Chemometrics and Intelligent Laboratory Systems 273 (2026) 105702

Contents lists available at ScienceDirect

Chemometrics and Intelligent Laboratory Systems

journal homepage: www.elsevier.com/locate/chemometrics

Gradient boosting applied to PTR-ToF-MS analysis of agrifood samples

Pablo M. Granitto [a,b,*], Maria Mazzucotelli [a], Michele Pedrotti [a], Iuliia Khomenko [a], Franco Biasioli [a]

[a] Research and Innovation Center, Fondazione Edmund Mach, Via E. Mach 1, San Michele all'Adige (Trento), Italy
[b] CIFASIS, CONICET-UNR, Ocampo y Esmeralda, Rosario, Argentina

ARTICLE INFO

Keywords:
PTR-toF-MS
XGBoosting
Food analysis

ABSTRACT

Rapid and non-invasive analysis of food products is essential in the agrifood sector for ensuring quality, safety and authenticity. In this context, Volatile Organic Compound (VOC) analysis plays a key role, and direct injection mass spectrometry, Proton Transfer Reaction Mass Spectrometry (PTR-ToF-MS) in particular, offers an optimal tool due to its speed and high sensitivity. The resulting datasets from these analyses are typically modeled using classification, regression, and peak selection methods. In these tasks, gradient boosting methods, and XGBoost in particular, have demonstrated outstanding performance, often surpassing classical machine learning techniques and deep learning approaches. In this work, we investigate the applicability of XGBoost to PTR-ToF-MS datasets of food VOCs in detail. We show that XGBoost requires careful (and time-consuming) optimization to achieve competitive results in this specific domain. Our results indicate that the performance of XGBoost on food products is better in classification than in other analysis tasks, and is comparable on regression and peak selection to that of other state-of-the-art methods, when all methods are appropriately tuned. Given the inherent difficulty of modeling small and noisy real world datasets, our work highlights the importance of carefully evaluating methods within each specific domain, rather than extrapolating their performance as a given.

1. Introduction

Proton Transfer Reaction Time of Flight Mass Spectrometry (PTR-ToF-MS) [1] is a widely used analytical technique for the analysis of volatile organic compounds (VOCs), offering high sensitivity and time resolution. Its applications range from disease detection [2] to air quality monitoring [3]. In the food industry, PTR-ToF-MS has been instrumental in characterizing the chemical profiles of food products, enabling the evaluation of quality, origin and shelf life [4]. However, the high-dimensional and complex nature of PTR-ToF-MS data, which often contain hundreds of peaks, poses significant challenges for data interpretation and predictive modeling.

Machine learning (ML) approaches have emerged as powerful tools to address these challenges, offering the ability to model complex, often non-linear relationships with scarce, high-dimensional data [5]. In the last decade, ML has been dominated by Deep Learning Methods [6]. In spite of its success in almost all types of data, Deep Learning has not been able to clearly outperform traditional methods in tabular datasets [7], leaving place to the application of kernel-based or ensemble methods. Among these methods, eXtreme Gradient Boosting (XGBoost, or XGB) [8] stands out as a versatile and efficient algorithm that combines gradient boosting with regularization to improve predictive accuracy and reduce overfitting.

XGB has been used successfully to model some PTR-ToF-MS datasets. Liu et al. [9] used a pool of machine learning methods, including XGB, to predict the concentrations of two typical human-related VOCs in the classroom over a period of five days. Li et al. [10] applied XGB to identify and validate biomarkers in breath for the screening of lung cancer. Temerdashev et al. [11] developed another study on early detection of lung cancer. Only a very recent work has used XGBoost in the analysis of food products. Kan et al. [12] studied the properties of soy sauce during fermentation using PTR-ToF-MS and two other analytical methods. As part of the study, they applied a pool of ML methods to predict the stage of the fermentation, including XGB, without any particular analysis of the performance or tuning of the classifiers.

In this study, we investigate in depth the application of XGBoost to model PTR-ToF-MS data from food product analysis. This context is particularly challenging, as there are typically very few samples available, combined with significant variability among them. Additionally, measurements are often noisy, which further complicates the

* Corresponding author.
E-mail address: pablo.granitto@fmach.it (P.M. Granitto).

https://doi.org/10.1016/j.chemolab.2026.105702
Received 21 February 2025; Received in revised form 4 February 2026; Accepted 6 March 2026
Available online 7 March 2026

modeling process. It is not clear whether the characteristics of XGB that lead to great performance in other domains are useful in this context. Using several, diverse example datasets, we analyze the performance of XGBoost and compare it with other ML methods on both classification and regression tasks. We also evaluate the impact of the selection of hyperparameters on the performance of the methods.

This paper is organized as follows. In Section 2 we describe all methods and datasets, in Section 3 we show and analyze the corresponding results and then we draw some conclusion and describe future work on the last section.

2. Methods

2.1. XGBoost

XGB is an ensemble ML method, meaning it combines the predictions of many simple models to create a more powerful and accurate model. Specifically, XGBoost is a gradient boosting algorithm [13], where the ensemble is built sequentially and each individual model is trained to correct the errors of the previous ones by minimizing a specific cost function. This sequential process allows XGB to focus on hard instances, usually improving overall performance and making it highly effective for a wide range of tasks. It adds the use of random sampling of both samples and features, following Random Forest (RF) [14], which helps to avoid overfitting. Moreover, the combination of prediction trees and cost penalization makes the method intrinsically resistant to outliers or leverage points [15]. It has been used in several machine learning challenges with great success [16]. It is considered as one of the most efficient methods for tabular data [7]. XGB combines the positive properties of both gradient boosting and RF methodologies. It shows great capability in modeling complex and highly non-linear relationships within the data, making it well-suited for challenging analytical tasks. Furthermore, XGB is highly efficient when applied to large datasets owing to its scalable design. Its robustness to noise is enhanced through the use random sampling of data and features during training, which avoids overfitting and produces better generalization.

Unlike RF, XGB has several parameters that need to be tuned to achieve good performance. The method relies on gradient descent, and therefore includes a learning rate (*alpha*) as a key parameter. As a boosting method, the performance is also influenced by the complexity of the individual classifiers. In XGB, this complexity can be controlled through various parameters; in particular, we will use the maximum depth of the trees (*max_depth*). Additionally, the method incorporates random sampling to prevent overfitting. Among the various sampling techniques it offers, we will specifically focus on tuning the proportion of data randomly selected to train each tree (*subsample*) and the proportion of features chosen for that task (*colsample_bytree*).

The final relevant parameter in XGB is the number of trees that form the ensemble. In this case, we opted to use a fixed and large number of trees. To support this decision, we conducted an evaluation comparing it with two alternative approaches for selecting the ensemble size. These alternative methods included dynamically determining the optimal number of trees based on early stopping criteria and selecting with cross validation (CV) over a set of predefined number of trees. By evaluating these strategies, we were able to justify our choice of a large, fixed ensemble size, balancing model performance and computational efficiency.

2.2. Comparison methods

We use several classical ML methods to produce an extended unbiased evaluation of XGB.

For classification datasets we selected four very diverse methods. RF [14] is a popular ML base method for any classification or regression problem, which can be used as an "off the shelf" method. It is particularly efficient for noisy problems and we consider it as the base comparison method for PTR-ToF-MS datasets. We also use Linear Discriminant Analysis with shrinkage (LDA-S) [17], a regularized variant of the well known statistical method. Partial Least Squares coupled with LDA (PLSDA) [18] is one of the classical methods for mass spectrometry. Last, we use Support Vector Machines (SVM) [19], an ML method with a strong mathematical basis which shows excellent results when properly tuned.

For regression datasets we use the corresponding variants of RF and SVM, and compare also with the LASSO variant of the least angle regression method [20], which is specifically designed for high dimensional problems.

In all cases we used the same optimization strategy as in XGB, using CV with a grid search, leaving the rest of the setup with the default values of the Scikit-Learn implementations [21]. For RF we used a fixed setup with 1000 trees for both regression and classification. For LDA-S we used the automatic selection of the shrinkage parameters, for PLSDA we tuned the number of PLS components. In classification with SVM we optimized the C constant with a linear kernel, while in regression we also used a linear kernel but optimized both C and the size of the tube, ϵ. For this parameters we made the selection over a set of fractions of the standard deviation of the value to predict on each dataset. Last, for LASSO we optimized α, the regularization parameter.

2.3. Datasets

We compare all the methods using diverse datasets comprising the evaluation of food related products. We discuss here the characteristic of our datasets that are more relevant to the modeling process and left other information to Appendix A.

In general, tabular data are extracted from PTR-ToF-MS raw data following the procedure described in Cappellin et al. [22], which includes m/z calibration and various preprocessing steps for noise reduction and baseline removal before peak extraction.

The final data are represented as a table containing the estimated concentration for each peak and each sample with sample identification in the first column and the accurate m/z values of each peak as column headings.

Tables 1 and 2 summarize the key characteristics of each dataset. In both tables, datasets are ordered by the sample-to-peak ratio, presented in the last column. Our selection encompasses a wide range of real-world scenarios in terms of the number of classes and sample-to-peak ratios.

Both tables also include the number of batches considered for each dataset. The concept of a "batch" is essential to modeling food-related samples, as it represents a group of samples that is independent from other groups but not necessarily within themselves. Studies focusing on specific food-related problems may sometimes analyze and discuss batch effects. In this work, we limit ourselves to a simple statistical perspective. Even when a batch contains diverse samples rather than simple technical replications, internal dependencies typically exist. A batch may correspond, for instance, to samples from a specific factory, a particular place of origin, or a specific production year or season.

The choice of batch composition can significantly influence the nature of the modeling problem. The same dataset may contain samples from different geographical origins and multiple years of production, leading to diverse predictive challenges. For example, predicting a product's property for a new geographical origin versus predicting it for a new production year can pose fundamentally different problems. In regression tasks, we accounted for two of these scenarios in most datasets.

Based on these considerations, we employed a leave-group-out approach for all evaluations, using a CV strategy in which each group corresponds to a batch.

Table 1
Details on Datasets for classification tasks.

Dataset	Samples	Batches	Peaks	Classes	S/P
Tea	456	21	161	4	2.83
Gum2	267	27	167	2	1.60
Gum3	267	27	167	2	1.60
Mush 21	593	50	383	6	1.55
Mush 20	396	65	402	12	0.99
Mush 13	54	19	125	6	0.43
Fish	104	32	259	3	0.40
Peppers	96	32	253	2	0.38
Spinach	72	24	333	2	0.22
Ham	54	18	427	3	0.13
Lacto	102	20	798	2	0.13
Coffee	36	12	563	6	0.06

Table 2
Details on Datasets for regression tasks.

Dataset	Samples	Batches	Peaks	S/P
Gum1_s	267	27	167	1.60
Gum2_s	267	27	167	1.60
Gum3_s	267	27	167	1.60
Gum1_b	267	3	167	1.60
Gum2_b	267	3	167	1.60
Gum3_b	267	3	167	1.60
Noc_S	72	24	380	0.19
Noc_O	72	3	380	0.19
Noc_3T	60	20	383	0.16

2.4. Software

We used open-source Python implementations of each method, including an extended use of the SciKit Learn library [21]. All our code and data are available at https://github.com/CIFASIS/XGB_PTRMS.

2.5. Peak selection

One of the most useful characteristics of ensemble methods is their ability to perform efficient feature selection, or peak selection in our case. Typically, feature importance scores are combined with Recursive Feature Elimination (RFE) [23] to produce the final selection. XGBoost can be used in this context. Since one of XGBoost's advantages is its ability to incorporate randomness in a way similar to Random Forest, we evaluated two approaches for feature selection using this method: a deterministic (or fixed) method (XGB-all) and an XGB variant that introduces randomness both in the dataset and the feature selection process (XGB-rnd). In both cases, we used the fixed setup discussed before for all parameter values.

Feature selection methods can be evaluated for the quality of the selected subset, in terms of an appropriate error measure for classification or regression. It is also relevant to measure the stability of the method, i.e., the capability of producing the same selection in similar situations. Clearly, another property of interest is the identity of selected peaks, which we do not analyze in this work because we limit ourselves to global (statistical) measures of peak quality.

We compare both selection strategies with the well-known RFE-RF method [24], both in terms of quality and stability. To this end we produced replicated experiments using the leave-group-out setup, selecting features with RFE on the training sets and evaluating the subset each time on the corresponding test set.

3. Results

3.1. Tuning procedure

As discussed in Section 2.1, XGB requires the selection of multiple hyperparameters. Various optimization strategies can be applied in this context [25]. In this study, we used a straightforward grid search over the parameter space, considering a limited set of values for each parameter. This approach was chosen for its interpretability and because XGB demonstrated a relatively smooth performance dependence on these parameters. To support this decision, we compared this method with Random Search [26] and Bayesian Search [27], two widely used hyperparameter selection methods, over all classification datasets. All three methods show very similar performances (see Appendix B for details).

Despite the simplicity of the grid setup, the search complexity grows exponentially with the number of parameters. To address this, we also evaluated two alternative configurations: a reduced setup involving only two tunable parameters and a fixed configuration with values specifically chosen to accommodate high-noise scenarios. In the fixed setup we used *alpha = 0.12*, *max_depth = 3*, *subsample = 0.8* and *colsample_bytree = 0.75*, which we consider as a good trade-off between randomness and accuracy. In the reduced setup we choose to tune *max_depth* and *subsample*, leaving the other parameters in the same values as the fixed setup. We use the same setup for both classification and regression problems.

Table 3
Tuning analysis: mean classification error results on three strategies to setup model parameters on XGB. We include RF results as reference in the first column, followed by full tuning, reduced set tuning and fixed configuration.

Dataset	RF	XGB		
		Full	Reduced	Fixed
Tea	0.5461	0.4956	0.5000	0.5000
Gum2	0.1273	0.0974	0.1011	0.1161
Gum3	0.1049	0.0936	0.0899	0.0974
Mush 21	0.2192	0.1585	0.1535	0.1636
Mush 20	0.2449	0.2247	0.2475	0.2576
Mush 13	0.6667	0.5370	0.5741	0.5926
Fish	0.0385	0.0481	0.0481	0.0481
Peppers	0.2604	0.2604	0.2604	0.2500
Spinach	0.4028	0.3333	0.3333	0.3194
Ham	0.4259	0.2963	0.2778	0.3148
Lacto	0.0392	0.0392	0.0490	0.0686
Coffee	0.4167<	0.3611	0.4722	0.5833

Table 4
Tuning analysis for regression: NMSE results on three strategies to setup model parameters on XGB. We include RF results as reference in the first column, followed by full tuning,reduced set tuning and fixed configuration.

Dataset	RF	XGB		
		Full	Reduced	Fixed
Gum1_s	0.5899	0.4338	0.4980	0.5327
Gum2_s	0.5470	0.2607	0.2716	0.4907
Gum3_s	0.7891	0.8751	0.8366	0.9396
Gum1_b	0.2640	0.2378	0.2625	0.2657
Gum2_b	0.2894	0.2580	0.2622	0.3409
Gum3_b	0.4672	0.4772	0.5122	0.4587
Noc_S	0.1581	0.1602	0.1438	0.1606
Noc_O	0.2850	0.3292	0.3272	0.3143
Noc_3T	0.6549	0.4990	0.4059	0.3288

Table 3 shows the corresponding results for classification problems. We include RF results for each dataset to provide a reference value to confront the differences between the three strategies. In almost all cases there is a consistent behavior, with the full tune showing better results than the reduced setup, which shows improvements over the fixed setup. The results for regression problems, Table 4, show a similar pattern. We use Normalized Mean Squared Error (NMSE) to evaluate all regression results, because it has a meaningful scale. It is worth mentioning that the full tuning of the bigger problems in our experiments took several hours on a 64 cores server. The cost/benefit of this procedure should be taken into account in all cases.

All previous experiments used a fixed size for the ensemble. To justify this choice we performed a comparison with two alternative

Table 5
Tuning analysis: mean classification error results on three strategies to select the number of rounds for XGB: selecting the number of rounds with CV, individual setup using fixed train/validation split, and fixed number of rounds.

Dataset	C.V.	Split	Fixed
Tea	0.5175	0.5197	0.5000
Gum2	0.1161	0.1311	0.1161
Gum3	0.0936	0.1011	0.0974
Mush 20	0.2576	0.2778	0.2576
Mush 21	0.1636	0.1669	0.1636
Mush 13	0.5926	0.7037	0.5926
Fish	0.0481	0.0481	0.0481
Peppers	0.2500	0.2708	0.2500
Spinach	0.3472	0.4306	0.3194
Ham	0.3148	0.3704	0.3148
Lacto	0.0686	0.0686	0.0686
Coffee	0.5833	0.3889	0.5833

Table 6
Tuning analysis for regression: NMSE results on three strategies to select the number of rounds for XGB: selecting the number of rounds with C.V., individual setup using fixed train/validation split, and fixed number of rounds.

Dataset	C.V.	Split	Fixed
Gum1_s	0.5321	0.5431	0.5327
Gum2_s	0.4907	0.5151	0.4907
Gum3_s	0.9404	0.8513	0.9396
Gum1_b	0.2655	0.2613	0.2657
Gum2_b	0.3410	0.2858	0.3409
Gum3_b	0.4597	0.4548	0.4587
Noc_S	0.1607	0.1379	0.1606
Noc_O	0.3142	0.3168	0.3143
Noc_3T	0.3288	0.3992	0.3288

approaches for selecting the number of trees. First, on the CV approach we selected the size with cross validation over a set of predefined number of trees (125, 250, 500 and 1000). After identifying the optimal size, XGB was retrained using all available samples. This is the most computationally expensive setup. Also, we used the Split strategy, in which we split the data into a training and a validation set, and dynamically determine the optimal number of trees based on early stopping criteria on the validation set. In this case, no additional training step was performed after determining the optimal number of trees. In all cases the other parameters of XGB used the Fixed setup discussed before.

Tables 3 and 4 present the corresponding results for classification and regression problems. The Fixed column uses the same setup as the last column in the previous table and show the same results. The results indicate that the Split method is outperformed by the other two options in almost all cases. Additionally, using a fixed, large number of trees yields comparable results to optimizing the number of trees while requiring significantly less computational effort (see Tables 5 and 6).

3.2. Comparison with other methods

We performed a complete comparison of XGB with other selected methods in both classification and regression setups. For XGB we used the Full tuning strategy with a fixed size of 1000 trees. The setup of the other methods was discussed in Section 2.2.

Table 7 presents the classification results. Figs. 1 and 2 visually summarize these results. The Friedman test did not reveal statistically significant differences among the evaluated methods ($p = 0.11$ both in regression and classification). Accordingly, post-hoc Nemenyi tests did not identify any significant pairwise differences after correction for multiple comparisons. This result is consistent with the limited number of datasets and the conservative nature of non-parametric multiple-comparison tests, as discussed by Demšar [28]. Therefore, we analyze the frequency with which each method achieves the best performance and the total number of competing methods each method

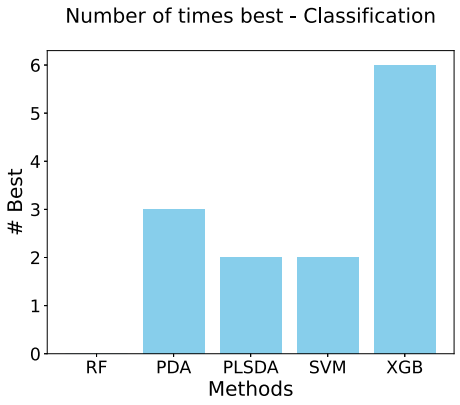

Fig. 1. Number of times that each method is the best method in classification tasks.

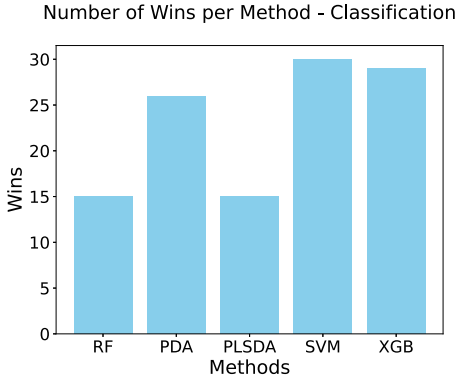

Fig. 2. Number of competing methods outperformed by each method in classification tasks.

outperforms across all datasets. These metrics provide a more robust characterization of method behavior than average error rates, as they are independent of dataset-specific error magnitudes. Notably, XGB demonstrated the best performance in six out of twelve datasets, and its overall performance, measured by the total number of outperformed methods, was comparable to SVM. Specifically, XGB seems to work better for problems with a bigger samples/features ratio, i.e., problems with more available information.

We repeated the analysis for regression problems. In this case, Table 8 presents the results and Figs. 3 and 4 visually summarize them. XGB does not show the same good performance in regression, giving the best result in only one dataset. The Lasso-Lars method, developed for high dimensional regression, seems to be the method of choice in this case, followed by SVR.

3.3. Peak selection

On the bottom panel of Fig. 5 we show the average classification error for the three selection strategies as a function of the size of the selected subset. The errors correspond to the mean value over the leave-group-out setup. We show errors for each dataset and also a thick black line with the mean values over all datasets. The results indicate a small advantage of RF-RFE over XGB-rnd and then over XGB-all. On the top panel we show the stability results, as the mean value of the RBO index [29] over all possible pairs of subsets of features of each size. RBO is 1 for complete agreement and 0 for complete disagreement, and gives more weight to first positions. Again, RF seems more stable than XGB-rnd and XGB-all.

PTR-ToF-MS datasets typically include several correlated peaks, corresponding to fragmentation products, isomers, or other related

Table 7
Classification: Comparison with typical methods for PTR-ToF-MS datasets. Average classification error over all samples in the dataset. Bold indicates best result.

D.S.	RF	PDA	PLSDA	SVM	XGB
Tea	0.5526	0.5000	0.5636	0.5241	**0.4956**
Gum2	0.1273	0.1273	0.1199	0.1348	**0.0974**
Gum3	0.1086	0.1049	0.1124	0.0974	**0.0936**
Mush 21	0.2108	**0.1298**	0.4327	0.1484	0.1585
Mush 20	0.2576	0.3106	0.5783	0.2955	**0.2247**
Mush 13	0.6667	0.6296	0.7778	0.5926	**0.5370**
Fish	0.0385	0.0096	**0.0000**	0.0096	0.0481
Peppers	0.2500	0.2604	0.2396	**0.1875**	0.2604
Spinach	0.3889	**0.2778**	**0.2778**	0.3194	0.3333
Ham	0.4259	0.4444	0.5370	0.4259	**0.2963**
Lacto	0.0294	0.0196	0.0196	**0.0000**	0.0392
Coffee	0.4444	**0.0278**	0.5833	0.1111	0.3611

Table 8
Regression: Comparison with typical methods for PTR-ToF-MS datasets. NMSE error over all samples in the dataset. Bold indicates best result.

Dataset	RF	LASSO	SVR	XGB
Gum1_s	0.5801	**0.1996**	0.2077	0.4129
Gum2_s	0.5620	0.4313	0.4148	**0.3816**
Gum3_s	0.7922	0.8568	**0.7222**	0.8942
Gum1_b	0.2603	**0.1355**	0.1507	0.2421
Gum2_b	0.2914	0.3662	**0.2134**	0.2561
Gum3_b	**0.4722**	0.5104	0.5675	0.4960
Noc_s	0.1561	**0.0631**	0.0658	0.1428
Noc_o	0.2871	**0.2062**	0.2669	0.3311
Noc_3T	0.6570	**0.2005**	0.2008	0.3354

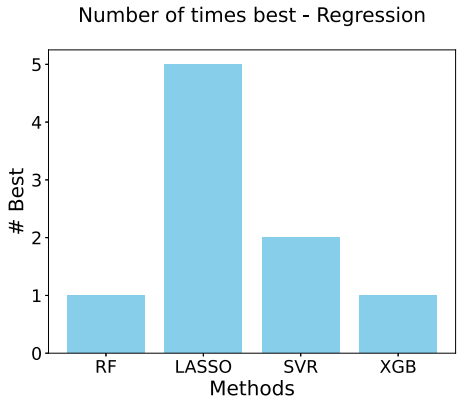

Fig. 3. Number of times that each method is the best method in regression tasks.

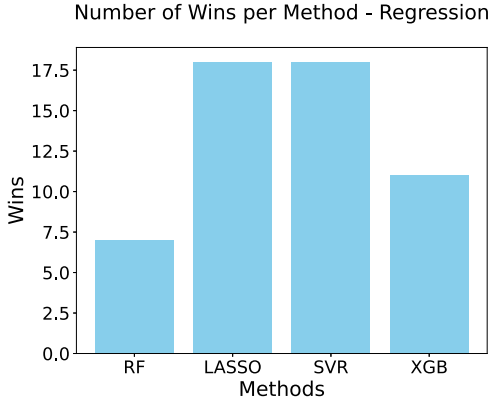

Fig. 4. Number of competing methods outperformed by each method in regression tasks.

compounds. As this can be particularly difficult to some feature selection methods, we repeated the procedure but first deleting all peaks with a high linear correlation with any other. We found that the results are qualitatively similar to using all peaks, indicating that correlation is not highly influential for the evaluated methods. We show the corresponding figures on Appendix B.

On regression problems, Fig. 6, RF is again the more stable method. On the other hand, XGB-rnd selects the subsets with the lower NMSE in this case, showing a small difference in particular with all peaks. The analysis without correlated peaks show the same qualitative results (Appendix B).

All together, our results suggest that, for PTR-ToF-MS food products datasets, using XGB-RFE for peak selection or the well established RF-RFE produce similar results, with small differences that are highly dependent on the dataset.

4. Conclusions

In this work we evaluated the use of a gradient boosting method, XGBoost, on PTR-ToF-MS data from food related samples. We first discussed the need for a correct setup for XGB, showing that the results improve with a correct setting, but there is a high computational cost involved that should be taken into account.

Overall, XGB shows better results for classification than regression problems in our experiments, and also better results in problems more informative, i.e., problems with a higher samples/features ratio.

The analysis of peak selection experiments indicates a similar performance of XGB and RF, both coupled with RFE, in quality and stability of the selection process.

As a guide to practitioners, our results suggest that XGBoost is more adapted to model classification PTR-ToF-MS problems using a reduced tuning procedure, involving mainly the complexity and the randomness of the classification trees.

While XGB has demonstrated remarkable performance in other domains, its application to PTR-ToF-MS analysis of agrifood samples, with its inherent challenges of limited and highly diverse samples, does not seem to achieve the same level of success. Nevertheless, XGB should be considered as a useful tool for this field, provided it is carefully tuned, as many other machine learning methods. At the end, our work highlights the importance of carefully evaluating methods within each specific domain, rather than extrapolating their performance as a given.

Our work exploited PTR-ToF-MS data as prototypical example of DIMS (Direct Injection Mass Spectrometry) and CIMS (Chemical Ionization Mass Spectrometry). Future research includes adding new types of agrifood data to the evaluation and consider other type of PTR-ToF-MS data, as for example the more complex data obtained adding ion mobility spectrometry to the time of flight data.

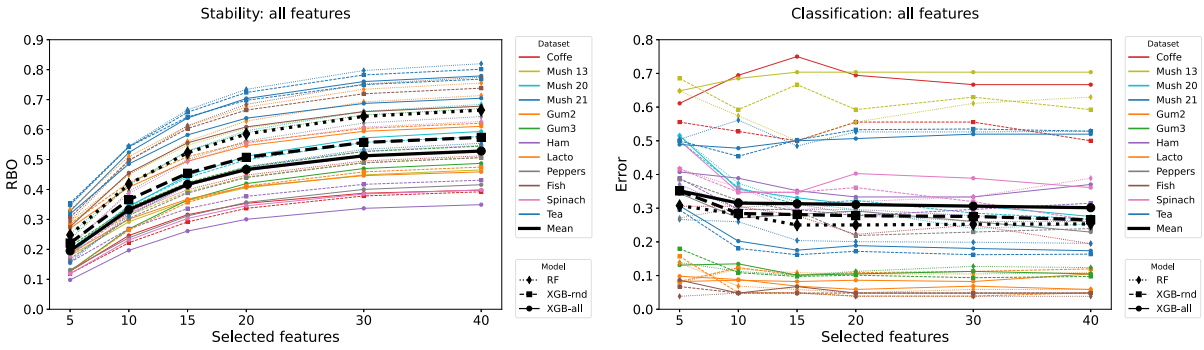

Fig. 5. Classification error (bottom) and stability of selections (top) as a function of the number of features selected by each method.

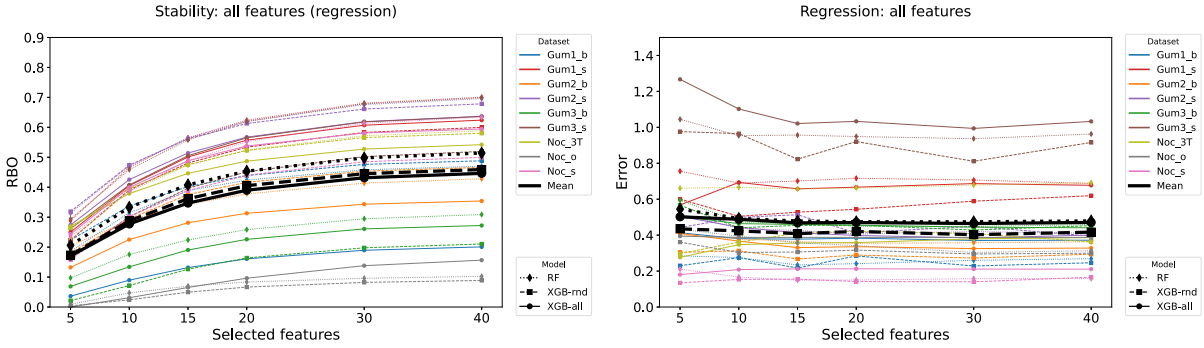

Fig. 6. Same as Fig. 5 for regression problems.

CRediT authorship contribution statement

Pablo M. Granitto: Writing – review & editing, Writing – original draft, Software, Methodology, Investigation, Formal analysis, Conceptualization. **Maria Mazzucotelli:** Writing – review & editing, Data curation. **Michele Pedrotti:** Writing – review & editing, Data curation. **Iuliia Khomenko:** Writing – review & editing, Data curation. **Franco Biasioli:** Writing – review & editing, Supervision, Project administration, Funding acquisition, Conceptualization.

Declaration of Generative AI and AI-assisted technologies in the writing process

During the preparation of this work the author(s) used ChatGPT and Gemini in order to improve the readability and language of the manuscript. After using these tools/services, the author(s) reviewed and edited the content as needed and take(s) full responsibility for the content of the published article.

Declaration of competing interest

We declare no conflicts of any kind.

Acknowledgments

Part of the results presented in this work have been obtained by using the facilities of the CCT-Rosario Computational Center, member of the High Performance Computing National System (SNCAD, Argentina). PMG thanks funding from "Visiting in Trentino" 2024 call. This work has been partially supported by the SISTERS project that has received funding from the European Union's Horizon 2020 research and innovation programme under Grant Agreement No. 101037796.

Table B.9
Tuning analysis: comparison on search methods to setup model parameters on XGB. The table shows mean classification error results on three search strategies, namely Grid Search, Random Search and Bayesian Search. We include RF results as reference in the first column.

Dataset	RF	XGB		
		Grid	Random	Bayesian
Tea	0.5461	0.4956	0.5044	0.5066
Gum2	0.1273	0.0974	0.1124	0.1086
Gum3	0.1049	0.0936	0.0974	0.0974
Mush 21	0.2192	0.1585	0.1619	0.1669
Mush 20	0.2449	0.2247	0.2399	0.2374
Mush 13	0.6667	0.5370	0.5370	0.5000
Fish	0.0385	0.0481	0.0481	0.0481
Peppers	0.2604	0.2604	0.2396	0.2500
Spinach	0.4028	0.3333	0.3194	0.3333
Ham	0.4259	0.2963	0.3148	0.2593
Lacto	0.0392	0.0392	0.0392	0.0392
Coffee	0.4167	0.3611	0.3889	0.5000

Appendix A. Datasets

In this section we include a brief description of each dataset used in this work. In the cases when data was already described in a publication, we include the corresponding citation.

A.1. Classification

Tea: Evaluation of leaves of green and black tea. The classes correspond to the geographical origin of each batch. Experimental details in Yener et al. [30].

Gum 2 & 3 Evaluation of samples of base material for chewing gum. The classes correspond to the presence or absence of two components in the base material, over a set of several possible combination of components.

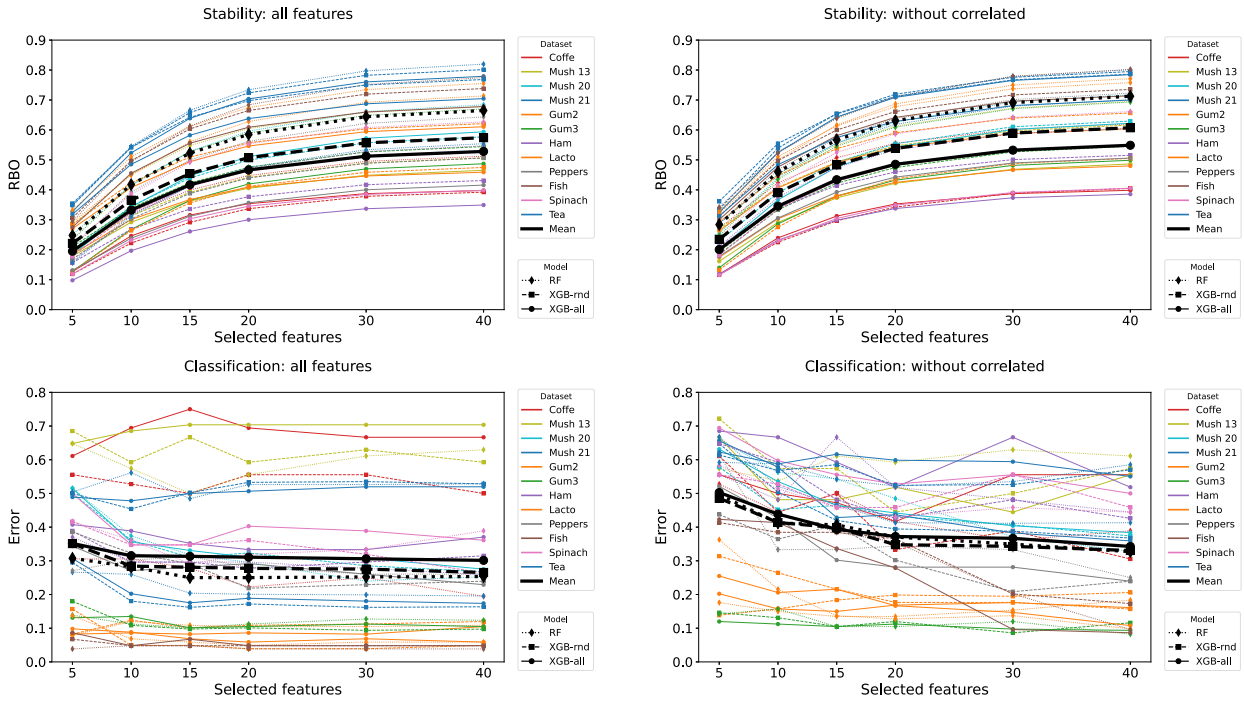

Fig. C.7. Classification error (bottom) and stability of selections (top) as a function of the number of features selected by each method. Left: all features. Right: correlated features were removed before features selection.

Mush 20 & 21 Evaluation of samples of diverse species of fungi. The classes correspond to the species (Mush 20) or to the condition of cultivation (Mush 21). Experimental details in Telagathoti et al. [31]

Mush 13 Evaluation of mushrooms samples of diverse species of the genus Armillaria. The classes correspond to the species. Experimental details are similar to Mush 20.

Fish Evaluation of fish meat samples. The classes correspond to different cooking processes. Experimental details in Khomenko et al. [32]

Peppers Evaluation of whole fresh peppers. The classes correspond to two different methods of conservation. Experimental details are similar to Spinach data set described below.

Spinach Evaluation of fresh spinach leaves. The classes correspond to two different methods of conservation. Experimental details in Khomenko et al. [33]

Ham Evaluation of samples of dry cured ham. The classes correspond to the geographical, controlled & protected origin of each batch. Experimental details in del Pulgar et al. [34]

Lacto Evaluation of samples of diverse strains of lactic acid bacteria during fermentation, taken at three consecutive times. The classes correspond to the temperature of the fermentation process. Experimental details in Rajendran et al. [35].

Coffee Evaluation of coffee powder. The classes correspond to the geographical origin of each batch. Experimental details in Yener et al. [36]

A.2. Regression

Gum 1, 2 & 3 Evaluation of samples of base material for chewing gum production. The predicted value correspond to the percentage concentration of three components in the base material, over a set of several possible combination of components. The batch correspond to technical replicates of a single sample (“s”) or to production batches (“b”).

Noc Evaluation of hazelnut paste samples, obtained processing raw kernels (Corylus avellana L.) from different geographical origins and different years. Samples were roasted using different times and temperatures. The predicted value correspond to the roasting time. The batchs correspond to technical replicates of a single sample (“s”) or to origin of the hazelnuts (“o”). More experimental details in Mazzucotelli et al. [37]

Noc_3T Evaluation of hazelnut paste samples as in the previous case. The predicted value correspond to the temperature of the oven during the roasting.

Appendix B. Tunning strategies

We considered three diverse strategies for hyperparameter optimization. In the main text we used Grid Search, where all combinations over a fixed set of values for each parameter are evaluated. We compare here that base method with two typical alternatives on the classification datasets:

- Random Search [26]: We define continuous or discrete ranges for each parameter, of similar size to the grid search, and use the same number of evaluations as the base method.
- Bayesian Search [27]: Again, we define continuous or discrete ranges for each parameter, of similar size to the grid search, and use the same number of evaluations as the base method. In particular, we use log scale for the learning rate and linear scale for other continuous parameters.

In the Table B.9, we show a comparison of the three methods for all classification problems. It is clear from the table that all methods produce similar results in our small and noisy datasets.

Appendix C. Additional figures

See Fig. C.8.

Data availability

All data and code is available at github as explained in the paper.

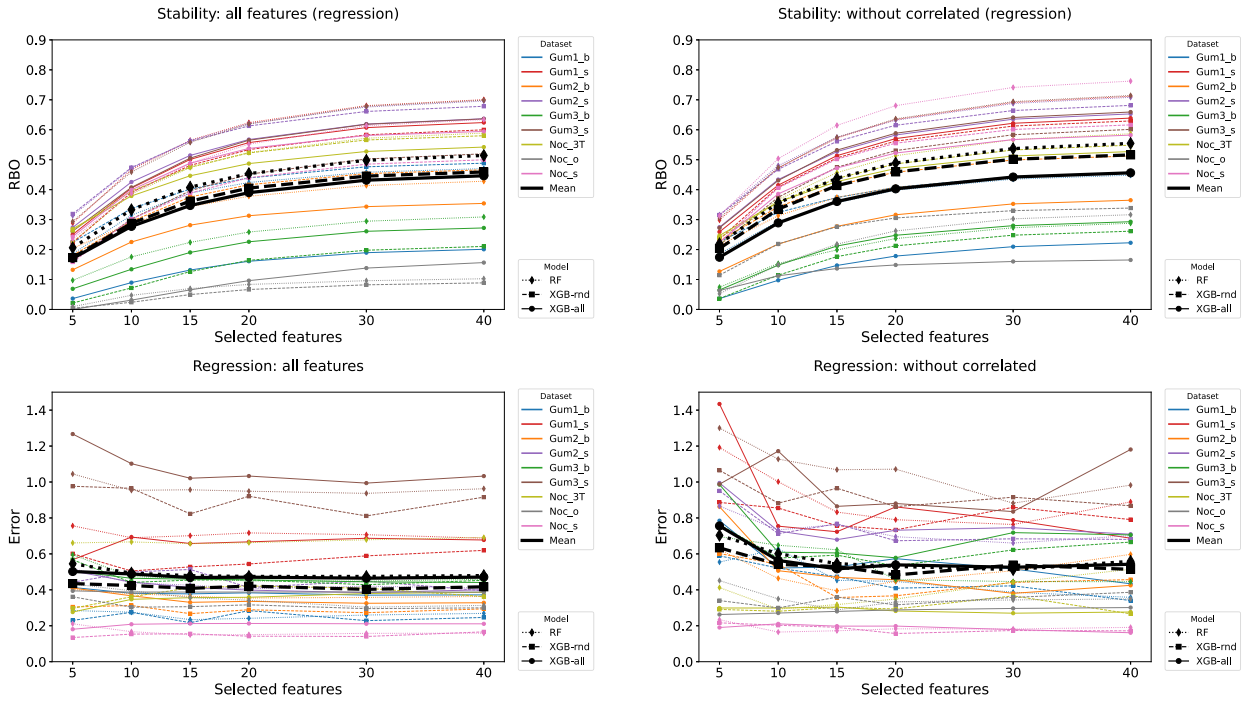

Fig. C.8. Same as Fig. C.7 for regression problems.

References

[1] W. Lindinger, A. Jordan, Proton-transfer-reaction mass spectrometry (ptr–ms): on-line monitoring of volatile organic compounds at pptv levels, Chem. Soc. Rev. 27 (1998) 347–375.

[2] B. Moser, F. Bodrogi, G. Eibl, M. Lechner, J. Rieder, P. Lirk, Mass spectrometric profile of exhaled breath—field study by ptr-ms, Respir. Physiol. Neurobiol. 145 (2005) 295–300, http://dx.doi.org/10.1016/j.resp.2004.02.002.

[3] J. De Gouw, C. Warneke, Measurements of volatile organic compounds in the earth's atmosphere using proton-transfer-reaction mass spectrometry, Mass Spectrom. Rev. 26 (2007) 223–257.

[4] F. Biasioli, F. Gasperi, C. Yeretzian, T.D. Märk, Ptr-ms monitoring of vocs and bvocs in food science and technology, TRAC Trends Anal. Chem. 30 (2011) 968–977, http://dx.doi.org/10.1016/j.trac.2011.03.009, biogenic Volatile Organic Compounds S.I.

[5] C.M. Bishop, Pattern Recognition and Machine Learning (Information Science and Statistics), Springer-Verlag, Berlin, Heidelberg, 2006.

[6] I. Goodfellow, Y. Bengio, A. Courville, Deep Learning, MIT Press, 2016, http://www.deeplearningbook.org.

[7] L. Grinsztajn, E. Oyallon, G. Varoquaux, Why do tree-based models still outperform deep learning on typical tabular data? Adv. Neural Inf. Process. Syst. 35 (2022) 507–520.

[8] T. Chen, C. Guestrin, Xgboost: A scalable tree boosting system, in: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '16, Association for Computing Machinery, New York, NY, USA, 2016, pp. 785–794, http://dx.doi.org/10.1145/2939672.2939785.

[9] J. Liu, R. Zhang, J. Xiong, Machine learning approach for estimating the human-related voc emissions in a university classroom, Build. Simul. 16 (2023) 915–925.

[10] J. Li, Y. Zhang, Q. Chen, Z. Pan, J. Chen, M. Sun, J. Wang, Y. Li, Q. Ye, Development and validation of a screening model for lung cancer using machine learning: A large-scale, multi-center study of biomarkers in breath, Front. Oncol. 12 (2022) 975563.

[11] A.Z. Temerdashev, E.M. Gashimova, V.A. Porkhanov, I.S. Polyakov, D.V. Perunov, E.V. Dmitrieva, Non-invasive lung cancer diagnostics through metabolites in exhaled breath: influence of the disease variability and comorbidities, Metabolites 13 (2023) 203.

[12] Q. Kan, L. Cao, L. He, P. Wang, G. Deng, J. Li, J. Fu, Q. Huang, C.-T. Ho, Y. Li, C. Xie, Y. Cao, L. Wen, Tracing the change of the volatile compounds of soy sauce at different fermentation times by ptr-tof-ms, e-nose and gc–ms, Food Chem.: X 25 (2025) 102002, http://dx.doi.org/10.1016/j.fochx.2024.102002.

[13] J.H. Friedman, Greedy function approximation: a gradient boosting machine, Ann. Stat. (2001) 1189–1232.

[14] L. Breiman, Random forests, Mach. Learn. 45 (2001) 5–32.

[15] M. Kuhn, K. Johnson, Applied Predictive Modeling, Springer, New York, 2013, http://dx.doi.org/10.1007/978-1-4614-6849-3.

[16] Kaggle, State of data science and machine learning 2021, 2021, URL: https://www.kaggle.com/kaggle-survey-2021. (Accessed 27 January 2025).

[17] Y. Guo, T. Hastie, R. Tibshirani, Regularized linear discriminant analysis and its application in microarrays, Biostatistics 8 (2007) 86–100.

[18] M. Barker, W. Rayens, Partial least squares for discrimination, J. Chemom.: A J. Chemom. Soc. 17 (2003) 166–173.

[19] C. Cortes, V. Vapnik, Support-vector networks, Mach. Learn. 20 (1995) 273–297, http://dx.doi.org/10.1023/A:1022627411411.

[20] B. Efron, T. Hastie, I. Johnstone, R. Tibshirani, Least angle regression, Ann. Statist. 32 (2004) 407–451.

[21] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, E. Duchesnay, Scikit-learn: Machine learning in Python, J. Mach. Learn. Res. 12 (2011) 2825–2830.

[22] L. Cappellin, F. Biasioli, P.M. Granitto, E. Schuhfried, C. Soukoulis, F. Costa, T.D. Märk, F. Gasperi, On data analysis in ptr-tof-ms: From raw spectra to data mining, Sensors Actuators B: Chem. 155 (2011) 183–190, http://dx.doi.org/10.1016/j.snb.2010.11.044.

[23] I. Guyon, J. Weston, S. Barnhill, V. Vapnik, Gene selection for cancer classification using support vector machines, Mach. Learn. 46 (2002) 389–422.

[24] P.M. Granitto, C. Furlanello, F. Biasioli, F. Gasperi, Recursive feature elimination with random forest for ptr-ms analysis of agroindustrial products, Chemometr. Intell. Lab. Syst. 83 (2006) 83–90, http://dx.doi.org/10.1016/j.chemolab.2006.01.007.

[25] L. Yang, A. Shami, On hyperparameter optimization of machine learning algorithms: Theory and practice, Neurocomputing 415 (2020) 295–316, http://dx.doi.org/10.1016/j.neucom.2020.07.061.

[26] J. Bergstra, Y. Bengio, Random search for hyper-parameter optimization, J. Mach. Learn. Res. 13 (2012) 281–305.

[27] J. Snoek, H. Larochelle, R.P. Adams, Practical bayesian optimization of machine learning algorithms, in: F. Pereira, C. Burges, L. Bottou, K. Weinberger (Eds.), in: Advances in Neural Information Processing Systems, vol. 25, 2012.

[28] J. Demšar, Statistical comparisons of classifiers over multiple data sets, J. Mach. Learn. Res. 7 (2006) 1–30.

[29] W. Webber, A. Moffat, J. Zobel, A similarity measure for indefinite rankings, ACM Trans. Inf. Syst. 28 (2010) http://dx.doi.org/10.1145/1852102.1852106.

[30] S. Yener, J.A. Sánchez-López, P.M. Granitto, L. Cappellin, T.D. Märk, R. Zimmermann, G.K. Bonn, C. Yeretzian, F. Biasioli, Rapid and direct volatile compound profiling of black and green teas (camellia sinensis) from different countries with ptr-tof-ms, Talanta 152 (2016) 45–53, http://dx.doi.org/10.1016/j.talanta.2016.01.050.

[31] A. Telagathoti, M. Probst, I. Khomenko, F. Biasioli, U. Peintner, High-throughput volatilome fingerprint using ptr–tof–ms shows species-specific patterns in mortierella and closely related genera, J. Fungi 7 (2021) http://dx.doi.org/10.3390/jof7010066.

[32] I. Khomenko, V. Ting, F. Brambilla, M. Perbellini, L. Cappellin, F. Biasioli, Ptr-tof-ms voc profiling of raw and cooked gilthead sea bream fillet (sparus aurata): Effect of rearing system, season, and geographical origin, Molecules 30 (2025) http://dx.doi.org/10.3390/molecules30020402.

[33] I. Khomenko, M. Pedrotti, E. Betta, D. Cliceri, I. Endrizzi, E. Aprea, F. Gasperi, F. Biasioli, Integrated approach for the evaluation of food loss and waste of fresh spinach during its storage, in: 8th MS Food Day, Torre Canne (BR), October 16–18, 2024, IT, 2024, pp. 207–208.

[34] J.S. del Pulgar, C. Soukoulis, F. Biasioli, L. Cappellin, C. García, F. Gasperi, P. Granitto, T.D. Märk, E. Piasentier, E. Schuhfried, Rapid characterization of dry cured ham produced following different pdos by proton transfer reaction time of flight mass spectrometry (ptr-tof-ms), Talanta 85 (2011) 386–393, http://dx.doi.org/10.1016/j.talanta.2011.03.077.

[35] S. Rajendran, I. Khomenko, P. Silcock, E. Betta, M. Pedrotti, F. Biasioli, P. Bremer, The effect of different medium compositions and lab strains on fermentation volatile organic compounds (vocs) analysed by proton transfer reaction-time of flight-mass spectrometry (ptr-tof-ms), Fermentation 10 (2024) http://dx.doi.org/10.3390/fermentation10060317.

[36] S. Yener, A. Romano, L. Cappellin, P.M. Granitto, E. Aprea, L. Navarini, T.D. Märk, F. Gasperi, F. Biasioli, Tracing coffee origin by direct injection headspace analysis with ptr/sri-ms, Food Res. Int. 69 (2015) 235–243, http://dx.doi.org/10.1016/j.foodres.2014.12.046.

[37] M. Mazzucotelli, I. Khomenko, P. Franceschi, B. Farneti, E. Betta, E. Gabetti, L. Falchero, A. Cavallero, E. Aprea, F. Biasioli, Characterization of hazelnut volatilome evolution during roasting by ptr-tof-ms, gc-ims, gc-ms and advanced data mining methods, in: Contributions 9th International Conference on Proton Transfer Reaction Mass Spectrometry and Its Applications, Innsbruck University Press, 2024, pp. 174–176.