# Diagnostic Status Prediction in Breast Cancer Patients using Machine Learning Models

Noah KWA MOUTOME Victor TAN

December 9, 2023

## 1 Introduction

Nowadays, in the realm of healthcare and data analytics, the diagnosis of breast cancer has become a crucial and challenging task. In 2020 alone, 685,000 deaths were attributed to breast cancer, according to the World Health Organization. Remarkably, nearly half of all breast cancers manifested in women for no apparent reason, emphasizing the need for comprehensive diagnostic approaches beyond traditional risk profiling based on sex and age alone.

The complexity of breast cancer necessitates advanced methods for diagnosis and prediction. Accurate identification of malignant tumors is of the utmost importance, as it empowers healthcare professionals to initiate preventive and effective treatments, and so potentially improving patient outcomes and reducing mortality rates. Moreover, such predictive capabilities offer the possibility of adaptive interventions, contributing to a proactive approach in managing the risks associated with breast cancer.

In this context, this project focuses on the diagnosis and prediction of breast cancer thanks to the "Breast Cancer Wisconsin (Diagnostic) Data Set" obtained from the computer sciences department at the University of Wisconsin. By employing sophisticated data analytics and machine learning techniques, we aim to develop predictive models capable of discerning whether a tumor is benign or malignant based on detailed characteristics extracted from images of tumor cells. Through this research, we try to contribute to the ongoing efforts to enhance diagnostic accuracy and advance our understanding of breast cancer, ultimately improving patient care and outcomes.

## 2 Preprocessing and Data Exploration

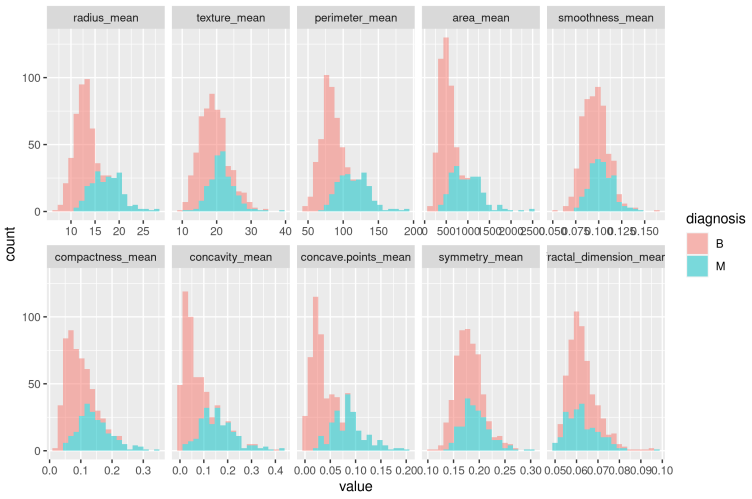
There's no missing values in this dataset, but we have to drop the last unnamed column and the "id" one which is not usefull.
So now, we have **569 observations** and **30 covariables** to predict 1 target variable, which is the diagnosis.

- `Texture`
- `Area`
- `Perimeter`
- `Symmetry`
- `Compactness` ($\frac{périmètre^2}{surface - 1.0}$)
- `Concave points` (number of concave portions of the contour)
- `Smoothness` (local variation in radius lengths)
- `Concavity` (severity of concave portions of the contour)
- `Fractal dimension` ("coastline approximation" - 1)
- `Radius` (mean of distances from center to points on the perimeter)

Then, the mean, the standard error, and the "worst" or largest (mean of the three largest values) of these features were computed for each image, resulting in 30 features. For instance, field 3 is Mean Radius, field 13 is Radius SE, field 23 is Worst Radius.
Furthermore, there are 212 Malignant diagnoses and 357 Benign diagnoses, which is a relatively balanced split: we do have enough data from each diagnoses to study the dataset correctly.

# 3 Description of the dataset

Let's now visualize the data to see if there's any outliers or if the data is normally distributed. These histograms are repetitive, so we will only show the means of the 10 features. The others are still available in the code for the reader to see. It looks like most of the features are normally distributed, and there seems to be no outliers. However, there's no clear separation between benign and malignant for most of the features except `concave.points_worst`, `perimeter_worst`, `radius_worst`, and `concave.points_mean`. So a visualization of the values of these features is not enough.




Also, we can see that the scale of the features is not the same, which could be a problem for the model. We will use the function "scale". Let's now take a look at the correlation plot:

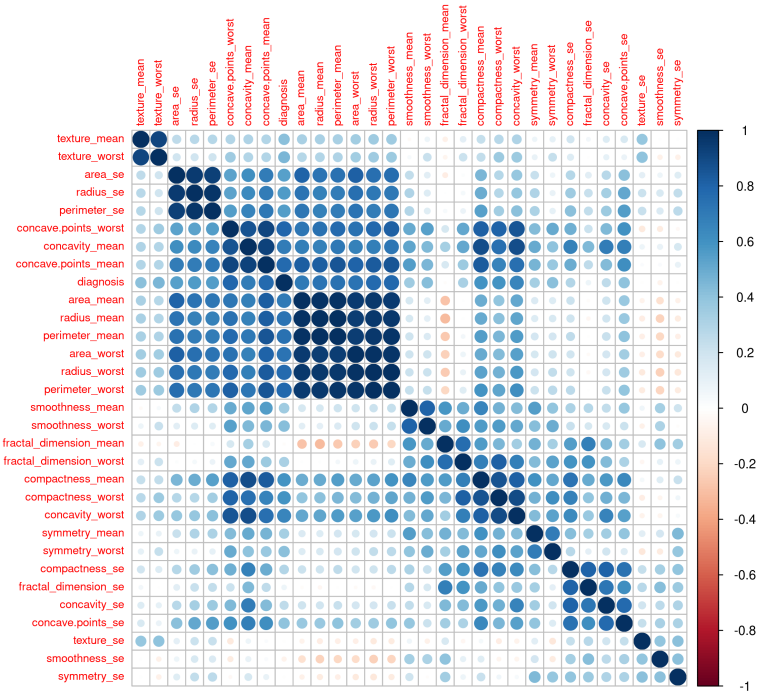



Figure 1: Correlation Matrix

We can see that there's a strong correlation between some features, which can be a problem for the model. Furthermore, there is a significant correlation between diagnosis and the following variables: `diagnosis`, `radius_mean`, `perimeter_mean`, `area_mean`, `concave.points_mean`, `radius_worst`, `perimeter_worst` `area_worst`, `concave.points_worst`. That could mean that these co-variables may be important for the prediction of the target variable.

# 4 Model training and prediction

We will try 5 approaches to study the dataset:
- Full dataset study (no treatment)
- Ridge Penalization
- Backward Selection
- Co-variables deletion by highest correlation
- Co-variables deletion by PCA

We will use a k-fold method to build a model training with the 5 approaches, and predict the diagnosis. We will use K = 5, meaning that 20% of the 569 observations will compose the test set and 80% will compose the training set.

## 4.1 Full dataset

We still use the same dataset and use the glm function.

### 4.1.1 No treatment

| **predictions_backward** | **0** | **1** |
|---|---|---|
| **true_values_backward** | | |
| **0** | 339 | 18 |
| **1** | 12 | 200 |
| Accuracy = 0.947 | | |

Table 1: Confusion Matrix for No Treatment Prediction

Without any treatment of the dataset, we get a 0.947 accuracy, which is already a good result. But let's see if we can improve it.

### 4.1.2 Ridge penalization

| **predictions_ridge** | **0** | **1** |
|---|---|---|
| **true_values_ridge** | | |
| **0** | 355 | 2 |
| **1** | 8 | 204 |
| Accuracy = 0.982 | | |

Table 2: Confusion Matrix for Ridge Regression Predictions

Without deleting any covariables, we get a 0.98 accuracy, which is a better result than the previous one. The Ridge method is a penalization method that shrinks the coefficients of the covariables that are not useful for the model. It reduces overfitting and may improve the accuracy of the model when covariables are correlated, which is the case here.

But now, let's see if we can get a better accuracy by deleting some covariables.

## 4.2 Backward Selection

We will now use the backward selection method to delete the co-variables that are not useful for the model with the function `step`.
With this method, the co-variables left are:

| | | | |
|---|---|---|---|
| diagnosis | radius_mean | texture_mean | |
| area_mean | smoothness_mean | compactness_mean | |
| concavity_mean | concave.points_mean | symmetry_mean | |
| fractal_dimension_mean | perimeter_se | area_se | smoothness_se |
| compactness_se | concavity_se | concave.points_se | symmetry_se |
| fractal_dimension_se | radius_worst | texture_worst | perimeter_worst |
| area_worst | concavity_worst | symmetry_worst | fractal_dimension_worst |

So we get a total of 24 co-variables, which means that only 6 co-variables have been deleted, which isn't that much. Here's the result of the `K-fold`:

| **predictions_backward** | **0** | **1** |
|---|---|---|
| **true_values_backward** | | |
| **0** | 340 | 17 |
| **1** | 12 | 200 |
| Accuracy = 0.949 | | |

Table 3: Confusion Matrix for Backward Selection Predictions

We can see that the accuracy is only slightly better than the default one with no treatment. So the backward method did improve the model, but not by a lot. Also, the model is less accurate than the Ridge one, so it should not be privileged over the Ridge method.

## 4.3 Co-variables deletion by highest correlation

Here, we use a simple principle to select the co-variables we'll use for the model. If two co-variables are highly correlated, (for instance, if the correlation is higher than 0.8), we will delete the one that has the highest mean of correlations with the others. We can use the function `findCorrelation` from the package `caret` to provide us the said co-variables that we will delete.
With this method, the co-variables left are :

| | | | |
|---|---|---|---|
| diagnosis | perimeter_mean | concave.points_mean | symmetry_mean |
| fractal_dimension_mean | radius_se | area_se | compactness_se |
| concavity_se | concave.points_se | symmetry_se | radius_worst |
| area_worst | concave.points_worst | symmetry_worst | fractal_dimension_worst |

So we get a total 15 co-variables, which is half of the initial variables. This approch is not perfect, because we only scan the correlation between two covariables once: there could still be high correlations between variables if 3 or more covariables were correlated together, and deleting more covariables could reduce too much information. Here's the results of the `K-fold`:

| **predictions_correlation** | **0** | **1** |
|---|---|---|
| **true_values_correlation** | | |
| **0** | 345 | 12 |
| **1** | 14 | 198 |
| Accuracy = 0.954 | | |

Table 4: Confusion Matrix for Correlation Deletion Predictions

With this method, we get a better accuracy than the default one. However, the accuracy is globally as good as the one with the backward selection method. Still, it is less accurate than the Ridge method, so it should not be privileged over the Ridge method.

## 4.4 Co-variables deletion by PCA

We will use the PCA method to reduce the number of co-variables. We will use the function `prcomp` from the package `stats` to do so. Because the dataset has already been centered and scale, we will use the parameters `center = FALSE` and `scale. = FALSE`.

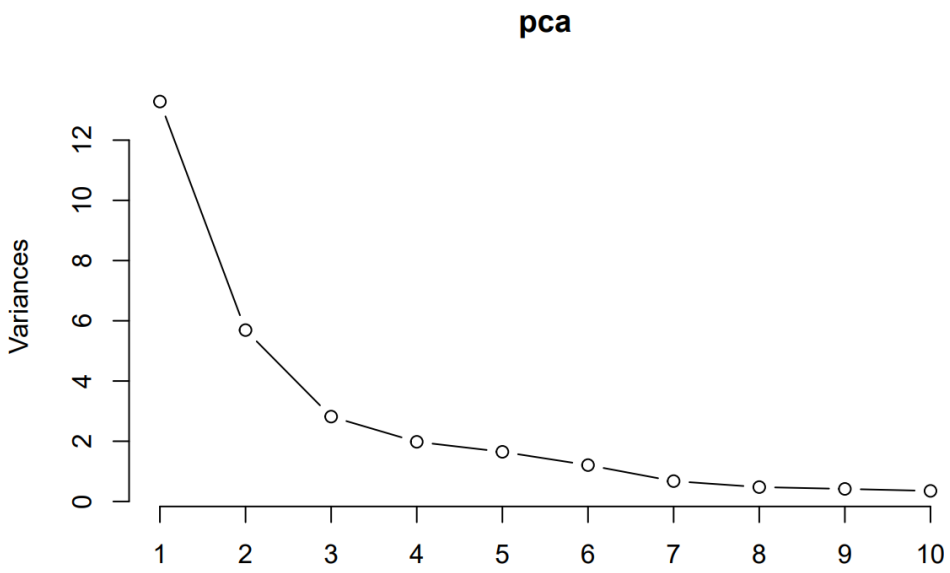



Figure 2: PCA - Variance given PC

Let's arbitrarily choose to keep enough co-variables to explain at least 99% of the variance. According to the summary, we need to keep 17 co-variables to do so (99.11% of the variance explained).
Here's the results of the predictions using the PCA method:

| **predictions_pca** | **0** | **1** |
|---|---|---|
| **true_values_pca** | | |
| **0** | 350 | 7 |
| **1** | 10 | 202 |
| Accuracy = 0.97 | | |

Table 5: Confusion Matrix for PCA Predictions

With this method, we get a better accuracy than the default one. The accuracy is globally better than the one with the backward selection method and the one with the co-variables deletion by highest correlation method. But, it is still less accurate than the Ridge method, so it should not be privileged over the Ridge method.

# 5 Conclusion

We can see that the Ridge method is the most accurate, followed by PCA by only a slight percentage. Moreover, the Ridge method gives the lowest False Negative, which is the only value that we want to minimize: False positives (when the prediction gives "Malignant" when the true value is "Benign") are not really a issue. But False negatives (when the prediction gives "Benign" when the true value is "Malignant") are the worst case scenario, as the patient might be in danger without our knowledge.

The results of our models obtained from the breast cancer classification project seem to be really accurate overall. The highest accuracy, Sensitivity and Specificity of the prediction on 569 observations is achieved by the Ridge Model. It may hold promise for supporting clinical decision-making of the diagnosis of tumors given images.
On a broader scale, we can then decide to make another type of model, where we can make a greater penalization on False negatives, which are the cases we want to avoid. It will probably reduce the accuracy of the current model, but would be more helpful in our scenarios.