The above flowchart represents the step wise flow of the solution. The importing of required libraries and the dataset were the first steps for development. The data was then analyzed, and the sentiment labels were replaced by 1(positive) and -1(negative). Then the data was pre-processed to remove html tags, and special characters. The data was converted to lower case, stop words were removed and the words were stemmed to its root. The data was finally vectorized to a numerical matrix.

Then the dataset was split into train and test sets. Multinomial Naïve Bayes model was created and trained using the train dataset. Finally, the model was tested using the test dataset and was evaluated based on multiple performance metrics to check the

efficiency and learning of the model. Confusion matrix and word clouds for positive and negative sentiment words were also created for further evaluation.

3.4. Development process

3.4.1. Explanation of used tools and technologies

Anaconda Navigator:

It is a software tool allowing to easily launch applications written in Python or R, manage conda packages, channels, and environments with its desktop friendly graphical user interface (Anaconda.Documentation, 2017).

Jupyter Notebook:

The Jupyter Notebook is a web browser application that allows developing Python based Data Science applications. It was used because it illustrates all the process followed (code, texts, outputs, images) for the solution in a step-by-step manner (tutorialspoint, 2019).



*Figure 9: Tools and Technologies Used
(Sources: UNL, 2017 and Datacamp, 2020)*

Python

It is an efficient object-oriented programming language that has pre-built libraries for ML and AI hence saving time of having to code from scratch. It has an easy-to-understand syntax hence allowing to focus on the complex systems with ease. It is flexible, platform independent and has a wide community (djangostars, 2022). Hence it was used for this project.

Microsoft Excel

It is a software program that uses spreadsheets to organize and analyse stored data (Rosenberg, 2022). The IMDB dataset of 50K movie reviews were stored in the excel sheet and later used from the program.

3.4.2. Explanation of used libraries

Libraries used for working with datasets:

- **NumPy:**

Numerical Python (NumPy) was used to easily operate and perform high-level mathematical functions on multi-dimensional arrays.

- **Pandas:**

It was used to analyze and manipulate data. It also allowed easy access to the matplotlib's and NumPy's methods with lesser code. It was used to visualize data in a DataFrame.

- **Matplotlib:**

It was used for creating word clouds for positive review and negative review words.

- **WordCloud:**

It was used to signify the frequency or importance of each word in a cloud of certain size.

Libraries used for pre-processing:

- **NLTK (Natural Language Toolkit):**

It is the leading library used for pre-processing and working with natural human language data. It is a Natural Language Processing Library with a wide range of text processing libraries for tasks like classification and stemming.

- **Stopwords from nltk. corpus:**

This library was used for finding stop words from the series of user reviews. Since, stop words are present in both negative and positive sentiment-based reviews and aren't of any value to the classification, it was removed to increase the efficiency of the model.

- **Word_tokenize from nltk. tokenize:** This library helps extract syllables from each word. This was used at the time of stopwords removal.

- **SnowballStemmer from nltk. stem:**

This library was used for stemming words having different forms to its root word.

- **Regular expression (re):**

This library was used for extracting strings having a certain pattern. Practically used in the project to remove HTML tags and special characters.

Library used for vectorizing data

- **TfidfVectorizer:**

In Term Frequency-Inverse Document Frequency Vectorizer (TF-IDF) library, the word counts are weighted by a measure of how frequently they appear in the documents. As a result, the authenticity of the rare words that contain a lot of information about a document was also preserved (Chaudhury, 2020).

Library used for creating a Machine Learning Model

- **Sklearn:**

This library was used for ML algorithms and model creation. The Multinomial NB model was created, and the data was fit into them. Also, various performance metrics like `accuracy_score`, `precision_score`, `recall_score`, `f1_score` and `confusion_matrix` were imported from *sklearn.metrics*.

3.4.3. Explanation of the Development Process

After having completed enough research on different problem domains, sentiment analysis was chosen with Naïve Bayes as the algorithm for developing the solution. Naïve Bayes was selected because it performs very well on the textual data. The rough blueprint for undertaking the solution was formulated during coursework 1 which served as the basis for the developments completed during this coursework. With respect to other research done on the problem domain, the required tools, and libraries to bring out a solution were well understood.

The development work began with importing required libraries stated above. Then the IMDB dataset with 50K movie reviews was imported and analysed. Missing values were checked and upon confirmation, the sentiments (positive and negative) were replaced by integers 1 and -1 respectively. Pre-processing of the data was done by removing HTML tags and special characters. Furthermore, the data was converted to lowercase, stop words were removed and words were stemmed to its root. The data was then vectorized using TfidfVectorizer with both default and custom parameters. Then the dataset was split into train and test sets in the ratio of 70:30. The train dataset was fit into the Multinomial Naïve Bayes model. Finally, the algorithm and the model were evaluated based on multiple performance metrics and word cloud for positive and negative sentiment words were created.

3.5. Achieved Results

3.5.1. Importing Required Libraries

First step of the solution was to import all the libraries required to run the application.

```
In [1]: #Data processing
import numpy as np
import pandas as pd

#NLTK
import nltk
from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize
from nltk.stem import SnowballStemmer

#for regex
import re

#Bag of Words
from sklearn.feature_extraction.text import TfidfVectorizer

#Model Creation
import sklearn
from sklearn.model_selection import train_test_split
from sklearn.naive_bayes import MultinomialNB
from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score, confusion_matrix

#WordCloud
import matplotlib.pyplot as plt
from wordcloud import WordCloud
```

Figure 10: Importing Required Libraries

3.5.2. Importing the Dataset

The dataset from the csv file was imported into a pandas DataFrame.

```
In [2]: #importing dataset
imdb_data= pd.read_csv('IMDB Dataset.csv')

#returns number of rows(50000) and columns(2)
print(imdb_data.shape)

#returns top 10 data
imdb_data.head(10)
```

(50000, 2)

Out[2]:

	review	sentiment
0	One of the other reviewers has mentioned that ...	positive
1	A wonderful little production. The...	positive
2	I thought this was a wonderful way to spend ti...	positive
3	Basically there's a family where a little boy ...	negative
4	Petter Mattei's "Love in the Time of Money" is...	positive
5	Probably my all-time favorite movie, a story o...	positive
6	I sure would like to see a resurrection of a u...	positive
7	This show was an amazing, fresh & innovative i...	negative
8	Encouraged by the positive comments about this...	negative
9	If you like original gut wrenching laughter yo...	positive

Figure 11: Importing the dataset

3.5.3. Data pre-processing

Checking for missing values in the data

Information about the DataFrame i.e., number of rows and column, datatype of each column and number of non-null values in each column is extracted.

Out of 50000 total rows, no null values were found.

```
In [3]: #checking for missing values in the data
imdb_data.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 50000 entries, 0 to 49999
Data columns (total 2 columns):
#   Column      Non-Null Count  Dtype
---  ---
0   review      50000 non-null   object
1   sentiment   50000 non-null   object
dtypes: object(2)
memory usage: 781.4+ KB
```

Figure 12: Checking for missing values

Converting Positive to 1 and Negative sentiment to -1

Since, no null values were present in the data, positive sentiments were replaced by 1 and negative sentiments by -1.

```
In [4]: #since no null data is present,
#converting positive sentiments to 1 and negatives to -1
imdb_data.sentiment.replace('positive',1,inplace=True)
#inplace=True specifies to modify the existing DataFrame
#rather than to create a new one
imdb_data.sentiment.replace('negative',-1,inplace=True)
imdb_data.head(10)
```

Out[4]:

	review	sentiment
0	One of the other reviewers has mentioned that ...	1
1	A wonderful little production. The...	1
2	I thought this was a wonderful way to spend ti...	1
3	Basically there's a family where a little boy ...	-1
4	Petter Mattei's "Love in the Time of Money" is...	1
5	Probably my all-time favorite movie, a story o...	1
6	I sure would like to see a resurrection of a u...	1
7	This show was an amazing, fresh & innovative i...	-1
8	Encouraged by the positive comments about this...	-1
9	If you like original gut wrenching laughter yo...	1

Figure 13: Converting Positive to 1 and Negative sentiment to -1

Removal of HTML tags

HTML tags were removed from the data by using a regular expression.

```
In [5]: #removal of HTML tags
def cleanHTMLtags(text):
    cleaned = re.compile(r'<.*?>')
    return re.sub(cleaned, '', text)

#uncleaned data
print(imdb_data.review[0])
imdb_data.review = imdb_data.review.apply(cleanHTMLtags)
#clean data
imdb_data.review[0]
```

One of the other reviewers has mentioned that after watching just 1 Oz episode you'll be hooked. They are right, as this is exactly what happened with me. The first thing that struck me about Oz was its brutality and unflinching scenes of violence, which set in right from the word GO. Trust me, this is not a show for the faint hearted or timid. This show pulls no punches with regards to drugs, sex or violence. It is hardcore, in the classic use of the word. It is called Oz as that is the nickname given to the Oswald Maximum Security State Penitentiary. It focuses mainly on Emerald City, an experimental section of the prison where all the cells have glass fronts and face inwards, so privacy is not high on the agenda. Emerald City is home to many..Aryans, Muslims, gangstas, Latinos, Christians, Italians, Irish and more....so scuffles, death stares, dodgy dealings and shady agreements are never far away. I would say the main appeal of the show is due to the fact that it goes where other shows wouldn't dare. Forget pretty pictures painted for mainstream audiences, forget charm, forget romance...Oz doesn't mess around. The first episode I ever saw struck me as so nasty it was surreal, I couldn't say I was ready for it, but as I watched more, I developed a taste for Oz, and got accustomed to the high levels of graphic violence. Not just violence, but injustice (crooked guards who'll be sold out for a nickel, inmates who'll kill on order and get away with it, well mannered, middle class inmates being turned into prison bitches due to their lack of street skills or prison experience) Watching Oz, you may become comfortable with what is uncomfortable viewing....thats if you can get in touch with your darker side.

Out[5]: "One of the other reviewers has mentioned that after watching just 1 Oz episode you'll be hooked. They are right, as this is exactly what happened with me. The first thing that struck me about Oz was its brutality and unflinching scenes of violence, which set in right from the word GO. Trust me, this is not a show for the faint hearted or timid. This show pulls no punches with regards to drugs, sex or violence. It is hardcore, in the classic use of the word. It is called Oz as that is the nickn

Figure 14: Removal of HTML tags

Removing special characters

Special characters were removed from the data by finding such characters and replacing them with an empty string ''.

```
In [6]: print(imdb_data.review[1])

#removing special characters: punctuations and non-alphanumeric
def removeSpecialCharacters(text):
    for i in text:
        if i.isalnum()==False and i != " ":
            text=text.replace(i,'')
    return text

imdb_data.review = imdb_data.review.apply(removeSpecialCharacters)
imdb_data.review[1]
```

A wonderful little production. The filming technique is very unassuming- very old-time-BBC fashion and gives a comforting, and sometimes discomforting, sense of realism to the entire piece. The actors are extremely well chosen- Michael Sheen not only has got all the polari but he has all the voices down pat too! You can truly see the seamless editing guided by the references to Williams' diary entries, not only is it well worth the watching but it is a terrifically written and performed piece. A masterful production about one of the great masters of comedy and his life. The realism really comes home with the little things: the fantasy of the guard which, rather than use the traditional 'dream' techniques remains solid then disappears. It plays on our knowledge and our senses, particularly with the scenes concerning Orton and Halliwell and the sets (particularly of their flat with Halliwell's murals decorating every surface) are terribly well done.

Out[6]: 'A wonderful little production The filming technique is very unassuming very oldtimeBBC fashion and gives a comforting and sometimes discomforting sense of realism to the entire piece The actors are extremely well chosen Michael Sheen not only has got all the polari but he has all the voices down pat too You can truly see the seamless editing guided by the references to Williams diary entries not only is it well worth the watching but it is a terrifically written and performed piece A masterful production about one of the great masters of comedy and his life The realism really comes home with the little things the fantasy of the guard which rather than use the traditional dream techniques remains solid then disappears It plays on our knowledge and our senses particularly with the scenes concerning Orton and Halliwell and the sets particularly of their flat with Halliwell's murals decorating every surface are terribly well done'

Figure 15: Removal of special characters

Converting all text to lower case

All reviews were converted to lower case using the `.lower()` function.

```
In [7]: #converting all texts to lower case
def convertToLower(text):
    return text.lower()

imdb_data.review = imdb_data.review.apply(convertToLower)
imdb_data.review[0]
```

Out[7]: 'one of the other reviewers has mentioned that after watching just 1 oz episode youll be hooked they are right as this is exactly what happened with methe first thing that struck me about oz was its brutality and unflinching scenes of violence which set in right from the word go trust me this is not a show for the faint hearted or timid this show pulls no punches with regards to drugs sex or violence its is hardcore in the classic use of the wordit is called oz as that is the nickname given to the oswald maximum security state penitentiary it focuses mainly on emerald city an experimental section of the prison where all the cells have glass fronts and face inwards so privacy is not high on the agenda em city is home to manyaryans muslims gangstas latinos christians italians irish and moreso scuffles death stares dodgy dealings and shady agreements are never far awayi would say the main appeal of the show is due to the fact that it goes where other shows wouldnt dare forget pretty pictures painted for mainstream audiences forget charm forget romanceoz doesnt mess around the first episode i ever saw struck me as so nasty it was surreal i couldnt say i was ready for it but as i watched more i developed a taste for oz and got accustomed to the high levels of graphic violence not just violence but injustice crooked guards wholl be sold out for a nickel inmates wholl kill on order and get away with it well mannered middle class inmates being turned into prison bitches due to their lack of street skills or prison experience watching oz you may become comfortable with what is uncomfortable viewingthats if you can get in touch with your darker side'

Figure 16: Converting text to lowercase

Removing stopwords

Even after importing all the required libraries, additional downloads were required. The reviews were looped through to discard all the stopwords.

```
In [8]: #removing stop words like "the", "in", "for", "where", "to", etc
nltk.download('stopwords')
nltk.download('punkt')
def removeStopwords(text):
    stop_words = set(stopwords.words('english'))
    #word_tokenize extracts syllables from each word
    words = word_tokenize(text)
    return [w for w in words if w not in stop_words]

imdb_data.review = imdb_data.review.apply(removeStopwords)
imdb_data.review[0]
```

[nltk_data] Downloading package stopwords to
[nltk_data] C:\Users\Subriti\AppData\Roaming\nltk_data...
[nltk_data] Package stopwords is already up-to-date!
[nltk_data] Downloading package punkt to
[nltk_data] C:\Users\Subriti\AppData\Roaming\nltk_data...
[nltk_data] Package punkt is already up-to-date!

Out[8]: ['one',
'reviewers',
'mentioned',
'watching',
'1',
'oz',
'episode',
'youll',
'hooked',
'right',
'exactly',

Figure 17: Removing stopwords

Stemming words to its root

The reviews were looped through, and each word was stemmed to its root word using SnowballStemmer. 'watched', 'watching' were converted to 'watch'.

```
In [9]: #stemming words to its root; watching,watched to watch
def stemText(text):
    snowballStem = SnowballStemmer('english')
    return " ".join([snowballStem.stem(w) for w in text])

imdb_data.review = imdb_data.review.apply(stemText)
imdb_data.review[0]

Out[9]: 'one review mention watch 1 oz episod youll hook right exact happen meth first thing struck oz brutal unflinch scene violenc set right word go trust show faint heart timid show pull punch regard d rug sex violenc hardcor classic use wordit call oz nicknam given oswald maximum secur state penite ntari focus main emerald citi experiment section prison cell glass front face inward privaci high agenda em citi home manyaryan muslim gangsta latino christian italian irish moreso scuffl death st are dodgi deal shadi agreement never far awayi would say main appeal show due fact goe show wouldn t dare forget pretti pictur paint mainstream audienc forget charm forget romanceoz doesnt mess aro und first episod ever saw struck nasti surreal couldnt say readi watch develop tast oz got accusto m high level graphic violenc violenc injustic crook guard wholl sold nickel inmat wholl kill order get away well manner middl class inmat turn prison bitch due lack street skill prison experi watch oz may becom comfort uncomfort viewingthat get touch darker side'
```

Figure 18: Stemming words to its root

Vectorizing data

With TfidfVectorizer taking min_df 1, max_df 1.0 and ngram_range (1,1) as default parameters, the output performance metrics gave less values for accuracy, precision, recall and F1 score i.e., 85.98%, 87.12%, 84.45% and 85.77% respectively.

```
In [10]: #converting collection of raw documents into a matrix of TF-IDF features
tfidf = TfidfVectorizer()
x = tfidf.fit_transform(imdb_data.review)
y = np.array(imdb_data.sentiment.values)

#setting random_state to a fixed value 2 means that every time the code is run,
#the function will randomly split the data in the same way
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.30, random_state=2)

#separating 30% data for test and 70% data for training
```

Figure 19: Separating Training and Test Values with default vectorizer

```
In [11]: #creating MultinomialNB model
m=MultinomialNB(alpha=1.0,fit_prior=True)

#fitting training data
m.fit(x_train,y_train)

#predicting on test data
prediction=m.predict(x_test)

#Calculating performance metrics|
print("Multinomial accuracy = ",accuracy_score(y_test,prediction))
print("Multinomial precision= ",precision_score(y_test,prediction))
print("Multinomial recall= ",recall_score(y_test,prediction))
print("Multinomial F1 score= ",f1_score(y_test,prediction))

Multinomial accuracy = 0.8598666666666667
Multinomial precision= 0.8712694264887911
Multinomial recall= 0.8445540594587388
Multinomial F1 score= 0.8577037638776063
```

Figure 20: Creating, fitting, and evaluating the model with default vectorizer

3.5.4. Separating Train and Test Values

Hence, `TfidfVectorizer (min_df=2, max_df=0.5, ngram_range= (1,2))` was used for the final solution. The processed dataset was then split into train and test data with the ratio of 70:30.

```
In [12]: #converting collection of raw documents into a matrix of TF-IDF features
#min_df=2 specifies the word must appear in atleast two documents to be included
#max_df=0.5 specifies the word must appear in less than 50% of the documents in order to be included in the vocabulary.
#ngram_range=(1,2) extracts all unigrams (single words) and bigrams (pairs of words).
tfidf = TfidfVectorizer(min_df=2, max_df=0.5, ngram_range=(1,2))
x = tfidf.fit_transform(imdb_data.review)
y = np.array(imdb_data.sentiment.values)

#setting random_state to a fixed value 2 means that every time the code is run,
#the function will randomly split the data in the same way, which is useful for experimentation and debugging
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.30, random_state=2)

#separating 30% data for test and 70% data for training
```

Figure 21: Separating Training and Test Values with parameterized vectorizer

A split of 70% for training and 30% for testing was chosen because

- It provides a large enough training set to allow the model to learn while still having enough examples in the test set to evaluate the model's performance.
- It provides the model to be trained and tested on a diverse range of data, which can help improve its generalizability and make it more robust.

3.5.5. Creating and training the model

Multinomial Naïve Bayes model was created, and training data was fit into it.

```
In [13]: #creating MultinomialNB model
m=MultinomialNB(alpha=1.0,fit_prior=True)

#fitting training data
m.fit(x_train,y_train)

Out[13]: MultinomialNB()
```

Figure 22: Creating and fitting the model

3.5.6. Performance metrics

The model was then made to predict on the test data. Performance metrics were calculated by cross-checking on the predicted data and the actual y-labels (sentiments).

```
In [14]: #predicting on test data
         prediction=m.predict(x_test)

         #Calculating performance metrics|
         print("Multinomial accuracy = ",accuracy_score(y_test,prediction))
         print("Multinomial precision= ",precision_score(y_test,prediction))
         print("Multinomial recall= ",recall_score(y_test,prediction))
         print("Multinomial F1 score= ",f1_score(y_test,prediction))

         Multinomial accuracy =  0.8849333333333333
         Multinomial precision=  0.8878441907320349
         Multinomial recall=    0.8812158378882816
         Multinomial F1 score=   0.8845175966813864
```

Figure 23: Testing and Evaluating the model

1. Accuracy calculation

It is a metric used to evaluate the performance of a model. It is the fraction of predictions that the model got right.

It is calculated by the formula given below:

```
Accuracy = (Number of correct predictions) / (Total number of predictions) or,
Accuracy = (TP + TN) / (TP + TN + FP + FN)
```

Result: The model yielded an accuracy of 88.49%.

2. Precision calculation

It is the fraction of correct positive predictions out of all positive predictions made by the model.

It is calculated by the formula given below:

```
precision = true_positives / (true_positives + false_positives)
```

Result: The model yielded a precision of 88.78%.

3. Recall calculation

It is the proportion of correct positive predictions out of all actual positive observations.

It is calculated by the formula given below:

```
recall = true_positives / (true_positives + false_negatives)
```

Result: The model yielded a recall score of 88.12%.

4. F1 score calculation

It is a measure of a model's accuracy that combines precision and recall bringing out a mean value. Higher mean indicates a better balance between precision and recall.

It is calculated by the formula given below:

$$F1 \text{ score} = 2 * (\text{precision} * \text{recall}) / (\text{precision} + \text{recall})$$

Result: The model yielded a F1 score of 88.45%.

Therefore, Final performance metrics yielded to 88.49% accuracy, precision of 88.78%, recall of 88.12% and F1 score of 88.45%.

5. Confusion matrix

It is a table used to evaluate the performance of a classification algorithm. The rows of the matrix represent the predicted classes while the columns represent the actual classes (Brownlee, 2020).

	Actual true	Actual false
Predicted true	True positive	False positive
Predicted false	False negative	True negative

Table 1: Confusion Matrix

- True Positive (TP) is the number of instances that are correctly predicted to be positive.
- False Positive (FP) is the number of instances that are incorrectly predicted to be positive.
- False Negative (FN) is the number of instances that are incorrectly predicted to be negative.
- True Negative (TN) is the number of instances that are correctly predicted to be negative.

Confusion matrix was created by cross-checking on the predicted data and the actual y-labels (sentiments).

```
In [15]: confusionMatrix= confusion_matrix(y_test, prediction)
          print(confusionMatrix)

[[6664  835]
 [ 891 6610]]
```

Figure 24: Confusion Matrix

Plotting the confusion matrix using Matplotlib

The matrix was plotted using Matplotlib for easy visualization of the positives and negatives predicted by the model.

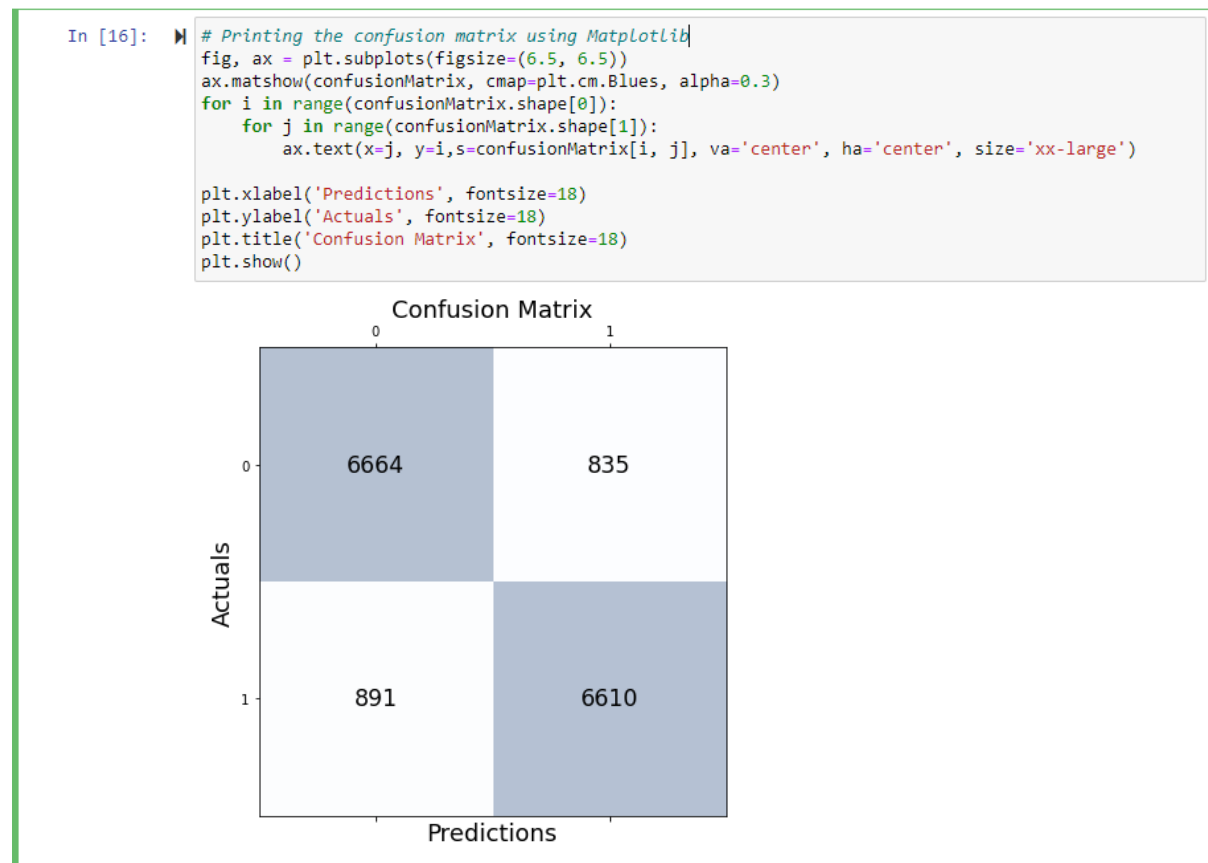


Figure 25: Plotting confusion matrix using Matplotlib

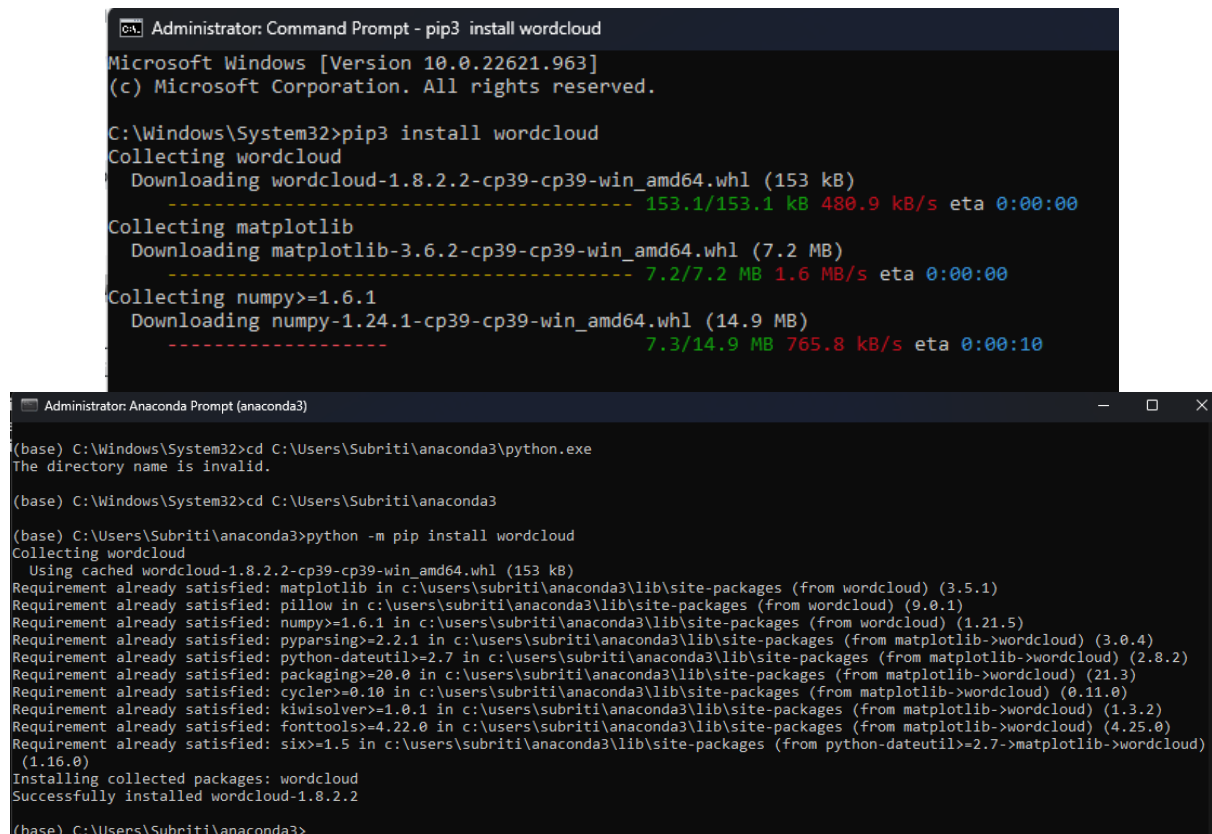
Since the matrix output has been flattened, the values for the matrix are as follows:

- True Positive: 6610
- False Positive: 835
- False Negative: 891
- True Negative: 6664

With maximum true predictions, we can infer that the model is performing quite well (Agrawal, 2021)

6. Word cloud of positive and negative sentiments

For displaying a word cloud, it had to be manually installed. Installation was done from the command prompt as well as the anaconda prompt until it was finally installed.



```

Administrator: Command Prompt - pip3 install wordcloud
Microsoft Windows [Version 10.0.22621.963]
(c) Microsoft Corporation. All rights reserved.

C:\Windows\System32>pip3 install wordcloud
Collecting wordcloud
  Downloading wordcloud-1.8.2.2-cp39-cp39-win_amd64.whl (153 kB)
----- 153.1/153.1 kB 480.9 kB/s eta 0:00:00
Collecting matplotlib
  Downloading matplotlib-3.6.2-cp39-cp39-win_amd64.whl (7.2 MB)
----- 7.2/7.2 MB 1.6 MB/s eta 0:00:00
Collecting numpy>=1.6.1
  Downloading numpy-1.24.1-cp39-cp39-win_amd64.whl (14.9 MB)
----- 7.3/14.9 MB 765.8 kB/s eta 0:00:10

Administrator: Anaconda Prompt (anaconda3)
(base) C:\Windows\System32>cd C:\Users\Subriti\anaconda3\python.exe
The directory name is invalid.

(base) C:\Windows\System32>cd C:\Users\Subriti\anaconda3

(base) C:\Users\Subriti\anaconda3>python -m pip install wordcloud
Collecting wordcloud
  Using cached wordcloud-1.8.2.2-cp39-cp39-win_amd64.whl (153 kB)
Requirement already satisfied: matplotlib in c:\users\subriti\anaconda3\lib\site-packages (from wordcloud) (3.5.1)
Requirement already satisfied: pillow in c:\users\subriti\anaconda3\lib\site-packages (from wordcloud) (9.0.1)
Requirement already satisfied: numpy>=1.6.1 in c:\users\subriti\anaconda3\lib\site-packages (from wordcloud) (1.21.5)
Requirement already satisfied: pyparsing>=2.2.1 in c:\users\subriti\anaconda3\lib\site-packages (from matplotlib->wordcloud) (3.0.4)
Requirement already satisfied: python-dateutil>=2.7 in c:\users\subriti\anaconda3\lib\site-packages (from matplotlib->wordcloud) (2.8.2)
Requirement already satisfied: packaging>=20.0 in c:\users\subriti\anaconda3\lib\site-packages (from matplotlib->wordcloud) (21.3)
Requirement already satisfied: cycler>=0.10 in c:\users\subriti\anaconda3\lib\site-packages (from matplotlib->wordcloud) (0.11.0)
Requirement already satisfied: kiwisolver>=1.0.1 in c:\users\subriti\anaconda3\lib\site-packages (from matplotlib->wordcloud) (1.3.2)
Requirement already satisfied: fonttools>=4.22.0 in c:\users\subriti\anaconda3\lib\site-packages (from matplotlib->wordcloud) (4.25.0)
Requirement already satisfied: six>=1.5 in c:\users\subriti\anaconda3\lib\site-packages (from python-dateutil->matplotlib->wordcloud) (1.16.0)
Installing collected packages: wordcloud
Successfully installed wordcloud-1.8.2.2

(base) C:\Users\Subriti\anaconda3>
  
```

Figure 26: Installing wordcloud

Positive and Negative Reviews were extracted from pandas DataFrame by being based on sentiment column's value using the following formula (Naveen, 2021).

```

positiveReviews= imdb_data [imdb_data['sentiment'] ==1] ['review']
negativeReviews= imdb_data [imdb_data['sentiment'] ==-1] ['review']
  
```

These lists of reviews were then merged into a single bulk text of positives and negatives reviews to generate a word cloud.

For Negative Review Word Cloud

```
In [17]: #word cloud for reviews words
WC=WordCloud(width=1000,height=500,max_words=2000,min_font_size=5, background_color="white")

plt.figure(figsize=(10,10))

#imdb_data['sentiment']==-1 gives true or false for the condition
#imdb_data[imdb_data['sentiment']==-1] gives all the value/ columns fulfilling the condition
#imdb_data[imdb_data['sentiment']==-1]['review'] returns only the negative reviews column
negativeReviews=imdb_data[imdb_data['sentiment']==-1]['review']
print(negativeReviews)

for review in negativeReviews:
    #combining all the negative reviews into one text
    negative_text= "" .join(review)
negative_words=WC.generate(negative_text)

#interpolation='bilinear' makes the displayed image appear more smoothly
plt.imshow(negative_words,interpolation='bilinear')
plt.show
```

```
3      basic there famili littl boy jake think there ...
7      show amaz fresh innov idea 70s first air first...
8      encourag posit comment film look forward watch...
10     phil alien one quirki film humour base around ...
11     saw movi 12 came recal scariest scene big bird...

...
49994   typic junk comedither almost laugh genuin mome...
49996   bad plot bad dialogu bad act idiot direct anno...
49997   cathol taught parochi elementari school nun ta...
49998   im go disagre previous comment side maltin one...
49999   one expect star trek movi high art fan expect ...
Name: review, Length: 25000, dtype: object
```

```
Out[17]: <function matplotlib.pyplot.show(close=None, block=None)>
```



Figure 27: Negative review word cloud

The word clouds were generated using WordCloud and plot using the Matplotlib library. It can be noted that words like movie, unfortun, best, charact, worst, cring, far, avoid, worth and watch were among the words commonly written in a negative review.

For Positive Review Word Cloud

```
In [18]: plt.figure(figsize=(10,10))

positiveReviews=imdb_data[imdb_data['sentiment']==1]['review']
print(positiveReviews)

for review in positiveReviews:
    #combining all the positive reviews into one text
    positive_text= "".join(review)
    positive_words=WC.generate(positive_text)

#interpolation='bilinear' makes the displayed image appear more smoothly
plt.imshow(positive_words,interpolation='bilinear')
plt.show
```

```
0      one review mention watch 1 oz episod youll hoo...
1      wonder littl product film techniqu unassum old...
2      thought wonder way spend time hot summer weeke...
4      petter mattei love time money visual stun film...
5      probabl alltim favorit movi stori selfless sac...
...
49983  love fan origin seri alway wonder back stori w...
49985  imaginari hero clear best film year complet ut...
49989  got one week ago love modern light fill true c...
49992  john garfield play marin blind grenad fight gu...
49995  thought movi right good job wasnt creativ orig...
Name: review, Length: 25000, dtype: object
```

```
Out[18]: <function matplotlib.pyplot.show(close=None, block=None)>
```



Figure 28: Positive review word cloud

It can be noted that words like movie, classic, enjoy, creative, expect, fun, good, recommend, see and proud were among the words commonly written in a positive review.

These word clouds certainly help us analyse the patterns of popularity or downfall of the movie with a glance. Also, these common words in the data will have the highest frequencies in the documents and will therefore be given more weight by the TF-IDF vectorizer. This affects the model's predictions because the words that are given the most weight will be the ones that the model considers most important when making its predictions.

4. Conclusion

4.1. Analysis of the work done

In conclusion, the project was based on the concepts of AI, ML and NLP. Among the researched domains, sentiment analysis of the user reviews for movies was selected. The IMDB dataset of 50K Movie Reviews was found from Kaggle and used for this system. The methodology chosen for analysis of sentiments in the user reviews was Naïve Bayes Classifier method.

The development work started with importing of libraries and the dataset followed by a quick data analysis. Data were pre-processed to remove unwanted characters, and words were stemmed to its root. These were done to increase the efficiency of the model. Then data were vectorized i.e., converted to a numerical matrix using TfidfVectorizer. Finally, the multinomial naïve bayes model was created and test dataset was fit into it. The model performed very well on the provided dataset. It yielded an accuracy of 88.49% with precision of 88.78%, recall of 88.12% and F1 score of 88.45% on the test dataset. The concept of confusion matrix was researched on and hence created to further evaluate the model. Finally, the word cloud for positive and negative words were created to visualize the most common words in a large collection of text data.

Hence, with appropriate and meticulous data pre-processing, the model yielded good results on the given dataset. The model showed a noticeable increase in the accuracy when the data was vectorized with custom parameters.

Since, the goal of the model was to accurately classify all reviews, regardless of whether they are positive or negative, reflection on a high accuracy score (88.49%) confirms that the model is correctly classifying a large proportion of the reviews.

Therefore, the project was of great help in strengthening the knowledge on the concepts of Data Analysis, Machine Learning, and its algorithms. The use of classification algorithm Naïve Bayes was deeply explored and implemented into the system to solve a problem of sentiment analysis. Hence, the experience gained from this coursework would surely help me undertake more such projects in the future.

4.2. How the solution addresses real world problems

Since, IMDb hosts a plethora of movies, and with it comes a lot of reviews of the users having some sentiments and emotions. With sentiment analysis can the corporations take sudden actions in response to the reviews written.

With sentiment analysis, the movie makers have the chance to follow online conversations/ reviews about themselves and their rivals in real time. They also learn quantitatively how positively or negatively they are perceived at the same time. This provides them with an opportunity to make amends (movie scenes or dialogs) and increase customer satisfaction.

The system will also help new users analyze a certain movie to see if it is worth their time. Sentiment analysis can certainly help analyze the popularity of the movie and find the influence of any characters/ actors (FreeProjectz, 2019).

4.3. Further work

Moving forward, a proper web or mobile application could be developed using the project's research and solution to determine whether a user review will have a positive or a negative sentiment. This would help corporations/people track sentiment of the reviews in real-time and take quick decisions.

Also, other classification techniques could be discovered to find the one yielding the highest accuracy on the dataset to ensure the predicted data has no margin of error.

References

- Agrawal, S., 2021. *Understanding the Confusion Matrix from Scikit learn*. [Online]
Available at: <https://towardsdatascience.com/understanding-the-confusion-matrix-from-scikit-learn-c51d88929c79>
[Accessed 8 January 2023].
- Ajmera, A., 2022. Sentiment Analysis of IMDb Movie Reviews. *International Journal for Research in Applied Science & Engineering Technology (IJRASET)* | ISSN: 2321-9653, 10(XII), pp. 1-9.
- Anaconda.Documentation, 2017. *Anaconda Navigator*. [Online]
Available at: <https://docs.anaconda.com/navigator/index.html>
[Accessed 7 January 2023].
- Banoula, M., 2022. *Supervised and Unsupervised Learning in Machine Learning*. [Online]
Available at: <https://www.simplilearn.com/tutorials/machine-learning-tutorial/supervised-and-unsupervised-learning>
[Accessed 13 December 2022].
- Bredava, A., 2022. *Top 10 best free and paid sentiment analysis tools*. [Online]
Available at: <https://awario.com/blog/sentiment-analysis-tools/>
[Accessed 10 December 2022].
- Brownlee, J., 2020. *What is a Confusion Matrix in Machine Learning*. [Online]
Available at: <https://machinelearningmastery.com/confusion-matrix-machine-learning/>
[Accessed 8 January 2023].
- Chaudhury, S., 2020. *Building a Sentiment Analyzer With Naive Bayes*. [Online]
Available at: <https://medium.com/swlh/building-a-sentiment-analyzer-with-naive-bayes-c96cc8aa52a5>
[Accessed 11 December 2022].
- djangostars, 2022. *8 Reasons Why Python is Good for AI and ML*. [Online]
Available at: <https://djangostars.com/blog/why-python-is-good-for-artificial->

intelligence-and-machine-learning/

[Accessed 7 January 2023].

- Educative Answers Team, 2022. *Definition: Model fitting*. [Online]
Available at: <https://www.educative.io/answers/definition-model-fitting>
[Accessed 12 December 2022].
- FreeProjectz, 2019. *Sentiment Analysis for IMDb Movie Review*. [Online]
Available at: <https://www.freeprojectz.com/paid-project/python-sentiment-analysis-project/imdb-movie-review>
[Accessed 12 December 2022].
- Glover, E., 2022. *Artificial Intelligence*. [Online]
Available at: <https://builtin.com/artificial-intelligence>
[Accessed 10 December 2022].
- Goyal, A. & Parulekar, A., 2015. *Sentiment Analysis for Movie Reviews*, California: CSE. University of California, San Diego.
- H., S. et al., 2019. Sentiment Analysis of Movie Review using data Analytics Techniques. *IRE Journals | ISSN: 2456-8880*, 2(9), pp. 1-5.
- JavaTpoint, 2021. *Classification Algorithm in Machine Learning*. [Online]
Available at: <https://www.javatpoint.com/classification-algorithm-in-machine-learning>
[Accessed 13 December 2022].
- Lodhi, F., 2021. *Sentiment Analysis of Movie Reviews using Logistic Regression*. [Online]
Available at: <https://furqanlodhi.medium.com/sentiment-analysis-of-movie-reviews-using-logistic-regression-269aa31f53c0>
[Accessed 11 December 2022].
- Maas, A. L. et al., 2011. Learning Word Vectors for Sentiment Analysis. *Learning Word Vectors for Sentiment Analysis*, Volume Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, pp. 142-150.
- Mesevage, T. G., 2020. *Top Machine Learning Algorithms Explained: How Do They Work?*. [Online]
Available at: <https://monkeylearn.com/blog/machine-learning-algorithms/>
[Accessed 10 December 2022].

- MonkeyLearn, 2022. *Sentiment Analysis*. [Online]
Available at: <https://monkeylearn.com/sentiment-analysis/>
[Accessed 10 December 2022].
- Narayanan, V., Arora, I. & Bhatia, A., n.d. *Fast and accurate sentiment classification using an enhanced Naive Bayes model*. [Online]
Available at: <https://arxiv.org/ftp/arxiv/papers/1305/1305.6143.pdf>
[Accessed 11 December 2022].
- Naveen, 2021. *Pandas Extract Column Value Based on Another Column*. [Online]
Available at: <https://sparkbyexamples.com/pandas/pandas-extract-column-value-based-on-another-column/>
[Accessed 8 January 2023].
- Pascual, F., 2022. *Getting Started with Sentiment Analysis using Python*. [Online]
Available at: <https://huggingface.co/blog/sentiment-analysis-python>
[Accessed 10 December 2022].
- Reza, M. et al., 2020. Improved Accuracy of Sentiment Analysis Movie. *Journal of Physics: Conference Series* 1641 012060.
- Roldós, I., 2020. *NLP, Machine Learning & AI, Explained*. [Online]
Available at: <https://monkeylearn.com/blog/nlp-ai/>
[Accessed 10 December 2022].
- Rosenberg, E., 2022. *The Importance of Excel in Business*. [Online]
Available at: <https://www.investopedia.com/articles/personal-finance/032415/importance-excel-business.asp>
[Accessed 7 January 2023].
- Shivanandhan, M., 2020. *What is Natural Language Processing? An NLP Definition and Tutorial for Beginners*. [Online]
Available at: <https://www.freecodecamp.org/news/what-is-natural-language-processing-an-nlp-definition-and-tutorial-for-beginners/>
[Accessed 10 December 2022].
- Stecanella, B., 2017. *A practical explanation of a Naive Bayes classifier*. [Online]
Available at: <https://monkeylearn.com/blog/practical-explanation-naive-bayes->

classifier/

[Accessed 11 December 2022].

- tutorialspoint, 2019. *Machine Learning - Jupyter Notebook*. [Online]
Available at:
https://www.tutorialspoint.com/machine_learning_with_python/machine_learning_with_python_jupyter_notebook.htm
[Accessed 7 January 2023].
- UKEssays, 2018. *The Importance Of Online Reviews*. [Online]
Available at: <https://www.ukessays.com/essays/film-studies/the-importance-of-online-reviews-film-studies-essay.php#citethis>
[Accessed 10 December 2022].
- Yassen, M. & Tedmori, S., 2019. *Movies Reviews Sentiment Analysis and Classification*. Amman, Jordan, 10.1109/ JEEIT.2019.8717422.
- Yilmaz, B., 2022. *15 Sentiment Analysis Statistics by Users, Region & Sector*. [Online]
Available at: <https://research.aimultiple.com/sentiment-analysis-stats/>
[Accessed 11 December 2022].