Application of machine learning algorithms to detecting breast cancer using the breast cancer dataset

Dhanasekaran, Roshan
Faculty of Engineering, Environment and Computing
Msc Data Science and Computational Intelligence
Coventry University
Coventry, United Kingdom
dhanasekar@coventry.ac.uk

Yakubu, Suleiman Adamu
Faculty of Engineering, Environment and Computing
Msc Data Science and Computational Intelligence
Coventry University
Coventry, United Kingdom
yakubus6@coventry.ac.uk

***Abstract*- This paper is meant to show how machine learning techniques can be used to detect breast cancer. The paper follows some analysis done on a dataset. The analysis includes exploratory data analysis, preprocessing, feature selection, algorithm selection, and finally, algorithm evaluation using selected metrics. The algorithms implemented are: Support Vector Machines, Stochastic Gradient Descent, Random Forest, and Multilayer Perceptron. MLP had the best performance.**

I. INTRODUCTION

Cancer is an insidious condition that causes some cells in the body to grow abnormally and uncontrollably. These cells grow to form a lump, or tumor. However, not all tumors that grow from abnormal cells are cancerous. Cancerous tumors, also called malignant tumors are the cells that can grow and multiply to other parts of the body other than the part that they first developed in. The other types of tumors, called non-cancerous or benign tumors do not spread, they may still cause harm to health when they grow and press on nearby tissues or organs. Thus, breast cancer is a malignant tumor that develops in the breast. It is the second leading cause of women's death, the first being lung cancer and constitutes 25% of all female cancer cases(J Sivapriya et al., 2019). There are currently no preventive measures in medicine and so early detection of the tumor is vital. It is important to note that most lumps or tumors discovered in the breast are benign.

Breast cancer is usually detected after a lump or tumor is noticed. Images of the breast tissue are taken using a procedure called a mammogram. These images can show doctors if there is a tumor and it is benign. If there is a suspicion of cancer however, a sample of the tumor is taken for a test called a biopsy. This test can then

confirm if the tumor is malignant. Machine learning can be used to analyze the results of a biopsy and tell, with better accuracy than a human expert, if the tumor is malignant.

II. THE DATASET AND PROBLEM DEFINITION

The dataset, called the Breast Cancer Wisconsin (Diagnostic) Dataset, is sourced from the UCI Machine Learning Repositories and can also be found on Kaggle. It contains 569 instances with 33 attributes with the label being the diagnosis attribute. The diagnosis is a categorical entity either “B” for benign or “M” for malignant. The dataset has no missing values and contains readings that describe the characteristics of a cell nucleus present in a digitized image taken during a biopsy process known as Fine Needle Aspiration (FNA). Ten (10) real valued attributes are computed for each cell. They include:

1. Radius: This is the distance from the nucleus to the cell wall. Thus, the larger the radius, the larger the cell
2. Texture: The standard deviation of
3. Perimeter: The total length of the outer cell wall
4. Area: The area of the nucleus
5. Smoothness: This is the difference between the length of one radial line and the mean length of the two radial lines surrounding it. Smaller numbers means smoother contours.
6. Compactness:
7. Concavity:
8. Concave points:
9. Symmetry:
10. Fractal dimensions:

In the dataset, the mean, standard error, and worst (or largest mean) values of these attributes are recorded. This brings the total number of our attributes to 30. The attributes are then rounded out by the following:

1. ID number: A unique number to identify the sample taken.
2. Diagnosis (B = benign, M = malignant)

The goal of this paper is to predict the nature of a tumor (malignant, or benign) and thus diagnose if a patient has breast cancer or not given the information that is defined in the dataset

III. DATA PREPARATION

This section covers any Exploratory Data Analysis (EDA) and preprocessing done to the data to make it suitable for the machine learning algorithms to consume. Some of the issues treated in this stage include class imbalance, feature selection and dimensionality reduction, as well as categorical data encoding. Some basic data cleaning was carried out as well. This entailed dropping a column in the dataset called “Unnamed:32” that contained only null values. The correlation matrix was also computed and a correlation heatmap drawn. The correlation heatmap helps to see the relation between attributes. This can be used to drop out attributes that have a high correlation as it might mean that they contain redundant information.

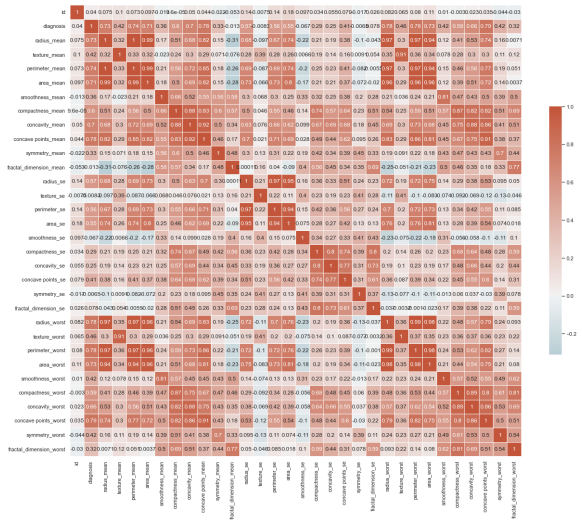


Fig 1: Correlation heatmap of the dataset.

A. Class Imbalance

An imbalance in the classes simply means the number of examples is skewed in favor of one class. This can lead to machine learning algorithms underperforming (Johnson, J., & Khoshgoftaar, T. 2019). To remedy this, there are a number of acceptable approaches, these include; assigning a high cost to misclassification of the minority class, or sampling methods. In this paper, sampling methods were used, Synthetic Minority Oversampling Technique (SMOTE) in particular. SMOTE works by increasing the samples of the minority class examples to a level that the number of examples of both classes are similar. It does this by selecting examples that are near to each other in the feature space, drawing a line along the examples, and then creating a new sample at some random point along the line.

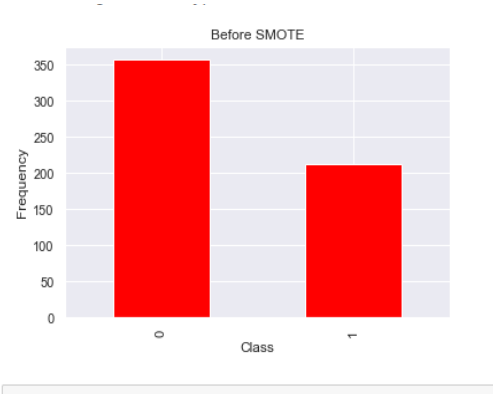


Figure 2: The distribution of classes in the dataset before SMOTE

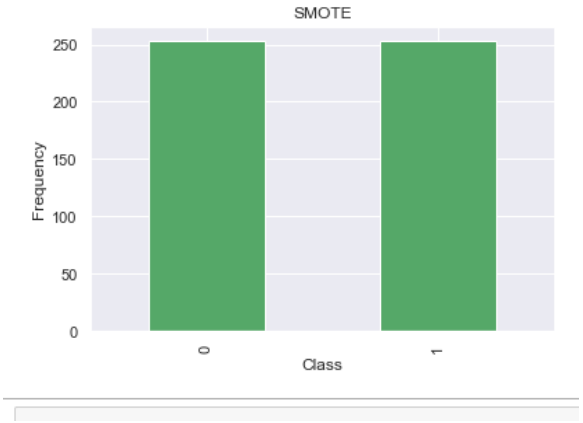


Figure 3: The distribution of classes in the dataset after SMOTE

B. Categorical Data Encoding

Most machine learning algorithms cannot interpret categorical data. For this reason, they are usually encoded into binary and passed to the algorithm. This is usually done by means of an encoder. However, I used Panda's map function to map the categorical column to an int (0 or 1). The column in question was the label column (diagnosis).

C. Dimensionality Reduction

Dimensionality reduction, as the name suggests, is a means by which we reduce the complexity of a dataset by reducing its

dimensions. We do this by eventually reducing the number of attributes in the dataset. The question remains, which attributes are safe to remove without losing too much data? This question is answered by Principal Component Analysis (PCA). What PCA produces are the Principal Components of the datasets. These are new variables that are linear combinations of the original ones and are uncorrelated. These variables are set up in such a way that the most information is compressed into the first ones. PCA tries to put the maximum amount of information into the first variable, then the maximum remaining information into the second and so on. What we then have is a feature vector that holds as much information as possible in the first variables(Zakaria Jaadi, 2021). PCA was done on this dataset using the pca module of the sklearn library.

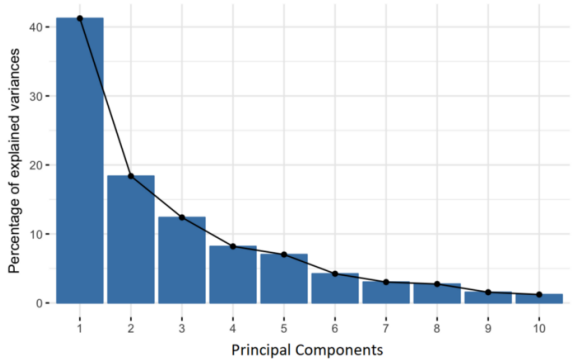


Figure 4: PCA showing the principal components (Zakaria Jaadi, 2021)

D. Feature Selection

Sometimes datasets contain redundant or useless information for the tasks at hand. Some attributes do not contribute significantly to the required classification and can be done away with. These attributes can then be eliminated from the dataset to improve performance. Recursive Feature Elimination (RFE) was done with the dataset to select the optimum number of features for training. RFE efficiently selects the weakest attributes in the dataset and recursively eliminates them till a required number of attributes is reached (YellowBrick, 2019). RFE was used from the python sklearn library.

IV. CLASSIFICATION ALGORITHMS USED

A. Support Vector Machine

Support Vector Machines (SVM) is a supervised learning algorithm that works well for classification problems, although it can be used with regression problems as well. It works by mapping training examples to points in space that maximize the width of the gap between the classes being predicted. Given a new data point, it then maps that data point into the same space and predicts the class of the new data point based on which part of the gap it falls. SVM can efficiently perform linear and non-linear classifications using kernels (Bonaccorso, G. 2017). SVM was implemented in this project the sklearn library implementation.

B. Random Forest Classifier

Random Forest Classifier is an ensemble machine learning algorithm. This means that it is a combination of some other simpler algorithms to create better results. In this case, Random Forest is a collection of decision trees. It is widely used for both classification and regression problems. It

works by constructing several decision trees for random samples of the dataset and taking their majority vote, in the case of a classification problem (Lutins, E., 2017). For regression, the Random Forest algorithm takes the average of the decision trees. The Random Forest algorithm was used in this project via an sklearn implementation.

C. Stochastic Gradient Descent

Understanding Stochastic Gradient Descent requires one to understand gradient descent. Gradient descent is a common way used to optimize the objective function of an optimization problem model. It is used to minimize the error in fitting the line to the data by taking the gradient of the error functions against the weight and stepping in the negative direction of the gradient until the error function is reduced to its minimum (Shalev-Shwartz and Ben-David 2014).

D. Multilayer Perceptron

Multilayer Perceptrons are feed forward neural networks that generate a set of outputs given a set of inputs. What they do in effect is approximate a function e.g f *. A multilayer perceptron consists of several layers of nodes that are set up like a connected graph between the inputs and the outputs (Ian Goodfellow et al., 2016). The Multilayer Perceptron Object () of the sklearn library was used in this project

V. RESULTS

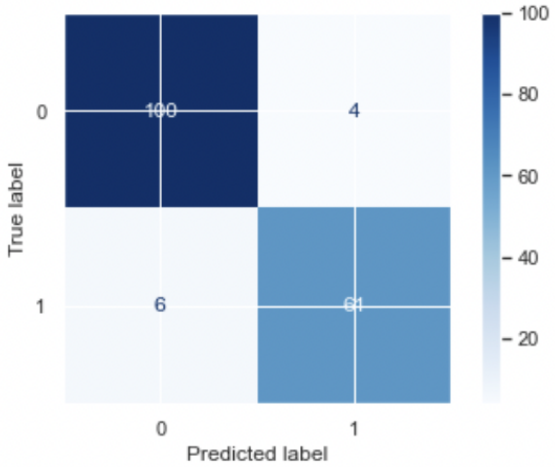
The algorithms and results obtained thereof were gotten on a Lenovo Ideapad 320 with 4GB of RAM. The code was run using Jupyter Notebook. The models were initially trained without preprocessing the data (i.e without PCA, RFE, and standardization). The models were then evaluated using various metrics. The data was then preprocessed, the models trained again, and evaluated and all the results noted. This was done in a pipeline. The metrics used for evaluation are as follows:

A. Accuracy
B. Precision
C. Recall
D. F1 score
E. Macro average
F. Weighted average

The results are as shown in the following sections:

A. Support Vector Machine

Using the Support Vector Classifier available from sklearn, the below results were obtained:



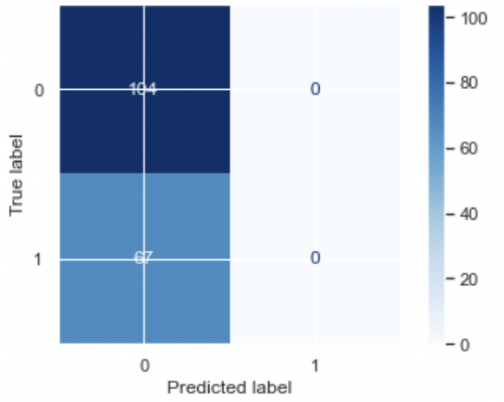


Fig 5: Confusion matrices for SVM

Support Vector Machine		
Algorithm parameters		
C	2	2
Kernel	linear	sigmoid
Gamma	auto	auto
Results		
	Before	After
No of samples	357	252
Accuracy	0.9532	0.9766
precision	0.95	0.97
Macro avg	0.95	0.97
Weighted avg	0.95	0.96
F1 score	0.95	0.97
Support	171	171
Confusion matrix		
No of samples	357	252
True Positive	61	104
True Negative	100	0
False Positive	6	0
False Negative	4	67

Table 1: Results and metrics for SVM

B. Random Forest

The Random Forest classifier from the sklearn.ensemble module produced the results below

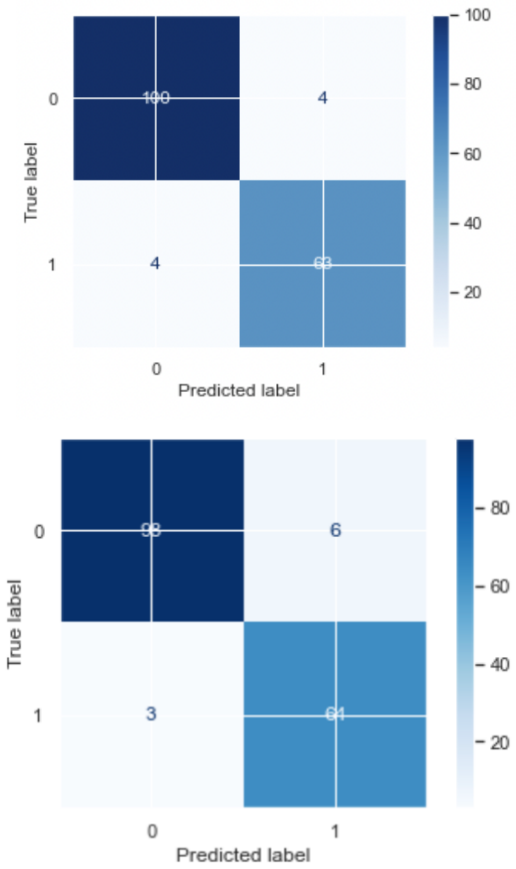


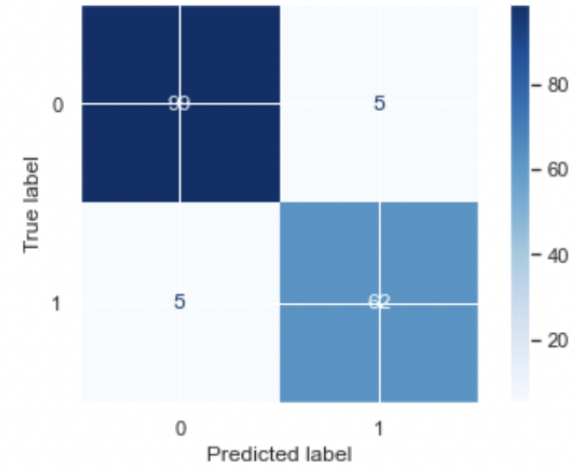
Fig 6: Confusion matrix for Random Forest

Random Forest		
Algorithm parameters		
Max Depth	2	4
Random state	0	2
Results		
	Before	After
No of samples	357	252
Accuracy	0.941	0.9473
Precision	0.95	0.96
Macro avg	0.94	0.97
Weighted avg	0.94	0.97
F1 score	0.94	0.95
Support	171	171
Confusion matrix		
No of samples	357	252
True Positive	63	61
True Negative	100	98
False Positive	4	3
False Negative	4	6

Table 2: Results and metrics for Random Forest

C. Stochastic Gradient Descent

The stochastic gradient descent algorithm used via the sklearn python library produced the following results



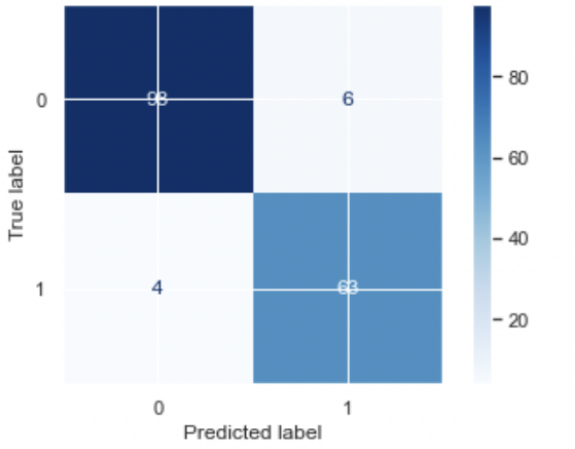


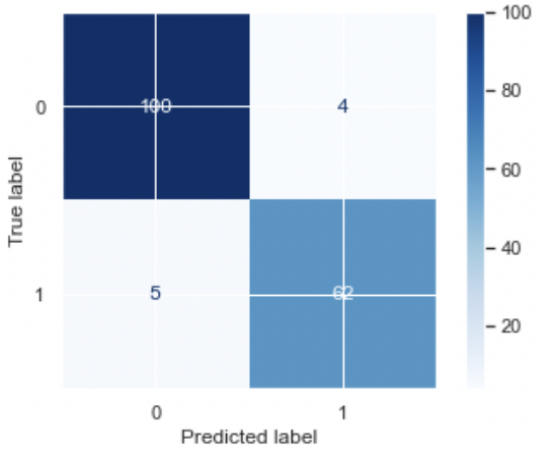
Fig 7: Confusion matrices for SGD

Stochastic Gradient Descent		
Algorithm parameters		
Loss	hinge	hinge
Penalty	12	12
Max iter	5	5
Results		
	Before	After
No of samples	357	252
Accuracy	0.877	0.941
Precision	0.94	0.96
Macro avg	0.87	0.94
Weighted avg	0.88	0.92
F1 score	0.88	0.92
Support	171	171
Confusion matrix		
No of samples	357	252
True Positive	62	63
True Negative	99	98
False Positive	5	4
False Negative	5	6

Table 1: Results and metrics for SGD

D. Multilayer Perceptron

The multilayer perceptron used in the project produced the following:



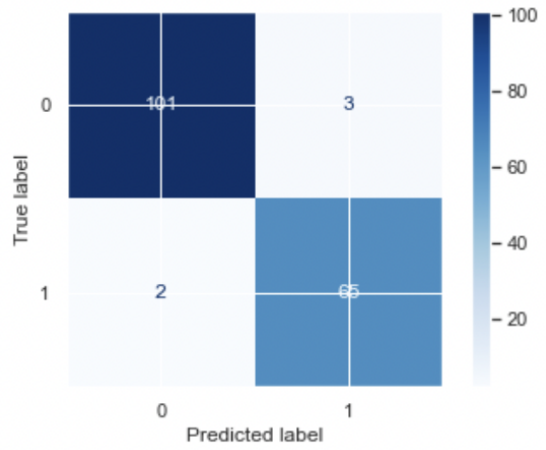


Fig 8: Confusion matrices for MLP

Multilayer perception		
Algorithm parameters		
Solver	lbfgs	lbfgs
alpha	1e-5	1e-5
Hidden layer	(5,2)	(5,2)
Random state	1	1
Results		
	Before	After
No of samples	357	252
Accuracy	0.947	0.970
Precision	0.95	0.96
Macro avg	0.95	0.96
Weighted avg	095	0.96
F1 score	0.94	0.97
Support	171	171
Confusion matrix		
No of samples	357	252
True Positive	62	65
True Negative	100	101
False Positive	5	2
False Negative	4	3

Table 1: Results and metrics for MLP

VI. DISCUSSION

In my case, after training, the multilayer perceptron performed better than the other algorithms, although that was not by a large margin. The SVM and even Random Forest had admirable results as well. Because of time, and due to the constraints of my machine, more hyperparameter tuning could not be done and as such further performance gains could not be explored. The preprocessing did increase the performance of the models, although not by a large amount. This could be due to the fact that the data was relatively clean, and most of the readings came from similar domains and so had similar scales. Feature selection and SMOTE did have a significant effect on making the models less complex and resource intensive.

VII. REFERENCES

Bonaccorso, G. (2017). Machine Learning Algorithms [online]. Birmingham: Packt Publishing, Limited. Available from <http://ebookcentral.proquest.com/lib/coventry/detail.action?docID=4926962>

Ian Goodfellow, Yoshua Bengio, & Aaron Courville. (2016). *Deep learning* (1st ed.). MIT Press.

J Sivapriya, V Aravind Kumar, S Siddarth Sai, & S Sriram. (2019). Breast cancer prediction using machine learning. *International Journal of Recent Technology and Engineering, 8*(4), 4879-4881. https://doi.org/10.35940/ijrte.D8292.118419

Johnson, J., & Khoshgoftaar, T. (2019). Survey on deep learning with class

imbalance. ***Journal Of Big Data*****, *6*(1). https://doi.org/10.1186/s40537-019-0192-5**

Lutins, E., (2017). Ensemble Methods in Machine Learning: What are They and Why Use Them?. [Online]Available at: https://towardsdatascience.com/ensemble-

Methods-in-machine-learning-what-are-they-and-why-use-them-68ec3f9fef5f [Accessed 19 april 2022].

Shalev-Shwartz, S. and Ben-David, S. (2014) 'Stochastic Gradient Descent'. in Understanding Machine Learning: From Theory to Algorithms. ed. by Anon Cambridge: Cambridge University Press, 150-166

YellowBrick. (2019). *Recursive feature elimination.* https://www.scikit-yb.org/en/latest/api/model_selection/rfecv.html

Zakaria Jaadi. (2021). *A step-by-step explanation of principal component analysis (PCA).* https://builtin.com/data-science/step-step-explanation-principal-component-analysis

IV. APPENDIX

Dataset Link : https://www.kaggle.com/datasets/uciml/breast-cancer-wisconsin-data

All code used in this project is available from the link below: https://github.com/suleiManiac/MLCU