A Major Project Final Report on

Sentiment Analysis System for Business Reviews

Submitted in Partial Fulfillment of the Requirements for

the Degree of **Bachelor of Engineering in Computer Engineering**

under **Pokhara University**

Submitted by:

Bibek Chandra Poudel, 171205

Manish Kumar Mandal, 171216

Puspha Raj Pandeya, 171229

Sandesh Paudel, 171238

Email: sandesh.171238@ncit.edu.np

Under the supervision of

Rishi K. Marseni

Date: 9 Sept 2022

Department of Computer Engineering

NEPAL COLLEGE OF INFORMATION TECHNOLOGY

Balkumari, Lalitpur, Nepal

ACKNOWLEDGEMENT

This project would not have been possible without the joint efforts of many individuals. It has been a pleasure for us to acknowledge the assistance and contributions that were very important and supportive throughout the project. We would like to extend our sincere thanks to all of them. We owe special thanks to several people who have devoted much of their time and expertise without which it would have been very difficult for us to complete our project entitled "**Sentiment Analysis System for Business Reviews**". We are highly indebted to our Project Supervisor, **Er. Rishi K. Marseni,** for his valuable guidance, encouragement, and support, throughout the project.

We take this opportunity to thank our associate professor, **Dr. Roshan Chitrakar,** for providing us with the opportunity to embark on this project. We also express our respect and love to our guardians and all other family members for their love and encouragement throughout our project development period. Last but not least, we would like to thank all the staff in the **Department of Computer Engineering** and colleagues who have knowingly or unknowingly helped us to achieve the successful completion of the project.

We feel obliged to submit this project as part of our curriculum for the partial fulfillment of the requirements leading to the award of the degree of **Bachelor of Computer Engineering.** Although there may be many who are unacknowledged in this humble vote of thanks, there are none who remain unappreciated.

I was involved in a "Sentiment Analysis for Business Reviews" research project. The research objective was to compare how Transformer-based models like BARD are better than traditional Neural Network models like CNN for sentiment analysis. The finding was due to the ability of transformers to memorize long sequences and the order of words, they are better than Neural Network models. The accuracy we achieved using BARD was roughly 92% whereas with CNN it was around 85%.

ABSTRACT

Sentiment analysis is currently of broad and current interest in the domain of social media. With the multiplication of social media platforms, which offer anonymity, easy access, and online community formation and online debate, the issue of sentiment detection and tracking has become a growing challenge to society, individuals, policy-makers, and researchers. As the volume of online mixed and confusing sentiments is increasing, methods that automatically detect those sentiments are very much required.

Moreover, these problems have also been attracting the Natural Language Processing and Machine Learning communities a lot. The goal of "Sentiment Analysis System for Business Reviews" is to look at how Natural Language Processing applies to detect mixed and confusing sentiments expressed by someone on a particular topic.

We have planned to develop the application using the Python language. We will use Jupyter Notebook as an IDE to create a model, VS Code for backend coding, and the Flask framework for creating APIs.

Keywords: *anonymity, API, Flask, Jupyter Notebook, Machine Learning, Natural Language Processing, policy-makers, sentiment analysis, VS code*

Contents

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

API	Application Programming Interface
BERT	Bidirectional Encoder Representation from Transformers
CNN	Convolutional Neural Network
ER	Entity Relationship
IDE	Integrated Development Environment
KPI	Key Performance Indicators
NER	Named-entity Recognition
NLP	Natural Language Processing
RNN	Recurrent Neural Network
SRS	Software Requirements Specifications
VS	Visual Studio

1. INTRODUCTION

Analyzing the situation of the country, technology is found to be less used by the people of every sector. If we can properly use the technology, it will be very beneficial. The proposed work attempts to look at how natural language processing applies to detecting mixed and confusing sentiments.

Sentiment analysis is the scanning of words written or said by a person to determine the emotions they're most likely feeling at the time(*Sentiment Analysis Guide*, n.d.). We can monitor real-time conversations about our company and its products or services to measure consumer sentiment. We can use the data from sentiment analysis to determine which products and services our customers want or how they feel about a brand.

Sentiment analysis means analyzing and finding the emotion or intent behind a piece of text, speech, or any mode of communication. One of the examples may be hate speech, which is commonly defined as any communication that disparages a person or a group based on some characteristic such as race, color, ethnicity, gender, sexual orientation, nationality, religion, or other characteristics. With the multiplication of social media platforms, which offer anonymity, easy access, and online community formation and online debate, the issue of sentiment detection and tracking has become a growing challenge to society, individuals, policy-makers, and researchers. Over the past years, interest in online sentiment analysis particularly the automatization of this task has continuously grown, along with the social impact of the phenomenon(Raj, 2021).

The main aim of this project is to find out how Natural Language Processing techniques can contribute to the detection of sentiments hidden behind text, speech, or any other means of communication. We are taking datasets from Kaggle that have 3 labels, or targets.

1. Positive Sentiment
2. Negative Sentiment
3. Neutral Sentiment

1.1. PROBLEM STATEMENT

The anonymity and flexibility afforded by the Internet have made it easy for users to communicate with mixed and confusing sentiments. We, humans, communicate with each other in a variety of languages, and any language is just a mediator or a way in which we try to express ourselves. Moreover, whatever we say has a sentiment associated with it. It might be positive or negative, or it might be neutral as well.

In most business cases, when we launch a product or service on the market, it's very important to get the customer's feedback on our product. If we don't analyze customer reviews, we can never know what we are lacking and we cannot grow our business efficiently. We can also find out the true needs of the customers. By analyzing the reviews from the customers, we can significantly grow our business.

We, humans, communicate with each other in a way that we call "Natural Language," and it's easy for us to interpret, but it's much more complicated and messier if we look into it. Because there are billions of people and they have their style of communicating, i.e., a lot of tiny variations are added to the language, a lot of sentiments are attached to it, which is easy for us to interpret, but it becomes a challenge for the machines(Raj, 2021).

1.2. PROJECT OBJECTIVES

Sentiment analysis can be an excellent source of information and can provide insights that can:

- Help businesses analyze their products based on reviews.
- Help businesses find market trends.
- Test beta products (beta testing).
- Classify texts into negative or positive sentiments.
- Help businesses determine their marketing strategy on time.

1.3. SIGNIFICANCE OF STUDY

Sentiment analysis is the process of detecting positive or negative sentiments in text. It's often used by businesses to detect sentiment in social data, gauge brand reputation, and understand customers. The growth of sentiment analysis as one of the most active research areas in the last 10 years is due to different reasons. First, sentiment analysis has a wide array of applications in almost every domain. Second, it offers many challenging research problems that have never been studied before. Third, with the advent of big data technologies, we now have a huge volume of opinionated data recorded and easily accessible in digital forms on the web. Most of the work regarding polarity classification usually considers text as unique information to infer sentiment, disregarding that social networks are actually networked environments. A novel research branch is developing to take advantage of both natural language and social network relationships(*Sentiment Analysis in Social Networks*, 2017).

The findings of this study will greatly contribute to the benefit of society as the study outcomes will help in extracting sentiment(meaning) from the text(*IRA IRYANI_15187.Pdf*, n.d.). It enables computers to understand and interpret sentences, paragraphs, or entire documents by analyzing their grammatical structure and identifying relationships between individual words in a given context. Therefore, the goal is to extract the exact meaning or dictionary meaning from the text. The job of a sentiment analyzer is to check the text for meaningfulness.

1.4. SCOPE AND LIMITATIONS

Scope and limitations are two terms that address the details of project. The term scope refers to the extent to which we plan to study/research our topic. Limitations refers to the shortcomings - things we believe our product lacked or ways in which it could have been better.

1.4.1. Scope

Some of the scope that our product tries to cover are listed below

➢ **Identifying and predicting market trends:**

It enables you to analyze large amounts of market research data in order to spot emerging trends and better understand consumer buying habits.

➢ **Keeping an eye on the brand's image:**

It is used to investigate user perceptions of a product or topic.

➢ **Examining public opinion polls and political polls:**

Public and political polls severely hamper businesses. So, examining public and political polls on time can help businesses take necessary actions early on.

➢ **Data from customer feedback is being analyzed:**

Data from customer feedback can be used to identify areas for improvement.

1.4.2. Limitations

The list of shortcomings that our product lacked are as follows:

- Unable to recognize things like sarcasm, irony, negation, jokes, and exaggerations.
- With short sentences and pieces of text, there might not be enough context for reliable sentiment analysis.
- An individual’s sentiment toward a brand or product may be influenced by one or more indirect causes; someone might have a bad day and give a negative remark.

- Texts written in Nepali or Devanagari cannot be determined as of now.

2. LITERATURE REVIEW

(“E-Commerce Product Review Sentiment Classification Based on a Naïve Bayes Continuous Learning Framework,” 2020) Introduced an NB method for multi-domain and large-scale e-commerce platform product review classification of sentiment. Consequently, the parameter evaluation method was extended in NB to a continuous learning fashion. Later, for fine-tuning the learned distribution on the basis of three types of assumptions, many ways were introduced for acquiring the best performance. The results have shown that the suggested model has high accuracy in Amazon product and movie review sentiment datasets.

(*Tourism Mobile App With Aspect-Based Sentiment Classification Framework for Tourist Reviews*, n.d.) Recommended a novel approach to aspect-based sentiment classification, which recognized the features in a precise manner and attained the best classification accuracy. Moreover, the scheme was developed as a mobile application, which assisted tourists in identifying the best hotel in town, and the proposed model was analyzed using real-world data sets. The results have shown that the presented model was effective in both recognition and classification.

(Brownlee, 2017) Recommended a deep convolutional neural network for sentiment analysis (Text classification). He used word embedding for representing text where different words with similar meanings have a similar real-valued vector representation. The results have shown that he gets around 76% of accuracy.

(*What Is BERT (Language Model) and How Does It Work?*, n.d.) Deepak Moonat, 2021 recommended a BERT model for Sentiment Analysis. He used fine-tuning BERT model that takes around two hours on GPU to complete training, with just 1 epoch where he achieve around 85% accuracy on validation.

3. PROPOSED METHODOLOGY

It is a web application that helps in extracting sentiments (meaning) from the text. We have planned to work according to these methodologies of applying knowledge, skills, tools, and techniques for a wide range of activities to meet the needs of our project sentiment analysis.

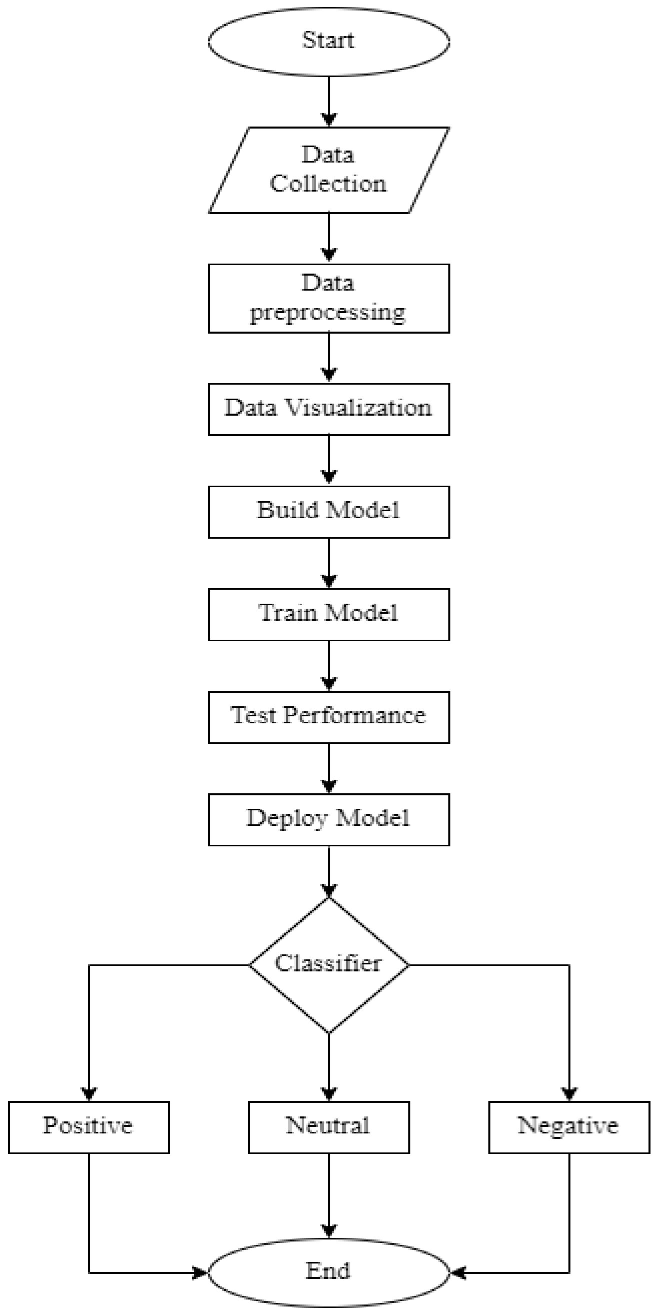

Figure 1: System Flow Diagram for model selection

The system flow diagram is basically a visual representation of data flow, excluding the minor parts and including the major parts of the system in a systematic sequential manner. The diagram consists of several steps that identify where the input is coming into the system and the output going out of the system. The system flow diagram for model training and choosing in our system is as shown above.

3.1. SOFTWARE DEVELOPMENT LIFE CYCLE

The framework that we planned to incorporate for developing this project was the Iterative Incremental Model. Firstly, we gathered user requirements to increase the developer's understanding of the research area. Then, it was carried out to model user requirements into detailed computer-based specifications. At the system and software design stages, architectural design, database design, and interface design were developed. After the implementation, each module was tested as a unit to ensure the system fulfilled business and design requirements.

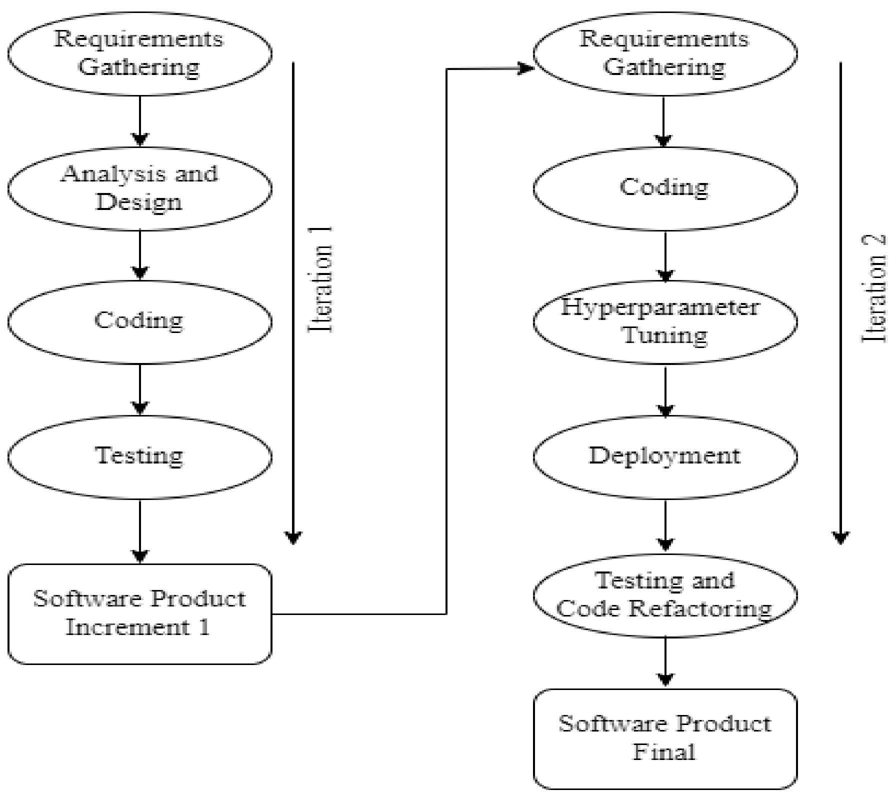

Figure 2: Iterative Incremental Model

A Sentiment Analysis System was developed by using an iterative incremental model because it is easy to understand and, after the competition of each iteration, we can again start from the beginning and solve the problem that occurs at a particular phase. The user provides feedback on the product for the planning stage of the next iteration cycle, and the development team responds, often by solving the problem that arises in the respective phase. This incremental cycle can continue until the product is shipped.

The iterative incremental model includes the following phases:

Analytic Phase- In this phase, the requirements of the software were analyzed, which resulted in **“Software Required Specifications."**

Design Phase- In this phase, the analysis of SRS was translated into the system’s design. The Context Diagram, Use-Case Diagram, ER Diagram, and Class Diagram were developed.

Coding Phase- This phase involves the coding as per the design and the formation of a working system at the end of the process.

Testing Phase- In this phase, the system was tested. With each test, certain changes were made as per the suggestions. This was done incrementally until a satisfactory result was achieved.

3.2. BENEFITS OF USING AN ITERATIVE INCREMENTAL MODEL

- ➢Generates working software quickly and early in the software life cycle.
- ➢More flexible—less costly to change scope and requirements.
- ➢Easier to test and debug during a smaller iteration.
- ➢Easier to manage risk because risky pieces are identified and handled during their iteration.
- ➢Each iteration is an easily managed milestone.

3.3. DIFFERENT STAGES OF ITERATION

3.3.1. First Iteration

In this phase we focused on the analysis and design of our system with reference to our objective. We gathered all requirements, and datasets from Kaggle. We define and draw different UML diagrams like Use case diagrams, sequence diagrams, etc. In this iteration, we focused on training a model using CNN for our sentiment analysis process. Also, we created a frontend for the user interface and a backend for functionality. We take simple input from the user, feed it to the model, and then the model generates the result, i.e, sentiment analysis. Overall tasks done in this iteration can be listed as,

- ➢ Gathered requirements and datasets.
- ➢ Created system design and used case diagrams.
- ➢ Created a basic model for sentiment analysis using CNN.
- ➢ Created a simple backend and frontend application that takes data from the user, feeds it to the model, and generates results.
- ➢ Documentation

3.3.2. Second Iteration

In this iteration phase, we worked on increasing accuracy for our previous CNN by doing hyperparameter tuning. And for even more accuracy than the previous CNN model we created BERT model which is based on the transformer. The previously designed frontend was also improved using better CSS and some more features were added. Also, the backend code was tested and refactored. And finally, an overall system test was also done in this phase. The overall task done in this phase can be listed as,

- ➢ Created a model using BERT which gave better accuracy than CNN.
- ➢ Also improved the CNN model by doing hyperparameter tuning.
- ➢ Improved frontend CSS and added some features.
- ➢ Deployment.
- ➢ Backend code refactoring and testing.
- ➢ Documentation.

3.4. USE CASE DIAGRAM

A use case diagram is used to represent the dynamic behavior of a system. It encapsulates the system's functionality by incorporating use cases, actors, and their relationships. A use case diagram is usually simple. It does not show the detail of the use cases:

- ➢ It only summarizes some of the relationships between use cases, actors, and systems.
- ➢ It does not show the order in which steps are performed to achieve the goals of each use case.

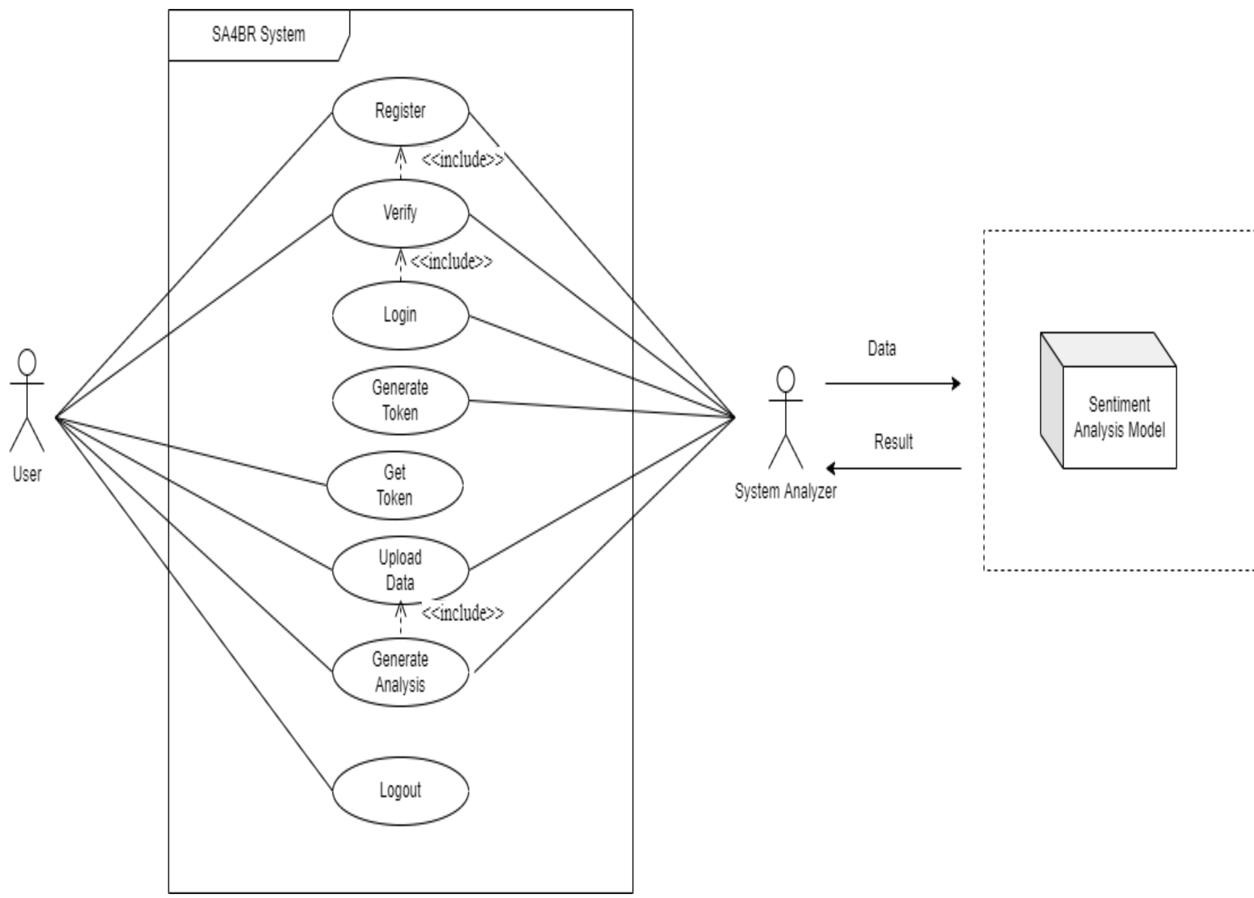

Figure 3: Use-Case Diagram

Use Case Description:

Use Case: 1

Use Case name: Register

Description: User has to provide his/her user details to register for the first time when using our system. User details contain name, email, and password.

Use Case: 2

Use Case name: Verify

Description: After a user has provided his/her user details, our system verifies whether the provided details are valid or not. If valid, a profile is created for them otherwise the registration process is terminated and logged back to the registration page.

Use Case: 3

Use Case name: Login

Description: A user is able to log in to the system with his/her credentials and the system checks for the validation if he/she is a legitimate user or not.

Use Case: 4

Use Case name: Generate Token

Description: After a successful login, a token is generated by our system which will be sent to the user.

Use Case: 5

Use Case name: Get Token

Description: The token generated by the system is sent back to the user and the user attached this token with every request s/he made to the server.

Use Case: 6

Use Case Name: Upload Data

Description: Users will upload texts, files, or API of their reviews for analysis of their products and to get results that will help them in their business.

Use Case: 7

Use Case name: Generate Analysis

Description: Sentiments are generated from the input data whether it is positive or negative. In general, the user will give input to the system, which will be forwarded to the model. Then in the model, the input file or text will be processed accordingly using a pre-defined model which will generate a result. Here result means analysis of sentiments in terms of negative or positive. It can help companies to know the customer view toward their product and helps in taking future decisions about quality service in their product and customer retention.

We can also build a use case display window to show two columns of information in the Main Success Scenario - usually, to include some sort of action performed, then a system response - similar to how test cases work.

Two-column use case format for our use case 7 named Generate Analysis is shown in the table below:

Actor Action	**System Response**
1. This use case begins when a user login to the system and gets a token provided by the system	
2. The user gives input to the system which can be in texts, files, or end-points.	
	3. System then provides user input to the

	model.
	4. Model uses a pre-trained model to generate results that are given to the system.
	5. System then provides the results generated by the model to the user in the visual form
6. Finally, users can see the results generated by the system and log out of our system	

Table 1: Two-column use case for Generate Analysis

Use Case: 8

Use Case Name: Logout

Description: The user logout from our system by clicking on the logout button and it will end the user session.

3.5. SEQUENCE DIAGRAM

A sequence diagram is an interaction diagram that emphasizes the time ordering of messages. It shows a set of objects and the messages sent and received by those objects. Graphically, a sequence diagram is a table that shows objects arranged along the X axis and messages, ordered in increasing time, along the Y axis.

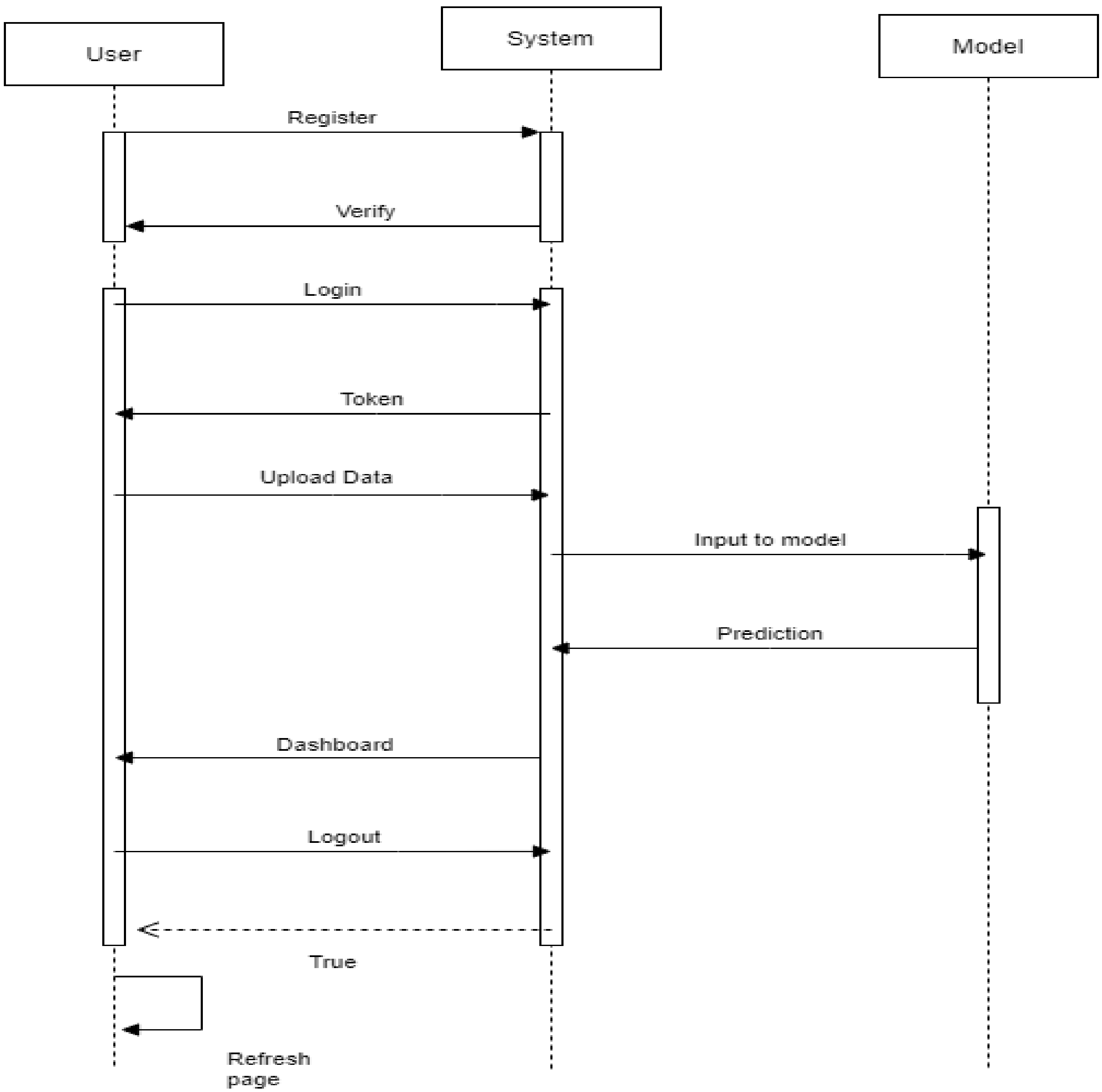

Figure 4: Sequence diagram

3.6. MACHINE LEARNING ALGORITHMS

3.6.1. Convolutional Neural Networks

A Convolutional Neural Network (ConvNet/CNN) is a Deep Learning algorithm which can take in an input text as well as image, assign importance (learnable weights and biases) to various aspects/objects in the text and be able to differentiate one from the other. The pre-processing required in a ConvNet is much lower as compared to other classification algorithms. The architecture of a ConvNet is analogous to that of the connectivity pattern of Neurons in the Human Brain and was inspired by the organization of the Visual Cortex.(Brownlee, 2017)

We use a Convolutional Neural Network (CNN) as they have proven to be successful at document classification problems. A conservative CNN configuration is used with 64 filters (parallel fields for processing words) and a kernel size of 3 with a rectified linear ('relu') activation function. This is followed by a pooling layer that reduces the output of the convolutional layer by one-third.

Next, the output from the CNN part of the model is flattened to one long 2D vector to represent the 'features' extracted by the CNN with 64 filters and a kernel size of 5 with a rectified linear activation function. This is followed by a pooling layer that reduces the output of the convolutional layer by one-fifth.

Again, the output from the above CNN part is flattened with 64 filters and a kernel size of 5 with rectified linear activation function. The back-end of the model is a standard Multilayer Perceptron layer to interpret the CNN features. The output layer uses a sigmoid activation function to output a value between 0 and 1 for the negative and positive sentiment in the review.

The image below shows how we build a model for CNN along with filters and the number of kernel sizes defined by dense value and activation function. Also, we use a binary cross entropy loss function for binary classification.

```python
12]: def build_model():
    sequences = layers.Input(shape=(MAX_LENGTH,))
    embedded = layers.Embedding(MAX_FEATURES, 64)(sequences)

    x = layers.Conv1D(64, 3, activation='relu')(embedded)

    #x = layers.BatchNormalization()(x)

    x = layers.MaxPool1D(3)(x)

    x = layers.Conv1D(64, 5, activation='relu')(x)
    #x = layers.BatchNormalization()(x)

    x = layers.MaxPool1D(5)(x)
    # Here, x
    x = layers.Conv1D(64, 5, activation='relu')(x)
    # Here, x
    x = layers.GlobalMaxPool1D()(x)

    x = layers.Flatten()(x)

    x = layers.Dense(100, activation='relu')(x)

    predictions = layers.Dense(1, activation='sigmoid')(x)

    model = models.Model(inputs=sequences, outputs=predictions)
    model.compile(
        optimizer='rmsprop',
        loss='binary_crossentropy',

        metrics=['binary_accuracy']

    )
    return model
model = build_model()
```

Figure 5: CNN model build

A Confusion matrix is an N x N matrix used for evaluating the performance of a classification model, where N is the number of target classes. The matrix compares the actual target values with those predicted by the machine learning model. This gives us a holistic view of how well our classification model is performing and what kinds of errors it is making.

Below figure is the confusion matrix of convolutional neural network which shows we get an accuracy of 0.944. As we can see, confusion matrix is nothing but a comparison matrix between true label of given datasets and predicted label by our model.

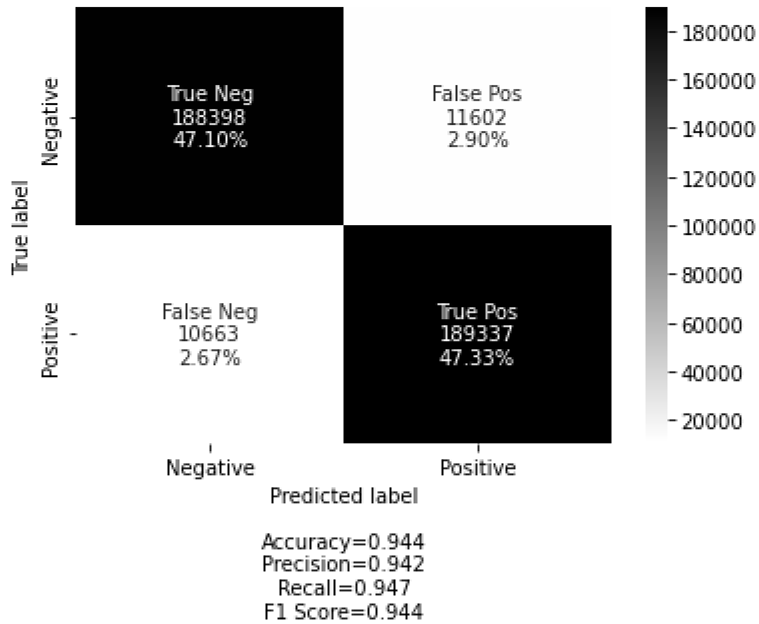

Figure 6: Confusion Matrix of CNN

The crucial term used in confusion matrix are listed below:

True Positive (TP)

The predicted value matches the actual value

The actual value was positive and the model predicted a positive value

True Negative (TN)

The predicted value matches the actual value

The actual value was negative and the model predicted a negative value

False Positive (FP) – Type 1 error

The predicted value was falsely predicted

The actual value was negative but the model predicted a positive value

False Negative (FN) – Type 2 error

The predicted value was falsely predicted

The actual value was positive but the model predicted a negative value

3.6.2 Bert

BERT is an open-source machine learning framework for natural language processing (NLP). BERT is designed to help computers understand the meaning of ambiguous language in the text by using surrounding text to establish context. The BERT framework was pre-trained using text from Wikipedia and can be fine-tuned with a question and answer datasets.(*What Is BERT (Language Model) and How Does It Work?*, n.d.)

BERT, which stands for Bidirectional Encoder Representations from Transformers, is based on Transformers, a deep learning model in which every output element is connected to every input element, and the weightings between them are dynamically calculated based upon their connection. In NLP, this process is called attention.(*What Is BERT (Language Model) and How Does It Work?*, n.d.)

Historically, language models could only read text input sequentially either left-to-right or right-to-left but couldn't do both at the same time. BERT is different because it is designed to read in both directions at once. This capability, enabled by the introduction of Transformers, is known as bi-directionality.

BERT is made possible by Google's research on Transformers. The transformer is the part of the model that gives BERT its increased capacity for understanding context and ambiguity in language. The transformer does this by processing any given word in relation to all other words in a sentence, rather than processing them one at a time. By looking at all surrounding words, the Transformer allows the BERT model to understand the full context of the word, and therefore better understand searcher's intent.

The below image shows how we extract BERT from pre-trained transformers,

```
In [1]: from transformers import BertTokenizer, TFBertForSequenceClassification
        from transformers import InputExample, InputFeatures

        model = TFBertForSequenceClassification.from_pretrained("bert-base-uncased")
        tokenizer = BertTokenizer.from_pretrained("bert-base-uncased")
```

Figure 7: BERT extraction

We now display the model summary so that we can see all the input and output layers used.

```
model.summary()

Model: "tf_bert_for_sequence_classification"
_________________________________________________________________
 Layer (type)                Output Shape              Param #   
=================================================================
 bert (TFBertMainLayer)      multiple                  109482240 
                                                                 
 dropout_37 (Dropout)        multiple                  0         
                                                                 
 classifier (Dense)          multiple                  1538      
                                                                 
=================================================================
Total params: 109,483,778
Trainable params: 109,483,778
Non-trainable params: 0
_________________________________________________________________
```

Figure 8: BERT model summary

The image above shows all the input and output layers we have initialized for our model. The output also shows the total params, trainable params, and non-trainable params.

Figure below shows a confusion matrix for BERT model. Here, we see for true positive we get around 46% and for true negative we get about 47%. And all together we get an accuracy of 0.94 from our BERT model which is slightly higher than our previous CNN model.

Figure 9: Confusion matrix of BERT

3.7 DATASETS

Here we are using datasets from Kaggle(*Amazon Reviews for Sentiment Analysis*, n.d.). There we have two files, one for training and another for testing purposes. The datasets have three labels, i.e. positive, negative and neutral. For training, we used a dataset containing 3.6 million rows of data and for testing, we used a dataset containing 4 hundred thousand rows of data. A sample of the dataset is shown below:

__label__2 Tyler Rocks: there is only one word to describe tyler hilton-Talent. i love this CD, it just shows off tylers amazing talent for writing music and his vocals are amazing. i strongly reccomend you buy this CD and his new EP the Tracks Of when it comes out late this year. Tyler is definately a talent to look out for in the future! __label__2 AWESOME: Absolutely amazing so relieving of my neck pain. I have an invert table and this does the same thing but is much easier and takes up so much less space. Its feels like having a chiropractor pull up on your neck like when they do traction. I love it. __label__1 What A Slap In The Face To Masami Ueda: Do NOT buy this cd. Ever. This was probably just released as a test to see if REfans would buy anything that had "Resident Evil/Biohazard" on it.How dare this metamorphis guy ruin such perfect scores.Masami Ueda probably had a heart attack when he heard these awful remixes of his songs.Trance....bah………………

Figure 10: Sample Datasets

For preprocessing we apply many steps like normalization, tokenization, etc. Some of the sample after preprocessing of dataset are shown below:

Before normalization our dataset looks like:

'Great CD: My lovely Pat has one of the GREAT voices of her generation. I have listened to this CD for YEARS and I still LOVE IT. When I\'m in a good mood it makes me feel better. A bad mood just evaporates like sugar in the rain. This CD just oozes LIFE. Vocals are jusat STUUNNING and lyrics just kill. One of life\'s hidden gems. This is a desert isle CD in my book. Why she never made it big is just beyond me. Everytime I play this, no matter black, white, young, old, male, female EVERYBODY says one thing "Who was that singing ?"'

Figure 11: dataset before normalization

Now after applying normalization techniques like stemming and lemmatization above dataset is converted as shown below

'great cd my lovely pat has one of the great voices of her generation i have listened to this cd for years and i still love it when i m in a good mood it makes me feel better a bad mood just evaporates like sugar in the rain this cd just oozes life vocals are jusat stuunning and lyrics just kill one of life s hidden gems this is a desert isle cd in my book why she never made it big is just beyond me everytime i play this no matter black white young old male female everybody says one thing who was that singing

Figure 12: dataset after normalization

3.8. DEPLOYMENT MODEL

Model deployment is simply the engineering task of exposing an ML model to real use. The term is often used quite synonymously with making a model available via real-time APIs. A sample of the deployment model for our system is shown below;

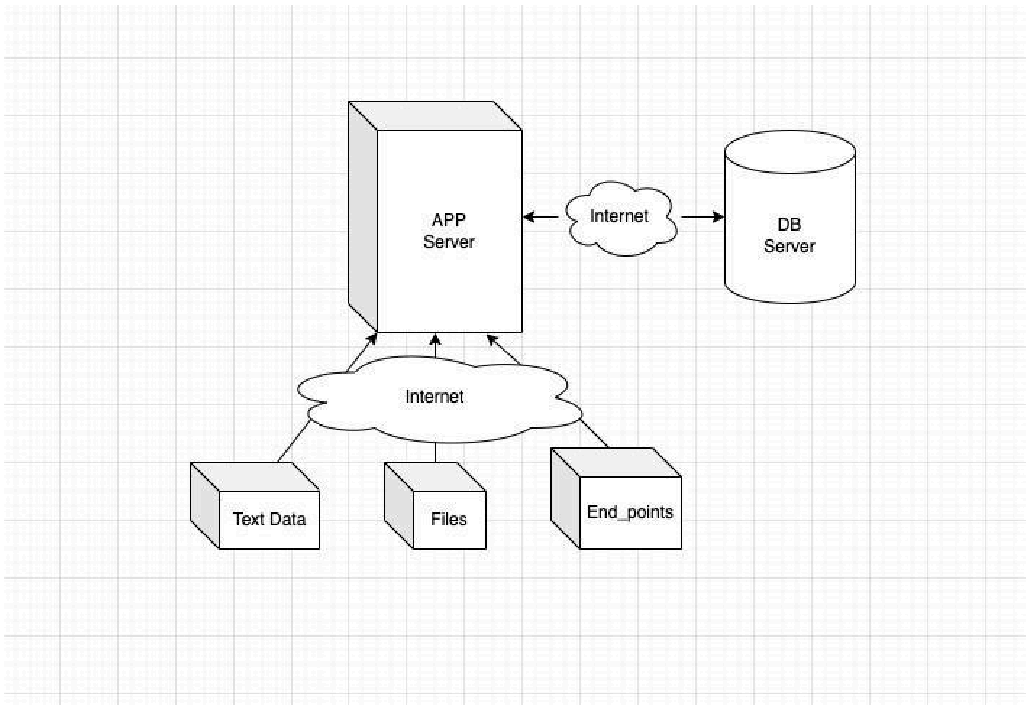

Figure 13: Deployment Model

4. TOOLS AND TECHNOLOGIES

4.1. TOOLS USED

Tools	Usage
Jupyter Notebook (v6.4.8)	IDE for model creation.
VS Code (v1.71)	An IDE for writing code
Mongo Compass (v1.32)	Platform to work with the MongoDB database.
Git (v2.37.3)	A version control system
Postman (v9.20.0)	API testing
Docker (v20.10.17)	To work with docker containers

Table 2: Tools Used

4.2. TECHNOLOGIES USED

- **Flask (v2.1.2):** Flask is an open-source Python web framework for backend web apps.
- **React (v18.2.0):** React is a free and open-source front-end JavaScript library for building user interfaces based on UI components.
- **Python (3.9.7):** Python is a computer programming language that is often used to build websites and software, automate tasks, and conduct data analysis.
- **HTML (v5)** and **CSS (3)** to develop interactive user interfaces.
- **GitHub (CE)**: platform for a team of developers to work concurrently on the same project.
- **Docker (v20.10.17):** To containerize our application to solve dependency issues.

5. WORK BREAKDOWN

Task	**Project Member**
Planning and scope discussion	Sandesh, Bibek, Manish, Puspha
Requirement analysis and system design	Bibek, Manish, Puspha, Sandesh
Data collection, cleaning and pre-processing	Bibek, Manish
Database design	Puspha, Bibek
Data training and model development	Sandesh, Puspha
Backend development	Puspha, Manish
Frontend design	Sandesh, Bibek
Frontend-Backend integration	Puspha, Sandesh
Test system module	Sandesh, Puspha, Bibek, Manish
Overall system test	Puspha, Manish, Sandesh, Bibek
Documentation	Bibek, Manish

Table 3: Work Breakdown

6. OUTCOMES

Our project delivers a web-based application that helps in the analysis of sentiments based on texts, which will have the following feature:

- Classify reviews into negative or positive sentiments.
- Users can enter their file of reviews and get an analysis of that.
- Users can enter any text data and get their respective sentiments.
- Users can also add the API of their reviews and get analysis.
- Login, signup, and profile section for users.
- A dashboard to graphically visualize the analysis.

7. TIME SCHEDULE

The project schedule has been designed as per requirement and constraint involved.

Task Name	Start Date	Finish Date	Duration(Days)
Iteration 1	7/6/2022	8/9/2022	34
Requirements Gathering	7/6/2022	7/13/2022	7
Analysis and Design	7/14/2022	7/18/2022	4
Coding	7/19/2022	8/2/2022	14
Testing	8/3/2022	8/7/2022	4
Documentation	7/11/2022	8/9/2022	29
Iteration 2	8/7/2022	9/9/2022	33
Requirements Gathering	8/7/2022	8/10/2022	3
Coding	8/11/2022	8/18/2022	7
Hyperparameter Tuning	8/19/2022	8/23/2022	4
Deployment	8/24/2022	8/28/2022	4
Testing and Code Re-factoring	8/29/2022	9/2/2022	4
Updated Documentation	8/10/2022	9/9/2022	30

Table 4: Time schedule

7.1 GANTT CHART

\

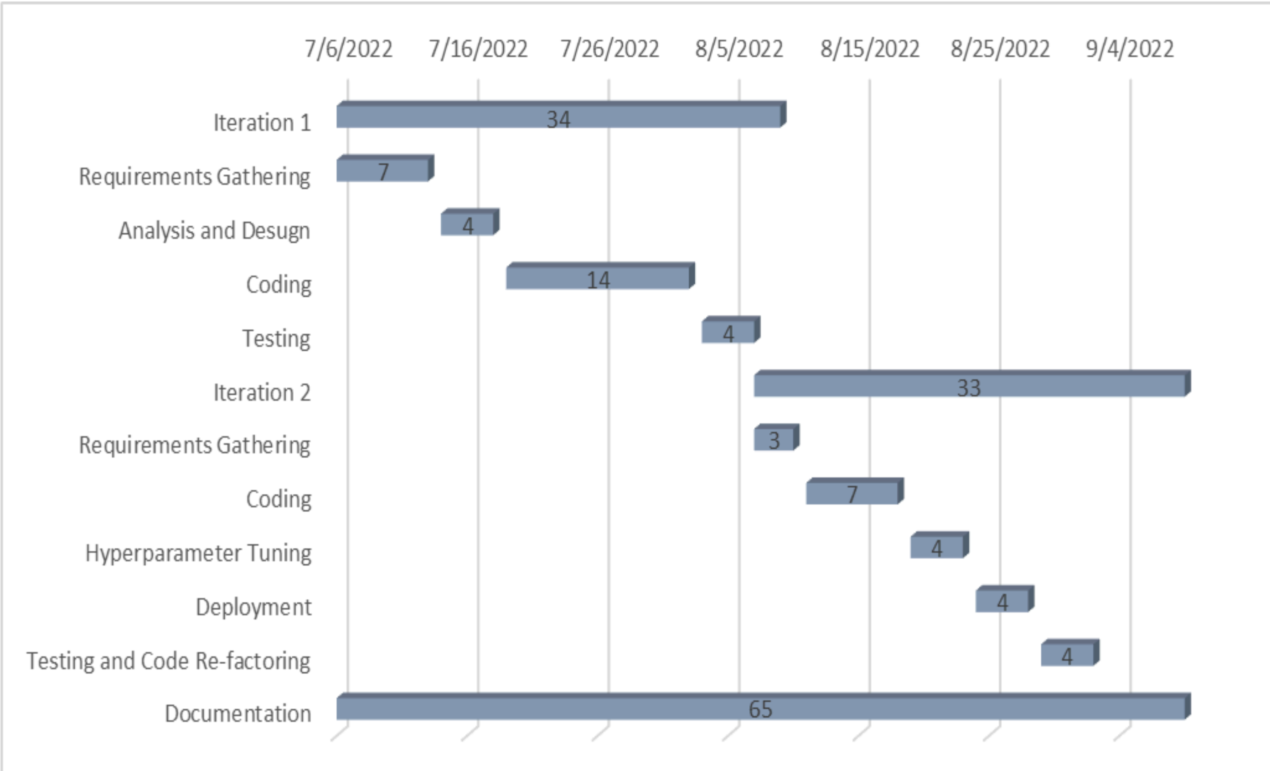

Figure 14: Gantt Chart

8. CONCLUSION

As we said, we have created a sentiment analysis system to identify the view or emotion behind a situation through text. Businesses depend highly on the reviews and feedback of their customers. So, proper analysis of the reviews and feedback in businesses is crucial. Reading and analyzing reviews manually takes a lot of time and isn't that efficient. This is why we need some process that makes computers understand natural language as we humans do, and this is what we call Natural Language Processing (NLP). And, as we know, sentiment analysis is a sub-field of NLP and, with the help of machine learning techniques, it tries to identify and extract insights.

BERT models perform fairly well in comparison to CNN models. However, they require high computational power and a large amount of time to train on a model. Thus, unless the dataset is complex and the application requires high accuracy, we can also use simpler models as they are faster to train with fewer computational power requirements and give fairly efficient results.

We really hope that our project succeeds in what it aims to do.

9. REFERENCES

Amazon Reviews for Sentiment Analysis. (n.d.). Retrieved September 24, 2022, from https://www.kaggle.com/datasets/bittlingmayer/amazonreviews

Brownlee, J. (2017, October 29). Deep Convolutional Neural Network for Sentiment Analysis (Text Classification). *Machine Learning Mastery*. https://machinelearningmastery.com/develop-word-embedding-model-predicting-movie-review-sentiment/

E-commerce product review sentiment classification based on a naïve Bayes continuous learning framework. (2020). *Information Processing & Management*, *57*(5), 102221. https://doi.org/10.1016/j.ipm.2020.102221

IRA IRYANI_15187.pdf. (n.d.). Retrieved September 24, 2022, from http://utpedia.utp.edu.my/13548/1/IRA%20IRYANI_15187.pdf

Raj, N. (2021, June 15). Sentiment Analysis | Sentiment Analysis in Natural Language Processing. *Analytics Vidhya*. https://www.analyticsvidhya.com/blog/2021/06/nlp-sentiment-analysis/

Sentiment Analysis Guide. (n.d.). MonkeyLearn. Retrieved September 24, 2022, from https://monkeylearn.com/sentiment-analysis/

Sentiment Analysis in Social Networks. (2017). Elsevier. https://doi.org/10.1016/C2015-0-01864-0

Tourism Mobile App With Aspect-Based Sentiment Classification Framework for Tourist Reviews. (n.d.). Retrieved September 24, 2022, from https://ieeexplore.ieee.org/document/8680692

What is BERT (Language Model) and How Does It Work? (n.d.). SearchEnterpriseAI. Retrieved September 24, 2022, from https://www.techtarget.com/searchenterpriseai/definition/BERT-language-model

Appendix

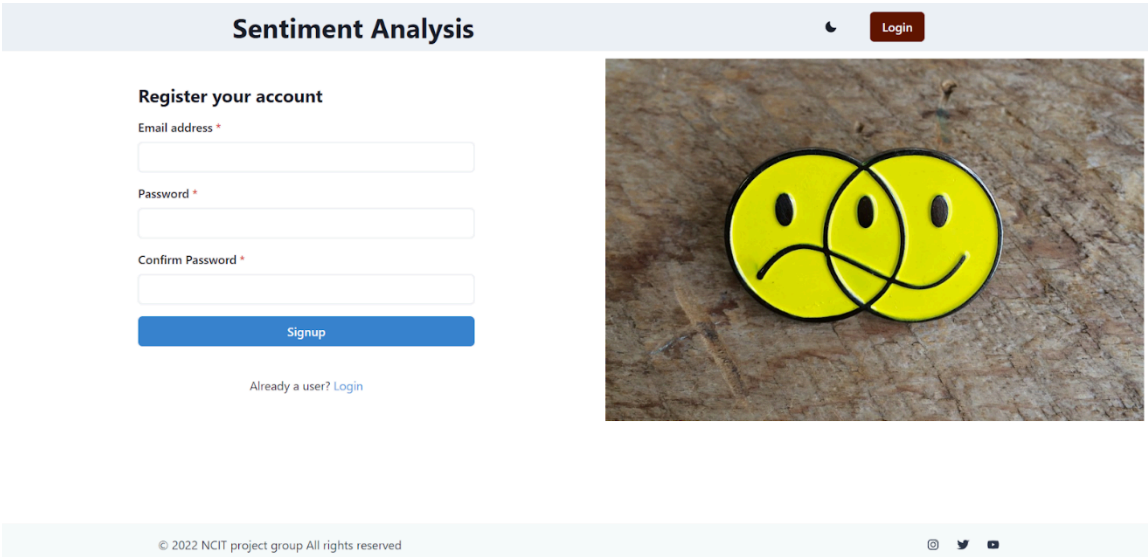

Figure 15: SignUp Page

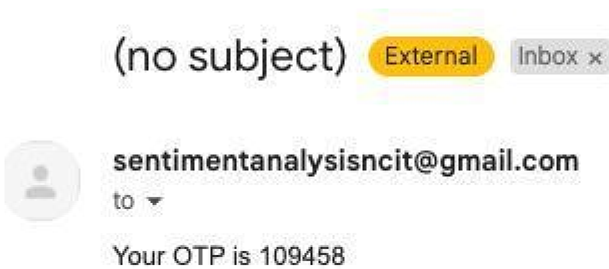

Figure 16: getting OTP

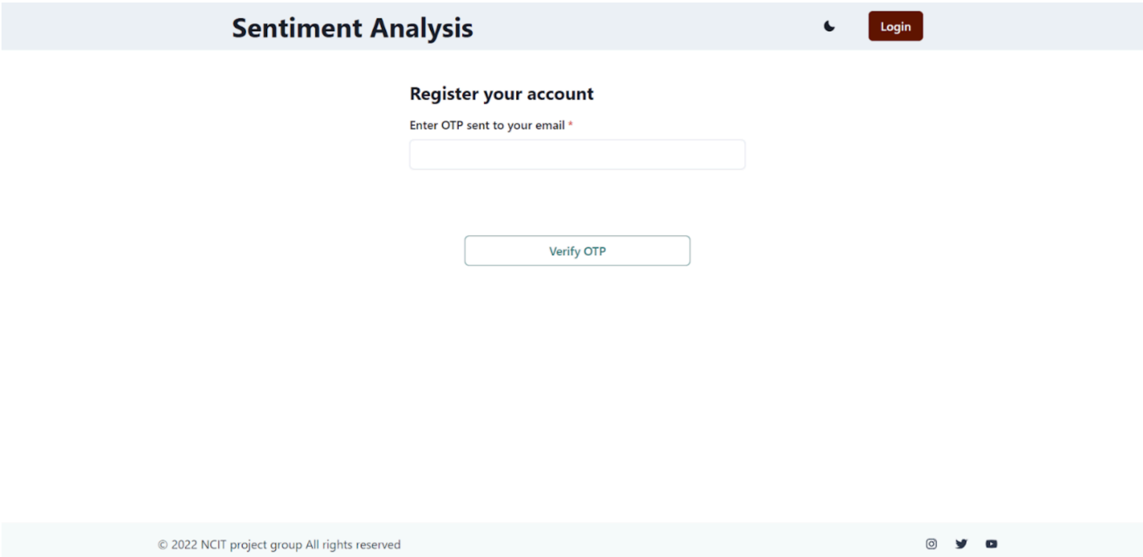

Figure 17: Verifying OTP

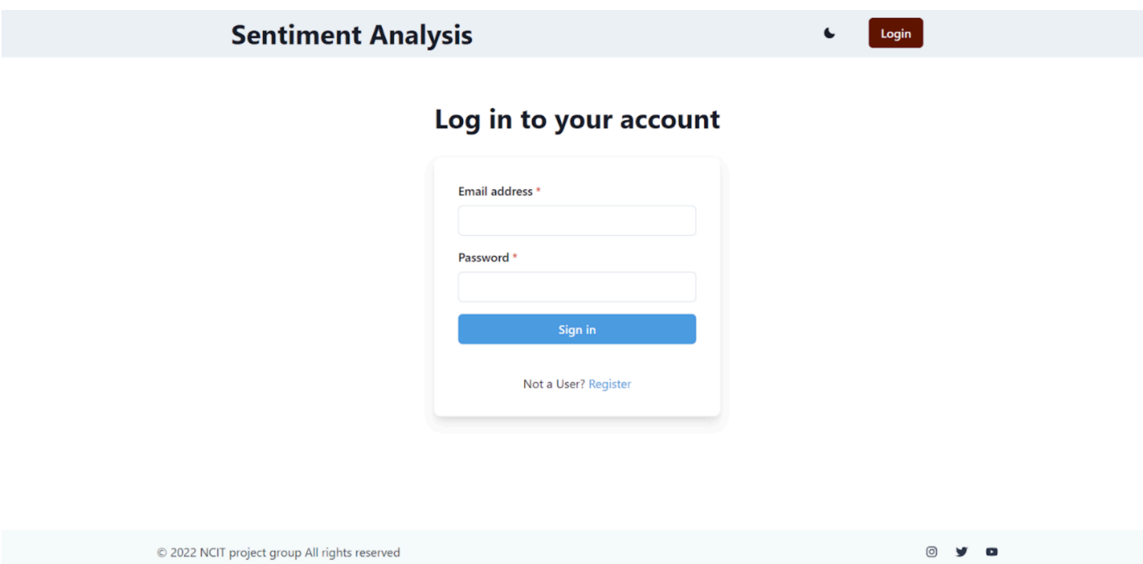

Figure 18: Login Page

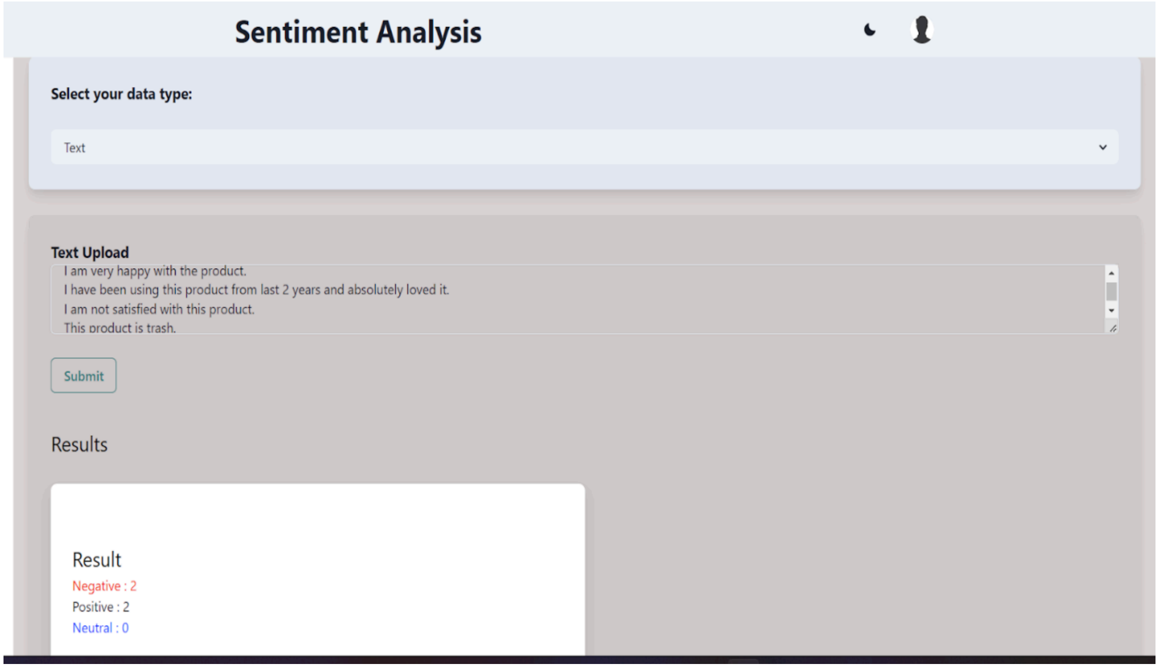

Figure 19: Text entering field

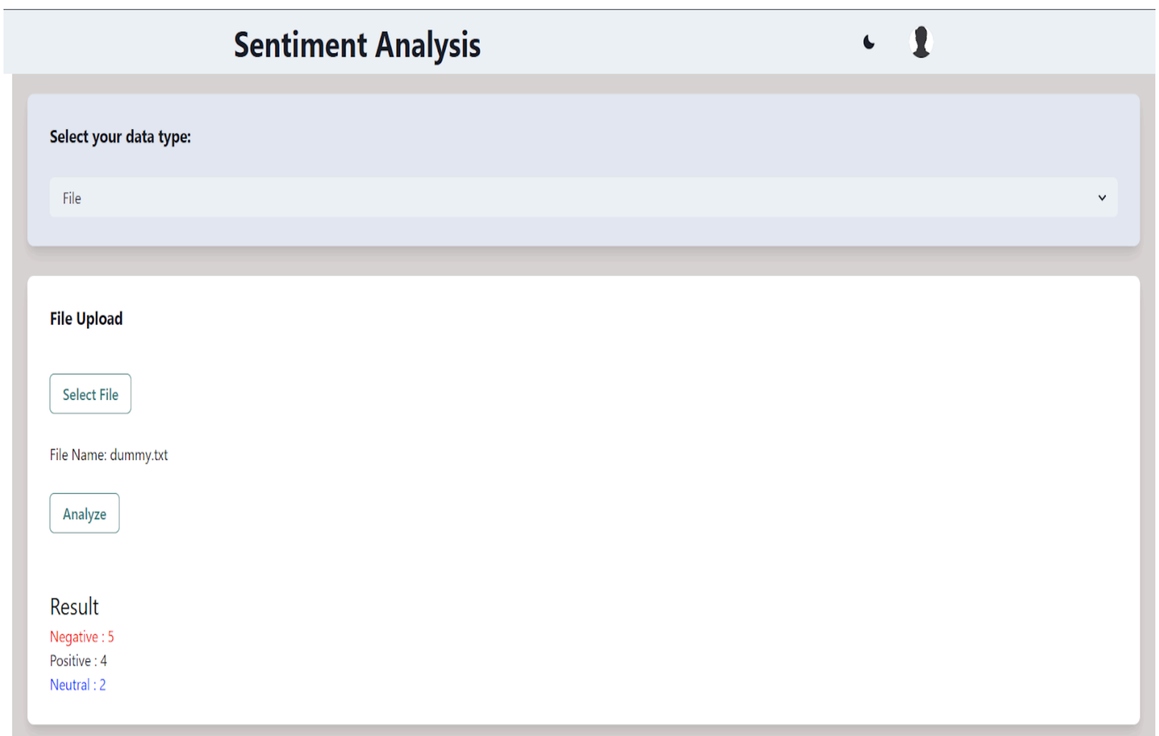

Figure 20: File upload field

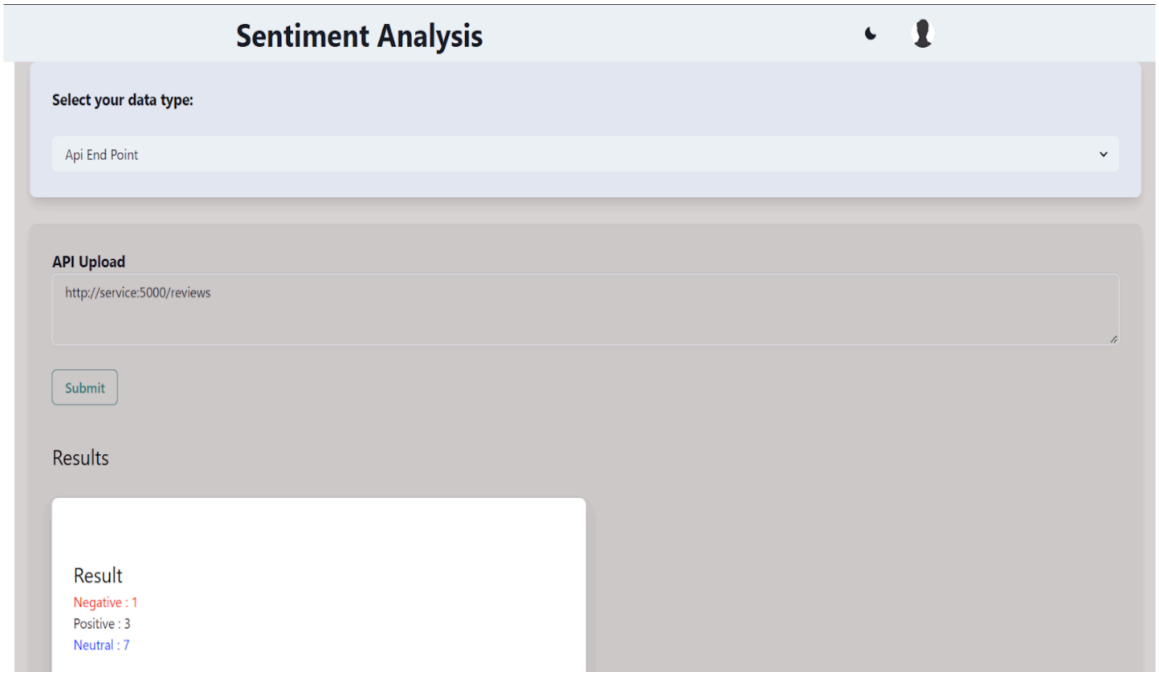

Figure 21: API input field