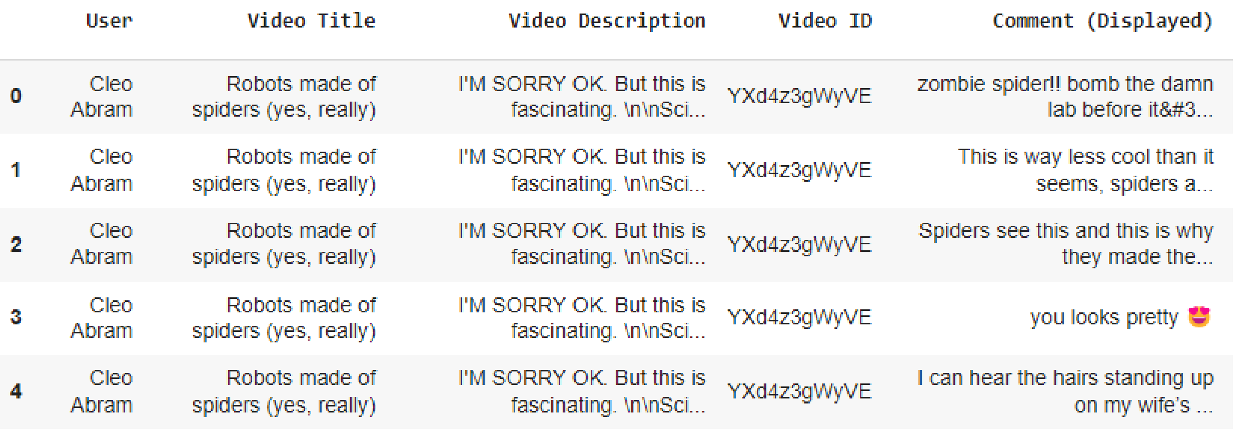
Initial Data -

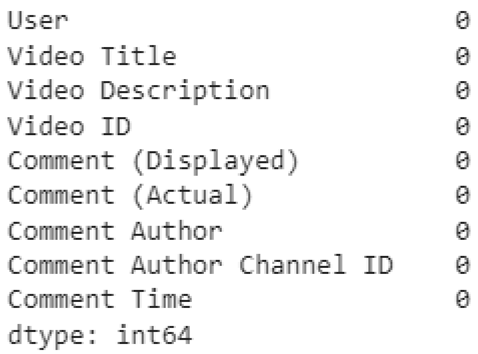
- Unlabelled
- Contains emojis, symbols that create problem in clustering



	User	Video Title	Video Description	Video ID	Comment (Displayed)
0	Cleo Abram	Robots made of spiders (yes, really)	I'M SORRY OK. But this is fascinating. \n\nSci...	YXd4z3gWyVE	zombie spider!! bomb the damn lab before it&#3...
1	Cleo Abram	Robots made of spiders (yes, really)	I'M SORRY OK. But this is fascinating. \n\nSci...	YXd4z3gWyVE	This is way less cool than it seems, spiders a...
2	Cleo Abram	Robots made of spiders (yes, really)	I'M SORRY OK. But this is fascinating. \n\nSci...	YXd4z3gWyVE	Spiders see this and this is why they made the...
3	Cleo Abram	Robots made of spiders (yes, really)	I'M SORRY OK. But this is fascinating. \n\nSci...	YXd4z3gWyVE	you looks pretty 😍
4	Cleo Abram	Robots made of spiders (yes, really)	I'M SORRY OK. But this is fascinating. \n\nSci...	YXd4z3gWyVE	I can hear the hairs standing up on my wife's ...

Preprocessing -

- Removing all the symbols and emojis and apostrophes.
- Checking for null values and removing them -

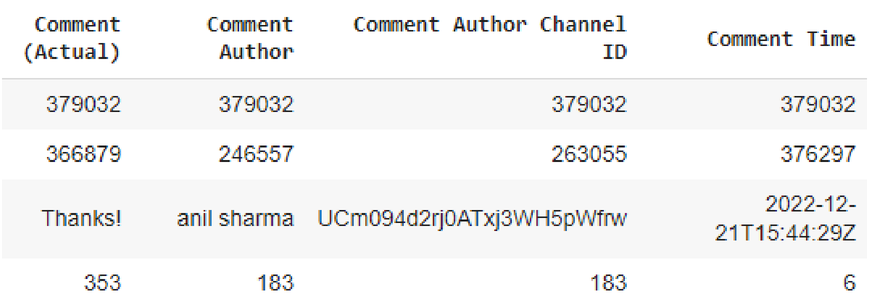


```
User                        0
Video Title                 0
Video Description           0
Video ID                    0
Comment (Displayed)         0
Comment (Actual)            0
Comment Author              0
Comment Author Channel ID   0
Comment Time                0
dtype: int64
```

There are no null values

- From df.describe()

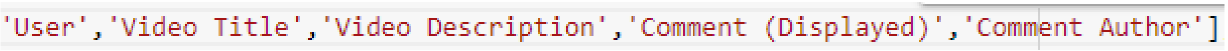
There are 2.6 lakhs unique Author Channel ID's of comments



Comment (Actual)	Comment Author	Comment Author Channel ID	Comment Time
379032	379032	379032	379032
366879	246557	263055	376297
Thanks!	anil sharma	UCm094d2rj0ATxj3WH5pWfrw	2022-12-21T15:44:29Z
353	183	183	6

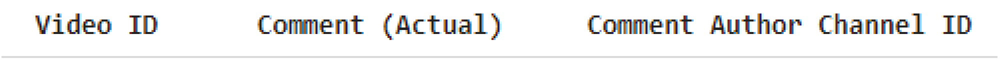
There is one idea of marking all those comments as spam, which have their author write comments whose number exceeds a certain threshold value, but that would interfere with making of clusters and labeling data.

- Dropping unnecessary columns -



```
'User','Video Title','Video Description','Comment (Displayed)','Comment Author']
```

These columns were similar to the columns which were kept -

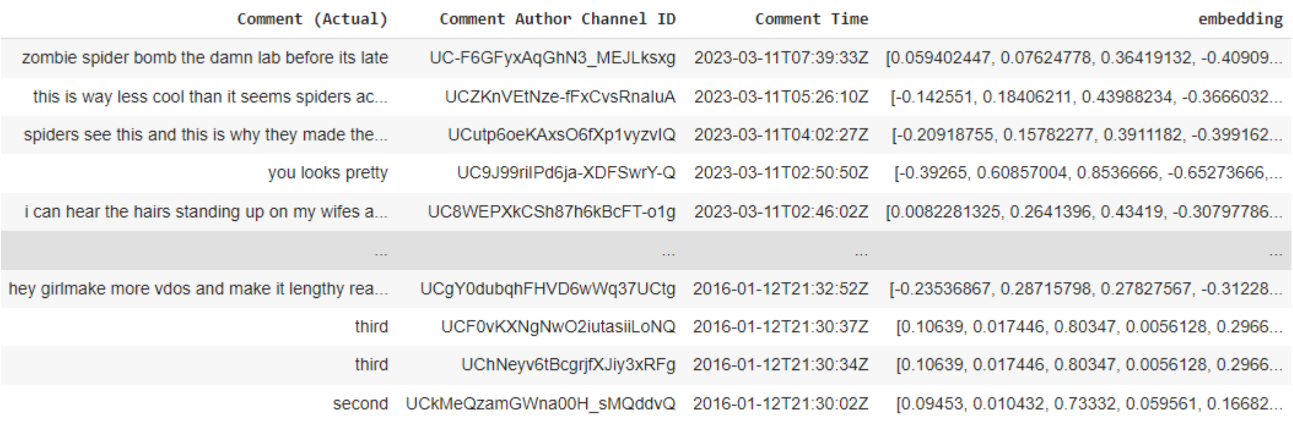


Video ID	Comment (Actual)	Comment Author Channel ID

Like , Video Title and Video ID were same, Comment(Actual) And Comment (Displayed) were similar. So one of each was dropped.

Word Embedding -

- Gensim library was used to create vectors of the comments,
    - The Gensim library used Glove's model dictionary to get vectors for each word and then append them to the complete vector



Comment (Actual)	Comment Author Channel ID	Comment Time	embedding
zombie spider bomb the damn lab before its late	UC-F6GFyxAqGhN3_MEJLksxg	2023-03-11T07:39:33Z	[0.059402447, 0.07624778, 0.36419132, -0.40909...
this is way less cool than it seems spiders ac...	UCZKnVEtNze-fFxCvsRnaluA	2023-03-11T05:26:10Z	[-0.142551, 0.18406211, 0.43988234, -0.3666032...
spiders see this and this is why they made the...	UCutp6oeKAxsO6fXp1vyzvlQ	2023-03-11T04:02:27Z	[-0.20918755, 0.15782277, 0.3911182, -0.399162...
you looks pretty	UC9J99rilPd6ja-XDFSwrY-Q	2023-03-11T02:50:50Z	[-0.39265, 0.60857004, 0.8536666, -0.65273666,...
i can hear the hairs standing up on my wifes a...	UC8WEPXkCSh87h6kBcFT-o1g	2023-03-11T02:46:02Z	[0.0082281325, 0.2641396, 0.43419, -0.30797786...
...	...	...	...
hey girlmake more vdos and make it lengthy rea...	UCgY0dubqhFHVD6wWq37UCtg	2016-01-12T21:32:52Z	[-0.23536867, 0.28715798, 0.27827567, -0.31228...
third	UCF0vKXNgNwO2iutasiiLoNQ	2016-01-12T21:30:37Z	[0.10639, 0.017446, 0.80347, 0.0056128, 0.2966...
third	UChNeyv6tBcgrjfXJiy3xRFg	2016-01-12T21:30:34Z	[0.10639, 0.017446, 0.80347, 0.0056128, 0.2966...
second	UCkMeQzamGWna00H_sMQddvQ	2016-01-12T21:30:02Z	[0.09453, 0.010432, 0.73332, 0.059561, 0.16682...

Labeling the data -

KMeans -

- We created 50 clusters of the embedded data using KMeans
- Then printed 20 comments from each cluster to manually check for spam and then label that cluster as spam or not spam in whole
- One way to automate this manual labeling process was to create a new dataset only containing spam comments and calculate the similarity of this spam dataset with the clusters and label them as spam or not spam -
    - But, since the dataset is limited and the new dataset of spams will be very widespread, we dropped this idea.

Labeled Dataset -

Supervised Learning -

- After getting the new labeled dataset -
    - We create new dataset of 2500 comments
- This new small dataset gets trained on 3 supervised learning algorithms namely -
    - Decision Tree
    - Gaussian Naive Bayes
    - Logistic Regression

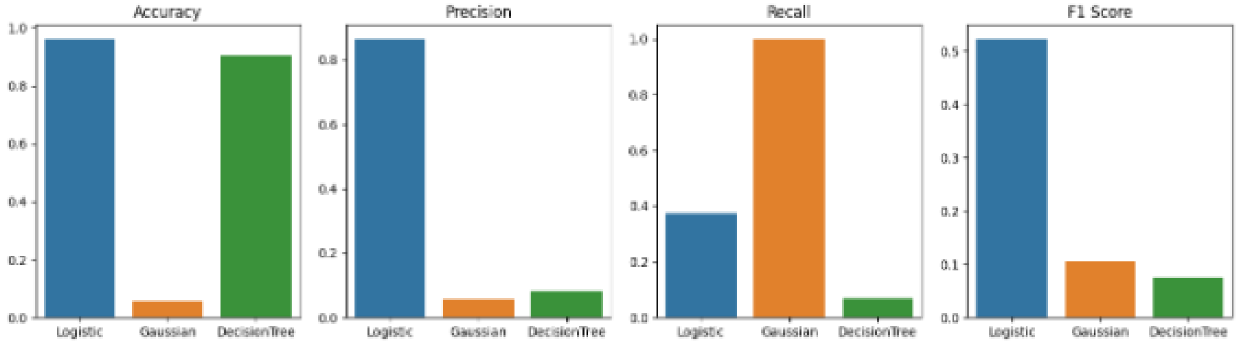
- The scores obtained are -

```
Evaluation metrics for Logistic Regression:
Accuracy: 0.9616364845184575
Precision: 0.8647740816989923
Recall: 0.3751527686377738
F1 Score: 0.5232927908730289

Evaluation metrics for Naive Bayes:
Accuracy: 0.05612718715042529
Precision: 0.05612718715042529
Recall: 1.0
F1 Score: 0.10628868915279811

Evaluation metrics for the Decision Tree:
Accuracy: 0.9034250406298149
Precision: 0.08169167803547067
Recall: 0.07036758484535113
F1 Score: 0.07560796989822975
```

- From the above scores we can say that LR seems to be biasing the data by not correctly identifying the spam comments. Its precision and recall scores are lower than its accuracy, indicating that it may have struggled with correctly identifying positive instances in the data.

- Whereas The Naive Bayes gave biased results only in favor of the spam comments, i.e. it classified all the comments as spam giving it a perfect recall score bit very less accuracy and precision. The poor performance of Naive could be due to it assuming that the input variables are independent.

- While the Decision tree is opposite of Naive Bayes, Its precision score is very low, which means it made many false positive predictions. Its recall score is also low, indicating that it failed to identify many positive instances in the data.This could be due to the fact that Decision Tree is prone to overfitting if not properly regularized or if the depth of the tree is too high.



Finally KNN applied -

- For the process to work correctly, we needed to set up the pipeline such that given a dataset of comments we could identify the cluster of spam comments which finishes the work once and for all.
- For this task, we applied KNN training and testing on the labeled data from KMeans
  - Complete dataset was divided into a 7:3 training to testing ratio
  - The scores obtained for KNN are -

    Accuracy: 0.9801424676809427

    Precision: 0.9198966408268734

    Recall: 0.7138670368656487

    F1 Score: 0.8038909154073302
- We can see that the final accuracy score is 98%, while maintaining F1-score of 0.8
  - This tells us that our pipeline is able to detect spam vs non-spam without biasing between both.
- We tried to apply DBSCAN to find single cluster for spam detection