Technical Report

By Hongnan Gao

A technical report on detecting breast cancer.

1 Introduction and Use Case

We are presented with a dataset that is taken from a biopsy procedure called **Fine Needle Aspiration (FNA)** performed on the breast. Studies show that FNA has a relatively **low recall rate** at 86.3%, consequently, the **cost** is the delay in treatment of patients with malignancy.

The use case in this project is to design a machine learning classifier that has **low false negatives**, which can correctly classify malignant cases relatively well. Furthermore, this can serve as a second opinion for doctors/pathologists to make more informed decision when examining tumor cells.

2 Problem Definition and Algorithm

2.1 Dataset Description

The dataset, Breast Cancer Wisconsin (Diagnostic), is taken from UCI repository. It has 569 unique observations with slight imbalance, in which $\sim 40\%$ are malignant. There are 10 base features[1] for each observation; the mean, standard error and the worst[2] measurements are computed for each base feature, resulting in a total of 30 features. The target variable is diagnosis, taking on either **malignant** or **benign**.

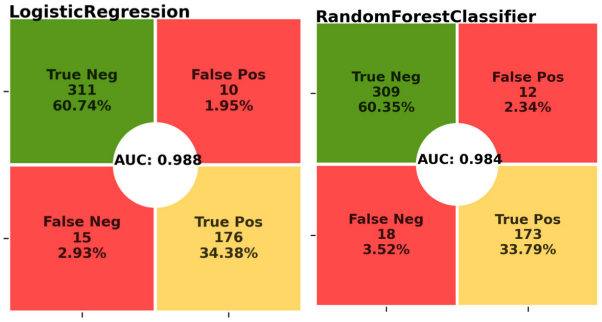
2.2 Problem Statement

The task at hand is to develop a **machine learning classifier** that can classify whether a tumor is benign or malignant, while placing more emphasis on **correctly classifying** the malignant tumors.

2.3 Choice of Classifier: Logistic Regression

There is a tradeoff between a classifer's **flexibility** and **interpretability**. After running multiple models **(see Figure 1. not inclusive of all models trained)**, and taking into consideration the No Free Lunch Theorem and Occam's Razor, **Logistic Regression (LR)** turns out to be a reasonable choice for this task.

Figure 1: Out of fold confusion matrix



LR is both **interpretable** and outputs **well calibrated probabilities**. In addition, its performance is on par with more complicated models such as Random Forest Classifier in this setting. A slightly more formal treatment of Logistic Regression can be found in Appendix.

3 Choice of Performance Measure: ROC and Brier Score

We will use metrics that can help us **reduce false negatives**, and at the same time, outputs **meaningful predictions**. In order to achieve both, we will use **Receiver operating characteristic (ROC)** as the primary metric for the model to maximize, and **Brier Score**, a **proper scoring rule** to measure the performance of our probabilistic predictions. We will go into some details in the next two subsections to justify our choice.

3.1 Receiver operating characteristic

Definition 1. *The ROC curve plots the True Positive Rate on the y-axis and False Positive Rate on the x-axis parametrized by a threshold vector $\vec{t}$.*

The choice of ROC over other metrics such as Accuracy is detailed in the notebook. **We also established we want to reduce false negatives (FN), since misclassifying a positive patient as benign is way more costly than misclassifying a negative patient.** One can choose to minimize Recall in order to reduce FN, but this is **less than ideal during training** because it is a thresholded metric, and does not provide at which threshold the recall is at minimum. We take a look at how ROC solves the aforementioned issues:

3.1.1 Threshold Invariant

ROC computes the pair $TPR \times FPR$ over all thresholds t, making the Area Under ROC (AUROC) threshold invariant, allowing us to look at the model's performance over all thresholds. We note that ROC may not be that reliable in the case of very imbalanced datasets where majority is in the negative class, as $FPR = \dfrac{FP}{FP+TN}$ may seem deceptively low as denominator may be made small by the sheer amount of TN, in this case, we may also look at the Precision-Recall curve.

3.1.2 Scale Invariant

This is not the desired property that we need, as this means that the ROC is non-proper in scoring, it can take in non-calibrated scores and still perform relatively well. The following code shows that the ROC scores a full score even though the probabilities are ill-conditioned, furthermore, the AUROC score remains the same with the same preserved ranking, regardless of scale. This can lead to overly-optimistic results, with that, we introduce Brier Score as a complement.

```
y1 = [1,0,1,0]
y2 = [0.52,0.51,0.52,0.51]
y3 = [52,51,52,51]
uncalibrated_roc = roc(y1,y2) == roc(y1,y3)
print(f"{uncalibrated_roc}") -> 1.0
```

Listing 1: Non-Proper Property

[1] radius, texture, perimeter, area, smoothness, compactness, concavity, concave points, symmetry, fractal dimension

[2] worst is defined as the three largest values

3.2 Brier Score

Definition 2. *Brier Score computes the squared difference between the probability of a prediction and its actual outcome.*

Brier Score is a strictly proper scoring rule[3] and the lower the Brier Score, the better the predictions are calibrated. We can first compute the AUROC score of the model, and compute Brier Score to give us how well calibrated (confident) the predictions are.

4 Methodology

4.1 Data Cleaning, EDA and Feature Selection

Inspection of data and analyzing data through extensive EDA is done to ensure that we make sound decisions during modelling. From EDA, we discovered that there is **multicollinearity** amongst predictors, furthermore, predictors are not on the same scale; these prompted us to create a preprocessing pipeline to standardize the predictors and also recursively remove multicollinear predictors using Variance Inflation Factor.

4.2 Cross-Validation Strategy

We first shuffle and split the whole dataset into X_{train} and X_{test} with a ratio of $9:1$, stratified by the target variable to ensure equal representation in the splits. We will use X_{train} for cross-validation and model selection, while keeping X_{test} untouched until the last stage, where **we perform an out-of-sample evaluation on this "unseen" test set for an estimation of generalization error.**

There is slight imbalance in the dataset, and since the observations are indepedent and unique, we can use **StratifiedKFold** as our cross-validation strategy. We also note that we **carefully included two preprocessing steps in our cross-validation pipeline, this is to avoid data leakage.** The pseudo-code can be found in the Appendix.

4.3 Hyperparameter Tuning

We will use an old-fashioned way to search for hyperparameters, which is brute force Grid Search. The time complexity of Grid Search is high and if you have many hyperparameters to tune and further steps can be taken to better this process, for example, we can use Optuna for Bayesian Optimization.

5 Analysis of Results

5.1 Feature Importance

As shown in Table 1 (only select 2 features for reference), all else being equal, for every square unit increase in mean cell area, the odds of the tumor being malignant increases by a factor of ~ 4.19. The variation[4] of the characteristics of cells also are deemed important by the model, for example, **area se** played an important role in determining whether a cell is malignant; intuitively, if some cells are noticably larger than the rest (high standard error), then it is also a good indicator of malignancy.

Feature	Coefficient
area mean	1.43
area se	1.21
...	...

Table 1: Feature Importance.

One thing to note further is the seemingly contradictory dynamics for **fractal dimension**, there is a tug of war in terms of the coefficients of its mean versus the worst. Even during my EDA, the univariate plot of this feature distribution is overlapping, making it hard to differentiate. Taking all these into consideration, we may need to investigate this predictor further as a next step.

5.2 Threshold Optimization

Our narrative is to have a classifier that can identify malignant tumors correctly. In other words, the cost-benefit analysis tells us that we need to minimize false negatives, possibly at the expense of an increase in false positives. We looked at both the ROC and PR curve and deduced a suitable cutoff point, $t = 0.35$ based on the train results. The table below shows the result of the precision, recall, f1 and roc metrics when the threshold is 0.35.

Precision	Recall	f1	roc	brier
0.93	0.95	0.94	0.99	0.03

Table 2: Classification Report on Positive Class

5.3 Evaluate on Test Set

In the final phase, we will predict our model using our best classifier selected during the Grid Search Hyperparameter phase. This step is to **estimate our out of sample generalization error**.

Precision	Recall	f1	roc	brier
0.87	0.95	0.91	0.983	0.04

Table 3: Classification Report on Positive Class (Test)

Using the same threshold, the table above shows the results on the test set, and they are mostly similar, except for a decrease in precision, indicating the increase of false positives.

6 Next Steps

Once we find a stable and well performing classifier, we can spend more time on interpreting feature importance in greater details so as to be able to explain to our stakeholders on why certain features are deemed more important than the others by the classifier.

[3] Damage Caused by Classification Accuracy and Other Discontinuous Improper Accuracy Scoring Rules

[4] standard error