Active Learning for Spam Detection on YouTube Comment Section

Gonçalo Vasconcelos Correia, João Geirinhas, João Marques
Affiliation
{uc2019216331, uc2019216397, uc2018278463}@student.uc.pt

Abstract. This paper proposes an Active Learning approach for detecting YouTube commentary spam detection. This platform lives in a non-deterministic environment (Internet) that is highly dynamic and continuously evolving. Furthermore, since the users are humans and they are responsible for the comments, we can assume that spam can appear in diverse ways due to human unpredictability. Consequently, supervised learning strategies [5], a support vector machine, and K-nearest [2] neighbor were used to tackle these constraints. However, since the available datasets don´t contain many examples, we believe that an Active Learning (AL) strategy would obtain better results. We will test AL with a multinomial Naïve Bayes that is suitable for classification with discrete features and compare them with the results from the papers since they tested their approaches with the same datasets. There are four datasets with different numbers of spam comments so we can test the efficiency of this approach with a different comment section.

Keywords: Spam Detection · Active Learning · Classification · Comments

1. Introduction

Artificial intelligence is based on the ability of computers to emulate human thoughts and develop tasks in real-world environments. While machine learning concerns the algorithms and technologies used to make systems able to recognize patterns and make decisions. This paper will address a specific type of machine learning, active learning. This is characterized by a learning algorithm where the user can interact with the agent, giving it correct outputs so that the agent can learn the right path. It selects labeling instances that have the most information (be more informative) using them so that it can analyze the other cases better, making it more perfect.

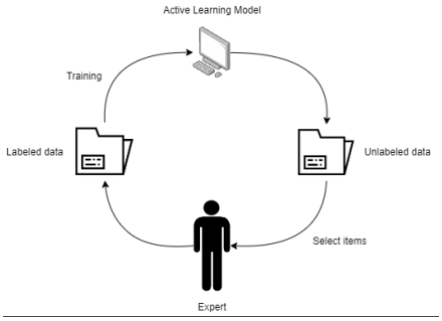

Fig 1- Active Learning Approach

As we can in observe in figure 1, firstly there is an unlabeled data set. After processing this data set some rows (most uncertain) will be selected to be labeled by an expert. Finally, the labeled data will be trained by the active learning model. These steps will be repeated a chosen number of times. The use of various machine learning algorithms may sometimes not be able to be used in certain real-world situations, so sometimes there must be the action of an expert who analyzes the experiences and communicates them to the algorithm. Furthermore, this kind of algorithm is relevant for reducing costs in acquiring data, which explains why AL has so much potential and uses. As mentioned before, this learning algorithm is widely used in traditional Supervised Learning as it saves a lot of resources for machine learning (ML) teams. Furthermore, the main areas where we can find and use this algorithm are mainly, computer vision, image classification, object detection, and image restoration, among others. As can be seen,

this type of ML has a lot of impacts and uses in classification services.
Since the pandemic, Spam campaigns have become much more active. According to Symantec, the most common technique is snowshoeing, which is based on the use of multiple IPs and domains for spam, making it easier for these campaigns to go undetected. Furthermore, around 85% of emails are spam, costing companies an average of $20.5 billion per year [3]. In addition, spam despite not being considered a type of virus can be quite dangerous because there can be scams and frauds. Although these two topics only account for about 2.5% of spam, it is known that 73% of phishing is sent via spam [3]. In fact, YouTube lives in a non-deterministic environment (Internet) that is highly dynamic and continuously evolving. Furthermore, nowadays almost everyone uses this platform to watch videos for entertainment or learning purposes. Moreover, creators post videos on YouTube not only for pleasure but also as a job. Taking into account the data provided by Alexa rankings, Youtube is the second most visited site in the world (about 2 billion users), just behind Google (owner of the YouTube platform). In addition, in 2021 about 51.4% [4] of comments on Youtube were considered inappropriate or in the spam category, and in 2020 about 25.5% [4] of the videos on this platform also fell into this category. These data lead us to believe that is crucial to have an effective way to detect YouTube spam. In fact, after analyzing the different ML algorithms that exist, we considered that Active learning would be the most correct, so we will present a spam classifier that detects messages and assigns them to the Spam or Ham category.

2. Related Work

For this document, some basis of our work will be centered on a paper called “An approach for Spam Detection in YouTube Comment Based on Supervised Learning” [2]. In this paper, the authors give us a background of the role of YouTube in nowadays society and its success. They also wrote about the motives that led users to start polluting this application with spam comments and in consequence the reasons why YouTube requested a tool to deal with this unwanted content. Their approach is based on Supervised learning, and they use a preprocessing technique where they fed their data set into a Tern frequency-Inverse document frequency. In this scope, these two authors gathered various information about multiple algorithms for YouTube spam detection such as Multilayer Perceptrons (MLPs), Support Vector Machine(SVM), Naïve Bayes(NB), Decision Tree(DT), Random Forest(RF), Logistic Regression(LG), and k-Nearest Neighbor (kNN) pipelines respectively. s To compare them they used Evaluation Criteria. This topic will be addressed later, however, these authors were based on the following criteria metrics: Accuracy or Correct Classification Rate (CCR), True Positive Rate (TPR), False Positive Rate (FPR), Precision, Error Rate (ER), and F1 Score.

3. Materials & Data set

To build this model of spam classification we will use a dataset that depends on humans' help to guide the algorithm, determining whether a certain comment is classified as spam or not. For that, we will apply this methodology to the Spambase dataset that can be found in the UCI Machine Learning Repository [6]. In this repository, we have five distinct datasets from five different YouTube videos with a different number of spam comments. Besides that, it is the same dataset used in the paper detailed previously. The following table shows the different datasets:

Dataset	ID	Spam	Ham	Total
Psy	9bZkp7q19f0	175	175	350
Katy Perry	CevxZvSJLk8	175	175	350
LMFAO	KQ6zr6kCPj8	236	202	438
Eminem	uelHwf8o7_U	245	203	448
Shakira	pRpeEdMmmQ0	174	196	370

Table 1 – Data set information

In the following list, we present the attributes/features that compose each data set.

- Comment Id
- Author
- Date
- Content
- Class

The comment Id is a unique value that refers to the id of the user. Furthermore, the author is the username of the user and the date is the time when the comment was published. Moreover, the content is a string that contains what the user wrote. Finally, the class can be 0 or 1, where 0 means that the comment is HAM and 1 means that is SPAM. A comment example is "ilove this so much. AND also I Generate Free Leads on Auto Pilot & You Can Too! http://www.MyLeaerGate.com/moretraffic ". In this case, it is classified as SPAM since it is a self-promoting comment.

To achieve our results and experiments, we will use the modAL tool [7] to provide the active learner. In addition, we will use the Rubrix framework [8] where we can label some of the selected comments to fulfill the purpose of this algorithm.

4. Methods

In this section of the paper, we will present and discuss the methodology that was applied to the project. In addition, we will also discuss our contribution and how the tools we used. The methodology that the project is based on consists of the following steps:

1. Preprocessing of the dataset. We divide the dataset 85% of instances for training and 15% for testing. Moreover, we do a class-balancing approach. We have the same number of ham and spam instances for training.

2. After that, the expert (we) labels a defined number of instances selected according to their uncertainty value. The more uncertain are selected.

3. Then, a query with the labeled data is fed to the learner and trained over a defined number of iterations. In each iteration, the classifier gives its predictions and with that information, we calculate the statistics that are shown in the results section.

4. The process of labeling and feeding queries with the label instances to the active learner is repeated a selected number of times.

In this table, we can observe the number of instances that training, and testing have.

Data sets	Número de SPAM treino	Número de HAM treino	Número de SPAM teste	Número de HAM teste
PSY	149	149	26	26
Katy Perry	149	149	26	26
LMFAO	186	186	50	16
Eminem	190	190	55	13
Shakira	157	157	17	39

Table 2- Training and Testing number of instances

Our code can be seen in a GitHub repository in point nine in the references section[9].

5. Metrics

In order to compare our work with other alternatives we will use some of the metrics that the paper mentioned before used as well as others. Such metrics are:

5.1 Accuracy

Classification Accuracy is what we usually mean when we use the term accuracy. It is the ratio of the number of correct predictions to the total number of input samples. The values can range between 0 and 100 percent (100 being a perfect score). For this metric, as previously mentioned, we'll need to obtain the well-classified predictions (Total Correct) and divide them by the total number of predictions (Total).

$$\frac{Total\ Correct}{Total}$$

5.2 Sensitivity (True Positive Rate)

Sensitivity is a measure of how well a machine learning model can detect positive instances. It is used to evaluate model performance because it allows us to see how many positive instances the model was able to correctly identify. A model with high sensitivity will have few false negatives, which means that it is missing a few of the positive instances. For this metric, we will need to obtain the total number of well-classified SPAM instances (TP) and the number of HAM instances that should be SPAM instances (FN). The sum of sensitivity (true positive rate) and false negative rate would be 1. The higher the true positive rate, the better the model is at correctly identifying the positive cases.

$$\frac{TP}{TP + FN}$$

5.3 Specificity

Specificity measures the proportion of true negatives that are correctly identified by the model. This implies that another proportion of actual negatives will be predicted as positive and could be termed as false positives. This proportion could also be called a True Negative Rate (TNR). The sum of specificity (true negative rate) and false positive rate would always be 1. High specificity means that the model is correctly identifying most of the negative results. For this metric, we will need to obtain the total number of well-classified non-spam instances (TN) and the number of SPAM instances that should be HAM instances (FP).

$$\frac{TN}{TN + FP}$$

5.4 Error Rate

This metric tells you what fraction of predictions were incorrect. In fact, the lower the value, the better the model's classification can predict the response variable's outcomes.

With the results obtained for the FP, FN, TN, and FN values we can calculate the error associated with the experience. It can also be used to calculate the Accuracy or be calculated with it using the formula:
Accuracy = 1 – Error rate.

$$\frac{FP + FN}{TP + FP + TN + FN}$$

5.5 Precision

In fact, Precision is a metric that quantifies the number of correct positive predictions made. It calculates the accuracy for the minority class and is calculated as the ratio of correctly predicted positive examples (TP) divided by the total of true positives plus the false negatives (FN). The range for this metric is between 0 and 1, with being 1 the best value and 0 being the worst. We can calculate the precision with the results obtained for the TP and FN values.

$$\frac{TP}{TP + FP}$$

5.6 F1 Score

This metric, F1 Score, is the Harmonic Mean between precision and recall. The range for this metric is between 0 and 1, with being 1 the best value possible and 0 being the worst. In fact, it tells how precise your classifier is – how many instances it classifies correctly – and how robust it is. The greater this metric is the better the performance of our model.

So that we can calculate the F1 Score we first need to obtain the precision (obtained in the same way as explained in 5.5) and the *Recall* value (this value is the number of correct positive results divided by the number of all relevant samples). The following fraction shows how to obtain this value.

$$\frac{TN}{TN + FP}$$

With the recall and the precision value, we can now calculate the F1 Score metric.

$$\frac{2 \cdot Recall \cdot Precision}{Recall + Precision}$$

6. Results

The results obtained in this section have the following specifications. Only one query with ten labeled instances. The classifier used was the Multinomial Naïve Bayes classifier. This is a Naïve Bayes variant (where the data are typically represented as word vector counts) that is usually used for text classification problems. The training process has a duration of five iterations. Tests were conducted with five different data sets (PSY, Eminem, Shakira, Katy Perry and LMFAO) so that we don't fall into an overfitting problem. For this first experiment, only four data sets could obtain real results since for the LMFAO data set, the selected comments for labeling all belonged to one class.

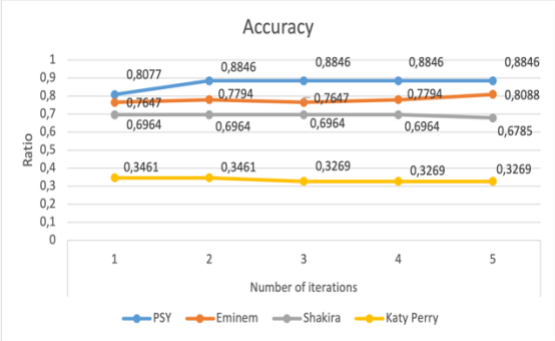

Fig 2 – Accuracy

As we can observe in figure 2, the accuracy obtained for the four different data sets had some variations. For the PSY data set, we obtained really good results and for the Eminem and Shakira we also obtained good values. However, for the Katy Perry data set with only one querie we couldn't obtain positive results.

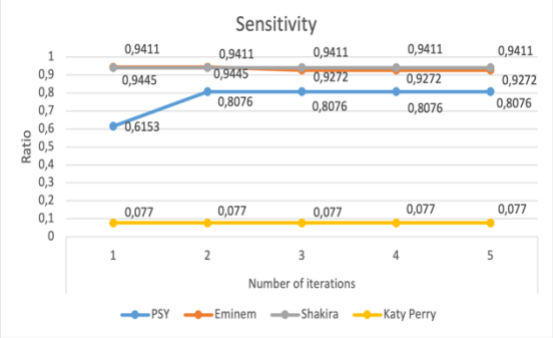

Fig 3 – Sensitivity

Taking into account figure 3, we can see that very high values for two data sets (Eminem and Shakira). This good performance in this metric means that almost every spam comment was classified well, suggesting that the algorithm is highly capable of detecting spam comments. We also obtained good results for the PSY data set. However, the algorithm had not a good performance in detecting spam instances for the Katy Perry data set.

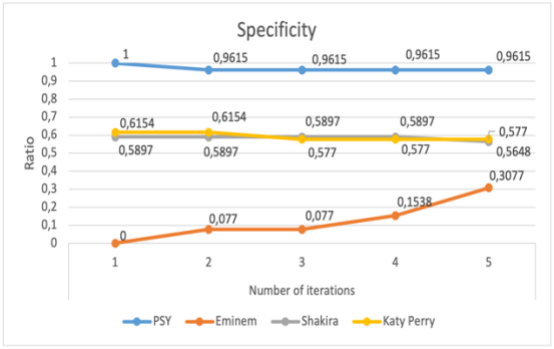

Fig 4 – Specificity

Analyzing figure 4 it is possible to see that for the PSY data set we obtained nearly perfect values. This means that for this data set the algorithm can have a very good performance for detecting non-spam (HAM) instances, indicating that miss-classification problems will not occur. For the Shakira and Eminem data sets, we obtained acceptable values having more than half classified correctly. On the other hand, we couldn't obtain decent results for the Katy Perry algorithm.

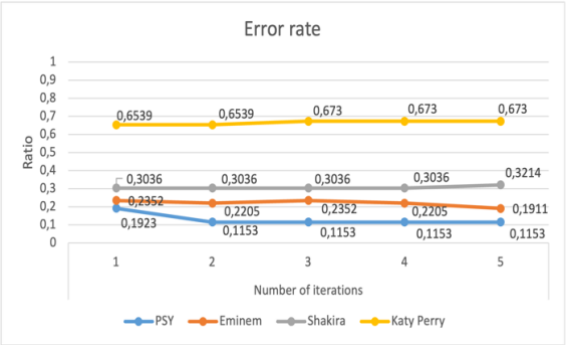

Fig 5 – Error Rate

Considering figure 5 it is possible to see that the results obtained for the PSY, and Eminem were really low, which means that the algorithm had a very good performance in predicting the correct class for all the instances. Also, for the Shakira data set, we can see that the algorithm performs well. However, the values obtained for the Katy Perry data set were very high which means that there is a high probability that the algorithm will fail in predicting the correct class for the instances.

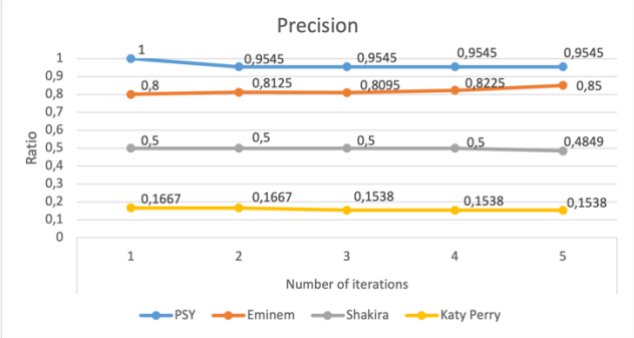

Fig 6 - Precision

As we can see in figure 6 when it comes to this metric for the PSY data set the algorithm has very high precision and for the Eminem data set it obtains very good values. On the other hand, for the Shakira data set we obtain almost positive results, and for the Katy Perry data set the results are low.

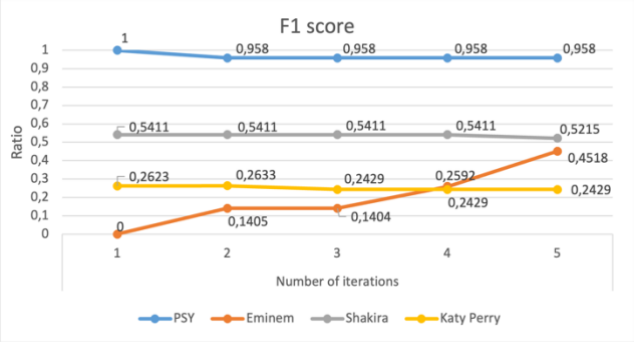

Fig 7- F1 score

Regarding figure 7 we can conclude that for the PSY data set the results of the F1 score metric we're almost perfect, having values of 0.95 or more. We also got positive results for the Shakira data set and almost positive for the Eminem data set. However, for the Katy Perry data set the results didn't go as well as the others.

7. Discussion

Having in mind the results obtained in the results section, we can first conclude that our approach didn't fall into an overfitting model. In fact, we obtained very good results for the PSY data set, and we also got good values for some metrics for the Shakira and Eminem data sets. Some bad results can be explained by a disproportional queue, i.e., there were much more instances of one class than the other. In the Shakira and Eminem data set we can see that we got very high values for sensitivity (0,9411 and 0,9277) but not-so-great values for specificity (0,577 and 5648). This is a result of having more spam instances in the labeled example than ham instances. In addition, comparing our results to the related work paper, we can see that our approach to Active Learning isn't always better than the Supervised Learning approach. In fact, comparing for example the F1 score metric, which represents the robustness of the results, we can see that our best value is 0,958, which is better than their best value of 0,9538. However, for our other data sets, we couldn't obtain as good results as this. However, for example for the sensitivity values above mentioned (0,9411 and 0,9277), the Shakira and Eminem data sets surpassed almost every single result of the Supervised Learning approach. Another relevant stat to observe is the growth of the algorithm over the iterations. For example, for the sensitivity value, the PSY data set initially had a 0,6153 ratio, but in the second iteration increased to 0,8076. On the other hand, the specificity value of this data set decreased. As we can see when one metric increases the other one decreases, occurring a tradeoff between these two metrics.
Finally, we can conclude that having in mind these results, our approach is highly dependent on the quality of the selected comments for the queue that is fed to the active learner.

8. Conclusion

In conclusion, we believe that we fulfilled the primary objective of trying a different approach to the Spam detection problem. Instead of Supervised learning, we tried an Active Learning approach that led us to some better results than the results of the related work paper. However, since not all results were better, and some were not as good as we expected we have the ambition to resolve these problems.

9. Future Work

For future work, we plan on trying to get better results for all our data sets especially the Katy Perry one. We plan on testing the other classifiers such as the Logistic Regression one. Furthermore, we intend to feed more queues to our active learner so we can reach a point where we have a certain number of queues that performs well in all the datasets. Finally, we also have the motivation to try a different number of instances per queue since for the LMFAO data set the queue only had one class present. Finally, we intend to investigate ways to better select the comments for the queues that are fed to the activity learner.

10. References

[1] Top Social Platforms by Monthly Active Users (2021) https://www.visualcapitalist.com/ranked- social-networks-worldwide-by-users/

[2] A.Azis, Cik Feresa Mohd Foozy, Palaniappan Shamala, Zurinah Suradi "Youtube spam comment detection using support vector machine and K–nearest neighbor" https://www.researchgate.net/publication/327249513_Youtube_spam_comment_detection_using_support_vector_machine_and_K-nearest_neighbor

[3] What's On the Other Side of Your Inbox - 20 SPAM Statistics for 2022 https://dataprot.net/statistics/spam-statistics/

[4] 140 Impressive YouTube Stats, Facts, and Figures in 2022 https://thrivemyway.com/youtube-stats/

[5] Amir Ali and Muhammad Zain Amin, "An approach for Spam Detection in YouTube Comments Based on Supervised Learning" https://www.researchgate.net/publication/337826806_An_Approach_for_Spam_Detection_in_YouTube_Comments_Based_on_Supervised_Learning

[6] YouTube Spam Collection Dataset (UCI Machine Learning Repository) http://archive.ics.uci.edu/ml/datasets/YouTube+Spam+Collection#

[7] modAL-python https://github.com/modAL-python/modAL

[8] Active Learning with modal and scikit-learn https://rubrix.readthedocs.io/en/stable/tutorials/05-active_learning.html

[9] Link for the GitHub repository https://github.com/GoncaloVCorreia/Active-Learning-IA