Assignment

- **Scraping youtube data**

➢ I used youtube data api. First I created developer key(Now changed due to security reasons) in google console
➢ Reference - https://developers.google.com/youtube/v3/docs/
➢ Created youtube_search function which finds data till last page.
➢ I took 2000+ sample of each category by kernel interruption
➢ Saved individual category data to individul csv
➢ Finally merged all data into file named all_category_data_merged.csv in which there are two columns text and category.
➢ Text column has video description and category has label names

---

Text Classification

Please look at **jupyter notebook** for all outputs and graphs.
Python file is also attached.

➢ Imported neccessary libraries like numpy,sklearn,etc
➢ Loaded data as df
➢ Dealt with nan values in text column
➢ Text Preprocessing -

For this particular data set, our text cleaning step includes HTML decoding, remove stop words, change text to lower case, remove punctuation, remove bad characters, and so on.

➢ Splitted data into test and train data
➢ Defined a function to plot classification report for different models

➢ Word Embeddings

A word embedding is a form of representing words and documents using a dense vector representation. The position of a word within the vector space is learned from text and is based on the words that surround the word when it is used. Word embeddings can be trained using the input corpus itself or can be generated using pre-trained word embeddings such as Word2Vec.

There are four essential steps:

1. Loading the pretrained word embeddings
2. Creating a tokenizer object
3. Transforming text documents to sequence of tokens and pad them
4. Create a mapping of token and their respective embeddings

◆ ML Models

➢ Model 1 - Logistic Regression(Accuracy-0.979)

Used ml Pipeline to chain all steps together which can then be applied to training data

Why Used Logistic over Naive Bayes?

- Naive Bayes fits feature weights independently while logistic regression accounts for correlations amongst features. As a result, Naive Bayes classifiers are often poorly calibrated, meaning that the predicted probabilities from Naive Bayes can be a poor fit for the empirical frequencies of outcomes.

Logistic regression won't suffer from this problem as much, as it accounts for the correlations and implicitly aims to make calibrated predictions.

➢ Model 2 - Bagging(Random Forest)(Accuracy-0.978)

Used ml Pipeline to chain all steps together which can then be applied to training data

Why Random Forest?

Random forest builds multiple decision trees and merges them together to get a more accurate and stable prediction.

Random Forest adds additional randomness to the model, while growing the trees. Instead of searching for the most important feature while splitting a node, it searches for the best feature among a random subset of features. This results in a wide diversity that generally results in a better model.

➢ Model 3 - Bidirectional RNN

Used ml Pipeline to chain all steps together which can then be applied to training data

Why Bidirectional LSTM?

LSTM in its core, preserves information from inputs that has already passed through it using the hidden state.

Unidirectional LSTM only preserves information of the past because the only inputs it has seen are from the past.

Using **bidirectional** will run your inputs in two ways, one from past to future and one from future to past and what differs this approach from unidirectional is that in the LSTM that runs backwards you preserve information from the future and using the two hidden states combined you are able in any point in time to preserve information from both past and future.

What they are suited for is a very complicated question but BiLSTMs show very good results as they can understand context better, I will try to explain through an example.

Lets say we try to predict the next word in a sentence, on a high level what a unidirectional LSTM will see is

The boys went to ....

And will try to predict the next word only by this context, **with bidirectional LSTM** you will be able to see information further down the road for example

Forward LSTM: The boys went to ...

Backward LSTM: ... and then they got out of the pool

You can see that using the information from the future it could be easier for the network to understand what the next word is.