Breast Cancer Classification - Machine Learning Perspective

Anastasios Papazafeiropoulos Tristan Hermanns

June , 2023

1 Introduction

Breast cancer is a significant global health concern that affects many individuals. Breast cancer is a disease in which cells in the breast grow out of control. There are different kinds of breast cancer. The kind of breast cancer depends on which cells in the breast turn into cancer. Breast cancer can begin in different parts of the breast. A breast is made up of three main parts: lobules, ducts, and connective tissue. The lobules are the glands that produce milk. The ducts are tubes that carry milk to the nipple. The connective tissue (which consists of fibrous and fatty tissue) surrounds and holds everything together. Most breast cancers begin in the ducts or lobules. Breast cancer can spread outside the breast through blood vessels and lymph vessels. When breast cancer spreads to other parts of the body, it is said to have metastasized. For all these reasons, accurate classification and early detection of breast cancer cases are vital for effective treatment planning and improving patient outcomes. Machine learning techniques have the potential to automate the classification process by analyzing various factors associated with breast cancer cases [1].

1.1: Related work

It is increasingly common nowadays for machine learning to be involved in the field of medicine to assist doctors, either for diagnosis or for the optimal selection of treatment. For this reason, numerous studies have been conducted on this topic. Most of them focus on predicting breast cancer using different machine-learning approaches applying demographic, laboratory, and mammographic data [2].

1.2: Our work

In this project, our aim is to develop a breast cancer classification model to predict if the person who has been diagnosed with this type of cancer is alive or not. This will help the doctors to decide about the best treatment and calculate the survival odds in a more accurate way for each client. To achieve this, we use a dataset that includes relevant features, which is going to be presented further in the next paragraph. We are going to implement eight different models, namely K-Nearest Neighbors (KNN) (with k=3, 5, 15), Logistic Regression, MLP (with “relu” and “sigmoid” activation function. To evaluate the

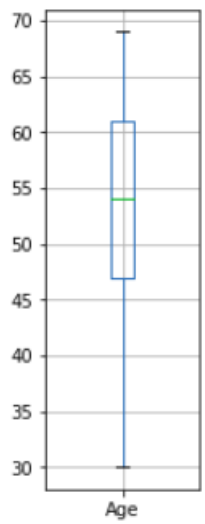
performance of each model, we are using k-Fold Validation with k=5 and present a comparison table which includes all the implemented models.

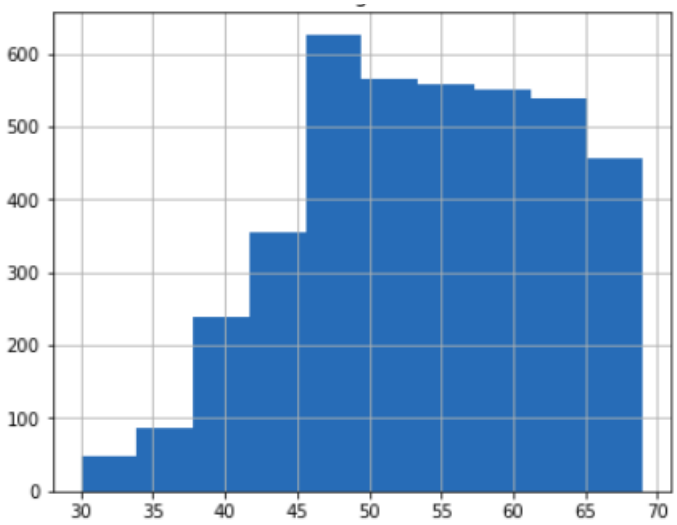
2. Dataset and Features

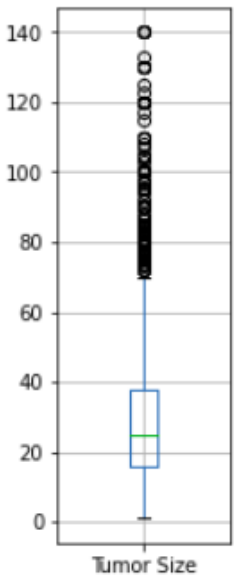
The dataset used in this project is a dataset of breast cancer patients that was obtained from the 2017 November update of the SEER Program of the NCI and provides information on population-based cancer statistics. It involved female patients with infiltrating duct and lobular carcinoma breast cancer (SEER primary cites recode NOS histology codes 8522/3) diagnosed in 2006-2010. The dataset contains a comprehensive set of attributes that are crucial for breast cancer classification. It encompasses demographic, clinical, and pathological characteristics of patients diagnosed with breast cancer. These attributes capture various aspects of the patients' profiles, providing valuable insights for predicting the outcome of breast cancer. They include, also, features such as age, race, marital status, tumor size, and tumor grade, which quantifies the aggressiveness of the tumor on a scale of 1 to 4. Patients with unknown tumor size, examined regional LNs, positive regional LNs, and patients whose survival months were less than 1 month were excluded; thus, 4024 patients were ultimately included. Additionally, the dataset provides information on the extent of cancer spread, indicating whether the cancer is localized or has spread to regional lymph nodes. These features collectively encompass critical demographic, clinical, and pathological factors that significantly influence the prognosis and treatment strategies for breast cancer patients.

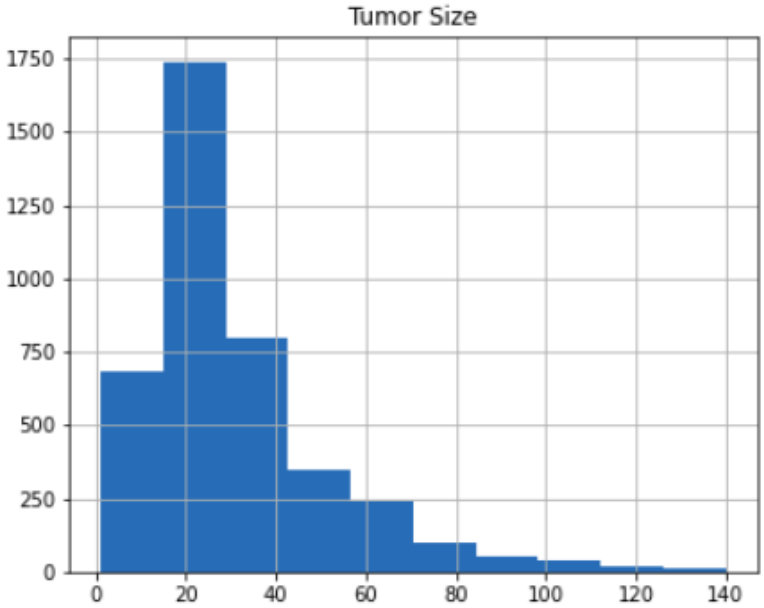
2.1: Preprocessing

First, the data was converted from the .csv file to a dataframe to make it accessible. Then, unnecessary columns were removed, and categorical variables were converted to numerical using the labelRencoder() function, which assigned numeric values based on the alphabetical order of the data. The "Status" column was chosen as the target value representing the outcome of the patients, differentiating between those who have passed away and those who are still alive with the following mapping: "Dead"→ 1 and "Alive"→ 0. In other words, the dataset was splitted into two components: the feature matrix (X), which includes all the independent variables, and the target variable (y), corresponding to "Status" column. This division enables the machine learning models to learn from the features and predict the survival status of future patients based on their unique characteristics. Then, outliers were founded based on the IQR Method for the columns "Age" and "Tumor Size" and the following diagrams were constructed:

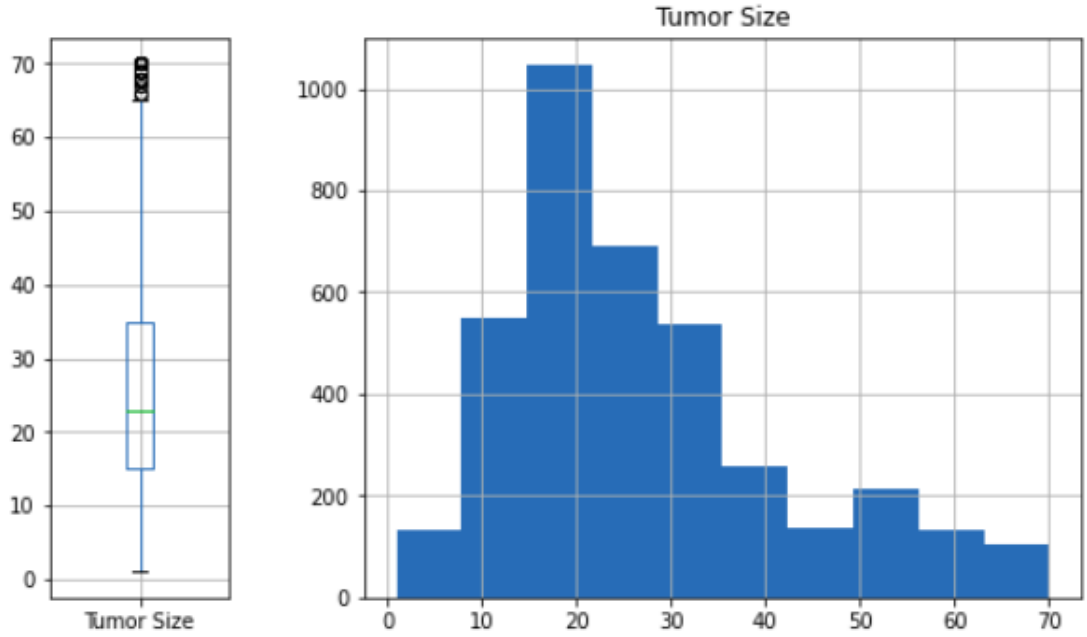








As emerged from the analysis, it was found that there are no outliers for the Age and there are only 222 outliers (bigger values) for the Tumor size. This is logical based on the monitoring of the above diagrams. After the trimming, the Boxplot for the Tumor Size has the following shape:



So, after all the dataset is the following:

	Age	Race	Marital Status	T Stage	N Stage	6th Stage	differentiate	Grade	A Stage	Tumor Size	Estrogen Status	Progesterone Status	Regional Node Examined	Reginol Node Positive	Status
0	68	2	1	0	0	0	1	3	1	4	1	1	23	0	0
1	50	2	1	1	1	2	0	2	1	35	1	1	13	4	0
2	58	2	0	2	2	4	0	2	1	63	1	1	13	6	0
3	58	2	1	0	0	0	1	3	1	18	1	1	1	0	0
4	47	2	1	1	0	1	1	3	1	41	1	1	2	0	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
4019	62	1	1	0	0	0	0	2	1	9	1	1	0	0	0
4020	56	2	0	1	1	2	0	2	1	46	1	1	13	7	0
4021	68	2	1	1	0	1	0	2	1	22	1	0	10	2	0
4022	58	0	0	1	0	1	0	2	1	44	1	1	10	0	0
4023	46	2	1	1	0	1	0	2	1	30	1	1	6	1	0

3802 rows × 15 columns

Before the models implementation, the data were splitted into training and testing sets (0.8 and 0.2 of the size of data, respectively) and the features were scaled by using StandardScaler() function.

3 Models and Results

3.1 K-Nearest Neighbors (KNN)

KNN is a non-parametric algorithm that classifies instances based on the majority class of their neighboring points in the feature space. In this approach, the KNN classifier was initialized with k=3, 5, 15 considering the k nearest neighbors to make a prediction. The algorithm calculates the distances between the instances and selects the most common class label among the neighbors as the prediction. The following results were obtained:

k=3:

```
KNeighborsClassifier(n_neighbors=3)
```

	precision	recall	f1-score	support
0	0.89	0.95	0.92	661
1	0.38	0.20	0.26	100
accuracy			0.85	761
macro avg	0.63	0.58	0.59	761
weighted avg	0.82	0.85	0.83	761

k=5:

```
KNeighborsClassifier()
```

	precision	recall	f1-score	support
0	0.89	0.97	0.93	661
1	0.51	0.19	0.28	100
accuracy			0.87	761
macro avg	0.70	0.58	0.60	761
weighted avg	0.84	0.87	0.84	761

k=15:

```
KNeighborsClassifier(n_neighbors=15)
```

	precision	recall	f1-score	support
0	0.88	0.99	0.93	661
1	0.54	0.07	0.12	100
accuracy			0.87	761
macro avg	0.71	0.53	0.53	761
weighted avg	0.83	0.87	0.82	761

3.2: Logistic Regression

Logistic Regression is a linear model commonly used for binary classification problems. It models the relationship between the input features and the probability of belonging to a specific class using a logistic function. In this approach, logistic regression was applied to the scaled training data to build a classification model. The model estimates the coefficients for each feature, which are then used to predict the probability of breast cancer for each instance. By selecting a suitable threshold (usually 0.5), the instances were classified as either positive or negative (“Alive or “Dead). The following results were obtained:

```
LogisticRegression()
```

	precision	recall	f1-score	support
0	0.88	0.99	0.93	661
1	0.60	0.12	0.20	100
accuracy			0.87	761
macro avg	0.74	0.55	0.57	761
weighted avg	0.84	0.87	0.84	761

3.3: Neural Networks

A standard neural network (NN) consists of many simple, connected processors called neurons, each producing a sequence of real-valued activations. Input neurons get activated through sensors perceiving the environment, other neurons get activated through weighted connections from previously active neurons. Some neurons may influence the environment by triggering actions. Learning or credit assignment is about finding weights that make the NN exhibit desired behavior, such as driving a car. Depending on the problem and how the neurons are connected, such behavior may require long causal chains of computational stages, where each stage transforms (often in a non-linear way) the aggregate activation of the network. Deep Learning is about accurately assigning credit across many such stages. [3] In the following work, the neural networks were implemented by using the scikit learn library.

3.3.1 Multi-Layer Perceptron

The multilayer perceptron is the most known and most frequently used type of neural network. On most occasions, the signals are transmitted within the network in one direction: from input to output. Layers which are not directly connected to the environment are called hidden. Each neuron has an activation function. The power of the multilayer perceptron comes precisely from non-linear activation functions. Almost any non-linear function can be used for this purpose, except for polynomial functions. [4] Currently, the functions most commonly used today are the single-pole (or logistic) sigmoid and the ReLu function. So in this study, two Multilayer perceptrons were implemented, one with each function. As a solver, 'lbfgs' was selected and 1000 was defined as the maximum number of iterations in both implementations. The following results were obtained:

-logistic:

	precision	recall	f1-score	support
0	0.87	1.00	0.93	661
1	0.00	0.00	0.00	100
accuracy			0.87	761
macro avg	0.43	0.50	0.46	761
weighted avg	0.75	0.87	0.81	761

-ReLu:

	precision	recall	f1-score	support
0	0.87	1.00	0.93	661
1	0.00	0.00	0.00	100
accuracy			0.87	761
macro avg	0.43	0.50	0.46	761
weighted avg	0.75	0.87	0.81	761

3.3.1: Gaussian Naive Bayes:

Learning can be greatly simplified by the Naïve Bayes classifier by supposing that features are independent of a given class. Although, the assumptions of independence are poor in general. Practically, with a more sophisticated classifier, Naive Bayes often competes effectively. The feature vector described that to a given example, Bayesian classifiers allocate the most likely class, i.e., $P(C) = \prod_i^n p(X_i|C)$, where C is the classifier,

and is a feature vector. In practice, the resulting classifier known as Naïve Bayes is very successful despite this unrealistic assumption, by competing again and again with much more sophisticated techniques. The model of Naïve Bayes is a simplified version of Bayesian probability. The operations of the Naïve Bayes classifier are done on a sturdy independence assumption. This means that one attribute of the probability of one does not have any effect on the probability of the rest of the attributes [5]. In this study, a Gaussian Naive Bayes was implemented by using the GaussianNB() function from the scikit learn library. The following results were obtained:

	precision	recall	f1-score	support
0	0.91	0.88	0.89	661
1	0.34	0.39	0.36	100
accuracy			0.82	761
macro avg	0.62	0.64	0.63	761
weighted avg	0.83	0.82	0.82	761

3.4 Model Evaluation

3.4.1: Results Interpretation and Evaluation

The metrics which were used above must be defined. So, we introduce the following definitions:

• Accuracy: Measures the proportion of correctly classified instances out of the total number of instances. It provides an overall assessment of the model's correctness.

• Precision: Represents the proportion of correctly identified positive cases out of the total instances predicted as positive. It measures the model's ability to avoid false positives.

• Recall: Also known as sensitivity or true positive rate, measures the proportion of actual positive cases that the model correctly identifies. It indicates the model's ability to retrieve positive cases.

• F1-score: The harmonic mean of precision and recall, providing a single metric that considers both measures. It is useful when we want to balance precision and recall, as it gives equal weight to both.

In machine learning, and specifically in classification, the interpretation and evaluation of results are a complex and multifactorial process. More specifically, to determine which model is better for a problem, parameters such as the problem itself,

resolution time, cost of resolution (both in monetary and computational terms) need to be taken into account. In the specific problem now, the goal is to predict the outcome of the disease so that the appropriate treatment can be selected by the doctor. In this case, the most important factor is the prediction of survival for a woman with certain features. Therefore, it is crucial to make accurate predictions, even if the computational cost is slightly higher. Specifically, the accuracy should be as high as possible along with a high recall. In the case where the goal was the diagnosis of the disease, the focus would be on minimizing False Negative results as much as possible. Now, we want to keep the False Negatives low, but it is critical to also keep the False Positives low because in this case, the doctor may choose a more "aggressive" treatment that could potentially cause side effects to the patient. Therefore, in this context, we primarily want the accuracy to be high, as well as the recall. In this perspective, the two best models developed so far are the logistic regression model and the multilayer perceptron (using the ReLU activation function was selected for the continuation of the study). Both models achieve accuracy equals to 0.87, that means correct prediction in 87% of the cases.

3.4.2: K-fold Cross Validation:

Cross-validation is a technique for evaluating a machine learning model and testing its performance. It is used commonly in applied ML tasks. It helps in comparing and selecting an appropriate model.CV tends to have a lower bias than other methods. The numerical value of k in a k-fold cross-validation training technique of machine learning predictive models is an essential element that impacts the model's performance. A right choice of k results in better accuracy, while a poorly chosen value for k might affect the model's performance. In literature, the most commonly used values of k are five (5) or ten (10), as these two values are believed to give test error rate estimates that suffer neither from extremely high bias nor very high variance. However, there is no formal rule [6]. After using this method with k=5 to evaluate all the implemented models, the following results were obtained:

	F-score (cv)	Precision (cv)	Recall (cv)	Accuracy	Time(s)
Logistic Regression	0.550065	0.732945	0.546754	0.855969	0.043566
MLP-relu	0.45957	0.425189	0.5	0.850378	0.015791
KNN-15	0.509218	0.662048	0.522577	0.848734	0.010908
KNN-5	0.54828	0.614632	0.5444	0.836566	0.008242
KNN-3	0.552718	0.583192	0.547026	0.819467	0.020387
MLP-sigmoid	0.552718	0.583192	0.547026	0.819467	0.374980
Gaussian Naive Bayes	0.618122	0.612956	0.625279	0.795461	0.002538

3.4.3: Principal Components Analysis

PCA is a classical statistical method for transforming attributes of a dataset into a new set of uncorrelated attributes called principal components (PCs). PCA can be used to reduce the dimensionality of a dataset, while still retaining as much of the *variability* of the dataset as possible. The goal of this research is to determine if PCA can be used to improve the performance of machine learning methods in the classification of such high-dimensional data [7]. As derived from the above comparative table, the best model for this specific problem is Linear Regression. So, after applying PCA to this model, the following results are obtained:

```
LogisticRegression()
```

	precision	recall	f1-score	support
0	0.88	0.99	0.93	661
1	0.53	0.09	0.15	100
accuracy			0.87	761
macro avg	0.70	0.54	0.54	761
weighted avg	0.83	0.87	0.83	761

So now, the comparison table is:

	F-score (cv)	Precision (cv)	Recall (cv)	Accuracy	Time(s)
Logistic Regression	0.550065	0.732945	0.546754	0.855969	0.043566
Logistic Regression-PCA	0.538643	0.731794	0.539835	0.854981	0.032149
MLP-relu	0.45957	0.425189	0.5	0.850378	0.015791
KNN-15	0.509218	0.662048	0.522577	0.848734	0.010908
KNN-5	0.54828	0.614632	0.5444	0.836566	0.008242
KNN-3	0.552718	0.583192	0.547026	0.819467	0.020387
MLP-sigmoid	0.552718	0.583192	0.547026	0.819467	0.374980
Gaussian Naive Bayes	0.618122	0.612956	0.625279	0.795461	0.002538

5 Conclusions

In this breast cancer classification project, we employed eight different models: K-Nearest Neighbors (KNN) (with k=3, 5, 15), Logistic Regression, MLP (with “relu” and “sigmoid” activation function. On the other hand, the neural network model showed a more balanced performance, with a higher recall value. The neural network model captured a larger number of positive cases but at the cost of increased false positives. It is important to note that further fine-tuning and optimization of the models may enhance their

performance. Additionally, incorporating additional features or exploring alternative algorithms could be considered in future iterations of this project.
It is worth noting that machine learning models have an advisory role towards doctors, which is increasingly growing. However, we should never fully rely on a model and the solution it provides to a problem without supervising the problem itself.
Overall, this project demonstrates the application of machine learning techniques for breast cancer classification, providing a foundation for developing more accurate models in the future.

6 Bibliography

[1]: Non-BRCA1/BRCA2 high-risk familial breast cancers are not associated with a high prevalence of BRCAness, Lars v. B. Andersen, Martin J. Larsen

[2]: Prediction of Breast Cancer using Machine Learning Approaches Reza Rabiei,1 Seyed Mohammad Ayyoubzadeh,2 Solmaz Sohrabei,3* Marzieh Esmaeili,2 and Alireza Atashi4Author information Article notes Copyright and License information Disclaimer

[3]: Schmidhuber, J. (2015). Deep learning in neural networks: An overview. Neural Networks, 61, 85–117. doi:10.1016/j.neunet.2014.09.003

[4]: Multilayer Perceptron and Neural Networks MARIUS-CONSTANTIN POPESCU1 VALENTINA E. BALAS2 LILIANA PERESCU-POPESCU3 NIKOS MASTORAKIS4

[5]:Gaussian Naïve Bayes Algorithm: A Reliable Technique Involved in the Assortment of the Segregation in Cancer M. Vijay Anand,1B. KiranBala,2S. R. Srividhya,3Kavitha C.,3Mohammed Younus,4and Md Habibur

[6]: Performance of Machine Learning Algorithms with Different K Values in K-fold Cross-Validation Isaac Kofi Nti

[7]: The effect of principal component analysis on machine learning accuracy with high-dimensional spectral data Tom Howley a , Michael G. Madden a,*, Marie-Louise O'Connell b