Breast Cancer Diagnosis Using Support Vector Machine and Random Forest Classifiers

Jacob Dean (ID: 20046542)
School of Computer Science
University of Nottingham
psxjd13@nottingham.ac.uk

Alfred Adjei (ID: 20331004)
School of Computer Science
University of Nottingham
psxaa39@exmail.nottingham.ac.uk

***Abstract*—We apply Support Vector Machine and Random Forest classification methods to the original version of the Breast Cancer Wisconsin dataset. Interestingly, our best performance is attained by applying a Random Forest model to the preprocessing methodology originally intended for SVM. This model yields a 97.1% accuracy and a 100% sensitivity. The high sensitivity value was obtained by using oversampling and by reducing the classification cutoff from 0.5 to 0.31. Although we observed other models that achieved higher accuracy values, sensitivity is of particular importance in the context of cancer diagnosis.**

Keywords—Support Vector Machine (SVM), Random Forest (RF), Wisconsin Breast Cancer Dataset (WBCD)

I. Introduction

In recent years, the use of computer science techniques in healthcare has largely focused on three major categories of ailment: cancerous, cardiovascular and neurological. It may not be immediately obvious how computer science can benefit such diverse medical fields. But if we look more closely, all three categories have the common property of there being a high correlation between early detection and patient recovery. Various papers have shown that early detection of disease can usually be modelled as a classification task (i.e., differentiating between datapoints based on two or more distinct groups). In the sections that follow, this paper investigates the challenge of diagnosing breast cancer by building a series of classification models using two techniques: Support Vector Machines and Random Forests. The dataset this paper will explore is the Wisconsin Breast Cancer Dataset obtained from the University of California Irvine's (UCI) machine learning repository.

The paper is organized as follows; section II reviews the relevant literature that is based on the same dataset, section III provides an overview of the dataset being explored, section IV outlines the preprocessing techniques chosen for each classification method, section V outlines the steps taken to properly tune our models for each classification method, section VI discusses the results and outcomes of the analysis, section VII explores follow-up tests leading to more relilable comparisons, and section VIII presents our conclusions as well as recommendations for future research.

II. Related Work

There is an extensive literature exploring the application of a wide range of classification methods on the BCW dataset. This literature can be split into two main categories: comparative papers that implement several distinct methodologies and focused papers that implement several variants of a single methodology.

Obaid et al. [1] compares the accuracy of SVM, kNN, and decision trees using 15-fold cross validation. Three variants of each classification method are explored, and the best performing variant is selected (quadratic SVM, medium kNN, and medium decision tree). Out of these three candidates, quadratic SVM performs the best with an accuracy of 98.1%. Vig [2] compares SVM, ANN, Random Forest, and Naïve Bayes. Unlike the previous paper, nested cross validation is used instead of simple cross validation to reduce bias, and quadratic SVM is the worst performing SVM model with an accuracy of 72.93%. SVM with a radial basis function kernel performs better with an accuracy of 93.63%, however Random Forest with 100 trees is the best performing overall with an accuracy of 95.64%. Showrov et al. [3] compare SVM, ANN, and Naïve Bayes. The original version of the BCW dataset (which contains missing values) is used here instead of the diagnostic version (which does not contain missing values) used in the previous two papers. For this reason, the authors perform several simple pre-processing steps such as replacing missing values with a rounded mean from the relevant column. 10-fold cross validation is used in this paper. Here, SVM using a linear kernel beats the RBF kernel that performed best in the previous paper with an accuracy of 96.72% compared with 95.72%. ANN using a RBF performs second-best with 95.88% accuracy.

All three comparative papers looked at SVM as a candidate classification method, however Polat & Güneş [4] focus solely on least square SVM, and their model performs better than the others with an obtained accuracy of 98.53%. This was achieved using 10-fold cross validation. Lavanya and Usha Rani [5] use decision trees as a single classification method and construct their trees using the CART algorithm (which involves the Gini Index for attribute selection alongside cost complexity pruning). The results demonstrate that feature selection can increase the performance of a classifier. Without feature selection, 94.84% accuracy was achieved on the original dataset, however by using the PrincipalComponentsAttributeEval feature selection method, a 96.99% accuracy was achieved. Nguyen et al. [6] focus on Random Forests in their paper, and like the previous paper, they use advanced feature selection (a four-step iterative method that involves ranking the features and eliminating those at the bottom of the list). In total, the reduced the number of features from 30 to 15. Random Forest is an extension of the decision tree methodology, which might explain why higher accuracy was achieved in this paper than the previous paper (99.82%). Interestingly, the authors used a relatively small number of trees (25 trees).

More recently, researchers have started applying ensemble methods to the BCW dataset, which bring together various classification methods under the hood of a single method. One such example is the Meta-Health Stack by Samieinasab et al. [7] which combines bagging, voting, and boosting algorithms into a single classifier. The Meta-Health Stack

achieved an accuracy of 98.2%, however a couple of single algorithms were able to achieve the same accuracy which raises the question of whether the additional complexity is needed. This is also the sentiment of Mushtaq et al. [8] who fine-tune the basic KNN algorithm by comparing a variety of feature selection methods, values for K, and distance functions. Their best performing model used Chi-square feature selection, K=1, and a Manhattan distance function to achieve an accuracy of 99.42%, which shows a fully optimised simple model can beat more complicated ensemble methods.

III. Dataset

Before delving into our analysis of this dataset, for the purpose of clarity, we will briefly outline the definition of cancer in modern healthcare. Cancer is a term used to describe a group of diseases characterized by an abnormal cell growth in one part of an organism's body which has the potential to spread to other parts. It should be noted that not every tumour (mass of cells) is cancerous. Certain varieties, although possibly discomforting, pose relatively little threat to health; these are known as benign tumours. Malignant tumours are most concerning to physicians and researchers because they possess the ability to spread or invade other cells in the body (metastasize).

Cancers have no specific point of origin and can start almost anywhere in the body of an organism. This data set is comprised of observations of cancers that start in the breast. There are three versions of the **Wisconsin Breast Cancer Dataset (WBCD)** which are distinguishable mainly by the number of observations as well as the number of attributes that they contain. For the purposes of this research, we consider the original variant of the WBCD which contains **699 observations and 11 attributes**. These are computed from a digitized image of a fine needle aspirate (FNA) of a breast mass and they describe characteristics of the cell nuclei present in the image. These attributes are described below:

Feature	Description
Clump thickness	Extent to which the epithelial aggregates were mono- or multi-layered.
Marginal adhesion	Cohesion of the peripheral cells of the epithelial cell aggregates.
Bare nuclei	Proportion of single epithelial nuclei that were devoid of surrounding cytoplasm.
Bland Chromatin	Blandness of nuclear chromatin.
Normal Nucleoli	Extent to which a cell's nucleolus was considered to be normal.
Cell shape	Uniformity of cell shape.
Cell size	Uniformity of cell size.
Single epithelial cell size	Diameter of the population of the largest epithelial cells relative to erythrocytes.
Mitoses	Infrequency of mitosis.

Fig. 1. Description of features

All these features are coded from 1-10 with 1 being 'most benign' and 10 being 'most malignant'. This means that the data is already normalised and there are no outliers, so we focus on other aspects of preprocessing in the following section.

IV. Preprocessing

A. Support Vector Machine (Method 1)

1) Selecting & Imputing Samples

An initial analysis shows 8 duplicated observations. Given the source of the data, three possible explanations arise. First: The duplicates may have been as a result of data observations being collected from a single individual at different points in time. Second: Owing to the discrete scaling of the data, it may also be possible that subtle nuances between the original values in these data observations were lost during processing. Third: Clerical error. In this analysis, the third explanation is chosen. Given that these observations account for less than 2% of the data points, it was determined that removing these observations would have little impact on our ability to train the model.

The dataset contains 16 missing values in the "Bare.nuclei" column. In dealing with these missing values, we investigate the effect of two imputation methods on the performance of a default Support Vector Machine (SVM) model: 1) Predictive Mean Matching and 2) conditional mean imputation [9]. In order to verify the performance of the default model on the two imputed datasets (mean-imputed and PMM-imputed), we use 10-fold cross validation[ref]. It is shown that PMM imputation outperforms mean imputation on both accuracy (97.1% against 96.4%) and sensitivity (98.64% against 98.57%) performance metrics. On account of these results, the values from PMM imputation are used.

2) Feature Selection

Besides the removal of the ID column, the significance of the dataset's features was assessed using two feature selection methods: Correlation Analysis and Principal Component Analysis [10]. Features in a dataset that are highly correlated provide redundant information. By eliminating such features, we can avoid a predictive bias for the information contained in those features. Correlation Analysis on this dataset reveals two features with a correlation higher than 90%: Cell Shape and Cell Size. In order to decrease the chance of predictive bias in our model, we remove the column that has the largest mean value, in this case the cell shape.

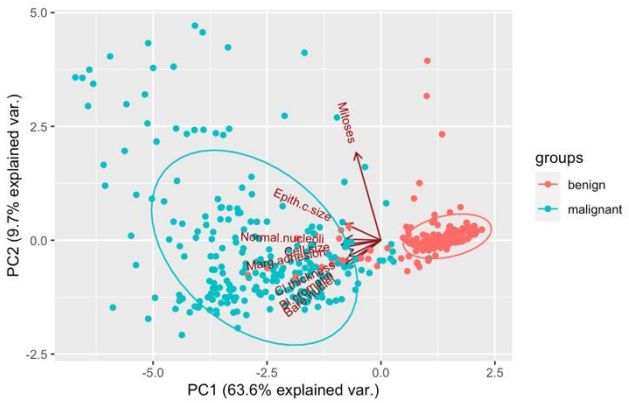

Fig. 2. PCA biplot of feature contributions to principal component 1 and 2

Using a PCA biplot, it can be seen that all variables contribute heavily to the first principal component, with larger values in these variables shifting the datapoints to the left side of the graph. It can also be seen from the separation of datapoints

that there seems to be a correlation between increased values in the all features and a class designation of malignant (the relationship is apparent but not as obvious for the Mitoses feature). Presented with this insight, we shall not remove any further features.

3) Data divison

A standard 80-20 split is used to separate the data into training and test partitions. In the classification section, parameters are tuned to find optimal values using ten-fold cross-validation on the training data, after which, the optimal model is tested on the test set.

B. Random Forest (Method 2)

1) Selecting & Imputing Samples

In this method no samples will be removed, unlike Method 1 where 8 were removed. The reason is that this method does not believe clerical error is the cause of the duplicates but instead follows the first two explanations. The reason for this is there are 54 duplicated ID values but 46 of these have values for the other features, suggesting they represent the same tumour observed at different times (the dataset was assembled over a two-year period). Since these samples represent new observations, they will not be removed.

For imputing the 16 missing values, a similar approach will be used to Method 1 by comparing an advanced imputation method to conditional mean imputation. However, this time random forest will be used as the advanced imputation method instead of PMM. This method works by using a random forest to build a proximity matrix keeping track of which samples are similar, using these similarity values to calculate weighted averages for the missing values, then repeating this process until the imputed values stabilise. We observe that random forest imputation yields a slightly lower out of bag (OOB) error rate (2.86%) compared to mean imputation (3.29%). Like in Method 1, the more advanced imputation method appears to perform better. However, since the performance difference is less than 0.5%, this may not be a statistically significant result.

2) Feature Selection

We begin by removing the ID column, because it should be uncorrelated with the proportion of malignant tumours (and any observed correlation would likely be unrepresentative of the wider population). Feature selection methods fall broadly under three categories: filtering, embedded, and wrapper. Methods from each category will be applied and then compared.

For our filtering method, we look at pairwise correlation. The 'Mitoses' feature has the weakest correlation with the outcome but after conducting a two-tailed test of the correlation coefficient's significance, we obtain a miniscule p-value less than 2.2*e^-16. Therefore, we do not remove the feature since it does seem to have some predictive power. Method 1 reached the same conclusion regarding 'Mitoses' but using the PCA plot rather than statistical tests.

For our embedded method, we construct a variable importance plot which looks at the decrease in mean Gini index if features were to be removed from a random forest model. Using this ordering of feature importance, we see that OOB accuracy falls by only 1.8% if the weakest 6 features are removed, suggesting several of these are redundant.

For our wrapper method, we use the Boruta algorithm, a method built specifically for random forest. This involves randomly shuffling columns to create 'shadow features' and then assesses whether doing so reduces accuracy. We observe that all 9 features were classed as important by this algorithm.

All in all, we have seen mixed results: correlation analysis and the Boruta algorithm suggest that all features should be retained, whilst the variable importance plot implies several features are not needed. However, computational time is not a major concern since the dataset is small, so all features will be retained. This is slightly different to Method 1 where the 'Cell Shape' column was removed due to its high correlation with 'Cell Size'. However, for random forests, predictive power is much more important than tree robustness, so multicollinearity is not a big concern.

3) Data Division

We will take an 85-15 training-test split since this retains over 100 observations for the testing set whilst retaining as much data as possible to train our models on. An important difference to Method 1 is that cross-validation will not be used here. The reason for this is that random forest trains each decision tree on a different bootstrapped dataset, and each of these datasets leaves OOB observations that can be used to tune the parameters of the model. In this way, bootstrapping will perform the same role as cross-validation.

V. CLASSIFICATION

A. Support Vector Machine (SVM)

1) Theoretical Background

Support vector machines as a classification technique emerged as a result of the inapplicability of the maximal margin classifier and support vector classifier on non-linearly separable data [9]. It works by enlarging the feature space using quadratic, cubic, or even higher-order polynomial functions of the predictors. Such functions are known as kernels and the process by which the feature space of a dataset is enlarged is known as the *kernel trick*. This trick allows for a linear separation boundary to be found in the larger feature space, essentially, finding a "line" between data observations that could not be found in the original feature space.

2) Model Tuning

Outside of the specific kernel used in classification, three parameters were tuned in this analysis;

a) Cost
b) Degree
c) Gamma

Cost is an essential parameter which sets a penalty for a wrong classification. The higher the cost, the more likely the model is to overfit. The gamma parameter is relevant to radial kernels and defines how far the influence of a single training example reaches, with low values meaning 'far' and high values meaning 'close' (far and close in terms of distance to other observations in the feature space). The degree

parameter is relevant to polynomial kernels and defines the curvature of the decision boundary. In essence, all three of these parameters influence the bias/variance trade-off but to varying degrees depending on the particular kernel.

The PCA plot in the previous section provided some evidence of separation among the data observations. Our main challenge was to determine if the data was linearly separable or not. Using a linear SVM kernel and an extremely large cost parameter (we will use a cost of 2^32 - typical cost values range from 0.1 to 100) we train a model which forces the decision boundary to be extremely thin (essentially overfitting the data). Once we train this model on our dataset using this parameter value and 10-fold cross validation, we may be fairly certain that any linearly separable data will produce predictions with 100% accuracy. If we don't obtain this accuracy value, the data is non-linear, in which case we explore using non-linear kernels. At a cost of 2^32, we find that accuracy falls to about 57% (considerably below 100%) and for this reason, we explore non-linear kernels in addition to the linear kernel.

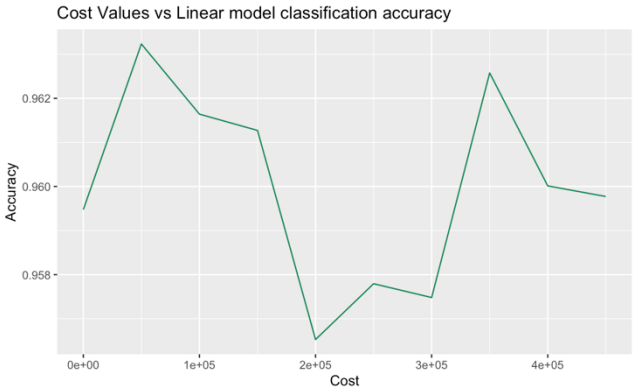

Fig. 3. Relationship between cost and accuracy

The figure above shows that as cost increases, accuracy rises and falls erratically. The relationship may not be accurately captured due to the non-linear nature of the data set. However, from out paragraph above, we know that at a cost of 2^32, accuracy falls to about 57%. It can also be seen that the accuracy never rises past its previous maximum. Given the above statements, we hypothesize that a graph plotted with a larger range of cost values would show a steady decline in the accuracy with increasing cost. In this case we are limited by our technological hardware.

B. Random Forest (RF)

1) Theoretical Background

Random forest is an ensemble classification method comprised of many independent decision trees and is often preferred to a single decision tree to prevent overfitting. Since each tree is constructed differently, the potential overfitting of individual trees is washed out when grouped together. There are two key features of decision trees: bagging and random subspaces.

Bagging is the process of taking a different bootstrapped dataset to train each tree in the forest. A bootstrapped dataset is formed by taking a random sample of rows (with replacement) equal to the number of rows in the main dataset. This means bagging is used to construct each tree with a different set of rows. Random Subspaces is the process of considering a random subset of the available features at each node of each decision tree and choosing the most powerful feature in this subset. This means each tree is constructed using a different set of columns.

We see that bagging and random subspaces mean each tree considers a different set of rows and columns, which is why each tree is built differently and therefore why random forest overcomes the problem of overfitting.

2) Model Tuning

We will consider four parameters when tuning our model:

a) Number of trees in the forest (ntree)
b) Number of variables at each node (mtry)
c) Balance of tumour classes (sampsize)
d) Percentage of votes required (cutoff)

a) Number of trees in the forest (ntree)

Since each tree is missing some features and observations, we expect adding more trees to increase model performance. But after a certain number of trees, all features and observations should be sufficiently represented, suggesting a limit to the gains from adding more trees. This hypothesis is supported by the following graph where we see a fall in OOB error only for the first few hundred trees. However, we do see a final drop after around 1750 trees so we select 1000 trees as a middle ground (note: default is 500 trees in R).

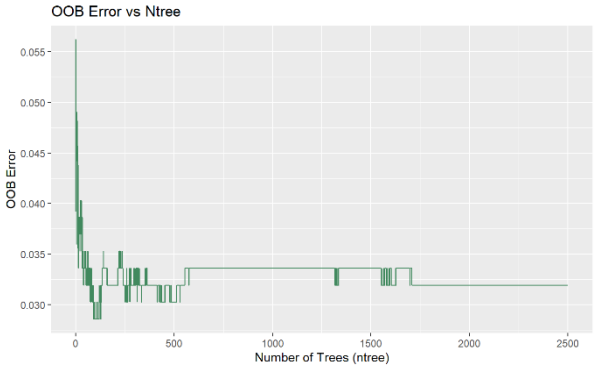

Fig. 4. Relationship between number of trees and error rate

b) Number of variables at each node (mtry)

With fewer features to choose from at each node, trees will be forced to use a diverse range of features, reducing the correlation between trees. However, forcing trees to use suboptimal also weakens their individual predictive power. These two antagonistic forces make the relationship between mtry and error rate ambiguous. When setting ntree at 1000 and considering mtry between 1 and 9 (corresponding to the total number of features), we find that an mtry of 2 minimises error.

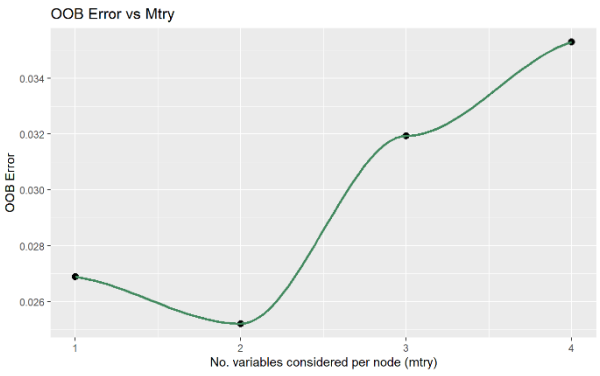

Fig. 5. Relationship between no. of variables considered and error rate

A strong parallel can be drawn between this graph and the graph of accuracy against cost for Linear SVM because both mtry and cost influence the level of overfitting in a similar way. Just like high values of mtry allow all trees to choose the same set of powerful features, high values for cost forces high penalties for misclassification and so incentivises the model to overfit. In this way, both graphs represent the bias-variance tradeoff where the left tail of the graph represents high bias whilst the right tail represents high variance.

c) Balance of tumour classes (sampsize)

The balance of our dataset is biased towards benign with 458 (66%) benign observations but only 241 (34%) malignant observations. This relative abundance of benign observations could make models particularly effective at diagnosing benign tumours. However, it is arguably more important to be strong at diagnosing malignant tumours due to the severe consequences of late breast cancer diagnosis.

Oversampling is one method that can correct the data imbalance. It involves randomly repeating malignant observations until they match the number of benign observations. An alternative method is stratifying each bootstrapped dataset to contain a higher proportion of malignant observations than default.

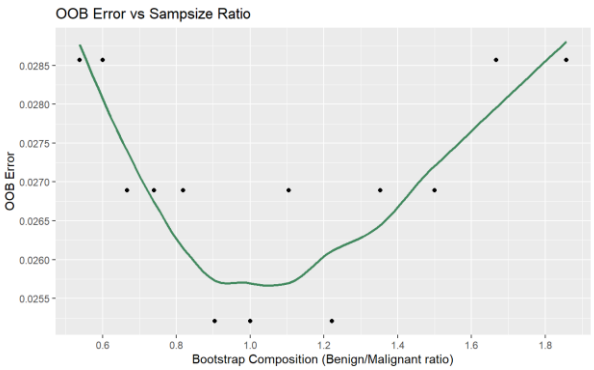

Fig. 6. Relationship between sample distribution and error rate

By varying the ratio of benign to malignant observations in each bootstrapped dataset, we would expect a trade-off between emphasising malignant observations and making best use of benign observations. When setting ntree as 1000 and mtry as 2 (following our previous steps), we observe that a ratio of 1:1 (in other words, a 50-50 bootstrapped sample) minimises error. Therefore, stratification seems to have a positive effect, considering the default ratio is about 2:1 (although this effect is small since the range on the y-axis is only about 0.3%)

d) Percentage of votes required (cutoff)

A random forest makes a final classification decision by comparing the decisions of the constituent decision trees. The default cut-off value is 0.5, meaning that for a random forest to classify a tumour as malignant, over 50% of the trees need to classify it as such. However, this cut-off is a parameter that can be varied. For example, cut-offs lower than 0.5 will make is easier to classify tumours as malignant since a lower proportion of the trees' votes are needed to do so. This would boost sensitivity at the cost of reducing specificity. In this way, tuning the cut-off amounts to selecting the optimal point on the ROC curve.

Due to the importance of successfully diagnosing malignant tumours, it may be beneficial to select a cut-off below 0.5. However, we also care about maintaining a degree of accuracy, so it makes sense to optimise a weighted average of accuracy and sensitivity. We select weightings of 75% for sensitivity and 25% for sensitivity to emphasise the importance of avoiding false negatives.

When plotting the relationships between performance and cut-off for our model using the oversampled dataset, we find that a cut-off of 0.4 maximises the weighted average. Due to oversampling 100% sensitivity is achieved at relatively high cut-off values, however this would not have been the case if we had a larger dataset. This suggests that the cut-off of 0.4 may not be fully tuned. For this reason we take an average between 0.4 and 0.22 (the optimal cut-off without oversampling) to get our final cut-off value.

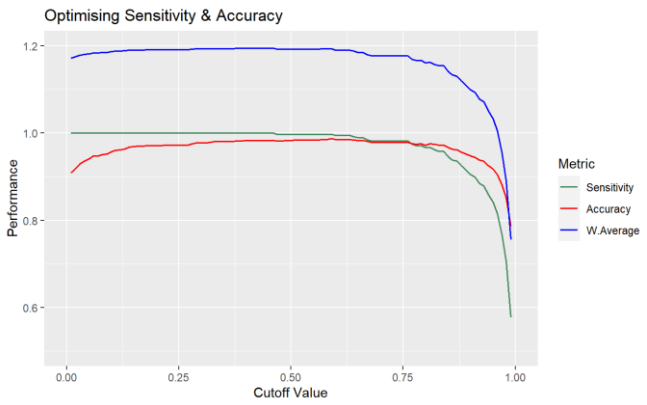

Fig. 7. Relationship between cut-off value and performance

3) Simultaneous Optimisation

So far, we have optimised parameters sequentially by optimising a single parameter, the feeding this into the model to optimise the next parameter. However, it is possible that we can only truly optimise our model by tuning all parameters simultaneously. This is because optimal performance with respect to one parameter doesn't necessarily translate to optimal performance when changing an additional parameter. When simultaneously optimising parameters, the number of trees will be fixed at 500 to limit computational load, which leaves mtry, sampsize, and cutoff as the parameters to optimise.

VI. Results

A. Support Vector Machine (SVM)

Employing 10-fold cross-validation on each kernel using typical ranges of cost, gamma and degree (0.1-100, 0.0001-10 and 1-10 respectively), the radial model with cost = 46.1 and gamma = 1e-04 was found to be the model with the lowest error rate on the training set data. (It should be noted that as randomisation was used in cross-validation, optimal parameter values are dependent on the set.seed value)

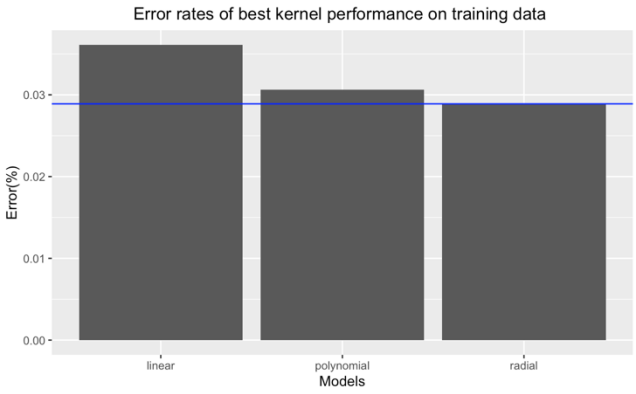

Fig. 8. Performance of kernel type on training data

Upon testing these models (fitted with their best parameters) on the test set data, the linear model with a cost of 0.1 is shown to be the best performing model. This result is confirmed by the figure above which shows hints that the polynomial and radial models overfit the training data. The table below shows the performances of four models on the test dataset:

Model	Accuracy (%)	Sensitivity (%)	AUC (%)
Default	95.6	98.0	99.1
Linear	**97.3**	**98.3**	**99.7**
Radial	95.0	95.2	99.7
Polynomial	95.7	95.2	99.7

Fig. 9. Table of results (SVM)

These results are consistent with the literature on SVM where the same dataset was used. In Showrov et al. [3] the linear model showed the best performance with an accuracy of 96.72%. Our linear model improves on this result slightly with an accuracy of 97.3%. The reason for this may be that our prior pre-processing steps (removal of the Cell Shape column) led to a more accurate model by removing predictive bias.

It is also interesting to note that the AUC does not change for all three kernels.

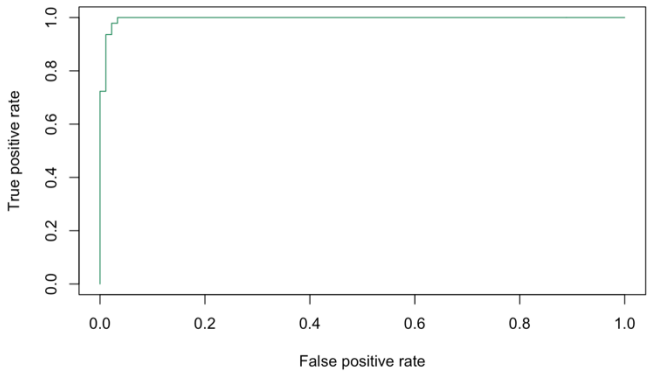

Fig. 10. ROC curve of linear model

B. Random Forest (RF)

We will consider the performance of five models corresponding to milestones throughout the RF methodology. The specifications and results of these models are captured in the following tables:

	Tuning Style	Sampling	Ntree/ Mtry	Cut-off
Default	*No tuning*	*Default*	*500/3*	*0.5*
Original	*Sequential*	*Default*	*1000/2*	
Oversampled	*Sequential (following 'Original')*	*Oversampling*		
Cutoff	*Sequential (following 'Oversample')*	*Oversampling*		*0.31*
Automatic	*Simultaneous*	*Stratified Sampling (50-50)*		*0.43*

Fig. 11. Description of models considered

	Accuracy (%)	Sensitivity (%)	AUC (%)
Default	95.2	91.7	98.9
Original	96.2	94.4	99.0
Oversampled	96.2	94.4	99.0
Cutoff	96.2	97.2	98.9
Automatic	98.1	100	99.1

Fig. 12. Table of results (Random Forest)

We see that the 'Default' RF model has the weakest performance for all metrics, demonstrating the importance of tuning ntree and mtry, although judging by the plateau in the earlier graph of error rate against ntree, it is likely mtry is responsible for this difference in performance. Note that the SVM default model shows slightly better performance with 0.4% higher accuracy and 6.3% higher sensitivity, so SVM may be naturally more appropriate for this dataset, perhaps because SVM is particularly well-suited to two-class problems such as this one.

The AUC values are close to 100% and extremely similar with a range of only 0.2%. A similar pattern was observed with SVM, albeit with a slightly higher range in AUC values of 0.6%. This pattern may be because the dataset's author has already coded the features from 'most benign' to 'most malignant' and AUC measures how well a model distinguishes between benign and malignant observations.

Changing the sampling style doesn't seem to be effective because the 'Original' and 'Oversampled' RF models show identical performance. We would have expected a higher sensitivity for 'Oversampled' because it was trained with a greater emphasis on malignant observations. However, this apparent ineffectiveness is somewhat inconsistent with the fact that the 'Automatic' RF model selected 50-50 bootstrap stratification over the default class ratio when simultaneously optimising parameters.

Cut-off appears to have a powerful ability to reduce false negatives. Both models that reduced cut-off below the default value of 0.5 were able to achieve significantly higher sensitivity at no cost to accuracy (although this does imply a drop in specificity).

Simultaneous optimisation seems to perform better than sequential optimisation because the 'Automatic' model is the best performing for all three metrics. However, we cannot have complete confidence in this model because it unexpectedly performed better on the testing data than the OOB data, unlike all the other models. If we consider OOB performance, the 'Cutoff' model performs better than the 'Automatic' model.

For this reason, we conclude that both 'Cutoff' and 'Automatic' are the best-performing random forest models.

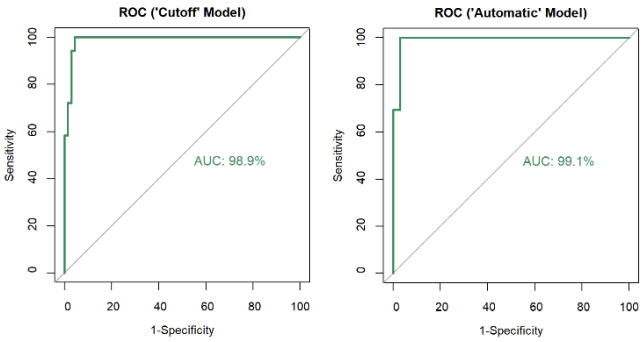

Fig. 13. ROC curves for best two Random Forest models

Our observed results for the 'Cutoff' model of 96.2% accuracy and 97.2% specificity are roughly in line with random forest models discussed in the literature. Vig [2] obtains slightly lower accuracies and sensitivities at 95.64% and 97% respectively, however this is to be expected considering they only used 100 trees whilst we used 1000 trees. On the other hand, Nguyen et al. [6] obtain higher accuracy and sensitivity values of 99.82% and 99.83% respectively. This is surprising since they only used 25 trees. For example, Vig observed a 5% drop in accuracy when moving from 100 trees to 10 trees. Nguyen et al. used the diagnostic version of the breast cancer dataset, whereas we used the original version, however this is unlikely to be the reason for the performance differences because Vig also used the diagnostic version. The results of Nguyen et al. are more in line with our 'Automatic' model which attained a 98.1% accuracy and 100% sensitivity. However, we recognised that these results could be an outlier, which suggests this is also a possibility with Nguyen et al.

We would expect our random forest models to perform better than individual decision trees since these are prone to overfitting. This is the case when comparing our results to Obaid et al. [1] who only achieved a 93.7% accuracy for their best decision tree. However, Lavanya and Usha Rani [5] managed to achieve a 96.99% accuracy using decision trees, beating the 96.2% we achieved in our 'Cutoff' model. This may be because of the advanced principal-components based feature selection method they applied. Nguyen et al. also used an advanced feature selection algorithm that they designed specifically for their study. This provides an alternative explanation of why they achieved better performance than us. All in all, it seems that feature selection can go a long way in improving model performance.

VII. Comparison

At this stage, it is unclear whether SVM or RF performs best on this dataset because we observe that the best SVM model (linear) fits between the two best RF models ('Automatic' and 'Cutoff') in terms of both accuracy and sensitivity. To gain more clarity we perform two follow-up tests on both our best and our default SVM and RF models:

a) Performing 10-fold CV using the same folds
b) Swapping our preprocessed datasets

a) Performing 10-fold CV using the same folds

Splitting a dataset on different indices has a surprisingly big impact on model performance. Therefore, to remove this source of variation and achieve more accurate comparisons, we can use an identical allocation of folds for both SVM and RF. We first notice that the 'Automatic' RF model now has 0.8% lower sensitivity but only 0.1% higher accuracy than the 'Cutoff' model and for this reason, we conclude that 'Cutoff' is our 'best' RF model.

Same Folds	**Support Vector Machine**		**Random Forest**	
	Accuracy (%)	**Sensitivity (%)**	**Accuracy (%)**	**Sensitivity (%)**
Default	96.2	97.9	97.1	97.0
Best	96.5	97.5	97.0	99.5

Fig. 14. Results from follow-up test (a)

We see there is almost no difference in accuracy between the default and best models for both SVM and RF. This can be explained by the earlier AUC values for the default models which were both around 99%. Based on this near-perfect value, there seems to be limited scope for improving accuracy by tuning parameters. However, we do observe an improvement in sensitivity of 2.5% for the best RF model, demonstrating the effectiveness of varying the class distribution and the cut-off value. All the SVM models were also built with a higher weighting placed on malignant observations which explains why their sensitivities are at least 1% higher than their accuracies.

b) Swapping preprocessed datasets

In this test, we build RF models on the preprocessed dataset previously used for SVM and vice versa. This allows us to test which preprocessed dataset yields better models.

Swapped Data	**Support Vector Machine**		**Random Forest**	
	Accuracy (%)	**Sensitivity (%)**	**Accuracy (%)**	**Sensitivity (%)**
Default	96.2	97.1	97.8	95.7
Best	95.2	95.7	97.1	100

Fig. 15. Results from follow-up test (b)

First, it is immediately clear that the conclusions from the first test are supported. We again see no gains in accuracy over the default models for both SVM and RF (indeed we see declines of 1% and 0.7% respectively), however we see a sizeable gain in sensitivity for the best RF model of 4.3%, and we also see that both SVM models have higher sensitivity than accuracy too.

Comparing our best models' performances on the swapped datasets, we notice that SVM now performs worse by 2.1% accuracy and 2.6% sensitivity, whilst RF now performs better by 0.9% accuracy. Before, the default SVM model performed better than the default RF model but now this relationship has reversed. This suggests that the SVM dataset (Method 1) is able to generate better models that the RF dataset (Method 2). Method 1 uses one fewer feature than Method 2, so feature selection is unlikely to be reason for this. Instead, it is likely to either be that PMM is the superior imputation method, or simply the fact that different seed values were used. But if we look back at the first follow up test, we saw the same pattern: SVM performed worse than before whilst RF performed better than before. In the first test, preprocessing was not switched, so it seems that it is the seed value rather than the preprocessing techniques driving the changes relative to the original results.

Based on the results from both follow-up tests, the best RF model shows both consistently higher accuracy and sensitivity than the best SVM model and for this reason we choose it as our final model (model summary: ntree=1000, mtry=2, using an oversampled dataset). However, we build this model based on Method 1 for preprocessing since this gives better performance on the testing dataset than Method 2 (preprocessing summary: 80% training set that removes duplicates, cell shape feature is removed, and PMM is used to impute missing values).

VIII. Conclusion

The aim of this work was to investigate the efficacy of applying the Support Vector Machine and Random Forest classification methods to the problem of breast cancer diagnosis. Through various methods of tuning, we were able to obtain accuracies of 97.3% and 97.8% for the SVM and Random Forest model respectively, both using Method 1 for preprocessing. However, we saw limited gains to accuracy relative to the default model specifications, suggesting limited scope for overall improvement. Despite this, we did observe significant scope to improving sensitivity by using sampling techniques to vary the class distribution and by lowering the cut-off rate. In future work, it would be interesting to explore the effects of varying cut-off on our SVM models as well, although SVM doesn't output probabilities directly so inference techniques such as Platt scaling would need to be applied. Although we have observed some favourable results, this work falls short in regards to the amount of data available for training and testing of models. Future iterations of this work should aim to test these techniques on larger data sets.

References

[1] Obaid, O., Mohammed, M., Ghani, M., Mostafa, S., Al-Dhief, F. (2018). Evaluating the Performance of Machine Learning Techniques in the Classification of Wisconsin Breast Cancer. *International Journal of Engineering & Technology, 7 (4.36), 160-166.*

[2] Vig, L. (2014). Comparative Analysis of Different Classifiers for the Wisconsin Breast Cancer Dataset. *OALib*, *01*(06), 1–7.

[3] Showrov, M. I. H., Islam, M. T., Hossain, M. D., & Ahmed, M. S. (2019). Performance Comparison of Three Classifiers for the Classification of Breast Cancer Dataset. *2019 4th International Conference on Electrical Information and Communication Technology (EICT).*

[4] Polat, K., & Güneş, S. (2007). Breast cancer diagnosis using least square support vector machine. *Digital Signal Processing*, *17*(4), 694–701.

[5] Lavanya, D., Usha Rani, K. (2011). Analysis of feature selection with classification: breast cancer datasets. *Indian Journal of Computer Scinece and Engineering*, Vol. 2 No.5, 756-763.

[6] Nguyen, C., Wang, Y., & Nguyen, H. N. (2013). Random forest classifier combined with feature selection for breast cancer diagnosis and prognostic. *Journal of Biomedical Science and Engineering*, *06*(05), 551–560.

[7] Samieinasab, M., Torabzadeh, S. A., Behnam, A., Aghsami, A., & Jolai, F. (2022). Meta-Health Stack: A new approach for breast cancer prediction. *Healthcare Analytics*, *2*, 100010.

[8] Mushtaq, Z., Yaqub, A., Sani, S., & Khalid, A. (2019). Effective K-nearest neighbor classifications for Wisconsin breast cancer data sets. *Journal of the Chinese Institute of Engineers*, *43*(1), 80–92.

[9] *Predictive Mean Matching Imputation (Example in R)*. (2022, March 15). Statistics Globe. https://statisticsglobe.com/predictive-mean-matching-imputation-method/

[10] James, G., Witten, D., Hastie, T., & Tibshirani, R. (2021). An Introduction to Statistical Learning: with Applications in R (Springer Texts in Statistics) (2nd ed. 2021 ed.). Springer.