

Mammographic Mass Data Set

Pedro Sobral - 98491 - (50%) - sobral@ua.pt, Eva Bartolomeu - 98513 - (50%) - evabartolomeu@ua.pt
Departamento de Eletrónica, Telecomunicações e Informática, University of Aveiro, Portugal
Fundamentos de Aprendizagem Automática - Course Instructor: Pétia Georgieva

Abstract—The main goal of this project is to apply suitable machine learning algorithms learned in class to solve a specific data science problem, in this case, models capable of identifying benign or malignant masses in mammography's. In this report, we try to obtain the best result of the accuracy of the implemented models.

Index Terms—Machine Learning, Classification Model, Logistic Regression, SVM, Neural Network, Normalization

I. INTRODUCTION

In the scope of the subject Fundamentals of Automatic Learning, a work was proposed on a problem (of our choice) of machine learning. The goal of this project is to apply suitable machine learning algorithms learned in class or self-learned to solve a specific data science problem (classification, regression, clustering). Represent the results in graphical/table formats and make analysis and conclusions. We decided to choose project proposal 3, Mammographic Mass Data Set. This data set can be used to predict the severity (benign or malignant) of a mammographic mass lesion from BI-RADS attributes and the patient's age. The aim is to discriminate benign from malignant cases assuming that all cases with BI-RADS assessments greater or equal a given value (varying from 1 to 5), are malignant and the other cases are benign. We chose this theme because we already work on project with a similar theme.

II. DATA DESCRIPTION AND PREPROCESSING

A. Data Description

Mammography is the most effective method for breast cancer [2] [4] screening available today. However, the low positive predictive value of breast biopsy resulting from mammogram interpretation leads to approximately 70% unnecessary biopsies with benign outcomes.

To reduce the high number of unnecessary breast biopsies, several computer-aided diagnosis (CAD) systems have been proposed in the last few years. This data set [23] can be used to predict the severity (benign or malignant) of a mammographic mass lesion from BI-RADS attributes and the patient's age.

It contains a BI-RADS assessment, the patient's age, and three BI-RADS attributes (Shape, Margin, Density) together with the ground truth (the severity field) for 516 benign and 445 malignant masses that have been identified on full-field digital mammograms collected at the Institute of Radiology of the University Erlangen-Nuremberg between 2003 and 2006. Each instance has an associated BI-RADS assessment ranging from 1 (definitely benign) to 5 (highly suggestive of

malignancy).

The attributes of the data-set are the following:

- BI-RADS assessment: 1 to 5 (ordinal)
- Age: patient's age in years (integer)
- Shape: mass shape: round=1 oval=2 lobular=3 irregular=4 (nominal)
- Density: mass density high=1 iso=2 low=3 fat-containing=4 (ordinal)
- Severity: benign=0 or malignant=1 (binominal)

B. Data Preprocessing

First of all, we notice that the original data-set has missing values, so on loading the data we substituted all the missing values with NaN, after that, we replaced those values with the mean of the particular attribute. The data-set contains some outliers in the BI-RADS attribute, in order to solve that we removed all the lines with outliers.

We decided to apply a normalization approach. Normalization is a re-scaling technique, that re-scaling an input variable to the range between 0 and 1.

III. DATA VISUALIZATION

In order to have deep knowledge about the data-set it is important to have many data visualization plots such as Box, Histograms, and Scatter. These plots help to understand more how the data are correlated. The data-set is balanced, this is important so that we have more chance that the model will be certain because the data is well distributed and non-bias. The next figure (Fig. 1) shows that the number of Benign and Malign cases is very identical.

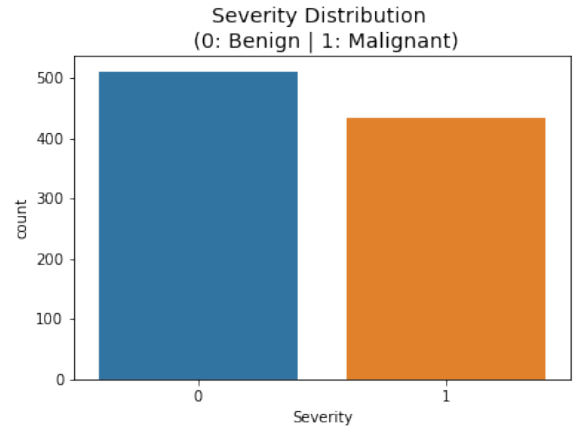


Fig. 1. Class Distribution

The percentages of the Severity are the following (Fig. 2).

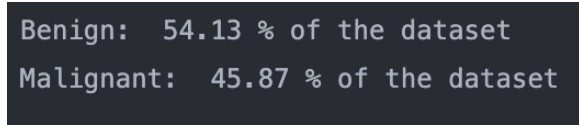


Fig. 2. Percentages of the Severity Distribution

In order to can see some pattern in the data, we plot all the data, like the following figure 3 shows.

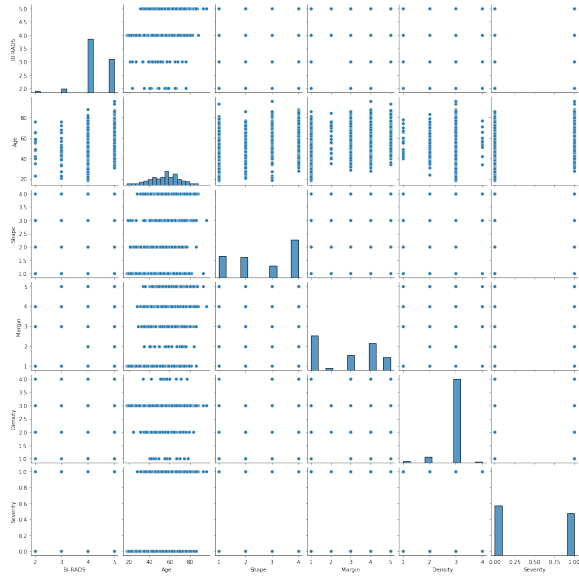


Fig. 3. Data

To see the attribute distributions we make some histograms, ones with benign (Fig. 4) cases and others with malignant (Fig. 5) cases. We make as well a Boxplot graphic to show all the features with the **Severity** (Fig 6).

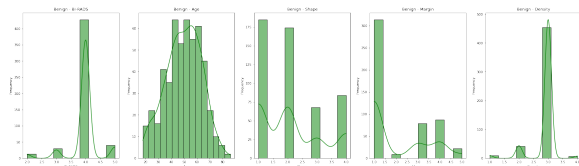


Fig. 4. Histograms - Benign

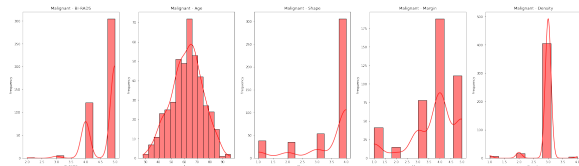


Fig. 5. Histograms - Malignant

Here we can see that the malignant cases, for example, are often in ages from 50 to 80, however, the benign cases appear

earlier and with a bigger frequency, but in a general analysis the histograms are identical. A very obvious consideration to make is that it is more probable to have a malignant case if the attribute "Shape" is 4. About "Margin" level, how higher the value of the attribute more malignant cases are, make the comparison with the "Margin" in the benign histograms, where tendentially the benign cases have a lower value to this attribute. About "Density" there are no assumptions to make since the histograms are very similar, as is possible to observe in the figures.

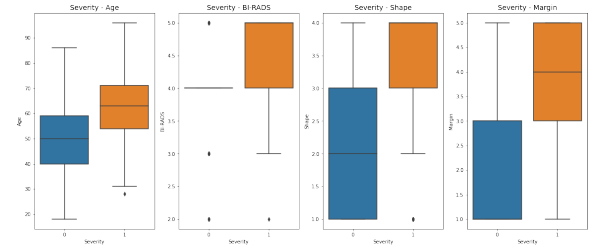


Fig. 6. Feature BoxPlot

So, in order to clarify this situation, and identify the most contributed features to our prediction model it is a good way to improve the accuracy of the data-set, eliminating the least contributed features.

A good first way to make some assumptions is by plotting a Correlation Matrix (Fig. 7), this matrix is good to verify the coefficients of connection between the features. Analyzing the matrix, the less contributed feature is the "Density", this attribute has the lowest values in the correlation matrix, as it is possible to see, it has a white color, so it means it is very low.

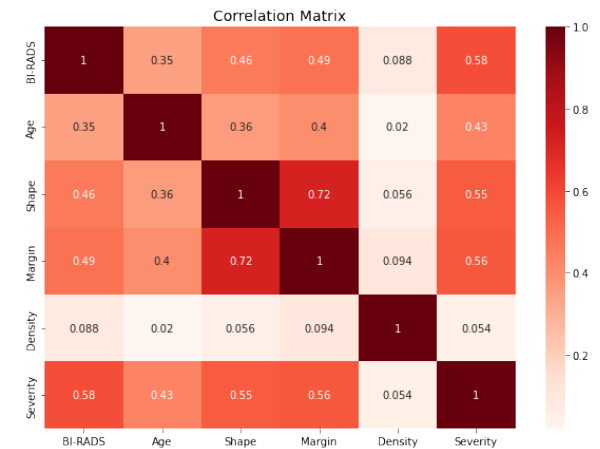


Fig. 7. Correlation Matrix

We have done some research [5] and decide to implement two techniques, the RFE and the PCA.

The RFE technique [22] is an algorithm that returns the feature's importance from the current set of features. So was implemented this algorithm to return the 3 most important

features. The output of this implementation is demonstrated in Fig 8.

```
Support: [ True True True False False]
Ranking: [1 1 1 2 3]
Feature importance: [0.52835375 0.31822601 0.15342024]
1.0
```

Fig. 8. RFE

With these results, the most important features are the first three: **BI-RADS**, **Age**, and **Margin**. Being the least important the **Margin** and **Density**.

Using the PCA technique [21], is the main linear algorithm for dimension reduction, this algorithm identifies and discards features that are less useful to make a valid approximation on a data-set. With this, we obtain the following results ($0.478 + 0.204 + 0.143$) with a sum of (83%) for the 3 most important features. (Fig. 9)

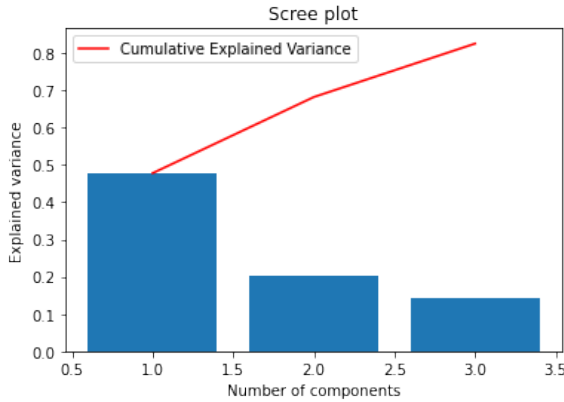


Fig. 9. PCA

So with all this information, we decided to remove the **Margin** and **Density** features of our data-set, remaining with the features **BI-RADS**, **Age** and **Margin**.

IV. MACHINE LEARNING MODELS

Before implementing the machine learning models, we decided that 25% of the data will be used for testing and 75% of the data will be used for training. For this we created two sets, the test, and the training, using the `sklearn.model_selection.train_test_split` library [18].

In each implemented machine learning algorithm, we run the algorithm with the base model, with the hyper-parameter selection model, and with K-Fold Cross Validation on the hyper-parameter selection model. For the execution of the described models, we created a function that trains the model passed as an argument and shows graphs, tables, and data in the output, in order to facilitate the analysis of the model. We also implemented a function that, through the `sklearn.model_selection.GridSearchCV` library [12], performs an exhaustive search on a list of parameters, training a given model with all the parameters, in the end, we return the

parameters that gave the best performance, according to the accuracy strategy.

We developed a function that splits the data set into k mixed sets, with training/test indices, using the `sklearn.model_selection.KFold` feature [13]. Still, within the function, we evaluate the metrics by cross-validation and record the fit/score times through `sklearn.model_selection.cross_validate` [11].

Finally, we create a function that calculates training and testing scores for an estimator with several different values on a specific parameter, these calculations are performed with the `sklearn.model_selection.validation_curve` [20]. We then graphed the average training scores, and the average test scores, on each specific parameter value.

A. Logistic Regression

The Logistic Regression [8] algorithm determines the probability of a binary event (which has only two different results). The Logistic Regression category is supervised learning.

In our problem, we use this algorithm, where the binary results are malignant or benign cancer. We performed the algorithm using the `sklearn.linear_model.LogisticRegression` library [15], and with the previously described functions.

B. Neural Network

Neural Network [7] is a deep learning algorithm, which teaches you to process data like a human brain. It uses interconnected nodes much like neurons, and teaches computers to learn from mistakes and improve. We implemented this algorithm with the `sklearn.neural_network.MLPClassifier` library [16].

C. Support Vector Machine

SVM [10] is a supervised learning algorithm, which analyzes data for classification and regression observation. Considering a set of training examples, where each example belongs to a category (there are two categories, two results), the SVM algorithm builds a model that classifies new data to a category. This classifier is a non-probabilistic linear binary. Examples are mapped to points in space, with a wide gap between categories. We use `sklearn.svm.SVC` [17] to develop this algorithm.

D. k-Nearest Neighbor

k-NN [6] is a non-parametric, supervised learning algorithm that uses the proximity of points to classify or predict, assuming that similar points are close. We implemented the algorithm using the `sklearn.neighbors.KNeighborsClassifier` [14] library.

E. Decision Tree

Decision Tree [9] is supervised learning, presents a tree structure, the internal nodes are the characteristics of a dataset, the branches are the decision conditions, and the leaf nodes are the results. Graphically represents all possible solutions to a problem. In the development of the algorithm, we used `sklearn.tree.DecisionTreeClassifier` [19].

V. RESULTS

A. Logistic Regression

First, we run this algorithm with the base model, with the hyper-parameter selection model, and with K-Fold cross-validation on the hyper-parameter selection model. In Table I we can see the parameters chosen in the hyper-parameter selection model.

TABLE I
BEST PARAMETERS IN LOGISTIC REGRESSION

C	class_weight	max_iter	penalty	solver
10	balanced	100	l2	liblinear

The value selected for parameter C was 10. The higher the value of this parameter, the greater the weight on the training data and the lower the weight on the complexity penalty. The smaller the value of this parameter, the smaller the weight in the training data and the greater the weight in the complexity penalty. This parameter is used to handle extreme data.

The parameter selected in the class_weight was “balanced”, this strategy implicitly replicates the class with the lowest weight, until there is the same number of samples as the class with the highest weight.

The chosen max_iter was 100, that is, the maximum number of iterations for the solvers to converge is 100.

The “l2” parameter was selected for the penalty. This parameter imposes the penalty of the square of the magnitude of the coefficients in the model. Thus reducing the coefficients with less contribution, that is, regularization is performed.

The parameter chosen in the solver was “liblinear”, this strategy is advised in small data sets, such as our set. This strategy behaves like a multiclass classified, through the coordinate descent algorithm, ie, separate binary classifiers are trained for all classes.

1) *Base Model*: The results for the Base Model application are:

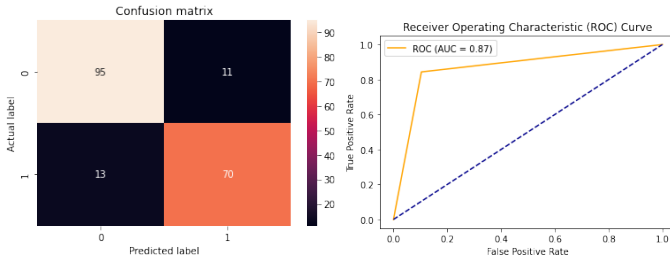


Fig. 10. Confusion Matrix ROC Curve

2) *Hyper-Parameter Selection Model*: The results for the Hyper-Parameter Selection Model are:

3) *K-Fold Cross-Validation Model*: The results for the K-Fold Cross-Validation Model are:

4) *Conclusions*: Observing the confusion matrices, we can conclude that the number of incorrectly classified examples, the sum of FP and FN, (13+11=24 and 10+16=26) is very

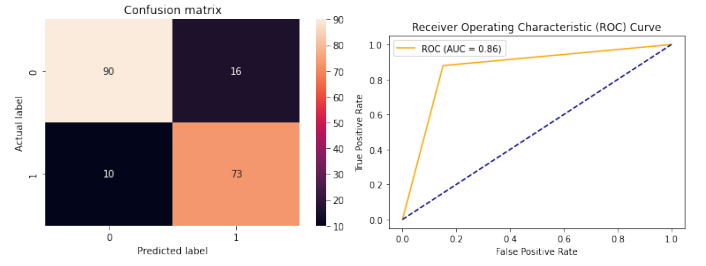


Fig. 11. 2 Confusion Matrix ROC Curve

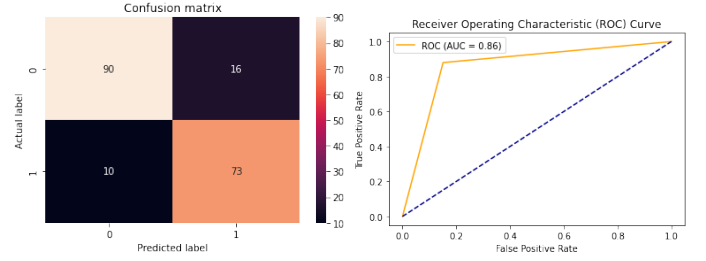


Fig. 12. Confusion Matrix ROC Curve

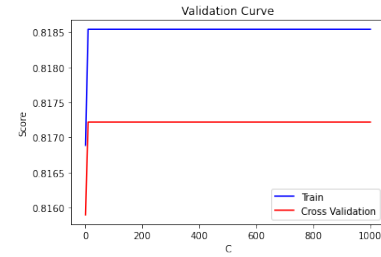


Fig. 13. Validation Curve

low compared to the number of correctly classified examples, sum of TP with TN (95+70 and 90+73). Therefore, we can consider the model accurate.

The higher the AUC score, the better the performance of a classifier for a given task. Therefore, through the ROC Curve graphs, we can say that the classifier performed well because the AUC value is 0.86/0.87. It has good predictive power.

In Fig 13, we can conclude that any value of C would be a good parameter, although a value between 0 and 50 is best, the difference is minimal. We also observed that from a value between 0 and 50 both the accuracy of the training score and the cross-validation score remain at a value, 0.8185 and 0.8170-0.8170 respectively.

TABLE II
CLASSIFICATION - ALL MODEL

	Accuracy	F1 Score	True Positive Rate
Base Model	0.873	0.853	0.843
Hyper-Parameter	0.862	0.848	0.879
K-Fold	0.862	0.848	0.879

By table II we can verify that the Accuracy, F1 score, and

True positive rate values of the 3 methods are very identical, with almost no difference in choosing the best method. The Base Model appears to have better accuracy, but the Hyper-Parameter and k-fold have a better True Positive Rate. For this problem, it is more important to have a better value of True Positive Rate than a better Accuracy, because the fewer positive cases pass as negative the better when cancer is malignant it is extremely important that it does not pass as benign.

B. Neural Network

We run this model in three different ways, the base model, the hyper-parameter selection model, and the K-Fold cross-validation with the hyper-parameter selected. The following Table III shows the parameters chosen by the algorithm.

TABLE III
BEST PARAMETERS IN NEURAL NETWORK

activation	alpha	hidden_layer_sizes	learning_rate
relu	1	(12, 12)	constant
learning_rate_init	max_iter	solver	
0.001	1000	adam	

1) *Base Model*: The results for the Base Model application are:

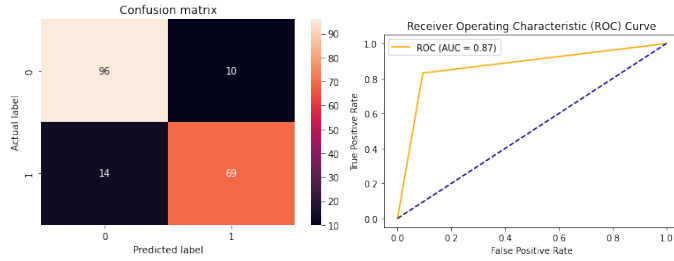


Fig. 14. Confusion Matrix ROC Curve

2) *Hyper-Parameter Selection Model*: The results for the Hyper-Parameter Selection Model are:

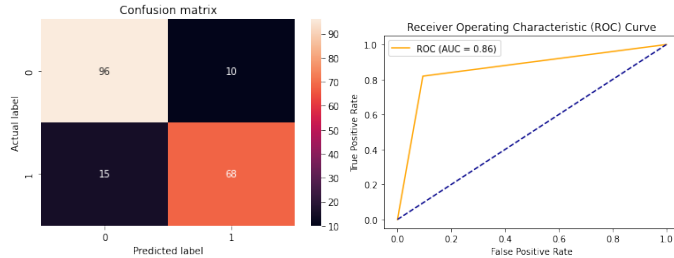


Fig. 15. 2 Confusion Matrix ROC Curve

3) *K-Fold Cross-Validation Model*: The results for the K-Fold Cross-Validation Model are:

4) *Conclusions*: As the graphics are very identical, a general conclusion will be made about the Confusion Matrix and ROC curve of all three implementations. As we can see, Fig.

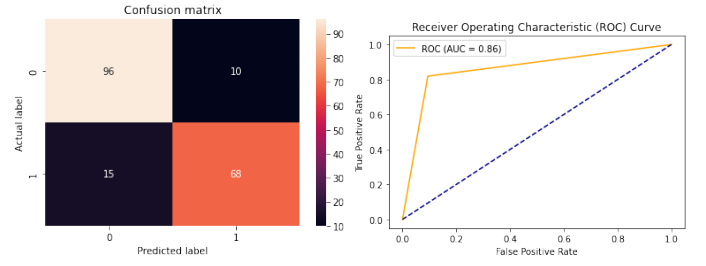


Fig. 16. Confusion Matrix ROC Curve

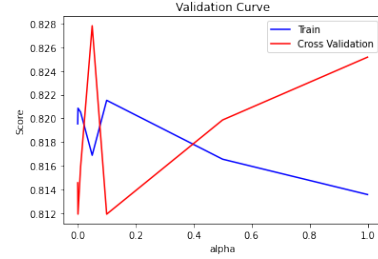


Fig. 17. Validation Curve

14, 15, 16 the Confusion Matrices are almost the same, so we see a bit of variation between the approaches, the ROC Curve is close to the top-left corner in the beginner, which is very good, how more the curve is closer to the top-left corner the better the performance, then the curve has a slope decrease to the end. About AUC (Area under the curve), how higher the value is (between 0 to 1) the better it is. As we can see in this article [1], values between 0.8 and 0.9 are considered excellent.

In Fig 17, we can conclude that the ideal value for the alpha parameter would be 0.4 and approximately 0.01 and 0.09. We can also observe that from alpha value 0.4, the accuracy of the training score decreases, and the cross-validation score increases.

TABLE IV
CLASSIFICATION - ALL MODEL

	Accuracy	F1 Score	True Positive Rate
Base Model	0.873	0.851	0.831
Hyper-Parameter	0.867	0.844	0.819
K-Fold	0.867	0.844	0.819

So we can say IV that with the Neural Network model, the Base Model approach is the best compared with the other two, where the **Accuracy** is 0.873 and the **True Positive Rate** is 0.831.

C. Support Vector Machine

We run this model in three different ways, the base model, the hyper-parameter selection model, and the K-Fold cross-validation with the hyper-parameter selected. The following Table V shows the parameters chosen by the algorithm.

1) *Base Model*: The results for the Base Model application are:

TABLE V
BEST PARAMETERS IN SVC

algorithm	$n_{neighbors}$	weights
auto	15	uniform

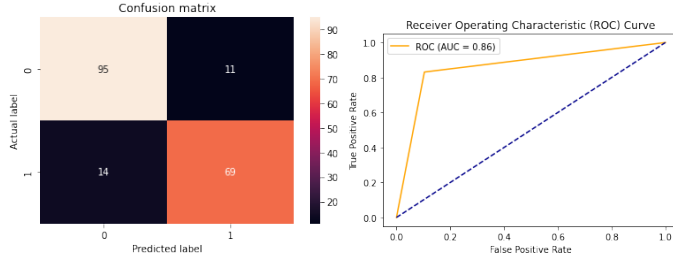


Fig. 18. Confusion Matrix ROC Curve

2) *Hyper-Parameter Selection Model*: The results for the Hyper-Parameter Selection Model are:

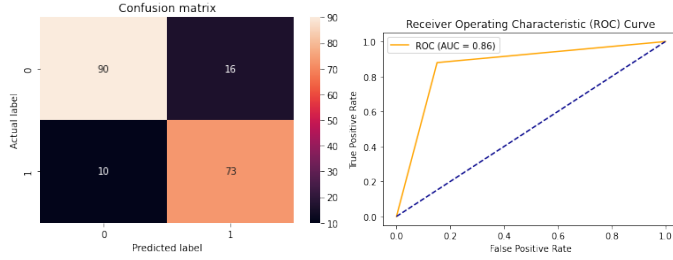


Fig. 19. 2 Confusion Matrix ROC Curve

3) *K-Fold Cross-Validation Model*: The results for the K-Fold Cross-Validation Model are:

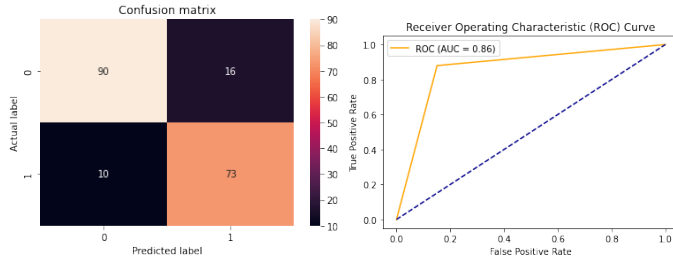


Fig. 20. Confusion Matrix ROC Curve

4) *Conclusions*: As the graphics are very identical, a general conclusion will be made about the Confusion Matrix and ROC curve of all three implementations. As we can see, Fig. 18, 19, 20 the Confusion Matrices are almost the same, so we see a bit of variation between the approaches, the ROC Curve is close to the top-left corner in the beginner, which is very good, how more the curve is closer to the top-left corner the better the performance, then the curve has a slope decrease to the end. About AUC (area under curve), how higher the value is (between 0 to 1) the better it is. As we can see in this article [1], values between 0.8 and 0.9 are considered excellent.

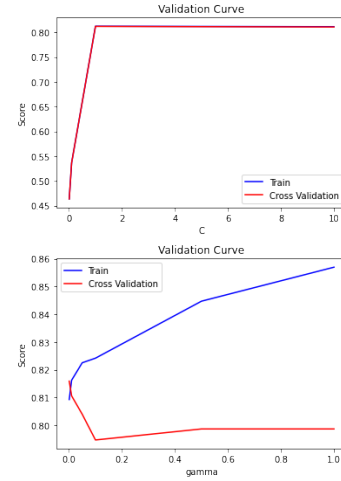


Fig. 21. Validation Curve

In Fig 21, we can conclude that any value of C would be a good parameter, as the Train values all coincide with the Cross Validation values. We also observed that from a value of 1 of C, more and less, both the accuracy of the training score and the Cross Validation score remain at a value of 0.80.

Still in Fig 21, we see that the ideal value for the gamma parameter is approximately 0. We also see that from a gamma value of 0, the accuracy of the training score increases, and the Cross Validation score goes down.

TABLE VI
CLASSIFICATION - ALL MODEL

	Accuracy	F1 Score	True Positive Rate
Base Model	0.867	0.846	0.831
Hyper-Parameter	0.862	0.848	0.879
K-Fold	0.862	0.848	0.879

So we can say (Table VI) that with the SVC model, the best approach is the Hyper-Parameter and the K-Fold, where the **Accuracy** is 0.862 and the **True Positive Rate** is 0.879.

D. k-Nearest Neighbor

We run this model in three different ways, the base model, the hyper-parameter selection model, and the K-Fold Cross Validation with the hyper-parameter selected. The following Table VII shows the parameters chosen by the algorithm.

C	$class_weight$	gamma	kernel
50	balances	0.001	rbf

TABLE VII
BEST PARAMETERS IN K-NEAREST NEIGHBOR

1) *Base Model*: The results for the Base Model application are:

2) *Hyper-Parameter Selection Model*: The results for the Hyper-Parameter Selection Model are:

3) *K-Fold Cross-Validation Model*: The results for the K-Fold Cross-Validation Model are:

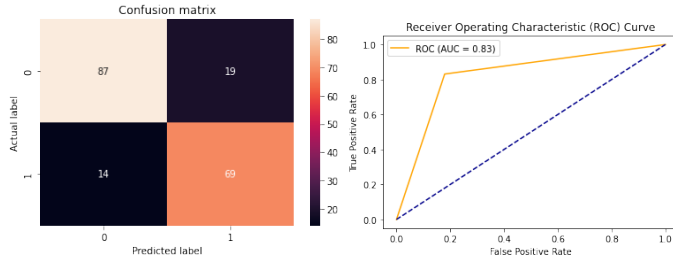


Fig. 22. Confusion Matrix ROC Curve

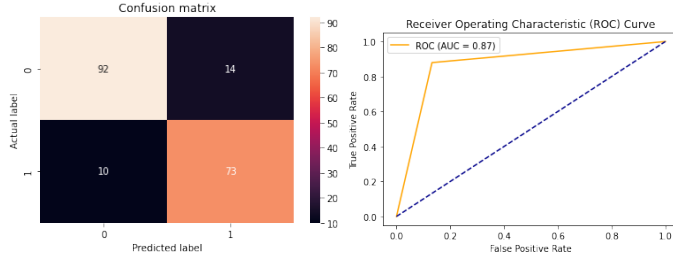


Fig. 23. 2 Confusion Matrix ROC Curve

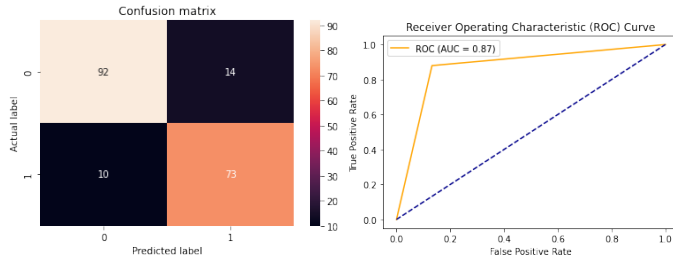


Fig. 24. Confusion Matrix ROC Curve

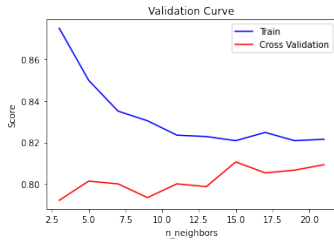


Fig. 25. Validation Curve

4) *Conclusions:* As the graphics are very identical, a general conclusion will be made about the Confusion Matrix and ROC curve of all three implementations. As we can see, Fig. 22, 23, 24 the Confusion Matrices are almost the same, so we see a bit of variation between the approaches, the ROC Curve is close to the top-left corner in the beginner, which is very good, how more the curve is closer to the top-left corner the better the performance, then the curve has a slope decrease to the end. About AUC (area under curve), how higher the value is (between 0 to 1) the better it is. As we can see in this article [1], values between 0.8 and 0.9 are considered excellent.

In Fig 25, we can conclude that the ideal values for the parameter `n_neighbors` are approximately 15 and 18.75. We also observed that from the value 15 of `n_neighbors`, the accuracy of the training score and the Cross Validation score, remain more or less at the same values, with small descents and ascents, in the approximate values of 0.82 and 0.81 respectively.

TABLE VIII
CLASSIFICATION - ALL MODEL

	Accuracy	F1 Score	True Positive Rate
Base Model	0.825	0.807	0.831
Hyper-Parameter	0.873	0.858	0.879
K-Fold	0.873	0.858	0.879

So we can say (Table VIII) that with the K-NN model, the best approach is the Hyper-Parameter and the K-Fold, where the **Accuracy** is 0.873 and the **True Positive Rate** is 0.879.

E. Decision Tree

We run this model in three different ways, the base model, the hyper-parameter selection model, and the K-Fold cross-validation with the hyper-parameter selected. The following Table IX shows the parameters chosen by the algorithm.

criterion	max_depth	min_samples_leaf	min_samples_split	splitter
entropy	5	8	9	random

TABLE IX
BEST PARAMETERS IN DECISION TREE

1) *Base Model:* The results for the Base Model application are:

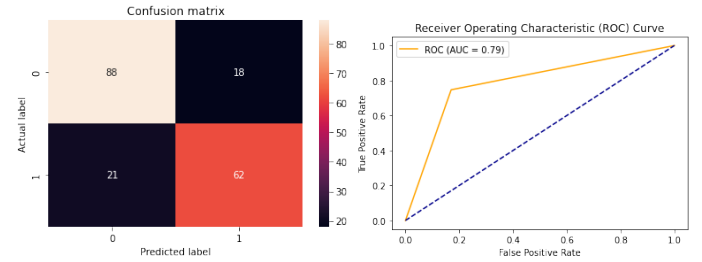


Fig. 26. Confusion Matrix ROC Curve

2) *Hyper-Parameter Selection Model:* The results for the Hyper-Parameter Selection Model are:

3) *K-Fold Cross-Validation Model:* The results for the K-Fold Cross-Validation Model are:

4) *Conclusions:* As the graphics are very identical, a general conclusion will be made about the Confusion Matrix and ROC curve of all three implementations. As we can see, Fig. 26, 27, 28 the Confusion Matrices are almost the same, so we see a bit of variation between the approaches, the ROC Curve is close to the top-left corner in the beginner, which is very good, how more the curve is closer to the top-left corner the better the performance, then the curve has a slope decrease to the end. About AUC (area under curve), how higher the value

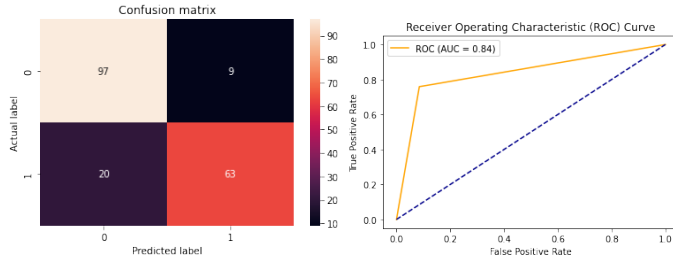


Fig. 27. 2 Confusion Matrix ROC Curve

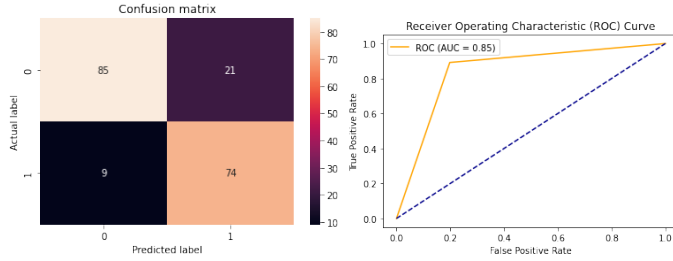


Fig. 28. Confusion Matrix ROC Curve

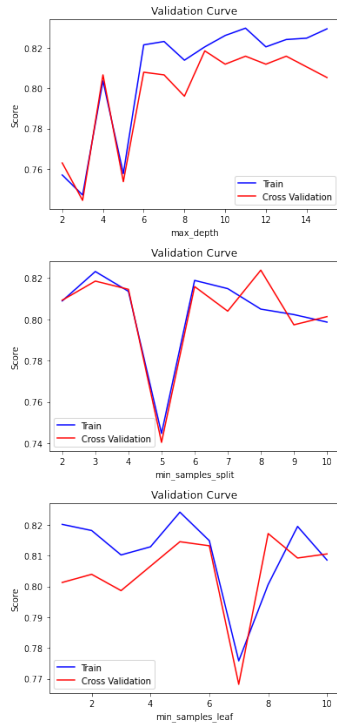


Fig. 29. Validation Curve

is (between 0 to 1) the better it is. As we can see in this article [1], values between 0.8 and 0.9 are considered excellent.

In Fig 29, we see that both the max_depth parameter and the min_samples_split parameter and min_samples_leaf can have any value, as they are practically all good, although there are better values (they are the intersection values of the train and the Cross Validation), the difference is minimum.

TABLE X
CLASSIFICATION - ALL MODEL

	Accuracy	F1 Score	True Positive Rate
Base Model	0.793	0.760	0.746
Hyper-Parameter	0.846	0.812	0.759
K-Fold	0.841	0.831	0.891

So we can say (Table X) that with the Decision Tree model, the best approach is with K-Fold, where the **Accuracy** is 0.841 and the **True Positive Rate** is 0.891.

F. Comparison between the models

The comparison here XI will be done with the best result from all models done in the previous section.

TABLE XI
CLASSIFICATION - ALL MODEL

	Accuracy	True Positive Rate
Logistic Regression (HP/K-Fold)	0.862	0.879
Neural Network (Base Model)	0.873	0.831
SVC (HP/K-Fold)	0.841	0.891
K-NN (HP/K-Fold)	0.841	0.891
Decision Tree (K-Fold)	0.831	0.891

Three of the five have the best **True Positive Rate** (SVC (HP/K-Fold), K-NN (HP/K-Fold), Decision Tree (K-Fold)), which is 0.891, however, the SVC and K-NN approach have a little more of **Accuracy** (more 0.01), so we decided that the best model to our problem will be one of the (SVC (HP/K-Fold) or K-NN (HP/K-Fold) approach.

VI. NOVELTY AND CONTRIBUTIONS

We found a published article A Machine Learning Predictive Model to Classify Severity of Breast Cancer Based on Mammographic Mass Dataset [3] that has the following results:

Machine Learning Model	Accuracy of models with reduced dataset	Standard deviation of models with reduced dataset
Logistic Regression	82.50%	0.042
Linear Discriminant Analysis	84.50%	0.037
K-Neighbors Classifier	82.06%	0.036
Decision Tree Classifier	82.82%	0.052
Gaussian Naïve Bayes	82.96%	0.033
Support Vector Classifier	83.74%	0.039

Fig. 30. Published Reference Results

Comparing our solution with theirs at accuracy levels (they don't have information about **True Positive Rate**), we have some models with more accuracy (Logistic Regression (HP/K-Fold) - 0.862 and Neural Network (Base Model) - 0.873) that the best of their (Linear Discriminant Analysis - 0.845).

We can say that we propose a better solution, with better performance of the Machine Learning models.

VII. CONCLUSION

Doing this project was very relevant since we not only successfully applied the machine learning algorithms learned in class, but we also self-learned new technics of machine learning using the internet. As we were developing our work, there were some periods where we found some difficulties, especially when we were dealing with data preprocessing since we had trouble deciding which features should be discarded. If we were to develop this work even further we would like to improve our data preprocessing or have more features in order to have better results. In the end, we are very content with the work we presented, since it made us increase our knowledge.

REFERENCES

- [1] Jayawant N. Mandrekar. "Receiver Operating Characteristic Curve in Diagnostic Test Assessment". In: *Journal of Thoracic Oncology* 5.9 (2010), pp. 1315–1316. ISSN: 1556-0864. DOI: <https://doi.org/10.1097/JTO.0b013e3181ec173d>. URL: <https://www.sciencedirect.com/science/article/pii/S1556086415306043>.
- [2] SR Islam and SM Aziz. "Mammography is the most effective method of breast cancer screening." In: *Myensingh medical journal: MMJ* 21.2 (2012), pp. 366–371.
- [3] Mahmud Parvez et al. "A Machine Learning Predictive Model to Classify Severity of Breast Cancer Based on Mammographic Mass Dataset". In: *2021 IEEE Fourth International Conference on Artificial Intelligence and Knowledge Engineering (AIKE)*. 2021, pp. 81–87. DOI: [10.1109/AIKE52691.2021.00018](https://doi.org/10.1109/AIKE52691.2021.00018).
- [4] cdc. *What is Breast Cancer Screening?* URL: https://www.cdc.gov/cancer/breast/basic_info/screening.htm.
- [5] *Feature Selection Techniques in Machine Learning*. URL: <https://www.analyticsvidhya.com/blog/2020/10/feature-selection-techniques-in-machine-learning/>.
- [6] IBM. *knn-definition*. URL: <https://www.ibm.com/topics/knn>.
- [7] IBM. *neural-networks-definition*. URL: <https://www.ibm.com/cloud/learn/neural-networks>.
- [8] javatpoint. *logistic-regression-definition*. URL: <https://www.javatpoint.com/logistic-regression-in-machine-learning>.
- [9] javatpoint. *tree-definition*. URL: <https://www.javatpoint.com/machine-learning-decision-tree-classification-algorithm>.
- [10] MonkeyLearn. *svm-definition*. URL: <https://monkeylearn.com/blog/introduction-to-support-vector-machines-svm/>.
- [11] scikit-learn. *cross-validate-documentation*. URL: https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.cross_validate.html.
- [12] scikit-learn. *GridSearchCV-documentation*. URL: https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html.
- [13] scikit-learn. *KFold-documentation*. URL: https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.KFold.html.
- [14] scikit-learn. *knn-documentation*. URL: <https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KNeighborsClassifier.html>.
- [15] scikit-learn. *logistic-regression-documentation*. URL: https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html.
- [16] scikit-learn. *MLPClassifier-documentation*. URL: https://scikit-learn.org/stable/modules/generated/sklearn.neural_network.MLPClassifier.html.
- [17] scikit-learn. *svm-documentation*. URL: <https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html>.
- [18] scikit-learn. *train-test-split-documentation*. URL: https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.train_test_split.html.
- [19] scikit-learn. *tree-documentation*. URL: <https://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeClassifier.html>.
- [20] scikit-learn. *validation-curve-documentation*. URL: https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.validation_curve.html.
- [21] sklearn. *PCA: Principal Component Analysis using Python (Scikit-learn)*. URL: <https://www.jcchouinard.com/pca-with-python/>.
- [22] sklearn. *sklearn - RFE*. URL: https://scikit-learn.org/stable/modules/generated/sklearn.feature_selection.RFE.html.
- [23] Machine Learning Repository - UCI. *Mammographic Mass Data Set*. URL: <https://archive.ics.uci.edu/ml/datasets/mammographic+mass>.