

Article

Not peer-reviewed version

---

# Machine Learning in Sensory Analysis of Mead. Case Study: Ensembles of Classifiers

---

[Krzysztof Przybył](#), [Daria Cicha-Wojciechowicz](#), [Natalia Drabińska](#), [Małgorzata Anna Majcher](#)\*

Posted Date: 10 July 2025

doi: 10.20944/preprints202507.0872.v1

Keywords: machine learning; ensembles of classifiers; mead aroma; sensory analysis; odor-active compounds



Preprints.org is a free multidisciplinary platform providing preprint service that is dedicated to making early versions of research outputs permanently available and citable. Preprints posted at Preprints.org appear in Web of Science, Crossref, Google Scholar, Scilit, Europe PMC.

Copyright: This open access article is published under a Creative Commons CC BY 4.0 license, which permit the free download, distribution, and reuse, provided that the author and preprint are cited in any reuse.

*Article*

# Machine Learning in Sensory Analysis of Mead. Case Study: Ensembles of Classifiers

Krzysztof Przybył, Daria Cicha-Wojciechowicz, Natalia Drabińska and Małgorzata Anna Majcher \*

Faculty of Food Science and Nutrition, Poznań University of Life Sciences, Wojska Polskiego 31, 60-624 Poznań, Poland

\* Correspondence: malgorzata.majcher@up.poznan.pl

## Abstract

This research focuses on the application of machine learning to the sensory analysis of mead, which opens up new possibilities in its classification and understanding. The aim was to use machine learning algorithms to classify mead types based on their sensory analysis. Machine learning algorithms such as Random Forest (RF), Adaptive Boosting (AdaBoost), Bootstrap Aggregating (Bagging), K-Nearest Neighbours (KNN) and Decision Tree (DT) were used in the analysis of chemical and sensory datasets. The Random Forest and K-Nearest Neighbours (KNN) algorithms were found to be the most effective in mead recognition, obtaining the highest scores. In contrast, the AdaBoost algorithm consistently produced the lowest accuracy results. Nonetheless, the Decision Tree algorithm achieved the highest accuracy value (0.909), indicating its potential in accurate classification based on aroma characteristics. The results suggest that the choice of an appropriate classification model can significantly affect the performance of the mead identification process in practical applications. Machine learning offers new opportunities in optimising mead production processes. The application of machine learning in the sensory analysis of mead is important for accurate classification, a better understanding of the factors affecting quality and the optimization of the production processes of this beverage, contributing to the development of interdisciplinary food research.

**Keywords:** machine learning; ensembles of classifiers; mead aroma; sensory analysis; odor-active compounds

## 1. Introduction

Honey has been a valued food product for centuries, renowned for its exceptional taste, aroma and health benefits. As the only available sweetener, honey has been an essential part of the diet since the beginning of *Homo sapiens*[1,2]. Its diversity is due to different nectar sources, natural environments, production techniques and fermentation processes. Depending on the type of plant from which bees collect nectar, honey can take on other colours, flavours and aromas, making it a unique product in each region. The processes of fermentation and storage of honey affect its final properties, including texture, shelf life and nutritional content. The main alcoholic beverage prepared from honey is mead, known for its unique and distinctive aroma and taste. The literature points to a number of research efforts to better understand and classify mead. The research by the team of Cicha-Wojciechowicz et al. (2024) focused on the effects of fermentation techniques and honey type on mead sensory profile [3], while the team of Pereira et al. (2019) studied the composition of volatile aromatic compounds [4]. Furthermore, Starowicz and Granvogl (2022) analyzed the effect of heat treatment on wort aroma changes[5], which is an important step in understanding the technological processes that influence the quality of this product. These research efforts help to know how different factors in

different types of honey can affect the sensory and chemical properties of mead. Nowadays, approaches using modern analytical technologies and machine learning methods are becoming crucial in the systematic analysis of honey.

The application of machine learning methods to the analysis of mead offers new possibilities for classifying honey types [6–8], identifying aromatic compounds, and conducting future experiments in the optimization of production processes. Machine learning algorithms such as classifiers based on supervised learning such as Random Forest (RF) [9,10], Adaptive Boosting (AdaBoost), Bootstrap Aggregating (Bagging), K-Nearest Neighbours (KNN) and Decision Tree (DT) [11–13] among others, can be applied to the analysis of chemical and sensory data sets [14,15]. DTs are some of the most widely used algorithms in data analysis, which allow the creation of simple yet explainable decision models. With DT, it is possible to both classify and predict the properties of food products based on the various descriptors identified in a given research question. In a transparent and easy to interpret way [16], DTs can be useful in the data analysis process, where the result must be understandable and practical for DTs to serve as a tool for implementation in the production process. In combination with other methods, such as RF or AdaBoost, decision trees form so-called ensembles of classifiers. This approach helps to improve the efficiency of classification and can be a crucial tool for improving mead production processes, assessing mead quality and distinguishing between different types of mead based on sensory and chemical data characteristics.

In order to improve the performance of the models, an optimization technique was applied by tuning the hyperparameters using the GridSearch method, which makes it possible to effectively identify relevant features of the data, even in the case of high complexity. This type of optimization not only improves the generalizability of the model, but also significantly increases its efficiency. As pointed out by Liu et al. (2021) [17], machine learning algorithms can exhibit poor performance due to inappropriate selection of hyperparameters. Optimization of hyperparameters becomes crucial for achieving better model performance [18].

The aim of this research was to investigate the possibility of using machine learning to classify mead types based on their sensory analysis, in particularly employing cluster map and k-means methods to identify relationships between mead groups based on their sensory characteristics. This approach will not only allow an accurate classification of mead, but also improve understanding of the factors that influence its quality.

This research contributes to the development of interdisciplinary research on food products, combining advanced methods of chemical analysis, sensory evaluation and artificial intelligence.

## 2. Results and Discussion

### 2.1. Hierarchical Analysis of Mead Aroma Compounds

A cluster map of odor related characteristics and odor compound concentration characteristics was generated (Figure 1). It was found that one variable had the highest value relative to the others. This was the odor sensory attribute, General Odor Intensity, which was the focus of the next stage of the experiments. Based on this analysis, combinations were made to create learning sets and then a model mead classification process. It was also observed that for the 'buckwheat' type of mead, in addition to General Odor Intensity, sensory attributes such as Malty, Rum, Honey, Fermented and Alcoholic achieved high scores. For comparison, the sensory attributes listed for the other types of tilia and acacia mead were also higher for buckwheat honey than for the other attributes. This means that these sensory aroma trait variables were more dominant than the aroma compound concentration traits. This made it possible to accurately understand and prepare classification models for the selected aroma-sensory attributes. It was also found that the dominant decision class for mead would be the variable responsible for the type of buckwheat.

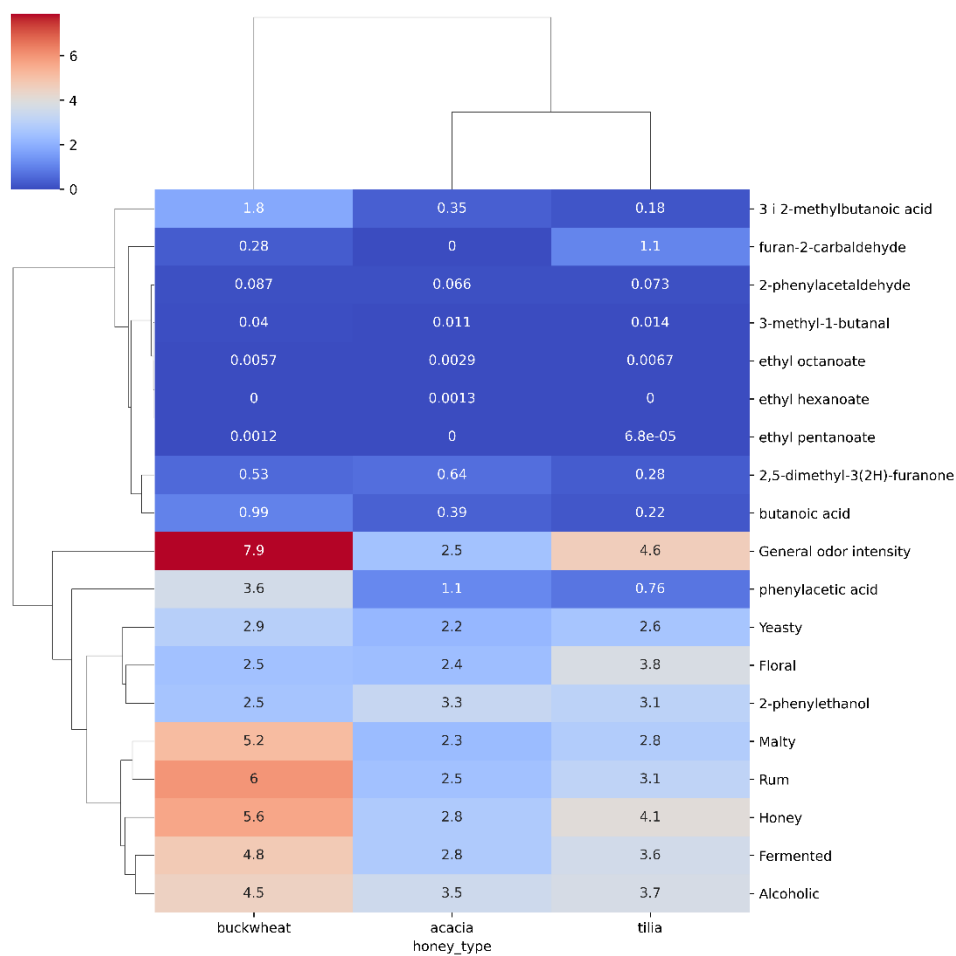
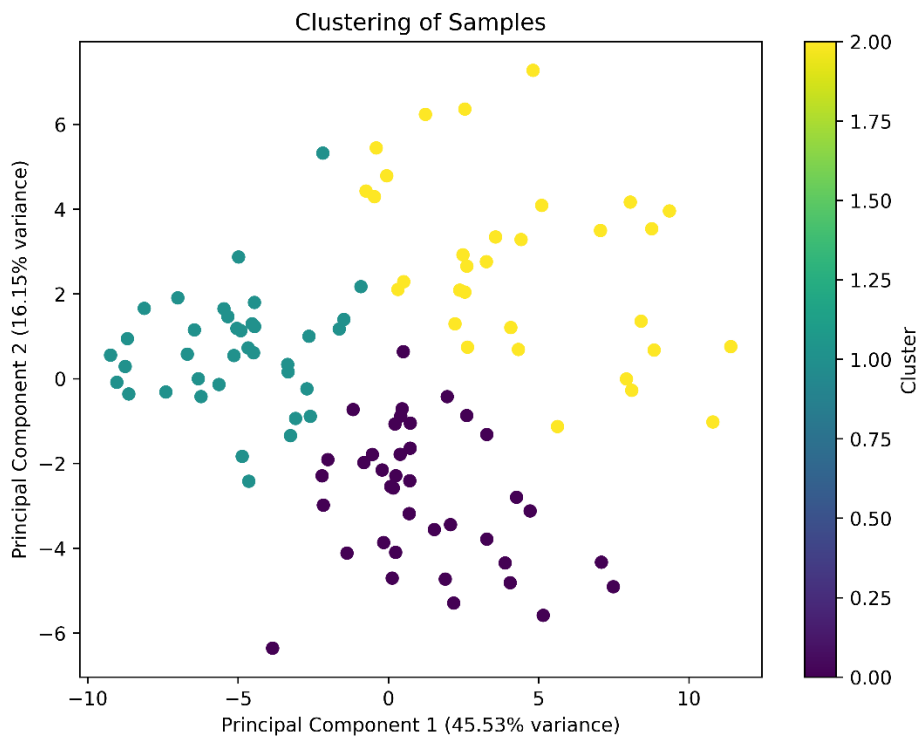


Figure 1. Clustermap of odor descriptors against compound concentrations.

2.2. Interpretation of K-Means of Honey Aroma Compounds

In the next stage of research, K-means analysis (Figure 2) was carried out between the descriptors (odor sensory attributes) identified from the clustermap patterns. In Table 1, it can be observed that cluster 2 had the highest value of variance for the sensor trait General odor intensity. This was confirmed from the clustermap analysis that General odor intensity strongly stands out in between cases in this cluster. It is possible that this descriptor plays a key role in identifying the type of mead. In this cluster, the General Odor Intensity interaction may facilitate the clustering of cases for other sensory attributes, which is reflected in the high values for other sensory attributes. For cluster 1, learning cases the sensory traits of mead aroma showed low intensity among the characteristics. This means that for this cluster 1, the learning cases do not distinguish among the characteristics. The highest variance value in this cluster 1 is also determined by the General Odor Intensity, whose mean value was 3.21. Cluster 0 gave an inter-mediate value between cluster 0 and cluster 1, suggesting that the General odor intensity descriptor was moderately expressed in the cases from this cluster. The learning cases in this cluster 0 may have been particularly characteristic of the moderate value of the aforementioned descriptor. In conclusion General Odor Intensity indeed represented an important sensory property for mead classification.



**Figure 2.** K-Means for descriptors: General\_odor\_intensity, Honey, Malty, Yeasty, Fermented, Alcoholic, Floral, Rum.

**Table 1.** Results of K-Means for descriptors: General\_odor\_intensity, Honey, Malty, Yeasty, Fermented, Alcoholic, Floral, Rum.

CLUSTER	General_odor_intensity	Honey	Malty	Yeasty	Fermented	Alcoholic	Floral	Rum
0	4.182051	4.125641	3.446154	4.164103	4.312821	4.997436	4.853846	4.164103
1	3.210811	2.435135	1.643243	0.943243	1.878378	2.097297	1.294595	1.513514
2	8.065625	6.165625	5.453125	2.468750	5.046875	4.656250	2.253125	6.203125

2.3. Machine Learning

In the analysis carried out on the basis of 21 different combinations of sensory features on the input variables classifying the mead, different machine learning algorithms were evaluated for their performance based on four key measures: accuracy (Acc), precision (Precision), sensitivity (Recall) and the F1-score (Table 2). The highest scores were obtained by the Random Forest (6 times) and KNN (6 times) algorithms. It can be concluded that these models were the most effective in classifying mead. It can be assumed that Random Forest, due to its use of multiple decision trees and random feature selection technique, had a high resistance to over-fitting. This translates into its stability and good performance in different configurations with mead based features [19]. On the other hand, KNN, based on nearest neighbour voting, also achieved high performance, especially in the evaluation of the performance metrics of this model [7,20,21]. Research shows that Random Forest, especially when combined with suitable feature extraction methods, achieves excellent results (e.g., Acc above 93%) [8,11,19,22,23]. Employing voting schemes and different distance metrics can further enhance classification performance. However, it should be noted that KNN is more sensitive to an increase in the number of features and may be less computationally efficient with large datasets.

In summary, both Random Forest and KNN can be very effective in sensory clas-sification tasks, but their advantage is due to different mechanisms of operation: RF from the strength of the tree ensemble and randomness, and KNN from local neighbour voting in feature space [22].



The rest of the machine learning algorithms such as Bagging (6 times), Naive Bayes (3 times) and Decision Tree (3 times) also performed well, but were not able to outperform Random Forest and KNN in terms of stability of results in different combinations. Bagging, a technique that involves training multiple models on different data samples, showed very robust results, but slightly less stable than the other algorithms. Naive Bayes and Decision Tree, although effective, achieved higher results, but only for selected correlations.

However, the highest metric for assessing the effectiveness of the model was obtained with Decision Tree for the sensory features General odor intensity, Yeasty, Floral, which reached an accuracy of 0.909, a precision of 0.929, a recall of 0.909 and an F1-score of 0.911. It was observed that when optimizing the algorithms with the Grid Search method for the most effective decision tree model, the value for the tree depth hyperparameter reached a score of 6. Depending on the combination of sensory features, the Grid Search method produced different results. This is due to the fact that for the Decision Tree model with data on these sensory features, the optimal tree depth was 6, whereas in other configurations, both for the Decision Tree and Random Forest models, values of less or more were obtained. The aim of this was to avoid overfitting the model, as an appropriate tree depth affects the ability of the model to generalize, and too much depth can lead to the model fitting the noise in the data rather than capturing the true relationships.

In contrast, the worst results were obtained with the AdaBoost algorithm. In fact, the AdaBoost model achieved the lowest accuracy value of 0.515 for aroma features such as General Odor Intensity, Rum and Yeasty, indicating its weaker performance compared to other algorithms. AdaBoost, which is a boosting algorithm, aims to improve the accuracy of the classifier by iteratively increasing the weights of incorrect examples and creating new weak classifiers.

Table 2. Machine learning algorithms with best hyperparameters.

No.	Descriptors	Alorithm AI	Best Hyperparameter
1	'General_odor_intensity', 'Honey', 'Malty'	<b>Random Forest</b>	<b>{'max_depth': 7, 'n_estimators': 500}</b>
		Decision Tree	{'max_depth': 5}
		AdaBoost	{'learning_rate': 0.01, 'n_estimators': 50}
		<b>Bagging</b>	<b>{'n_estimators': 100}</b>
		KNN	{'metric': 'chebyshev', 'n_neighbors': 3}
		Naive_Bayes	{'var_smoothing': 1e-09}
2	'General_odor_intensity', 'Honey', 'Fermented'	<b>Random Forest</b>	<b>{'max_depth': 6, 'n_estimators': 10}</b>
		Decision Tree	{'max_depth': 6}
		AdaBoost	{'learning_rate': 0.01, 'n_estimators': 50}
		Bagging	{'n_estimators': 50}
		KNN	{'metric': 'manhattan', 'n_neighbors': 5}
		Naive_Bayes	{'var_smoothing': 1e-09}
3	'General_odor_intensity', 'Honey', 'Rum'	Random Forest	{'max_depth': 3, 'n_estimators': 10}
		Decision Tree	{'max_depth': 5}
		AdaBoost	{'learning_rate': 0.01, 'n_estimators': 50}
		<b>Bagging</b>	<b>{'n_estimators': 200}</b>
		KNN	{'metric': 'manhattan', 'n_neighbors': 2}
		Naive_Bayes	{'var_smoothing': 1e-09}
4	'General_odor_intensity', 'Honey', 'Yeasty'	Random Forest	{'max_depth': 3, 'n_estimators': 100}
		Decision Tree	{'max_depth': 3}
		AdaBoost	{'learning_rate': 0.01, 'n_estimators': 50}
		<b>Bagging</b>	<b>{'n_estimators': 200}</b>
		KNN	{'metric': 'chebyshev', 'n_neighbors': 3}
		Naive_Bayes	{'var_smoothing': 1e-09}

5	'General_odor_intensity', 'Honey', 'Floral'	Random Forest Decision Tree AdaBoost Bagging <b>KNN</b> Naive_Bayes	{'max_depth': 7, 'n_estimators': 50} {'max_depth': 6} {'learning_rate': 0.01, 'n_estimators': 50} {'n_estimators': 100} {'metric': 'euclidean', 'n_neighbors': 7} {'var_smoothing': 1e-09}
6	'General_odor_intensity', 'Honey', 'Alcoholic'	<b>Random Forest</b> Decision Tree AdaBoost Bagging <b>KNN</b> <b>Naive_Bayes</b>	{'max_depth': 7, 'n_estimators': 1000} {'max_depth': 5} {'learning_rate': 0.01, 'n_estimators': 100} {'n_estimators': 100} {'metric': 'minkowski', 'n_neighbors': 2} {'var_smoothing': 1e-09}
7	'General_odor_intensity', 'Malty', 'Fermented'	Random Forest Decision Tree AdaBoost Bagging <b>KNN</b> Naive_Bayes	{'max_depth': 4, 'n_estimators': 50} {'max_depth': 7} {'learning_rate': 0.01, 'n_estimators': 50} {'n_estimators': 50} {'metric': 'manhattan', 'n_neighbors': 3} {'var_smoothing': 1e-09}
8	'General_odor_intensity', 'Malty', 'Rum'	Random Forest <b>Decision Tree</b> AdaBoost Bagging KNN Naive_Bayes	{'max_depth': 7, 'n_estimators': 10} {'max_depth': 5} {'learning_rate': 0.1, 'n_estimators': 50} {'n_estimators': 100} {'metric': 'manhattan', 'n_neighbors': 5} {'var_smoothing': 1e-09}
9	'General_odor_intensity', 'Malty', 'Yeasty'	Random Forest Decision Tree AdaBoost Bagging <b>KNN</b> Naive_Bayes	{'max_depth': 7, 'n_estimators': 50} {'max_depth': 6} {'learning_rate': 0.01, 'n_estimators': 50} {'n_estimators': 200} {'metric': 'minkowski', 'n_neighbors': 3} {'var_smoothing': 1e-09}
10	'General_odor_intensity', 'Malty', 'Floral'	Random Forest <b>Decision Tree</b> AdaBoost Bagging KNN Naive_Bayes	{'max_depth': 6, 'n_estimators': 10} {'max_depth': 4} {'learning_rate': 0.01, 'n_estimators': 50} {'n_estimators': 50} {'metric': 'chebyshev', 'n_neighbors': 3} {'var_smoothing': 1e-09}
11	'General_odor_intensity', 'Malty', 'Alcoholic'	Random Forest Decision Tree AdaBoost <b>Bagging</b> <b>KNN</b> Naive_Bayes	{'max_depth': 6, 'n_estimators': 10} {'max_depth': 4} {'learning_rate': 0.01, 'n_estimators': 50} {'n_estimators': 50} {'metric': 'manhattan', 'n_neighbors': 3} {'var_smoothing': 1e-09}
12	'General_odor_intensity', 'Fermented', 'Rum'	<b>Random Forest</b> Decision Tree AdaBoost Bagging KNN	{'max_depth': 5, 'n_estimators': 50} {'max_depth': 6} {'learning_rate': 0.1, 'n_estimators': 500} {'n_estimators': 100} {'metric': 'euclidean', 'n_neighbors': 2}

13	'General_odor_intensity', 'Fermented', 'Yeasty'	Naive_Bayes	{'var_smoothing': 1e-09}
		Random Forest	{'max_depth': 4, 'n_estimators': 10}
		Decision Tree	{'max_depth': 6}
		AdaBoost	{'learning_rate': 0.01, 'n_estimators': 50}
		<b>Bagging</b>	{'n_estimators': 50}
		KNN	{'metric': 'euclidean', 'n_neighbors': 5}
14	'General_odor_intensity', 'Fermented', 'Floral'	Naive_Bayes	{'var_smoothing': 1e-09}
		<b>Random Forest</b>	{'max_depth': 6, 'n_estimators': 100}
		Decision Tree	{'max_depth': 3}
		AdaBoost	{'learning_rate': 0.01, 'n_estimators': 50}
		Bagging	{'n_estimators': 100}
		KNN	{'metric': 'manhattan', 'n_neighbors': 5}
15	'General_odor_intensity', 'Fermented', 'Alcoholic'	Naive_Bayes	{'var_smoothing': 1e-09}
		Random Forest	{'max_depth': 3, 'n_estimators': 100}
		Decision Tree	{'max_depth': 3}
		AdaBoost	{'learning_rate': 0.01, 'n_estimators': 100}
		Bagging	{'n_estimators': 50}
		KNN	{'metric': 'euclidean', 'n_neighbors': 2}
16	'General_odor_intensity', 'Rum', 'Yeasty'	<b>Naive_Bayes</b>	{'var_smoothing': 1e-09}
		Random Forest	{'max_depth': 7, 'n_estimators': 50}
		Decision Tree	{'max_depth': 7}
		AdaBoost	{'learning_rate': 1.0, 'n_estimators': 50}
		Bagging	{'n_estimators': 100}
		KNN	{'metric': 'manhattan', 'n_neighbors': 5}
17	'General_odor_intensity', 'Rum', 'Floral'	Naive_Bayes	{'var_smoothing': 1e-09}
		Random Forest	{'max_depth': 4, 'n_estimators': 500}
		Decision Tree	{'max_depth': 3}
		AdaBoost	{'learning_rate': 0.1, 'n_estimators': 500}
		<b>Bagging</b>	{'n_estimators': 100}
		KNN	{'metric': 'chebyshev', 'n_neighbors': 2}
18	'General_odor_intensity', 'Rum', 'Alcoholic'	Naive_Bayes	{'var_smoothing': 1e-09}
		Random Forest	{'max_depth': 5, 'n_estimators': 100}
		Decision Tree	{'max_depth': 3}
		AdaBoost	{'learning_rate': 0.01, 'n_estimators': 50}
		Bagging	{'n_estimators': 100}
		KNN	{'metric': 'euclidean', 'n_neighbors': 3}
19	'General_odor_intensity', 'Yeasty', 'Floral'	Naive_Bayes	{'var_smoothing': 1e-09}
		Random Forest	{'max_depth': 6, 'n_estimators': 100}
		<b>Decision Tree</b>	{'max_depth': 6}
		AdaBoost	{'learning_rate': 0.01, 'n_estimators': 50}
		Bagging	{'n_estimators': 50}
		KNN	{'metric': 'chebyshev', 'n_neighbors': 3}
20	'General_odor_intensity', 'Yeasty', 'Alcoholic'	Naive_Bayes	{'var_smoothing': 1e-09}
		<b>Random Forest</b>	{'max_depth': 3, 'n_estimators': 50}
		Decision Tree	{'max_depth': 3}
		AdaBoost	{'learning_rate': 0.1, 'n_estimators': 50}
		Bagging	{'n_estimators': 50}
		KNN	{'metric': 'chebyshev', 'n_neighbors': 2}



21	'General_odor_intensity', 'Floral', 'Alcoholic'	Random Forest Decision Tree AdaBoost Bagging KNN Naive_Bayes	{'max_depth': 7, 'n_estimators': 10} {'max_depth': 3} {'learning_rate': 0.01, 'n_estimators': 100} {'n_estimators': 50} {'metric': 'manhattan', 'n_neighbors': 3} {'var_smoothing': 1e-09}
----	----------------------------------------------------	-----------------------------------------------------------------------------	-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

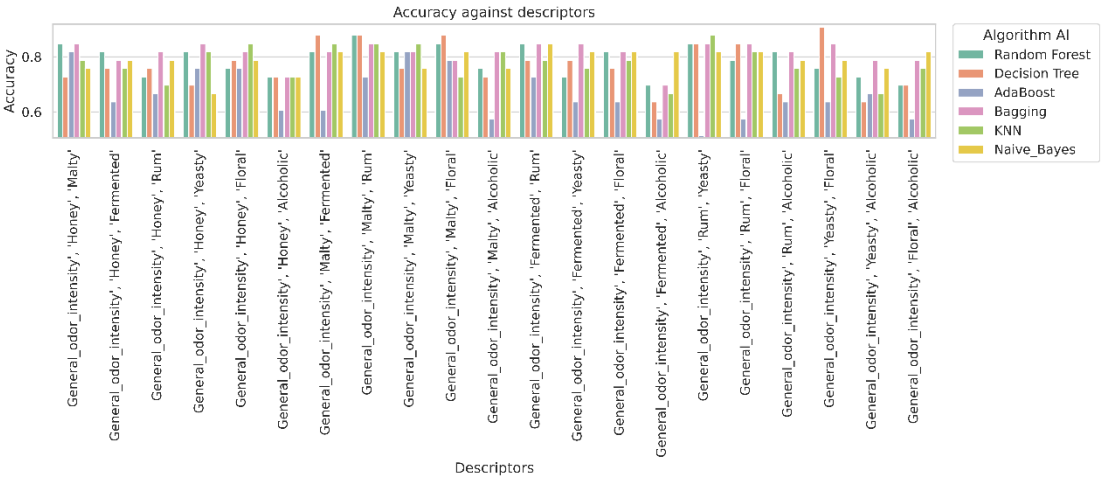


Figure 3. Impact of Descriptors on Algorithm Accuracy.

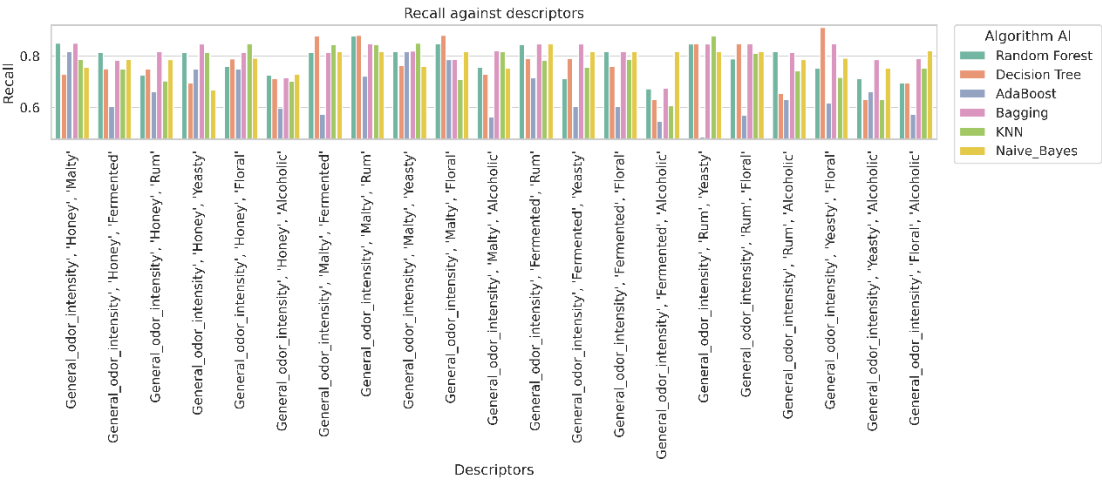


Figure 4. Impact of Descriptors on Algorithm Recall.

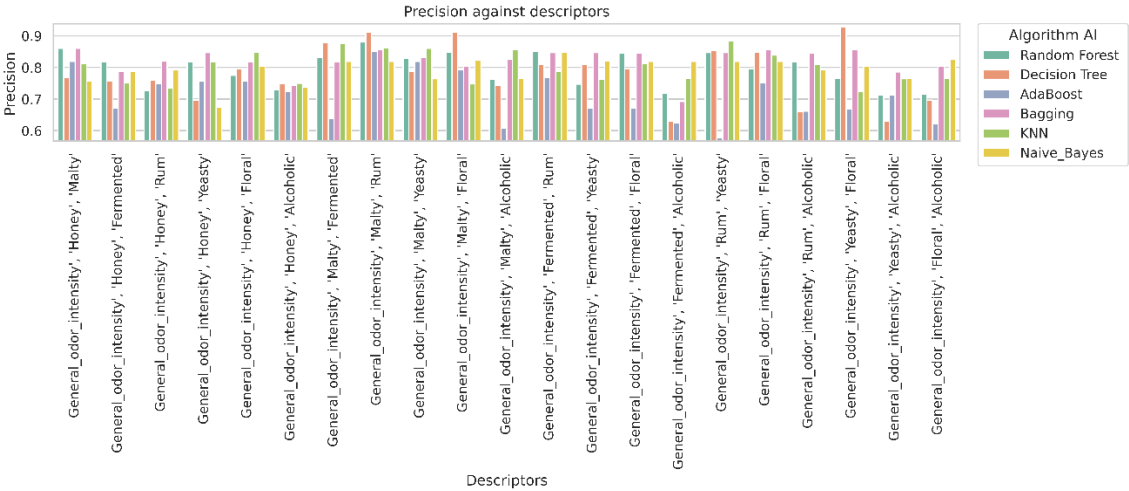


Figure 5. Impact of Descriptors on Algorithm Precision.

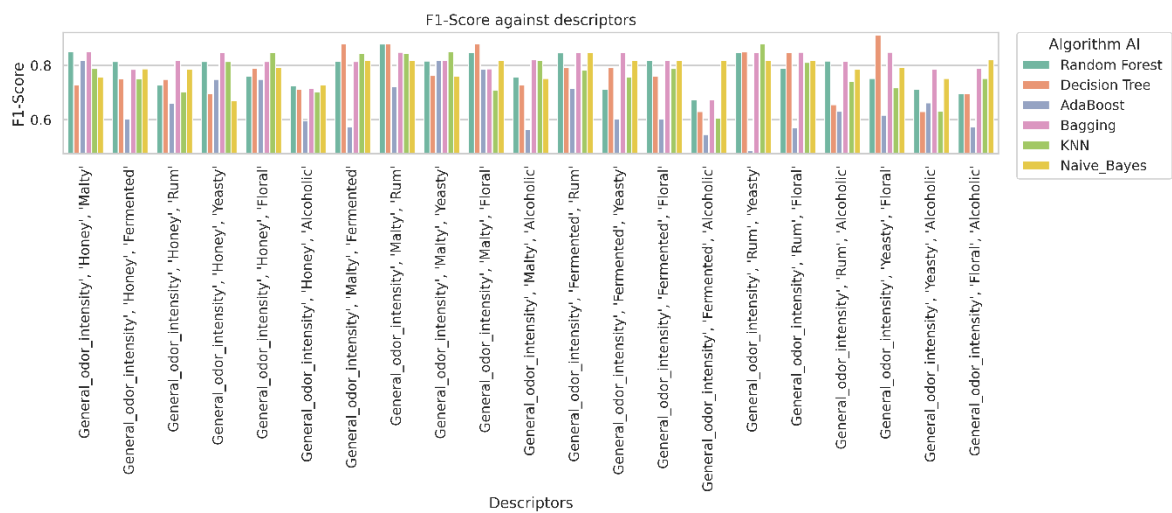


Figure 6. Impact of Descriptors on Algorithm F1-Score.

In Figure 3, it is observed that the Random Forest and K-Nearest Neighbours (KNN) algorithms generally achieve the highest accuracy results in classifying mead types. In Figure 4, the Random Forest and KNN algorithms also demonstrate high performance in terms of recall rate. Overall, these algorithms also provide high precision (Figure 5) scores and F1-Score (Figure 6). Although it is worth noting that for specific sensory characteristics of the mead, the Decision Tree algorithm obtained the highest single value for accuracy, recall, precision, and F1-Score. The results clearly indicate that choosing an appropriate classification model significantly impacts the performance of the mead identification process in practical applications. Additionally, selecting suitable sensory descriptors is crucial for achieving optimal mead identification performance.

When tuning the models using the K-NN algorithm, it was observed that the ‘Manhattan’ hyperparameter was the most frequently selected. The reason for this is that it gives a better assessment of model performance and stability [24]. In the case of the Random Forest algorithm, as in the case of the Decision Tree algorithm, the choice of hyperparameter was influenced by the avoidance of over-fitting and, at the same time, the goal of obtaining high model performance. In the case of the Naive Bayes model, the Grid Search method determined the exact optimal value of var\_smoothing equal to 1e-09 to improve the stability of the model, as well as its generalizability, avoiding overfitting problems or computational errors associated with zero variance. In summary, the selection of different hyperparameter values using different machine learning algorithms made it possible to test on the basis of which sensory characteristics the model effectively classifies the type of mead. In light of the above and observations in the literature, it is worth investigating the selection of hyperparameters suitable for a specific task [25]. In the age of modern machine learning and deep learning tools, it allows that optimization through Grid Search has become rapid. Observations and literature show that when choosing an optimization technique, it is also worth paying attention to the size of the dataset for a given problem [26].

2.4. Analysis of Classifier Performance Based on Confusion Matrix Results

In machine learning, confusion matrices are an important tool for evaluating the performance of classifiers. Confusion matrices allow a detailed analysis of the predictive performance for a test set to be performed. It has also been observed in the literature that it is the most commonly used visualization to present among others this in-formation [27–31]. It was found that it is usually presented to represent a single model. In our study, the recognition behavior of the selected mead type was also compared between models, which is crucial for selecting the appropriate model in a future implementation in an application [11,27].

In this experiment, the number of correct and incorrect classifications of mead in-to a particular class (buckwheat, acacia, tilia) was accurately determined. This analysis led to an understanding of the extent to which the model correctly classified learning instances related to aroma characteristics and where it went wrong. The confusion matrix also made it possible to identify specific difficulties that may have occurred with certain types of mead. Such a solution will be helpful in optimizing the model, which would lead to an assessment of the overall performance of the classifier, while attempting to improve these algorithms. A total of 126 confusion matrix plots were generated (Figure: S1-S126), corresponding to 21 combinations of sensory odor features generated from the similarity map analysis and the K-means method. Each combination contained 6 models and for each model a confusion matrix was plotted for the test set. In each matrix the class number corresponded to the type of mead, i.e. '1' was acacia, '2' was buckwheat and '3' was tilia. In order to explicitly identify individual types of mead, the classification analysis of the attributes 'General\_odor\_intensity', 'Honey' and 'Malty' from Table 1, Random Forest (Figure S1) and Bagging (Figure S4) successfully classified 11 cases for acacia and at the same time made an error in 1 case by assigning tilia to mead on the test set. In the class corresponding to buck-wheat-based mead, 8 cases were classified in the class corresponding to this buck-wheat, while 2 cases were incorrectly classified in the class corresponding to tilia. In the case of the last type of honey, 9 cases were clearly classified in favor of tilia honey, while only 2 cases were incorrectly classified in favor of tilia honey. In the second case (Table 3), for the characteristics 'general\_odor\_intensity', 'honey', 'fermented', the Random Forest model was the most successful, classifying acacia honey in 11 cases, buckwheat-based mead in 9 cases and tilia honey in 7 cases (Figure S7). In the third case (Table 3), considering the features 'general\_odor\_intensity', 'honey', 'rum', the Bagging model was the one that correctly classified acacia honey in 11 cases, buck-wheat-based mead in 9 cases and tilia honey in 8 cases (Figure S16). The analysis of the confusion matrix between the models and the selection of sensory attributes showed that the algorithms were more successful in classifying acacia honey. In the case of buckwheat and tilia honey, this depended on the choice of model and the sensory characteristics of the mead. Considering the most effective decision tree model for the sensory attributes 'General\_Odor\_Intensity', 'Yeasty', 'Floral' (Figure S109-S114), the confusion matrix on the test set correctly classified acacia honey in 10 cases, buckwheat-based mead in 9 cases and lime tree honey in 11 cases (Figure S110). In fact, it can be seen that the difficulty in classifying mead is due to the type of mead, i.e. tilia honey, where in case 19 (Table 3) tilia honey was classified more correctly than acacia honey. In the group of characteristics related to General\_Odor\_Intensity, Rum, Yeasty, where the AdaBoost model performed the worst (Figure S93), the KNN algorithm proved to be the most successful model correctly classifying tilia mead in 11 cases and buckwheat and tilia mead in 9 cases (Figure S95).

### 3. Materials and Methods

#### 3.1. Data Source

For the current research, data from a previously published study were used, including a quantitative olfactory profile analysis and the quantitation of important odor-active compounds for twelve types of mead. These meads were produced using three different honey varieties (acacia, buckwheat, and tilia) and three fermentation methods: spontaneous fermentation, inoculation with *Saccharomyces cerevisiae* yeast, and inoculation with *Galactomyces Geotrichum* molds. Additionally, both heated and unheated wort variations were analyzed. The methods used for sample analysis were thoroughly described before [3].

#### 3.2. Data Collection

The research examined a series of sensory attributes: 'general odor intensity', 'honey', 'malty', 'fermented', 'rum', 'yeasty', 'floral', 'alcoholic'. A total of these attributes were created. This process facilitated the creation of multiple input variables sets used for training the cluster models. The

combinations aimed to deepen the under-standing of relationships between the different sensory characteristics and potentially enhance the performance of the algorithms. By varying these patterns, it was possible to better capture important features that influence the accuracy of mead classification. As a result, 21 learning sets were designed for which the input variables were 3 sensory characteristics. The output variable in each set was the type of honey (acacia, talia, backwheat). The dataset consisted of 108 learning cases. The `train_test_split` function was used to split the set into a training set and a test set. This means that 30% of the learning cases belonged to the test set and 70% of the learning cases belonged to the training set. The selection of learning cases from the training set was done by the aforementioned function, which automatically decides which cases should go into the learning set and which into the test set. The only criterion set by the user is the proportion that determines how much data is included in the training set and how much in the test set. A random seed (`random_state`) was also specified to ensure reproducibility of the distribution, allowing results to be replicated in future experiments.

3.3. Parameter Selection for Classification Models

In the design of algorithms using ensembles of classifiers, an appropriate architecture was developed for each model (Table 3), which was then used to the classify the different types of mead. In addition, during the learning process, the selection of the hyperparameters of each model was automated in order to obtain only the best model in terms of the evaluation metrics. In this way, it was possible to optimize the performance of each classifier to obtain the best possible results in the classification of mead types. The automated hyperparameter selection process also ensured greater consistency and repeatability of results, eliminating the risk of errors due to manual calibration.

**Table 3.** The structure of hyperparameters in tuning algorithms for ensembles of classifiers.

Machine learning algorithm type	Hyperparameters used	Value
DecisionTree	max_depth	3,4,5,6,7
RandomForest	max_depth	3,4,5,6,7
RandomForest	n_estimators	10, 50, 100, 200, 500, 1000
AdaBoost	n_estimators	50, 100, 200, 500, 1000
AdaBoost	learning_rate	0.01, 0.1, 1.0
KNN	n_neighbors	2,3,5,7
KNN	metric	euclidean, manhattan, chebyshev, minkowski
KNN	weights	'uniform', 'distance'
Bagging	n_estimators	50,100,200
Naive_Bayes	var_smoothing	1e-9, 1e-8, 1e-7, 1e-6, 1e-5

In view of the fact that machine learning algorithms often contain several hyperparameters, this study used the optimization of selected hyperparameters to assess the performance of the models [32]. Within the Random Forest model, a set of hyperparameters was selected that included different values for the maximum tree depth (`max_depth`) and the number of estimators (`n_estimators`). This allowed different configurations to be tested and the optimum balance between accuracy and model complexity to be found.

In the research question for decision trees, only the maximum tree depth (`max_depth`) was considered as a hyperparameter, with specific values tested (Table 1). These values were intended to strike a balance between the complexity of the tree and its ability to extract meaningful patterns from the data.

Two important variables were introduced for the AdaBoost model: the number of estimators (`n_estimators`) and the learning rate (`learning_rate`). These values were tested over a wide range to optimise the learning rate and the number of weak classifiers in the ensemble.

The K-Nearest Neighbours algorithm used a more complex parametric grid, including the number of neighbours (`n_neighbours`), the distance measure (`metric`) and the way the neighbours were weighted (`weights`) (Table 3). Each of these parameters affected classification accuracy and computation time, so different values were tested to select the best settings.

Bagging (`param_grid_bag`) considered the parameter `n_estimators`, which determines the number of estimators in the ensemble. Three different values were tested (Table 1) in order to find the optimal number of submodels that best influenced the classification performance.

In the case of the Naive Bayes model, the `var_smoothing` hyperparameter was used, which was responsible for adding a small value to the variance of the features to avoid the problem of dividing by zero. The tested values (Table 3) allowed to check the effect of this parameter on the stability and accuracy of the model.

### 3.4. Model Training and Testing

In the next step, the learning process of the designed ensembles of classifiers [33–35] was carried out in order to evaluate their performance in recognizing mead types. During validation on the test set, the quality of learning was assessed using evaluation metrics such as accuracy, precision, recall and F1-score. A confusion matrix was also used to analyze the predicted results. The process of model estimation and learning was carried out using Python version 3.11.11 [36–39].

### 3.5. Statistical Analysis

In the process of clustering the data using the `sklearn.cluster` library, the clustering for the cases defining the different sensory characteristics was plotted on a graph. In the next step, the PCA (Principal Component Analysis) algorithm was used to reduce the data to two principal components, allowing them to be visualized in a two-dimensional graph. The K-Means algorithm [20,40] grouped the sensory features and the results of the clustering were presented in a graph with colour-coding for different groups separately. This visualization facilitated the evaluation of the structure of the learning instances corresponding to the sensory features and allowed us to understand the preparation of learning sets based on the three sensory features.

As part of the analysis of the sensory feature data, correlations between the different features were also determined. A cluster map of the sensory features and odors was created to identify feature similarities that could influence the classification results. Based on the clustermap results, it was possible to prepare learning sets for selected odor features.

## 4. Conclusions

Within the scope of our research, we identified mead on the basis of selected sensory characteristics of aroma, which allowed us to accurately classify this type of mead. By analyzing the performance of different classification algorithms, we found that Random Forest and K-Nearest Neighbours (KNN) algorithms proved to be the most effective models for mead recognition. Both models showed high performance, but it was the Decision Tree algorithm that achieved the highest accuracy value, suggesting its potential for accurate classification based on odor features. These results indicate that the choice of an appropriate classification model can significantly affect the performance of the mead identification process in practical applications. Analysis of the confusion matrix also showed that acacia honey was more easily identified by the algorithms than tilia or buckwheat mead. It is worth noting that the high performance of the decision tree algorithm influenced the fact that the number of classified cases using the confusion matrix method was higher for lime honey than for acacia honey. The highest model performance result was obtained using the decision tree for odor sensory attributes such as General odor intensity, Yeasty, Floral, which resulted in an accuracy value of 0.909, a precision of 0.929, a recall of 0.909 and an F1-score of 0.911.



Future research directions may focus on the optimization of mead production processes using machine learning methods, which open up new opportunities for experiments aimed at increasing the efficiency and quality of this beverage. A deeper understanding of the factors influencing the quality of mead both in terms of its chemical and sensory characteristics is also an important area for further analysis. In the future, it is also worth focusing on the selection and optimization of classification models used in practice, because, as the results so far have shown, the appropriate choice of algorithm significantly affects the effectiveness of mead identification. The difficulties in distinguishing between buckwheat and lime mead require special attention analysis of the confusion matrix has shown that they are classified less precisely than acacia honey, which may indicate the need for further research into improving the classification performance of these particular types of mead.

It is also worth highlighting the importance of developing interdisciplinary re-search that combines advanced methods of chemical analysis, sensory evaluation and artificial intelligence. This approach fosters the creation of modern research tools and the advancement of knowledge on the quality and characteristics of food products, including mead.

**Supplementary Materials:** The following supporting information can be downloaded at the website of this paper posted on Preprints.org. Figures S1-S126: Confusion matrix of the classifier ensembles calculated on the test set for all sensory attributes of the odor.

**Author Contributions:** Conceptualization, K.P., D.C.-W. and M.A.M.; methodology, K.P.; software, K.P.; validation, K.P., D.C.-W. and M.A.M.; formal analysis, K.P., D.C.-W. and N.D.; investigation, K.P.; resources, K.P. and M.A.M.; data curation, K.P., D.C.-W. and N.D.; writing—original draft preparation, K.P.; writing—review and editing, D.C.-W.; visualization, K.P.; supervision, K.P. and M.A.M.; project administration, D.C.-W. and M.A.M.; funding acquisition, M.A.M. All authors have read and agreed to the published version of the manuscript.

**Funding:** This work was supported by the National Science Centre (Poland) under grant no. UMO-2019/35/O/NZ9/00788.

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** All original data presented in this research are included in the article. Further information is available from the first author or the corresponding author upon reasonable request.

**Conflicts of Interest:** The authors declare no conflicts of interest.

## References

1. Ghosh, S.; Meyer-Rochow, V.B.; Jung, C. Honey Bees and Their Brood: A Potentially Valuable Resource of Food, Worthy of Greater Appreciation and Scientific Attention. *J Ecol Environ* 2021, 45, 31, doi:10.1186/s41610-021-00212-y.
2. Bogdanov, S. Honey as Nutrient and Functional Food : A Review. *Bee Product Science* 2015.
3. Cicha-Wojciechowicz, D.; Drabińska, N.; Majcher, M.A. Influence of Honey Varieties, Fermentation Techniques, and Production Process on Sensory Properties and Odor-Active Compounds in Meads. *Molecules* 2024, 29, 5913, doi:10.3390/MOLECULES29245913/S1.
4. Pereira, A.P.; Mendes-Ferreira, A.; Dias, L.G.; Oliveira, J.M.; Estevinho, L.M.; Mendes-Faia, A. Volatile Composition and Sensory Properties of Mead. *Microorganisms* 2019, 7, 404, doi:10.3390/microorganisms7100404.
5. Starowicz, M.; Granvogl, M. Effect of Wort Boiling on Volatiles Formation and Sensory Properties of Mead. *Molecules* 2022, 27, 710, doi:10.3390/molecules27030710.
6. Noviyanto, A.; Abdulla, W.H. Honey Botanical Origin Classification Using Hyperspectral Imaging and Machine Learning. *J Food Eng* 2020, 265, 109684, doi:10.1016/j.jfoodeng.2019.109684.



7. Ahmed, E. Detection of Honey Adulteration Using Machine Learning. *PLOS Digital Health* 2024, 3, e0000536, doi:10.1371/journal.pdig.0000536.
8. Ciulu, M.; Oertel, E.; Serra, R.; Farre, R.; Spano, N.; Caredda, M.; Malfatti, L.; Sanna, G. Classification of Unifloral Honeys from SARDINIA (Italy) by ATR-FTIR Spectroscopy and Random Forest. *Molecules* 2020, 26, 88, doi:10.3390/molecules26010088.
9. Yu, Y.; Wang, L.; Huang, H.; Yang, W. An Improved Random Forest Algorithm. *J Phys Conf Ser* 2020, 1646, 012070, doi:10.1088/1742-6596/1646/1/012070.
10. Granitto, P.M.; Gasperi, F.; Biasioli, F.; Trainotti, E.; Furlanello, C. Modern Data Mining Tools in Descriptive Sensory Analysis: A Case Study with a Random Forest Approach. *Food Qual Prefer* 2007, 18, 681–689, doi:10.1016/j.foodqual.2006.11.001.
11. Przybył, K.; Walkowiak, K.; Kowalczewski, P.L. Efficiency of Identification of Blackcurrant Powders Using Classifier Ensembles. *Foods* 2024, 13, 697, doi:10.3390/foods13050697.
12. ÇUKUR, T.; ÇUKUR, F. Determining Factors Affecting Cooperative Membership of the Beekeepers Using Decision Tree Algorithms. *Tarım Bilimleri Dergisi* 2022, 28, 25–32, doi:10.15832/ankutbd.739230.
13. Şengül, Z.; Yücel, B.; Saner, G.; Takma, Ç. Investigating the Impact of Climate Parameters on Honey Yield under Migratory Beekeeping Conditions through Decision Tree Analysis: The Case of İzmir Province. *ANADOLU Ege Tarımsal Araştırma Enstitüsü Dergisi* 2023, 33, 268–280, doi:10.18615/anadolu.1394787.
14. Chacko, R.; Jain, D.; Patwardhan, M.; Puri, A.; Karande, S.; Rai, B. Data Based Predictive Models for Odor Perception. *Sci Rep* 2020, 10, 17136, doi:10.1038/s41598-020-73978-1.
15. Nozaki, Y.; Nakamoto, T. Predictive Modeling for Odor Character of a Chemical Using Machine Learning Combined with Natural Language Processing. *PLoS One* 2018, 13, e0198475, doi:10.1371/journal.pone.0198475.
16. Przybył, K. Explainable AI: Machine Learning Interpretation in Blackcurrant Powders. *Sensors* 2024, 24, 3198, doi:10.3390/S24103198/S1.
17. Liu, Y.; Pu, H.; Sun, D.-W. Efficient Extraction of Deep Image Features Using Convolutional Neural Network (CNN) for Applications in Detecting and Analysing Complex Food Matrices. *Trends Food Sci Technol* 2021, 113, 193–204, doi:10.1016/j.tifs.2021.04.042.
18. Bischl, B.; Binder, M.; Lang, M.; Pielok, T.; Richter, J.; Coors, S.; Thomas, J.; Ullmann, T.; Becker, M.; Boulesteix, A.; et al. Hyperparameter Optimization: Foundations, Algorithms, Best Practices, and Open Challenges. *WIREs Data Mining and Knowledge Discovery* 2023, 13, doi:10.1002/widm.1484.
19. Martinez-Castillo, C.; Astray, G.; Mejuto, J.C.; Simal-Gandara, J. Random Forest, Artificial Neural Network, and Support Vector Machine Models for Honey Classification. *eFood* 2020, 1, 69–76, doi:10.2991/efood.k.191004.001.
20. Al-Mahasneh, M.; Al-U'datt, M.; Rababah, T.; Al-Widyan, M.; Abu Kaeed, A.; Al-Mahasneh, A.J.; Abu-Khalaf, N. Classification and Prediction of Bee Honey Indirect Adulteration Using Physiochemical Properties Coupled with K-Means Clustering and Simulated Annealing-Artificial Neural Networks (SA-ANNs). *J Food Qual* 2021, 2021, 1–9, doi:10.1155/2021/6634598.
21. Al-Awadhi, M.A.; Deshmukh, R.R. Honey Classification Using Hyperspectral Imaging and Machine Learning. In *Proceedings of the 2021 Smart Technologies, Communication and Robotics (STCR); IEEE, October 9 2021; pp. 1–5.*
22. Zuo, M.; Chen, X.; Sui, L. Evaluation of Machine Learning Algorithms for Classification of Visual Stimulation-Induced EEG Signals in 2D and 3D VR Videos. *Brain Sci* 2025, 15, 75, doi:10.3390/brainsci15010075.
23. Mateo, F.; Tarazona, A.; Mateo, E.M. Comparative Study of Several Machine Learning Algorithms for Classification of Unifloral Honeys. *Foods* 2021, 10, 1543, doi:10.3390/foods10071543.
24. Razavi-Termeh, S.V.; Sadeghi-Niaraki, A.; Razavi, S.; Choi, S.-M. Enhancing Flood-Prone Area Mapping: Fine-Tuning the K-Nearest Neighbors (KNN) Algorithm for Spatial Modelling. *Int J Digit Earth* 2024, 17, doi:10.1080/17538947.2024.2311325.
25. Fan, M.; Xiao, K.; Sun, L.; Zhang, S.; Xu, Y. Automated Hyperparameter Optimization of Gradient Boosting Decision Tree Approach for Gold Mineral Prospectivity Mapping in the Xiong'er-shan Area. *Minerals* 2022, 12, 1621, doi:10.3390/min12121621.

26. Lin, J.; Zhang, J. A Fast Parameters Selection Method of Support Vector Machine Based on Coarse Grid Search and Pattern Search. In Proceedings of the 2013 Fourth Global Congress on Intelligent Systems; IEEE, December 2013; pp. 77–81.
27. Pomme, L.-E.; Bourqui, R.; Giot, R.; Auber, D. Relative Confusion Matrix: Efficient Comparison of Decision Models. In Proceedings of the 2022 26th International Conference Information Visualisation (IV); IEEE, July 2022; Vol. 2022-July, pp. 98–103.
28. Khan, M.M. Enhancing Honey Quality Control: A Machine Learning-Based Approach Using Hyperspectral Imaging. *VFAST Transactions on Software Engineering* 2025, 13, 147–160, doi:10.21015/VTSE.V13I2.2130.
29. ZHOU, M.; LONG, T.; ZHAO, Z.; CHEN, J.; WU, Q.; WANG, Y.; ZOU, Z. Honey Quality Detection Based on Near-Infrared Spectroscopy. *Food Science and Technology* 2023, 43, e98822, doi:10.1590/fst.98822.
30. Saputro, A.H.; Aprichilia, C. Classification System of Honey Floral Origin Based on Visual Near-Infrared Imaging. In Proceedings of the 2019 International Conference on Sustainable Information Engineering and Technology (SIET); IEEE, September 1 2019; pp. 125–129.
31. Zhao, Z.; Chen, L.; Liu, F.; Zhou, F.; Peng, J.; Sun, M. Fast Classification of Geographical Origins of Honey Based on Laser-Induced Breakdown Spectroscopy and Multivariate Analysis. *Sensors* 2020, Vol. 20, Page 1878 2020, 20, 1878, doi:10.3390/S20071878.
32. Gomes Mantovani, R.; Horváth, T.; Rossi, A.L.D.; Cerri, R.; Barbon Junior, S.; Vanschoren, J.; Carvalho, A.C.P.L.F. de Better Trees: An Empirical Study on Hyperparameter Tuning of Classification Decision Tree Induction Algorithms. *Data Min Knowl Discov* 2024, doi:10.1007/s10618-024-01002-5.
33. Belgiu, M.; Drăguț, L. Random Forest in Remote Sensing: A Review of Applications and Future Directions. *ISPRS Journal of Photogrammetry and Remote Sensing* 2016, 114, 24–31, doi:10.1016/j.isprsjprs.2016.01.011.
34. Kumari, S.; Kumar, D.; Mittal, M. An Ensemble Approach for Classification and Prediction of Diabetes Mellitus Using Soft Voting Classifier. *International Journal of Cognitive Computing in Engineering* 2021, 2, 40–46, doi:10.1016/j.ijcce.2021.01.001.
35. Catal, C.; Tufekci, S.; Pirmitt, E.; Kocabag, G. On the Use of Ensemble of Classifiers for Accelerometer-Based Activity Recognition. *Appl Soft Comput* 2015, 37, 1018–1022, doi:10.1016/j.asoc.2015.01.025.
36. Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V.; et al. Scikit-Learn: Machine Learning in Python. *Journal of Machine Learning Research* 2011, 12.
37. van der Walt, S.; Schönberger, J.L.; Nunez-Iglesias, J.; Boulogne, F.; Warner, J.D.; Yager, N.; Gouillart, E.; Yu, T. Scikit-Image: Image Processing in Python. *PeerJ* 2014, 2, e453, doi:10.7717/peerj.453.
38. Hao, J.; Ho, T.K. Machine Learning Made Easy: A Review of Scikit-Learn Package in Python Programming Language. *Journal of Educational and Behavioral Statistics* 2019, 44, 348–361, doi:10.3102/1076998619832248.
39. Python in a Nutshell - Alex Martelli, Anna Martelli Ravenscroft, Steve Holden, Paul McGuire - Google Książki Available online: [https://books.google.pl/books?hl=pl&lr=&id=2WSmEAAAQBAJ&oi=fnd&pg=PT28&dq=python&ots=oUp1BWZF48&sig=kI8hqMTywjvESsgtVj8x9K1hr4U&redir\\_esc=y#v=onepage&q=python&f=false](https://books.google.pl/books?hl=pl&lr=&id=2WSmEAAAQBAJ&oi=fnd&pg=PT28&dq=python&ots=oUp1BWZF48&sig=kI8hqMTywjvESsgtVj8x9K1hr4U&redir_esc=y#v=onepage&q=python&f=false) (accessed on 24 May 2025).
40. Kanungo, T.; Mount, D.M.; Netanyahu, N.S.; Piatko, C.D.; Silverman, R.; Wu, A.Y. An Efficient K-Means Clustering Algorithm: Analysis and Implementation. *IEEE Trans Pattern Anal Mach Intell* 2002, 24, 881–892, doi:10.1109/TPAMI.2002.1017616.

**Disclaimer/Publisher's Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.