UNIVERSITAT POLITÈCNICA
DE CATALUNYA
BARCELONATECH

Machine Learning I

Data Science and Engineering

Binary Classification for critically ill patients

Anna Esteve Gallifa & Biel Altimira Tarter

Spring semester 2024

Contents

List of Figures

List of Tables

1 Introduction

This report aims to leverage the power of the SUPPORT2 dataset and the machine learning techniques learned in class to perform proper assessment on critical illness and patient outcomes. By diving deeper into the data, we can potentially enhance clinical decision-making and ultimately improve patient care. The following sections briefly outline the assets we are working with and our goals for this project.

1.1 Dataset

The SUPPORT2 [1] dataset *(Study to Understand Prognoses Preferences Outcomes and Risks of Treatment)* is a collection of 9105 instances carried out by the Vanderbilt University Department of Biostatistics, funded by the Robert Wood Johnson Foundation and made publicly available in 2023.

Each of these instances represent a critically ill patient who met the acceptance criteria to be addimted into a study regarding multiple categorized diseases. The studied individuals were adults, hospitalized throughout 1989-1994 and from five United States medical centers.

The raw version of this dataset comprises 47 features of mixed data types that will further be inspected for some feature selection. Nonetheless, it must be noted that these features provide insight on physiologic, demographics, and disease severity traits of the patients.

1.2 Objectives

Our central goal will be to perform a binary classification of the patient mortality risk based on their health status, this involves estimating the *death* variable. The reasoning behind this, is that it is of the upmost importance to be able to empower the health infrastructure to make more informed decisions regarding the most suitable procedures for the patients. Ensuring to reduce mechanical and painful dying processes, as well as maximise the well-being of patients suffering from extreme conditions.

From this rationale, we have extracted some secondary and more broken down targets:

1. Refine the given data by performing the appropriate prepossessing securing the most information.
2. Apply the learned core concepts to find a suitable and optimal solution for the posed problem.
3. Maintain the focus on interpretation of the obtained results.
4. Address the inherent ethical considerations that emerge from utilising machine learning models in such context.

1.3 Methodology

To achieve the objective of accurately classifying the mortality of several patients, we have meticulously followed the typical structure of machine learning projects. Therefore, this section digs into the techniques employed for this analysis, from exploring our dataset to shaping our data into the desired format, concluding with the comparison between different models and the evaluation of the best performing one.

2 Exploratory Data Analysis

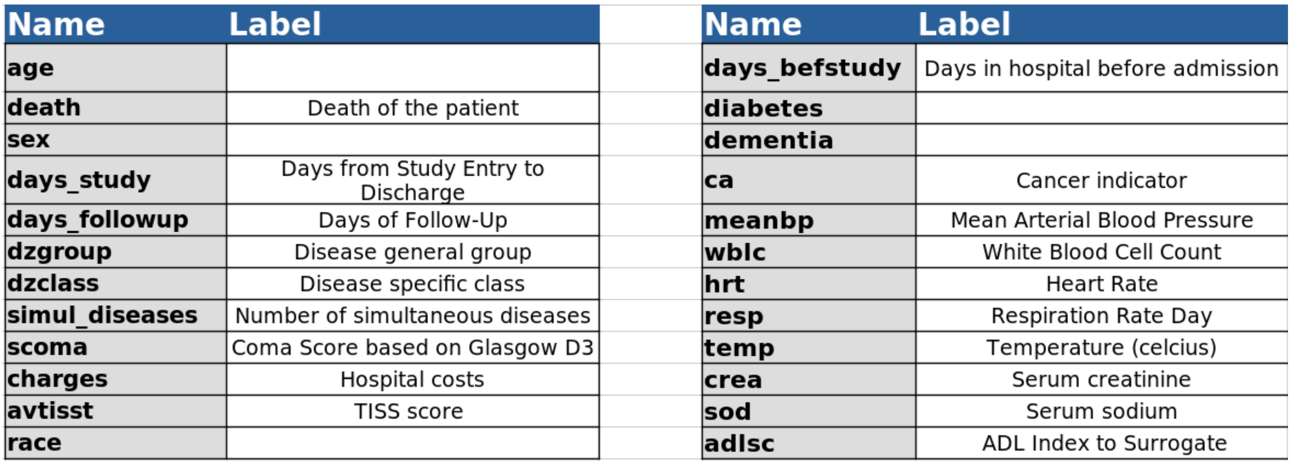
2.1 Feature Selection

It can be easily noted that the original dataset has a rather large amount of features that, nonetheless, can be simplified into a more manageable set. To start with, from the dataset documentation we are told not not use the following variables for any predictions, since they are only useful when using findings from previous models:

"aps", "sps", "surv2m", "surv6m", "prg2m", "prg6m", "dnr", "dnrday"

Next, we have a large amount of features that contain a huge and unreasonable amount of null values. The reasoning behind this, is that most of this variables either provide from nurses/patient/relatives input in the form of evaluation of the patient or seem to be extra information, thus not relevant. Since the loss of information from removing those instance is not worth it, neither is a possibility to find an accurate imputations for those values, we remove the features with more than 5% of NaN's.

We end up with a final set of 24 features representing:



Name	Label	Name	Label
age		days_befstudy	Days in hospital before admission
death	Death of the patient	diabetes	
sex		dementia	
days_study	Days from Study Entry to Discharge	ca	Cancer indicator
days_followup	Days of Follow-Up	meanbp	Mean Arterial Blood Pressure
dzgroup	Disease general group	wblc	White Blood Cell Count
dzclass	Disease specific class	hrt	Heart Rate
simul_diseases	Number of simultaneous diseases	resp	Respiration Rate Day
scoma	Coma Score based on Glasgow D3	temp	Temperature (celcius)
charges	Hospital costs	crea	Serum creatinine
avtisst	TISS score	sod	Serum sodium
race		adlsc	ADL Index to Surrogate

Figure 1: Selected features and brief description.

2.2 Main Concerns

From observing the selected features a few concerns raised. Firstly, with the help of the features histogram (8) we were able to detect skewness in some variables, this will need to be corrected since many models rely on the normality of the features as a statistical assumption. Moreover, we detected the presence of many outliers, causing the distributions to be uninterpretable. Finally, our target values seems to be imbalanced, as a consequence of having a sampling bias and most of the patients resulted to be dead. To properly handle this issue, we will need to perform oversampling or undersampling or use the appropriate performance metrics that will not be biased from this fact.

3 Preprocessing

Preprocessing the data is a major step of the machine learning pipeline to achieve accurate predictions. The following key points display the process of converting the original dataset into readable data that can be fed into the mathematical models and grants proper generalization from the obtained sample.

3.1 NaN Value Cleaning

Firstly, the removal of null values must be taken care of. It is important to recap that from the 47 original features, we are left 24 after feature selection, where we removed features with more than 5% of NaN values. Also, all the NaN values are indeed marked as NaN, thus identified by Pandas, after the inspection carried out in the exploratory data analysis, and no null values were found on the target variable.

Then, as shown in figure 2, some columns still have a reasonable amount of NaN's to be imputed. For this task, we created a custom *Scikit-Learn* transformer that utilises the *KNNClassifier* or the *KNNRegressor*, depending on the feature type working on, to make a prediction for the target variable containing NaN's based on the other predictors. It must be noted that we found 51 instances containing more than one feature with a null value, we decided to drop them to be able to apply this method.

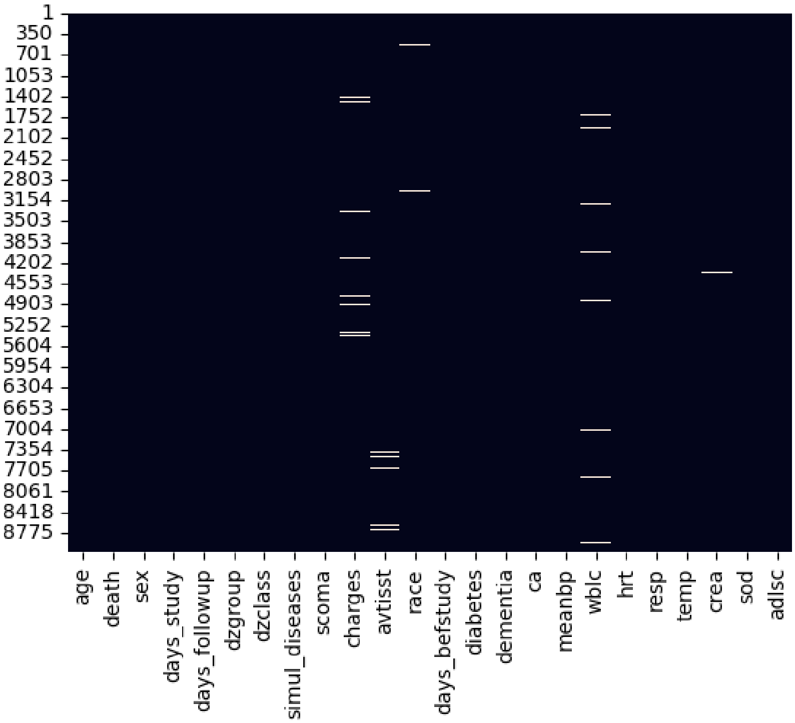


Figure 2: Visual representation of NaN values (white strips) for each feature.

At this point, to prevent any data leakage, we splited our data into a *train / test set* following a 80/20 proportion and trained our imputer on the NaN free instances of the training set. As follows, we used the trained *KNN* model to impute the values of both train and test set and repeating for every feature that needed handling.

We concluded with a total of 7243 instances for the train set and 1811 for the test set, completely null-free.

3.2 Outlier Detection

Once we have removed all the missing values in the dataset, we are going to identify the outliers to remove them, impute them or just acknowledge their existence and take into account in your analysis, depending on our interests.

To start, we define the outliers by using the IQR criteria. Then the outliers are going to be the ones that are smaller than Q1 - 1.5*IQR or bigger than Q3 + 1.5*IQR.

After calculating the outliers of each feature, we are going to remove this values and compare the new histogram with the old one. We can see that most of them adquire a more gaussian shape and a wider dinamic range. For the features that don't get this impact and mantain each former shape we won't take the outliers because they don't cause any interfence on the prediction.

For these reasons we are going to take the outliers of the categories: *"days_study", "charges" , "avtisst" , "days_befstudy" , "meanbp" , "wblc" , "hrt" , "resp" , "crea"* and *"sod".*

3.3 Gaussianity and transformations

We analyse the histograms and boxplots of the continous and categorical variables that we have been modifying. We observe that most of them follow a normal distribution, except some of the feature that are categorical which is natural because we have different number of exemplars of each category.

Even though there are few features that don't really obey a gaussian distribution. So we consider making transformations of the data. After analysing the corresponding plots we observe that *"charges"* would need a change of scale. So we will take logs for this feature.

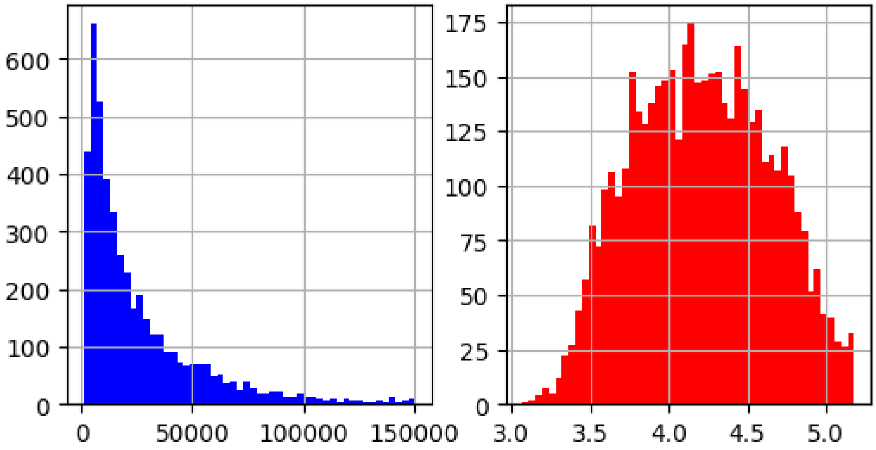


Figure 3: Transformation of the *charges* variable.

3.4 Feature Encoding

As we have different kinds of data (floats, integers and categorical) we make some changes to obtain a more comfortable dataset.

In the column of *"ca"*, which is a cancer indicator we change the categorical values for numerical indicators to make the prediction easier. In this case, we can do it because not having cancer, having cancer and having metastasis has a hierarchical meaning. In this way we change categories *"no", "yes" and "metastatic"* for *0, 1, 2.*

For the rest of the features we have observed that the categorical variables don't follow any ordinal relation, and we still have a reasonable amount of features to work with, so we have opted to perform One Hot encoding. This way, we ensure maintaining the relations between the categories at the expense of increasing the dimensionality. This means, for each categorical feature we detected, we substitute it by n boolean features indicating if the instance belongs to each of the n possible categories.

After this transformation, we end up with 39 features, all of them have a numerical type indicating discrete values, continuous measurements or booleans reproducing the categorical variables that needed to be addressed From now on, all the transformations that only supported numerical features can be applied.

3.5 Normalization

To avoid that some features adquire more importance for its high values we normalize or standarize our data. In this way, we force to have the same range for each variable.

We use the min-max scaling, that sends our data to the range [0,1].

4 Modeling

For the following section, after preprocessing our data, we have selected several algorithms and tuned their parameters to achieve optimal fit on the dataset. As we discussed, our main goal is to predict weather a patient will decease in the near future based on the selected features that mainly represent physical conditions and disease severity. Therefore, the model should be able to capture and generalize the records from the dataset in order to be able to predict the outcome for new critically ill patients. Nonetheless, when dealing with such a sensitive issue, many aspects need to be taken into consideration to ensure our model will only provide a benefit towards the patient's life

4.1 Modeling Methodology

Firstly, we must state what needs to be considered as a valid model. Our main goal is to classify, however, we need to contextualise our model based on the data we have and the challenge we are addressing. Regarding the first one, as observed in the exploratory data analysis, we are dealing with an imbalance on the target variable, most of the observations are patients who did not survive. Therefore, the accuracy metric is not reliable for this situation, we need to take into account a balance between precision and recall, that's why the F1 score would be suitable.

However, regarding the second piece of context, we need to take into account we are asserting mortality risk, mainly to provide more exhaustive care to those patients with a more severe life-threatening situation. For this reason, recall should have a greater impact on what we consider as a suitable model. We aim to correctly classify patients in terminal stages to allocate increased resources to them. This change may enable patient recovery, even if it means some less critical patients are considered for more intensive care.

From this last interpretation we derived that the most suitable metric was the F-beta score, with $\beta = 2$, where instead of having a harmonic mean between precision and recall, we double the weight on recall:

$$F_\beta = \frac{(1+\beta^2)\mathrm{TP}}{(1+\beta^2)\mathrm{TP} + \mathrm{FP} + \beta^2\mathrm{FN}}$$

To study the different models, we will also include some additional metrics that will provide supplementary insights on each model. To validate the models we will use cross validation to get an estimate of the generalisation performance by just using the training set. Both linear, nonlinear, instance and model based, ensembling and boosting methods will be explored. For each model, hyperparameter tuning will be performed to ensure the best possible performance from the gathered data. After that, all the models will be compared to select the most suitable one and the generalisation error will be estimated thorough the test set to obtain the final conclusions.

4.2 Linear Discriminant Analysis

The first model we will try is Linear Discriminant Analysis (LDA). LDA is a statistical method used for classification and dimensionality reduction, aiming to find a linear combination of features that best separates two or more classes. In other words, it projects data onto a lower-dimensional space, enhancing class separability and reducing computational costs.

LDA's key characteristics include its linear decision boundary, probabilistic approach and assumptions of normally distributed data with identical covariance matrices of the classes. This model is simple to implement and interpret, computationally efficient, and performs well with linearly separable data. Additionally, it helps mitigate the curse of dimensionality by reducing the feature space.

However, LDA relies heavily on its assumptions about data distribution. When these assumptions are not met, the performance can be suboptimal and sensitive to outliers. Despite this, our training data has been treated and normalized, resulting in good metrics when fitting the model to this data. Nonetheless, we cannot consider this model adequate for real-world data, as the assumptions are not fully met in the actual data.

Model	Accuracy	Precision	Recall	F1 Score	F-beta Score	ROC-AUC	Log Loss
LDA	0.838286	0.854327	0.916133	0.884044	0.902994	0.932521	0.349181

Table 1: Metrics of LDA

4.3 Quadratic Discriminant Analysis

The second model we will explore is Quadratic Discriminant Analysis (QDA). QDA is a statistical method utilized for classification, designed to identify a quadratic combination of features that optimally separates multiple classes. In contrast to Linear Discriminant Analysis (LDA), QDA provides a more flexible decision boundary by allowing each class to have its unique covariance matrix.

QDA's distinguishing characteristics include its quadratic decision boundary, probabilistic framework, and the assumption of normally distributed data with different covariance matrices for each class. This approach is particularly positive in scenarios where the classes exhibit different shapes and orientations within the feature space, enabling it to capture more complex relationships between features.

Nonetheless, QDA is highly dependent on the validity of its assumptions regarding data distribution. That is why it offers even worse classification capabilities compared LDA. So we remain with the results of LDA and keep trying other models.

Model	Accuracy	Precision	Recall	F1 Score	F-beta Score	ROC-AUC	Log Loss
QDA	0.700482	0.848630	0.676726	0.751231	0.704370	0.794206	2.901450

Table 2: Metrics of QDA

4.4 K-nearest neighbors

The following model that we have explored is the K-Nearest Neighbors (KNN) algorithm. KNN is a non-parametric, instance-based learning method used for classification and regression. It operates by finding the 'k' closest data points (neighbors) in the feature space to a given query point and making decisions based on the majority class among these neighbors.

KNN's key features include its simplicity, easy of implementation, and flexibility in choosing different distance metrics. We experimented with various distance metrics. The ones we have studied are attached in the annex 8.2.

Additionally, we have tested different values for 'k', the number of neighbors considered. After testing, we determined that using the Manhattan distance metric with 20 neighbors yielded the best performance.

The distinguishing characteristics of KNN include its lack of assumptions about the underlying data distribution and its ability to adapt to complex decision boundaries as defined by the local neighborhood of data points. For these reasons has performed better than QDA. However, KNN's performance can be significantly affected by the choice of 'k' and the distance metric, and it can be computationally expensive, especially with large datasets. And LDA keeps having better results than KNN.

Model	Accuracy	Precision	Recall	F1 Score	F-beta Score	ROC-AUC	Log Loss
KNN	0.808573	0.829724	0.900450	0.863540	0.885286	0.868093	0.485365

Table 3: Metrics of KNN

4.5 Gaussian Naive Bayes

The next model we will explore is Gaussian Naive Bayes (GNB). GNB is a probabilistic classification technique that assumes the features follow a normal (Gaussian) distribution. This method is built on Bayes' theorem, which provides a way to update the probability estimate for a hypothesis as more evidence or information becomes available.

GNB's distinguishing characteristics include its simplicity, speed, and scalability. It assumes that all features are GNB performs surprisingly well when the number of features is large, this assumption is actually not real in our case and that is why it gives a not really good result.

For this reason, as GNB is highly dependent on the validity of feature independence and normal distribution, GNB returns suboptimal results.

Model	Accuracy	Precision	Recall	F1 Score	F-beta Score	ROC-AUC	Log Loss
Gaussian Naive Bayes	0.637814	0.882662	0.532335	0.664076	0.578212	0.842532	1.871467

Table 4: Metrics of GNB

4.6 Logistic Regression

The following model that we have tried is Logistic Regression. It is a linear model for binary classification. It predicts the probability that a given input belongs to a particular class by modeling the relationship between the dependent binary variable and one or more independent variables using a logistic function.

Logistic Regression is based on the assumption that the log-odds of the probability of the event of interest is a linear combination of the independent variables.

This approach is particularly beneficial when the relationship between the features and the target variable is approximately linear. Logistic Regression provides not only a classification output but also probabilities, which can be useful for understanding the uncertainty of predictions.

However, it assumes a linear decision boundary, which may not capture more complex relationships between features and the target variable. Additionally, it can be sensitive to outliers and irrelevant features. Despite these limitations, Logistic Regression works quite well in this dataset and for the moment gives the best metrics results.

Model	Accuracy	Precision	Recall	F1 Score	F-beta Score	ROC-AUC	Log Loss
LDA	0.838286	0.854327	0.916133	0.884044	0.902994	0.932521	0.349181

Table 5: Metrics of LDA

4.7 Stochastic Gradient Descent

The sixth model we tested is Stochastic Gradient Descent (SGD). SGD is an iterative optimization technique used to minimize functions, particularly useful for training linear models. It processes one training example at a time, making it efficient for large datasets. So this method is particularly advantageous when computational resources are limited and if we are working with real-time or streaming data.

However, the method requires meticulous tuning of hyperparameters like the learning rate, and it can be quite sensitive to these choices. Additionally, it may not always converge to the global minimum, particularly in noisy datasets, and is susceptible to the influence of outliers. These limitations affect its performance on our medical dataset and it is reflected on the results.

Model	Accuracy	Precision	Recall	F1 Score	F-beta Score	ROC-AUC	Log Loss
SGDC	0.852498	0.910678	0.866843	0.887516	0.874837	0.925462	-

Table 6: Metrics of SGD

4.8 Support Vector Classifier

The seventh model has been a support vector classifier, which aims to fit a hyperplane that maximizes the separation of our data. This is still suitable for non-linearly separation data (our case) by using the kernel trick. After testing multiple hyperparameters, the Gauissan kernel has proven to be the most suitable. In addition, we have set the regularization parameter C to 100, which attempts to reduce the overall classification errors at the expense of a smaller margin of separation.

Model	Accuracy	Precision	Recall	F1 Score	F-beta Score	ROC-AUC	Log Loss
SVM	0.808573	0.829724	0.900450	0.863540	0.885286	0.868093	0.485365

Table 7: Metrics of SVM

4.9 Decision Tree

The eighth model we explored is the Decision Tree. Decision Trees are intuitive, non-parametric models used for both classification and regression tasks.

Decision Trees are significant for their simplicity and interpretability, making them easy to visualize and understand. They are effective at identifying non-linear relationships within the data, making them suitable for complex datasets. However, they tend to overfit the training data. They are also sensitive to small variations in the data, which can result in different tree structures. These drawbacks make them less reliable for our dataset compared to more advanced models.

Model	Accuracy	Precision	Recall	F1 Score	F-beta Score	ROC-AUC	Log Loss
Decision Tree	0.859823	0.902543	0.887648	0.894973	0.890549	0.915223	1.814702

Table 8: Metrics of Decision Tree

4.10 Random Forest

The ninth model we evaluated is Random Forest. Random Forest is an ensemble technique that builds multiple decision trees and merges their predictions to enhance accuracy and control overfitting.

Random Forest excels in robustness against overfitting due to its ensemble approach and its capability to handle a large number of input variables without variable deletion. Random Forest also helps us figure out which features are important, helping us understand the data better. But, it can take a lot of computing power and might not be as easy to understand as single decision trees. Even with these difficulties, Random Forest has done really well, so it's worth looking into more.

Model	Accuracy	Precision	Recall	F1 Score	F-beta Score	ROC-AUC	Log Loss
Random Forest	0.875971	0.932049	0.879960	0.905166	0.889852	0.947727	0.277282

Table 9: Metrics of Random Forest

4.11 Gradient Boosting

The tenth model we investigated is Gradient Boosting. Gradient Boosting constructs an ensemble of trees in a sequential manner, where each tree corrects the errors of the previous ones, leading to a powerful predictive model.

Gradient Boosting stands out for its high accuracy and excellent balance between precision and recall, indicating a strong overall performance. It is particularly effective in handling complex data patterns and improving predictive performance through careful adjustment of biases and variances. However, it requires significant computational resources and careful tuning to avoid overfitting, which are important considerations for its practical application. After choosing the proper parameters from performing a grid search we obtained the highest F-beta score, hence potentially choosing this as our definitive model.

Model	Accuracy	Precision	Recall	F1 Score	F-beta Score	ROC-AUC	Log Loss
Gradient Boosting	0.893413	0.943646	0.895327	0.918620	0.904448	0.960342	0.227130

Table 10: Metrics of Gradient Boosting

4.12 Ada Boosting

The eleventh model we considered is AdaBoosting. AdaBoosting, or Adaptive Boosting, is an ensemble method that focuses on improving the performance of weak classifiers by adjusting their weights based on classification errors.

AdaBoosting is known for its strong ability to enhance model performance by converting weak learners into a robust ensemble. It demonstrates a good balance between precision and recall, making it reliable for complex datasets. However, it is sensitive to noisy data and outliers, which can negatively impact its performance. Despite these issues, AdaBoosting has shown potential as a reliable model for our dataset, but Gradient Boosting's results keep being slightly more accurate.

Model	Accuracy	Precision	Recall	F1 Score	F-beta Score	ROC-AUC	Log Loss
AdaBoost	0.880922	0.924912	0.895965	0.910048	0.901508	0.952212	0.645760

Table 11: Metrics of AdaBoost

5 Model Evaluation

Considering all the models, we have obtained the table accesible at 8.3 with all the metrics. Following the criteria we stated, our model selection will be based on the F-beta score. This criteria is fulfilled by the Gradient Boosting model, also providing the lowest log loss, highest ROC-AUC, F1 score and accuracy.

We observe how QDA and Gaussian Naive Bayes performed very poorly due to the statistical asumptions not being correct. On the other hand SGDC proved to be fast and suitable for large amounts of data but not appropiate for the precision we are looking for. The rest of the models had an overall good performance, however, the most suitable one has been the Gradient Boosting. Many other models has a slightly higher recall, which is utimately what we are interested in, however, it was at the expense of a much lower precision, a tradeoff we will not consider.

The hyperparameters we have set for the model are a 0.1 learning rate, robust to prevent overfitting, since gradient boosting is based on decision trees, which are very prone to overfit the data. In addition, we have set a mas depth of 3 levels for each individual tree and a they require at least 2 samples for each leaf node. Moreover, we have a total of 200 estimators or trees to be used in the ensemble.

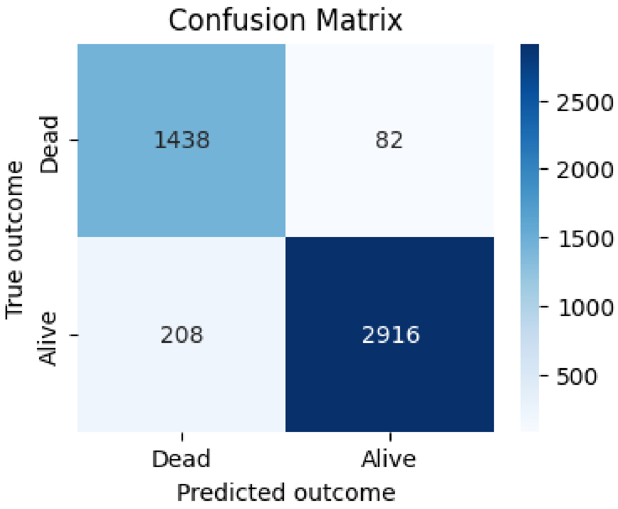


Figure 4: Confusion matrix of the GBC on the training data.

The overall fit on the data yields the following confusion matrix. It indicates that over all the 4644 records of patients, or model is able to correctly classify 93.7% of them. Leaving only 82 patients classified as survivors that actually died. That is a 1.76% of patients that could recieve potential harm from this model. The remaining 208 false negatives do not symbolize any significant drawback of our system.

However, our model cannot be properly assesed using the training data, it is time to utilize the test set, which containts some patients which the model has not seen yet. The resulting confusion matrix is the following:

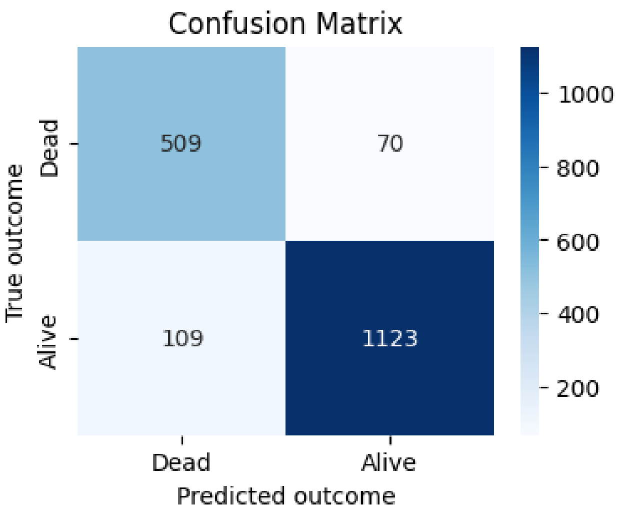


Figure 5: Confusion matrix of the GBC on the test data.

The interpretation is that for a set of 1811 new patients, our model would be able to correctly classify 90% of them. Leaving around 3.8% patients classified as alive whereas their medical conditions didn't allow them to survive. The follow table summarizes all the metrics for the test set, which shows a generally good result. The metrics already showed accurate classification rate for the training set, indicating that no underfitting was being produced. After obtaining also good performance on the test set, we can conclude that no overfitting is being produced either and our model has a good generalization performance.

Accuracy	Precision	Recall	F1 Score	F-beta Score	ROC-AUC
0.9011	0.9413	0.9115	0.9261	0.9173	0.8953

Table 12: Test metrics of GBC

5.1 Target Balancing

Finally, we also tried to perform different techniques in order to balance the target variable to have the same number os samples on each catagory. The first method makes the asumption we have many samples and we can eliminate the extra observations from the majority class, it's called undersampling. The second method is oversampling, where by using the *SMOTE* module we generate values for the minority class.

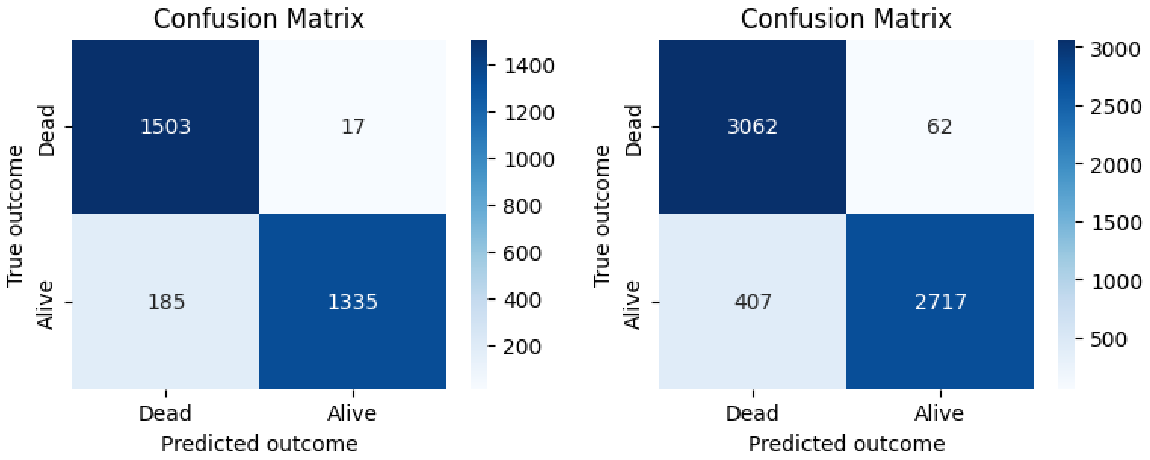


Figure 6: Confusion matrix on test of model trained on undersamples and oversampled data

The results show a much better identification of the critical cases, which we considered to be the false negatives. We make these critical mistakes with only 1% of the test set patients using undersampling and 1.6% using oversampling. This means that our features are descriptive enough to still be representative despite highly reducing the number of samples. Also, synthetical samples improve with regard to the base unbalanced dataset but perform worse compared to undersampling. It must be highlighted how the rest of the metrics are also very appropriate and do not deviate much from the original training set.

Accuracy	Precision	Recall	F1 Score	F-beta Score	ROC-AUC	Log Loss
0.903289	0.958965	0.842763	0.896259	0.86023	0.953915	0.241738

Accuracy	Precision	Recall	F1 Score	F-beta Score	ROC-AUC	Log Loss
0.903175	0.959649	0.842518	0.897089	0.863381	0.96363	0.225709

Table 13: Undersampling (above) and oversampling (below) metrics

6 Conclusions

In closing, we have been able to obtain a very accurate and precise classifier from the original raw data. We preprocessed the initial input to select the appropriate features, remove the incorrect values and outlying observations, applied suitable transformations, scaled and encoded the remaining data to be able to feed it into the different algorithms. All in all, we have been able to try and compare multiple linear, nonlinear, model-based, instance-based, and different kinds of ensembling algorithms to chose the most appropriate one. Finally, we estimated the generalization error and derived these conclusions.

For instance, we demonstrated that it is crucial to deeply analyze the problem that needs to be addressed. This will allow to find the suitable metrics, prior models and help to take an objective-based decision making. This is observable in our project from the shift to focusing on recall rather than any other metric. Because what we were trying to achieve was to find which of the critically ill patients had a more life threatening situation and would need intensive care and more resources in order to prevent a painful dying process.

This last statement also brings to the fore another main conclusion drawn. Machine learning can really have an impact on real world situations and be of great help, but in some cases, incorrect decisions while modeling in a sensitive context can result in violating certain ethical constrains. For example, if we decided to focus on a more precision-based metric instead of recall, we would be prioritizing making correct predictions. Therefore, our model would only be considered useful when trying to give resources only to patients that would survive and forget about dying patients which would be considered as not worth the try.

Also, in a real world situation a machine learning model cannot be used as a sole decision making instrument. It should be a source of information to contrast with many others by the field experts. Manly because despite obtaining great classification metrics, the model still makes bold decisions and would leave 17 patients without help, since they would be considered to survive, when in reality they wouldn't.

On the more technical side, the last part of the report outlined the great importance of treating imbalance. We obtained a much better performance on the balanced training sets despite seeming like an unharmful event at first and considering we were using the appropriate metric for treating an unbalanced dataset. In addition, we were able to obtain such good results since we relied on a great set of features that traded for the loss of instances when undersampling. In a parallel case where we didn't have such descriptive features, a greater reliance on data would be needed. In such case, oversampling could still provide great results too.

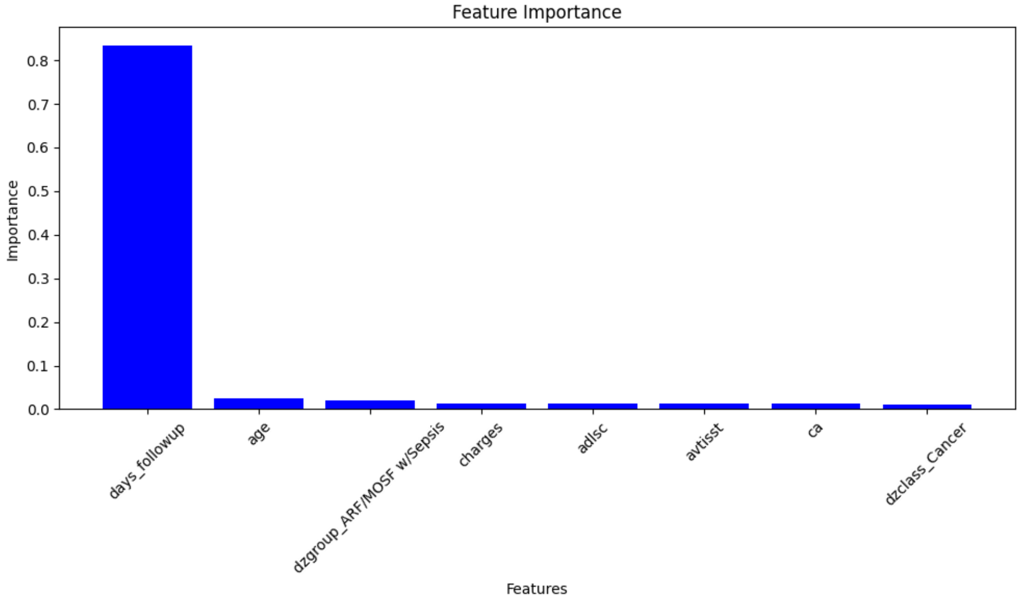


Figure 7: Feature importance based on the GBC method

Finally, we must also focus on the interpretability of our models, considering the prevalence of machine learning in critical decision-making processes, such as the one studied, interpretability becomes key. It ensures that these automated choices are fair, unbiased, and align with ethical values. Figure 7 shows the importance our model is granting to each variable. In other words, it gives us a sense of how the predictions are made and how the model is reasoning.

For the most part, it's taking into account the days of followup of the patient. Therefore, our model mainly relies on the days the patient has been continuously being inspected and tested after finishing treatment. This might indicate that the patients for the dataset were not properly being cured or given the necessary treatments for survival.

The rest of the variables show importance towards age, older patients had a higher risk of not surviving their conditions, MOSF with sepsis, indicating a general failure of many organs with infections and also cancer had a major correlation as expected.

It is also of the upmost importance to note how the charges variable has a great impact on weather a patient will decease. Coming from a private healthcare system in the USA, patients with more economical power are able to afford more treatments and resources which derive in more odds of surviving a harmful condition. Also, the adlsc feature gives high importance to the input from the nurses or patient relatives in a more subjective assessment of the patient.

All in all, machine learning techniques are valuable for decision-making, especially in complex models where interpreting results can be challenging. They create a powerful synergy when guided by humans, leading to more informed and ethical decisions. However, it's crucial to remember that these tools should support human judgment, not replace it entirely, as humans still have a significant influence over decision-making and should continue to do so.

7 References

[1] Frank Harrel. *SUPPORT2*. 2023. URL: https://archive.ics.uci.edu/dataset/880/support2.

8 Appendix

8.1 Exploratory Data Analysis

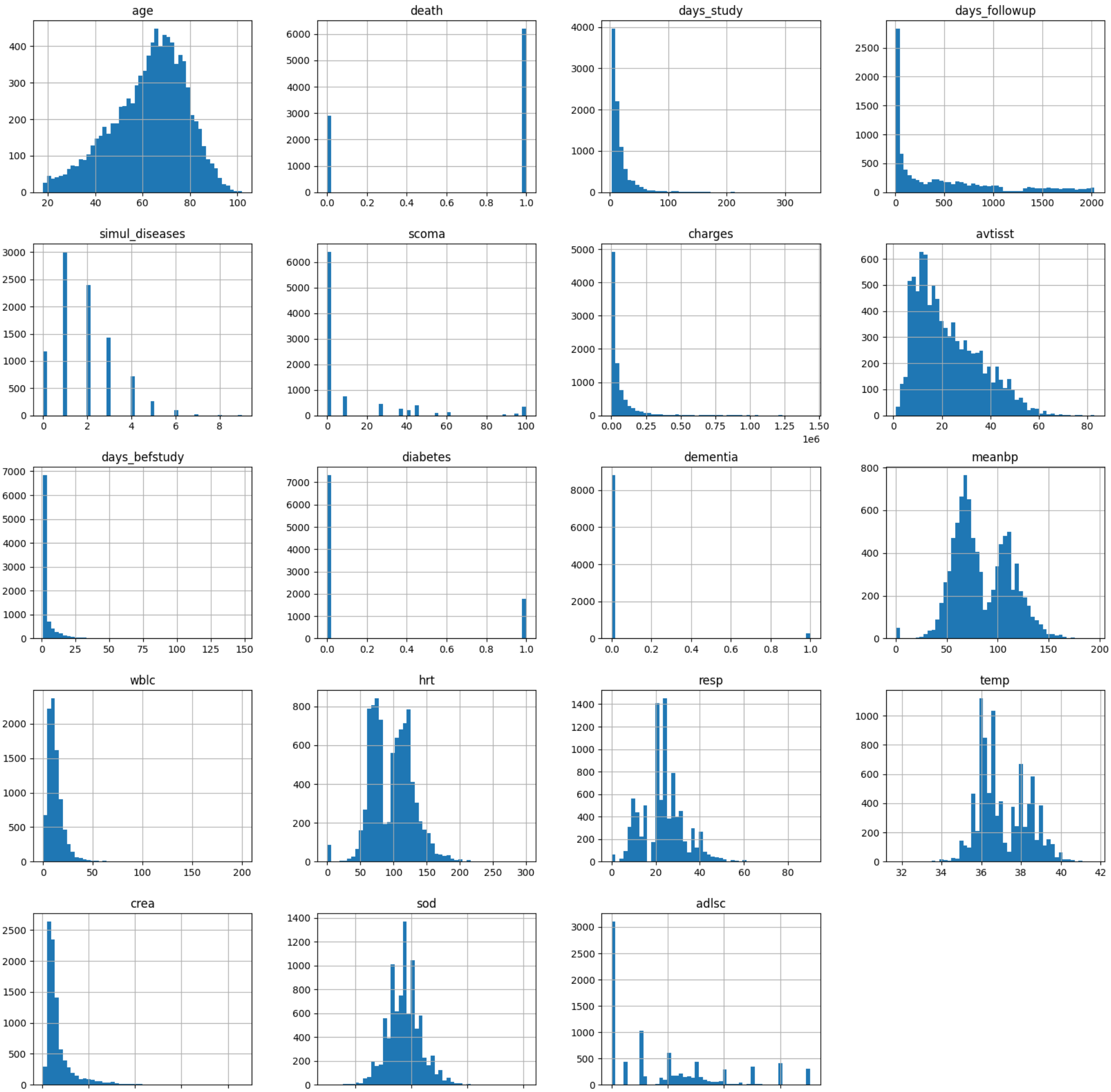


Figure 8: Histogram plots of the features before preprocessing

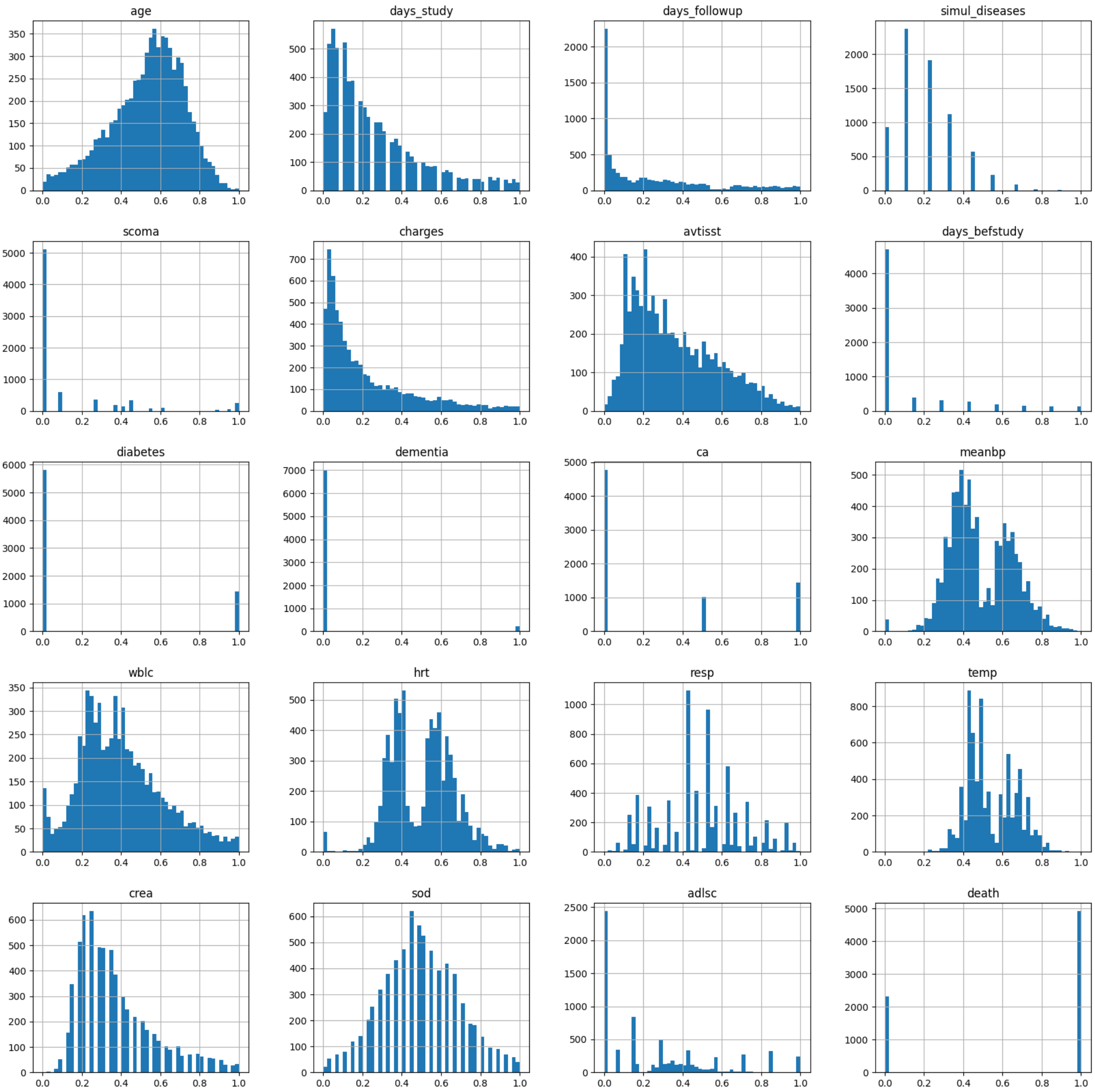


Figure 9: Histogram plots of the features after preprocessing

8.2 Distances

- **Manhattan Distance (L1 Norm):**

$$d_{manhattan}(x, y) = \sum_{i=1}^{n} |x_i - y_i|$$

- **Cityblock Distance:**

$$d_{cityblock}(x, y) = \sum_{i=1}^{n} |x_i - y_i|$$

- **Cosine Distance:**

$$d_{cosine}(x, y) = 1 - \frac{\sum_{i=1}^{n} x_i y_i}{\sqrt{\sum_{i=1}^{n} x_i^2}\sqrt{\sum_{i=1}^{n} y_i^2}}$$

- **L1 Distance:**

$$d_{L1}(x, y) = \sum_{i=1}^{n} |x_i - y_i|$$

- **L2 Distance (Euclidean Distance):**

$$d_{L2}(x, y) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2}$$

- **Euclidean Distance:**

$$d_{euclidean}(x, y) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2}$$

- **Minkowski Distance:**

$$d_{minkowski}(x, y) = \left(\sum_{i=1}^{n} |x_i - y_i|^p\right)^{\frac{1}{p}}$$

- **Manhattan Distance:**

$$d_{manhattan}(x, y) = \sum_{i=1}^{n} |x_i - y_i|$$

8.3 Metrics of all models

Model	Accuracy	Precision	Recall	F1 Score	F-beta Score	ROC-AUC	Log Loss
Gradient Boosting	0.893413	0.943646	0.895327	0.918620	0.904448	0.960342	0.227130
LDA	0.838286	0.854327	0.916133	0.884044	0.902994	0.932521	0.349181
Logistic Regression	0.847975	0.869517	0.911008	0.889658	0.902315	0.929282	0.331932
AdaBoost	0.880922	0.924912	0.895965	0.910048	0.901508	0.952212	0.645760
Decision Tree	0.859823	0.902543	0.887648	0.894973	0.890549	0.915223	1.814702
Random Forest	0.875971	0.932049	0.879960	0.905166	0.889852	0.947727	0.277282
KNN	0.808573	0.829724	0.900450	0.863540	0.885286	0.868093	0.485365
SVM	0.808573	0.829724	0.900450	0.863540	0.885286	0.868093	0.485365
SGDC	0.852498	0.910678	0.866843	0.887516	0.874837	0.925462	-
QDA	0.700482	0.848630	0.676726	0.751231	0.704370	0.794206	2.901450
Gaussian Naive Bayes	0.637814	0.882662	0.532335	0.664076	0.578212	0.842532	1.871467

Table 14: Comparison of all the obtained metrics