**ARTIFICIAL INTELLIGENCE**

**CREATE A CHATBOT USING PYTHON**

****

Name: Kalavathy K

Reg. No: 513521104019

Department: CSE

Year: III

NM ID: au513521104019

E-Mail: [kaviyakalavathy@gmail.com](mailto:karpagapriya062003@gmail.com)

Phase 3 (Devlopment Part-1)

**INTRODUCTION:**

"Welcome to the world of Chatbot development with Python! In this project, we embark on a journey to create an intelligent and interactive chatbot powered by Python's Natural Language Processing (NLP) capabilities. Our goal is to design a chatbot that can engage in meaningful conversations, answer questions, and provide assistance across a range of domains. Through data acquisition, model training, and iterative improvement, we aim to craft a chatbot that not only understands user queries but also offers contextually relevant responses. Join us as we explore the fascinating realm of conversational AI and create a chatbot that's ready to assist and engage with users."

In the continuation of the previous documentation AI\_Phase2, we will be looking forward on the development part of the project. In this phase, we will start building the model by loading the dataset and processing the data

**DATASET LINK:** [**https://www.kaggle.com/datasets/grafstor/simple-dialogs-for-chatbot**](https://www.kaggle.com/datasets/grafstor/simple-dialogs-for-chatbot)

The following steps are followed in this development part.

1. **EMPATHIZE**

Understanding user needs is a critical aspect of designing and developing a chatbot that is effective and user-friendly. Developers should consider the following key user needs:

**Clear Communication:** Users need a chatbot that communicates clearly and in a language they understand. The chatbot should avoid jargon and complex terminology, providing responses that are easy to follow.

**Efficiency**: Users value efficiency in their interactions with a chatbot. They expect quick and accurate responses to their queries or requests. Developers should prioritize minimizing response times and streamlining interactions.

**Relevance**: Users want the chatbot to provide information or assistance that is relevant to their needs or context. Developers should ensure that the chatbot's responses and actions are tailored to individual users when possible.

**Problem** **Solving**: Users often turn to chatbots to solve problems or complete tasks. Developers should design the chatbot to be capable of addressing common user issues and helping users accomplish their goals.

**Personalization**: Personalization is increasingly important to users. Developers can enhance user experience by allowing chatbots to remember user preferences, past interactions, and other relevant information to deliver a more personalized experience.

**User**-**Friendly** **Interface**: The chatbot's user interface should be intuitive and user-friendly. Users need to be able to navigate the chatbot easily and understand how to use it without confusion.

**Privacy** **and** **Security**: Users are concerned about their data and privacy. Developers must implement robust security measures to protect user information and communicate transparently about data handling practices.

**Availability**: Users expect chatbots to be available when needed. Ensure the chatbot is accessible 24/7 if possible, or communicate its operating hours clearly to users.

**Error** **Handling**: Users appreciate a chatbot that can gracefully handle errors and unexpected inputs. Developers should design the chatbot to respond appropriately when it encounters user mistakes or misunderstandings.

**Feedback** **and** **Help**: Users should have the ability to provide feedback on the chatbot's performance or request assistance when needed. Developers can implement mechanisms for users to report issues or seek human assistance when the chatbot cannot provide a solution.

Accessibility, Consistency, MultilingualSupport, Integration, Simplicity, Empathy, Quality Content etc are some of the user needs in development of chatbot.

By comprehensively understanding these user needs, developers can create chatbots that not only meet user expectations but also provide valuable and satisfying interactions. Regular user feedback and testing are essential to continuously refine the chatbot based on evolving user needs.

**2.DEFINE**:

"The problem to be solved is that users need an efficient and user-friendly way to schedule appointments with healthcare providers. Existing scheduling systems are often cumbersome, require phone calls, and are prone to errors, leading to patient frustration and missed appointments. The chatbot's objective is to streamline the appointment scheduling process, provide real-time availability information, and send timely reminders to both patients and healthcare providers, ultimately improving the patient experience and reducing appointment no-shows."This problem statement is clear and specific. It identifies the users' pain points (inefficient scheduling, missed appointments), the limitations of the current system, and the chatbot's role in addressing these issues (streamlining scheduling, real-time information, reminders). It also aligns with the user needs and objectives, making it a robust foundation for the chatbot's design and development.

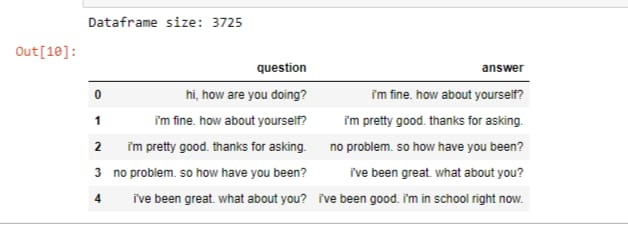
**3.Importing Libraries:**

In this step we will import all the required libraries and packages from the pre-installed modules.

**Importing libraries**



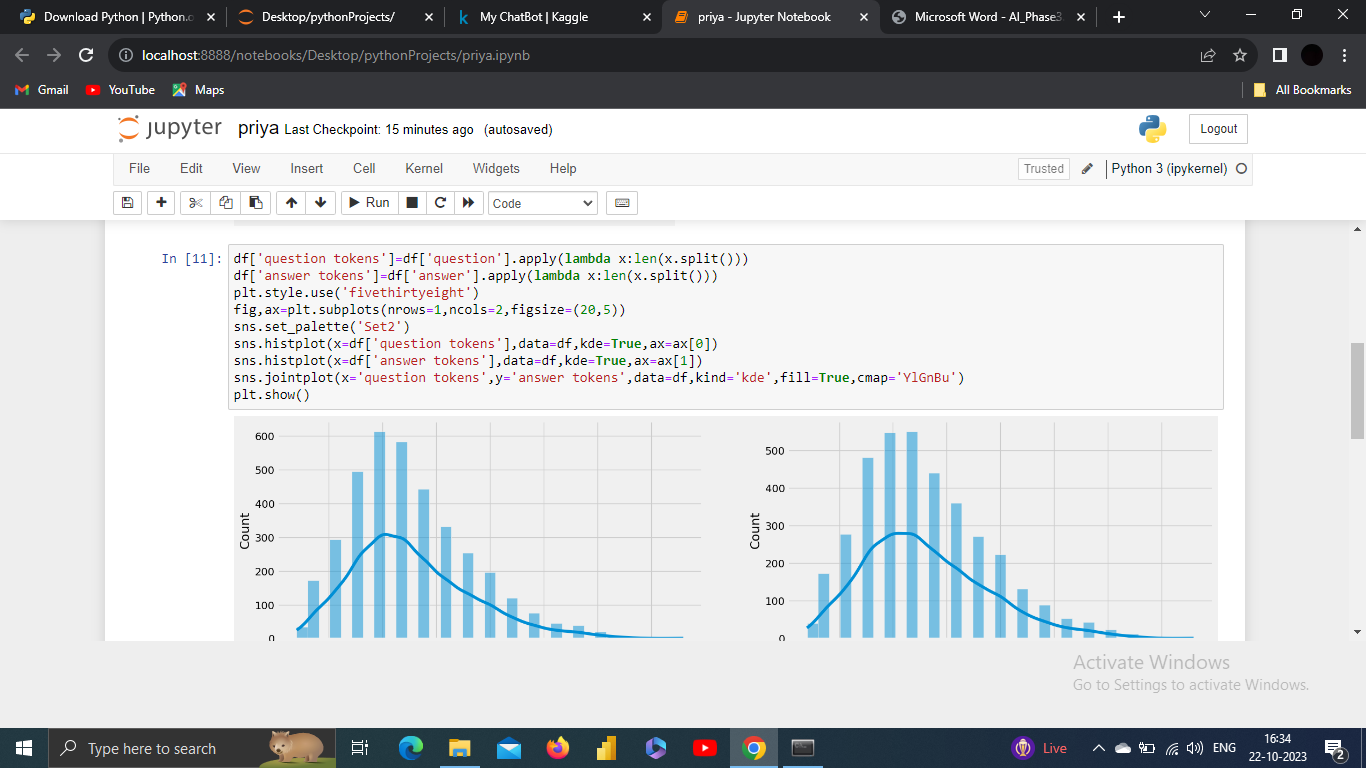
**4.Loading** **datasets**:



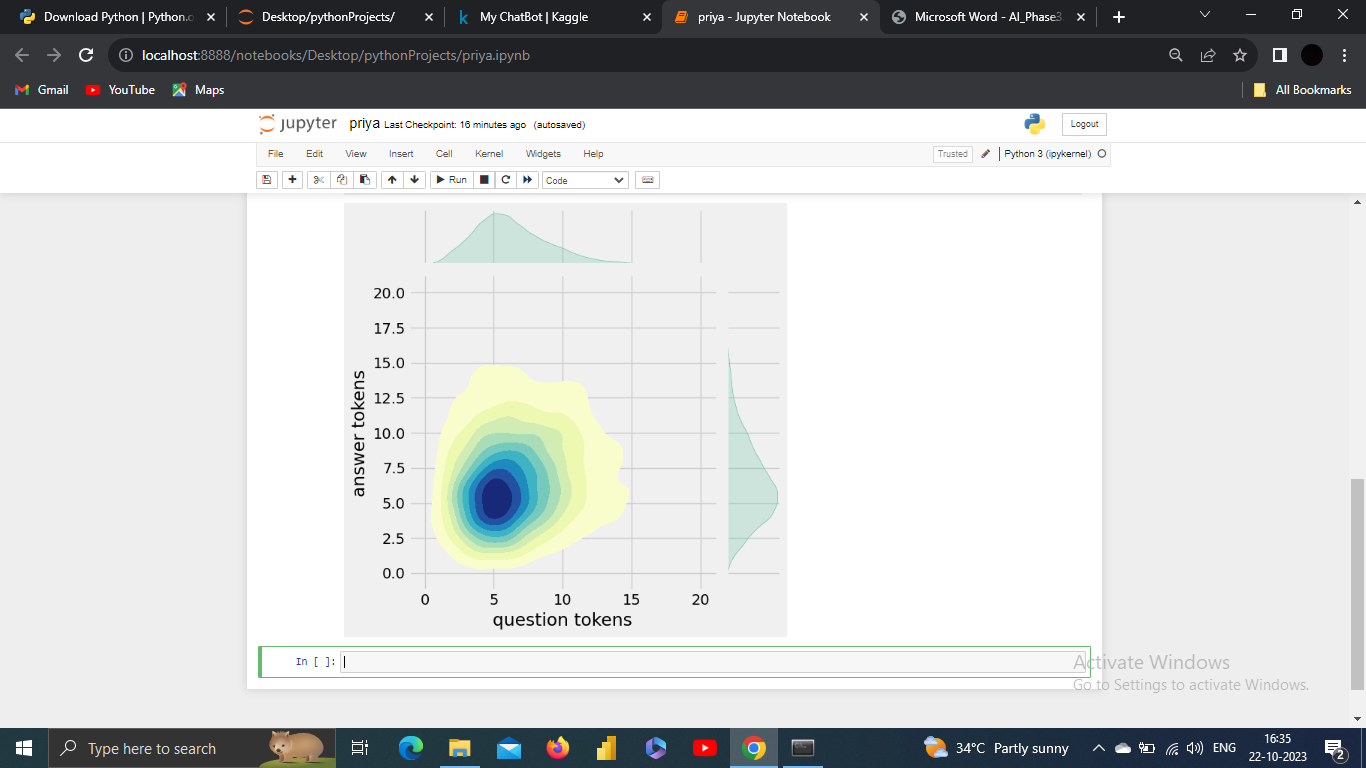
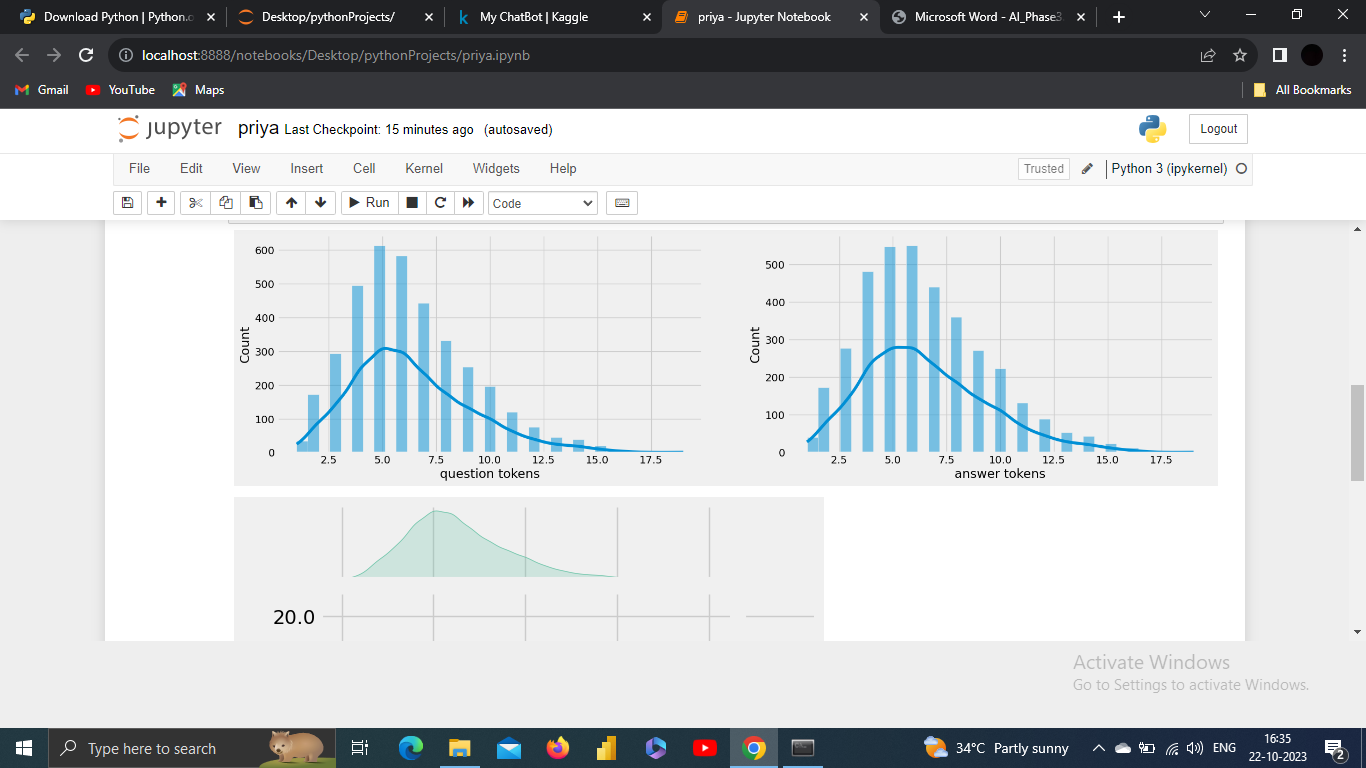
**5. Ideate**:

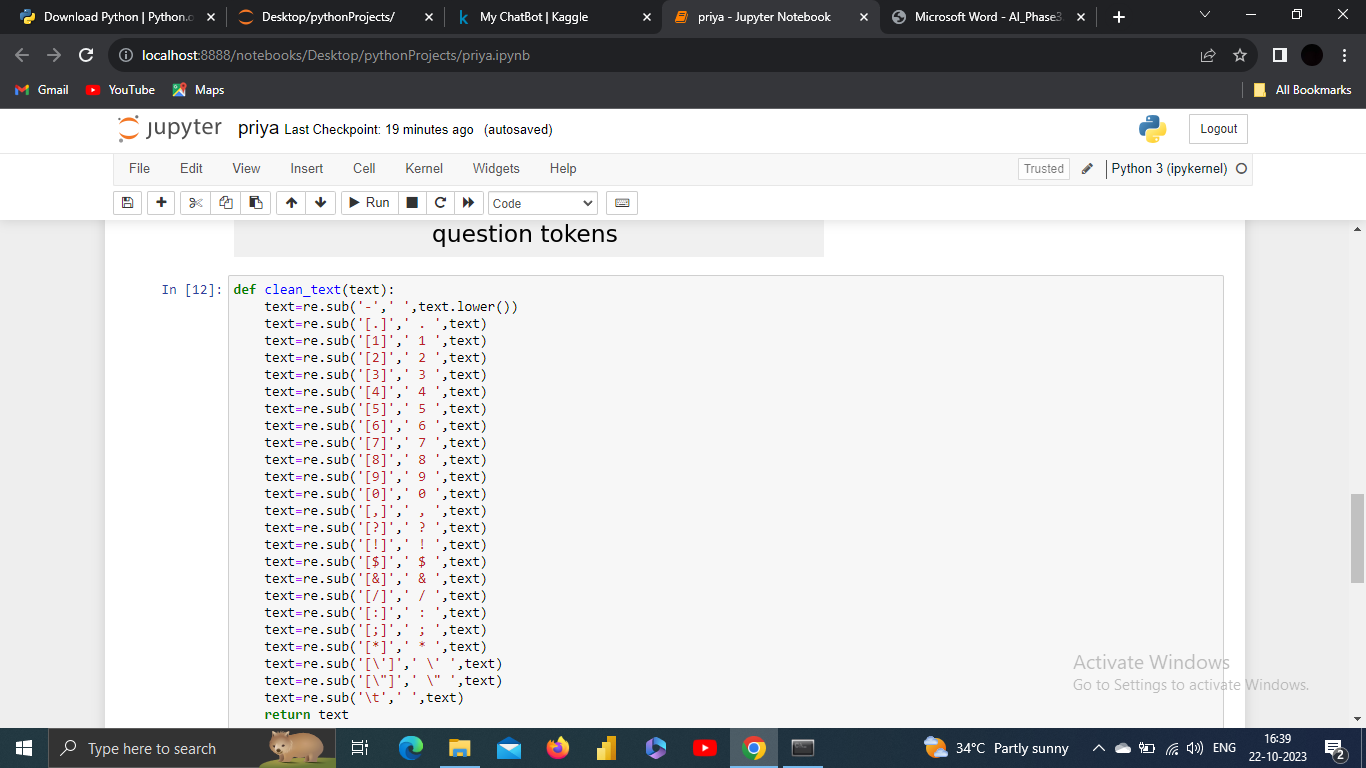
**Data Preprocessing: (Data Visualization)**

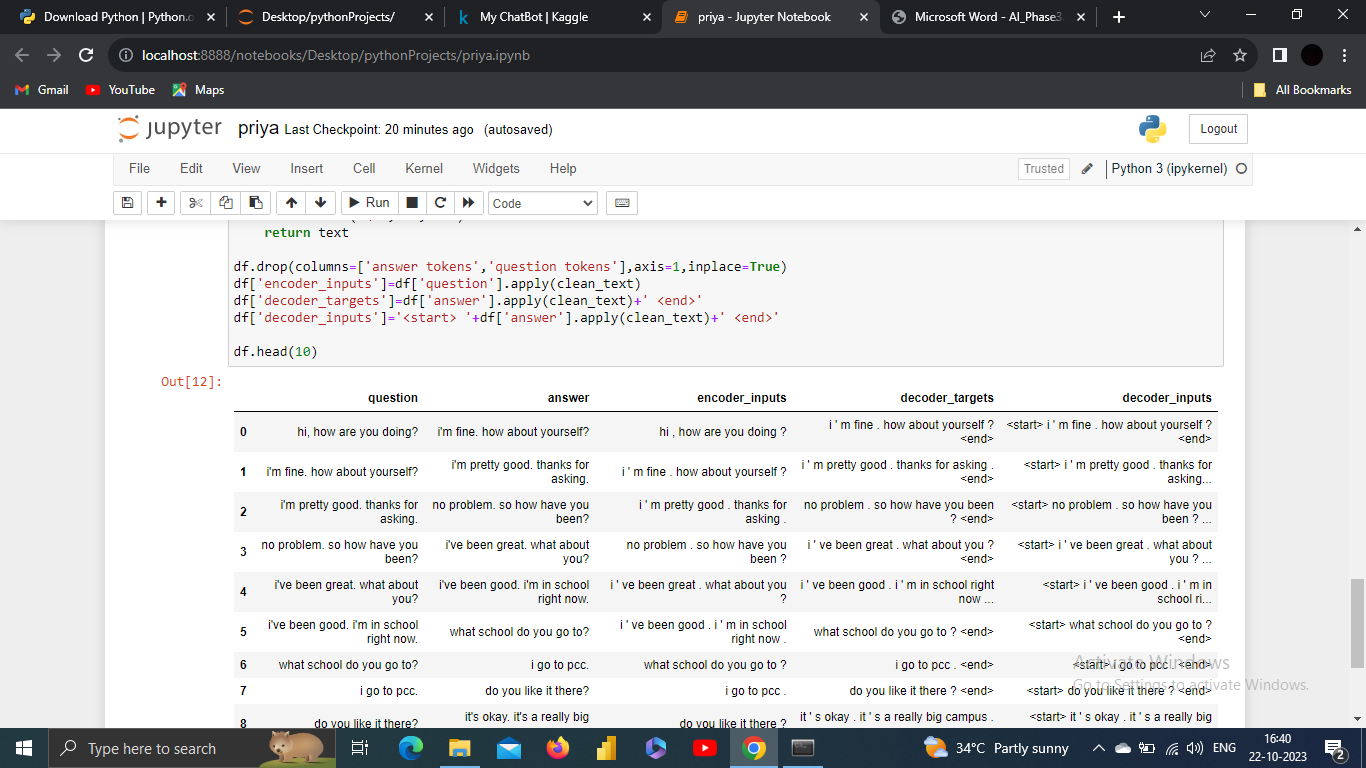
We will be pre-processing the data for the further development of model.

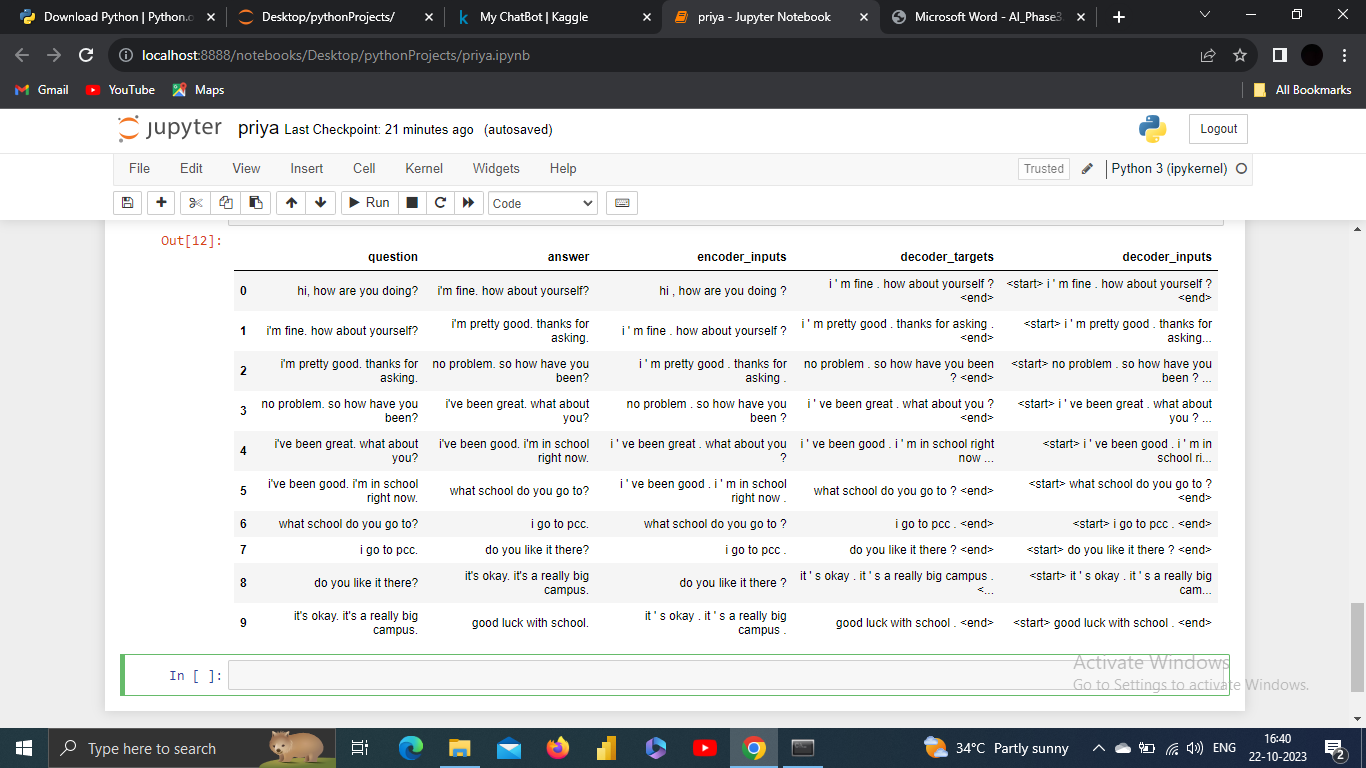


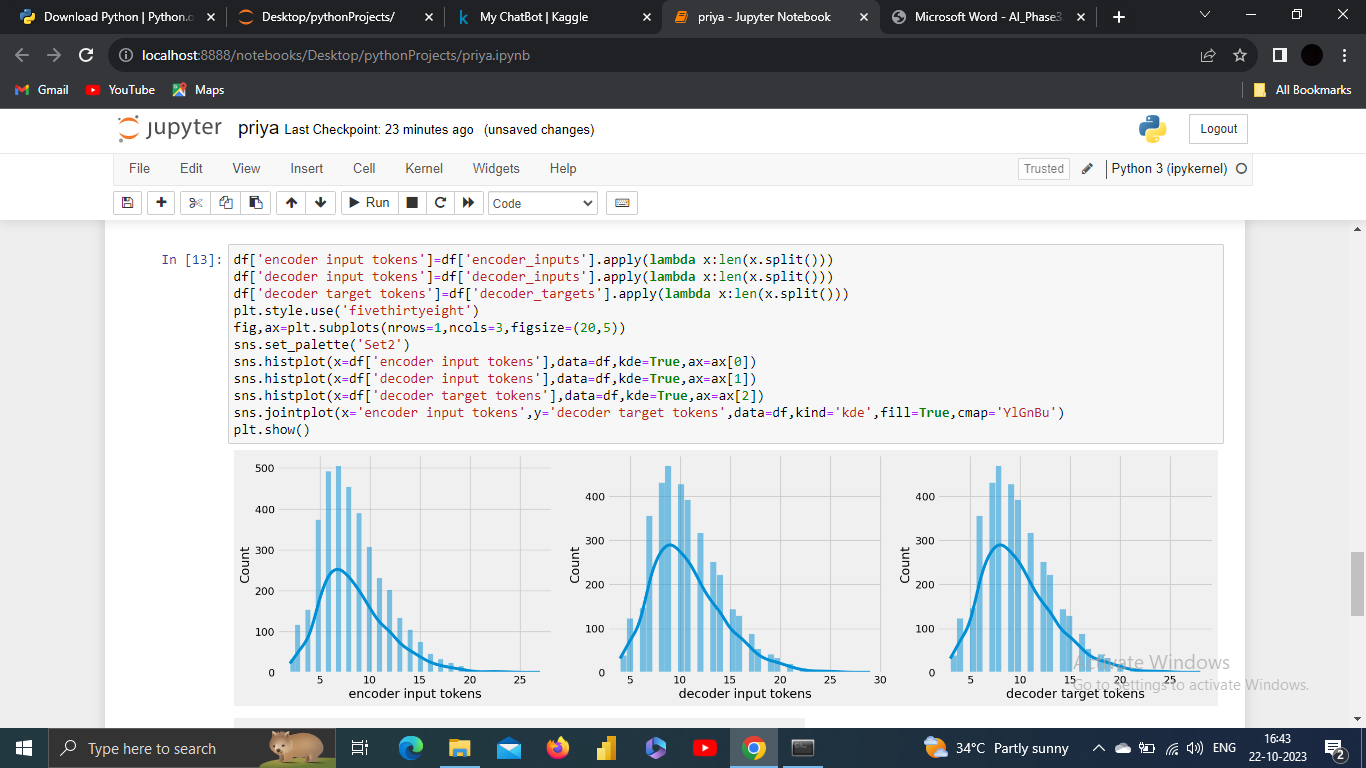
**O/P:**

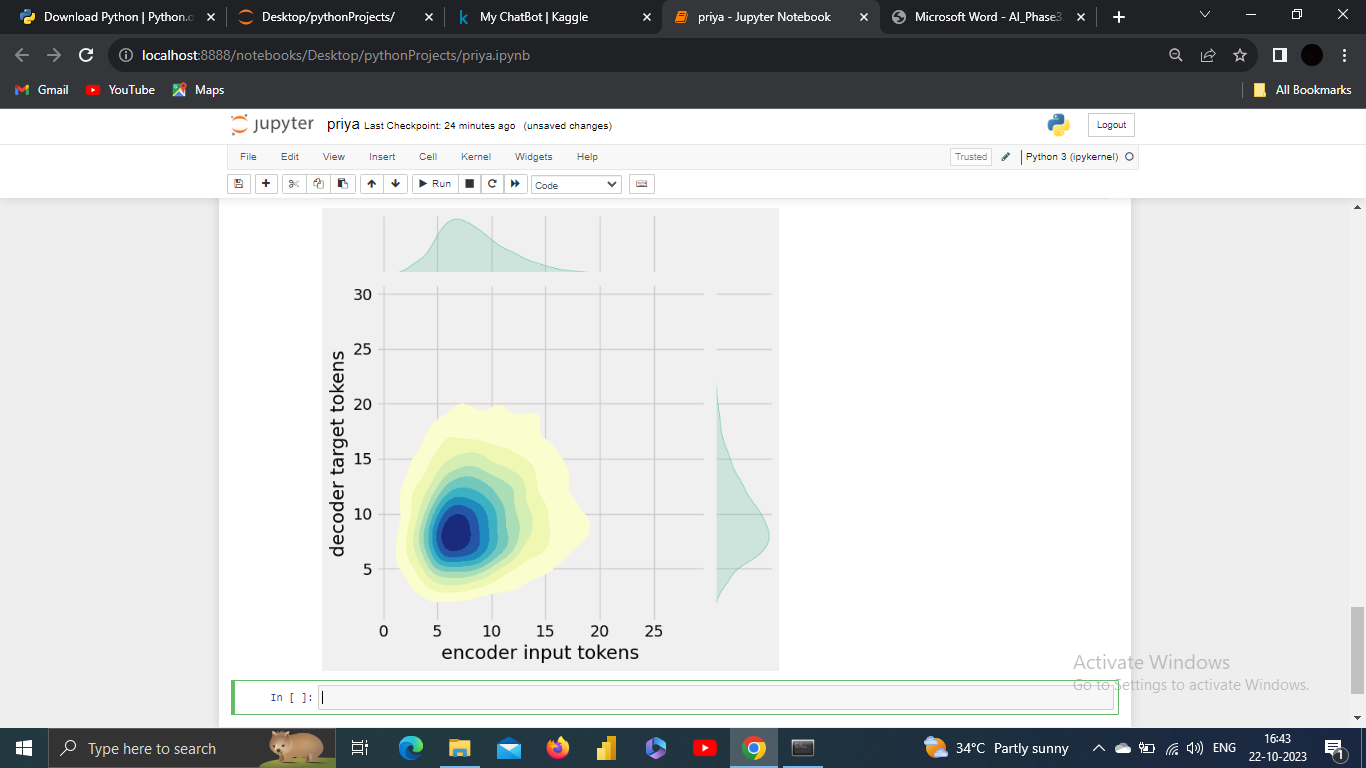


**6.Text** **Cleaning**:

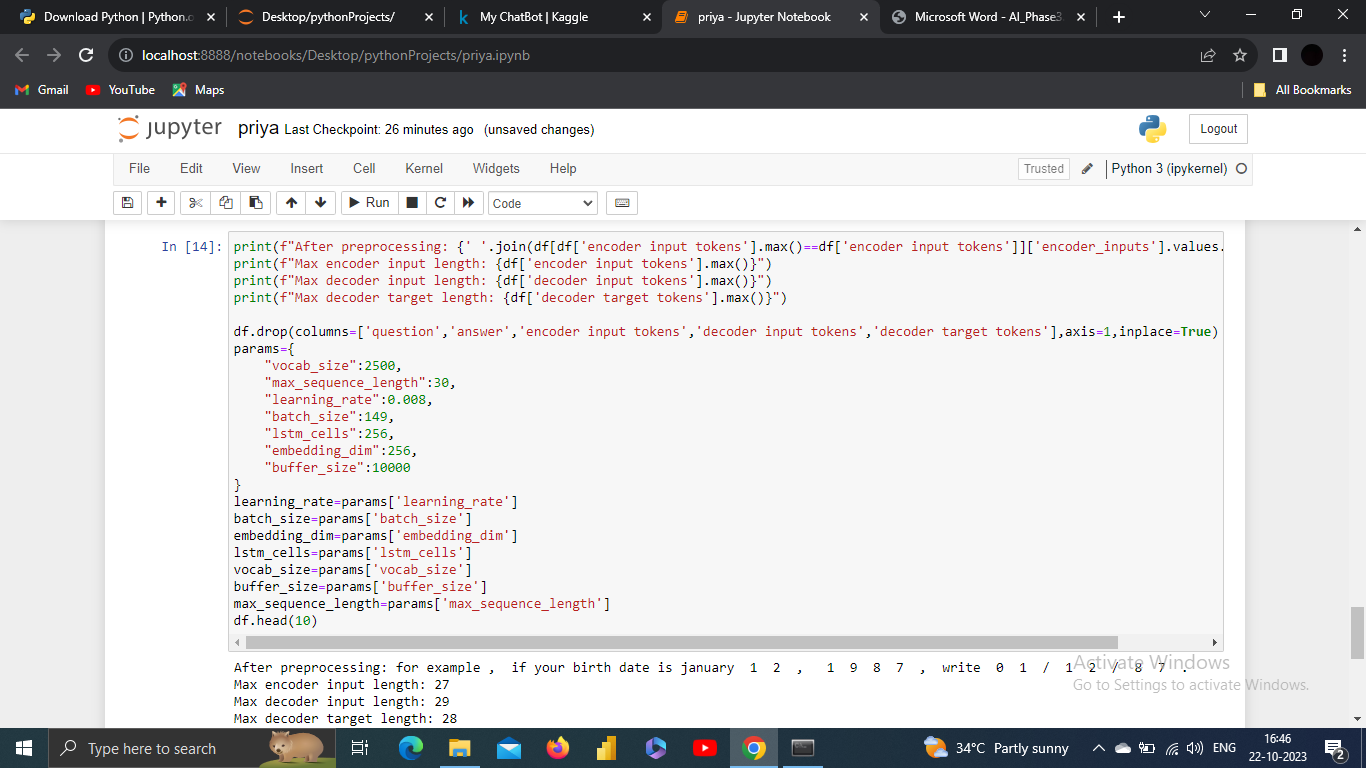
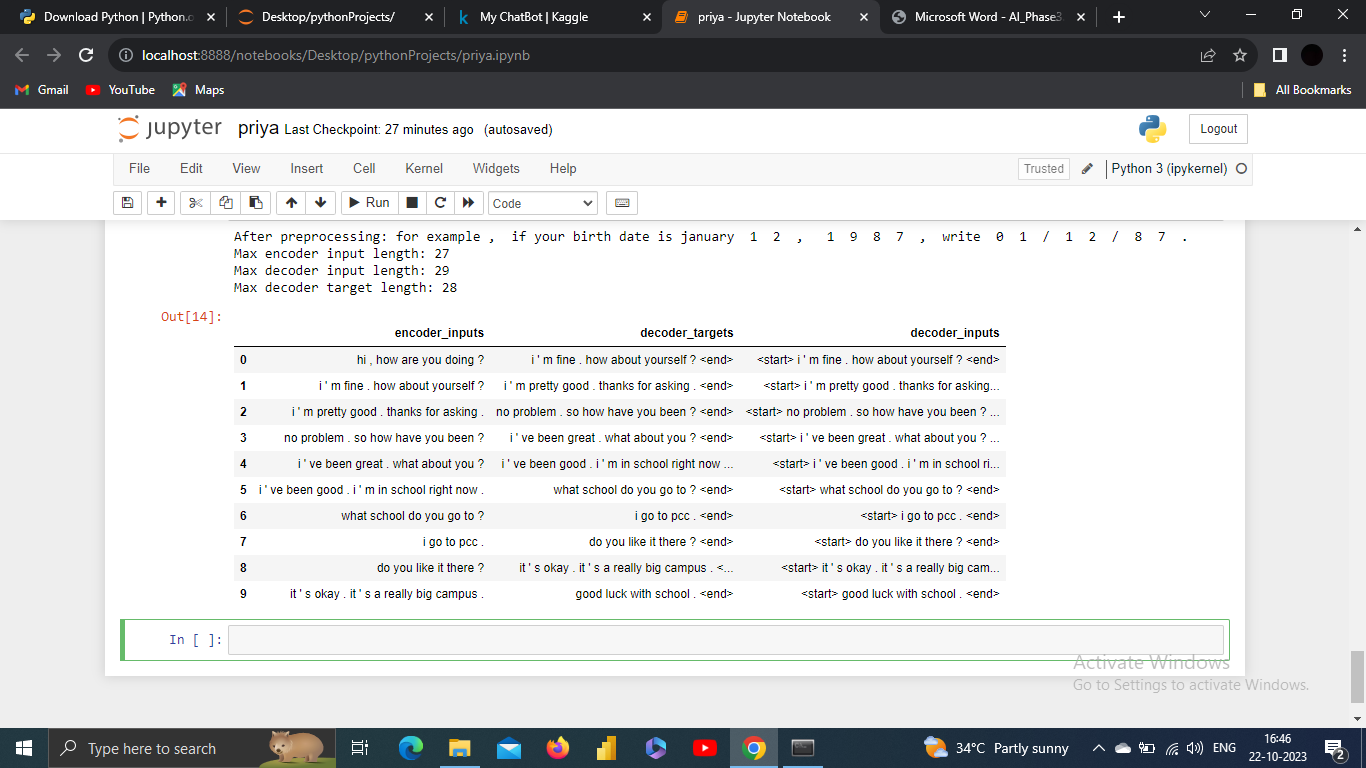




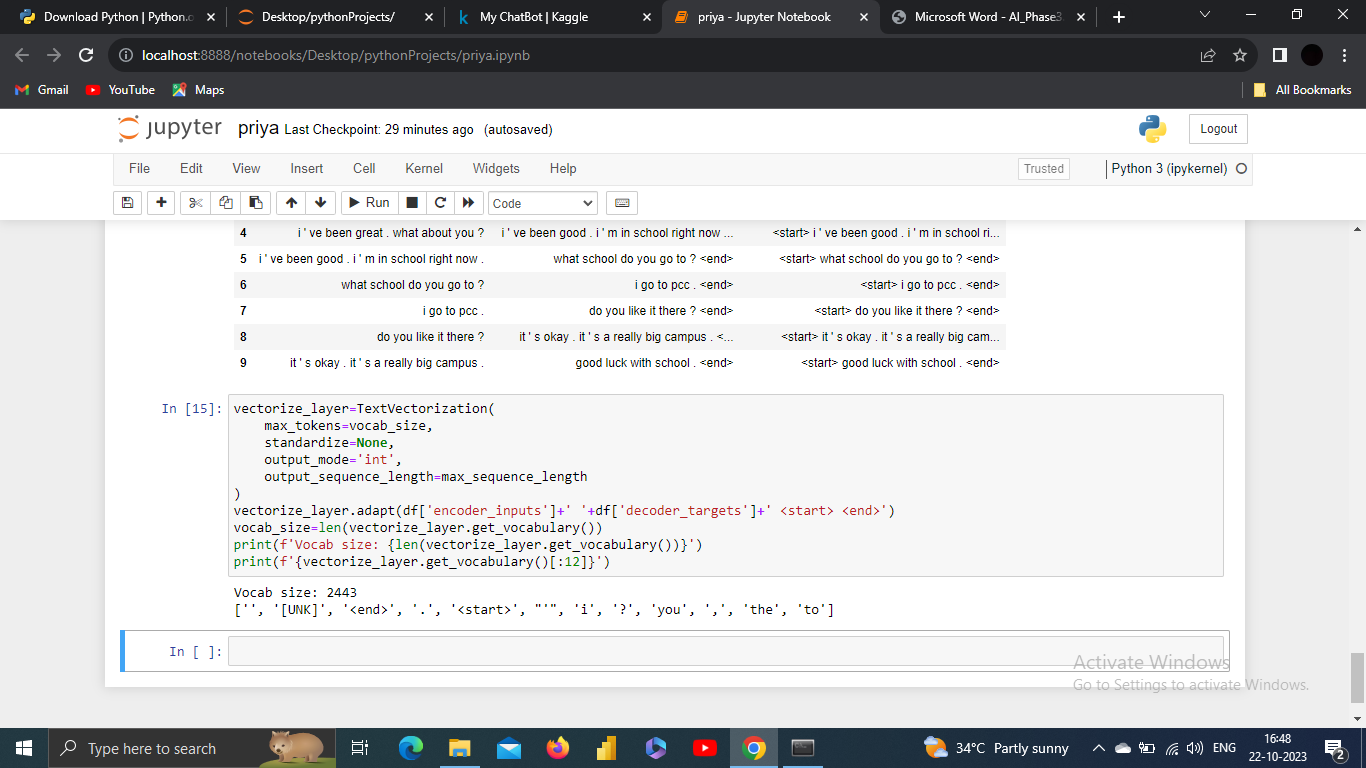


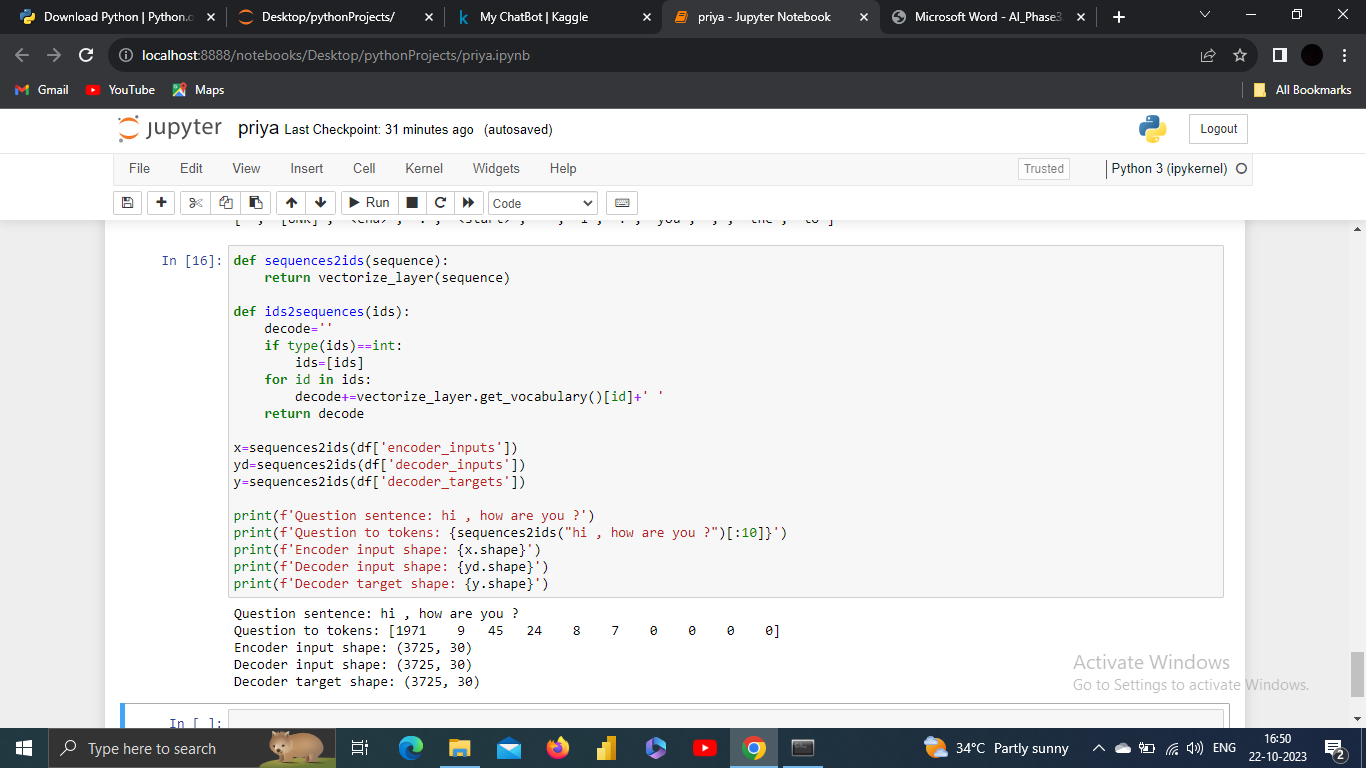


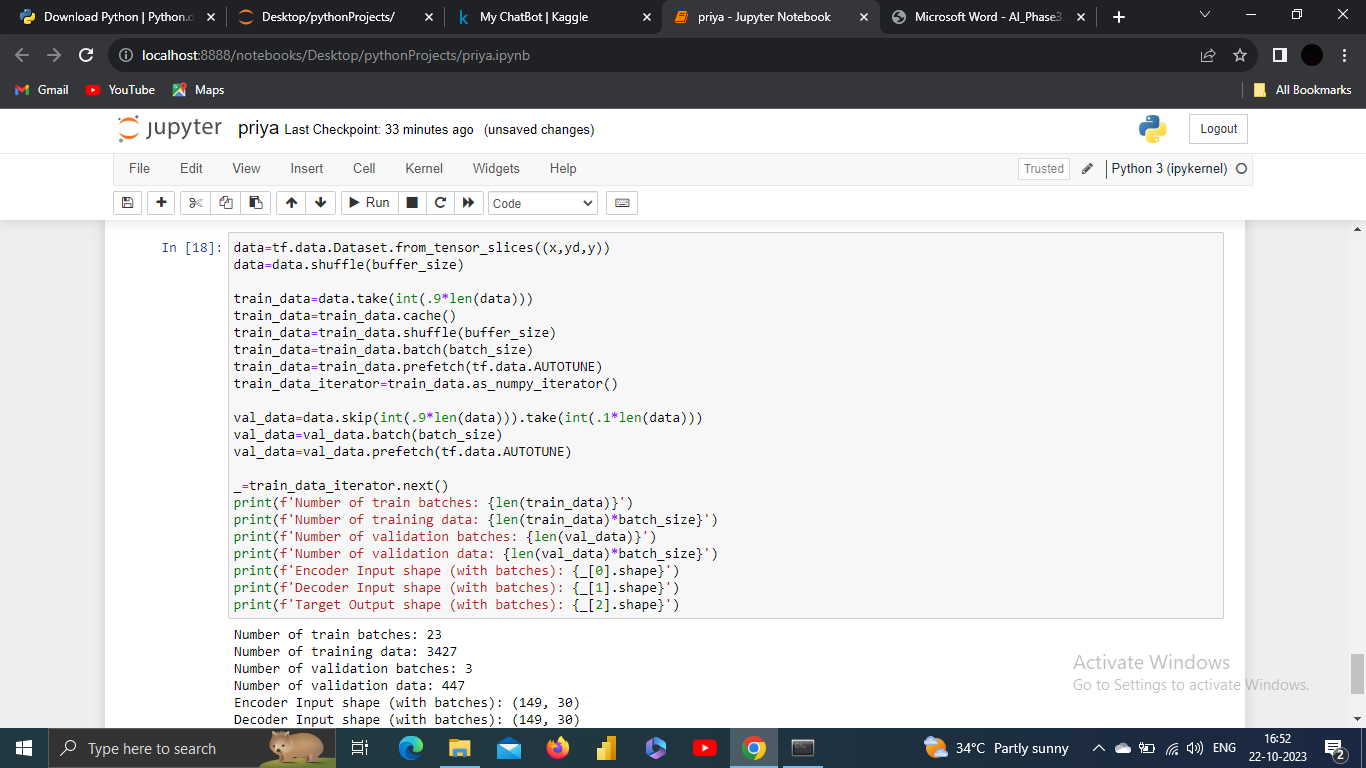
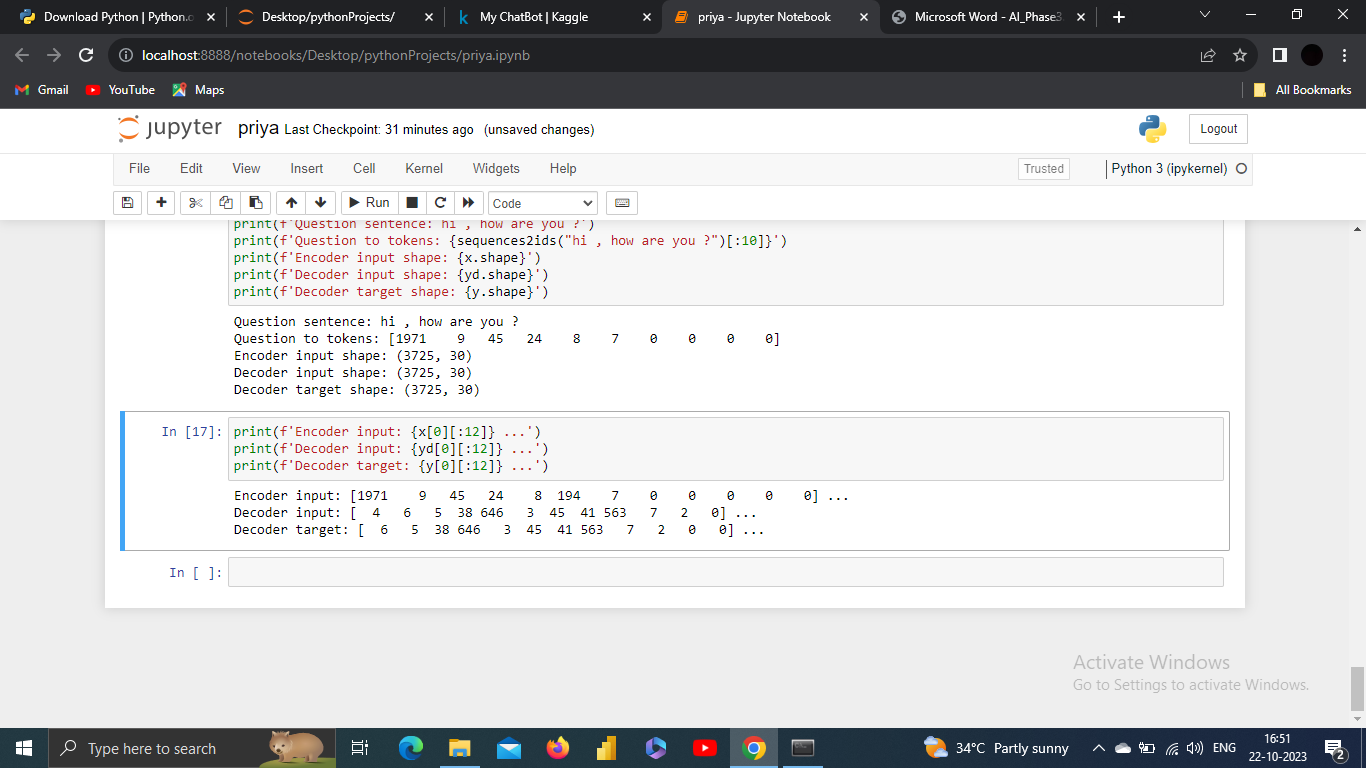
**After preprocessing,**

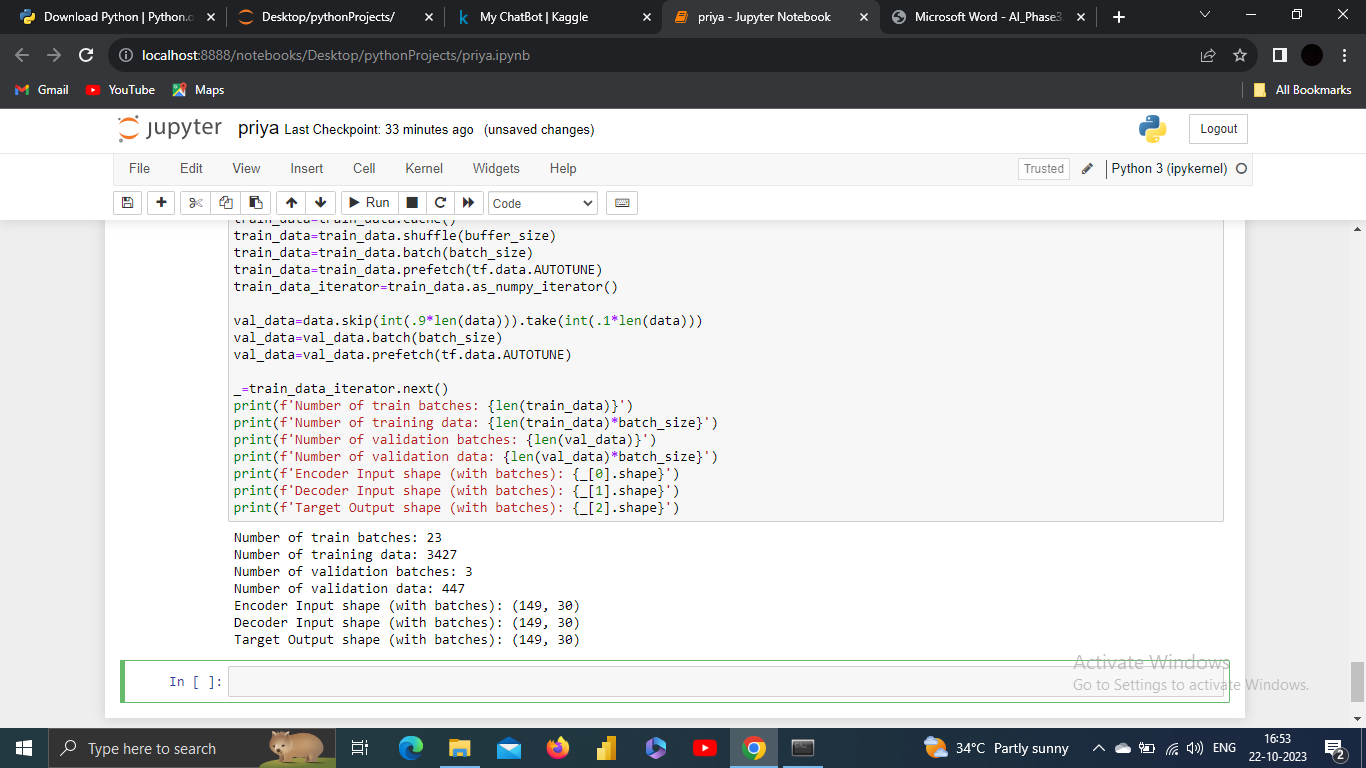
**7. Tokenization**:





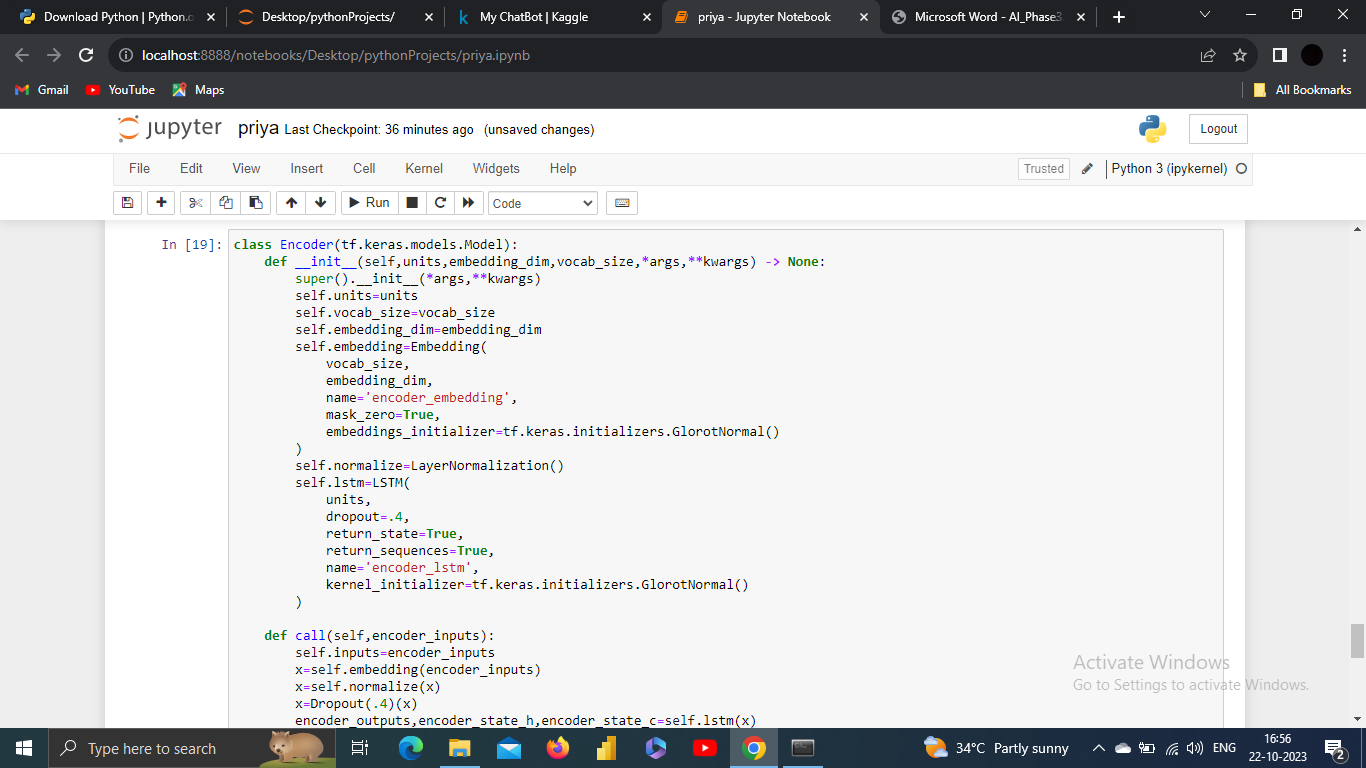


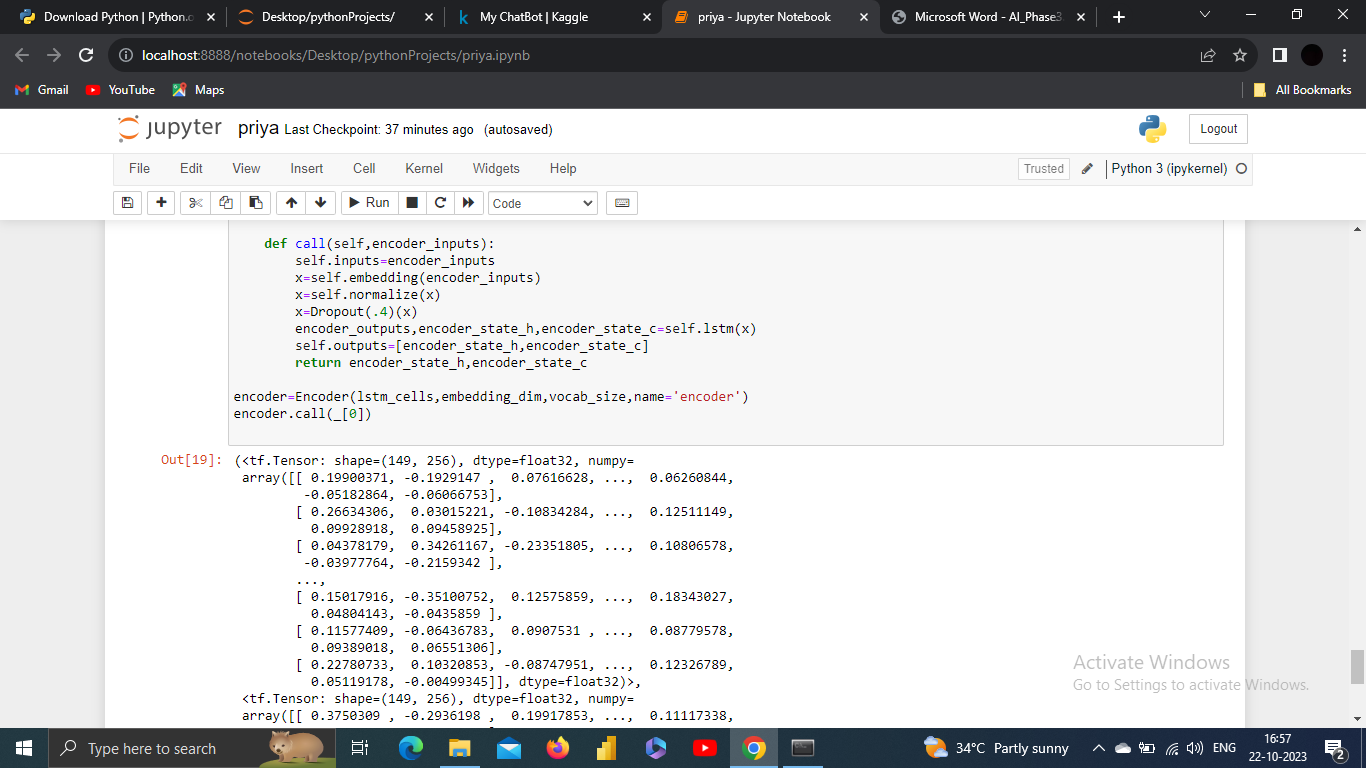
O\P:

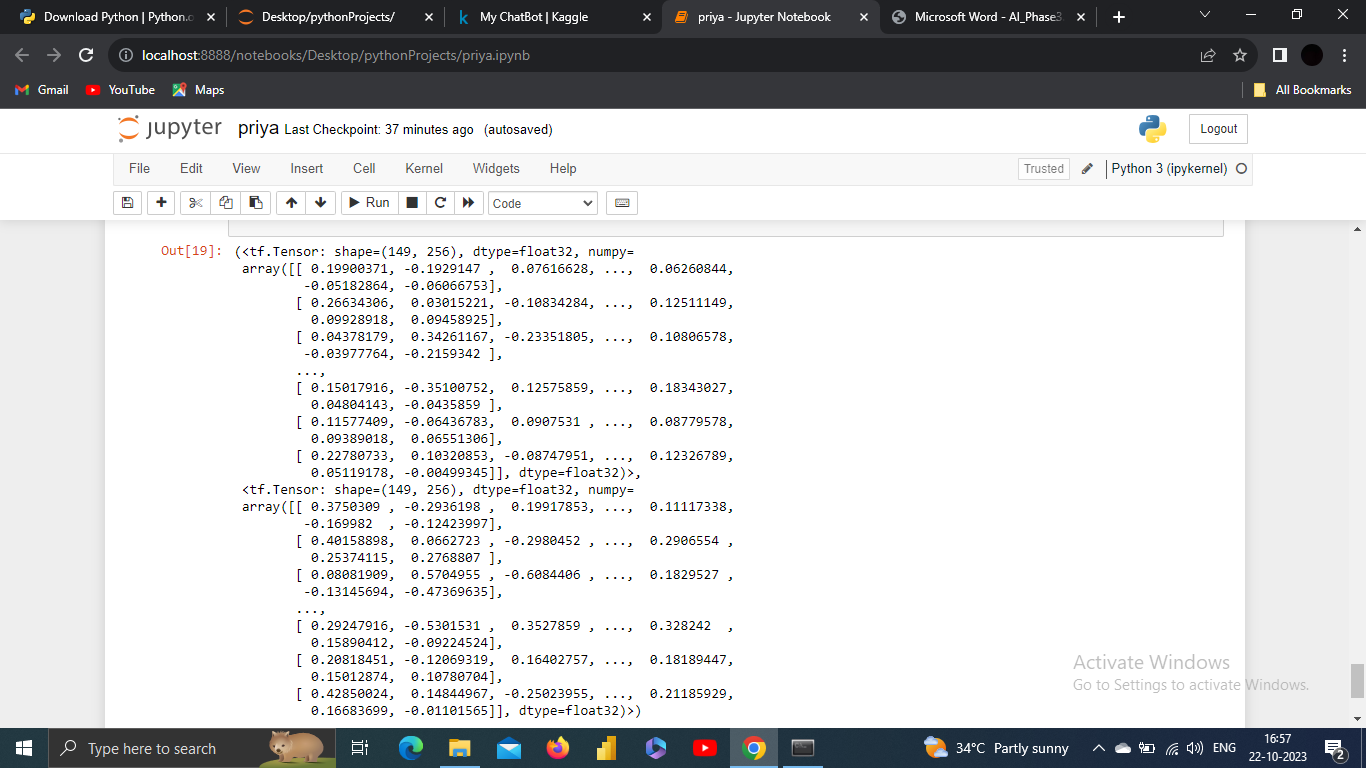


**8. Prototype**:

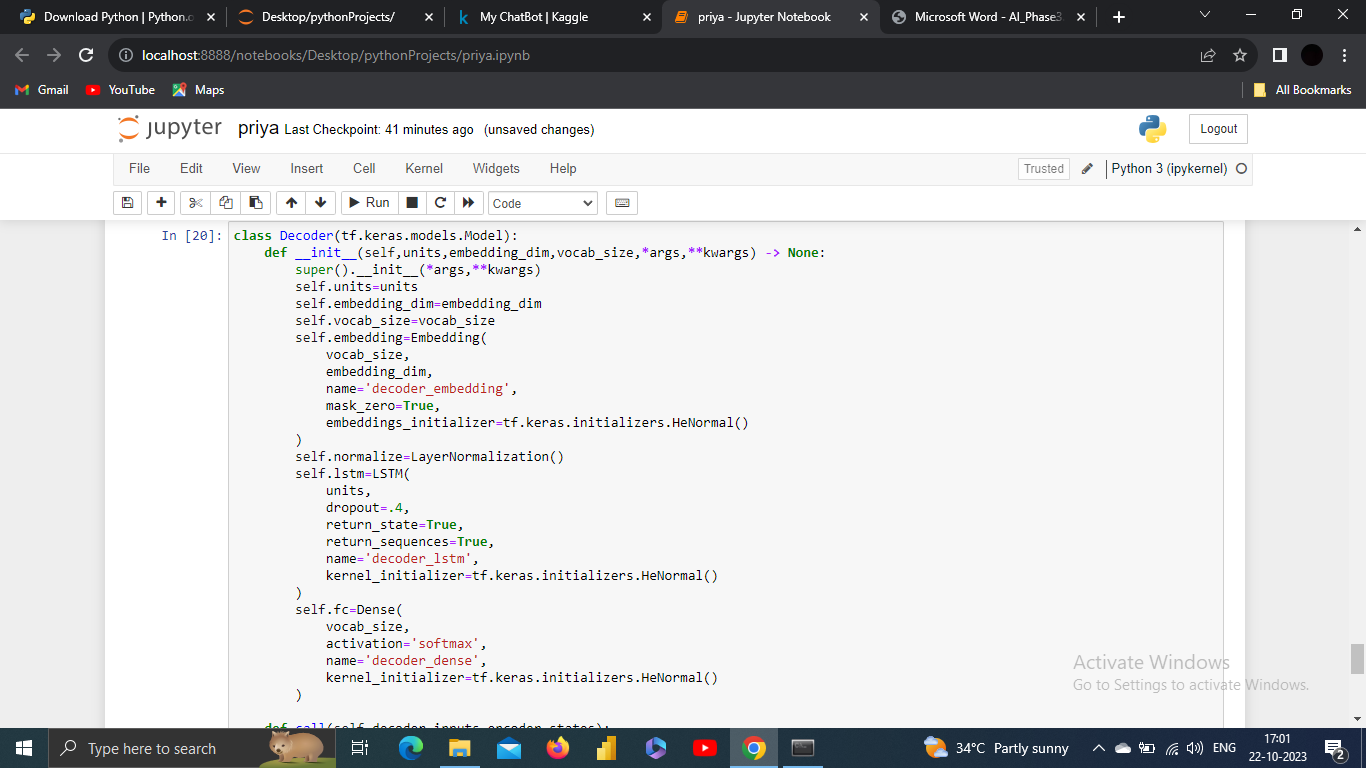
Build encoder,

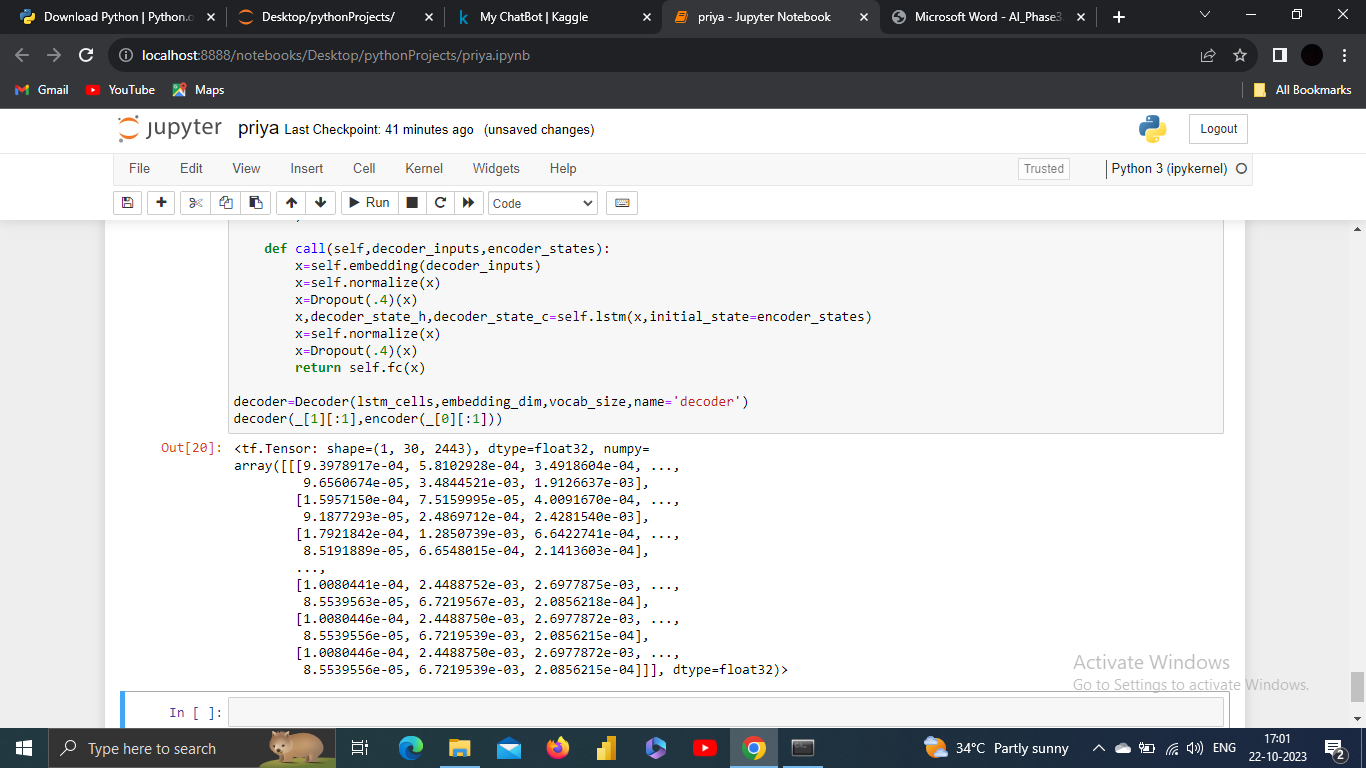






Build Decoder,





Build Training Model,

