ZHAW at eRisk 2022: Predicting Signs of Pathological Gambling - GloVe for Snowy Days

Samuel Stalder[1], Erman Zankov[1]

[1]ZHAW Zurich University of Applied Sciences, Technikumstrasse 9, 8401 Winterthur

Abstract

This paper describes the participation of our team from ZHAW in the eRisk 2022 shared task. eRisk provides three challenges in the field of "Early Risk Prediction on the Internet". These are text classification problems to detect mental disorders. This paper focuses on addressing the first task whose main objective is the early detection of users at risk of pathological gambling, based on their Reddit history. We addressed this issue by developing models with GloVe as feature extraction and SVM and XGBoost as Classification. It has been found that XGBoost without dart boost has slightly better decision-based evaluation results than SVM.

Keywords

Early Detection System, Natural Language Processing, GloVe, SVM, XGBoost

1. Introduction

The eRisk lab is the "Early Risk Prediction on the Internet" challenge in CLEF. The various tasks relate to textual analysis of social media data. The main goal of eRisk is to study topics such as evaluation methods, metrics, and other factors necessary for the development of research collections and the identification of problems for early risk detection. In turn, the participants, research teams from around the world analyze user data through appropriate methods to detect mental illnesses of any kind early and precisely. With CLEF 2022, eRisk is already entering its sixth round. Early Detection of Signs of Pathological Gambling was first introduced as a challenge in 2021[1] and continues this year. The only difference is that this year they provided labeled training data[2]. The work presented in this paper has been carried out in the course of a bachelor's thesis at the ZHAW Zurich University of Applied Sciences. Our work is based on the solution of team UNSL from 2021[3]. We looked for possible improvements by replacing single subsystems. This paper describes our team's participation in CLEF eRisk 2022 Task 1. The methodology and associated results are presented in this paper, as well as a discussion of future work.

CLEF 2022: Conference and Labs of the Evaluation Forum, September 5–8, 2022, Bologna, Italy

samuel.stalder1@gmail.com (S. Stalder); zankoverman@gmail.com (E. Zankov)

2. Background

2.1. Related Work

Our experiments are based on the results submitted by team UNSL in 2021. They achieved good results last year using three different approaches. In their first approach, they used standard classification models to identify high-risk users using a simple rule-based model. In the second approach, they used a deep learning model for classification and reinforcement learning for early detection. In the last approach, they used a simple and interpretable model that identifies high-risk users and alerts them with a global risk level. Since they did not receive any labeled training data last year, they had to obtain it themselves by using a web crawler that collects gambling and non-gambling users from Reddit.

For our setup, we used the same web crawler, preprocessing, and rule-based early detection system as in the first approach. Our customizations are mainly in the area of feature extraction and classification. For a good contextual extraction of the features, GloVe[4] is used in combination with SVM[5][6] and XGBoost[7], whereas in the original they chose BoW with SVM and doc2vec with logistic regression.

2.2. Preprocessing

In order to test our models, we were given training data by the organizers of this year's challenge. Each file is a XML file that contains all of a user's posts: The user ID for identification and a list of the user's postings. Each writing has a title, date, text, and info field. The title is optional, so many posts have no title. The date contains both the date and time when the post was written. The text represents the post itself, and the info field is just information about the type of post – in this case, all posts are Reddit posts.

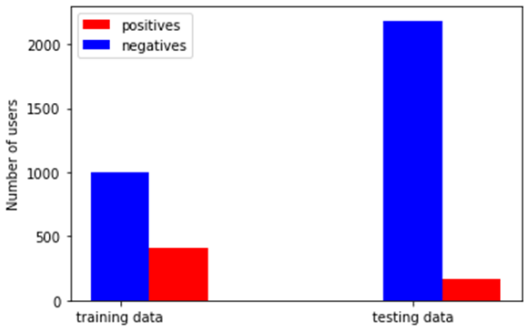

Figure 1: Distribution of positive and negative cases of training data and test data

To compare our results with the results of team UNSL, we used the data received from eRisk only for testing. The training data was collected with the help of the UNSL web crawler, just like in last year's challenge. It is important to note that our participation in this year's challenge is based on our bachelor's thesis, which compares UNSL's feature extraction with our implementation of GloVe. Hence, the training of our models was done with the collected data from the web crawler.

The web crawler uses Reddit's API to download the newest posts from several subreddits. For the acquisition of the positive cases, the subreddit "r/problemgambling" was used. For the negative cases, a mixture of different subreddits, not related to gambling, was used, such as "r/sports", "r/jokes", "r/gaming", "r/politics", "r/news" and "r/LifeProTips". After downloading a user's post, comments written by the same user on the same subreddit were downloaded in order to acquire a sufficient number of posts per user. The limit for the maximum number of posts is 100. The resulting training set contains 411 positive cases and 999 negative cases. We aimed at having a balanced positive to negative ratio.

For the preprocessing of our training and testing data, we used the preprocessing script written by team UNSL. Given that their team had the best scores in several performance metrics in last year's challenge, we decided not to change it in the interest of having the best possible comparison later down the line with our models. The preprocessing steps include setting all characters in the acquired posts to lowercase, replacing Unicode values with their corresponding symbols, HTML codes with their symbols, web links with the token "weblink", numbers with the token "number" and removing repeated white spaces, tabs and new lines. In case the post ends up empty after the above-mentioned preprocessing steps, the token "empty" is used instead. After each post by a user, the token "END_OF_POST" is appended to signal its end. We conducted experiments to ensure that there is no overlap between the training and testing datasets.

2.3. Early Detection System

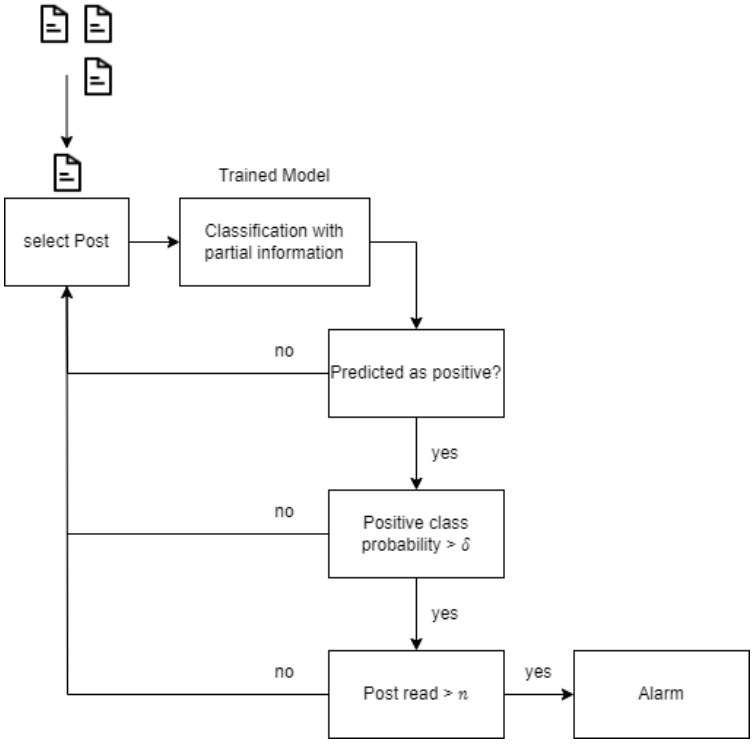

Figure 2: Policy to determine when to raise an alarm. It issues an alarm if the current document is predicted as positive, its probability of belonging to the positive class is greater than threshold δ and the number of posts read is over the threshold n.

The importance of our system is not the classification of online users with gambling addiction, but the early detection of such. In the real world, it is more important to detect emerging signs of addiction rather than simply detecting addiction after it is too late, or significantly more difficult to treat. To do that, we trained our model with a system designed to detect gambling addiction as early as possible based on the user's posts. Hence, the testing of the system differs from the standard testing using a test set. Rather than giving all of a user's posts at once, we are simulating a real-life situation, where each post is given to our system one by one.

Figure 2 visualizes the Early Detection System conceptually. The first step consists of determining if a post is classified as positive and its corresponding probability. Then, we keep track of the number of posts per user our model evaluates. All evaluations made from less than n posts are ignored. This ensures the reduction of false positive cases, i.e., having our model make a prediction based on too little input. If the threshold of n posts is reached and the probability of the post being positive is higher than threshold δ, then this user is classified as positive.

3. Models

This section describes the details of the models used by our team for this task. The following five models differ in hyperparameter and classification method. They all used GloVe for feature extraction. GloVe contains 685k keys, 343k unique vectors with 300 dimensions. Model #0 and #1 use SVM with the parameter C is "1.0", the type of kernel the algorithm used is "rbf", the kernel coefficient is "scale" and the tolerance of stopping criterion is "0.001". Model #2 and #3 use XGBoost with the loss function "log_loss", learning rate of 0.1, number of boosting stages is 100 and a "friedman_mse" criterion. The last model uses an additional dart boost. We decided to take over the thresholds from UNSL's paper, considering that we wanted to make a direct comparison between our models and theirs, by only changing the feature extraction and classification methods.

Table 1
Details of our models. For each run, the type of feature extraction, the type of Classification method and the thresholds δ and n for early detection policy. Our models are designated "stezmo3#[run] (ZHAW)" in this and in the following tables.

Model	Feature Extraction	Classfication	δ	n
stezmo3#0 (ZHAW)	GloVe	SVM	0.85	3
stezmo3#1 (ZHAW)	GloVe	SVM	0.75	10
stezmo3#2 (ZHAW)	GloVe	XGBoost	0.85	3
stezmo3#3 (ZHAW)	GloVe	XGBoost	0.75	10
stezmo3#4 (ZHAW)	GloVe	XGBoost + dart	0.85	5

4. Results

With our 5 models we have processed the first 30 posts, due to time constraints for all 2079 people. For an optimal outlook, all posts of the 2079 people would have had to be evaluated. Unfortunately, the prediction with our models took too long to fully process all posts, and additionally there were some server problems.

Table 2 shows the performance and the number of processed posts of each team. In total, only three of the nine participating groups were able to process all posts. We processed 30 posts with a total time of 12 hours and 30 minutes. With 5 min, the prediction time is one of the slowest. Only team SINAI and team RELAI reported slower execution times. The fastest was team UNED-NLP with 6s.

Early classification decision-based performance: Table 3 shows the results obtained for the decision-based performance metrics. As can be observed, our team was able to achieve solid average performance for the metrics P, R, F1, ERDE5 and ERDE50. Our model #2 was able to achieve slightly better results compared to the rest of our models. Normally, precision and recall have an inversely proportional behavior. So, if one of them is high, the other one is low. This is also reflected in our results. The recall measure is high, but the precision is low. This means that our models could find almost all users addicted to gambling, but many of the users classified as addicted to gambling do not belong to this class. A good mix between P and R is

Table 2
Details of the participating teams. For each team, the number of submitted models, number of processed posts and their total time is shown. In addition, the processing time for each post was calculated.

Team	#models	#posts	Total	Per post
UNED-NLP	5	2001	17:58:48	00:06
SINAI	3	46	108:54:03	47:20
BioInfo_UAVR	5	1002	22:35:47	00:16
RELAI	5	109	183:27:25	20:11
BLUE	3	2001	85:15:25	00:51
BioNLP-UniBuc	5	3	00:37:33	02:28
UNSL	5	2001	45:53:51	00:16
NLPGroup-IISERB	5	1020	381:30:48	04:29
stezmo3 (ZHAW)	*5*	*30*	*12:30:26*	*05:00*

evident in F1. Model UNED-NLP#4 and UNSL#1 in particular stand out here. In the area of early detection with the metrics ERDE5, ERDE50 and latency-weighted F1, models UNED-NLP#4, SINAI#0 and UNSL#1 are the best.

Table 3
Decision-based evaluation results. For comparison, the best models in the range P, R, F1, ERDE5, ERDE50 and latency-weighted F1 are listed together with the median and mean values of all models. The best values among all participating models are shown in bold.

Team	P	R	F1	ERDE5	ERDE50	latencyTP	speed	latency-weighted F1
UNED-NLP#4	0.809	0.938	**0.869**	0.02	**0.008**	3	0.992	**0.862**
SINAI#0	0.425	0.765	0.546	**0.015**	0.011	**1**	**1**	0.546
BioNLP-UniBuc#0	0.039	**1**	0.075	0.038	0.037	**1**	**1**	0.075
UNSL#1	0.461	0.938	0.618	0.041	**0.008**	11	0.961	0.594
NLPGroup-IISERB#1	**1**	0.074	0.138	0.038	0.037	41.5	0.843	0.116
stezmo3#0 (ZHAW)	*0.116*	*0.864*	*0.205*	*0.034*	*0.015*	*5*	*0.984*	*0.202*
stezmo3#1 (ZHAW)	*0.116*	*0.864*	*0.205*	*0.049*	*0.015*	*12*	*0.957*	*0.196*
stezmo3#2 (ZHAW)	*0.152*	*0.914*	*0.261*	*0.033*	*0.011*	*5*	*0.984*	*0.257*
stezmo3#3 (ZHAW)	*0.139*	*0.864*	*0.24*	*0.047*	*0.013*	*12*	*0.957*	*0.229*
stezmo3#4 (ZHAW)	*0.16*	*0.901*	*0.271*	*0.043*	*0.011*	*7*	*0.977*	*0.265*
Median	0.116	0.963	0.205	0.037	0.015	2.75	0.993	0.211
Mean	0.226	0.846	0.282	0.03	0.0209	4.82	0.985	0.300

Ranking-based performance: Table 4 shows the results obtained for the ranking-based performance metrics. Team UNED-NLP, BLUE and UNSL achieved the best results here. The only differences here are seen in NDCG100. For one writing processed, model BLUE#0 performs best, for 100 writings processed, model UNSL#1 performs best, and for 500 and 1000 writings processed, model UNED-NLP performs best. Our models #2, #3 and #4 are better than the average after processing the first writing. For the metrics P10, NDCG10 and NDCG100 after 100, 500 and 1000 writings processed, we have a score of 0. The reason for this is that we processed a total of only 30 out of 2001 posts.

Table 4
Ranking-based evaluation results. The values obtained for each metric are shown after processing 1, 100, 500, and 1000 writings. The best values among all participating models are shown in bold.

	1 writing			100 writings			500 writings			1000 writings		
Team	P10	NDCG10	NDCG100	P10	NDCG10	NDCG100	P10	NDCG10	NDCG100	P10	NDCG10	NDCG100
UNED-NLP#4	**1**	**1**	0.56	**1**	**1**	0.88	**1**	**1**	**0.95**	**1**	**1**	**0.95**
BLUE#0	**1**	**1**	**0.76**	**1**	**1**	0.81	**1**	**1**	0.89	**1**	**1**	0.89
UNSL#1	**1**	**1**	0.7	**1**	**1**	**0.9**	**1**	**1**	0.92	**1**	**1**	0.93
stezmo3#0 (ZHAW)	*0.1*	*0.06*	*0.26*	*0*	*0*	*0*	*0*	*0*	*0*	*0*	*0*	*0*
stezmo3#1 (ZHAW)	*0.1*	*0.06*	*0.26*	*0*	*0*	*0*	*0*	*0*	*0*	*0*	*0*	*0*
stezmo3#2 (ZHAW)	*0.5*	*0.58*	*0.61*	*0*	*0*	*0*	*0*	*0*	*0*	*0*	*0*	*0*
stezmo3#3 (ZHAW)	*0.5*	*0.58*	*0.61*	*0*	*0*	*0*	*0*	*0*	*0*	*0*	*0*	*0*
stezmo3#4 (ZHAW)	*0.5*	*0.58*	*0.61*	*0*	*0*	*0*	*0*	*0*	*0*	*0*	*0*	*0*
Median	0.3	0.19	0.32	0	0	0.1	0	0	0	0	0	0.02
Mean	0.41	0.41	0.37	0.30	0.29	0.29	0.21	0.20	0.23	0.22	0.21	0.23

5. Conclusion and future work

In this paper we presented the results of our team's participation in the eRisk2022 Task 1. Due to complications, not all posts could be evaluated. Server problems and the long time needed for the prediction, due to the low processing power of our hardware, made the whole evaluation difficult. A complete evaluation might have yielded better results. In addition, a complete ranking-based evaluation would have been possible. Despite these obstacles, we were able to achieve competitive results. With stronger hardware and more time, a complete prediction could have been done.

This year, there was a training set for task 1 from the organizer. However, we did not use it for training but only for testing. Possibly better results could have been achieved by mixing the data from the web crawler with the training data from eRisk and using an 80-20 split.

The potential of our approach has not been exhausted and future work could be about improving the decision-based performance with hyperparameter tuning of the thresholds δ and n. In addition, the preprocessing could be improved by filtering out the non-English sentences. Another approach would be to divide the prediction into two subpredictions. In the first step, it is determined whether the person is at risk of addiction and in the second step whether it is pathological gambling. This would be especially beneficial in scenarios where all different types of addictive behavior should be detected.

6. Acknowledgments

This research was enabled by the support of Prof. Dr. Martin Braschler at ZHAW Zurich University of Applied Sciences.

References

[1] J. Parapar, P. Martín-Rodilla, D. E. Losada, F. Crestani, Overview of eRisk at CLEF 2021: Early Risk Prediction on the Internet (Extended Overview), Technical Report, 2021. URL: https://www.dc.fi.udc.es/~parapar.

[2] J. Parapar, P. Martín-Rodilla, D. E. Losada, F. Crestani, Evaluation Report of eRisk 2022: Early Risk Prediction on the Internet, Technical Report, 2022. URL: https://www.dc.fi.udc.es/~parapar.

[3] J. M. Loyola, S. Burdisso, H. Thompson, L. Cagnina, M. Errecalde, UNSL at eRisk 2021: A Comparison of Three Early Alert Policies for Early Risk Detection under Creative Commons License Attribution 4.0 International (CC BY 4.0) (2021). URL: https://github.com/jmloyola/unsl_erisk_2021http://ceur-ws.org.

[4] J. Pennington, R. Socher, C. D. Manning, GloVe: Global Vectors for Word Representation, Technical Report, 2014. URL: http://nlp.

[5] B. E. Boser, I. M. Guyon, V. N. Vapnik, A Training Algorithm for Optimal Margin Classifiers, Technical Report, 1992.

[6] J. Piskorski, J. Haneczok, G. Jacquet, New Benchmark Corpus and Models for Fine-grained Event Classification: To BERT or not to BERT?, Technical Report, 2020. URL: https://tac.nist.gov/2017/KBP/Event/index.html.

[7] T. Chen, C. Guestrin, XGBoost: A scalable tree boosting system, in: Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, volume 13-17-August-2016, Association for Computing Machinery, 2016, pp. 785–794. doi:10.1145/2939672.2939785.