Breast Cancer Classification using Machine Learning: A Comparison of Algorithms using the Wisconsin Diagnostic Dataset

Abstract

Every year, there are around 55,900 new breast cancer cases in the UK, 23% of which are preventable with early diagnosis and treatment[i]. In this project, we investigated the effectiveness of three different machine learning algorithms when applied to the Wisconsin Diagnostic Breast Cancer dataset. The dataset consists of numerical data obtained through digitised images of breast tissue, that describe the characteristics of cell nuclei. We compared the performance of Logistic Regression, K-Nearest Neighbours and Support Vector Machine, on the task of classifying breast tissue as either malignant or benign, through measuring recall, specificity, F2-measure and Matthew's Correlation Coefficient. Our results showed that the logistic regression performed best, with a recall of 0.952 and Matthews Correlation Coefficient of 0.943.

Introduction

The United Kingdom's National Health Service (NHS) is under significant strain following the coronavirus pandemic. Cancer services are under particular pressure with a sharp rise in long waits for cancer therapy in the past four years. These delays have been described by Steven McIntosh of Macmillan Cancer Support as "traumatic" and may be having an impact on the chances of survival. One of the primary issues he has identified is that the NHS is "chronically short staffed". An area he notes as being particularly impeded by shortages is the process of diagnosis[ii].

The aim of this study is to evaluate the performance of a variety of machine-learning models to determine if they are sufficiently reliable in identifying malignancy in a sample of breast tissue. If a model with sufficient reliability can be found, our research could be used to provide support for the introduction of machine-learning tools to aid physicians in clinical diagnosis. It may also progress the search for the optimal pre-processing, algorithm and hyperparameter combination to be deployed in such tools. Doctors could be alerted to the most urgent cases requiring inspection, potentially allowing for faster diagnosis of cancer with less human oversight and freeing up resources for other areas of NHS care.

This report investigates a wide variety of algorithms and hyperparameter combinations to produce an optimal binary classifier. We will investigate the machine-learning models which have historically produced the best results and perform a rich hyperparameter analysis. The performance of the optimal models will then be evaluated and it will be determined if they are reliable enough to be considered for further development in real-world applications.

To train and evaluate the models, we will be using the 1995 Diagnostic Wisconsin Breast Cancer Database[iii]. The attributes of this dataset describe the characteristics of digitized images of breast mass cell nuclei obtained from fine needle aspirates (FNA). The samples are divided into malignant (cancerous) and benign (non-cancerous). The model's aim is to successfully identify the malignant samples.

Our problem is formally defined as a machine learning task by Tom Mitchell's definition[iv] as follows: Our model will improve over task **T**, which is to classify breast tissue through numerical values of digitized images, as malignant or benign. Such a model will learn with experience **E**, which is a subset of training data from 569 examples of numerical values relating to the nuclei of breast tissue, as judged by performance measure **P,** which is the model's Matthew's Correlation Coefficient.

Exploratory Data Analysis and Data Preparation

Each record within the Wisconsin diagnostic breast cancer dataset (WDBC) consists of 32 attributes. They are attributed with an ID number, diagnosis (malignant or benign) and the mean, standard error and worst measurements of 10 real-valued features across the nuclei in a sample. Those features are radius, texture, perimeter, area, smoothness, compactness, concavity, concave points, symmetry and fractal dimension. There are 569 samples within the dataset with no null values or duplicate records. We dropped the ID number attribute from the dataset as it provides no useful information for training or evaluating the models.The dataset is mildly imbalanced with the minority class (malignant) composing less than 40% of all records. Therefore, we decided that it would be best to stratify the data when dividing it into samples in order to reduce bias in sample selection.

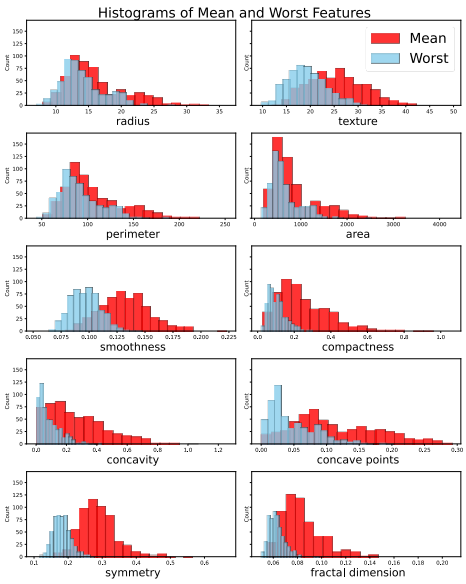


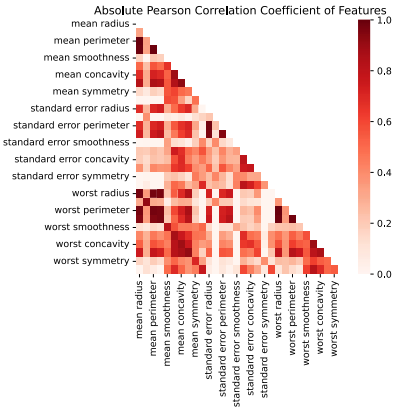
Figure 1

As can be seen in **Error! Reference source not found.**, the distribution of features largely resembles a bell-shaped curve with a positive skew. To ascertain whether the data were normally distributed, we applied the Shapiro-Wilk test which determined that there was evidence that none of the features followed a normal distribution.

The features are measured on widely different scales. The consequence of this is that without rescaling, we would be limited in the choice of models available and other pre-processing techniques (such as dimensionality reduction) we applied would not work as intended. Therefore, we deemed it necessary to apply standardisation and normalisation techniques when training the model. This would also assist in helping the models converge.

By analysing the correlations between the different features, we discovered a high degree of multicollinearity. For example, in Figure 2, the heatmap shows a very strong correlation between, area, perimeter and radius. Due to the presence of widespread collinearity and a large number of features, we decided that dimensionality reduction would be an effective strategy for reducing computational load and negating the risk of overfitting.

Figure 2



The dimensionality reduction technique we chose to employ was principal component analysis (PCA). This is because it deals effectively with multicollinearity with a low computational cost whilst maximising variance of data in a low-dimensional representation. The drawback of this is low interpretability of principle components. However, this information loss didn't matter to our study as we were not concerned with the influence of specific features in our analysis. We considered other process such as wrapper methods, but ultimately ruled them out due to high computational costs.

Many outliers were present throughout the data. PCA is very sensitive to outliers and therefore it was necessary to try to reduce their number. We did not consider the removal of the outlying data to be viable as it would be too significant of a decrease in the size of the dataset, increasing the risk of overfitting. Instead, we decided the best approach was to mitigate the effect of outliers through our choice of scalar. We hypothesised that a quantile transformation scaler would be best suited to address this as it is less severely affected by outliers. To confirm this, we decided to compare quantile transformation with scikit learn's standard scaler and robust scaler.

Learning Algorithm Selection

When choosing which machine learning algorithms to use, we considered the following heuristics:

1. The nature of the problem we were trying to solve
2. The relative advantages and disadvantages of different algorithms that we could apply, and how they would be suited to our data set and preparation methods.
3. What the literature supported as being most effective in solving breast cancer detection and classification

Considering these measures in order, we first identified that our problem was a supervised binary classification problem. From researching the most popular machine learning algorithms and re-examining our lecture notes we came to review a list of the following algorithms:

Logistic Regression, K Nearest Neighbours (KNN), Support Vector Machine (SVM), Decision Trees, Random Forest, and Naïve Bayes.

Upon re-reviewing our lectures, we decided to remove Decision Trees from this list. This was because we were advised that when considering tree-based learning methods, a random forest would generally yield better results due to it drawing upon multiple decision trees.

To reduce this list of 5 down, to 3, we reviewed the literature on breast cancer detection and classification to see which of these models had performed best with performance measures that we would be using to evaluate our models. The table below shows the findings from two recent studies which compared machine learning algorithms when applied to the WDBC dataset. The F2 measure for the first study was calculated using the precision and recall values provided. The table has been coloured to show which place each algorithm finished for each performance metric.

Rather surprisingly, the first study showed random forest to have the best performance in all performance metrics, whereas,

Metric	Study					
	Breast Cancer Detection and Classification: A Comparative Analysis Using Machine Learning Algorithms(v)			Feature Selection Using Correlation Analysis and Principal Component Analysis for Accurate Breast Cancer Diagnosis(vi)		
	Recall	MCC	F2-Measure	Recall	MCC	F2-Measure
SVM	0.857	0.849	0.874	0.939	-	0.935
Logistic Regression	0.857	0.869	0.878	0.952	-	0.957
Random Forest	0.929	0.944	0.942	0.829	-	0.880
KNN	0.786	0.836	0.821	0.897	-	0.913
Naïve Bayes	-	-	-	0.918	-	0.918

in the second, the complete opposite was displayed. Logistic regression and SVM were shown to perform well in both studies, so we selected them both for this report. We then had to decide the final algorithm to include. Naïve Bayes assumes that each feature makes an equal contribution to the target class, however, this would be unlikely given that the features in our dataset would be principal components. Furthermore, Naïve Bayes' main advantage is being computationally quick which was less relevant to us given the relatively small size of the WDBC dataset. For this reason, we decided to pick

KNN due to it being suited to smaller datasets and effective when applied to a smaller number of features, which we would ensure through PCA.

Model Training and Evaluation

Training Approach

To begin the process of training the models, we first split the dataset into a training and test set. We used the common rule of thumb of an 80/20 split as this follows the Pareto principle. With the training set, we performed hyperparameter tuning using a grid search with k-fold cross-validation.

Hyperparameter Selection and Tuning

When considering what k value to use, we tried to balance the trade-off between variance and bias that large and small values of k would respectively yield. In addition, we ideally wanted to repeat the cross-validation to lower the variance of our results, however, this was competing with other factors for our runtime such as the number of hyperparameters and their range.

When performing cross-validation without repetitions, we observed that the best hyperparameters would frequently change depending on how the training set was split into folds. This provided empirical support for the implementation of repeated cross-validation to reduce the variance in our results.

In determining k, we wanted to choose it to be a factor of the size of the training set, ensuring folds of equal size. Furthermore, we considered how a large k could negatively affect how representative our folds were of the distribution of diagnosis.

Our training set was a relatively small size of 455, leaving it more vulnerable to something like this occurring. As such, we decided to pick the factor k=7. We would have ideally used k=13, another factor of 455, however, one of our aims was to investigate a large range of hyperparameters; a goal which would only be achievable if we prioritized enough of our finite resources towards it. By selecting a smaller k, the different hyperparameter combinations would be calculated quicker, allowing us to investigate a larger range of them and their candidate values.

When considering which hyperparameters to tune, we wanted to find a balance between analytical depth and maintaining a reasonable run-time for the grid search (an unpractically long search would impede development). We also desired a good degree of compatibility between hyperparameter combinations to simplify the analytics we would examine from the training process. With this in mind, we chose the following hyperparameters to evaluate in a grid search:

Model	Hyperparameter	Candidate Values	Notes
Logistic Regression	Algorithm (solver)	lbfgs, liblinear, newton-cg, sag, saga	Removed newton-cholesky from training as it failed to converge. We did not want to sacrifice runtime by adding more maximum iterations, when this was the only algorithm that failed
	Inverse regularisation strength (C)	$10^{-3}, 10^{-2}, 10^{-1}, 10^{0}, 10^{1}$ $10^{2}, 10^{3}$	Selected increasing powers of 10 in order to test a range of orders of magnitude
	Penalty type	l1, l2	Removed 'none' option as models would fail to converge within the time and we didn't wish to increase iteration count at the cost of run-time
	Class Weight	balanced, none	We hypothesised that weighting would improve model performance by addressing the mild class imbalance in the dataset
K Nearest Neighbours	Number of neighbours	1,3,5,9,15,23	Selected odd values in order to avoid ties
	Weight function	uniform, distance	We selected all possible candidate values
	Algorithm	ball tree, kd tree, brute	Omitted auto as this would select one of the other algorithms
	Distance metric	minkowski, euclidean, manhattan	
Support Vector Machine	Inverse regularisation strength (C)	$10^{-3}, 10^{-2}, 10^{-1}, 10^{0}, 10^{1}$ $10^{2}, 10^{3}$	Selected increasing powers of 10 in order to test a range of orders of magnitude
	Kernel type	linear, poly, rbf, sigmoid	Only omitted 'precomputed'
	Kernel coefficient (gamma)	scale, auto	

Underfitting and Overfitting

One of our main concerns when preparing the data and training our models was the risk of either overfitting or underfitting. We integrated several strategies to mitigate the risk of either of these occurring.

Arguably our most important measure in preventing overfitting was the use of PCA. Due to our dataset being relatively small, yet containing a fair number of attributes, the data points would be sparsely separated; a characteristic that will inherently lead to a model separating the data without any patterns necessarily existing. Through using PCA, we heavily reduced the amount of noise captured whilst still capturing much of the variance. We coupled this, where we could, with

regularization to further remove the features which the model deemed unnecessary. At this point, we realised we had to be particularly careful with KNN due to it not having an option for regularization and that as an instance-based algorithm, it already tended to overfit. To counteract this, we made sure to include some larger values of k, so that the model would have the opportunity to become less well-fitted to the data. In addition to the measures presented, throughout our grid search, we used k-fold cross-validation to evaluate each combination of hyperparameters. By training and testing the model on different subsets of the data, we robustly ensured that a set of hyperparameters that performed well during training were truly good at generalising.

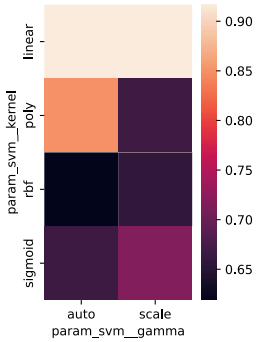
In addition to preventing overfitting, the use of k-fold validation also prevented our models from underfitting. We scrutinised each set of hyperparameters through training and testing 7 times for 3 iterations each. A simplistic model would not be able to consistently maintain a high enough score for it to have a chance of being selected through this process. One particular area of concern we had with underfitting our models was the use of PCA and regulirization, which could potentially remove too much information when reducing our data's dimensions. Accounting for this, we took care to provide both with a large range of candidate values, allowing us to select a value that would not cause the model to underfit as would be evidenced by its low k-fold validation score

Evaluation Metrics

To evaluate the performance of the models, two metrics were chosen: Matthews's Correlation Coefficient (MCC) and Recall. We selected MCC as the primary means of ranking performance as it takes into account all four measures in the confusion matrix and so provides a balanced view of a model's overall performance. This mitigates the risk of producing overoptimistic results which other metrics can produce.

For further analysis, we also considered recall score. This measure was suited to the context of our problem as the consequences of false negatives are very costly. A false negative may increase the time it takes for someone with cancer to be diagnosed and thus receive appropriate treatment. As the model's aim is to alert clinical professionals to potential cases which may require further investigation, false positives can be considered less consequential. However, as recall alone can easily produce a biased result, it is used to supplement the analysis rather than act as the primary ranking metric.

Figure 3

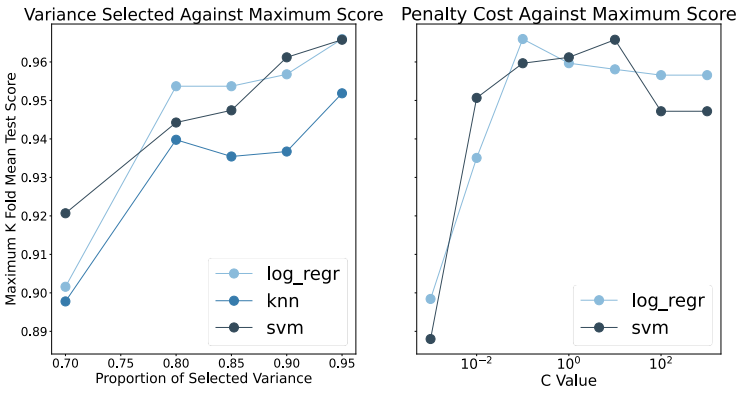


Model Comparison

Before being able to compare each of the models directly, we needed to conclude the optimal hyperparameters for each one.

Firstly, we investigated hyperparameters that were common amongst the different models, the results of which can be seen in Figure 4. Our results showed that all the models required a larger amount of variance than we expected to perform optimally.

Figure 4



For the inverse regularization strength, both models performed poorly when it was very small. We concluded this was likely caused by the models underfitting, as they would be heavily incentivised to use fewer features. Furthermore, we observed that when C was very high the models still performed well. This led us to hypothesise that we should add an option for no regularization at all, however when testing this, logistic regression failed to converge.

Our research led us to believe that we should not apply standardization to non-normally distributed attributes, however, our results showed otherwise, with the standard scaler performing the best for each model.

For each model we then plotted a bar chart for every hyperparameter, with a bar for each candidate values best performance. For logistic regression and KNN there seemed to be very minor differences between the candidate values best performance. An exception to this was KNN's neighbours, which peaked at 9 and fell off gradually either side. The hyperparameter with the most variance between its candidate values was the kernel in the support vector machine. Despite the rbf kernel having the best performing model overall, when looking at a heatmap of average performance we could clearly see it performing much worse as shown in Figure 3.

It is intriguing to note that the best performing logistic regression model did not benefit from class weighting despite our dataset being mildly imbalanced. We suspect this could be due to the class imbalance being on the boundary of balanced.

After analysis, the best performing model configurations were determined to be as follows:

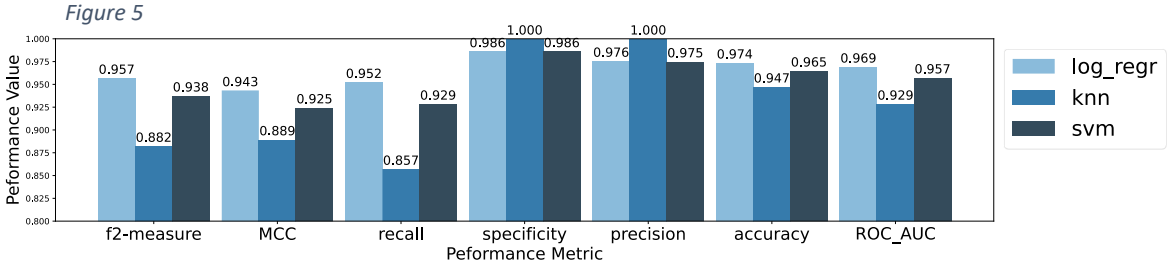
Model	Optimal Hyperparameters	MCC Score

Logistic Regression	**Scaler**: Standard Scaler, **PCA Components**: 0.95, **Algorithm**: liblinear, **C**: 0.1, **Penalty**: l2, **Class Weight**: none	0.9660
Support Vector Machine	**Scaler**: Standard Scaler, **PCA Components**: 0.95, **C**: 10, **Kernal**: rbf, **Gamma**: scale	0.9658
K Nearest Neighbours	**Scaler**: Standard Scaler, **PCA Components**: 0.95, **Number of Neighbours**: 9, **Weight function**: distance, **Algorithm**: ball tree, **Distance Metric**: manhattan	0.9519

These optimal models were then retrained on the entire training set and then tested against the test set. Their performances were evaluated using a range of metrics.

Figure 5 below shows how each model performed for F2 measure, MCC, recall, specificity, precision, accuracy and ROC AUC. We can more clearly see how these values were calculated from the confusion matrix in Figure 6.

Figure 5

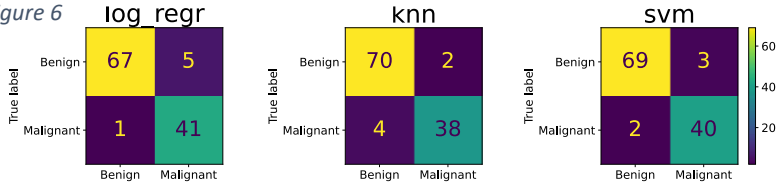


It was clear from these two figures that KNN had learned to use a higher threshold, favouring its true negative rate to the detriment of its true positive rate. This is characterised by its perfect specificity but comparatively low recall score. In the confusion matrix it can be seen to label all the negative examples correctly as benign, but at the cost of mislabelling 6 malignant examples as benign. In the context of the medical application of our model, such behaviour is particularly detrimental, making it less desirable when compared to the other models.

In contrast, the results of logistic regression and SVM are almost identical: logistic regression has just one less false negative. This difference, however, is enough for logistic regression to beat it in nearly every single metric.

K Nearest Neighbours performed best in terms of specificity and precision. However, in the context of real-world clinical application, these metrics are less salient than the others as they measure the model's ability to mitigate false classifications of malignancy. Although this is important, it is more vital that actually malignant samples are classified as such as misclassification in this case may severely delay diagnosis. Benign samples classified as malignant increases the number of cases for physicians to review, but the resulting delay is much less significant. Logistic regression achieved the highest scores on all other measures and so we concluded it to be most appropriate for a clinical setting.

Figure 6



Conclusion and Discussion

Process Summary

The aim of our project was to build upon the work of previous studies to develop and find the optimal hyperparameters for a machine-learning model which can successfully classify a sample of breast tissue as malignant or benign. Our model was trained and evaluated on the WBDC dataset which was composed of breast tissue samples obtained from fine needle aspirate. To identify which candidate models to train and compare, we referred to two previous studies on the same dataset and selected the best-performing algorithms. Our goal was to test and evaluate a rich array of preprocessing techniques and hyperparameters to improve upon the results of past reports. We wished to produce evidence to support the hypothesis that a machine-learning model can be created that is sufficiently reliable and accurate to be used safely in a real-world setting.

Results Summary

After training and testing, the best model generated, as measured by MCC and recall was logistic regression with the following preprocessing and hyperparameters:

Scaler: Standard Scaler, PCA components: 0.95, No class weighting, Algorithm (solver): liblinear, Inverse regularisation strength (C): 0.1, Penalty norm: (l2)

This model, on testing, produced an MCC score of 0.943 and a recall score of 0.952.

We were pleased to see that our optimised logistic regression model performed better than the best logistic regression model found in the previous studies we examined. Our recall score was equal to the best found and our MCC score was 8.5% higher than the previous best. Therefore, we succeeded in achieving the primary aim of our study.

Implications and Future Development

The results of our study provide further evidence that machine learning algorithms have the potential to perform well enough to aid physicians in the diagnosis of breast cancer. A study[vii] in the American Journal of Clinical Pathology examined the

performance of a human radiologist-pathologist team classifying malignancy of FNA of breast tissue. They produced a sensitivity score of 92.6% and a specificity score of 96.8%. When compared with our optimal model's respective scores of 94.3% and 98.6%, there is an indication that our discovered model configuration is aligns with acceptable medical standards. Of course, this is only a preliminary investigation using a relatively small dataset and so further research is required. To further develop this investigation, our optimal model should be trained and evaluated on a much larger dataset to see if the results hold at a larger scale.

Limitations and Improvements

A major limitation of our study is the processing speed of the hardware we had access to. We could not evaluate the full spectrum of hyperparameters and pre-processing techniques as we would have wished as it would have led to an unacceptably long run-time for the grid search. If we had access to more powerful machines, we would test more hyperparameters and pre-processors to discover a better potential combination.

Another limitation was the size of our dataset. With only 569 samples, the WBDC dataset is relatively small; therefore, we cannot be confident in the generalisability of our findings due to the risks of overfitting and low estimation precision. As a consequence, our study only serves as a preliminary investigation and requires much further research. To improve confidence in our results, we would run the study again on a much larger dataset.

Lastly, before engaging in this project, we had very limited knowledge of and experience in using the python libraries and the workflow involved in machine learning. This meant we had to spend a large amount of our time researching code semantics and understanding the structure of the required program. With more time, we would further investigate different areas such as dimensionality reduction and data exploration.

Lessons Learned

Throughout this project, our team learned much about teamworking and communication skills. Initially, we encountered some challenges with conflicting work being performed and a lack of clear vision. However, we quickly developed a plan and implemented agile development practices such as regular meetings and pair programming. In the end, we were able to work as a cohesive team with clear goals, schedules and communication.

We also deepened our knowledge about machine learning in general through extensive research into data preprocessing, model selection, evaluation metrics and hyperparameter tuning. Furthermore, we developed a familiarity with various python libraries including pandas, matplotlib and scikitlearn and feel confident in using them in future projects. Finally, through deep exploration of many previously unfamiliar topics, the process of developing our project has strengthened our independent and collaborative research skills and report writing.

---

[i] https://www.cancerresearchuk.org/health-professional/cancer-statistics/statistics-by-cancer-type/breast-cancer#:~:text=23%25%20of%20breast%20cancer%20cases%20in%20the%20UK%20are%20preventable.

[ii] Cancer care delays: How bad are they in your area? (2022). BBC News. [online] 10 Nov. Available at: https://www.bbc.co.uk/news/health-63573718.

[iii] archive-beta.ics.uci.edu. (n.d.). UC Irvine Machine Learning Repository. [online] Available at: https://archive-beta.ics.uci.edu/dataset/17/breast+cancer+wisconsin+diagnostic.

[iv] Mitchell, T.M. (1997). Machine learning. New York: Mcgraw-Hill.

[v] Sakib, Shadman & Yasmin, Nowrin & Tanzeem, Abyaz & Shorna, Fatema & Hasib, Khan & Alam, Saadia. (2022). Breast Cancer Detection and Classification: A Comparative Analysis Using Machine Learning Algorithms. 10.1007/978-981-16-8862-1_46j

[vi] Ibrahim, S., Nazir, S., & Velastín, S.A. (2021). Feature Selection Using Correlation Analysis and Principal Component Analysis for Accurate Breast Cancer Diagnosis. *Journal of Imaging, 7*. Please note we used table 2 from this paper as it was most relevant to how we had preprocessed our data. We applied, PCA but not wrapper methods, which the values in this table show. We also rounded the values to 3sf to be consistent with the other study

[vii] Farras Roca, J.A., Tardivon, A., Thibault, F., El Khoury, C., Alran, S., Fourchotte, V., Marck, V., Alépée, B., Sigal, B., de Rycke, Y., Rouzier, R. and Klijanienko, J. (2017). Diagnostic Performance of Ultrasound-Guided Fine-Needle Aspiration of Nonpalpable Breast Lesions in a Multidisciplinary Setting. American Journal of Clinical Pathology, 147(6), pp.571–579. doi:10.1093/ajcp/aqx009.