

**Veermata Jijabai Technological Institute, Mumbai 400019**

**Experiment No.:** 04

**Aim:** Implement a sentiment classifier using LSTM Network. Use word embeddings Word2Vec to represent words as input to the model that is sentiment classifier. Display accuracy of your model using 4-fold cross validation on the selected dataset.

**Hint -** Dataset can be used from Kaggle, Hugging Faces or product reviews given by customers.

**Name:** Kiran K Patil

**Enrolment No.:** 211070904

**Branch:** Computer Engineering

**Batch:** D

**Theory:**

**1. Sentiment Analysis:**

* Sentiment analysis is a natural language processing (NLP) task that involves determining the sentiment expressed in a piece of text. Sentiments are often categorized as positive, negative, or neutral. In the context of product reviews or customer feedback, sentiment analysis helps in understanding the opinions and emotions expressed by users.

**2. Long Short-Term Memory (LSTM) Networks:**

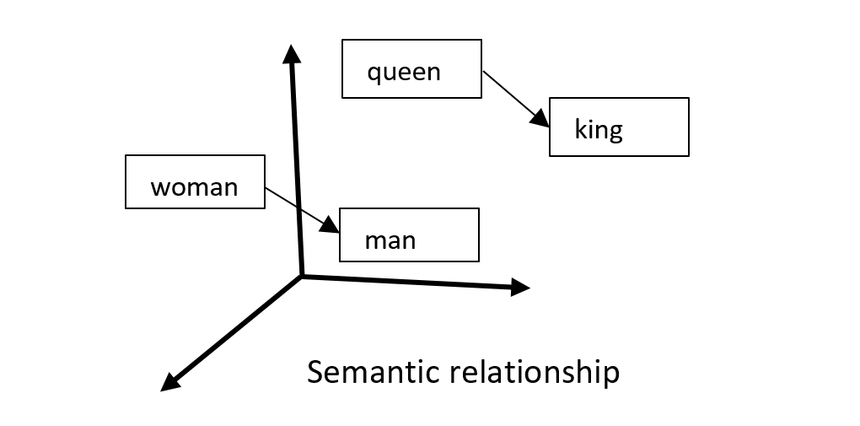
* LSTMs are a type of recurrent neural network (RNN) architecture designed to address the challenges of vanishing gradients in traditional RNNs. The LSTM units are equipped with memory cells and gates (input, output, and forget gates) that allow them to capture and propagate information over long sequences. This makes LSTMs well-suited for tasks involving sequential data, such as sentiment analysis.
* The architecture of an LSTM enables the model to retain information from earlier parts of a sequence, making them effective in capturing dependencies and patterns in language that span across multiple words.

**3. LSTM Python for Text Classification**

* There are many classic classification algorithms like Decision trees, RFR, SVM, that can fairly do a good job, then why to use LSTM for classification?
* One good reason to use LSTM is that it is effective in memorizing important information.
* If we look and other non-neural network classification techniques they are trained on multiple word as separate inputs that are just word having no actual meaning as a sentence, and while predicting the class it will give the output according to statistics and not according to meaning. That means, every single word is classified into one of the categories.
* This is not the same in LSTM. In LSTM we can use a multiple word string to find out the class to which it belongs. This is very helpful while working with Natural language processing. If we use appropriate layers of embedding and encoding in LSTM, the model will be able to find out the actual meaning in input string and will give the most accurate output class. The following code will elaborate the idea on how text classification is done using LSTM.

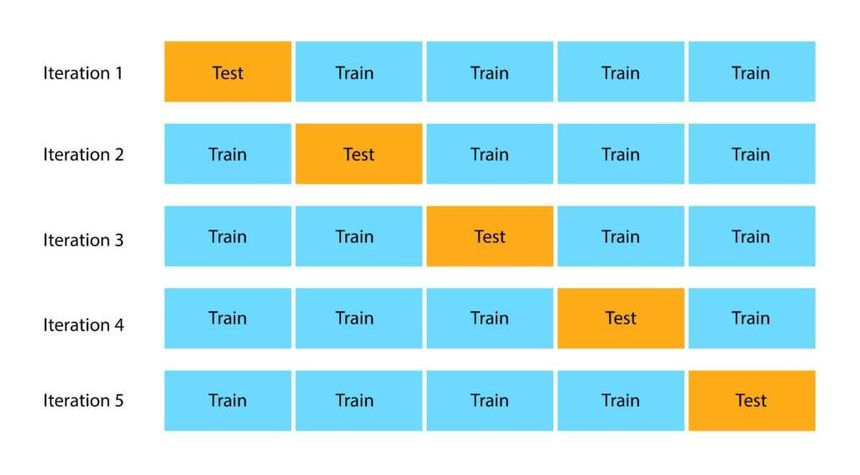
**4. Word Embeddings (Word2Vec):**

* Word embeddings are vector representations of words that capture semantic relationships. Word2Vec, developed by Mikolov et al., is a popular word embedding technique that learns continuous vector representations of words based on their context in a given corpus.
* The Word2Vec model is trained by predicting the context words around a target word in a given window. The resulting embeddings place similar words close to each other in the vector space. This helps the model to capture the meaning and relationships between words in a more meaningful way compared to traditional one-hot encoding.
* The learned word embeddings carry semantic information, allowing the model to understand the contextual meaning of words in the dataset.



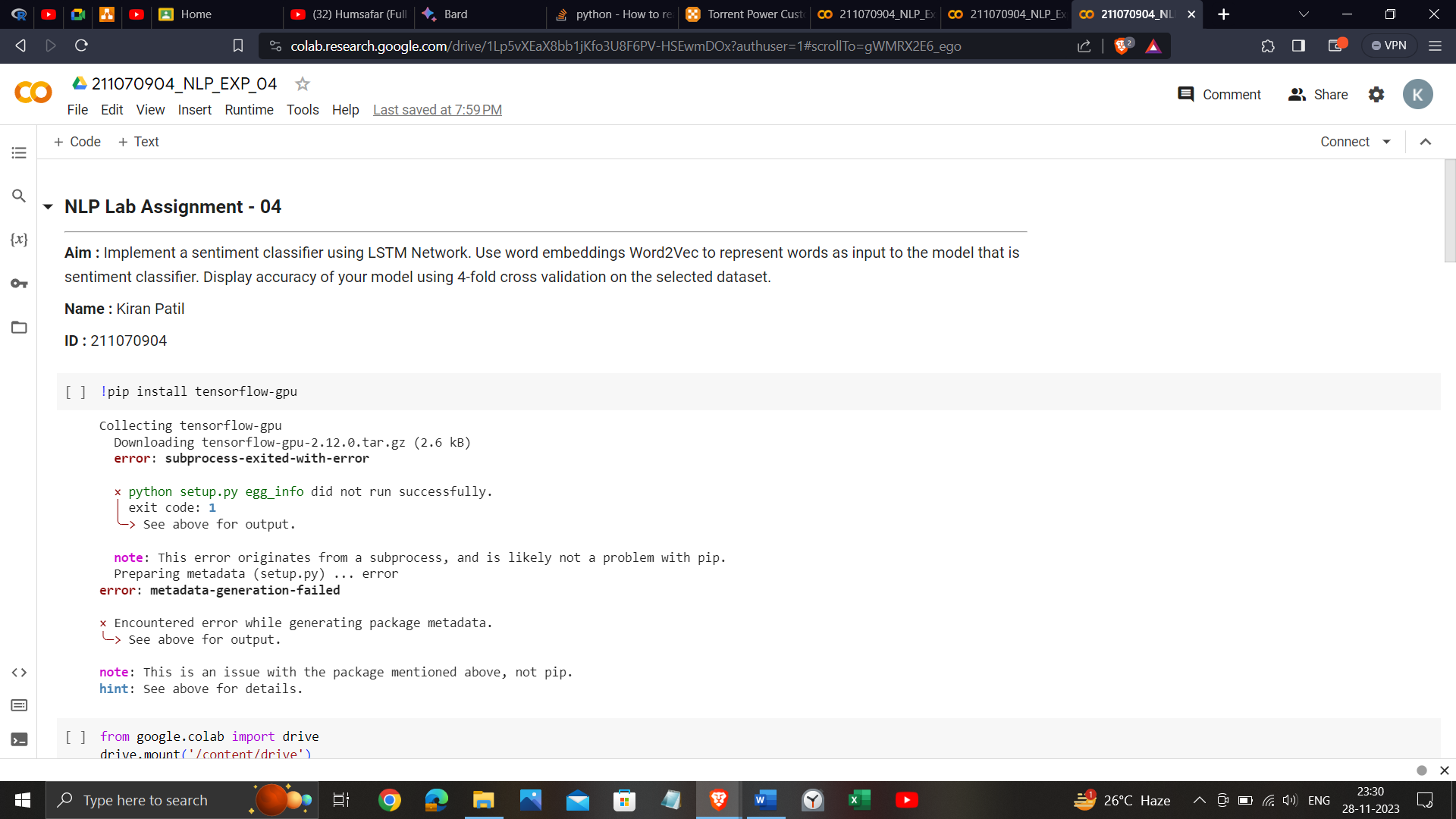
**4. Cross-Validation:**

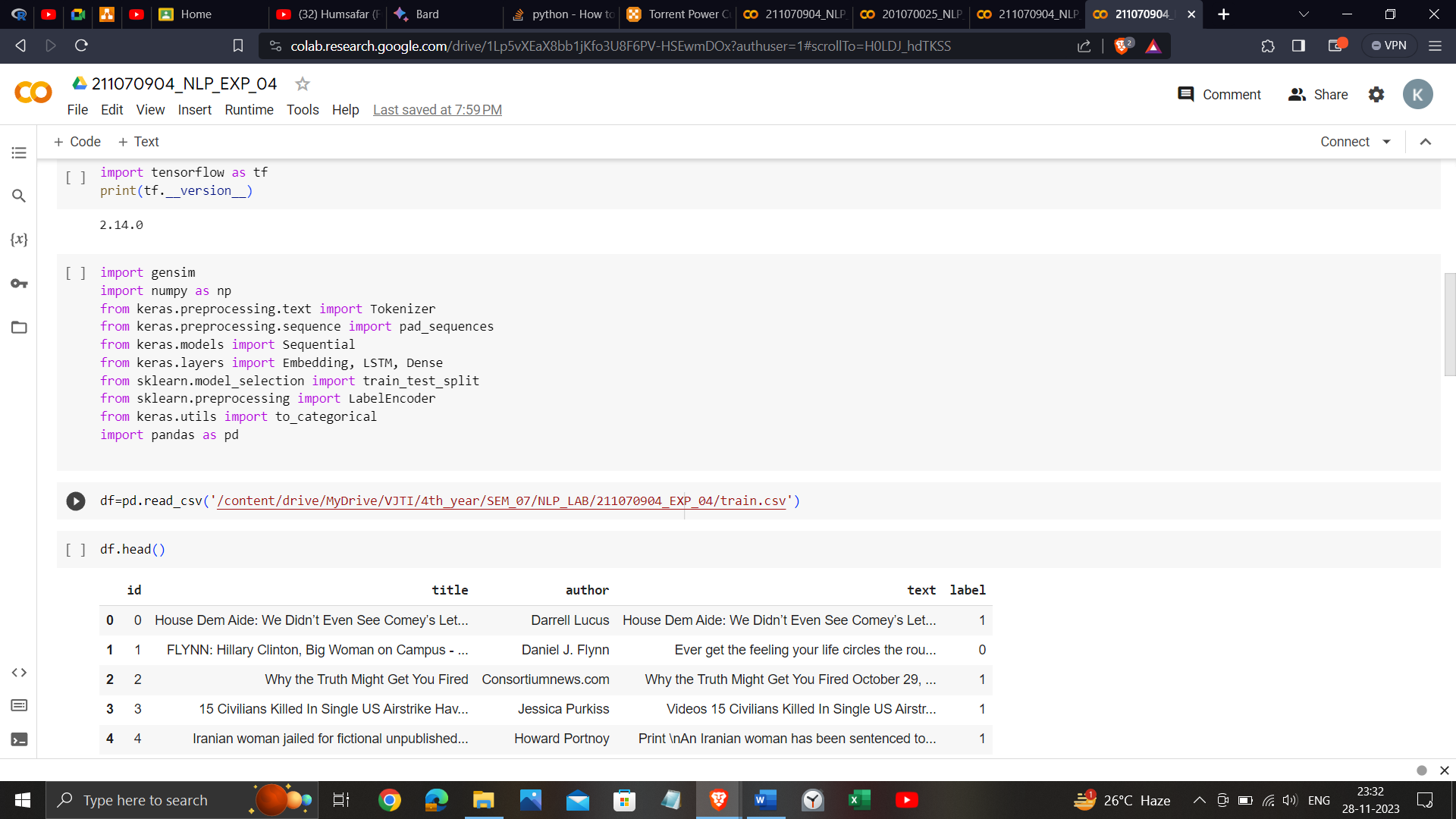
* Cross-validation is a resampling technique used to assess the performance of a machine learning model. In k-fold cross-validation:
* The dataset is divided into k subsets (folds).
* The model is trained k times, each time using k-1 folds for training and the remaining fold for validation.
* This process ensures that each data point is part of the validation set exactly once.
* Stratified k-fold ensures that the distribution of classes in each fold is representative of the overall distribution in the dataset. This is particularly important in sentiment analysis, where you want to ensure that each fold contains a balanced representation of positive and negative sentiments.
* Cross-validation helps in obtaining a more reliable estimate of the model's performance, reducing the risk of overfitting or underfitting to a specific subset of the data.

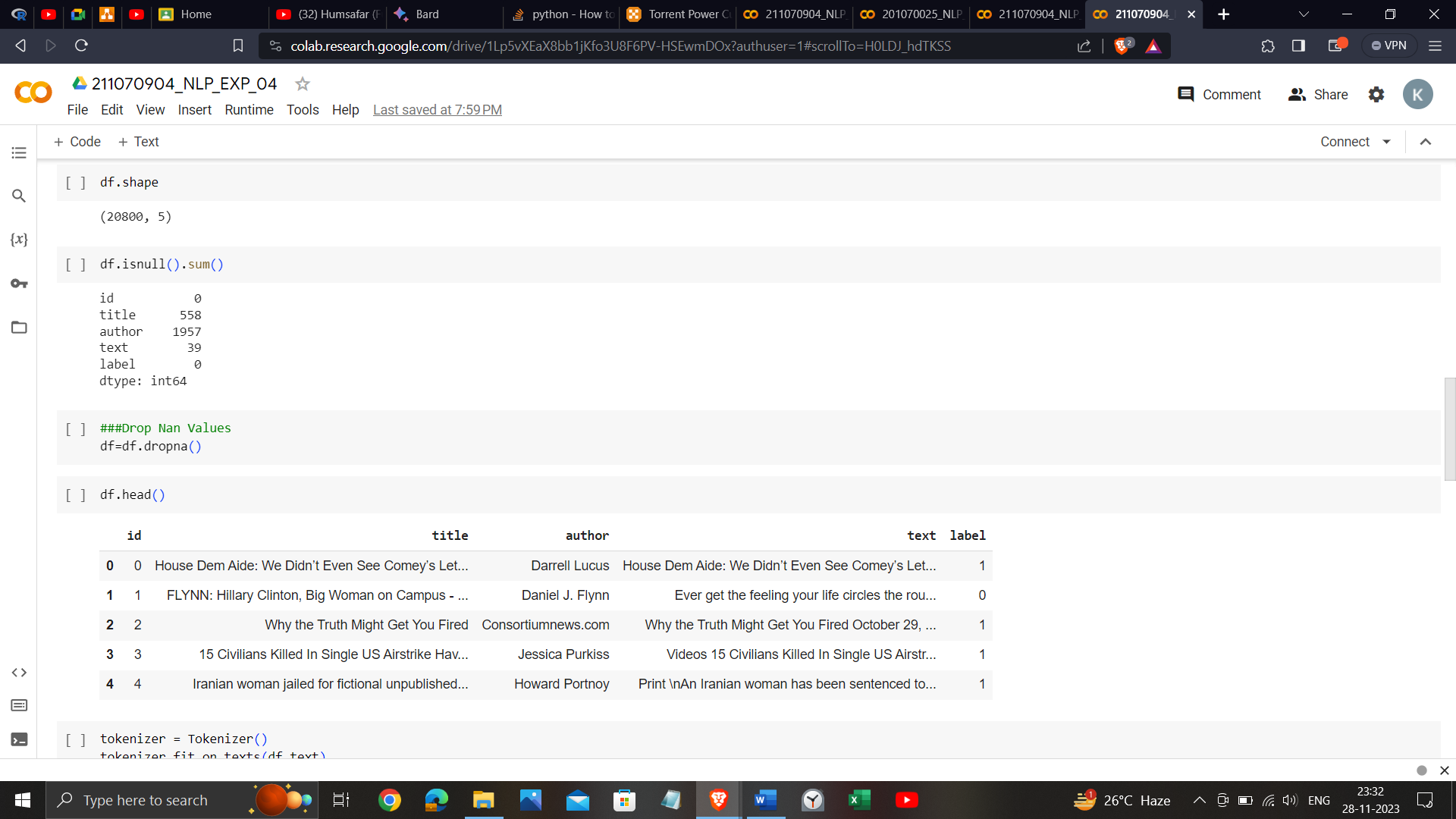


These foundational concepts lay the groundwork for building a sentiment classifier using LSTM networks with Word2Vec embeddings and performing a robust evaluation through cross-validation. The subsequent steps involve implementing these theories into code, training the model, and assessing its performance on real-world data.

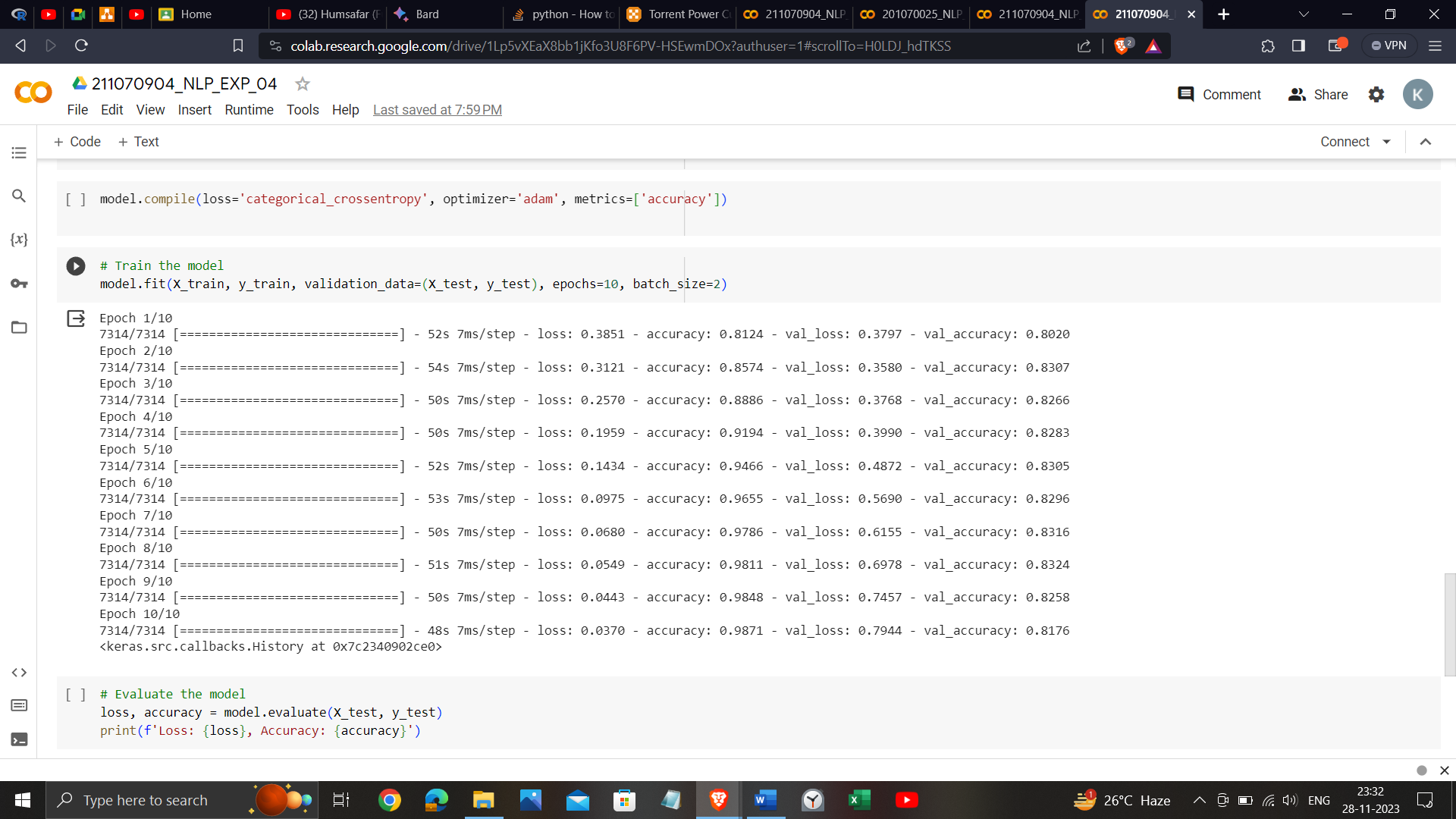
**Implementation:**











<https://colab.research.google.com/drive/1Lp5vXEaX8bb1jKfo3U8F6PV-HSEwmDOx?usp=sharing>

**Conclusion**:

In conclusion, this experiment provided a foundational understanding of text processing, covering character encoding, Unicode handling, regular expressions, and text normalization. These skills are crucial for effective data preprocessing, paving the way for more advanced natural language processing applications.