VRIJE
UNIVERSITEIT
BRUSSEL

BREAST CANCER CLASSIFICATION

Techniques of Artificial Intelligence

Amine Naqi | 0562497

May 21, 2023

Faculty of Engineering | ETRO

Contents

1 Introduction

1.1 Breast Cancer Statistics

In 2020, [e-c] there were 2.3 million women diagnosed with breast cancer and 685 000 deaths globally. As of the end of 2020, there were 7.8 million women alive who were diagnosed with breast cancer in the past 5 years, making it the world's most prevalent cancer. There are more lost disability-adjusted life years (DALYs) by women to breast cancer globally than any other type of cancer. Breast cancer occurs in every country of the world in women at any age after puberty but with increasing rates in later life.

Breast calcification [Org](are calcium deposits that form in the breast tissue. They are not related to the amount of calcium taken in the diet or obtained through supplements) are quite common and most are not associated with cancer. To be sure, the radiologist looks at their size, shape and arrangement on a mammogram, where they often appear as small white spots. Some of their characteristics, such as an irregular shape or certain groupings, may be suspicious.

There are two types of calcification: "macro-calcifications" and "micro-calcifications". Macro-calcifications are large deposits of calcium in the breast. They are more common in women over 50 years old. They are often associated with benign changes in the breast, such as aging of the breast arteries, old lesions, inflammation or masses such as fibroadenoma. For this reason, when these macro-calcifications are found, the radiologist does not routinely recommend a biopsy.

1.2 Project Objective

The objective of this project is to be able to detect breast cancer from data extracted from images. These images are the result of ct-scans, and the problem is a binary classification problem. The data-set is representative of wavelet analysis of 3562 images which represent 96 cases or 96 patients. The data-set also contains 150 attributes or features representing radio-mic data of the micro calcification. The goal of this work is to classify individual micros assuming all micros per subject have the same label, and to classify whether a subject has cancer based on the classification of the multiple micros per subject.

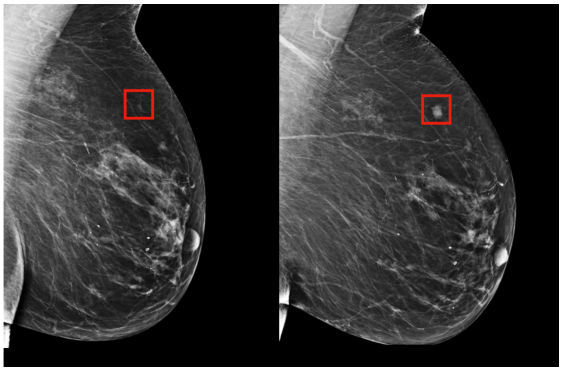

Figure 1: Mammogram of a Patient

2 Algorithms Used

2.1 Support Vector Machine

Support Vector Machine [IBMa] works by mapping data to a high-dimensional feature space so that data points can be categorized, even when the data are not otherwise linearly separable. A separator between the categories is found, then the data are transformed in such a way that the separator could be drawn as a hyperplane. Following this, characteristics of new data can be used to predict the group to which a new record should belong.

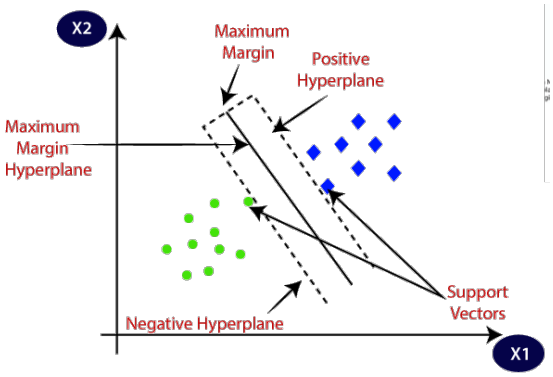

Figure 2: Support Vector Machine

2.2 Logistic Regression

Logistic regression [AWS] is a data analysis technique that uses mathematics to determine the relationships between two data factors. This relationship is then used to predict the value of one of these factors based on the other. The prediction usually has a limited number of outcomes, such as yes or no.

For example, let's say you want to guess whether or not your website visitor will click the checkout button in their shopping cart.

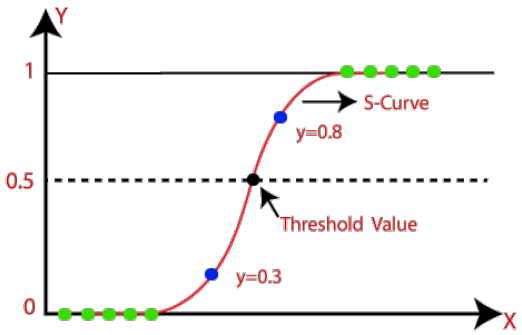

Figure 3: Logistic Regression

2.3 Bagging

Bagging [IBMb], also known as bootstrap aggregation, is the ensemble learning method that is commonly used to reduce variance within a noisy data-set. In bagging, a random sample of data in a training set is selected with replacement—meaning that the individual data points can be chosen more than once. After several data samples are generated, these weak models are then trained independently, and depending on the type of task—regression or classification, for example—the average or majority of those predictions yield a more accurate estimate.

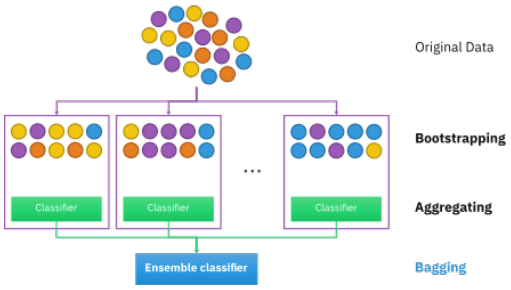

Figure 4: Ensemble Learning - Bagging

2.4 Random Forest

Random forest [IBMc] is a commonly-used machine learning algorithm trademarked by Leo Breiman and Adele Cutler, which combines the output of multiple decision trees to reach a single result. Its ease of use and flexibility have fueled its adoption, as it handles both classification and regression problems.

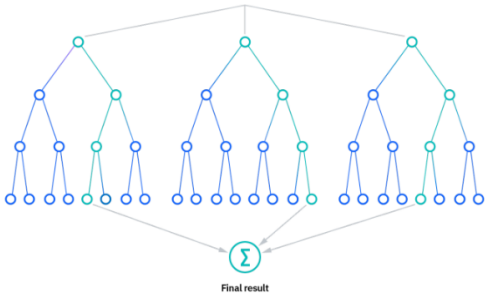

Figure 5: Random Forest

2.5 Other Algorithms Used

We have also used AdaBoost, Naives Bayes, and finally KNearest Neighbor for the classifying tasks. We have chosen all of these algorithms due to their success, and popularity among researchers, as well as software solutions. For the sake of staying under the max limitation of how many pages we can use, we have only targeted 3 algorithms which are "SVM", "Logistic Regression", "Bagging" and finally "Random Forest".

3 Preparing Data For Task 1

We have used the following steps in order to generate models to predict if the patient will have breast cancer.

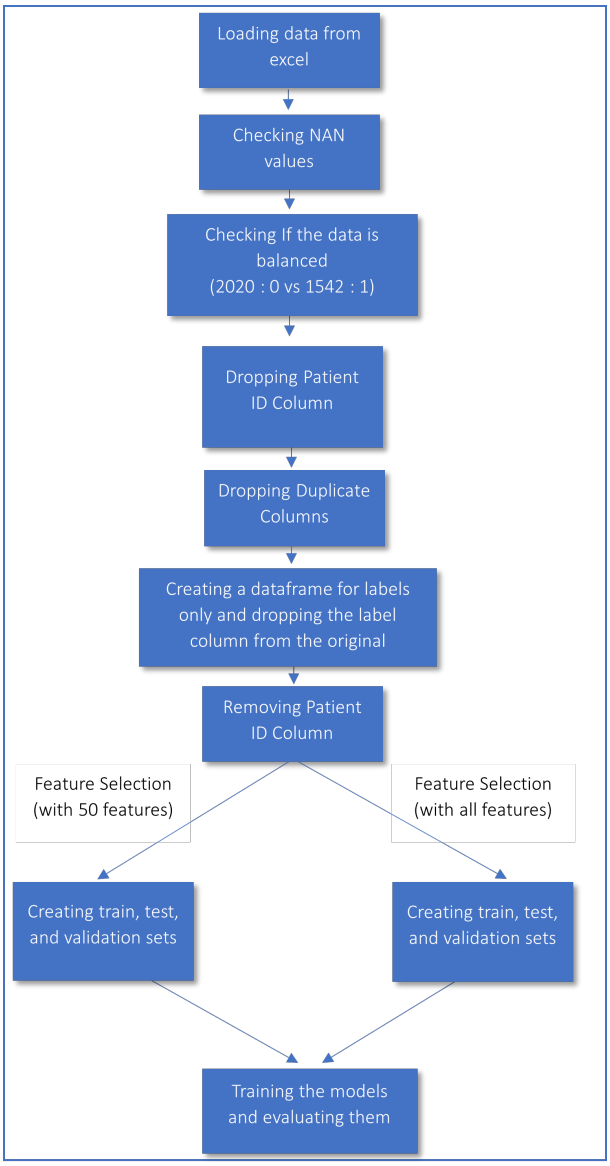

Figure 6: Training Diagram

The data used in this task comes from the same source as the second task, but the way data was processed and splitted is different which gave us different results using the same algorithms. The data-set in this case is used to train models to classify individual micros assuming all micros per subject have the same label.

4 Results

4.1 Using Support Vector Machine

4.1.1 For 50 features

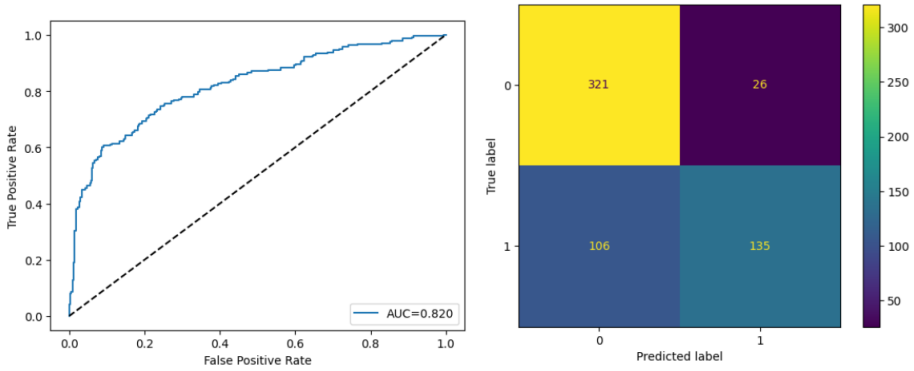

Figure 7: ROC and Confusions Matrix for SVM with 50 Features

4.1.2 For all features

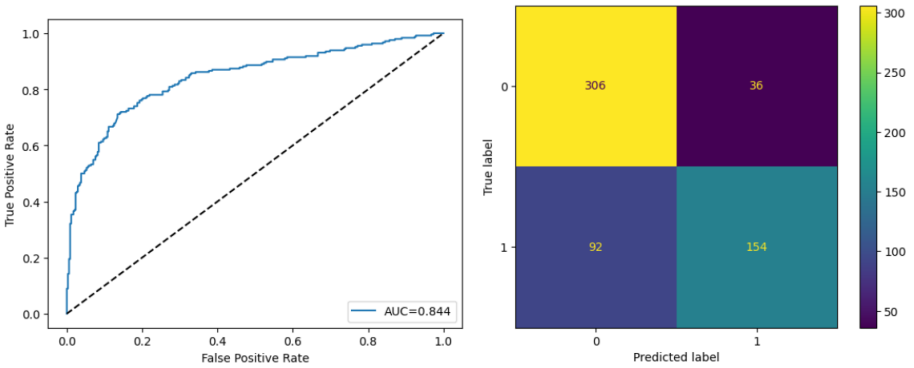

Figure 8: ROC and Confusions Matrix for SVM with all Features

Summary By using GridSearch and CrossValidation, We were able to tune the algorithm and extract the best possible model out of it. The hyper-parameters used to obtain these results are : "Kernel : RBF, Gamma : Auto". This model was able to reach 75.55% in accuracy, 83.85% in Precision, but only 56.01% in recall (for 50 features). On the other side (using full features) we were able to obtain to obtain 78.23% in accuracy, 81.05% in precision and 62.60% in recall. This proves that using the full range of features helps the algorithm to learn more in order to give better predictions. We can see this through the AUC results where the AUC is higher by 2.4% when we use all of the features, and the confusion matrix has a better repartition, as well as the model being able to predict more true positives.

4.2 Using Logistic Regression

4.2.1 For 50 features

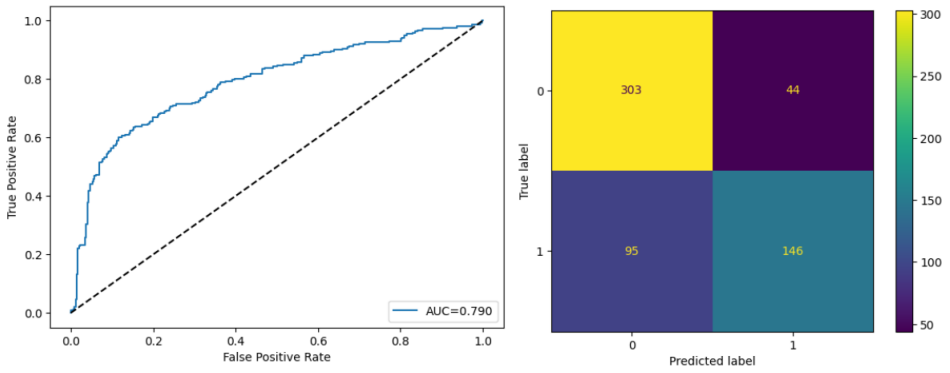

Figure 9: ROC and Confusions Matrix for LR with 50 Features

4.2.2 For all features

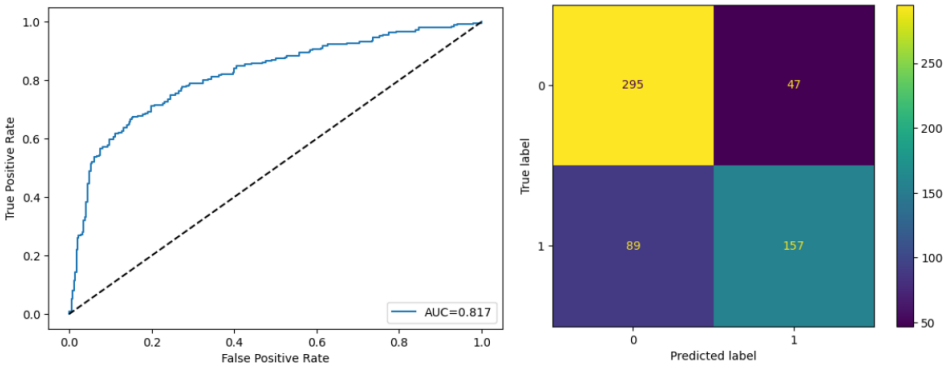

Figure 10: ROC and Confusions Matrix for LR with all Features

Summary By using GridSearch and CrossValidation, We were able to tune the algorithm and extract the best possible model out of it. The hyper-parameter used to obtain these results is : "Max iteration : 500" This model was able to reach 76.36% in accuracy, 76.84% in Precision, and finally 60.58% in recall (for 50 features). On the other side (using full features) we were able to obtain to obtain 76.87% in accuracy, 76.96% in precision and 63.82% in recall. This proves that using the full range of features helps the algorithm to learn more in order to give better predictions. We can see this through the AUC results where the AUC is higher by 2.7% when we use all of the features, and the confusion matrix has a better repartition as well, as the model is able to predict more true positives.

4.3 Using Bagging

4.3.1 For 50 features

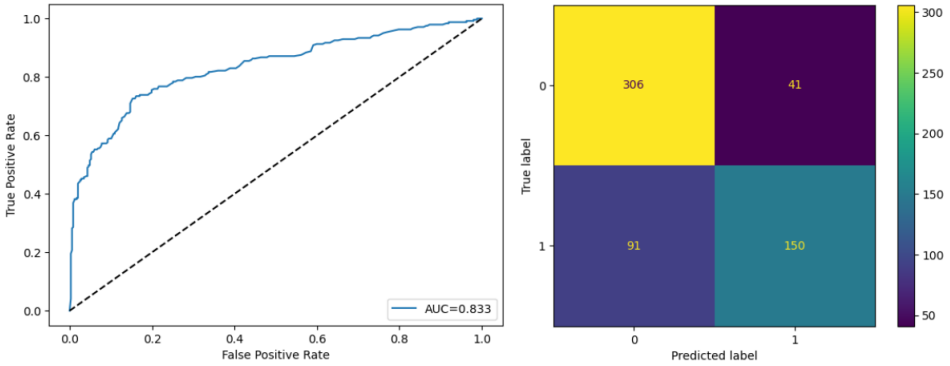

Figure 11: ROC and Confusions Matrix for Bagging with 50 Features

4.3.2 For all features

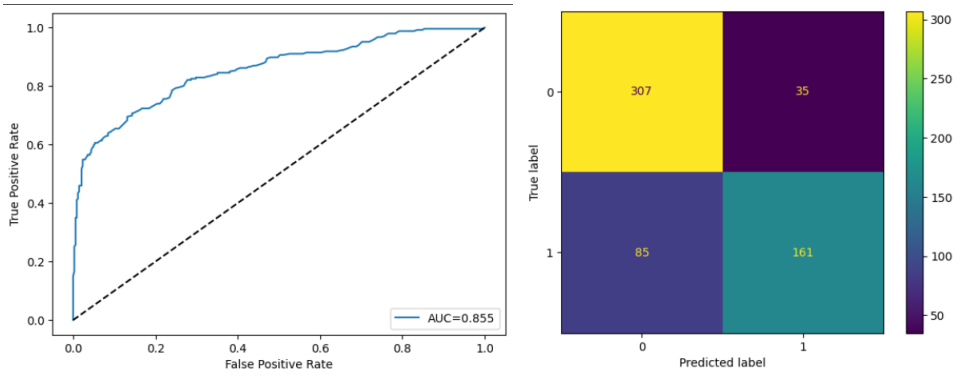

Figure 12: ROC and Confusions Matrix for Bagging with all Features

Summary By using GridSearch and CrossValidation, We were able to tune the algorithm and extract the best possible model out of it. The hyper-parameters used to obtain these results are : "Number of Estimators': 300" This model was able to reach 77.55% in accuracy, 78.53% in Precision, and finally 62.24% in recall (for 50 features). On the other side (using full features) we were able to obtain to obtain 79.59% in accuracy, 82.14% in precision and 65.44% in recall. This proves that using the full range of features helps the algorithm to learn more in order to give better predictions. We can see this through the AUC results where the AUC is higher by 2.2% when we use all of the features, and the confusion matrix has a better repartition, as well as the model being able to predict more true positives.

4.4 Using Random Forest

4.4.1 For 50 features

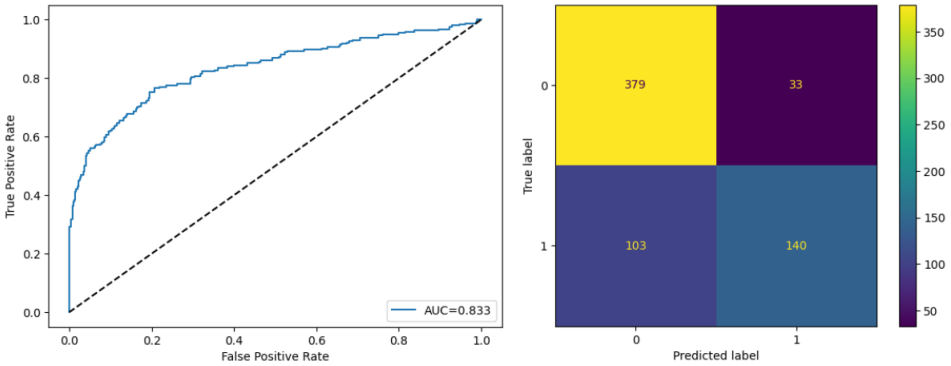

Figure 13: ROC and Confusions Matrix for Random Forest with 50 Features

4.4.2 For all features

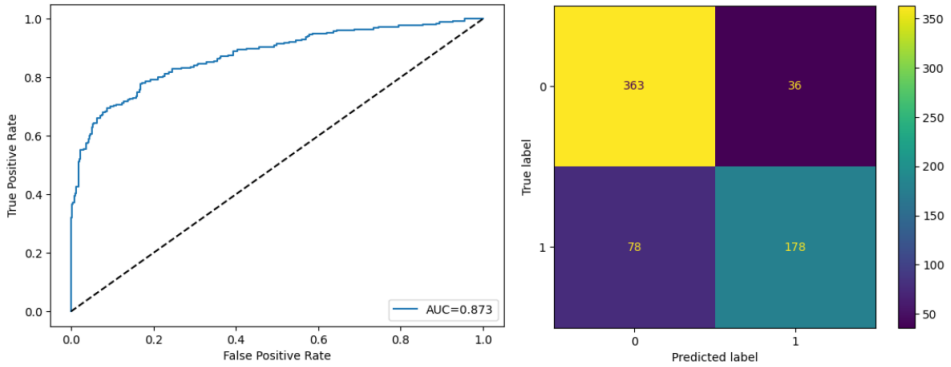

Figure 14: ROC and Confusions Matrix for Random Forest with all Features

Summary By using GridSearch and CrossValidation, We were able to tune the algorithm and extract the best possible model out of it. The hyper-parameters used to obtain these results are : "Number of Estimators : 100", "Max Depth = 10", "Min Samples Split = 10" This model was able to reach 78.57% in accuracy, 80.42% in Precision, and finally 63.07% in recall (for 50 features). On the other side (using full features) we were able to obtain to obtain 78.91% in accuracy, 81.12% in precision and 64.63% in recall. This proves that using the full range of features helps the algorithm to learn more in order to give better predictions. We can see this through the AUC results where the AUC is higher by 0.8% when we use all of the features, and the confusion matrix has a better repartition, as well as the model being able to predict more true positives.

4.5 Benchmark and Results

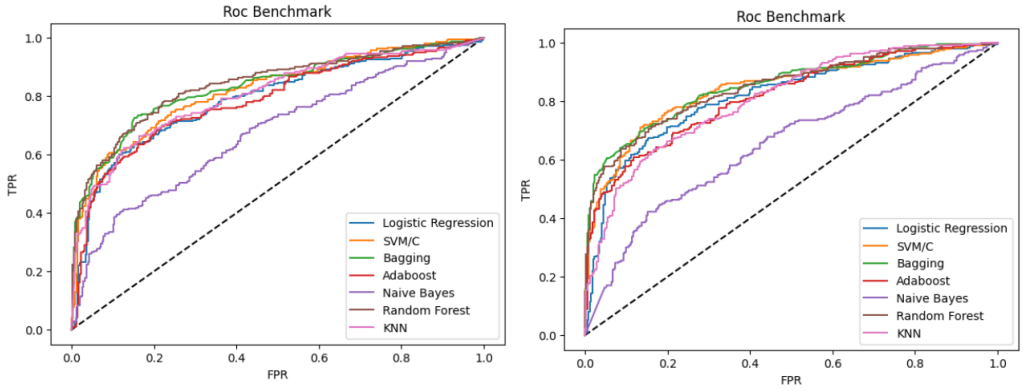

Figure 15: ROC of All Algorithms Used

Dataset	Algorithm	Accuracy	Precision	Recall
1	SVM	0.77	0.83	0.56
2	SVM	0.78	0.81	0.62
1	LR	0.76	0.76	0.60
2	LR	0.76	0.76	0.63
1	Bagging	0.77	0.78	0.62
2	Bagging	0.79	0.82	0.65
1	Adaboost	0.75	0.77	0.57
2	Adaboost	0.73	0.69	0.66
1	Naives Bayes	0.63	0.54	0.57
2	Naives Bayes	0.60	0.52	0.63
1	KNN	0.76	0.77	0.60
2	KNN	0.74	0.77	0.54
1	Random Forest	0.78	0.80	0.63
2	Random Forest	0.78	0.81	0.64

Table 1: Benchmark Results for Task 1.

4.5.1 Conclusion

From all of the previous summaries, we have noticed that having full features is always better (or just slightly better in some cases, besides Naïve Bayes classifier) when it comes to classifying the micro-calcifications. Naïve Bayes is an algorithm that is used for classification purposes. In fact this algorithm is well known for his ability to classify a multiple class data-set, as well as when we are dealing with categorical input. Our data-set contains binary targets, and numerical inputs which makes this algorithm not entirely suitable for our purpose.

For the model trained with only 50 features, we can observe from the results "that Random Forest" is the clear winner with an AUC of 0.841 compared to Bagging which is 0.833, and an accuracy, precision, and recall of 78.57%, 80.42%, and finally 63.07%, in comparison with bagging. As for the model trained with all features, we can say that by comparing the results that "Bagging" is the best performer with having an AUC of 0.855 compared to "Random Forest" that was able to get 0.849 on the AUC score.

5 Preparing Data For Task 2

We have used the following steps in order to generate models to predict if the patient will have breast cancer.

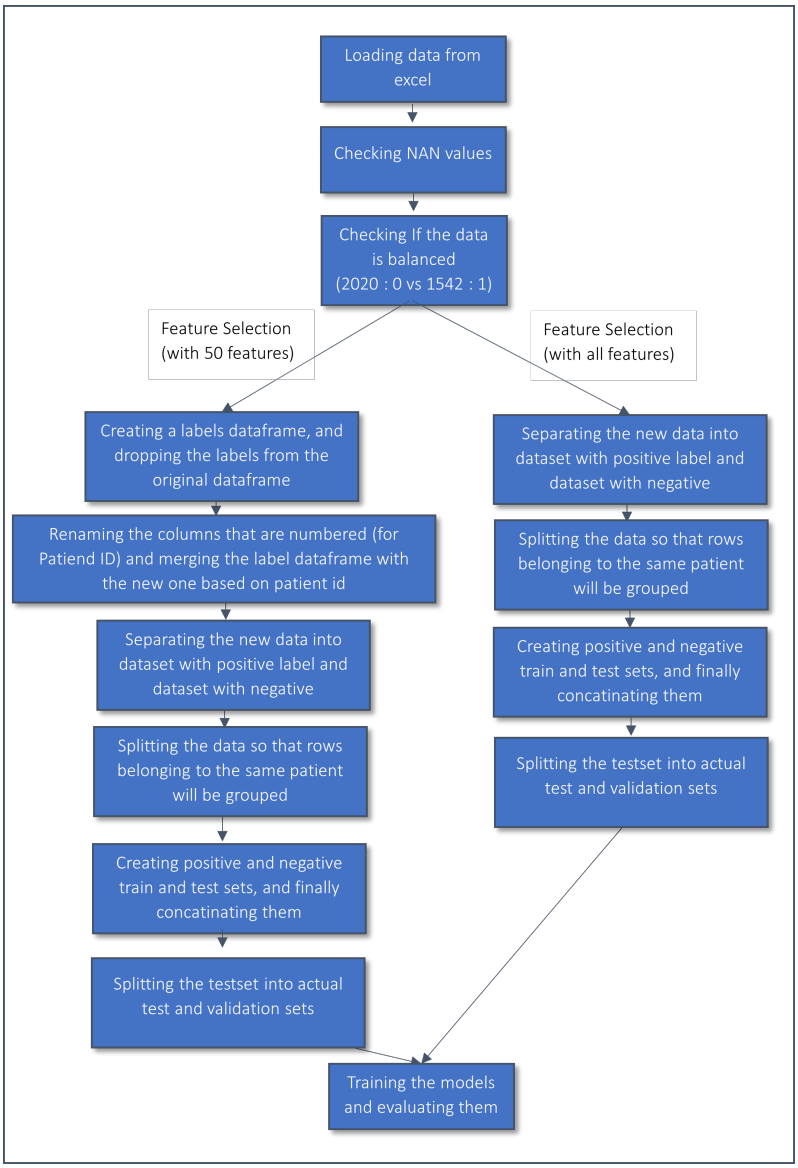

Figure 16: Training Diagram

The data-set used to train the model has been processed so that the models classify whether a subject has cancer based on the classification of the multiple micros per subject.

6 Results

6.1 Using Support Vector Machine

6.1.1 For 50 features

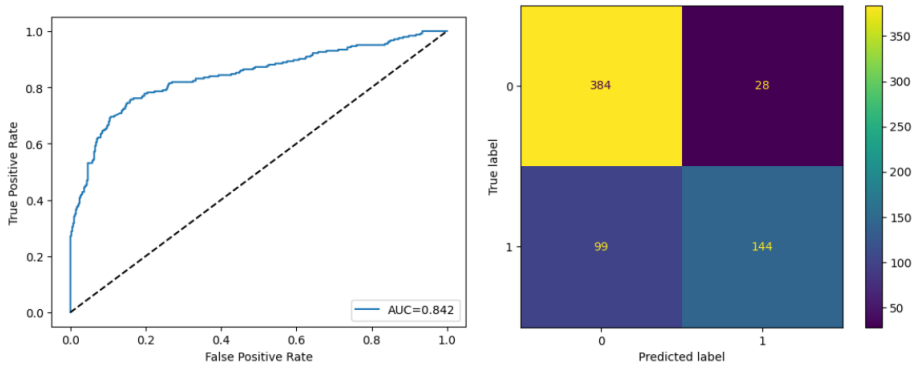

Figure 17: ROC and Confusions Matrix for Support Vector Machine with 50 Features

6.1.2 For all features

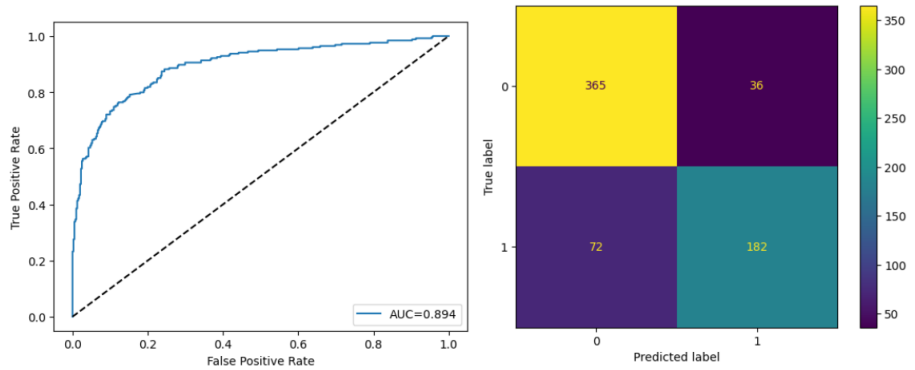

Figure 18: ROC and Confusions Matrix for Support Vector Machine with all Features

Summary By using GridSearch and CrossValidation, We were able to tune the algorithm and extract the best possible model out of it. The hyper-parameters used to obtain these results are : "Kernel : RBF, Gamma : Auto". This model was able to reach 80.61% in accuracy, 83.72% in Precision, but only 59.25% in recall (for 50 features). On the other side (using full features) we were able to obtain to obtain 83.51% in accuracy, 83.48% in precision and 71.65% in recall. This proves that using the full range of features helps the algorithm to learn more in order to give better predictions. We can see this through the AUC results where the AUC is higher by 5.2% when we use all of the features, and the confusion matrix has a better repartition, as well as the model being able to predict more true positives.

6.2 Using Logistic Regression

6.2.1 For 50 features

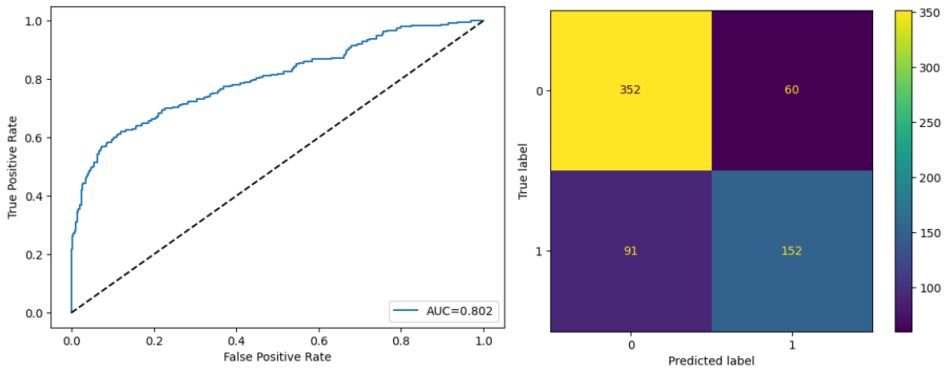

Figure 19: ROC and Confusions Matrix for Logistic Regression with 50 Features

6.2.2 For all features

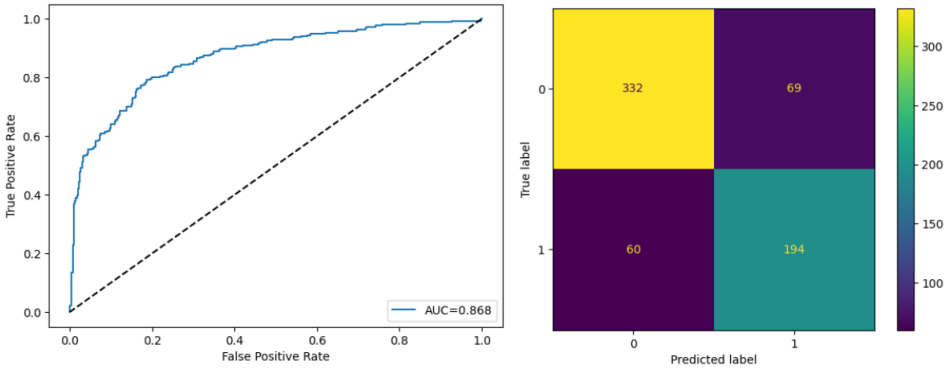

Figure 20: ROC and Confusions Matrix for Logistic Regression with all Features

Summary By using GridSearch and CrossValidation, We were able to tune the algorithm and extract the best possible model out of it. The hyper-parameter used to obtain these results is : "Max iteration : 500" This model was able to reach 76.94% in accuracy, 71.69% in Precision, and finally 62.55% in recall (for 50 features). On the other side (using full features) we were able to obtain to obtain 80.30% in accuracy, 73.76% in precision and 76.37% in recall. This proves that using the full range of features helps the algorithm to learn more in order to give better predictions. We can see this through the AUC results where the AUC is higher by 6.6% when we use all of the features, and the confusion matrix has a better repartition, as well as the model being able to predict more true positives.

6.3 Using Bagging

6.3.1 For 50 features

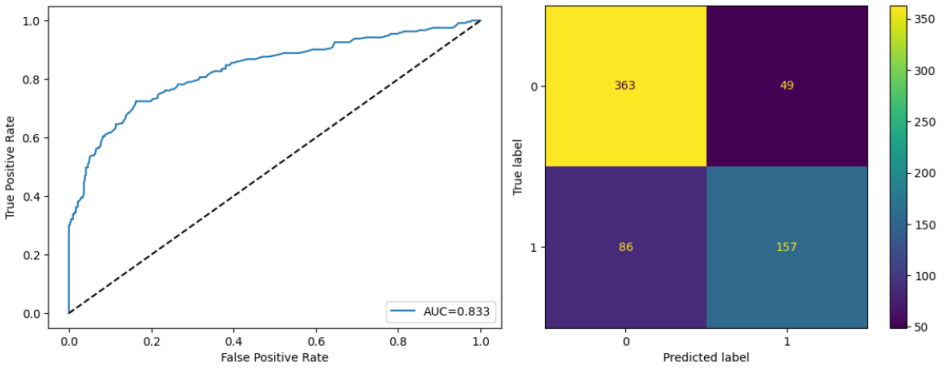

Figure 21: ROC and Confusions Matrix for Bagging with 50 Features

6.3.2 For all features

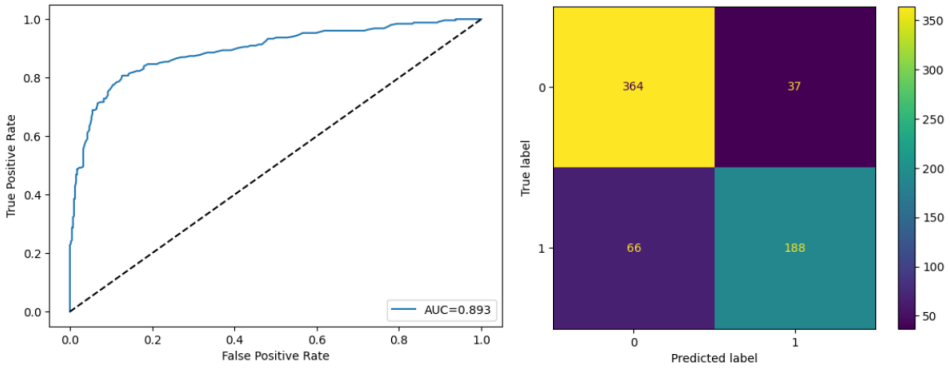

Figure 22: ROC and Confusions Matrix for Bagging with all Features

Summary By using GridSearch and CrossValidation, We were able to tune the algorithm and extract the best possible model out of it. The hyper-parameters used to obtain these results are : "Number of Estimators : 300" This model was able to reach 79.38% in accuracy, 76.21% in Precision, and finally 64.60% in recall (for 50 features). On the other side (using full features) we were able to obtain to obtain 84.27% in accuracy, 83.55% in precision and 74.01% in recall. This proves that using the full range of features helps the algorithm to learn more in order to give better predictions. We can see this through the AUC results where the AUC is higher by 6.0% when we use all of the features, and the confusion matrix has a better repartition, as well as the model being able to predict more true positives.

6.4 Using Random Forest

6.4.1 For 50 features

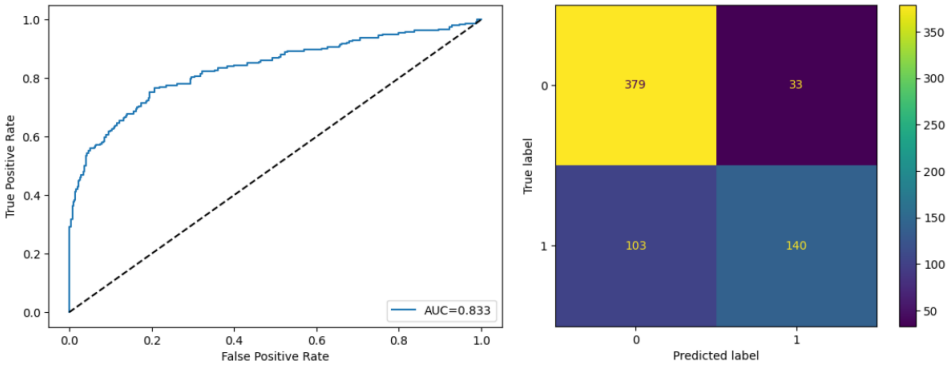

Figure 23: ROC and Confusions Matrix for Random Forest with 50 Features

6.4.2 For all features

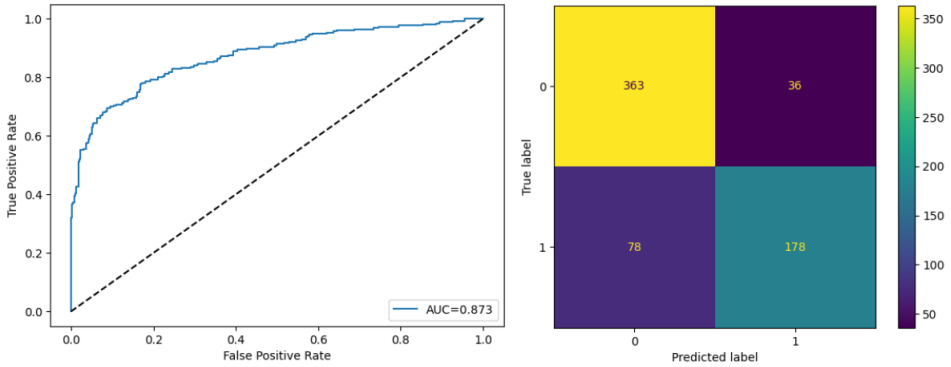

Figure 24: ROC and Confusions Matrix for Random Forest with all Features

Summary By using GridSearch and CrossValidation, We were able to tune the algorithm and extract the best possible model out of it. The hyper-parameters used to obtain these results are : "Number of Estimators : 100", "Max Depth = 10", "Min Samples Split = 10" This model was able to reach 79.23% in accuracy, 80.92% in Precision, and finally 57.61% in recall (for 50 features). On the other side (using full features) we were able to obtain to obtain 82.59% in accuracy, 83.17% in precision and 69.53% in recall. This proves that using the full range of features helps the algorithm to learn more in order to give better predictions. We can see this through the AUC results where the AUC is higher by 4.0% when we use all of the features, and the confusion matrix has a better repartition, as well as the model being able to predict more true positives.

6.5 Benchmark and Results

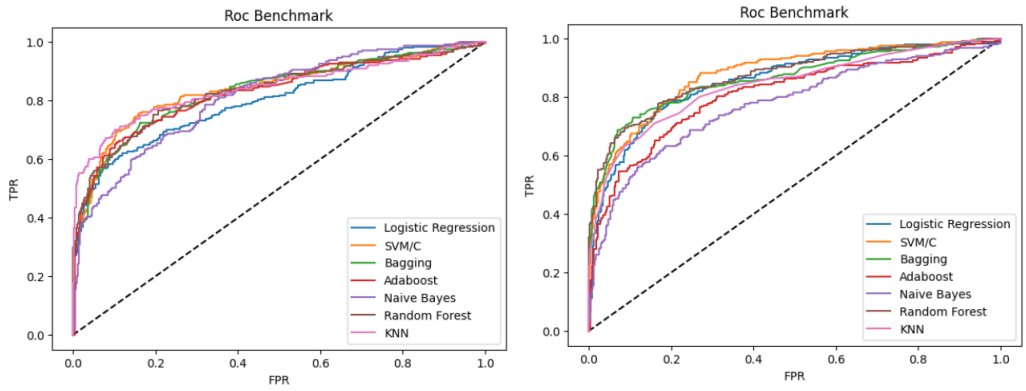

Figure 25: ROC of All Algorithms Used

Dataset	Algorithm	Accuracy	Precision	Recall
1	SVM	0.80	0.83	0.59
2	SVM	0.83	0.83	0.71
1	LR	0.76	0.71	0.62
2	LR	0.80	0.73	0.76
1	Bagging	0.79	0.76	0.64
2	Bagging	0.84	0.83	0.74
1	Adaboost	0.78	0.73	0.67
2	Adaboost	0.74	0.66	0.70
1	Naives Bayes	0.75	0.68	0.62
2	Naives Bayes	0.75	0.70	0.61
1	KNN	0.82	0.90	0.60
2	KNN	0.78	0.88	0.52
1	Random Forest	0.79	0.80	0.57
2	Random Forest	0.82	0.83	0.69

Table 2: Benchmark Results for Task 2.

6.5.1 Conclusion

From all of the previous summaries, we have noticed that having full features is always better (or slightly better in some cases) when it comes to classifying the micro-calcifications, even Naïve Bayes was able to generalise well using the second data-set. In the first part Naïve Bayes was not performing well when it comes to classifying the calcifications. But using the second data-set the model was able to perform well where the AUC went from 0.675 to 0.814 (13.9% performance improvement). The second data-set not only classifies the microcalcification per target, but also depending on the patient. This means that we always have the patient data grouped which gives more sense to the algorithm to learn from, and thus be effective to when it comes to classifying.

For the model trained with only 50 features, we can observe from the results that "Bagging" is the winner. As for the model trained with all features, the best algorithm was "Support Vector Machine" holding an AUC of 0.894, and an Accuracy, Precision, and Recall of 80%, 83%, and 71%.

7 Conclusion

Breast cancer classification (in our case Microcalcification) plays a vital role in the diagnosis of patients.

With new technologies and with the availability of large-scale data-sets from either public or private institutes, machine learning algorithms have demonstrated to us that they can be a powerful tool when it comes to classification.

Through out this paper, we have explored various methods and techniques employed in breast cancer classification. We have used so multiple machine learning algorithms such as support vector machines, random forests, and bagging, adaboost models and these approaches have demonstrated promising results in improving the accuracy, precision, and reliability of breast cancer diagnosis.

After having analysed the results comming from the multiple benchmarks that we had, we can conclude that the second data-set gave us the ability to create better models for microcalcification classification. The best model generate was done by using the SVM algorithm having hyperparameters initialised as "Kernel : RBF, Gamma : Auto". This model scored an AUC of 0.894, and was able to separate all prediction possiblities while highlighting True Negatives, and True Positives [Figure 18].

These results can be enhanced by training the models on bigger datasets, or by running multipme other ensemble learning algorithms such as "XGBoost". Parameter tuning is also a big factor in getting the best out of an algorithm, but this process takes a lot of time, and might require hours of training in order to find the perfect parameters which can also be sped up by having access to powerful computers.

8 Table Of Figures

List of Tables

List of Figures

9 Webography

Websites, and documentation used to write the report.

References

[AWS] AWS. *What is logistic regression ?* URL: https://aws.amazon.com/de/what-is/logistic-regression. (accessed : 05.05.2023).

[e-c] e-cancer.fr. *Calcifications mammaires.* URL: https://www.e-cancer.fr/Patients-et-proches/Les-cancers/Cancer-du-sein/Les-maladies-du-sein/Calcifications-mammaires. (accessed : 05.05.2023).

[IBMa] IBM. *How SVM Works.* URL: https://www.ibm.com/docs/en/spss-modeler/saas?topic=models-how-svm-works. (accessed : 05.05.2023).

[IBMb] IBM. *What is Bagging ?* URL: https://www.ibm.com/topics/bagging. (accessed : 05.05.2023).

[IBMc] IBM. *What is random forest?* URL: https://www.ibm.com/topics/random-forest. (accessed : 19.05.2023).

[Org] W. H. Organization. *Calcifications mammaires.* URL: https://www.who.int/news-room/fact-sheets/detail/breast-cancer. (accessed : 05.05.2023).