Machine Learning Project Report

Luc YAO, Richard CHEAM

November 3, 2024

Contents

1 Introduction

This study investigates the use of machine learning to classify ice quantities in Greenland using infrasonic signals and climate data from the ECMWF. By converting four infrasound measurements into binary classifications of low or high ice quantities, we aim to develop predictive models that capture ice dynamics. This approach leverages infrasonic monitoring as a tool for understanding glacier changes and will evaluate various classifiers to identify the model with the best accuracy and interpretability.

2 The data

Two separate datasets were provided, each consists of 2556 rows. One dataset represents 11 different features ($X_{i=1,2,...,11}$) and the other represents 4 target variables ($Y_1, ..., Y_4$). However, instead of four, only one quantitative variable (Y_1) will be considered in this analysis. Information of each feature is described as below:

- `time`: date
- `t2m`: 2 meter below sea temperature
- `SST`: sea-surface temperature
- `(u10, v10)`: a couple of variables indicating wind speed (in separate columns)
- `SIC`: sea ice concentration information
- `r1_MAR, r2_MAR, r3_MAR, r4_MAR, r5_MAR`: Greenland liquid water discharge simulated by Region Climate Models for 5 regions.

Apart from `time`, the other features are continuous variables.

2.1 Target variable | Y_1

From Figure 1 We can observe systematic jumps at the middle of each year. The highest infrasound detection occurs between 2016 and 2017, while the lowest is between 2018 and 2019.

Additionally, according to Figure 2, the values of Y_1 are mostly `0`, accounting for `88.50%` out of 2556 values. While there are `139` values in the range `1-10`, corresponding to `5.44%`, the other intervals contain very few values. Only one value reaches 433 (see Table 4), which is the highest point, occurring in mid-2016 (see Figure 1).

3 Encoding

For the purpose of classification, we need transform the target variable into a binary or dichotomous variable using an appropriate threshold. Hence, the transformation is described as below:

- `0`: no infrasound detected that day
- `1`: infrasound detected that day

After the transformation, we then have 2262 of class `0` and 294 of class `0`, which corresponds to `88.5%` and `11.5%` respectively.

4 Exploratory data analysis (EDA)

As feature `time` is a string type, it will not included in the numerical analysis.

4.1 Summary statistics

From Table 3, we can observe that:

- `t2m` has an average close to -10°C with a wide range from -32°C to almost 8°C, indicating substantial temperature variability.
- Wind components `(u10, v10)` show slight average values close to zero, suggesting no dominant wind direction on average, though both show high variability with values ranging widely in both positive and negative directions.
- `SST` has an average below zero, with most values between -1.69°C (25th to 75th percentiles) and a maximum of 6°C, suggesting cold water conditions with occasional warmer readings.
- `SIC` averages around 73% with many values close to 100%, showing a predominantly high sea ice concentration, although there are outliers at 0%.

- `r1_MAR, ..., r5_MAR` vary widely with low medians and high standard deviations. Many values are close to zero, especially for `r3_MAR` to `r5_MAR`, but extreme high values are observed, particularly in `r1_MAR` and `r2_MAR`, indicating occasional large discharges that are outliers relative to typical values.

4.2 Correlation matrix

Based on the Figure 3, we can see that the Greenland liquid water discharge simulated by Region Climate Models for 5 regions (`r1_MAR, r2_MAR, r3_MAR, r4_MAR, r5_MAR`) are highly correlated. Moreover, while `SST` and `t2m` have a noticeable positive correlation, we can also see that `t2m, SIC` and `SST, SIC` have the highest negative correlation coefficient.

4.3 Distribution of data

From Figure 5, data points in blue (class `0`) are more concentrated on the left, while those in red (class `1`) are wide spread to the right.

4.4 Feature distribution by class

According to Figure 4, we can say that:

- Though we can see some outliers for `v10` and a slightly wider spread of class 0, both classes seem to have a similar median and range (also `u10`). Hence, they might have less distinguishing power than the others.
- For `SIC`, while having a similar median and upper bound, the presence of low outliers at 0 suggests extreme deviations from typical values in both groups.
- Others features suggest notable difference which might have a strong saying (informative) for the classification.

5 Methodology

5.1 Weighted error

To achieve our objective, we trained multiple classifier models on the data. In this study, predicting no infrasound signals but observed the opposite have more impact than the other way around. Therefore, our focus is on **minimizing false negatives**

and having a decent accuracy, as missing true observations of ice could lead to significant misinterpretations.

To address this, we implemented a **weighted error function** with respect to formula below that assigns a higher penalty to false negatives, ensuring the model prioritizes correctly identifying true ice events over minimizing false positives. We chose that false negatives have **5 times** more impact than false positives.

$$\mathbb{E}_{X,Y}[L(Y, f(X))] = \frac{1}{n_{test}} \sum_{i \in Test} L(y_i, f(x_i))$$

where

$$L(Y, \hat{Y}) = \begin{cases} 1 & \text{if } Y = 0 \text{ and } \hat{Y} = 1 \text{ (False Positive)} \\ 5 & \text{if } Y = 1 \text{ and } \hat{Y} = 0 \text{ (False Negative)} \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

5.2 Experimenting on multiple models

We used multiple models for this binary target data set : Naive Bayes, Logistic regression, Decision tree, Random forest, Adaboost and Stacking.

For the decision tree, we initially trained an unconstrained model, then applied the optimal "error versus complexity" metric to identify the level of complexity that minimized the error on the test set.

6 Testing plan

To simplify the model, feature `time` was split into `day, month, year`. Plus, wind components `(u10, v10)` were removed and in order to evaluate model performance, cross-validation approach was used. The data was split into training and testing sets to avoid bias and overfitting, and each model was trained using cross-validation to optimize configurations. This method allowed us to assess model stability and generalizability across different data subsets.

Additionally, we calculated the mean cross-validation accuracy, weighted error, and the area under the ROC curve (AUC) for comprehensive performance metrics.

This approach enabled us to select the best-performing model based on its ability to balance predictive accuracy and the cost of classification errors in line with our study's objectives.

7 Conclusion on the performances

From Table 1, the models which reach a great compromise between accuracy and weighted error are Random Forest and Stacking but the stacking method is costly in time (running for 4 minutes).

Model	Accuracy	Weighted Error	AUC
Naive Bayes	0.923329	0.198748	0.841264
Logistic Regression	0.942889	0.229264	0.804968
Decision Tree	0.909634	0.279343	0.772506
Decision Tree (optimal alpha)	0.937422	0.244131	0.792995
Random Forest	0.943673	0.215962	0.817247
AdaBoost	0.936634	0.240219	0.796992
Stacking	0.943655	0.211268	0.821686

Table 1: Performance metrics for various classification models

8 To go further: data augmentation

To deal with the imbalanced dataset issue, 3 resampling methods was considered, namely:

- Random Undersampling
- Oversampling with SMOTE (Synthetic Minority Oversampling Technique)
- Combination of SMOTE and Edited Nearest Neighbors Undersampling (SMOTEENN)

Model	Accuracy	Weighted Error	AUC
Naive Bayes	0.888652	0.452069	0.892264
Logistic Regression	0.907610	0.437383	0.912806
Decision Tree	0.967162	0.083311	0.968184
Decision Tree (optimal alpha)	0.968765	0.086515	0.969832
Random Forest	0.981311	0.077437	0.981970
AdaBoost	0.928177	0.316422	0.931377

Table 2: Performance metrics with SMOTEENN

As a result, SMOTEENN outperformed the others and better performances were attained, specifically, Random Forest (see Table 2). However, using synthetic data (artificial) introduced potential risks of overfitting, as the model may rely on patterns in the generated samples that do not generalize well to unseen data.

A Appendix

A.1 Supplementary tables

	t2m	u10	v10	SST	SIC	r1_MAR	r2_MAR	r3_MAR	r4_MAR	r5_MAR
Count	2556	2556	2556	2556	2556	2556	2556	2556	2556	2556
Mean	-10.19	0.14	0.63	-0.86	73.27	18.79	11.52	1.33	4.38	5.19
Std	10.34	5.01	3.96	1.45	29.25	47.70	27.94	3.39	12.97	13.41
Min	-32.02	-13.85	-12.32	-1.69	0.00	0.00	0.00	0.00	0.00	0.00
25%	-19.88	-3.61	-2.08	-1.69	70.00	0.12	0.12	0.00	0.00	0.00
50%	-9.60	-0.19	0.91	-1.69	84.60	0.48	0.48	0.00	0.00	0.00
75%	0.17	3.81	3.48	-0.30	90.36	4.08	3.96	0.00	0.01	0.00
Max	7.84	14.64	12.81	6.05	99.50	479.72	281.67	23.24	115.88	88.05

Table 3: Descriptive Statistics for features

	Count	Mean	Std	Min	Max
Y_1	2556	3.53	18.98	0.00	433.00

Table 4: Descriptive Statistics for target variable (before encoding)

Model	**Accuracy**	**Weighted Error**	**AUC**
Naive Bayes	0.843688	0.659864	0.843537
Logistic Regression	0.855611	0.647959	0.855442
Decision Tree	0.756780	0.751701	0.751701
Decision Tree (optimal alpha)	0.847019	0.663265	0.846939
Random Forest	0.829982	0.646259	0.829932
AdaBoost	0.845324	0.664966	0.845238

Table 5: Performance metrics with random undersampling

Model	**Accuracy**	**Weighted Error**	**AUC**
Naive Bayes	0.841741	0.667551	0.841733
Logistic Regression	0.856330	0.647657	0.856322
Decision Tree	0.930814	0.188329	0.931919
Decision Tree (optimal alpha)	0.929044	0.188550	0.932582
Random Forest	0.954253	0.143899	0.954244
AdaBoost	0.881530	0.491600	0.881521

Table 6: Performance metrics with SMOTE (oversampling)

A.2 Supplementary figures

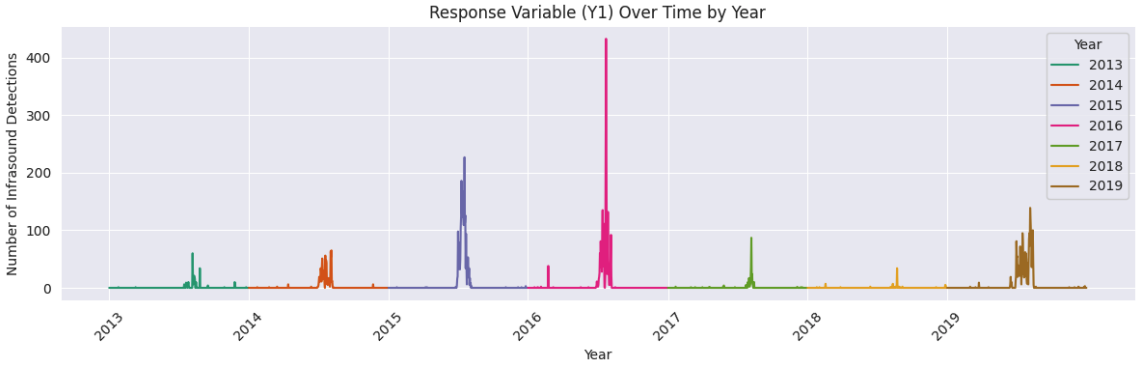

Figure 1: Values of target variable (Y_1) over time

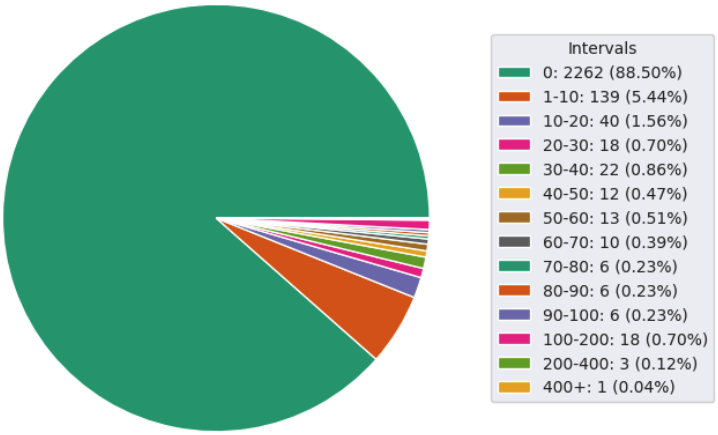

Figure 2: Occurrence of target variable Y_1 by interval in percentage (%)

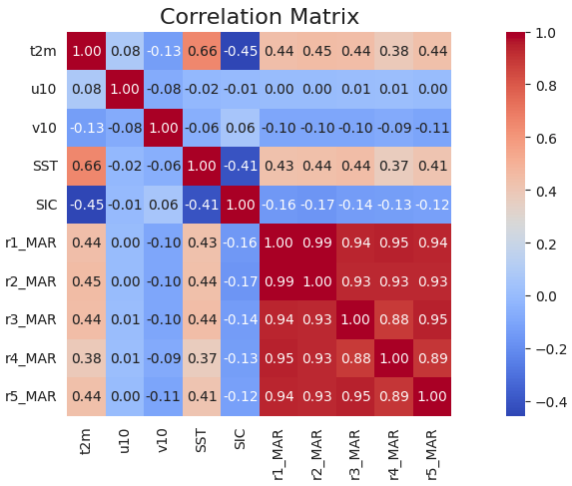

Figure 3: Correlation matrix between 10 features

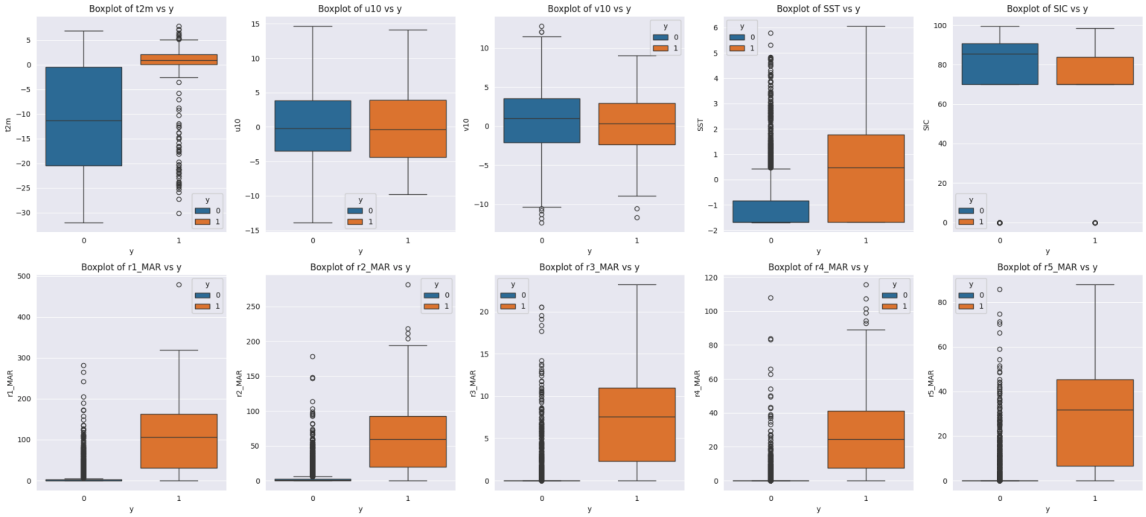

Figure 4: Box-plot of each feature by class

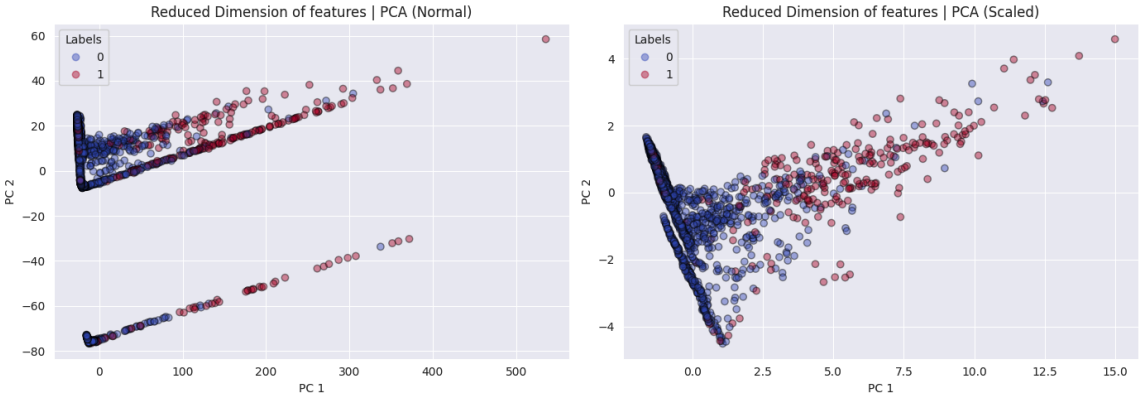

Figure 5: 2 dimensional space plot of initial data (left) and scaled date (right)

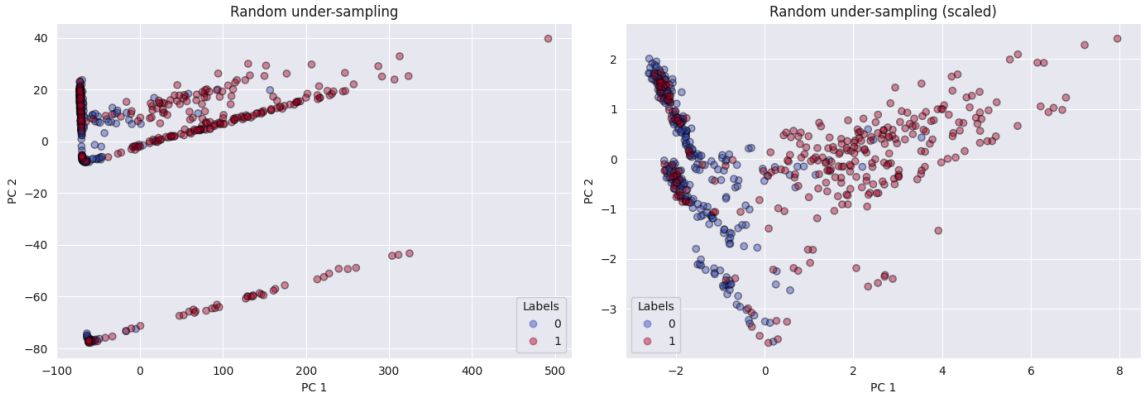

Figure 6: 2 dimensional space plot of random undersampling data

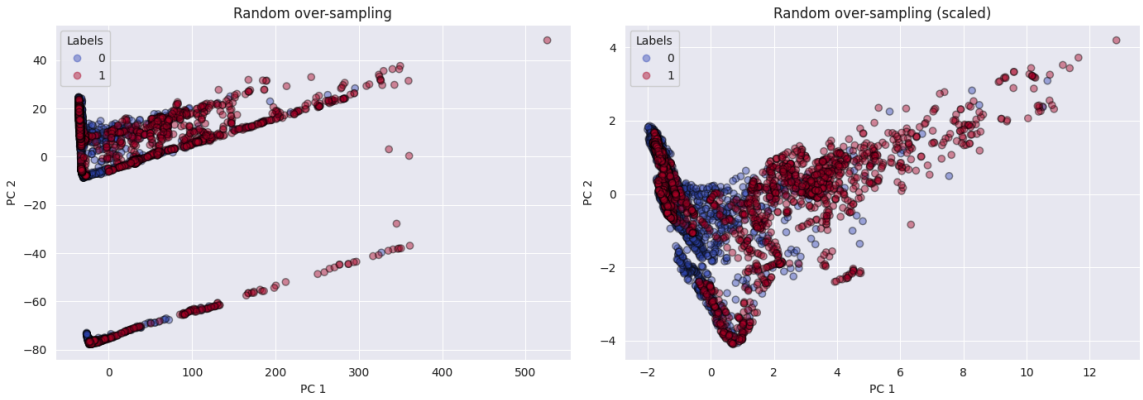

Figure 7: 2 dimensional space plot of oversampling data

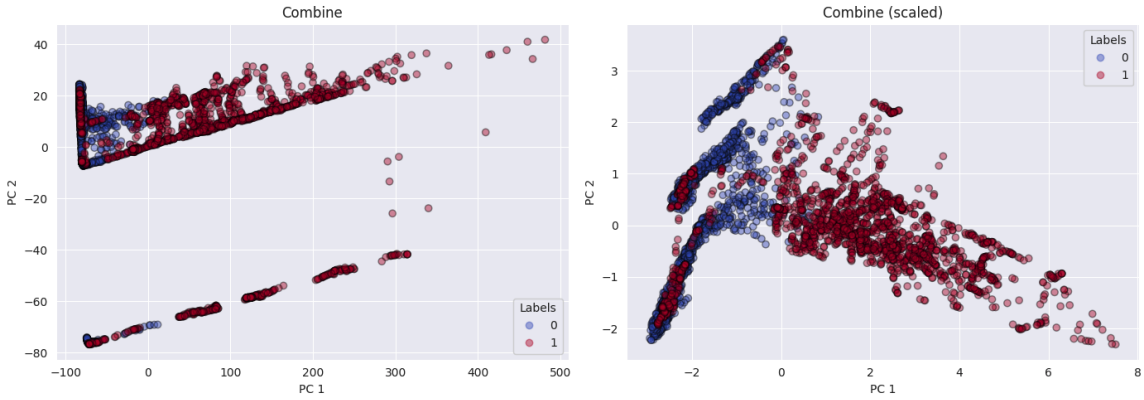

Figure 8: 2 dimensional space plot of SMOTEENN

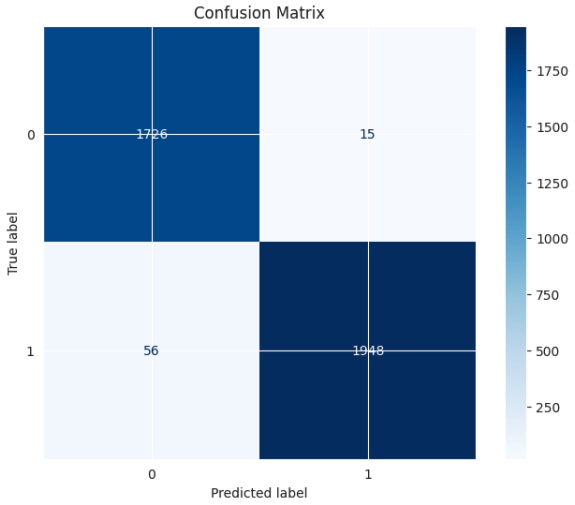

Figure 9: Confusion matrix of SMOTEENN with Random Forest

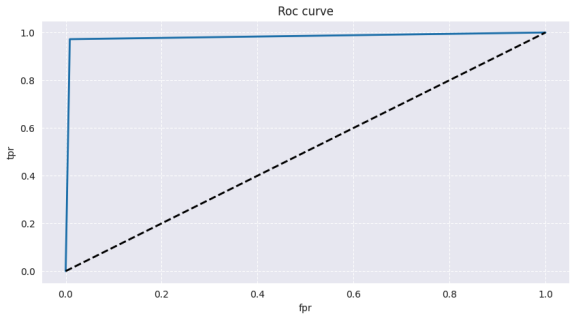

Figure 10: ROC curve of SMOTEENN with Random Forest