algorithms

MDPI

Article

Data Mining Techniques for Endometriosis Detection in a Data-Scarce Medical Dataset

Pablo Caballero [1,*], Luis Gonzalez-Abril [2], Juan A. Ortega [3,*] and Áurea Simon-Soro [4]

[1] Computer Engineering Programme, International Doctorate School, University of Seville, 41013 Seville, Spain
[2] Department of Applied Economic I, Faculty of Economics and Business Sciences, University of Seville, 41018 Seville, Spain; luisgon@us.es
[3] Department of Computer Science, Higher Technical School of Computer Engineering, University of Seville, 41012 Seville, Spain
[4] Department of Stomatology, Faculty of Dentistry, University of Seville, 41009 Seville, Spain; asimon@us.es
* Correspondence: pabcabper@alum.us.es (P.C.); jortega@us.es (J.A.O.)

Abstract: Endometriosis (EM) is a chronic inflammatory estrogen-dependent disorder that affects 10% of women worldwide. It affects the female reproductive tract and its resident microbiota, as well as distal body sites that can serve as surrogate markers of EM. Currently, no single definitive biomarker can diagnose EM. For this pilot study, we analyzed a cohort of 21 patients with endometriosis and infertility-associated conditions. A microbiome dataset was created using five sample types taken from the reproductive and gastrointestinal tracts of each patient. We evaluated several machine learning algorithms for EM detection using these features. The characteristics of the dataset were derived from endometrial biopsy, endometrial fluid, vaginal, oral, and fecal samples. Despite limited data, the algorithms demonstrated high performance with respect to the F1 score. In addition, they suggested that disease diagnosis could potentially be improved by using less medically invasive procedures. Overall, the results indicate that machine learning algorithms can be useful tools for diagnosing endometriosis in low-resource settings where data availability and availability are limited. We recommend that future studies explore the complexities of the EM disorder using artificial intelligence and prediction modeling to further define the characteristics of the endometriosis phenotype.

Keywords: endometriosis; machine learning; artificial intelligence; biomarkers; microbiome; oral systemic; healthcare; SVM

check for updates

Citation: Caballero, P.; Gonzalez-Abril, L.; Ortega, J.A.; Simon-Soro, Á. Data Mining Techniques for Endometriosis Detection in a Data-Scarce Medical Dataset. *Algorithms* **2024**, *17*, 108. https://doi.org/10.3390/a17030108

Academic Editor: Umberto Michelucci

Received: 1 February 2024
Revised: 22 February 2024
Accepted: 27 February 2024
Published: 4 March 2024

Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Endometriosis (EM) is a chronic inflammatory estrogen-dependent disorder characterized by endometrial-like tissue outside the uterus [1]. It affects 10% of women worldwide, causing pelvic pain and infertility [2]. The uterus is not a sterile organ [3]. The inner lining of the uterus, the endometrium, contains resident microorganisms that are different in type and number from the microorganisms that reside in the vagina [4]. A conclusive clinical diagnosis of EM usually requires a combination of medical history, physical examination, imaging techniques (such as ultrasound or MRI), and sometimes laparoscopic surgery. Currently, no single definitive biomarker can diagnose EM with 100% accuracy. Recent studies show that the microbial composition in EM differs from that in healthy individuals. This implies that it plays a significant role in disease and reproductive outcomes [5].

The composition and diversity of the microbiota [6] can indicate the onset or progression of the disease. Distal microbial changes can serve as biomarkers for EM. Gastrointestinal biomarkers, such as those taken from the intestinal tract or oral cavity, are quantifiable indicators that can offer valuable information on overall health. Using samples from the oral microbiome could have advantages over other internal body sites in the female reproductive tract because the process for collection is noninvasive, and specialized sample

collection procedures are not required for the evaluation of EM. Consequently, the early detection of EM can help avoid complications such as reduced fertility. Therefore, the timely diagnosis of this condition is crucial for effective treatment and for the preservation of the patient's reproductive health.

Machine learning algorithms have proven their effectiveness in solving classification problems [7]. These algorithms can employ methods such as logistic regression, decision trees, SVMs, neural networks, and fuzzy classifiers, among others [8,9]. Machine learning has been applied to various problems in the domain of endometrial disease detection [10–12]. For example, some researchers have proposed a novel machine learning algorithm to create precision prognostication systems for endometrial cancer [13,14], while others have used deep learning to classify MRI images of endometrial cancer [15]. These studies demonstrate the potential of machine learning to improve the diagnosis and treatment of endometrial diseases. Another study using noninvasive biomarkers [16], such as blood, urine, or endometrial biopsy, was carried out using the QUADAS-2 tool as an example of a failed attempt, and the conclusion was that they could not replace laparoscopy as a diagnostic procedure. It is important to note that while these biomarkers are promising, none of them are currently used as an independent diagnostic tool for EM [17].

Data scarcity is a common challenge for all algorithms, not only for those used for classification purposes. This problem limits AI applications in the real world, especially in medicine, where data are costly and public funding is needed [18]. Moreover, the availability of public records is limited [19]. In these cases, algorithms fail to achieve an adequate generalization capacity, which results in poor performance. Another major problem facing classification learning algorithms is the imbalance between classes in datasets [20]. That is, one or more classes have most of the examples, while the remaining classes are underrepresented. The problem becomes more severe when the class with the least amount of data is the most relevant [21].

In this study, several machine learning algorithms have been considered in order to provide a classification of EM disease from the microbiome. Our hypothesis asserts that the bacterial taxa present in the oral or fecal microbiome could serve as a surrogate biomarker for the diagnosis of EM in women. If the hypothesis is confirmed, then the resulting impact would have economic and clinical benefits due to the need for less invasive and lower-cost sample collection procedures (see Figure 1).

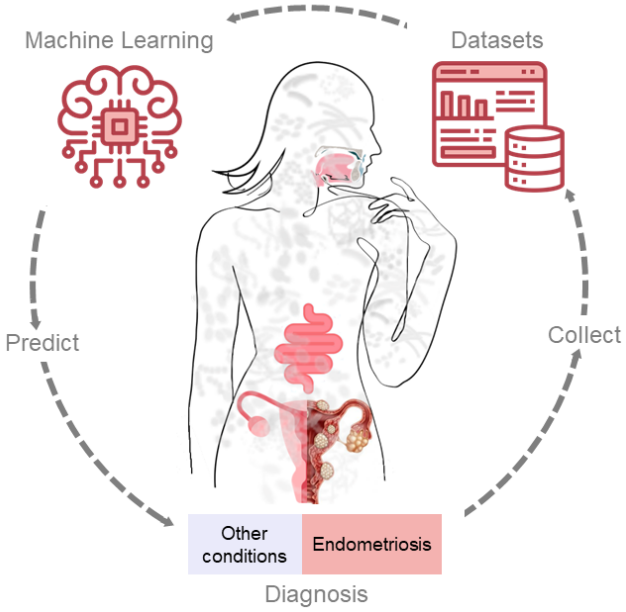

Figure 1. Process schema.

2. Materials and Methods

2.1. Data Sources

A total of twenty-one patients diagnosed with some pathology associated with infertility were included in the analysis. From the total, seven were diagnosed with EM. We included 5 sample types corresponding to the gastrointestinal tract (GIT) and the female reproductive tract (FRT). The gastrointestinal tract included oral and fecal sample types. In addition, the FRT included endometrial fluid (EF), endometrial biopsy (EB), and vaginal (VA) sample types. When evaluating bacterial communities, we used each sample type using microbiome abundances. The dataset consisted of up to 438 different bacterial taxa (the link to the input data is available in the "Data Availability Statement" Section).

This list is an initial approach based on the available datasets:

- Address each sample type independently:
  - Dataset for oral samples.
  - Dataset for fecal samples.
  - Dataset for endometrial fluid.
  - Dataset for endometrial biopsies.
  - Dataset for the vaginal samples.
- Group some sample types:
  - The dataset for the GIT merges the oral–fecal datasets.
  - The dataset for the FRT merges the endometrial fluid, endometrial biopsy, and vaginal microbiomes.
  - The dataset for FRT2 merges the endometrial biopsy and vaginal microbiomes.

Please note that the data size for all datasets was very small and that the proportion of subjects with EM was low compared to that of the other patients; therefore, it was difficult to obtain reliable results from the classification process. Otherwise, from a medical perspective, even if the data size is small, this process is essential because it will enhance patient care if a satisfactory classifier performance can be obtained with the oral or fecal microbiome since it might be a less invasive procedure.

2.1.1. Data Source for Oral Region

All patients in the dataset suffer from an infertility-related disease, either EM or another condition. These conditions are polyps, erythroplakia, hydrosalpinx, ovarian failure, chronic endometritis, polycystic ovary syndrome, or a unicornuate uterus.

In Tables 1 and 2, we can see that some of the bacteria are more frequent and some of them are almost negligible. We had to use a logarithmic scale to observe the variations in values in Figure 2. *Firmicutes Streptococcus* is undoubtedly the most prevalent bacterium in this region.

Table 1. The most frequent bacteria based on the global mean ($\overline{X}$) of the relative abundance in the oral region for EM patients.

Bacteria Name	$\overline{X}$ (%)
Firmicutes Streptococcus	49.14%
Proteobacteria Neisseria	13.14%
Proteobacteria Haemophilus	4.58%
Actinobacteria Actinomyces	3.63%
Firmicutes Veillonella	3.51%

In Figure 2, the x-axis shows the patients and the y-axis shows the relative abundance. Each point has a color that represents a different type of bacteria. However, we do not provide the legend of the colors because they are only informative and the distribution is the important aspect of the chart. We use the same criterion for Figures 3–6.

Table 2. The most frequent bacteria based on the global mean ($\overline{X}$) of the relative abundance in the oral region for non-EM patients.

Bacteria Name	$\overline{X}$ (%)
Firmicutes Streptococcusm	45.36%
Proteobacteria Haemophilusm	22.53%
Fusobacteria Fusobacteriumm	3.28%
Proteobacteria Pasteurellaceae	3.06%
Proteobacteria Neisseriam	2.69%

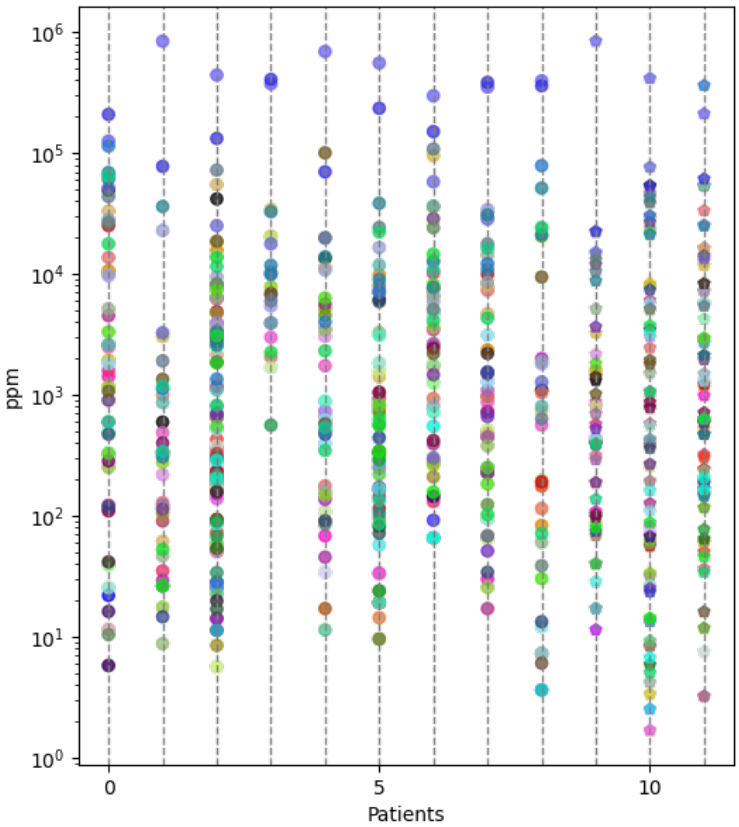

Figure 2. Biome distribution in oral region per patient.

2.1.2. Data Source for Fecal Region

In Tables 3 and 4, we can see that *Firmicutes Lachnospiraceae* is the most prevalent bacterium in this region. However, the second and third ranks vary by the patient's disease.

Table 3. The most frequent bacteria based on the global mean ($\overline{X}$) of the relative abundance in the fecal region for EM patients.

Bacteria Name	$\overline{X}$ (%)
Firmicutes Lachnospiraceae	35.52%
Proteobacteria Enterobacteriaceae	21.14%
Firmicutes Streptococcus	13.89%
Firmicutes Ruminococcaceae	6.23%
Firmicutes Ruminococcus	3.98%
Firmicutes Blautia	3.59%
Firmicutes Faecalibacterium	3.27%

Table 4. The most frequent bacteria based on the global mean ($\overline{X}$) of the relative abundance in the fecal region for non-EM patients.

Bacteria Name	$\overline{X}$ (%)
Firmicutes Lachnospiraceae	24.78%
Firmicutes Ruminococcus	17.33%
Firmicutes Faecalibacterium	8.47%
Proteobacteria Enterobacteriaceae	8.46%
Firmicutes Ruminococcaceae	7.27%
Firmicutes Blautia	6.57%
Firmicutes Bacillus	3.91%
Firmicutes Streptococcus	3.47%
Firmicutes Erysipelotrichaceae	3.34%
Firmicutes Enterococcus	3.17%

As shown in Figure 3, the fecal region has a lower concentration of the dominant bacteria. However, the same bacteria are among those with the highest relative abundance.

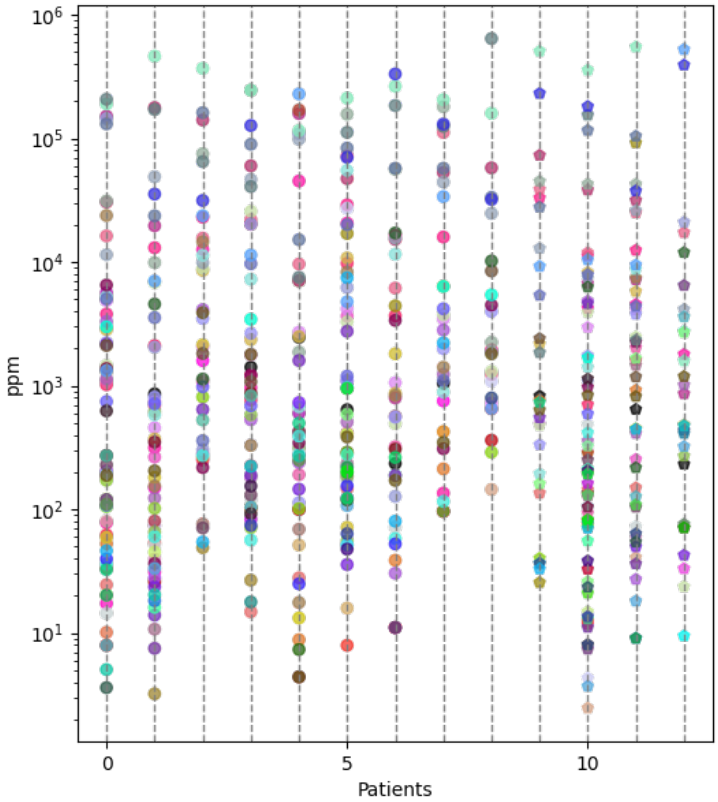

Figure 3. Biome distribution in fecal region per patient.

2.1.3. Data Source for Endometrial Fluid Region

In Tables 5 and 6, we can see that *Firmicutes Lactobacillus* is the most prevalent bacterium in this region, followed by *Actinobacteria Gardnerella*. In Figure 4, the bacterial distribution in the EF region is shown.

Table 5. The most frequent bacteria based on the global mean ($\overline{X}$) of the relative abundance in the EF region for EM patients.

Bacteria Name	$\overline{X}$ (%)
Firmicutes Lactobacillus	67.61%
Actinobacteria Gardnerella	10.83%
Proteobacteria Rhizobiaceae	3.05%
Firmicutes Lachnospiraceae	2.43%
Firmicutes Megasphaera	1.83%

Table 6. The most frequent bacteria based on the global mean ($\overline{X}$) of the relative abundance in the EF region for non-EM patients.

Bacteria Name	$\overline{X}$ (%)
Firmicutes Lactobacillus	61.20%
Actinobacteria Gardnerella	14.89%
Proteobacteria Enterobacteriaceae	6.78%
Proteobacteria Rhizobiaceae	2.22%
Proteobacteria Vibrio	1.62%

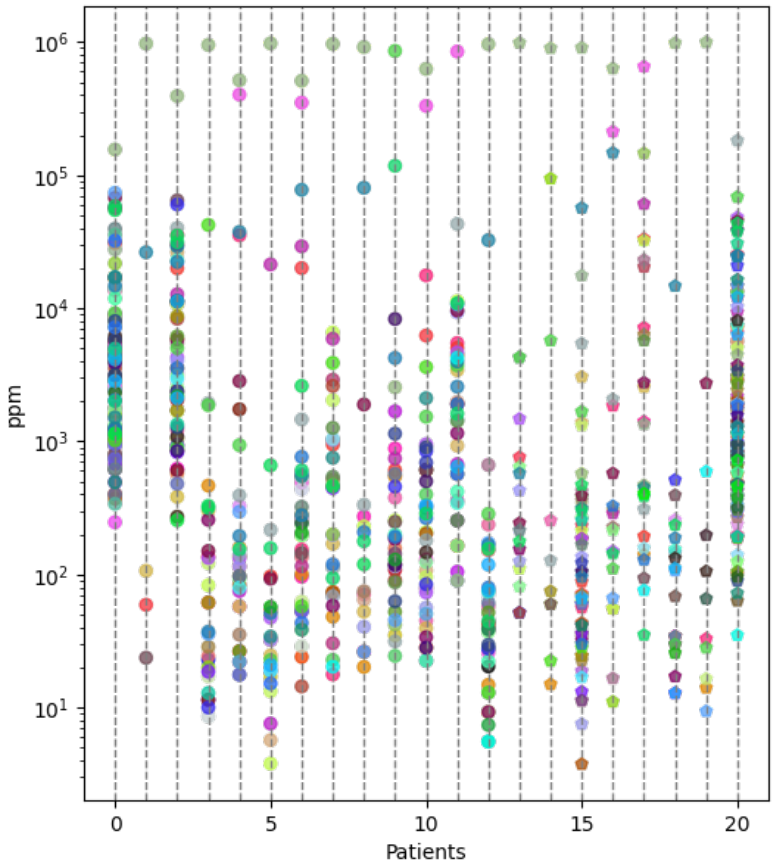

Figure 4. Biome distribution in EF region per patient.

2.1.4. Data Source for Endometrial Biopsy Region

In Tables 7 and 8, we can see that *Firmicutes Lactobacillus* is the most prevalent bacterium in this region, similarly to the EF region. In Figure 5, the bacterial distribution in the EB region is shown.

Table 7. The most frequent bacteria based on the global mean ($\overline{X}$) of the relative abundance in the EB region for EM patients.

Bacteria Name	$\overline{X}$ (%)
Firmicutes Lactobacillus	82.63%
Proteobacteria Rhizobiaceae	3.41%
Actinobacteria Gardnerella	2.62%
Firmicutes Mogibacteriaceae	2.44%
Firmicutes Lachnospiraceae	1.94%

Table 8. The most frequent bacteria based on the global mean ($\overline{X}$) of the relative abundance in the EB region for non-EM patients.

Bacteria Name	$\overline{X}$ (%)
Firmicutes Lactobacillus	49.62%
Firmicutes Lachnospiraceae	8.65%
Proteobacteria Enterobacteriaceae	5.49%
Proteobacteria Rhizobiaceae	3.51%
Firmicutes Streptococcus	3.05%
Bacteroidetes Bacteroides	2.89%
Actinobacteria Gardnerella	2.79%

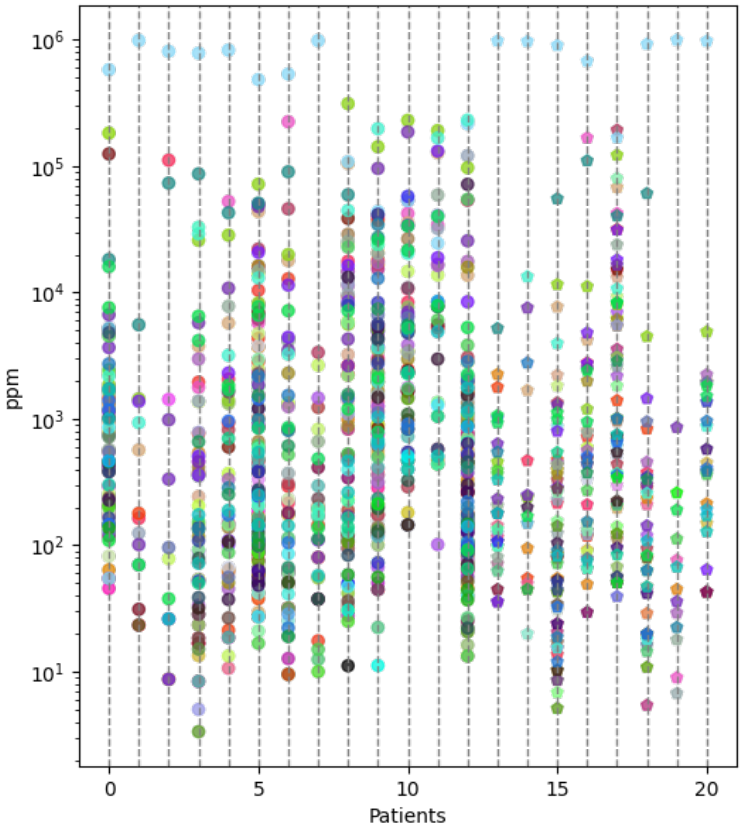

Figure 5. Biome distribution in EB region per patient.

2.1.5. Data Source for Endometrial Vaginal Region

In Tables 9 and 10, we can see that *Firmicutes Lactobacillus* is the most prevalent bacterium in this region, followed by *Actinobacteria Gardnerella* by a large margin. In Figure 6, the bacterial distribution in the vaginal region is shown.

Table 9. The most frequent bacteria based on the global mean ($\overline{X}$) of the relative abundance in the vaginal region for EM patients.

Bacteria Name	$\overline{X}$ (%)
Firmicutes Lactobacillus	82.63%
Firmicutes Lactobacilluss	82.93%
Actinobacteria Gardnerellas	7.43%
Firmicutes Mogibacteriaceae	2.38%
Bacteroidetes Prevotella	1.94%

Table 10. The most frequent bacteria based on the global mean ($\overline{X}$) of the relative abundance in the vaginal region for non-EM patients.

Bacteria Name	$\overline{X}$ (%)
Firmicutes Lactobacilluss	70.54%
Actinobacteria Gardnerellas	9.92%
Proteobacteria Enterobacteriaceaes	7.35%
Firmicutes Enterococcuss	2.93%
Firmicutes Lactobacillaceae	1.68%

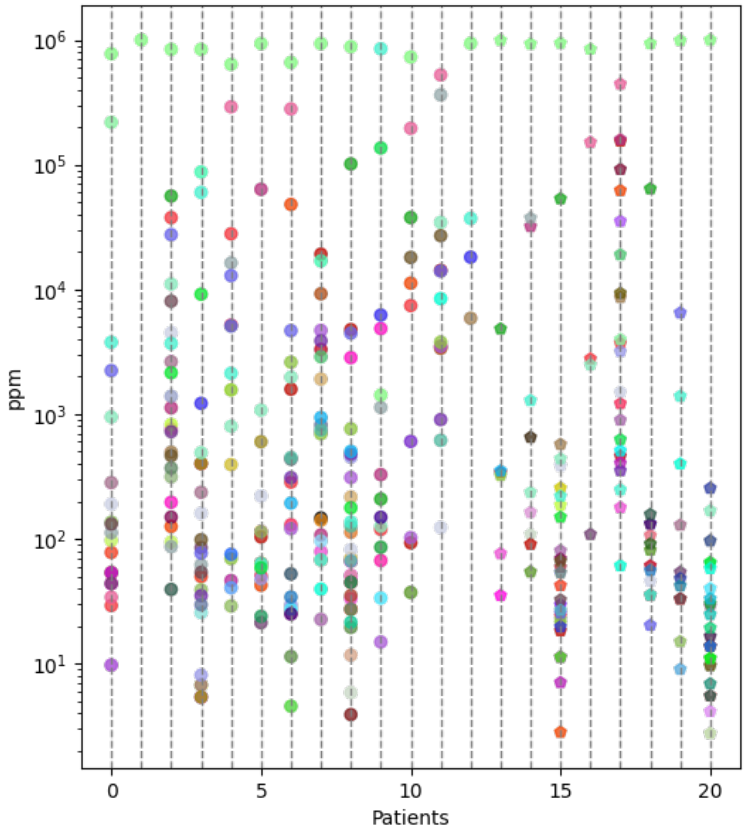

Figure 6. Biome distribution in vaginal region per patient.

2.2. Preprocessing Data

Microbiome analysis was performed using the QIIME2 bioinformatics platform (version 2022.8) to process raw 16S rRNA gene sequences [22]. Sequences were quality-filtered, denoised, and duplicated using the DADA2 algorithm implemented in the denoise-pyro plugin, which was specifically designed for single-end demultiplexed pyrosequencing sequences [23]. To remove low-quality data, the first 15 bases at the 5′ ends of the sequences were trimmed, and samples with a mean Phred quality score below 20 were excluded from further analyses. The sequence variants of amplicons (ASVs) obtained were taxonomically

classified using the Greengenes 13_8 99% Operational Taxonomic Unit (OTU) reference sequence [24,25] with the VSEARCH tool [26]. The VSEARCH tool was used to validate and process raw data from the microbiome analysis. This ensured the quality and reliability of the data before applying any machine learning techniques. We note that our study has no outliers, but some of the datasets have missing values. In these cases, the default value is zero, indicating that there are no bacteria of this type. Each column in the dataset represents a relative abundance and must satisfy Equation (1). This equation ensures that for each patient and area, the sum of all levels is equal to one since the units are proportions.

$$\forall i(\text{rows}) : \sum_{j=first\_relative\_abundance\_column}^{last\_relative\_abundance\_column} X_{i,j} = 1 \quad (1)$$

2.2.1. Filtering

All columns with zero values were discarded, as they did not contribute to the models. Thus, the resulting datasets were reduced according to Table 11.

Table 11. Total of microbiome columns per region.

Region	Number of Columns	Number of Discarded Columns
EB	256 (58.45%)	182 (41.55%)
EF	230 (52.52%)	208 (47.48%)
Vaginal	230 (58.45%)	208 (47.48%)
Oral	144 (32.88%)	294 (67.12%)
Fecal	124 (28.31%)	314 (71.69%)

2.2.2. Scaling

Some bacterial taxa were very rare in the microbiome, which can cause rounding errors in the analysis. To avoid this, relative abundances were expressed in units per million (ppm). A further explanation of the rationale behind using the ppm unit of measure is as follows: First, we found the maximum of all abundance values for a given sample type. In the following step, we looked for the smallest maximum abundance that was not zero, as this value must have an integer part. For example, the smallest abundance in the EB sample type was for the bacterium *Firmicutes Desulfurispora*, which has a value of 5.028848829 ppm. After applying Equation (2), the resulting models should be exactly equivalent if the predictions are also scaled.

$$\forall i(\text{rows}), j(\text{columns}) : X_{i,j} \leftarrow X_{i,j} \times 10^6 \quad (2)$$

2.2.3. Class Imbalance

For this study, we used a diagnosis associated with infertility from the patient's clinical history. Dealing with unbalanced datasets is a challenging task for machine learning algorithms. Some authors suggest downsampling the majority class, but this would result in a significant loss of information in such a small dataset like the one analyzed in our study [27].

Synthetic data (SD) are often used to augment or replace real data in various domains, such as healthcare, where data privacy and availability are major concerns [28]. Moreover, clinical trial data typically have a relatively small number of participants; however, the suggested sizes are always larger than a thousand entries [29–31].

Some authors propose generating synthetic data for the minority class, but this could lead to overfitting the model to artificial samples [32].

SD have some drawbacks that limit their usefulness and reliability for building robust and accurate models [33]. Creating a realistic and representative SD model is challenging and time-consuming. It requires a lot of domain knowledge, expertise, and effort. It is not easy to capture the complexity, variability, and correlations of real data in a synthetic model.

SD may contain omissions and inconsistencies and may not reflect all relevant features, patterns, and anomalies of the real data. This is especially true if the model is based on incomplete or outdated information. Moreover, SD may introduce artificial noise or bias that can affect the performance and the validity of models trained on it. The quality of SD is largely dependent on the skill of the expert building the model. Different experts may have different assumptions, preferences, and methods for generating SD, which can lead to inconsistent and incomparable results. Furthermore, the quality of SD can degrade over time, as real data evolve and change. SD may miss some important insights or discoveries that can only be revealed by analyzing real data. This is because SD are oblivious to the ground truth hidden in real field data that is not yet known in theory. For example, SD may not capture some rare cases that are not well understood or documented in the literature. Even so, SD were tested at an early stage of the study and generated worse results. Therefore, we decided not to use SD for our model based on these drawbacks in the healthcare domain.

2.3. Dataset from One Area

Supervised learning algorithms were based on the diagnostic values of EM in each patient, which were obtained by using a clinical detection method. This detection could involve methods such as the histological examination of the lesions or laparoscopic surgery, among others [34,35]. Laparoscopic surgery is a minimally invasive procedure that allows the visualization of the pelvic organs and the confirmation of the presence or absence of endometriosis lesions. The diagnosed values are then used as labels for the training and testing of the machine learning models.

There are two possible states for a binary problem, so a Boolean variable was used to perform the classification. The criterion for selection was defined as follows: If *endometriosis* is present, then the value is `true`. If endometriosis is not present, then the value is `false`. By applying this premise, the binary classification problem was resolved.

2.4. Machine Learning Models

We conducted different experiments with machine learning algorithms using scikit-learn (version 1.3.0) [36], a free software library for Python, with the main support of the free software libraries numpy (version 1.24.3) and pandas (version 2.0.3). The experiments were run in a Conda environment (version 23.7.3) with Python (version 3.11.4). The experiments compared the viability of several classification techniques using real-world datasets.

Cross-validation [37] is the standard technique to estimate the performance of machine learning models on unseen data. Data were divided into k groups or folds, and one fold was used as the test set and the rest as the training set. The choice of k affects the bias–variance trade-off of the model, where a smaller k leads to a higher bias but a lower variance, and a larger k leads to a lower bias but a higher variance. A single execution of k-fold cross-validation may result in noisy values, because the order of the data may affect the performance values, especially if the datasets are small or imbalanced [38]. However, if the computational cost is not a problem, repeated k-fold cross-validation, which repeats the process n times, each time using a different random seed to shuffle the data before splitting it into folds, can provide a more robust and less dependent estimate [39]. To achieve statistical stability, we used a group of 100 random state values for each test conducted in this study. Thus, we repeated the cross-validation process 100 times with different random seeds and averaged the results. Considering that the dataset consists of 21 observations, we set the number of splits in cross-validation to 7. Then, we had a large enough number of data points in the training set to be able to obtain an adequate classifier. Consequently, the number of training data points was 18, compared to 3 validation data points. If the dataset size was less than 21, the number of splits varied as indicated in Section 2.1.

The general algorithm is described in Algorithm 1. The function performs the classification task using the given classifier and the repeated k-fold cross-validation method [40]. The function uses four inputs: the X matrix, the Y matrix, the number of folds for cross-

validation, and the specific classifier for the chosen method of classification. In this study, we used 100 repetitions for cross-validation.

The function returns a dictionary with the tuples of the mean ($\overline{X}$), the standard deviation (σ), and the length of each cross-validation score of each metric. The function also handles cases where there is a zero-division error in calculating precision and recall by filtering out those scores.

The standard deviation (σ) was used to monitor and prevent overfitting [41], which is a risk of having models with a large variance (σ^2). Overfitting considers the model's capture of noise or the specific features of the training set but does not generalize well to new or unseen data. The standard deviation measures how much the model's predictions deviate from the mean.

We used the following four common metrics to evaluate the performance of machine learning models in classification tasks: *accuracy* (see Equation (3)), *precision* (see Equation (4)), *recall* (see Equation (5)), and *F1* (see Equation (6)). Accuracy measures how often the model predicts the correct class. Precision measures how often the model's positive predictions are correct. Recall measures how often the model detects positive cases. The F1 score is the harmonic mean of precision and recall; it captures the trade-off between precision and recall, which was used to rank the performance of each algorithm due to the imbalance in the classes. Another interesting measure of fit for unbalanced data is the geometric mean [42]. However, during the experiment, we observed that there were many test sets with one class. In such cases, the result provided by this metric was 0, which invalidated this metric.

To make a final decision about the most relevant model, we focused on the F1 score. The F1 score could be more informative due to accuracy, or the receiver operating characteristic curve (ROC). When dealing with a small number of items, both false positives and false negatives must be considered and could have a significant effect on a small sample.

All the variables used by Equations (3)–(6) are defined in Table 12.

Table 12. Confusion matrix.

		Actual Classification	
		Positive	Negative
Predicted classification	Positive	True positive (TP)	False positive (FP)
	Negative	False negative (FN)	True negative (TN)

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{3}$$

$$Precision = \frac{TP}{TP + FP} \tag{4}$$

$$Recall = \frac{TP}{TP + FN} \tag{5}$$

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall} = \frac{2 \times TP}{2 \times TP + FP + FN} \tag{6}$$

In order to avoid overfitting, we used the `RepeatedStratifiedKFold` validator from the scikit-learn library to evaluate the performance of the model. This validator randomly splits the data into k folds and repeats the process n times while preserving the class distribution of the data in each fold. This is more suitable for classification problems with imbalanced classes, as it ensures that each fold has a representative sample of each class. On the other hand, the `RepeatedKFold` validator from the same library does not preserve the class distribution. This can result in some folds having very few or no samples of some classes, which in turn can affect the performance of the model. We compared both validators and found that the `RepeatedStratifiedKFold` validator scores were slightly higher than the `RepeatedKFold` validator scores.

Algorithm 1: `classify(X, Y, k, classifier)`

X : Input. X matrix.
Y : Input. Y matrix.
k : Input. The number of folds for cross-validation.
classifier: Input. The specific classifier for the chosen means of classification.
Result: This method returns a dictionary with the tuples of the mean, standard deviation, and length of each cross-validation score of each metric; the keys of the dictionary are ''`accuracy`'', ''`precision`'', ''`recall`'', and ''`f1`''.

```
// Create metrics
metrics ← ∅
metrics["accuracy"] ← {new: Accuracy()}
metrics["precision_zero_division_0"] ← new: Precision(zero_division = 0)
metrics["precision_zero_division_1"] ← new: Precision(zero_division = 1)
metrics["recall_zero_division_0"] ← new: Recall(zero_division = 0)
metrics["recall_zero_division_1"] ← new: Recall(zero_division = 1)
// Create a repeated k-fold cross-validator generator with 100
   repetitions.
cv ← new: RepeatedStratifiedKFold(k, number_of_repetitions = 100)
// Evaluate cross-validation.  It returns a dictionary of arrays
   with all the scores.  One entry per metric.
scores ← cross_validate(X, Y, classifier, cv, metrics)
// Filter all zero-division scores.
accuracy ← scores["accuracy"]
precision_zd0 ← scores["precision_zero_division_0"]
precision_zd1 ← scores["precision_zero_division_1"]
recall_zd0 ← scores["recall_zero_division_0"]
recall_zd1 ← scores["recall_zero_division_1"]
filtered_scores ←
 filter_non_zero_division(accuracy, precision_zd0, precision_zd1, recall_zd0, recall_zd1)
// Calculate F1.
filtered_scores["f1"] ← get_f1(filtered_scores("precision"), filtered_scores["recall"])

// Calculate the mean and standard deviation for the scores.
result ← ∅
for each: score_name, score_array in: filtered_scores do
    global_mean ← mean(score_array)
    global_std ← std(score_array)
    length ← len(score_array)
    result[score_name] ← ⟨global_mean, global_std, length⟩
end
return result
```

When there were no data from the EM class in the test set, we faced a major zero-division problem in calculating precision, recall, and F1. In these cases, we considered the zero-division argument and calculated the classification for the two possible values for the score: `0`, which would be penalized, and `1`, which would be rewarded. Subsequently, we only took scores with the same value for the same classification that were not affected by this situation, and then we filtered them using Algorithm 2. Although there were no zero-division problems in the accuracy calculation, we calculated this metric only when precision and recall were well defined to ensure the consistency of the results. We applied this method to all the algorithms in this study.

Algorithm 2: `filter_non_zero_division(precision_zd0, precision_zd1, recall_zd0, recall_zd1)`

accuracy : Input. An array of decimals with all the accuracy metrics. The length of this array is `n`.
precision_zd0 : Input. An array of decimals with all the precision metrics when `zero_division` is `0`. The length of this array is `n`.
precision_zd1 : Input. An array of decimals with all the precision metrics when `zero_division` is `1`. The length of this array is `n`.
recall_zd0 : Input. An array of decimals with all the recall metrics when `zero_division` is `0`. The length of this array is `n`.
recall_zd1 : Input. An array of decimals with all the recall metrics when `zero_division` is `1`. The length of this array is `n`.
Result: This method returns a dictionary with an array of cross-validation scores per metric when zero_division has not happened.

$filtered\_accuracy \leftarrow \emptyset$
$filtered\_precision \leftarrow \emptyset$
$filtered\_recall \leftarrow \emptyset$
for $i \leftarrow 1$ **to** n **do**
  if $precision\_zd0(i) = precision\_zd1(i)$ `AND` $recall\_zd0(i) = recall\_zd1(i)$ **then**
    $filtered\_accuracy \leftarrow filtered\_accuracy \cup \{accuracy(i)\}$
    $filtered\_precision \leftarrow filtered\_precision \cup \{precision\_zd0(i)\}$
    $filtered\_recall \leftarrow filtered\_recall \cup \{recall\_zd0(i)\}$
  end
end
$result \leftarrow \emptyset$
$result[\text{“}accuracy\text{”}] \leftarrow filtered\_accuracy$
$result[\text{“}precision\text{”}] \leftarrow filtered\_precision$
$result[\text{“}recall\text{”}] \leftarrow filtered\_recall$
return $result$

For the calculation of the F1 score, the method defined in Algorithm 3 based on Equation (6) (previously shown) was used. This method prevented a zero-division error, which occurs when the divisor in the formula is zero, by returning not a number, or `NaN`. The scikit-learn library expected the default value of 0 or 1. However, we were interested in using `NaN` as the default value. Using this approach, we could later discard the undefined values in our evaluation.

Algorithm 3: `get_f1(precision, recall)`

precision : Input. An array of decimals with all the precision metrics. The length of this array is `n`.
recall : Input. An array of decimals with all the recall metrics. The length of this array is `n`.
Result: This method returns a set of arrays of cross-validation scores when zero_division has not happened.

$result \leftarrow \emptyset$
for $i \leftarrow 1$ **to** n **do**
  if $precision(i) + recall(i) \neq 0$ **then**
    $f1 \leftarrow 2 \cdot \frac{precision(i) \cdot recall(i)}{precision(i) + recall(i)}$
    $result \leftarrow result \cup \{f1\}$
  end
  else
    $result \leftarrow result \cup \{\texttt{NaN}\}$
  end
end
return $result$

2.4.1. Logistic Regression Classification

Algorithm 4 is a binary classification algorithm that uses `LogisticRegression` [36,43] from the scikit-learn library as the classifier, with the default value of `C=1.0` as the inverse of the regularization strength. The regularization strength is a parameter that controls the complexity of the model and prevents overfitting. The solver "`liblinear`" was chosen because it performed well on small datasets.

Algorithm 4: `classify_by_logistic_regression(X, Y, k)`

X: Input. X matrix.
Y: Input. Y matrix.
k: Input. The number of folds for cross-validation.
Result: This method returns a dictionary with the tuples of the mean, standard deviation, and length of the cross-validation scores for each metric. The keys of the dictionary are ''`accuracy`'', ''`precision`'', ''`recall`'', and ''`f1`''.

```
C ← 1.0 // Inverse of regularization strength
solver ← "liblinear"
classifier ← new: LogisticRegression(C, solver)
// Evaluate classification
result ← classify(x, y, k, classifier)
return result
```

2.4.2. Decision Tree Classification

Algorithm 5 is a classification algorithm that uses `DecisionTreeClassifier` [36,44] from the scikit-learn library to predict EM disease based on a set of decision rules. `DecisionTreeClassifier` is a nonparametric supervised learning method that can handle both numerical and categorical data. We determined that the maximum number of features to be considered when looking for the optimal split to be the square root of the total number of features. In our case, this means 20 features out of 438 features in our original data. We then partitioned and evaluated our data using repeated k-fold cross-validation for our experiment.

Algorithm 5: `classify_by_decision_tree(X, Y, k)`

X: Input. X matrix.
Y: Input. Y matrix.
k: Input. The number of folds for cross-validation.
Result: This method returns a dictionary with the tuples of the mean, standard deviation, and length of the cross-validation scores for each metric. The keys of the dictionary are ''`accuracy`'', ''`precision`'', ''`recall`'', and ''`f1`''.

```
max_features ← "sqrt" // Square root of the number of columns of X
classifier ← new: DecisionTreeClassifier(max_features)
// Evaluate classification
result ← classify(x, y, k, classifier)
return result
```

2.4.3. Support Vector Classification

Algorithm 6 is a classification algorithm that uses a support vector classifier, `svm.SVC` [36,45], from the scikit-learn library to predict EM disease based on a subset of training points, which are called support vectors. The support vectors define a decision boundary that maximizes the margin between classes. The support vector classifier uses a kernel function to map the input data to a higher-dimensional feature space, where the decision boundary can be found more easily. Depending on the kernel function, the `gamma` coefficient is applied or not to control the influence of individual training points on the decision boundary. We then partitioned and evaluated our data using repeated k-fold cross-validation for our experiment.

For the `SVC` classification model, we experimented with two types of kernels: `linear` and radial basis function (`rbf`). The `linear` kernel used liblinear as the underlying solver, while the `rbf` kernel was suitable for nonlinear problems.

Algorithm 6: `classify_by_svc(X, Y, k, kernel, C)`

X : Input. X matrix.
Y : Input. Y matrix.
k : Input. The number of folds for cross-validation.
kernel : Input. Algorithm to use in the optimization problem. Possible values: ''`linear`'' or ''`rbf`''.
C : Input. The regularization parameter.
Result: This method returns a dictionary with the tuples of the mean, standard deviation, and length of the cross-validation scores for each metric. The keys of the dictionary are ''`accuracy`'', ''`precision`'', ''`recall`'', and ''`f1`''.

```
// Create C-Support vector classifier
if kernel is "rbf" then
    gamma ← "scale"
    classifier ← new: SVC(C, kernel, gamma)
else
    classifier ← new: SVC(C, kernel)
end
// Evaluate classification
result ← classify(x, y, k, classifier)
return result
```

2.5. Dataset from a Single Sample Type

Table 13 shows the number of patients per sample type. We used repeated k-fold cross-validation to evaluate the performance of our machine learning model on different sample types. We divided the data into k equal parts and tested the model on each part. For the FRT sample type, we used k = 7, so each part contained 3 samples (EB, EF, and vaginal). For the GIT sample type, we used k = 4, so each part also contained 3 samples (oral and fecal).

Table 13. Patients per region.

Region	Number of Patients	Patients with EM
EB	21	7 (33.33%)
EF	21	7 (33.33%)
Vaginal	21	7 (33.33%)
Oral	12	3 (25.00%)
Fecal	13	4 (30.76%)

2.6. Dataset from Multiple Sample Types

In Section 2.1, we present a data analysis focused on the GIT and FRT regions. This section also explains how we implemented the proposed method.

There were regions with missing data for some sample types due to a variety of factors, including the complexity and/or invasiveness of the sample collection process, contamination, or the degradation of sample quality. As a result, we only took into account sample types that had complete datasets for each patient and omitted the others from the study.

Table 14 shows that the GIT region had a very limited sample size and an unbalanced distribution of patients with EM disease. For the GIT region, we could use k = 3, so each part would contain 3 samples. However, this led to an insufficient number of samples per class for classification purposes, resulting in the lowered reliability of the evaluation metrics, such as accuracy or the F1 score. Therefore, we did not evaluate the classification performance for the GIT region in this study.

Table 14. Patients per sample type.

Region	Number of Patients	Patients with EM
GIT (oral + fecal)	08	3 (37.50%)
FRT (EB + EF + vaginal)	21	7 (33.33%)
FRT2 (EB + vaginal)	21	7 (33.33%)

The FRT could be partitioned into 7 subsets, each containing 3 samples that could be used to conduct an evaluation. Therefore, a new dataset was required that aggregated features by patient. The merging method presented in Algorithm 7 transformed a dataset with a list of features into a new dataset, where each patient had a single row with all their features. For the initial part of the execution, we created a list of columns for all the types and features of the samples, plus a dictionary of datasets for each ‘‘`PatientID`’’. The next step was to check whether there were any missing sample types for each patient. If the patient was missing a sample type, then they were skipped. Separately, the algorithm added the column ‘‘`IsEndometriosis`’’ and grouped the dataset by ‘‘`SampleType`’’. Next, the values of each type of sample were appended to an array; the array was scaled and then added to the list of rows. The final output was a dataset with these rows and columns.

A total of 438 bacterial taxa were provided as input data before the merge. The merge operation increased the number of features to 1314, as each feature had 3 values for the FRT, 1 for each sample type. Nevertheless, applying the filtering described in Section 2.2.1 reduced the number of features to 716, so 54.49% of the columns were discarded.

The algorithm for FRT classification is based on the previous methods presented in Algorithm 8. It first combines the data and then applies a supervised machine learning method to classify the patients.

Algorithm 7: `merge(dataset, features, sample_types=[ ''EF'', ''EB'', ''Vaginal'' ])`

dataset : Input. A dataset where the rows are the relative abundances, and each column represents a feature. At least the columns ''IsEndometriosis'', ''PatientID'', ''SampleType'', and all the features should be present.

features : Input. A list of column names that represent the features of the input data.

sample_types: Input. A list of the different types of samples. The values are filtered by the column ''SampleType''.

Result: This method creates a new dataset, where each row contains all the patient's features.

```
column_names ← {"IsEndometriosis", "PatientID"}
for each: sample_type in: sample_types do
    for each: feature in: features do
        column_name ← concat(sample_type, "-", feature)
        column_names ← columns ∪ {column_name}
    end
end
rows ← ∅
dataset_per_patient ← group_by(D, "PatientID")
for each: patient_ID, dataset_per_patient in: dataset_per_patient do
    if NOT∀sample_type ∈ sample_types, ∃row ∈
     dataset_per_patient, dataset_per_patient["SampleType"] = sample_type then
        // Ignore it.
    end
    else
        // All the rows have the same value for this column
        y ← dataset_per_patient["IsEndometriosis"].first()
        // Only rows with the given sample types are allowed
        dataset_per_patient ←
         filter(dataset_per_patient, sample_types, "SampleType")
        // Group by column SampleType
        dataset_per_sample_types ← group_by(dataset_per_patient, "SampleType")
        // Recreate the new row
        for each: sample_type, dataset_per_sample_type in:
         dataset_per_sample_types do
            row_x ← ∀j(columns), column ∈ features : dataset_per_sample_type_{0,j}
            row_x ← scale_to_ppt(row_x)
        end
        row ← {y, sample_type} ∪ row_x
        rows ← rows ∪ {new: Array(row)}
    end
end
result ← new: Dataset(rows, columns)
return result
```

Algorithm 8: `classify_frt(dataset, features, k, classifier)`

dataset : Input. A dataset where the rows are the relative abundances, and each column represents a feature. At least the columns ''`IsEndometriosis`'', ''`PatientID`'', ''`SampleType`'', and all the features should be present.

features : Input. A list of column names that represent the features of the input data.

k : Input. The number of folds for cross-validation.

classifier: Input. The specific classifier for the chosen means of classification.

Result: This method returns a dictionary with the tuples of the mean and standard deviation of each cross-validation score of each metric. The keys of the dictionary are ''`accuracy`'', ''`precision`'', ''`recall`'', and ''`f1`''.

```
// Merge feaures
sample_types ← ["EF", "EB", "Vaginal"]
merged_dataset ← merge(dataset, features, sample_types)
// Select the inputs for classification
X ← ∀column, column ∈ features : merged_dataset[column]
Y ← merged_dataset["IsEndometriosis"]
// Evaluate classification
result ← classify(X, Y, k, classifier)
return result
```

2.7. Hyperparameter Optimization

Hyperparameter optimization of the selected classifier may not necessarily improve its performance. Furthermore, the influence of hyperparameter optimization could be affected by the variability induced by the `random_state` hyperparameter, controlling both the model and the cross-validation process. Therefore, 100 different values of `random_state` were tested in order to achieve the statistical stability of the results of each classification task.

For the SVC algorithm, the regularization hyperparameter `C` was adjusted following a manual grid search [46] to find the optimal values. The regularization strength varied inversely with `C`, which required a strictly positive value. Following the standard approach, we set the initial value of `C` to a default of 1 and then evaluated powers of 10, namely, 1, 10, and 100, for comparison.

The parameter γ in the rbf kernel was considered by default since preliminary experimentation showed that, in the specified range, the performance did not vary significantly.

3. Results

It is worth noting that if a random classifier [47] from Table 13 was considered, the accuracy and the F1 score were $\frac{10}{16}$ (62.5%) and $\frac{3}{12}$ (25%) in the oral region, respectively. Equations (7)–(10) prove the previous figures.

$$\begin{aligned} \mathbb{P}[EM] &= \frac{3}{12} & \mathbb{P}[Success\ EM] &= \frac{3}{12} \\ \mathbb{P}[Non\ EM] &= \frac{9}{12} & \mathbb{P}[Success\ Non\ EM] &= \frac{9}{12} \end{aligned} \tag{7}$$

$$Accuracy = \mathbb{P}[Success] = \mathbb{P}[Success\ EM] \times \mathbb{P}[EM] + \mathbb{P}[Success\ Non\ EM] \times \mathbb{P}[Non\ EM] \tag{8}$$

$$Accuracy = \frac{3}{12} \times \frac{3}{12} + \frac{9}{12} \times \frac{9}{12} = (\frac{1}{4})^2 + (\frac{3}{4})^2 = \frac{10}{16} \tag{9}$$

$$Accuracy = \frac{10}{16} \qquad F1 = \mathbb{P}[EM] = \frac{3}{12} \tag{10}$$

In the fecal sample type, the accuracy and the F1 score were $\frac{97}{169}$ (57.4%) and $\frac{4}{13}$ (30.8%), respectively. Equations (11) and (12) prove the previous figures.

$$\begin{aligned} \mathbb{P}[EM] &= \frac{4}{13} & \mathbb{P}[Success\ EM] &= \frac{4}{13} \\ \mathbb{P}[Non\ EM] &= \frac{9}{13} & \mathbb{P}[Success\ Non\ EM] &= \frac{4}{13} \end{aligned} \tag{11}$$

$$Accuracy = \frac{4}{13} \times \frac{4}{13} + \frac{9}{13} \times \frac{9}{13} = (\frac{4}{13})^2 + (\frac{9}{13})^2 = \frac{97}{169} \qquad F1 = \mathbb{P}[EM] = \frac{4}{13} \tag{12}$$

In the other regions, the accuracy and the F1 score were $\frac{5}{9}$ (55.6%) and $\frac{1}{3}$ (33.3%), respectively. See Equations (13) and (14).

$$\begin{aligned} \mathbb{P}[EM] &= \frac{7}{21} & \mathbb{P}[Success\ EM] &= \frac{7}{21} \\ \mathbb{P}[Non\ EM] &= \frac{14}{21} & \mathbb{P}[Success\ Non\ EM] &= \frac{14}{21} \end{aligned} \tag{13}$$

$$Accuracy = \frac{7}{21} \times \frac{7}{21} + \frac{14}{21} \times \frac{14}{21} = (\frac{7}{21})^2 + (\frac{14}{21})^2 = \frac{245}{441} = \frac{5}{9} \qquad F1 = \mathbb{P}[EM] = \frac{7}{21} = \frac{1}{3} \tag{14}$$

As a result, we can see that the random classifier performed poorly in all regions. Thus, we recommend using models that reach at least these performance thresholds.

Considering the number of folds of cross-validation and the number of repetitions (100), the classifier was trained a total of 300 times in the oral and fecal regions and 700 times in the other regions. Nevertheless, only classifiers not presenting zero division were considered for obtaining the statistics and are reported in the corresponding tables.

As mentioned in Section 2.4, the F1 score was used to compare the performance of the different machine learning classifiers. Next, the best classifiers obtained for each model—logistic regression, decision tree, SVM linear, and SVM rbf—after the grid search are presented and analyzed.

3.1. Logistic Regression Classification

The main parameters for this classification are shown in Table 15. Table 16 summarizes the main findings of this experiment using these parameters.

Table 15. The parameters for the logistic regression classification model.

Name	Value	Description
`Classifier`	`LogisticRegression`	Python classifier from `sklearn.linear_model`
`solver`	`liblinear`	Algorithm to use in optimization problem
`max_iter`	`1,000,000`	Maximum number of iterations
`C`	`1.0`	Inverse of regularization strength
`n_repeats`	`100`	Number of repetitions in `RepeatedKFold`

Table 16. Global mean ($\overline{X}$) and global standard deviation (σ) for scores for logistic regression classification.

Metric	Statistic	EB	EF	Vaginal	Oral	Fecal	FRT	FRT2
	#	536	469	625	179	335	610	608
Accuracy	$\overline{X}(\%)$	38.43	37.67	29.97	80.07	59.38	39.78	46.38
	σ	0.25	0.26	0.25	0.29	0.25	0.25	0.26
Precision	$\overline{X}(\%)$	23.54	22.28	20.27	71.97	40.72	30.16	35.8
	σ	0.33	0.32	0.26	0.4	0.33	0.3	0.32
Recall	$\overline{X}(\%)$	36.29	33.48	38.8	81.01	69.25	54.75	61.84
	σ	0.46	0.45	0.47	0.39	0.46	0.48	0.47
F1	#	215	175	267	145	232	360	398
	$\overline{X}$ (%)	26.46	25.04	25.31	74.77	49.72	36.54	42.87
	σ	0.34	0.34	0.3	0.39	0.36	0.32	0.34

# denotes the number of classifiers used in the calculation of the statistics.

As shown in Table 16, the F1-score of the random classifiers for the EB, EF, and vaginal samples did not exceed the minimum threshold of 33.33% given by Equation (12).

The F1 scores for the oral and fecal samples were 74.77% and 49.72%, respectively. GIT sample types were higher than the minimum thresholds of the random classifier, which were 25% for oral samples and 30.8% for fecal samples. These results validated the initial hypothesis that the classification of the oral and fecal sample types is better than in the reproductive region. In particular, the result obtained in the oral region is very promising.

3.2. Decision Tree Classification

The main parameters for this classification are shown in Table 17. Table 18 summarizes the main findings of this experiment using these parameters.

Table 17. The parameters for the decision tree regression classification model.

Name	Value	Description
`Classifier`	`DecisionTreeClassifier`	The Python classifier from sklearn.tree.
`max_features`	`sqrt`	The number of features to consider when looking for the best split: this parameter affected the performance and complexity of the decision tree.
`n_repeats`	`100`	The number of repetitions in `RepeatedKFold`.

Table 18. Global mean ($\overline{X}$) and global standard deviation (σ) for scores for decision tree classification.

Metric	Statistic	EB	EF	Vaginal	Oral	Fecal	FRT	FRT2
	#	456	510	452	178	270	534	547
Accuracy	$\overline{X}(\%)$	40.86	29.61	38.2	73.6	58.73	51.56	46.86
	σ	0.29	0.25	0.25	0.26	0.31	0.28	0.28
Precision	$\overline{X}(\%)$	27.89	13.01	21.2	61.42	39.85	39.61	33.24
	σ	0.36	0.28	0.32	0.38	0.41	0.34	0.36
Recall	$\overline{X}(\%)$	41.67	20.69	32.3	80.34	54.81	63.39	50.18
	σ	0.48	0.4	0.45	0.4	0.5	0.46	0.48
F1	#	205	112	160	143	148	356	294
	$\overline{X}$ (%)	31.06	14.93	23.81	67.6	44.52	46.49	37.68
	σ	0.36	0.3	0.33	0.37	0.43	0.36	0.37

# denotes the number of classifiers used in the calculation of the statistics.

Table 18 shows that the random classifiers for the EF and vaginal samples had F1 scores below 33.3%, which is the minimum threshold given by Equation (14). This value suggests that they performed more poorly than the random classifier.

3.3. Support Vector Classification with the linear kernel

The main parameters for this classification are shown in Table 19. Table 20 summarizes the main findings of this experiment using the listed parameters.

Table 19. The parameters for the support vector classification model (linear kernel).

Name	Value	Description
`Classifier`	`SVC`	Python classifier from sklearn.svm
`kernel`	`linear`	Kernel type to be used
`C`	`1.0`	Regularization parameter
`n_repeats`	`100`	Number of repetitions in `RepeatedKFold`

Table 20. Global mean ($\overline{X}$) and global standard deviation (σ) for scores for SVC classification with linear kernel and `C = 1`.

Metric	Statistic	EB	EF	Vaginal	Oral	Fecal	FRT	FRT2
	#	476	491	667	154	278	572	642
Accuracy	$\overline{X}(\%)$	38.31	35.17	25.69	79.44	69.9	47.03	44.29
	σ	0.25	0.25	0.24	0.29	0.24	0.27	0.26
Precision	$\overline{X}(\%)$	21.46	20.03	18.04	70.78	53.99	35.81	33.52
	σ	0.31	0.31	0.24	0.41	0.34	0.33	0.30
Recall	$\overline{X}(\%)$	34.77	30.86	36.21	79.87	81.65	59.97	60.44
	σ	0.46	0.44	0.46	0.4	0.39	0.47	0.47
F1	#	181	169	264	123	227	363	412
	$\overline{X}$ (%)	25.18	22.52	22.88	73.59	62.91	42.37	40.93
	σ	0.34	0.32	0.29	0.39	0.34	0.35	0.33

# denotes the number of classifiers used in the calculation of the statistics.

The results of the SVC with the linear kernel shown in Table 20 are very similar to those obtained with logistic regression. Therefore, the results support the initial hypothesis that the classification in the oral and fecal regions is better than in the reproductive region. Furthermore, by taking into account the unbalanced EM class, the results of precision and recall in the oral and fecal regions are very promising.

3.4. Support Vector Classification with the rbf kernel and `C = 100`

The main parameters for this classification are shown in Table 21. Table 22 summarizes the main findings of this experiment using these parameters. It should be noted that in this case, `C` was much larger.

Table 21. The parameters for the support vector classification model (rbf kernel and `C = 100`).

Name	Value	Description
`Classifier`	`SVC`	Python classifier from sklearn.svm
`kernel`	`rbf`	Kernel type to be used
`gamma`	`scale`	Kernel coefficient
`C`	`100`	Regularization parameter

Table 22. Global mean ($\overline{X}$) and global standard deviation (σ) for scores for SVC classification with rbf kernel, and `C = 100`.

Metric	Statistic	EB	EF	Vaginal	Oral	Fecal	FRT	FRT2
	#	623	349	574	124	261	575	613
Accuracy	$\overline{X}(\%)$	67.58	25.02	54.59	76.34	66.03	62.72	63.4
	σ	0.27	0.14	0.28	0.32	0.26	0.27	0.26
Precision	$\overline{X}(\%)$	60.54	1.72	44.37	68.82	47.19	54.84	55.85
	σ	0.34	0.07	0.37	0.4	0.39	0.36	0.32
Recall	$\overline{X}(\%)$	84.27	5.16	64.55	80.65	68.2	76.17	85.24
	σ	0.34	0.22	0.45	0.4	0.47	0.4	0.33
F1	#	547	18	397	100	178	466	540
	$\overline{X}$ (%)	67.05	2.58	49.56	72.45	53.96	60.23	64.13
	σ	0.31	0.11	0.36	0.39	0.4	0.33	0.29

# denotes the number of classifiers used in the calculation of the statistics.

Table 22 shows the results of the SVC classification with the rbf kernel. The F1 score for FRT2 was high, indicating a good performance. The F1 scores for the other regions were similar to those of the other models, except for EF. We have crossed out the F1 score of the EF sample type in Table 22 because the number of classifiers was very small, and it did not produce meaningful statistics.

4. Conclusions

This paper presents a comparative analysis of three machine learning classification algorithms: logistic regression, decision trees, and support vector machines. We applied these algorithms to seven sample types to detect EM disease. Our results show that logistic regression outperforms the other two algorithms in terms of the F1 score. The logistic regression achieved an accuracy of 80.07%, a precision of 71.97%, a recall of 81.01%, and an F1 score of 74.77%. Our findings confirm our initial hypothesis that the oral sample type contains relevant information to predict EM disease. They also suggest that a machine learning model based on logistic regression could be a reliable and noninvasive tool for early diagnosis.

The findings from our study have several implications for both health and research. They could reduce the costs and risks associated with endometrial analysis and provide insight into surrogate biomarkers associated with EM disease. The machine learning model's results can assist the specialist in considering EM as a potential diagnosis. However, there are some limitations that we faced that future studies should address.

First, our sample size was relatively small, with only 12 patients in the oral sample type region and 3 of them with EM disease. Having a small sample size allows for some bias or noise in the data, which could affect the scalability and robustness of our model. Although more information was available for the other sample types, the data were still limited.

Second, our study was cross-sectional, which means that we did not follow the patients over time to monitor the progression of the disease or the response to treatment. This could limit our understanding of the causal relationships between bacteria and EM disease and the temporal dynamics of surrogate biomarkers. Therefore, we suggest that future studies include larger cohorts, thus increasing the sample size, and conduct longitudinal studies to track changes in the oral sample type over time.

Finally, we encourage future studies to explore additional machine learning algorithms or methodologies that may improve the performance or comprehension of our model. This could include fuzzy classifiers, neural networks, ensemble techniques, transfer learning, pre-trained models, or more sophisticated models that could be enhanced by synthetic data. Further investigations using machine learning algorithms could reveal the microbial role in the disease and the local (FRT) and distal (GIT) host–microbial homeostasis related to EM.

Author Contributions: Conceptualization, P.C., L.G.-A., J.A.O. and Á.S.-S.; methodology, P.C. and L.G.-A.; software, P.C.; validation, P.C. and L.G.-A.; formal analysis, P.C. and L.G.-A.; investigation, P.C.; resources, P.C. and Á.S.-S.; data curation, P.C., L.G.-A. and J.A.O.; writing—original draft preparation, P.C., L.G.-A. and Á.S.-S.; writing—review and editing, P.C., L.G.-A., J.A.O. and Á.S.-S.; visualization, P.C.; supervision, L.G.-A. and J.A.O.; project administration, J.A.O.; funding acquisition, J.A.O. All authors have read and agreed to the published version of the manuscript.

Funding: This research has been partially supported by the "Generation of Reliable Synthetic Health Data for Federated Learning in Secure Data Spaces" Research Project (PID2022-141045OB-C42 (AEI/FEDER, UE)) funded by MCIN/AEI/10.13039/501100011033 and by "ERDF A way of making Europe" by the "European Union".

Institutional Review Board Statement: The present study was approved by the Ethics Committee of Universidad del Atlántico 1532-N-21 and European University on 22 December 2020. The research was carried out following the guidelines of the Declaration of Helsinki on Medical Research in Human Subjects and Good Clinical Practice. Written informed consent was obtained from all participants.

Informed Consent Statement: Informed consent was obtained from all subjects involved in the study.

Data Availability Statement: Patient data are confidential, so they have been anonymized. They are openly available in figshare at 10.6084/m9.figshare.24125037. The Jupyter notebooks with all the used code are available at 10.6084/m9.figshare.23905125. The results datasets that support the findings of this study and the figures of the scores are openly available in figshare at the following locations: LogisticRegression scores and figures: 10.6084/m9.figshare.23904903 and 10.6084/m9.figshare.24114696; DecisionTree scores and figures: 10.6084/m9.figshare.23904861 and 10.6084/m9.figshare.24114702; SVC scores and figures (linear kernel and C = 1): 10.6084/m9.figshare.23904933 and 10.6084/m9.figshare.24114687; SVC scores and figures (rbf kernel and C = 1): 10.6084/m9.figshare.24114681 and 10.6084/m9.figshare.24114579; SVC scores and figures (rbf kernel and C = 10): 10.6084/m9.figshare.23904954 and 10.6084/m9.figshare.24114570; SVC scores and figures (rbf kernel and C = 100): 10.6084/m9.figshare.23904972 and 10.6084/m9.figshare.24114561.

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

Abbreviations

The following abbreviations are used in this manuscript:

ASVs	Amplicon Sequence Variants
CSV	Comma-Separated Values
DADA2	Divisive Amplicon Denoising Algorithm 2
EB	Endometrial biopsy
EF	Endometrial fluid
EM	Endometriosis
FRT	Female reproductive tract
GIT	Gastrointestinal tract
MRI	Magnetic Resonance Imaging
OTUs	Operational Taxonomic Units
RBF	Radial basis function
ROC	Receiver operating characteristic curve
rRNA	Ribosomal Ribonucleic Acid
SDK	Software Development Kit
SVC	Support vector classifier
SVM	Support vector machine

References

1. Bullon, P.; Navarro, J.M. Inflammasome as a Key Pathogenic Mechanism in Endometriosis. *Curr. Drug Targets* **2017**, *18*. [CrossRef]
2. Zondervan, K.T.; Becker, C.M.; Missmer, S.A. Endometriosis. *N. Engl. J. Med.* **2020**, *382*, 1244–1256. [CrossRef]
3. Moreno, I.; Codoñer, F.M.; Vilella, F.; Valbuena, D.; Martinez-Blanch, J.F.; Jimenez-Almazán, J.; Alonso, R.; Alamá, P.; Remohí, J.; Pellicer, A.; et al. Evidence that the endometrial microbiota has an effect on implantation success or failure. *Am. J. Obstet. Gynecol.* **2016**, *215*, 684–703. [CrossRef] [PubMed]

4. Riganelli, L.; Iebba, V.; Piccioni, M.; Illuminati, I.; Bonfiglio, G.; Neroni, B.; Calvo, L.; Gagliardi, A.; Levrero, M.; Merlino, L.; et al. Structural Variations of Vaginal and Endometrial Microbiota: Hints on Female Infertility. *Front. Cell. Infect. Microbiol.* **2020**, *10*, 350. [CrossRef]
5. Moreno, I.; Garcia-Grau, I.; Perez-Villaroya, D.; Gonzalez-Monfort, M.; Bahçeci, M.; Barrionuevo, M.J.; Taguchi, S.; Puente, E.; Dimattina, M.; Lim, M.W.; et al. Endometrial microbiota composition is associated with reproductive outcome in infertile patients. *Microbiome* **2022**, *10*, 1. [CrossRef] [PubMed]
6. Bhattacharya, K.; Dutta, S.; Sengupta, P.; Bagchi, S. Reproductive tract microbiome and therapeutics of infertility. *Middle East Fertil. Soc. J.* **2023**, *28*, 11. [CrossRef]
7. Mitchell, T. *Machine Learning*; McGraw-Hill Education: New York, NY, USA, 1997.
8. Rabcan, J.; Levashenko, V.; Zaitseva, E.; Kvassay, M. EEG Signal Classification Based on Fuzzy Classifiers. *IEEE Trans. Ind. Inform.* **2022**, *18*, 757–766. [CrossRef]
9. Bonissone, P.; Cadenas, J.M.; Carmen Garrido, M.; Andrés Díaz-Valladares, R. A fuzzy random forest. *Int. J. Approx. Reason.* **2010**, *51*, 729–747. [CrossRef]
10. Visalaxi, S.; Punnoose, D.; Muthu, T.S. An Analogy of Endometriosis Recognition Using Machine Learning Techniques. In Proceedings of the 2021 Third International Conference on Intelligent Communication Technologies and Virtual Mobile Networks (ICICV), Tirunelveli, India, 4–6 February 2021. [CrossRef]
11. Esfandiari, N.; Babavalian, M.R.; Moghadam, A.M.E.; Tabar, V.K. Knowledge discovery in medicine: Current issue and future trend. *Expert Syst. Appl.* **2014**, *41*, 4434–4463. [CrossRef]
12. Wang, L.; Zheng, W.; Ding, X.; Yu, J.; Jiang, W.; Zhang, S. Identification biomarkers of eutopic endometrium in endometriosis using artificial neural networks and protein fingerprinting. *Fertil. Steril.* **2010**, *93*, 2460–2462. [CrossRef]
13. Praiss, A.M.; Huang, Y.; St. Clair, C.M.; Tergas, A.I.; Melamed, A.; Khoury-Collado, F.; Hou, J.Y.; Hu, J.; Hur, C.; Hershman, D.L.; et al. Using machine learning to create prognostic systems for endometrial cancer. *Gynecol. Oncol.* **2020**, *159*, 744–750. [CrossRef]
14. Bhardwaj, V.; Sharma, A.; Parambath, S.V.; Gul, I.; Zhang, X.; Lobie, P.E.; Qin, P.; Pandey, V. Machine Learning for Endometrial Cancer Prediction and Prognostication. *Front. Oncol.* **2022**, *12*. [CrossRef]
15. Chen, X.; Wang, Y.; Shen, M.; Yang, B.; Zhou, Q.; Yi, Y.; Liu, W.; Zhang, G.; Yang, G.; Zhang, H. Deep learning for the determination of myometrial invasion depth and automatic lesion identification in endometrial cancer MR imaging: A preliminary study in a single institution. *Eur. Radiol.* **2020**, *30*, 4985–4994. [CrossRef] [PubMed]
16. Nisenblat, V.; Prentice, L.; Bossuyt, P.M.; Farquhar, C.; Hull, M.L.; Johnson, N. Combination of the non-invasive tests for the diagnosis of endometriosis. *Cochrane Database Syst. Rev.* **2016**, *2016*, CD012281. [CrossRef] [PubMed]
17. Anastasiu, C.V.; Moga, M.A.; Elena Neculau, A.; Bălan, A.; Scârneciu, I.; Dragomir, R.M.; Dull, A.M.; Chicea, L.M. Biomarkers for the Noninvasive Diagnosis of Endometriosis: State of the Art and Future Perspectives. *Int. J. Mol. Sci.* **2020**, *21*, 1750. [CrossRef]
18. Mukhamediev, R.I.; Popova, Y.; Kuchin, Y.; Zaitseva, E. Review of Artificial Intelligence and Machine Learning Technologies: Classification, Restrictions, Opportunities and Challenges. *Mathematics* **2022**, *10*, 2552. [CrossRef]
19. Rychnovská, D. Anticipatory Governance in Biobanking: Security and Risk Management in Digital Health. *Sci. Eng. Ethics* **2021**, *27*, 30. [CrossRef] [PubMed]
20. Nuñez, H.; Gonzalez-Abril, L.; Angulo, C. Improving SVM Classification on Imbalanced Datasets by Introducing a New Bias. *J. Classif.* **2017**, *34*, 427–443. [CrossRef]
21. Gonzalez-Abril, L.; Angulo, C.; Nuñez, H.; Leal, Y. Handling binary classification problems with a priority class by using Support Vector Machines. *Appl. Soft Comput.* **2017**, *61*, 661–669. [CrossRef]
22. Bolyen, E.; Rideout, J.R.; Dillon, M.R.; Bokulich, N.A.; Abnet, C.C.; Al-Ghalith, G.A.; Alexander, H.; Alm, E.J.; Arumugam, M.; Asnicar, F.; et al. Reproducible, interactive, scalable and extensible microbiome data science using QIIME 2. *Nat. Biotechnol.* **2019**, *37*, 852–857. [CrossRef]
23. Callahan, B.J.; McMurdie, P.J.; Rosen, M.J.; Han, A.W.; Johnson, A.J.A.; Holmes, S.P. DADA2: High-resolution sample inference from Illumina amplicon data. *Nat. Methods* **2016**, *13*, 581–583. [CrossRef] [PubMed]
24. McDonald, D.; Price, M.N.; Goodrich, J.; Nawrocki, E.P.; DeSantis, T.Z.; Probst, A.; Andersen, G.L.; Knight, R.; Hugenholtz, P. An improved Greengenes taxonomy with explicit ranks for ecological and evolutionary analyses of bacteria and archaea. *ISME J.* **2012**, *6*, 610–618. [CrossRef]
25. Bokulich, N.A.; Kaehler, B.D.; Rideout, J.R.; Dillon, M.; Bolyen, E.; Knight, R.; Huttley, G.A.; Gregory Caporaso, J. Optimizing taxonomic classification of marker-gene amplicon sequences with QIIME 2's q2-feature-classifier plugin. *Microbiome* **2018**, *6*. [CrossRef]
26. Rognes, T.; Flouri, T.; Nichols, B.; Quince, C.; Mahé, F. VSEARCH: A versatile open source tool for metagenomics. *PeerJ* **2016**, *4*, e2584. [CrossRef]
27. Barandela, R.; Sánchez, J.; García, V.; Rangel, E. Strategies for learning in class imbalance problems. *Pattern Recognit.* **2003**, *36*, 849–851. [CrossRef]
28. Chen, R.J.; Lu, M.Y.; Chen, T.Y.; Williamson, D.F.K.; Mahmood, F. Synthetic data in machine learning for medicine and healthcare. *Nat. Biomed. Eng.* **2021**, *5*, 493–497. [CrossRef]
29. Azizi, Z.; Zheng, C.; Mosquera, L.; Pilote, L.; El Emam, K. Can synthetic data be a proxy for real clinical trial data? A validation study. *BMJ Open* **2021**, *11*, e043497. [CrossRef] [PubMed]

30. Esteban Lasso, A.; Martínez Toledo, C.; Perosanz Amarillo, S. *Diseño de un Modelo Para Generar Datos Sintéticos en Investigación Médica*; Universidad de Alcalá: Alcalá de Henares, Spain, 2023; Volume 12.
31. Reiner Benaim, A.; Almog, R.; Gorelik, Y.; Hochberg, I.; Nassar, L.; Mashiach, T.; Khamaisi, M.; Lurie, Y.; Azzam, Z.S.; Khoury, J.; et al. Analyzing Medical Research Results Based on Synthetic Data and Their Relation to Real Data Results: Systematic Comparison From Five Observational Studies. *JMIR Med. Inform.* **2020**, *8*, e16492. [CrossRef]
32. Chawla, N. Data Mining and Knowledge Discovery Handbook. In *Data Mining and Knowledge Discovery Handbook*; Chapter Data Mining for Imbalanced Datasets: An Overview; Springer: New York, NY, USA, 2010; pp. 875–886. [CrossRef]
33. Murtaza, H.; Ahmed, M.; Khan, N.F.; Murtaza, G.; Zafar, S.; Bano, A. Synthetic data generation: State of the art in health care domain. *Comput. Sci. Rev.* **2023**, *48*, 100546. [CrossRef]
34. Spaczynski, R.Z.; Duleba, A.J. Diagnosis of Endometriosis. *Semin. Reprod. Med.* **2003**, *21*, 193–208. [CrossRef]
35. Hsu, A.L.; Khachikyan, I.; Stratton, P. Invasive and non-invasive methods for the diagnosis of endometriosis. *Clin. Obstet. Gynecol.* **2010**, *53*, 413–419. [CrossRef] [PubMed]
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. *J. Mach. Learn. Res.* **2011**, *12*, 2825–2830.
37. Ramezan, C.A.; Warner, T.A.; Maxwell, A.E. Evaluation of Sampling and Cross-Validation Tuning Strategies for Regional-Scale Machine Learning Classification. *Remote Sens.* **2019**, *11*, 185. [CrossRef]
38. Santos, M.S.; Soares, J.P.; Abreu, P.H.; Araujo, H.; Santos, J. Cross-Validation for Imbalanced Datasets: Avoiding Overoptimistic and Overfitting Approaches [Research Frontier]. *IEEE Comput. Intell. Mag.* **2018**, *13*, 59–76. [CrossRef]
39. Wong, T.T.; Yeh, P.Y. Reliable Accuracy Estimates from k-Fold Cross Validation. *IEEE Trans. Knowl. Data Eng.* **2020**, *32*, 1586–1594. [CrossRef]
40. Simon, R. Supervised Analysis When the Number of Candidate Features (p) Greatly Exceeds the Number of Cases (n). *SIGKDD Explor. Newsl.* **2003**, *5*, 31–36. [CrossRef]
41. Cawley, G.; Talbot, N. On Over-fitting in Model Selection and Subsequent Selection Bias in Performance Evaluation. *J. Mach. Learn. Res.* **2010**, *11*, 2079–2107.
42. Gonzalez-Abril, L.; Nuñez, H.; Angulo, C.; Velasco, F. GSVM: An SVM for handling imbalanced accuracy between classes inbi-classification problems. *Appl. Soft Comput.* **2014**, *17*, 23–31. [CrossRef]
43. Peng, C.; Lee, K.; Ingersoll, G. An Introduction to Logistic Regression Analysis and Reporting. *J. Educ. Res.* **2002**, *96*, 3–14. [CrossRef]
44. Quinlan, J. Induction of Decision Trees. *Mach. Learn.* **1986**, *1*, 81–106. [CrossRef]
45. Gonzalez-Abril, L.; Angulo, C.; Velasco, F.; Català, A. Dual unification of bi-class support vector machine formulations. *Pattern Recognit.* **2006**, *39*, 1325–1332. [CrossRef]
46. Syarif, I.; Prugel-Bennett, A.; Wills, G. SVM Parameter Optimization using Grid Search and Genetic Algorithm to Improve Classification Performance. *TELKOMNIKA (Telecommun. Comput. Electron. Control)* **2016**, *14*, 1502. [CrossRef]
47. Falomir, Z.; Museros, L.; Sanz, I.; Gonzalez-Abril, L. Categorizing paintings in art styles based on qualitative color descriptors, quantitative global features and machine learning (QArt-Learn). *Expert Syst. Appl.* **2018**, *97*, 83–94. [CrossRef]

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.