Topic Identification using LSTM

**Team Members:**

|  |  |  |
| --- | --- | --- |
| **S. No** | **Student Name** | **Student ID** |
| 1 | Vineela Dachepally | 11533067 |
| 2 | Akhila Kura | 11591726 |
| 3 | Teja Sree Mandadi | 11607666 |
| 4 | Panthangi Nikitha | 11592479 |

**Goals and Objectives:**

* **Motivation**:

Every one of us likes to read newspapers, books, and online articles to enhance our knowledge, but it takes a lot of time to read an entire article and understand every topic line by line and search for articles. To gather information based on category, this project analyzes the entire data and categorizes data based on topic.

* **Significance**:

It reduces the time and effort and understands long articles or books and categorizes data based on topics.

* **Objectives**:

The main objective of this project is to separate the data based on topics for easy understanding and searching.

* **Features**:

Quick understanding, text analyzing and topic identification.

**Related Work:**

Topic identification at times is a tedious task. In order to figure out the topic, we generally need to go through lengthy articles that aren't possible every time. Applications like this help us figure out swiftly and accurately what the article primarily focuses on. This saves a lot of time and effort and yet gives us the best results.

**Datasets:**

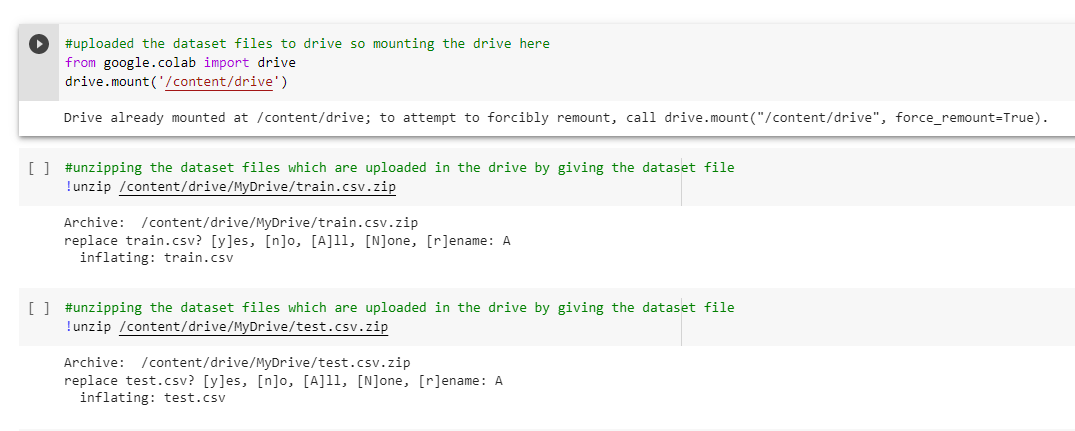
The training dataset that we used here basically consists of fields that are subjects like Chemistry, Physics etc. and has the id, title and abstract of that article. We trained the model to work efficiently and predict the output about which of these subjects hold a bigger portion in the article and what the topic is mainly about.

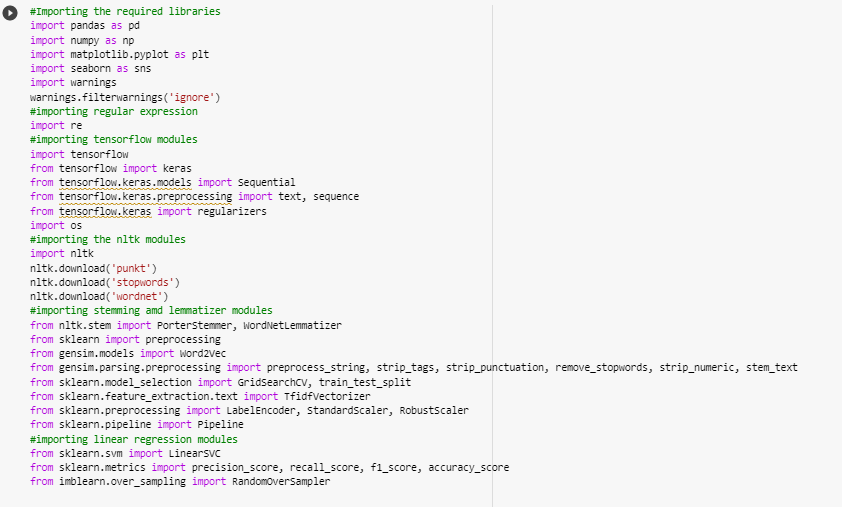
**Detail Design of Features and Analysis:**

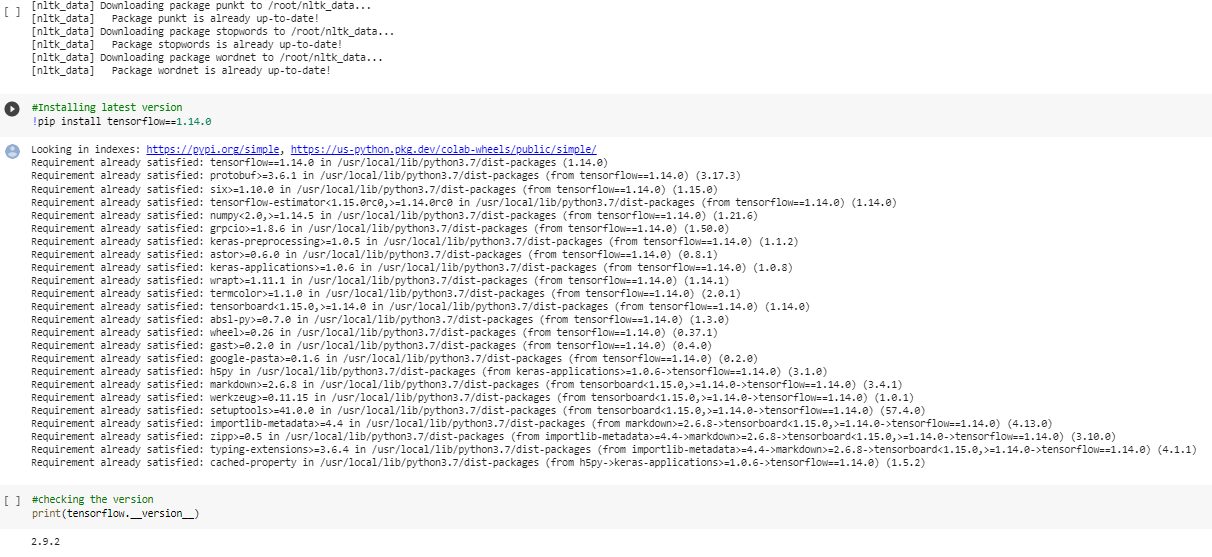
* Built and trained using Keras with TensorFlow that leads to the flexibility of our model.
* Used an open-source library, Gensim, that is used for unsupervised topic modeling, retrieving information based on similarity and to perform techniques like indexing.
* The implementation of RNN makes the prediction efficient by sequential characteristics and this is a key task in order to identify the topic based on the dependency of one word on the other for accurate topic prediction.
* As a part of the same, LSTM (long short time memory) is an RNN technique that is capable of long-term dependencies i.e., along with the sequential data, these long term dependencies result in accurate identification of the topic. It not only processes the data at specific data points but also the entire sequence of data.

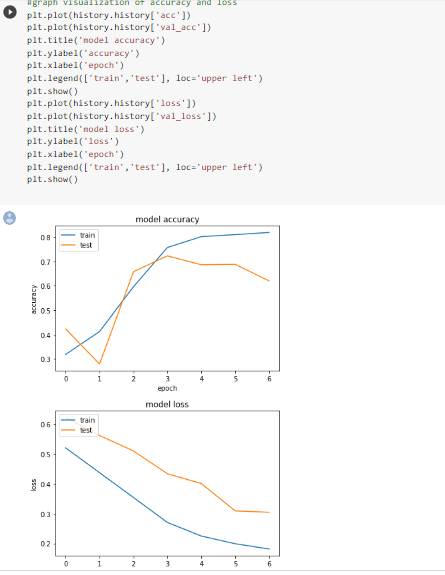
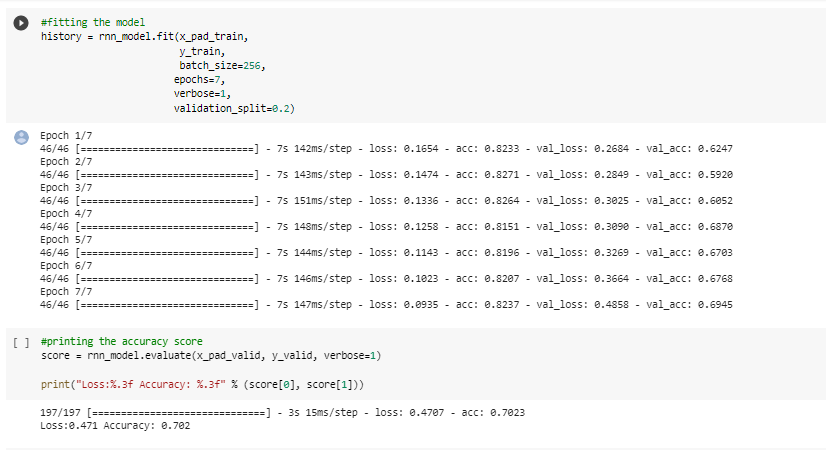
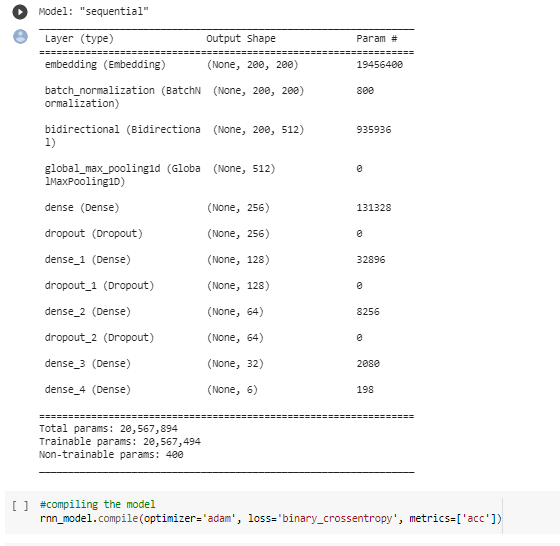
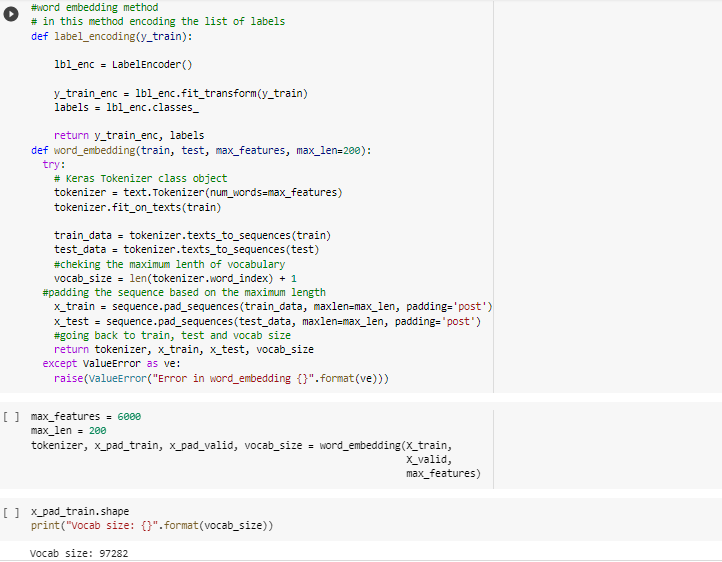
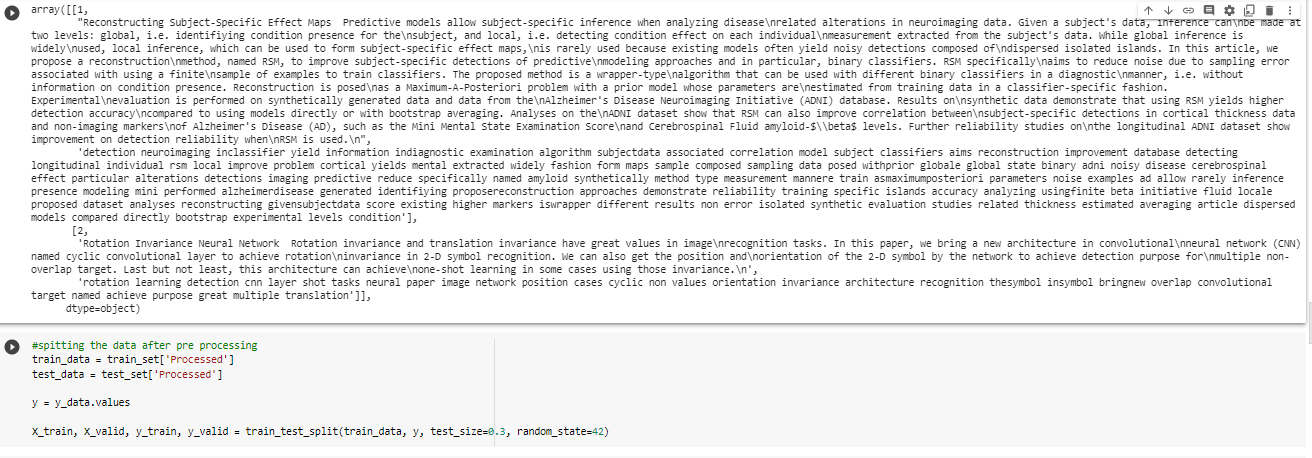
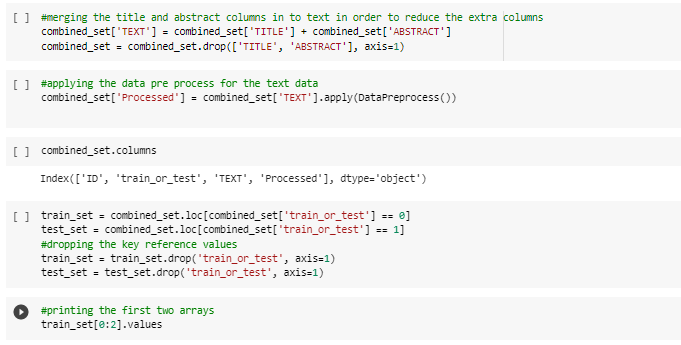
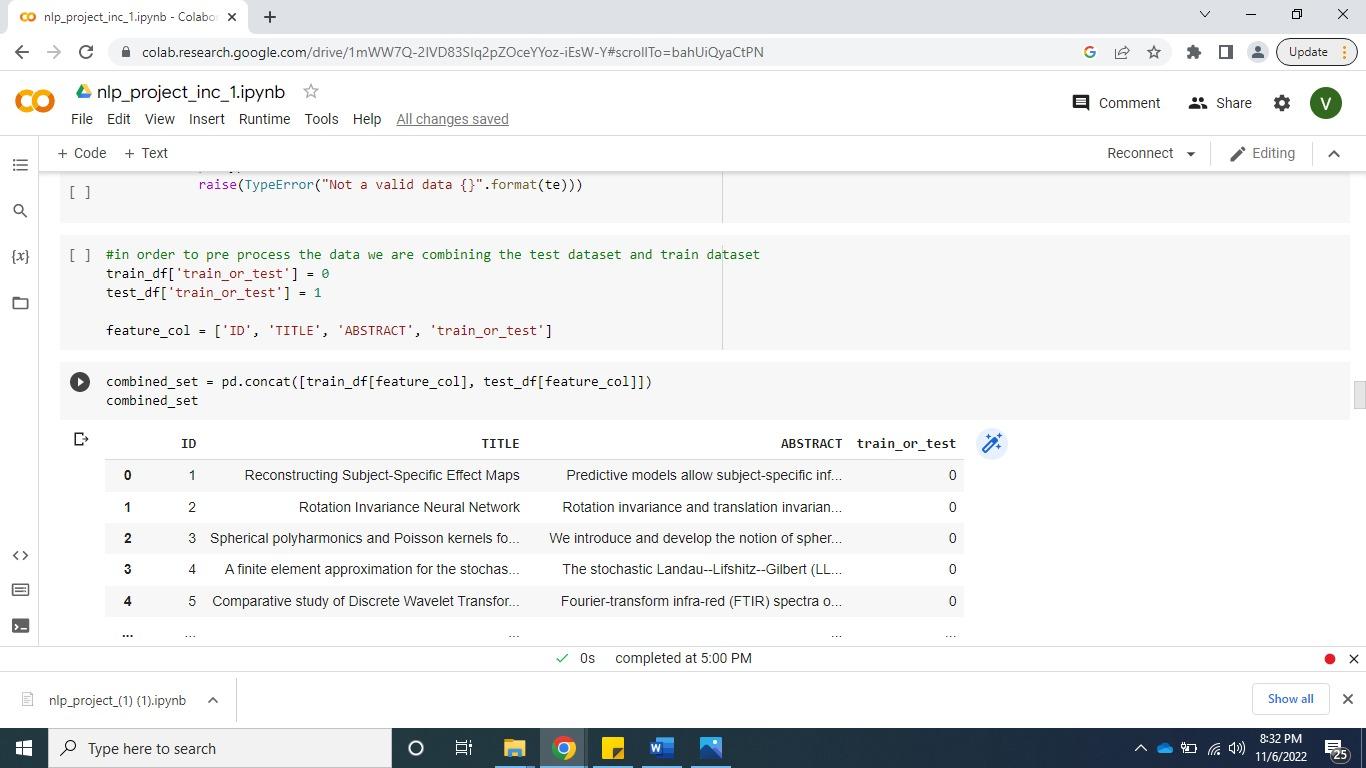
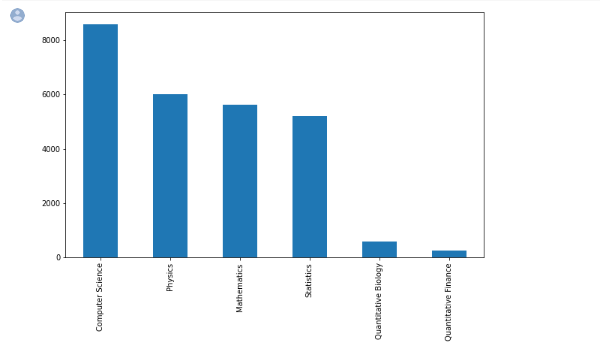
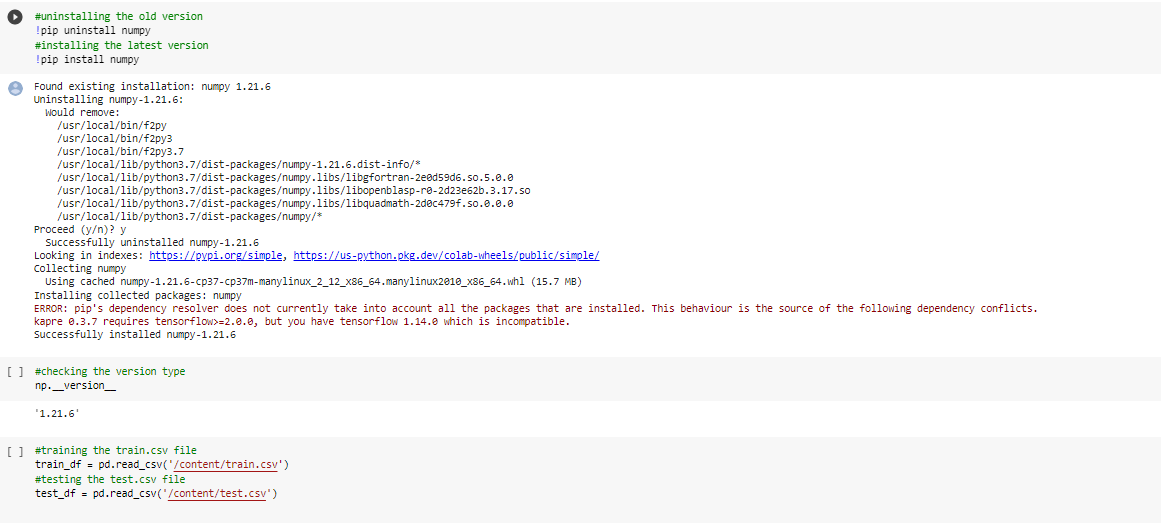
**Implementation:**

The code was implemented as below along with its respective results.

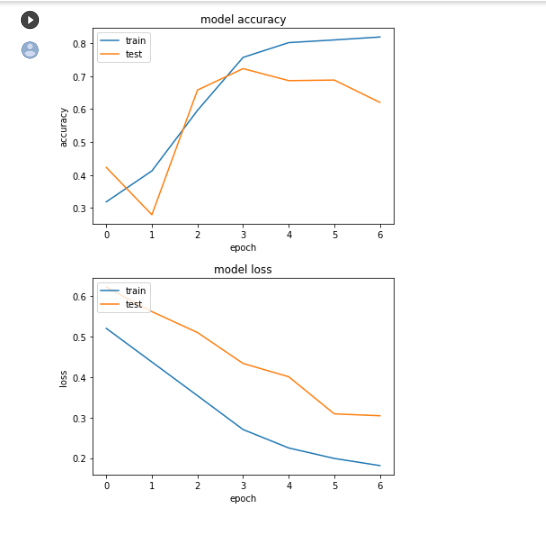
****

****

****

****

**Preliminary Results:** At the end of Increment 1, the following is the accuracy and the loss of our model.

****

**Project Management:**

Work completed:

1. Teja Sree Mandadi:

Implemented the lemmatization and stemming

Contribution: 25%

1. Akhila Kura:

Implemented the data preprocessing

Contribution: 25%

1. Panthangi Nikitha:

Implemented the word embedding

Contribution: 25%

1. Vineela Dachepally:

Implemented the model and fitting the model

Contribution: 25%

Work to be completed:  
• Description:

1. Need to test the model and find the accuracy. Represent it in visualization
2. Trying to improve the accuracy score
3. Trying to apply some techniques to boost the model

• Responsibility (Task, Person)

1. Testing the model – Vineela
2. Implement the techniques – Akhila, Nikitha
3. Data visualization - Tejasree

• Issues/Concerns:

The accuracy score of the model is like other models. So, need to improvise the score similarity

**References/Bibliography:**

<https://www.tensorflow.org/hub/tutorials/tf2_text_classification>

<https://www.machinelearningplus.com/nlp/gensim-tutorial/>

<https://www.analyticsvidhya.com/blog/2021/06/lstm-for-text-classification/#h2_6>

https://www.simplilearn.com/tutorials/deep-learning-tutorial/rnn