SAFE-LAND

Mitigating the risk of flooding and landslides via artificial intelligence with a view to extreme climate events

Deliverable 4.2:
Trustworthy AI models: risk awareness

Mitigating the risk of flooding and landslides via artificial intelligence with a view to extreme climate events

Co-funded by the
European Union

e-CAMPUS UNIVERSITÀ
UNIVERSITÀ DI PISA
MEĐIMURSKA ŽUPANIJA

Deliverable Title	Trustworthy AI models: risk awareness
Deliverable number	4.2
Deliverable Lead	UNIPI
Related Work Package	WP4: Trustworthy AI for hydrogeological risk assessment and evaluation of people's risk awareness of existing areas
Author(s)	Francesco Pistolesi (UNIPI), Michele Baldassini (UNIPI), Matteo Mugnai (UNIPI), Marco Avvenuti (UNIPI)
Dissemination Level	Public
Due Submission Date	11/11/25
Actual Submission	12/01/25
Project Number	101140345
Status	Version 1.0 (12/01/25)
Reviewed (Authors)	Elisabetta Cattoni (eCampus), Francesco Focacci (eCampus)
Start Date of Project	12 February 2024
Duration	24 months
Abstract	This document is the Deliverable 4.2 of the project *"Mitigating the risk of flooding and landslides via artificial intelligence with a view to extreme climate events (SAFE-LAND)"*. This deliverable describes the models implemented for risk awareness, their architectures and learning processes.
Status changes history	Version 1.0 (12/01/25) — First release

Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union. Neither the European Union nor the granting authority can be held responsible for them.

Table of Contents

Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union. Neither the European Union nor the granting authority can be held responsible for them.

1. Introduction

The SAFE-LAND project addresses the emerging risks of landslides and floods driven by extreme climate events. It aims to build a decision support framework that integrates technical hazard assessments with social analysis to enhance community resilience and public preparedness.

While accurate physical modeling is essential for hazard prediction, the effectiveness of mitigation strategies relies heavily on the population's preparedness and perception of risk. Understanding how individuals perceive hazards is crucial for designing effective communication strategies and emergency plans. SAFE-LAND addresses this challenge by developing AI-based classification models capable of predicting key psychological and cognitive dimensions related to risk. These models aim to quantify individuals' *Experience*, *Knowledge*, *Awareness*, and *Worry* regarding floods and landslides, allowing for a better assessment of social vulnerability in at-risk areas.

WP4 focuses on the design, training, and validation of supervised learning models to predict these risk perception indicators. These classifiers were trained on a dataset derived from a comprehensive socio-demographic survey. The dataset gathers information from respondents regarding their personal characteristics, past experiences with natural disasters, and emotional responses to potential hazards.

To achieve robust predictive performance, various machine learning architectures were implemented and compared, including tree-based ensembles (Random Forest, Gradient Boosting, XGBoost), Support Vector Machines (SVM), and Multilayer Perceptrons (MLP). Furthermore, to ensure the trustworthiness of the AI predictions, *Explainable AI (XAI)* techniques based on SHAP (SHapley Additive exPlanations) were employed. This allows for the identification of the most influential features driving risk perception, ensuring that the models provide interpretable insights consistent with domain knowledge.

This deliverable describes the results obtained during the development and testing of these risk awareness models. It is organized as follows: Section 2 provides a theoretical background of the supervised learning algorithms and explainability methods adopted; Section 3 details the experimental pipeline, including dataset preprocessing, feature selection, model training, and a comprehensive evaluation of performance and interpretability for both flood and landslide tasks; Section 4 draws the conclusions.

Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union. Neither the European Union nor the granting authority can be held responsible for them.

2. Background

Supervised learning is perhaps the most widely used type of machine learning, where the main objective is to learn a mapping from a set of input variables to one or more output variables. The input variables consist of measurable quantities that describe a particular problem, while the output variables, known as targets, represent the quantities that we aim to predict. The central assumption of supervised learning is that a training dataset is available, containing examples of inputs paired with their corresponding outputs. Using this dataset, the learning algorithm attempts to infer a function that can accurately predict the output for new, unseen inputs. Although it is generally easy and inexpensive to measure or collect feature vectors, obtaining the corresponding target values is often difficult, costly, or time-consuming.

Supervised learning can be applied to regression problems, where the target variable is continuous, or to classification problems, where the target is categorical. A wide range of models can be used in supervised learning, from simple linear models to complex non-linear methods. Decision trees are among the most intuitive models: they recursively partition the feature space into regions associated with specific outputs. However, single decision trees are prone to overfitting and high variance, which can limit their predictive performance. To overcome these limitations, ensemble methods combine multiple models to produce a stronger predictor. Bagging-based ensembles, such as Random Forests, reduce variance by averaging many decorrelated trees, while boosting-based ensembles, such as Gradient Boosting and its modern variants, build models sequentially to correct previous errors. These approaches leverage the strengths of decision trees while mitigating their weaknesses.

Beyond tree-based methods, supervised learning also includes neural and margin-based models. Support Vector Machines (SVMs) aim to find an optimal boundary that separates classes, with the flexibility of kernel methods to handle nonlinear patterns. Perceptrons and their generalization into Multilayer Perceptrons (MLPs) form the basis of neural networks, capable of approximating highly complex, nonlinear relationships through layers of interconnected neurons.

In the following sections, we provide an overview of the supervised learning models utilized in this work, starting with tree-based ensembles, moving to gradient boosting and its modern variants, and concluding with SVM and MLP.

2.1 Random Forest

Random Forest [1] is an ensemble learning method that builds a collection of decision trees and combines their predictions to produce a more accurate and stable model. It can be applied to both classification and regression problems. The main idea behind Random Forest is to reduce the overfitting and high variance that often affect single decision trees, while keeping their ability to capture non-linear relationships and complex feature interactions.

Each tree in the forest is trained on a bootstrap sample of the training data — a randomly selected subset obtained by sampling with replacement. This process, known as bagging (bootstrap aggregating), ensures that each tree is trained on a slightly different dataset. However, Random Forest introduces an additional source of randomness that

Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union. Neither the European Union nor the granting authority can be held responsible for them.

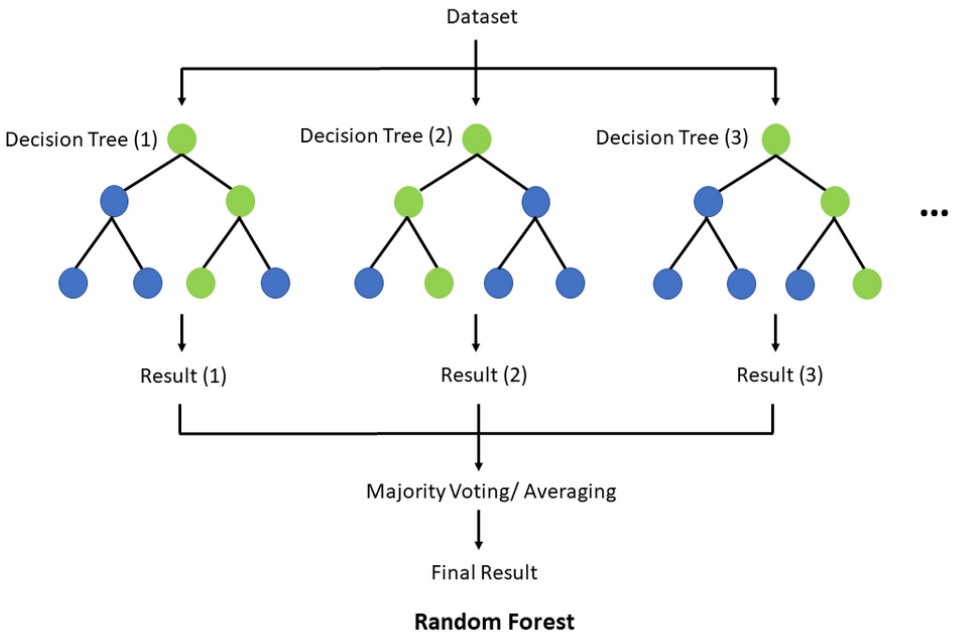


Fig. 1: Schematic of the Random Forest algorithm, illustrating how an ensemble of decision trees generates individual results that are aggregated (via majority voting or averaging) to produce a final prediction.

distinguishes it from standard bagged trees: at each split in the tree, instead of evaluating all available features, the algorithm randomly selects a subset of features (usually $\sqrt{p}$ for classification or $\log_2 p$ for regression, where p is the total number of features). The best split is then chosen only among this subset, based on the criterion that maximizes the decrease in impurity (e.g., Gini index or variance reduction).

This random selection of features decorrelates the trees, meaning that their errors are less likely to overlap. When predictions from many weakly correlated models are averaged (in regression) or combined by majority vote (in classification), the overall model variance is significantly reduced. This mechanism makes Random Forest both robust and reliable, even when the data are noisy or contain irrelevant variables. The idea of Random-Forest is showed in Fig. 1.

Random Forest combines the ideas of bagging and random feature selection to produce a strong ensemble predictor. Its main advantages include high predictive performance, the ability to handle high-dimensional data, and built-in estimates of feature importance, which make it a widely used and interpretable model in many domains.

2.2 Gradient Boosting

Gradient Boosting [2], [3] is an ensemble learning technique that extends the concept of boosting by interpreting it as a gradient descent process in function space. It builds a strong predictive model by sequentially adding weak learners (typically shallow decision trees) in a way that each new model focuses on correcting the residual errors of the combined models trained so far. Unlike Random Forest, where trees are built independently, Gradient Boosting trains trees in a stage-wise manner, with each iteration aimed at reducing the

Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union. Neither the European Union nor the granting authority can be held responsible for them.

overall prediction error. The method starts with an initial model $F_0(x)$, which is often a simple constant prediction such as the mean of the target variable. At each iteration m, the algorithm computes the negative gradient of the loss function with respect to current predictions. These gradients, known as pseudo-residuals, represent the direction in which the model should move to reduce the loss most effectively. A new weak learner $h_m(x)$ is then fitted to these pseudo-residuals, effectively learning how to correct the remaining errors of the ensemble.

Once $h_m(x)$ is trained, the algorithm finds an optimal multiplier that minimizes the loss. The model is then updated additively as

$$F_m(x) = F_{m-1} + \nu h_m(x) \tag{1}$$

where ν is the learning rate, typically a small value (e.g., 0.1 or less) to ensure gradual improvement and to avoid overfitting. The process repeats for a fixed number of iterations or until no further reduction in the loss is observed.

One of the key advantages of Gradient Boosting is its flexibility: it can optimize any differentiable loss function, such as the squared error for regression or the logistic loss for classification. This makes it a general and powerful framework capable of adapting to many types of predictive problems. However, since the trees are built sequentially, the algorithm can be sensitive to noise and prone to overfitting if the number of iterations is too large or if the trees are too deep.

2.3 Histogram Gradient Boosting

Histogram-based Gradient Boosting [4] is an efficient variant of the traditional Gradient Boosting algorithm. Like standard Gradient Boosting, it builds a strong predictive model by sequentially adding weak learners that focus on correcting the errors of the previous models. The key difference lies in how the algorithm handles the input data and splits during tree construction, which results in significant improvements in speed and memory usage. In HistGradientBoosting, continuous input features are first binned into discrete histograms. Instead of considering all possible split points for a feature, the algorithm evaluates splits only at the edges of these bins. This approximation drastically reduces the number of computations required to find the best splits, making the method particularly efficient for large datasets. Despite this simplification, the predictive accuracy remains comparable to traditional Gradient Boosting.

The training process follows the same stage-wise additive principle as standard Gradient Boosting. At each iteration, the algorithm computes the negative gradient of the loss function with respect to current predictions (pseudo-residuals), fits a weak learner to these residuals, and updates the model with a weighted contribution from the new tree. A learning rate controls how much each new tree influences the overall model, helping to prevent overfitting.

HistGradientBoosting inherits the flexibility of Gradient Boosting: it can optimize any differentiable loss function, such as squared error for regression or logistic loss for classification. Additionally, its histogram-based approach allows it to handle large datasets more efficiently, both in terms of computation time and memory consumption. It is also less sensitive to the exact choice of split points, which can improve robustness in practice.

Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union. Neither the European Union nor the granting authority can be held responsible for them.

2.4 Extreme Gradient Boosting

Extreme Gradient Boosting (XGBoost) [5] is an optimized and highly efficient implementation of gradient boosting that has become one of the most widely used algorithms in machine learning. Introduced in 2014 by Tianqi Chen and Tong He, it quickly became one of the most popular machine learning algorithms due to its combination of high predictive accuracy, scalability, and computational efficiency. It has been widely adopted in both academic research and industry applications, particularly where performance and speed are critical.

Like standard Gradient Boosting, XGBoost builds an ensemble of decision trees sequentially, where each new tree attempts to correct the errors made by the previous ones. However, XGBoost introduces several important improvements that make it faster and more robust:

- The first key innovation is the use of a regularized objective function that explicitly penalizes the complexity of individual trees. This regularization helps prevent overfitting by discouraging overly deep or irregular trees and by shrinking leaf weights toward smaller values.

- The second major improvement lies in how XGBoost optimizes the loss function. Instead of relying only on the gradient, XGBoost uses a second-order Taylor expansion of the loss function to approximate changes in the objective. This allows the algorithm to incorporate both first-order (gradient) and second-order (curvature) information, effectively performing a Newton–Raphson–style optimization in function space. As a result, XGBoost can converge faster and make more accurate updates compared to standard gradient boosting methods.

At each boosting iteration, XGBoost fits a new regression tree that minimizes the regularized objective function. Each leaf of the tree receives a score based on the weighted sum of gradients and Hessians of the loss with respect to the current predictions. These scores are then adjusted according to the regularization parameters, which control the trade-off between model complexity and generalization. Splits in the tree are determined by maximizing the gain, a measure derived from how much a potential split reduces the objective function.

Beyond its algorithmic improvements, XGBoost also benefits from several engineering optimizations, including parallelized tree construction, efficient memory usage, and native handling of sparse or missing data. These make XGBoost a highly scalable and reliable algorithm, often serving as a benchmark for structured and tabular data tasks.

2.5 Support Vector Machine

Support Vector Machines (SVMs) [6] are among the most classic and conceptually simple machine learning models, yet they remain highly effective for many classification tasks. SVMs belong to the family of margin-based models, which focus on finding the best boundary between classes rather than modeling the distribution of each class directly.

The central idea, as showed in Fig. **??**, is to identify a hyperplane that separates data points with the maximum possible margin — the distance between the boundary and the

Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union. Neither the European Union nor the granting authority can be held responsible for them.

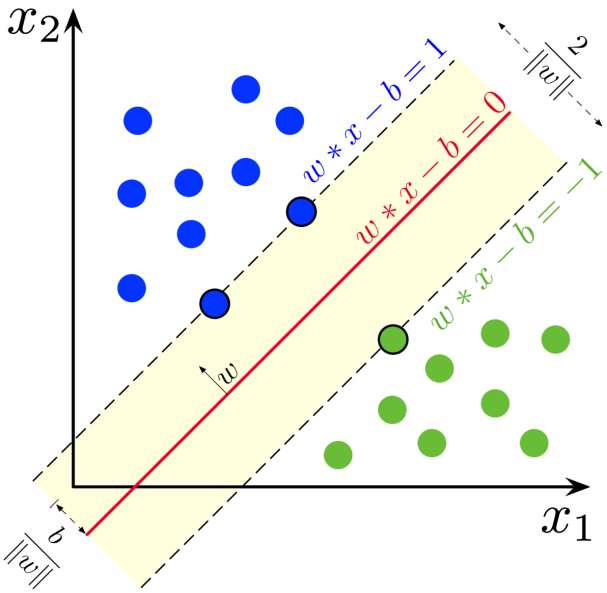


Fig. 2: Illustration of a linear Support Vector Machine classifier. The model identifies an optimal hyperplane (red line) that maximizes the margin (yellow area) between the two classes (blue and green circles).

closest samples from each class, known as support vectors. This emphasis on maximizing the margin helps improve generalization and robustness to noise, even when the number of features is large relative to the dataset size.

However, many real-world problems are not linearly separable. To handle such cases, SVMs can be kernelized. Kernelized SVMs use a mathematical technique called the kernel trick to implicitly map data into a higher-dimensional space, where a linear separation becomes possible. This allows SVMs to model complex, nonlinear decision boundaries without explicitly computing the high-dimensional representation, which would be computationally expensive.

Two commonly used kernels are:

- Polynomial kernel – computes all polynomial combinations of the original features up to a specified degree.
- Radial Basis Function (RBF) or Gaussian kernel – maps the data into an effectively infinite-dimensional space, considering all polynomial combinations with decreasing importance for higher degrees.

During training, the SVM identifies which data points are essential for defining the decision boundary—these are the support vectors. Only these points influence the final model. When predicting a new point, the SVM measures the similarity (or distance) of this point to the support vectors, weighted by their learned importance, to assign a class label.

SVMs combine the geometric intuition of linear classifiers with the flexibility of nonlinear mappings, making them powerful and versatile tools for classification.

Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union. Neither the European Union nor the granting authority can be held responsible for them.

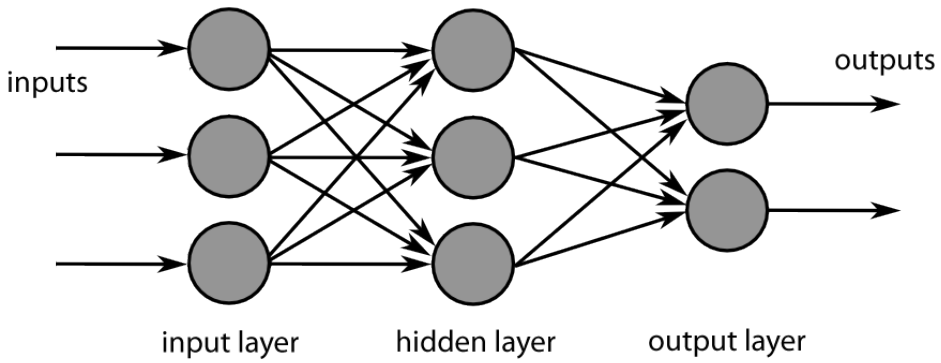


Fig. 3: Architecture of a simple feedforward neural network (Multi-Layer Perceptron) with one hidden layer, showing the connections between input, hidden, and output neurons.

2.6 Multilayer Perceptron

A Multilayer Perceptron (MLP) [7] is a type of supervised learning model that belongs to the family of artificial neural networks. MLPs can be used for classification or regression tasks, but they are particularly well-suited for capturing complex, nonlinear relationships in the data. MLPs are parametric models, meaning they learn a set of weights during training that define how inputs are transformed into outputs.

At its core, an MLP consists of layers of interconnected nodes, or "neurons." Each neuron computes a weighted sum of its inputs, applies a nonlinear activation function, and passes the result to the next layer. Fig. 3 shows an example of a neural network. The network typically includes:

- Input layer – receives the features of the data.
- One or more hidden layers – perform nonlinear transformations that allow the network to learn complex patterns.
- Output layer – produces the final prediction, either as class probabilities for classification or continuous values for regression.

The power of an MLP comes from these hidden layers and non-linear activations, which enable the network to approximate almost any continuous function given enough neurons and data—a property known as the universal approximation theorem. During training, the network adjusts the weights using backpropagation combined with an optimization algorithm (like gradient descent) to minimize a chosen loss function.

Unlike SVMs, where only a subset of data points (the support vectors) defines the decision boundary, an MLP uses all training points to iteratively update its weights. This allows it to capture highly intricate patterns, but it also makes MLPs more sensitive to overfitting, requiring careful tuning of architecture, learning rate, and regularization.

In practice, MLPs are highly flexible and widely used, especially for datasets where relationships between features are nonlinear and complex, making them a cornerstone of modern deep learning.

Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union. Neither the European Union nor the granting authority can be held responsible for them.

2.7 Explainable AI

As predictive models are increasingly used to inform decisions, understanding why a model produces a given prediction becomes as important as the prediction itself. Explainability supports trust, error analysis and fairness checks: it helps practitioners to verify that models rely on plausible signals rather than spurious correlations or leaked information, and it provides stakeholders with interpretable evidence for automated decisions. In this work we rely on SHAP (SHapley Additive exPlanations) as a primary explanation method because it offers a principled, unified way to attribute a model's output to individual input features.

SHAP is founded on the Shapley value concept from cooperative game theory: each feature is treated as a "player" that contributes to the final prediction, and the Shapley value for feature i quantifies its average marginal contribution across all possible coalitions of features. Formally, for a model f defined on a feature set N, the Shapley value ϕ_i is given by the weighted sum of marginal contributions

$$\phi_i = \sum_{S \subseteq N \setminus \{i\}} \frac{|S|!\,(|N| - |S| - 1)!}{|N|!} \big(f(S \cup \{i\}) - f(S)\big) \tag{2}$$

which guarantees desirable axiomatic properties such as local accuracy (additivity of contributions), consistency (if a feature's contribution increases under a model change its attribution does not decrease), and missingness (features absent from the model receive zero attribution). In practice SHAP provides both local explanations for single predictions and global summaries obtained by aggregating local attributions across a dataset, thus connecting instance-level reasoning with population-level insights.

Different SHAP implementations make the approach practical for real models and datasets. TreeSHAP exploits the structure of tree ensembles to compute exact Shapley values efficiently for decision-tree-based models, while KernelSHAP approximates Shapley values for arbitrary black-box predictors via a weighted sampling scheme. Typical diagnostic visualisations include summary plots (showing feature importance and directionality across many instances), dependence plots (revealing how a feature's effect varies with its value and with interacting features), and force or decision plots (highlighting the feature contributions for individual predictions). In this work, two main SHAP visualisations are used: *bar plots* and *beeswarm plots*. The bar plots report the mean absolute SHAP value for each feature, offering a global ranking of importance across the dataset, while the beeswarm plots expand on this view by showing the distribution and sign of the SHAP values for each feature, with colour encoding the raw feature value. Together, they allow both an overall assessment of which variables are most influential and a detailed understanding of how their values push predictions toward specific classes, ensuring that model behaviour can be interpreted in light of domain knowledge.

SHAP has important limitations. First, attributions depend on the fitted model and on the background data used to compute marginal contributions: different models or different background samples can produce different explanations. Second, when features are strongly correlated, Shapley attributions can be harder to interpret since contribution may be spread across correlated variables. In this report, SHAP is employed to validate that selected features carry sensible predictive signal, and to inspect why the model succeeds

Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union. Neither the European Union nor the granting authority can be held responsible for them.

or fails on particular classes using compact visual summaries. These explanations are used as diagnostic tools rather than definitive proofs, helping to guide further data curation, feature engineering and model refinement.

3. Predicting risk awareness

3.1 Dataset and preprocessing

The dataset used in this study originates from a survey conducted within the SAFE-LAND project, which collected information from respondents about socio-demographic characteristics, personal experiences, and perceptions related to natural hazards such as floods and landslides. Each row in the dataset represents an individual respondent, whereas each column corresponds to a specific survey question or derived variable.

The raw dataset consists of 778 samples and 77 attributes. During preprocessing, we applied various cleaning and transformation operations to ensure data consistency and suitability for modeling. A set of unnecessary columns (including identifiers such as `Codiceanonimo` (ID code) and open-text responses like `@110a_TSaltro` (other notes) were removed. Categorical attributes representing geographical information (e.g., city and region of origin) were converted into a numerical format using label encoding. Then, the dataset was split into two subsets corresponding to two distinct classification tasks:

- **Flood task**: focused on predicting individuals' experience, knowledge, worry, and **awareness** of people related to flood risk.
- **Landslide task**: focused on predicting the same four perceptual dimensions, but in relation to landslide risk.

Each subset combines a dedicated set of hazard-specific features with a shared pool of generic socio-demographic and psychological variables. After the feature-selection stage used in the pipeline, each task dataset contains **47 predictor attributes** and **4 target variables**.

The targets for each task and their values are listed below. For both floods and landslides, we assessed four dimensions of risk perception: *Knowledge*, *Awareness*, *Worry*, and *Experience*, as specified in Table 5 of Deliverable 3.2. Each dimension was originally measured through multiple survey items and then recoded into discrete classes to obtain harmonized target variables for the modeling phase.

In the following, we describe the conceptual rationale for each factor, the items used for measurement, and the final scoring scheme adopted for the flood and landslide targets. In some cases, adjacent or sparsely populated categories were merged to reduce class imbalance and improve the robustness of model training, ensuring smoother and more interpretable distributions of target variables. Figures 4a and 4b show the distribution of classes before mapping.

Knowledge. Good knowledge can promote an adequate level of risk perception. Knowledge was measured through four items: (1) knowledge of previous local floods or landslides (Yes/No); (2) knowledge of correct response behaviors and emergency management procedures (protective behaviors, warning systems, safe areas, etc.); (3) frequency of keeping

Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union. Neither the European Union nor the granting authority can be held responsible for them.

informed about flood, landslide, and weather warnings (0=not informed to 3=very informed); (4) perceived knowledge on how to respond in case of a flood or landslide (0=no knowledge to 3=a lot of knowledge). These items were combined into a single variable, `Knowledge`. Participants with low knowledge were assigned a score of −1, whereas those with adequate or high knowledge were assigned a score of 0.

Variable	Cod_feature	Questionnaire question	Feature explanation	Value coding	Final variable
EXPERIENCE	@21aESPERIENZAalluvioni @21bESPERIENZAfrane	Which of the following sentences best indicates your experience with floods/landslides?	Type of past experience with floods or landslides	0= No experience 1= Read/seen information in the news or other media 2= Direct experience of a friend/relative 3= Direct experience of a friend/relative + Read/seen information in the news or other media 4= Direct personal experience 5= Direct personal experience + Direct experience of a friend/relative 6= Direct personal experience + Direct experience of a friend/relative + Read/seen information in the news or other media RECODE from 0 to 3 = 0 (indierect experience) from 4 to 6 = 1 (direct experience)	Based on this item, we created a final unique variable titled Experience. For this variable, we assigned to each participant score of 0 in case of indirect experience and a score of 1 in case of direct experience
KNOWLEDGE	@22aCONOSCENZAPASSATO alluvioni @22bCONOSCENZAPASSATO frane	Do you know if in the area where you live there have been floods/landslides in the past?	Knowledge of Previous local floods or landslides	0= no 1= yes	Based on these four items, we created a final unique variable titled Knowledge. For this variable, we assigned to each participant a score of -1, in case of low knowledge and a score of 0 in case of adequate or high knowledge.
	@29aPREPARAZIONEIN BASEACONOSCENZE alluvione @29bPREPARAZIONE INBASEACONOSCENZE frana	Tick the boxes for which you feel informed about the following issues in the event of a flood or landslide	Knowledge of response behaviours and emergency management in case of floods or landslides (e.g., protective behaviors, warning systems, risk/safe areas, etc.) (Selection of known options)	0= not informed 1= a little informed (1-2 acquaintances) 2= somewhat informed (3-4 acquaintances) 3= very well informed (5-7 acquaintances)	
	@210bPREPARAZIONEI NBASEALTENERSIINFORMATO alluvioni @210aPREPARAZIONE INBASEALTENERSIINFORMATO frane	Do you keep abreast of floods/landslide warnings?	Keeping up to date with flood and landslide warnings	0 = not informed 1= a little informed 2 = somewhat informed 3 = very well informed	
	@212aAUTOEFFICACIA alluvione @212bAUTOEFFICACIA frana	Do you have enough knowledge to protect yourself/response measures in the event of a flood/landslide?	Level of knowledge on how to protect oneself/respond in case of flood/flood, or landslide	0=I have no knowledge 1= I have little knowledge 2= I have moderate knowledge 3=I have a lot of knowledge	
WORRY	@23aEMOZIONE frana @23bEMOZIONE alluvione	What emotion did you feel or do you think you would feel if experienced a flood/landslide?	Emotional response experienced in the past or anticipated in the future if a flood or landslide were to occur	1 = Calm 2 = Nervousness 3 = Worry 4= Anxiety 5 = Fear 6 = Panic 7 = Terror	Based on these two items, we created a final unique variable titled Worry. For this variable, we assigned to each participant a score of -1, in case of low worry and a score of 0 in case of adequate worry and a score of 1 in case of high worry
	@24aPREOCCUPAZIONE ALLARMEalluvione @24bPREOCCUPAZIONE ALLARMEfrana	If the news were to report a severe flood or landslide warning predicted for tomorrow, how would you feel? Please indicate your level of worry	Level of worry in response to a flood or landslide warning for the following day	0= not worried 1= little worried 2= moderately worried 3= very worried	
AWARENESS	@26aCONSAPEVOLEZZA RISCHIOCASAalluvione @26bCONSAPEVOLEZZA RISCHIOCASAfrana	Is your residence located in the flood risk area?	Awareness of living in a flood/landslide risk area	0= no 1= I don't know 2= yes	Based on these three items, we created a final unique variable titled Awareness. For this variable, we assigned to each participant a score of -1, in case of low awareness and a score of 0 in case of adequate awareness and a score of 1 for high awareness
	@27aCONSAPEVOLEZZA RISCHIOCITTAalluvione @27bCONSAPEVOLEZZA RISCHIOCITTAfrana	Do you think there are parts of your town/city more exposed to landslide or flood risk than others?	Awareness of areas in the city most at risk of flooding or landslides	0 = I don't know 1 = There is no risk in the city 2 = Yes, there are parts of the city that are more at risk 3 = The city has the same risk everywhere	

Tab. 2
Overview of all questionnaire items used to construct the four target variables—*Experience*, *Knowledge*, *Worry*, and *Awareness*—for both floods and landslides. For each item, the table reports the corresponding dataset code, the original questionnaire question, a brief explanation of the feature, and the exact value coding as used in the raw dataset. The final column describes how each set of items was combined to generate the unified target variables employed in the modelling tasks.

To make the link between questionnaire items and the dataset explicit, we report here the exact item codes used (see Table 2). The raw questionnaire items that contribute to the flood (landslide) Knowledge variable are: `@22a(b) CONOSCENZA PASSATO alluvioni (frane)` (knowledge of previous local events), `@29a(b) PREPARAZIONE IN BASE A CONOSCENZE alluvione (frana)` (knowledge of response behaviors), `@210a(b) PREPARAZIONE IN BASE AL TENERSI INFORMATO alluvione (frana)`(frequency of keep-

Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union. Neither the European Union nor the granting authority can be held responsible for them.

ing informed) and `@212a(b) AUTO EFFICACIA alluvione (frana)` (self-reported preparedness/knowledge).

For modeling purposes, this variable was expanded to a five–class scale, resulting in the following final categories:

$$[-4.0, -3.0, -2.0, -1.0, 0.0].$$

Awareness. Low awareness can hinder risk perception and reduce the effectiveness of protective responses. Awareness was assessed through three items: (1) awareness of living in a risk area (0=No, 1=Don't know, 2=Yes); (2) perception of which city areas are the most at risk (3=same risk everywhere to 0=don't know); (3) awareness of the causes of floods or landslides. Based on these items, a single variable `Awareness` was created, assigning a score of -1 for low awareness, 0 for adequate awareness, and 1 for high awareness. To obtain balanced classes for modeling, extreme categories (-3.0 and 2.0) were merged, leading to the following four classes:

$$[-2.0, -1.0, 0.0, 1.0].$$

The dataset features that feed the Awareness target are `@26a(b) CONSAPEVOLEZZA RISCHIO CASA alluvione(frana)` (awareness of residence risk), `@27a(b) CONSAPEVOLEZZA RISCHIO CITTA alluvione (frana)` (perception of parts of the city at risk) and `@28a(b) CONSAPEVOLEZZA CAUSE alluvioni (frane)` (knowledge of causes). These items and their codings are listed in Table 2. In preprocessing, the three item codes were combined into an intermediate Awareness score (preserving original per-item codings), which was then recoded into the four modeling classes above. The exact scoring function and the thresholds used for binning are recorded in the pipeline JSON.

Worry. Worry can promote higher levels of risk perception. Worry was measured through two items: (1) emotional response experienced or anticipated in the case of a flood or landslide (from calm to terror); (2) worry in response to an anticipated flood or landslide event (0=not worried at all to 3=extremely worried). These items were synthesized into a variable `Worry`, with scores of -1 (low worry), 0 (adequate worry), and 1 (high worry). For the final targets, the original value -2.0 was merged with -1.0, resulting in four classes:

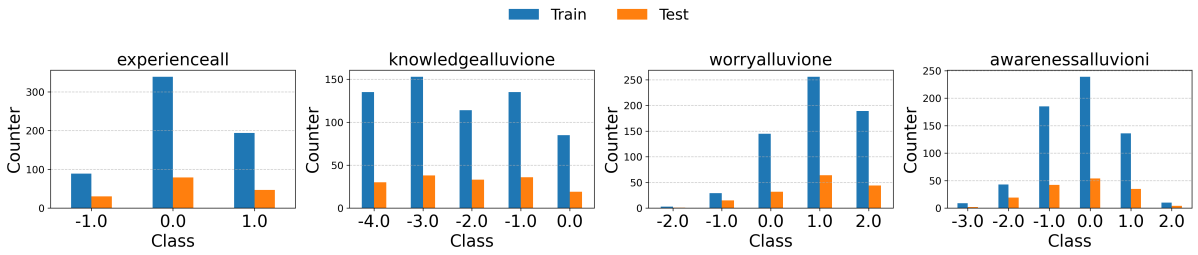
$$[-1.0, 0.0, 1.0, 2.0].$$

The dataset variables used are `@23a(b) EMOZIONE alluvione (frana)` (emotion label; coded 1–7 in the questionnaire) and `@24a(b) PREOCCUPAZIONE ALLARME alluvione (frana)` (level of worry when a warning is issued; coded 0–3). Table 2 reports the original emotion labels and their numeric codes.

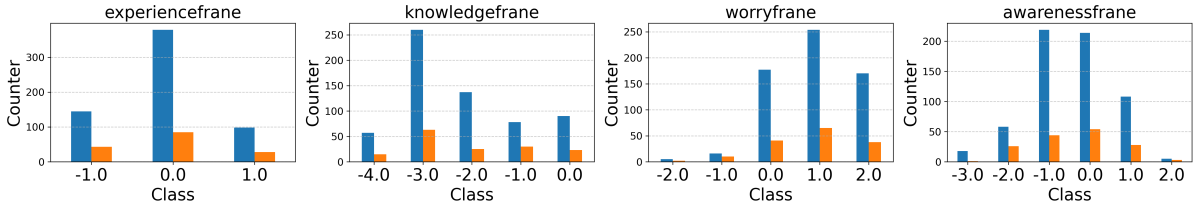
Experience. Direct experience of past events tends to increase risk perception. Past experience was evaluated through one item: type of previous experience (direct personal experience; direct experience of a friend/relative; information from media; none). A final variable `Experience` was created by assigning -1 for no experience, 0 for indirect experience, and 1 for direct experience. This variable was directly used as a three–class target:

$$[-1.0, 0.0, 1.0].$$

Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union. Neither the European Union nor the granting authority can be held responsible for them.



(a) Class distribution for flood risk.



(b) Class distribution for landslide risk.

Fig. 4: Class distribution for the target classes.

The raw dataset feature is `@21a ESPERIENZA alluvioni` and `@21b ESPERIENZA frane` (see Table 2).

Flood and Landslide Targets. The same scoring scheme and class structure were applied separately for floods (`experienceall`, `knowledgealluvione`, `worryalluvione`, `awarenessalluvioni`) and landslides (`experiencefrane`, `knowledgefrane`, `worryfrane`, `awarenessfrane`). Thus, each dimension preserves identical interpretability across the two hazard types, with the following final target classes:

- `experienceall / experiencefrane`: $[-1.0, 0.0, 1.0]$ (3 classes)
- `knowledgealluvione / knowledgefrane`: $[-4.0, -3.0, -2.0, -1.0, 0.0]$ (5 classes)
- `worryalluvione / worryfrane`: $[-1.0, 0.0, 1.0, 2.0]$ (4 classes)
- `awarenessalluvioni / awarenessfrane`: $[-2.0, -1.0, 0.0, 1.0]$ (4 classes)

Finally, the cleaned datasets for both tasks are split using an 80/20 train/test ratio (same indices used for both tasks), producing **622 training** and **156 test** samples per task. The resulting splits are saved for reproducible downstream experimentation.

3.2 Feature-Selection

The feature-selection stage implements an end-to-end pipeline for selecting informative features before fitting the final models. It first removes trivial or redundant inputs, then ranks features by relevance using both univariate and model-based methods, and finally applies a wrapper search with cross-validation to produce a compact feature set for final training and evaluation. The main steps are:

- **Prefilter (variance and correlation).** Low-variance features are removed with a variance threshold filter, which discards features that contain little or no variability

Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union. Neither the European Union nor the granting authority can be held responsible for them.

across samples (these typically carry no predictive information). Then highly correlated features are pruned: the absolute correlation matrix is inspected and, for pairs above a configurable threshold (default 0.95), one feature is removed. This two-step prefilter reduces dimensionality and multicollinearity while keeping the most informative representatives of correlated groups.

- **Univariate ranking by mutual information (top-K).** Remaining features are ranked using *mutual information* (MI) with the target. MI is a general measure of statistical dependence that quantifies how much knowing a feature X reduces the uncertainty about the target Y. Formally, for discrete variables:

$$I(X;Y) \;=\; \sum_{x,y} p(x,y)\,\log\frac{p(x,y)}{p(x)\,p(y)} \tag{3}$$

  where $p(x,y)$ is the joint probability of $X = x$ and $Y = y$, and $p(x)$, $p(y)$ are the marginals. A value of $I(X;Y) = 0$ indicates independence, while larger values indicate stronger dependence (including non-linear relationships). In practice, MI is estimated from the available samples and used as a univariate score: features are sorted by their MI with the target and the top k features (default $k = 25$) are retained as a candidate pool. This step provides a fast, model-agnostic screening that highlights individually informative predictors, but it does not account for interactions between features.

- **Model-based ranking via permutation importance.** To capture multivariate effects and interactions that univariate MI cannot detect, the pipeline next assesses feature relevance with a *model-aware* criterion based on permutation importance. The idea is simple and intuitive: measure how much a trained model's performance degrades when the values of a single feature are randomly permuted in a validation set. If permuting feature j produces a large drop in score, that feature is important to the model. If the score is largely unchanged, the feature is likely irrelevant.

  Let S_{base} be the baseline validation score of the trained model and $S_j^{(r)}$ the validation score after the r-th random permutation of feature j. The permutation importance for feature j is estimated as the average performance decrease as follows:

$$\Delta_j \;=\; S_{\text{base}} \;-\; \frac{1}{R}\sum_{r=1}^{R} S_j^{(r)} \tag{4}$$

  where R is the number of repetitions. The procedure returns both the mean Δ_j and its standard deviation across iterations, providing a measure of effect size and stability. Because this method evaluates features in the context of an actual predictive model, it naturally accounts for interactions and collinearity as experienced by that model. The resulting ranked list of features (mean decrease in score ± standard deviation) is used to select candidates for the subsequent wrapper stage (RFECV).

- **Features selection with RFECV.** From the top-ranked features obtained through permutation importance, the pipeline applies *Recursive Feature Elimination with*

Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union. Neither the European Union nor the granting authority can be held responsible for them.

Cross-Validation (RFECV), a wrapper-based approach that evaluates the contribution of each feature set by training and validating a predictive model multiple times. Unlike univariate or model-based filters, RFECV performs feature selection in a supervised and iterative manner: at each iteration, a model (in this case, a Random Forest classifier) is trained on the current subset of features, their relative importances are computed, and the least relevant features are removed. This process continues recursively until a minimum number of features is reached.

The quality of each feature subset is estimated through k-fold cross-validation, where the dataset is repeatedly partitioned into k training and validation splits. For each subset size n, RFECV computes the mean validation score $\bar{s}(n)$ (the balanced accuracy) across all folds:

$$\bar{s}(n) = \frac{1}{k} \sum_{i=1}^{k} s_i(n) \tag{5}$$

where $s_i(n)$ denotes the validation score in fold i. The optimal number of features n^* is then chosen as the one that maximizes $\bar{s}(n)$:

$$n^* = \arg\max_n \bar{s}(n) \tag{6}$$

By using stratified folds, the procedure preserves class balance during cross-validation, ensuring fair evaluation even in the presence of uneven class distributions. Once the optimal subset of features is identified, a final Random Forest model is retrained using only the selected predictors, and its performance is evaluated on the held-out test set using balanced accuracy as a metrics.

Figure 5 shows the RFECV validation curves for all flood target variables. In each case, performance increases rapidly and then stabilizes as more features are added. The selected number of features (red dashed line) does not always coincide with the absolute maximum score, as RFECV follows a greedy elimination path and favors the most stable subset rather than the global optimum. This results in compact feature sets that balance accuracy and model robustness between targets. This also happens for land-slide targets.

- **Reproducibility.** The pipeline is deterministic via a configurable random state and writes a compact JSON summary per target containing the task type, numbers of features at each stage, the final selected features and evaluation metrics. Intermediate results (mutual information series, permutation importance table, and the RFECV object) are also returned for further inspection.

This modular design combines univariate filters (variance, correlation, mutual information) with model-aware methods (permutation importance and RFECV) to produce a small, performant feature set while controlling computational cost.

Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union. Neither the European Union nor the granting authority can be held responsible for them.

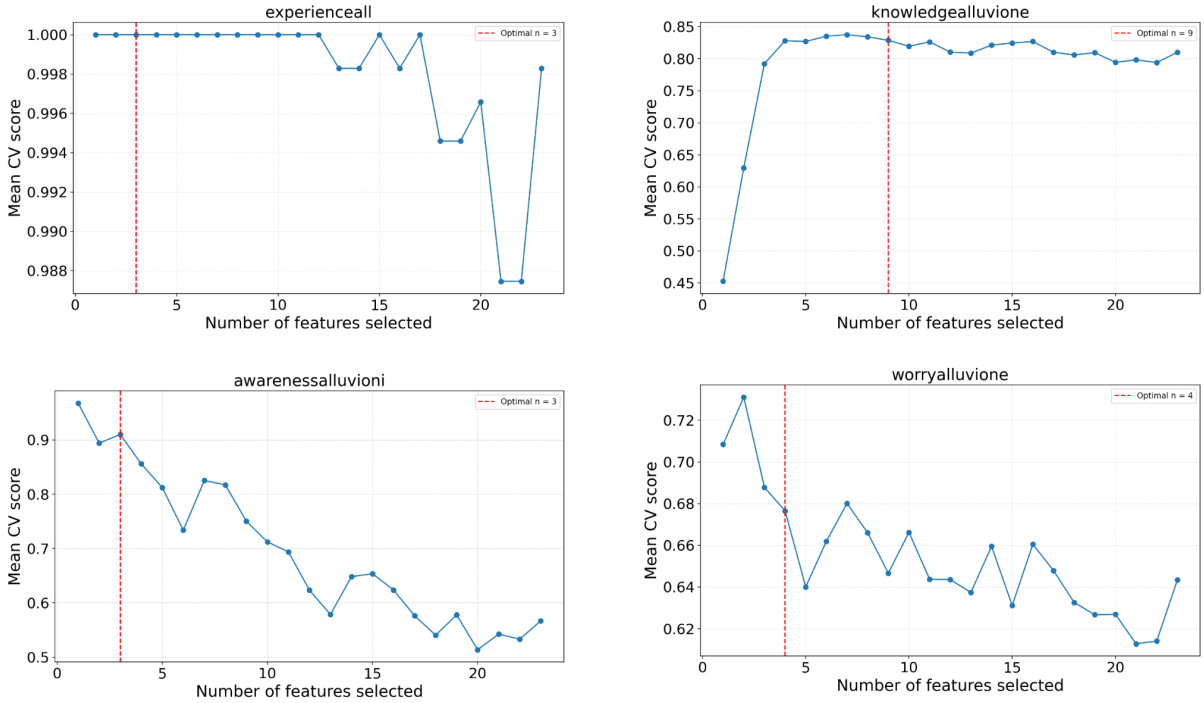


Fig. 5: RFECV cross-validation curves for the four flood-related targets. Each plot shows the mean validation score as a function of the number of selected features. The red dashed line marks the subset chosen by RFECV, representing the best-performing configuration along the elimination path.

3.3 Train and Test

All classifiers, both for flood and landslide risk tasks, were trained and evaluated following the same protocol. For each model we performed an exhaustive grid search with five-fold cross-validation, optimizing the balanced accuracy to account for class imbalance. Formally, given a candidate hyperparameter configuration θ, the cross-validated score is the average validation score across folds, and the selected configuration is

$$\hat{\theta} = \arg\max_{\theta} \frac{1}{k} \sum_{i=1}^{k} s_i(\theta), \tag{7}$$

where $s_i(\theta)$ is the balanced accuracy on fold i and $k = 5$. After selecting $\hat{\theta}$ the estimator was retrained on the full training set and evaluated on the held-out test set; predictions and fitted models were saved for reproducibility.

The Random Forest classifier served as a strong, tree-based baseline. Its grid varied the number of trees, tree depth and simple split/leaf regularization:

- $n\_estimators \in \{100, 200, 300\}$: increasing the number of trees generally improves stability and reduces variance, while keeping training costs manageable.
- $max\_depth \in \{\text{None}, 10, 20, 30\}$: limiting tree depth controls overfitting; the value `None` allows fully grown trees as a reference.
- $min\_samples\_split \in \{2, 5, 10\}$: larger values enforce more conservative splits, helping to regularise trees and avoid fragmentation of small classes.

Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union. Neither the European Union nor the granting authority can be held responsible for them.

- $min\_samples\_leaf \in \{1, 2, 4\}$: increasing the minimum number of samples per leaf smooths the decision boundaries and reduces overfitting.
- $max\_features \in \{\text{sqrt}, \text{log2}\}$: these widely used heuristics determine the number of candidate features considered at each split, affecting tree diversity and ensemble decorrelation.

A Gradient Boosting classifier was tuned next, with a compact grid over:

- $n\_estimators \in \{100, 200\}$: as previously noted, increasing the number of trees generally improves performance, but in boosting it also interacts with the learning rate; thus we limited the search to moderate values to balance accuracy and training cost.
- $max\_depth \in \{3, 5\}$: unlike Random Forests, boosting typically benefits from shallow trees that act as weak learners; depths of 3–5 are standard choices to control model complexity while retaining expressive power.
- $learning\_rate \in \{0.1, 0.05\}$: the learning rate governs how much each tree contributes to the ensemble; smaller values provide more conservative updates and reduce overfitting, at the cost of requiring more trees. We tested two commonly adopted rates to assess this trade–off.

XGBoost was treated carefully with label encoding: the target labels were encoded into integer indices $0 \dots K-1$ on the training set and decoded back to the original labels after prediction. XGBoost's parameters are the following:

- $n\_estimators \in \{100, 200, 400\}$: larger values were included due to XGBoost's efficient shrinkage and regularisation, which often benefit from deeper boosting rounds.
- $max\_depth \in \{3, 5, 7\}$: deeper trees (up to depth 7) allow more complex interactions to be captured, while still keeping the model within reasonable computational limits.
- $learning\_rate \in \{0.1, 0.05, 0.01\}$: a wider range was explored to balance aggressiveness of updates (0.1) against more conservative, fine–grained boosting (0.01).
- $subsample \in \{1.0, 0.8\}$: subsampling the training instances acts as an additional regulariser, reducing variance and improving robustness, especially for noisier targets.
- $colsample\_bytree \in \{1.0, 0.8\}$: column subsampling helps decorrelate trees and limit overfitting, in line with standard XGBoost practice.

The XGBoost models used a multiclass log-loss evaluation metric internally during training.

The histogram-based Gradient Boosting implementation (HistGradientBoosting) was also tuned, considering the parameters:

Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union. Neither the European Union nor the granting authority can be held responsible for them.

- $max\_iter \in \{100, 200\}$: this parameter controls the number of boosting iterations; moderate values were tested to balance convergence and computational efficiency, given the model's fast training procedure.
- $max\_leaf\_nodes \in \{30, 60\}$: the number of leaf nodes influences the expressiveness of the individual trees; these ranges allow the model to capture non-linear patterns while preventing excessive complexity.
- $learning\_rate \in \{0.1, 0.05\}$: smaller learning rates yield more conservative updates and help reduce overfitting, whereas larger values speed up convergence.

Support Vector Machines were trained on standardized features: a `StandardScaler` was fit on the training set and applied to both train and test. The SVM grid covered regularization and kernel hyperparameters, allowing the search to find either a simple linear separator or a nonlinear kernel solution depending on the data geometry.

- $C \in \{1, 10, 100, 1000\}$: the regularization parameter controls the trade-off between maximizing the margin and minimizing classification error. A wide range was tested to cover both flexible and highly constrained models.
- $\gamma \in \{1, 0.1, 0.001, 0.0001\}$: for the RBF kernel, γ defines the influence of individual points; smaller values yield smoother decision boundaries, while larger values allow more complex, localized patterns.
- $kernel \in \{linear, rbf\}$: testing both linear and radial basis function kernels lets the model adapt to either linearly separable or more complex nonlinear patterns in the feature space.

Finally, a Multilayer Perceptron (MLP) was trained within a pipeline that standardizes inputs before the neural network. The search space covered model architecture, optimization, and regularization parameters:

- $hidden\_layer\_sizes \in \{(50,), (100,), (100, 50)\}$: different layer sizes and depths were tested to explore trade-offs between model complexity and generalization.
- $activation \in \{\text{relu}, \tanh\}$: commonly used activation functions allowing either piecewise-linear (ReLU) or smooth nonlinear (tanh) transformations.
- $solver \in \{\text{adam}, \text{lbfgs}\}$: optimization algorithms were chosen to compare adaptive stochastic gradient descent (Adam) with a quasi-Newton method (LBFGS), balancing convergence speed and stability.
- $\alpha \in \{10^{-4}, 10^{-3}\}$: L2 regularization coefficient, tested to reduce overfitting while keeping the network sufficiently flexible.
- $learning\_rate\_init \in \{10^{-3}, 10^{-2}\}$: initial learning rate for the optimizer, controlling step size in weight updates; values were selected to balance convergence and stability.

Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union. Neither the European Union nor the granting authority can be held responsible for them.

- $max\_iter \in \{1000, 2000\}$: maximum number of training epochs to allow sufficient convergence without excessive computation.
- $tol \in \{10^{-4}, 10^{-5}\}$: convergence tolerance for the optimization; smaller values enforce stricter stopping criteria.

For each model the best estimator found by grid search was used to predict the test set and then evaluated with the same metrics reported throughout the work (accuracy, balanced accuracy, macro F_1 and the confusion matrix).

3.4 Metrics

To quantitatively assess the performance of a multiclass classification model, we consider two fundamental metrics: *accuracy* and the *macro* F1-score. Both metrics are computed from the confusion matrix, where for each class c we define: true positives (TP_c), false positives (FP_c), false negatives (FN_c), and true negatives (TN_c).

Accuracy. Accuracy measures the proportion of samples that are correctly classified among all samples. In a multiclass setting, it is defined as:

$$\text{Accuracy} = \frac{\sum_{c=1}^{K} TP_c}{N},$$

where K is the number of classes and N is the total number of samples. Although easy to interpret, accuracy alone may be insufficient when dealing with imbalanced datasets, as it can be overly influenced by the majority classes.

Macro F1-score. For each class c, we first compute *precision* and *recall*:

$$\text{Precision}_c = \frac{TP_c}{TP_c + FP_c}, \qquad \text{Recall}_c = \frac{TP_c}{TP_c + FN_c}.$$

The F1-score for class c is then the harmonic mean of precision and recall:

$$\text{F1}_c = 2 \cdot \frac{\text{Precision}_c \cdot \text{Recall}_c}{\text{Precision}_c + \text{Recall}_c}.$$

In the macro version of the metric, each class contributes equally, regardless of its frequency in the dataset. The macro F1-score is obtained by averaging the per-class F1 values:

$$\text{Macro F1} = \frac{1}{K} \sum_{c=1}^{K} \text{F1}_c.$$

This metric is particularly useful in multiclass scenarios with unbalanced class distributions, as it provides a more balanced view of model performance across all classes.

Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union. Neither the European Union nor the granting authority can be held responsible for them.

	RF	GB	HGS	XGB	SVM	MLP
Experienceall	Accuracy: 1.0000 F1: 1.0000	Accuracy: 1.0000 F1: 1.0000	Accuracy: 1.0000 F1: 1.0000	Accuracy: 1.0000 F1: 1.0000	Accuracy: 1.0000 F1: 1.0000	Accuracy: 1.0000 F1: 1.0000
Knowledgeall	Accuracy: 0.8013 F1: 0.7788	Accuracy: 0.8205 F1: 0.8032	Accuracy: 0.7244 F1: 0.7106	Accuracy: 0.7500 F1: 0.7423	Accuracy: 0.7372 F1: 0.7226	Accuracy: 0.6987 F1: 0.6981
Awarenessall	Accuracy: 0.9487 F1: 0.9432	Accuracy: 0.9487 F1: 0.9432	Accuracy: 0.9487 F1: 0.9432	Accuracy: 0.9423 F1: 0.9353	Accuracy: 0.9359 F1: 0.9254	Accuracy: 0.9359 F1: 0.9254
Worryall	Accuracy: 0.8333 F1: 0.7913	Accuracy: 0.8654 F1: 0.8601	Accuracy: 0.7949 F1: 0.7335	Accuracy: 0.8462 F1: 0.8135	Accuracy: 0.7821 F1: 0.7633	Accuracy: 0.7372 F1: 0.6927
Experiencefrane	Accuracy: 1.0000 F1: 1.0000	Accuracy: 1.0000 F1: 1.0000	Accuracy: 1.0000 F1: 1.0000	Accuracy: 1.0000 F1: 1.0000	Accuracy: 1.0000 F1: 1.0000	Accuracy: 1.0000 F1: 1.0000
Knowledgefrane	Accuracy: 0.7949 F1: 0.7262	Accuracy: 0.7628 F1: 0.6929	Accuracy: 0.7436 F1: 0.6772	Accuracy: 0.8077 F1: 0.7399	Accuracy: 0.6795 F1: 0.6015	Accuracy: 0.7372 F1: 0.6917
Awarenessfrane	Accuracy: 0.8397 F1: 0.8351	Accuracy: 0.8269 F1: 0.8247	Accuracy: 0.7821 F1: 0.7810	Accuracy: 0.8333 F1: 0.8288	Accuracy: 0.8590 F1: 0.8579	Accuracy: 0.7885 F1: 0.7839
Worryfrane	Accuracy: 0.9167 F1: 0.9353	Accuracy: 0.9167 F1: 0.9353	Accuracy: 0.8590 F1: 0.7963	Accuracy: 0.9231 F1: 0.9405	Accuracy: 0.9167 F1: 0.9274	Accuracy: 0.8451 F1: 0.8370

Tab. 3
Summary of test set performance across classifiers and targets. The table reports Accuracy and F1 score for each task. The F1 metric denotes the macro-averaged F1 score.

3.5 Model Evaluation

Table 3 summarizes test-set performance across all targets and classifiers. The two experience targets (*experienceall* and *experiencefrane*) reach perfect scores on the test set, while the perceptual targets (*knowledge*, *awareness* and *worry*) achieve lower but still informative performance under the best tree-based models. Ensemble methods (Random Forest, Gradient Boosting, XGBoost) generally yield the most robust results, but the performance landscape is nuanced. Gradient Boosting and XGBoost achieve the highest F1 scores in the *worry* tasks (e.g., 0.8601 for GB on *worryalluvione*) and *knowledge* tasks. However, SVM proves highly competitive, surprisingly outperforming all tree-based models on the *awarenessfrane* target (F1: 0.8579). Conversely, the Multilayer Perceptron (MLP) consistently lags behind, particularly in the *knowledge* and *worry* domains (e.g., nearly 17 percentage points lower than GB on *worryalluvione*), confirming that for this specific tabular dataset, simpler margin-based or ensemble approaches are more effective than neural architectures.

The confusion matrices (Figs 14–19) demonstrate that the classifiers successfully distinguish between different levels of risk perception, avoiding the pitfall of defaulting to the majority class. The distribution of predictions shows distinct True Positive counts across all target categories, confirming the models' ability to handle class imbalance. Furthermore, the off-diagonal elements reveal that misclassifications are not random but primarily occur between adjacent classes (e.g., confusing 'low worry' with 'moderate worry'). This pattern indicates that the models capture the underlying ordinal nature of the data, where errors reflect the nuanced boundaries between neighboring intensity levels rather than a systematic failure to generalize.

SHAP complements these quantitative results by identifying the features that drive

Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union. Neither the European Union nor the granting authority can be held responsible for them.

Other Variables	**@25aCONSAPEVOLEZZAINFORMAZIONI alluvione** **@25bCONSAPEVOLEZZAINFORMAZIONI frana**	How informed do you feel about the landslide and flood risk in your area?	Awareness of local landslide and flood risk	0 = not informed 1 = a little informed 2 = somewhat informed 3 = very well informed
	@28aCONSAPEVOLEZZACAUSE alluvioni **@28bCONSAPEVOLEZZACAUSE frane**	What are the options that in your view are the main causes of floods or landslides?	Awareness of the causes of floods and landslides	0 = not informed 1 = a little informed (1-2 causes) 2 = somewhat informed (3-8 causes) 3 = very well informed (9-12 causes)
	@210cPREPARAZIONEINBASE ALTENERSIINFORMATOmeteo	Do you keep a breast of floods/landslide warnings?	Keeping up to date with weather warnings	0 = not informed 1 = a little informed 2 = somewhat informed 3 = very well informed
	@211aMISUREPREVENZ Alluvioni **@211bMISUREPREVENZ Frana**	How adequate are the measures taken to prevent floods/landslide that may cause disasters in your city?	Adequacy of local disaster prevention measures in the case of floods or landslides	0 = I don't know 1 = Very inadequate 2 = Fairly adequate 3 = Very adequate
	@213aSICUREZZA alluvione **@213bSICUREZZAfrana**	How safe would you feel in case of flood or landslide?	Perception of safety in the case of floods or landslides	0 = not at all certain 1 = somewhat certain 2= fairly certain 3= very certain

Tab. 4
Overview of additional questionnaire items related to risk information, causes, prevention measures, and safety perception, reporting their feature codes, explanations, and value coding.

model decisions. Global importance plots (Figures 20 and 21) summarize the mean absolute SHAP values and show which variables exert the largest overall influence. The beeswarm summary plots (Figures 6–13) add directionality and instance-level detail: dot colour encodes the raw feature value and horizontal position the SHAP contribution for a given class, so it is possible to see whether high or low values push the model toward a specific class.

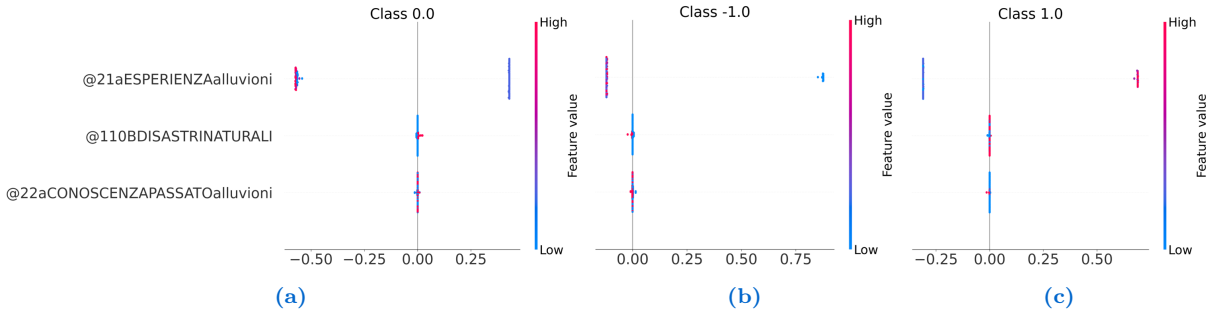


Fig. 6: Beeswarm plot illustrating feature contributions for the flood experience target.

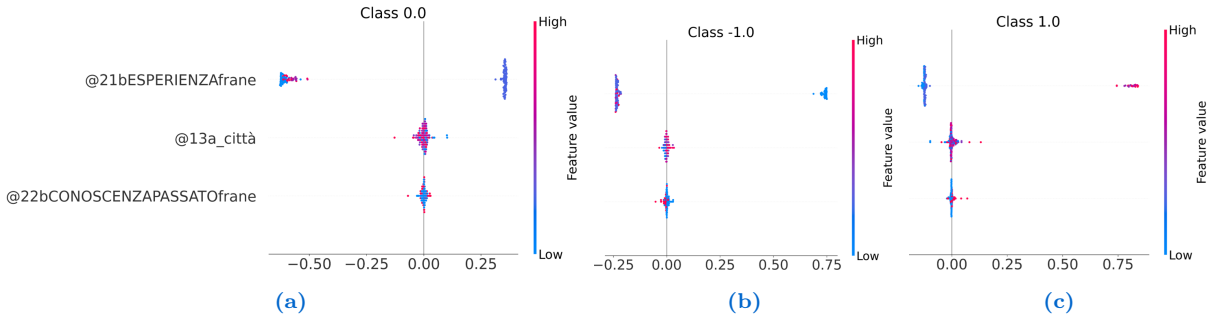


Fig. 7: Beeswarm plot illustrating feature contributions for the landslide experience target.

Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union. Neither the European Union nor the granting authority can be held responsible for them.

3.5.1 Experience

The experience models are dominated by the direct experience variables `@21a Esperienza Alluvioni` and `@21a Esperienza Frane`. High values on these items produce large positive SHAP contributions toward the "experienced" class and thus explain the near-perfect accuracy for those targets. As shown in the beeswarm plots (Fig. 6 and Fig. 7), the separation is stark and deterministic:

- For the "High Experience" class (Class 1.0): high values (red dots) of feature `@21a` (Direct experience) result in high positive SHAP values, strongly pushing the prediction toward this class. This aligns perfectly with the coding in Table 2, where higher values represent direct personal experience.
- For the "No Experience" class (Class -1.0): low values (blue dots) of `@21a` corresponding to "No experience" codes generate large positive SHAP contributions for this specific class.

This confirms that the model has correctly learned to map the raw survey code directly to the target variable without noise.

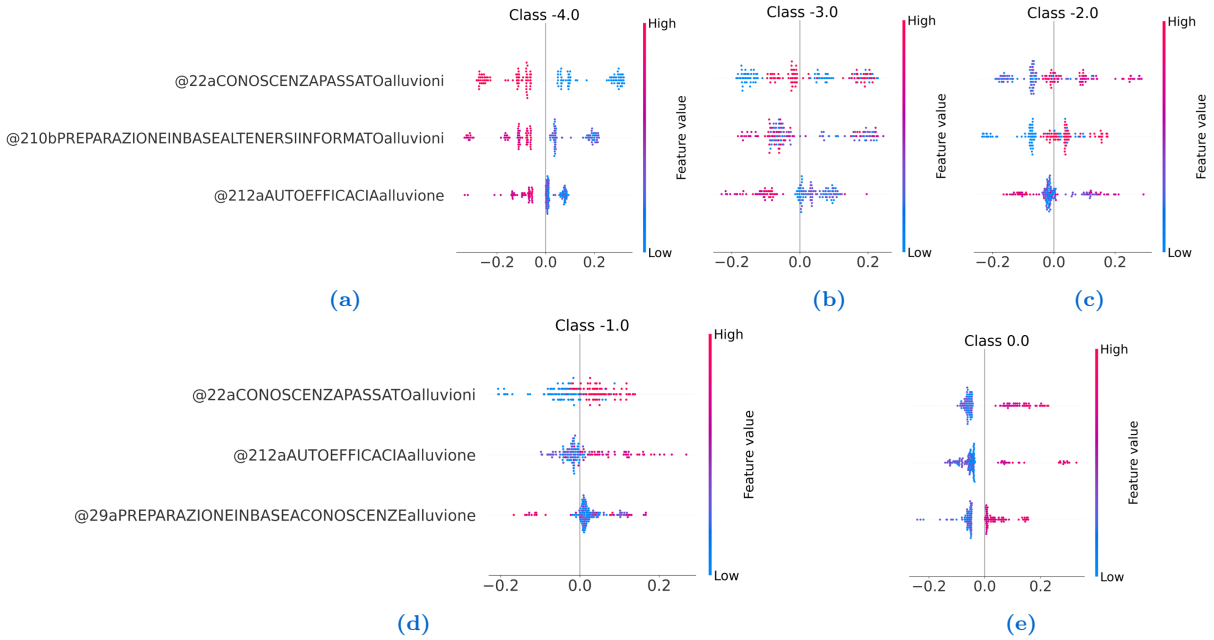


Fig. 8: Beeswarm plot illustrating feature contributions for the flood knowledge target.

3.5.2 Knowledge

For *knowledge* , the beeswarm plots (Figures 8 and 9) show that self-reported past knowledge and proactive information behaviors shift predictions toward higher knowledge classes, whereas low values on the same items produce negative attributions that favor lower classes. In particular, we have:

Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union. Neither the European Union nor the granting authority can be held responsible for them.

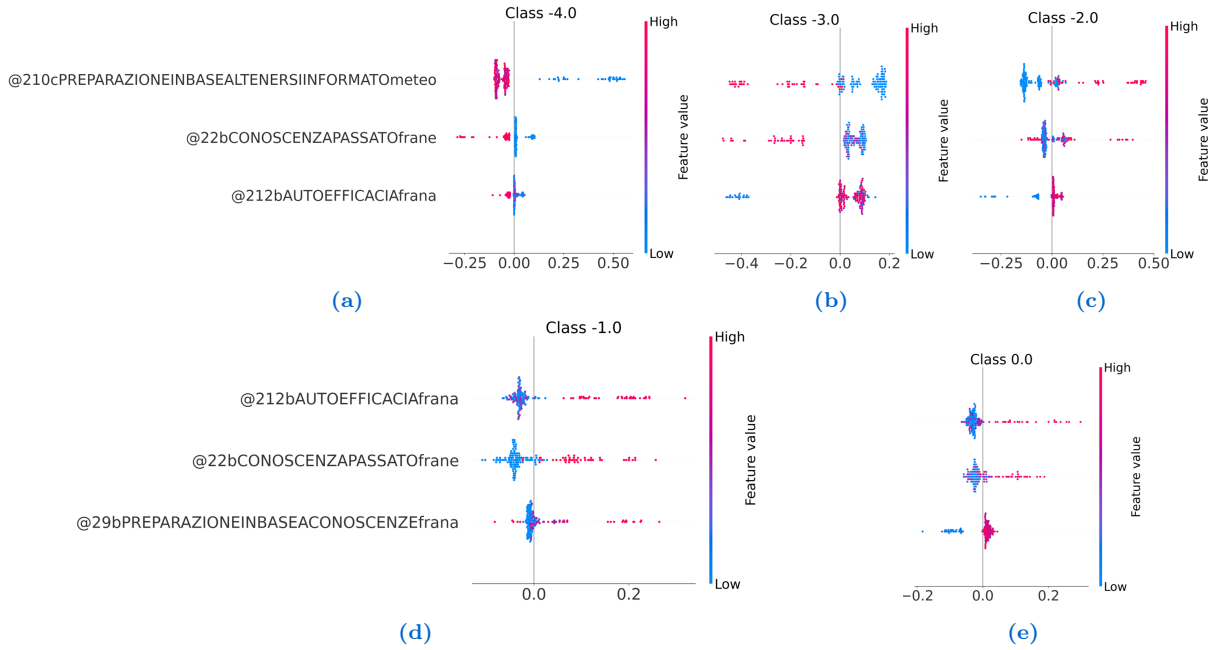


Fig. 9: Beeswarm plot illustrating feature contributions for the landslide knowledge target.

- Self-Efficacy (`@212a/b`): This feature, representing perceived knowledge on how to protect oneself, exhibits a clear gradient. In Fig. 8-9e (Class 0.0 - High Knowledge), high feature values (red dots) result in positive SHAP values. Conversely, in Fig. 8-9a (Class -4.0 - Low Knowledge), these same high values result in negative SHAP values, effectively reducing the probability of classifying a knowledgeable person as "uninformed."

- Information Seeking (`@210b/c`): As detailed in Table 2, this item measures "Keeping up to date with warnings". The SHAP plots reveal that respondents who actively check weather warnings (high values) are pushed toward higher knowledge classes.

- Past Knowledge (`@22a/b`): Knowledge of previous local events acts as a foundational feature. In the Landslide task (Fig. 9), the absence of past knowledge (blue dots) is a strong predictor for the lowest knowledge classes (Class -3.0 and -4.0), indicated by the cluster of blue dots with positive SHAP values on the right side of the plot.

3.5.3 Awareness

Awareness predictions are driven principally by household and city risk perception items. The interaction between "House Risk" and "City Risk" is visible in Figs. 10 and 11:

- House Risk (`@26a/b`): Defined in Table 2 as "Awareness of residence risk", this is the most polarizing feature. For the high awareness class (Class 1.0), high values (red dots) in `@26a` produce the strongest positive SHAP contributions (Fig. 10c). This indicates that believing one's own home is at risk is the single strongest indicator of overall high awareness.

Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union. Neither the European Union nor the granting authority can be held responsible for them.

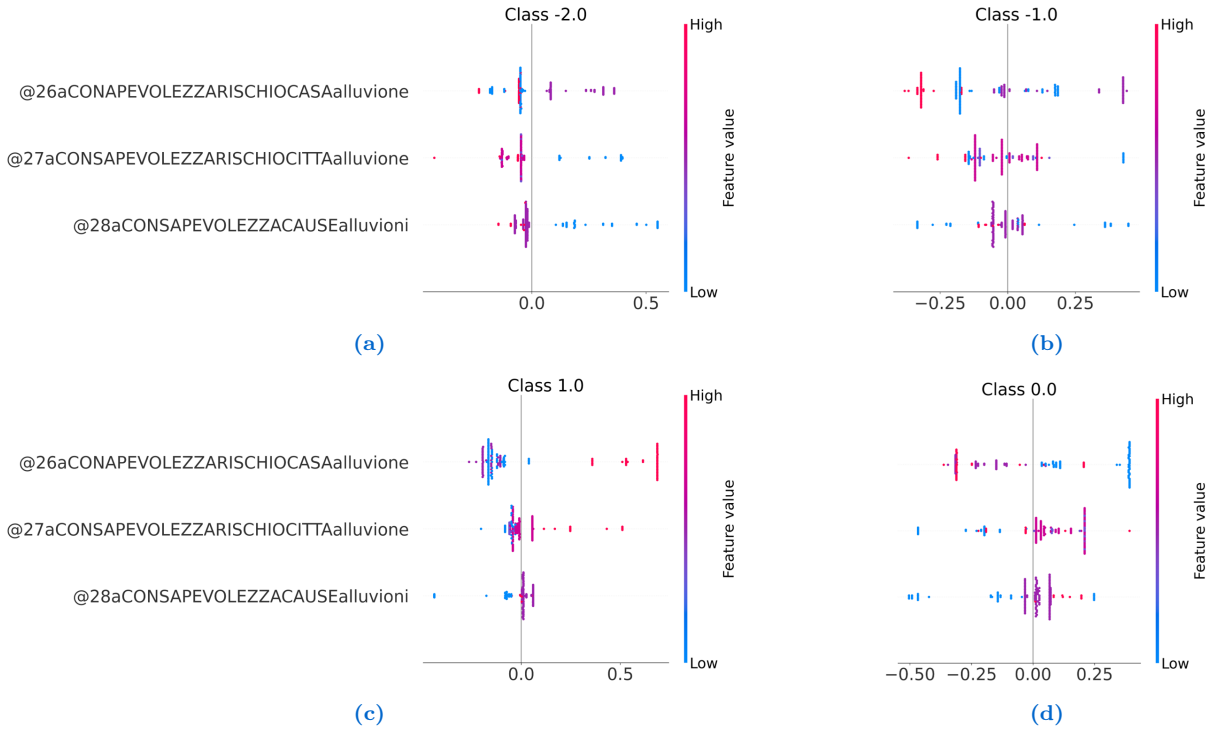


Fig. 10: Beeswarm plot illustrating feature contributions for the flood awareness target.

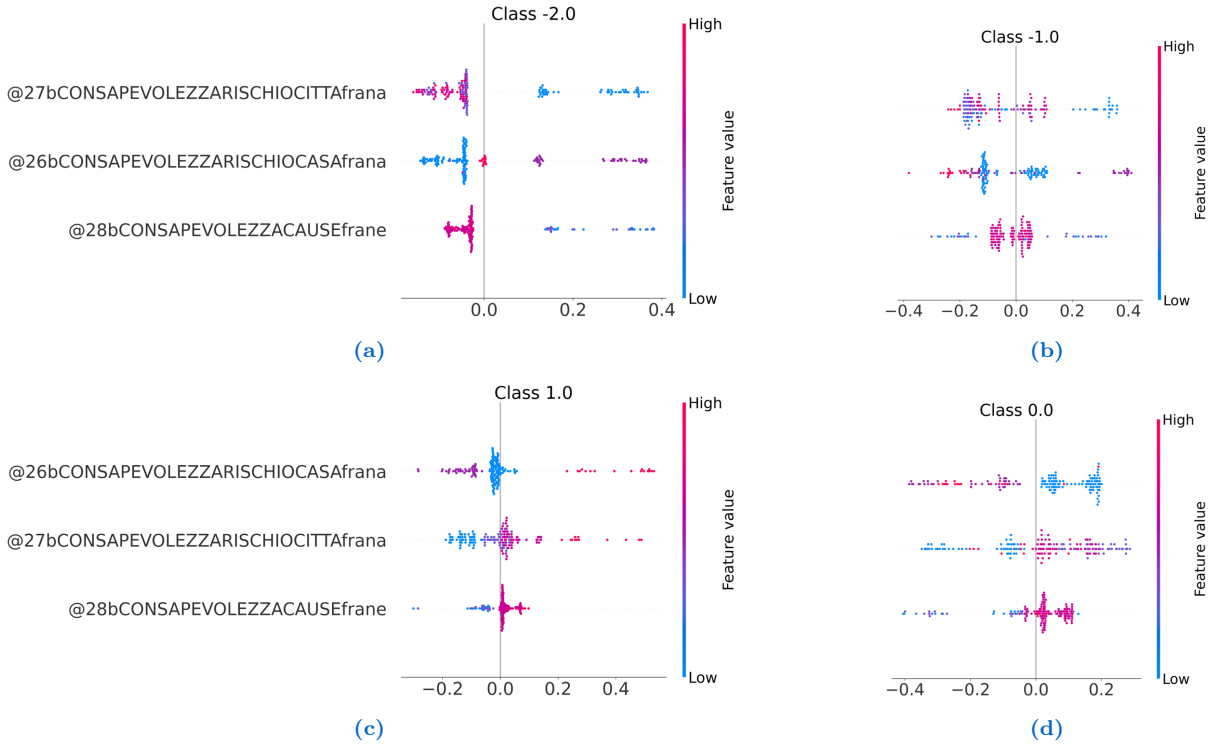


Fig. 11: Beeswarm plot illustrating feature contributions for the landslide awareness target.

- City Risk (`@27a/b`): This feature acts as a secondary support. Even if a respondent does not believe their specific house is at risk, acknowledging that parts of their city

Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union. Neither the European Union nor the granting authority can be held responsible for them.

are exposed (@27 = Yes) prevents the model from classifying them into the lowest awareness categories (Class -2.0), as seen in the negative SHAP values for high @27 values in Fig. 10a.

- Flood and landslides causes: These features, described in Table 4, capture participants' understanding of the underlying causes of floods and landslides and have a smaller impact, as shown in 10 and 11.

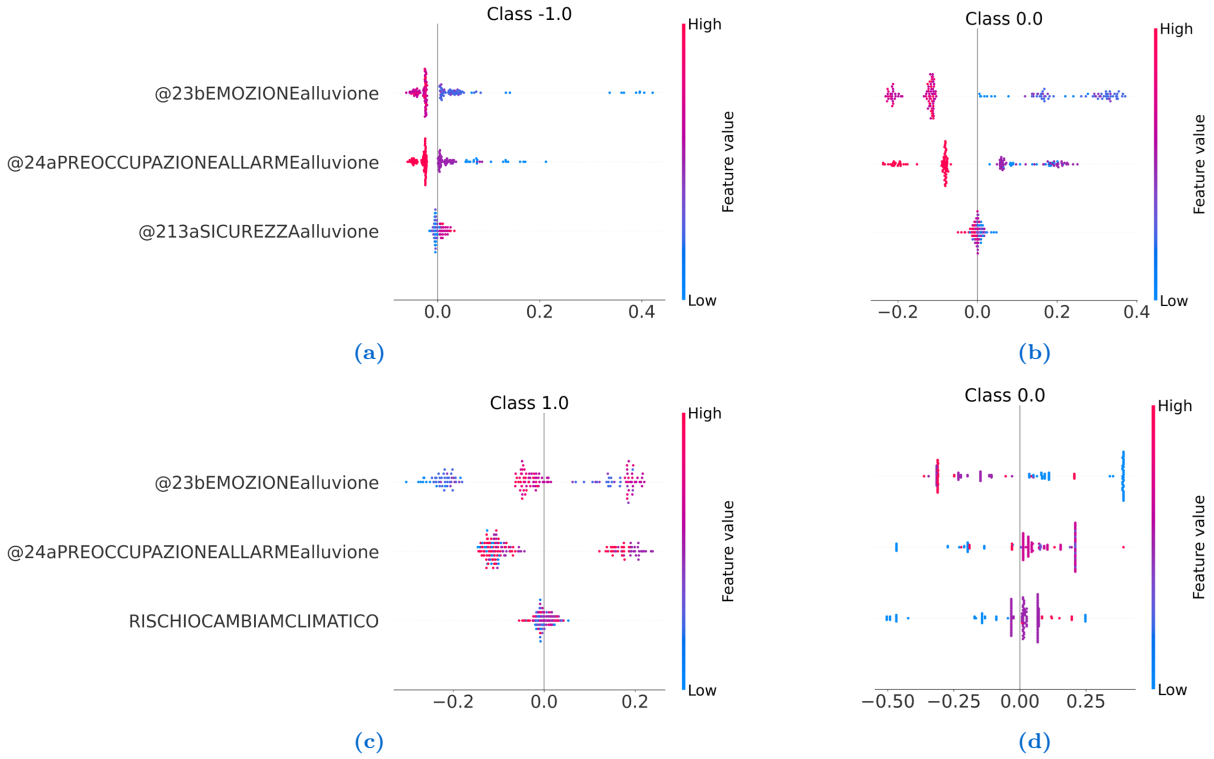


Fig. 12: Beeswarm plot illustrating feature contributions for the flood worry target.

3.5.4 Worry

Worry is clearly associated with affective items. Figures 12 (Flood) and 13 (Landslide) demonstrate a strong correlation between emotional self-reporting and the target class:

- Emotional Response (`@23a/b`): This variable, coded from 1 (Calm) to 7 (Terror), shows a linear relationship with the model's output. In Fig. 12d (Flood, Class 2.0 - High Worry), the red dots (high fear) are clustered entirely on the right (positive SHAP), showing that intense emotional responses are the primary signal for high worry.

- Alarm Reaction (`@24a/b`): Similarly, the "Level of worry in response to a warning" reinforces this pattern. Low values (blue dots) on this question are the strongest drivers for Class -1.0 (Low Worry), as seen in Fig. 12a.

Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union. Neither the European Union nor the granting authority can be held responsible for them.

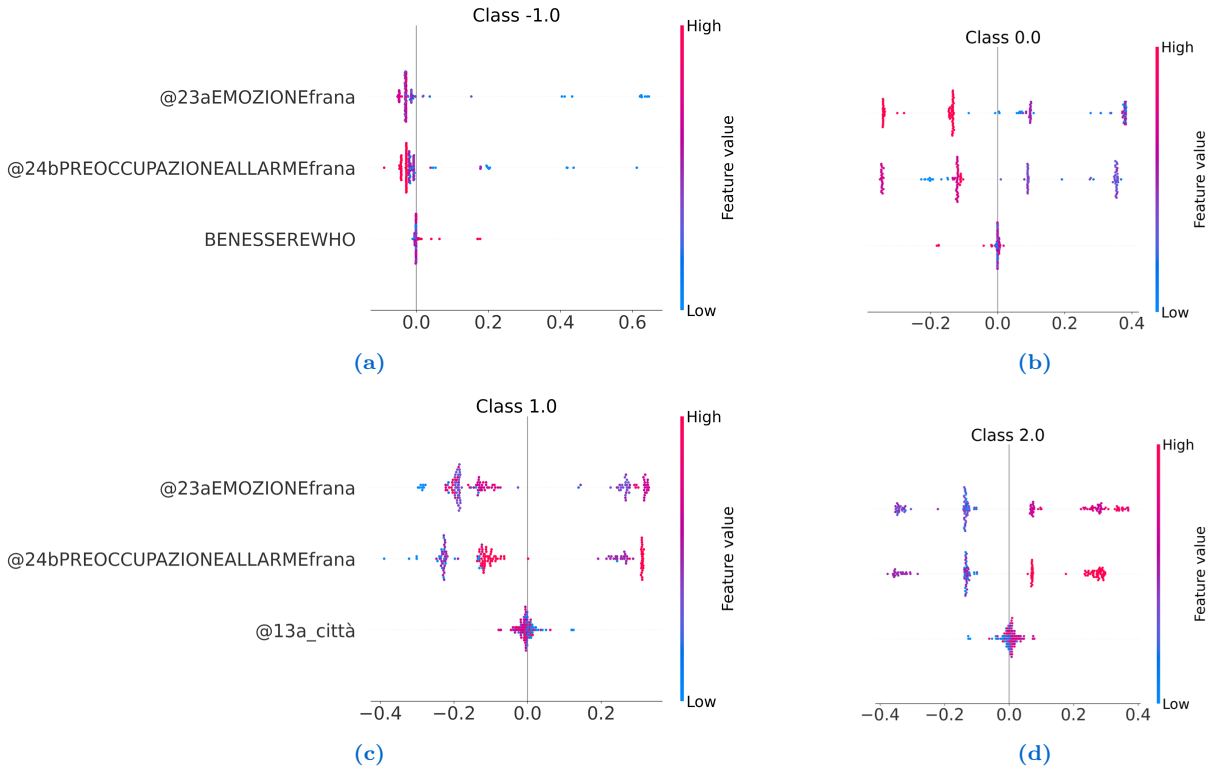


Fig. 13: Beeswarm plot illustrating feature contributions for the landslide worry target.

- Climate change risk: Interestingly, Fig. 12c highlights that the perception of climate change risk appears as a relevant feature for flood worry. High concern about climate change (red dots) contributes positively to the High Worry class (Class 1.0), suggesting that for floods, general environmental anxiety reinforces specific hazard worry.

SHAP explanations are most informative when read alongside the confusion matrices: the matrices expose which class boundaries produce errors, and SHAP shows which features contributed to those borderline decisions. Two caveats are important to mention. First, SHAP attributions depend on the chosen model and the background distribution used to compute expectations; feature rankings can change if either is altered. Second, correlated survey items can split attribution so that individually important variables may understate a combined effect; reporting grouped attributions or feature correlations helps mitigate this issue.

4. Conclusions

This deliverable described the development and evaluation of Trustworthy AI models designed to classify four key dimensions of risk perception: *Experience*, *Knowledge*, *Awareness*, and *Worry*, for both flood and landslide hazards. Starting from a processed socio-demographic survey dataset, a machine learning pipeline, incorporating feature selection and hyperparameter tuning, was implemented to compare various architectures, including Random Forest, Gradient Boosting variants, SVM, and Multilayer Perceptrons.

Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union. Neither the European Union nor the granting authority can be held responsible for them.

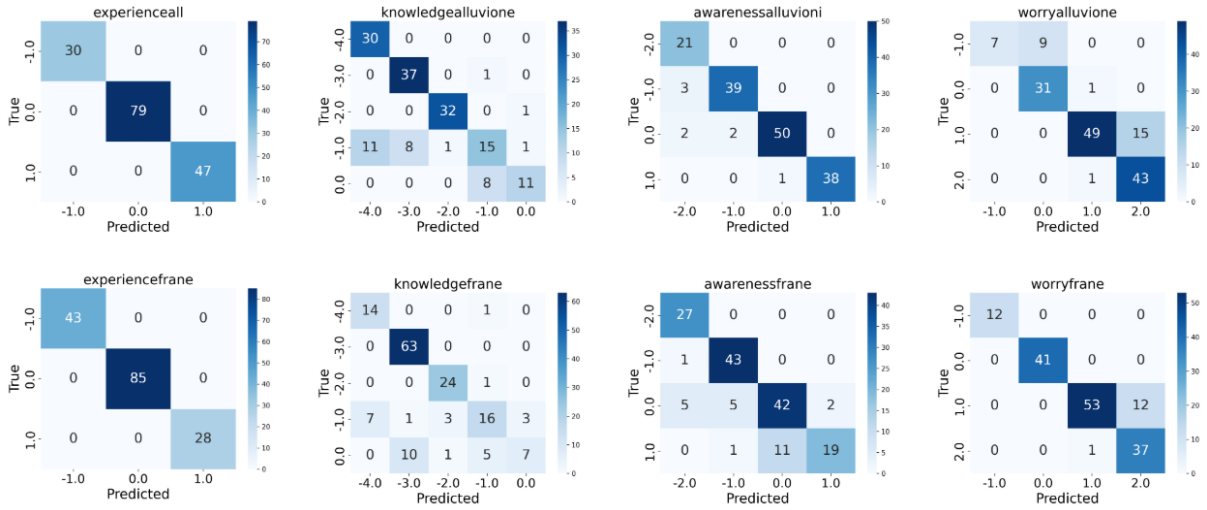


Fig. 14: Test-set confusion matrices for the Random Forest classifier.

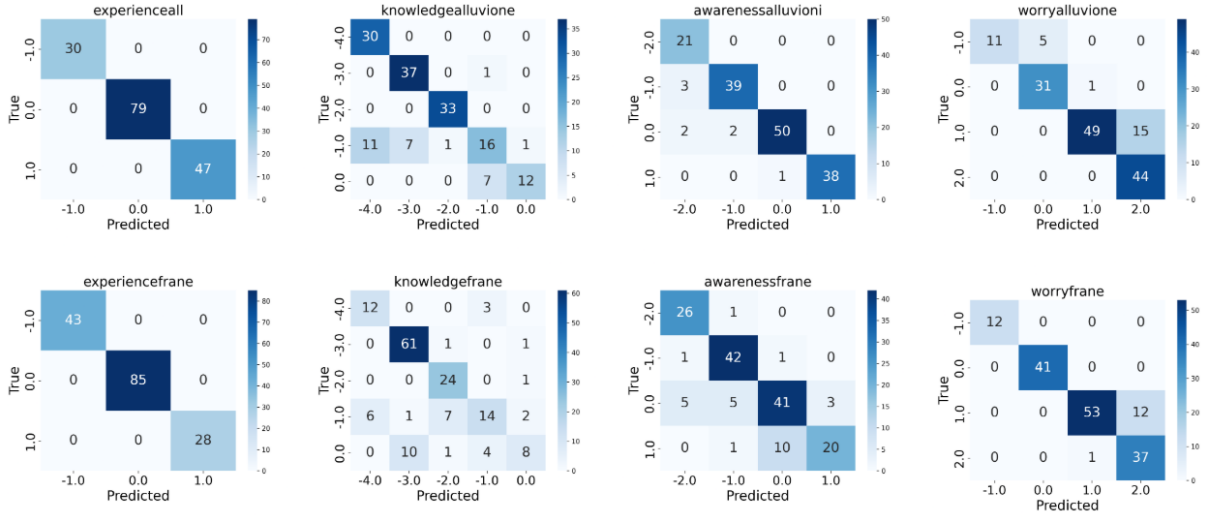


Fig. 15: Test-set confusion matrices for the Gradient Boosting classifier.

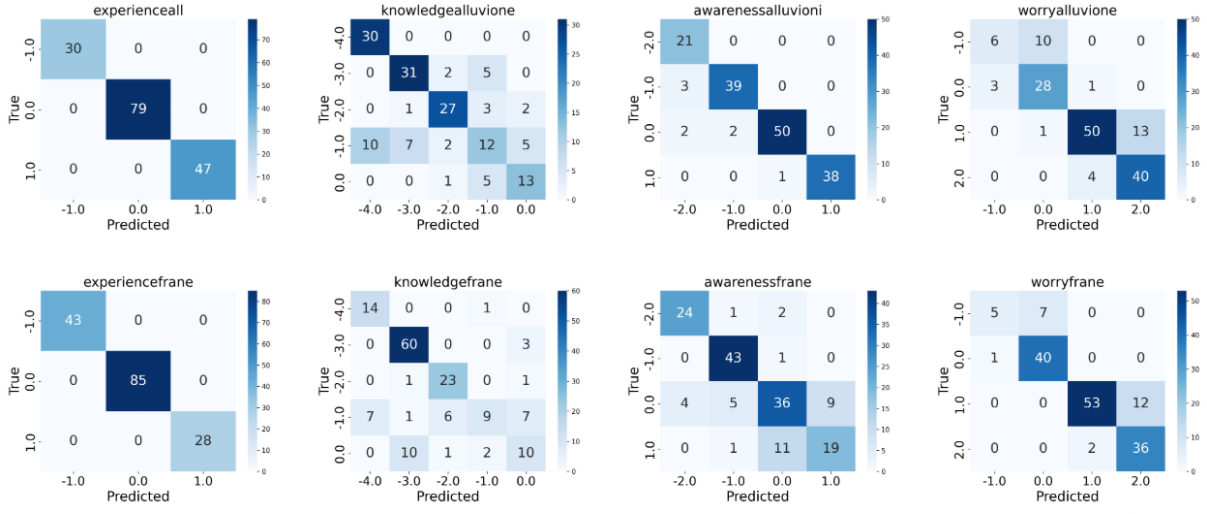


Fig. 16: Test-set confusion matrices for the Hist Gradient Boosting classifier.

Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union. Neither the European Union nor the granting authority can be held responsible for them.

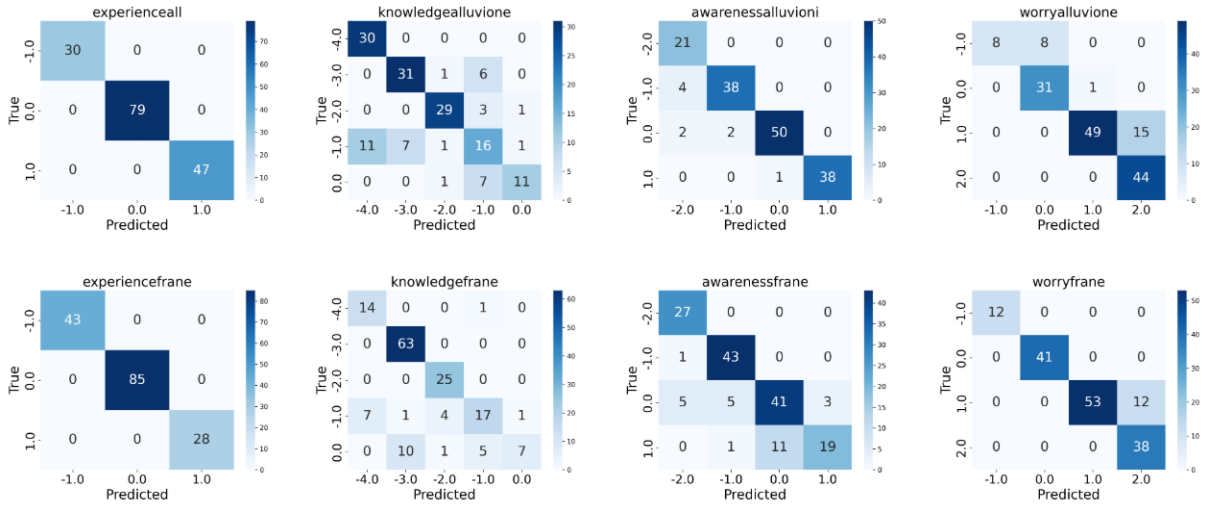


Fig. 17: Test-set confusion matrices for the Extreme Gradient Boosting classifier.

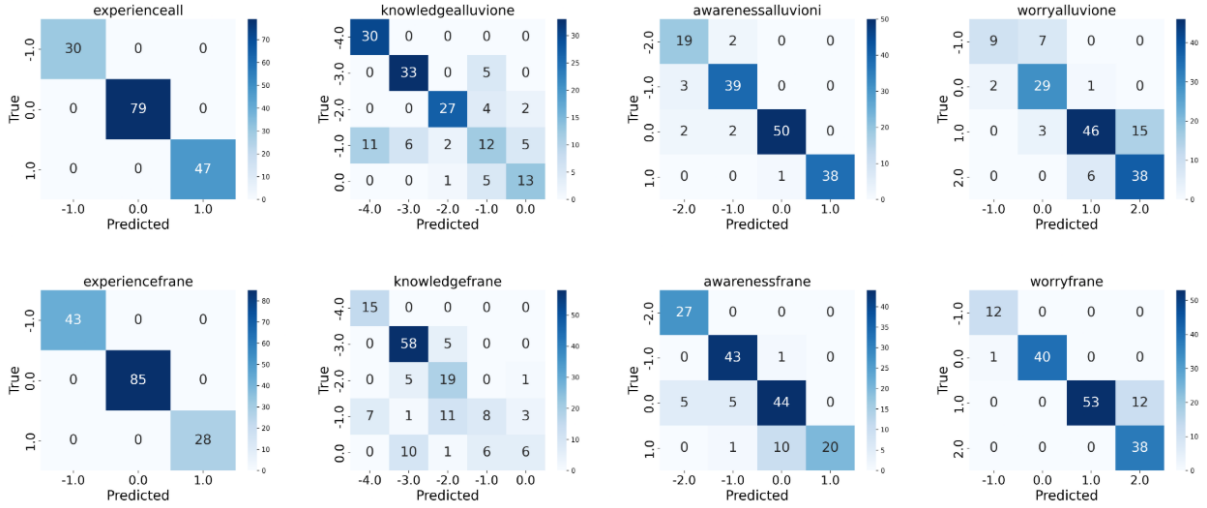


Fig. 18: Test-set confusion matrices for the Support Vector classifier.

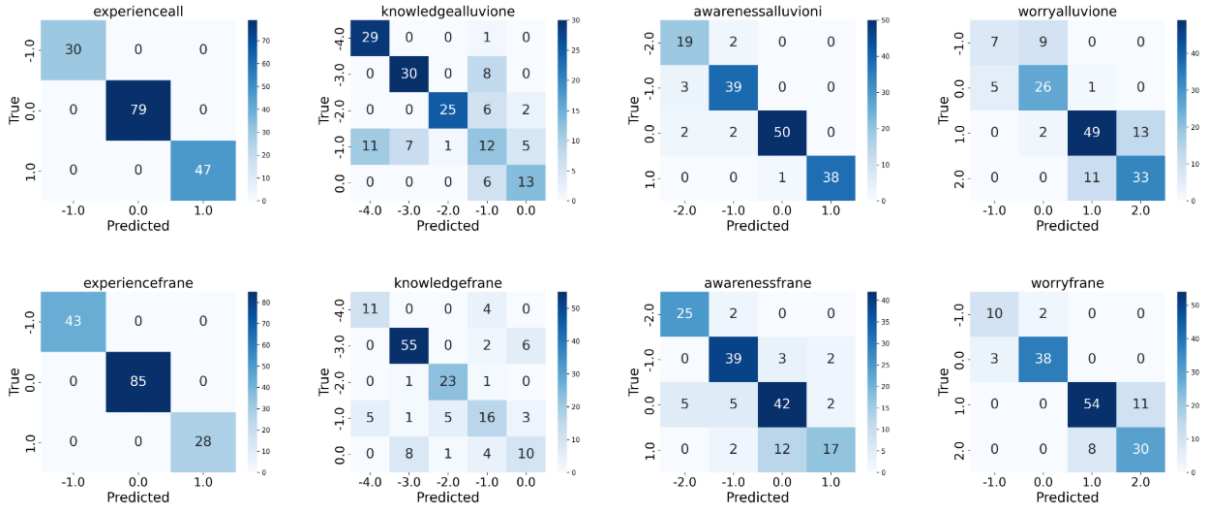


Fig. 19: Test-set confusion matrices for the Multilayer Perceptron classifier.

Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union. Neither the European Union nor the granting authority can be held responsible for them.

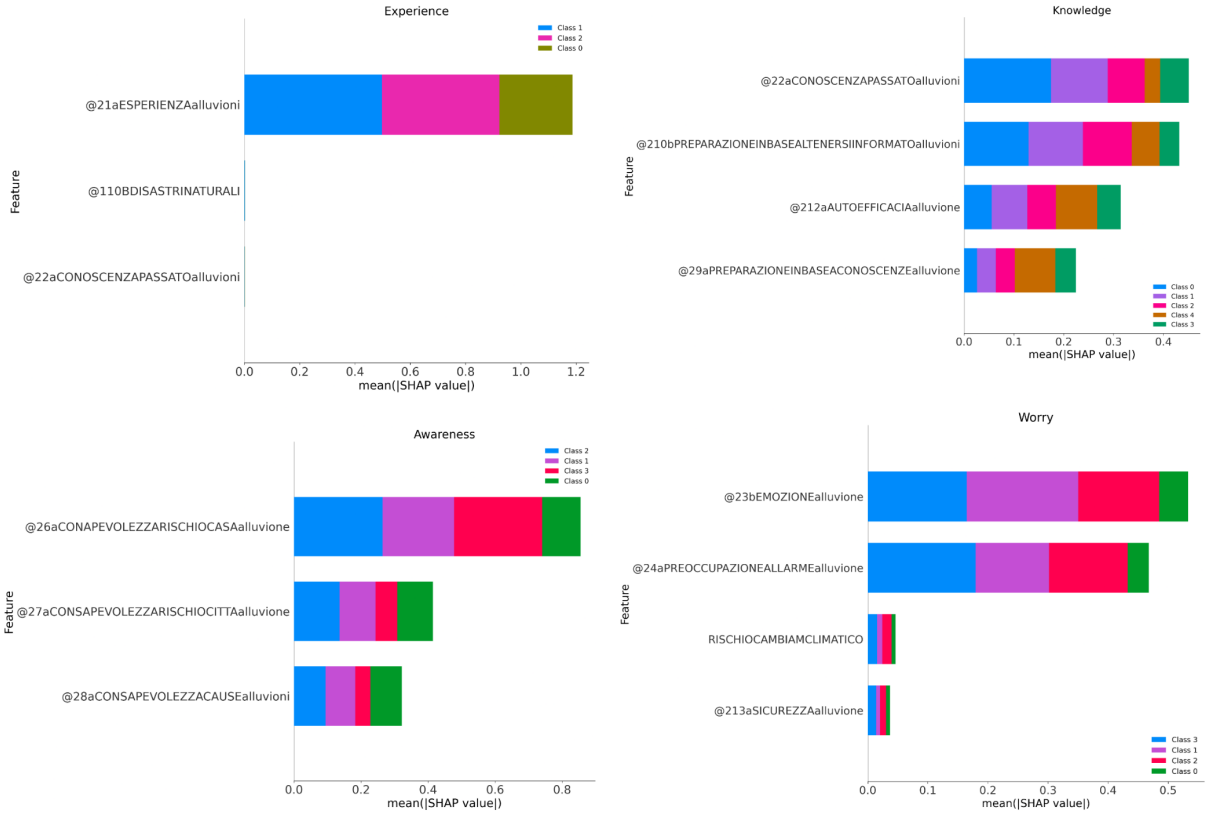


Fig. 20: Flood Risk feature importance

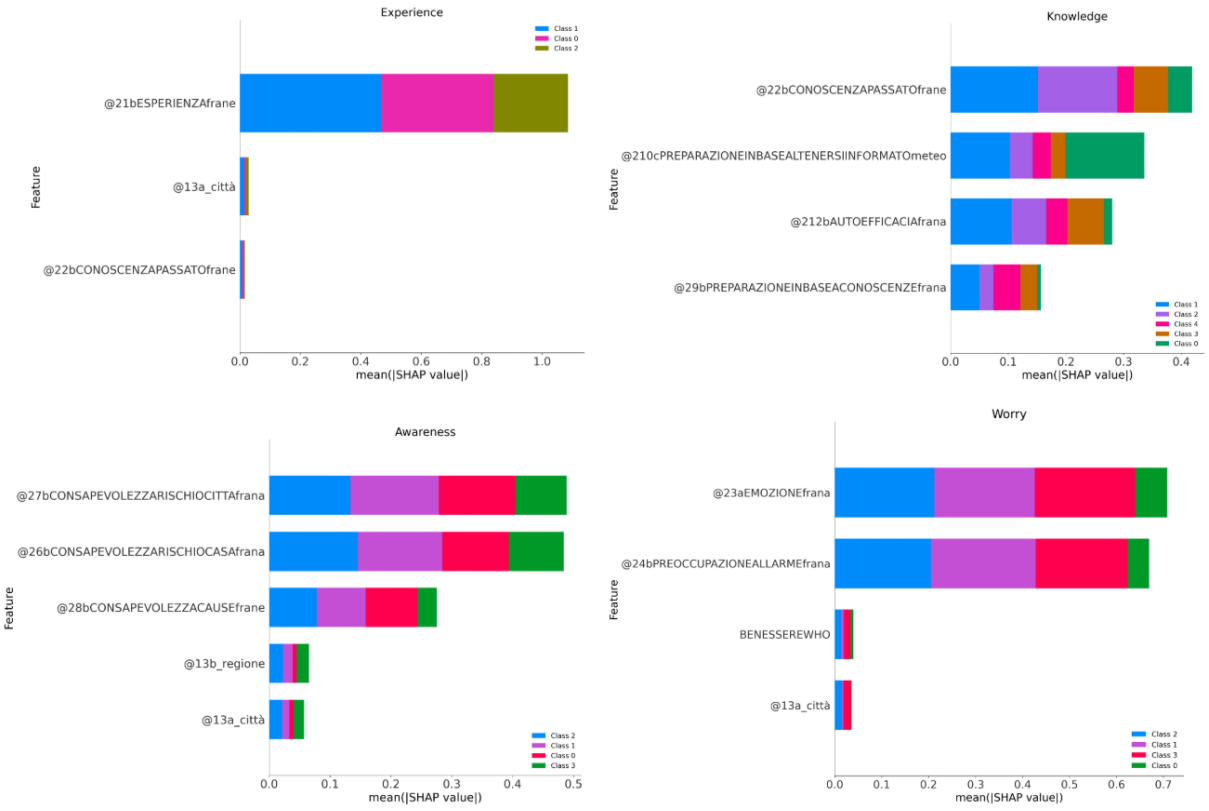


Fig. 21: Landslide Risk feature importance

Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union. Neither the European Union nor the granting authority can be held responsible for them.

The experimental results indicate that tree-based ensemble methods (Gradient Boosting and XGBoost) generally offer the most robust performance for perceptual targets, with Support Vector Machines proving highly competitive in specific tasks such as landslide awareness. The integration of SHAP validated the trustworthiness of the models, demonstrating that the predictions are driven by coherent and interpretable features rather than spurious correlations, providing transparent insights into the decision-making process and revealing that a compact subset of highly discriminative features is sufficient to drive accurate predictions. The analysis revealed that *Experience* is predicted by the direct history of past events, while *Knowledge* relies heavily on self-efficacy and proactive information-seeking habits. *Awareness* was found to be strongly tied to the perception of risk to one's own residence and city, whereas *Worry* is primarily driven by emotional reactions and the level of alarm in response to warnings.

These findings confirm that the developed models are reliable, interpretable, and effective for enhancing the risk assessment capabilities of the SAFE-LAND framework.

References

[1] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, Oct. 2001.

[2] J. H. Friedman, "Greedy function approximation: A gradient boosting machine.," *The Annals of Statistics*, vol. 29, no. 5, Oct. 2001.

[3] J. H. Friedman, "Stochastic gradient boosting," *Computational Statistics amp; Data Analysis*, vol. 38, no. 4, pp. 367–378, Feb. 2002.

[4] A. Guryanov, "Histogram-based algorithm for building gradient boosting ensembles of piecewise linear decision trees," in *Analysis of Images, Social Networks and Texts*. Springer International Publishing, 2019, pp. 39–50.

[5] T. Chen and C. Guestrin, "Xgboost: A scalable tree boosting system," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '16, ACM, Aug. 2016, pp. 785–794.

[6] C. Cortes and V. Vapnik, "Support-vector networks," *Machine learning*, vol. 20, no. 3, pp. 273–297, 1995.

[7] D. E. Rumelhart and J. L. McClelland, "Learning internal representations by error propagation," in *Parallel Distributed Processing: Explorations in the Microstructure of Cognition: Foundations*. 1987, pp. 318–362.

Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union. Neither the European Union nor the granting authority can be held responsible for them.