**PRESTIGE INSTITUTE OF MANAGEMENT AND RESEARCH, INDORE**

(An Autonomous Institution Established in1994, Accredited Twice Consecutively with Grade A‟NAAC’’ (UGC) ISO 9001: 2008 Certified Institute, AICTE / UGC Approved Programs affiliated to DAVV, Indore)



**(Session 2023 – 2023)**

**Capstone Project Report on**

“YouTube Comments Spam Detection”

**Faculty Mentor: Submitted By:**

Mr. Atul Astai Hussaina Haidary (21110300009)

Tanishq Singh (21110113467)

Tashmeet Kaur Saluja (21110313642)

**Class:**

BBA Business Analytics

VI Semester (A)

***Introduction:***

YouTube is world’s most viewed video sharing platform. It presents the user base all categories of information, news, facts, music, live telecasts of sports, concerts and even millions of channels that can exhibit their content. In the information age, there has been tremendous amount of knowledge shared through various ways. Even the sources of the information have been increased by ample amount. Voluminous advices, suggestions, entertainment is available at ease for the consumers. According to a market trend analysis of social media platforms, it is known that YouTube has more than 2 billion users and 500 hours of video been uploaded for every 60 seconds 43,200,000 hours of video on every day. This statistic also collaterally indicates the earning of millions of dollars. In this view, much of the YouTubers request their users for likes, shares, comments and subscribes as it would earn them larger user base and hence bountiful feast of earnings. Along with the pros, there are always associated downsides and here, that is comments – more specifically spam. Discussion and analysis of various happenings in the world and related matter in the comment section is welcoming as it also plays role in educating the users. But online incivility, spamming is one of the contemporary issues that has entangled the world. Even though, YouTube has spam reporting mechanism, ascertaining comments as spam and control of it has not been fully successful. The proposed work would make the nascent steps into detecting spam that should be eliminated and this should form the footing ground for further detection of online Incivility and removal of them.

***Methodology followed:***

1. Data acquisition
2. Data cleaning
3. Splitting and sampling of data
4. Feature extraction
5. Executing machine learning models
6. ***Data Acquisition:***

The proposed work is applied on the dataset that consists of comments from five of the most watched YouTube videos by 2015. The data is captured through YouTube Data API link and available for public use. The dataset has 5 CSV files that correspond to five most watched videos respectively, which totally has 1956 comments. The dataset has following details:

* COMMENT\_ID: Unique id for each comment
* AUTHOR: Name of the user who has commented
* DATE: Date on which the comment was published with time stamp
* CONTENT: Content of the comment
* CLASS: 1 or 0

Following are the data files used in the project:

1. 'Youtube01Psy.csv'

2. 'Youtube02-KatyPerry.csv'

3. 'Youtube03-LMFAO.csv'

4. 'Youtube04-Eminem.csv’

5. 'Youtube05-Shakira.csv'.

All the comments in these files are labeled as spam or ham (legitimate), where class value 1 corresponds to spam and 0 corresponds to ham comments

1. ***Data cleaning:***

Data cleaning is the phenomenon of finding wrong - formatting, construction, spellings, unwanted characters, spaces and etc. As data is the core component of data analysis, cleaning the data to format it to the requirement is imperative. After reading the data from all the five 5 files, it is added to the data files array. In the proposed work, data cleaning is done in two stages:

1. Removing insignificant columns: Comment ID, Author and Date columns are removed as they do not add much value for spam comment analysis. Now, the data files array contains only Content and Class columns.
2. Processing the Content Column: Content i.e., comments on YouTube might contain lot of punctuations, special characters, numbers and other superfluous content. In this stage, only the characters ranging from A-Z are retained and all the characters in the comments are converted to lowercase.
3. ***Splitting and Sampling data:***

The processed data now should be split into training and testing data sets. As the name suggests, training data set is used for training the Machine Learning Models in further stages. To test the accuracy of model training, test data is used. An important aspect of this stage is – whole process should be done randomly as it yields in better training the models as it produces less biased data sets. To accomplish to random splitting and sampling of data, python library SciKit – splitter function is used.

In this project, train\_test\_split function is used to split the 80% of the data as training set and remaining 20% as test set by specifying test\_size = 0.2 and random state=57 that constitutes 1563 comments as training and 393 comments as testing samples. Random State parameter facilitates random splitting of data as training and testing set. X parameter is the Processed Content column and Y parameter is the Class column.

1. ***Feature Extraction:***

On completion of cleaning and splitting the data into training and testing samples, it entails the process of Feature Extraction. Data set now contains the comments and its corresponding class value suggesting whether it is a Spam or a ham. Comments are a string of words and therefore bag-of-words model can be correlated on the data to extract the all words from all the sentences and then finding its size in the bag. For an example, Let following be the 5 comments in the data set:

1. “hi how are you”
2. ii. “hi who are you”
3. “i like this song”
4. “hi do you like the song”
5. “hi click on the link”

Applying Bag-of-Words technique on this, all the words from all the comments are extracted, [‘hi’, ‘how’, ‘are’, ‘you’, ‘who’, ‘i’, ‘like’, ‘this’, ‘song’, ‘do’, ‘the’, ‘click’, ‘on’, ‘link’] On closer observation, it can be noticed that all the words are treated as individual entities and duplicate terms are not considered for the bag. This data cannot be fed into the models directly as the machine does not understand the context of these words. Next task is to convert the extracted words into the binary language. For conversion to binary language, let us find the multiplicity of the words i.e.

|  |  |
| --- | --- |
| **Terms** | **Occurrence count** |
| Hi | 4 |
| How | 1 |
| Are | 2 |
| You | 2 |
| Who | 1 |
| I | 1 |
| Like | 2 |
| This | 1 |
| Song | 2 |
| Do | 1 |
| The | 2 |
| Click | 1 |
| On | 1 |
| Link | 1 |

Subsequently, vectors can be formed using the extracted words and the data set will be created as the following:

1. “hi how are you” = [1,1,1,1,0,0,0,0,0,0,0,0,0,0]
2. “hi who are you” = [1,0,1,1,1,0,0,0,0,0,0,0,0,0]
3. “i like this song” = [0,0,0,0,0,1,1,1,1,0,0,0,0,0]
4. “hi do you like the song” = [1,0,0,1,0,0,1,0,1,1,1,0,0,0]
5. “hi click on the link” = [1,0,0,0,0,0,0,0,0,0,1,1,1,1]

The above explained process is implemented using SciKitCountVectorizer. CountVectorizer function invokes both ‘tokenization’ i.e. extracting words and ‘counting’. It also creates vectors that assign the binary value to each word on the row basis. In the proposed work – each word is treated as Unigram. The data created using results in tokens of words. But this might reflect the actual context of spam comments on YouTube, because certain words in English like – ‘a’, ‘the’, ‘and’ etc. are used profusely in the sentences. Inclusion of such words in the comments might not actually indicate any resemblance to spam comments. These such ‘stop words’ are removed by passing the parameter – “stop\_words: english”. SciKit already has a list of unhelpful words and removes it from the bag of words. Now, we have list of words and its occurrence count in the comments. As shown in the above example, ‘hi’ has a total count of 4 and ‘click’, ‘link’ – have total count of 1. In the context of identifying the spam comments – a term that has the highest frequency cannot be considered as the indicative. Apparently, even in the above example, ‘hi’ does not have much correspondence towards being the spam word. Contrarily, words like ‘click’, ‘link’ may have affinity towards being the spam words even though their frequency is just one. If this data is fed to the model for training, then the words which have less meaningful info might obscure the actual spam words and hence necessitates the process of re-weighting the terms. To reweight the terms based on the context of the sentences, SciKit ‘Tf-idf term weighting’ function is employed on the vectorized data. “Tf means term frequency and idf means inverse document frequency. TF-IDF reassigns weightage to each word of Training set and testing sets.

1. ***Executing machine learning models***

On transforming the training and testing data sets to required format with rightly assigned weightages, machine learning models have to be built to run on the training data and then to evaluate the efficiency of the models on identifying the spam comments. In this process, two machine learning models are chosen to conduct the proposed task. They are; Logistic Regression and Random Forest Classifier.

1. **Logistic Regression**:

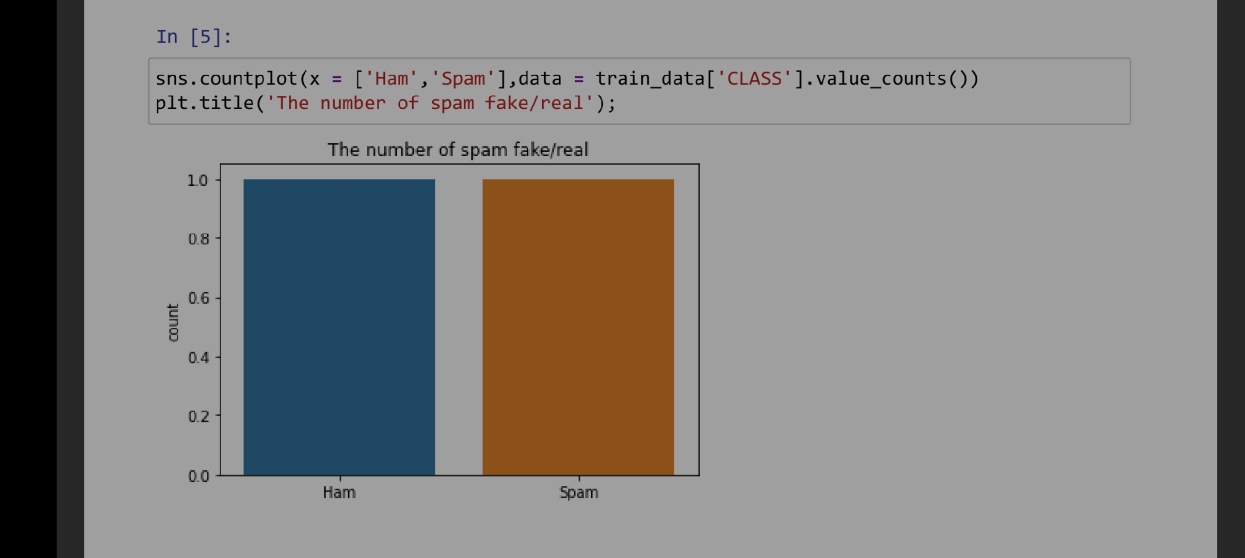
Logistic Regression is used here as it is generally used to predict the happening of an event with respect to the depending factors. Logistic regression is binary model that provides result in the form - whether a word is a spam word or not i.e. 1 belongs to spam and 0 belongs to ham. Logic regression in python is accomplished using SciKit Logistic Regression model.

1. **Random Forest Classifier**:

Random Forest Classifier (RMF) is an ensemble of decision trees and built on the basis of supervised algorithm. Usually, over fitting is the issue in the machine learning models as the models fit the training data too close to the set of data points. This issue happens when the model understands the noise too closely. Usually, to prevent over fitting, regularization is added or the nodes in the hidden layers are diminished. But RMF does it by default as the group of decision trees run altogether. In the project, Random Forest Classifier model of SciKit is used to build the model and run on the training samples. The model is trained with 1563 comments as training samples with their corresponding weight and class values. The same is tested against testing sample set of 393 comments. The best accuracy achieved is 95.15%.

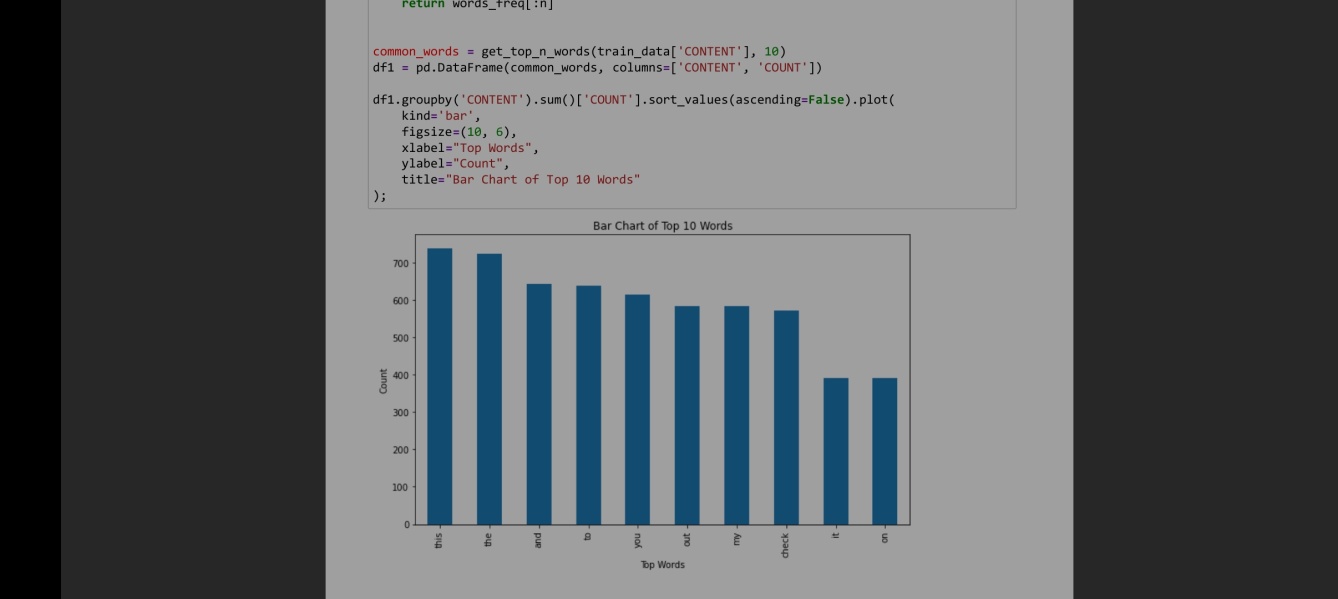
***Data Visualisation:***

Data visualization has been done using the matplotlib.pyplot, seaborn and WordCloud libraries.

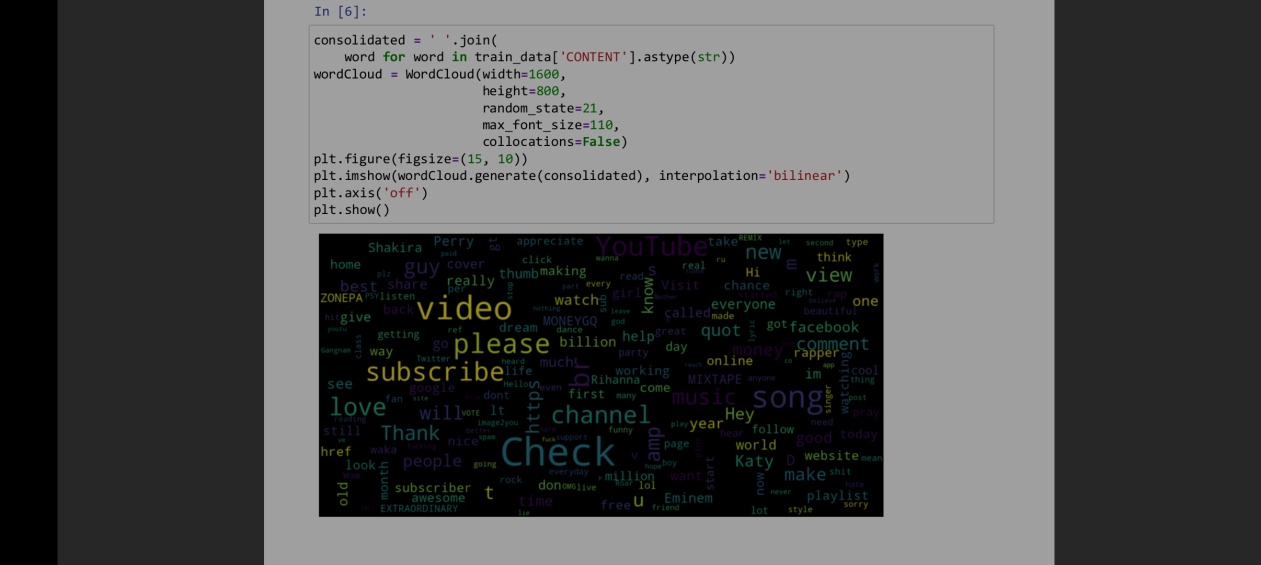


1.The following graph depicts the

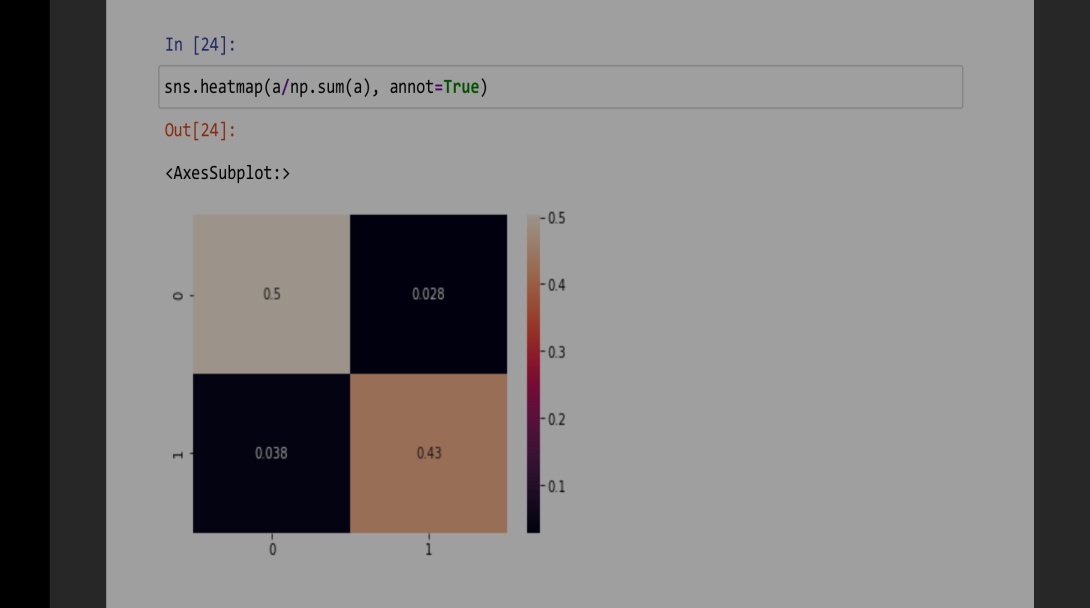
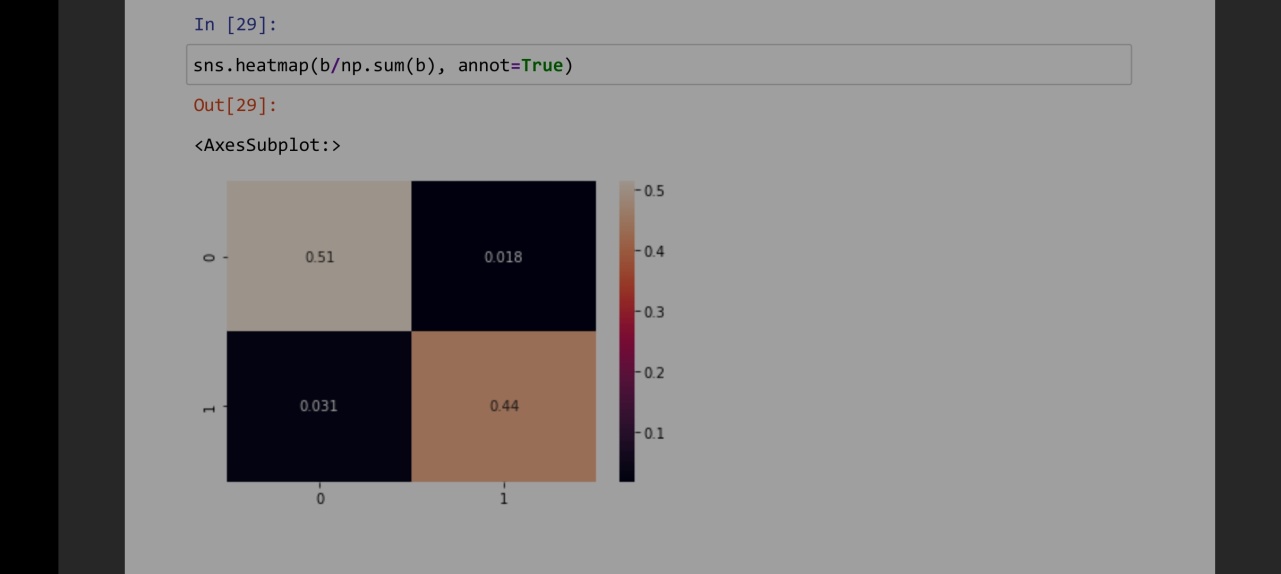
number of spam and non spam (ham) comments present in the data:



2.The presented graph shows the top 10 most used:

3.The following Word Cloud highlight various words used throughout the dataset:

4.The following heat maps show the confusion matrices in various models:



***Conclusion and summary:***

From the outset of data acquisition to lastly studying the various machine learning models, understanding their training configuration, its effects and subsequently its performance in identifying the comments as spam or ham. Logistic Regression and Random Forest Classifier models are used for the work. The accuracy of each of the machine learning model is calculated and Random Forest Classifier model provides the best accuracy of 95.15%. RMF is known to provide best results as aggregation of decision trees form the forest and unbiased estimate is furnished based on the model voting method.

On working with this project, we learnt the process of feature extraction using bag-of-words and tf-idf techniques. The heart of the matter topics of tokenizing, counting and vectorization of data is inspected and understood. Then, learnt the working of machine learning models and implementing each of them in python using SciKit libraries.