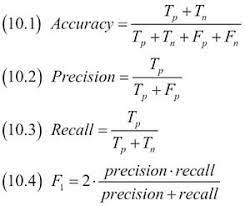
YouTube Popularity Analysis: Project Source Code Documentation

This file contains a list of all files submitted, and an explanation for the use of the same. Instructions to reproduce the results of each file are mentioned as well

# List of files Submitted

* YouTube - Cleaning.ipynb
* Youtube Baselines-MLNB.ipynb
* Youtub\_baselines\_lstm.ipynb
* Youtube - distilBERT, epoch=2.ipynb
* Youtube - DistilBERT training (1 epoch)
* ReadMe – Documentation.docx
* PDF Folder – pdf versions of all codes

# Metrics used

As this problem is a binary classification problem the metrics used to describe the results of the file are the following:

* Accuracy score
* Precision score
* Recall score
* F1 score
* Confusion matrix – For better visualizations

# File wise Documentation

YouTube - Cleaning.ipynb

* This is the file used to clean the data and store it into a new “Youtube-Cleaned.csv” file.
* Initial data can be downloaded from Kaggle – Trending YouTube Video statistics: <https://www.kaggle.com/datasnaek/youtube-new>
* Change the path to the US and CN .csv files in the python notebook to the path of the files on your machine. (There is a comment in the code for the same)
  + E.g., USFile= pd.read\_csv(<Your\_path\_here>)
* This IPYNB file removes punctuations of the data, and duplicates between US and CN files.
* It also has code to remove punctuations and tokenize the data.
* Lastly the data is stemmed and Lemmatized
* The label is also created as a binary function of the view count -> for every title , if the view counts is greater than the median view count of the dataset it is considered popular i.e. label=1
* Results from this file: Youtube-Cleaned csv
  1. Youtube Baselines-MLNB.ipynb
* This is the file uses data from “Youtube-Cleaned.csv” file for Multinomial Naïve Bayes prediction
* Change the path to the “Youtube-Cleaned.csv” files in the python notebook to the path of the files on your machine. (There is a comment in the code for the same)
  + E.g., USFile= pd.read\_csv(<Your\_path\_here>)
* This IPYNB file utilizes Tf-idf vectorizer in a pipeline with sklearn.naive\_bayes’ Multinomial naïve bayes model
  + TFIDF or term frequency inverse document frequency is used from sklearn , with parameter smooth\_idf =True, This ensures that if there is a word which is not seen before the inverse document frequency value will be 1 and not 0
  + Multinomial Naïve bayes is often used for word-frequency related tasks like this one. It was used from sklearn
* Results from this file: Accuracy, Classification report and confusion matrix (Reported in results.docx)
  1. Youtube\_baselines\_lstm.ipynb
* Here lstm from tensorflow.keras.layers was used , this ipynb file was also run on Google Colab , hence downloading tensorflow is not necessary.
* Change the path to the “Youtube-Cleaned.csv” files in the python notebook to the path of the files on your machine. (There is a comment in the code for the same)
  + E.g., USFile= pd.read\_csv(<Your\_path\_here>)
* This ipynb file uses a on hot encoding for each word in the corpus , and then runs an lstm with a dropout of =0.3 .It utilizes a sigmoid function to return the binary output
* Lastly results are seen from the test set are : Accuracy, Classification report and confusion matrix (Reported in results.docx)
  1. Youtube - distilBERT, epoch=2
* Environment /equivalent needed to run this file:
  + `transformers` version: 4.18.0
  + - Platform: Windows-10-10.0.19041-SP0
  + - Python version: 3.8.8
  + - Huggingface\_hub version: 0.5.1
  + - PyTorch version (GPU?): 1.11.0+cu113
* Change the path to the “Youtube-Cleaned.csv” files in the python notebook to the path of the files on your machine. (There is a comment in the code for the same)
  + E.g., USFile= pd.read\_csv(<Your\_path\_here>)
* This ipynb file trainer the huggingface distilbert base model, it also utilizes the distilbert tokenizer for the same
* This also utilizes huggingface’s dataset to split the tokenized dataset into train test and validation sets 80-10-10
* The trainer class is used to train the dataset’s train split and run evaluations on the validation split
* Lastly results are seen from the test set are : Accuracy, Classification report and confusion matrix (Reported in results.docx)
  1. Youtube – DistilBERT training (1 epoch)
* This model is exactly the same as the above but was trained on 1 epoch , it has lesser accuracy as shown in results doc , further instructions are the same as above .
* Environment /equivalent needed to run this file:
  + `transformers` version: 4.18.0
  + - Platform: Windows-10-10.0.19041-SP0
  + - Python version: 3.8.8
  + - Huggingface\_hub version: 0.5.1
  + - PyTorch version (GPU?): 1.11.0+cu113
* Change the path to the “Youtube-Cleaned.csv” files in the python notebook to the path of the files on your machine. (There is a comment in the code for the same)
  + E.g., USFile= pd.read\_csv(<Your\_path\_here>)
* This ipynb file trainer the huggingface distilbert base model, it also utilizes the distilbert tokenizer for the same
* This also utilizes huggingface’s dataset to split the tokenized dataset into train test and validation sets 80-10-10
* The trainer class is used to train the dataset’s train split and run evaluations on the validation split
* Lastly results are seen from the test set are : Accuracy, Classification report and confusion matrix (Reported in results.docx)
  1. PDF Folder
* This contains all the ipynb in pdf format as backup, contains all code and results as well