2.1 Introduction

In this chapter we concern about to discuss the reviews of other solutions for similar kind of approaches are related to our proposed solution as well as what are the advantages and disadvantages of our proposed solution which was addressed in above. In order to deliver a successful solution for the mentioned above, it is very important to refer previous researches, surveys to get the essential knowledge about what are we going to do and it is very important to identify barriers which previous researchers have faced when they deliver the solution. We are supposed to discuss what are the current limitations and also challenges are appearing when we deliver our proposed solution. The proposed solution has four main submodules which are integrated each other and this chapter includes an overall literature survey for the proposed solution a s well as four individual literature surveys for sub modules.

2.2 Literature survey

When consider about hate content filtering and trend analysis in Sinhala YouTube contents module. It has never been made before. Our research module is the one only module for detect Sinhala YouTube hate contents and trend analysis in available in the field. but when we consider about four sub modules in main module .it has plenty of previous research works have been done by various researchers in various countries for varies languages.

The related work is illustrated under four research areas which are the main modules of the project.

* Filter spam comments
* Extract text in thumbnails
* Identify hate videos
* Predict the trending nature of hate video

2.2.1 Spam comment filtering

When consider about existing solutions for spam comments filtering approaches, it has several approaches for make automate the spam comments filter in YouTube. most of spam comments are very short of words length and most of time those comments include slangs, short meaning words, emoji’s and many more. there for it is very difficult to even tokenize the comments. When look at the comments of YouTube it has a less textual illustration and these comments are commented by human not bots. Therefor these comments have a higher variety with compared with blog spams. Spams are usually unrelated low quality data and it is aim is get attention of a large audients for fulfill the spam commenters target. Because of that usually spammers try to deliver their message more attractively such as use emotions, repeat their comments continuously, post a big comment.

YouTube comments based researches began more than one decade ago. But It has never done a research about Sinhala comment spams in YouTube. However, spamming in the glob is almost similar to Sri Lanka also. It has solidly different spamming patterns with compared the globe. one of the oldest studies of this field is detect spam comments work of MIshne et al. [11]. They built a simple coded model that not requires a training model for detecting spams like links in blogs. their model totally depended on the language commenter used for comment. But usually spam comments not follow the semantic of the language. even their model gives 7.5% of false positive rate and 11% of false negative rate it was very far from being a practical model for detect spam comments. Thomason had investigated about importance of blog spam in general and did a review study on it [12]. he studied number of spamming contexts and available methodologies for overcome spam comments. and also Romero and his fellow members studied blog spam comments from different point of views, they compared several machine learning models and evaluated the result of the accuracy for detecting spam comments in each models. they used Neural Network(NN), Naive Bayes (NB), K- Nearest Neighbor (KNN) and support vector machine (SVM). they used 10-fold cross validation for each technique. their average best outcome was 84.61% accuracy rate with SVM. number of studies had done by using sentimental analysis of YouTube comments [14-17] one of the studies indicated concept of text mining in YouTube comment spam was proposed by Sureka [18]. Author has used text mining techniques to find spam behaviors. There assumption was if more than one comments are in the same content or repeatedly post comments in various kind of video (unrelated video), then that user is named as a spammer [18]. but the main issue of that built method is it was tested on a low number of data and it makes a question about model accuracy. Alberto et al. built an online spam comment filter for YouTube. they collected data set from famous musicians in their YouTube profile. they used normal classification techniques (support vector machine, Logistic Regression) for detect comment spams [19]. They were able to get a high level of performance in their models. But there were various factors exists such as accuracy rate, false positive rate, false negative rate. therefore, some models even give a high accuracy rate but it gave a higher false positive rate also. in YouTube spam filtering domain higher false positive rate is considered as a very important factor for determine a model suitability. All of researches have done for you tube spam comments filter by using only comments details but there is no feature to measure the relatability of a comment to its content. in our proposed solution it has several new features to increase the accuracy in a model with more productive such as YouTube content – comment similarity, this feature compares the similarity of a comments with its own content and give a scale of related behavior of comments. and it has a new feature that determined the time gap between video content and each of comments. usually spammers comment as soon as possible when a content has released. And in this proposed solution it has introduced a new feature as it checks whether a comment has a period marks sequence. Because of most of Sir Lankans has habituated to type sequence of period marks. Therefor we have included that behavior as a feature for filter spam comments in YouTube Sinhala comments.

2.2.2 Text Extraction from images

In this section we illustrate an overview about related works in text extraction from images. Mainly deep learning algorithms use for extract the texts form images and there are plenty of solutions ranged from simple classifier trained on manual features to multi stage pipelines aggregated different algorithms for text detection. The common features are consisted from edge features, shape contexts and texture descriptors. Apart from that some systems have highly flexible learning methods to learn all essential information from labeled data. Amritha S Nadarajan &; Thamizharasi A (2018) has delivered a new approach for get the stork width value of images. Their introduced algorithm able to identify most of fonts and languages which are embedded in images. Their approach is included preprocessing, text localization, classification and relevant character identification. They have used many classification algorithms for their approach such as convolutional neural network, support vector machine, Adaboost and Text-CNN etc. and also they have provided a detailed study of text identification of images. It analyses and compare the different methods and techniques to overcome ongoing text detection challenges in this particular area. They have used different datasets which have extracted from different sources for detect the texts of images and evaluate a comparative study to different text identification methods. Their comparative study has identified the CNN is a much better algorithm for detect text from images. Chandio, A. A., Pickering, M., and Shafi, K. (2018), have been used a hybrid technique for increase the quality of naturel images and also thy have used robust techniques for subtract the image background, they have identified few supervised learning algorithms for map an input to an output. Have used Gaussian algorithms filters for detect and remove noise form images. Tridib Chakraborty et al (2017, find the skewness and noise of the image as the preprocessing. Then image is converted to gray scale after that into binary form. After all of above steps then image can be segmented. Therefor texts which are embedded in image can be extracted out from the image to individual characters. there are some methods for extract the text from images such as text detection, text tracking and text localization. Pirker J, Wurzinger G (2016) have introduced OCR (optical character recognition) system for this. In their study they have mainly focused for old fonts such as historical texts. They have used different training systems such as neural networks and export systems. Jeong, M., Jo, K. H. (2015), in their study they have used edge detection for detect the objects of a specific boundary area, edge components labelling first and after that labeled edge components label new edge components. Using a component aggregation function, it can find the relation between components. Jacob, J., Thomas, A. (2015), in their study they have used preprocessing of images, edge detection for detect the edges, detect stork width, filter the noises and remove noises, feature extraction and after that classification. they have used random forest as the classification algorithm in their study. Rani, N. S., & Vasudev, T. (2015), in their study they have used optical character recognition for identify texts and extraction algorithms have applied for detect the texts. A. Ruikar, S. D. (2014), in their study they have used text region identification method for identify the text. And also they have used Niklack’s local banalization algorithm for segment the image. B. Epshtein, E. Ofek and Y. Wexler, (2010), in this study they have used the image retrieval algorithms for identify the text. It has used many techniques for develop their text detection such as SVM, filter, convolutional neural network (CNN).

Hate video analysis module

Hateful objects such as comments, contents, images detection on web platform is an ongoing popular research area in nowadays. There are plenty of hate content analyses researches have done for the English language. But when look at other languages still it is very rare to find. According to the past related researches on hate content filled, hate contents analysis has carried through basically two main approaches such as machine leaning based hate content analysis and lexicon based hate content analysis. And some of researches have done by using as a combination of machine learning based techniques and lexicon based techniques. H.M.S.T. Sandaruwan, S.A.S. Lorensuhewa, M.A.L. Kalyani (2020), in their study they have used both lexicon approach and machine learning based approach because of when it uses only lexicon based techniques we have to totally depend on the words which include in pre-built lexicon dictionary in system. But when time goes the hate words cn be changed add new words and it is difficult to fully depend on a pre build lexicon dictionary. When consider about machine learning based approach it allows to computer to learn from the input data and it is dynamic not static like lexicon based techniques. Machine learning algorithms makes new inference of the data according to the training data therefor same algorithm can be used for build different models according to different data sets. Köffer, S.et al.(2018) this research has based on German language and it has used bag of words techniques for detect hate speeches in social media contents and it has worked with as English without occurring a language different. Dias, D. S., Welikala, M. D. and Dias, N. G. J. (2019), in this research has only used word bi-gram feature for identify hate speeches. But hate speeches are not limited to racism based comments therefor we want to consider other factors when we are detection hate speeches such as religion, gender, sexual orientation etc. there is another limitation of this research of they have used support vector machine as a training model for their proposed solution and it is one and only model that they have used it makes us that other models are whether capable or not for Sinhala hate speeches identification. When we refer other related researches we noticed that most of researches have used python as the programming language for their proposed solution and NLTk libraries for preprocessing, post processing and model creating tasks. Furthermore, most of related researches have used supervised machine learning algorithms for model building and it is very less to see deep learning algorithms have used for them researches. Malmasi, S. and Zampieri, M. (2017), they have used algorithms like word n-grams, character n-grams and brown cluster with SVM algorithm. Here they have got over 78% testing accuracy for character four-gram feature. As the research Davidson, T. et al. (2017), in their research they have used unigram, bigram and trigram as features with TF\_IDF algorithm for the detect the hate speeches. They have been able to get 0.91 accuracy for logistic regression model and it is the best model to detect hate speeches their research. Recent research Alfina, I. et al. (2018) in their research have done for Indonesian language and published with the Indonesian language. They have used word n-grams and character n-grams features and with few models such as Naïve Bayes, support vector machine, Bayesian Logistic Regression, and RFDT classifiers. They have obtained the best model as over 93.5% of f-measure score for RFDT classifier with word n-grams feature. Kwok and Wang, has used bag of words algorithm with Naïve Bayes classifier and Grevy et al., has used bag of words algorithm with conjunction to support vector machine. But it has given a high false positive rate of bag of words there for other researches have used more new approaches for extract features for classification models.

. 2.2.4 Trend analysis module

The popularity of social media contents has been increased in recent years there for predicting the trend videos in social media researches is most attractive area in these days. In YouTube trend analysis research area, it has plenty of researches for it but not for Sinhala YouTube video trend analysis. It is very important to select most suitable features for predicting social media contents popularity. Chelaru et al. have mainly aimed on ranking methods used Random Forest, support vector machine, GBRT etc. And extracted most important social features like comments, likes, dislikes and Flavio etc. Cheng et al. have conducted a large scaled of analysis for provide the statistical details of you tube videos. Trend has not a scientific definition but it can be defined with different perspectives such as using different learning algorithms. Peifeng Yin et al has built a “conformer maverick model” to act out a voting algorithm for rank the trend. But it has a huge drawback as it is depending from old voting patterns without concerning the prior information pattern. Therefore, it given results are not believable. Yu et al, has done for research for social marketing field to decide whether it can be popular or not and they have used support vector machine classifier and Naïve Bayes classifier but their approaches narrow to textual contents. Szabo et al. has conducted a traditional way of algorithm model to predict the trend of YouTube videos. S. D. Roy et al.has built a model which is using twitter trending topics. Cha et al. [5] and Gill et al. [14] have done a very earliest analysis for YouTube videos and they have first aimed for crewel data from YouTube, the latter part was based on YouTube traffic statistics. Wattenhofer et al. [20] has researched about correlation between YouTube content popularity with different online social network properties which built among users who use the system. These studies concentrated on sole shots of statistics. Therefor they did not consider for the temporal information which made by video content. As a result of it is not suitable for long term trend analysis. Broxton et al. [4] has established an algorithm to measure a video will be gone viral due to viral fractions. If a video gets a big popularity with a large fraction of likely to be viral. Brodersen et al. have built the same model to measure with the help of other factors, if a viral content gets most views from the same geographical area. And also Borghol et al. have determined the importance of referrer and features of particular contents to the last trending videos. In earliest researches have provided some perceptions to decide which features are important for measure the popularity of videos. But there are still some little doubts about which are the most important features for determine trending videos. Such as popularity features, content and referrer and also system mechanism helps to most of evolution of trending videos. Yin et al. have built a model to determine the rank of trending contents. For do it they have taken into account peoples’ behaviors when molding votes. They have used a Bayesian algorithm for predict the rank. y Lerman and Hogg have made a voting classifier models for predict content popularity and also they have applied their models to news sharing application but the efficiency of not much convincible and debatable. y Szabo and Hubberman and Pinto et al. have measured you tube contents and they have identified long term popularity is likely correlated for early trending at a scale. y Borghol et al. have built a simple prediction algorithms based on linear regression and it was based only from a one feature such s early popularity views at certain period of time. Even it is a very simple model for prediction it has achieved an accurate consistent prediction of popularity for YouTube video contents.

3.1 Introduction

This chapter mainly describes a straightforward explanation about the technologies which are used to implement the solution. For the get and use the YouTube data and find the hate content from those data and find the relationship between trending video and hate contents required technologies and their importance is clearly highlighted as follows

3.2 Technologies Adopted for Implementation

3.2.1 Programming Language

Python Python will be used as the main programming language for the system implementation. Apart from being an open source programming language, python is a great object-oriented, interpreted, and interactive programming language. It has gained high popularity in using as a scientific programming language and efficiency. It provides many libraries for machine learning and image processing approaches. Python is also easy-to-use scripting or automation interfaces.

3.2.2 Development Tools

Spyder

Python is used as the main programming language for the system developments Spyder (short for the Scientific Python Development Environment) IDE is selected to be used. There are very useful features such as Spyder has integrated IPython console, profiler, debugger, documentation viewer, variable explorer, and also IPython console that makes testing small code parts really comfortable.

PyCharm

PyCharm is an integrated development environment for Python programming. Like in other IDEs, PyCharm too includes IntelliSense code auto completion, which assures the implementation without any slips or mistakes. PyCharm also includes its own interpreter so that Python can be used more efficiently in big data and data science projects and supports the widely used libraries such as NumPy etc.

Jupyter notebook

The Jupyter Notebook is an open-source editor which enable you to make and share documents which insist live visualizations, codes, narrative text and equations. It is very famous for exploratory data analysis and feature engineering tasks.

3.2.3 Libraries

SKLearn

Scikit-learn an open source library built on NumPy, SciPy, and Matplotlib. that implements a range of machine-learning, pre-processing and visualization algorithms such as Naive Bayes, K-Nearest Neighbors, K-Means and K-Means++ .

NumPy and Pandas

These two libraries interoperate with each other. NumPy adds support for large, multidimensional arrays and matrices along with high-level mathematical functions whereas Pandas depends on NumPy. Both of these makes the importing and analysis of data easier.

Natural Language Toolkit (NLTK)

It contains text processing libraries for tokenization, stemming, Lemmatization, Punctuation, Character Count, and Word count etc. It is very elegant and easy to work with it. It is an open source library for python programming. This is important to work with natural languages such as wee used YouTube comments and thumbnail texts for measuring hate contents and trend analysis.

Seaborn

It gives interactive plotting facilities for illustrate graphiclly different kind of relationships among features, data distribution, outliers illustrations, feature importance illustrations etc.

Sinling

Singling is language processing tool for Sinhalese which provides collection of NLP tools. They are Sinhala Tokenizer, stemmer, Part-of-Speech tagger, word joiner and word splitter.

OpenCV

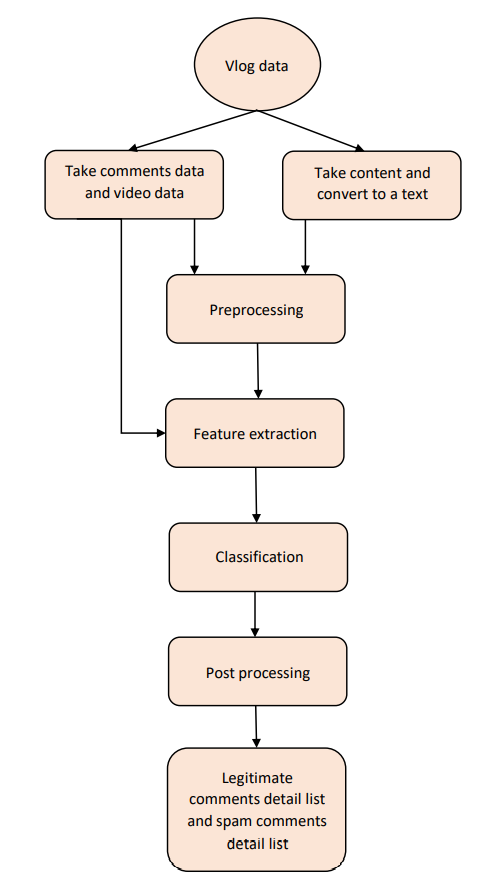
OpenCV is an inbuilt library for python programming language that is very use is writing computer vision applications.

3.3 Summary

This chapter gives an overview of the technologies that will be used in the implementation of the system and why those technologies are chosen. Next chapter illustrates the approach of implementing the system.

High-level Architecture of the Individual Modules

Comment spams filtering sub module



Input = you tube video id is an input for this sub module

Video comments data set = after input you tube video id; sub module starts to download all of Sinhala comments to a file by using you tube data API. It downloads Video id, Video posted date and time, Comment id, Comment content, commented channel id, like count, video title, Comment published date and time, Comment updated date and time.

Video content data set = here download a video contents data set by using pytube3 and you tube dll libraries. It downloads particular video first and then it converted to an audio then using google speech recognition it converts to a text. Then converted text store in a file with its video id.

Pre-processing module = here do few pre-processing according to the requirements. It does remove punctuation marks, remove emoji’s, tokenization as words and character tokenization as pre-processing tasks.

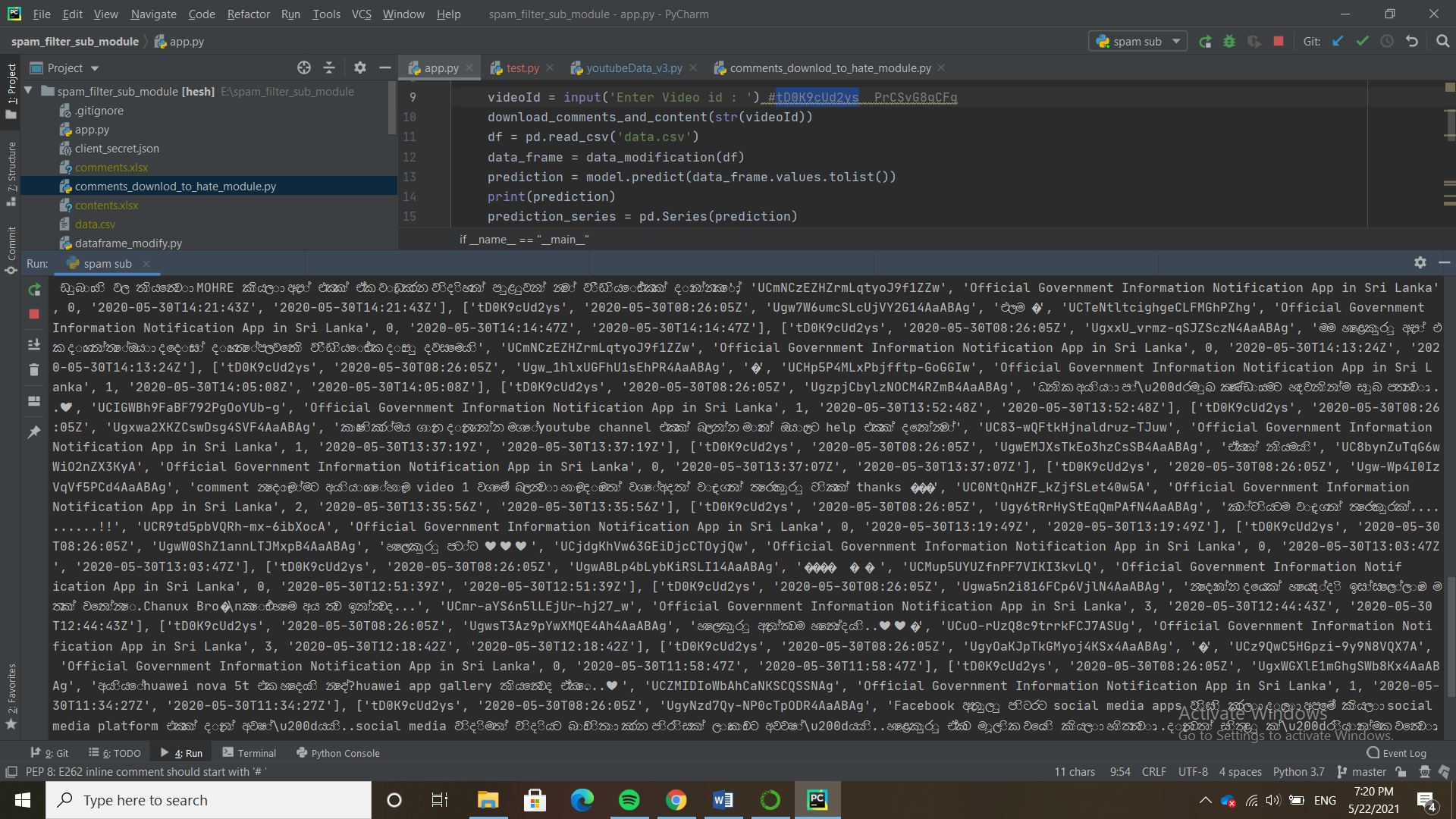
Feature extraction module = in this module, it extracts features from downloaded comments detail files and content detail files such as Similarity between content and comment, Repeating duplicates comments which are posted by same user in a same video, Similarity between comment - comment (surrounding comments), Word count of a comment, Words duplication ratio of a comment, Number of sentences of a comment, Number of punctuation marks in a comment , Black words ration words in a comment, Length of a comment, Availability of url, Availability of YouTube url , Availability of phone numbers in a comment , availability of period marks sequence in a comments, Time gap of Video posted date time and comment posted date time from seconds.

Classification module = in this module, it takes extracted features as an input then it predicts a particular comment as a spam comment or legitimate comment.

Post processing module = this module modifies the output given by classification module according to the requirements of next main sub module (Hate content analysis and clustering module).

Implementation of spam sub module

Here I have implemented my sub module for filter spam comments from YouTube comments section and make a file with only legitimate comments for pass the next sub module.

At the first stage I download YouTube comment details and YouTube content as a text by using YouTube data API and YouTube dll library.

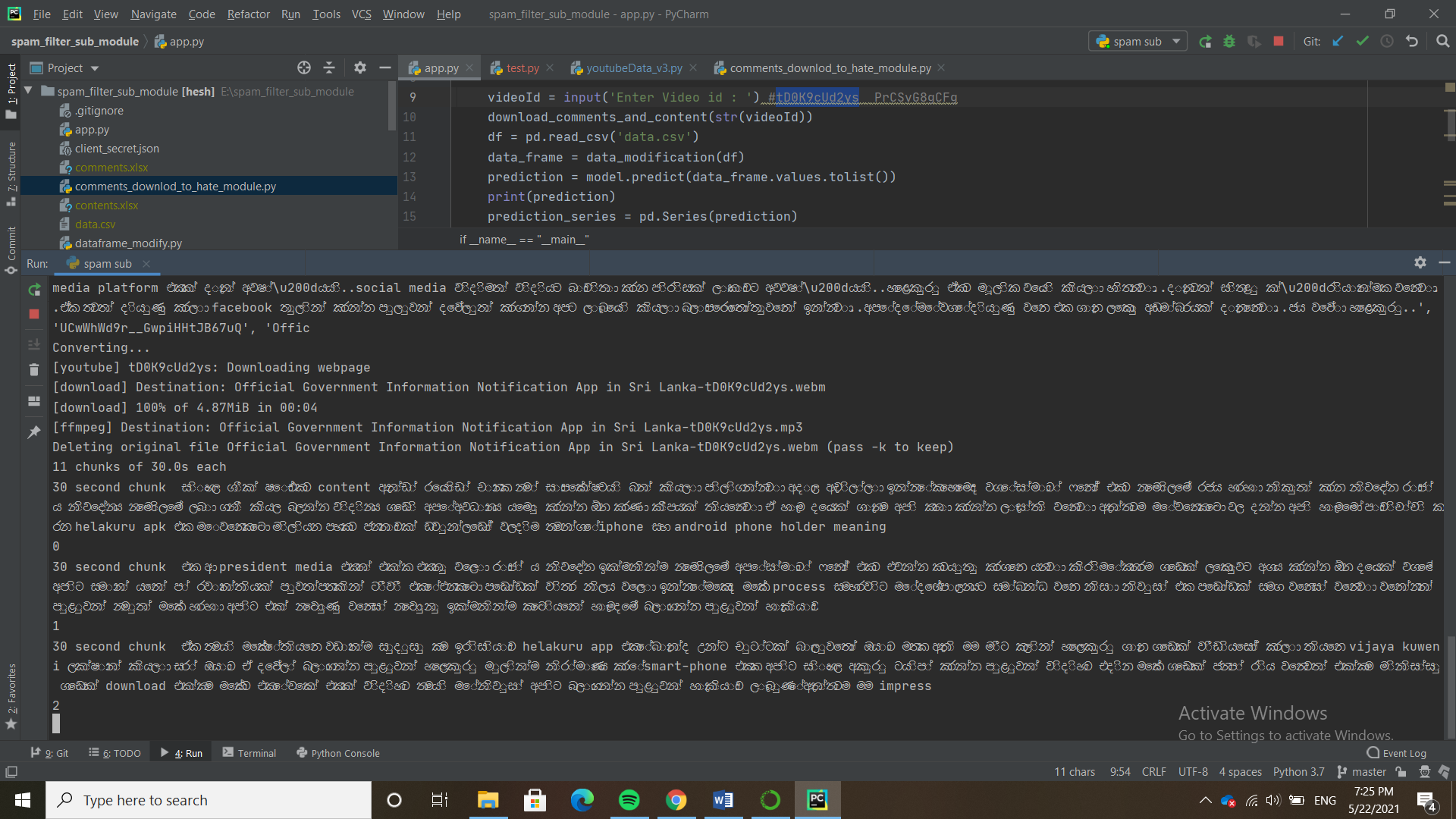
Figure 1.1 YouTube comments details download

Figure 1.2 YouTube content download as a text using google speech recognition

After that I extract features from downloaded comment details file and content text file

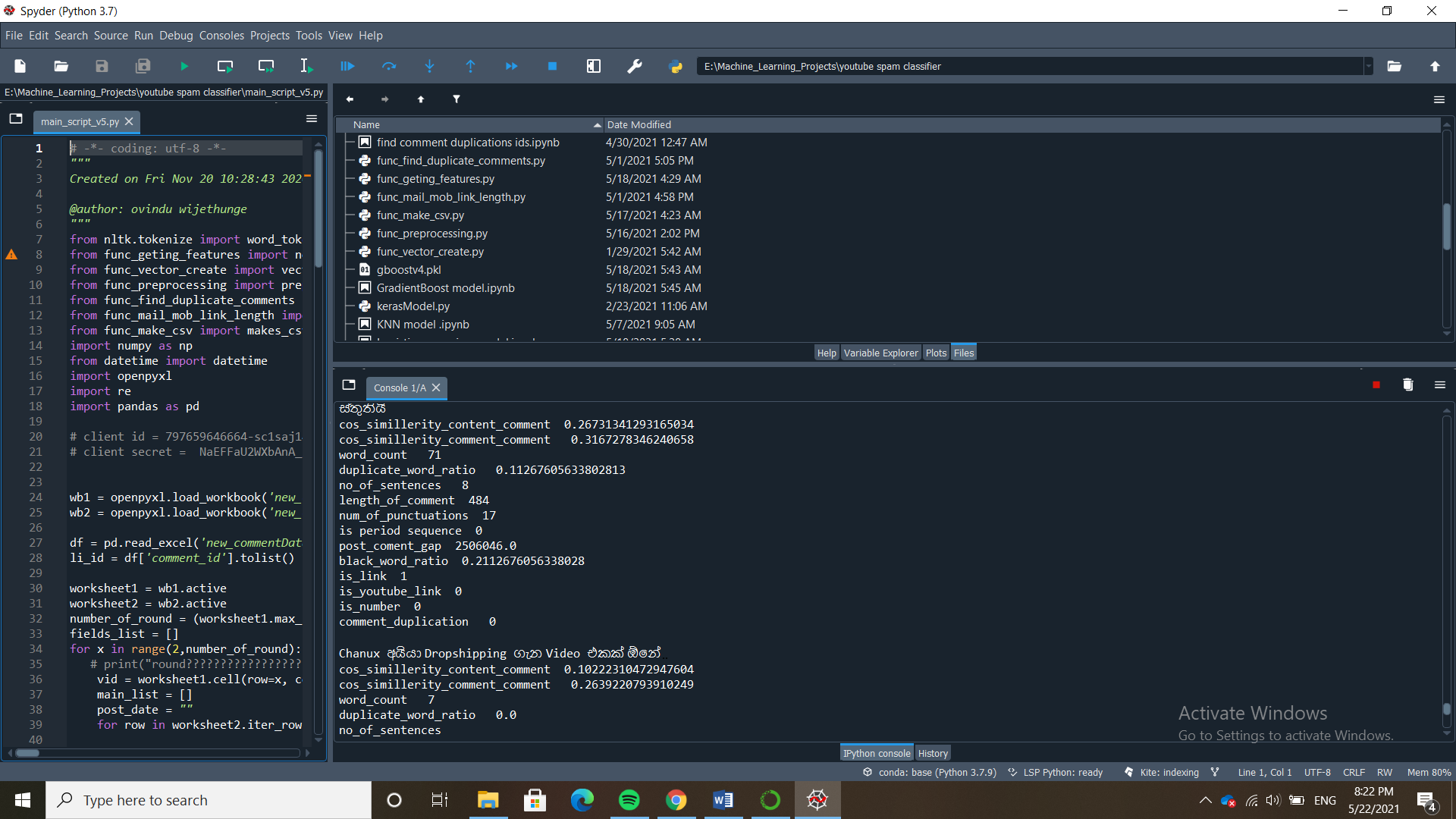


Figure 1.3 features extraction

Then I feed my inputs for my trained model . I have choose gradient boosting model for predict comments.

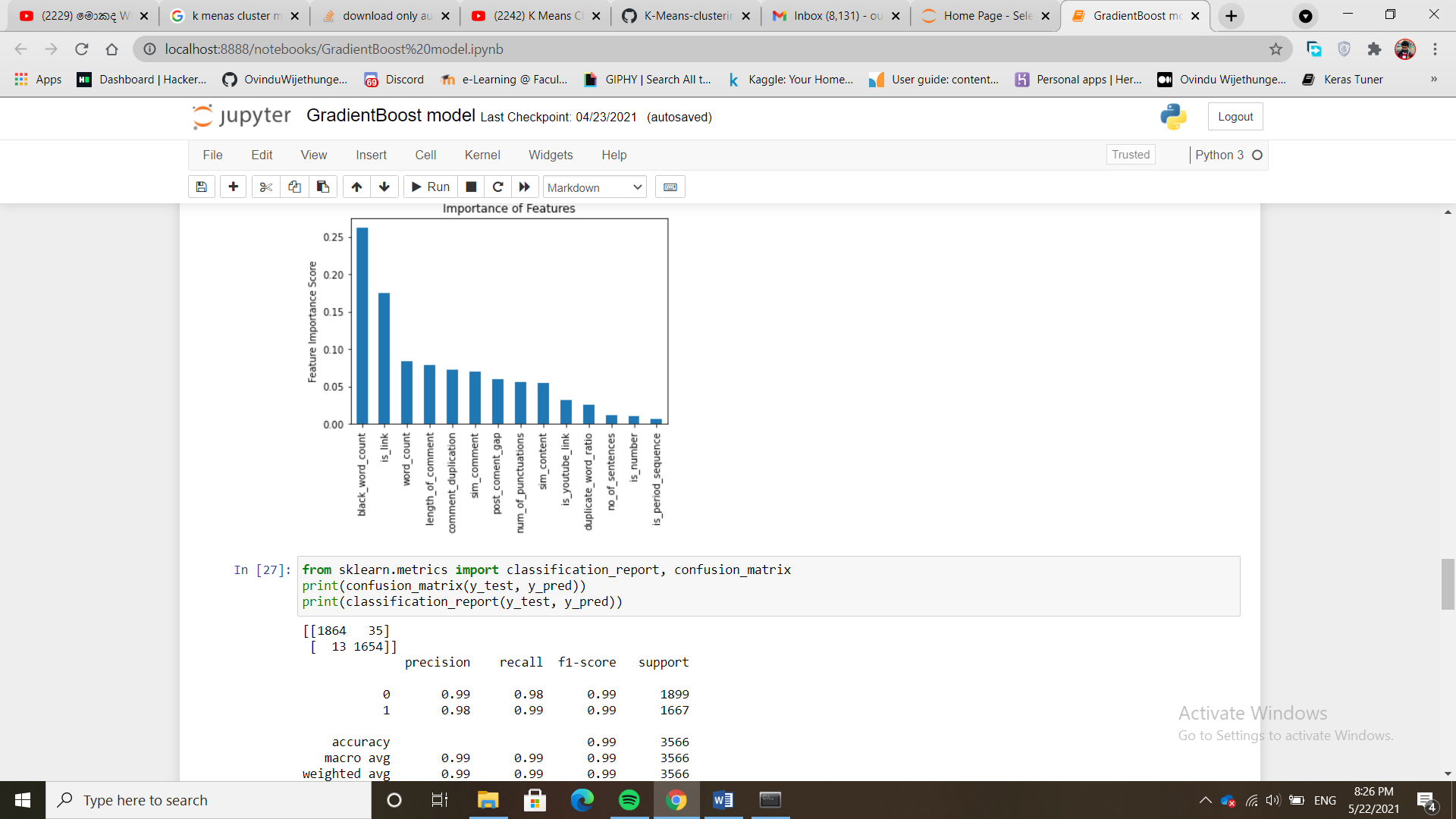


Figure 1.4 gradient boosting model

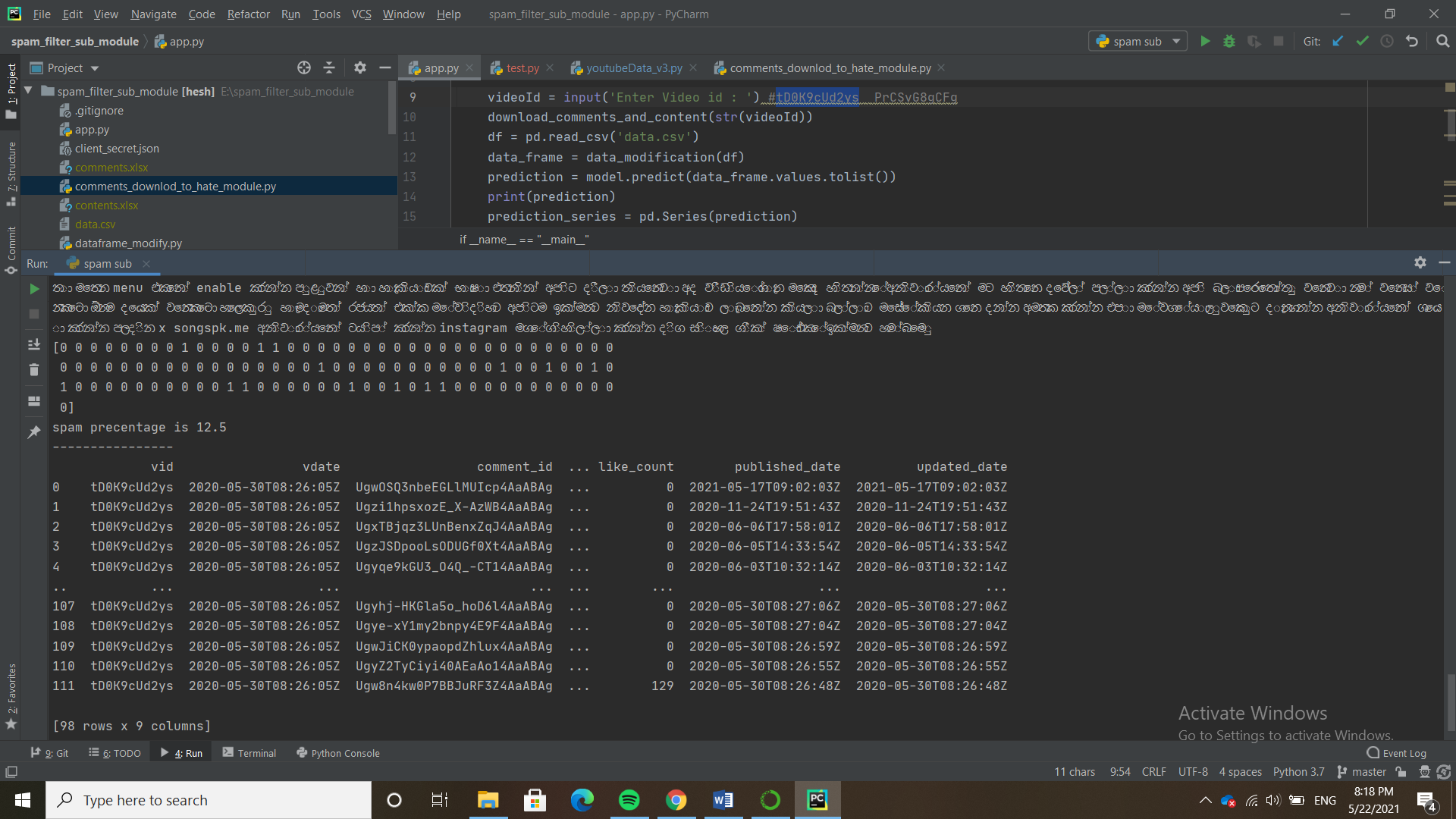


Figure 1.5 predicted output from model

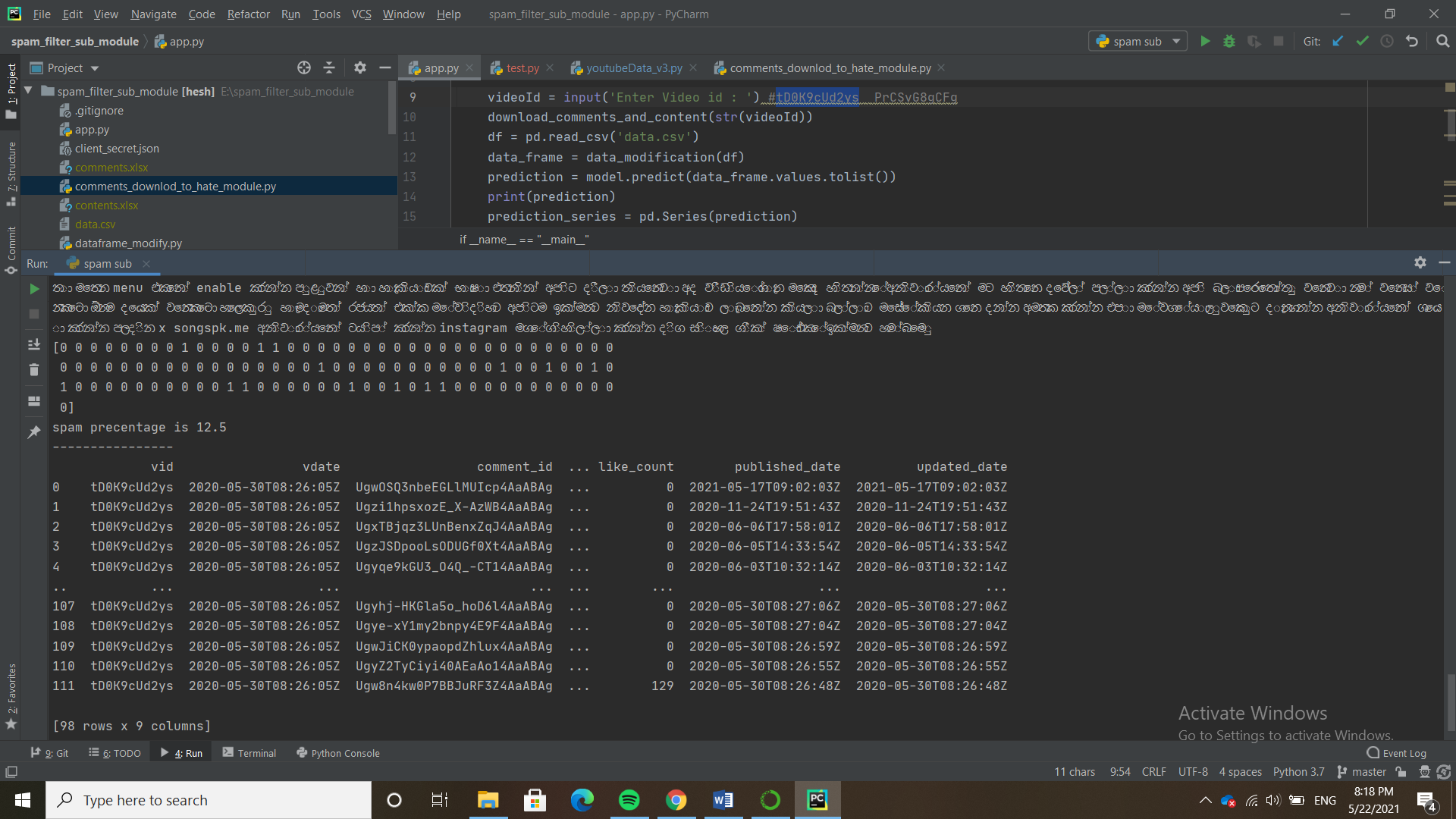


Figure 1.6 legitimate comments data frame as the final output

Evaluation

In spam filtering sub module, it concerns few evaluation techniques to determine which model is the best model for it.

1. Accuracy

Total number of correct predictions / the total number of instances in input (data set)

TP = True Positive

TN = True Negative

FN = False Negative

FP = False Positive

Accuracy = TP +TN / (TP+TN+FP+FN)

True Positive

This indicate the number of instances which have classified as positive class but in real they are belongs to positive class too.

True Negative

This indicate the number of instances which have classified as negative class but in real they are belongs to negative class too.

False Negative

This indicate the number of instances which have classified as negative class but in real they are belongs to positive class.

False Positive

This indicate the number of instances which have classified as positive class but in real they are belongs to negative class. With compared false negative false positive has a big impact of spam comments filtering because of even a spam comment predict as a legitimate comment is not a big issue but a legitimate comment predict as a spam comment is a crime in spam filtering domain.

1. Spam caught rate (SC)

The ratio of correctly classified comments spam and actual comments spam. (true positive rate)

SC = TP / (TP+FN)

TP = True Positive

FN = False Negative

1. Precision

The ratio of correctly predicted positive class comments and total number of predicted positive class comments

Precision = TP/ (TP +FP)

TP = True Positive

FP = False Positive

1. F-measure

F1-score is the harmonic mean of the precision and recall

Precision = True Positives / (True Positives + False Positives)

Recall = True Positives / (True Positives + False Negatives)

F1- score = F-Measure = (2 \* Precision \* Recall) / (Precision + Recall)

1. ROC and AUC curve

This indicates relationship between true positive rate and false positive rate and also area under the curve indicate how much better for a particular model for prediction. When area under the curve is high we ca say it is good fitted model for a particular problem.

We have trained nine machine learning models and evaluate each of models according to the above evaluation criteria.

We have trained,

1. Logistic Regression (LR).
2. Artificial Neural Network (ANN).
3. Random Forest Classifier (RF).
4. Naive Bayes Classifier (NB).
5. K - Nearest neighbor Classifier (KNN).
6. Decision Tree Classifier (DT).
7. Support Vector Machine (SVM).
8. Gradient Boost Classifier (Gboost).
9. Extremely Gradient Boost Classifier (Egboost).

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Algorithm | Accuracy | Recall | Precision | F1 | AUC |
| LR | 0.89 | 0.85 | 0.93 | 0.89 | 0.89 |
| ANN | 0.87 | 0.79 | 0.92 | 0.85 | 0.86 |
| RF | 0.99 | 0.99 | 0.98 | 0.99 | 0.99 |
| NB | 0.81 | 0.64 | 0.93 | 0.76 | 0.80 |
| KNN | 0.96 | 0.98 | 0.94 | 0.96 | 0.96 |
| DT | 0.95 | 0.97 | 0.92 | 0.95 | 0.95 |
| SVM | 0.94 | 0.96 | 0.91 | 0.94 | 0.94 |
| Gboost | 0.99 | 0.99 | 0.98 | 0.99 | 0.99 |
| Egboost | 0.99 | 0.98 | 0.98 | 0.99 | 0.99 |