Prediction on Breast Cancer: An application of Logistic Regression

Ankita Das
adas4@buffalo.edu

Abstract

This paper presents the development and the simulation of a machine learning model of two class problem created with logistic regression to predict breast cancer . For obtaining a reasonable model, we did use a real dataset delivered by Wisconsin Diagnostic Breast Cancer (WDBC) database where the features used for classification are pre-computed from images of a fine needle aspirate (FNA) of a breast mass. Our objective is to classify suspected FNA cells to Benign (class 0) or Malignant (class 1) using logistic regression as the classifier. The first part of this work has been dedicated to pre-process the data to optimize the classifier. Next, we need to examine the dataset, what it contains, when and how it was created, if it is noisy, if it has missing values. This section is important to understand what are the issues that will need to be processed while preparing the data to create the classifier. For the implementation of the model, the dataset was partitioned in the following fashion: 80% for training phase, and 20% for the testing and validation phase. The next step is to apply Logistics Regression methods and algorithms to optimize the training set. The hyper-parameters have been assigned manually for the learning process and we estimate the best model parameters which gives us an accuracy of 94.74%.

1. Introduction

Breast cancer (BC) is one of the most common cancers among women worldwide, representing the majority of new cancer cases and cancer-related deaths according to global statistics, making it a significant public health problem in today's society.

The early diagnosis of BC can improve the prognosis and chance of survival significantly, as it can promote timely clinical treatment to patients. Further accurate classification of benign tumors can prevent patients from undergoing unnecessary treatments. Thus, the correct diagnosis of BC and the classification of patients into malignant or benign groups is the subject of much research. Because of its unique advantages in critical features detection from complex BC datasets, machine learning (ML) is widely recognized as the methodology of choice in BC pattern classification and forecast modeling.

Classification is an effective way to classify data. Especially in the medical field, where those methods are widely used in diagnosis and analysis to make decisions.

This paper presents a study on the said topic using logistic regression. For obtaining a reasonable model, we did use a real dataset delivered by Wisconsin Diagnostic Breast Cancer (WDBC)database where it is filled by real data from patients. The database has 569 instances and each one corresponds to a patient. This type of data is anonymized because they are sensitive and private data. for creating a model upon the breast cancer disease.

2. Dataset Definition

The dataset used in this paper was created by Dr. William H. Wolberg, physician at the University of Wisconsin Hospital at Madison, Wisconsin, USA and used for training, validation and testing. The dataset contains 569 instances (212 – Malignant, 357 – Benign) with 32 attributes (ID, diagnosis (B/M), 30 real-valued input features). Features are computed from a digitized image of a fine needle aspirate (FNA) of a breast mass. Computed features describe the following characteristics of the cell nuclei present in the image:

1. radius (mean of distances from center to points on the perimeter)
2. texture (standard deviation of gray-scale values)
3. perimeter
4. area
5. smoothness (local variation in radius lengths)
6. compactness (perimeter^2 / area - 1.0)
7. concavity (severity of concave portions of the contour)
8. concave points (number of concave portions of the contour)
9. symmetry
10. fractal dimension (“coastline approximation” - 1)

The mean, standard error, and “worst” or largest (mean of the three largest values) of these features were computed for each image, resulting in 30 features.

3. Pre-Processing

We are using Jupyter notebook to work on this dataset. We will first go with importing the necessary libraries and import our dataset wdbc.csv to Jupyter notebook. We can visualize the data set using the pandas’ head and tail method. Visualization of data is an imperative aspect of data science. It helps to understand data and also to explain the data to another person.

	1	2	3	4	5	6	7	8	9	10	...	22	23	24	25	26	27	28	29	30	3
0	1	17.99	10.38	122.8	1001.0	0.11840	0.27760	0.3001	0.14710	0.2419	...	25.38	17.33	184.6	2019.0	0.1622	0.6656	0.7119	0.2654	0.4601	0.1189
1	1	20.57	17.77	132.9	1326.0	0.08474	0.07864	0.0869	0.07017	0.1812	...	24.99	23.41	158.8	1956.0	0.1238	0.1866	0.2416	0.1860	0.2750	0.0890
2	1	19.69	21.25	130.0	1203.0	0.10960	0.15990	0.1974	0.12790	0.2069	...	23.57	25.53	152.5	1709.0	0.1444	0.4245	0.4504	0.2430	0.3613	0.0875

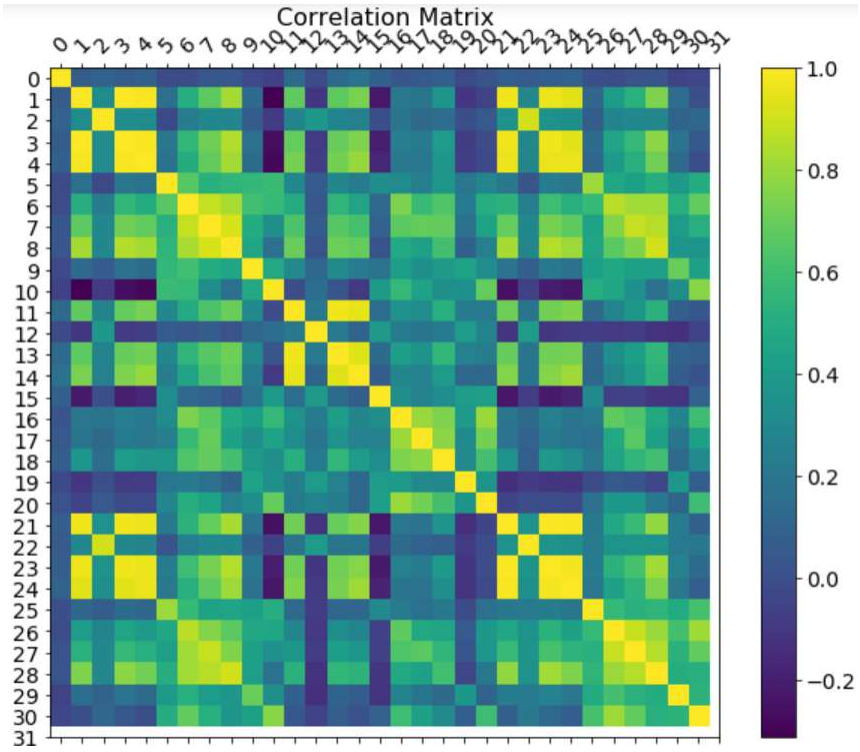
3 rows × 31 columns

We checked the dimensions of the data set using the panda dataset ‘shape’ attribute and here the dimension is 569 X 32

We also checked for any missing or null data points of the data set.

```
0    0.0
1    0.0
2    0.0
3    0.0
4    0.0
dtype: float64
```

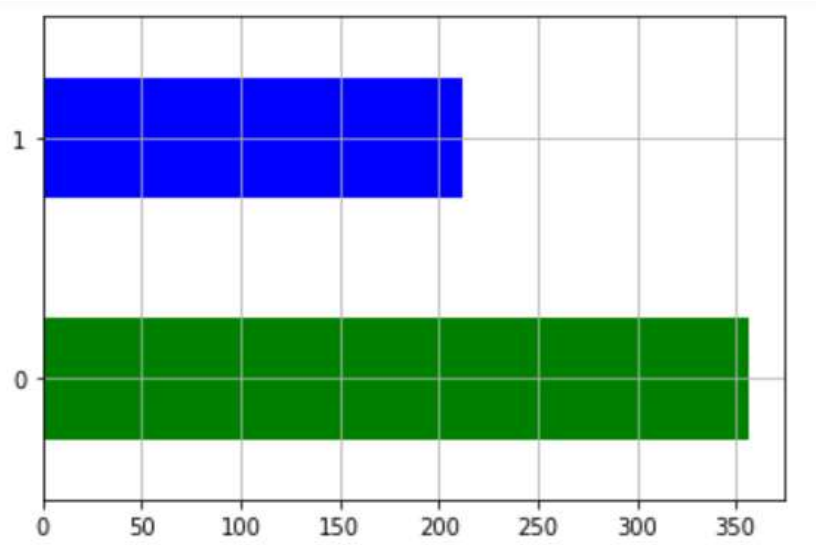
By performing Visual Exploratory Analysis, the Correlation Matrix is as follows:



As we are not using the Id column to develop the model, we are dropping it.

'Column 2' of the dataset (Diagnosis((B/M)) is indicating if an FNA cell is Benign (class = 0) or Malignant (class = 1) and we are going to predict this column only. As it contains categorical data i.e., variables that contain label values rather than numeric values, we are converting it as Malignant: 1 and Benign: 0

We can identify that out of the 569 persons, 357 are labeled as B (benign) and 212 as M (malignant).



We have partitioned the dataset into training data, test data and validation data. The training data contains 80% data of the dataset and test data contains 20% of the dataset and validation data contains 10% of the test data. The training set contains a known output and the model learns on this data to be generalized to other data later on. We have the test dataset to test our model's prediction on this subset and the introduction of the validation dataset allows us to evaluate the model on different data than it was trained on and select the best model architecture, while still holding out a subset of the data for the final evaluation at the end of our model development.

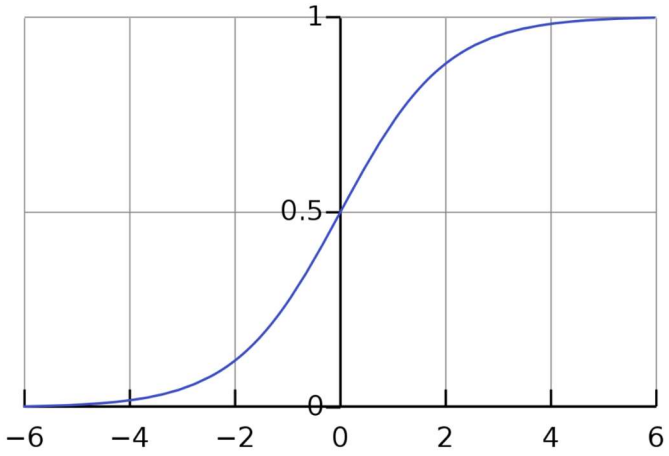
Next, we normalize the data to change the values of numeric columns in the dataset to a common scale, without distorting differences in the ranges of values.

4. Model Architecture:

Logistic Regression: Logistic Regression is a Machine Learning algorithm which is used for the classification problems, it is a predictive analysis algorithm and based on the concept of probability.

We can call a Logistic Regression a Linear Regression model, but the Logistic Regression uses a more complex cost function, this cost function can be defined as the '**Sigmoid function**' or also known as the 'logistic function' instead of a linear function.

Sigmoid Function (Logistic Function): Logistic regression algorithm also uses a linear equation with independent predictors to predict a value. The predicted value can be anywhere between negative infinity to positive infinity. We need the output of the algorithm to be class variable, i.e. 0-no, 1-yes. Therefore, we are squashing the output of the linear equation into a range of [0,1]. To squash the predicted value between 0 and 1, we use the sigmoid function.



Sigmoid Function Graph

The linear equation for the above curve can be represented as:

$$z = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_3 + \theta_4 x_4 + \ldots\ldots\ldots$$

$$z = (\theta^T)x + b$$

$$a = \sigma(z) \qquad \text{[where Sigmoid Function(} \sigma) = \frac{1}{1+e^{-z}}]$$

As shown in the above graph we have chosen the threshold as 0.5, if the prediction function (a) returned a value greater or equal to 0.5 then we would classify this observation as '**Malignant**'. If our prediction returned a value smaller than 0.5 then we would classify the observation as **'Benign'**

The cost function for logistic regression looks like

$$\boldsymbol{cost\ function(J\theta) = -\frac{1}{m}\sum_{i=1}^{m} y log\sigma(z) + (1-y) log(1-\sigma(z))}$$

Now to reduce the cost value the idea of Gradient Descent has been introduced. The main goal of Gradient Descent is to minimize the cost value**.** i.e. min $(\boldsymbol{J\theta})$.

Now To minimize the cost function we must run the gradient descent function on each parameter:

repeat until convergence {

$$b := b - \alpha \Delta b$$

$$w := w - \alpha \Delta w$$

}

[Where $(\alpha)$is the learning rate]

5. Results:

To correctly analyze the results, it is important to keep in mind that for this application of machine learning, having an accurate classifier is as important as having a low rate of false-negative when classifying a malignant lump, because each instance miss classified as a benign lump can delay the correct diagnosis and turn the treatment even more difficult.

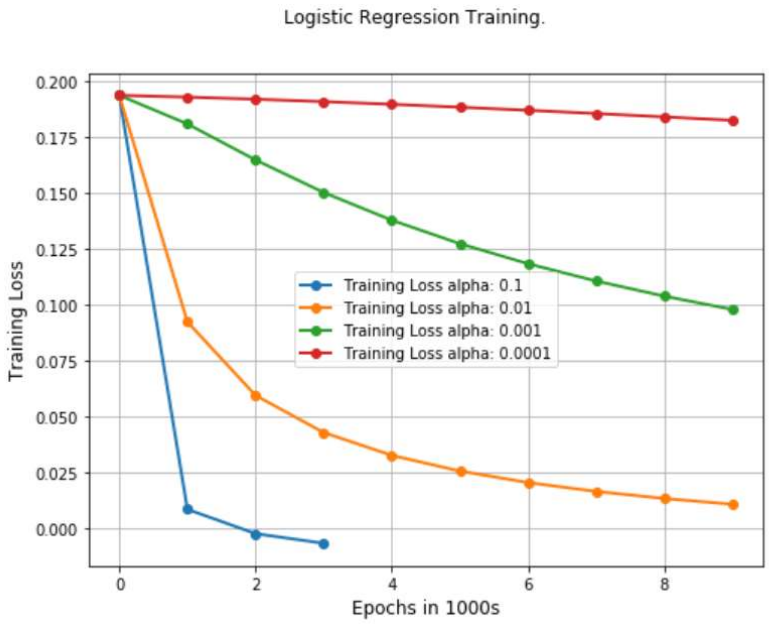
hyperparameter tuning: In machine learning, hyperparameter optimization or tuning is the problem of choosing a set of optimal hyperparameters for a learning algorithm. A hyperparameter is a parameter whose value is used to control the learning process. By contrast, the values of other parameters (typically node weights) are learned.

The same kind of machine learning model can require different constraints, weights or learning rates to generalize different data patterns. These measures are called hyperparameters and have to be tuned so that the model can optimally solve the machine learning problem. Hyperparameter optimization finds a tuple of hyperparameters that yields an optimal model which minimizes a predefined loss function on given independent data. The objective function takes a tuple of hyperparameters and returns the associated loss. Cross-validation is often used to estimate this generalization performance.

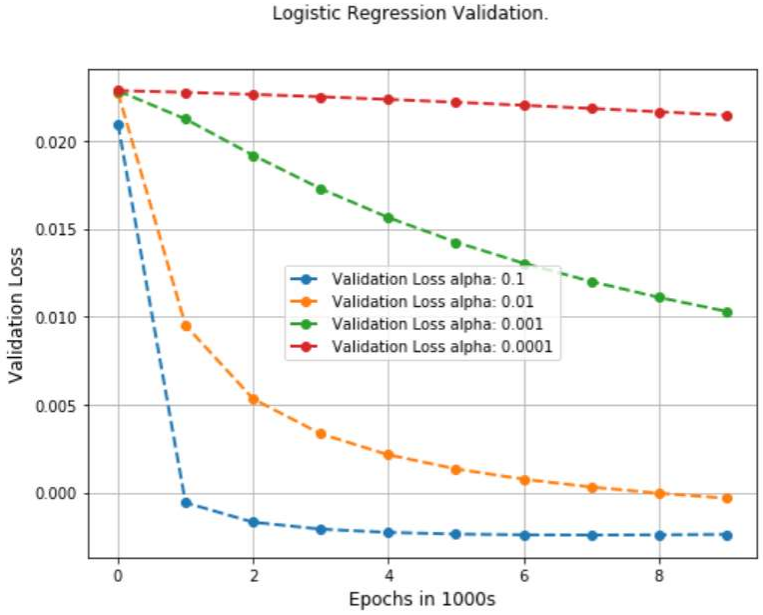
Here, we have considered one hyperparameter as learning rate with different values as 0.1, 0.01, 0.001, 0.0001.

The value of w and b is being updated iteratively with respect to the Training and Validation data to estimate the best parameter for accuracy.

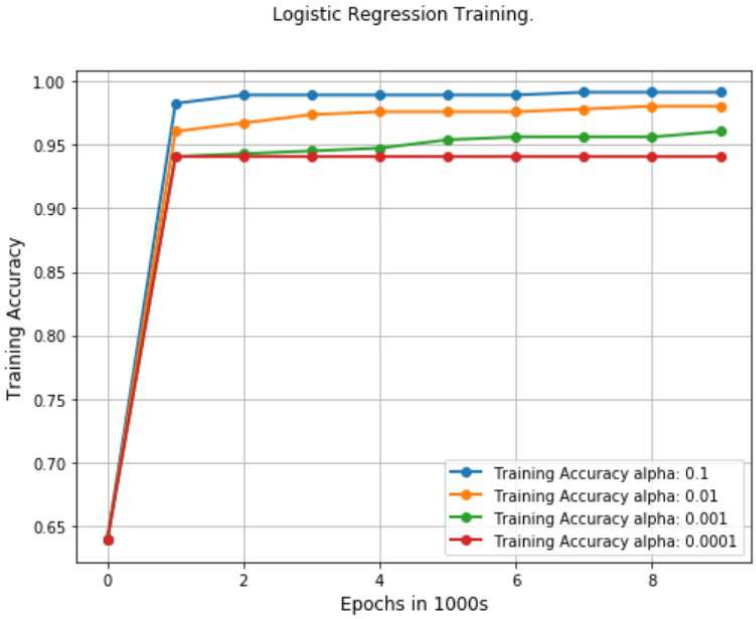
We have recorded the training loss for different learning rate as follows:



We have recorded the Validation loss for different learning rate as follows:



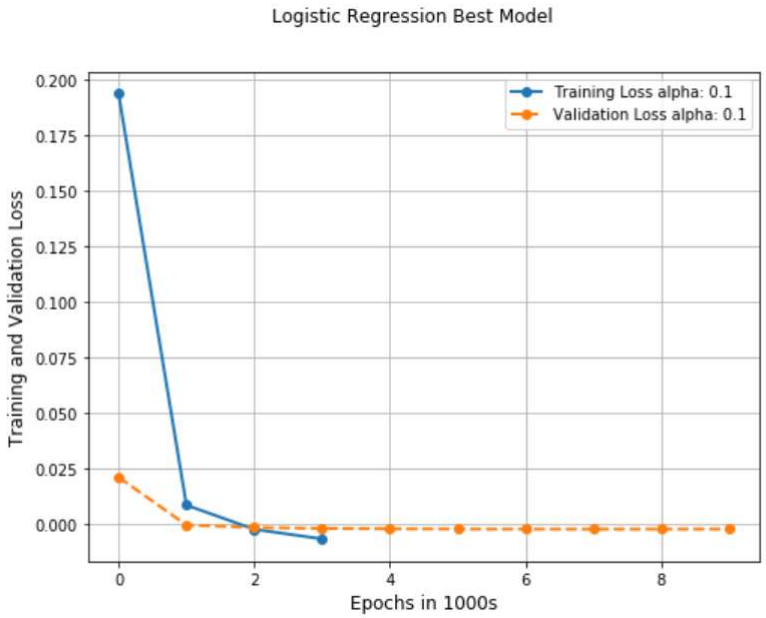
We have recorded the Training Accuracy for different learning rate as follows:



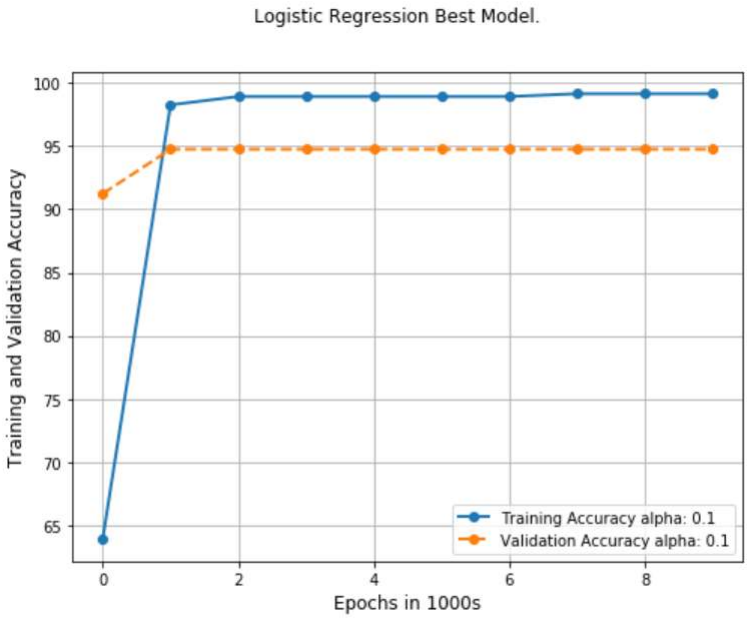
We have recorded the Validation Accuracy for different learning rate as follows:

As per our estimation, the best learning rate parameters for the model is 0.1 which gives us a training accuracy of 99.12%. Hence, we can keep the model with the best hyperparameters.

The Training and Validation Loss plot for best parameter (i.e. $\alpha = 0.1$):



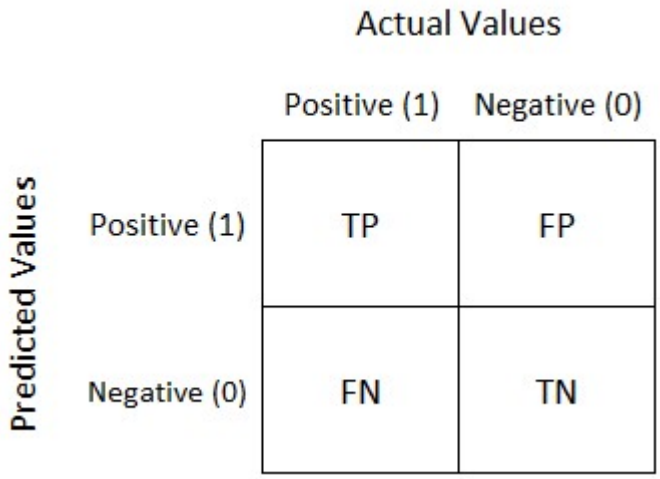
The Training and Validation Accuracy plot for best parameter (i.e. $\alpha = 0.1$):



We now fix the learning rate α and model parameter and test our model's performance on the testing set which gives us an accuracy of 94.74%

	Training Data	Test Data
Accuracy	99.12%	94.74%

Confusion Matrix: To examine the accuracy of test dataset, here we have introduced the concept of confusion matrix method of metrics class. Confusion Matrix is a performance measurement for machine learning classification problem where output can be two or more classes. It is a table with 4 different combinations of predicted and actual values.



True Positive (TP): We predicted positive and it's true.
True Negative (TN): We predicted negative and it's true.
False Positive (FP): You predicted positive and it's false.
False Negative (FN): You predicted negative and it's false

In our case the confusion matrix is as follows:

	TP	**FP**
FN	**30**	**0**
TN	**3**	**24**

Using Confusion Matrix, we can calculate Recall, Precision and Accuracy.

Accuracy is defined as the ratio of the number of correct predictions to the total number of input samples.

Hence, Accuracy = $\frac{TP+TN}{TP+TN+FP+FN}$ = 54/57 = 94.74%

Precision is defined as the ratio of the number of correct positive results to the number of positive results predicted by the classifier

Hence, Precision $= \frac{TP}{TP + FP} = 30/30 = 100\%$

Recall is defined as the ratio of number of correct positive results to the number of all relevant samples (all samples that should have been identified as positive).

Hence, Recall $= \frac{TP}{TP + FN} = 30/33 = 90.91\%$

At a glance the final results are as follows:

Accuracy	Precision	Recall
94.74%	100%	90.91%

6. Conclusion

This paper presents the development and the simulation of a machine learning model of two class problem created with logistic regression to predict breast cancer. The presented ML algorithm exhibited high performance on the two-class problem of breast cancer, i.e. determining whether benign tumor or malignant tumor. Consequently, the statistical measures on the classification problem were also satisfactory.

Acknowledgments

We are extremely grateful to Professor Sargur Srihari and Mihir Chauhan for teaching all the necessary concepts related to Logistic Regression and helping in this project throughout.

References

[1] Breast Cancer: Current Research: https://www.omicsonline.org/open-access/breast-cancer-prediction-by-logistic-regression-with-cuda-parallel-programming-support-bccr-1000111.php?aid=77421
[2] Logistic Regression: https://sebastianraschka.com/faq/docs/logistic-why-sigmoid.html
[3] Analysis of the Wisconsin Breast Cancer Dataset and Machine Learning for Breast Cancer Detection: https://www.researchgate.net/publication/311950799_Analysis_of_the_Wisconsin_Breast_Cancer_Dataset_and_Machine_Learning_for_Breast_Cancer_Detection
[4] Logistic Regression Model Tuning with scikit-learn: https://towardsdatascience.com/logistic-regression-model-tuning-with-scikit-learn-part-1-425142e01af5