DOCSUMO DATA VERSE 2023

Visualize Train Conquer

Event Date : 17th January, 2023 - 23rd January, 2023

A

Data Insight Report On

ArXiv Datasets for Scholarly Articles

Author

Team Non-Linear

Nirajan Bekoju

Nishant Luitel

Submission Date

23rd January, 2023

Contents

List of Figures

List of Tables

Chapter 1

Introduction to Docsumo DataVerse 2023

1.1 Abstract

ArXiv is a public service repository and open-source archive for scholarly articles in the fields of physics, mathematics, computer science, quantitative biology, quantitative finance, statistics, electrical engineering and systems science, and economics.

1.2 Problem Description

In this data challenge, Datasets that contains abstracts and categories columns that are collected from the arXiv portal are given to the participants. There are altogether 157 subject categories. For this competition, datasets having 23 classes are provided.

The task is to build a model to predict the category given paper abstract and title.

1.3 Evaluation Metric

The evaluation metric for this competition is Mean F1-Score. The F1 score, commonly used in information retrieval, measures accuracy using the statistics precision(p) and recall(r).

Precision is the ratio of true positives (tp) to all predicted positives (tp + fp). Recall is the ratio of true positives (tp) to all actual positives (tp + fn).

The F1 score is given by:

$$\mathrm{F1} = 2\frac{\mathrm{p}\cdot\mathrm{r}}{\mathrm{p}+\mathrm{r}} \quad \text{where} \quad \mathrm{p} = \frac{\mathrm{tp}}{\mathrm{tp}+\mathrm{fp}}, \quad \mathrm{r} = \frac{\mathrm{tp}}{\mathrm{tp}+\mathrm{fn}}$$

The F1 metric weights recall and precision equally, and a good retrieval algorithm will maximize both precision and recall simultaneously. Thus, moderately good performance on both will be favored over extremely good performance on one and poor performance on the other.

1.4 Dataset Overview

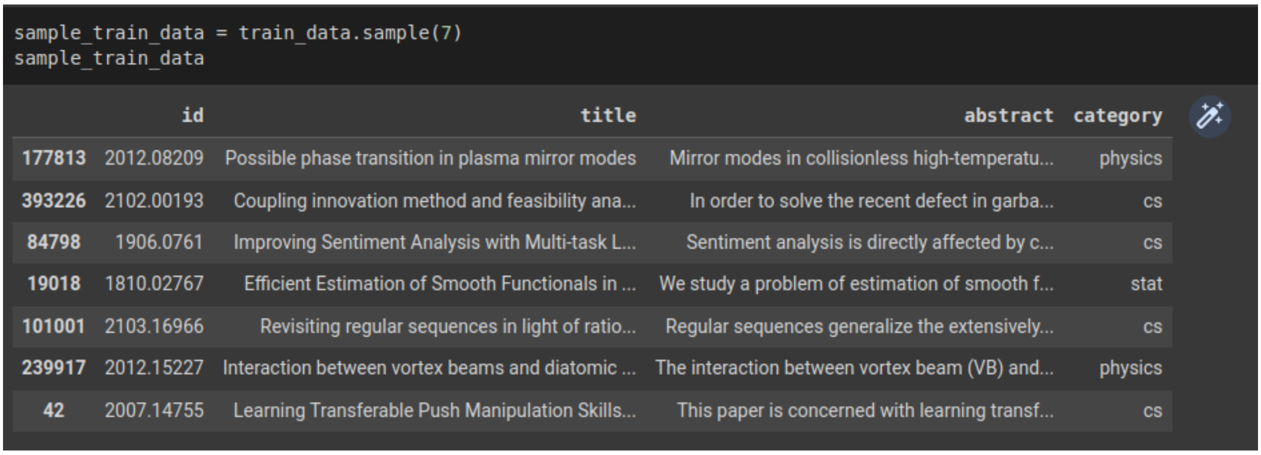
Participants were given three files.

1. **train.csv** : the training set
2. **test.csv** : the test set
3. **sample_submission.csv** : a sample submission file in the correct format

1.4.1 Train Dataset Overview

The train dataset contains total four columns as follow:

1. **id** : Unique id of the article
2. **title** : Title of the article
3. **abstract** : Abstract of the article
4. **category** : Label of the article



```
sample_train_data = train_data.sample(7)
sample_train_data
```

	id	title	abstract	category
177813	2012.08209	Possible phase transition in plasma mirror modes	Mirror modes in collisionless high-temperatu...	physics
393226	2102.00193	Coupling innovation method and feasibility ana...	In order to solve the recent defect in garba...	cs
84798	1906.0761	Improving Sentiment Analysis with Multi-task L...	Sentiment analysis is directly affected by c...	cs
19018	1810.02767	Efficient Estimation of Smooth Functionals in ...	We study a problem of estimation of smooth f...	stat
101001	2103.16966	Revisiting regular sequences in light of ratio...	Regular sequences generalize the extensively...	cs
239917	2012.15227	Interaction between vortex beams and diatomic ...	The interaction between vortex beam (VB) and...	physics
42	2007.14755	Learning Transferable Push Manipulation Skills...	This paper is concerned with learning transf...	cs

Figure 1.1: Train Data Sample

Altogether, there are 8,61,236 entries in the training dataset. None of id, title and abstract column are null. However, 4 row for category column are null. The train dataset is of size 995.08 MB.

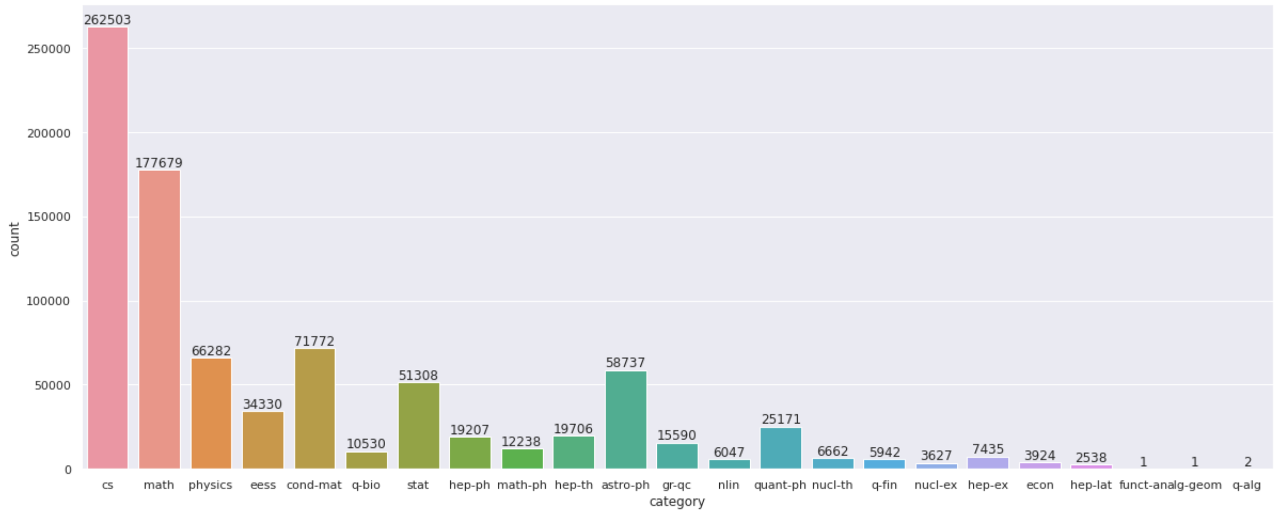


Figure 1.2: Train Data Category Countplot

The name of the categories in the train data are cs, math, physics, eess, cond-mat, q-bio, stat, hep-ph, math-ph, hep-th, astro-ph, gr-qc, nlin, quant-ph, nucl-th, q-fin, nucl-ex, hep-ex, econ, hep-lat, funct-an, alg-geom, and q-alg.

From the graph, we can observe that the count of some labels like cs, math, physics, cond-mat, astro-ph are greater than 50,000 while some labels have their count range from 2,000 to 30,000 and label "funct-an", "alg-geom" and "q-alg" have 2, 1 and 1 count respectively.

So, it is clear that the given training data is highly imbalanced and so we have to take the imbalanced into consideration. Otherwise, the model will be highly biased toward the majority class. Hence, we will have to use various sampling methods to balance the data and feed to the model for increasing the reliablility of the classifier model.

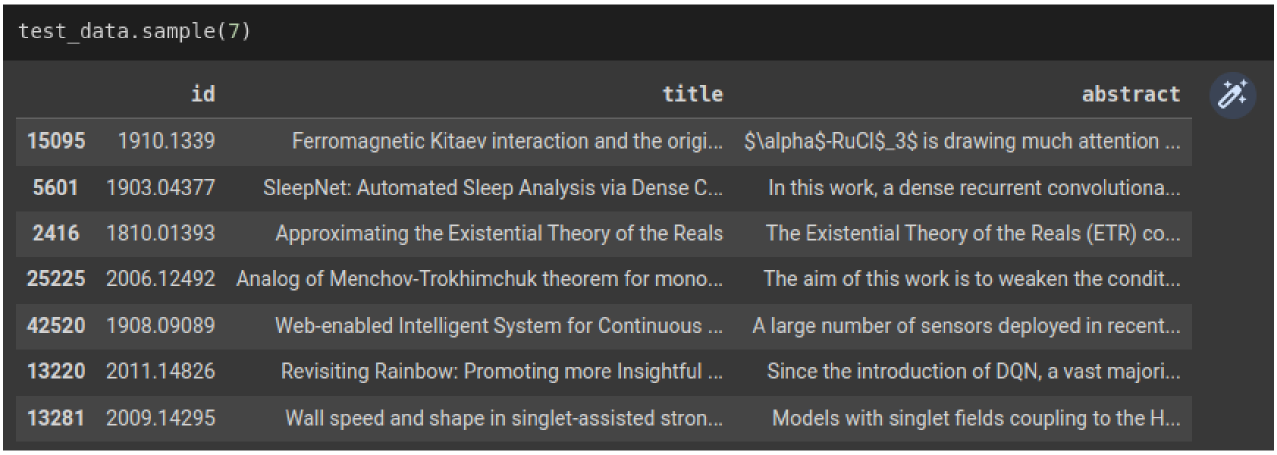
Some sample title and their category are shown below:

id	title	category
2012.08209	Possible phase transition in plasma mirror modes	physics
2102.00193	Coupling innovation method and feasibility analysis of garbage	cs
1810.02767	Efficient Estimation of Smooth Functionals in Gaussian Shift Models	stat

Table 1.1: Train Data Title Category Sample

1.4.2 Test Dataset Overview

The test dataset contains total of four columns : id, title,abstract and category. The size of test dataset is of 52.13 MB. There are 43,785 entries in the test dataset.



```
test_data.sample(7)
```

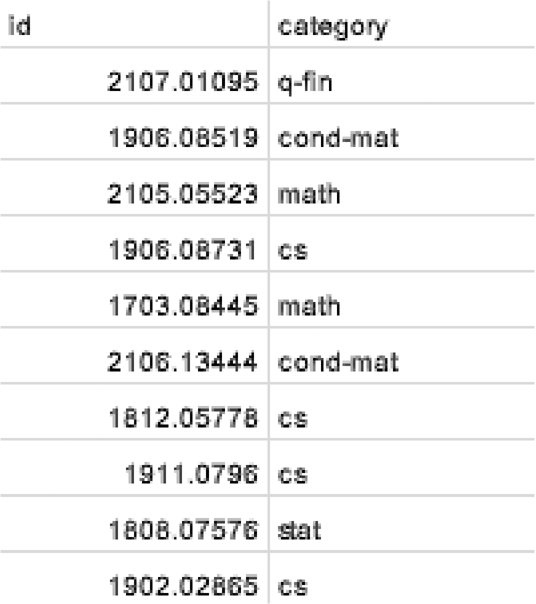
	id	title	abstract
15095	1910.1339	Ferromagnetic Kitaev interaction and the origi...	α-RuCl$_3$ is drawing much attention ...
5601	1903.04377	SleepNet: Automated Sleep Analysis via Dense C...	In this work, a dense recurrent convolutiona...
2416	1810.01393	Approximating the Existential Theory of the Reals	The Existential Theory of the Reals (ETR) co...
25225	2006.12492	Analog of Menchov-Trokhimchuk theorem for mono...	The aim of this work is to weaken the condit...
42520	1908.09089	Web-enabled Intelligent System for Continuous ...	A large number of sensors deployed in recent...
13220	2011.14826	Revisiting Rainbow: Promoting more Insightful ...	Since the introduction of DQN, a vast majori...
13281	2009.14295	Wall speed and shape in singlet-assisted stron...	Models with singlet fields coupling to the H...

Figure 1.3: Test Data Sample

1.4.3 Submission Format

For every author in the dataset, submission files should contain two columns: ID and Category. The file should contain a header and have the following format in csv file:

The file should contain a header and have the following format:



id	category
2107.01095	q-fin
1906.08519	cond-mat
2105.05523	math
1906.08731	cs
1703.08445	math
2106.13444	cond-mat
1812.05778	cs
1911.0796	cs
1808.07576	stat
1902.02865	cs

Figure 1.4: Submission File Format

Chapter 2

Data Insights

2.1 Text Data Analysis

Text data analysis were performed in the training datasets. Analysis like title length based on character frequency, title length based on word frequency were analysed.

2.1.1 Title length based on character frequency

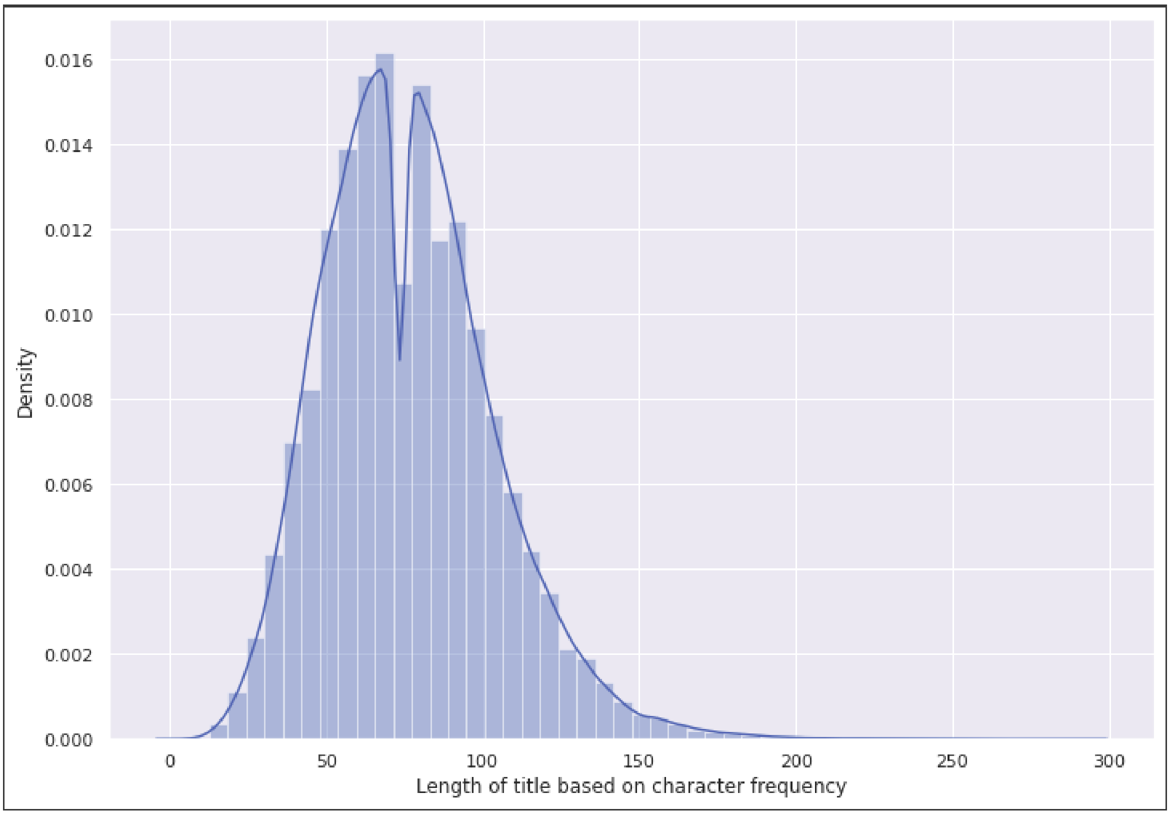


Figure 2.1: Title length based on number of characters (Bi-modal Distribution)

From the figure above, it can be deduced that the title length based on number of characters follow bimodal distribution. Its basic statistics value can be observed in following box plot.

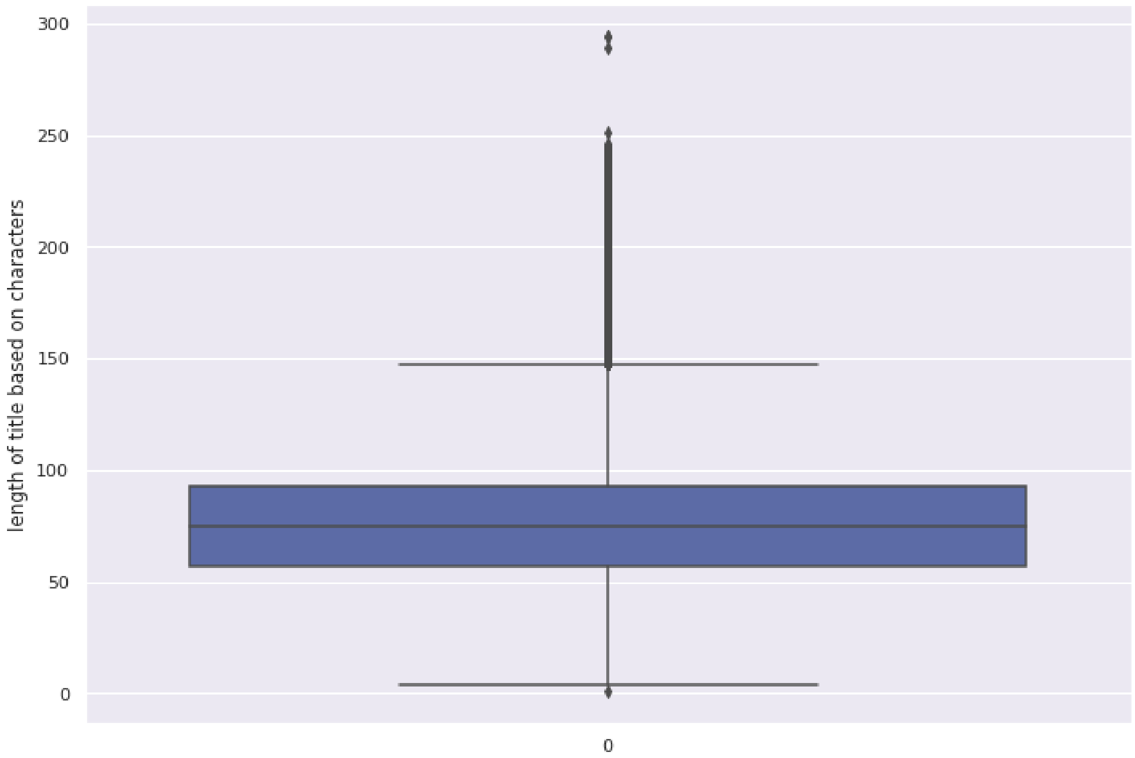


Figure 2.2: Box plot of title length based on character frequency

From the box plot, the average number of characters in a title in the training dataset is found to be 76.3678 ± 27.2314. The statistics information are shown below.

mean	76.3678
standard deviation	27.2314
minimum	1
first quartile	57
median	75
third quartile	93
maximum	294

Table 2.1: Title length statistics based on character frequency

2.1.2 Title length based on word frequency

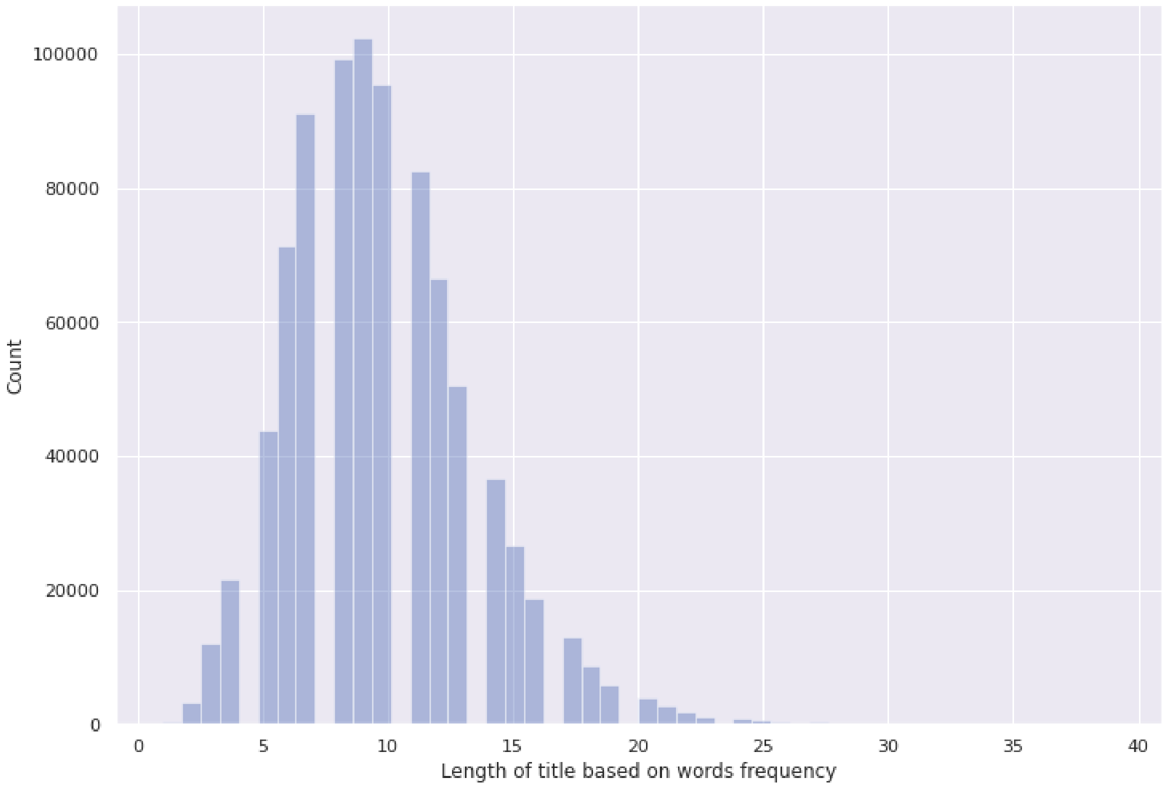


Figure 2.3: Title length based on word frequency (Right Skewed Normal Distribution)

From the above distribution plot, it can be deduced that the title length based on word frequency follow right skewed normal distribution.

From the box plot Fig : 2.4, the average number of words in a title in the training dataset is found to be 9.7681 ± 3.6153. The statistics information are shown below.

mean	9.7681
standard deviation	3.6153
minimum	1
first quartile	7
median	9
third quartile	12
maximum	39

Table 2.2: Title length statistics based on word frequency

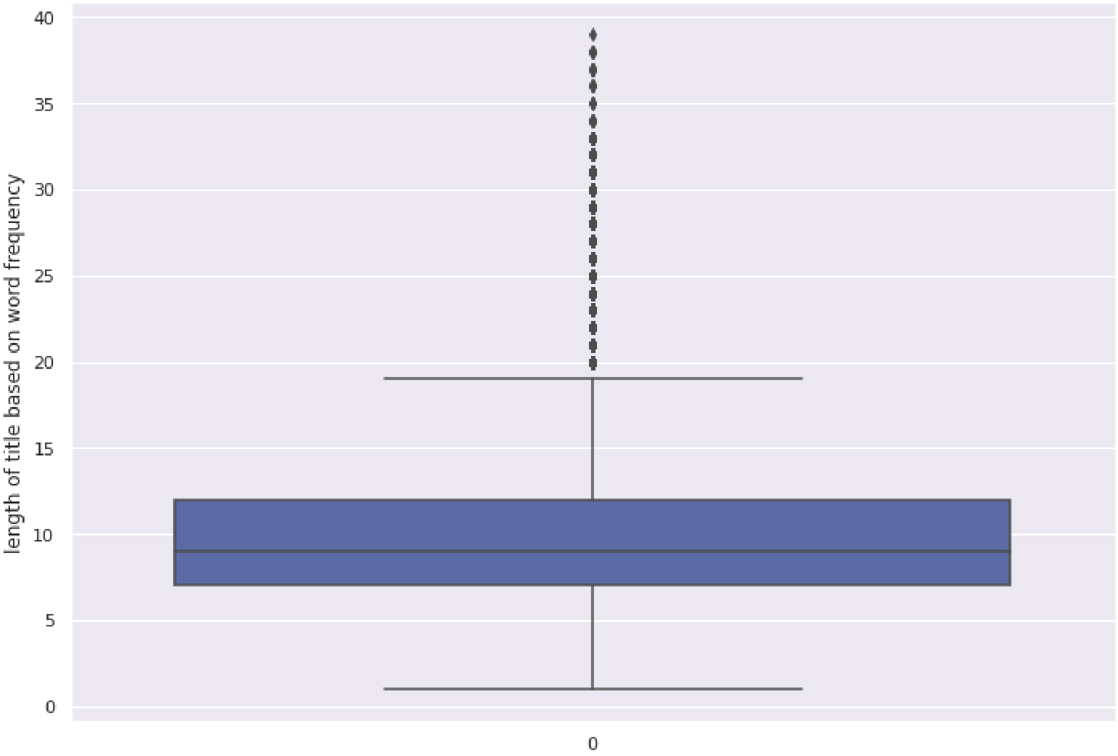


Figure 2.4: Box plot of title length based on word frequency

2.2 Word Cloud

A word cloud (also known as a tag cloud) is a visual representation of words. Cloud creators are used to highlight popular words and phrases based on frequency and relevance. They provide quick and simple visual insights that can lead to more in-depth analyses.

For this report, word cloud is created using the title column of the training data removing all stopwords. From the word cloud Fig 2.5, we can see the clear dominance of the majority class : cs, math, physics, cond-mat, stat, astro-ph, and quant-ph.

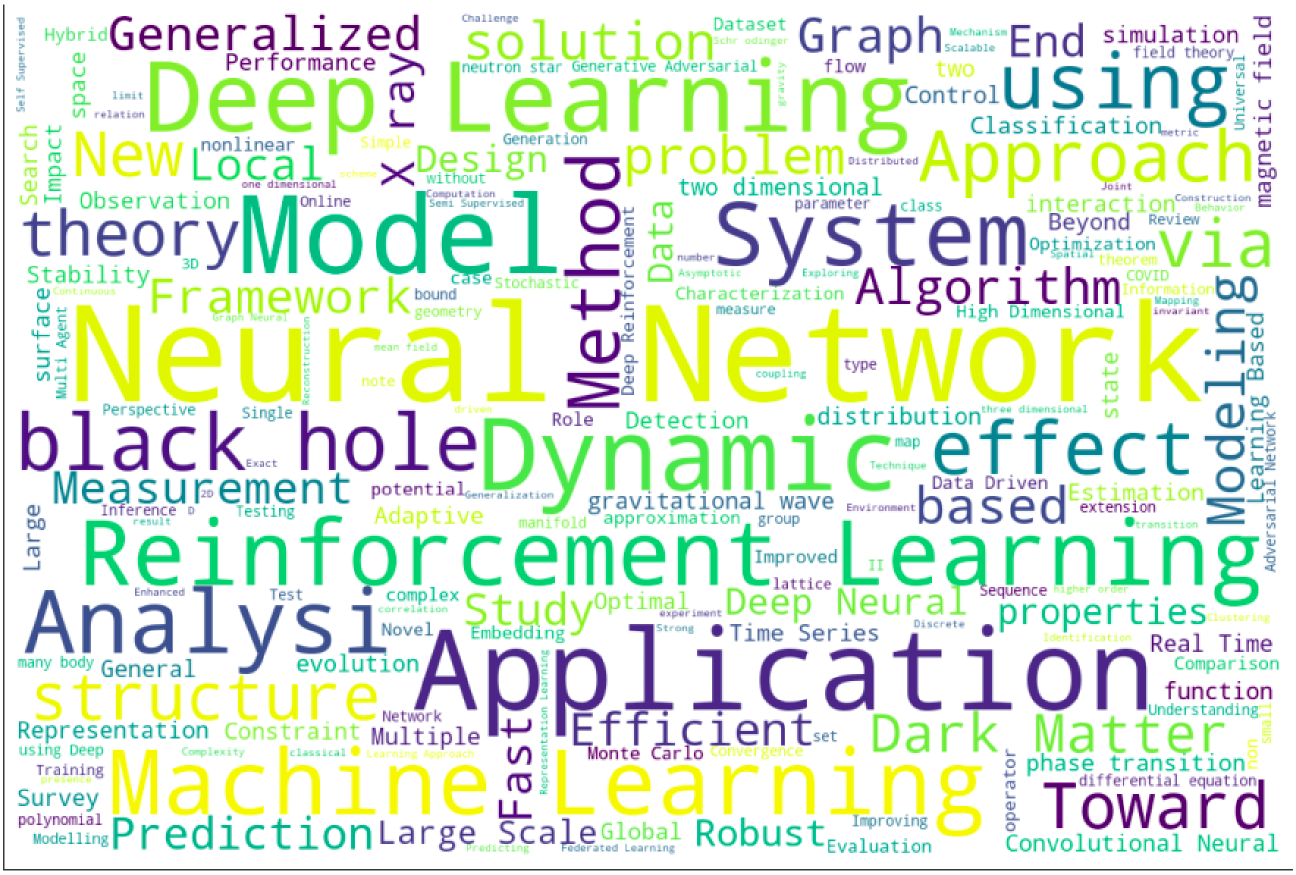


Figure 2.5: Title Word Cloud

Let's view the word cloud on specific topics.

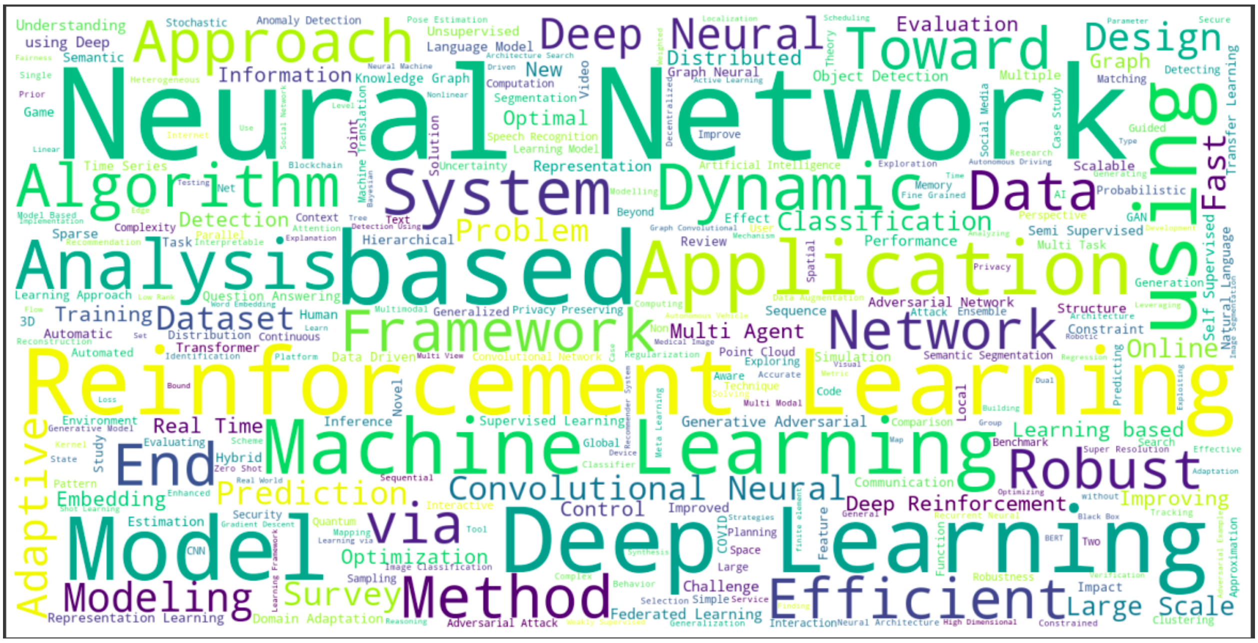


Figure 2.6: CS Word Cloud

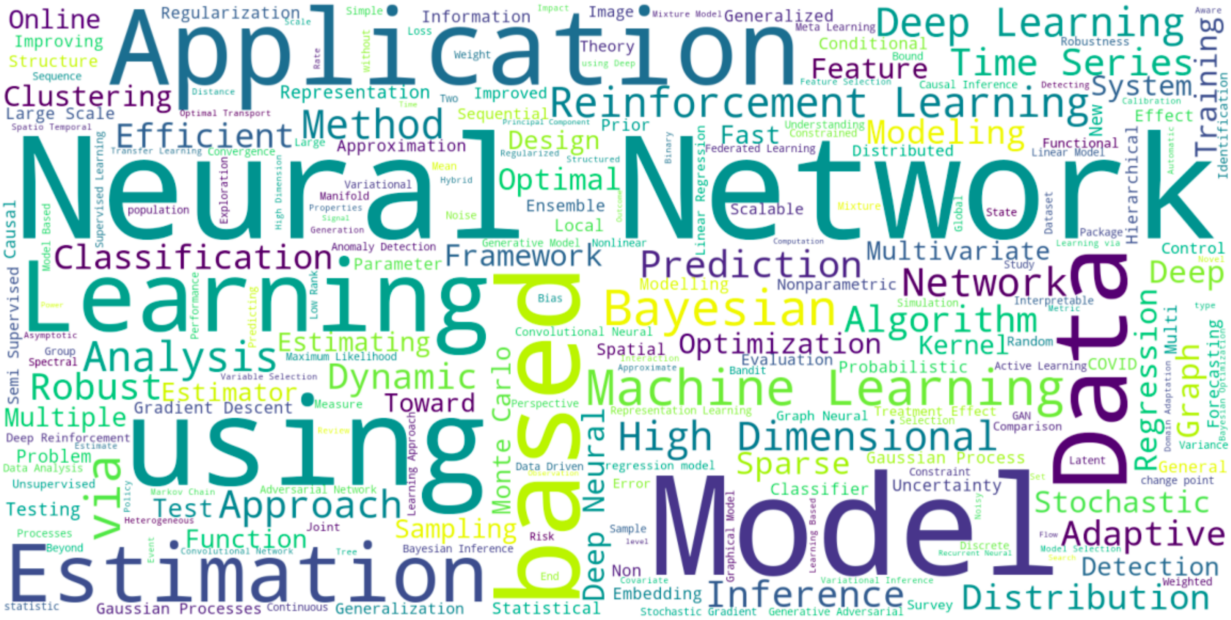


Figure 2.7: Stat Word Cloud

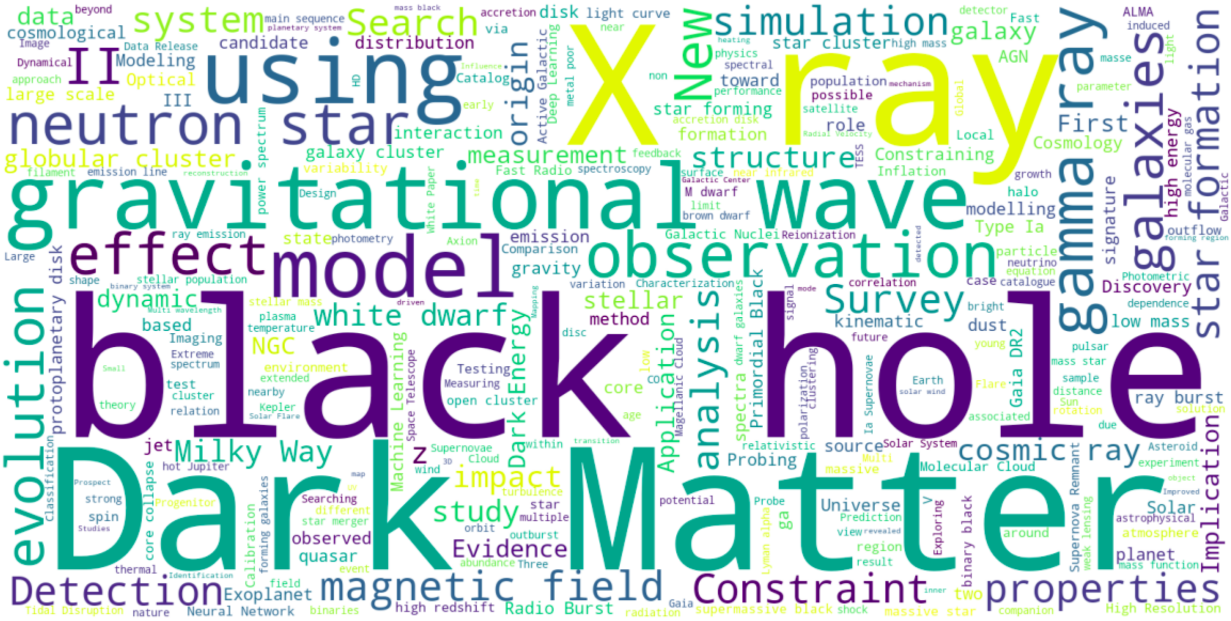


Figure 2.8: Astro-ph Word Cloud

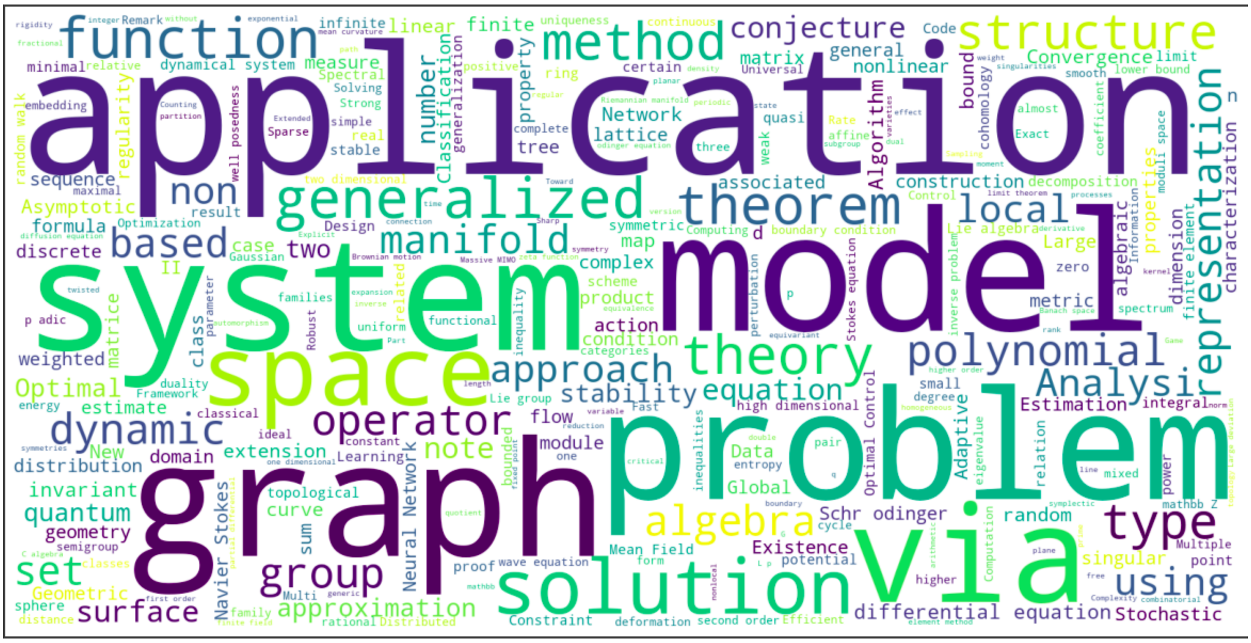


Figure 2.9: Math Word Cloud

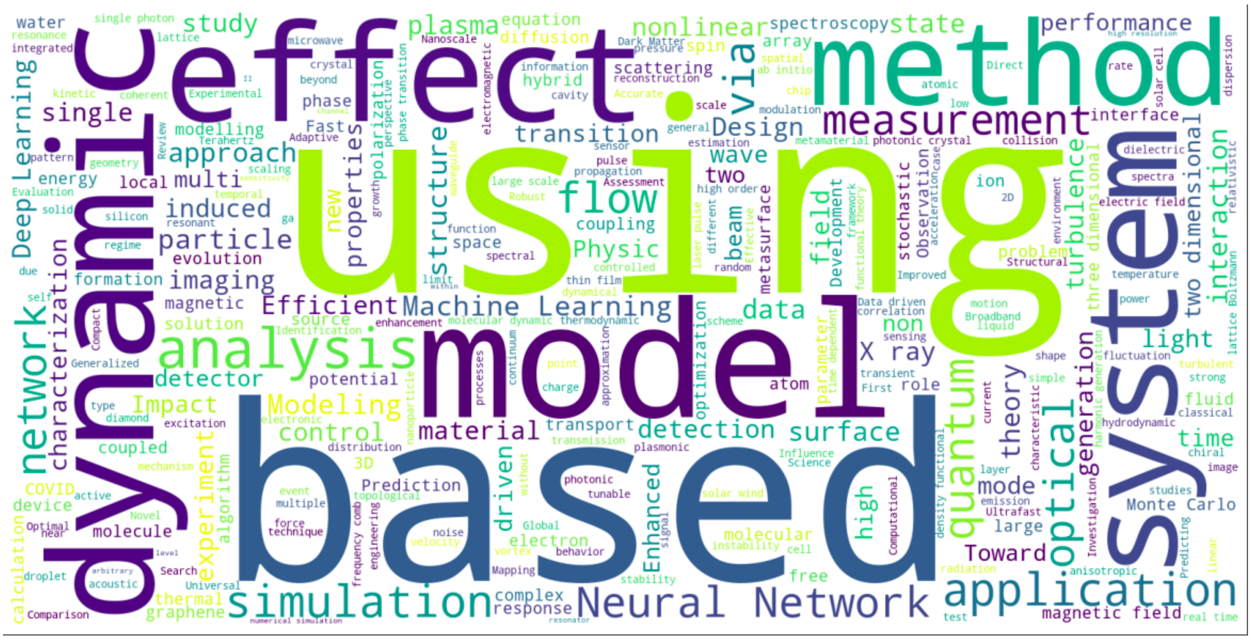


Figure 2.10: Physics Word Cloud

2.2.1 Inference from label based word cloud

From cs and stat word cloud, we can see lots of similar word in both labels like model, neural network, machine learning, network, data, distribution, etc. When looking on to the abstract of these labels, just by judging through the corpus, some of them were even hard for human to classify between them. Similar was the case with astro-ph and physics which can be clearly visualized in the word cloud.

2.2.2 Minority class word cloud

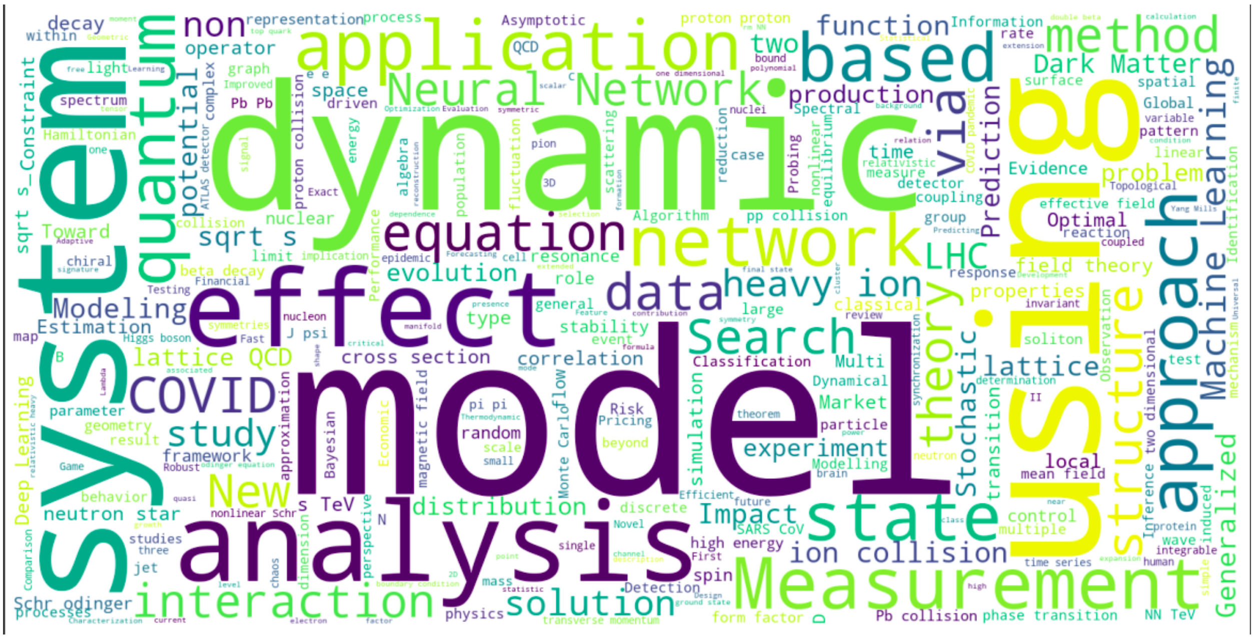


Figure 2.11: Minority class Word Cloud

Minority class includes labels with number of data less than 15000 in training datasets. The classes are q-bio, hep-ex, math-ph, nucl-th, nlin, q-fin, econ, nucl-ex, hep-lat, q-alg, funct-an, and alg-geom.

2.3 N-gram Exploration

For the exploration of the most frequency unigram, stopwords were removed from the corpus created from the whole training corpus. Then, we got the count of all unique word from the corpus in a hash map. Using the hash map, we drew this bar graph by taking the most frequent words. The result is as follow :

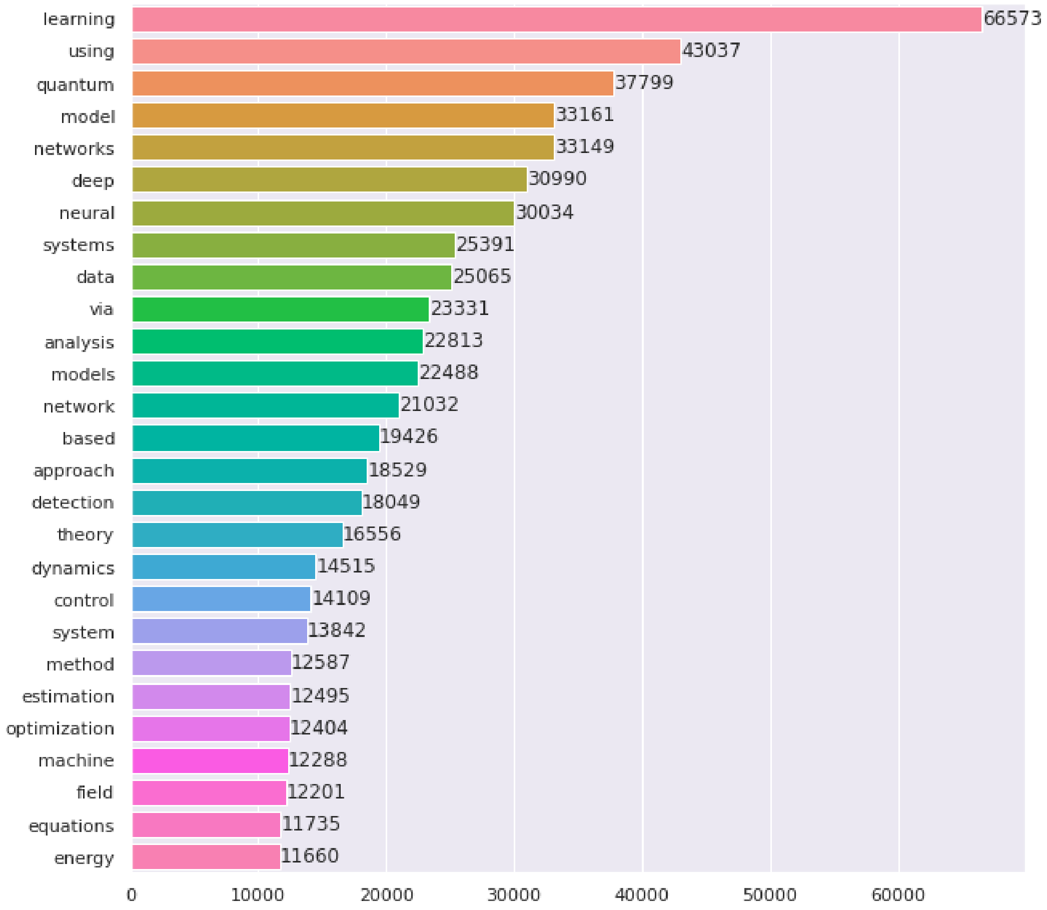


Figure 2.12: Most frequent words

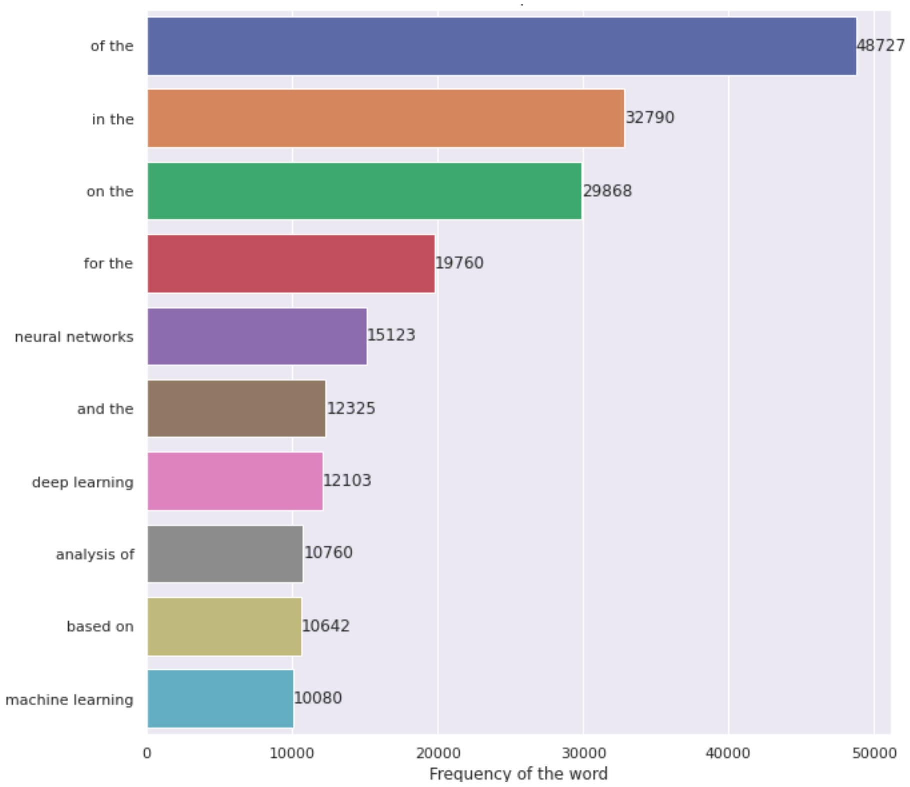


Figure 2.13: Most frequent bi-grams

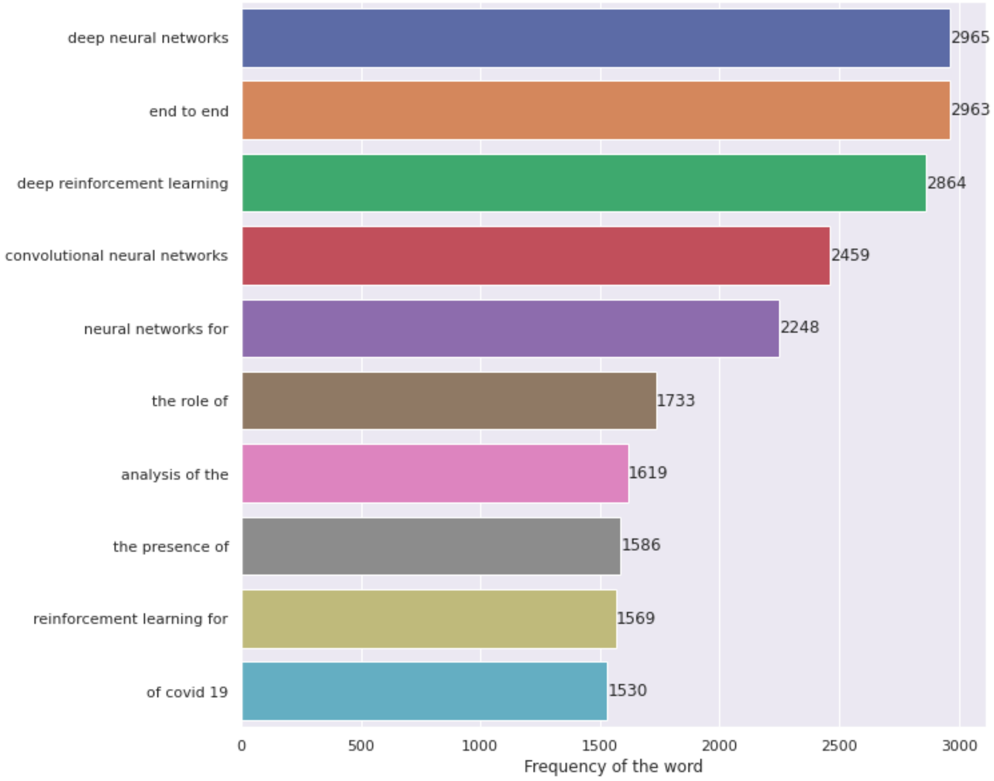
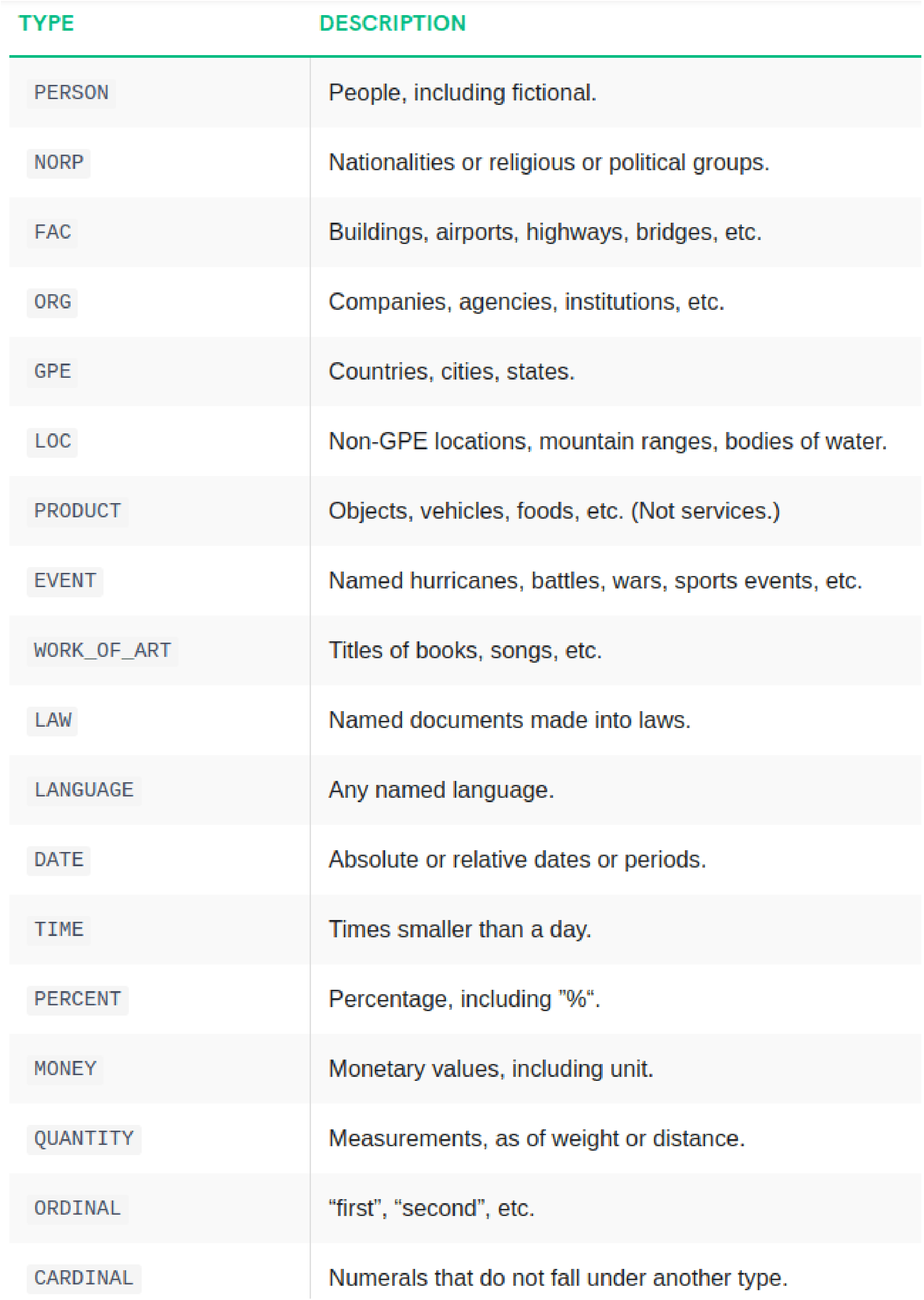


Figure 2.14: Most frequent tri-grams

2.4 Named Entity Recognition

Named-entity recognition (NER) (also known as (named) entity identification, entity chunking, and entity extraction) is a subtask of information extraction that seeks to locate and classify named entities mentioned in unstructured text into pre-defined categories such as person names, organizations, locations, medical codes, time expressions, quantities, monetary values, percentages, etc. Following table shows the label of NER along with its description.



TYPE	DESCRIPTION
PERSON	People, including fictional.
NORP	Nationalities or religious or political groups.
FAC	Buildings, airports, highways, bridges, etc.
ORG	Companies, agencies, institutions, etc.
GPE	Countries, cities, states.
LOC	Non-GPE locations, mountain ranges, bodies of water.
PRODUCT	Objects, vehicles, foods, etc. (Not services.)
EVENT	Named hurricanes, battles, wars, sports events, etc.
WORK_OF_ART	Titles of books, songs, etc.
LAW	Named documents made into laws.
LANGUAGE	Any named language.
DATE	Absolute or relative dates or periods.
TIME	Times smaller than a day.
PERCENT	Percentage, including ”%“.
MONEY	Monetary values, including unit.
QUANTITY	Measurements, as of weight or distance.
ORDINAL	“first”, “second”, etc.
CARDINAL	Numerals that do not fall under another type.

Figure 2.15: NER Label Description

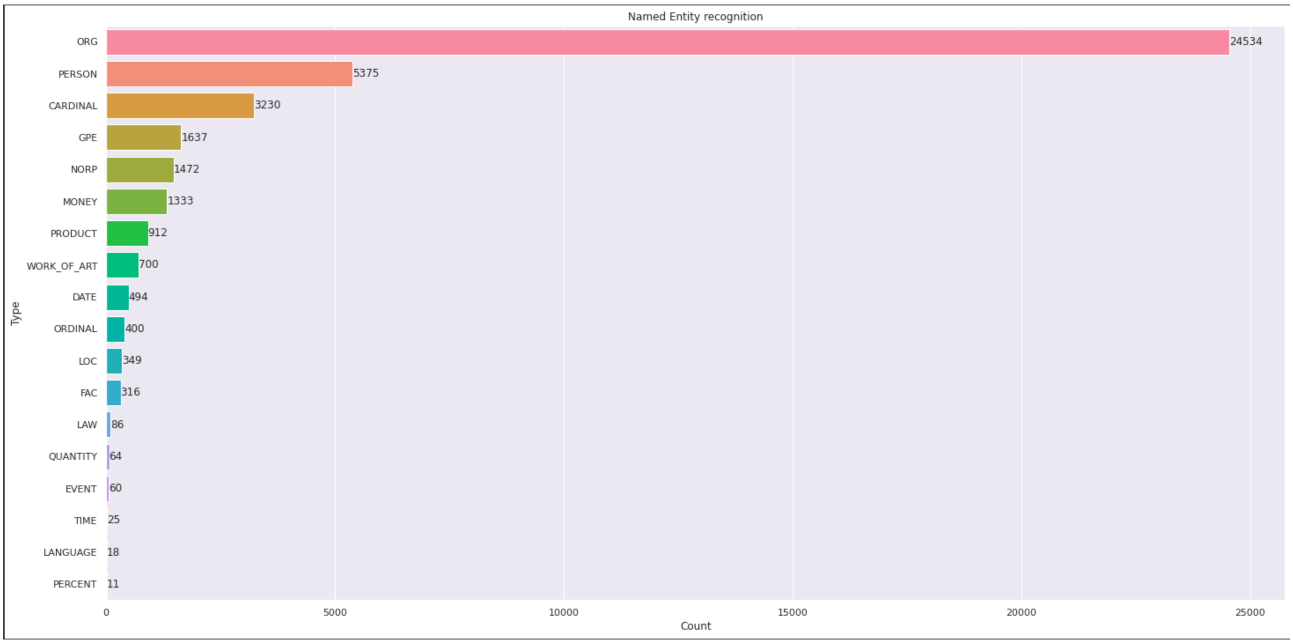


Figure 2.16: NER Countplot

2.4.1 NER Label Exploration

For the generation of following plot based on specific NER label, 50,000 sample were taken from the title column of the training dataset.

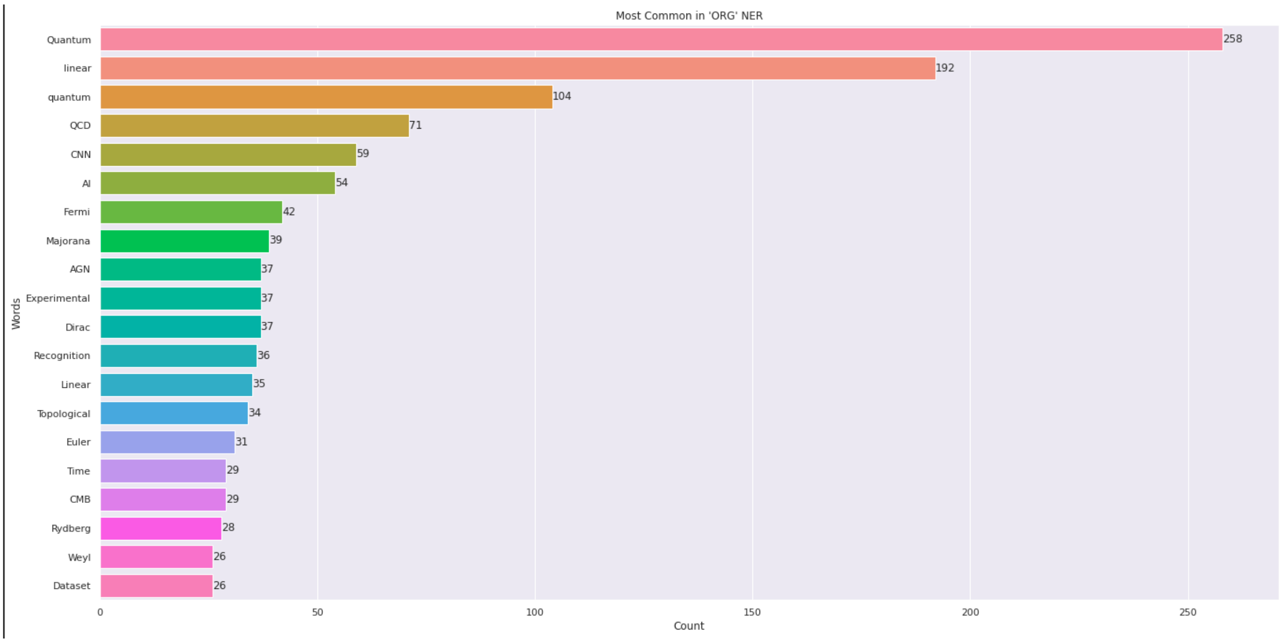


Figure 2.17: Most common word in ORG NER

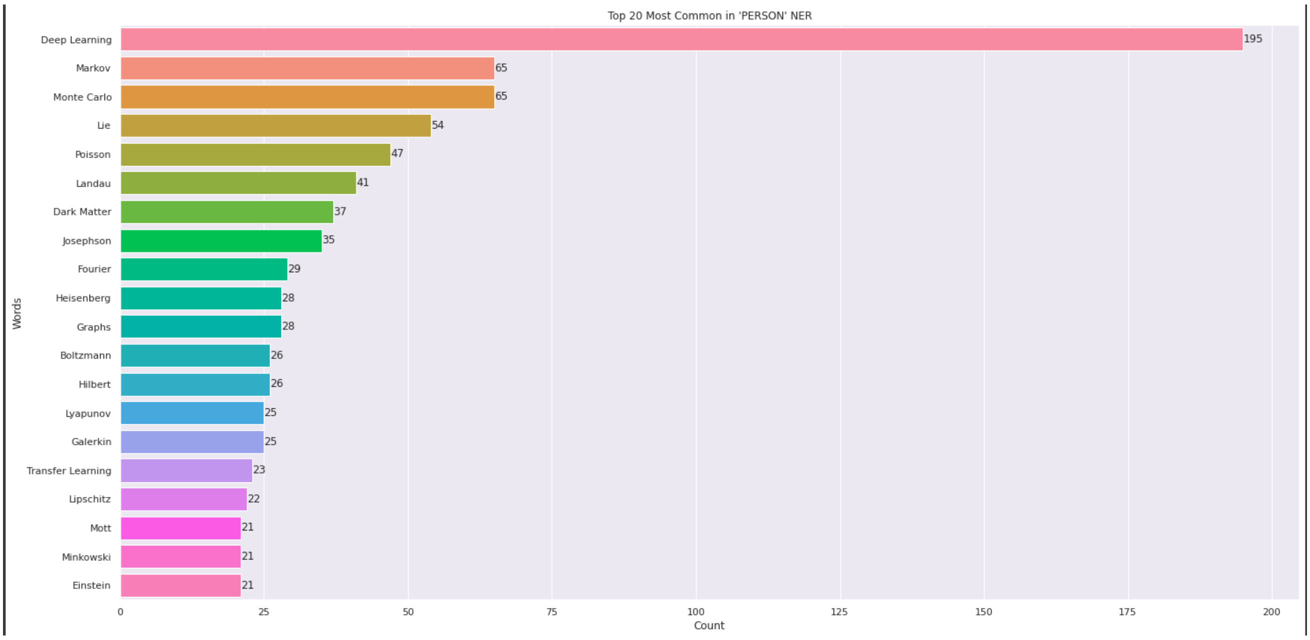


Figure 2.18: Most common word in PERSON NER

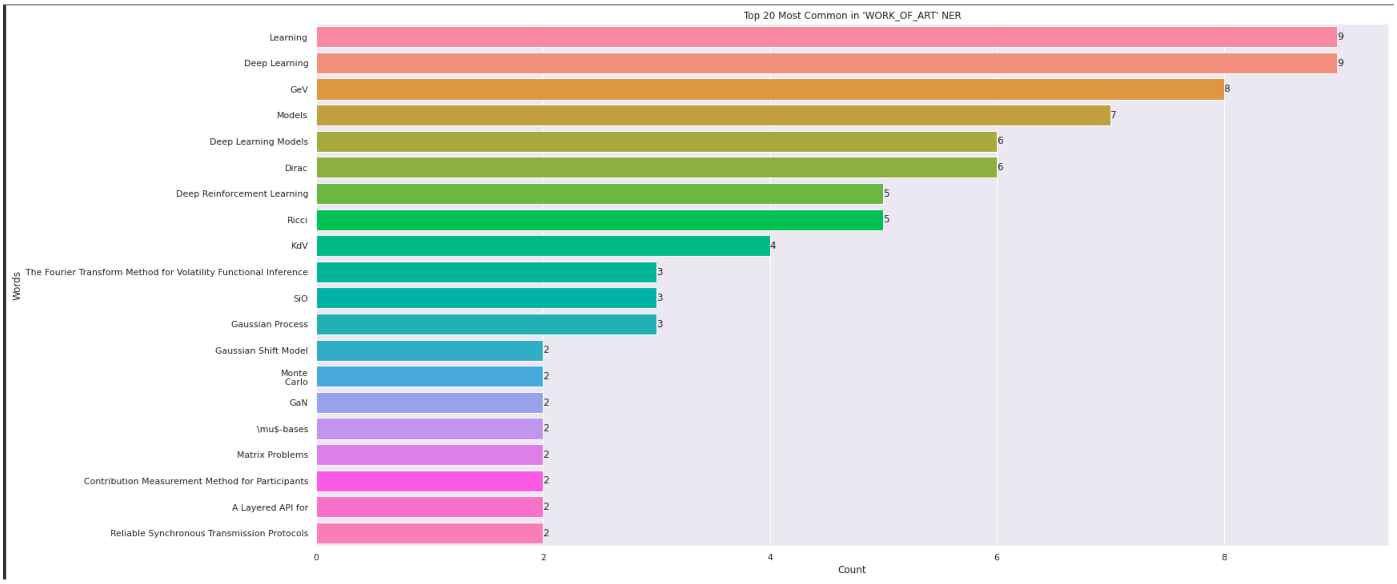


Figure 2.19: Most common words in WORK OF ART NER

2.4.2 Inference from NER

From fig 2.16, it can be deduced that ORG highly dominate the title of the articles following PERSON, CARDINAL, etc.

2.5 Part of Speech Tagging

Parts of speech (POS) tagging is a method that assigns part of speech labels to words in a sentence.

NNP	Proper Noun, Singular
NN	Noun, Singular
IN	Preposition or subordinating conjunction
JJ	Adjective
NNS	Noun, plural
DT	Determiner
CC	Coordinating conjunction
VBG	Verb, gerund or present participle
:	Mid-sentence punctuation (: ; ... – -)
$	Currency Sign

Table 2.3: POS Label and Description

2.5.1 Label based POS exploration

For the POS exploration, 1,00,000 sample of data were taken from the training data whose distribution is shown in fig 2.20.

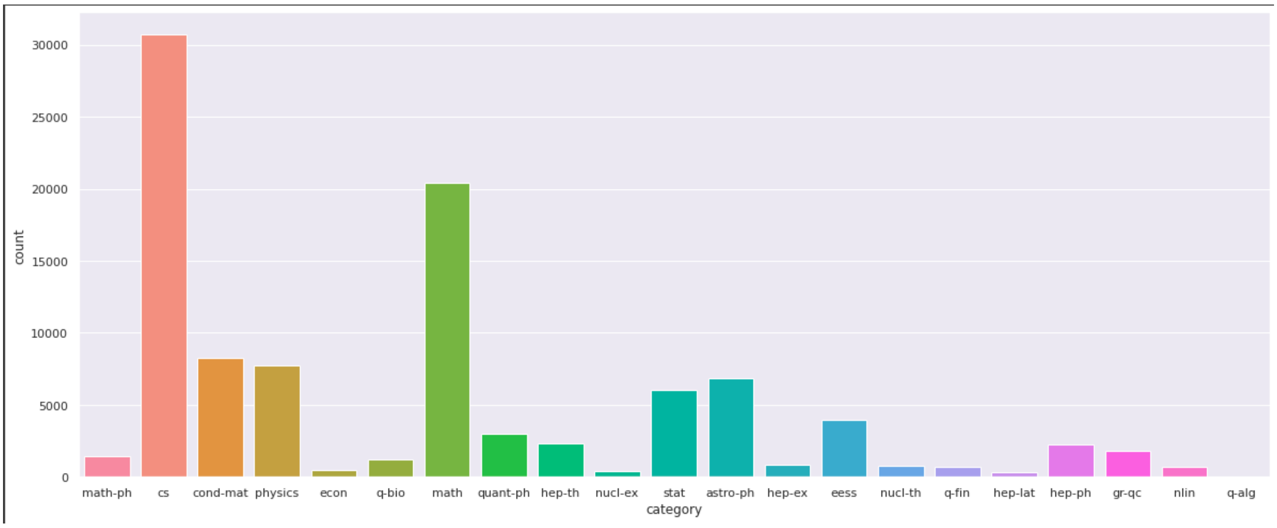


Figure 2.20: 1,00,000 sample data for POS

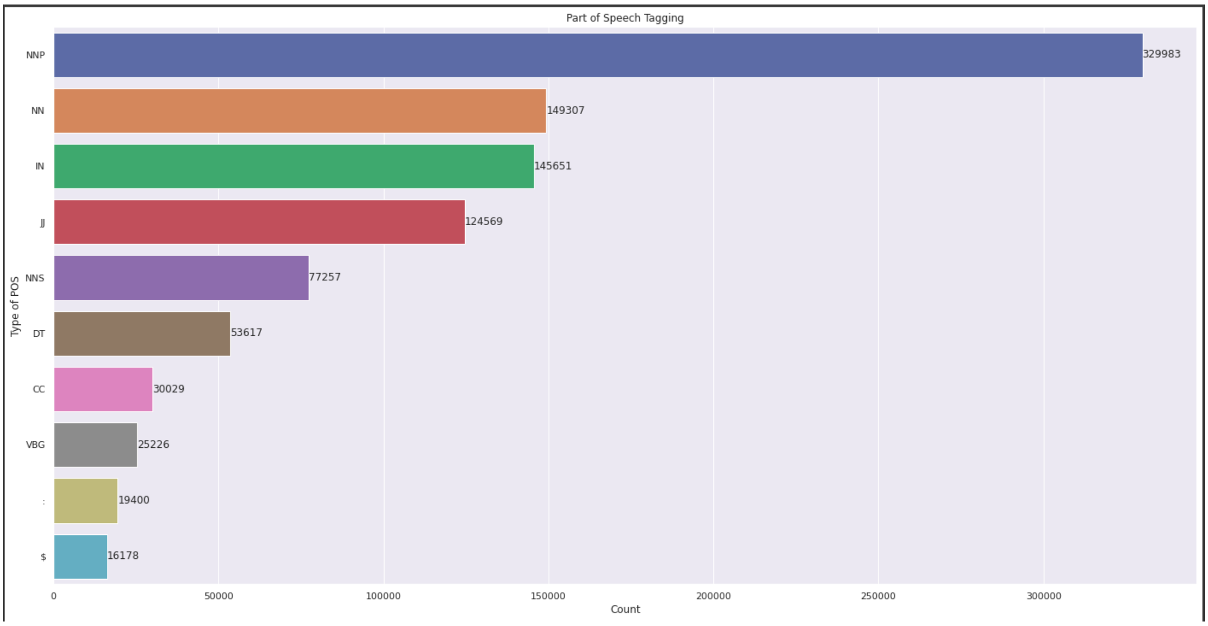


Figure 2.21: POS Countplot for the 1,00,000 sample data

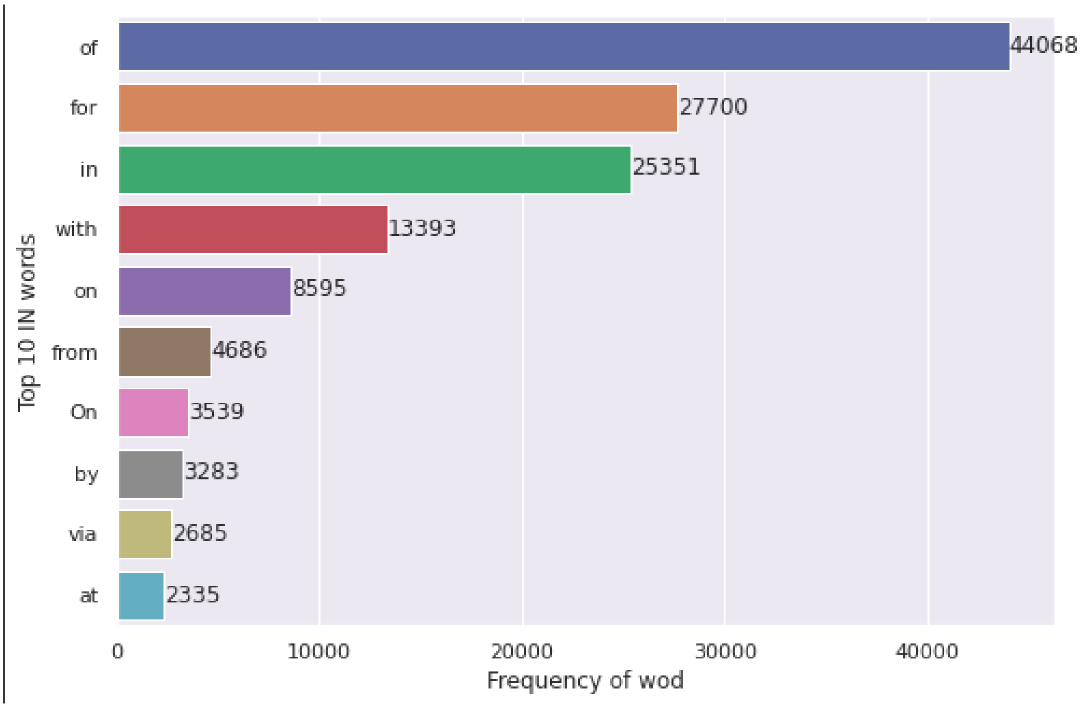


Figure 2.22: Top 10 preprosition in title

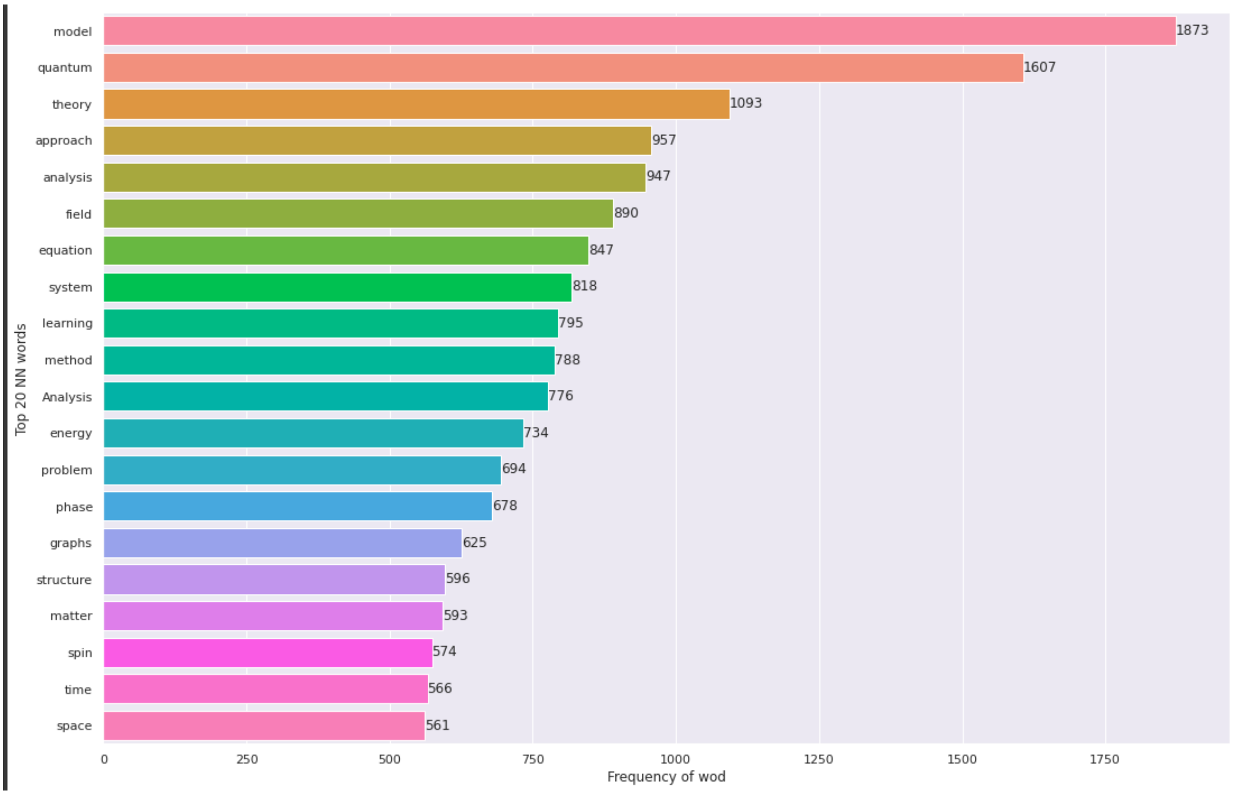


Figure 2.23: Top 20 NN in title

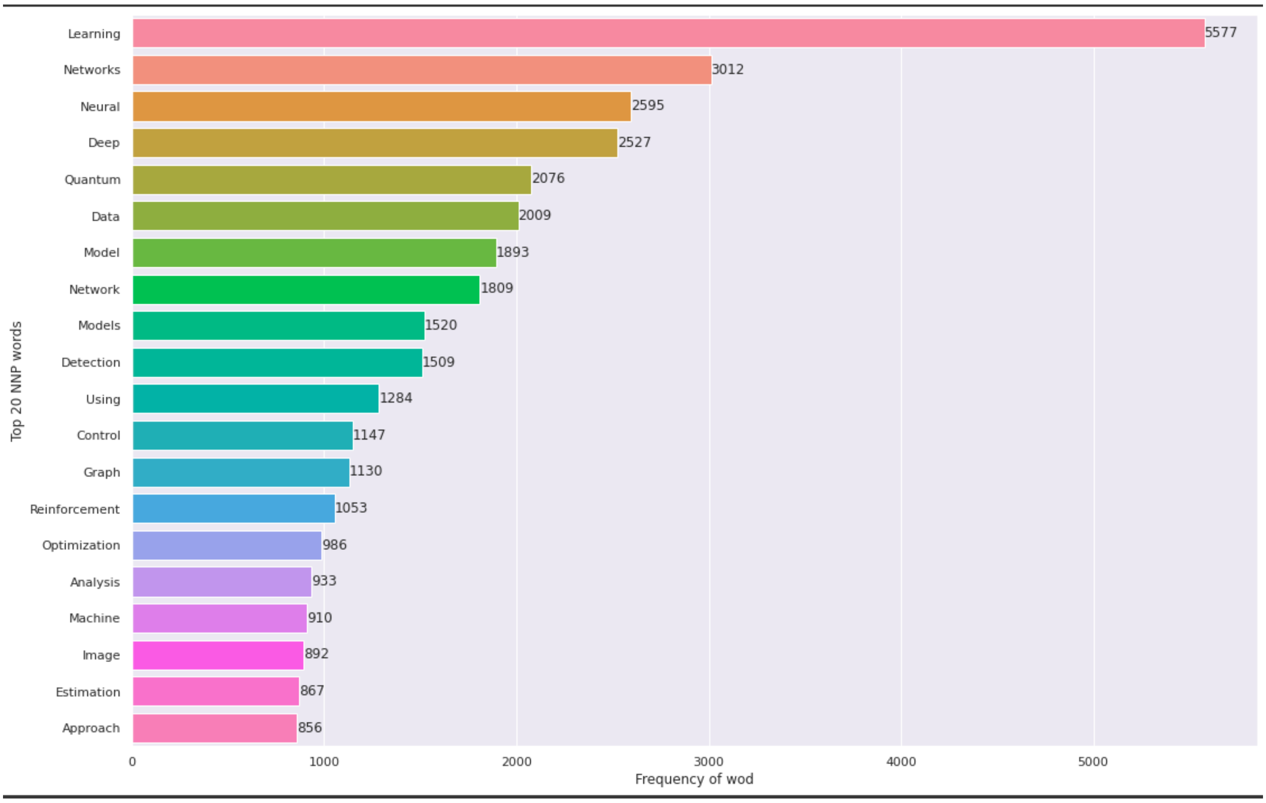


Figure 2.24: Top 20 NNP in title

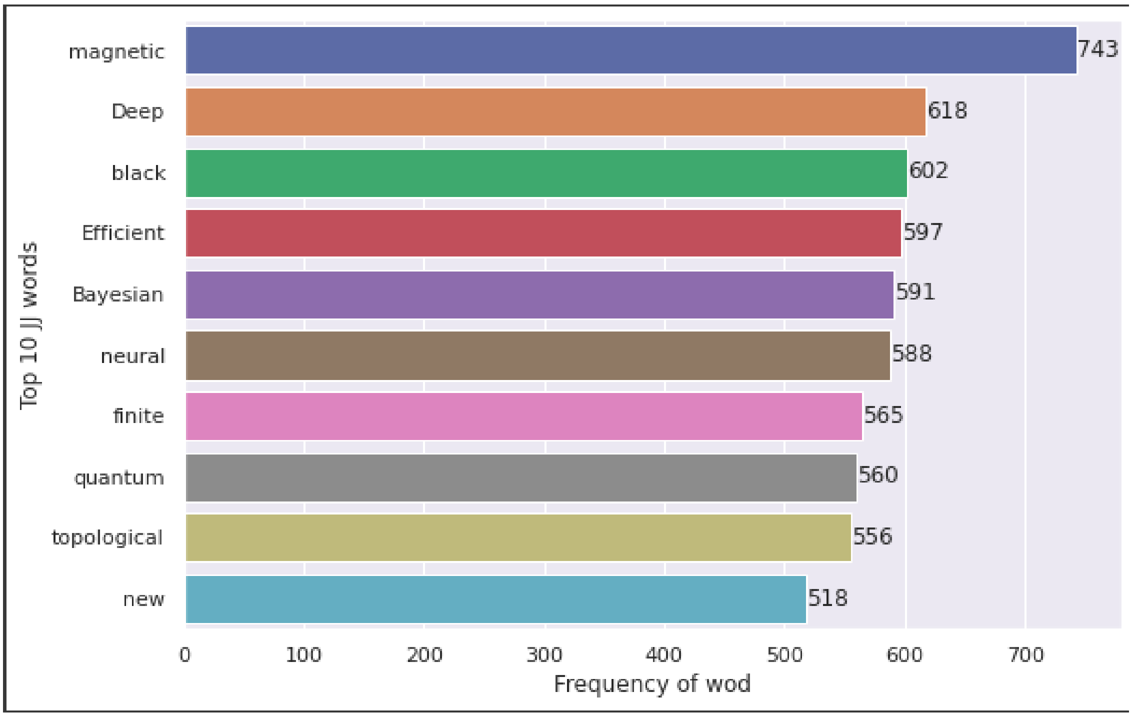


Figure 2.25: Top 10 JJ in title

2.5.2 Inference from POS exploration

From fig 2.21, proper singular noun, prepositions and adjectives were found to dominate the title of the training data which are normally the case. In NNP count plot, most words were found to be from majority class as in fig 2.20.

Chapter 3

Model Development and Evaluation

3.1 Text Preprocessing

For the preprocessing of the given data, the "title" and "abstract" column were joined to form a new colum "text". Then the preprocessing was done on the text column and feeded for tokenization process and model training process. Same was applied to the testing dataset as well.

The steps taken to preprocess the text were as follow:

1. Lower case the text
2. Replace the "-" with white space
3. Remove all the remaining punctuations from string.punctuation
4. Remove all the numbers and words containing numbers
5. Lemmatize the words
6. Return the sentences

For further improvement in preprocessing, instead of remove the numbers we can provide $\langle NUM \rangle$ token for the numbers and words containing numbers.

3.2 Tokenization

For training purpose, title and abstract of the training data were concatenated whose length distribution are shown below.

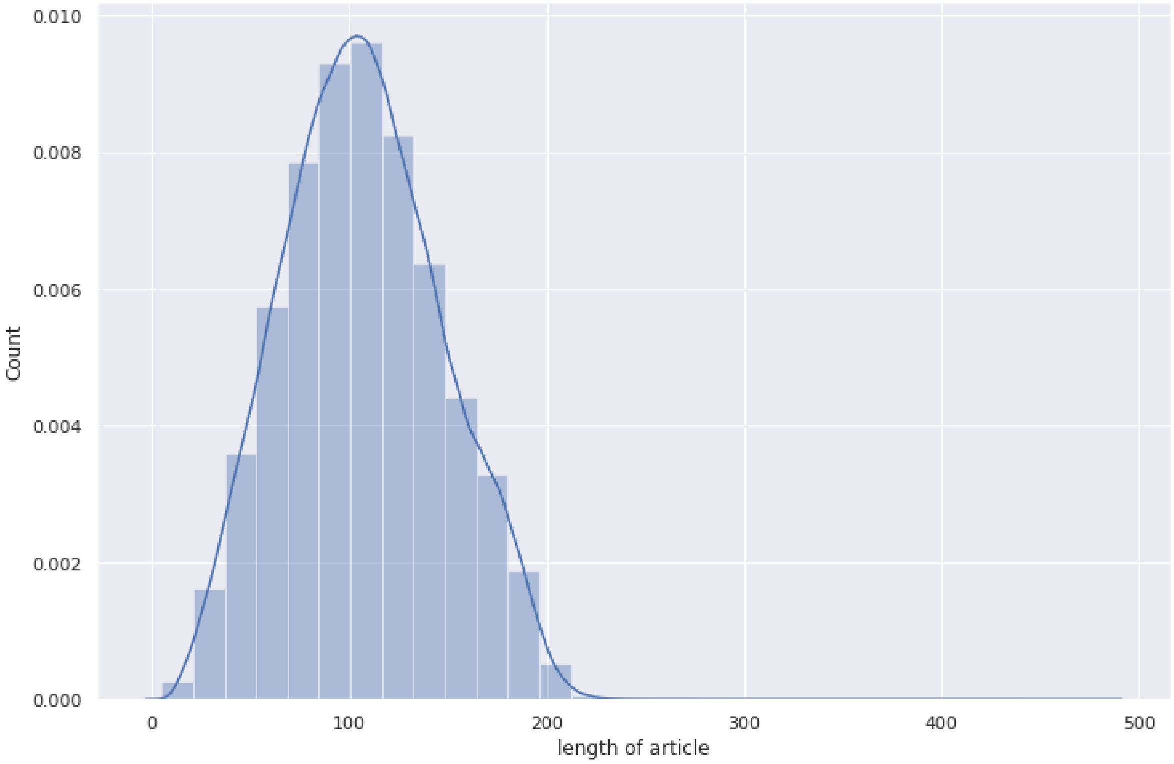


Figure 3.1: Concatenated title and abstract length in training data

From the above distribution of the text length for training data, we get the following statistics value:

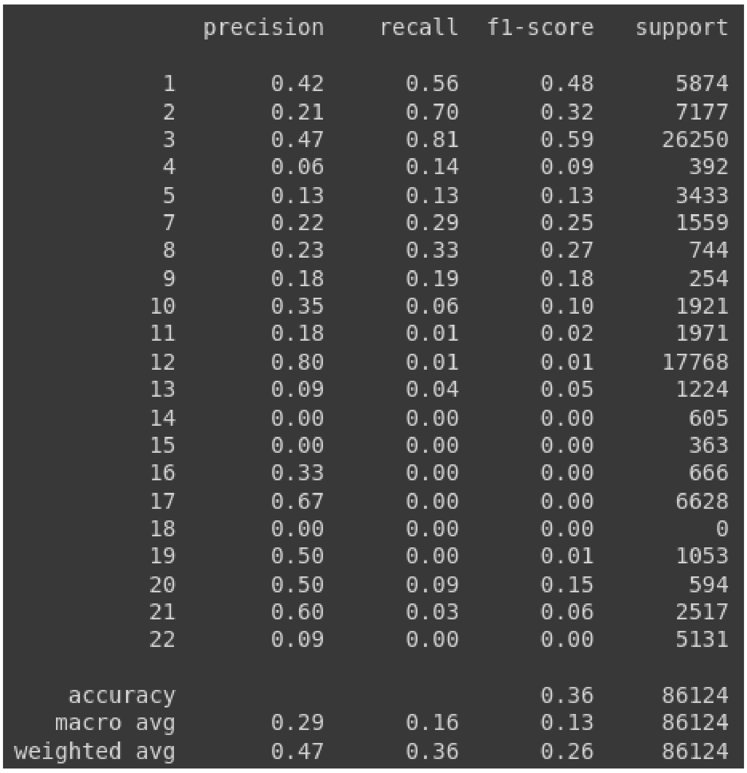
mean	106.9657
standard deviation	39.87
minimum	5
first quartile	78
median	105
third quartile	134
maximum	483

Table 3.1: Title and abstract length statistics based on word frequency

Hence, we choose MAX_PAD_LENGTH = 140 for the tokenization of the data training the LSTMs Deep Neural Networks.

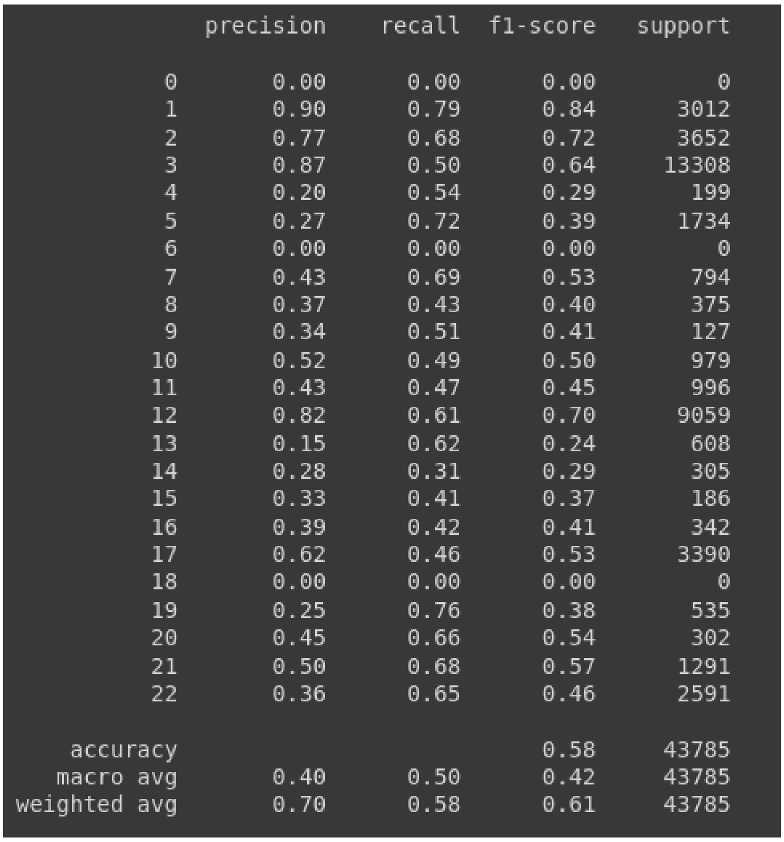
3.3 Machine Learning Model

Support vector machine and Multinomial NB model were trained in the training dataset given. Test dataset was used for the validation and generation of the classification report which is as follow. Multinomial NB was found to be perfoming better on test dataset with accuracy 0.58, f1-score 0.42 and weighted averate f1-score of 0.61.



```
              precision    recall  f1-score   support

           1       0.42      0.56      0.48      5874
           2       0.21      0.70      0.32      7177
           3       0.47      0.81      0.59     26250
           4       0.06      0.14      0.09       392
           5       0.13      0.13      0.13      3433
           7       0.22      0.29      0.25      1559
           8       0.23      0.33      0.27       744
           9       0.18      0.19      0.18       254
          10       0.35      0.06      0.10      1921
          11       0.18      0.01      0.02      1971
          12       0.80      0.01      0.01     17768
          13       0.09      0.04      0.05      1224
          14       0.00      0.00      0.00       605
          15       0.00      0.00      0.00       363
          16       0.33      0.00      0.00       666
          17       0.67      0.00      0.00      6628
          18       0.00      0.00      0.00         0
          19       0.50      0.00      0.01      1053
          20       0.50      0.09      0.15       594
          21       0.60      0.03      0.06      2517
          22       0.09      0.00      0.00      5131

    accuracy                           0.36     86124
   macro avg       0.29      0.16      0.13     86124
weighted avg       0.47      0.36      0.26     86124
```

Figure 3.2: SVM classification report



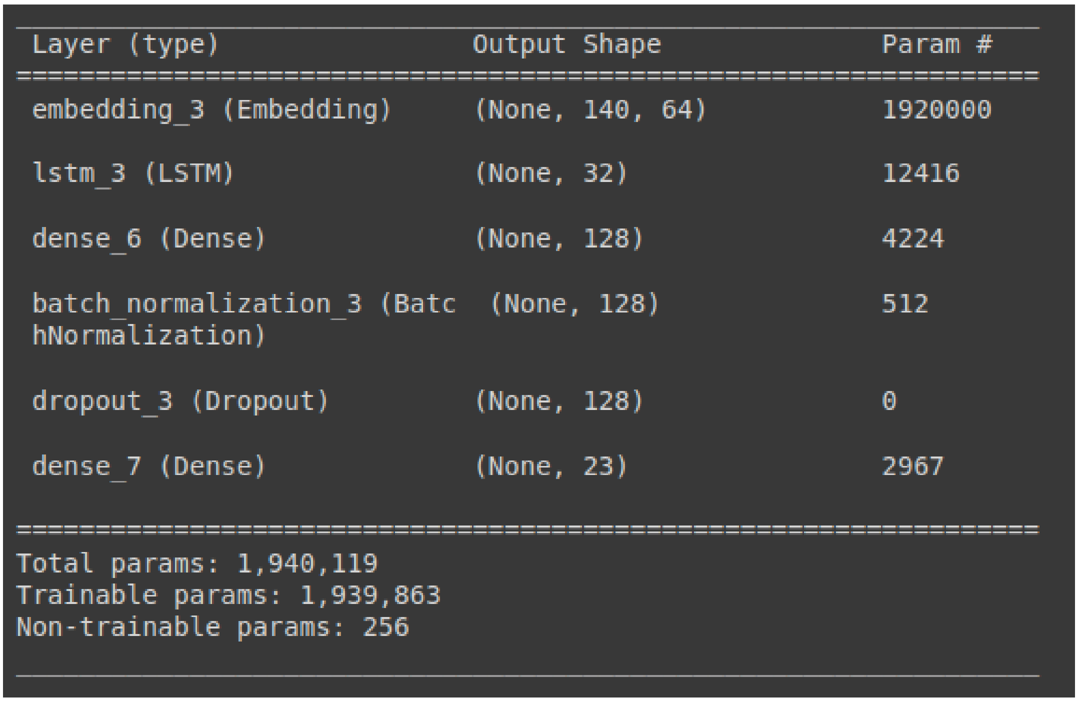
```
              precision    recall  f1-score   support

           0       0.00      0.00      0.00         0
           1       0.90      0.79      0.84      3012
           2       0.77      0.68      0.72      3652
           3       0.87      0.50      0.64     13308
           4       0.20      0.54      0.29       199
           5       0.27      0.72      0.39      1734
           6       0.00      0.00      0.00         0
           7       0.43      0.69      0.53       794
           8       0.37      0.43      0.40       375
           9       0.34      0.51      0.41       127
          10       0.52      0.49      0.50       979
          11       0.43      0.47      0.45       996
          12       0.82      0.61      0.70      9059
          13       0.15      0.62      0.24       608
          14       0.28      0.31      0.29       305
          15       0.33      0.41      0.37       186
          16       0.39      0.42      0.41       342
          17       0.62      0.46      0.53      3390
          18       0.00      0.00      0.00         0
          19       0.25      0.76      0.38       535
          20       0.45      0.66      0.54       302
          21       0.50      0.68      0.57      1291
          22       0.36      0.65      0.46      2591

    accuracy                           0.58     43785
   macro avg       0.40      0.50      0.42     43785
weighted avg       0.70      0.58      0.61     43785
```

Figure 3.3: Multinomial NB classification report

3.4 LSTMs Dense Neural Network

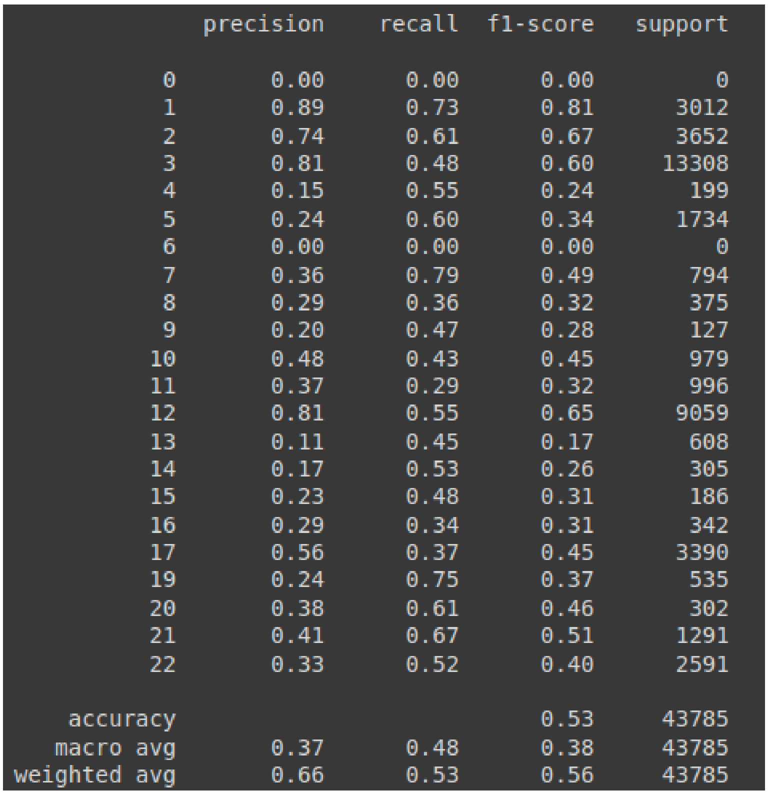
3.4.1 LSTMs Version 1

Any label with count less than 5000 were over-sampled to 5000 and any label with count greater than 10,000 were under-sampled to 10,000 for training in this model. The model summary and the classification report are shown below.



```
_________________________________________________________________
 Layer (type)                Output Shape              Param #   
=================================================================
 embedding_3 (Embedding)     (None, 140, 64)           1920000   
                                                                 
 lstm_3 (LSTM)               (None, 32)                12416     
                                                                 
 dense_6 (Dense)             (None, 128)               4224      
                                                                 
 batch_normalization_3 (Batc  (None, 128)              512       
 hNormalization)                                                 
                                                                 
 dropout_3 (Dropout)         (None, 128)               0         
                                                                 
 dense_7 (Dense)             (None, 23)                2967      
                                                                 
=================================================================
Total params: 1,940,119
Trainable params: 1,939,863
Non-trainable params: 256
_________________________________________________________________
```

Figure 3.4: Model summary for version 1

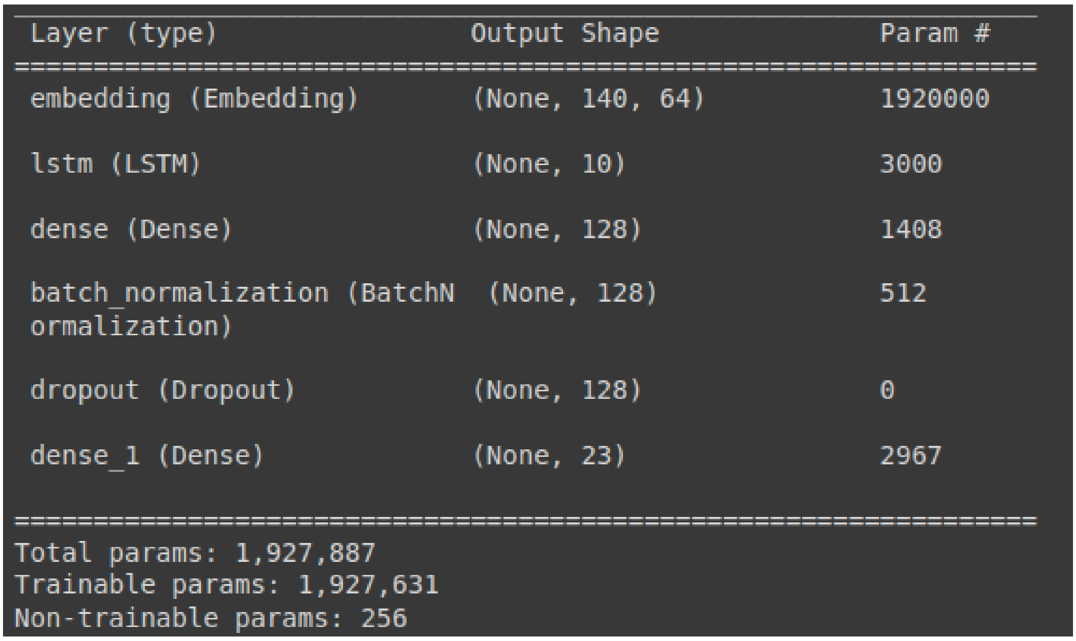


```
              precision    recall  f1-score   support

           0       0.00      0.00      0.00         0
           1       0.89      0.73      0.81      3012
           2       0.74      0.61      0.67      3652
           3       0.81      0.48      0.60     13308
           4       0.15      0.55      0.24       199
           5       0.24      0.60      0.34      1734
           6       0.00      0.00      0.00         0
           7       0.36      0.79      0.49       794
           8       0.29      0.36      0.32       375
           9       0.20      0.47      0.28       127
          10       0.48      0.43      0.45       979
          11       0.37      0.29      0.32       996
          12       0.81      0.55      0.65      9059
          13       0.11      0.45      0.17       608
          14       0.17      0.53      0.26       305
          15       0.23      0.48      0.31       186
          16       0.29      0.34      0.31       342
          17       0.56      0.37      0.45      3390
          19       0.24      0.75      0.37       535
          20       0.38      0.61      0.46       302
          21       0.41      0.67      0.51      1291
          22       0.33      0.52      0.40      2591

    accuracy                           0.53     43785
   macro avg       0.37      0.48      0.38     43785
weighted avg       0.66      0.53      0.56     43785
```

Figure 3.5: Classification report for Version 1

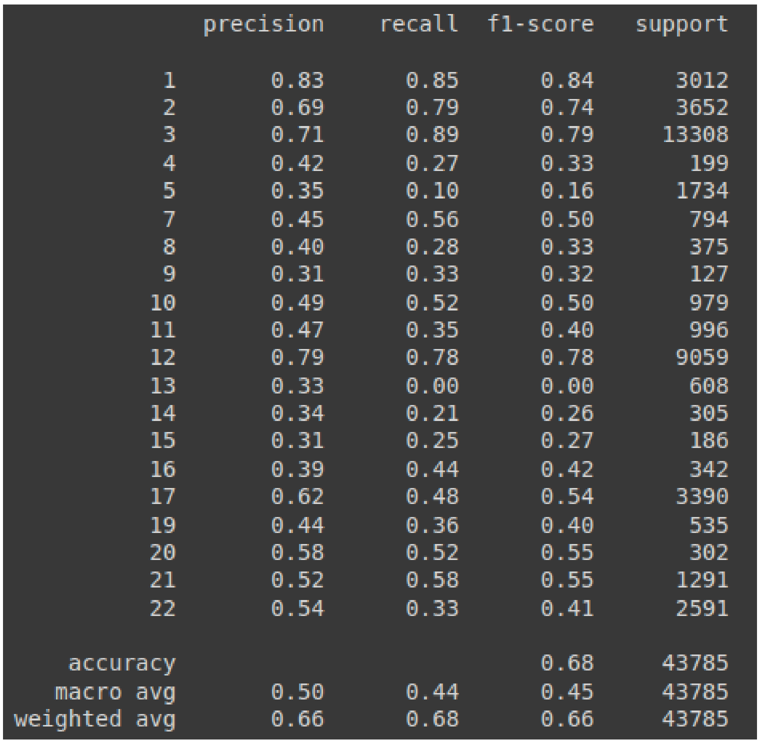
3.4.2 LSTMs Version 2

The model was same as in the Version 1. However, no. of LSTMs unit was reduced from 32 to 10 in this version and the result obtained were as follow. We can observe that the test accuracy improves from 0.53 to 0.68, macro average F1-score from 0.38 to 0.45.



```
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 embedding (Embedding)       (None, 140, 64)           1920000

 lstm (LSTM)                 (None, 10)                3000

 dense (Dense)               (None, 128)               1408

 batch_normalization (BatchN  (None, 128)              512
 ormalization)

 dropout (Dropout)           (None, 128)               0

 dense_1 (Dense)             (None, 23)                2967

=================================================================
Total params: 1,927,887
Trainable params: 1,927,631
Non-trainable params: 256
```

Figure 3.6: Model summary for version 2



```
              precision    recall  f1-score   support

           1       0.83      0.85      0.84      3012
           2       0.69      0.79      0.74      3652
           3       0.71      0.89      0.79     13308
           4       0.42      0.27      0.33       199
           5       0.35      0.10      0.16      1734
           7       0.45      0.56      0.50       794
           8       0.40      0.28      0.33       375
           9       0.31      0.33      0.32       127
          10       0.49      0.52      0.50       979
          11       0.47      0.35      0.40       996
          12       0.79      0.78      0.78      9059
          13       0.33      0.00      0.00       608
          14       0.34      0.21      0.26       305
          15       0.31      0.25      0.27       186
          16       0.39      0.44      0.42       342
          17       0.62      0.48      0.54      3390
          19       0.44      0.36      0.40       535
          20       0.58      0.52      0.55       302
          21       0.52      0.58      0.55      1291
          22       0.54      0.33      0.41      2591

    accuracy                           0.68     43785
   macro avg       0.50      0.44      0.45     43785
weighted avg       0.66      0.68      0.66     43785
```

Figure 3.7: Classification report for Version 2

The graph regarding loss, accuracy, f1-score and fbeta-score are as follow:

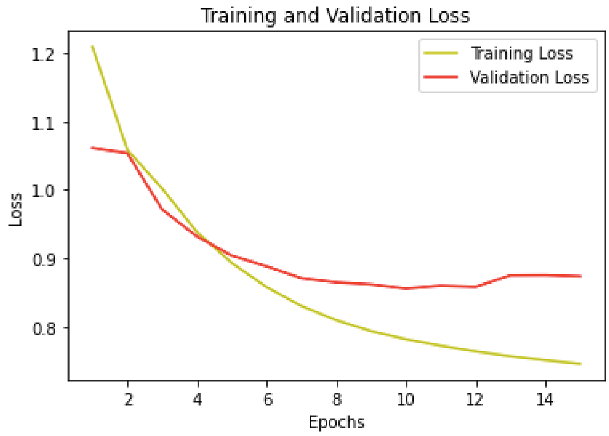


Figure 3.8: LSTMs V2 Loss curve

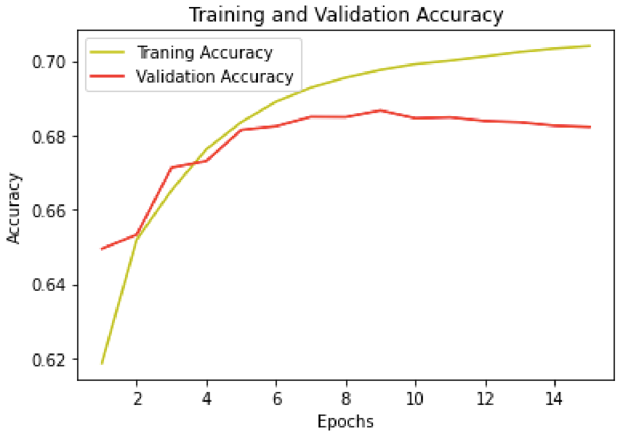


Figure 3.9: LSTMs V2 accuracy curve

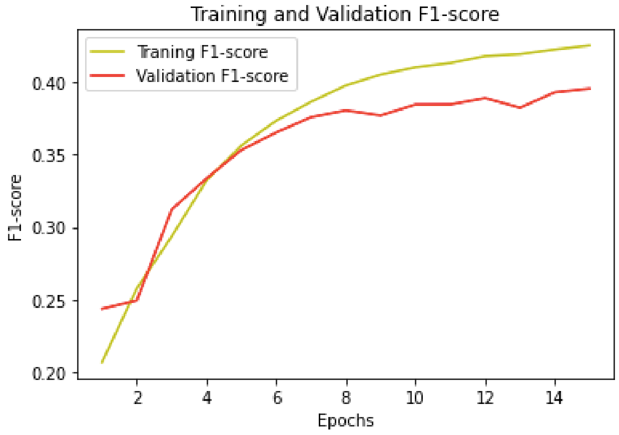


Figure 3.10: LSTMs V2 F1-score curve

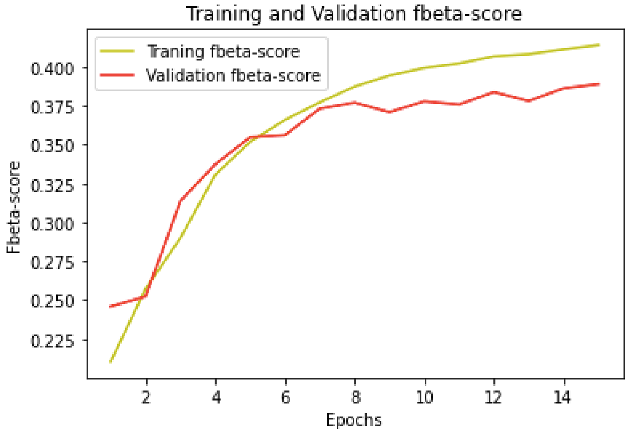


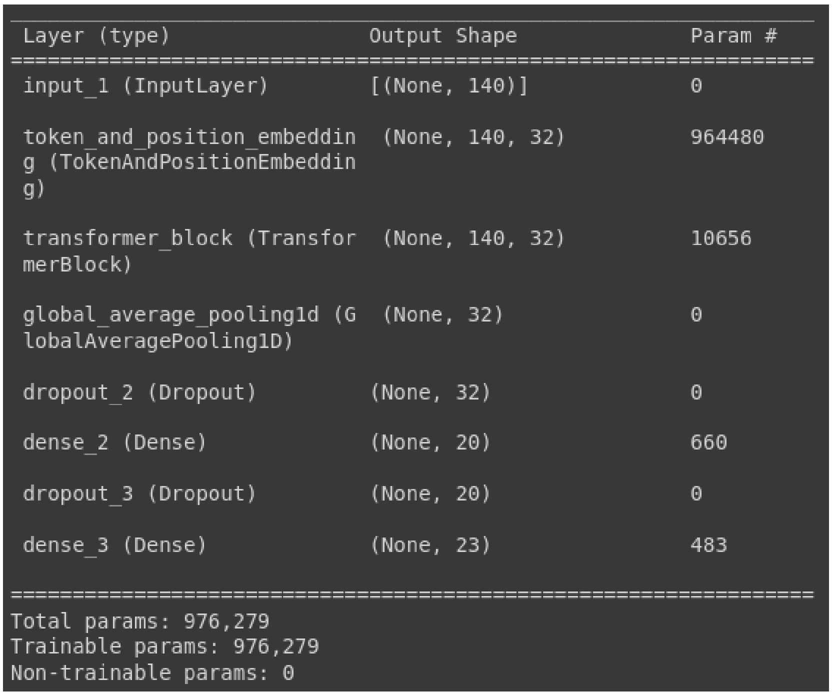
Figure 3.11: LSTMs V2 fbeta-score curve

3.5 Transformer Model

For the test data while training transformer model, we observed that the 32,404 rows in test data are duplicates of the training data. Hence We removed those rows from the test data for creating the classification report.

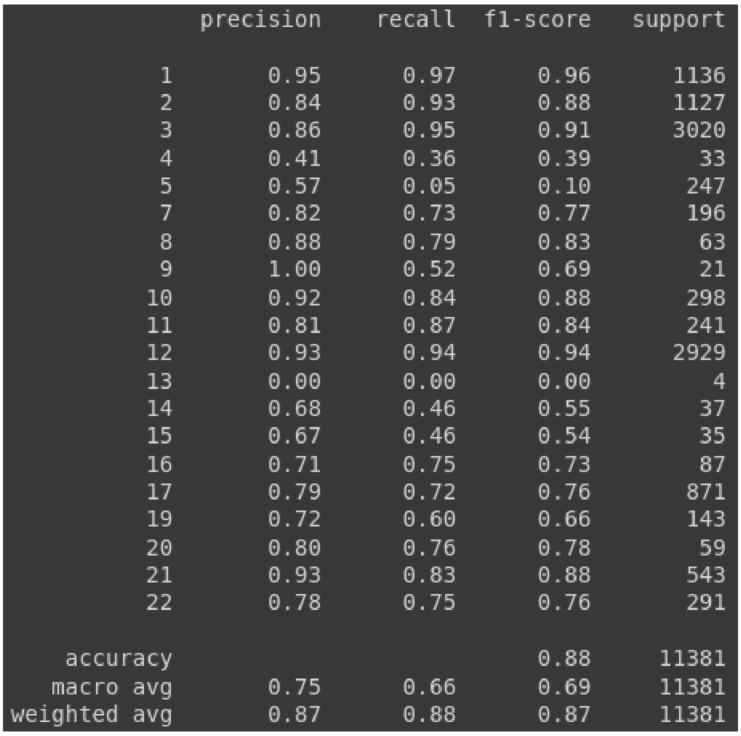
3.5.1 Transformer Version 1

We can observe form the classification report that the accuracy improves to 0.69 and macro average f1-score to 0.46.



```
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 input_1 (InputLayer)        [(None, 140)]             0

 token_and_position_embeddin  (None, 140, 32)          964480
 g (TokenAndPositionEmbeddin
 g)

 transformer_block (Transfor  (None, 140, 32)          10656
 merBlock)

 global_average_pooling1d (G  (None, 32)               0
 lobalAveragePooling1D)

 dropout_2 (Dropout)         (None, 32)                0

 dense_2 (Dense)             (None, 20)                660

 dropout_3 (Dropout)         (None, 20)                0

 dense_3 (Dense)             (None, 23)                483

=================================================================
Total params: 976,279
Trainable params: 976,279
Non-trainable params: 0
```

Figure 3.12: Transformer Model summary for version 1



```
              precision    recall  f1-score   support

           1       0.95      0.97      0.96      1136
           2       0.84      0.93      0.88      1127
           3       0.86      0.95      0.91      3020
           4       0.41      0.36      0.39        33
           5       0.57      0.05      0.10       247
           7       0.82      0.73      0.77       196
           8       0.88      0.79      0.83        63
           9       1.00      0.52      0.69        21
          10       0.92      0.84      0.88       298
          11       0.81      0.87      0.84       241
          12       0.93      0.94      0.94      2929
          13       0.00      0.00      0.00         4
          14       0.68      0.46      0.55        37
          15       0.67      0.46      0.54        35
          16       0.71      0.75      0.73        87
          17       0.79      0.72      0.76       871
          19       0.72      0.60      0.66       143
          20       0.80      0.76      0.78        59
          21       0.93      0.83      0.88       543
          22       0.78      0.75      0.76       291

    accuracy                           0.88     11381
   macro avg       0.75      0.66      0.69     11381
weighted avg       0.87      0.88      0.87     11381
```

Figure 3.13: Transformer Classification report for Version 1

The graph regarding loss, accuracy, f1-score and fbeta-score are as follow:

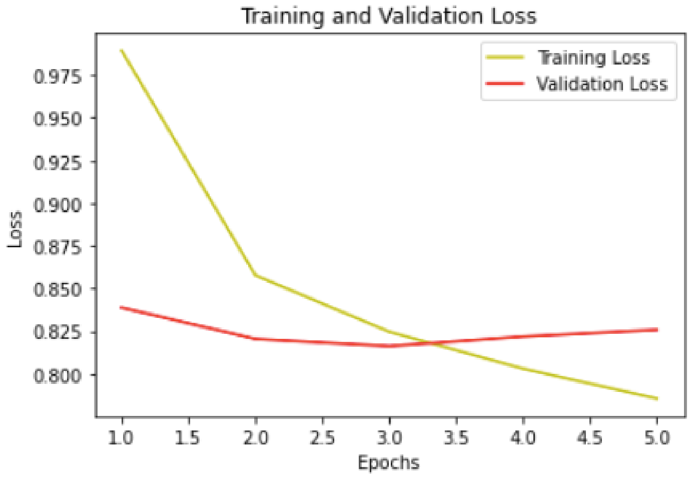


Figure 3.14: Transformer V1 Loss curve

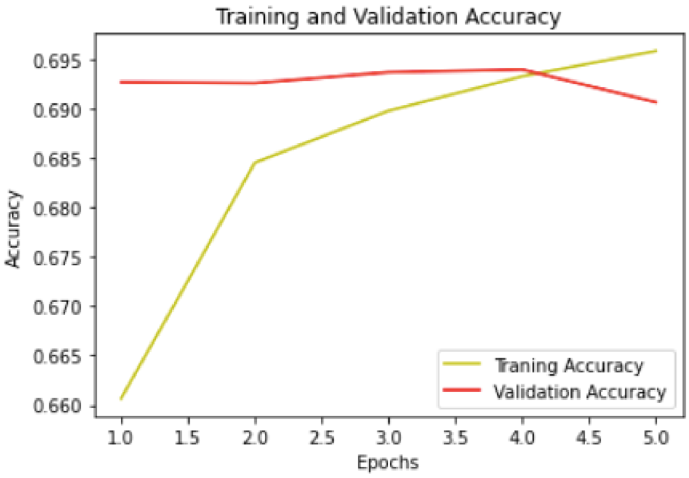


Figure 3.15: Transformer V1 accuracy curve

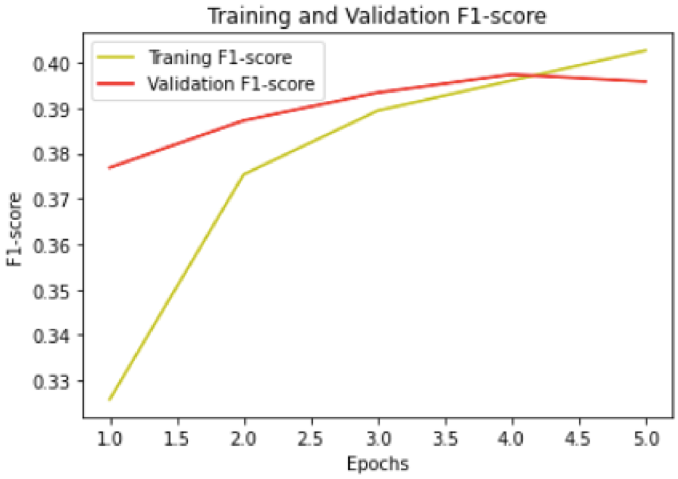


Figure 3.16: Transformer F1-score curve

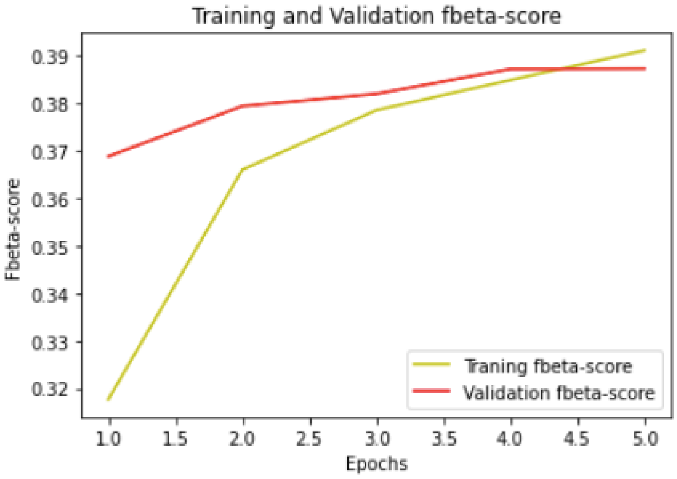
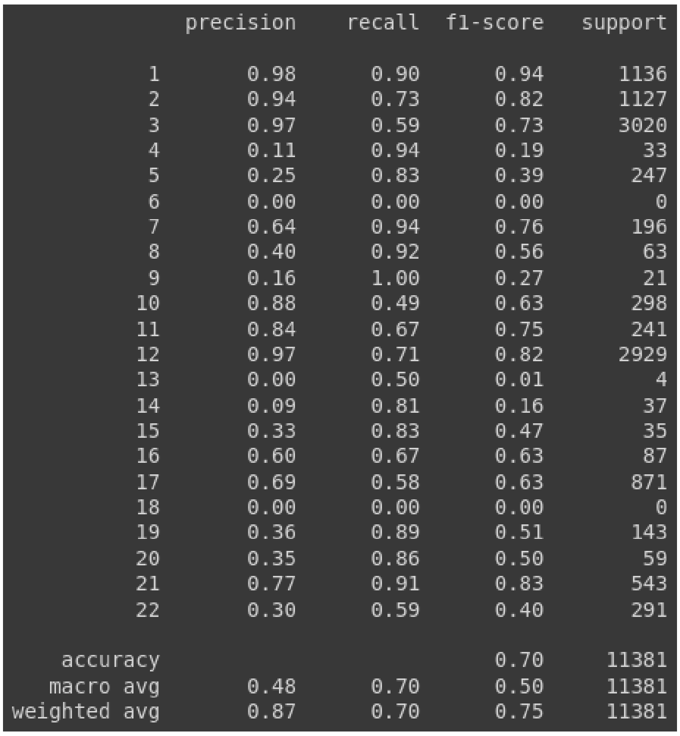


Figure 3.17: Transformer V1 fbeta-score curve

3.5.2 Transformer Version 2

The model summary was same as in transformer version 1. However, for training this model, we oversampled the label with count less than 1000 to 1000 and provided class weights using sklearn.utils.class_weights. The classification report and model history curve are as follow:



```
              precision    recall  f1-score   support

           1       0.98      0.90      0.94      1136
           2       0.94      0.73      0.82      1127
           3       0.97      0.59      0.73      3020
           4       0.11      0.94      0.19        33
           5       0.25      0.83      0.39       247
           6       0.00      0.00      0.00         0
           7       0.64      0.94      0.76       196
           8       0.40      0.92      0.56        63
           9       0.16      1.00      0.27        21
          10       0.88      0.49      0.63       298
          11       0.84      0.67      0.75       241
          12       0.97      0.71      0.82      2929
          13       0.00      0.50      0.01         4
          14       0.09      0.81      0.16        37
          15       0.33      0.83      0.47        35
          16       0.60      0.67      0.63        87
          17       0.69      0.58      0.63       871
          18       0.00      0.00      0.00         0
          19       0.36      0.89      0.51       143
          20       0.35      0.86      0.50        59
          21       0.77      0.91      0.83       543
          22       0.30      0.59      0.40       291

    accuracy                           0.70     11381
   macro avg       0.48      0.70      0.50     11381
weighted avg       0.87      0.70      0.75     11381
```

Figure 3.18: Transformer Classification report for version 2

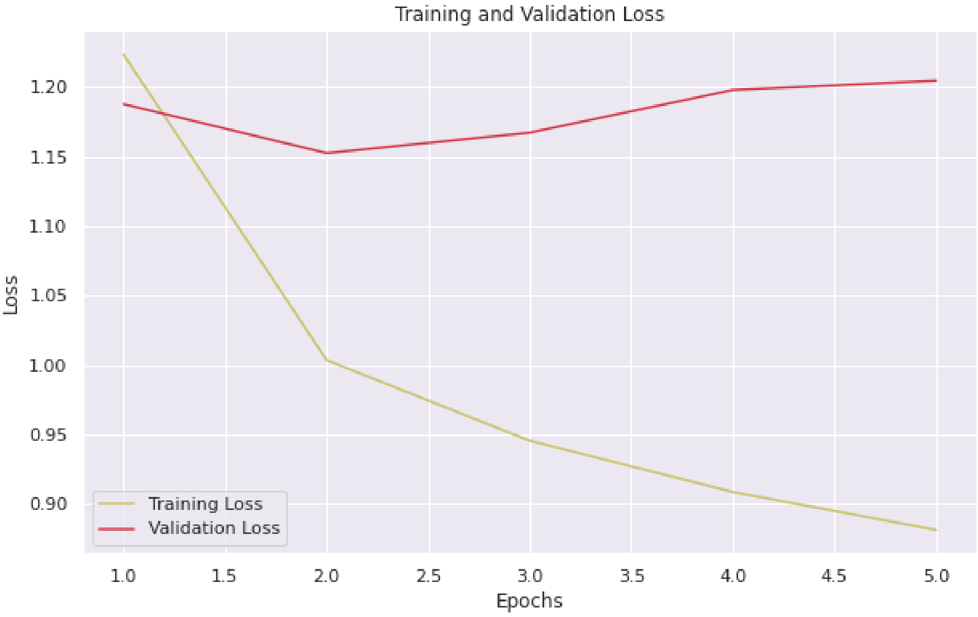


Figure 3.19: Transformer V2 Loss curve

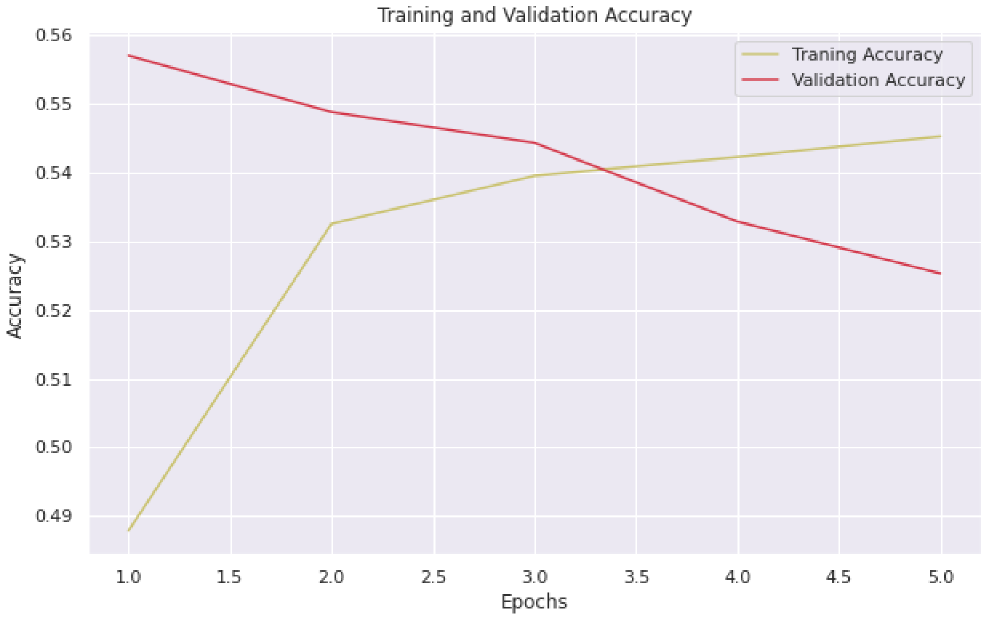


Figure 3.20: Transformer V2 accuracy curve

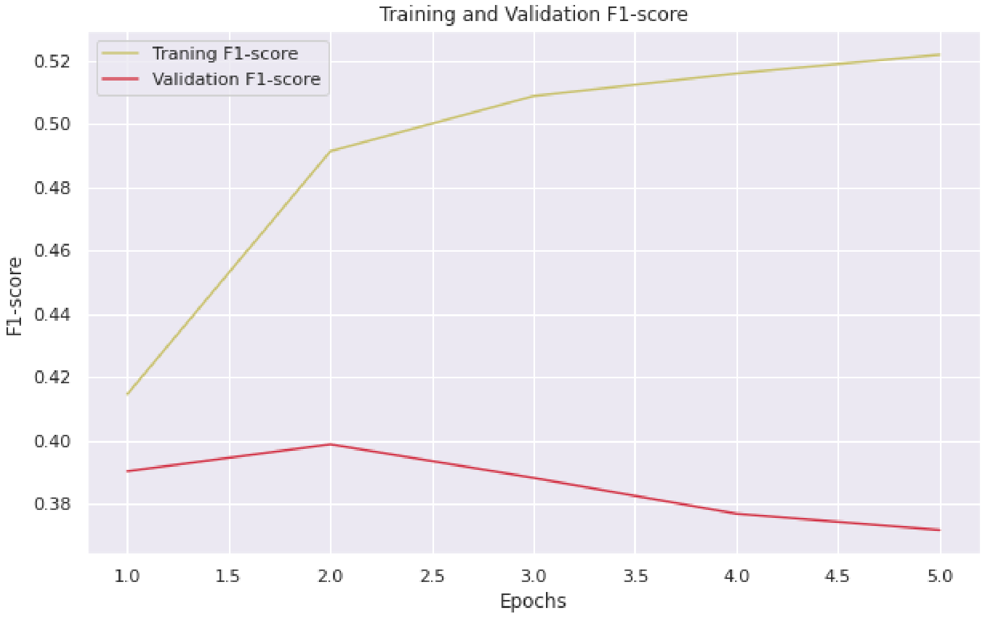


Figure 3.21: Transformer F2-score curve

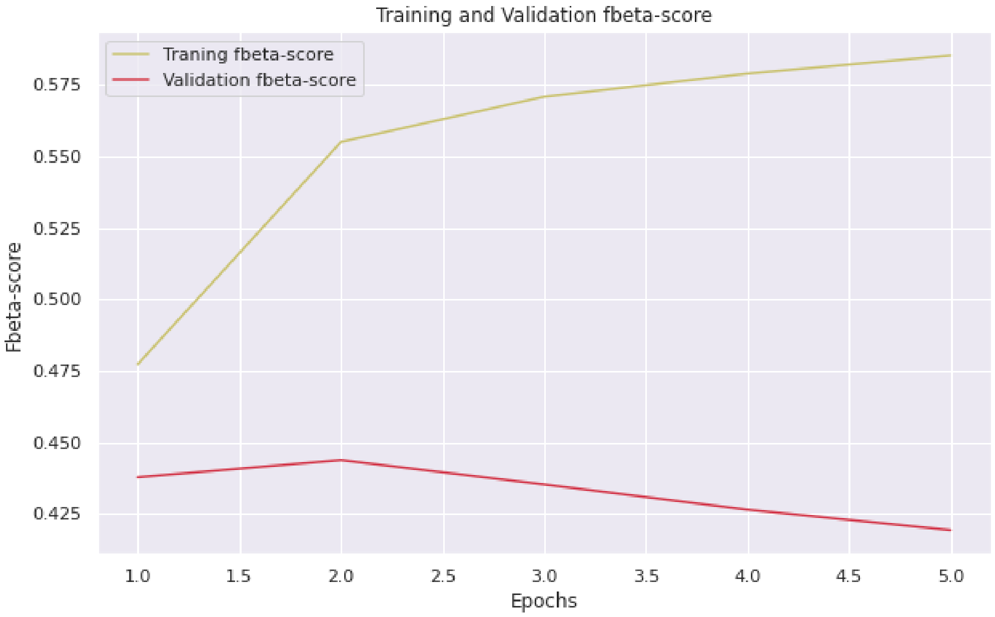
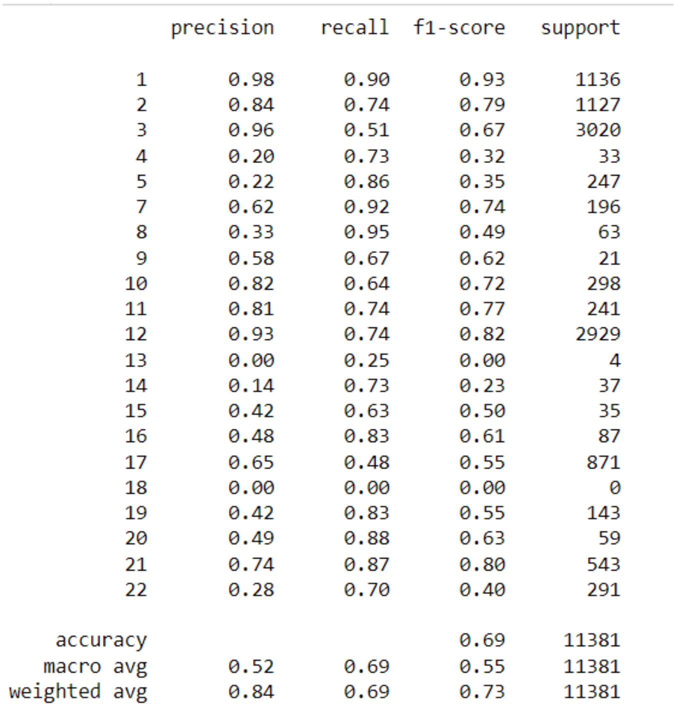


Figure 3.22: Transformer V2 fbeta-score curve

3.5.3 Pretrained BERT Model

For Bert, the pretrained model was taken from hugging face transformers. The sentences were tokenized using pretrained bert tokenizer. The pretrained encoding from model was taken, on top of which 3 layerd multi-layered perceptron of layer_size = (480,360,180) was used with dropout layer(p = 0.1). The classification report was as follow:



```
              precision    recall  f1-score   support

           1       0.98      0.90      0.93      1136
           2       0.84      0.74      0.79      1127
           3       0.96      0.51      0.67      3020
           4       0.20      0.73      0.32        33
           5       0.22      0.86      0.35       247
           7       0.62      0.92      0.74       196
           8       0.33      0.95      0.49        63
           9       0.58      0.67      0.62        21
          10       0.82      0.64      0.72       298
          11       0.81      0.74      0.77       241
          12       0.93      0.74      0.82      2929
          13       0.00      0.25      0.00         4
          14       0.14      0.73      0.23        37
          15       0.42      0.63      0.50        35
          16       0.48      0.83      0.61        87
          17       0.65      0.48      0.55       871
          18       0.00      0.00      0.00         0
          19       0.42      0.83      0.55       143
          20       0.49      0.88      0.63        59
          21       0.74      0.87      0.80       543
          22       0.28      0.70      0.40       291

    accuracy                           0.69     11381
   macro avg       0.52      0.69      0.55     11381
weighted avg       0.84      0.69      0.73     11381
```

Figure 3.23: Bert Classification Report

3.6 Model Comparision

Model Name	Accuracy	Macro Average	Weighted Average
Multinomial NB	0.58	0.42	0.61
SVM	0.36	0.13	0.26
LSTMs V1	0.53	0.38	0.56
LSTMs V2	0.68	0.45	0.66
Transformer V1	.88	.69	.87
Transformer V2	.70	.50	.75
BERT	.69	.55	.73

Table 3.2: Model Comparision based on classification report

From the table 3.2, we found that transformer V1 is better than the transformer V2 and LSTMs V2 is better than LSTMs V1.

In both LSTMs V1 and transformer V2, we did sampling and provided class weights for correcting the class imbalance. However the result was not good. This is because, the test data provided itself was highly imbalanced and biased toward the majority class as in training data which we can observe from the support column of the classification report. Hence, we got better accuracy and f1-score in the model when no sampling and class weights are provided as in the LSTMs V2 and transformer V1.

However, if the test data would have contained minority classes data too, then the result could change.

Chapter 4

Conclusion

Text data analysis helped determining various parameters like maximum padding length and vocab size. Similarly, word cloud, N-gram exploration, Named Entity Recognition Data, Part of Speech Tagging made more clear about the training data that we were working on which eventually improves the model learning capability.

The best model we have is transformer version 1 with accuracy 0.88, macro average(f1-score) of 0.69 and weighted average of 0.87.