Tables Of Contents

|  |  |  |
| --- | --- | --- |
| Sr No. | Title | Page No |
| 1. | Introduction to python and NLP | 6 |
| 2. | Why this topic ? | 9 |
| 3. | Begin with Analysis | 10 |
| 4. | Conclusion | 23 |
| 5. | Bibliography | 25 |

INTRODUCTION

***Introduction of python:***

Python is a widely used general-purpose, high level programming language. It was created by Guido van Rossum in 1991 and further developed by the Python Software Foundation. It was designed with an emphasis on code readability, and its syntax allows programmers to express their concepts in fewer lines of code.

Python is a programming language that lets you work quickly and integrate systems more efficiently.

Python is an interpreted, object-oriented, high-level programming language with dynamic semantics. Its high-level built in data structures, combined with dynamic typing and dynamic binding, make it very attractive for Rapid Application Development, as well as for use as a scripting or glue language to connect existing components together. Python's simple, easy to learn syntax emphasizes readability and therefore reduces the cost of program maintenance. Python supports modules and packages, which encourages program modularity and code reuse. The Python interpreter and the extensive standard library are available in source or binary form without charge for all major platforms, and can be freely distributed.

Often, programmers fall in love with Python because of the increased productivity it provides. Since there is no compilation step, the edit-test-debug cycle is incredibly fast. Debugging Python programs is easy: a bug or bad input will never cause a segmentation fault. Instead, when the interpreter discovers an error, it raises an exception. When the program doesn't catch the exception, the interpreter prints a stack trace. A source level debugger allows inspection of local and global variables, evaluation of arbitrary expressions, setting breakpoints, stepping through the code a line at a time, and so on. The debugger is written in Python itself, testifying to Python's introspective power. On the other hand, often the quickest way to debug a program is to add a few print statements to the source: the fast edit-test-debug cycle makes this simple approach very effective

***Introduction of NLP:***

Natural language processing (NLP) refers to the branch of computer science—and more specifically, the branch of [artificial intelligence or AI](https://www.ibm.com/topics/artificial-intelligence)—concerned with giving computers the ability to understand text and spoken words in much the same way human beings can.

NLP combines computational linguistics—rule-based modeling of human language—with statistical, machine learning, and deep learning models. Together, these technologies enable computers to process human language in the form of text or voice data and to ‘understand’ its full meaning, complete with the speaker or writer’s intent and sentiment.

NLP drives computer programs that translate text from one language to another, respond to spoken commands, and summarize large volumes of text rapidly—even in real time. There’s a good chance you’ve interacted with NLP in the form of voice-operated GPS systems, digital assistants, speech-to-text dictation software, customer service chatbots, and other consumer conveniences. But NLP also plays a growing role in enterprise solutions that help streamline business operations, increase employee productivity, and simplify mission-critical business processes.

NLP use cases

**NLP USES CASES:**

Natural language processing is the driving force behind machine intelligence in many modern real-world applications. Here are a few examples:

* **Spam detection:**You may not think of spam detection as an NLP solution, but the best spam detection technologies use NLP's text classification capabilities to scan emails for language that often indicates spam or phishing. These indicators can include overuse of financial terms, characteristic bad grammar, threatening language, inappropriate urgency, misspelled company names, and more. Spam detection is one of a handful of NLP problems that experts consider 'mostly solved' (although you may argue that this doesn’t match your email experience).
* **Machine translation:**Google Translate is an example of widely available NLP technology at work. Truly useful machine translation involves more than replacing words in one language with words of another.  Effective translation has to capture accurately the meaning and tone of the input language and translate it to text with the same meaning and desired impact in the output language. Machine translation tools are making good progress in terms of accuracy. A great way to test any machine translation tool is to translate text to one language and then back to the original. An oft-cited classic example: Not long ago, translating “*The spirit is willing but the flesh is weak”* from English to Russian and back yielded “*The vodka is good but the meat is rotten*.” Today, the result is “*The spirit desires, but the flesh is weak*,” which isn’t perfect, but inspires much more confidence in the English-to-Russian translation.
* **Virtual agents and chatbots:** [Virtual agents](https://www.ibm.com/products/watson-assistant) such as Apple's Siri and Amazon's Alexa use speech recognition to recognize patterns in voice commands and natural language generation to respond with appropriate action or helpful comments. [Chatbots](https://www.ibm.com/topics/chatbots) perform the same magic in response to typed text entries. The best of these also learn to recognize contextual clues about human requests and use them to provide even better responses or options over time. The next enhancement for these applications is question answering, the ability to respond to our questions—anticipated or not—with relevant and helpful answers in their own words.
* **Social media sentiment analysis:**NLP has become an essential business tool for uncovering hidden data insights from social media channels. Sentiment analysis can analyze language used in social media posts, responses, reviews, and more to extract attitudes and emotions in response to products, promotions, and events–information companies can use in product designs, advertising campaigns, and more.
* **Text summarization:**Text summarization uses NLP techniques to digest huge volumes of digital text and create summaries and synopses for indexes, research databases, or busy readers who don't have time to read full text. The best text summarization applications use semantic reasoning and natural language generation (NLG) to add useful context and conclusions to summaries.

Why I chose to text analysis of Mahabharat ?

It narrates the struggle between two groups of cousins in the Kurukshetra War and the fates of the Kaurava and the Pāṇḍava princes and their successors.

It also contains philosophical and devotional material, such as a discussion of the four "goals of life" or puruṣārtha. Among the principal works and stories in the Mahābhārata are the Bhagavad Gita, the story of Damayanti, the story of Shakuntala, the story of Pururava and Urvashi, the story of Savitri and Satyavan, the story of Kacha and Devayani, the story of Rishyasringa and an abbreviated version of the Rāmāyaṇa, often considered as works in their own right.

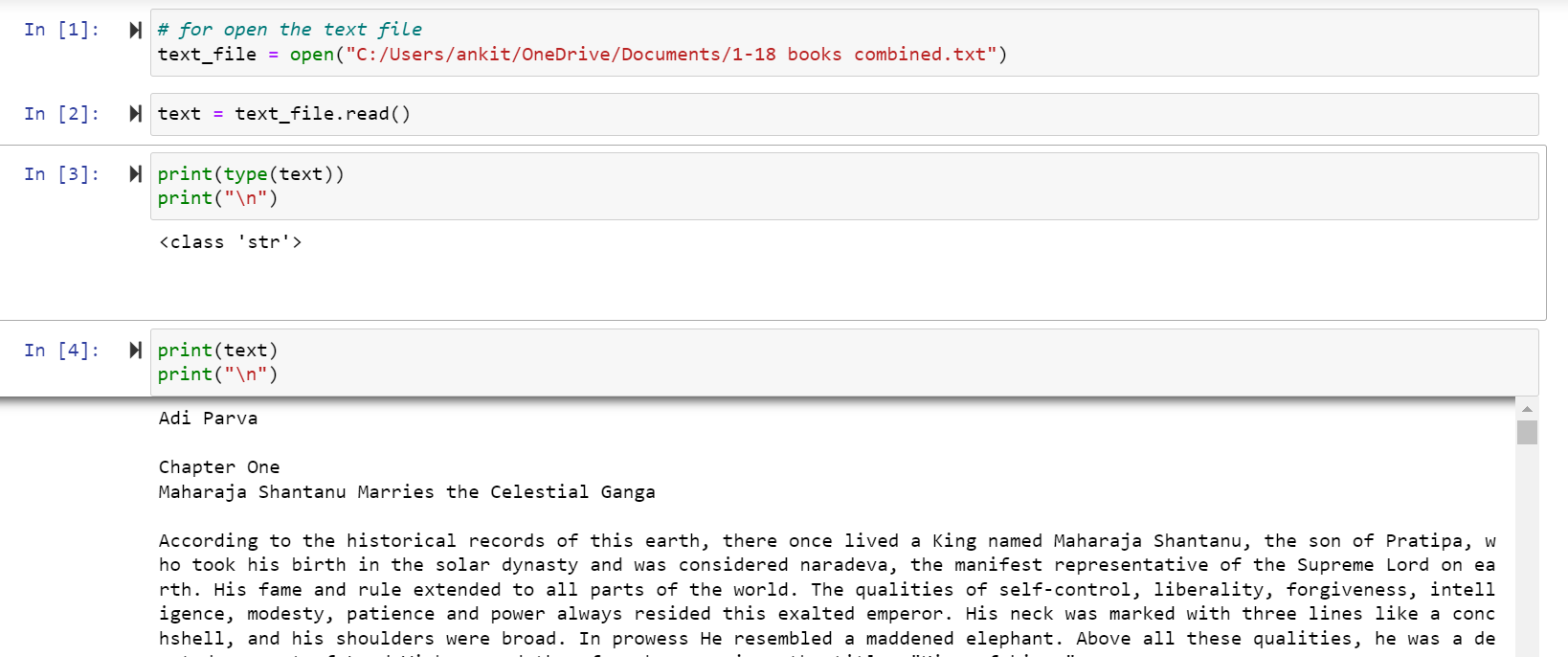
Krishna and Arjuna at Kurukshetra, 18th–19th-century painting. Traditionally, the authorship of the Mahābhārata is attributed to Vyāsa. There have been many attempts to unravel its historical growth and compositional layers. The bulk of the Mahābhārata was probably compiled between the 3rd century BCE and the 3rd century CE, with the oldest preserved parts not much older than around 400 BCE. The text probably reached its final form by the early Gupta period (c. 4th century CE).

The Mahābhārata is the longest epic poem known and has been described as "the longest poem ever written".

Begin with the Project

STEP 1 : Installing the nltk and Downloading the data

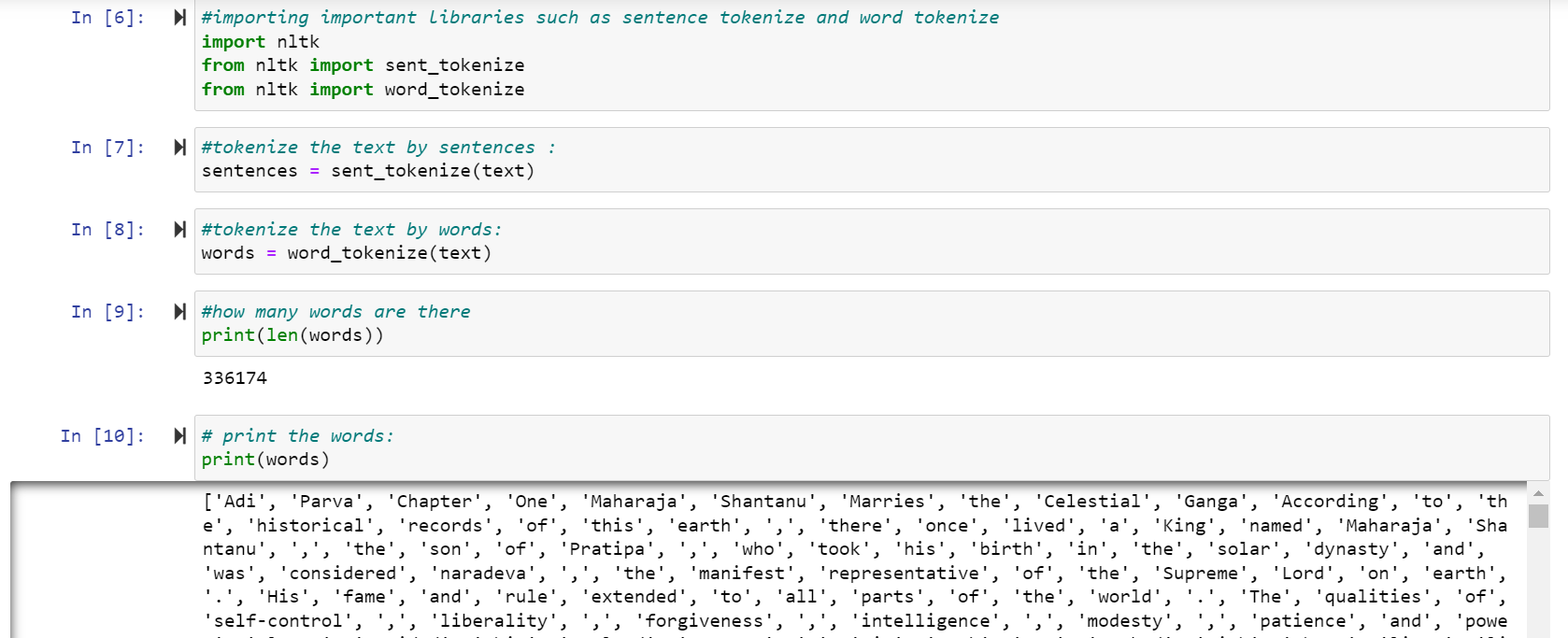
We will use the NLTK package in Python. In this step we will install NLTK and open and jread the text text file

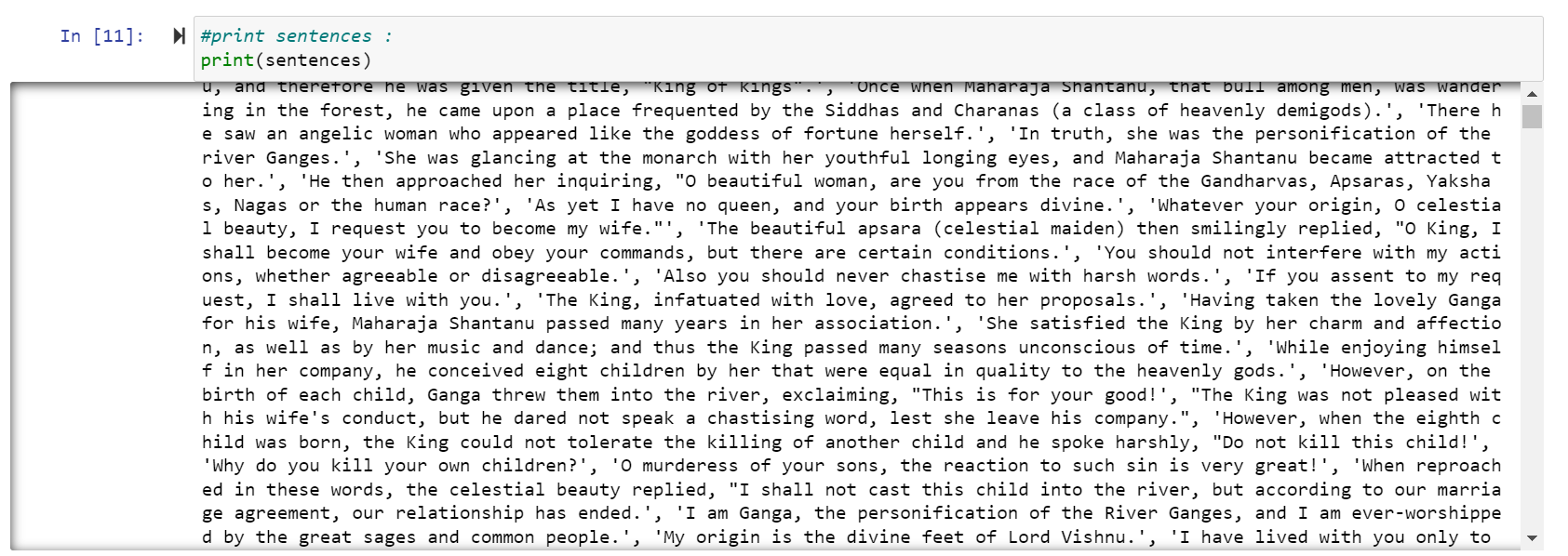


STEP 2 : Tokenize the data

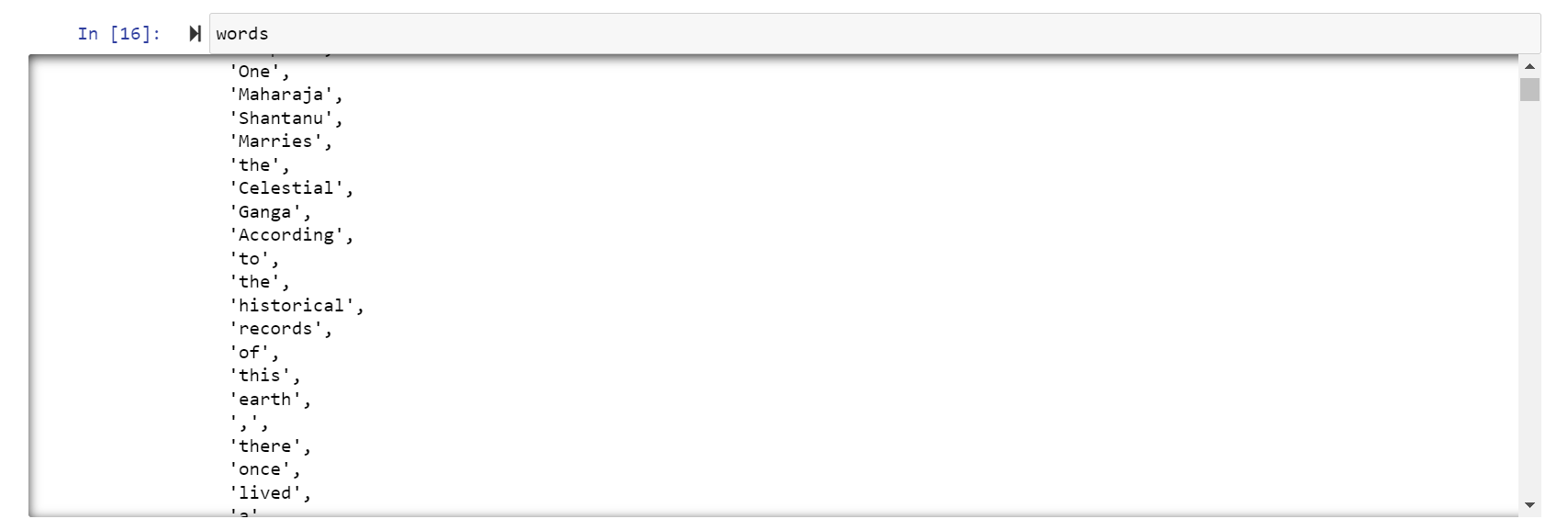
Language in its original form cannot be accurately processed by a machine, so you need to process the language to make it easier for the machine to understand. The first part of making sense of the data is through a process called tokenization, or splitting strings into smaller parts called tokens.

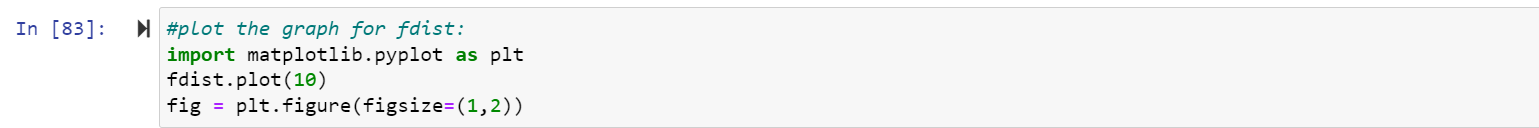
A token is a sequence of characters in text that serves as a unit. Based on how you create the tokens, they may consist of words, emoticons, hashtags, links, or even individual characters. A basic way of breaking language into tokens is by splitting the text based on whitespace and punctuation.

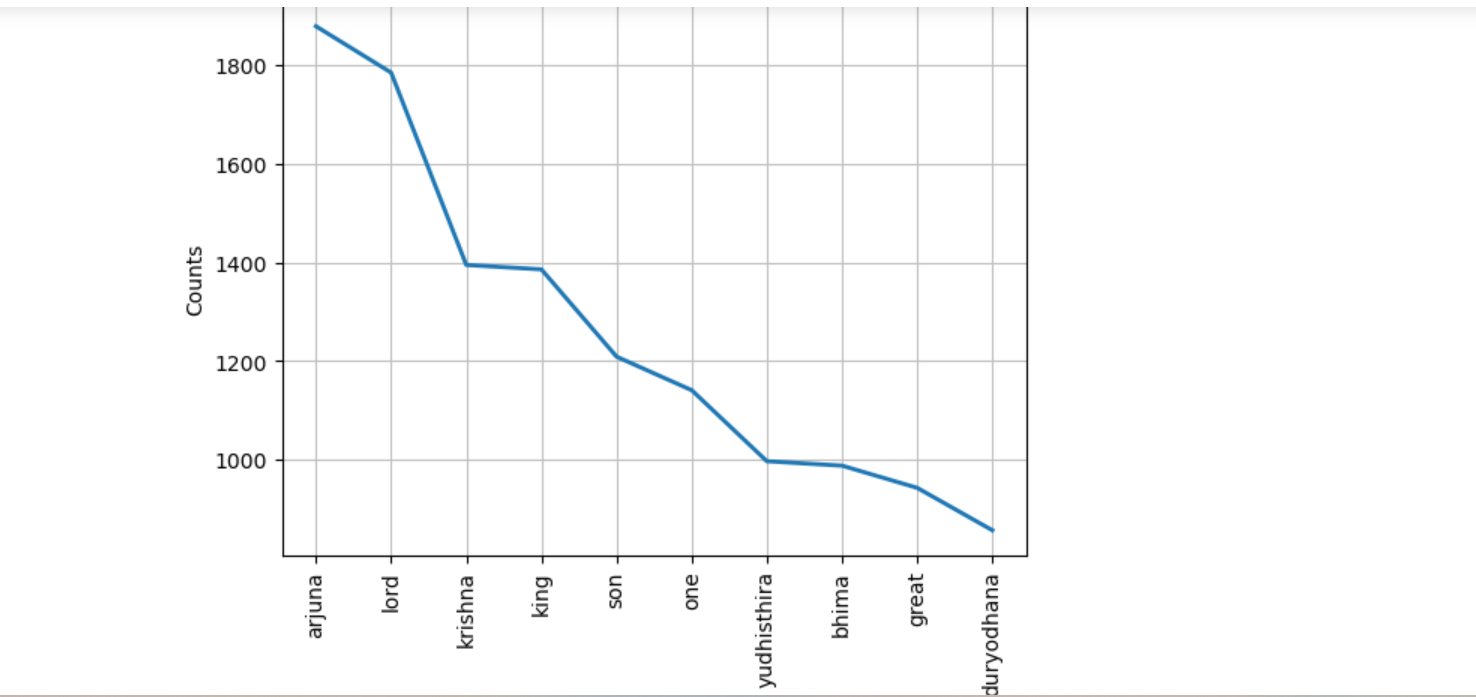










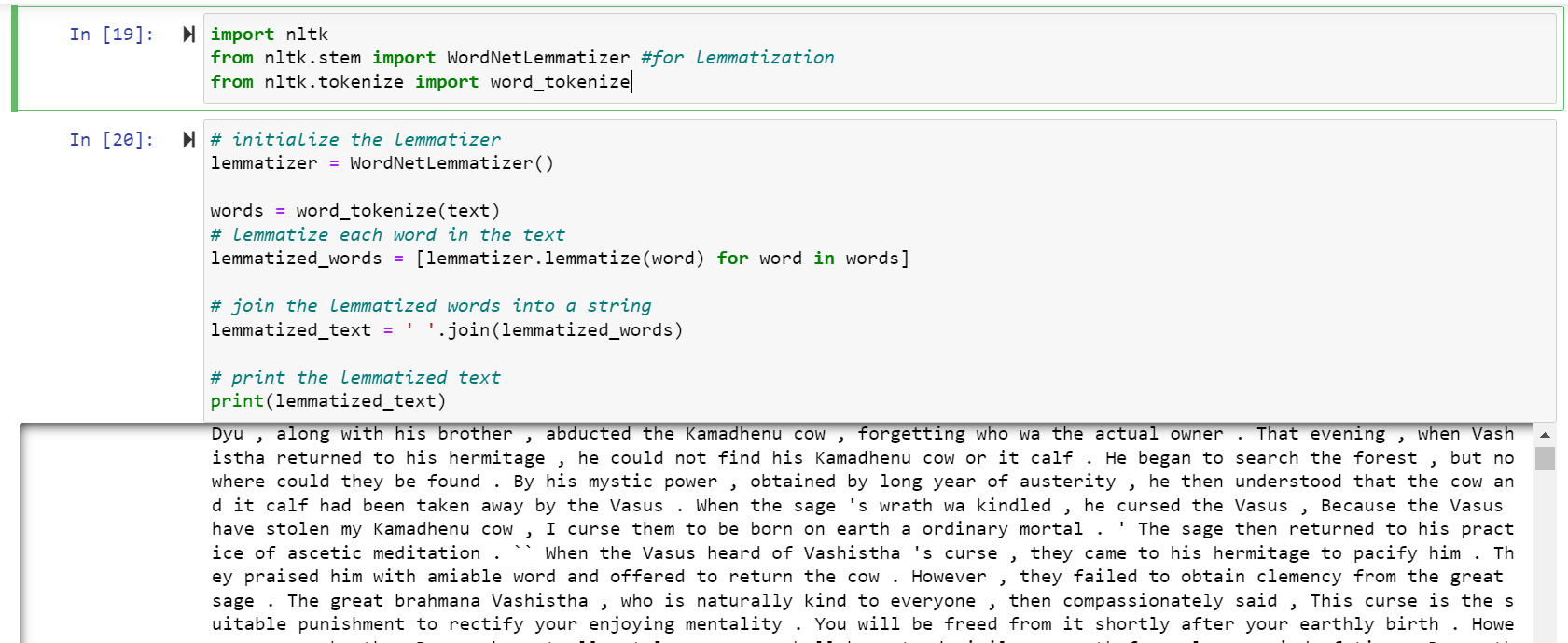


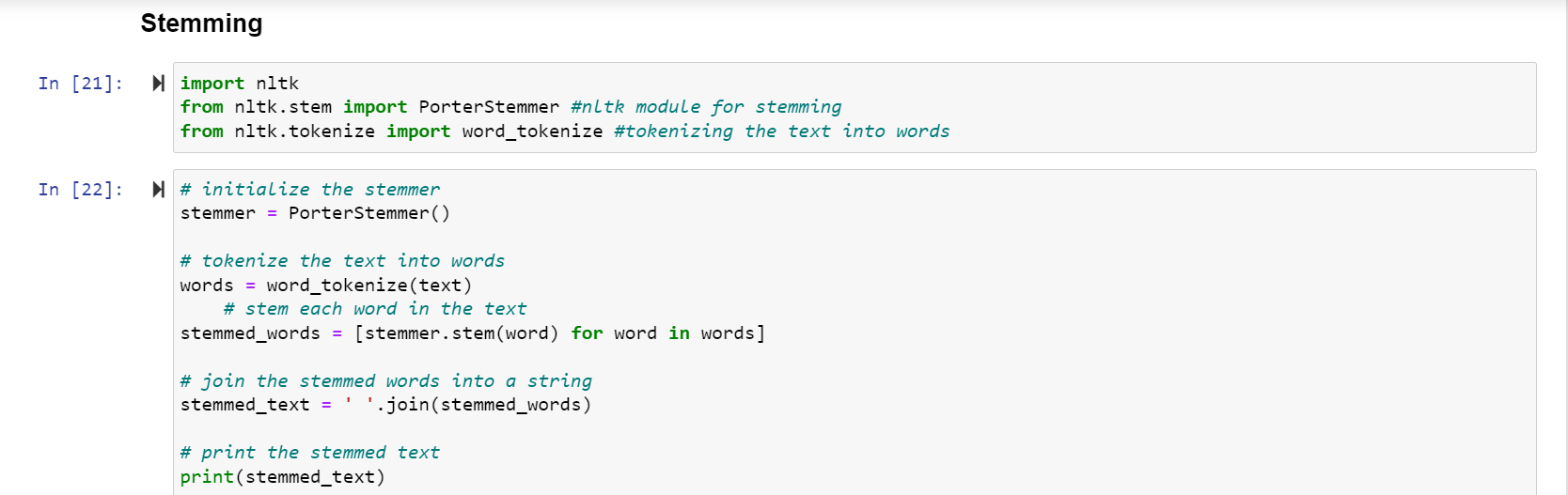
STEP 3 : Normalizing the Data

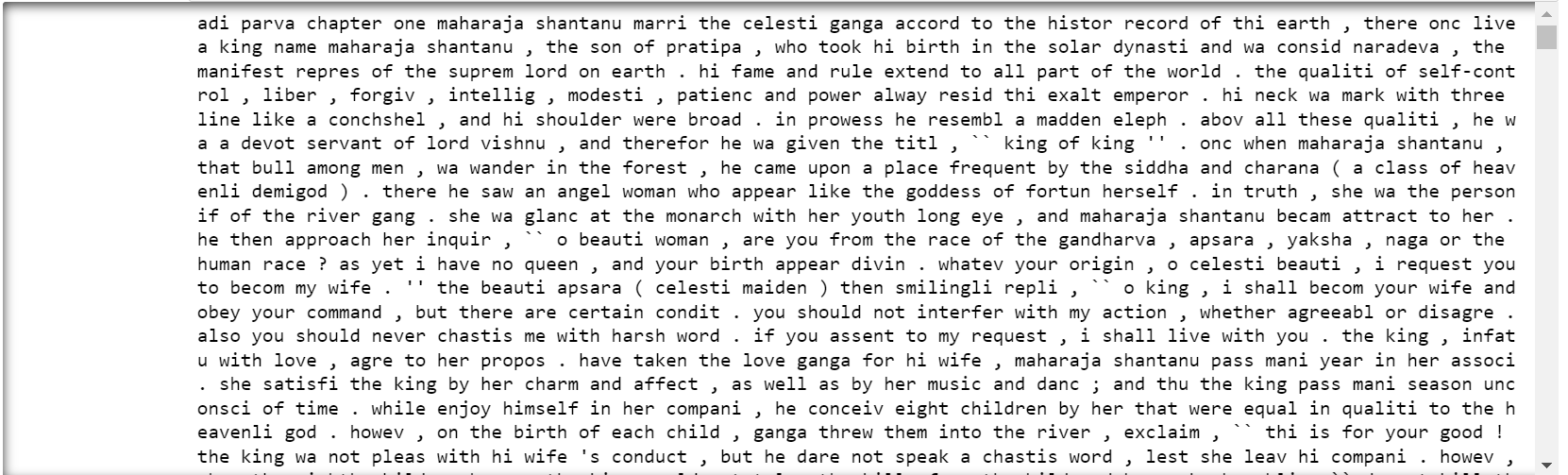
Normalization helps group together words with the same meaning but different forms. Without normalization, “ran”, “runs”, and “running” would be treated as different words, even though you may want them to be treated as the same word.

Stemming is a process of removing affixes from a word. Stemming, working with only simple verb forms, is a heuristic process that removes the ends of words.

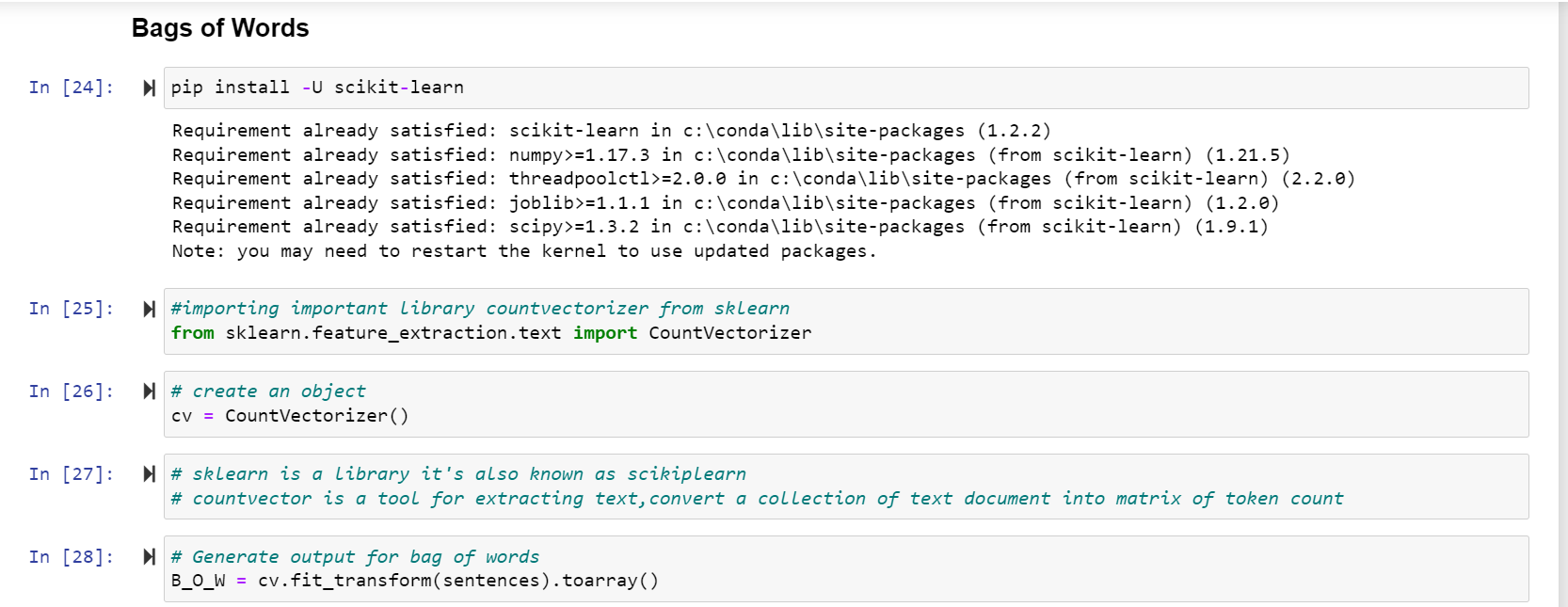
The lemmatization algorithm analyzes the structure of the word and its context to convert it to a normalized form. Therefore, it comes at a cost of speed. A comparison of stemming and lemmatization ultimately comes down to a trade off between speed and accuracy.











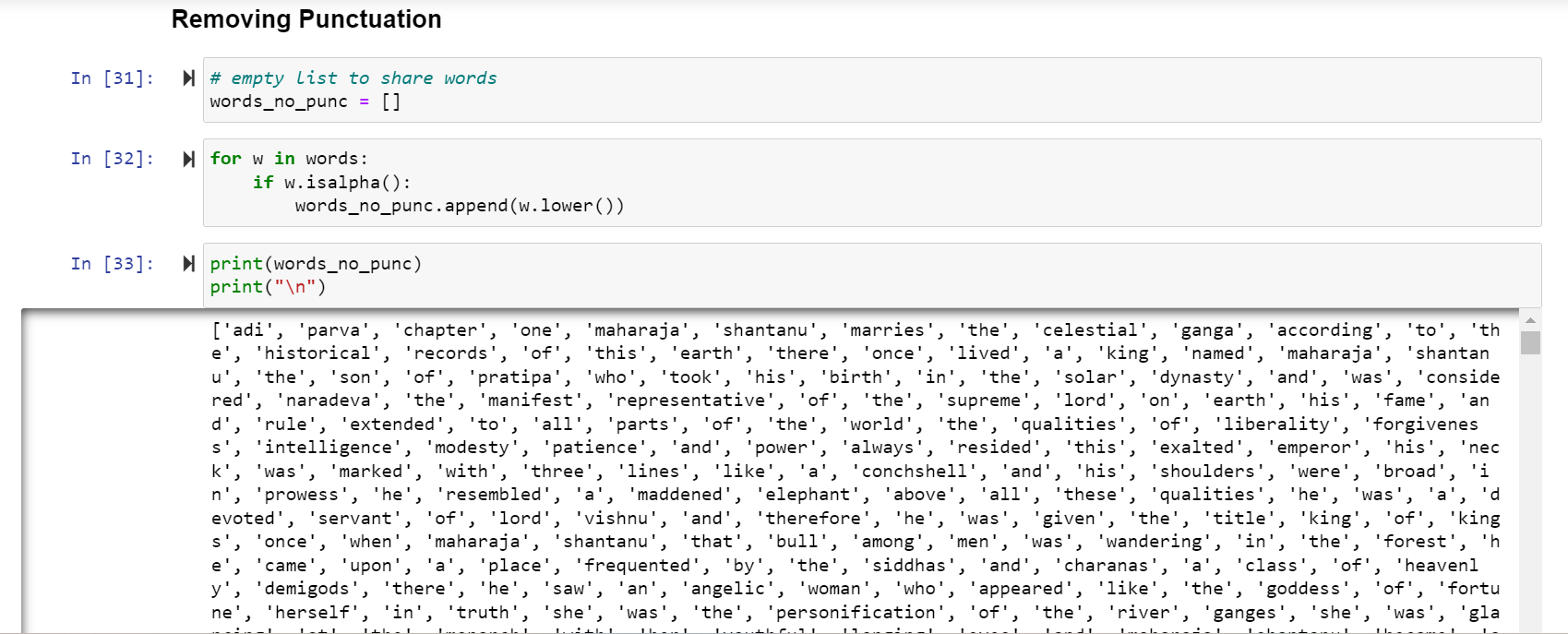


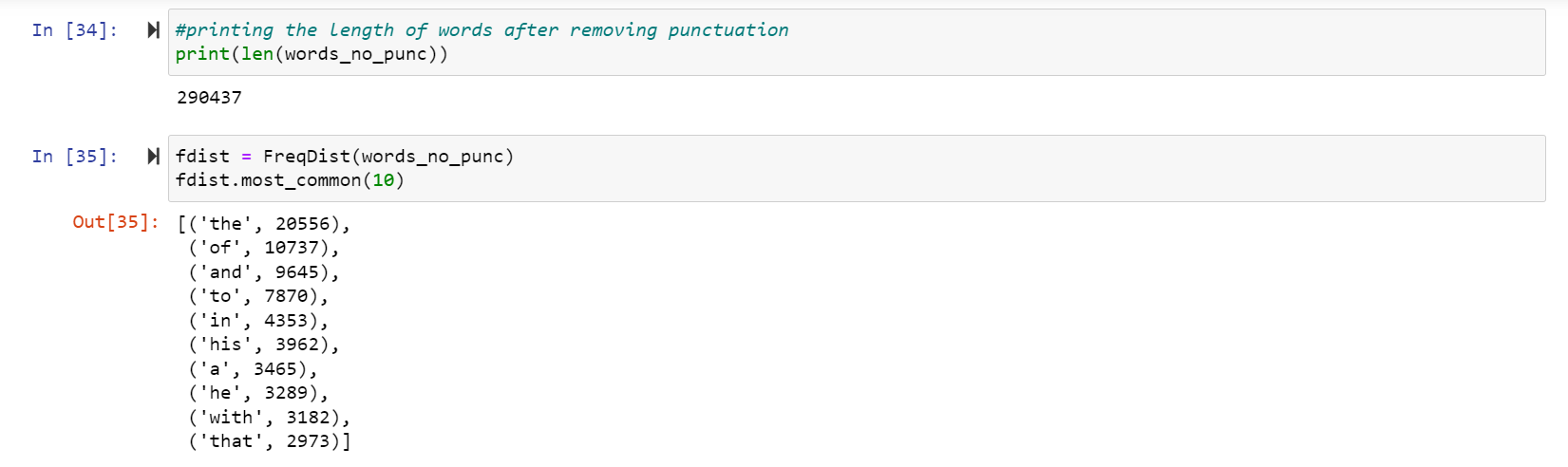


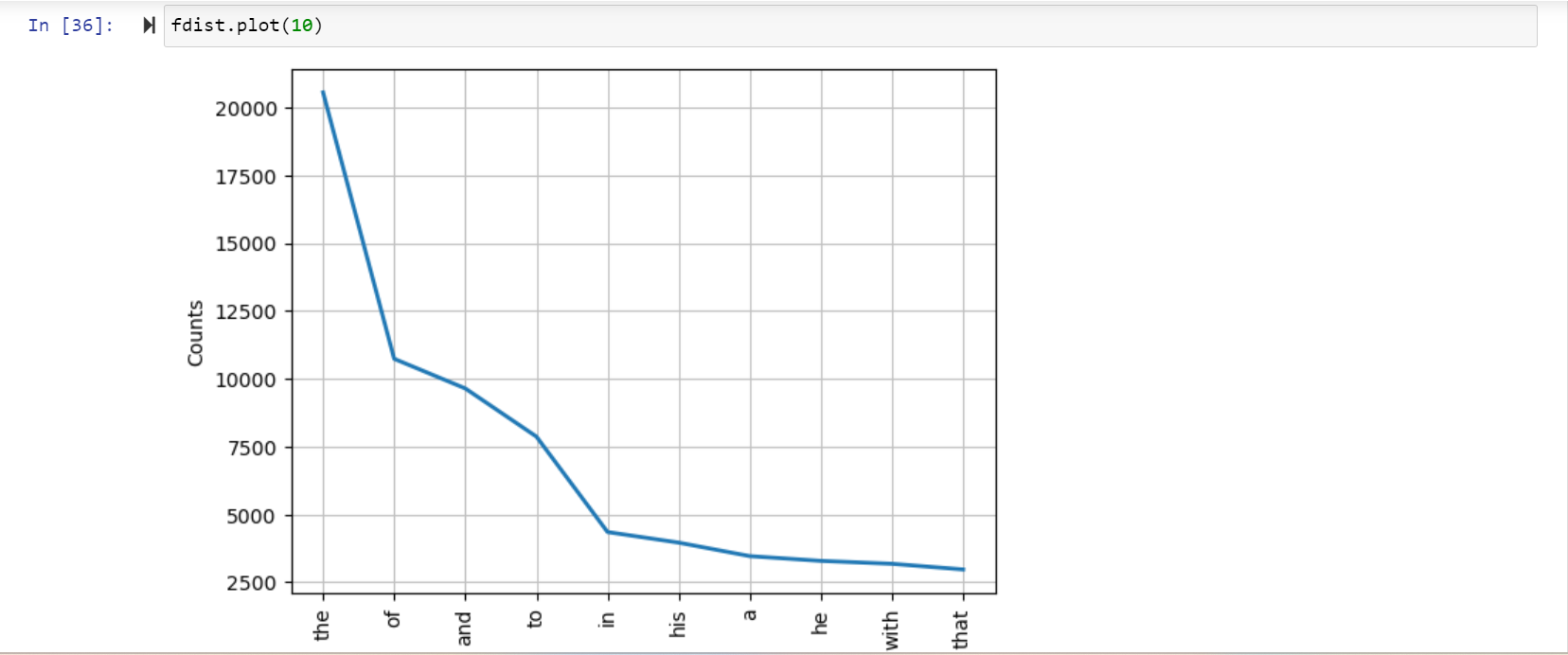
STEP 4 : Removing Noise from the Data

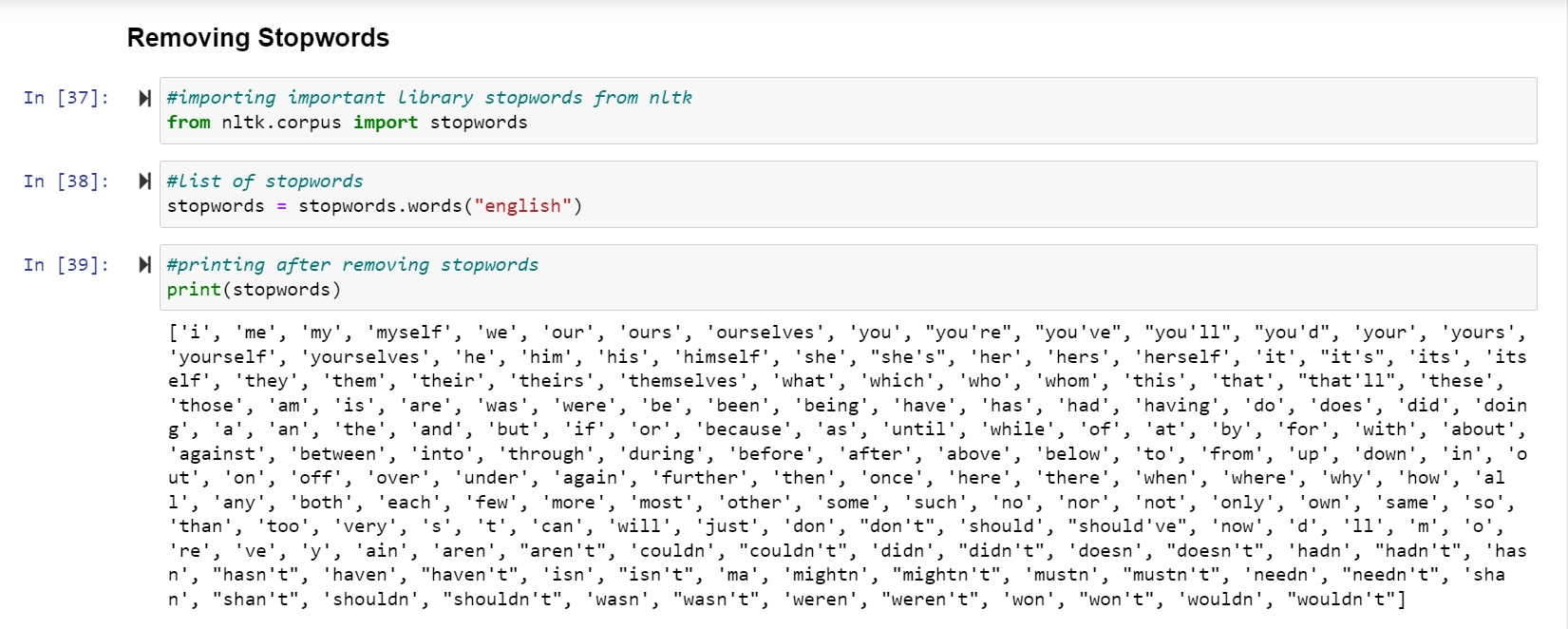
In this step, we will remove noise from the dataset. Noise is any part of the text that does not add meaning or information to data.

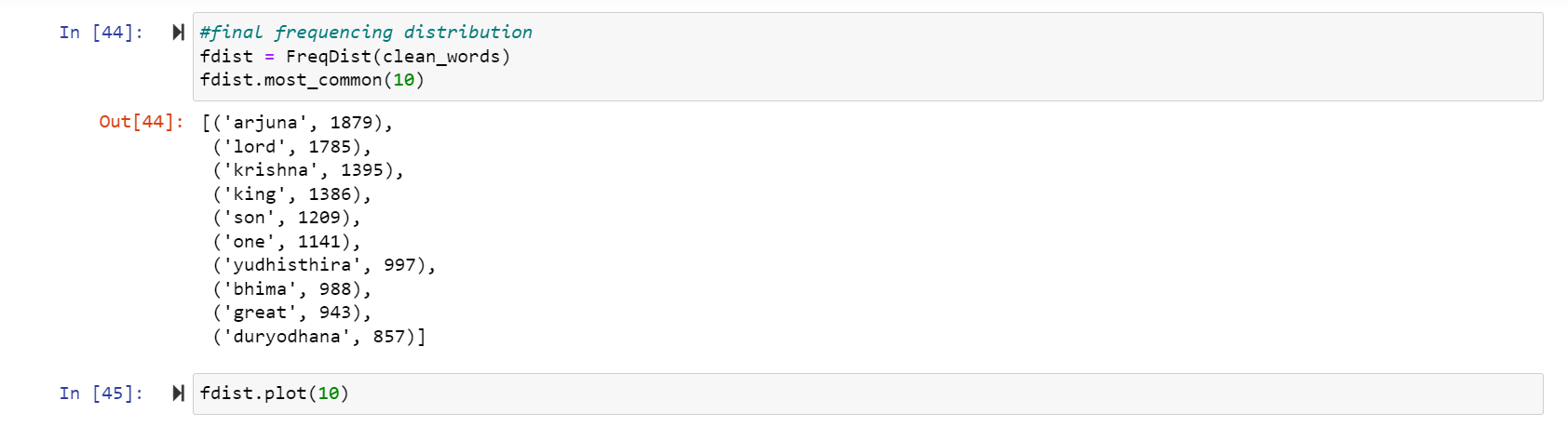
Noise is specific to each project, so what constitutes noise in one project may not be in a different project. For instance, the most common words in a language are called stop words. Some examples of stop words are “is”, “the”, and “a”. They are generally irrelevant when processing language, unless a specific use case warrants their inclusion.

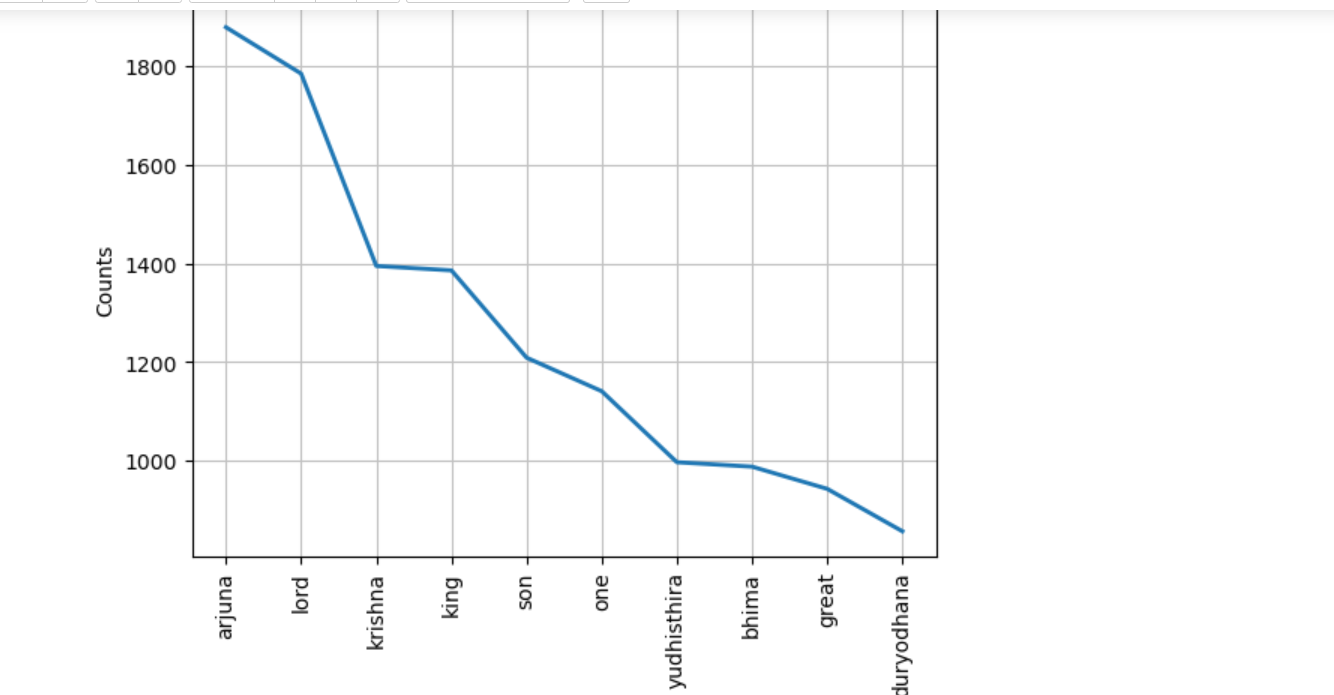










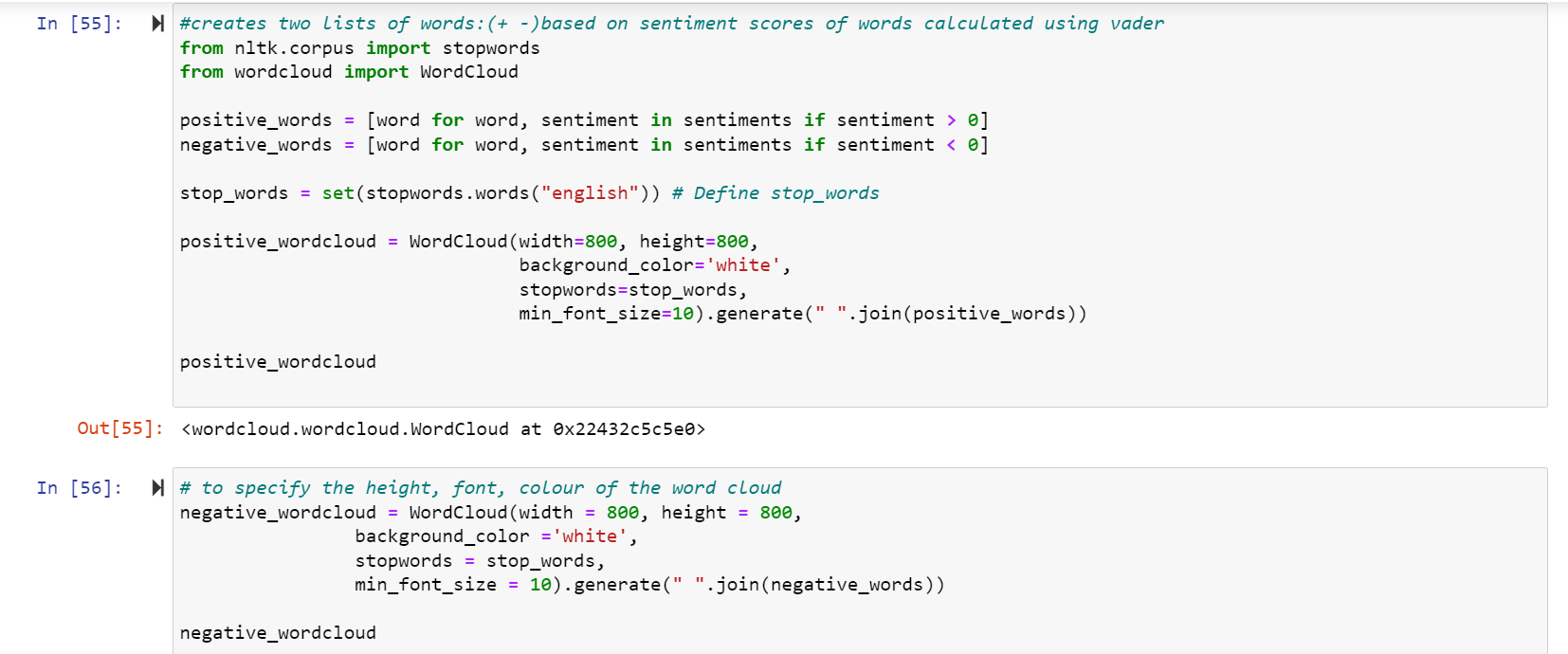
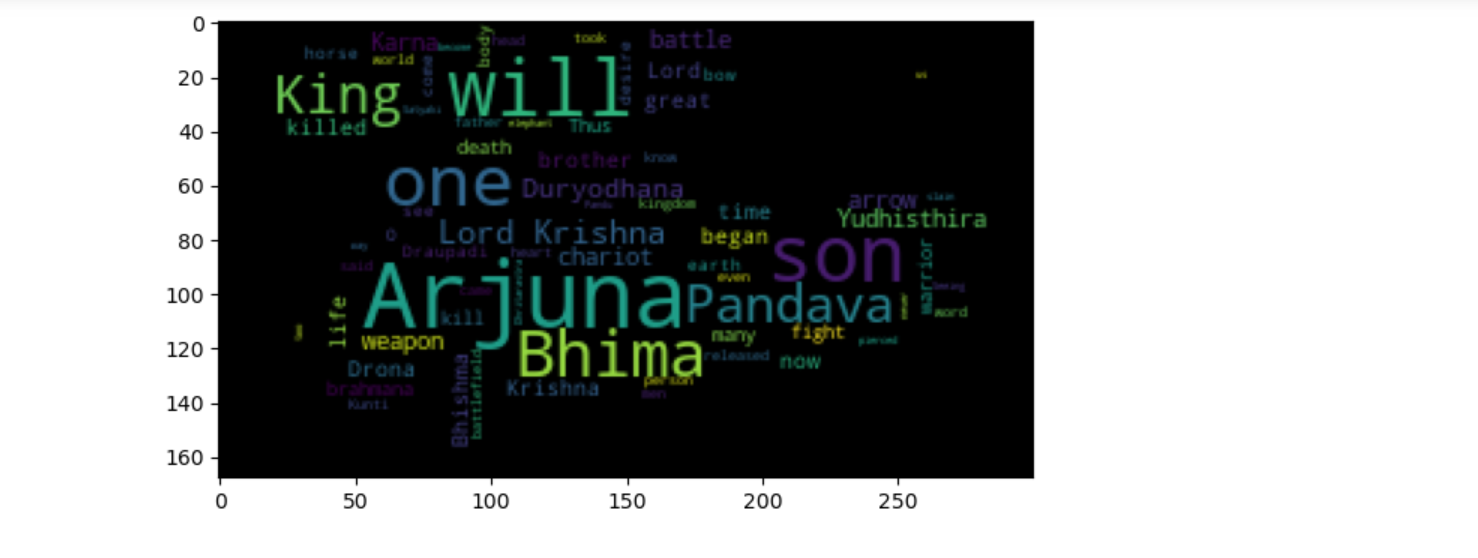


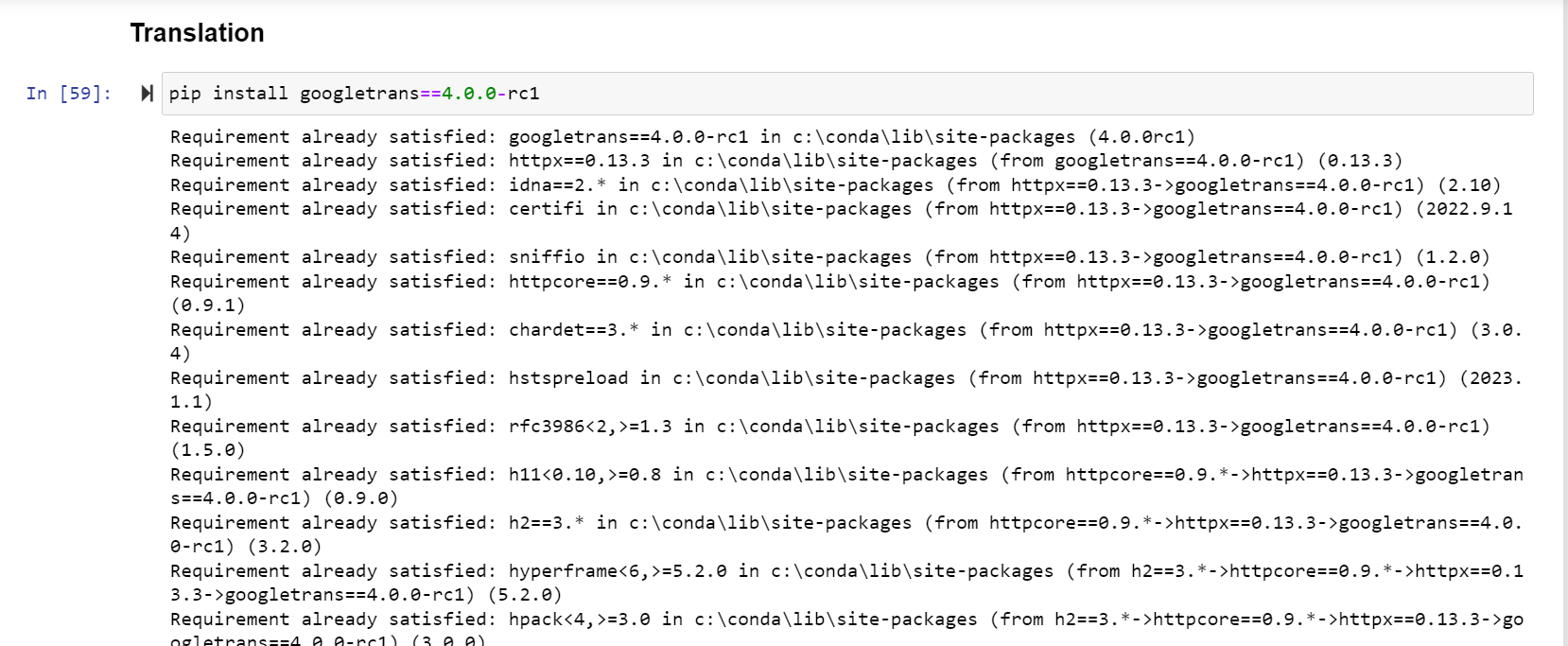


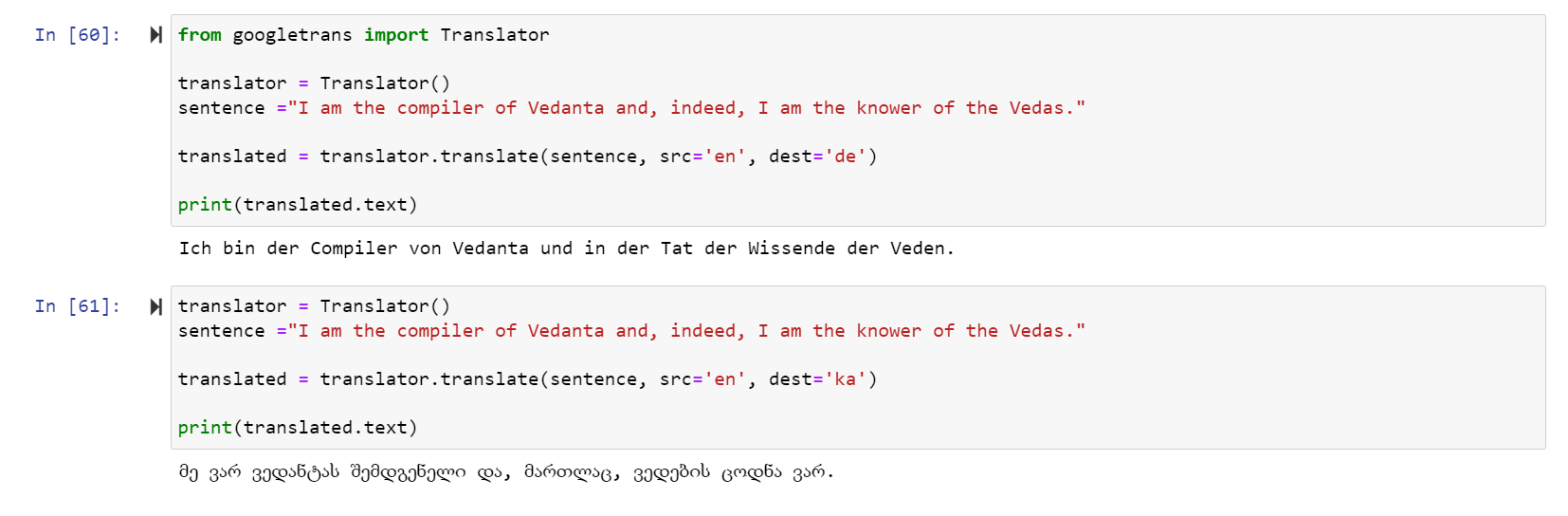
STEP 5 : Creating Wordcloud

Word Cloud provides an excellent option to analyze the text data through visualization in the form of tags, or words, where the importance of a word is explained by its frequency.

It's important to think about the context of the text and the specific meanings of the words being used. Be wary of outliers: Sometimes, a word may appear very large in the word cloud simply because it appears frequently, even if it's not particularly meaningful or relevant to the overall message of the text.







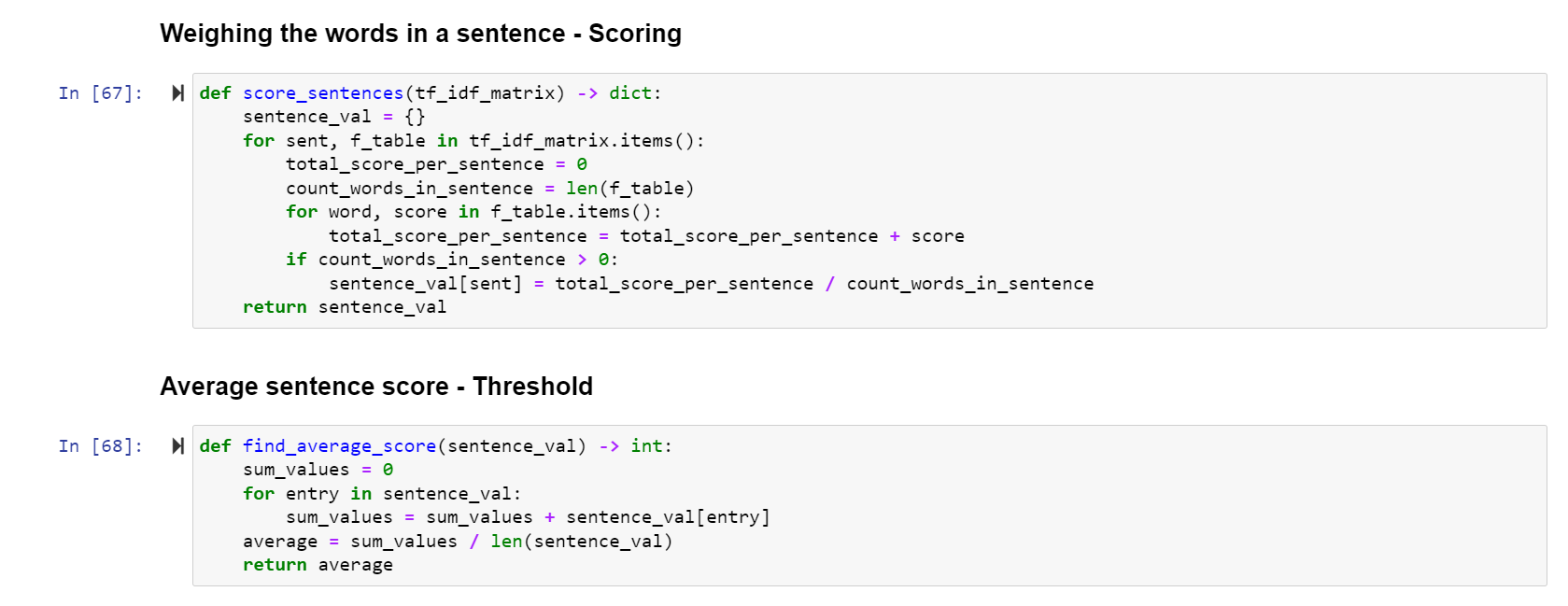
STEP 6 : Summarizing

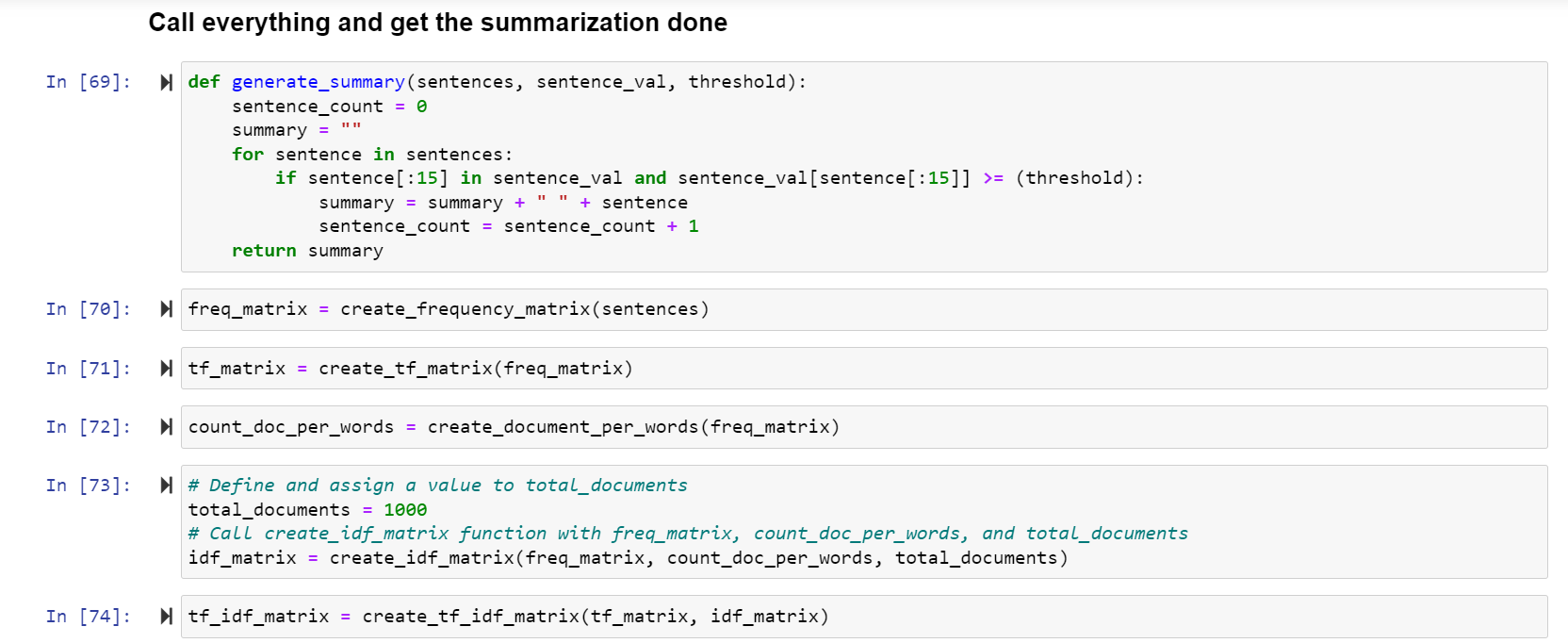
Text summarization in NLP is the process of summarizing the information in large texts for quicker consumption.

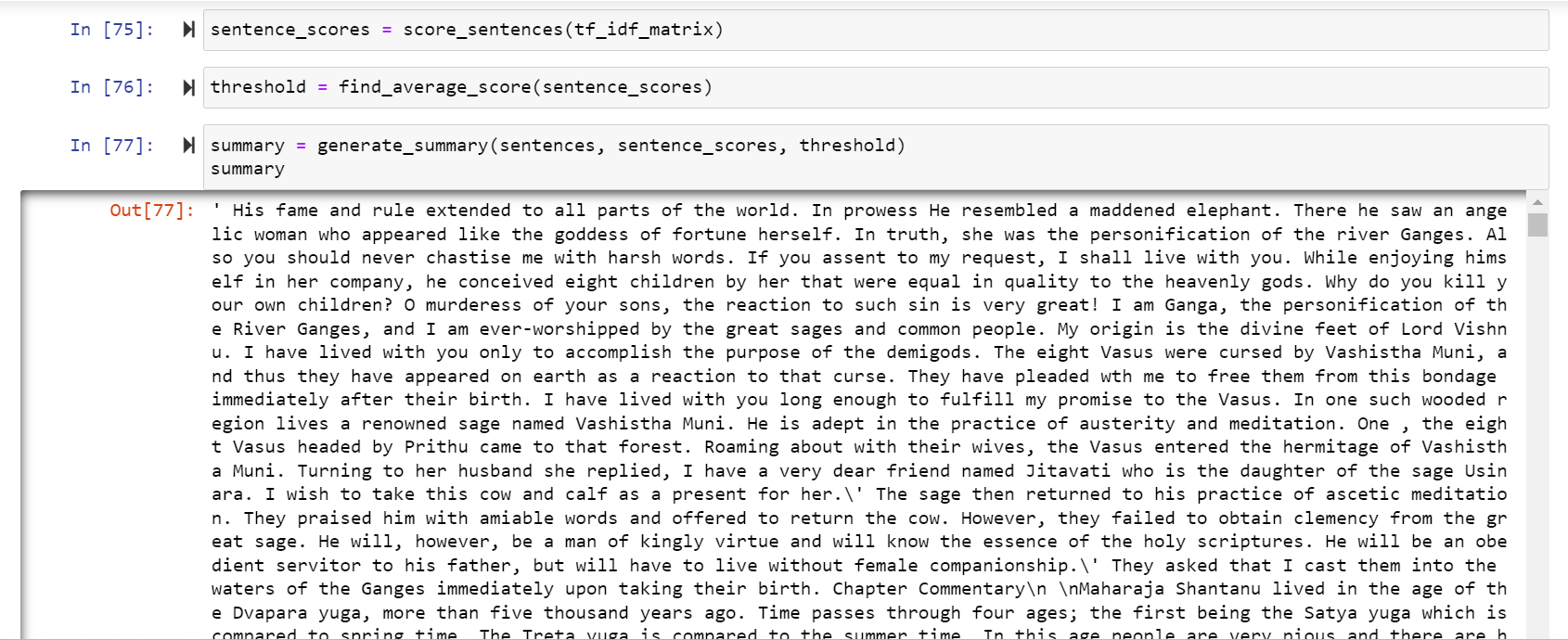












Conclusion :

In this project, first we import, open, read & print the script of mahabharat in a text form to begin with, first we check the type of data this files contain i.e.,string type,and the length of the file. Next step is to tokenize the file on the bases of words and sentences by importing important libraries such as sent\_tokenize & word\_tokenize from nltk package,Now let's find the frequency of the most common words in the file, from which, I have imported the FreqDist function. With the help of matplotlib,I have created the graph of the total words & sentences. After this remove unnecessary words,i.e removing punctuation and make the plot from it. To increase the search performance, I have removed stopwords i.e remove the low-level information from our text in order to give more focus to the important information.Next step is to be done with Stemming and Lemmatization, so that it reduced those words which are not important and make our data easy to work with. Now, I will categorize the words in a text (corpus) in correspondence with a particular part of speech, i.e POS tagging and analyse text document based on word count (Bag of Words).Now to summarize the data I will be creating frequency Matrix for the tokens. Making of word cloud on the basis of sentiment analysis of words having positive, negative or netural meaning. For that I have imported few images in which shape I want to make my word cloud.I wanted to check how the data of each paragraph is interrelated with each other, and on the basis of that data, I have done clustering using scatterd plot. This will show which data belongs to which cluster and how they are correspondence with each other. For clustering, we need to import libraries like pandas, numpy, matplotlib, kmeans, TfidfVectorizer.To get the data in any other language we can translate the data into that language. To check this I have translate one sentence into Korean and Hindi, for which I have to install Googletrans library and from it importing the translator.

Bibliography :

* <https://en.wikipedia.org/wiki>
* <https://www.kaggle.com>
* <https://monkeylearn.com>
* <https://www.ontotext.com>