CREATE A CHATBOT WITH PYTHON

Team Leader : ROHINEE G (211521104127)

Members :

SHRUTHI P (211521104146)

UMA MEENAKSHI R (211521104173)

VASUMATHI D (211521104175)

YUVASREE G (211521104189)

PHASE 3

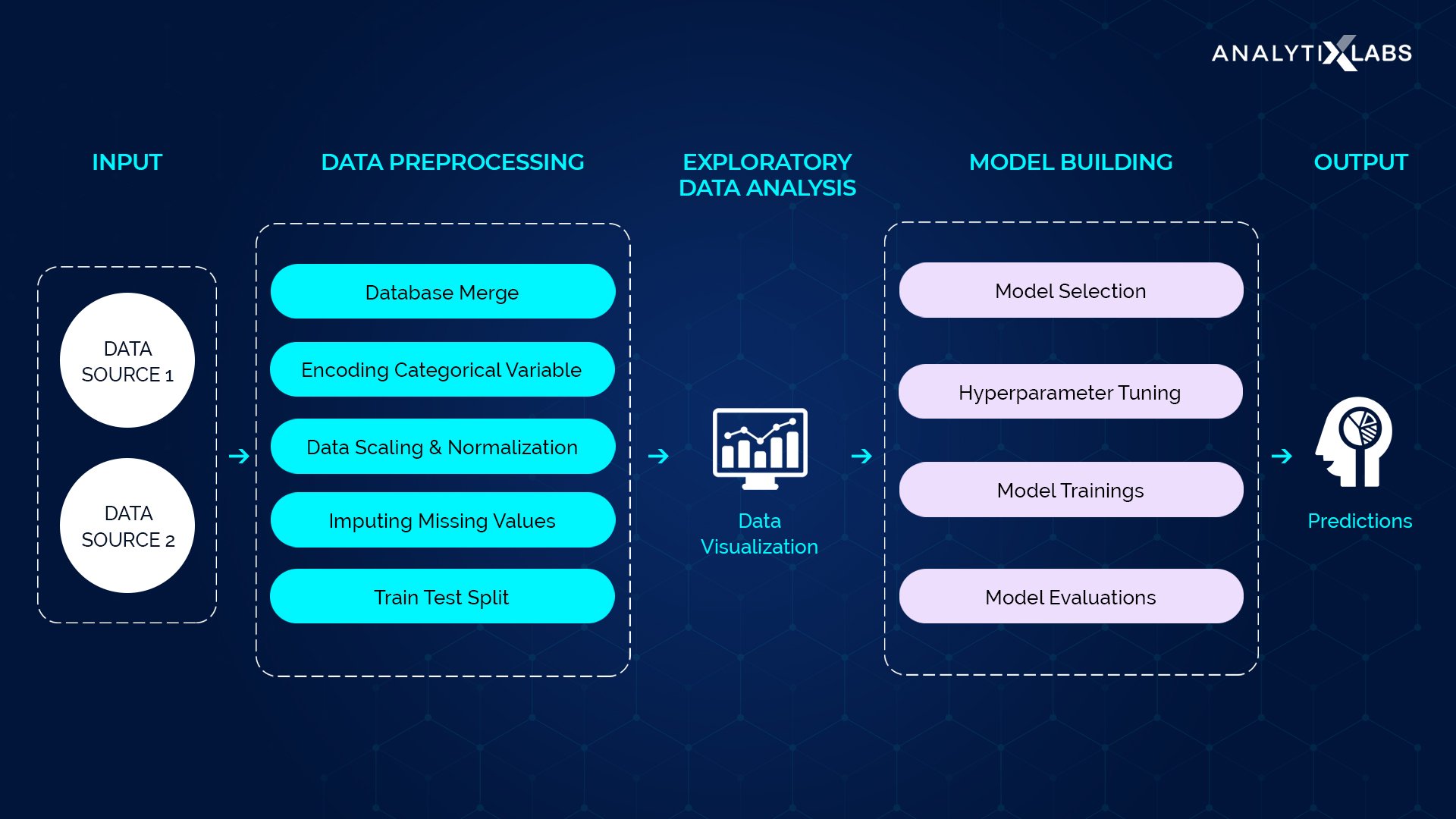
DATA PREPROCESSING

Introduction:

A chatbot is a computer program that simulates and processes human conversations, allowing humans to interact with digital devices as if they are communicating with a real person. Chatbots, like other AI tools, will be used to further enhance human capabilities and makes our world a more efficient and connected place to live and work.

DATA PREPROCESSING

Data Preprocessing is the process of preparing the raw data and making it suitable for the machine learning model. It is the first and the most crucial step in machine learning. The data that we have are not proper, clean and formatted. For the machine learning model to perform efficiently, the data must be cleaned and put in a formatted way before performing any other process.



WHY DO WE NEED DATA PREPROCESSING?

Data is not always pure and clean. Data usually contains noise, missing values or sometimes the data might be unformatted. This cannot be used by the machine learning model as they might reduce the accuracy of the model. Data preprocessing involves the following steps:

* Importing the necessary libraries
* Importing the dataset
* Finding missing data
* Encoding categorical data
* Splitting the dataset into training and testing dataset
* Feature scaling

IMPORTING THE NECESSARY LIBRARIES:

Python has many libraries which can perform specific data preprocessing functions. Importing all these libraries is essential for data preprocessing. Python libraries like numpy, pandas, matplotlib, sklearn are essential libraries required for data preprocessing.

IMPORTING THE DATASET

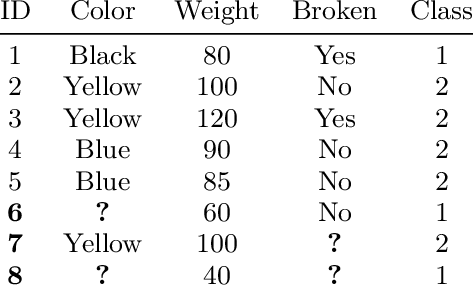
The dataset that should be used in the model must be imported. Importing the dataset is done using the read\_csv() function in Pandas library. This function can read a CSV (Comma Separated File) file. We can also import a HTML file or an Excel file. After importing the dataset, the dependent and independent variables should be extracted and labelled separately. For this, we use the iloc[] function from the Pandas library. This function can extract selected rows and columns from the dataset.

FINDING MISSING VALUES

The data that we use may have missing values which when used in the machine learning model would reduce the accuracy and efficiency of the model. The model might draw faulty conclusions from this dataset. So, to avoid these problems, missing values are identified and corrected during the data preprocessing phase. There are two ways to handle missing values:

* Deleting a particular row - In this method, you remove a specific row that has a null value for a feature or a particular column where more than 75% of the values are missing. However, this method is not 100% efficient, and it is recommended that you use it only when the dataset has adequate samples. You must ensure that after deleting the data, there remains no addition of bias.
* Calculating the mean - This method is useful for features having numeric data like age, salary, year, etc. Here, you can calculate the mean, median, or mode of a particular feature or column or row that contains a missing value and replace the result for the missing value. This method can add variance to the dataset, and any loss of data can be efficiently negated. Hence, it yields better results compared to the first method (omission of rows/columns).

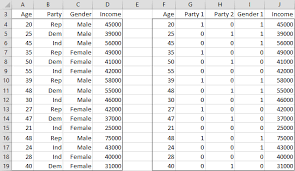
To handle missing values, Scikit-learn library has many libraries used for building machine learning models. Sklearn.preprocessing library has Imputer class which replaces the missing data with mean, median or mode. Median value is always preferred because mean value is influenced by outliers and skewness in the data. There are many types of imputers like SimpleImputer, IterativeImputer, KNNImputer etc.



ENCODING A CATEGORICAL DATA

Categorical data refers to the information that has specific categories within the dataset. Age, gender, educational qualification are examples of categorical data. Categorical data must be encoded because the machine learning model cannot use categorical data in raw form. Categorical data are of two types:

* Ordinal categorical variable: Ordinal categorical variables can be ordered. For example, grades’ can be arranged in order from A to E. For ordinal categorical variable, LabelEncoder is used.
* Nominal categorical variable: Nominal categorical variables can never be ordered. For example, the color of the eye lens can never be arranged in an order. For nominal categorical variable, OneHotEncoder is used.



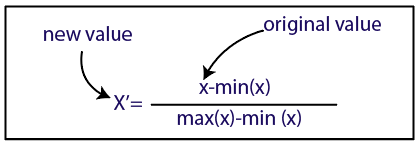
SPLITTING THE DATASET INTO TRAINING AND TEST SET

After data preparation, the dataset is divided into two parts – training set, and the test set. The training data is used to build the model. It identifies the patterns inside the dataset and generates model parameters. The training helps to identify the overfitting and underfitting in the data. The test set is used to predict the output by calculating the accuracy. Usually the split is in the ratio 80:20 or 70:30. To implement training and test set splitting, we have to import train\_test\_split function from sklearn.model\_selection library. Thus, the train\_test\_split() function includes four parameters, the first two of which are for arrays of data. The test\_size function specifies the size of the test set. The test\_size maybe .5, .3, or .2 – this specifies the dividing ratio between the training and test sets. The last parameter, “random\_state” sets seed for a random generator so that the output is always the same.

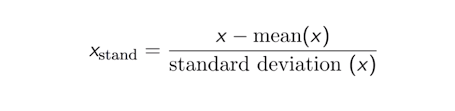
FEATURE SCALING

Feature Scaling is the process of standardizing the independent variables of a dataset within a specific range. Feature scaling limits the range of variables so that you can compare them. There are two ways of feature scaling:

* Normalisation: Normalization is a scaling technique in which values are shifted and rescaled so that they end up ranging between 0 and 1. It is also known as Min-Max scaling.



* Standardisation: Standardization is another scaling method where the values are centered around the mean with a unit standard deviation. This means that the mean of the attribute becomes zero, and the resultant distribution has a unit standard deviation.



To implement feature scaling, we import StandardScaler class from sklearn.preprocessing library.

TEXT PREPROCESSING

In Natural Language Processing, text preprocessing is the practice of cleaning and preparing text data. NLTK and re are the usual python libraries used for text preprocessing.

STEPS INVOLVED IN TEXT PREPROCESSING

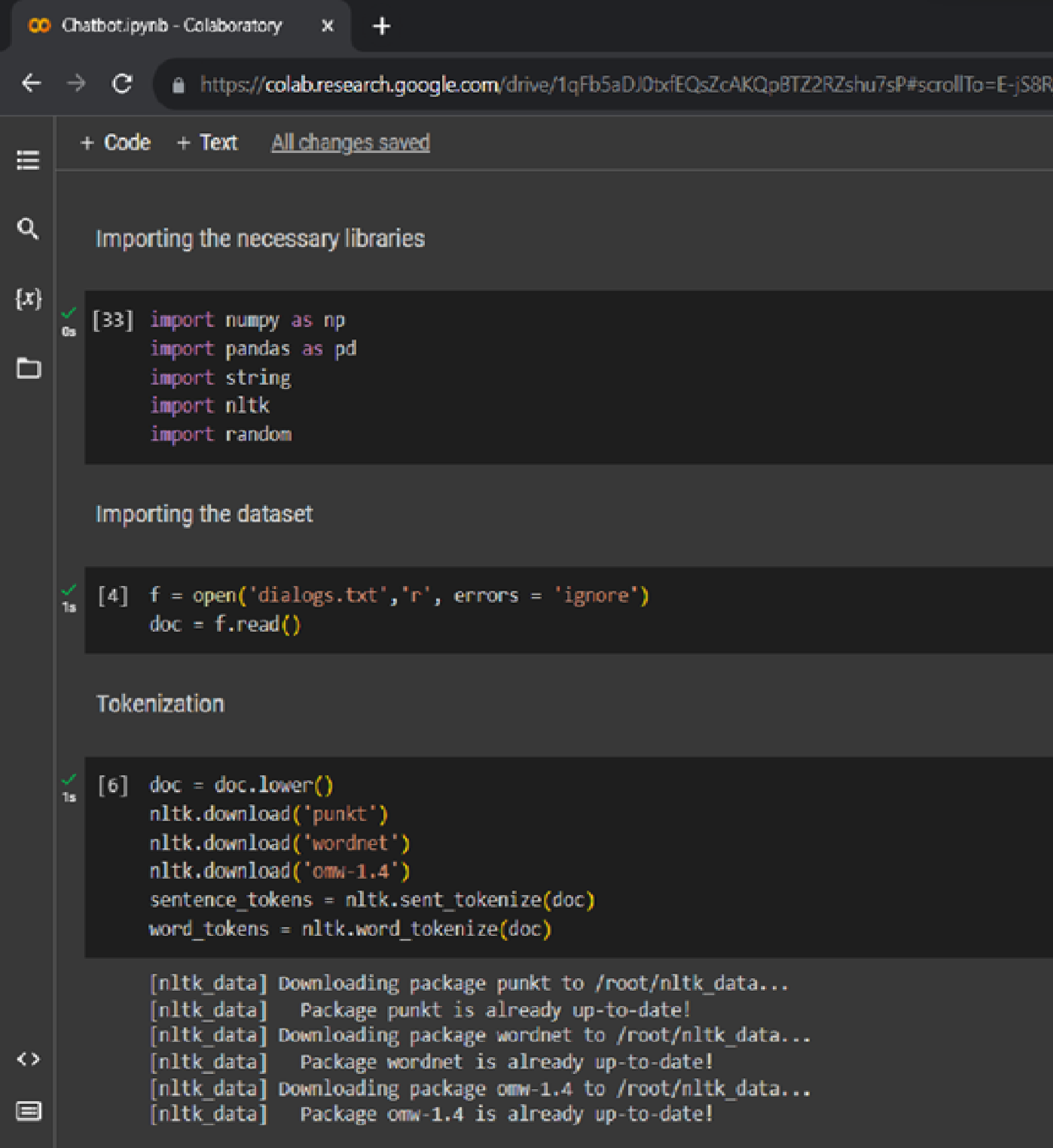
* Noise removal: It is a text preprocessing task that removes characters, digits and pieces of text that can interfere with the text analysis.
* Tokenisation: It is the text preprocessing task of breaking up text into smaller components of text (known as tokens).
* Text normalisation: Normalization encompasses many text preprocessing tasks including stemming, lemmatization, upper or lowercasing, and stopwords removal.
* Stemming: It is the text preprocessing normalization task concerned with bluntly removing word affixes (prefixes and suffixes).
* Lemmatization: It is the text preprocessing normalization task concerned with bringing words down to their root forms.
* Stopword removal: It is the process of removing words from a string that don’t provide any information about the tone of a statement.
* Part-of-speech tagging: It is the process of assigning a part of speech to every word in a string. Using the part of speech can improve the results of lemmatization.

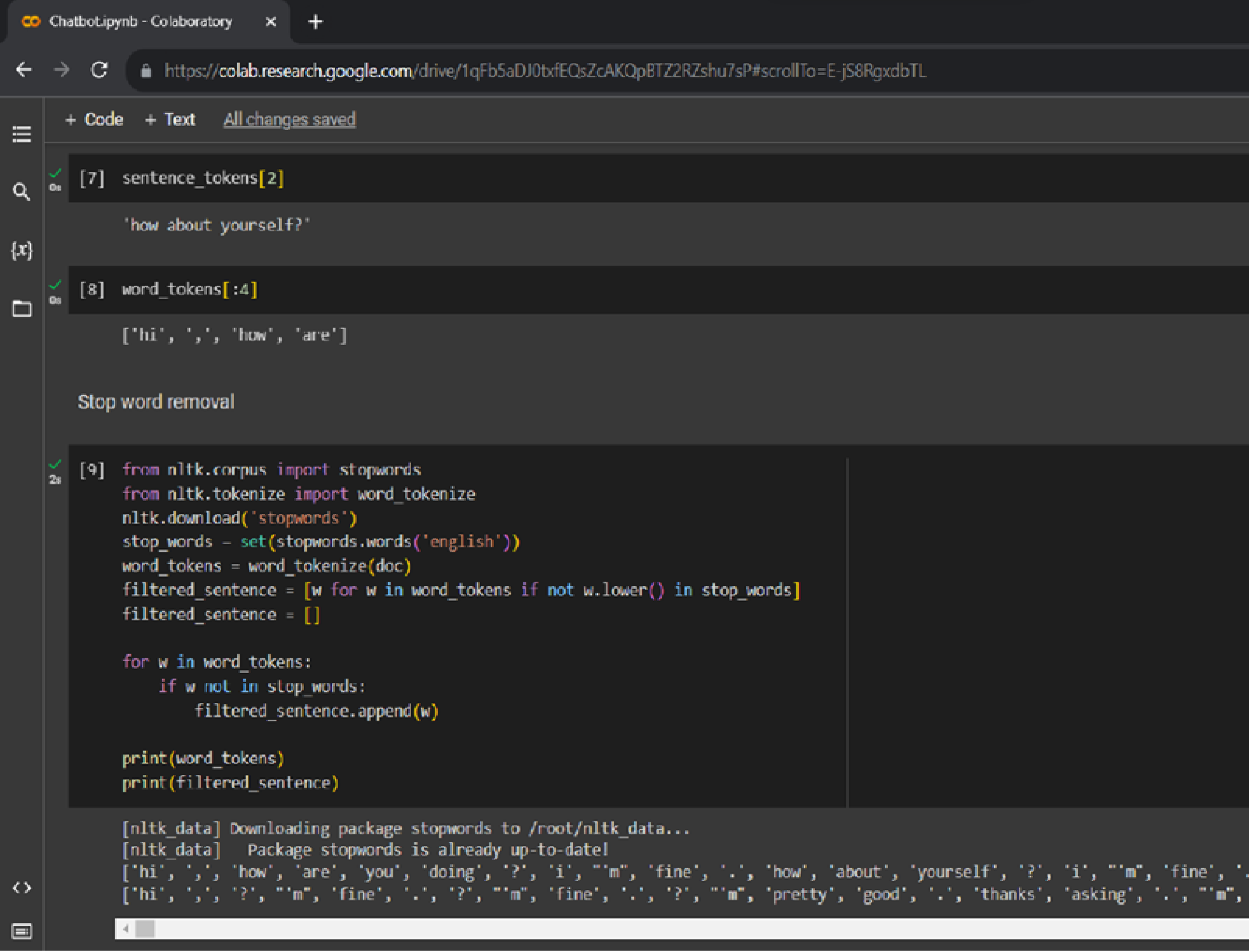
CODE IMPLEMENTATION

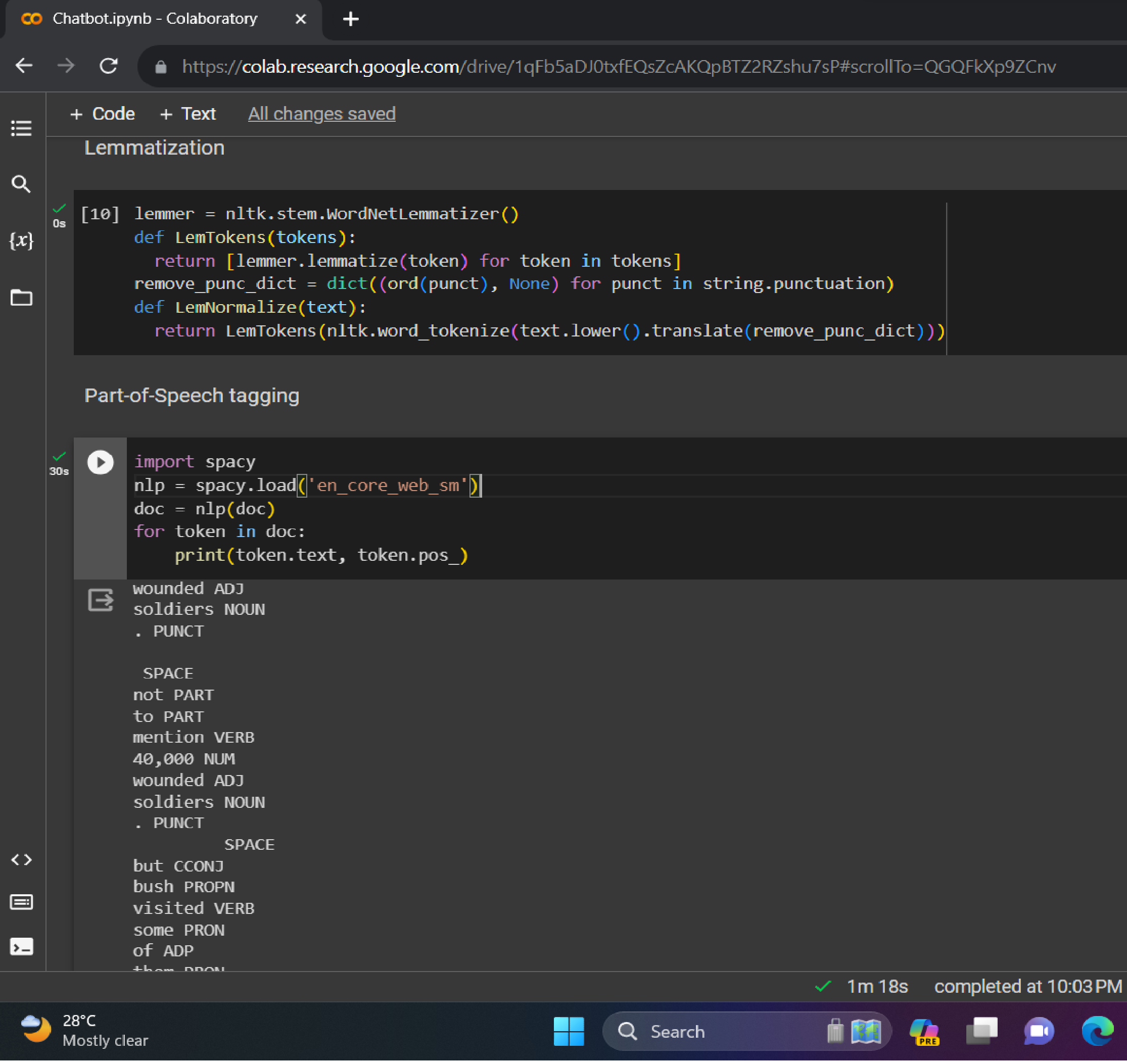
DATASET: <https://www.kaggle.com/datasets/grafstor/simple-dialogs-for-chatbot>

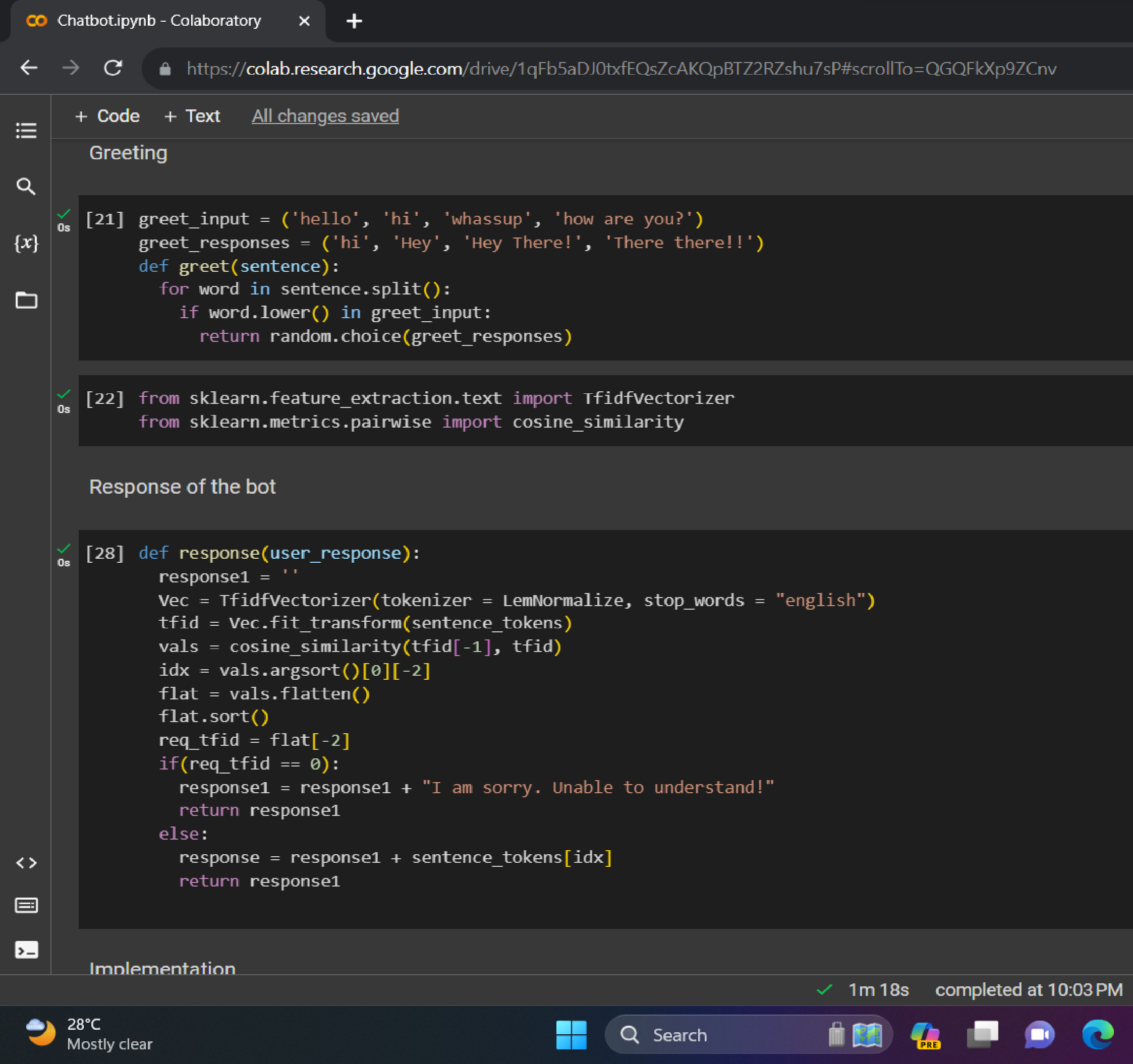
WORKING ENVIRONMENT: Google colaboratory notebook

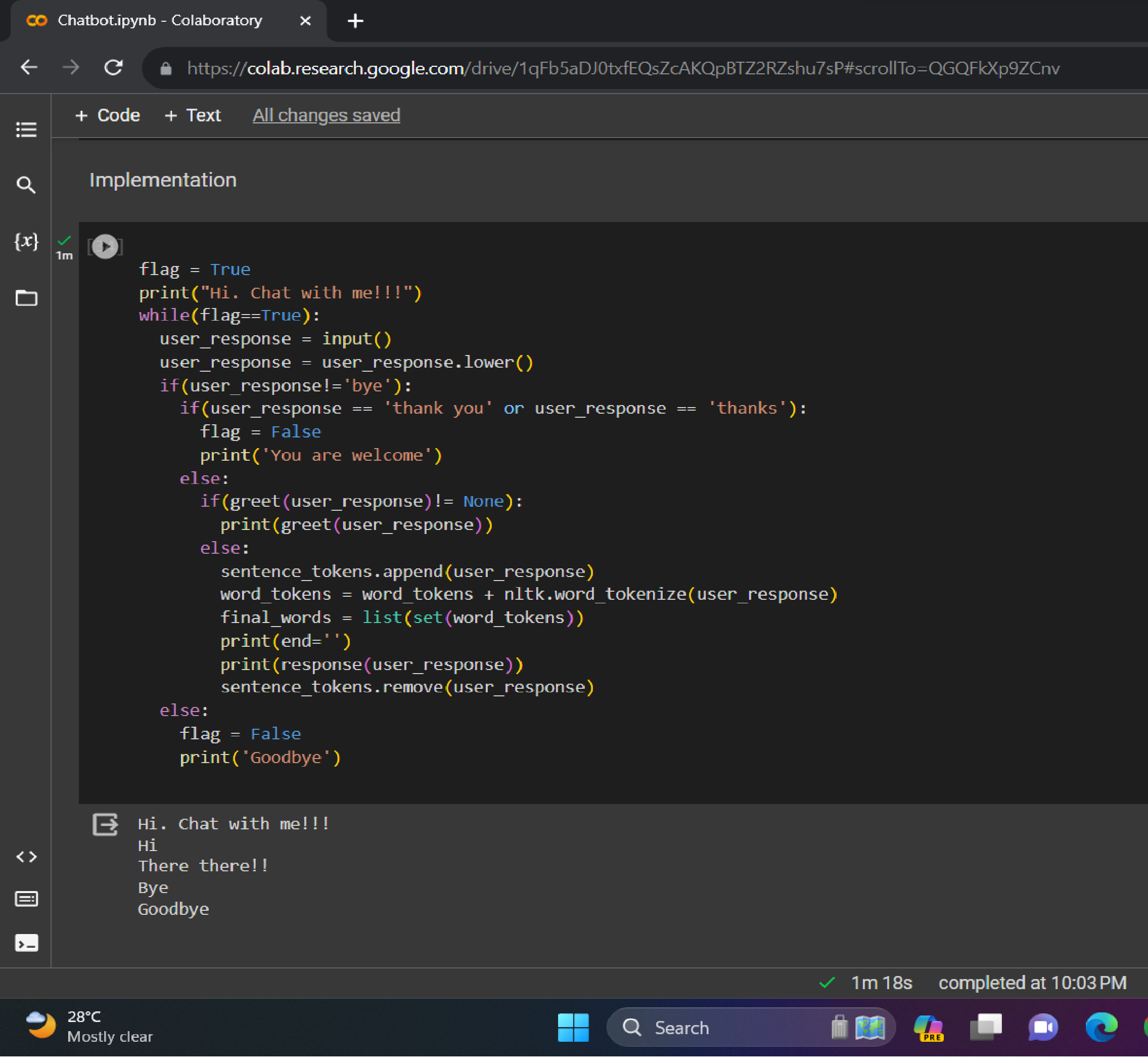
CODE:











CONCLUSION

Thus, the basic implementation of the chatbot is done using the necessary python libraries. The dataset has been loaded and data preprocessing has been performed on the dataset.