

# HULAT-UC3M at Task1@MentalRiskES 2025: Detecting Gambling Disorders Using Machine Learning Approaches

Javier Campos-Molina<sup>1</sup>, Paloma Martínez<sup>1</sup>

<sup>1</sup>Computer Science and Engineering Department, Universidad Carlos III de Madrid, Leganés, Madrid, Spain

## Abstract

This paper describes the participation of HULAT-UC3M research group at Task 1: Risk Detection of Gambling Disorders at MentalRiskES 2025 shared task [1] at Iberlef 2025 [2]. Three approaches based on machine learning algorithms have been tested in classifying users in high risk or low risk in gambling addiction. The first and second approaches (run 0 and run 1) are based on a SVM classifier that receives embeddings corresponding to users messages. Third system (run 2) integrates a data augmentation approach to extend training data, a sentence vectorization and a Random Forest model to predict user risk. Best performance is achieved with third approach with a macro-F1 of 0.488 that ranks HULAT-UC3M 9th in the ranking of results.

## Keywords

Natural Language Processing, Machine learning, Gambling detection, Classification, Data augmentation, LLM, Spanish

## 1. Introduction

Gambling-related harm has become a significant public health concern worldwide, affecting millions of individuals and communities. According to recent studies, problematic gambling behaviors can lead to serious financial, psychological, and social consequences, and are often associated with other mental health issues such as depression and substance abuse [3, 4]. It has also been detected that it is a mental problem in adolescents and young women according to several studies that relate gambling with the probability of consuming drugs and being alcoholic [5, 6].

Early identification of individuals at risk of developing gambling problems is essential for timely intervention and the development of effective prevention strategies. Beyond traditional clinical approaches, the growing availability of digital data from forums, social media, and messaging platforms enables the creation of automated systems for the detection and monitoring of gambling-related harm. Recent research has explored the application of large language models (LLMs) [7, 8] and other machine learning techniques [9, 10] to detect gambling-related risks from user-generated content. This work contributes to that direction by exploring the use of machine learning and LLM generation to classify gambling-related behaviors in Spanish-language messages collected from the Telegram platform.

The main objective of the MentalRiskES competition [1] is the early detection of gambling in users. In particular for task 1, that is the task in which we are competing, we will receive round of users and the goal is to detect as soon as possible if the user is a gambler or is not. The dataset provided for training gives us some users related to gambling and others that do not [11]. We have 178 users with label 0 (considered as non gambling) and 172 users with label 1 (considered as gambling).

This paper will be divided into several sections: first, a brief related work in analysis of gambling in Internet users or tasks related to the mental problems associated with gambling. It will continue with a general description of the mentalRisk task [7], followed by a detailed description of the proposed solution. Finally, results, discussion and some improvements are described.

---

*Iberlef 2025, September 2025, Zaragoza, Spain*

\*Corresponding author.

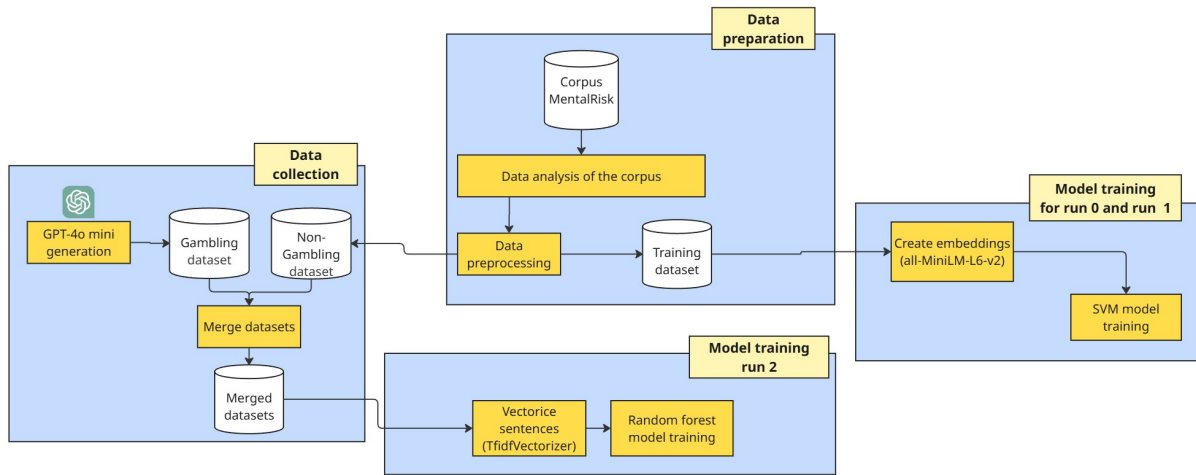
† These authors contributed equally.

✉ 100472233@alumnos.uc3m.es (J. Campos-Molina); pmf@inf.uc3m.es (P. Martínez)

id 0000-0003-3013-3771 (P. Martínez)



© 2025 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).



**Figure 1:** High level architecture of the models training for the runs submitted to MentalRiskES task 1 by HULAT-UC3M team

## 2. Related Work

This task focuses on detecting gambling-related content in user messages across different social media platforms. Since this topic was not addressed in previous editions of the MentalRisk challenge, we had to look for external resources related to gambling detection. In the related literature, we find works that apply machine learning techniques [9, 10], and others that explore deep learning approaches using large language models (LLMs) and transformer-based architectures [7, 8].

The goals of research in the gambling detection domain vary widely. Some works aim to detect gambling-related messages, as we intend to do in this task [7, 9, 10], while others focus on identifying gambling-related locations such as casinos using voice recordings [8]. Data collection strategies also differ significantly—from using user surveys [9] to harvesting messages from online forums [7].

Some of these projects apply deep learning approaches within the NLP field, including the use of recurrent neural networks (RNNs) such as Long Short-Term Memory (LSTM) and Gated Recurrent Units (GRU). Variants like Contextual LSTM (CLSTM) have shown good performance in capturing long-term dependencies in comparison to LSTM [12], while other studies compare multiple models of different variations of GRU and LSTM [13], remarking the strong performance of convolutional neural networks (CNNs), which are frequently used in similar tasks [14].

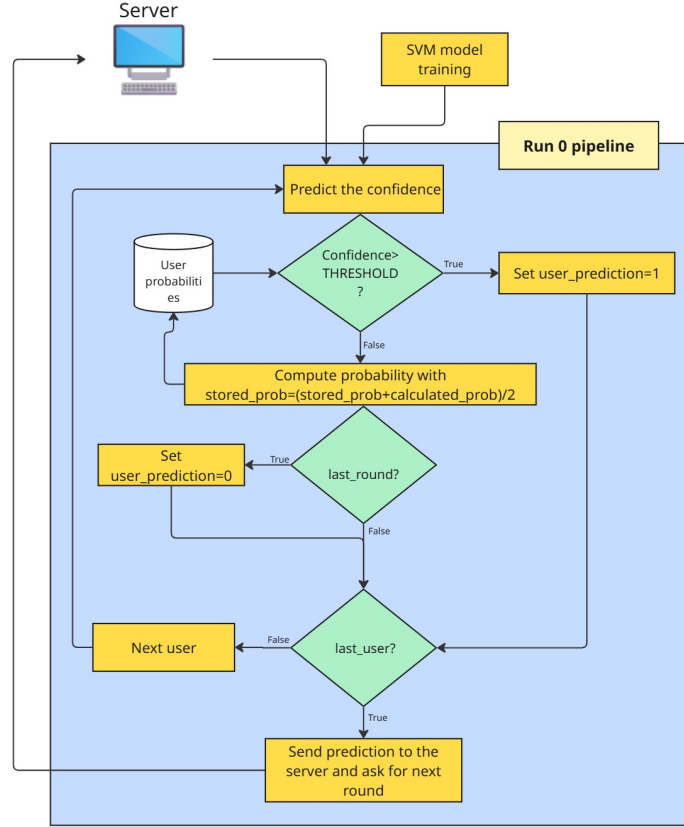
## 3. Method and system description

Before the development of our solution, the team performed a data analysis of the labeled data given by the MentalRiskES organization to check that there is enough data to compute a model and checking that all the labels were in the correct format.

After this process, we can start discussing about the best model to approach this classification task. We decided to work on three architectures differentiated by the machine learning model used in prediction and the data used in training these models. Additionally, we tried two approaches to analyze inputs from the server. One of the methods sentence-by-sentence using the probabilities for each messages or in an incremental way taking into account the whole text received for each user until that round. The figure 1 represents a good overall view of all the processes followed for submitting our results.

### 3.1. Using embeddings and SVM to classify

For architectures used in generating runs 0 and 1, we used the same model, that was trained with a dataset that was computed concatenating all the sentences from an user and its corresponding label, 1 if



**Figure 2:** High level architecture of the mechanism of communication for the run 0 submitted to MentalRiskES task 1 by HULAT-UC3M team

its an user labeled ad gambling and 0 if the user is not labeled as gambling. The final dataset is composed of two columns, the first one will be the whole text for the user and the second one will be the label. For the text column, we used the sentence transformer “all-MiniLM-L6-v2”<sup>1</sup> to create the embeddings before training the model. The chosen model was Support Vector Machine (SVM), and we took the implementation directly from sklearn library<sup>2</sup>. The model was computed with the hyperparameter probabilities set to True as it will be needed for the mechanism of detecting the predictions. The rest of the parameters were the standards fixed by the library. We chose SVM because it regularly has a good performance in binary classification and because it has the hyperparameter probabilities that we will need for the predictions as said before. The training of the model are represented in the figure 3

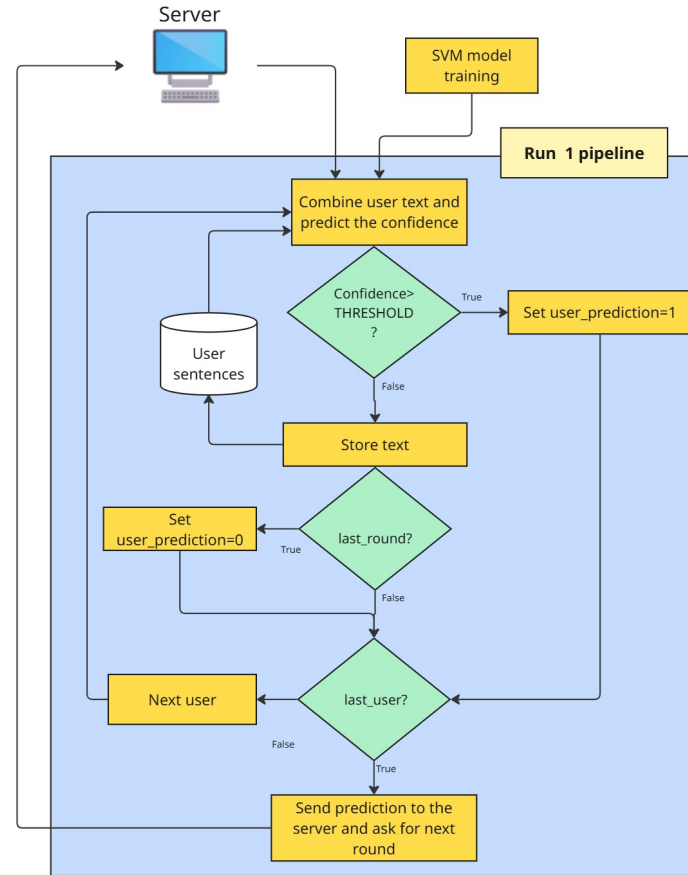
In the case of run 0, we predict the probability that a received message is related to gambling using the SVM model. When the first round is received from the server, the system computes the probability of gambling for each user. If the probability exceeds a predefined threshold, the system outputs a prediction of 1 for that user in the first round. In the other hand, if the probability does not exceed the threshold, it is stored in a dictionary to be used in the next round. This procedure is applied to all users participating in the current round. In cases where no message is received from a user, or the message is empty, the system assigns a prediction of 0 for that user. In the following round, upon receiving a new message from the user, a new probability is computed. The stored and current probabilities are then averaged using next formula:

$$\text{Updated Probability} = \frac{\text{Stored Probability} + \text{Current Probability}}{2}$$

If the updated probability exceeds the threshold, the system sends a prediction of 1; otherwise, the

<sup>1</sup><https://huggingface.co/sentence-transformers/all-MiniLM-L6-v2>

<sup>2</sup><https://scikit-learn.org/stable/modules/svm.html>



**Figure 3:** High level architecture of the mechanism of communication for the run 1 submitted to MentalRiskES task 1 by HULAT-UC3M team

new value is stored again in the dictionary for use in the following rounds. This process continues until all users have been sent. The threshold selected for this task was 0.6, as it provided a suitable balance between precision and recall in preliminary experiments.

For run 1, the model predicts the probability based on the received text. As in run 0, if the probability exceeds the threshold, a prediction of 1 is sent for that user. However, instead of storing the probability, we store the text itself. In the next round, the model processes the concatenation of all previously stored text with the new text, treating it as a single input. This allows the model to make predictions based on the full message history up to that point. As in run 0, if a user does not appear in the current round or their message is empty, a prediction of 0 is sent for that user. The same threshold value is used for this approach. The formula representing the text input used for prediction is:

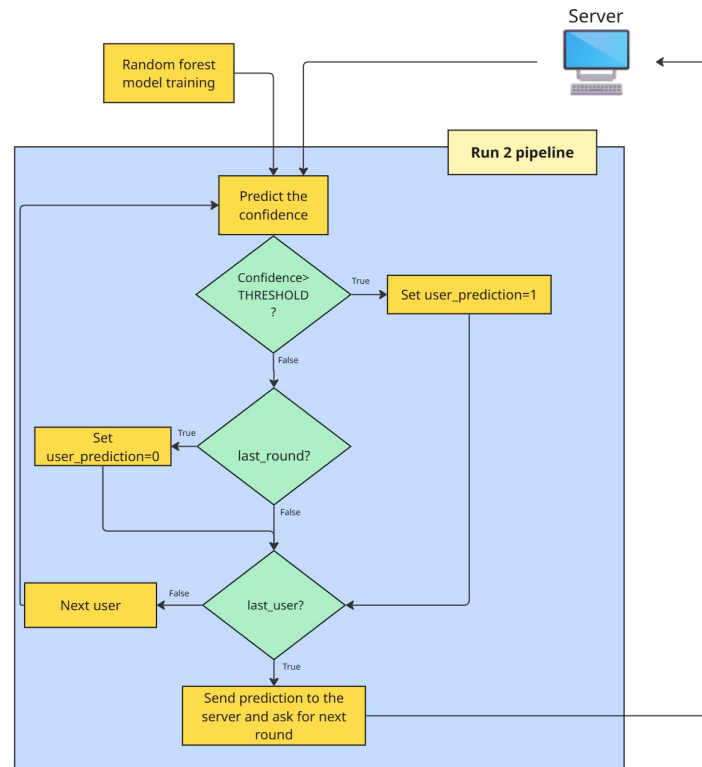
$$\text{Predicted Text} = \text{Stored Text} + \text{New Text}$$

### 3.2. Using Data augmentation and Random Forest to classify

For run 2, a data augmentation approach is implemented. GPT-4o mini model<sup>3</sup> is used for generating labeled messages instead of labeled users. In this case, we asked two different prompts to the model. Firstly, the prompt includes an instruction to generate random sentences about gambling in users; following an few-shot approach, the prompt was the following:

Necesito que generes frases relacionadas con el juego. Las frases deben ser mensajes de usuarios en redes sociales, chats etc, y que tengan relación con alguno de los tipos de gambling. Esto serían

<sup>3</sup><https://chatgpt.com/>



**Figure 4:** High level architecture of the mechanism of communication for the run 2 submitted to MentalRiskES task 1 by HULAT-UC3M team

algunos ejemplos de frases que necesito generar: (I need you to generate phrases related to gambling. The phrases should be messages from users on social media, chats, etc, and should be related to one of the types of gambling. Here are some examples of phrases I need to generate:)

- Aposté a la victoria (Betting on victory).
- Confié en el pick (Trust the pick).
- Vi una cuota buenísima. (I saw a huge fee)
- Entré fuerte en esta apuesta. (I went strong on this bet)

Some examples of generated messages are:

- Apuéstale todo al rojo (go all in on the red).
- Perdí todo apostando al Real Madrid (I lost everything betting on Real Madrid).
- No puedo dejar de entrar a Bet365 (I can't stop entering Bet365).

After reviewing the messages generated by the GPT model, we decided to expand the dataset by generating sentence variations using Python with the GPT model. The script produces similar sentences by substituting certain words with related alternatives. For example, for the sentence "No puedo dejar de entrar a Bet365" we generated more sentences substituting the word "Bet365" for other ones related like "Betway" or "Codere". All these sentences were created using regular expressions and using the model GPT-4o for the code generation that creates the dataset. Afterwards, a list with related words to each item was created. Some examples of sentence patterns are listed below: :

- Aposté al EQUIPO (I bet on TEAM)
- Aposté a CUOTANUMBER (I bet a QUOTANUMER)
- Confié en JUGADOR (I trusted PLAYER)

where:

- EQUIPO (TEAM) will be substituted for team names as Real Madrid, Manchester City etc.
- CUOTANUMBER (QUOTANUMBER) will be substituted by numbers as 1.5, 2.5 etc
- JUGADOR (PLAYER) will be substituted for player names as Messi, Haland etc.

We thought this approach would be good if data generation works properly because we need to predict for each message and we did not have labeled messages in the dataset provided by MentalRiskES organizers. It is important to remark that some of particular gambling messages will be not detected as the generation of those messages introduced to the model training were focused on football, crypto and casinos that are the most popular methods of gambling nowadays.

In order to add to the model some non-gambling messages we just took non-gambling users and we labeled the messages as non-gambling. It wasn't possible to do the same with gambling as a user can have some messages regarding gambling and others that do not have, so this could have created a very unstable model. After the data augmentation process, we used the vectorizer called TfidfVectorizer<sup>4</sup> with max features hyperparameter equal to 5000 and the stop words Spanish dictionary<sup>5</sup> in order to convert the text to numbers, and finally we computed the model random forest classifier with hyperparameters class weight to balanced and number of estimators to 100 from the implementation in sklearn<sup>6</sup>.

Finally, we have the random forest model for the run 2. In this case neither probabilities nor text messages are stored as the main idea when training this model was to predict over each message individually as the model was trained in such a form. When receiving each round the model is used to compute the probability of being a message of gambling or not and if it's higher than the threshold it sends a prediction of 1 for that user. The prediction of 0 is then sent for the user that is not on the round or the user that we received an empty list. The threshold applied for this run was 0.4 and the reason of the threshold is explained in section 4.

## 4. Results and discussion

In Table 1, the results given by the MentalRiskES organizers for the participants for task 1 are shown, in order to compare our performance with the other participants of the task. The ranking has been defined using the metric macro F1.

Referring to the Table 1, it is observed that run 1 was not considered valid; this is because for one of the users in test dataset prediction 0 was not submitted. We believe this was caused by stopping the program while sending that prediction. However, we do have data for run 0 that use the same model as run 1 and in this task, our SVM model has not performed correctly despite having an accuracy of 0.56 on the internal validation dataset created dividing by users in training and validation datasets. We believe this may be due to the fact that, when training with very large texts, it does not generalize well with small texts for gambling and non-gambling for the first rounds and further the embedding creator, all-MiniLM-L6-v2, is not particular for gambling sentences that makes it even more difficult for the SVM to differentiate between both classes.

In the case of the run 3 the results are much better compared to the other valid run in terms of Macro F1 but not in terms of accuracy. The threshold has a lot of importance here, as we tested manually over some messages, but the messages generated are not general for all the concepts related to gambling as mentioned in section 3.2 because we just focused on generate messages related to popular methods of gambling. In an internal test dividing the dataset created by GPT into train and valid we obtained 0.97 in accuracy for messages. Following this approach and doing much better data generation the results could be even better when using it in the competition than the ones achieved with our run despite being in the 9th position in the table.

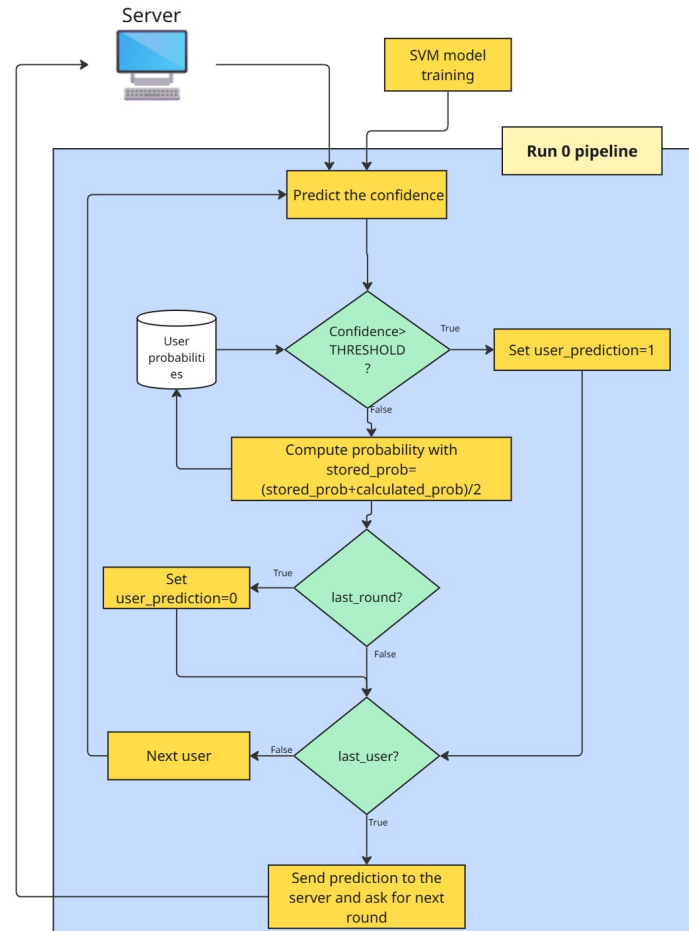
For configuration 2, an initial exploration of true positives, false positives, and false negatives was carried out. It was really important to establish the threshold to be used when predicting with users

<sup>4</sup>[https://scikit-learn.org/stable/modules/generated/sklearn.feature\\_extraction.text.TfidfVectorizer.html](https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.TfidfVectorizer.html)

<sup>5</sup><https://pythonspot.com/nltk-stop-words/>

<sup>6</sup><https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html>





**Figure 5:** High level architecture of the mechanism of communication for the run 1 submitted to MentalRiskES task 1 by HULAT-UC3M team

coming from the server. A preliminary test, as explained in the implementation and tools section, involved using 20 addiction users and another 20 non-addiction users. Table 2 shows the results in terms of TP, FN, and FP from the experiment.

In this case, a threshold set at 0.4 properly separates addicted users from non-addicted ones. Additional tests were also conducted at the message level to detect which messages were responsible for the TP, FP, and FN cases. In the case of relatively high confidence in gambling predictions, we have TPs such as:

- "Ya le metí mil" (I already put a thousand) with confidence of 0.4922
- "Metí verde" (I put green) with confidence of 0.4903
- "Y cayó rojo" ( And it came out red) with confidence of 0.5133
- "Esta es la de hoy" (This is the one for today) with a confidence of 0.4235

In other hand, we have some sentences not predicted as gambling, that they should be (false negatives). As mentioned before, because the generation of GPT does not contemplate similar messages to these ones.

- "Pero que sepas que falta una parte entera pa dos corners y es verde" (But know that a whole part is missing for two corners and it is green) with a confidence of 0.0200
- "No me parece tan descabellada por eso 1 eurillo" (I do not think it is so ridiculous for that 1 eurillo) with confidence of 0.0300
- "Está a buena cuota de que gana en prórroga" (It is at a good odds to win in overtime) with confidence of 0.0300

**Table 1**

Performance of teams participating in Task 1 at MentalRiskES 2025 ranked by macro F1 Score

Rank	Team	Run	Accuracy	Macro_P	Macro_R	Macro_F1
1	UNSL	2	0.569	0.568	0.567	0.567
2	UNSL	0	0.581	0.586	0.574	0.563
3	I2C-UHU-Rigel	1	0.556	0.555	0.553	0.551
4	ELiRF-UPV	1	0.550	0.564	0.556	0.540
5	ELiRF-UPV	2	0.538	0.543	0.542	0.534
6	ELiRF-UPV	0	0.538	0.535	0.535	0.533
7	PLN_PPM_ISB	2	0.538	0.535	0.534	0.532
8	UC3Mental	1	0.513	0.507	0.506	0.495
9	<b>HULAT_UC3M</b>	<b>2</b>	<b>0.488</b>	<b>0.488</b>	<b>0.488</b>	<b>0.488</b>
10	PLN_PPM_ISB	1	0.569	0.636	0.554	0.484
11	SoloResearch	2	0.506	0.497	0.498	0.475
12	NLP-UNED	0	0.531	0.533	0.519	0.468
13	SoloResearch	1	0.500	0.486	0.490	0.455
14	NLP-UNED	2	0.494	0.476	0.483	0.446
15	SoloResearch	0	0.494	0.476	0.483	0.446
16	UNSL	1	0.475	0.459	0.467	0.444
17	PLN_PPM_ISB	0	0.550	0.632	0.534	0.436
18	UC3Mental	2	0.550	0.632	0.534	0.436
19	NLP-UNED	1	0.506	0.487	0.493	0.429
20	Robertuito	–	0.550	0.657	0.533	0.428
21	PUXai	0	0.538	0.622	0.520	0.403
22	PUXai	2	0.525	0.539	0.508	0.396
23	UC3Mental	0	0.506	0.465	0.490	0.385
24	I2C-UHU-Rigel	2	0.481	0.490	0.498	0.371
25	PUXai	1	0.525	0.594	0.507	0.367
26	purple_john	0	0.519	0.509	0.500	0.353
27	VerbaNexAI Lab	0	0.519	0.259	0.500	0.342
28	VerbaNexAI Lab	1	0.519	0.259	0.500	0.342
29	VerbaNexAI Lab	2	0.519	0.259	0.500	0.342
30	I2C-UHU-Rigel	0	0.519	0.259	0.500	0.342
31	Roberta Base	–	0.519	0.259	0.500	0.342
32	wangkongqiang	2	0.519	0.259	0.500	0.342
33	wangkongqiang	1	0.519	0.259	0.500	0.342
34	wangkongqiang	0	0.519	0.259	0.500	0.342
35	purple_john	1	0.519	0.259	0.500	0.342
36	<b>HULAT_UC3M</b>	<b>0</b>	<b>0.513</b>	<b>0.258</b>	<b>0.494</b>	<b>0.339</b>
37	purple_john	2	0.513	0.258	0.494	0.339
38	<b>HULAT_UC3M</b>	<b>1</b>	<b>No valid run</b>	–	–	–

**Table 2**Comparison of results (TP, FP, FN) for thresholds 0.4 and 0.5 using the *RF* model

Threshold	True Positives (TP)	False Positives (FP)	False Negatives (FN)
0.4	16	5	4
0.5	13	2	7

Finally, we can notice some of the messages that were labeled as gambling from users that were not referring to gambling considered false positives (FP):

- "ALL IN!!!" with confidence of 0.5720
- "Increíble" (Incredible) with confidence of 0.5487
- "Go" with confidence of 0.4100



- "Que sabrás tu de rojo" (you have no clue about red!) with confidence of 0.5133
- "De que casa ?" (Which house?) with confidence of 0.4684

In this project, the responsible for MentalRiskES shared task propose to capture some metrics according to the consumption in terms of energy and CO2 produced with the library Code Carbon. The results are represented in the table 3. In one hand, the table represents the CO2 produced by the column `emissions_mean`, and in the other hand, it shows the KW needed with the column `energy_consumed_mean`. The CO2 produced was 4.78E-06 KG, that compared with the 2.37KG that produces 1L of diesel [15], it's not a big quantity. In comparison with the other teams, the emissions produced by our runs is one of the lowest in the whole table. For comparing the results of energy, in terms of KW, the team wanted to compare the consumption to Tesla car. Tesla model S car consumed 17,5KW per 100km [16] and our run takes 2.75E-05 KW, that means that if we take our energy consumption in terms of distance with the car, it would mean 16cm approximately. In energy consumed, the team is also one of the best in the table.

## 5. Conclusions and Future Work

We have presented a general overview of our participation in Task 1 at MentalRiskES [1] on early detection mechanisms for gambling users on social networks. As mentioned in previous sections, we can notice that the machine learning approaches followed in the runs were not adequate and did not work as planned, especially in using the whole text as input to model training. For possible solutions and future work about this, it would be a good idea to try to generate more generalized data using decoder-based LLM (such as Llama 3 or GPT models). It worked better than expected with not many data generated and it could be refined with more data and more testing in the selection of the threshold number.

On the other hand, instead of the machine learning model using a pre-trained model with an SVM classifier, it would be good to use one of those pre-trained models to adapt it to create embeddings related to gambling instead of general text. In addition, the use of transformers to create a self-attention matrix to detect dependencies between words would be a good approach to this task to capture semantic relationships. Then, this attention matrix could be used in a convolutional neural network (CNN) in order to detect more dependencies as it was already done in some projects with very good accuracy [14, 17, 13]. In addition to the CNN, we will need max-pooling to reduce dimensions and ReLU activation function to improve generalization as it removes negative values. It could be ended using the activation function softmax for classifying between gambling and non-gambling in the last layer.

## Acknowledgments

This work was partially supported by Grant PID2023-148577OB-C21 (Human-Centered AI: User-Driven Adapted Language Models-HUMAN AI) by MICIU/AEI/ 10.13039/501100011033 and by FEDER/UE.

## Declaration on Generative AI

The team has used generative AI, particularly ChatGPT, in spelling check for creating this document and code regarding latex format. In addition and as mentioned in the proposed solution section for run 2, GPT-4o mini model has been used in order to create data labeled per message for run 2. Finally, some minor errors in the code have been fixed using it.

**Table 3**

Energy consumption and emissions per run in Task 1 at MentalRiskES 2025

Rank	Team	Run	emissions_mean	energy_consumed_mean
1	UNSL	2	1.66E-07	6.99E-05
2	UNSL	0	1.66E-07	6.98E-05
3	I2C-UHU-Rigel	1	4.19E-03	7.37E-03
4	ELiRF-UPV	1	1.27E-04	7.32E-04
5	ELiRF-UPV	2	1.27E-04	7.32E-04
6	ELiRF-UPV	0	1.27E-04	7.32E-04
7	PLN_PPM_ISB	2	2.65E-03	4.12E-03
8	UC3Mental	1	9.07E-05	5.21E-04
9	HULAT_UC3M	2	4.78E-06	2.75E-05
10	PLN_PPM_ISB	1	2.65E-03	4.12E-03
11	SoloResearch	2	4.24E-05	2.44E-04
12	NLP-UNED	0	4.73E-08	2.72E-07
13	SoloResearch	1	4.24E-05	2.44E-04
14	NLP-UNED	2	3.62E-07	2.08E-06
15	SoloResearch	0	4.24E-05	2.44E-04
16	UNSL	1	1.66E-07	6.99E-05
17	PLN_PPM_ISB	0	2.65E-03	4.12E-03
18	UC3Mental	2	9.07E-05	5.21E-04
19	NLP-UNED	1	2.00E-07	1.15E-06
20	Robertuito			
21	PUXai	0	6.75E-05	1.47E-04
22	PUXai	2	6.75E-05	1.47E-04
23	UC3Mental	0	9.07E-05	5.21E-04
24	I2C-UHU-Rigel	2	4.19E-03	7.37E-03
25	PUXai	1	6.75E-05	1.47E-04
26	purple_john	0	3.40E-05	7.52E-05
27	VerbaNexAI Lab	0	4.71E-04	1.82E-03
28	VerbaNexAI Lab	1	4.71E-04	1.82E-03
29	VerbaNexAI Lab	2	4.71E-04	1.82E-03
30	I2C-UHU-Rigel	0	4.19E-03	7.37E-03
31	Roberta Base			
32	wangkongqiang	2	1.20E-05	3.43E-05
33	wangkongqiang	1	1.20E-05	3.43E-05
34	wangkongqiang	0	1.20E-05	3.43E-05
35	purple_john	1	3.40E-05	7.52E-05
36	HULAT_UC3M	0	4.78E-06	2.75E-05
37	purple_john	2	3.40E-05	7.52E-05
38	HULAT_UC3M	1		

## References

- [1] A. M. Mármol-Romero, P. Álvarez-Ojeda, A. Moreno-Muñoz, F. M. P. del Arco, M. D. Molina-González, M.-T. Martín-Valdivia, L. A. Ureña-López, A. Montejo-Ráez, Overview of mentalriskes at iberlef 2025: Early detection of mental disorder risk in spanish, *Procesamiento del Lenguaje Natural* 75 (2025).
- [2] J. Á. González-Barba, L. Chiruzzo, S. M. Jiménez-Zafra, Overview of IberLEF 2025: Natural Language Processing Challenges for Spanish and other Iberian Languages, in: *Proceedings of the Iberian Languages Evaluation Forum (IberLEF 2025)*, co-located with the 41st Conference of the Spanish Society for Natural Language Processing (SEPLN 2025), CEUR-WS. org, 2025.
- [3] M. E. Bellringer, S. Janicot, T. Ikeda, Changes in some health and lifestyle behaviours are significantly associated with changes in gambling behaviours: Findings from a longitudinal new zealand population study, *Addictive Behaviors* 149 (2024) 107886.

- [4] N. Butler, Z. Quigg, R. Bates, M. Sayle, H. Ewart, Gambling with your health: Associations between gambling problem severity and health risk behaviours, health and wellbeing, *Journal of Gambling Studies* 36 (2020) 527–538.
- [5] L. Macía, A. Estévez, P. Jáuregui, Gambling: Exploring the role of gambling motives, attachment and addictive behaviours among adolescents and young women, *Journal of gambling studies* 39 (2023) 183–201.
- [6] L. Tozzi, C. Akre, A. Fleury-Schubert, J.-C. Surís, Gambling among youths in switzerland and its association with other addictive behaviours: a population-based study., *Swiss medical weekly* 143 (2013) 1–6.
- [7] E. Smith, J. Peters, N. Reiter, Automatic detection of problem-gambling signs from online texts using large language models, *PLOS Digital Health* 3 (2024) e0000605.
- [8] K. Yokotani, T. Yamamoto, H. Takahashi, M. Takamura, N. Abe, Sounds like gambling: detection of gambling venue visitation from sounds in gamblers’ environments using a transformer, *Scientific reports* 15 (2025) 340.
- [9] W. S. Murch, S. Kairouz, M. French, Establishing the temporal stability of machine learning models that detect online gambling-related harms, *Computers in Human Behavior Reports* 14 (2024) 100427.
- [10] H. Christiansen, M.-L. Chavanon, O. Hirsch, M. H. Schmidt, C. Meyer, A. Müller, H.-J. Rumpf, I. Grigorev, A. Hoffmann, Use of machine learning to classify adult adhd and other conditions based on the conners’ adult adhd rating scales, *Scientific reports* 10 (2020) 18871.
- [11] P. Álvarez-Ojeda, M. V. Cantero-Romero, A. Semikozova, A. Montejo-Ráez, The precom-sm corpus: Gambling in spanish social media, in: *Proceedings of the 31st International Conference on Computational Linguistics*, 2025, pp. 17–28.
- [12] S. Ghosh, O. Vinyals, B. Strope, S. Roy, T. Dean, L. Heck, Contextual lstm (clstm) models for large scale nlp tasks, *arXiv preprint arXiv:1602.06291* (2016).
- [13] Z. Zhang, J. Zhu, Z. Guo, Y. Zhang, Z. Li, B. Hu, Natural language processing for depression prediction on sina weibo: Method study and analysis, *JMIR Mental Health* 11 (2024) e58259.
- [14] M. M. Lopez, J. Kalita, Deep learning applied to nlp, *arXiv preprint arXiv:1703.03091* (2017).
- [15] Michelin, Cómo calcular emisiones de co2, Sitio web, 2025. URL: <https://connectedfleet.michelin.com/es/blog/calcular-emisiones-de-co2/>, access: 31 may 2025.
- [16] Tesla, Consumo de energía del vehículo, Sitio web, 2025. URL: [https://www.tesla.com/es\\_es/support/power-consumption](https://www.tesla.com/es_es/support/power-consumption), access: 16 june 2025.
- [17] W. Yin, K. Kann, M. Yu, H. Schütze, Comparative study of cnn and rnn for natural language processing, *arXiv preprint arXiv:1702.01923* (2017).