# Breast cancer classification using logistic regression

Reece Hill and Tom Aldridge

The development of machine learning techniques, alongside computing power, has progressed in recent decades. Artificial intelligence is now an area of research in most disciplines, particularly medicine. We explore the use of a simple, logistic regression model that is optimised to categorise breast tumours as either malignant or benign. The model is assessed on its accuracy, precision, sensitivity, specificity, and negative log likelihood. These metrics are used to evaluate the effect on predictions of increasing the size of a given training set, relative to the dataset on which it is tested. We also compare the performance of this model against that of a clinician, and we discuss the utility of machine learning, including its drawbacks, as a tool for medical diagnoses.

***Introduction.***– Examples of machine learning in medical diagnostics can be found for most test modalities: neurophysiological [1–3], imaging [4–6], and surgical robotics [7, 8]. This paper concerns the histopathological context, where data is obtained at a cellular level.

We task the model with correctly identifying malignant from benign breast cancer tumours. Breast cancer affects 1 in 1200 people in the UK [9], costing the NHS over £700 million per year [10]. By testing machine learning methods in oncology, we can explore ways in which the delivery of healthcare can be improved. After all, a successful model could prompt the automation of care pathways; thus, reducing the national cost per diagnosis. If impressive enough, artificial intelligence could successfully detect the presence of cancer in patients that would have otherwise gone unnoticed.

We split this report into sections. First, we begin with a summary of the data that was used to train and test the model. Then, we describe and justify the chosen model. This is followed by a description of our methods before an analysis and review of the model's results are made. We conclude in the traditional way with a discussion, where we briefly evaluate the real-world use of such a model.

***Data.***– The data we are using has been provided by the UCI Machine Learning Repository[11, 12]. Discarding the sample's number, the 699-row dataset is formed of 9 features and 1 target. These features are as follows: clump thickness, uniformity of cell size, uniformity of cell shape, marginal adhesion, single epithelial cell size, bare nuclei, bland chromatin, normal nucleoli and mitoses.

The respective grading scale for each feature histologically derived, using observations of percentage cover. For instance, a grade of 1 for "uniformity of cell size" implies that 100% of cells in the sample were completely uniform. The percentage of uniform cells decreases, such that grade 10 indicates that all cells are of different sizes. For cases where the characteristic is proportional to malignancy, such as single epithelial cell size, the rule is reversed, and the percentage cover increases with grade. Thus, we evolve a scale where 1 describes the most benign features, and 10 suggests poor inter-cellular differentiation that is indicative of malignancy.

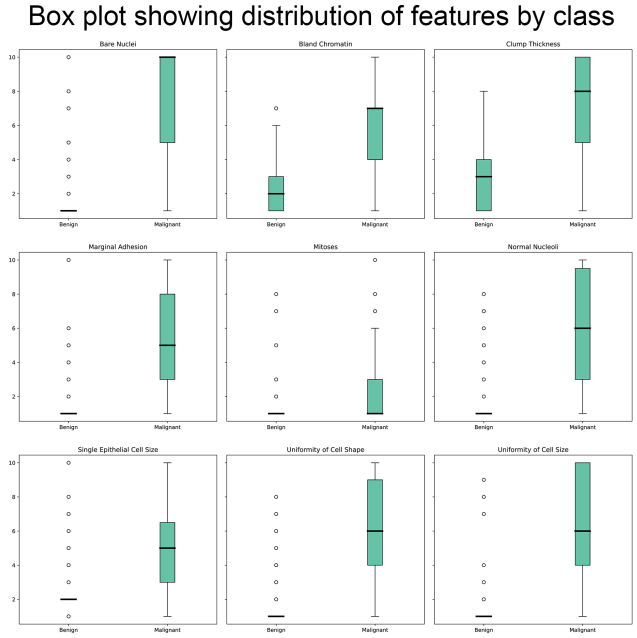


FIG. 1. **Box-plot for distribution of each feature's data** Showing data generated for Experiment 1; entries with missing data were removed. Each feature shows data grouped by class label (benign/malignant). It can be seen that high grades commonly correspond to malignant tumours, while benign tumours may be characterised by lower grades.

For completeness, we outline the target. This is a binary classification task and thus we consider a single target of two possibilities: benign or malignant. Labelled in the dataset as "class", the former is encoded by "2" and the latter, malignancy, by "4". For use in our model, we reassign these values (see section: **Method**) and assume certainty for any given label. We also append a vector of ones to the start of the data, forming the intercept vector.

***Model.***– To perform this task, we have chosen a logistic regression model. Logistic regression models are a favoured method for binary classification tasks due to their ease of use and thorough documentation. Logistic regression has also been used elsewhere in cancer diag-

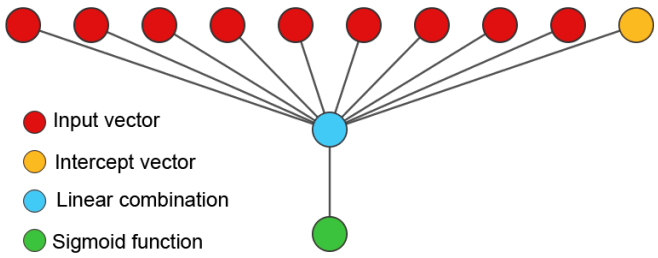


FIG. 2. **Map of the network.** Graphical representation of our model. The data is linearly combined, then passed through a sigmoid activation function in order to perform logistic regression.

nostics, such as [13, 14]. To perform logistic regression, we first construct a linear model,

$$\hat{\mathbf{Y}} = \boldsymbol{X}\boldsymbol{\beta}^* \tag{1}$$

where $\hat{\mathbf{Y}}$ is the vector of predicted values, $\mathbf{X}$ is the design matrix of features, and $\boldsymbol{\beta}^*$ is the optimal parameter defined by $\boldsymbol{\beta}^* = (\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{Y}$. We then pass this prediction through the sigmoid function, $\sigma(\boldsymbol{X}\boldsymbol{\beta}^*)$, to restrict the output to be between 0 and 1. This forms the conditional probability that the tumour is malignant given our model. Our model is visualised in Figure 2.

***Method.***– The model was coded using Python 3.7.12[15] with additional packages[16, 17]. Data was retrieved from a GitHub repository, then scaled by a factor of $10^5$ to mitigate rounding errors in Python and avoid singular matrices. We performed four experiments on the data, varying how missing data were managed. The first experiment had entries with missing data completely removed from the dataset so as to not affect the dataset's distribution. Accompanying this approach, we then trialled replacing data gaps with the respective feature's mean. Conversely, the latter two experiments attempted to show the effect of improper data manipulation. The third experiment substituted with the respective feature's minimum values, and the fourth used the maximum values.

Once complete, the dataset was randomly stratified by class to form mutually exclusive datasets for training and testing our model. Randomness was reproducible due to seeding and we performed stratification methodically. Data was separated into a training set and the test set, such that the intersection of these sets was empty. We examined the effect of the training set size on model performance by iterating through every possible size of the training set; starting from only one entry for training and the rest for testing, until all but one entries were used for training. Consequently, 244,650 data rows per seed were passed to the model.

In order to use the threshold of 50% probability as our classification threshold, we reassign the targets to be either -1 or 1 instead of 2 or 4. This means that when the model is ran, and the predicted values pass through the sigmoid function, we classify as per:

$$\text{Class} = \begin{cases} \text{Benign} & \text{if } \sigma(\hat{\mathbf{Y}}) < 0.5 \\ \text{Malignant} & \text{if } \sigma(\hat{\mathbf{Y}}) > 0.5 \\ \text{Inconclusive} & \text{if } \sigma(\hat{\mathbf{Y}}) = 0.5 \end{cases}$$

Seeding affected the entries that were sampled for any given trial, and so we repeated each trial for 100 unique seeds. We averaged over all seeds to compute 5 performance metrics for our model and a given training-test size split: mean accuracy, mean precision, mean sensitivity, mean specificity, and mean negative log likelihood (NLL). This calculation was performed for all the data-replacement experiments previously outlined. Of all measurements, the NLL, calculated by the binary cross-entropy loss function, is of poor value given our use of optimal $\boldsymbol{\beta}^*$ values. Hence, this indicator is included in Figure 3 for reference only.

***Results.***– The performance of our model can be seen in Table I, with accompanying graphs plotting performance against training set size found in Figure 3. For all methods of data manipulation we see a positive correlation between performance of predictions (made on unseen features from the testing dataset) and training set sizes. This is also true for sizes past 70%, though these are overlooked to reduce over-fitting.

Removing the rows of data with missing entries yielded the best results, with 96% accuracy, 97% precision, 99% specificity, and 92% sensitivity. For all experiments, we observe sensitivity to be the most difficult metric to achieve good scores. Thus, whilst this model may rarely produce false positives (high specificity), it may yield slightly more false negatives. Arguably, this is the most favourable balance of the two metrics, if one had to be lower. Altering a dataset, be it with the mean, minimum or maximum values of any feature, skews the data in a way that may be unrealistic.

Replacing the missing data with the mean value performed nearly as well as removing the data entirely, but achieved these results for a much lower training set size. However, despite the decreased training cost this model still skews the data. This is incredibly undesirable due to the costs involved with assigning false negatives in a medical context. Replacing missing data with the minimum or maximum of features performed similarly to the mean, but with an increased training cost.

It is interesting to note the behaviour of all of the models as the training set size passes 90%. Each metric starts to jump and spike away from the trend, sometimes trending higher than previously achieved accuracy, but also falling below the benchmark. This is because the model overfits, and is very dependent on the remaining test cases. As the model is unable to generalise with training sets this size, the accuracy of the model drops significantly.

| Metric/Method | Removed | Mean | Min | Max |
|---|---|---|---|---|
| Accuracy | 96% (70%) | 96% (44%) | 96% (44%) | 96% (44%) |
| Precision | 97% (70%) | 96% (40%) | 96% (70%) | 96% (70%) |
| Specificity | 99% (70%) | 98% (40%) | 98% (70%) | 98% (70%) |
| Sensitivity | 92% (55%) | 92% (57%) | 91% (49%) | 91% (49%) |

TABLE I. Maximum metric achieved by each method, with the training set size that achieved it.

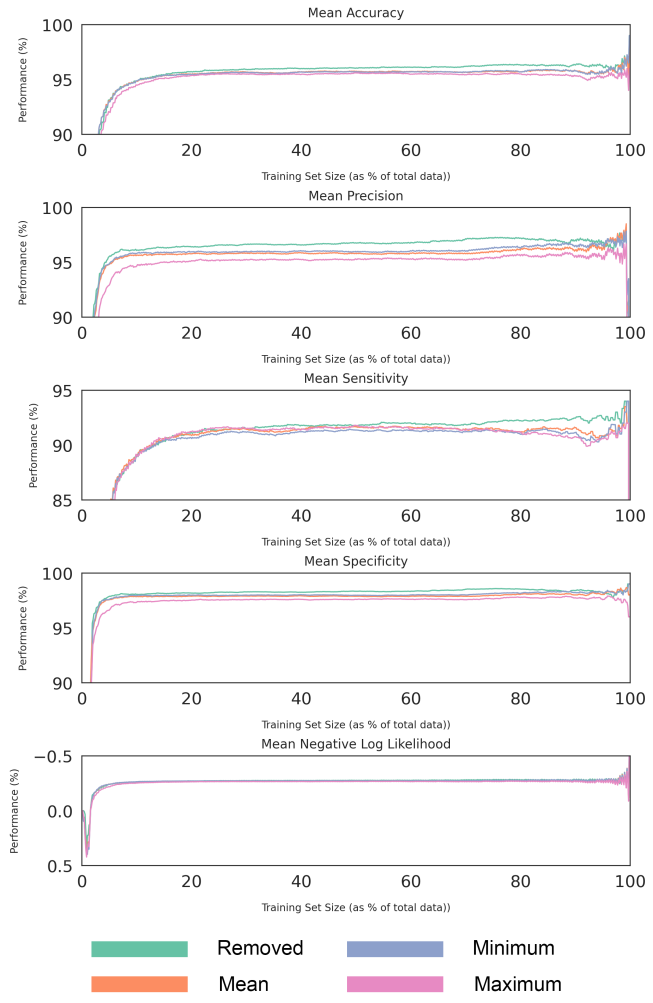


FIG. 3. **Plots showing performance metrics against training set size.** All metrics increase sharply as the training set size increases before starting to plateau. After the training set size reaches 90%, the metrics jump and spike.

***Discussion.***– Despite this model showing promising results, it first seems reasonable to put them into context. This task is removed from clinical reality for the labels "benign" and "malignant", though useful, are descriptive terms for sub-types of non-cancerous and cancerous changes. For example, different sub-types of malignancy exist, each presenting differently and with unique features correlations. There is also a diagnostic grey area, such as carcinoma in situ [cite ICD-10], not considered by this model. Thus, a binary classification appears a crude approach, and the clinical use of this model's output is difficult.

Although clinicians continue to perform better in classification [cite italian], machine learning models are faster. However, in order to use our model we first require that features undergo discretisation (i.e. ranked). Presently, this demands humans input and thus is a bottleneck in a process with potential for automation. Other machine learning methods, perhaps taken from image recognition (eg. convolutional neural networks) could assist here. However, in doing so we attract the ethical issue of depersonalising healthcare and treating a disease, rather than advocating holistic styles of medicine.

This report also only explored linear regression. Other supervised learning methods (eg. support vector machines or decision trees) could yield better prediction performance for similar datasets. Additionally, and as seen for all supervised learning problems, a larger dataset from which the model can be trained will always be beneficial. Here we used a logistic regression model for a binary classification task. For our case, we demonstrated that missing data are best removed entirely from sets prior to learning. We also provided evidence of the effect of over-fitting. To conclude, our report summarises the literary findings that suggest caution is needed when selecting and using data for supervised learning, especially in medical contexts.


[1] S. Lee, Y. Chu, J. Ryu, Y. J. Park, S. Yang, and S. B. Koh, Yonsei medical journal **63**, S93 (2022).
[2] S. Hong, Y. Zhou, J. Shang, C. Xiao, and J. Sun, Computers in Biology and Medicine **122**, 103801 (2020).
[3] B. Abbasi and D. M. Goldenholz, Epilepsia **60**, 2037 (2019).
[4] D. W. Langerhuizen, S. J. Janssen, W. H. Mallee, M. P. Van Den Bekerom, D. Ring, G. M. Kerkhoffs, R. L. Jaarsma, and J. N. Doornberg, Clinical orthopaedics and related research **477**, 2482 (2019).
[5] C. Kriza, V. Amenta, A. Zenié, D. Panidis, H. Chassaigne, P. Urbán, U. Holzwarth, A. V. Sauer, V. Reina, and C. B. Griesinger, European Journal of Radiology **145**, 110028 (2021).
[6] M. S. Sadaghiani, S. P. Rowe, and S. Sheikhbahaei, Annals of Translational Medicine **9** (2021).
[7] A. Moglia, K. Georgiou, E. Georgiou, R. M. Satava, and A. Cuschieri, International Journal of Surgery **95**, 106151 (2021).
[8] T. J. Loftus, P. J. Tighe, A. C. Filiberto, P. A. Efron, S. C. Brakenridge, A. M. Mohr, P. Rashidi, G. R. Upchurch, and A. Bihorac, JAMA surgery **155**, 148 (2020).
[9] "Breast cancer statistics," (2022).
[10] R. Morton, M. Sayma, and M. S. Sura, Breast Cancer: Targets and Therapy **9**, 217 (2017).
[11] D. Dua and C. Graff, "UCI machine learning repository," (2017).

[12] O. L. Mangasarian and W. H. Wolberg, *Cancer diagnosis via linear programming*, Tech. Rep. (University of Wisconsin-Madison Department of Computer Sciences, 1990).

[13] J. Chhatwal, O. Alagoz, M. J. Lindstrom, C. E. Kahn Jr, K. A. Shaffer, and E. S. Burnside, AJR. American journal of roentgenology **192**, 1117 (2009).

[14] X. Zhou, K.-Y. Liu, and S. T. Wong, Journal of Biomedical Informatics **37**, 249 (2004), biomedical Machine Learning.

[15] G. Van Rossum and F. L. Drake Jr, *Python tutorial* (Centrum voor Wiskunde en Informatica Amsterdam, The Netherlands, 1995).

[16] T. pandas development team, "pandas-dev/pandas: Pandas," (2020).

[17] C. R. Harris and K. J. Millman, "Array programming with numpy," (2020).