CLASSIFICATION ALGORITHM APPLICATION FOR PREDICTION OF DISEASE EVOLUTION

Teresa Vega Martínez
Alberto Montero Solera

teresavegamar@uma.es 13albertomontero@uma.es
Ingeniería de la Salud.
Universidad de Málaga.

In this project, the classification of a set of data has previously been carried out using 5 different methods. After applying these methods, the false positives, false negatives, true positives and true negatives have been collected in a table for each method. Based on this information, we are going to perform a detailed analysis of the performance of each method. To do this, we will calculate different quality metrics with precision or recall. Finally according to the analysis, we will determine which method is better overall and why.

1 Introduction

Predicting disease evolution through classification algorithms has become an essential tool in the medical field. These methods offer significant advantages, such as enabling early diagnosis, forecasting disease progression, and assisting healthcare professionals in making informed decisions. However, to maximize the utility of these algorithms, it is crucial to evaluate and compare the performance of different methods in a structured and fair manner. This work focuses on comparing several supervised learning methods applied to disease prediction tasks, using key performance metrics that are widely used in the literature.

A central aim of this study is to establish a fair comparison between methods that address a specific problem, relying on well-defined metrics such as accuracy, sensitivity, and specificity. These metrics are chosen for their ability to reflect the strengths and weaknesses of each method, particularly when dealing with imbalanced datasets, which are often encountered in medical applications. In addition to comparing the methods, this work will provide a thorough understanding of the procedure for training and testing a supervised learning model, including an exploration of the challenges and critical aspects that influence the effectiveness of these methods.

This report also introduces the concept of Deep Learning, an advanced form of supervised learning. Deep Learning algorithms have gained prominence due to their ability to handle complex data and capture intricate patterns that traditional methods might miss. We will examine how Deep Learning methods differ from conventional approaches and discuss their potential advantages in the context of disease prediction.

By the end of this report, we will present a comprehensive analysis of the results obtained from each method, offering a clear and fair comparison. The goal is to identify the method that provides the most accurate and reliable predictions for disease evolution, considering both traditional supervised learning and Deep Learning approaches.

2 Dataset

It is essential to understand the nature and composition of the dataset prior to any classification task. This knowledge enables us to make informed decisions about suitable preprocessing techniques and classification methods. In this section, we examine key aspects of the dataset to which the classification algorithms have been applied.

The dataset comprises a total of 1000 instances, of which 100 are labeled as positive and 900 as negative. This substantial disparity between positive and negative instances indicates that the dataset is highly imbalanced, with only 10% of the instances belonging to the positive class. Such class imbalance can pose significant challenges for machine learning models, as they may tend to favor the majority class, leading to potentially misleading performance metrics. Specific techniques, such as resampling or specialized evaluation metrics, may be necessary to handle this imbalance effectively.

The table below presents the classification results obtained with five different methods. Each method's performance is measured through four key metrics: True Positives (TP), False Positives (FP), False Negatives (FN), and True Negatives (TN). These metrics provide insight into how accurately each method distinguishes between the positive and negative classes.

Method	TP	FP	FN	TN
A	100	900	0	0
B	80	125	20	775
C	25	25	75	875
D	50	50	50	850
E	0	0	100	900

Table 1. The classification performance of each method across these metrics. The variations in TP, FP, FN, and TN values highlight the differences in each method's approach to handling the imbalanced data and their respective tendencies toward false positives and negatives. These results underscore the importance of selecting the right method based on the application's sensitivity to misclassifications and the degree of imbalance in the dataset.

3 Metrics

In the performance analysis of classification models, especially in the context of disease prediction, it is essential to use metrics that allow evaluating different aspects of the accuracy and effectiveness of a model. These metrics help you understand how well the model correctly classifies positive and negative cases and also provide insight into what errors you might be making and the implications of these errors in a clinical context.

Below is a detailed description of the evaluation metrics that we are going to use in this project, explaining what it represents, its definition and the ideal values at which each one should be placed to optimize precision and minimize errors in the measurement classification. This will allow us to better interpret the results when we use these metrics to evaluate the methods used.

- **Precision:** measures the proportion of true positives (correct predictions of the positive class) against the total positive predictions made by the model. It is used to evaluate how accurate a positive model prediction is. It is particularly useful in contexts where false positives are costly, since a high value means that the model makes few such errors.
  A high precision value indicates that most positive predictions are correct, with 1 being the ideal value and 0 being the worst.

$$Precision(PR) : \frac{TP}{TP + FP}$$

  - **0.8 - 1:** high precision, Indicates that the majority of positive predictions are correct
  - **0.5 - 0.79:** moderate precision, the model is reasonably accurate in its positive predictions, but there may still be false positive errors.
  - **0 - 0.49:** low precision, the model incorrectly classifies many cases as positive

- **Recall:** measures the proportion of true positives out of the actual positives (true positives plus false negatives). It is used to evaluate the model's ability to identify all positive instances. A high recall is desirable in contexts where it is critical to detect all positive cases, as it minimizes false negatives.
  A high recall value means that the model correctly captures most of the positive cases, with 1 being ideal and 0 indicating very poor recall.

$$Recall(RC) : \frac{TP}{TP + FN}$$

  - **0.8 - 1:** high recall, indicates the model correctly identifies most positive cases
  - **0.5 - 0.79:** moderate recall, the model captures a reasonable amount of positive cases but may miss some
  - **0 - 0.49:** low recall, the model fails to identify many positive cases

- **Specificity:** measures the proportion of true negatives (correct predictions of the negative class) against the total actual negatives. It is used to evaluate how well the model identifies negative cases and avoids false positives. Specificity is particularly useful in contexts where it is important to correctly identify negatives, as in screening for rare diseases, where false positives can lead to unnecessary further testing.
  A high specificity value indicates that most negative cases are correctly identified, with 1 being the ideal value and 0 being the worst.

$$Specificity(SP) : \frac{TN}{TN + FP}$$

- **0.8 - 1:** high specificity, indicates that the model correctly identifies most negative cases and makes few false positive errors.
- **0.5 - 0.79:** moderate specificity, the model is reasonably accurate in identifying negatives, but may still produce some false positives.
- **0 - 0.49:** low specificity, the model incorrectly identifies many cases as positive, leading to a high rate of false positives.

– **False Negative Rate (FNR):** measures the proportion of actual positives that the model incorrectly predicts as negative. It is the complement of recall and indicates how many positive instances are missed by the model. A low FNR is ideal, especially in contexts where false negatives are highly detrimental.
A low FNR means that the model omits few positive cases, with 0 being the best value and 1 indicating the worst.

$$FalseNegativeRate(FNR) : \frac{FN}{TP + FN}$$

- **0 - 0.2:** low FNR, indicates that the model misses very few positive cases
- **0.21 - 0.5:** moderate FNR, some positive cases are missed, but it may still be acceptable
- **0.51 - 1:** high FNR, the model fails to capture many positive cases, which can be problematic in sensitive contexts

– **False Positive Rate (FPR):** measures the proportion of actual negatives that the model incorrectly classifies as positive. It reflects the model's tendency to make false positive errors. A low FPR is desirable, especially in applications where avoiding false positives is important.
A low FPR indicates the model makes few false positives, with 0 being the best value and 1 the worst.

$$FalsePositiveRate(FPR) : \frac{FP}{FP + TN}$$

- **0 - 0.2:** low FPR, the model makes few false positive errors
- **0.21 - 0.5:** moderate FPR, a moderate number of false positives occur, which may be tolerable in some cases
- **0.51 - 1:** high FPR, the model incorrectly classifies many cases as positive, which could lead to unnecessary interventions

– **Accuracy (ACC):** measures the percentage of correct predictions (both positives and negatives) relative to the total predictions made by the model. It provides a general measure of model reliability. While high accuracy indicates that the model is generally effective, it can be misleading in imbalanced datasets.
High accuracy suggests the model is generally dependable, with 1 as the ideal and 0 as the worst possible value.

$$Accuracy(ACC) : \frac{TN + TP}{TP + FN + FP + TN}$$

- **0.8 - 1:** high accuracy, the model is generally very reliable in its predictions
- **0.5 - 0.79:** moderate accuracy, the model is reasonably reliable but has some notable errors
- **0 - 0.49:** low accuracy, the model fails in the majority of its predictions, indicating the need for improvement

– **Spatial Accuracy (S) or Jaccard Index (J):** measures the overlap between the predicted and actual areas of interest (e.g., in image segmentation). It is critical in fields where location accuracy is essential, such as medical imaging. High spatial accuracy indicates a strong match between the predicted and actual regions.
A high value means that the model correctly identifies and locates most regions of interest, with 1 being ideal and 0 being the worst.

$$SpatialAccuracy(S) : \frac{TP}{TP + FN + FP}$$

- **0.8 - 1:** high spatial accuracy, the model correctly identifies the regions of interest
- **0.5 - 0.79:** moderate spatial accuracy, the model locates areas of interest but may miss some detail
- **0 - 0.49:** low spatial accuracy, the model poorly identifies the regions of interest, which may hinder its usefulness

– **F-measure (Fm):** is the harmonic mean of precision and recall, providing a balanced measure that considers both false positives and false negatives. It is particularly useful when both types of errors are costly or when there is a need for balance between precision and recall.
A high F-measure indicates a good trade-off between precision and recall, with 1 being ideal and 0 the worst.

$$F - Measure(Fm) : \frac{(2 * PR * RC)}{PR + RC}$$

- **0.8 - 1:** high F-measure, the model has a balanced and strong performance in both precision and recall
- **0.5 - 0.79:** moderate F-measure, the model has a reasonable balance but could be improved in precision or recall
- **0 - 0.49:** low F-measure, the model has significant weaknesses in either precision, recall, or both

4 Results and Analysis

We have developed an algorithm that calculates different metrics that allow us to measure the quality of each method. You can see the code implementation in the attached jupyter notebook called *Lab2.py*. After obtaining all the results, we can create the next table which represents all metrics applied to each method.

Metrics	**A**	**B**	**C**	**D**	**E**
Precision (PR)	0.1	0.3902	**0.5**	**0.5**	0
Recall (RC)	**1**	0.8	0.25	0.5	0
Specificity (SP)	0	0.8611	0.9722	0.9444	**1**
FNR	**0**	0.2	0.75	0.5	1
FPR	1	0.1389	0.0278	0.0556	**0**
Accuracy (ACC)	0.1	0.855	**0.9**	**0.9**	**0.9**
Spatial accuracy	0.1	**0.3556**	0.2	0.3333	0
F-measure (Fm)	0.1818	**0.5246**	0.3333	0.5	0

Table 2. Comparison of methods by metric with highlighted best values.

4.1 Precision

· The best methods found are **C** and **D** with **0.5**.
· The worst method is **E** with **0**.
· Methods C and D have a 0.5 precision, meaning they correctly identify half of the positive cases. Method E, however, fails to correctly predict any positive cases, resulting in a 0 precision.

4.2 Recall

· The best method found is **A** with **1**.
· The worst method is **E** with **0**.
· Method A has a perfect recall of 1, indicating it identified all positive cases. Method E does not identify any positive cases, resulting in a 0 recall.

4.3 Specificity

· The best method found is **E** with **1**.
· The worst method is **A** with **0**.
· Method E has perfect specificity with a 1, meaning it correctly classifies all negative cases. Method A has a specificity of 0, indicating it failed to correctly identify any negative cases.

4.4 FNR

· The best method found is **A** with **0**.
· The worst method is **E** with **1**.
· Method A has an FNR of 0, meaning it did not misclassify any positive cases as negative. Method E has an FNR of 1, meaning it misclassified all positive cases.

4.5 FPR

· The best method found is **E** with **0**.
· The worst method is **A** with **1**.
· Method E has a 0 FPR, meaning it did not misclassify any negative cases as positive. Method A has an FPR of 1, indicating it misclassified all negative cases as positive.

4.6 Accuracy

· The best method found are **C**, **D** and **E** with **0.9**.
· The worst method is **A** with **0.1**.
· Methods C, D, and E have high accuracy of 0.9, meaning they made many correct predictions. Method A has low accuracy of 0.1, indicating it made mostly incorrect predictions.

4.7 Spatial accuracy

· The best method found is **B** with **0.3556**.
· The worst method is **E** with **0**.
· Method B has the highest value of 0.3556, indicating a good balance between true positives and true negatives. Method E has a Jaccard index of 0, indicating no overlap between correctly predicted positives and negatives.

4.8 F-measure

· The best method found is **B** with **0.5246**.
· The worst method is **E** with **0**.
· Method B has the highest value of 0.5246, showing a good balance between detecting positives and minimizing false positives. Method E has an F-measure of 0, indicating poor performance in both precision and recall.

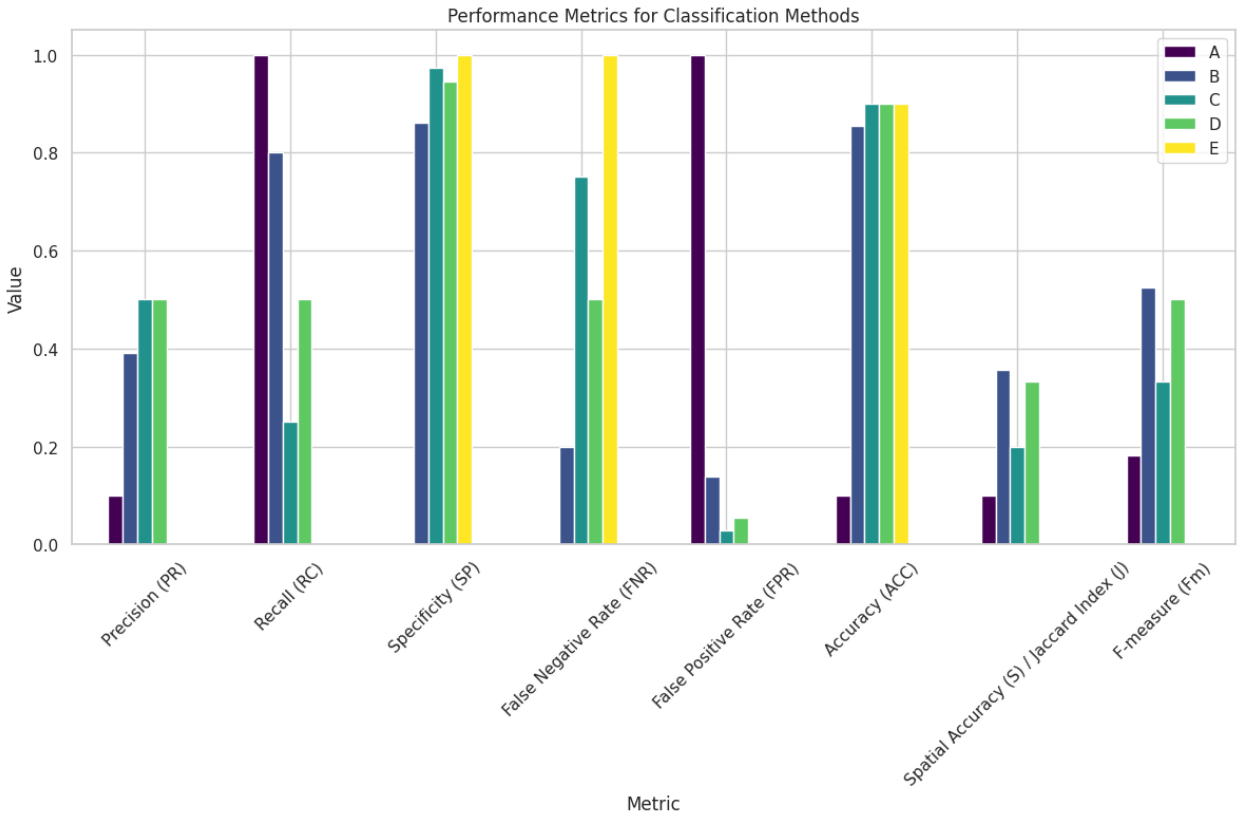

Fig. 1. Visual Results of Performance Metrics

From the Jupyter notebook code, Figure 1 shows metric values per method. Precision, Spatial Accuracy, and F-measure are generally low, while Specificity and Accuracy are high—except for Method A, with near-zero values. Figures 2, 3, and 4 illustrate key metric relationships.

In Figure 2, the **Accuracy (ACC)** is plotted against the **F-measure (Fm)** for each method. Higher values for both metrics indicate better performance. The best method in this comparison is Method B, as it has the highest combination of accuracy and F-measure, indicating a good balance. In contrast, Method E has the lowest values for both metrics, indicating it is the worst performer, as it has neither high accuracy nor a high F-measure score.

Furthermore, in Figure 3, the **False Negative Rate (FNR)** is plotted against the **False Positive Rate (FPR)**. Lower values on both axes indicate better performance, as lower FNR and FPR values mean the model makes fewer errors in both false positives and false negatives. Method B appears to be the best method, as it achieves a good balance between FNR and FPR. D also presents a balanced result with intermediate values on both axes, so it could also be a reasonable choice depending on the application.

Finally, in Figure 4, **Precision (PR)** is plotted against **Recall (RC)**. A higher position in both metrics is desirable because it reflects better ability to identify true positives accurately and consistently. Method B appears to have the best balance between precision and recall. It shows relatively high values on both axes, which means it has fewer false positives. Conversely, Method E is the worst performer here, as it has the lowest values for both metrics, indicating poor precision and recall.

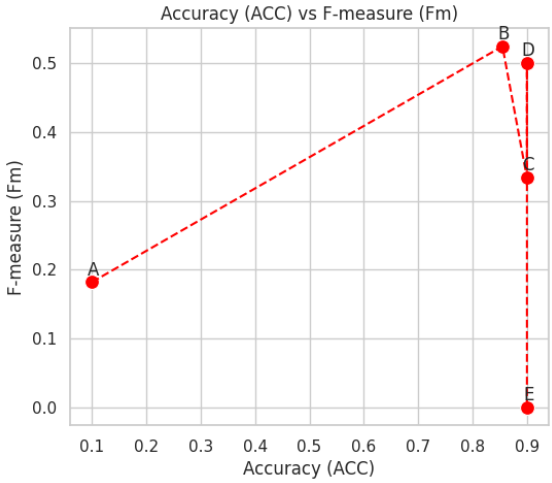

Fig. 2. ACC vs Fm

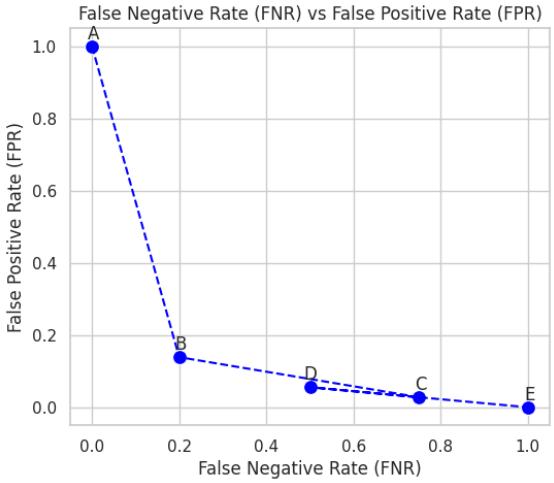

Fig. 3. FPR vs FNR

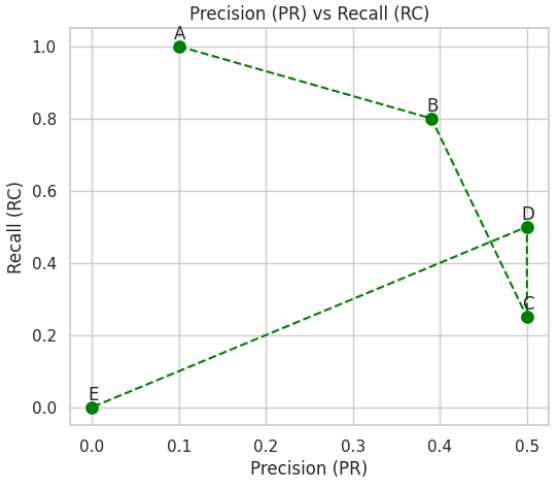

Fig. 4. RC vs PR

5 Conclusion

After examining the results, we can conclude that **Method B** is the best overall, while **Method E** is the least effective. Below, we summarize the reasons for these conclusions.

Method B stands out as the best overall performer. It achieves high accuracy and the highest F-measure, reflecting a strong balance between precision and recall. Additionally, **Method B** has low FNR and FPR values, indicating that it minimizes both types of classification errors. This makes **Method B** the most reliable choice for applications where a balanced approach to false positives and false negatives is essential.

Method D also shows competitive results, particularly in precision, where it outperforms **Method B**. This higher precision indicates that **Method D** is effective at reducing false positives, which may be advantageous in scenarios where it is critical to avoid false alarms, even if it means missing some true positives. However, **Method D**'s slightly lower F-measure and higher FNR compared to **Method B** make it a secondary choice when recall is equally important as precision.

Methods A and C show mixed performance. **Method A** achieves perfect recall, meaning it captures all positive cases, but this comes at the cost of high false positive rates and low precision, which may not be suitable for many applications. **Method C** has moderate accuracy and specificity but suffers from lower recall, making it appropriate only in situations where missing some positives is acceptable.

Finally, **Method E** is the poorest performer, with low values in both accuracy and F-measure, as well as high FNR. This indicates that **Method E** struggles to identify positive cases effectively and is therefore unsuitable for applications requiring reliable classification.

In summary, **Method B** is recommended as the most balanced and effective choice, with **Method D** as a viable alternative when high precision is prioritized. **Methods A and C** may be useful in specific cases, but **Method E** is not recommended due to its consistently poor performance across metrics.