**Mini Project Report on**



**Product Review Analyzer Using ML**



**Submitted in partial fulfillment of the requirement for the award of the degree of**

**BACHELOR OF TECHNOLOGY**

**IN**

**COMPUTER SCIENCE & ENGINEERING**

**Submitted by:**

**Shivang Mahendra**  **University Roll No. 2019103**

***Under the Mentorship of***

**Dr. Neeraj Kumar Pandey**

**Associate Professor**



**Department of Computer Science and Engineering**

**Graphic Era (Deemed to be University)**

**Dehradun, Uttarakhand**

**July-2024**

GEU logo

**CANDIDATE’S DECLARATION**

I hereby certify that the work which is being presented in the project report entitled **“Title of the project”** in partial fulfillment of the requirements for the award of the Degree of Bachelor of Technology in Computer Science and Engineeringof the Graphic Era (Deemed to be University), Dehradun shall be carried out by the under the mentorship of **Mentor Name, Designation**, Department of Computer Science and Engineering, Graphic Era (Deemed to be University), Dehradun.

Shivang Mahendra University Roll no 2019103

**Table of Contents**

|  |  |  |
| --- | --- | --- |
| **Chapter No.** | **Description** | **Page No.** |
| Chapter 1 | Introduction | **1-2** |
| Chapter 2 | Literature Survey | **3-4** |
| Chapter 3 | Methodology | **5-10** |
| Chapter 4 | Result and Discussion | **11-12** |
| Chapter 5 | Conclusion and Future Work | **13** |
|  | References | **14** |

**Chapter 1**

**Introduction**

* 1. **Problem Statement**

We have to create a Product Review Analyzer that will analyze the reviews provided by user of the products used by them and classify them as positive or negative depending on the user’s review.

* 1. **Introduction**

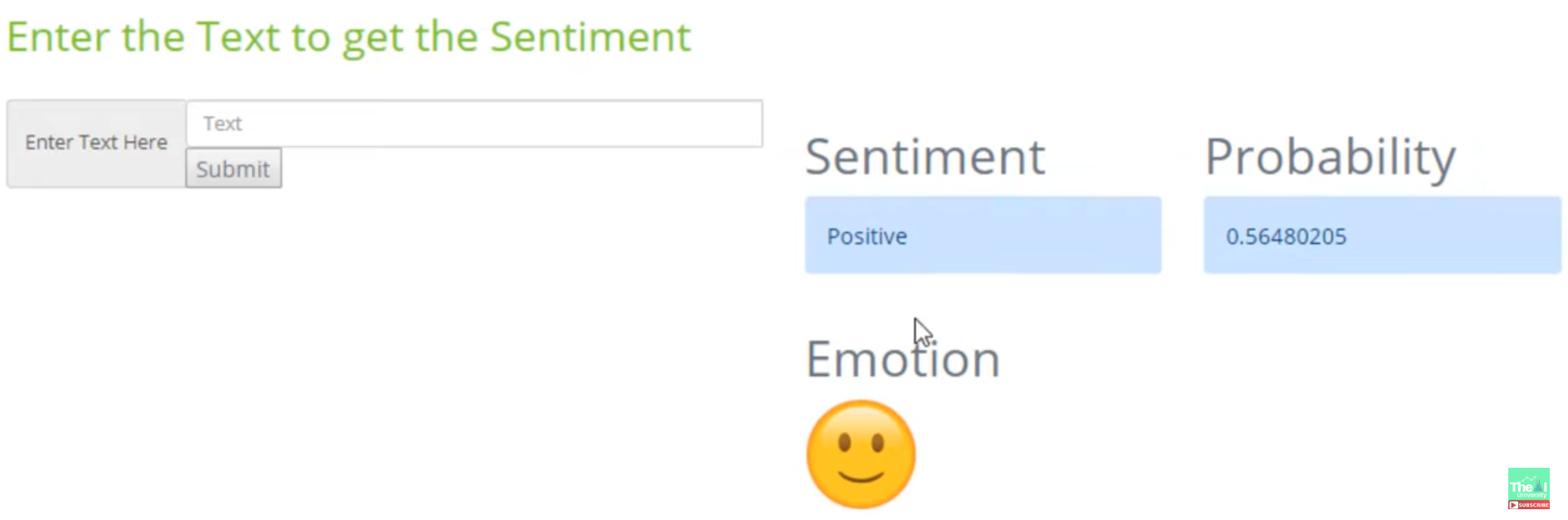
In this project, I have created a WebApp using html, CSS, bootstrap CSS for frontend and flask python framework for backend.

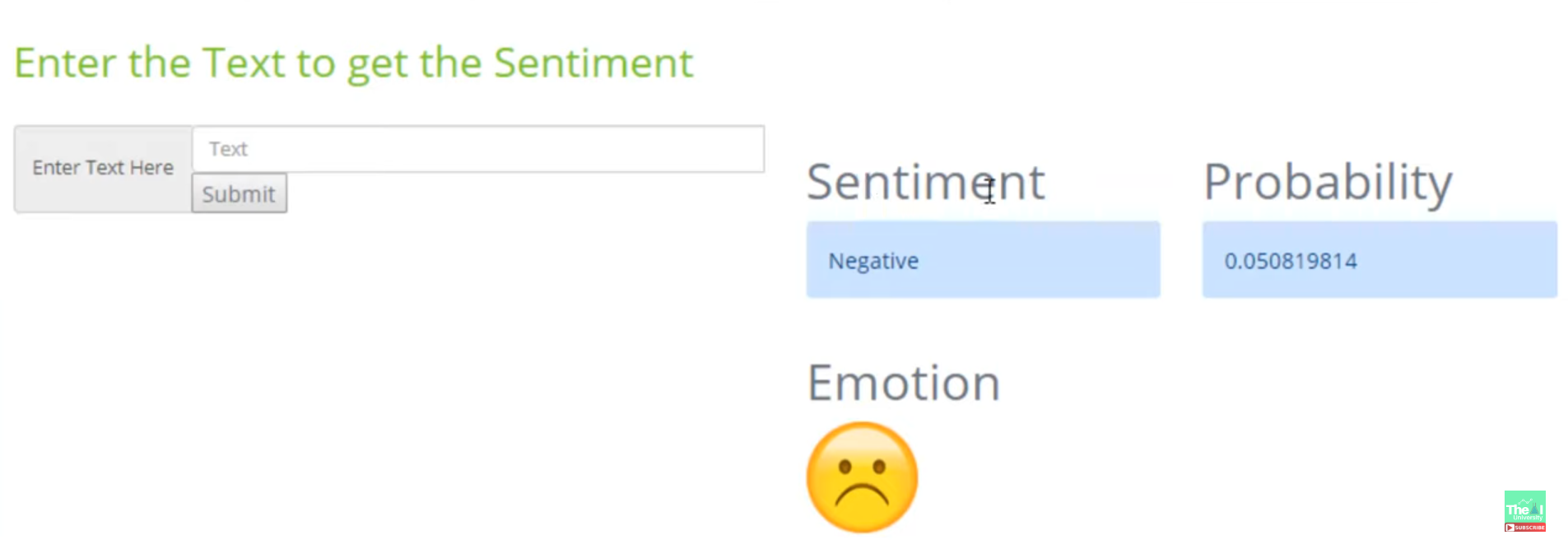
In frontend we take review as input from user or owner and in the backend using machine learning Long Short Term Memory (LSTM) algorithm we analyze the input and provide the results as positive or negative on the WebApp.

In backend, we preprocess the input which we receive from frontend and then pass it to our model which is based on LSTM then process it and give the output back to the WebApp and show it to the user.

This is very useful for product owners as they can insert large number of reviews and get the analysis of whether their product is being liked by the public or not and what further improvements are needed to be done. This will help owners to provide better products and services to customers and earn more.

This can be very useful for customers also as they can get valuable information from vast amounts of data reviews within seconds and can decide on what to purchase and what not to and also for comparison between different products and services.





**Chapter 2**

**Literature Survey**

**2.1 Introduction**

Product Review Analyzer is a very important feature of modern e-commerce platforms like Amazon, Flipkart, etc. which helps customers as well as sellers/owners to know valuable information about the product and how much satisfied other users are.

The use of Natural Language Processing (NLP) and various machine learning algorithms has improved product review analysis systems very much.

**2.2 Background and Motivation**

Product Reviews are a rich source of information for both business ownners and customers as well. Product Review Analyzer helps in analyzing and extracting meaningful information from vast amounts of reviews which helps both customers who want to buy the product and the business owners who want to improve their products for more earning and provide better user experience.

**2.3 Review of Related Work**

Sentiment Analysis is the fundamental component of product review analyzer systems.

Various approaches have been used such as:

* Lexicon-Based Methods: This method uses predefined dictionaries of sentiment related words to classify text. But struggle with domain related language. (Taboada et al., 2011)
* Machine Learning Approaches: Various supervised learning techniques such as SVM(support vector machines), Naïve Bayes, and Random Forests have been tried and tested for sentiment analysis. But these techniques requires labeled datasets and they show high level of accuracy based on sufficient datasets used for training models. (Pang et al., 2002)
* Deep Learning Techniques: Most recent approaches use deep learning algorithms and models for sentiment analysis such as CNN (Convolutional Neural Networks). These techniques have shown much better accuracy for sentiment analysis tasks. (Zhang et al., 2018)

Feature Extraction is another crucial task for transforming raw text into meaningful representations using common=n techniques such as:

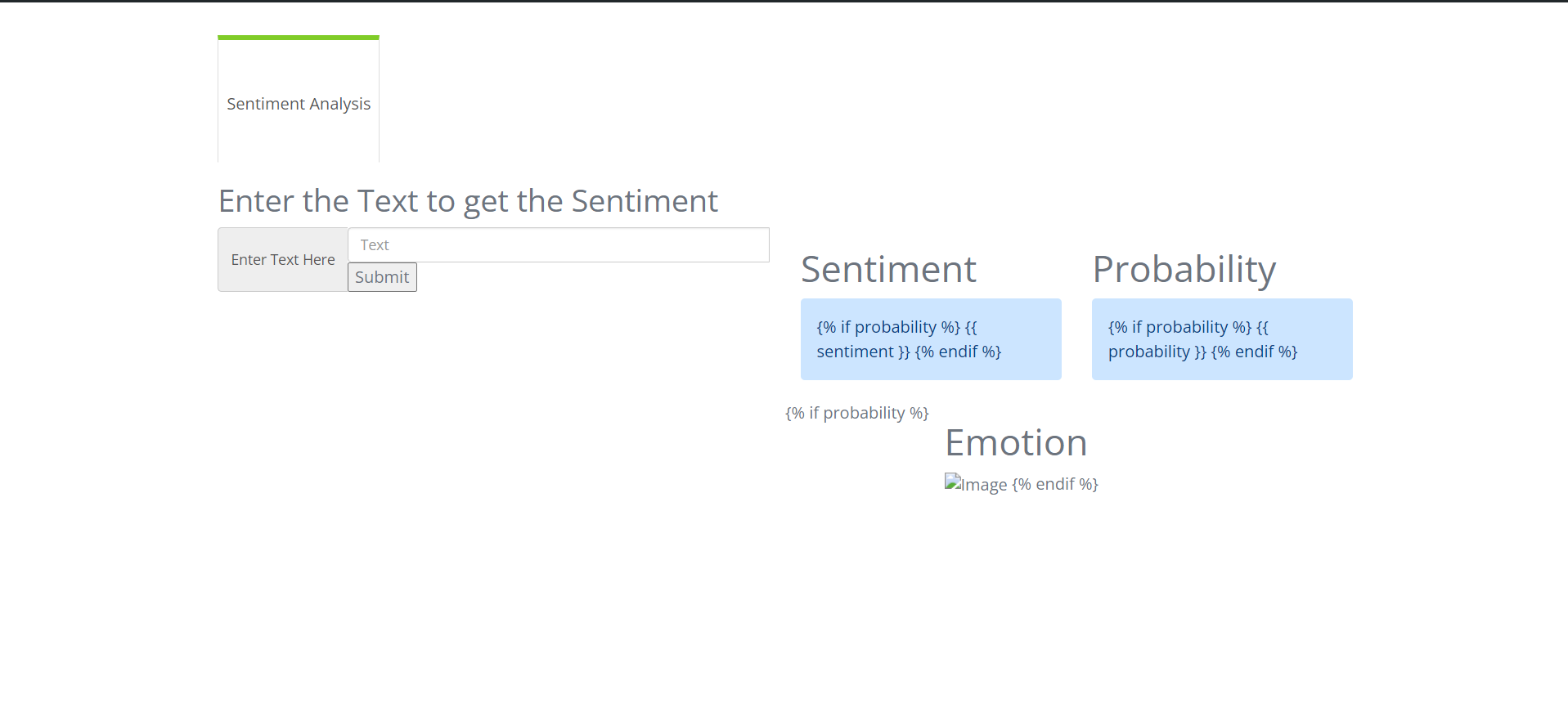
* Bag-of-Words (BoW): This representation represents text as a collection of words disregarding grammar and order of words.It is a very simple approach and useful in many places but not very helpful in sentiment analysis as it does not maintain semantic relationship and order of words. (Harris, 1954)
* TF-IDF (Term Frequency – Inverse Document Frequenncy): It improves BoW by considering the importance of a word in a document relative to its occurrence in entire document. (Salton & McGill, 1986)
* Word2Vec: It creates a dense vector representation for words, with capturing semantic relationships of words. These are used as standards in modern NLP applications. (Mikolov et al., 2013)

**Chapter 3**

**Methodology**

**3.1 Front-End**

For Front-End of Product Review Analyzer, I have used HTML, CSS and bootstrap framework CSS. In this front end we take review from user and pass it to backend which processes it and gives back the sentiment classification to front end and user can view it.

****

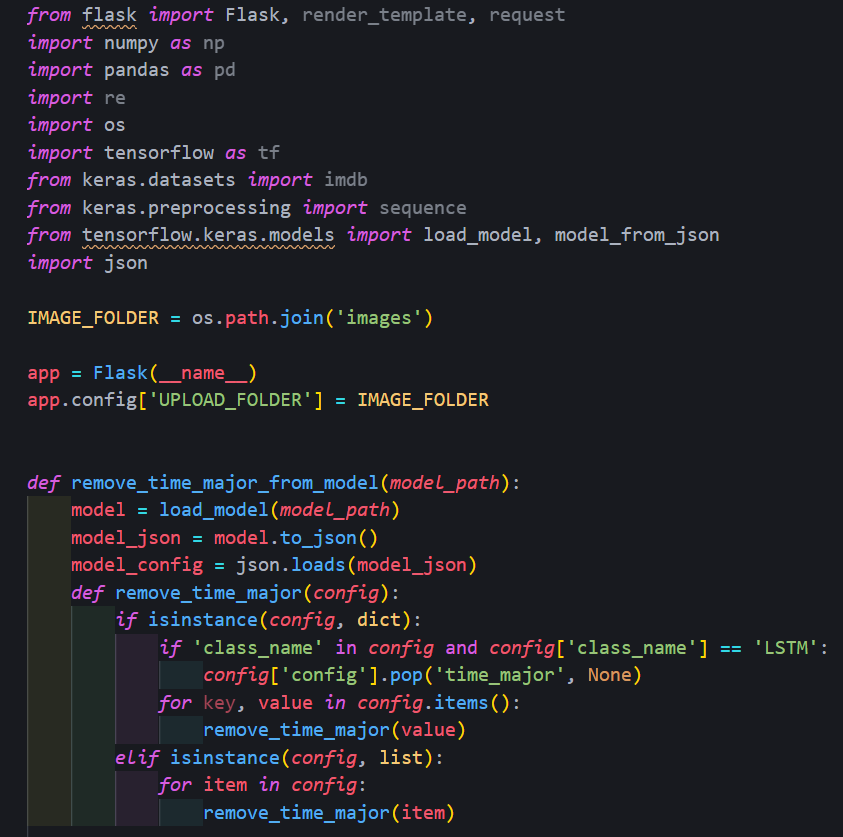
**Fig 3.1: Front-End of Product Review Analyzer**

**3.2 Back-End**

For Back-End of this Project on Product Review Analyzer, I have python framework Flask. Back-End passes the input to the LSTM model and gets the output from it i.e. sentiment class (positive or negative).

Flask is a light-weight web microframework for python, designed to make it easy to create web applications. It is highly extensible, allowing developers to add functionality they need through extensions. Its core features include:

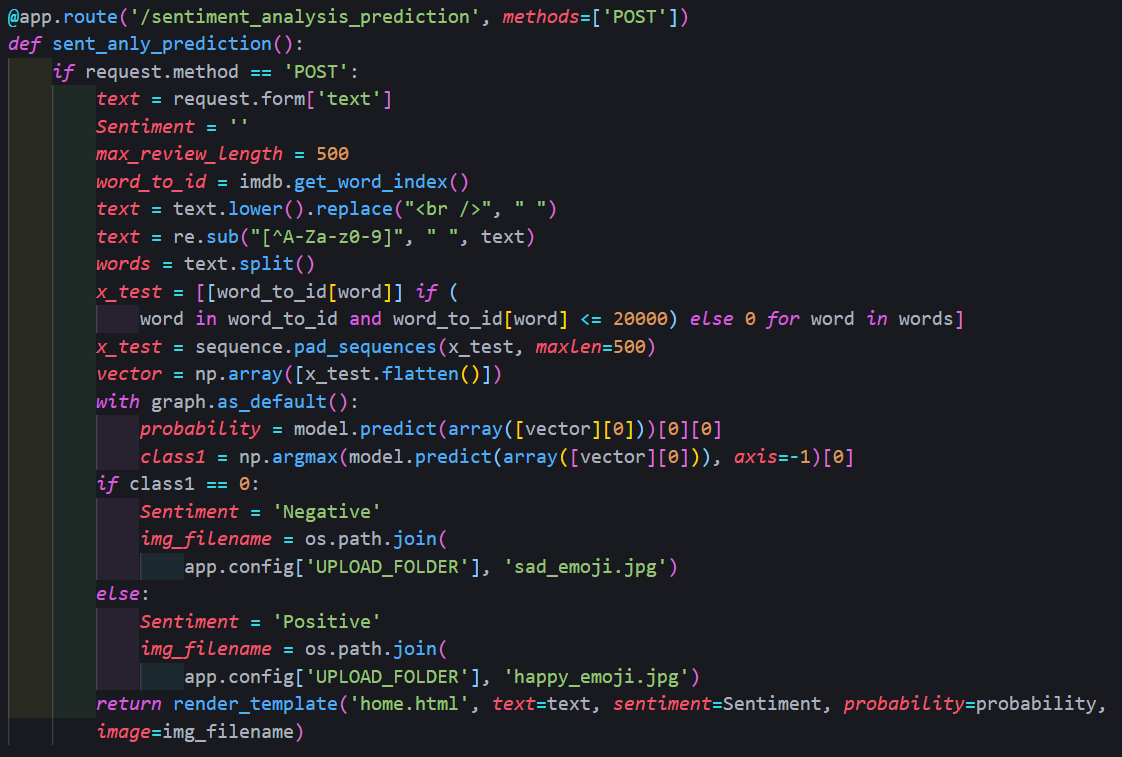
* Routing
* Templates
* Request Handling
* Development Server
* Blueprints



A screen shot of a computer program

Description automatically generated

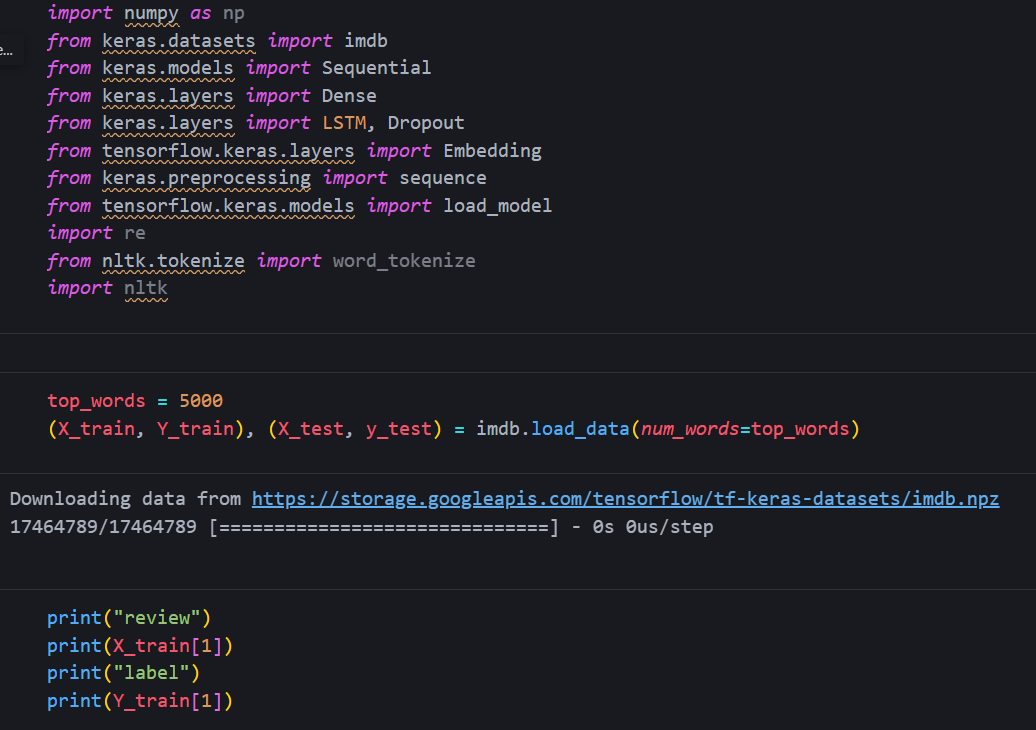
**Fig 3.2: Loading the model and presenting the initial Home Page to user**

****

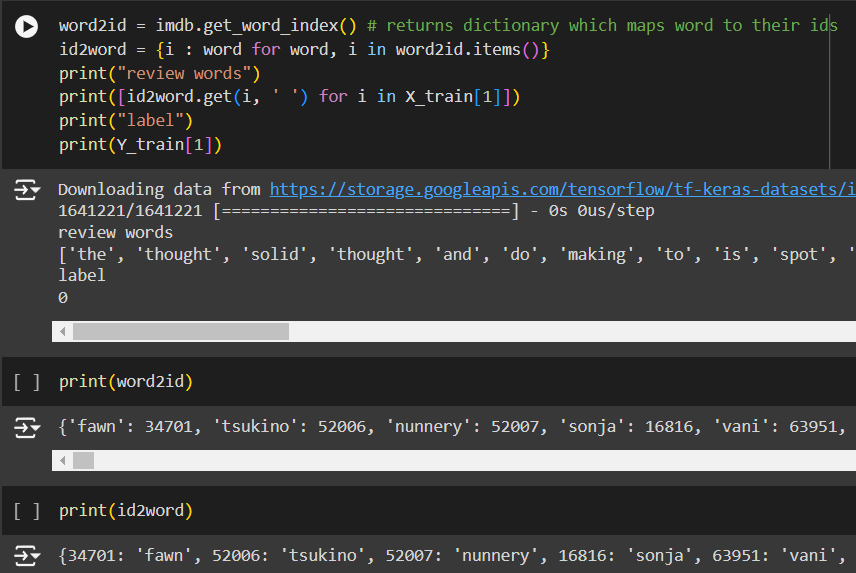
**Fig 3.3: Returning the output of the model to front-end**

**3.3 LSTM Model**

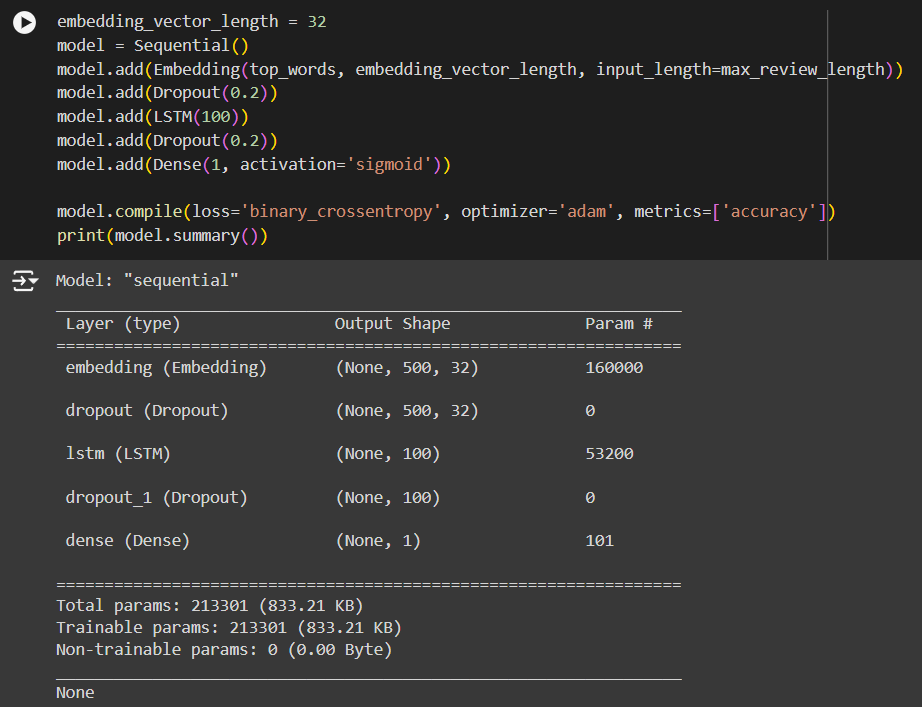
The input is passed to the trained LSTM model which preprocesses the input and passes it the model and returns the class. The model is trained on the imdb dataset using 1 embedding vector layer, then 1 LSTM layer and 1 dense layer (which gives the predicted classes) and 2 dropout layers in between to prevent over-fitting and under-fitting of the data in layers. This model gives us a high accuracy of 86.5%.

****

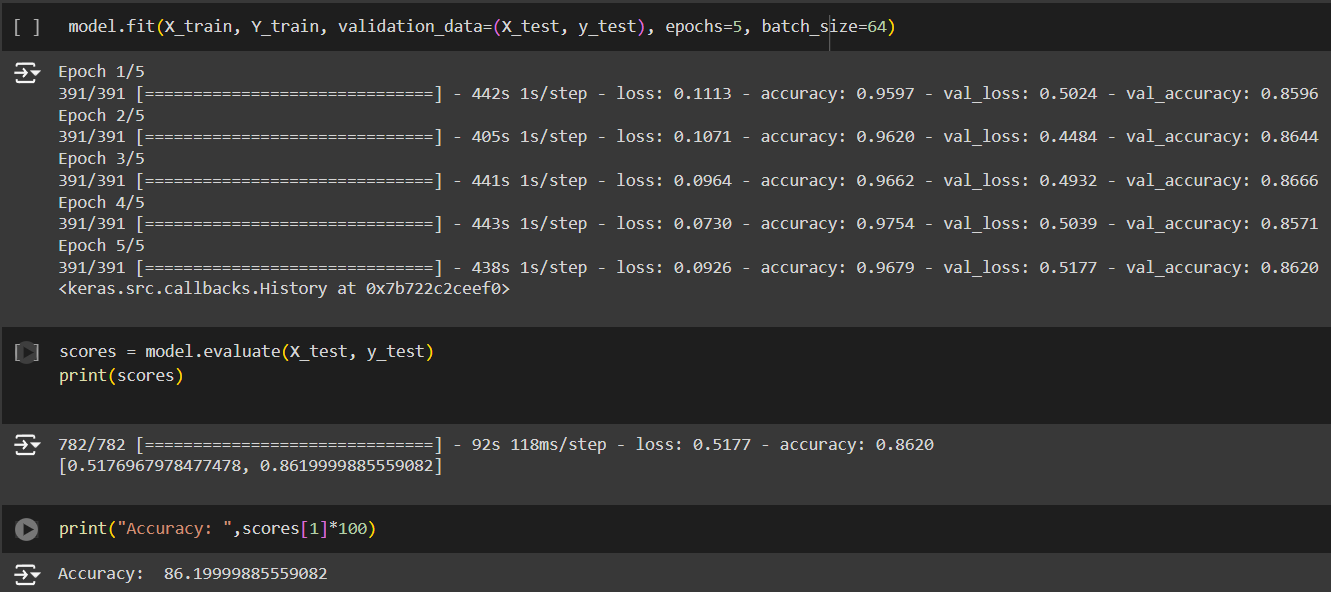
**Fig 3.4: Importing Libraries and loading and dividing dataset**

****

**Fig 3.5: converting data from their numerical ids to their meaningful words**

****

**Fig 3.6: Creating Model by adding various layers for better output**

****

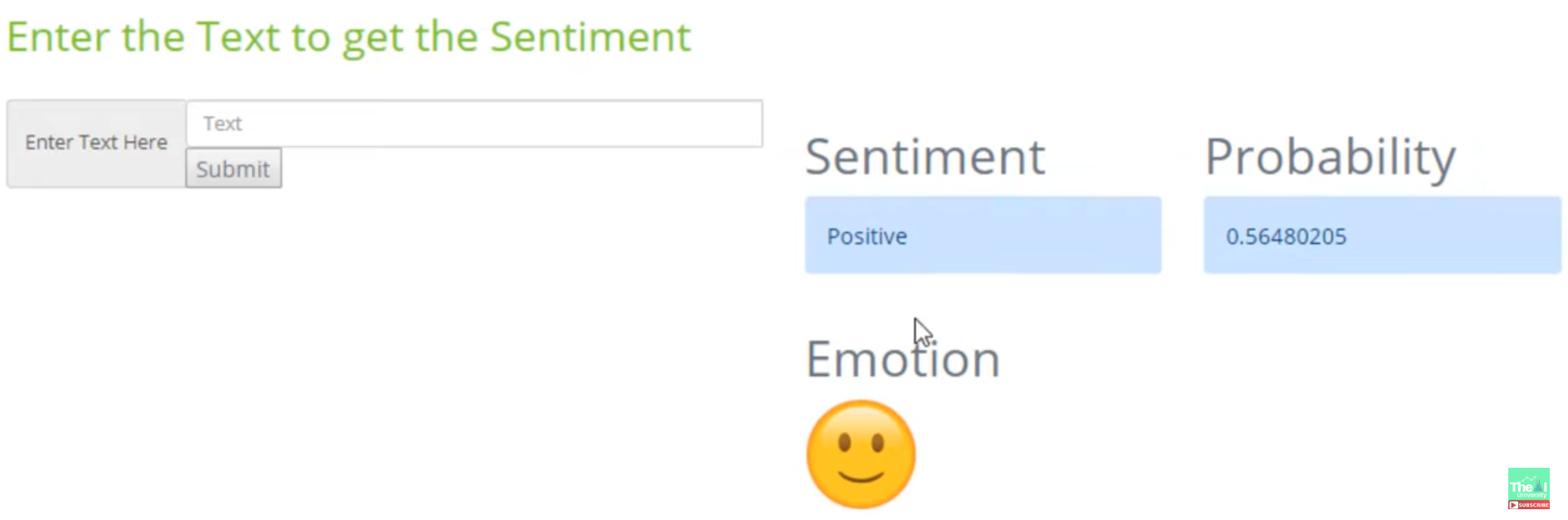
**Fig 3.7: Training the model and checking accuracy**

**Chapter 4**

**Result and Discussion**

The above shown methodology uses the LSTM model and it gives us a high frequency of 86.5% which is much higher than other machine learning and deep learning models used in which the accuracy ranges from 60-75%.

For complete code, please [Click Here](https://drive.google.com/drive/folders/16U9zCVpJso8YYTpQ8dZpB1aZofPRz5RY?usp=sharing).



A screenshot of a computer

Description automatically generated

**Chapter 5**

**Conclusion and Future Work**

**Conclusion**

With the help of above-mentioned methodology, we have successfully completed the task of creating a Product Review Analyzer System and a WebApp for easy access for users with a high accuracy of 86.5%.

**Future Work**

* Better Quality and large Dataset: Better quality and large datasets will help in the training of LSTM model and give better frequency.
* Handling Sarcasm and Irony: Detecting sarcasm in reviews is difficult due to the tone of context and is often missed by automated systems.
* Real-Time Analysis: Processing of large amounts of revies in real-time will require better infrastructure, powerful servers, efficient algorithms and high preprocessing of data on which the WebApp is deployed.
* Multilingual Sentiment Analysis: Making the sentiment analysis to accept and process reviews in different local languages.

**References**

[1] Liu, B. (2012). Sentiment Analysis and Opinion Mining. Synthesis Lectures on Human Language Technologies, 5(1), 1-167

[2] Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient Estimation of Word Representations in Vector Space. arXiv preprint arXiv:1301.3781.

[3] Pang, B., Lee, L., & Vaithyanathan, S. (2002). Sentiment Classification Using Machine Learning Techniques. Proceedings of the ACL-02 Conference on Empirical Methods in Natural Language Processing (EMNLP), 10(10), 79-86.

[4]Zhang, Y., Wang, S., & Liu, X. (2018). Deep Learning for Sentiment Analysis: A Survey. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, 8(4), e1253.

[5] The AI University, youtube channel, <https://www.youtube.com/@TheAIUniversity>