Forecast the diagnostic group (PTSD or CUD) from the intrusion characteristics

Team ROC-stars: Clara Chappuis, Camille Pittet, Renuka Singh Virk
Machine Learning Project II (ML4Science) - EPFL

***Abstract*—This paper aims to assess intrusive memory characteristics to predict whether an intrusive memory is due to Post-Traumatic Stress Disorder (PTSD) or Cocaine Use Disorder (CUD) based on the symptoms. We provide insights into the dataset, its analysis, and the implementation of machine learning methods to make predictions. We explore several classification methods and find the best results with MLP, Logistic Regression and Gradient Boosting. As the results within these three models do not differ much, we decide that the best model to use is Logistic Regression with a Ridge regularization, which renders an accuracy of 86.8% and an F1 score of 81.6%.**

I. Introduction

Intrusions are unwanted, uncontrollable intrusive memories or thoughts. Often, intrusive memories are not verbal thoughts but manifest as individual images, feelings, sounds, odors, tastes, or physical sensations.

The EMemory study was conducted using ecological momentary assessment (EMA). For the two groups, Post-Traumatic Stress Disorder (PTSD) and Cocaine Use Disorder (CUD) patients, two types of surveys are used. We focus on the *event–based approach*. In this survey the patients are asked to complete daily diaries on the frequency and characteristics of their trauma– or drug–related intrusions on 14 consecutive days using a smartphone app. The questions cover the content, duration and modality of the intrusion, the intrusion characteristics (intrusiveness, vividness), the emotional responses (anger, fear, guilt, shame, helplessness, etc.), as well as the cognitive behavioral responses (dwelling, suppression, distraction medication or drug intake, etc.) and potential triggers for the intrusion.

The subjects are required to complete the event-based questionnaire after the occurrence of an intrusion event.

The responses of participants, apart from the text answers, are coded numerically. The survey differentiates between slider bar questions and single/multiple-choice questions. Depending on the answer given by the participant, follow–up questions might have been asked. These follow-up questions are either tick boxes or in the form of text boxes. Intrusiveness, vividness, anxiety, emotional state, loss of control, distress, craving, and valence are interrogated using slider bar questions with values going from 0 to 10. Single or multiple choice questions are used to capture information on duration, modality and trigger type of an intrusion, as well as emotional, physical, and behavioral responses. Free text questions are used to ask the participants what went through their heads during the intrusion event.

This project focuses on using machine learning to predict whether intrusive memories are due to PTSD or CUD, based on the symptoms. Ultimately, the laboratory is interested in determining whether there are clear patterns in the characteristics and symptoms surrounding intrusive memories in both groups. A successful classification of both classes indicates that there are indeed significant differences in the intrusive memories nature.

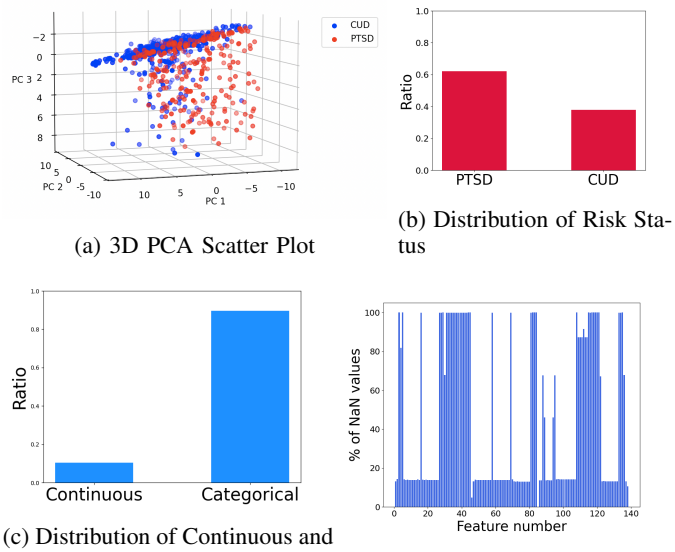
Our goal is to find the most effective machine learning methods to correctly map the entries to either PTSD or CUD.

II. Data Visualization and Exploration

A. Dataset

The data is composed of one `.xlsx` file which we convert into a `.csv` file. The data frame is of shape (1002, 654) and thus entails 1001 survey answers as well as a header row, and 654 features comprising one categorical column which we are interested in predicting, namely 'Intrusionsfragebogen (T)' for PTSD, or 'Intrusionsfragebogen (K)' for CUD patients. Note that there are multiple rows per patient.

B. Data visualization



(a) 3D PCA Scatter Plot

(b) Distribution of Risk Status

(c) Distribution of Continuous and Categorical Values

(d) NaN Values Distribution

Fig. 1: Visualization of Data

The analysis of the features and risk status using Figure 1 focuses on four aspects:

Principal component	PC1	PC2	PC3	PC4	PC5	PC6	PC7	PC8	PC9	PC10
% of variance explained	34	15	10	6	4.5	4.2	4	1.9	1.7	1.4

TABLE I: % of variance explained by first 10 principal components (PCs).

a) *PCA Scatter Plot*:
We plot the high-dimensional data in three dimensions by applying Principal Component Analysis (PCA) to the dataset. We choose a 3D representation because the three first principal components PC1, PC2 and PC3 explain respectively 34%, 15%, and 10% of the variance, and the next principal components do not explain a lot of the variance as is illustrated in Table I.
We observe on Figure 1a that the two classes are mostly overlapping, meaning that they are not linearly separable in the reduced feature space defined by PC1, PC2 and PC3. Because PCA is typically more effective for continuous data, the prevalence of categorical features in our dataset might be a contributing factor to the lack of success.

b) *Slightly Unbalanced Data*: The training set consists of 62% of PTSD and 38% of CUD patients, showing a moderate bias.

c) *Different Distributions*: The patients' dataset consists of two types of variables: continuous (10%) and categorical (90%). The categorical features are binary and can either take the value of 0 or 1.

d) *Missing Values*: There are several missing values in the patients' dataset, which must be handled during data cleaning. These missing values either take the form of `<no-response>` or `<not-shown>`.

e) *Irrelevant Values*: As mentioned in section I, the slider bar questions can score from 0 to 10. However, we observe a few values larger than 10. These are due to a shifting mistake in the .xlsx file during download. Indeed, for some participants, time-related objects that can be larger than 10 are erroneously placed in a column where the values should range from 0 to 10. These values are handled during data cleaning.
There are also free text answers, in which the patients describe what they feel during an intrusive memory. Such answers are not interpretable and cannot be used for our models.

C. Data Cleaning

1) Merging CUD and PTSD features: Some features are present twice, with information filled for PTSD participants in one feature and information corresponding to CUD participants in the other feature. We merge the corresponding columns by removing 'CUD' or 'PTSD' from the headers and merging columns with identical headers. We also remove the features corresponding to questions that were asked only to the CUD patients, which we can not use for our prediction.

2) Handling Missing Values: We replace all `<no-reponse>` and `<not-shown>` values by `NaN` and then remove all columns containing over 15% of `NaN` values. We choose 15% because the `NaN` percentage across features is either below 15 or above 50, there are no in-between (see Figure 1d). We also remove all rows with more than 80% of `NaN`, which correspond to patients who stopped responding to the questionnaire, therefore not providing useful information.

3) Handling Irrelevant Values : We remove all features with '_RT' in their header as they contain additional information on how long each participant spent on the question, which is not relevant to our analysis.
We replace all categorical values that are out of range (below 0 or above 10) with `NaN` values.
We also replace all non-numeric values with `NaN`.

4) Removing Outliers: Some features might present outliers that can be removed using the 1.5 IQR (inter-quartile range) rule. However extreme values must not necessarily be interpreted as outliers, especially considering that the answers are in a precise range defined by the laboratory. In the file `outliers.py`, we remove outliers from continuous features of both the training and testing sets. This is done via the function `remove_outliers` in the file `helper.py`. The logistic regression model is then tuned on both the training set with and without outliers and predictions are generated.

As indicated in Table II, removing outliers significantly decreases the performance of the model, which is why we do not remove outliers.

Scenario	Accuracy	F1 Score
With Outliers	0.868	0.816
Without Outliers	0.787	0.704
With Feature Augmentation	0.862	0.809
Without Feature Augmentation	0.868	0.816

TABLE II: Assessing the Impact of Ablation on Logistic Regression Performance

5) Low Standard Deviation Features: Some features consist of constants or very similar values thus not providing helpful information for our predictions. We remove all columns with a low standard deviation. In our case, the three smallest standard deviations are 0.0 for the 'INTRUSION_ERLEBT' feature, 0.188333 for the 'STRATEGIE_4' feature, and 0.214472 for the 'TRIGGER_9' feature. We thus set the standard deviation threshold to 0.1, removing the feature with a standard deviation of 0.

The final cleaned dataset consists of 870 rows and 86 columns (85 features, plus the target).

III. Models and Methods

A. Target Variable Isolation

Because our target variables are a feature of the whole dataset, we isolate them and store them in a `y` data frame. The remaining set `X` contains the input features. The training and testing sets are generated using `train_test_split` from `sklearn.model_selection`. We do not create a validation set as we later use cross-validation for tuning and thus do not require separate training and validation sets.

B. Preprocessing

1) Variable Transformation for Binary Classification: The feature containing our target variables has the header 'SURVEY_NAME', which can either take on the value 'Intrusionsfragebogen (T)' for PTSD participants or 'Intrusionsfragebogen (K)' for CUD participants. We transform these values into 0 for PTSD and 1 for CUD.

2) Standardization: We standardize continuous features so they have a mean of 0 and a standard deviation of 1. This step ensures that the features share a common scale, making it easier to compare them across different models and enhancing the convergence of the model. Categorical features are left unchanged since standardization is applied to numerical features.

Note that models that rely on distance to assess classification are most affected by the scaling of the features. [1]

3) Feature Augmentation: Because our cleaned dataset is relatively small, we implement polynomial feature augmentation. We do so by using the `PolynomialFeatures()` method from `sklearn.preprocessing`. We create a new data frame consisting of all the interactions possible (up to degree 2) between the features. We then test whether we obtain better results with Logistic Regression, but the accuracy does not improve (see Table II). On top of that, the running time increases significantly with a much larger dataset, we thus choose not to proceed with this approach.

We introduce a constant term (intercept). The model can thus shift its decision boundary, which is not forced to pass through the origin anymore. This can potentially provide a more balanced classification.

4) Hyperparameters Tuning: The best hyperparameters for a model are determined using K-fold cross-validation with 5 folds on the training set. We choose 5-fold cross-validation so that at each iteration, 80% of the training data is used for training and 20% is used for validation, providing a good balance. To find the best parameters, the model performance is evaluated under various hyperparameters, and the ones rendering the highest accuracy score are selected.

C. Comparison of models

We implement various methods:

- Logistic Regression (LR)
- Random Forest Classifier (RF)
- Gradient Boosting (GB)
- Support Vector Machine (SVM)
- K-Nearest Neighbours (KNN)
- Multilayer Perceptron (MLP)
- Gaussian Mixture Model (GMM)

Each model training is carried out on a training set consisting of 80% of the data, and the remaining 20% is used as a test set to assess the accuracy of the models. In order for the code to be reproducible, we split the dataset consistently by setting the `random_state` of the function `train_test_split` to 0. Since our dataset is quite small, the accuracies of our models are significantly influenced by how the dataset is split. However, when we split the data differently, we never got accuracies and F1 scores below 0.7 or above 0.9, so the performance remains good overall.

We evaluate model performance by examining both accuracy and F1 score. Since our data is not strongly unbalanced, these two scores are similar to each other within each model. They also evolve similarly, i.e. if the F1 score of a model increases, its accuracy is larger or equal.

As a reminder, the F1 score takes into account the precision and the recall:

$$F1 = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{1}$$

The accuracy compares the predictions to the actual set and returns the % of exactitude between both sets.

The results of each method are summarized in Table III.

	Accuracy	F1 Score
MLP manual	0.868	0.819
LR	0.868	0.816
GB	0.868	0.816
RF	0.862	0.803
MLP random search	0.856	0.803
PSVM	0.851	0.794
MLP grid search	0.845	0.777
SVM	0.839	0.774
KNN	0.810	0.703
GMM	0.799	0.685

TABLE III: Performance of the models; best (top) to worst (bottom).

IV. Results

Out of the eight models we tune, the best are Multilayer Perceptron (MLP), Logistic Regression (LR), and Gradient Boosting (GB) (see Table III). We thus choose to investigate these three methods further.

1) Multilayer Perceptron: The tuning of Multilayer Perceptron is very time-consuming since it is a fully connected neural network. Such methods typically work best for big datasets, which we do not have in our case. Additionally, the risk of overfitting is non-negligible with MLP. While the model yields great results, the investment of 12 hours may not be justified, particularly when we can achieve outcomes almost as good using less complex methods.

Tuning using grid search cross-validation can miss optimal combinations of hyperparameters, depending on how the grid space is defined, which is why the grid search tuning does not

render the best hyperparameters for MLP (see Table III).
We thus implement random search tuning and get slightly better results, but again, not the best MLP can render.
Indeed with manual testing of combinations of hyperparameters, we are able to obtain better results than both of the tuning methods cited above. Of course, the results found are not the ultimate best ones, only the best we were able to find manually. We find these by doing 5-fold cross-validation on our training set and looking at the mean accuracy. Note that it is very important not to choose the hyperparameters that maximize the accuracy on the test set directly, in order to avoid overfitting.

2) Logistic Regression: The main advantage of Logistic Regression is that it is fast to tune. The `LogisticRegression()` class from the `scikit-learn` library uses a default threshold of 0.5. For unbalanced data, it might be necessary to change this threshold, which is why we test various thresholds and assess performance for each threshold. The threshold rendering both the best F1 score and accuracy is 0.5. This is expected since our dataset is not particularly unbalanced. The ROC curve is plotted in Figure 2. The AUC (0.95) indicates that our model is quite performant.

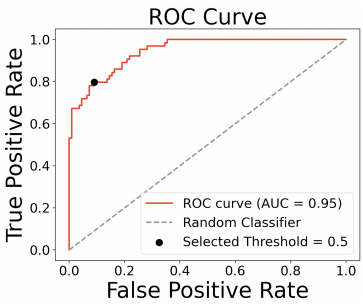
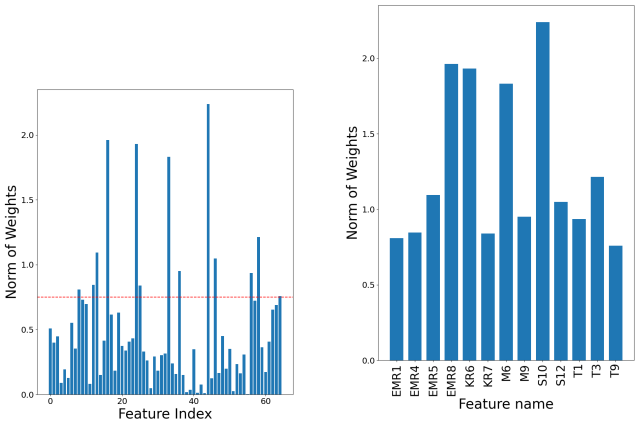


Fig. 2: Receiver Operating Characteristic (ROC) curve of Logistic Regression model

3) Gradient Boosting: With Gradient Boosting we can obtain the same performance level as with Logistic Regression. However, because the second model is less complex and faster to tune, we will rather choose Logistic Regression over Gradient Boosting.

V. Model Analysis

To further study and understand the Logistic Regression model, we analyze the weights to assess which features are the most important for the prediction. This can be useful to assess what the main differences between the CUD and PTSD patients are. In order to find the most significant features we plot the weights (Fig 3a) of each feature and retrieve the ones with weights larger than a threshold of 0.75. The threshold is selected by graphically identifying the limit between the average weights and the outlying ones. Gaining insights into the features that significantly impact the prediction of the Logistic Regression model proves valuable in discerning the differences between the nature of intrusive memories in both categories. This can be investigated further in future studies.



(a) Weight of each feature

(b) Features with highest weight norm[1]

Fig. 3: Features' weights of Logistic Regression model with ridge regularization and $C = 1.0$

VI. Discussion

It is important to keep in mind that the answers provided by the participants are very subjective. Indeed, individuals come from diverse backgrounds and each of them has a different understanding of the scale for expressing their experiences. For instance, an anger level rated at 5 by a participant might translate to an 8 by another, depending on their separate perceptions. Therefore, the subjective nature of responses introduces challenges to create a universally applicable machine learning model. A further step to improve this project could be to construct a model able to adapt to this variability. This could be done by collecting more data, using more advanced methods, considering additional information or finding a way to normalize subjective answers.

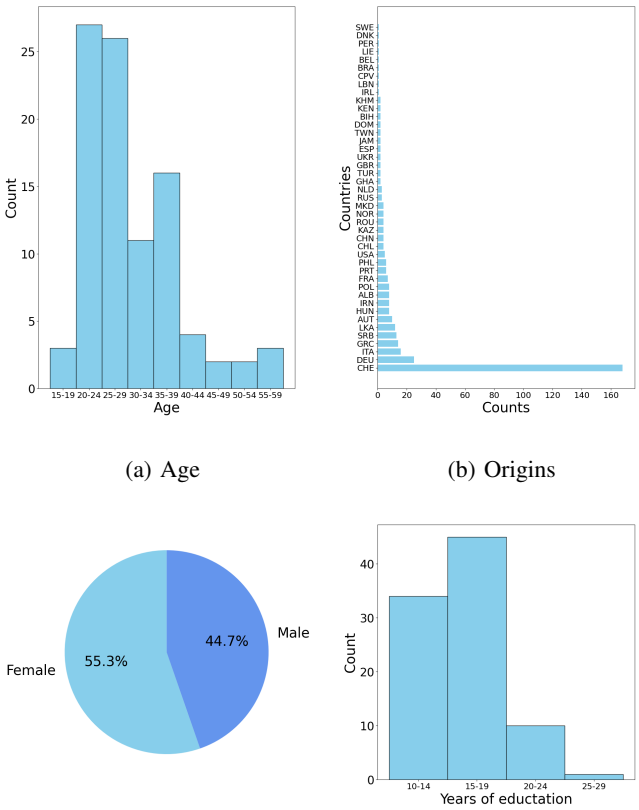
VII. Summary

In summary, we start by visualizing the data and observe that we do not have a strongly unbalanced dataset, but that many features contain a lot of `NaN` values. These are handled during data cleaning. We also see that our features are mainly categorical. We then implement various classification methods and conclude that, due to its simplicity and to the accurate results it renders, Logistic Regression (with Ridge regularization) is the best model to use. We also plot the ROC curve and observe that we have an AUC of 0.95, which again indicates a good accuracy of the model.

The fact that we are able to correctly classify the two groups based on the symptoms indicates that there are different patterns in the nature of the intrusive memories between the two groups. In order to assess which features are the most significant to distinguish PTSD from CUD patients, we analyze the weights of the model.

[1]'EMR' = 'EM_REAKTION_', 'KR' = 'KOERP_REAKTION_', 'M' = 'MODALITAET_', 'S' = 'STRATEGIE_', 'T' = 'TRIGGER_'

VIII. Ethical Risk Assessment



(a) Age (b) Origins

(c) Distribution of Gender (d) Years of education

Fig. 4: Patients characteristics

We see on Figure 4 the distribution of age, origins, gender and years of education amongst the population of the study. There is a nice balance between the genders with approximately half of the patients being male or female. However, some populations are underrepresented.
Namely, there are very few patients over the age of 40. The origins of the patients are very similar with mostly Swiss participants. The years of education between the participants are similar and around 15. However this parameter might not be the most relevant one.
The lack of representation of older population as well as the lack of diversity in the origins of the participants might translate to a poor generalization of the model to unseen populations. This could lead to poor predictions for underrepresented patients (e.g. over the age of 40 or non-Swiss citizens), and one must thus keep in mind that the conclusions drawn from this project apply only to a certain population.
To take this problem into account we could have had either make the data more representative of all the population by balancing the different categories. For example adding more data about patients over 40 years old or by removing some data of the over represented population. In conclusion we would have to gather more diverse participants.

In the context of this clinical study, parameters such as age and origin (difference in genetic background) might play a role in the occurrence of intrusive memories. The risk is to get predictions that only work for specific subsets of patients, thus missing important characteristics of what distinguishes the intrusive memories in PTSD and CUD patients.

IX. Acknowledgements

We wish to thank Lina Dietiker and Amelie Zacher from the laboratory of *Experimentelle Psychopathologie und Psychotherapie* at the University of Zurich for supervising us.

References

1. *Feature engineering, Scaling and Standardization* Accessed on Date 13.12.2023. https://www.analyticsvidhya.com/blog/2020/04/feature-scaling-machine-learning-normalization-standardization/s.
2. Author, J. B. *Discover Feature Engineering, How to Engineer Features and How to Get Good at It* Accessed on Date 14.10.2023. https://machinelearningmastery.com/discover-feature-engineering-how-to-engineer-features-and-how-to-get-good-at-it/.
3. Author, A. N. *Advice for applying Machine Learning, Stanford* Accessed on Date 17.10.2023. https://cs229.stanford.edu/materials/ML-advice.pdf.