

**CSE523 Machine Learning**

**Weekly Report 1**

**Group Name: Precision Précis**

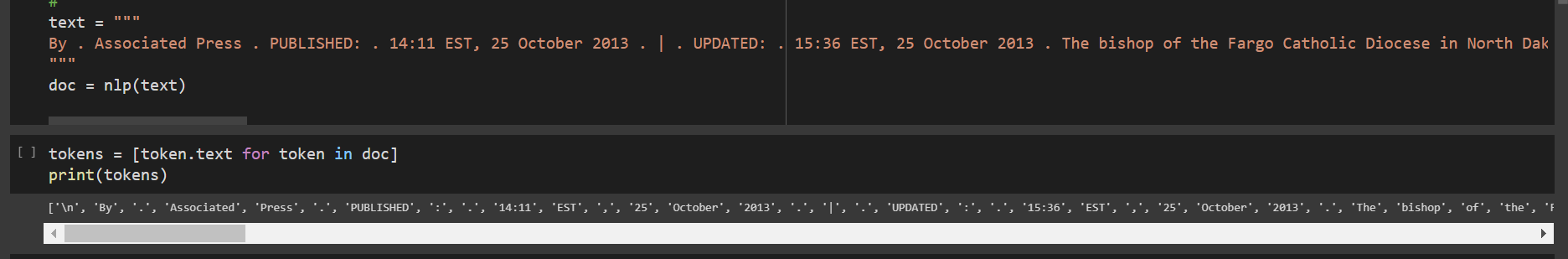
| **NAME** | **ENROLLMENT NUMBER** |
| --- | --- |
| Twinkle Popat | AU2040216 |
| Kathan Bhavsar | AU2040170 |
| Rushali Moteria | AU2040210 |
| Neel Buddhdev | AU2040176 |

1. **Tasks performed in the week and the outcomes of the tasks performed.**

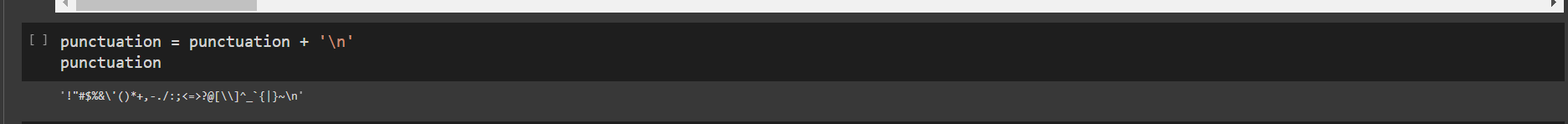
At the first level of our project, we tried to implement tokenization, followed by assigning frequencies to each word and sentence and then generating a sentence with the highest frequency score. The CNN Daily Mail News dataset consists of three columns: unique article id, Original Article, and highlights representing the actual output. The following steps have been followed to achieve the aforementioned outcome:

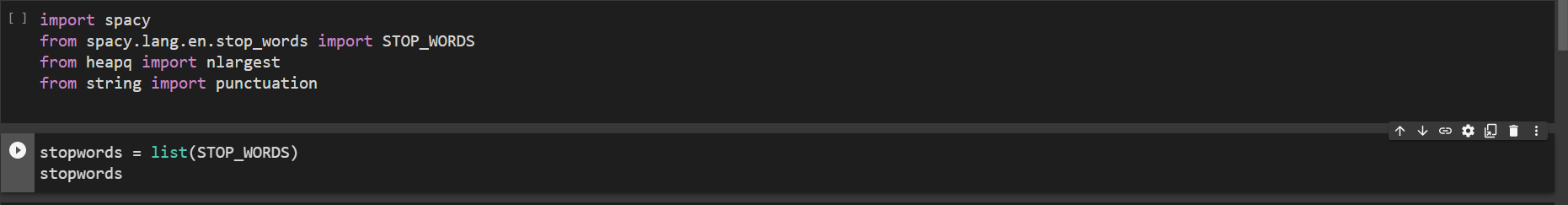
* **Preprocessing:**

As a part of preprocessing, we have done tokenization using the nlp library of python called spaCy in which the text is broken down into words or tokens. The outcome of word tokenization will be the spacy.tokens.doc.Doc object which is further used to form an text array of all the entities such as word and punctuation from the original text.



Further stop words and punctuation are removed to find the relevant frequencies of words which provide actual meaning to the sentence so that the final scores of sentences can be evaluated. For the initial sample code we have employed the already existing set of Stopwords from spaCy library.

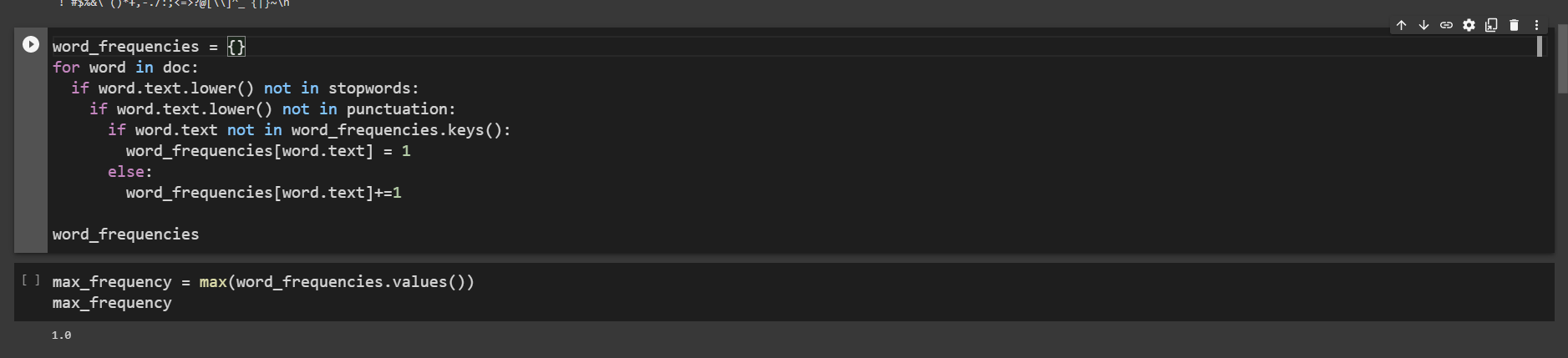




* **Text Analysis:**

As a part of text analysis, a dictionary (word\_frequencies) is created in which the frequency of each word is counted and stored. Frequencies will be counted for only those words that do not occur in stopwords and punctuation created earlier using spaCy.

In the end, we printed the maximum frequency, followed by the normalization of each frequency i.e., frequency / maximum frequency.



* **Sentence Selection:**

Next, we implement sentence tokenization, dividing the text into individual sentences. Depending on the frequency of the sentence's words, each sentence's score is calculated. Then according to the amount of summary we want to generate, we decide the sentence length (here, 0.3 that the summary will be 30% of the original Text).



Finally, the summary is printed by contacting the sentences with maximum scores.

1. **Tasks to be performed in the upcoming week.**

In our initial approach, we hard-coded the percentage of the summary to be printed. However, this is not appropriate for generating summaries for all kinds of texts. In the upcoming week, we aim to use TF-IDF Vectorization or Bag-of-words model for feature extraction and converting the sentences into numerical matrices of scores.TF-IDF or Bag-of-words using python libraries such as NLTK or scikit-learn.