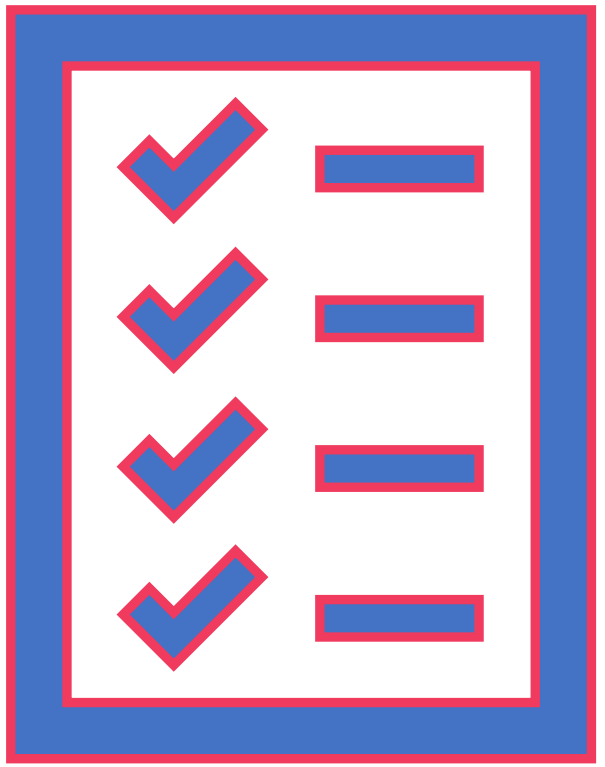
COMPUTER SCIENCE
VIRGINIA TECH

Yusuf Elnady
yelnady@vt.edu

Classifying YouTube videos into Unbiased, Informative and False Rumor Videos

Outline

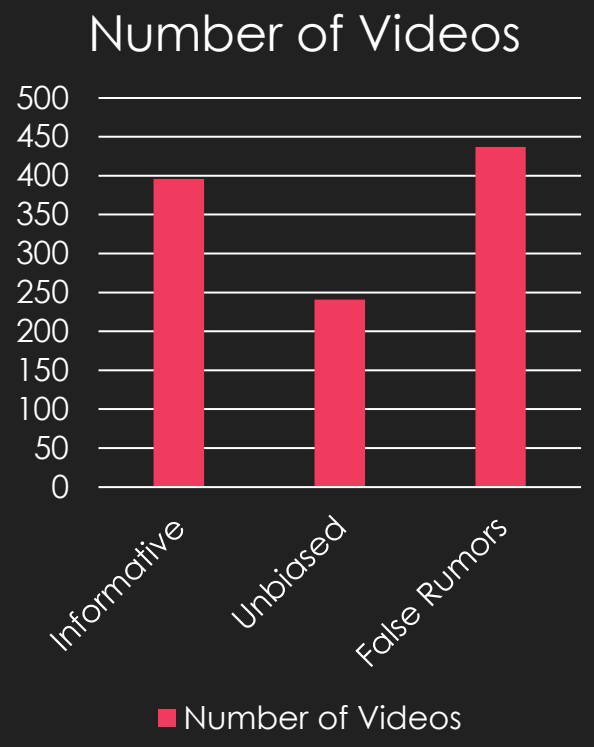
- Problem statement
- Data description
- Data preprocessing
- Data exploration
- Model building
- Model evaluation
- Real-world insights
- Lessons Learned



Problem Statement

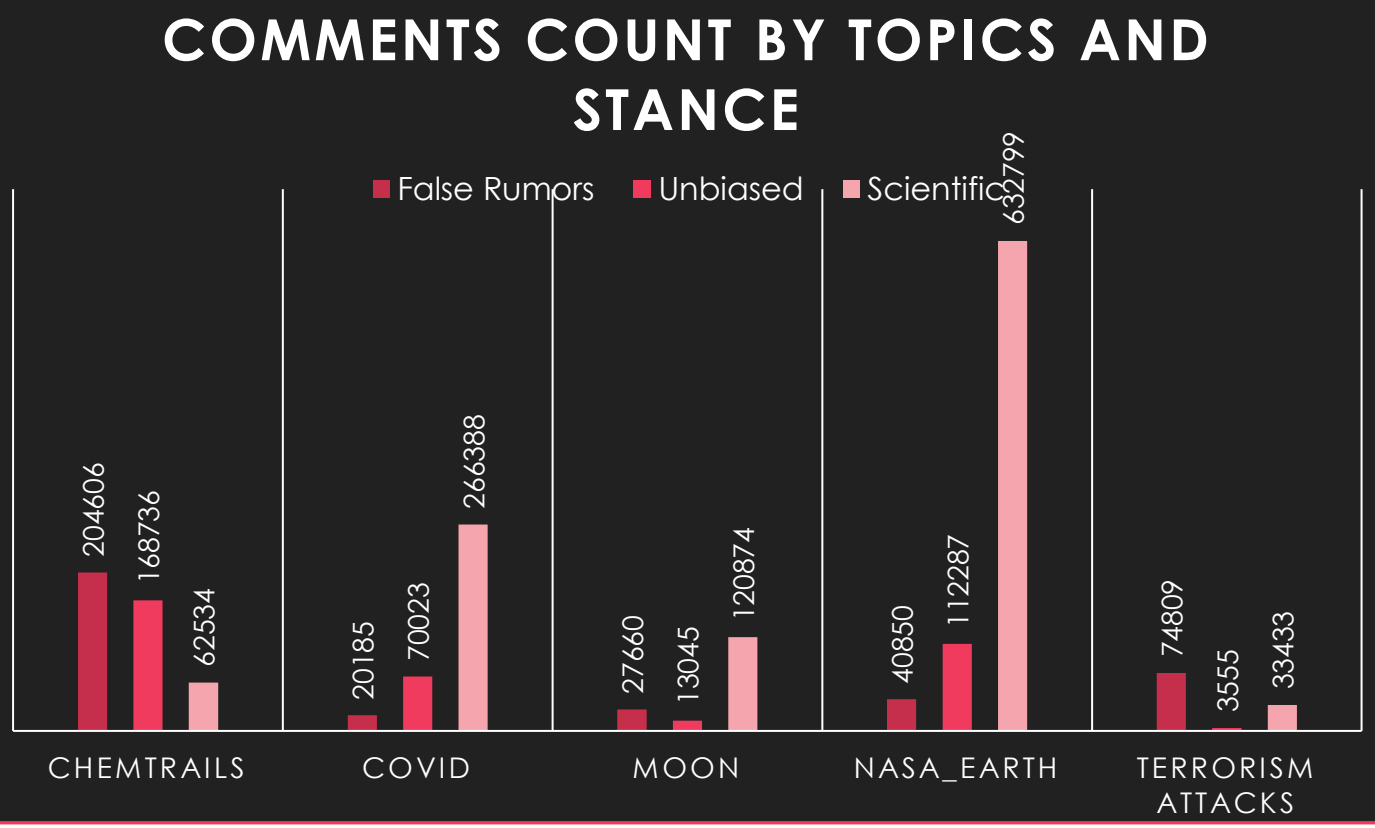
- YouTube is a very big site that contains a lot of information where almost everyone in the world can post anything, and only few videos that clashes with YouTube videos are being rejected (e.g., those containing nudity, terrorism, or race differentiation).
- The idea of this project is to focus on some specific controversial topics of the YouTube videos and try to build a classifier that detects whether the information provided in this video is correct, or it's false rumors that misleads the people.
- The classification is done based on the comments of users on the videos.

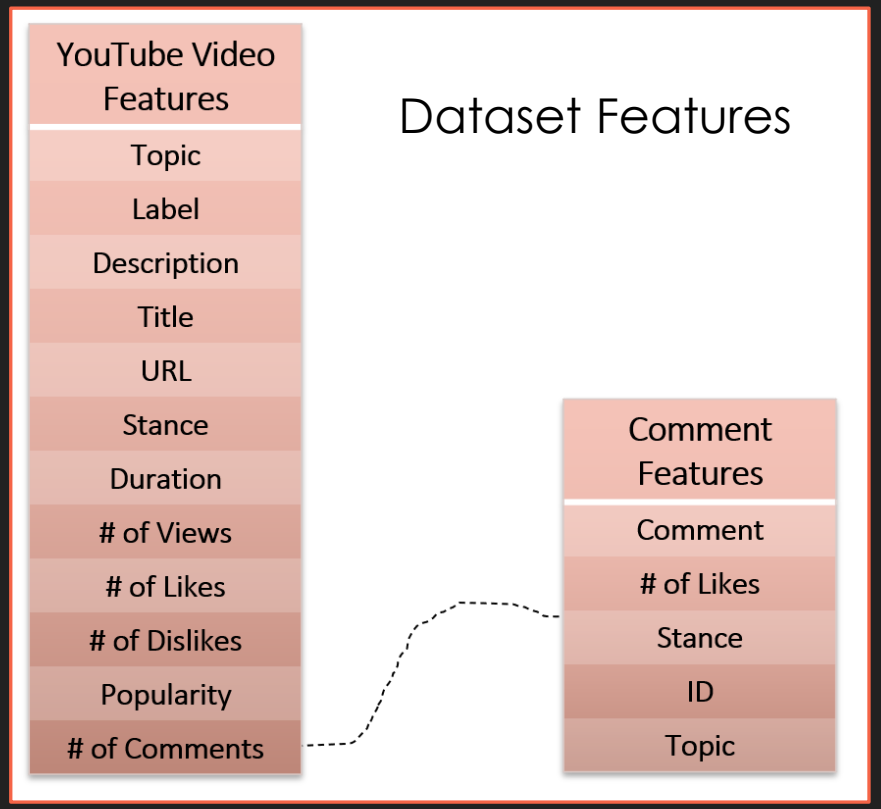
Data Description



- The initial dataset consists of more than 2900 videos that are shown in this figure
- Videos has the following features: {'Topic_Name', 'Label', 'Description', 'Title', 'URL', etc.. }.
- These videos consist of different topics as shown in the Topic_Name attribute.
- Each video is annotated by its stance as either +1, 0, -1 which are informative, unbiased, or false rumor videos.
- The results were more than 2900 files; a JSON file for each video that has been converted to a CSV file.
- Every comments file has the following features: { 'comment', 'No_of_Likes', 'Topic', 'stance', 'ID }.
- The final number of comments I got is 1,851,784 comments that cover all topics together.

Data Description





You can access these dataset (More than 330 MB) using the following link:
https://drive.google.com/drive/folders/1v0JOVtLRx2hPfg1tro8-DgT6wZ34CdP?usp=sharing

Dataset Summary

This is a summary of the comments have been queried for each type of videos.

Topic	Data	Scientific	Unbiased	False Rumors	Total
Terrorism Attacks	No. of videos	31	115	26	172
	No. of comments	204606	168736	62534	435876
Nasa Moon	No. of videos	14	55	90	159
	No. of comments	27660	13045	120874	161579
COVID-19 Controversies	No. of videos	88	42	213	343
	No. of comments	20185	70023	266388	356596
Earth is flat	No. of videos	23	20	71	114
	No. of comments	40850	112287	632799	785936
Chemtrails	No. of videos	240	9	37	286
	No. of comments	74809	3555	33433	11797

Data Preprocessing

Dealing with Missing Info

- To get the YouTube comments you need to be connected to YouTube API. Once you can query the videos you get their comments. The problem is when I started processing the data, there was an error that comments of float type cannot be processed, which was later found to be Nan comments.
- They are comments that are deleted by users, so their content doesn't exist. The solution I followed is to completely drop any row that has a Nan comment.

Removing Unwanted Words and Characters

- Some comments are very short, and they just contain some symbols or special character that does not make any sense to remain in the data.
- Other comments that are just having single letters as "w w x w w" without having a complete word that gives the comment a meaning.
- Other comments are written in languages that are not English which is not helpful in our analysis.
- These variations in the comments are due to the variation in the users from all over the world accessing YouTube.
- I removed all of these words, special characters, and digits to make the comments clean.

Stemming

- Stemming describes the process of transforming a word into its root form.
- Stemming collapses provisionally linked terms.
- I stemmed the comments to increase the accuracy of predicting the YouTube position and acts as a word normalization method.
- This figure shows an example of a comment after it has been stemmed.

Data Preprocessing

Stemming

- Stemming describes the process of transforming a word into its root form.
- Stemming collapses provisionally linked terms.
- I stemmed the comments to increase the accuracy of predicting the YouTube position and acts as a word normalization method.
- This figure shows an example of a comment after it has been stemmed.

Landing people on the moon and returning to Earth is so amazing that people can't even believe that

land	peopl	on	the	moon	and	return	to	earth
is	so	amaz	that	peopl	Can't	even	believ	that

Tokenization

- Tokenization describes the general process of breaking down a text corpus into individual elements that serve as input for various natural language processing algorithms.
- A token is an instance of a sequence of characters that are grouped together as a useful semantic unit for further processing.
- This figure illustrates the process of tokenization on a comment from the dataset.

Why didn't they keep the film's like this also?

Why	didn't	they	keep	the	film's	like	this	also?

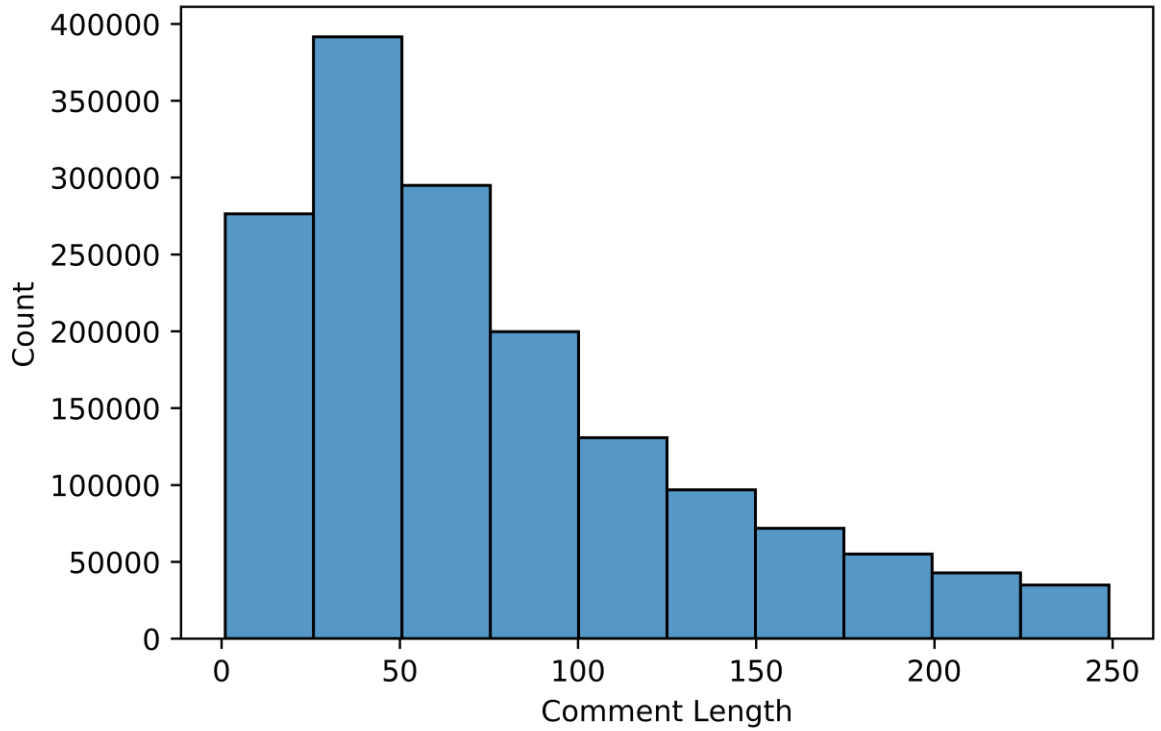
Data Preprocessing

Removing Stop Words

- The last step that is heavily used in many text classification is to remove the stop words.
- Stop words are words that are especially frequent in a corpus of text and therefore are considered somewhat uninformative (e.g., words such as so; and; or; and the).
- However, I ended up not removing stop words from the comments.
- Later in the slides, I report the metrics I got from running the different preprocessing on many data and showing that I obtained higher accuracy without removing the stop words; otherwise, that will alter the sentiment of the comment.

Data Exploration

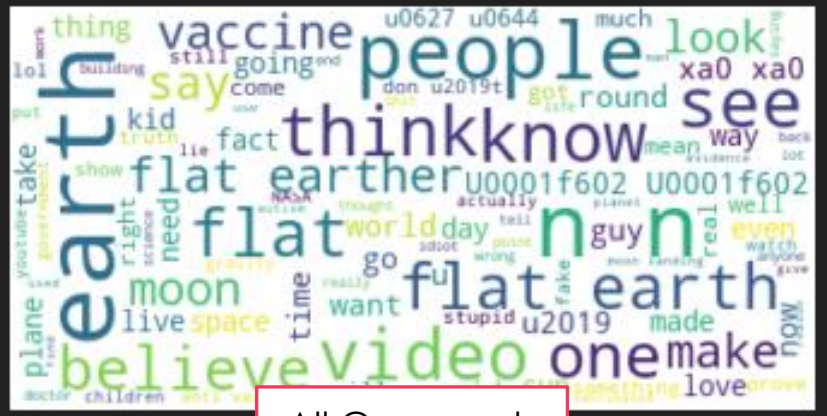
- The dataset is consisted of 1,851,784 comments and their distribution is shown in this histogram.
- I have removed the outlier comments with a huge length so the data can be shown clearly
- Outlier comments may have a length of 70,000 characters.
- The mode of the length of the comment is 28 words which is enough length to provide a piece of information or ask a question
- The shape of the distribution which is right-skewed.



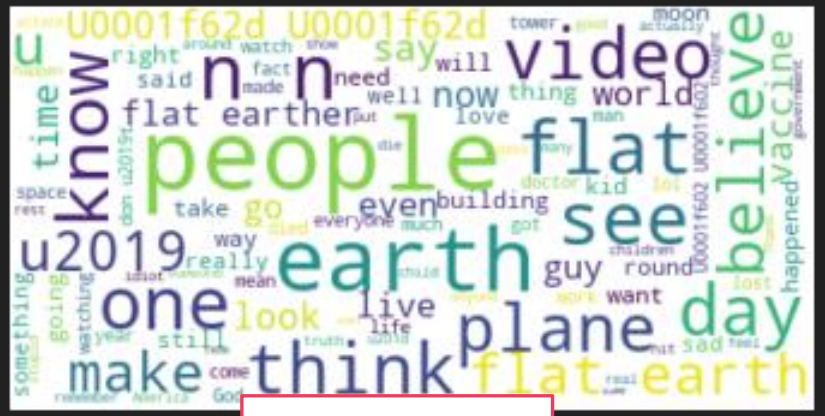
Data Exploration

- To get more information about the most frequent words in the comments.
- Here is an example of the words that exist in scientific and credible videos of my list.
- I used the WordCloud library to produce this beautiful combination of words in the following figures

Scientific



All Comments



Unbiased

False Rumors

Model Building – Feature Selection

TF-IDF Vectorizer

- TF-IDF is an abbreviation for Term Frequency Inverse Document Frequency.
- It is a very common algorithm to transform the text into a meaningful representation of numbers which is used to fit the machine learning algorithm for prediction
- In TF-IDF vectorizer, Sklearn provides a lot of parameters that must be hyper-tuned in order to let the algorithm works best for your solution.
- Using the grid search provided by Sklearn, I grid searched values for specific parameters of the TF-IDF to tune it and get the best numbers to use.
- The best parameters are reported in the paper

Latent Semantic Analysis

- It is a dimensionality reduction technique that aims to capture the latent structure and semantics of the comment.
- One of the major problems in Natural Language processing is the presence of a large number of features extracted from the text which leads to complex models and high computation requirements, ultimately leading to low accuracy and poor results.
- LSA match topics instead of exact words i.e., words occurring in a similar context.
- I made a matrix generated by unigrams and apply Singular Value Decomposition, SVD, to yield three matrices USV out of which V forms our final feature matrix, then you can construct a word-review matrix D, of dimensions of the unigram corpus. Finally, perform the Singular Value Decomposition on D.

Model Building – Learning Process

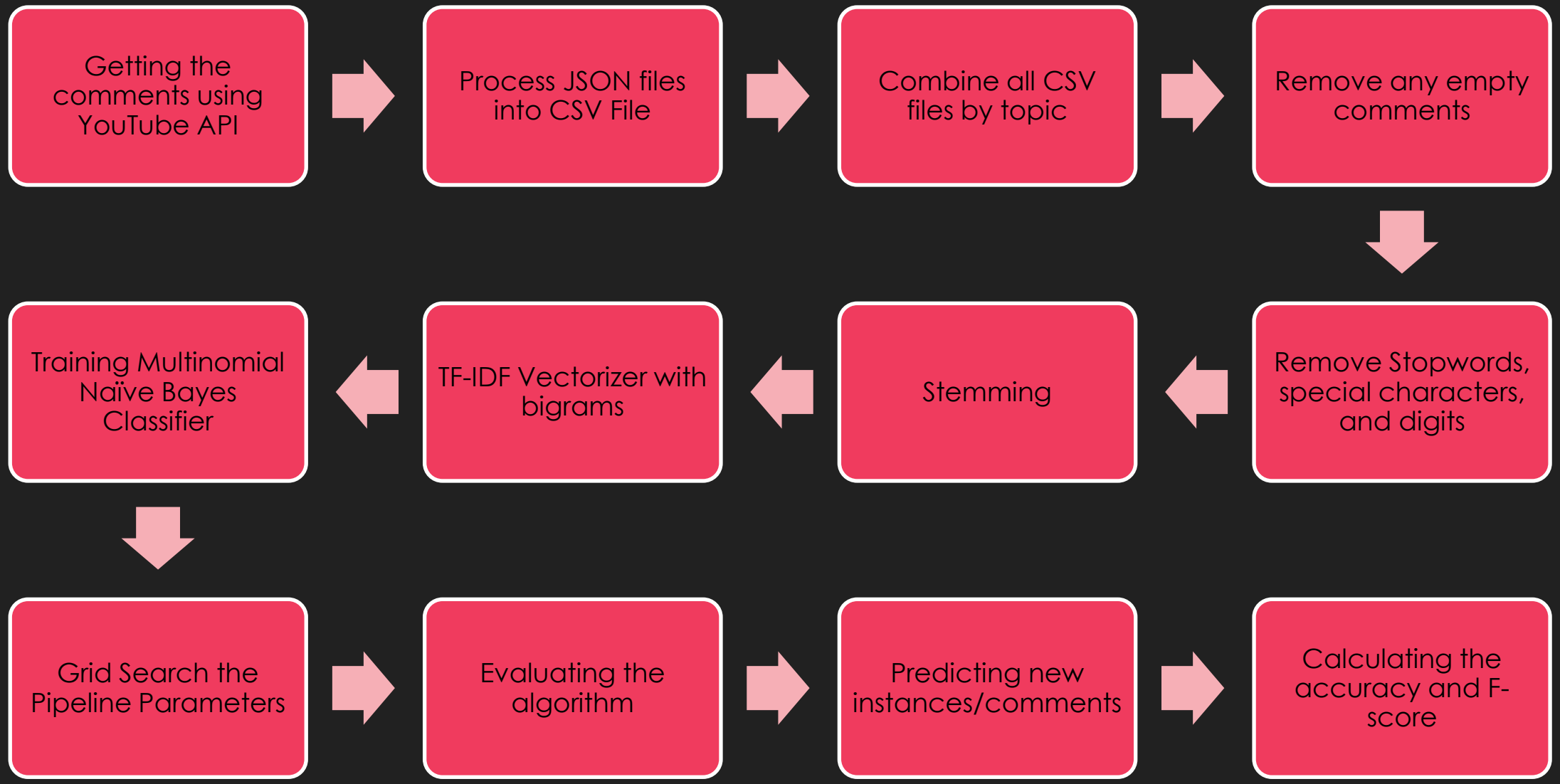
Multinomial Naïve Bayes

- The Naïve Bayes Classifier assumes that given a class the conditional probability between any two features is independent of each other.
- I calculate the posterior probability given a feature using this assumption and build the model.
- When a new feature is encountered, we compute all the joint probability values of the class for that feature and the highest probability is the output as the final class label for this new feature.
- Multinomial Naïve Bayes is generally used in document classification as it works well for data that can be converted into frequencies
- I used the grid search algorithm provided by Sklearn for hyper-tuning my pipeline, and I got the best alpha, the additive smoothing parameter, to be 0.1.

Stochastic Gradient Descent (SGD)

- Stochastic gradient descent is a very popular and common algorithm used in various Machine Learning algorithms, most importantly forms the basis of Neural Networks.
- There are a few downsides to the gradient descent algorithm. For example, we need to take a closer look at the amount of computation we make for each iteration of the algorithm, or otherwise we will run out of memory.
- SGD solved this problem as is also common to sample a small number of data points instead of just one point at each step and that is called “mini-batch” gradient descent.
- Mini-batch tries to strike a balance between the goodness of gradient descent and the speed of SGD.
- Using grid search, the best parameters are loss = 'hinge', penalty = 'l2', alpha = 0.0001, max_iter = 1500.

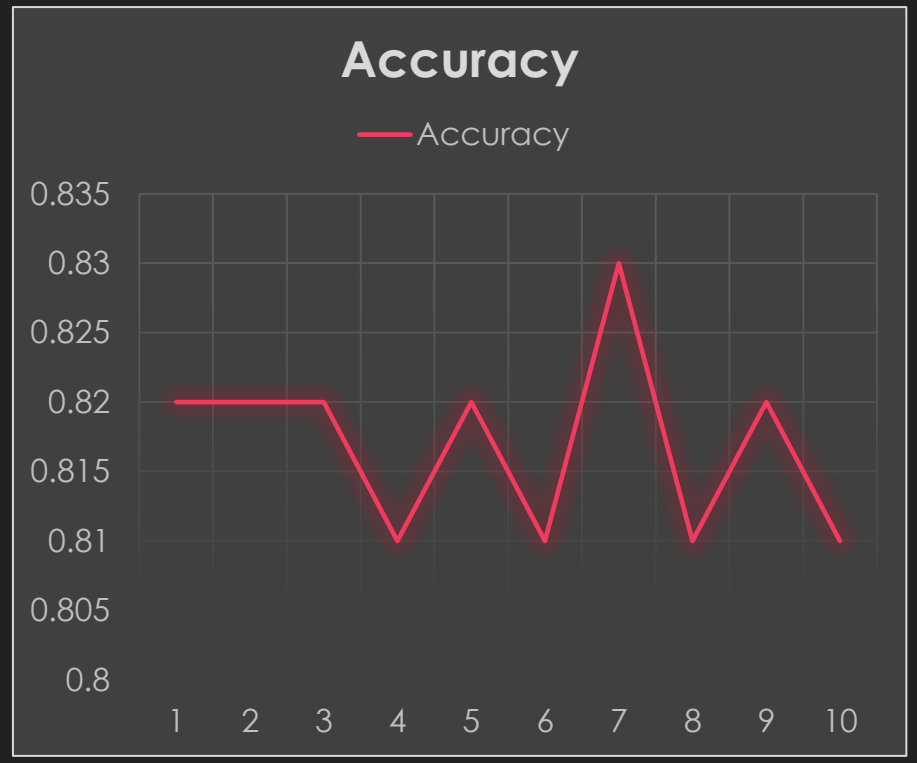
Model Building Pipeline



Model Evaluation

An example of the model accuracy that has been trained on the Flat Earth data set

ID	Settings and Classifier	Accuracy
1	MultinomialNB without removing stop words using unigrams	0.82
2	SGD Classifier without removing stop words using unigrams	0.82
3	MultinomialNB Classifier without removing stop words using Bigrams	0.83
4	SGD Classifier Classifier without removing stop words using Bigrams	0.81
5	MultinomialNB Classifier without removing stop words using Trigrams	0.82
6	SGD Classifier without removing stop words using Trigrams	0.81
7	MultinomialNB with removing stop words using bigrams	**0.83**
8	SGD Classifier with removing stop words using bigrams	0.81
9	MultinomialNB Classifier with removing stop words and using stemming with bigrams	0.82
10	Logistic Regression Classifier using Latent Semantic Indexing	0.81



Model Evaluation

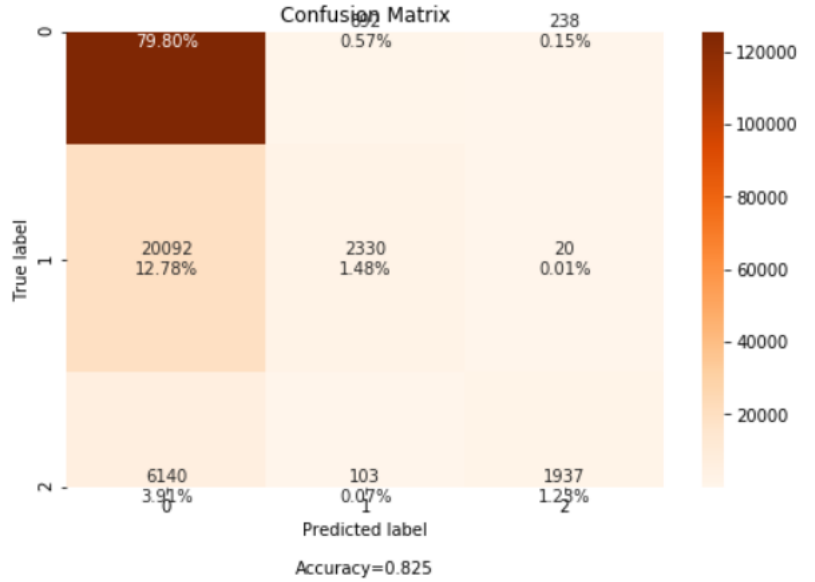
- For each of the 10 different settings I consider, I have reported in the code different metrics that helps understanding the results.
- The following figure shows the different metrics we got on the test set after training our model: **MultinomialNB Classifier without Removing Stop Words and Using TF-IDF Vectorizer with Bigrams.**

- You can refer to the Jupyter notebook to see the output of each specific model.
- The reasons why I choose this model is described in the paper.

```
Accuracy: 0.83
Area Under The Curve: 0.76
Detail:
              precision    recall  f1-score   support

          -1       0.83      0.99      0.90    126556
           0       0.70      0.10      0.18     22442
           1       0.88      0.24      0.37      8180

    accuracy                           0.83    157178
   macro avg       0.80      0.44      0.49    157178
weighted avg       0.81      0.83      0.77    157178
```



Real-world Insights

- Comments are important in helping understanding the data and analyzing it.
- Choosing correct Feature Selection technique is as important as choosing the correct learning technique.
- Traditional Machine Learning techniques perform well on the text provided that the features fed into them are robust and good.
- Like Count cannot be used as a feature to weigh the samples because it is varying from user to the other, and people who watch misinformative content will have more of these videos and they will intend to like these comments that support their perspective, while other may dislike it because it doesn't support their scientific beliefs.
- The results showed that in some topics when removing stop words, the classification accuracy and F1-score decreases, which can be explained by the fact that some of the deleted words contain is critical to the meaning of the comment and for the classification issue, and after removing them the model becomes unable to perform well. In general, comments tend to take the form of revulsion and exclamatory questions and removing these words greatly affected the classification.
- Having one model for classifying any video into misinformative, unbiased, or scientific-based is not a good idea. Rather it's better to have models trained for specific models that has some controversies around it.

Lessons Learned

- In this project I dived deeper into the field of NLP, data preprocessing, and text classification.
- I learned about using TF-IDF Vectorizer, Count Vectorizer, and Latent Semantic Indexing.
- I tried to dive into word2vec, but I didn't have time to learn the deep learning (e.g., LSTM).
- I have also practiced a lot using machine learning libraries, such as Sklearn, pandas, NumPy, matplotlib, nltk, genism, and seaborn.
- I learned some useful components that Sklearn provides such as Pipeline, Grid Search, Feature Selection, Truncated SVD, NMF, and many others.
- I have also learned how to train models using my data and how to clean and preprocess it before feeding to the classifier. As I have tried many settings in order to get the highest accuracy and F1-score I can approach.
- I learned using the YouTube API and Google Cloud Platform.

If I have the same opportunity again

- I prefer starting with the other advanced state-of-the-art techniques, such as RNNs, Word2Vec, LSTM and BERT.
- I would have started learning deep learning and neural networks and testing on them.
- I would do more hyperparameter tuning to boost accuracy.
- I would use boosting and stacking.
- I would think of how to use the features that are related to the video itself as auxiliary information to help further in predicting the stance of the video.
- I would have used a GPU for training, so I don't waste a lot of time waiting for the code to complete and produce the results.

THANK YOU!