EXECUTIVE SUMMARY

The Youtube industry is booming nowadays with billions of videos being uploaded daily. Youtube gives income to the video creators from the revenue collected from advertisements while providing useful information to the subscribers. Some immoral users try to exploit this feature by intentionally uploading videos with misleading titles to generate income. These are called clickbait videos and these create a bad environment for Youtube advertisers, viewers and investors. By collecting a lot of data from youtube videos, this project is going to overcome the issue by detecting clickbait video titles and improving the overall experience of the Youtube system. In our proposed methodology, we have two datasets, one with English videos and one with multilingual video titles to understand the features of clickbait titles in multiple languages. These titles along with numerical features of the video such as the likes, dislikes and views it has received are used to build our classifiers. The ML algorithms that we have used consist of Random Forest, Support Vector Machine(SVM) and Multinomial Naive Bayes which provides a basis for comparison. Transfer learning is a method where a model trained for a particular task is used as the starting point for a model being trained for a different task. The transfer learning model we are using, Bidirectional Encoder Representations from Transformers (BERT) has an advantage over other language models because it is able to gain context of a word from both ends of a sentence and it has been pre trained over a massive dataset. As text is the most important feature in our problem, this is the model we will be implementing.

CONTENTS

Page No.

LIST OF FIGURES

1. INTRODUCTION

1.1. OBJECTIVE

First we are going to discuss the related work in this field to realize what are the past works done and what are the different methodologies used there. This will help us in identifying different ways of transforming the data so that it could be used properly by the machine learning algorithms. After that we will do the Exploratory Data Analysis which will help us understand the dataset. It is followed by the data pre-processing which is an important step as it helps in converting the string data to values understandable by the algorithms. After that the last step is training the ML model and comparing the accuracy and other different results.

1.2. MOTIVATION

There has been exponential growth in the popularity of video-sharing and social media platforms. This has happened due to the increase in the utilization of Web 2.0 Technologies. The social media giants such as Twitter, Instagram and Youtube have the majority of users. Our main focus is Youtube, which is a social media video uploading and streaming platform which is used by billions of users.

1.3 BACKGROUND

Previously much research has been done on this and many Machine Learning algorithms have been implemented to classify between the clickbait and non-clickbait titles of the project. We found out that in many researches they have not done the Exploratory Data Analysis of the dataset and they have not used the BERT classification algorithm to classify the youtube clickbait titles.

2. LITERATURE SURVEY

Authors and Year	Title	Theoretical Model	Dataset details	Methodology	Relevant Findings	Gaps identified
Neha Reddy Vadde, Piyush Gupta, Prasham Mehta, Puneet Gupta, Vikranth BM 2020	Analysis of YouTube Videos: Detecting Click bait on YouTube	The authors have tokenized the data in the pre-processing stage and they are comparing the various classification algorithms using the Receiver Operating Characteristics curve to compare their robustness.	The dataset was extracted from the Youtube API. The dataset contains all the details of the trending Youtube videos along with its likes, dislikes, comments, tags and views for each video for a particular year.	They have divided the process into 3 modules i.e. Comments Characteristics Analysis, Metadata Characteristics Analysis and Supervised Classification. They have trained three models i.e. SVM, LSTM and Random Forest.	In their evaluation they found out that Random Forest is working most effectively followed by SVM and then LSTM. The Random Forest gives an accuracy of 97.14 %, SVM gives 96.76 % and LSTM gives 93.79 %.	They have not done Exploratory Data Analysis to show how the data that they have on different parameters. They didn't try any different deep learning algorithms to find what does the accuracy may come from using that.
Natnicha Wongsap, Lisha Lou, Sasiwimol Jumun,	Thai Clickbait Headline News Classification and its	The authors study the characteristics of Thai clickbait	Two datasets are used- one with the special characters !, ?, # and one without.	Text mining techniques such as TF-IDF and n gram are used in SVM, Decision Tree and	The authors found that there is a high correlation between the	Some of the research gaps in the paper are: 1. A simple percentage of correctly classified headlines is used to

Tastanya Prapphan, Sarawoot Kongyoung, Natsuda Kaothanthong 2018	Characteristic	headlines and the correlation between special punctuations and the clickbait headlines.		Naive Bayes classifiers and the results are compared.	presence of special characters !,?,# in the headline. The accuracy given by the decision tree was highest (99.90%) in this case.	measure the accuracy of this model rather than standard metrics such as F-score.
Suraj Manjesh, Tushar Kanakagiri, Vaishak P, Vivek Chettiar, Shobha G 2017	Clickbait Pattern Detection and Classification of News Headlines using Natural Language Processing	The authors extract syntactic and semantic features of the clickbait titles and use these as input alongside the title as input to the classifiers.	The clickbait headlines used for classification are scraped from websites like BuzzFeed, NewsWeek etc and non clickbait headlines re scraped from sources such as The Hindu, The Guardian, The Economist and so on.	N-gram models were considered from each headline and their TFIDF vectors were formed. The features of the titles such as sentiment, readability score and mean title length were added to the formed vector. Naive Bayes Models and LSTM models were then trained with this vector and accuracy noted.	The authors found that clickbait headlines tend to start with numbers, have a number of dots and contain abbreviations more than non-clickbait headlines. The LSTM method gave better accuracy than The ML	The gap in this article was that the features used as input were restricted to the headline only and did not consider facts such as likes, dislikes,no of reads etc.

					models.	
Mahmud Hasan Munna, Md Shakhawat Hossen 2021	Identification of Clickbait in Video Sharing Platform	The authors create a manually labeled dataset consisting of english and bengali video titles and features such as views, likes, dislikes.	The dataset was prepared by the authors by scraping Youtube and manually labeling the video titles. The dataset consists of 3000 titles, out of which 1571 are clickbait while the rest are non-clickbait.	The numerical data points were scaled and punctuations were removed. Tf Idf was used to create a vector which is then used as input to the following models: SVM, Gradient Boost, Logistic Regression and Decision Tree.	The authors have found that the accuracy achieved were higher in the case where both numerical and text features were used. The highest accuracy was achieved using the Decision Tree and Gradient Boost method.	The gap identified in this paper is that the authors discard punctuations and special characters while preprocessing. As we have studied from our literature survey, punctuations are an important feature while identifying clickbaits.
Prateek Nima 2020	Automatic Filtration of Misleading Youtube Videos using Data Mining Techniques	The author has first extracted the data, after that understood the data. The data pre-processing stage is after that followed	The author had a set of channels which only post videos which are non clickbait and then use the Youtube API to get all the data of those	The author has done the Exploratory Data Analysis to understand the data that has been extracted. He has also done the different pre-processing such	The author found out that there was imbalance in the data so using the random oversampling technique this	The author was not able to take factors such as likes, dislikes, number of comments etc. which are some important features. The author could have also used some more deep learning algorithms

		by Feature Extraction of Data and Random Oversampling and then the Classification models have been trained.	channel's videos. Similarly he had a list of clickbait channels and he got their data using the Youtube API.	as Tokenization of Data, Removal of Stopwords, Removal of Non-Alphabetic Text, Stemming etc. Countvectorizer approach has been used for feature extraction so that only relevant features can be extracted from the data. The author then has used Machine Learning algorithms such as Support Vector Machines (SVM), Decision Tree, Random Forest, Naive Bayes and Convolutional Neural Network.	problem was solved. The best accuracy was given by Convolutional Neural Network (CNN) with an accuracy of 93.23 %. But when the oversampling was done then Random Forest was giving the best accuracy of 94.44 %.	such as Recurrent Neural Network (RNN). The author has also said that he would have done image analysis on the thumbnail of the video to consider the video for clickbait or non clickbait.

3. PROJECT DESCRIPTION AND GOALS

In this project, we are getting the data from kaggle which contains many types of different datasets. The dataset that we have found will serve its purpose sufficiently in this project. Natural Language Processing along with Machine Learning has been used for training and testing purposes. The various algorithms will be briefly discussed in the below sections. Machine learning algorithms used in this project include Multinomial Naive Bayes, Support Vector Machine (SVM), Random Forest and Bidirectional Encoder Representations from Transformers (BERT). Many issues could be curbed from the detection of clickbaits such as:

- Recognition of the video titles which is clickbait.
- The advertisers, which are the main source of income for Youtube, will have more trust in Youtube’s system as this will ensure that advertisements are only shown on the videos which do not have negative impact on the users or viewers.
- As the clickbait video is detected, it would help the Youtube recommendation system to downgrade these types of videos and it will not be recommended to the users, this will ensure that the videos with valid data will get good recommendation.
- The most important thing is this will improve the user experience of the users which is very significant.

4. TECHNICAL SPECIFICATIONS

Google collab with inbuilt GPU.
Libraries of python :

- Scikit Learn
- Numpy
- Pandas
- Seaborn,
- Matplotlib
- Tensorflow

The innovation in our approach is the use of a multilingual dataset of video titles. Using dataset 2 that has headlines in Indian regional languages like Hindi, Malayalam, Tamil, Telugu and Kannada, we plan to identify and analyze the syntactic features of clickbait video titles. The use of transfer learning with BERT to perform classification is also unique. BERT being pre-trained specifically for NLP tasks can be used to solve this problem statement as the input consists of textual data in the form of video titles that need contextual understanding. Thus we expect BERT to perform better than deep learning models like CNN that were used for this task previously as observed in the literature survey.

5. DESIGN APPROACH AND DETAILS

5.1. DESIGN APPROACH

The following steps were performed:

- To form the complete dataset consisting of both clickbait and non-clickbait titles, first a column titled isClickbait to both the clickbait and non clickbait datasets. The isClickbait value was filled with 1 for the clickbait dataset and 0 for non-clickbait dataset. The two csv files were then merged and the rows randomized.
- Excess columns Video_ID and Favorites are dropped
- The text in the video title column was lowercase, extra spaces and stopwords removed. Punctuations are retained as they are important features in clickbait classification as seen in the literature survey.
- The video titles are lemmatized using the WordNetLemmatizer from NLTK library.
- Tf-Idf vectorization is used to convert the title texts into a matrix of numeric values.
- Features of the video title such as no punctuations, mean length of title, views to likes ratio, views to dislikes ratio and sentiment score are extracted for each title and added to the vector as a new feature.
- The dataset is split into train and test sets in a 8:2 ratio.
- The dataset is fit into the ML models one by one and predictions made on the test set. The accuracy scores are noted.

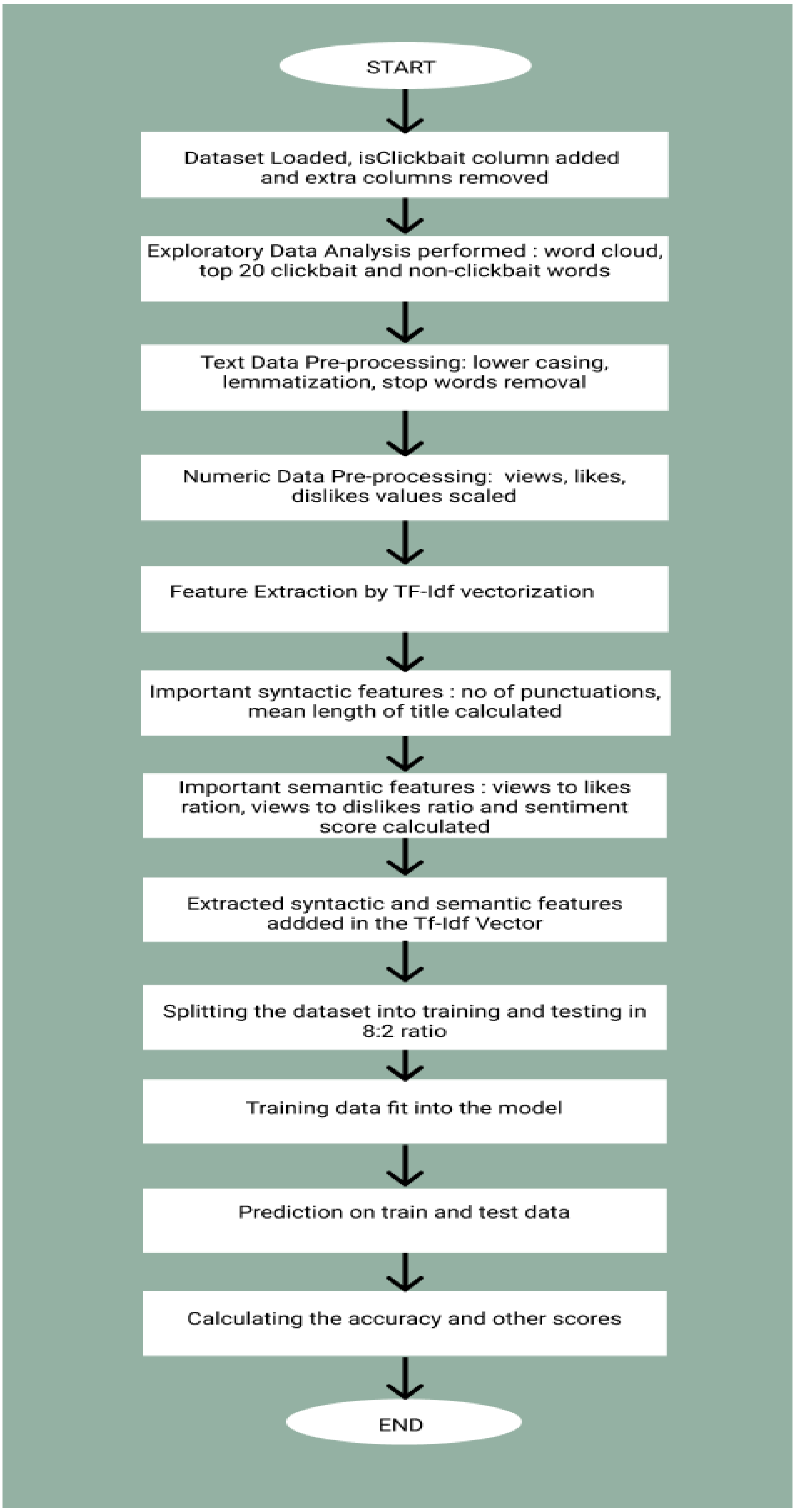

START
Dataset Loaded, isClickbait column added and extra columns removed
Exploratory Data Analysis performed : word cloud, top 20 clickbait and non-clickbait words
Text Data Pre-processing: lower casing, lemmatization, stop words removal
Numeric Data Pre-processing: views, likes, dislikes values scaled
Feature Extraction by TF-Idf vectorization
Important syntactic features : no of punctuations, mean length of title calculated
Important semantic features : views to likes ration, views to dislikes ratio and sentiment score calculated
Extracted syntactic and semantic features addded in the Tf-Idf Vector
Splitting the dataset into training and testing in 8:2 ratio
Training data fit into the model
Prediction on train and test data
Calculating the accuracy and other scores
END

Fig.1. Flow Chart of the Project

In the case of BERT:

- Each lemmatized video title is tokenized and converted to a vector representation using the pretrained tokenizers of BERT.
- A base pretrained model for BERT is imported as a local variable and additional layers according to our problem statement are being added and compiled(transfer learning).
- The vectors are then sent as input to the BERT model
- The output of the BERT model is trained further using the tensorflow Dense layers.
- The models are trained based on this data and used to predict the final classification for the isClickbait column.

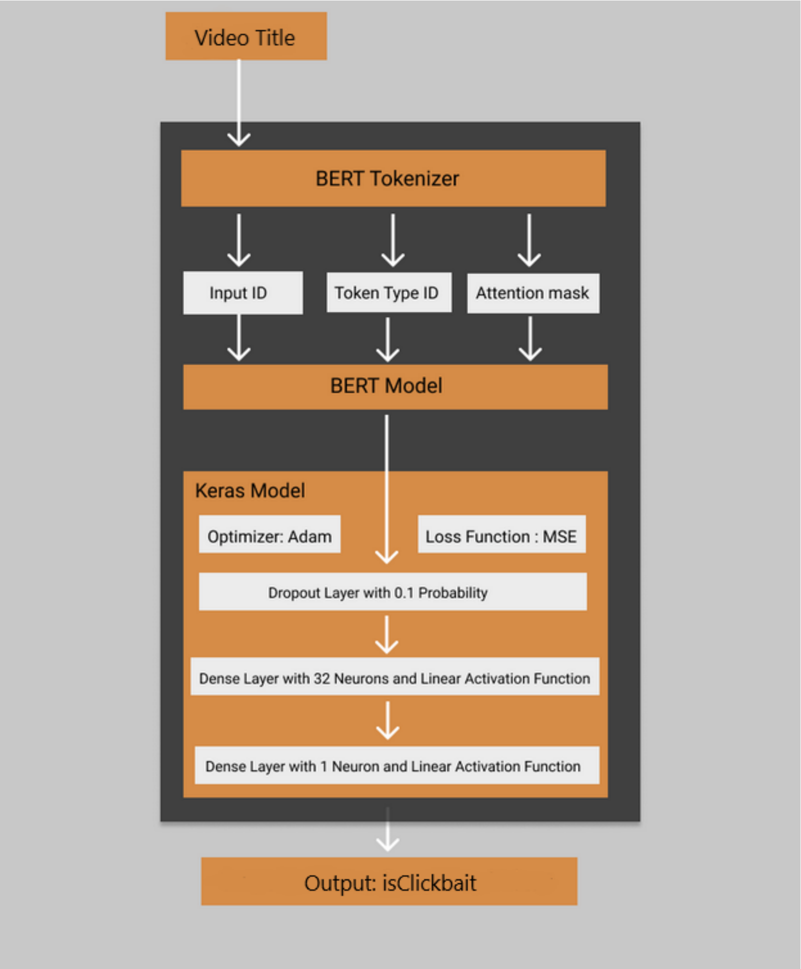

Video Title
BERT Tokenizer
Input ID
Token Type ID
Attention mask
BERT Model
Keras Model
Optimizer: Adam
Loss Function : MSE
Dropout Layer with 0.1 Probability
Dense Layer with 32 Neurons and Linear Activation Function
Dense Layer with 1 Neuron and Linear Activation Function
Output: isClickbait

Fig.2. Details about the BERT Text Classifier

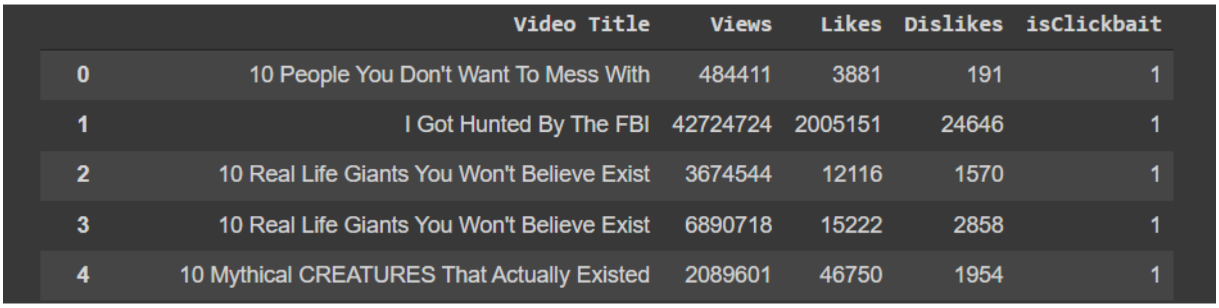
The dataset has been taken from a well known online community for data scientists known as kaggle. There are two different types of datasets that we have used.

The first dataset had the attributes ID, Video Title, Views, Likes, Dislikes and Favorites. There were two csv files for the first dataset. The first csv file consisted of the data which were clickbaits and the second csv file was for the data which were non clickbaits. As these were two different files so we had to import them and then merge the data from both the files. After merging the datas from both the files we randomized the rows so that the clickbait and non clickbait data get mixed properly. This dataset contains more than 32,000 rows of data. This dataset contains youtube titles from various channels available on youtube. In this dataset the features such as ID and Favorites are of no use in training the Machine Learning and Deep Learning models because they are not affecting the result if the video is clickbait or not. So we are going to use the rest of the features such as Video Title, Views, Likes and Dislikes to train the models and then use it to classify other Videos. We added another column named "isClickbait". The clickbait data were put as 1 and non clickbait data were put as 0 in this column and then jumbled.

Columns:

ID, Video Title, Views, Likes, Dislikes, Favorites, isClickbait

Sample from dataset:



	Video Title	Views	Likes	Dislikes	isClickbait
0	10 People You Don't Want To Mess With	484411	3881	191	1
1	I Got Hunted By The FBI	42724724	2005151	24646	1
2	10 Real Life Giants You Won't Believe Exist	3674544	12116	1570	1
3	10 Real Life Giants You Won't Believe Exist	6890718	15222	2858	1
4	10 Mythical CREATURES That Actually Existed	2089601	46750	1954	1

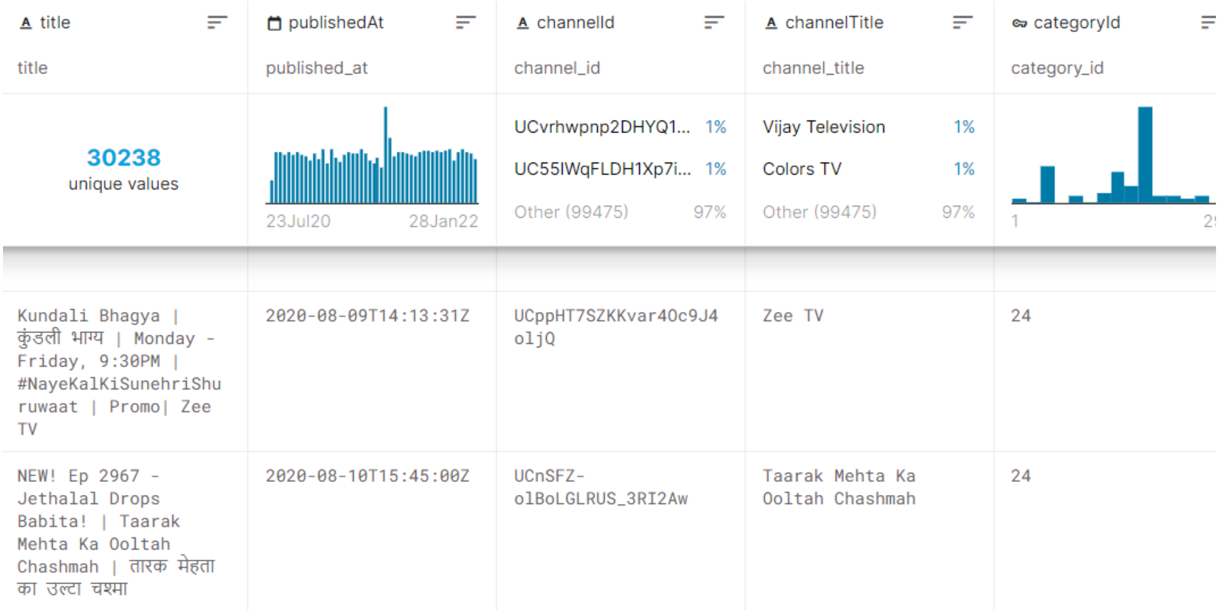
Fig.3. English Video Dataset

The second dataset that we are going to use is updated on a daily basis. It has many more features compared to the first dataset. It has features such as Video ID, Video Title, Published At, Channel ID, Channel Title, Category ID, Trending Date, View Count, Likes, Dislikes, Comment Count and Description. The unique proposition of this dataset is that it has headlines in Hindi, Malayalam, Tamil, Telugu and Kannada, so with this dataset we can even classify titles with multilingual headlines. The features such as Video ID, Published At, Channel ID, Category ID, Trending Date are not useful for training because they are just used as identifiers. The main problem with this dataset is that we have to manually read the rows and classify it as clickbait or non-clickbait. So we have to add another column called "isClickbait" and put all the values there after the manual classification. The useful features that we are going to use are Video Title, Channel Title, View Count, Likes, Dislikes, Comment Count and Description, these will be used to train the different models.

Columns:

Video_ID, Video_Title, Published_At, Channel_ID, Channel_Title, Category_ID, Trending_Date, View_Count, Likes, Dislikes,Comment_Count, Description, isClickbait

Sample from Dataset:



title	publishedAt	channelId	channelTitle	categoryId
title	published_at	channel_id	channel_title	category_id
30238 unique values	23Jul20 28Jan22	UCvrhwpnp2DHYQ1... 1% UC55IWqFLDH1Xp7i... 1% Other (99475) 97%	Vijay Television 1% Colors TV 1% Other (99475) 97%	1 2
Kundali Bhagya | कुंडली भाग्य | Monday - Friday, 9:30PM | #NayeKalKiSunehriShuruwaat | Promo| Zee TV	2020-08-09T14:13:31Z	UCppHT7SZKKvar40c9J4oljQ	Zee TV	24
NEW! Ep 2967 - Jethalal Drops Babita! | Taarak Mehta Ka Ooltah Chashmah | तारक मेहता का उल्टा चश्मा	2020-08-10T15:45:00Z	UCnSFZ-olBoLGLRUS_3RI2Aw	Taarak Mehta Ka Ooltah Chashmah	24

Fig.4. Multi-lingual Video Titles Dataset

5.2. CONSTRAINTS, ALTERNATIVES AND TRADEOFFS

There could be other alternatives to the model that we have used above such as using some other Neural Network that would work more properly with the text data. We have seen that we can modify and stack some ML models such that the accuracy may be increased.

One of the major limitations of the BERT text classification algorithm is lack of ability to handle long text sequences. By default, BERT supports up to 512 tokens. There are multiple ways to overcome it: Ignore text after 512 tokens.

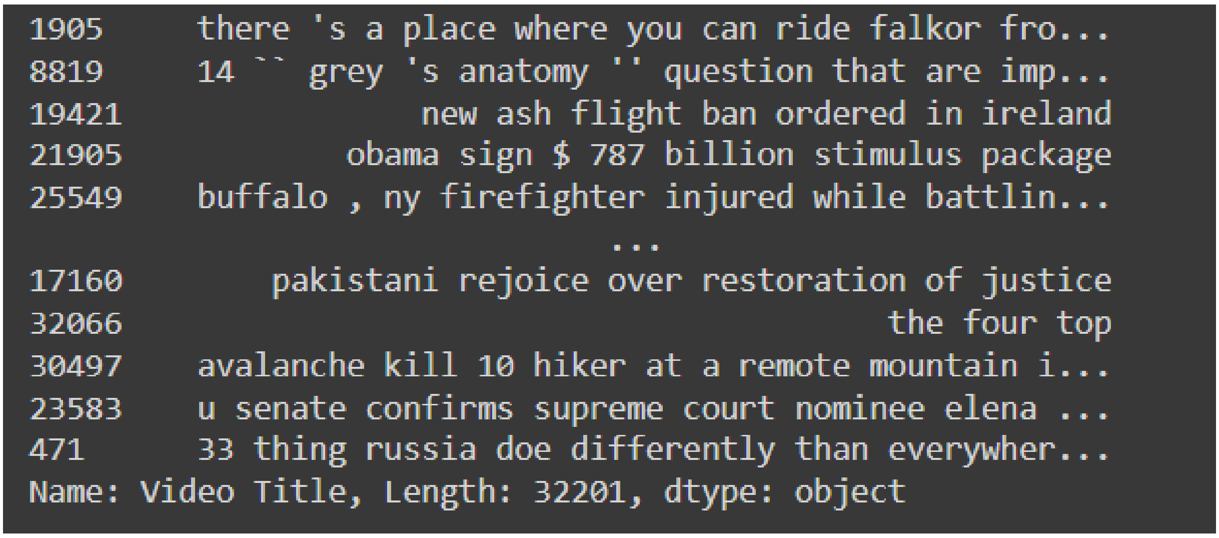
6. SCHEDULE, TASKS AND MILESTONES

SCHEDULE:
Jan-Feb: Find a suitable project title
Feb-Mar: FInding a suitable dataset and data preprocessing
Mar-Apr: Implementing the chosen model and analyzing results

MILESTONES:
Milestone 1: Finding the proper dataset for the model
Milestone 2: Annotation of the dataset
Milestone 3: Text preprocessing and data cleaning
Milestone 4: Implementing the chosen ML models
Milestone 5: Implementing the chosen NN model
Milestone 6: Comparing results

7. RESULTS AND DISCUSSIONS

7.1. Preprocessing Results: Lower Casing, Stop Word Removal and Lemmatization



```
1905     there 's a place where you can ride falkor fro...
8819     14 `` grey 's anatomy '' question that are imp...
19421                new ash flight ban ordered in ireland
21905            obama sign $ 787 billion stimulus package
25549    buffalo , ny firefighter injured while battlin...
                              ...
17160        pakistani rejoice over restoration of justice
32066                                         the four top
30497    avalanche kill 10 hiker at a remote mountain i...
23583    u senate confirms supreme court nominee elena ...
471      33 thing russia doe differently than everywher...
Name: Video Title, Length: 32201, dtype: object
```

Fig.5. Pre processing Results

7.2. Exploratory Data Analysis Results

a. Missing Numbers in Dataset

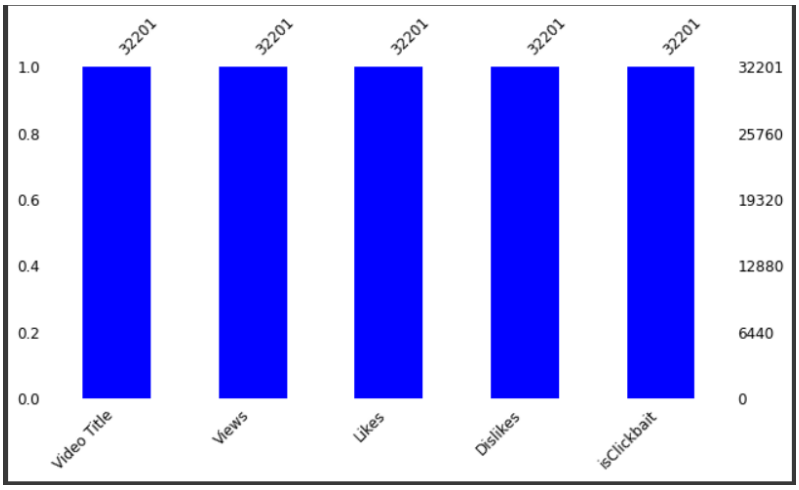


Fig.6.

b. Word Cloud of Video Titles

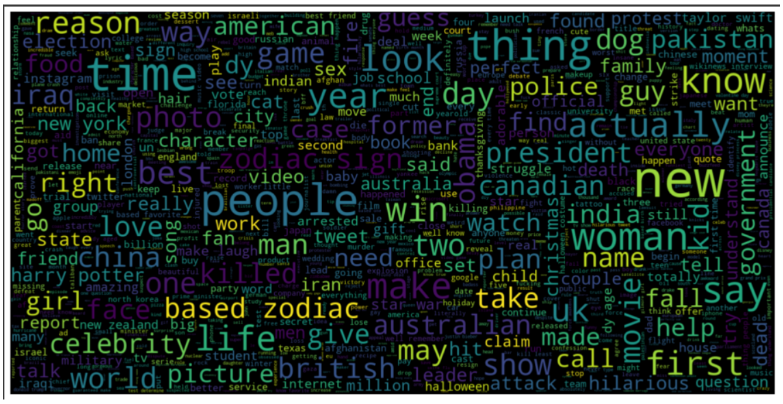


Fig.7.

c. Top 20 Unigrams

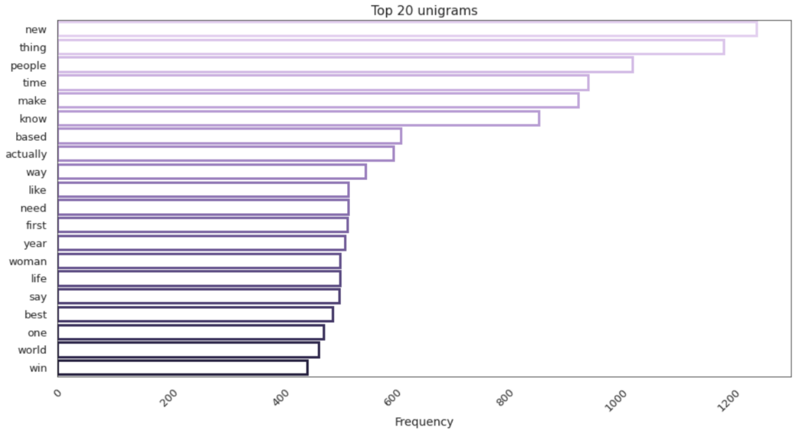


Fig.8.

d. Top 20 Bigrams

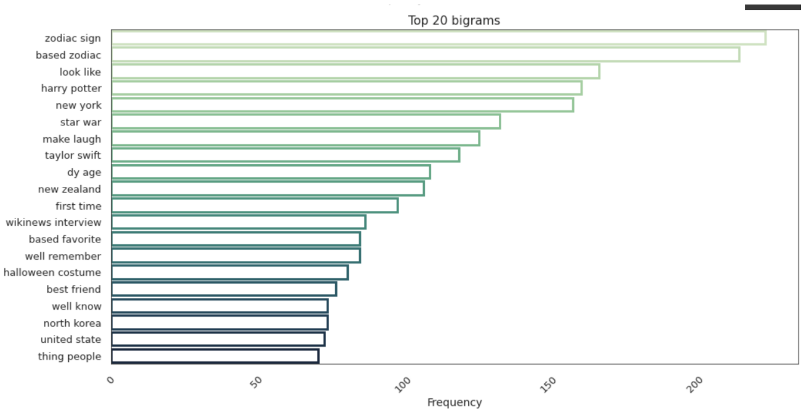


Fig.9.

e. Top 20 Trigrams

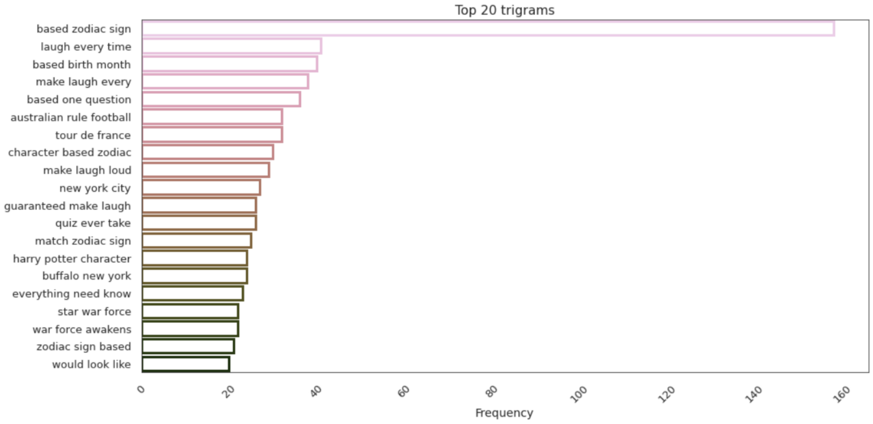


Fig.10.

f. Target Distribution

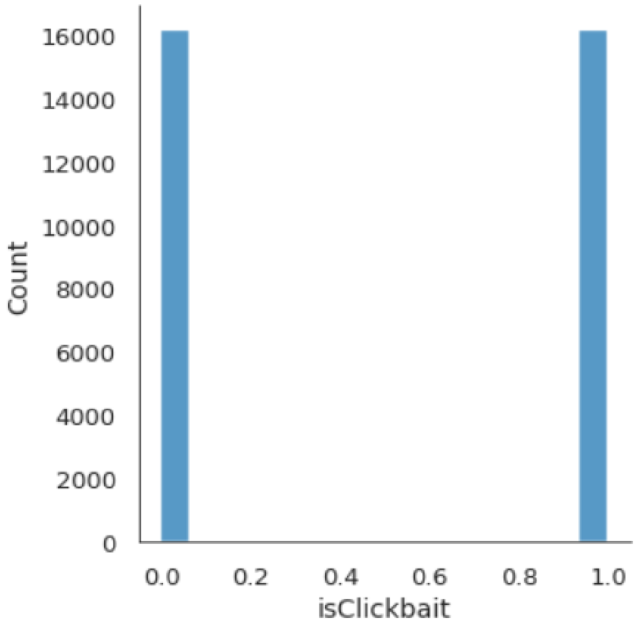


Fig.11.

g. Top 20 Clickbait Headline Words

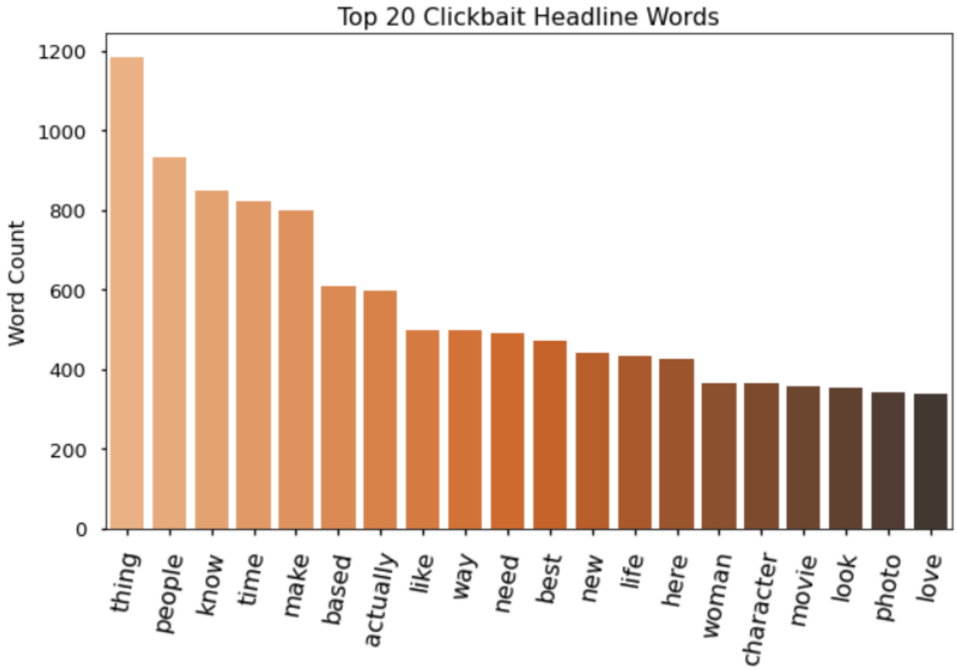


Fig.12.

h. Top 20 Non-Clickbait Headline Words

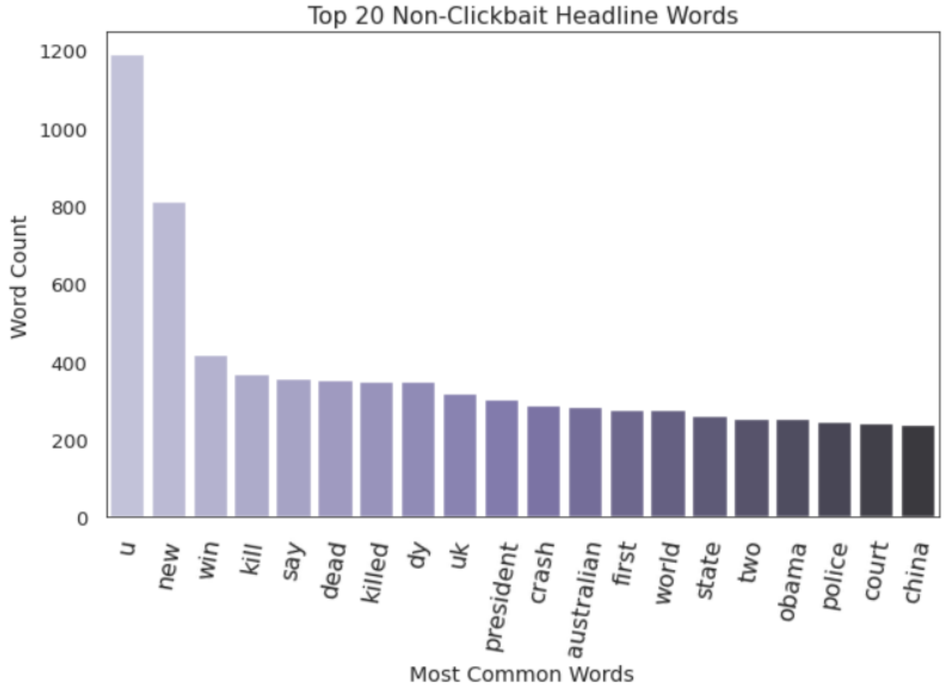
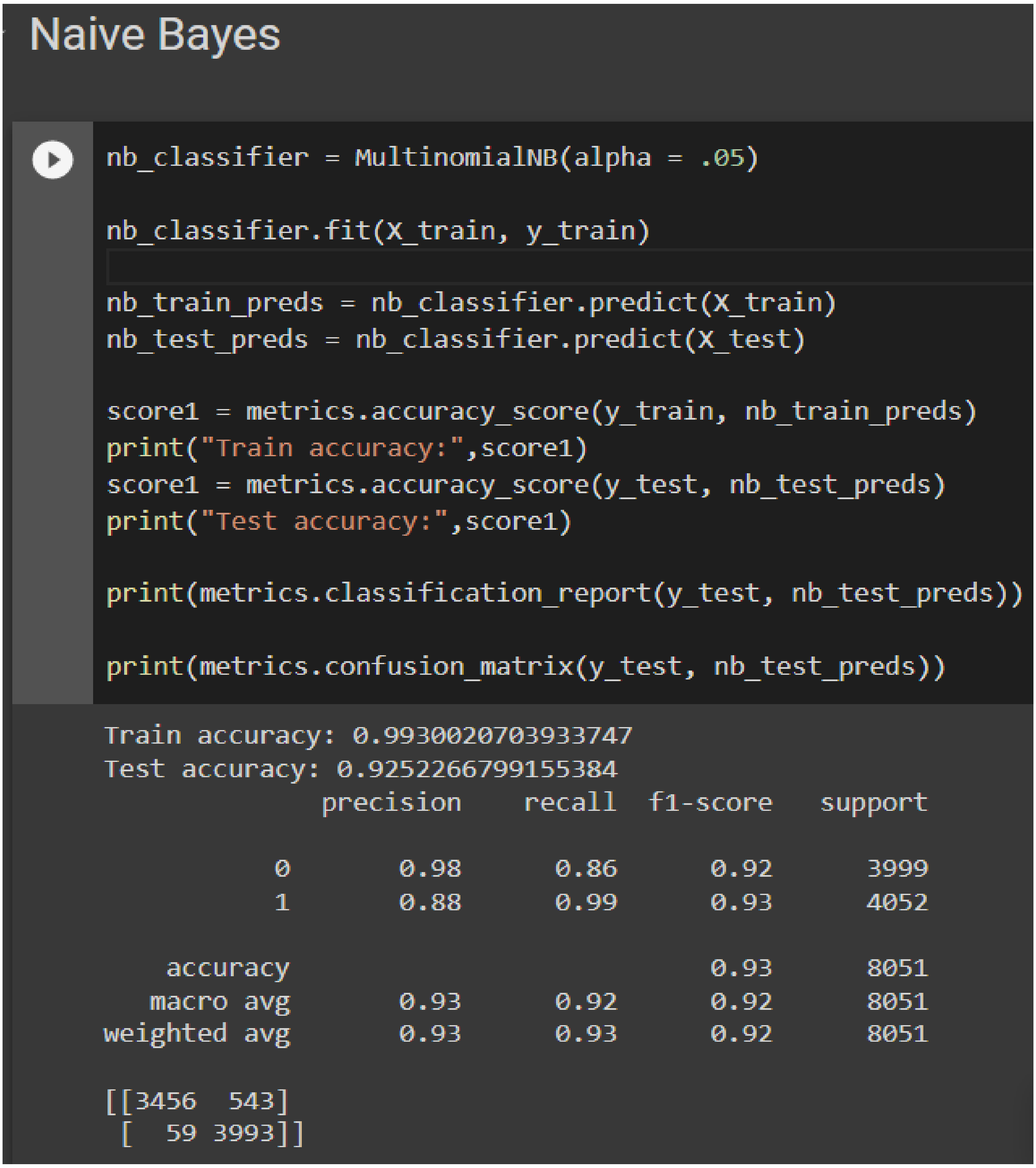


Fig.13.

Final Expectation : The final output for each row of data is a binary label of 0 referring to the title being non-clickbait and 1 referring to the title being clickbait
We expect an accuracy above 95% for the english video titles dataset and above 90% for the multilingual video titles dataset.

Naive Bayes



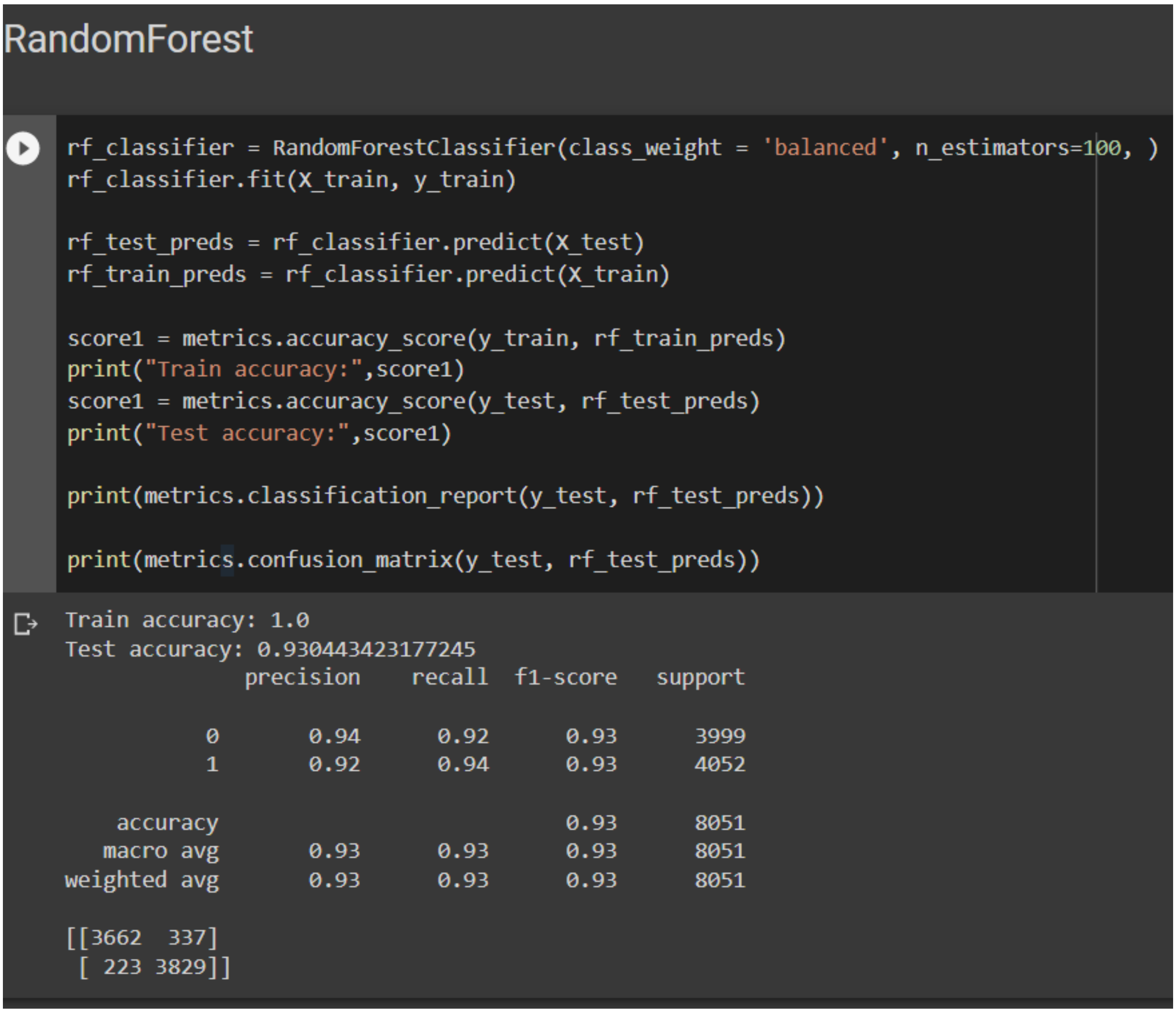
Naive Bayes

```
nb_classifier = MultinomialNB(alpha = .05)

nb_classifier.fit(X_train, y_train)

nb_train_preds = nb_classifier.predict(X_train)
nb_test_preds = nb_classifier.predict(X_test)

score1 = metrics.accuracy_score(y_train, nb_train_preds)
print("Train accuracy:",score1)
score1 = metrics.accuracy_score(y_test, nb_test_preds)
print("Test accuracy:",score1)

print(metrics.classification_report(y_test, nb_test_preds))

print(metrics.confusion_matrix(y_test, nb_test_preds))
```

```
Train accuracy: 0.9930020703933747
Test accuracy: 0.9252266799155384
              precision    recall  f1-score   support

           0       0.98      0.86      0.92      3999
           1       0.88      0.99      0.93      4052

    accuracy                           0.93      8051
   macro avg       0.93      0.92      0.92      8051
weighted avg       0.93      0.93      0.92      8051

[[3456  543]
 [  59 3993]]
```

Fig.14. Naive Bayes Classifier Results

Random Forest



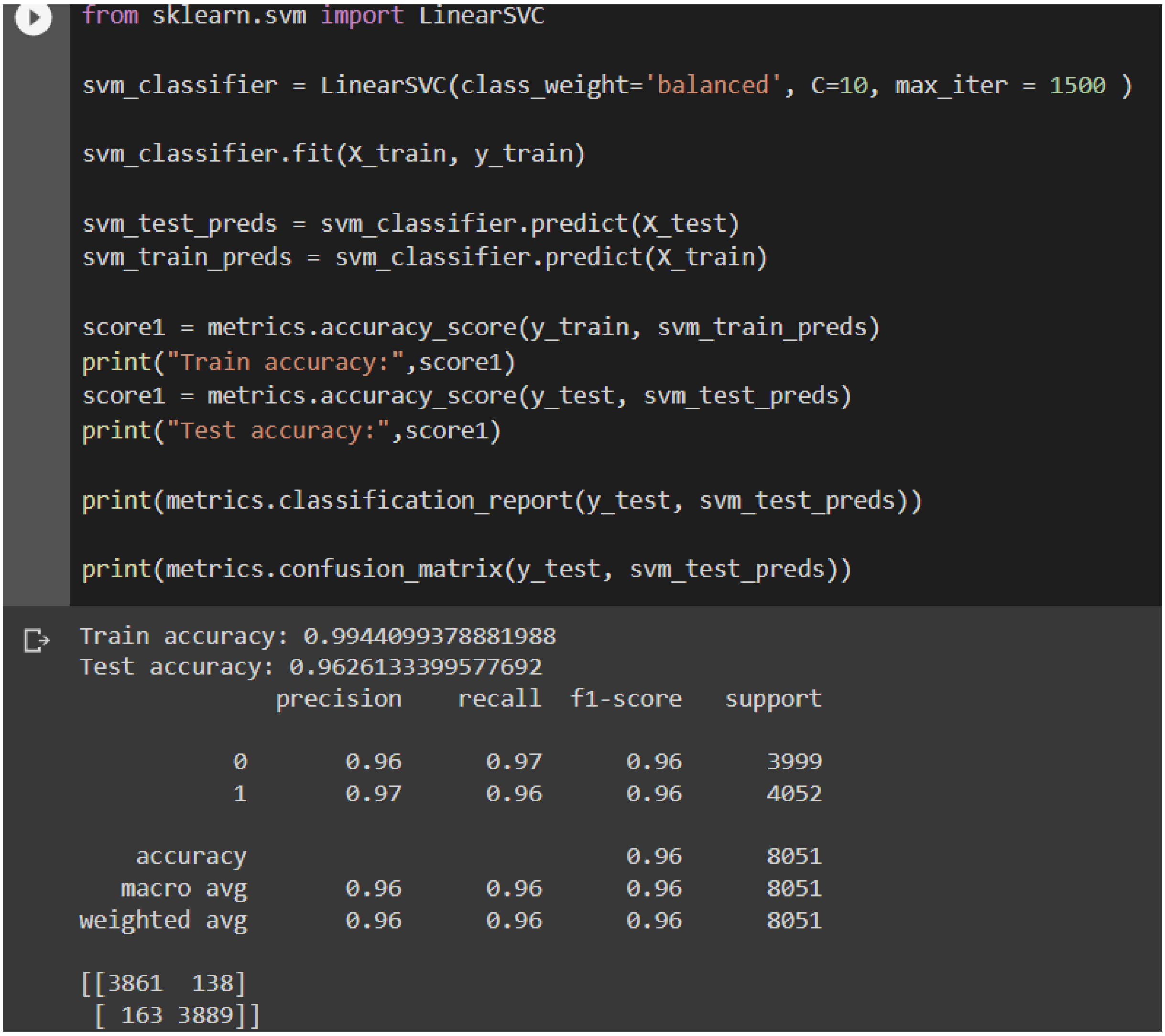
RandomForest

```
rf_classifier = RandomForestClassifier(class_weight = 'balanced', n_estimators=100, )
rf_classifier.fit(X_train, y_train)

rf_test_preds = rf_classifier.predict(X_test)
rf_train_preds = rf_classifier.predict(X_train)

score1 = metrics.accuracy_score(y_train, rf_train_preds)
print("Train accuracy:",score1)
score1 = metrics.accuracy_score(y_test, rf_test_preds)
print("Test accuracy:",score1)

print(metrics.classification_report(y_test, rf_test_preds))

print(metrics.confusion_matrix(y_test, rf_test_preds))
```

```
Train accuracy: 1.0
Test accuracy: 0.930443423177245
              precision    recall  f1-score   support

           0       0.94      0.92      0.93      3999
           1       0.92      0.94      0.93      4052

    accuracy                           0.93      8051
   macro avg       0.93      0.93      0.93      8051
weighted avg       0.93      0.93      0.93      8051

[[3662  337]
 [ 223 3829]]
```

Fig.15. Random Forest Classifier Results

SVM

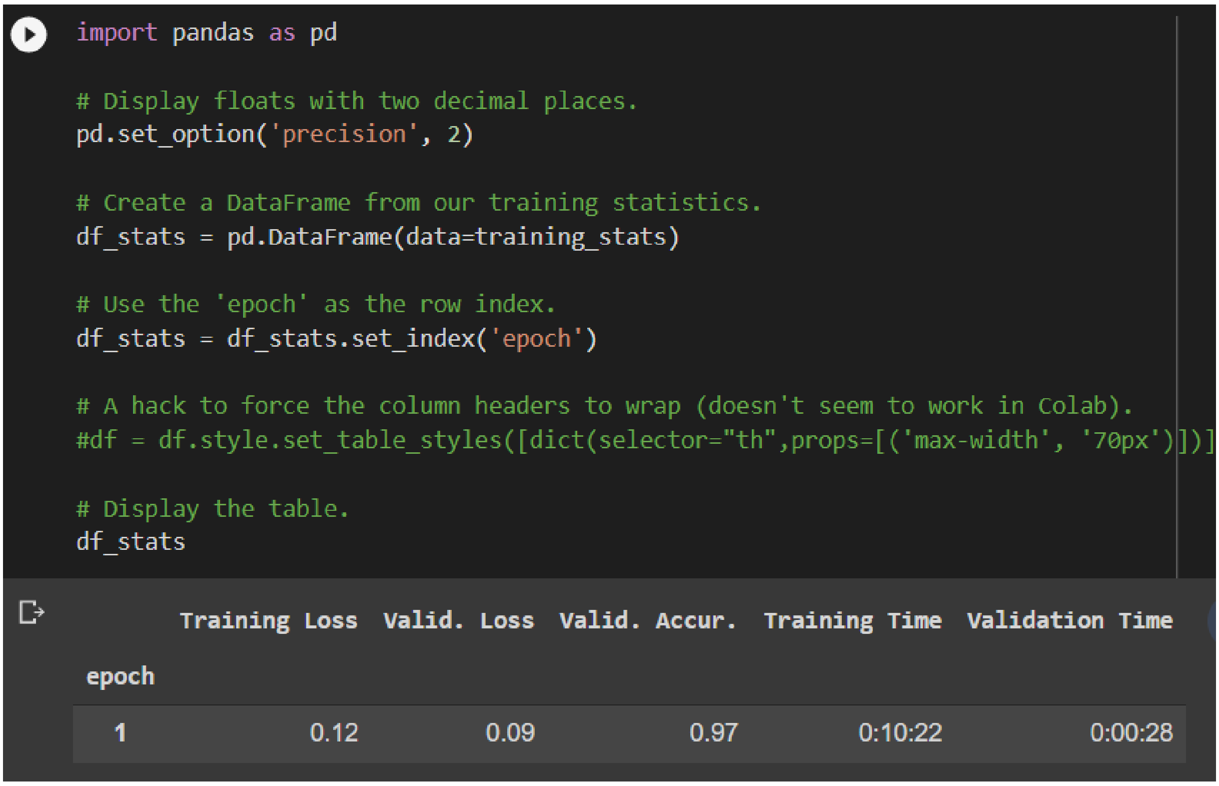


```
from sklearn.svm import LinearSVC

svm_classifier = LinearSVC(class_weight='balanced', C=10, max_iter = 1500 )

svm_classifier.fit(X_train, y_train)

svm_test_preds = svm_classifier.predict(X_test)
svm_train_preds = svm_classifier.predict(X_train)

score1 = metrics.accuracy_score(y_train, svm_train_preds)
print("Train accuracy:",score1)
score1 = metrics.accuracy_score(y_test, svm_test_preds)
print("Test accuracy:",score1)

print(metrics.classification_report(y_test, svm_test_preds))

print(metrics.confusion_matrix(y_test, svm_test_preds))
```

```
Train accuracy: 0.9944099378881988
Test accuracy: 0.9626133399577692
              precision    recall  f1-score   support

           0       0.96      0.97      0.96      3999
           1       0.97      0.96      0.96      4052

    accuracy                           0.96      8051
   macro avg       0.96      0.96      0.96      8051
weighted avg       0.96      0.96      0.96      8051

[[3861  138]
 [ 163 3889]]
```

Fig.16. SVM Classifier Results

XG Boost

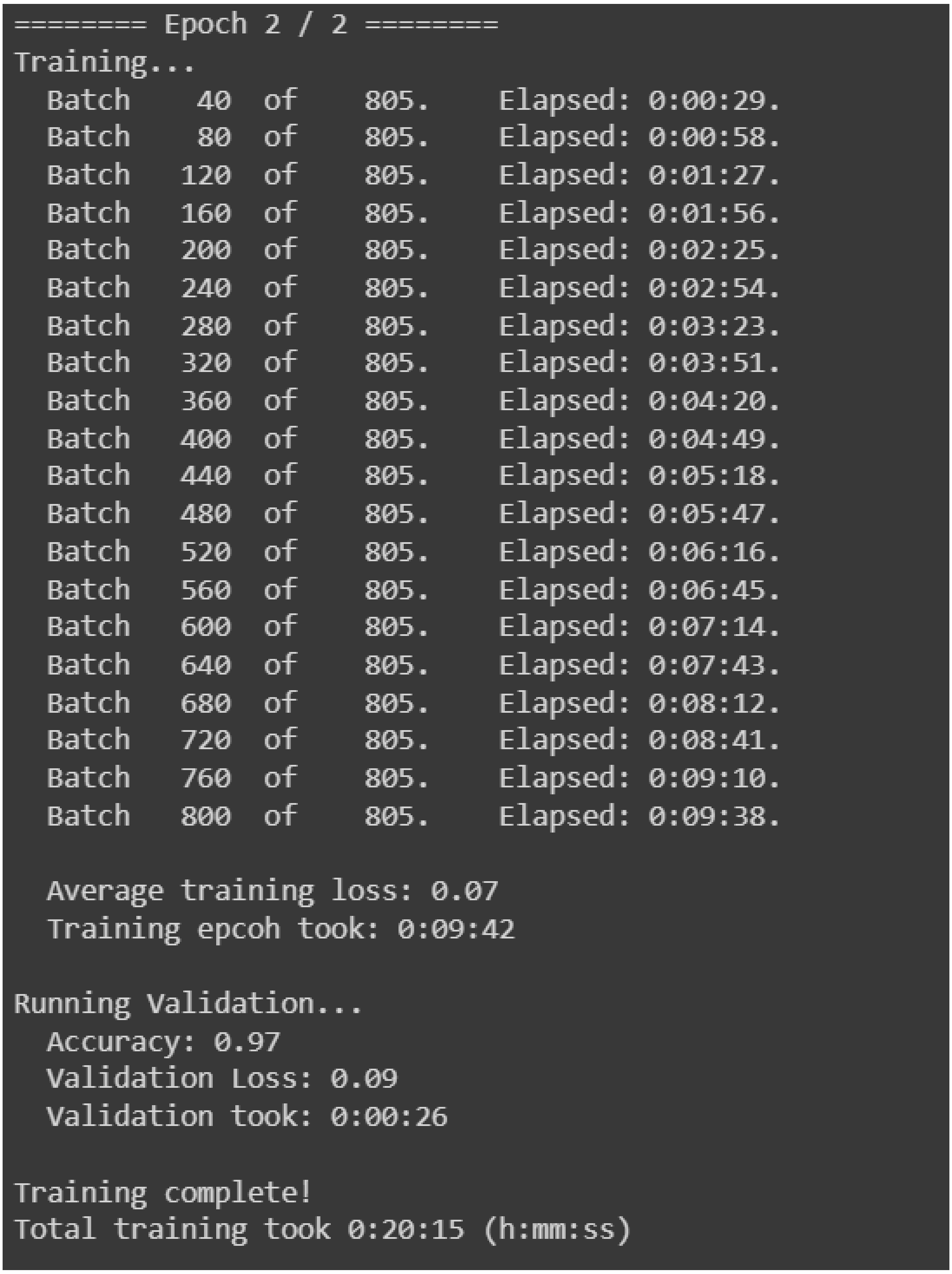


```
import pandas as pd

# Display floats with two decimal places.
pd.set_option('precision', 2)

# Create a DataFrame from our training statistics.
df_stats = pd.DataFrame(data=training_stats)

# Use the 'epoch' as the row index.
df_stats = df_stats.set_index('epoch')

# A hack to force the column headers to wrap (doesn't seem to work in Colab).
#df = df.style.set_table_styles([dict(selector="th",props=[('max-width', '70px')])]

# Display the table.
df_stats
```

```
       Training Loss  Valid. Loss  Valid. Accur.  Training Time  Validation Time
epoch
  1         0.12          0.09          0.97          0:10:22          0:00:28
```

Fig.17. Feature Extraction using XG Boost

BERT



```
======== Epoch 2 / 2 ========
Training...
  Batch    40  of    805.    Elapsed: 0:00:29.
  Batch    80  of    805.    Elapsed: 0:00:58.
  Batch   120  of    805.    Elapsed: 0:01:27.
  Batch   160  of    805.    Elapsed: 0:01:56.
  Batch   200  of    805.    Elapsed: 0:02:25.
  Batch   240  of    805.    Elapsed: 0:02:54.
  Batch   280  of    805.    Elapsed: 0:03:23.
  Batch   320  of    805.    Elapsed: 0:03:51.
  Batch   360  of    805.    Elapsed: 0:04:20.
  Batch   400  of    805.    Elapsed: 0:04:49.
  Batch   440  of    805.    Elapsed: 0:05:18.
  Batch   480  of    805.    Elapsed: 0:05:47.
  Batch   520  of    805.    Elapsed: 0:06:16.
  Batch   560  of    805.    Elapsed: 0:06:45.
  Batch   600  of    805.    Elapsed: 0:07:14.
  Batch   640  of    805.    Elapsed: 0:07:43.
  Batch   680  of    805.    Elapsed: 0:08:12.
  Batch   720  of    805.    Elapsed: 0:08:41.
  Batch   760  of    805.    Elapsed: 0:09:10.
  Batch   800  of    805.    Elapsed: 0:09:38.

  Average training loss: 0.07
  Training epcoh took: 0:09:42

Running Validation...
  Accuracy: 0.97
  Validation Loss: 0.09
  Validation took: 0:00:26

Training complete!
Total training took 0:20:15 (h:mm:ss)
```

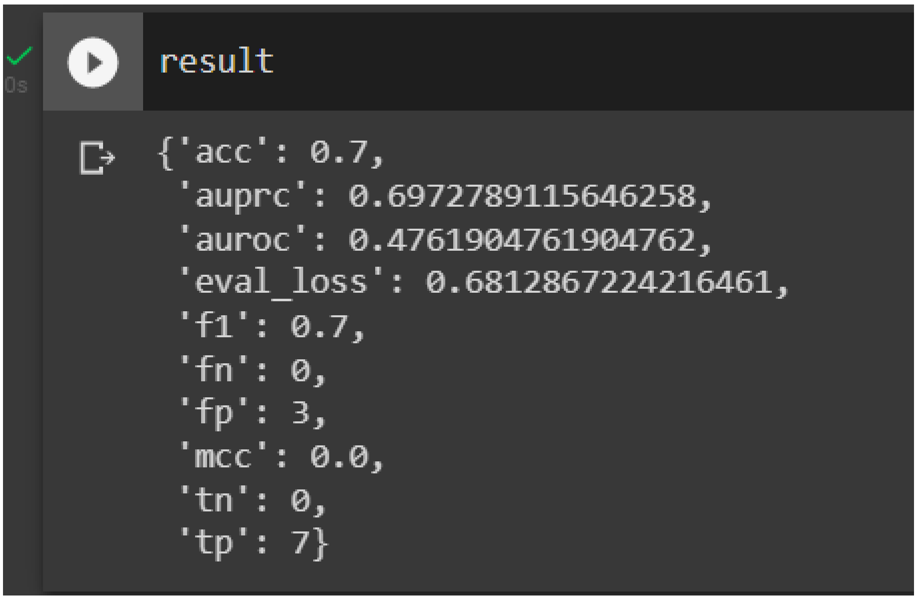
Fig.18. Training the BERT Model

Comparison of results :

Model	*Train Accuracy*	*Test Accuracy*
Naive Bayes	0.99	0.92
Random Forest	0.1	0.93
SVM	0.99	0.96
BERT	0.97	0.975

Table.1. Result Comparison

INDIAN LANGUAGES DATASET



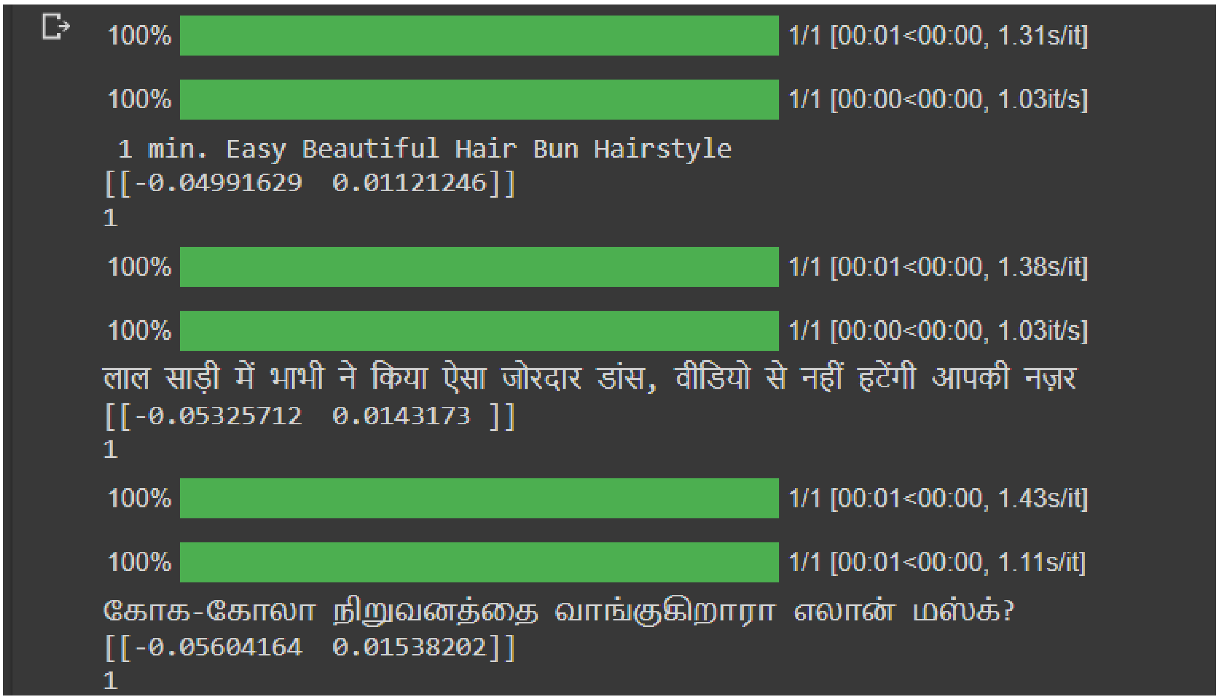
```
result
```

```
{'acc': 0.7,
 'auprc': 0.6972789115646258,
 'auroc': 0.47619047619047 62,
 'eval_loss': 0.6812867224216461,
 'f1': 0.7,
 'fn': 0,
 'fp': 3,
 'mcc': 0.0,
 'tn': 0,
 'tp': 7}
```

Fig.19. Result from the Indian Language Dataset

As seen above, the accuracy of prediction is 70% on the Indian languages youtube dataset. This is almost at par with the state of the art results obtained by training mBERT

RESULT OF CLASSIFICATIONS



```
100%  1/1 [00:01<00:00, 1.31s/it]

100%  1/1 [00:00<00:00, 1.03it/s]
 1 min. Easy Beautiful Hair Bun Hairstyle
[[-0.04991629  0.01121246]]
1

100%  1/1 [00:01<00:00, 1.38s/it]

100%  1/1 [00:00<00:00, 1.03it/s]
लाल साड़ी में भाभी ने किया ऐसा जोरदार डांस, वीडियो से नहीं हटेंगी आपकी नज़र
[[-0.05325712  0.0143173 ]]
1

100%  1/1 [00:01<00:00, 1.43s/it]

100%  1/1 [00:01<00:00, 1.11s/it]
கோக-கோலா நிறுவனத்தை வாங்குகிறாரா எலான் மஸ்க்?
[[-0.05604164  0.01538202]]
1
```

Fig.20. Manual Classification Results on Indian Language Trained Model

As shown above we have correctly classified three clickbait titles in English, Hindi and Tamil.

9. SUMMARY

The results that we have got from the BERT text classification algorithm is better than other Machine Learning algorithms. The testing accuracy for BERT was 97% in the case of the English Language dataset and 70% in the case of the Indian Language dataset. For the Machine Learning algorithms such as Naive Bayes, Random Forest and SVM, the testing accuracy was 92%, 93% and 96% respectively.

10. REFERENCES

1. Vadde, Neha Reddy, et al. "Analysis of YouTube Videos: Detecting Click bait on YouTube."
2. Wongsap, Natnicha, et al. "Thai clickbait headline news classification and its characteristics." *2018 International Conference on Embedded Systems and Intelligent Technology & International Conference on Information and Communication Technology for Embedded Systems (ICESIT-ICICTES)*. IEEE, 2018.
3. Manjesh, Suraj, et al. "Clickbait pattern detection and classification of news headlines using natural language processing." *2017 2nd International Conference on Computational Systems and Information Technology for Sustainable Solution (CSITSS)*. IEEE, 2017.
4. Munna, Mahmud Hasan, and Md Shakhawat Hossen. "Identification of Clickbait in Video Sharing Platforms." *2021 International Conference on Automation, Control and Mechatronics for Industry 4.0 (ACMI)*. IEEE, 2021.
5. Nima, Prateek. "Automatic Filtration of Misleading Youtube Videos using Data Mining Techniques." *National College of Ireland 26p* (2020).