UNIVERSITÀ DI PISA

DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE

Master's Degree in Artificial Intelligence and Data Engineering

Data Mining and Machine Learning Project

IceSentinel: A Machine Learning Approach to Avalanche Danger Level Prediction

Author:

Emanuele Respino

Project GitHub Repository: https://github.com/Emanuele.rsp/IceSentinel

ACADEMIC YEAR 2024/2025

Contents

Chapter 1

Introduction

Avalanches remain one of the most deadly natural hazards in mountainous regions, claiming lives and causing significant damage each year. In Europe alone, the European Avalanche Warning Services (EAWS) report an average of about 100 lives lost per winter season despite advances in rescue techniques and safety equipment [4].

Accurate nowcasting of avalanche danger levels is critical not only for saving lives but also for maintaining the viability of alpine tourism and ensuring the safety of backcountry and resort operations. The Alpine region records approximately 375 million overnight stays each year, underlining the high stakes involved in hazard prediction for both local economies and visiting skiers [5]. To support public and commercial decision-making, EAWS forecasters issue daily avalanche danger levels - ranging from 1 (Low) to 5 (Very High), see Fig. 1.1 - based on standardized criteria including snowpack stability, spatial distribution of weak layers, and potential avalanche size [6].

In parallel with developments in numerical weather prediction, environmental forecasting has seen a rapid shift toward data-driven, machine-learning approaches. Within avalanche science, several studies have shown the promise of similar meth-

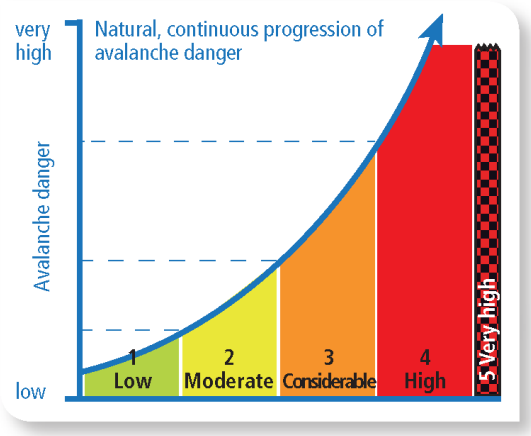


Figure 1.1: European Avalanche Danger Scale (Source: EAWS).

ods: Pérez-Guillén et al. [2] successfully trained random forest classifiers on over two decades of IMIS station data and SNOWPACK simulations to predict regional avalanche danger levels in the Swiss Alps, and data-driven models are increasingly being integrated into operational forecasting workflows across Europe [1].

1.1 Problem Statement

Avalanche danger nowcasting focuses on determining the current hazard level-on the standard five-point scale from Low (1) to Very High (5) at specific mountain sites in near real time. Rather than predicting tomorrow's conditions, our goal is to automate the classification of today's avalanche risk by learning directly from hourly observations and pre-computed SNOWPACK features. Traditional workflows rely on expert judgment, interpreting snow-layer stability through time-consuming physical simulations; this can introduce subjective bias, delay updates and limit spatial or temporal resolution. By contrast, a machine-learning framework can ingest diverse sensor measurements and snow-model outputs to deliver fast, repeatable and spatially detailed nowcasts without re-running full physics models.

1.2 Objectives

This project aims to build and evaluate a suite of machine-learning pipelines for avalanche-danger nowcasting at IMIS stations. By comparing Random Forest, XGBoost and SVM classifiers -each paired with different oversampling, scaling and dimensionality-reduction steps- we identify the best end-to-end workflow. After an initial zero-tuning screening to shortlist candidates, we carry out hyperparameter optimization (avoiding any temporal leakage), select the final model based on its statistical performance on held-out data and perform an addictional model optimization based on feature selection and re-tuning. Alongside this, we use feature-importance measures and SHAP values to reveal which variables drive the forecasts.

Chapter 2

Dataset

2.1 Data Sources

In our reference study, the original database comprised all hourly IMIS observations (half a million) and corresponding multi-layer *SNOWPACK* outputs, but was then filtered to retain only records with certified SLF avalanche danger levels (almost 30 thousands observations), discarding any entries lacking a verified rating as described in Pérez-Guillén et al. [2]. Over twenty winters of data - from 1997/98 through 2019/20 - were compiled from the Swiss Intercantonal Measurement and Information System (**IMIS**), a dense network of automatic weather stations deployed at potential avalanche starting zones [8], autonomously recording and transmitting hourly meteorological measurements to the SLF data server in Davos.

These raw observations feed a continuous, 1D, multi-layer *SNOWPACK* simulation chain, run automatically each winter to generate detailed snow-stratigraphy profiles alongside pre-processed meteorological fields. Input data are first quality-controlled via the *MeteoIO* library-filtering spurious values and gap-filling missing entries through temporal or spatial interpolation-then used to drive SNOWPACK's physics-based layer growth and metamorphism routines.

As targets, the dataset uses the daily regional avalanche danger levels published by the Swiss national warning service (SLF) at 17:00 LT, updated at 08:00 LT since 2013, each valid for the next 24 h and covering 130 warning regions ($\sim$200 km^2 each) across the Alps. Regions are assigned a level from 1 (*Low*) to 5 (*Very High*), with accompanying critical elevation bands and slope aspects. As described in Pérez-Guillén et al. [2], only those daily danger ratings for which the published bulletin ($D_{forecast}$) matched independent local nowcast assessments-or were corroborated by avalanche activity reports and prior verification studies-were included in the so-called D_{tidy} subset.

This fully pre-processed and filtered database (D_{tidy}) was made available via *EnviDat* [3] as a CSV file containing 29 296 records and 79 feature columns, each row representing a 24 h aggregation of measured and simulated variables aligned to the certified dry-snow danger level at the corresponding station and timestamp.

No further aggregation or alignment was necessary before proceeding to residual cleaning, feature pruning and model development.

2.2 Features Description

For a complete list of all the features and their definitions, please refer to the dataset documentation. The dataset contains 79 columns which can be grouped as follow:

Target Variable: Avalanche Danger Level. The classification target is the certified avalanche danger level, a *categorical variable* from 1 to 5 [6]:

1. **Low**. Snowpack is generally very stable. Small avalanches are only possible on extremely steep slopes or under unusually heavy loading (e.g. a large group of skiers).

2. **Moderate**. Stability is good in most places, but on some very steep slopes it is only moderate. Avalanches can be triggered by heavy loads, especially on known weak slopes, but spontaneous avalanches are not expected.

3. **Considerable**. Stability is moderate or poor on many steep slopes. Even a single skier or a moderate new-snow load can trigger avalanches, and medium-sized or large spontaneous releases are possible in some areas.

4. **High**. Stability is poor on most steep slopes. Avalanches are likely even under light loading, and multiple medium-sized or large spontaneous avalanches are expected.

5. **Very High**: The snowpack is generally unstable everywhere. Large avalanches are very likely on even moderate slopes, with many spontaneous releases expected without any additional loading.

Raw Meteorological Variables. These *continuous-valued* features are the direct hourly measurements from IMIS stations, aggregated over 24 h windows. They include air temperature, precipitation, wind speed and humidity. Each of those variables can directly influences snow-layer growth, melting and transport processes.

SNOWPACK-Derived Snowpack Features. Driven by the quality-controlled meteorological inputs, SNOWPACK produces detailed snow-cover profiles on a per-station, from which layer-specific indices (*continuous-valued* except 3 *binary* ones) were extracted. These indices capture critical metamorphic and mechanical properties of the snowpack that are not directly measurable at each time step.

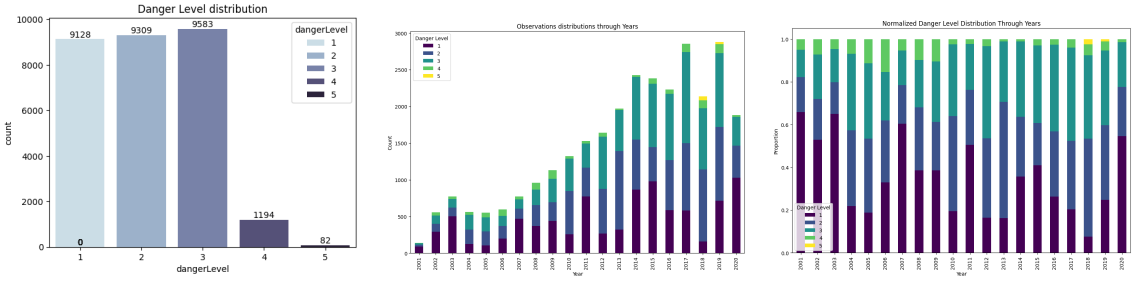
Temporal and Lagged Features. To embed short-term trends and memory effects, some aggregated and lagged versions of variables were included (up to one week earlier). These features can help the model recognize rapid warming or loading events-which often trigger avalanches-by providing explicit temporal context.

Derived Metrics. Also higher-level derived metrics were included such as the net long wave radiation, which summarize complex processes in a single value.

2.3 Exploratory Data Analysis (EDA)

Data structure. The analysis began by loading the D_{tidy} dataset (29 296 records, 79 columns), inspecting its structure via `df.info()`, `df.describe()` and `df.dtypes`, and quickly identifying a constant column (`zS5`) and three binary flags (`base_pwl`, `pwl_100_15`, and `pwl_100`) by counting unique values.

Target distribution. Numerical summaries revealed that over the full twenty-year span, the dataset exhibits a pronounced imbalance among danger levels (Fig. 2.1a): the low/mid-range classes (1-Low, 2-Moderate and 3-Considerable) each contribute on the order of nine thousand samples, whereas levels 4-5 together account for only about 1 300 observations; *Very High* (level 5) occurs in just 82 cases (2018-2019 only, Fig. 2.1b). Year by year, the volume of observations grows from a few hundred records per season in the early 2000s, when the IMIS network was small, to nearly three thousand by 2018-19. When we normalize by year (Fig. 2.1c), we see that early winters were dominated by Low ratings, mid-decade seasons shifted toward a more even mix of 2-Moderate and 3-Considerable days.



(a) Target variable imbalance.

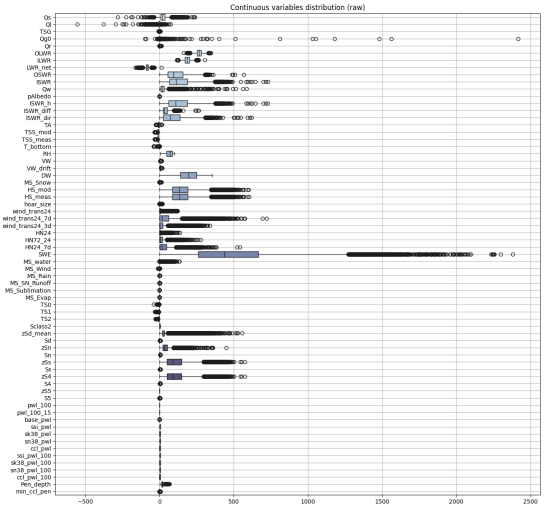
(b) Number of observation through the years.

(c) Danger level distribution over the years.

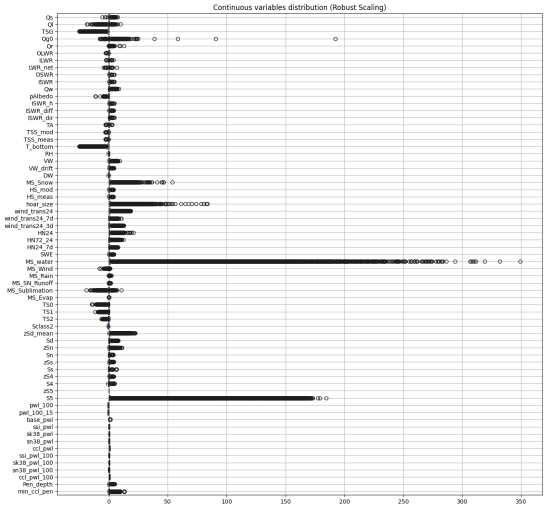
Figure 2.1: Target distribution.

Missing values. Looking at the missing values for each feature, five columns (`Qg_mean`, `mAlbedo_mean`, `MS_Soil_Runoff_mean`, `TS3_mean` and `TS4_mean`) contain no data at all. Overall, only 22 597 tuples (77.13% of the full dataset) have no missing values at all.

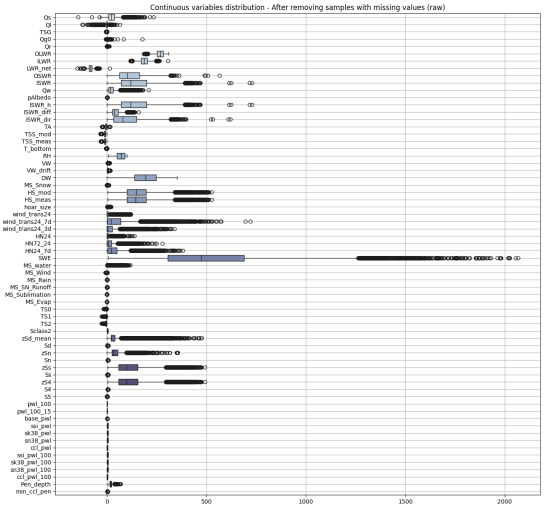
Features distribution. During our exploratory analysis of the continuous variables, box-plots of the raw data (Fig. 2.2a) immediately revealed different value ranges across features, and the presence of extreme outliers. For instance, the snow water equivalent (`SWE`) has a median of almost $500[kg/m^2]$ while most of the other features cluster near zero, with some instances registering extreme values on many attributes.



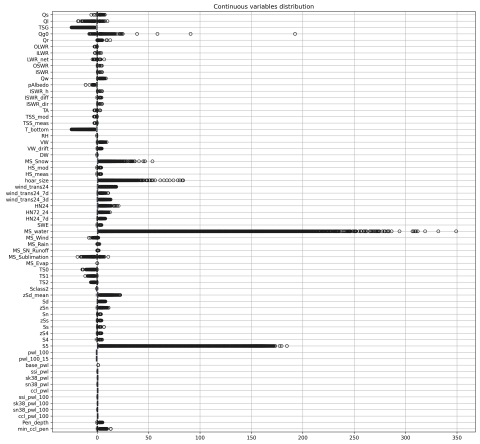
(a) Before Missing-value removal and Robust Scaling.



(b) Before Missing-value removal - After Robust Scaling.



(c) After Missing-value removal - Before Robust Scaling.



(d) After Missing-value removal and Robust Scaling.

Figure 2.2: Features distribution.

To address both the scale disparities and the undue influence of these extremes, we applied a *Robust Scaler*, which rescales each feature based on its inter-quartile range rather than its standard deviation. After transformation (Fig. 2.2b), most

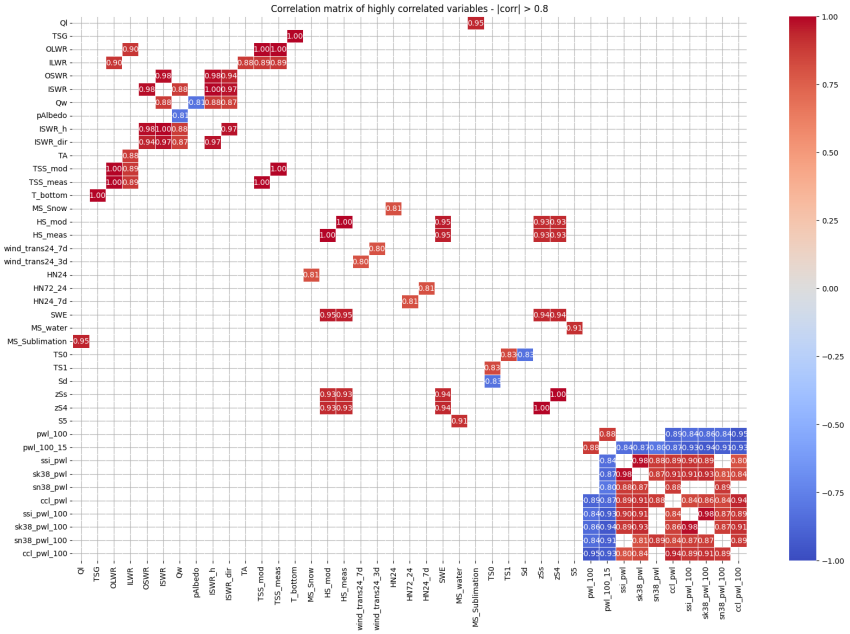


Figure 2.3: Correlation matrix.

variables appear much more homogeneously spread around zero, with extreme points still visible but less pronounced.

After scaling, the bulk of each feature's distribution contracts around zero, yet the true outliers remain visible-most notably in `Qg0`, `S5` and `MS_water`. These pronounced tails could reflect rare but physically meaningful snowpack states.

Correlation analysis and PCA. After dropping non-numeric identifiers, we computed the correlation matrix of the remaining features (Fig. 2.3) and applied an absolute correlation threshold of 0.80 to detect very strong linear relationships using a filtered heatmap: this allow to immediately show, despite the number of features, how many features are highly correlated. This can be due to the fact that some features are derived from other, or that they are measuring the same thing but in different ways.

We decided to use a **global PCA** to get a sense of the overall intrinsic dimensionality. Using the robust-scaled data, we fitted a standard PCA and plotted the cumulative explained variance as a function of the number of components. We observed that with StandardScaler, almost 30-35 components are required to capture 95% of the variance, whereas with RobustScaler, the first two components already exceed that mark. This global PCA confirmed that, even before isolating correlated groups, the dataset's effective dimensionality is far lower than the raw feature count.

With this insights, we extracted the groups of highly correlated variables, using a `NetworkX graph` where each node is a feature and an edge connects any two features whose absolute correlation exceeds 0.90 (Fig. 2.4). We then extracted the connected components of this graph to identify seven distinct groups of highly

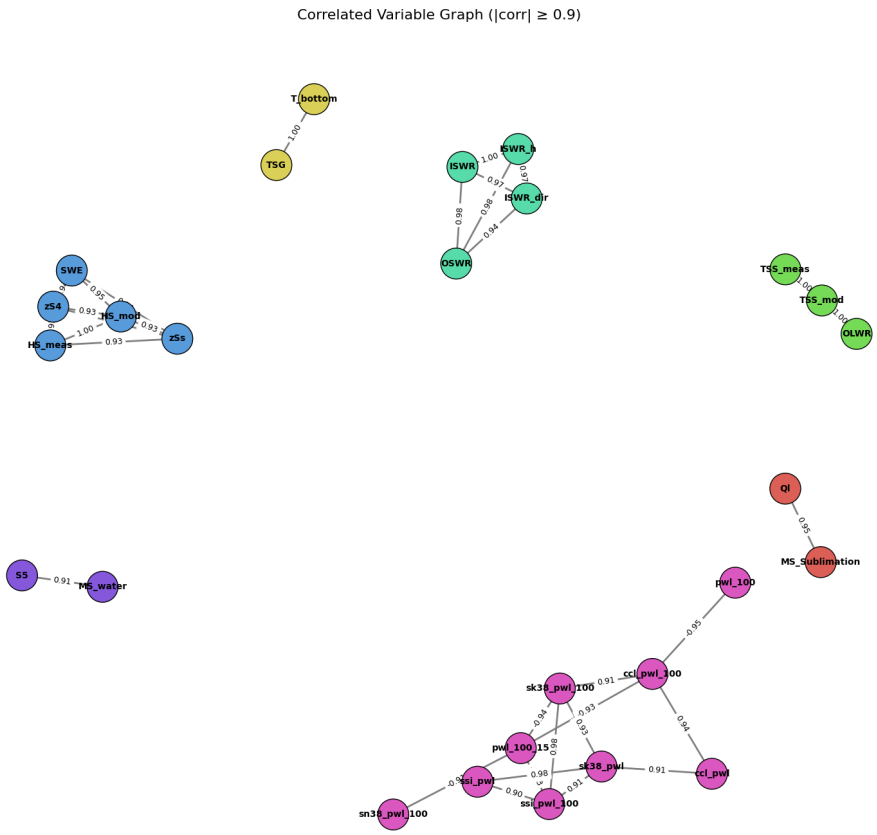


Figure 2.4: High correlated groups of variables.

correlated variables, making clear which features essentially carry duplicate or near-duplicate information-often because they measure the same physical process. We finally proceeded to group-wise PCA on each cluster of highly correlated variables, using the same 95% explained-variance criterion to replace each group with its top components. In every case, the first one to three principal components captured at least 95% of the group's variance.

2.4 Preprocessing

Starting from the D_{tidy} dataset (29 296 rows, 79 columns), we performed the following steps:

Feature pruning We removed the five columns that contained only missing values (`Qg_mean`, `mAlbedo_mean`, `MS_Soil_Runoff_mean`, `TS3_mean`, `TS4_mean`) and the constant column (`zS5`). Administrative or location identifiers (`station_code`, `sector_id`, `warnreg`, `elevation_station`) were dropped to avoid encoding site-specific biases. We also excluded `elevation_th`, which directly reflects the official bulletin and would leak target information. We retained only `datnum` to enforce

strictly chronological cross-validation splits.

Missing-value removal Next, rather than attempt further imputation, we removed every record with at least one missing entry. This decision was driven by the fact that meteorological inputs were already quality-controlled and interpolated [2] and that residual gaps likely indicate problematic measurements. This decision reduced the dataset from 29 296 to 22 597 rows (a 22.9% reduction) without materially altering the class balance of the target variable (see before/after distribution in Figg. 2.2). Although some extreme feature values were pruned, subsequent boxplots of the cleaned data show distributions that remain comparable to the originals once a Robust Scaler is applied.

Class merging The original danger level 5 (*Very High*) occurred in only 82 cases (0.3% of D_{tidy}), making reliable learning extremely difficult and contributing high labeling noise. To mitigate this, we merged levels 4 and 5 into a single *High* class (4). This increases top-class sample size, reduces rare-class instability, and simplifies the multi-class task-yielding more stable, generalizable models without aggressive resampling.

After preprocessing, the cleaned dataset comprises 22 601 complete records, 68 input features, the merged four-class target (1-Low, 2-Moderate, 3-Considerable, 4-High) and `datnum`. Label encoding (1-4 to 0-3) will be applied prior to model fitting to ensure compatibility with XGBoost and other classifiers.

2.5 Dataset Splitting

Our primary objective is to develop a reliable in-network nowcasting model for the Swiss Alpine IMIS stations, since all available data originate from this specific geographic and climatic context. We do not possess a representative sample of external stations to ensure true out-of-network generalization; attempting a leave-one-station-out or station-based group CV would yield folds with too few samples and results that are difficult to interpret. Consequently, we focus our validation on strict temporal rigor (expanding-window) within the known network.

Nested CV Considerations We opted not to implement a full nested cross-validation for two main reasons. First, the **computational cost** of nesting k-fold tuning loops inside k-fold validation loops-across dozens of pipeline variants and almost 22 000 samples would have been prohibitive. Second, in a time-series setting a standard nested CV cannot by itself prevent temporal leakage unless the outer folds also respect seasonal ordering, further shrinking training sets and destabilizing results. Our *outer hold-out + expanding-window CV* design (Fig. 2.5) thus provides a rigorous, leakage-free evaluation while remaining tractable.

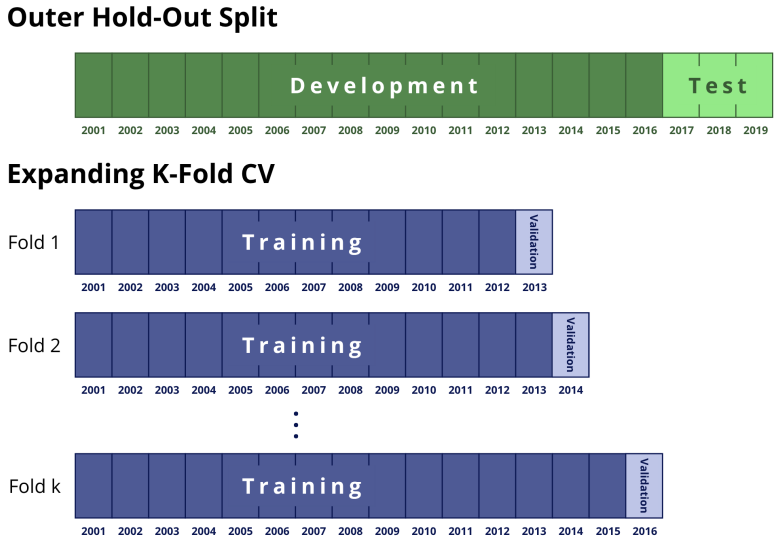


Figure 2.5: Splitting Criterion.

Season assignment First, we computed a `season ID` for each observation by assigning dates in November-March to the preceding winter year. This allowed us to treat each winter season as an indivisible block for time-aware splitting.

2.5.1 Outer hold-out (test) set

The last three seasons (2017, 2018, 2019) were reserved as a final out-of-time hold-out, representing approximately 25% of all observations. These data are excluded from all screening and tuning procedures to ensure the test performance reflects truly unseen future conditions, providing a realistic estimate of operational performance.

2.5.2 Inner cross-validation (training/validation)

The remaining seasons (2001-2016) form the development pool. Instead of a naïve random k-fold, we implemented an *expanding-window* cross-validation, where each fold trains on all seasons up to year Y and validates on season $Y + 1$. This strategy prevents any look-ahead bias and simulates sequential model updates. The helper function used is `get_expanding_splits(groups, train_seasons, n_folds)`, which yields a sequence of `(train_idx, val_idx)` pairs for a chosen number of folds.

By varying `n_folds` (5 folds for baseline screening and fine tuning, up to 15 for model selection), we balance statistical rigor and computational cost.

Use of datum The original timestamp column `datum` is retained solely to drive these chronological splits and is never provided as a model feature. This disciplined, time-aware splitting ensures no future information leaks into training or validation.

Chapter 3

Methodology

Evaluation Metric Throughout our model development and comparison, we adopt the macro-averaged F_1 score as the primary evaluation criterion. This choice aligns with prior data-driven avalanche forecasting studies, facilitating direct comparison of performance. More importantly, macro-F_1 treats each danger class equally, ensuring that the rare but critical *High* events carry the same weight as more common mid-range days. In practical terms, under-predicting a high-risk day can have life-threatening consequences, whereas over-predicting mainly incurs economic costs; macro-F_1 inherently balances these asymmetric stakes by penalizing false negatives in the highest-risk class as strongly as errors elsewhere.

3.1 Model Training

3.1.1 Pipeline Design

To systematically explore the space of possible workflows, each model was constructed as a four-step pipeline:

1. **Sampling** To address class imbalance, we evaluated three strategies:
    - *None* `[NoS]` (no sampling),
    - *RandomOverSampler* `[ROS]`, which replicates existing minority-class samples by random sampling with replacement until class frequencies are balanced,
    - *SMOTE* (Synthetic Minority Over-Sampling Technique), which generates new synthetic minority samples by interpolating between minority-class neighbors in feature space.
2. **Scaling** Given the wide feature ranges and presence of outliers, we compared:
    - *None* `[NoS]` (no scaling),

- *StandardScaler* `[SS]`, which centers on the mean and scales by the standard deviation,
- *RobustScaler* `[RS]`, centering on the median and scaling by the interquartile range.

3. **Dimensionality Reduction** To reduce redundancy and noise, three approaches were tested:

   - *None* `[NoDR]` (No dimensionality reduction)
   - *Global PCA* `[PCA]` (Principal Component Analysis),
   - *Group-wise PCA* `[L-PCA]`, applying PCA separately to each highly correlated feature cluster previously found.

   For both PCA methods, we set 0.95 as cumulative variance threshold for Principal Components selection.

4. **Classification** The processed feature vectors were then passed to one of three classifiers:

   - *Random Forest* `[RF]`, the most widely used tree-based ensemble in avalanche hazard prediction literature due to its robustness to noise, interpretability via feature importance measures, and ability to handle heterogeneous, nonlinear relationships [2],
   - *XGBoost* `[XGB]`, a gradient-boosted decision-tree algorithm that often achieves state-of-the-art accuracy through built-in regularization and efficient handling of sparse data
   - *Support Vector Machine* `[SVM]`, a kernel-based method well-suited to high-dimen-sional spaces and offering a contrasting decision boundary paradigm that can capture complex patterns with a relatively small number of support vectors.

   We selected three complementary classifiers based on their strong performance in related applications.

We also added two reference variants from Pérez-Guillén et al. [2] (the base RF1 and RF optimized). In total, avoiding the inclusion of useless pipelines (e.g. PCA method without a previous scaling step), we generated 56 pipelines.

3.1.2 Baseline screening

Before investing in expensive hyperparameter searches, we first perform a quick *sanity check* on every candidate pipeline using its default settings. This initial screening serves two purposes: identify and discards any workflows that fail to produce reasonable performance out-of-the-box, and focus our compute budget on the most promising few.

Concretely, we run a 5-fold, time-series-aware cross-validation (using the expanding-window splits on the 2001–2016 development pool) and record the macro-averaged F_1 score for each pipeline.

We then rank all the pipelines by their mean macro-F_1 score. To balance performance and methodological diversity, we first identify the top 20-ranked pipelines and from these, we manually select nine of the best performers ensuring a certain degree of representation across all sampling, scaling, reduction and classifier strategies, plus the reference model from Pérez-Guillén et al. [2].

This shortlist of **8 pipelines** (Tab. 3.1) -plus the reference model- sets the stage for targeted hyperparameter optimization.

Table 3.1: F_1 Mean Score for selected pipelines after baseline screening.

Pipeline	**Mean F_1**	**Std. dev.**
RF_ROS_NoS_L-PCA	0.5935	0.0373
RF_ROS_RS_NoDR	0.5934	0.0384
RF_ROS_SS_NoDR	0.5931	0.0382
XGB_NoR_RS_NoDR	0.5895	0.0384
XGB_NoR_NoS_L-PCA	0.5889	0.0368
RF_SMOTE_NoS_L-PCA	0.5867	0.0333
XGB_ROS_RS_NoDR	0.5864	0.0438
SVM_NoR_SS_PCA99	0.5843	0.0387
Paper_Optimized	0.5805	0.0435
RF_NoR_NoS_NoDR	0.5724	0.3556

3.1.3 Hyperparameter Optimization

Each pipeline was paired with a custom hyperparameter grid at construction time, tailored to its steps (e.g. number of trees and maximum depth for Random Forest).

We performed a **randomized search** over each grid with $N_{\text{iter}} = 100$ iterations, using the same 5-fold expanding-window splits on the 2001–2016 development pool. This yields approximately

$$9 \times 100 \times 5 \;\approx\; 4\,500$$

inner-CV fits, an order of magnitude lower than a full grid search, yet sufficient to explore the regions of each hyperparameter space.

Baseline-vs-tuned sanity check. For each pipeline, we compared the best-CV score from random search against its original default-parameter score from the baseline screening. If tuning did not improve performance (occasionally when random draws are suboptimal) we reverted to the default hyperparameters. This safety-net prevents degradation due to unlucky search samples. In this phase, a rollback to

default parameters was made only for `RF_SMOTE_NoS_L-PCA`. Results are shown in Tab. 3.2.

Table 3.2: Best CV Score e Used Default for selected pipelines.

Pipeline	Best CV Score	Used Default
RF_ROS_NoS_L-PCA	0.598049	False
RF_ROS_RS_NoDR	0.596792	False
RF_ROS_SS_NoDR	0.596768	False
RF_SMOTE_NoS_L-PCA	0.586731	True
RF_NoR_NoS_NoDR	0.575359	False
XGB_NoR_RS_NoDR	0.597877	False
XGB_NoR_NoS_L-PCA	0.600498	False
XGB_ROS_RS_NoDR	0.597030	False
SVM_NoR_SS_PCA99	0.591932	False
Paper_Optimized	0.589039	False

3.2 Model Selection

After hyperparameter tuning, we compared the nine candidate pipelines (plus the reference *Paper_ Optimized*) via a statistical comparison and then, among the statistically best ones, we analyzed their generalization capacity using the hold-out (test) set for selecting a final model.

Statistical Comparison To obtain sufficient replicates for non-parametric comparison, we re-ran an expanding-window CV with $k = 15$ folds on the 2001–2016 development pool. We then applied the Friedman test (χ^2, $\alpha = 0.05$) to the per-fold macro-F_1 scores, confirming a significant difference among models. A subsequent Nemenyi post-hoc analysis isolated the top seven performers whose pairwise differences were not statistically significant.

Model evaluation on Hold-out (test) set The seven finalist pipelines were retrained on the full 2001–2016 data and evaluated on the out-of-time hold-out set (2017–2019). We assessed performance via **Normalized confusion matrices**, **ROC curves** and corresponding AUC values for each class (Fig. 3.1), in addition to the metrics shown in Tab. 3.3.

Final model choice We chose `XGB_ROS_RS_NoDR` as our primary model because it, alongside the reference model, achieved over 50% recall on the rare, high-risk class 4 in the hold-out confusion matrix, but unlike the paper's original configuration, our

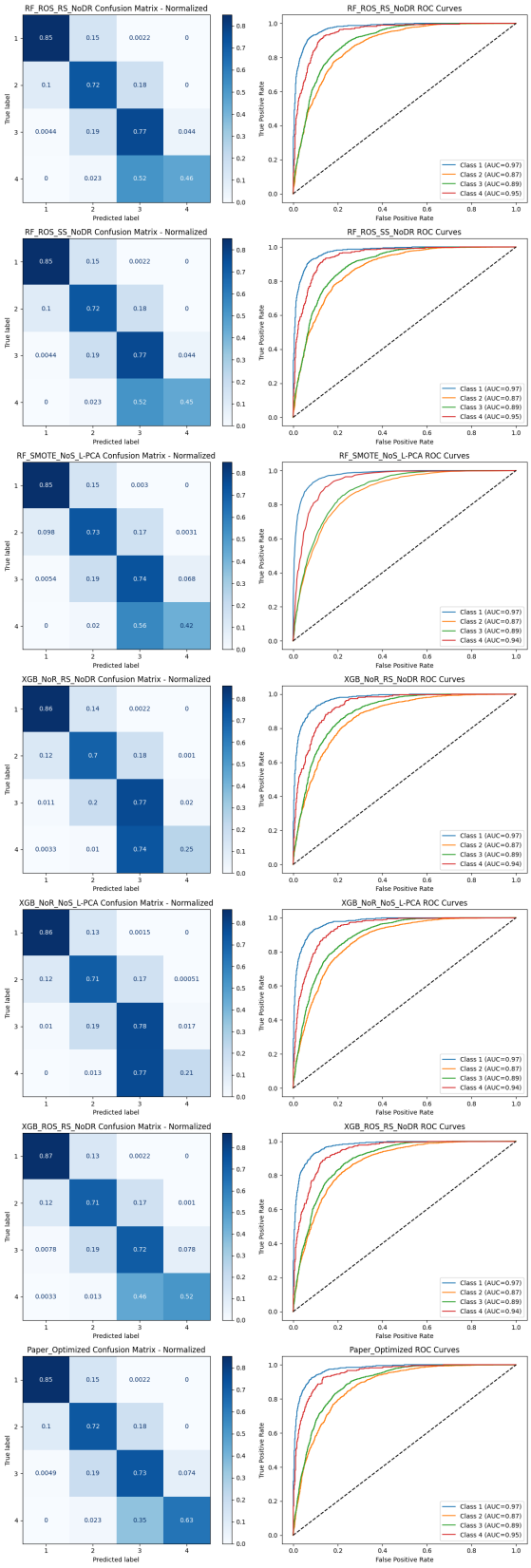


Figure 3.1: Evaluation of selected models on hold-out (test) set.

Table 3.3: Pipeline evaluation metrics on hold-out (test) set.

Pipeline	Accuracy	F_1-macro	Balanced Acc.	MCC
RF_ROS_RS_NoDR	0.754209	**0.710624**	**0.698313**	0.641582
RF_ROS_SS_NoDR	**0.754563**	0.710500	0.697977	**0.642067**
RF_SMOTE_NoS_L-PCA	0.742867	0.686936	0.682776	0.626233
XGB_NoR_RS_NoDR	0.739677	0.661388	0.645076	0.618530
XGB_NoR_NoS_L-PCA	0.745880	0.657204	0.641983	0.627126
XGB_ROS_RS_NoDR	0.742158	0.700881	0.706312	0.627547
Paper_Optimized	0.748361	0.722989	0.731236	0.636519

`XGB_ROS_RS_NoDR` parameters were selected under a fully leakage-free, expanding-window scheme, ensuring unbiased performance estimates. Although two of the RF-based pipelines (`RF_ROS_RS_NoDR` and `RF_ROS_SS_NoDR`) showed marginally higher overall accuracy, F_1-macro and MCC on the test set, we prioritized `XGB_ROS_RS_NoDR` because it delivered the strongest ability to identify class 4 under a rigorously leakage-free validation. To preserve flexibility, we also retain the `RF_ROS_SS_NoDR` and `Paper_Optimized` models.

Finally, the selected pipelines were retrained on the complete dataset (2001–2019) to maximize available training information before saving them.

3.3 Model Optimization

For the selected model, we tried to perform the procedure described in Pérez-Guillén et al. [2] for model optimization.

Removing Less-Important, Highly Correlated Features First, feature importance from the tuned pipeline are used to rank all inputs. We then drop any lower-ranked feature whose absolute Pearson correlation with a higher-ranked partner exceeds 0.90, ensuring that only the most informative variable remains in each correlated cluster.

Determining the Minimal Feature Subset Size We evaluate progressively larger prefixes of the ranked feature list (e.g. $k = 10, 20, 30 \ldots$) via 5-fold expanding-window CV, recording macro-F_1. We choose the smallest k for which performance is statistically indistinguishable (Wilcoxon, $\alpha = 0.05$) from the full set.

Recursive Feature Elimination (RFE) and Pipeline Modification Having fixed k, we perform RFE inside the existing pipeline to select exactly k features before any other transformations: we clone the tuned pipeline and replace its classifier with an RFE wrapper that selects k features, and then we fit `opt_model` on

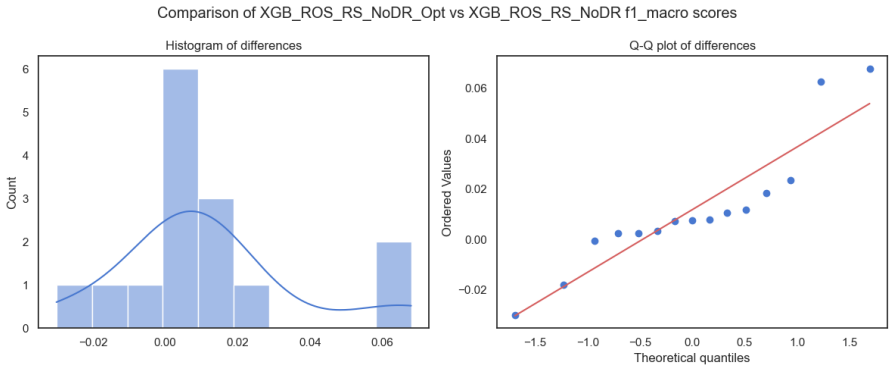


Figure 3.2: Comparison of final models.

the training data to extract the final selected feature names. Finally, we prepend a `ColumnTransformer` to the pipeline to permanently drop unselected features.

This ensures that every CV fold and subsequent retraining uses only the k most critical and non-redundant features.

Re-tuning on the Reduced Feature Set We adjust the hyperparameter search grids to focus around the previously optimized values and run a randomized search with `N_ITER=100` iterations over the reduced-feature pipeline, again using the 5-fold expanding-window splits.

Statistical Comparison of Original vs Optimized Both the full- and reduced-feature pipelines are evaluated via a 15-fold expanding-window CV. We collect per-fold macro-F_1 scores and apply a Wilcoxon signed-rank test (differences are not Gaussian as show in Fig. 3.2, confirmed by T-Test), setting a confidence level $\alpha = 0.05$ to confirm that the reduced pipeline is statistically non-inferior (or superior) to the original.

Final Hold-Out Evaluation The optimized pipeline (`XGB_ROS_RS_NoDR_Opt`) is retrained on the full 2001–2016 data and evaluated on the 2017–2019 hold-out, reporting absolute and normalized confusion matrices, and ROC, AUC and PR curve for each class (Fig. 3.3).

3.4 Second-Stage Study for Discriminating Class 5

To attempt recovery of the rare danger level 5 (which was merged into level 4 in the first-stage classifier), we evaluated several postprocessing strategies. Each method was applied exclusively to test samples whose true labels were 4 or 5, and was assessed using precision, recall, F_1, and AUPR.

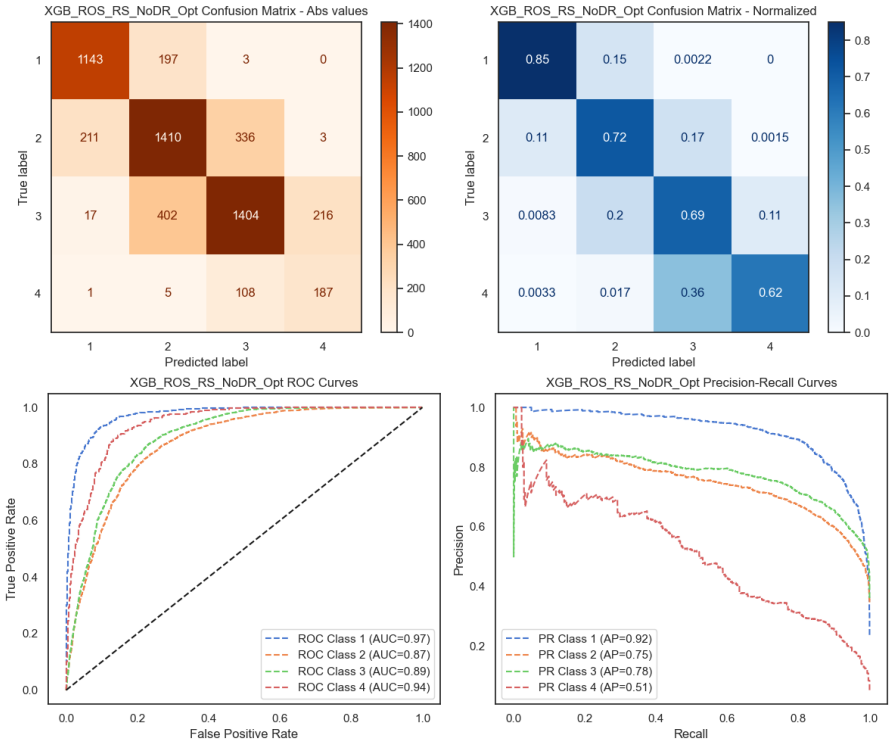


Figure 3.3: Evaluation of optimized model.

3.4.1 Baseline (First-Stage Only)

We first apply the pretrained first-stage model, trained on classes $\{1,2,3,4 = 4 \cup 5\}$, directly to the test set. We generate predictions for all test samples and then focus on those whose actual label is 4 or 5, computing a confusion matrix and classification report. Observing how many true-5 samples end up classified as 4 versus how many are misclassified into classes 1–3 provides a clear numeric baseline, how many true examples of class 5 the first stage manages to at least classify as 4 (and therefore can move on to the second stage) compared to how many end up directly in 1, 2 or 3 (and are lost).

3.4.2 Margin Analysis

Next, we examine the first-stage model's probability outputs to see if the difference between the confidence for class 4 and the confidence for class 3 can distinguish true-5 events from true-4 events. We extract each test sample's probabilities for class 4 and class 3, compute their difference (*margin*), and focus on samples truly labeled 4 or 5. Treating true 5 as positive, we calculate AUPR using the margin to assess how well it ranks 5s above 4s. We then find the single margin threshold that maximizes F_1, checking if any univariate cutoff can recover real 5s without too many false positives.

Logistic Meta-Classifier on $\{p_4, p_4 - p_3\}$ We fit a small logistic-regression model on the stage probability for class 4 and the difference between class 4 and class 3 probabilities, using only training examples labeled 4 or 5 (coded as 0 or 1, respectively). At test time, the same two values are extracted for samples truly belonging to 4 or 5, and the model's binary output is remapped to 4 or 5. Any improvement over the simple margin threshold would show that combining both features gives slightly better separation.

3.4.3 IsolationForest Outlier Detection

We train an IsolationForest only on true-4 samples (setting its contamination rate to the fraction of class-5 in training) and then score test samples whose true label is 4 or 5. Any test point with a negative anomaly score is labeled 5; otherwise, it is labeled 4.

Chapter 4

Model explainability

Modern ensemble methods such as Random Forest and XGBoost register good performances at forecasting avalanche danger levels but often operate as *black boxes*, obscuring the reasoning behind each prediction. In safety-critical contexts, understanding model behavior is essential: it builds trust among forecasters, aids debugging and feature refinement, and ensures accountability when issuing high-risk alerts.

Among available explanation tools, we focus on SHAP (SHapley Additive exPlanations) for its solid theoretical foundation in cooperative game theory, its model-agnostic applicability, and its ability to generate both global and local interpretability.

In our explainability analysis, SHAP values are computed with respect to the model's predicted class probabilities rather than the discrete class labels. This choice allows us to capture the influence of each feature on the likelihood of every danger level, providing a richer and more informative interpretation of the model's decision process. The explanations presented here are calculated specifically for observations from the most recent winter season (2019-2020), ensuring insights are relevant to the latest conditions on a reasonable sized set (1 691 samples).

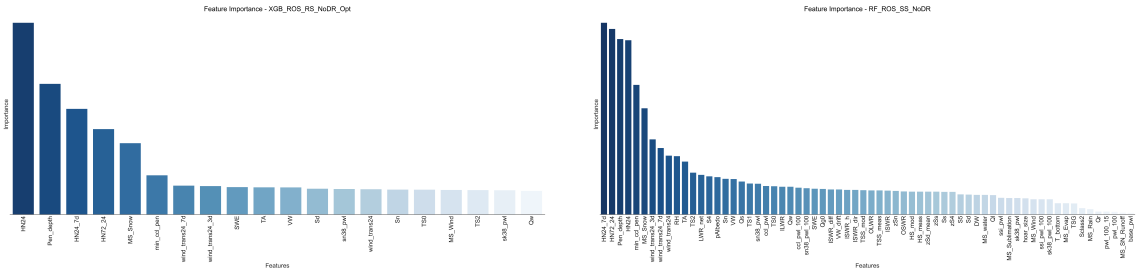
4.1 Global Explainability

4.1.1 Traditional Feature Importance

To better understand the drivers behind our predictive models, we compare the feature importance rankings produced by the Random Forest and XGBoost pipelines. This comparison helps reveal both common influential variables and algorithm-specific nuances, offering insights into the physical factors most relevant for avalanche danger nowcasting.

In both models (Fig. 4.1), the top six features stand out above the rest in terms of importance. These are `HN24_7d`, `HN24`, `HN72_24`, `MS_Snow`, `Pen_depth`, and `min_ccl_pen`. Variables related to wind and temperature follow with slightly lower

importance.



(a) XGB_ROS_RS_NoDR_Opt (b) RF_ROS_SS_NoDR

Figure 4.1: Feature importance.

Among the top features, the `HN` indicators and `Pen_depth` dominate. The Random Forest model places slightly more emphasis on `HN24_7d` and `HN72_24`, while the XGBoost model favors `HN24` and `Pen_depth`. Despite minor differences in ranking, these features maintain similar and significant influence in both models.

This consistency highlights that recent snowfall amounts and the penetration depth, among other physical snowpack characteristics are key predictors of avalanche danger, consistent with domain knowledge.

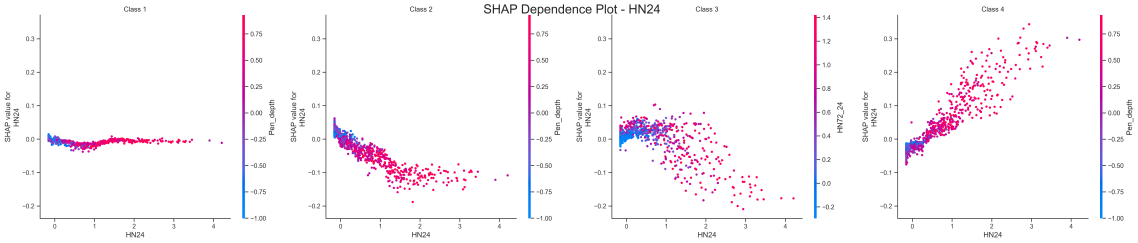
4.1.2 SHAP Dependence plot

We can enter deeper into details using SHAP Dependence plot (Fig. 4.2), which allow to understand better the relationship between features and classes. In our analysis we focus on the 3 main features governing the models: `HN24`, `HN72_24` and `Pen_depth`:

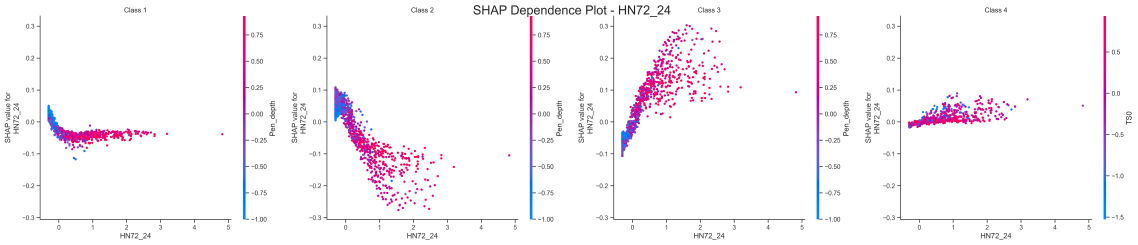
- **`HN72_24`** primarily related to `Pen_depth` but also `TS0`, it mainly determines moderate/considerable risk levels, increasing the overall danger.
- **`HN24`** plays a fundamental role in determining the *high* risk level, highlighting the importance of recent snowfalls in avalanche hazard.
- **`Pen_depth`** influences significantly all the classes, in particular 1 and 2: lower values determines lower risk, while higher ones push increase the avalanche activity.

4.1.3 SHAP Summary Plot

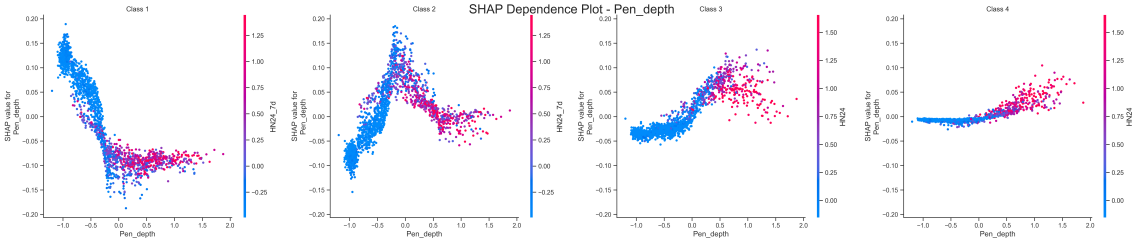
Now, we focus on the SHAP summary plot for the optimized XGBoost model (`XGB_ROS_RS_NoDR_Opt`), which ranks features by their average absolute impact on the model output across all samples, specifically entering into the details of how features influences each hazard class.



(a) Feature `HN24`.



(b) Feature `HN72_24`.



(c) Feature `Pen_depth`.

Figure 4.2: Dependence plots for the 3 most important features

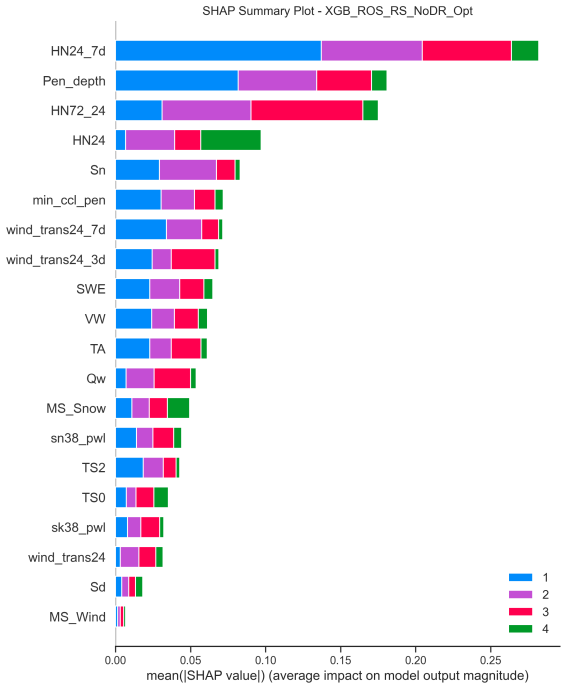


Figure 4.3: SHAP Summary Bar plot for XGB_ROS_RS_NoDR_Opt.

As shown in Fig. 4.3, the SHAP summary bar plot reveals that the driving features differ somewhat by avalanche danger class. Classes 1 (*Low*) and 2 (*Moderate*) are primarily influenced by the 7-day cumulative snowfall (`HN24_7d`) and the snowpack penetration depth (`Pen_depth`), reflecting how recent snow accumulation and snowpack stability contribute to these moderate risk levels. Class 3 (*Considerable*) is predominantly driven by the 3-day snowfall total (`HN72_24`), indicating higher impact of recent window there. Finally, the highest risk category 4 (*High*) is mostly associated with the 24-hour snowfall (`HN24`), suggesting that very recent snowfall spikes strongly influence the prediction of severe danger.

Class 1 - Low Risk The SHAP summary plot for Class 1 (Fig. 4.4a) reveals how feature values affect the likelihood of this class. For instance, higher values of `HN24_7d` (7-day snowfall total) tend to decrease the likelihood of Class 1, pushing predictions away from low risk towards higher danger levels. Similarly, increased `Pen_depth` reduces the chance of a low-risk classification, reflecting deeper snowpack penetration as a sign of instability. Conversely, lower wind transport over the last 7 days (`wind_trans24_7d`) supports a higher probability of Class 1, indicating stable, calm snow conditions. Other features such as `Sn` (natural stability index) and `min_ccl_pen` (minimum critical cut length at a deeper snowpack layer) positively contribute to the prediction of Class 1 (Low risk) with higher values, suggesting a stable snowpack structure.

Class 2 - Moderate Risk Notably (Fig. 4.4b), lower values of `HN24_7d` tend to increase the probability of Class 2, while high values of `HN72_24` have the opposite effect, reflecting that a class 2 has an higher probability with recent snowfalls that are not concentrated in the last 3 days. The penetration depth (`Pen_depth`) shows a more balanced effect, with moderate values engraving the most the prediction toward a Moderate Risk.

Class 3 - Considerable Risk The SHAP summary plot for Class 3 (Fig. 4.4c) highlights that this intermediate-high danger level is most strongly influenced by cumulative snowfall features such as `HN72_24` (3-day total) and `HN24_7d` (7-day total), which push the prediction towards this class when high. The snowpack penetration depth (`Pen_depth`) also plays an important role, with higher values increasing the likelihood of Class 3. The 24-hour snowfall (`HN24`) exhibits a mixed effect, with high values influencing the likelihood of class 3 in different ways, but mainly decreasing it in favor of an higher risk.

Class 4 - High Risk As we can observe in Fig. 4.4d, the most influential features are recent snowfall and snowpack properties. Specifically, the 24-hour snowfall (`HN24`) mainly but also the 7-day cumulative snowfall (`HN24_7d`) have strong positive effects on increasing the predicted risk to this highest level. The solid precipitation rate (`MS_Snow`) and the penetration depth (`Pen_depth`) further contribute, with higher values pushing the model towards class 4. 3-day cumulative snowfall (`HN72_24`) also plays a role but with less impact compared to the shorter-term snowfall metrics.

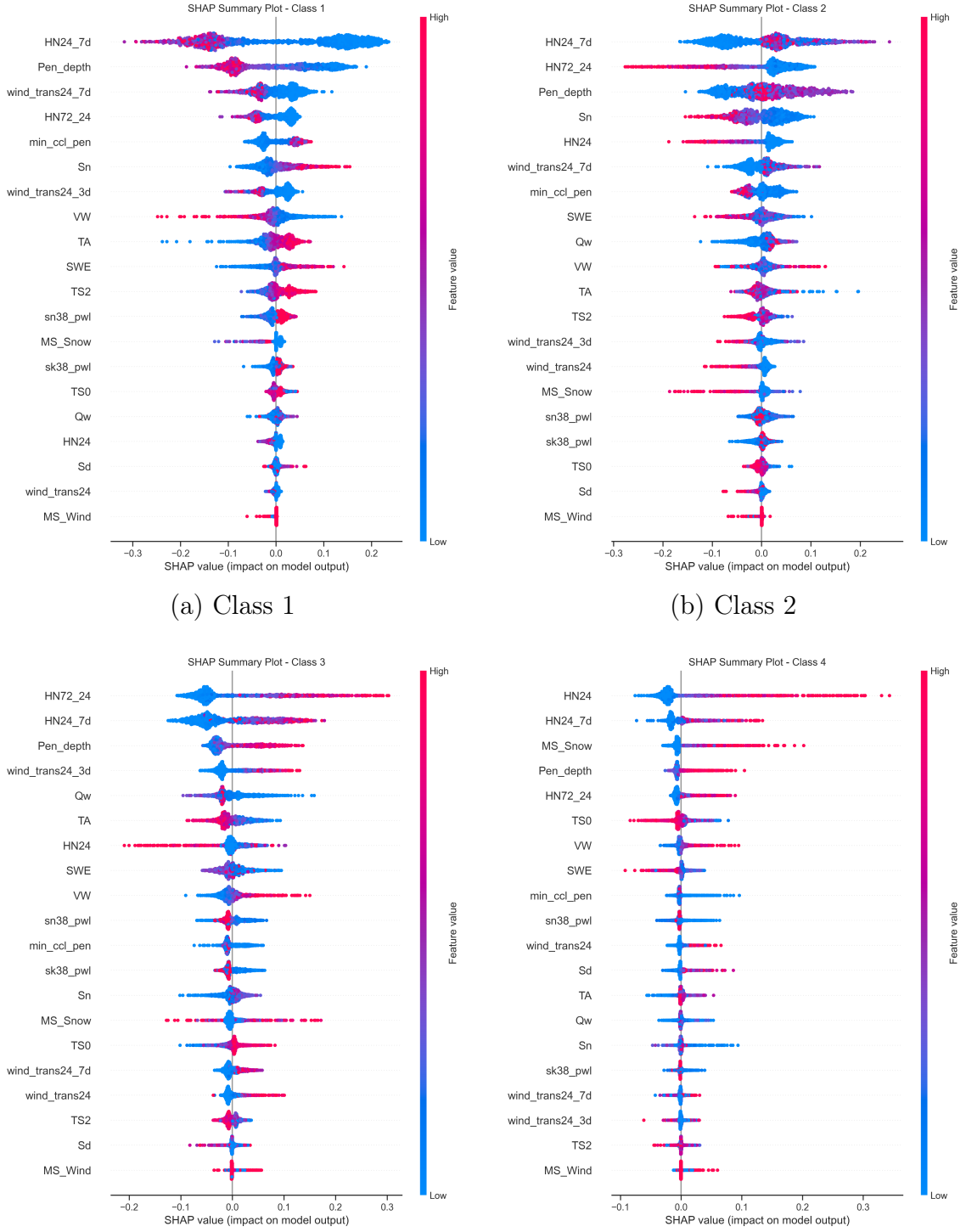
4.2 Local Explainability

Understanding the reasoning behind individual model predictions is crucial in avalanche risk forecasting, where single cases can have significant safety implications. Local explainability methods allow us to dissect and visualize how specific feature values contribute to the model's decision for each observation, providing actionable insights and building user trust.

Now, we present a concrete example of local explainability using a waterfall plot. We selected a sample from Class 3 because it is relatively common and represents a meaningful (*Considerable*) risk level, making it particularly interesting for analysis.

A practical example of local explanation As we can see in Fig. 4.5, the waterfall plots illustrate the contribution of each feature to the prediction score for the selected sample across all classes. Each plot shows how features either increase or decrease the likelihood of the sample belonging to a particular class compared to the average prediction.

We observe that a high value of `HN24_7d` decreases the likelihood of the sample being classified as *Low* risk, while increasing the probabilities for higher danger



(a) Class 1

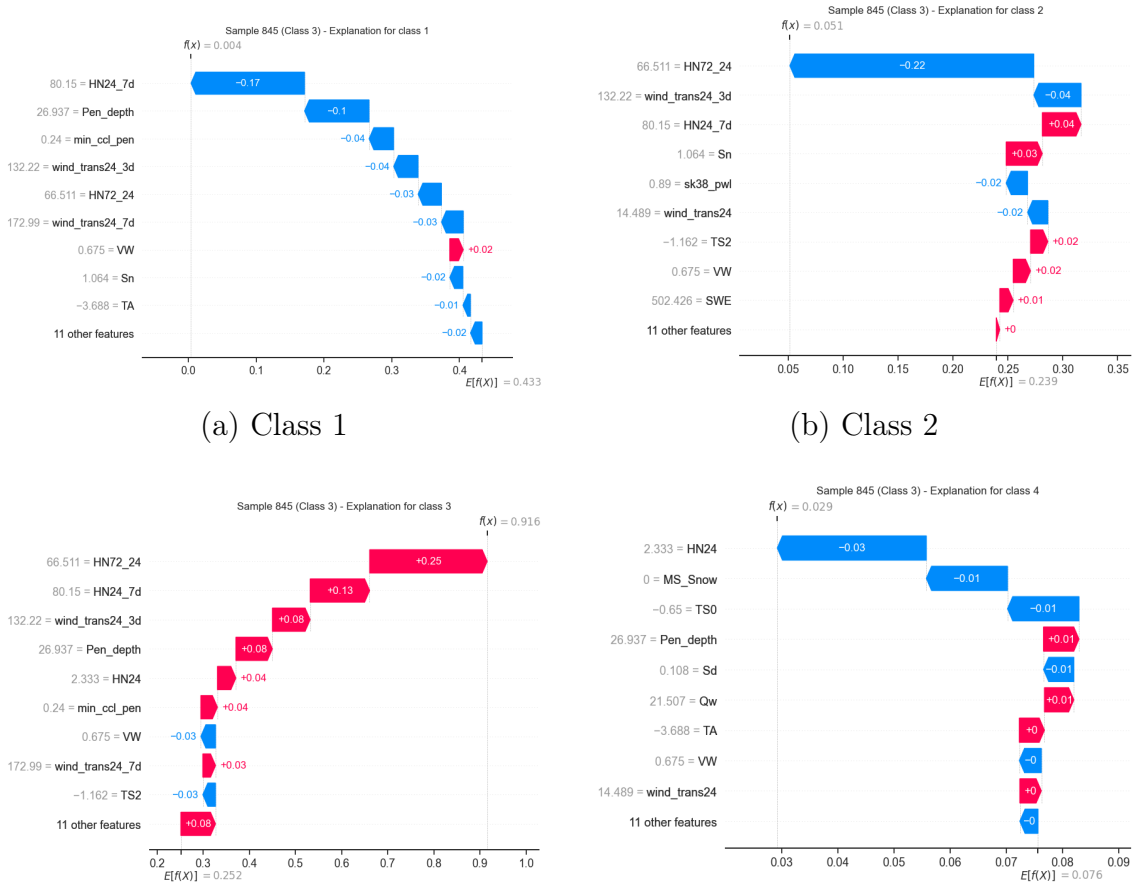
(b) Class 2

(c) Class 3

(d) Class 4

Figure 4.4: SHAP summaries for each class.

levels. Similarly, the high value of more recent snowfall (`HN72_24`) strongly reduces the probability of a *Moderate* risk, favoring instead a *Considerable* risk. Finally, the *High* danger level is effectively excluded due to the low snowfall in the last day (`HN24`). This pattern, combined with other positive but less dominant factors such as penetration depth and wind, leads the model to assign this sample to Class 3, corresponding to a *Considerable* avalanche danger level.



(a) Class 1 (b) Class 2

(c) Class 3 (d) Class 4

Figure 4.5: SHAP waterfall explanation for a class 3 sample.

This detailed feature-level explanation highlights the complex interactions the model uses to differentiate among the danger levels for this particular example.

Chapter 5

Graphical User Interface

The Graphical User Interface (GUI) was developed to make the avalanche danger level prediction model easily accessible for users. The primary goal is to allow users to load data, visualize predictions, and gain insights into the model's reasoning behind its decisions. By making the predictive model more transparent and interactive, the GUI enhances trust in the system and helps users make informed decisions about avalanche risk management.

5.1 Architecture and User Flow

The GUI was built using **Streamlit**, a popular Python framework for creating interactive web applications. Streamlit allows for rapid prototyping and an intuitive user interface, making it ideal for our project.

5.1.1 Backend

The system is not a server application, however it does include a minimal backend implemented within the same script. This backend handles the loading of pre-trained models, using `joblib` to load the saved models for inference. The avaliable models are as follows:

- `Paper_Optimized`: The model optimized in the original paper.
- `RF_ROS_SS_NoDR`: The Random Forest model developed during this project.
- `XGB_ROS_RS_NoDR`: The base XGBoost model selected for deployment.
- `XGB_ROS_RS_NoDR_Opt`: The optimized version of the XGBoost model.

Additionally, SHAP values for the most recent winter season are pre-calculated and stored, ensuring that users can access global explanations without recalculating them each time.

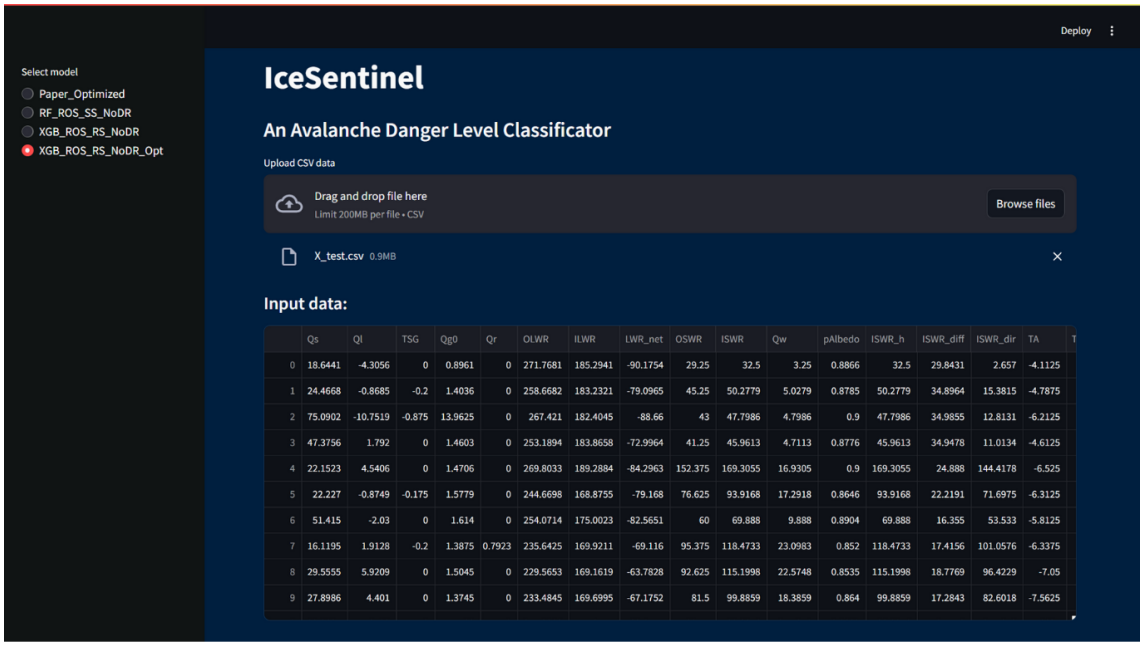


Figure 5.1: Users can select a model and upload data.

5.1.2 Frontend

The GUI provides a set of key features to the user, enhancing usability and transparency in avalanche risk prediction in real time, while state persistence across user interactions is managed with `st.session_state`, allowing the app to retain information across different actions, ensuring a seamless experience.

1. Loading and Visualizing Input Data. Users can upload a CSV file containing feature values for one or more samples (Fig. 5.1): the system displays a preview of the input data, ensuring that users can verify the file's structure before submitting it for prediction.

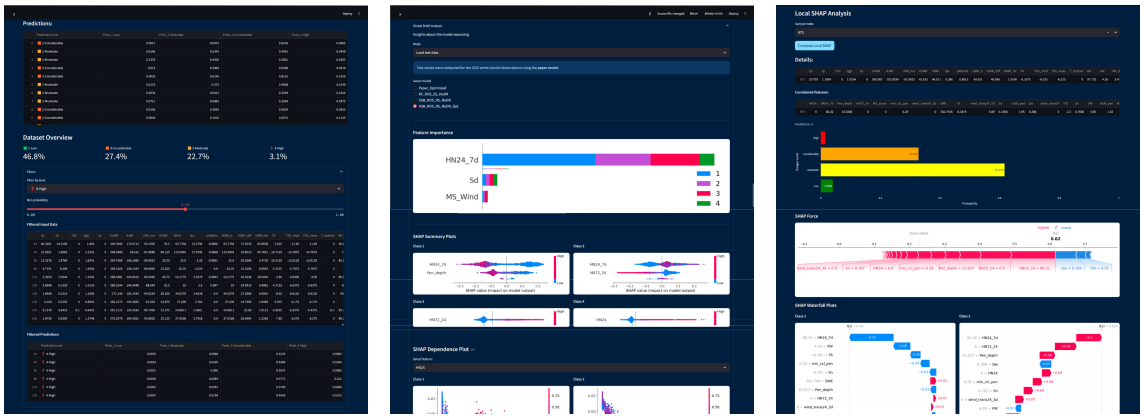
2. Model Selection. Users can choose (Fig. 5.1) which model to apply from the list of available trained models: the system will then run the selected model on the provided input data and generate predictions.

3. Prediction results. For each sample, the predicted avalanche danger level is displayed (Fig. 5.2a) along with the associated probabilities for each class (*Low*, *Moderate*, *Considerable*, and *High*). A dashboard shows the percentage of samples assigned to each danger level class, and users can filter the observations by predicted class or by the probability threshold to view specific subsets of the data.

4. Global SHAP Visualizations. The global SHAP visualizations (Figg. 5.2b) provide insight into the model's decision-making process. The SHAP summary plot

shows the importance of each feature across all predictions, helping users understand which factors most influence the model's output. Class-specific SHAP plots offer a deeper look into how features affect predictions for each class. Additionally, dependence plots allow users to explore how variations in feature values influence the predicted danger level, providing clarity on feature relationships.

5. Local Explanation for Selected Sample. Users can select any sample from the dataset and view its local explanation through **SHAP force plots** or **waterfall plots**. These visualizations (Fig. 5.2c) show exactly how the feature values for that specific observation influence the final prediction, providing a transparent look at how the model arrived at its decision.



(a) Predictions. (b) Global SHAP. (c) Local SHAP.

Figure 5.2: App front-end.

Chapter 6

Results and Conclusion

6.1 Main Findings

The final selected pipeline, `XGB_ROS_RS_NoDR`, was one of only two models, alongside the reference model, to exceed 50% recall on class 4 (High danger) in the hold-out set but, unlike the paper's original parameters, our model was tuned and validated under a strictly leakage-free expanding-window scheme, ensuring the robustness and credibility of its performance, particularly on rare but critical events.

Performances Considerations Despite a ROC-AUC of 0.94 for class 4, indicating strong ranking capability, the model recovered only 52% recall under the default argmax classification rule-nearly half of the true high-danger days were misclassified as class 3.

We ran three exploratory threshold sweeps (Fig. 6.1), each optimizing for F1-score, precision, and recall respectively. None yielded a fully satisfactory trade-off. As a demonstration, we selected a custom threshold of 0.41, which raised recall to 62% with a manageable 10% false-positive rate. However, this threshold was chosen based on the test set, and therefore cannot be used for production. Any operational threshold must be selected via nested cross-validation or on a hold-out within the training data to maintain an unbiased evaluation.

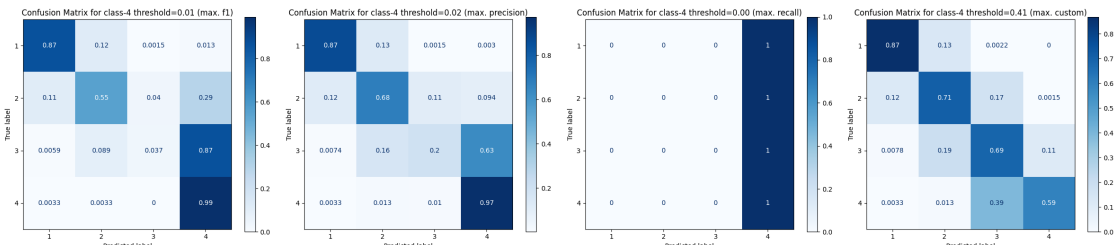


Figure 6.1: CM of `XGB_ROS_RS_NoDR` for different class-4 thresholds.

6.1.1 Impact of Pipeline Design Choices

The comparative analysis of 56 pipelines highlighted three main factors with the greatest impact:

- **Model choice**: Tree-based methods, especially XGBoost, delivered the most robust and accurate predictions. Their ability to capture non-linear interactions and tolerate irrelevant or correlated features made them ideal for this task.
- **Oversampling strategy**: Introducing synthetic or duplicated minority class examples was essential for performance on rare classes. Among the methods tested, RandomOverSampler consistently outperformed SMOTE. This may be due to the fact that SMOTE interpolates feature vectors, which, in the context of physical processes like snow and weather, could produce non-realistic or physically invalid samples.
- **Dimensionality reduction**: The best-performing pipelines often avoided dimensionality reduction altogether. When used, local PCA (applied to correlated feature groups) provided better results than global PCA, likely due to its ability to preserve the internal structure of meaningful variable clusters without distorting feature relationships.
- **Scaling**: no consistent pattern emerged. Pipelines using StandardScaler, RobustScaler, and even no scaling at all appeared among the top performers, suggesting that model and sampling strategy had a far greater influence than the normalization method. This is expected since RF and XGB are tree-based and use split points rather than distance metrics.

6.1.2 Model Optimization

After the feature pruning phase (removal of the less important and more correlated variables) and RFE, the optimized model maintains its ranking ability (ROC and PR curves almost identical), but on the test set only the balanced accuracy improves, while accuracy , F_1-macro and MCC are slightly lower than the baseline model (Tab. 6.1). The Wilcoxon test on the development set ($\alpha = 0.05$) returned $p_{value} = 0.041$, confirming the statistical superiority of the optimized model on the development set, but on the test set the only tangible gain is in the balanced accuracy. The fact that the balanced accuracy goes from 0.6983 to 0.7201 is significant: in the real world, correctly identifying high-risk days can be worth much more than a small drop in overall accuracy. This trade-off – that is, sacrificing some overall performance to better recognize class 4 – may be acceptable.

6.1.3 Explainability Insights

Global SHAP Analysis allowed to identify the most influential features such as snow accumulation variables (e.g. `HN24_7d`, `HN72_24`) and the snow penetration depth.

Table 6.1: Metrics comparison between optimized and baseline final models.

Pipeline	Accuracy	F_1-macro	Balanced Acc.	MCC
XGB_ROS_RS_NoDR_Opt	0.734361	0.700612	**0.720076**	0.618952
XGB_ROS_RS_NoDR	0.742158	0.700881	0.706312	0.627547
RF_ROS_SS_NoDR	**0.754563**	**0.710500**	0.697977	**0.642067**

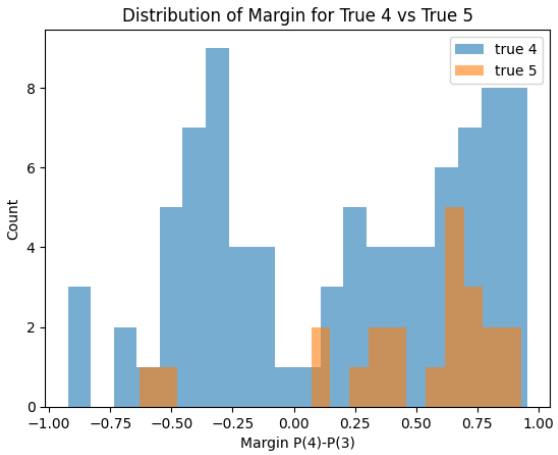


Figure 6.2: Margin distributions for true 4 and 5 samples.

These align with domain knowledge and demonstrate how tree-based models can encode expert-relevant reasoning. SHAP dependence plots provided useful insights into how changes in input features affect predictions, particularly in critical transitions between moderate and considerable danger levels.

6.2 Considerations about Second-Stage Analysis

Across all second-stage experiments, none achieves reliable recovery of true danger-5 events.

- **Baseline analysis** showed that of the 22 true-5 samples, 20 ended up in *predicted=4* (and therefore could be passed to any second-stage method), while 2 were misclassified into lower classes (1–3) and thus irrecoverable.
- The **margin-based cutoff** instead yields $F_1 = 0.342$ but with very low precision, reflects strong overlap between true-4 and true-5 margin distributions (Fig. 6.2).
- Training a **logistic regression** on the two-dimensional space of $\{P(4),\ P(4) - P(3)\}$ yields similar performance to margin thresholding. The 86 true-4 test samples are labeled correctly in 88% of cases ($precision_4 = 0.79, recall_4 = 0.88$), but only 9% of the 22 true-5 samples are recovered ($precision_5 = 0.17, recall_5 =$

0.09). The resulting AUPR remains at 0.282, showing no meaningful gain over the simpler margin rule.

- When treating true-5 samples as anomalies relative to true-4, the **Isolation-Forest** trained only on class-4 data fails to flag any of the 22 true-5 points: Nearly all true-4s (84 out of 86) are identified correctly ($precision_4 = 0.79, recall_4 = 0.97$), but zero of the 22 true-5 points were detected.

Given the scarcity of true-5 examples and their close proximity to true-4 samples, distance-based methods such as k-nearest neighbors (KNN) are unlikely to improve discrimination. In an imbalanced setting, each sample's nearest neighbors will typically be dominated by class 4, effectively drowning out any subtle differences that might exist for class 5.

These consistently results for class 5 indicate that, in the feature representation learned by the first-stage model, level 5 is too similar to level 4 for any probability-based or outlier-based filter to separate them.

Bibliography

[1] C. Pérez-Guillén, F. Techel, M. Volpi, and A. van Herwijnen, "Assessing the performance and explainability of an avalanche danger forecast model," *Natural Hazards and Earth System Sciences*, vol. 25, no. 4, pp. 1331–1351, 2025. [Online]. Available: https://nhess.copernicus.org/articles/25/1331/2025/

[2] C. Pérez-Guillén, F. Techel, M. Hendrick, M. Volpi, A. van Herwijnen, T. Olevski, G. Obozinski, F. Pérez-Cruz, and J. Schweizer, "Data-driven automated predictions of the avalanche danger level for dry-snow conditions in switzerland," *Natural Hazards and Earth System Sciences*, vol. 22, pp. 2031–2056, 2022. [Online]. Available: https://doi.org/10.5194/nhess-22-2031-2022

[3] ——, "Weather, snowpack and danger ratings data for automated avalanche danger level predictions," EnviDat dataset, 2022. [Online]. Available: https://doi.org/10.16904/envidat.330

[4] European Avalanche Warning Services (EAWS), "Fatalities 2024/25," https://www.avalanches.org/fatalities/fatalities-2024-25/, 2025.

[5] AlpNet, "Association of Alpine Tourism Organisations," https://www.alp-net.eu/home/, [Accessed: 27-May-2025].

[6] European Avalanche Warning Services (EAWS), "European Avalanche Danger Scale (2018/19)," https://www.avalanches.org/standards/avalanche-danger-scale/, 2021, [Online; accessed 28-May-2025].

[7] A. van Herwijnen, S. Mayer, C. Pérez-Guillén, F. Techel, M. Hendrick, and J. Schweizer, "Data-driven models used in operational avalanche forecasting in switzerland," in *Proceedings of the International Snow Science Workshop (ISSW)*, Bend, Oregon, USA, October 2023, pp. 321–326, [Accessed: 27-May-2025]. [Online]. Available: https://arc.lib.montana.edu/snow-science/objects/ISSW2023_P1.33.pdf

[8] P. Bartelt and M. Lehning, "A physical snowpack model for the swiss avalanche warning: Part i: numerical model," *Cold Regions Science and Technology*, vol. 35, no. 3, pp. 123–145, 2002.