SUM SPCL 22-23

J Component Report

Technical Answers for Real World Problems

EEE 1901

GROUP MEMBERS

Ayush Gupta- 19BEE0092
Lakshay Singh- 19BEE0220
Roshan Kumar- 19BEE0236
V S Akshit- 19BEE0435

FACULTY: Dr. Raja Singh R

SLOT: TA1

Machine Learning based topic recognition system for researchers

1. ***Abstract:***

Our project's objective is to categorize any research paper by examining the publication abstract. Our method involves training a model which analyses merely the paper's abstract to determine the category using machine learning and text categorization. Lemmatization and the removal of missing values from a clean dataset will be used to construct the model. And when it comes time to classify the input text, deep learning techniques will be used. To complete this objective, sequential neural networks will indeed be employed.

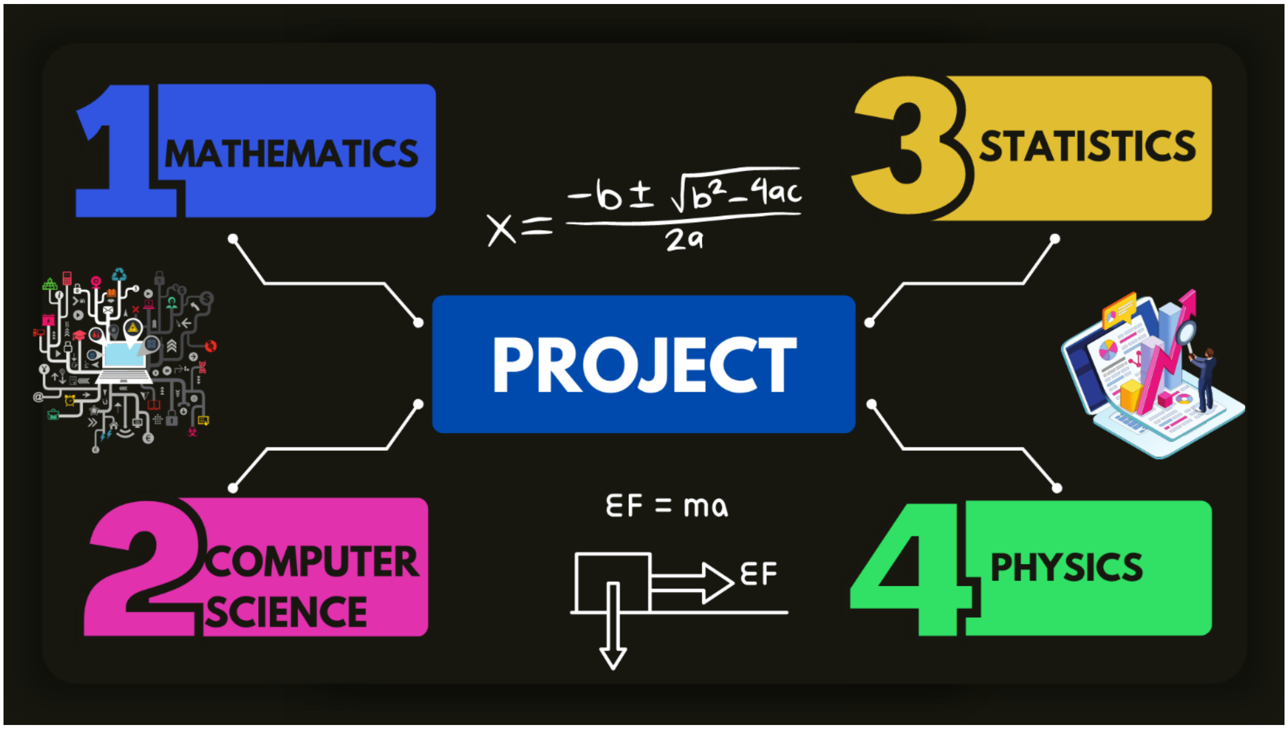
By analysing the research article's abstract, our project tries to identify the group in which a research article falls. Among the tags are those for mathematics, statistics, computers, physics, applications of partial differential equations, artificial intelligence, computation and language, computer vision and pattern matching, cosmology, and non-galactic astrophysics, as well as algorithms and data structures differential geometry, and astrophysics of the Earth and planets. Numerous topics are covered, such as fluid mechanics, information theory, astronomical instruments and techniques, machine learning and material science, methodology, and number theory. A scientific paper could have several tags. The subsequent four subjects are the sources again for research report abstracts:

- ***Computer Science***
- ***Mathematics***
- ***Physics***
- ***Statistics.***

Our This model would take the abstract of a research paper as input and return one of the four different categories to which that specific research paper falls under.

A webpage will be developed which would allow users to submit abstracts up to a certain word limit, and it would then generate the best-fitting subcategory it belongs to as well as a confidence score or accuracy rating.

One of the really significant uses for this project would be as a "Digital Library," where articles could be categorised into distinct and designated topics, making it easier for any user or researcher to find what they're looking for and improving the sense of professionalism of the web resources. Quite a few projects have explored text classification for research purposes, and this effort is one of the first that deploy sequential neural networks in this application.



2. Introduction

Accessibility to high quality content has greatly increased along with the general rise in globalisation levels brought on by the development of the internet. Since the materials needed to satisfy anyone's interest are easily accessible nowadays, the only significant obstacle standing in the way is that person's own curiosity.

It is essential for the creation of user-assistance tools because the beholder now has access to an unparalleled volume of information. With the aid of our initiative, we hope to meet this critical need by enabling users to identify the genre of any article or published work and decide the extent to which it corresponds to their specific areas of interest.

This initiative would also assist researchers and aspirants in locating the top publications or conventions they might participate in. Researchers frequently struggle with making decisions related to these issues, which is precisely where this study may help. The relevance of the possible applications provided by this initiative is further highlighted by the recent increase in the use of electronic information resources.

This report seeks to succinctly and analytically compile the advances achieved upon the project and, using a variety of methodologies, to clearly explain various parts of it.

Hence, this paper could act as a reference of the suggested architecture diagram, the dataset and the reasoning for using it, and the execution that has been carried out so far, including the data preprocessing.

3. Problem Statement:

As, technology has paved the way for high accessibility, with easy access to resources over the internet, the average person’s lifestyle has been getting busier at the same rate. As researchers look forward to reading and assessing articles of their choice, they often find it hard to identify relevant papers as per their interests.

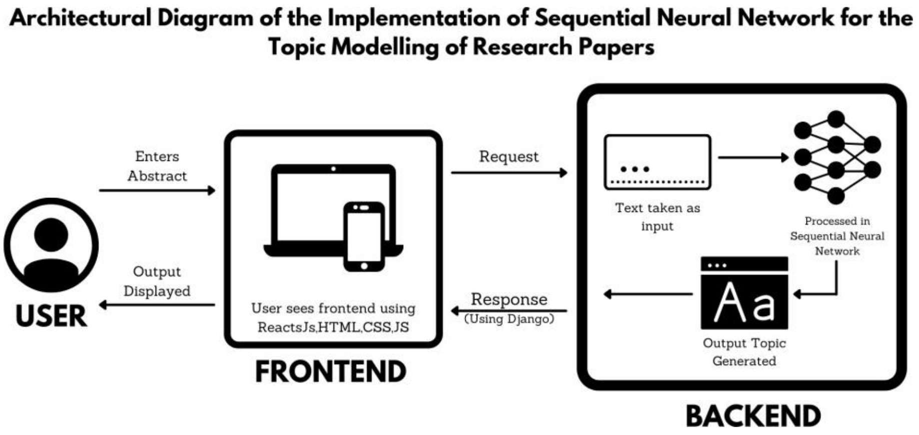
Similarly, paper authors often struggle to identify the ideal journal or publication for their work, as the article which they aim to publish covers a large number of parameters. This is where the concept of our project can act as an efficient gateway to save time and increase accessibility for various members of the research community and more.

The common person's living has been increasing grouchier at the same time that technology has laid the foundation for enhanced access, with simple access to resources over the internet. While analysts like reading and evaluating the articles of their choice, they frequently struggle to find the papers that are pertinent to their areas of interest.

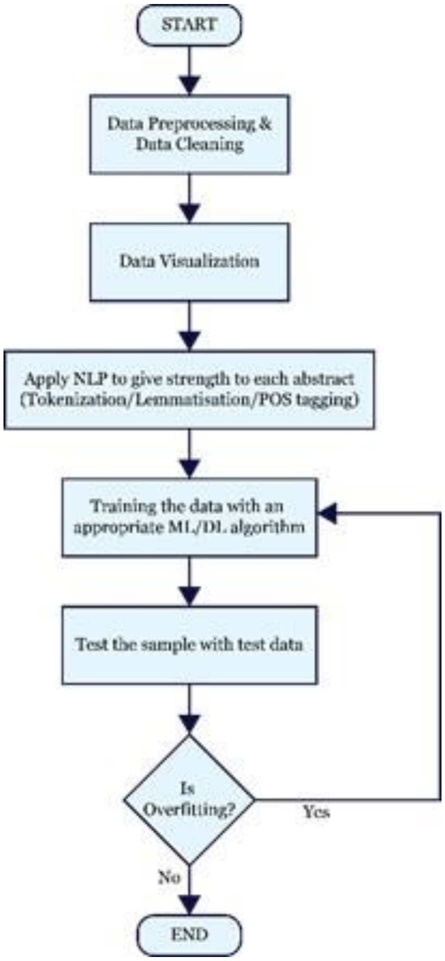
The article that authors want to publish spans a wide range of parameters, thus they frequently struggle to find the best magazine or publisher for their work. Here, the idea behind our work can serve as a useful entryway to cut down on time and promote access for a wide range of research community members and much more.

Information about a topic's word distributions before reading materials that have strong affinities for that topic to conduct a more thorough investigation.

3.1. Architecture Diagram


Architectural Diagram of the Implementation of Sequential Neural Network for the Topic Modelling of Research Papers
Enters Abstract
Output Displayed
USER
User sees frontend using ReactsJs,HTML,CSS,JS
FRONTEND
Request
Response
(Using Django)
Text taken as input
Processed in Sequential Neural Network
Aa
Output Topic Generated
BACKEND

3.2. Flow Diagram


START
Data Preprocessing & Data Cleaning
Data Visualization
Apply NLP to give strength to each abstract (Tokenization/Lemmatisation/POS tagging)
Training the data with an appropriate ML/DL algorithm
Test the sample with test data
Is Overfitting?
Yes
No
END

3.3. PseudoCode

```
BEGIN:
        READ Import Necessary Packages //As listed below
        READ: Import Train.csv and Test.csv  //using pandas

        //Cleaning Dataset
        DETERMINE Db ← Drop Columns 6 to 31 in Db
        DETERMINE Db ← Drop ID Column in Db
        SET blanks=[]
        FOR each column in Db
        IF Abstract Column is Blank
                ADD blanks ← blanks[i]
        END IF
END FOR
DISPLAY blanks
IF blanks is empty
        CONTINUE
ELSE
        DETERMINE Db ← Drop Columns with index i
END IF
SET Db[‘label] ← 0
FOR each row in Db
        IF Row[‘Mathematics’]==1 and Row[‘Other Columns’]==0
                ADD Db[‘label’] ← M
        ELSE IF Row[‘Computer Science’’]==1 and Row[‘Other Columns’]==0
                ADD Db[‘label’] ← C
        ELSE IF Row[‘Physics’’]==1 and Row[‘Other Columns’]==0
                ADD Db[‘label’] ← P
        ELSE
                ADD Db[‘label’] ← S
        END IF
END FOR

//Tokenization
DETERMINE tokens ← nltk.word_tokenizer(row[‘ABSTRACT’])
FOR w for w in tokens
        IF w.isalpha()
                ADD token_words ← token_words[w]
        END IF
END FOR
DETERMINE db[‘abstract’] ← db.apply[identify_tokens]
```

```
//Stemming
READ PorterStemmer from nltk.stem()
COMPUTE stemming ← PorterStemmer
COMPUTE my_list ← row[‘ABSTRACT’]
FOR stemming.stem(word) for word in my_list
    ADD stemmed_list ← stemmed_list[word]
END FOR
DETERMINE db[‘stemmed_words’] ← db.apply[stem_list]

//Removing Stop Words
READ stopwords from nltk.corpus
COMPUTE stops ← stopwords.words[“english”]
COMPUTE my_list ← row[‘stemmed_words’]
FOR w for word in my_list
    IF not w in stops
        ADD meaningful_words ← meaningful_wordst[w]
    END IF
END FOR
DETERMINE db[‘stem_meaningful’’] ← db.apply[remove_stops]

//Brief Pseudocode for Proposed Implementation ahead

//Using Created Tokens to create a new vocabulary of all words
READ tokenizer and pad_sequences from keras
COMPUTE vocab_size using Tokenizer

//Perform One-Hot Encoding for each of obtained tokens using the vocabulary list

READ Sequential from Tensorflow
DETERMINE Sequential model with parameters listed below
```

```
//Use the keras library provided by tensorflow to build the sequential model and optimize the parameters of Embedding, Flatten and Dense
//Use the encoded training data such that the sequential model can train itself
//Implement the word embeddings approach and tune the dense layer weights
DETERMINE Model Summary and Fit the Model
//Optimize all the parameters to optimize the model and its accuracy
DETERMINE F-measure of the Model
//Test the accuracy of the model using F-measure and ensure low presence of false positives and false negatives
END
```

4. ***Experiments and Results***

Dataset Used and Requirement Analysis:

Through Kaggle, we acquired this dataset. It takes the shape of an excel spreadsheet, including one column having the abstracts of research articles and the other the tags. Then, we will pre-process and sanitize the data (removing the missing data, etc.). The strength of each abstract will thereafter be determined using a variety of NLP techniques, such as tokenization, lemmatization, etc., applied in the Abstracts column. Furthermore, depending on the tags and categories, we will classify the abstract using Deep Learning and Machine Learning techniques.

I. Functional requirement:

This model uses an entry of an abstract from a research paper to determine which area the research article falls under.

II. Software required:

1. Google Collab
2. Jupyter notebook
3. VS Code

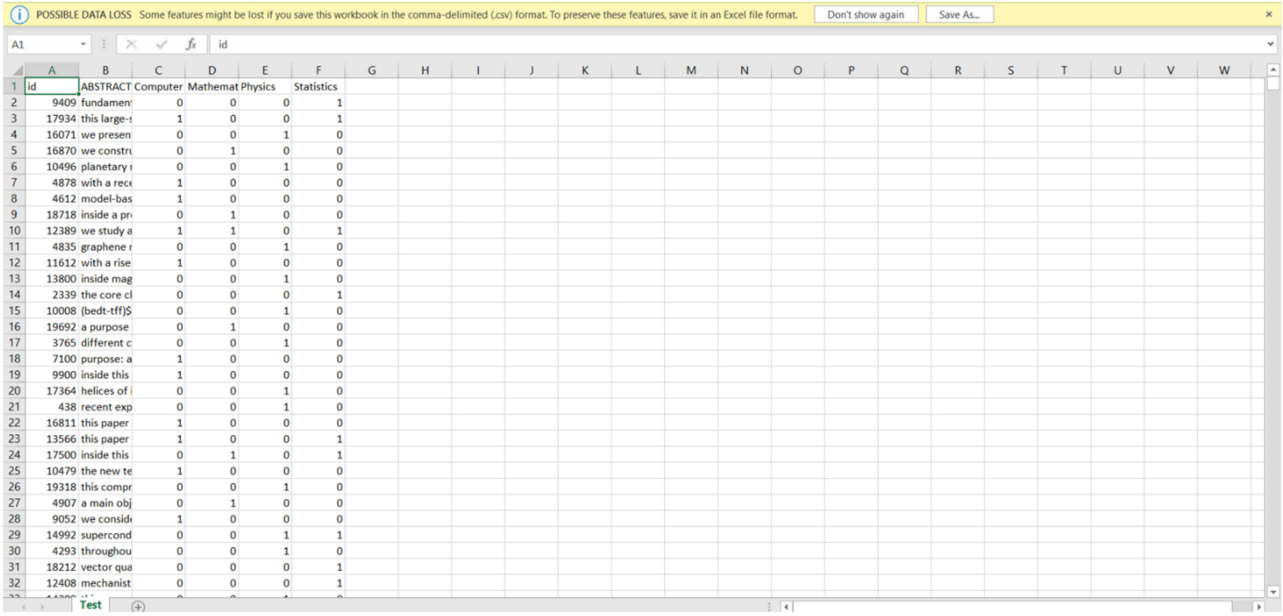
III. Packages required:

1. Nltk

2. Flask
3. Matplotlib
4. Pandas
5. Numpy
6. Seaborn
7. Scikit learn
8. Tensorflow
9. Keras
10. Spacey

4.1 ***Sample of the Datasets***

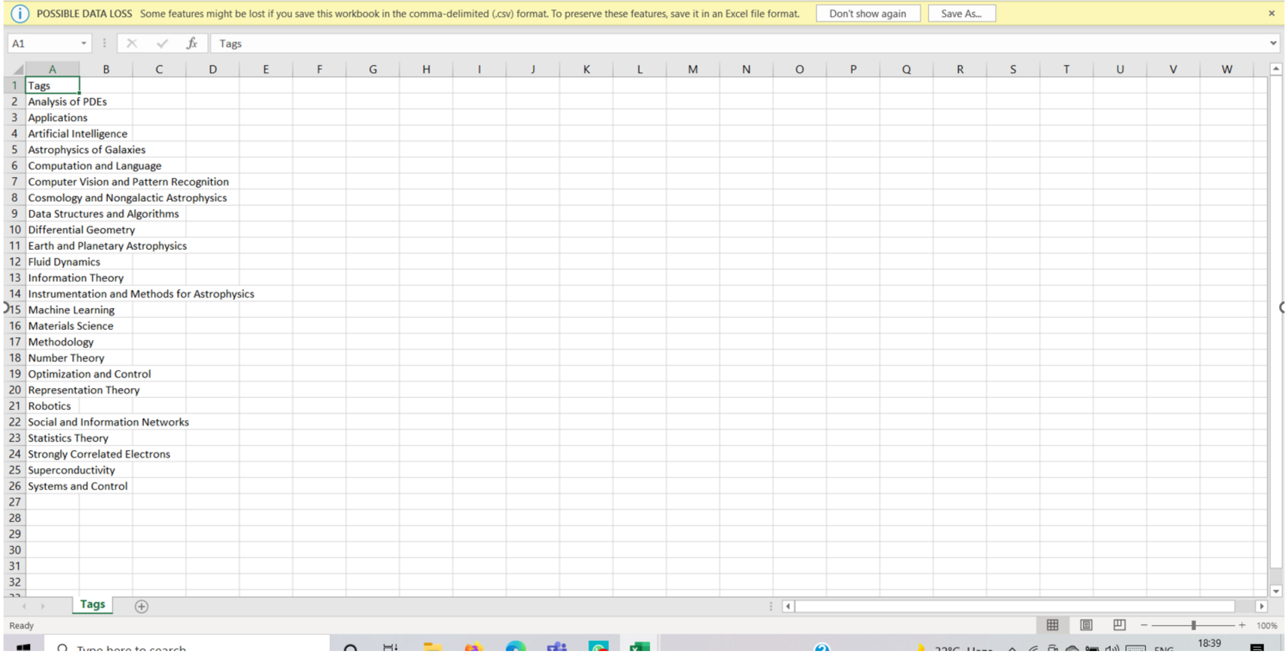
Test.csv: *Data collected used to evaluate the trained model against the data*



POSSIBLE DATA LOSS Some features might be lost if you save this workbook in the comma-delimited (.csv) format. To preserve these features, save it in an Excel file format.

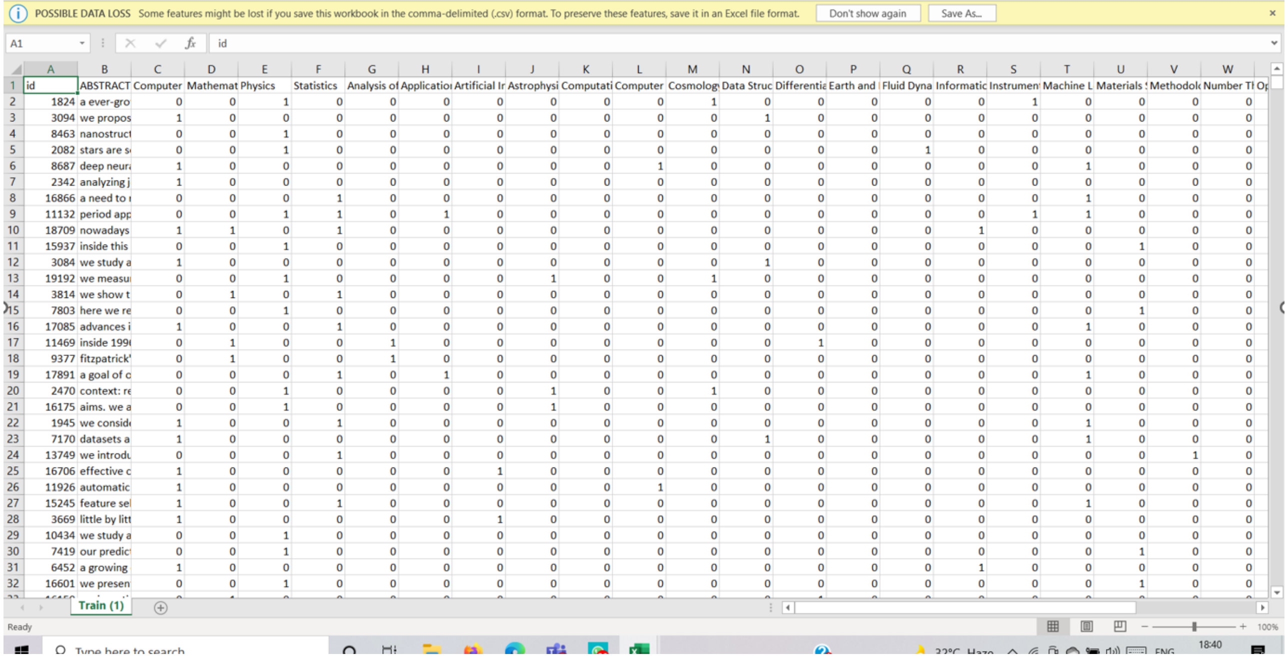
id	ABSTRACT	Computer	Mathemat	Physics	Statistics
9409	fundament	0	0	0	1
17934	this large-s	1	0	0	1
16071	we presen	0	0	1	0
16870	we constru	0	1	0	0
10496	planetary r	0	0	1	0
4878	with a rece	1	0	0	0
4612	model-bas	1	0	0	0
18718	inside a pr	0	1	0	0
12389	we study a	1	1	0	1
4835	graphene r	0	0	1	0
11612	with a rise	1	0	0	0
13800	inside mag	0	0	1	0
2339	the core cl	0	0	0	1
10008	(bedt-tff)$	0	0	1	0
19692	a purpose	0	1	0	0
3765	different c	0	0	1	0
7100	purpose: a	1	0	0	0
9900	inside this	1	0	0	0
17364	helices of i	0	0	1	0
438	recent exp	0	0	1	0
16811	this paper	1	0	0	0
13566	this paper	1	0	0	1
17500	inside this	0	1	0	1
10479	the new te	1	0	0	0
19318	this compr	0	0	1	0
4907	a main obj	0	1	0	0
9052	we conside	1	0	0	0
14992	supercond	0	0	1	1
4293	throughou	0	0	1	0
18212	vector qua	0	0	0	1
12408	mechanist	0	0	0	1

<u>Tags.csv</u>: *For displaying the planned and available sample tags*


POSSIBLE DATA LOSS Some features might be lost if you save this workbook in the comma-delimited (.csv) format. To preserve these features, save it in an Excel file format.
Don't show again
Save As...
A1
Tags
Ready
100%
18:39

	A
1	Tags
2	Analysis of PDEs
3	Applications
4	Artificial Intelligence
5	Astrophysics of Galaxies
6	Computation and Language
7	Computer Vision and Pattern Recognition
8	Cosmology and Nongalactic Astrophysics
9	Data Structures and Algorithms
10	Differential Geometry
11	Earth and Planetary Astrophysics
12	Fluid Dynamics
13	Information Theory
14	Instrumentation and Methods for Astrophysics
15	Machine Learning
16	Materials Science
17	Methodology
18	Number Theory
19	Optimization and Control
20	Representation Theory
21	Robotics
22	Social and Information Networks
23	Statistics Theory
24	Strongly Correlated Electrons
25	Superconductivity
26	Systems and Control

<u>Train.csv:</u> *Dataset used for training the model. The dataset underwent extensive cleaning methods before being used*


POSSIBLE DATA LOSS Some features might be lost if you save this workbook in the comma-delimited (.csv) format. To preserve these features, save it in an Excel file format.
Don't show again
Save As...
A1
id
Train (1)
Ready
100%
18:40

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
1	id	ABSTRACT	Computer	Mathemat	Physics	Statistics	Analysis of	Applicatio	Artificial Ir	Astrophysi	Computati	Computer	Cosmolog	Data Struc	Differentia	Earth and	Fluid Dyna	Informatic	Instrumen	Machine L	Materials !	Methodolc	Number Tl
2	1824	a ever-gro	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0
3	3094	we propos	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
4	8463	nanostruct	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	2082	stars are s	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
6	8687	deep neur	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0
7	2342	analyzing j	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8	16866	a need to	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
9	11132	period app	0	0	1	1	0	1	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0
10	18709	nowadays	1	1	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
11	15937	inside this	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
12	3084	we study a	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
13	19192	we measu	0	0	1	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0
14	3814	we show t	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
15	7803	here we re	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
16	17085	advances i	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
17	11469	inside 199	0	1	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
18	9377	fitzpatrick'	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
19	17891	a goal of c	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
20	2470	context: re	0	0	1	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0
21	16175	aims. we a	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
22	1945	we conside	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
23	7170	datasets a	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0
24	13749	we introdu	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
25	16706	effective c	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
26	11926	automatic	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
27	15245	feature sel	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
28	3669	little by litt	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
29	10434	we study a	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
30	7419	our predic	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
31	6452	a growing	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
32	16601	we presen	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0

4.1.1 Explain methodology with the dataset

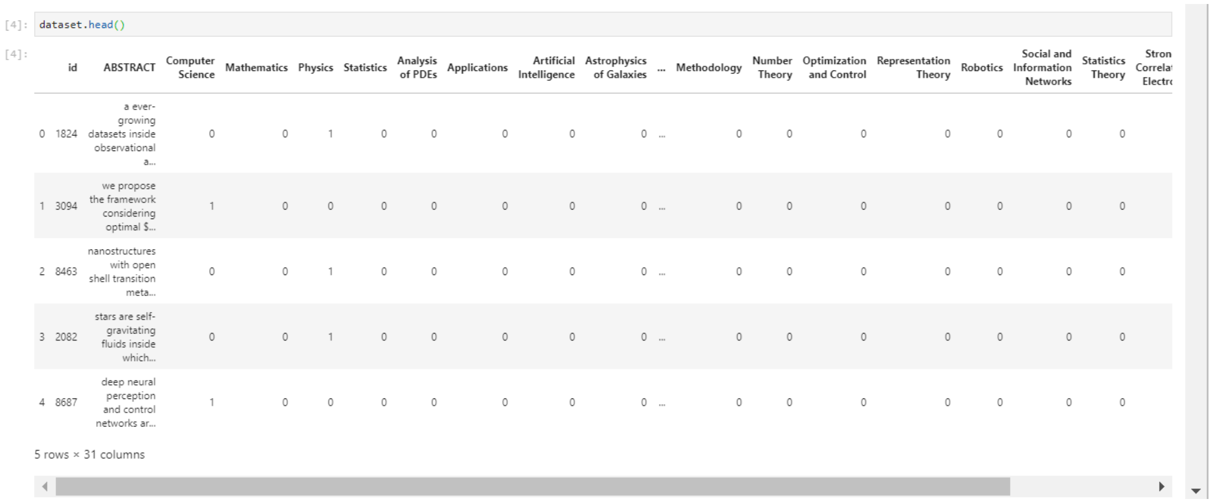
- Use a variety of data visualisation tools to choose the classification model or neural network that would provide the best fit for our case.
- Make a bag with each term in our dataset.
- Check to see how many times a specific word appears in our sentence.
- Make a sequential neural network, then train it with the given dataset.
- Classify sentences (abstracts) using the trained model into a certain field of study, including such computer science, mathematics, physics, or statistics, using the Bag of Words and Relevant method.

4.2 Output

Cleaning the Dataset

Utilizing the methods described in the pseudocode detailed above, the training Dataset has been cleaned.

Displaying the head, or the first few rows of the dataset:



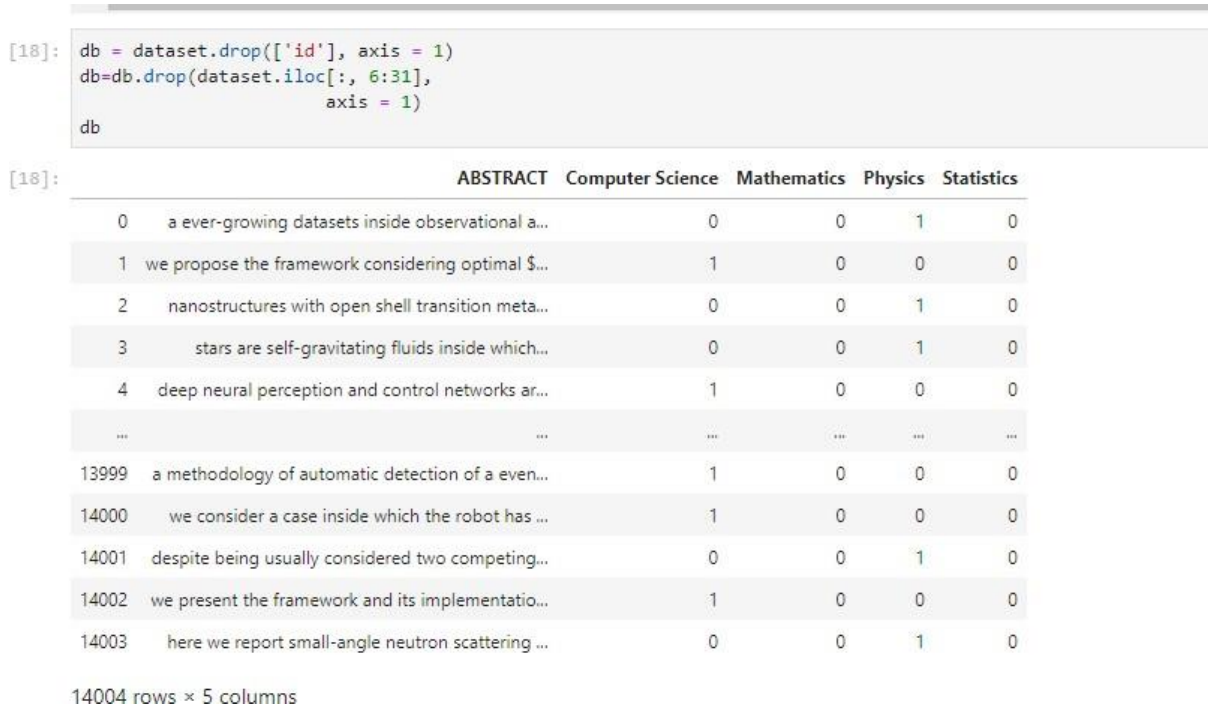
```
[4]: dataset.head()
```

[4]:

	id	ABSTRACT	Computer Science	Mathematics	Physics	Statistics	Analysis of PDEs	Applications	Artificial Intelligence	Astrophysics of Galaxies	...	Methodology	Number Theory	Optimization and Control	Representation Theory	Robotics	Social and Information Networks	Statistics Theory	Stron Correlat Electr
0	1824	a ever-growing datasets inside observational a...	0	0	1	0	0	0	0	0	...	0	0	0	0	0	0	0	
1	3094	we propose the framework considering optimal $...	1	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	
2	8463	nanostructures with open shell transition meta...	0	0	1	0	0	0	0	0	...	0	0	0	0	0	0	0	
3	2082	stars are self-gravitating fluids inside which...	0	0	1	0	0	0	0	0	...	0	0	0	0	0	0	0	
4	8687	deep neural perception and control networks ar...	1	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	

5 rows × 31 columns

Dropping the Columns to Make Room for four (4) sections for Analyzing:



[18]:

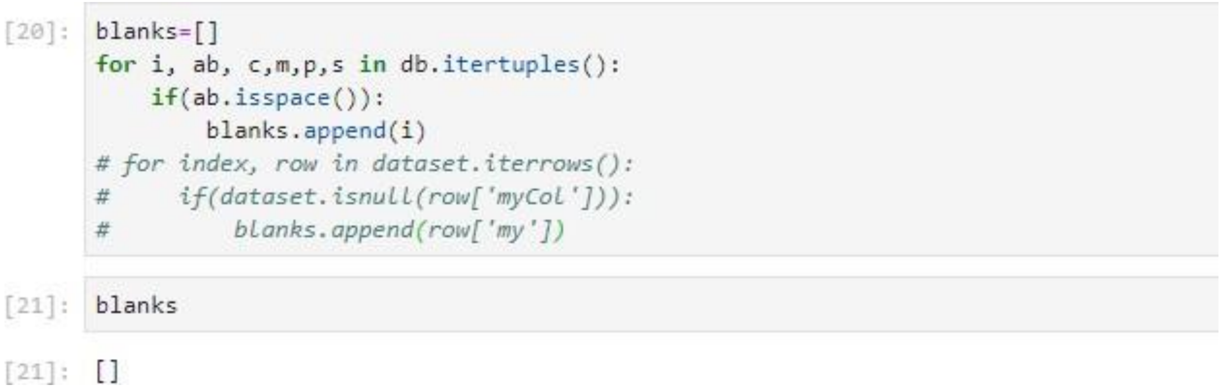
```
db = dataset.drop(['id'], axis = 1)
db=db.drop(dataset.iloc[:, 6:31],
                        axis = 1)
db
```

[18]:

	ABSTRACT	Computer Science	Mathematics	Physics	Statistics
0	a ever-growing datasets inside observational a...	0	0	1	0
1	we propose the framework considering optimal $...	1	0	0	0
2	nanostructures with open shell transition meta...	0	0	1	0
3	stars are self-gravitating fluids inside which...	0	0	1	0
4	deep neural perception and control networks ar...	1	0	0	0
...	...	...	...	...	...
13999	a methodology of automatic detection of a even...	1	0	0	0
14000	we consider a case inside which the robot has ...	1	0	0	0
14001	despite being usually considered two competing...	0	0	1	0
14002	we present the framework and its implementatio...	1	0	0	0
14003	here we report small-angle neutron scattering ...	0	0	1	0

14004 rows × 5 columns

Using a blanks [] array, determining whether the ABSTRACT column contains any zeros.



[20]:

```
blanks=[]
for i, ab, c,m,p,s in db.itertuples():
    if(ab.isspace()):
        blanks.append(i)
# for index, row in dataset.iterrows():
#     if(dataset.isnull(row['myCol'])):
#         blanks.append(row['my'])
```

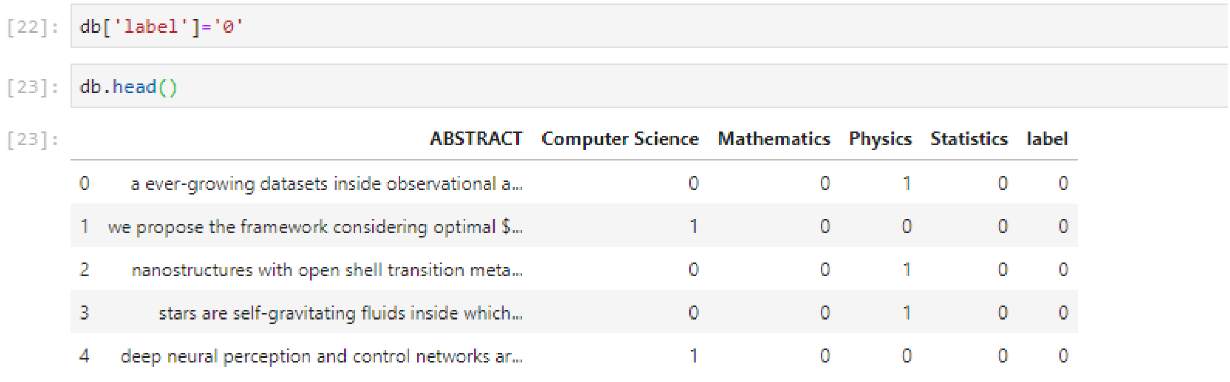
[21]:

```
blanks
```

[21]:

```
[]
```

Introducing a label column and setting its value to "0"



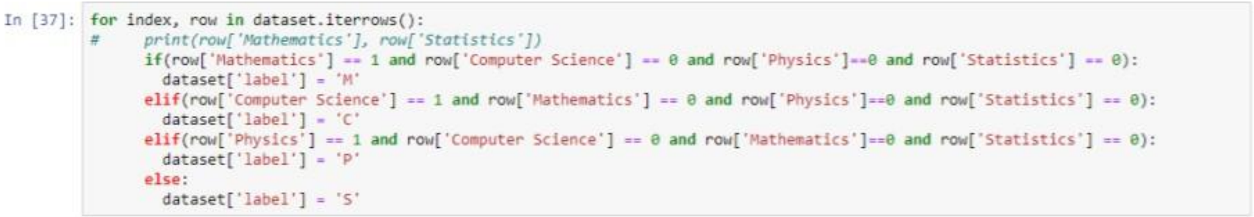
```
[22]: db['label']='0'
```

```
[23]: db.head()
```

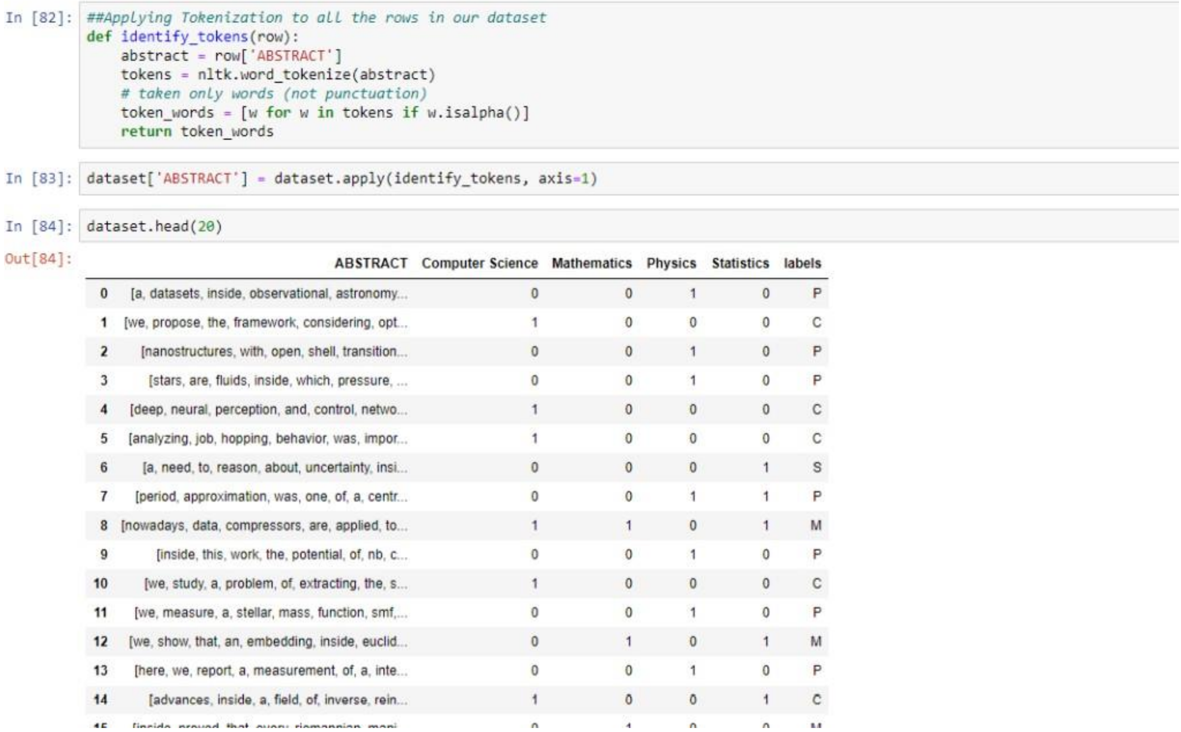
[23]:

	ABSTRACT	Computer Science	Mathematics	Physics	Statistics	label
0	a ever-growing datasets inside observational a...	0	0	1	0	0
1	we propose the framework considering optimal $...	1	0	0	0	0
2	nanostructures with open shell transition meta...	0	0	1	0	0
3	stars are self-gravitating fluids inside which...	0	0	1	0	0
4	deep neural perception and control networks ar...	1	0	0	0	0

Tokenization Implementation



```
In [37]: for index, row in dataset.iterrows():
         #     print(row['Mathematics'], row['Statistics'])
               if(row['Mathematics'] == 1 and row['Computer Science'] == 0 and row['Physics']==0 and row['Statistics'] == 0):
                 dataset['label'] = 'M'
               elif(row['Computer Science'] == 1 and row['Mathematics'] == 0 and row['Physics']==0 and row['Statistics'] == 0):
                 dataset['label'] = 'C'
               elif(row['Physics'] == 1 and row['Computer Science'] == 0 and row['Mathematics']==0 and row['Statistics'] == 0):
                 dataset['label'] = 'P'
               else:
                 dataset['label'] = 'S'
```



```
In [82]: ##Applying Tokenization to all the rows in our dataset
         def identify_tokens(row):
             abstract = row['ABSTRACT']
             tokens = nltk.word_tokenize(abstract)
             # taken only words (not punctuation)
             token_words = [w for w in tokens if w.isalpha()]
             return token_words
```

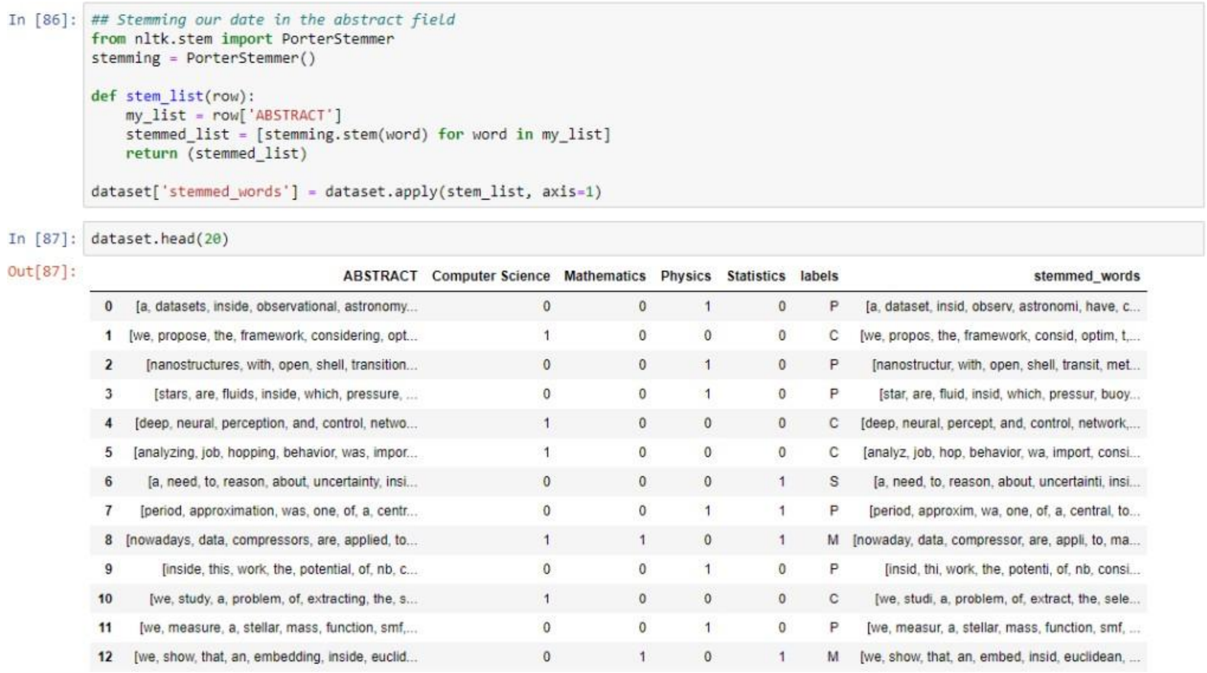
```
In [83]: dataset['ABSTRACT'] = dataset.apply(identify_tokens, axis=1)
```

```
In [84]: dataset.head(20)
```

Out[84]:

	ABSTRACT	Computer Science	Mathematics	Physics	Statistics	labels
0	[a, datasets, inside, observational, astronomy...	0	0	1	0	P
1	[we, propose, the, framework, considering, opt...	1	0	0	0	C
2	[nanostructures, with, open, shell, transition...	0	0	1	0	P
3	[stars, are, fluids, inside, which, pressure, ...	0	0	1	0	P
4	[deep, neural, perception, and, control, netwo...	1	0	0	0	C
5	[analyzing, job, hopping, behavior, was, impor...	1	0	0	0	C
6	[a, need, to, reason, about, uncertainty, insi...	0	0	0	1	S
7	[period, approximation, was, one, of, a, centr...	0	0	1	1	P
8	[nowadays, data, compressors, are, applied, to...	1	1	0	1	M
9	[inside, this, work, the, potential, of, nb, c...	0	0	1	0	P
10	[we, study, a, problem, of, extracting, the, s...	1	0	0	0	C
11	[we, measure, a, stellar, mass, function, smf,...	0	0	1	0	P
12	[we, show, that, an, embedding, inside, euclid...	0	1	0	1	M
13	[here, we, report, a, measurement, of, a, inte...	0	0	1	0	P
14	[advances, inside, a, field, of, inverse, rein...	1	0	0	1	C
15	[inside, proved, that, every, riemannian, mani...	0	1	0	0	M

Implementing Stemming



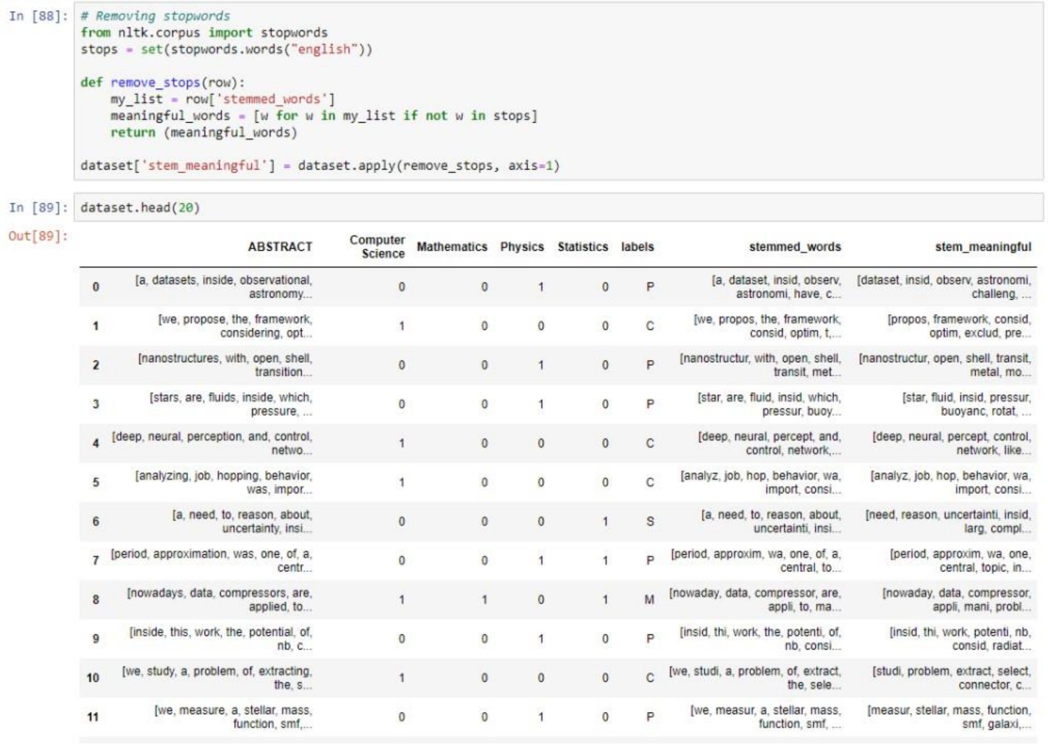
```
In [86]: ## Stemming our date in the abstract field
         from nltk.stem import PorterStemmer
         stemming = PorterStemmer()

         def stem_list(row):
             my_list = row['ABSTRACT']
             stemmed_list = [stemming.stem(word) for word in my_list]
             return (stemmed_list)

         dataset['stemmed_words'] = dataset.apply(stem_list, axis=1)
```

```
In [87]: dataset.head(20)
```

Out[87]:

	ABSTRACT	Computer Science	Mathematics	Physics	Statistics	labels	stemmed_words
0	[a, datasets, inside, observational, astronomy...	0	0	1	0	P	[a, dataset, insid, observ, astronomi, have, c...
1	[we, propose, the, framework, considering, opt...	1	0	0	0	C	[we, propos, the, framework, consid, optim, t,...
2	[nanostructures, with, open, shell, transition...	0	0	1	0	P	[nanostructur, with, open, shell, transit, met...
3	[stars, are, fluids, inside, which, pressure, ...	0	0	1	0	P	[star, are, fluid, insid, which, pressur, buoy...
4	[deep, neural, perception, and, control, netwo...	1	0	0	0	C	[deep, neural, percept, and, control, network,...
5	[analyzing, job, hopping, behavior, was, impor...	1	0	0	0	C	[analyz, job, hop, behavior, wa, import, consi...
6	[a, need, to, reason, about, uncertainty, insi...	0	0	0	1	S	[a, need, to, reason, about, uncertainti, insi...
7	[period, approximation, was, one, of, a, centr...	0	0	1	1	P	[period, approxim, wa, one, of, a, central, to...
8	[nowadays, data, compressors, are, applied, to...	1	1	0	1	M	[nowaday, data, compressor, are, appli, to, ma...
9	[inside, this, work, the, potential, of, nb, c...	0	0	1	0	P	[insid, thi, work, the, potenti, of, nb, consi...
10	[we, study, a, problem, of, extracting, the, s...	1	0	0	0	C	[we, studi, a, problem, of, extract, the, sele...
11	[we, measure, a, stellar, mass, function, smf,...	0	0	1	0	P	[we, measur, a, stellar, mass, function, smf, ...
12	[we, show, that, an, embedding, inside, euclid...	0	1	0	1	M	[we, show, that, an, embed, insid, euclidean, ...

Removing the Stop Words



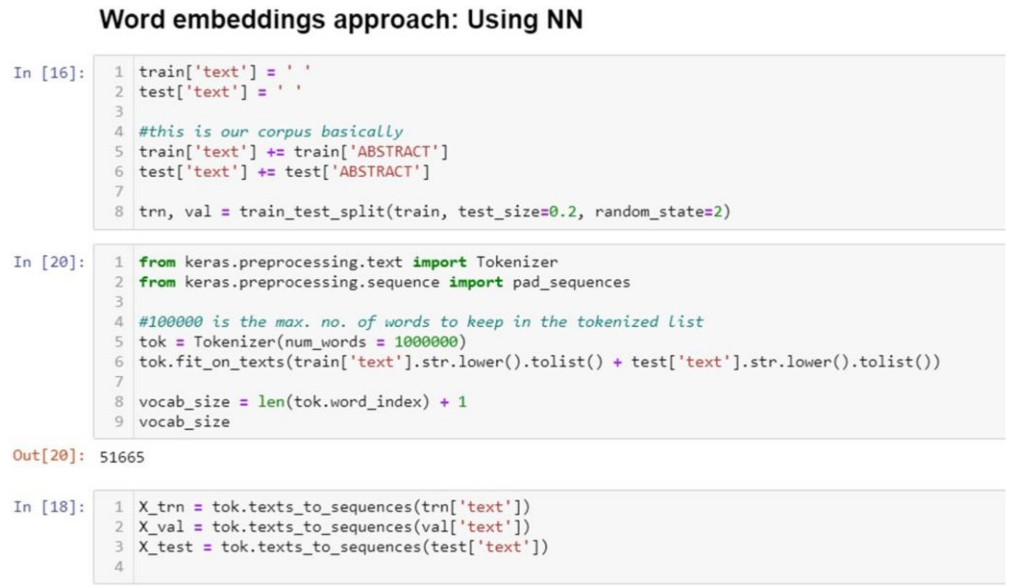
```
In [88]: # Removing stopwords
         from nltk.corpus import stopwords
         stops = set(stopwords.words("english"))

         def remove_stops(row):
             my_list = row['stemmed_words']
             meaningful_words = [w for w in my_list if not w in stops]
             return (meaningful_words)

         dataset['stem_meaningful'] = dataset.apply(remove_stops, axis=1)
```

```
In [89]: dataset.head(20)
```

Out[89]:

	ABSTRACT	Computer Science	Mathematics	Physics	Statistics	labels	stemmed_words	stem_meaningful
0	[a, datasets, inside, observational, astronomy...	0	0	1	0	P	[a, dataset, insid, observ, astronomi, have, c...	[dataset, insid, observ, astronomi, challeng, ...
1	[we, propose, the, framework, considering, opt...	1	0	0	0	C	[we, propos, the, framework, consid, optim, t,...	[propos, framework, consid, optim, exclud, pre...
2	[nanostructures, with, open, shell, transition...	0	0	1	0	P	[nanostructur, with, open, shell, transit, met...	[nanostructur, open, shell, transit, metal, mo...
3	[stars, are, fluids, inside, which, pressure, ...	0	0	1	0	P	[star, are, fluid, insid, which, pressur, buoy...	[star, fluid, insid, pressur, buoyanc, rotat, ...
4	[deep, neural, perception, and, control, netwo...	1	0	0	0	C	[deep, neural, percept, and, control, network,...	[deep, neural, percept, control, network, like...
5	[analyzing, job, hopping, behavior, was, impor...	1	0	0	0	C	[analyz, job, hop, behavior, wa, import, consi...	[analyz, job, hop, behavior, wa, import, consi...
6	[a, need, to, reason, about, uncertainty, insi...	0	0	0	1	S	[a, need, to, reason, about, uncertainti, insi...	[need, reason, uncertainti, insid, larg, compl...
7	[period, approximation, was, one, of, a, centr...	0	0	1	1	P	[period, approxim, wa, one, of, a, central, to...	[period, approxim, wa, one, central, topic, in...
8	[nowadays, data, compressors, are, applied, to...	1	1	0	1	M	[nowaday, data, compressor, are, appli, to, ma...	[nowaday, data, compressor, appli, mani, probl...
9	[inside, this, work, the, potential, of, nb, c...	0	0	1	0	P	[insid, thi, work, the, potenti, of, nb, consi...	[insid, thi, work, potenti, nb, consid, radiat...
10	[we, study, a, problem, of, extracting, the, s...	1	0	0	0	C	[we, studi, a, problem, of, extract, the, sele...	[studi, problem, extract, select, connector, c...
11	[we, measure, a, stellar, mass, function, smf,...	0	0	1	0	P	[we, measur, a, stellar, mass, function, smf, ...	[measur, stellar, mass, function, smf, galaxi,...

Sequential Neural Network



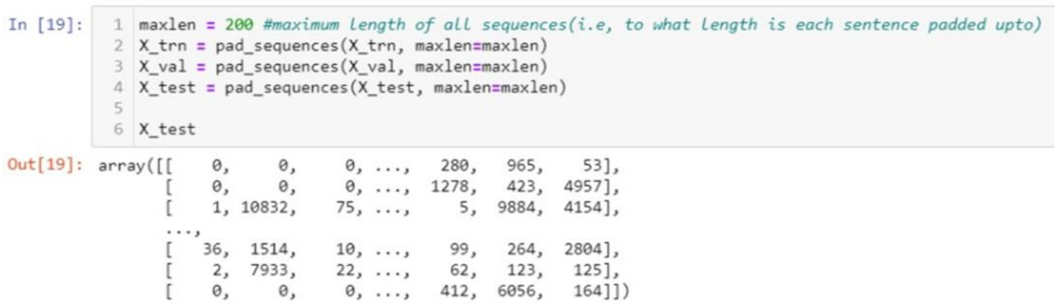
Word embeddings approach: Using NN

```
In [16]: train['text'] = ' '
         test['text'] = ' '

         #this is our corpus basically
         train['text'] += train['ABSTRACT']
         test['text'] += test['ABSTRACT']

         trn, val = train_test_split(train, test_size=0.2, random_state=2)
```

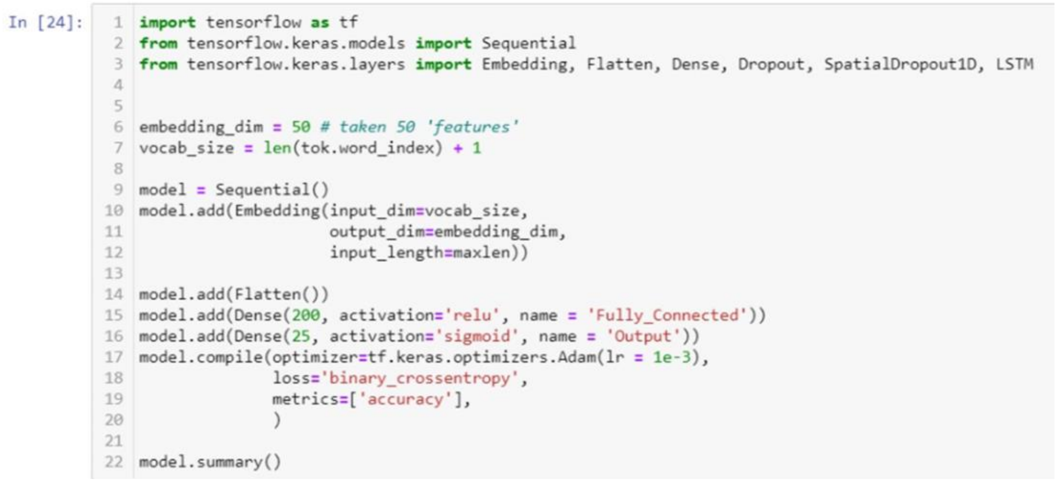
```
In [20]: from keras.preprocessing.text import Tokenizer
         from keras.preprocessing.sequence import pad_sequences

         #100000 is the max. no. of words to keep in the tokenized list
         tok = Tokenizer(num_words = 1000000)
         tok.fit_on_texts(train['text'].str.lower().tolist() + test['text'].str.lower().tolist())

         vocab_size = len(tok.word_index) + 1
         vocab_size

Out[20]: 51665
```

```
In [18]: X_trn = tok.texts_to_sequences(trn['text'])
         X_val = tok.texts_to_sequences(val['text'])
         X_test = tok.texts_to_sequences(test['text'])
```

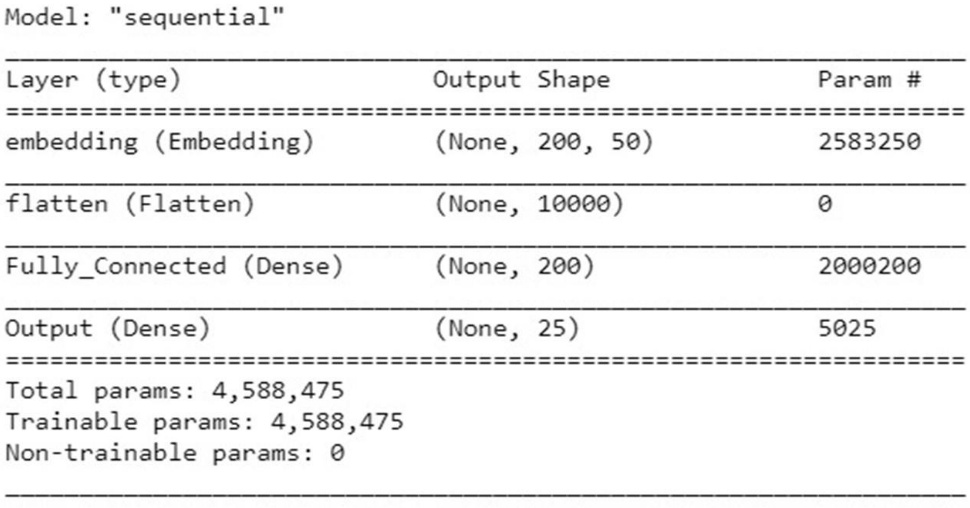
Embedding the model and using back propagation to learn the model



```
In [19]: maxlen = 200 #maximum length of all sequences(i.e, to what length is each sentence padded upto)
         X_trn = pad_sequences(X_trn, maxlen=maxlen)
         X_val = pad_sequences(X_val, maxlen=maxlen)
         X_test = pad_sequences(X_test, maxlen=maxlen)

         X_test

Out[19]: array([[    0,     0,     0, ...,   280,   965,    53],
                [    0,     0,     0, ...,  1278,   423,  4957],
                [    1, 10832,    75, ...,     5,  9884,  4154],
                ...,
                [   36,  1514,    10, ...,    99,   264,  2804],
                [    2,  7933,    22, ...,    62,   123,   125],
                [    0,     0,     0, ...,   412,  6056,   164]])
```

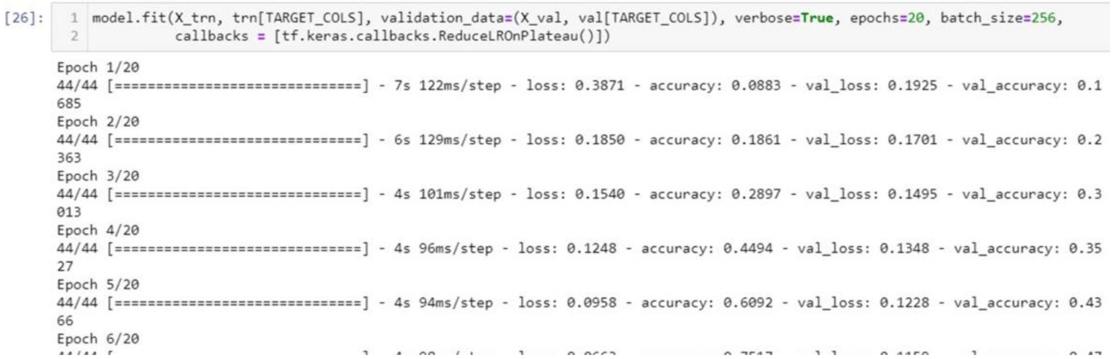


```
In [24]: import tensorflow as tf
         from tensorflow.keras.models import Sequential
         from tensorflow.keras.layers import Embedding, Flatten, Dense, Dropout, SpatialDropout1D, LSTM


         embedding_dim = 50 # taken 50 'features'
         vocab_size = len(tok.word_index) + 1

         model = Sequential()
         model.add(Embedding(input_dim=vocab_size,
                             output_dim=embedding_dim,
                             input_length=maxlen))

         model.add(Flatten())
         model.add(Dense(200, activation='relu', name = 'Fully_Connected'))
         model.add(Dense(25, activation='sigmoid', name = 'Output'))
         model.compile(optimizer=tf.keras.optimizers.Adam(lr = 1e-3),
                       loss='binary_crossentropy',
                       metrics=['accuracy'],
                       )

         model.summary()
```



```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
embedding (Embedding)        (None, 200, 50)           2583250
_________________________________________________________________
flatten (Flatten)            (None, 10000)             0
_________________________________________________________________
Fully_Connected (Dense)      (None, 200)               2000200
_________________________________________________________________
Output (Dense)               (None, 25)                5025
=================================================================
Total params: 4,588,475
Trainable params: 4,588,475
Non-trainable params: 0
_________________________________________________________________
```

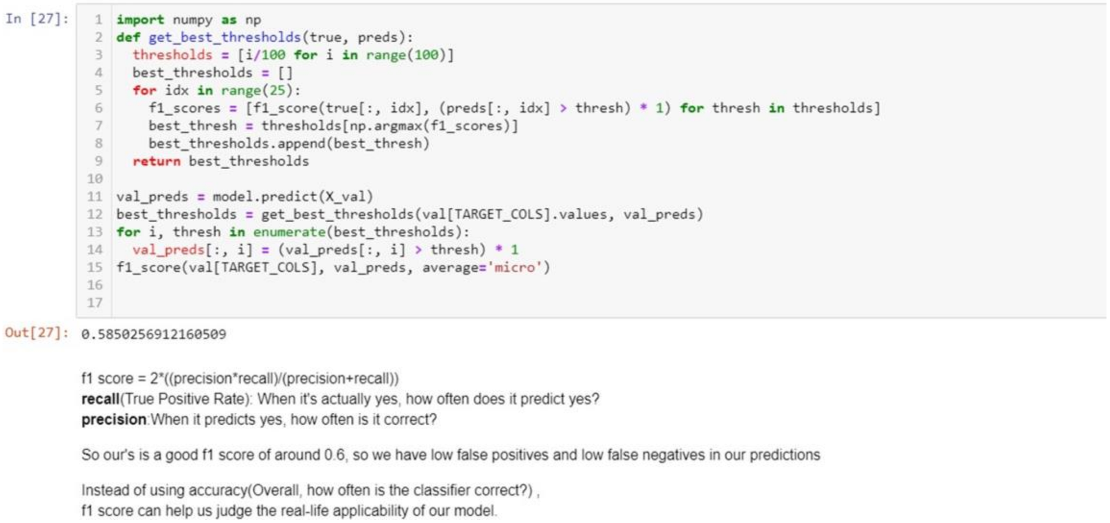
Fitting the model



```
[26]:  model.fit(X_trn, trn[TARGET_COLS], validation_data=(X_val, val[TARGET_COLS]), verbose=True, epochs=20, batch_size=256,
                 callbacks = [tf.keras.callbacks.ReduceLROnPlateau()])
```

```
Epoch 1/20
44/44 [==============================] - 7s 122ms/step - loss: 0.3871 - accuracy: 0.0883 - val_loss: 0.1925 - val_accuracy: 0.1
685
Epoch 2/20
44/44 [==============================] - 6s 129ms/step - loss: 0.1850 - accuracy: 0.1861 - val_loss: 0.1701 - val_accuracy: 0.2
363
Epoch 3/20
44/44 [==============================] - 4s 101ms/step - loss: 0.1540 - accuracy: 0.2897 - val_loss: 0.1495 - val_accuracy: 0.3
013
Epoch 4/20
44/44 [==============================] - 4s 96ms/step - loss: 0.1248 - accuracy: 0.4494 - val_loss: 0.1348 - val_accuracy: 0.35
27
Epoch 5/20
44/44 [==============================] - 4s 94ms/step - loss: 0.0958 - accuracy: 0.6092 - val_loss: 0.1228 - val_accuracy: 0.43
66
Epoch 6/20
```

Calculating the precision,recall and F1 values



```
In [27]:
import numpy as np
def get_best_thresholds(true, preds):
  thresholds = [i/100 for i in range(100)]
  best_thresholds = []
  for idx in range(25):
    f1_scores = [f1_score(true[:, idx], (preds[:, idx] > thresh) * 1) for thresh in thresholds]
    best_thresh = thresholds[np.argmax(f1_scores)]
    best_thresholds.append(best_thresh)
  return best_thresholds

val_preds = model.predict(X_val)
best_thresholds = get_best_thresholds(val[TARGET_COLS].values, val_preds)
for i, thresh in enumerate(best_thresholds):
  val_preds[:, i] = (val_preds[:, i] > thresh) * 1
f1_score(val[TARGET_COLS], val_preds, average='micro')
```

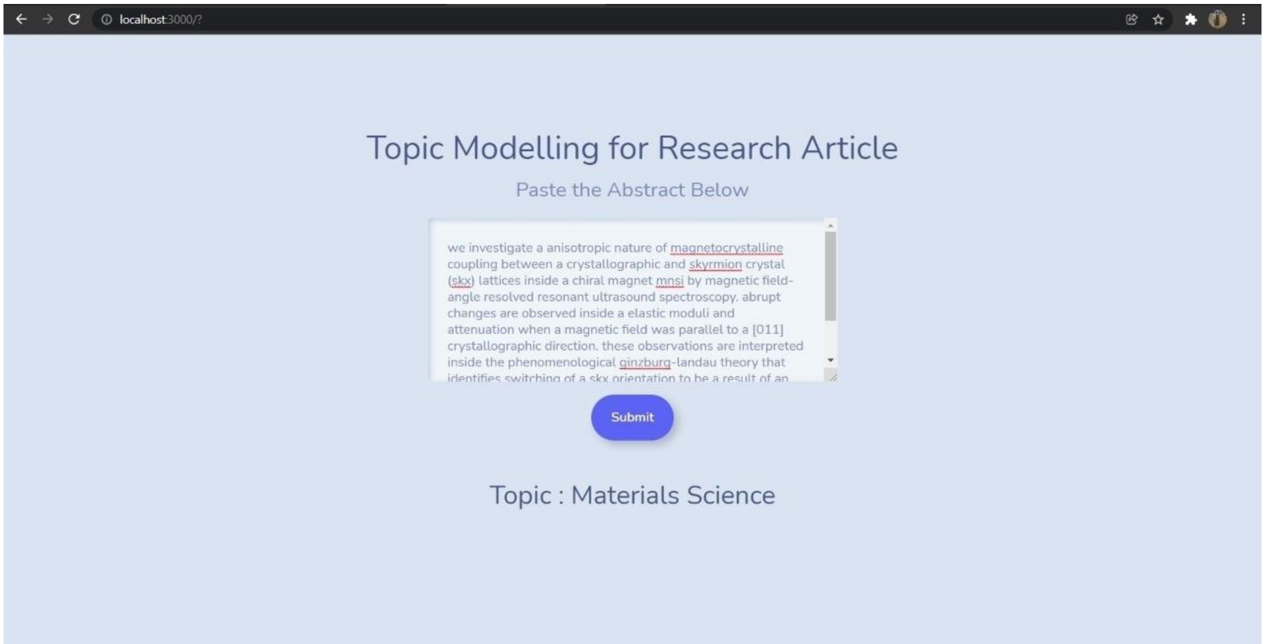
```
Out[27]: 0.5850256912160509
```

f1 score = 2*((precision*recall)/(precision+recall))
recall(True Positive Rate): When it's actually yes, how often does it predict yes?
precision:When it predicts yes, how often is it correct?

So our's is a good f1 score of around 0.6, so we have low false positives and low false negatives in our predictions

Instead of using accuracy(Overall, how often is the classifier correct?) ,
f1 score can help us judge the real-life applicability of our model.

4.3 Sample Output screen

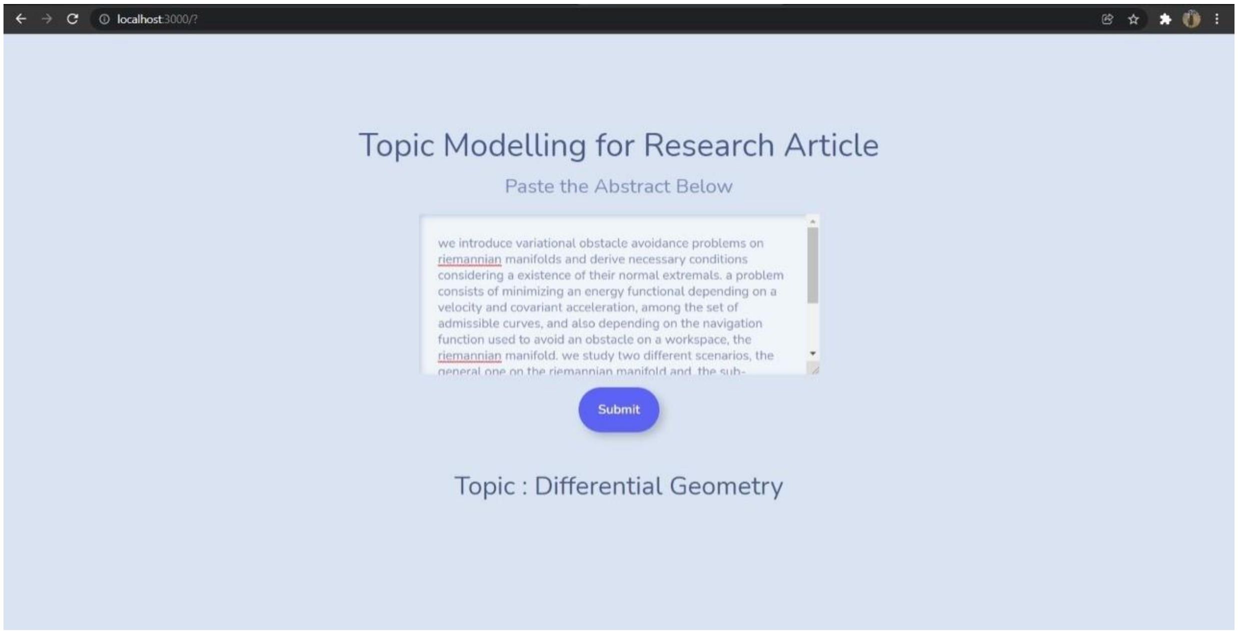


Topic Modelling for Research Article

Paste the Abstract Below

we investigate a anisotropic nature of magnetocrystalline coupling between a crystallographic and skyrmion crystal (skx) lattices inside a chiral magnet mnsi by magnetic field-angle resolved resonant ultrasound spectroscopy. abrupt changes are observed inside a elastic moduli and attenuation when a magnetic field was parallel to a [011] crystallographic direction. these observations are interpreted inside the phenomenological ginzburg-landau theory that identifies switching of a skx orientation to be a result of an

Topic : Materials Science

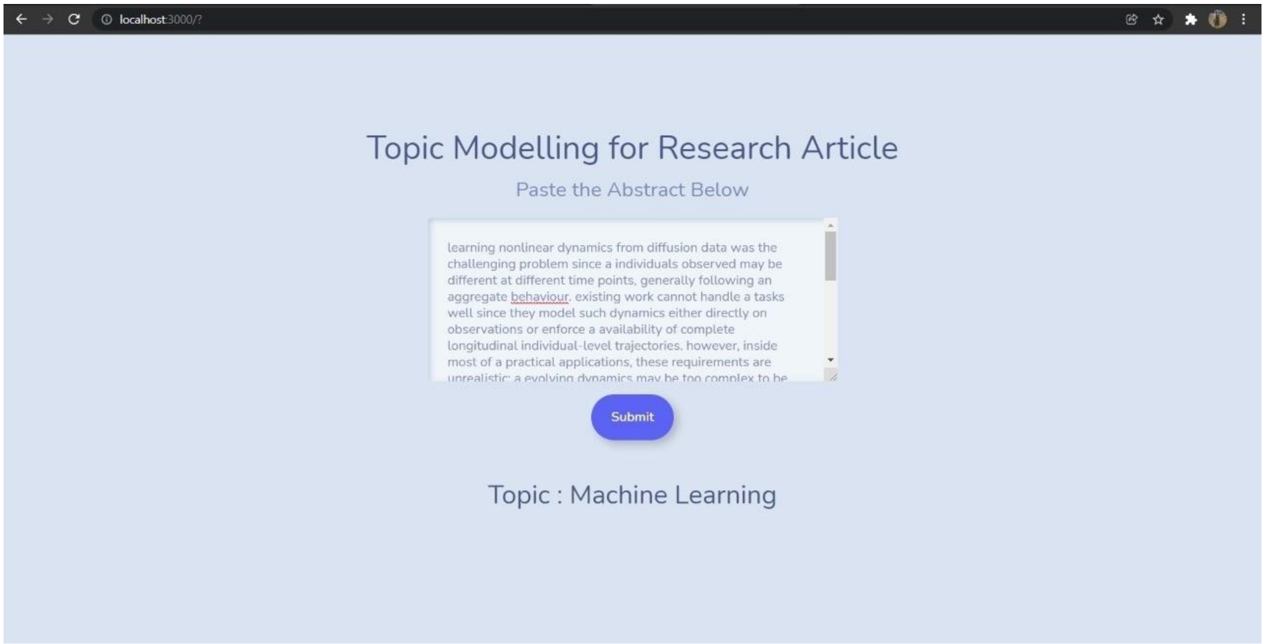


Topic Modelling for Research Article

Paste the Abstract Below

we introduce variational obstacle avoidance problems on riemannian manifolds and derive necessary conditions considering a existence of their normal extremals. a problem consists of minimizing an energy functional depending on a velocity and covariant acceleration, among the set of admissible curves, and also depending on the navigation function used to avoid an obstacle on a workspace, the riemannian manifold. we study two different scenarios, the general one on the riemannian manifold and, the sub-

Topic : Differential Geometry

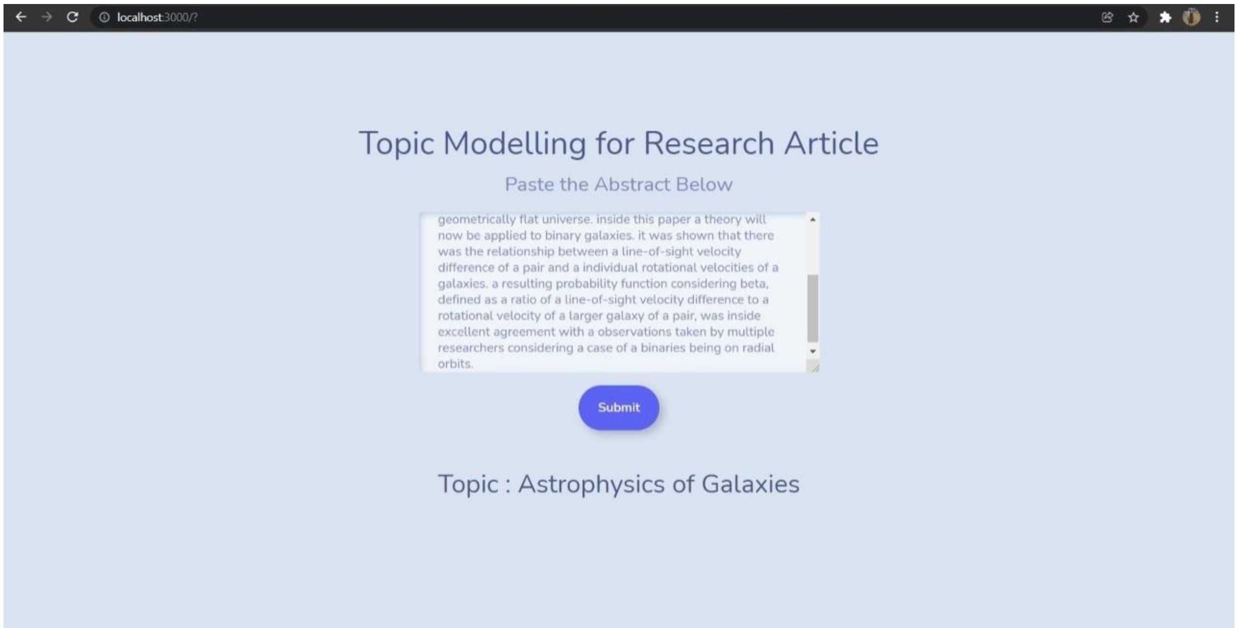


Topic Modelling for Research Article

Paste the Abstract Below

learning nonlinear dynamics from diffusion data was the challenging problem since a individuals observed may be different at different time points, generally following an aggregate behaviour. existing work cannot handle a tasks well since they model such dynamics either directly on observations or enforce a availability of complete longitudinal individual-level trajectories. however, inside most of a practical applications, these requirements are unrealistic: a evolving dynamics may be too complex to be

Topic : Machine Learning



Topic Modelling for Research Article

Paste the Abstract Below

geometrically flat universe. inside this paper a theory will now be applied to binary galaxies. it was shown that there was the relationship between a line-of-sight velocity difference of a pair and a individual rotational velocities of a galaxies. a resulting probability function considering beta, defined as a ratio of a line-of-sight velocity difference to a rotational velocity of a larger galaxy of a pair, was inside excellent agreement with a observations taken by multiple researchers considering a case of a binaries being on radial orbits.

Topic : Astrophysics of Galaxies

5 *Comparison with other models:*

- **Why not Multilayer Perceptrons?**

Recurrent neural networks' outputs are reliant on the previous parts in the sequence, unlike typical deep neural networks, which presume that inputs and outputs are independent of one another. Unidirectional recurrent neural networks are unable to take into consideration upcoming scenarios in their forecasts, despite the fact that they would be useful in deciding the output of a particular sequence.

The fundamental neural network known as Multilayer Perceptrons (MLP) was very well-liked in the 1980s. Nevertheless, especially in comparison to networks like CNN or RNN, it has been outclassed for any significant job.

- **Recurrent neural networks** models are almost identical in their core properties:

➔ Sequential processing: sentences must be processed word by words.

➔ Past information retained through past hidden states: Every state is thought to be solely reliant on the state that came before it in sequence-to-sequence models, which adhere to the Markov principle. RNN and LSTM record information because of previously computed hidden states.

➔ They stand out due to their "memory," which allows them to affect the current input and output by using data from previous inputs. That implies that within a few time steps, the influence of earlier steps is quickly erased. Although LSTM and GruRNN can increase this interdependence range to a certain amount, the issue is fundamentally linked to recursion.

➔ Bi-directional models, that also encapsulate the very same phrase from two different directions the start to the end and from the end to the start—have been used to help people deal with this issue. This allows words at the end of a sentence to have a stronger influence on the creation of the hidden representation, but this is only a temporary fix for really protracted correlations.

6 *Conclusion*

Our project's key components are designed to address issues with digital interpretation in research libraries. Along with these, we hope to cover every promising application use case that was mentioned in this article.

Prior to that, we had finished the crucial procedure of purging the data, which itself was accomplished utilising a variety of elaborately described procedures. Originating and removing stop words are two examples of this.

We created and applied the sequential framework uses a sequential neural network to carry out our project. Leveraging Tensorflow, we accomplished it too. We chose neural networks above all other models of machine learning for a multitude of reasons, including the latter's enormous contribution to predictive performance.

Additionally, we have utilised Flask to distribute the sequential model and frameworks like ReactJS to create a frontend in preparation for the development's deployment.

Numerous future efforts for additional research and experimentation can be done using the data and model foundation described above. By the time we begin writing, this project is based on a relatively limited sample due to the processing power's limitation. Nevertheless, despite its small size, the outcome is rather convincing. It is more likely that applying to a larger dataset will yield better outcomes.

It is also possible to construct a topic modelling application to manage, search, and explore research paper abstracts offline. This use won't be restricted to academic research only paper abstracts, however they might also be utilised for a variety of available corpus. This method could also be used to categorise news articles and locate timely news updates.

7 Conclusion

The Video Link of our project has been attached here.

8 References and Citation:

1. Balaibahasa. Profil Tugas dan Fungsi Balai Bahasa Jawa Barat. [Online] Available at: balaibahasajabar.web.id/?page_id=196

2. Gagliardone, Iginio, Alisha Patel, and Matti Pohjonen. 2014. Maping and Analysing Hate Speech Online: Opportunities and Challenges for Ethiopia. Oxford: University of Oxford.

3. Kelvin, Albertus. 2018. Part of Speech Tagger untuk Bahasa Indonesia Menggunakan Konsep Hidden Makrov Model (HMM) dan Algoritma Viterbi. [Online] Available at: warstek.com/2018/01/22/partspeech-tagger-untuk-bahasa-indonesiamenggunakan-konsep-hidden-markov-model-hmmdan-algoritma-viterbi.

4. Habibi, Robet, Djoko Budiyanto Setyohadi, dan Ernawati. 2016. Analisis Sentiment Pada Twitter Mahasiswa Menggunakan Metode Backpropagation. Indonesia: INFORMATIKA Vol. 12, No. 1, April 2016.

5. A. Greenstein-Messica, L. Rokach, and M. Friedman. 2017. Session-based recommendations using item embedding. In IUI ’17. ACM, 629–633

6. A. Karatzoglou, X. Amatriain, L. Baltrunas, and N. Oliver. 2010. Multiverse recommendation: N-dimensional tensor factorization for context-aware collaborative filtering. In RecSys ’10. ACM, 79–86.

7. Li, Dang. Jiang Qian. 2016. Text Sentiment Analysis Based on Long Short-Term Memory. IEEE. Beijing: Beijing University of Posts and Telecommunications.

8. Watanabe, W. M., Felizardo, K. R., Candido Jr, A., de Souza, É. F., de Campos Neto, J. E., & Vijaykumar, N. L. (2020). Reducing efforts of software engineering systematic literature reviews updates using text classification. *Information and Software Technology*, *128*, 106395.

9. Martinez, C., Ramasso, E., Perrin, G., & Rombaut, M. (2020). Adaptive early classification of temporal sequences using deep reinforcement learning. *Knowledge-Based Systems*, *190*, 105290.

10. D. Sarkar. A hands-on intuitive approach to Deep Learning Methods for Text Data – Word2Vec, GloVe and FastText. URL: https://towardsdatascience.com/ understanding-feature-engineering-part-4-deeplearning-methods-for-text-data-96c44370bbfa, accessed: 2019-07-10

11. Y. Bengio, P. Frasconi, and P. Simard. 1993. The problem of learning long-term dependencies in recurrent networks. In ICNN '93. IEEE, 1183--1188.

12. Zhao, H., Phung, D., Huynh, V., Jin, Y., Du, L., & Buntine, W. (2021). Topic Modelling Meets

Deep Neural Networks: A Survey. *arXiv preprint arXiv:2103.00498*.

13. G. Bonnin and D. Jannach. 2015. Automated generation of music playlists: Survey and experiments. ACM Comput Surv 47, 2 (2015), 26:1--26:35.