THE LONDON SCHOOL OF ECONOMICS AND POLITICAL SCIENCE

Predictive Modelling of the Wisconsin Breast Cancer Dataset

Candidate Number: 36084, 35946, XXXXX

Course: MA429 Algorithmic Techniques for Data Mining

May 3, 2020

Executive Summary

This report looks at the Wisconsin Breast Cancer Dataset [1]. We apply various data mining methods with the goal to accurately predict the classification (Malignant or Benign) of tumourous cells in the breast. We present a literature survey of what has already been done with this dataset and provide a full introduction to the topic of cancerous cells. We then show that various data mining methods can accurately predict the classification of such cells. Specifically, we show that a logistic classifier can achieve a high accuracy rate garnering a recall of 95.31% and a success rate of 97.66%. We also provide the reader with extra information about the method of XGBoost and explain why feature reduction methods combined with a logistic classifier achieve optimal results. Lastly, we instruct the reader how they should proceed with the information in this report for future work on the topic.

Contents

1. Introduction

1.1 Background Information

In order to understand the information presented in this report we must first learn about tumours. Tumours can be divided into two types, malignant and benign. Benign means that the cells are **non-cancerous**, specifically this means that they are unable to spread to nearby tissue or organs and are not considered that dangerous. Malignant tumours are cells that are able to spread to nearby tissue and are **cancerous**. This is what is referred to as *metastasizing*, where the cancer cells spread through the blood or lymph system and form new tumours elsewhere in the body. While the latter is of more concern, it is possible for benign tumours to become dangerous as time passes. They can grow to very large sizes and are particularly threatening when detected in places like the brain or colon. This is why people often have them removed despite them being non-cancerous. [2]

Tumour cells are able to form in any part of the body, while this report will focus primarily on tumours found in the breast, our findings could shed some light on tumours found in other parts of the body. Something that we are going to be particularly interested in is the size and geometry of the cells. As already indicated, tumours can spread and grow in size and a simple observation is that the larger the tumour the more dangerous is it is. A large malignant tumour would suggest the spread of cancerous cells to multiple parts of the body and is obviously very dangerous and often results in fatality.

Breast cancer cells are most commonly found in women, they are detected usually by a woman finding a lump on her breast or confirmed by x-ray imaging. Whilst it is more common in women, men can still get breast cancer it is just far less likely. To compare, according to the American Cancer Society men are 0.12% likely to get breast cancer whilst women are at much larger risk at 13%. [3]

Due to the large amount of people this puts at risk, it is vital that we learn more about the key indicators of breast cancer as this helps detect it much sooner and as a result will directly save lives.

The dataset we are looking at originates from an experiment in 1993 [4] by the University of Wisconsin and the features are extracted from digitized images of a *fine needle aspirate* (FNA) of a breast mass [1]. These features describe characteristics of the cell nuclei present in the image. FNA is a type of biopsy whereby doctors remove a

chunk of tissue and examine it for disease. We can see this kind of image in Figure 1.1.

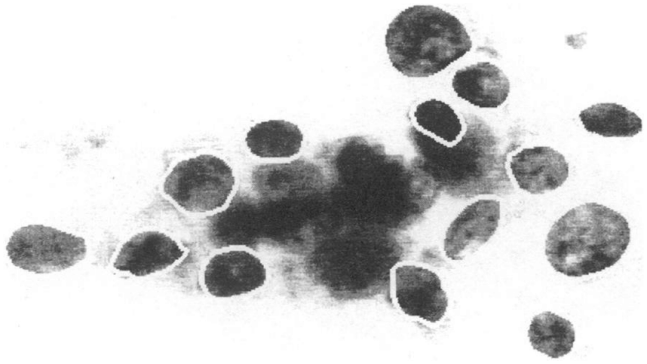

Figure 1.1: A magnified image of a malignant breast fine needle aspirate

The extraction method of the data is out of the scope of this report and the reader should consult the paper by K. P. Bennett and O. L. Mangasarian [5] if they wish to know more.

1.2 Literature Survey

A key thing to point out about this dataset is that it is widely studied with great success and people have achieved over 90% accuracy with various classifiers. In light of this, a good question to ask is what is making this small number of training examples slip through the net?

This report will be focusing on investigating models which fall in to either of the following categories: **i)** models that have various hyperparameters to tune; **ii)** more advanced models or variations of models that have been tried and tested. By a model having more hyperparameters we can investigate how changing these hyperparameters may affect the classification rate, meaning that something like a neural network classifier is primed for use in this report.

At first sight this dataset seems to have a large number of overlapping or correlated features due to most of them being related to the geometry of cells, meaning that feature transformation could be of good use here.
Principal Component Analysis (PCA) seemed to work fairly well in most cases but there were other methods that seem to yield better results when data mining methods were applied. In general there were various other methods that were successful at reducing the set of features to 5 or less. *Univariate feature selection* or *recursive feature elimination* were techniques that seemed to work well and we may utilise them in this report.

2. Pre-Processing and Input Data

2.1 Data Cleaning

The dataset used in this project is the Wisconsin Diagnostic Breast Cancer (WDBC) Dataset, it consists of 569 observations and 32 different features. The first two features are ***ID*** (patient unique identification number) and ***diagnosis result*** (benign or malignant). The other 30 features comprise of statistics about the geometry of the cells. These include average, standard error and the worst value of the ***radius, perimeter, area, smoothness, compactness, concavity, number of concave points*** as well as the mean, standard deviation and maximum value of the ***symmetry and fractal dimension***. Something to note is that these values are all positive numbers, while looking at the mean values, we can get a general view that the mean values for radius of cells is within the range 6 and 29 micrometers, which is larger than the normal cell radius(1-20 micrometers). Meanwhile, the mean value for the texture also has a relatively wide range, which is within 9 and 40. We can deduce from this that the range for different features is quite different from normal values due to the existence of malignant tumour cells.

(Further details can be found in the appendix and Section 3.2)

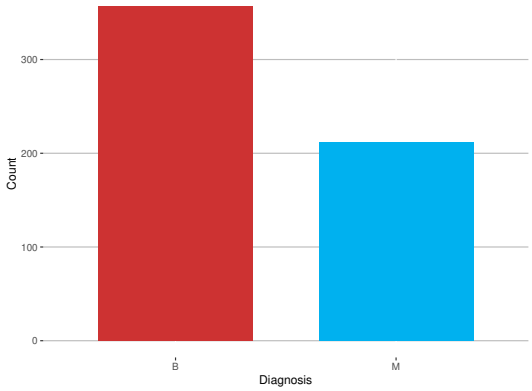

Figure 2.1: The number of diagnoses in each class

Firstly, we want to check the influence of missing values. It can be found through querying the dataset for NA values that the number of missing values is 0. In general, the data dimension matches the description, which means we do not need to remove any missing values in this project. Figure 2.1 shows that the response variable has only two values B (Benign) and M (Malignant). According to the description, the

response variable has no abnormal values.

Next, we check whether outliers exist in the mean of ten variables. From the density curve in Figure 2.2, we can find that for these ten different observation indicators, all records are greater than 0, of which *radius*, *texture* and $fractal\_dimension$ are approximately normal distributed. The other variables are right-biased, without obvious anomalies. We can also see that we get similar results for the standard deviation data and the worst value of 10 different features in Figure 2.3 and Figure 2.4.

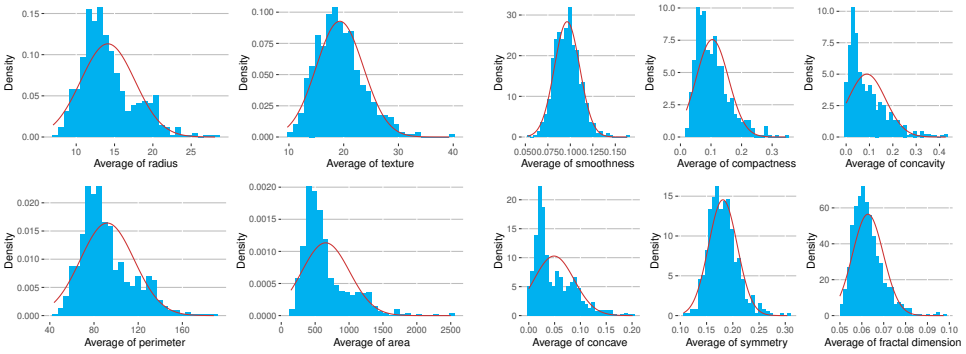

Figure 2.2: The average of 10 different features

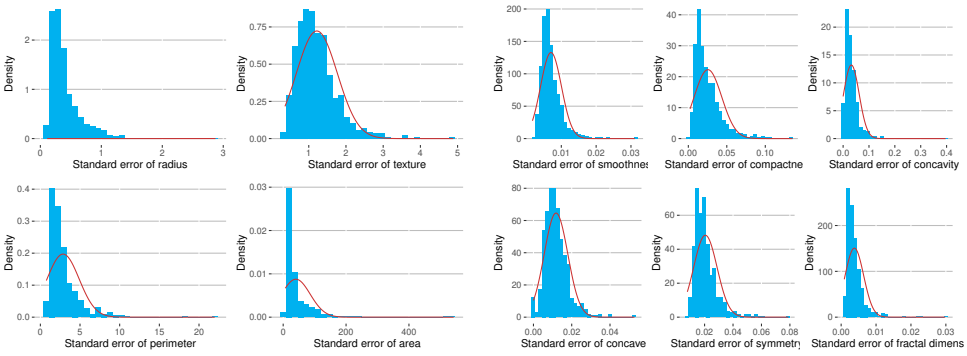

Figure 2.3: The standard error of 10 different features

To conclude, it can be found that there are no obvious outliers and missing values in this dataset, indicating that all features are available for use. We will however omit ID as it is the unique identification number of a patient as it gives no useful information for our analyses. We shall retain all other variables.

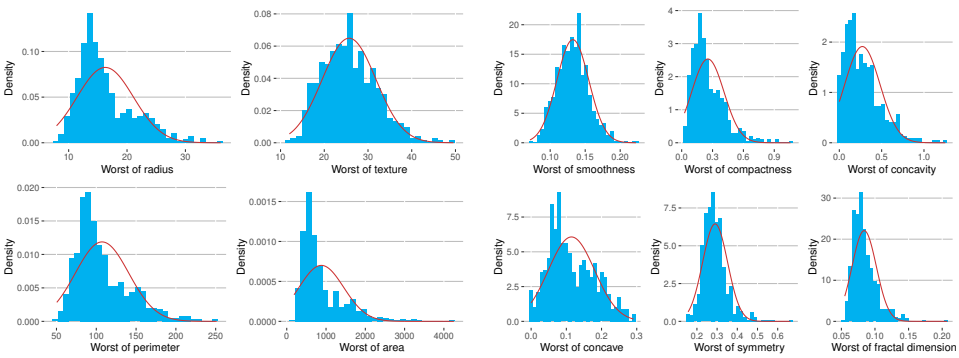

Figure 2.4: The worst of 10 different features

3. Preliminary Analysis

3.1 Hypotheses

Due to the nature of malignant tumours, it would not be unreasonable to expect in many cases features included in this dataset such as *radius*, *area* and *perimeter* to indicate malignancy. As previously indicated malignant tumours are prone to spreading and in fact the literature suggests size could be a great factor for determining the type of tumour as shown in the paper by Erasmus et. al, "Small size and smooth, well-defined margins are suggestive of but not diagnostic for benignity." [6]

Another feature in the dataset is *smoothness*, which represents the local variation in radius lengths of the tumour. According to the same paper by Erasmus et. al, one would expect that tumours with lower values of smoothness could potentially indicate that the tumour is benign.

We could also consider the feature *texture* and it's importance in differentiating between benign and malignant tumours. Benign tumours do not tend to invade neighbouring cells, hence we expect them to be more tightly packed and expect less variation in their greyscale values which is what the feature represents. We have also touched on another point here, due to the invasive nature of malignant tumours we might expect them to be of irregular shapes in comparison to benign tumours. This might mean the feature *concavity.points*, when large could be a good indicator of malignancy, as it measures the number of concave points in the tumour.

Lastly, the when the feature *compactness* is large it might turn out to be a good indicator of malignancy, as it was shown to have good predictive value in the paper by RM Rangayyan and colleagues [7].

One thing to consider is that whilst many of these measurements might suggest that a tumour is either malignant or benign, a common theme in the literature was that such a measurement should not be the only factor a diagnosis is based on. For example, if we find clustered at an extreme of one of the features in our dataset, this does not mean that benign tumours can't also crop up with a similar measurement or that at the other extreme we will find tumours solely of the other classification. This means that we should approach blanket conclusions about the data with caution, especially when the sample size is most likely not large enough to support such claims.

3.2 Basic Analysis

Before doing any comprehensive analysis by machine learning methods, we can do a basic analysis on the data set to check whether our initial hypotheses are plausible.

For the cell *radius*, the average value of the *radius* of malignant tumour cells is generally larger than that of benign tumour cells, and the worst values are also significantly higher in malignant tumour cells. Also, from Figure 3.1, we can find that the fluctuation of cell *radius* is relatively large in malignant tumours. We can check whether our hypothesis related to the size of tumours is true after analysing the same way for *area* and *perimeter*.

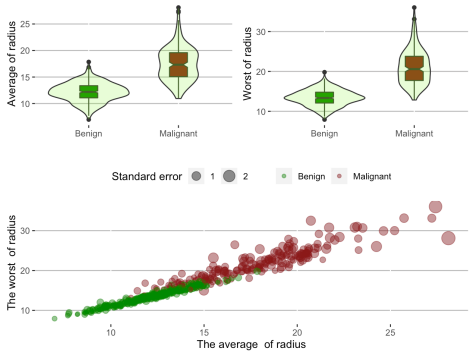

Figure 3.1: Radius

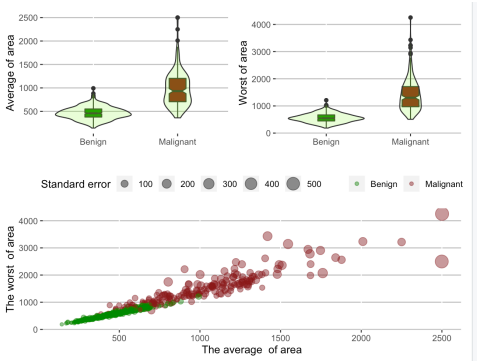

Figure 3.2: Area

The same goes for *area*, from Figure 3.2, it can be found that the average *area* and worst values for *area* of benign tumour cells are lower than those of malignant tumour cells. Meanwhile, the area fluctuation of malignant tumour cells is significantly higher than that of normal benign tumour cells.

In addition to this, the *perimeter* of the cells is explored and analysed, Figure 3.3 shows that the average *perimeter* of malignant tumour cells is higher than that of benign tumour cells, same for the worst values and fluctuations. In general, the *radius*, *area* and *perimeter* of malignant tumour cells are higher than those of benign tumour cells. These observations indicate that our hypothesis for the size part is true, it is more likely for malignant tumour cells to spread.

Then we need to check the correctness of the hypothesis for the *smoothness*. The difference in *smoothness* is not as obvious as the cell size, although to some extent, we can say the malignant tumour cells have higher values of *smoothness* by observing the graphs for the mean of average and the worst values for *smoothness*, which can be matched with our hypothesis. But according to Figure 3.4, the extreme value for the average of benign tumours is higher than that of malignant tumour cells. As the volatility is not obvious, more observations are required for accurate analysis.

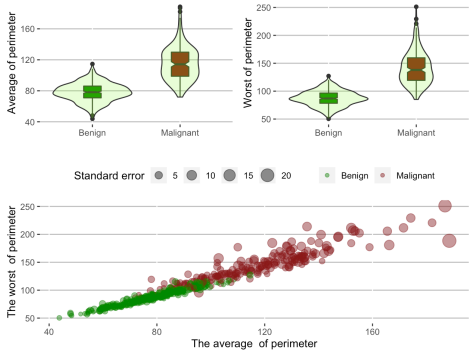

Figure 3.3: Perimeter

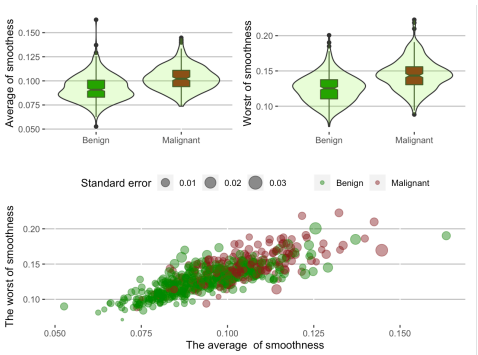

Figure 3.4: Smoothness

Our hypothesis for the *concavity* says that large *concavity* can be a good indicator of malignancy. Analysing in detail, from Figure 3.5, we can deduce that the average *concavity* degree of malignant tumour cells is higher than that of benign tumour cells, and same for the worst values, while this can be matched with our hypothesis for *concavity*. However, from the average of *concavity*, the scatter plot of the extreme average value indicates that benign tumour cells may also have high severity of concave portions. Although comparing with malignant tumour cells, the fluctuation for the average of *concavity* of benign tumour cells is low, which means its concavity extent can not be maintained at a high value. To check whether our hypothesis for the *concavity* is completely right, it is necessary to observe more cells to evaluate the volatility.

Meanwhile, the hypothesis for the $concave_points$ can be verified by observing Figure 3.6. The proportion of $concave_points$ in malignant tumour cells is significantly higher than that of benign tumour cells, same for the worst values, which means number of $concave_points$ may be a better indicator than the concavity extent.

Next, to evaluate the *compactness* between cells, Figure 3.7 indicates that the average value of the *compactness* of malignant tumour cells is generally higher than that of benign tumour cells, and same for the worst values. Meanwhile, based on the scatter plot, our hypothesis for the *compactness* is true, this feature can be used as an indicator to distinguish malignant tumour cells and benign tumour cells.

Finally, we can check the hypothesis for the *texture*. Although many reports state that texture is an important feature for differentiating between benign and malignant tumours, our data shows that there is no significant difference in both average and worst values of *texture* for different kind of tumour cells. This result conflicts with our initial hypothesis, we need explore more data sets for ensuring whether *texture* can be a good indicator.

Combined with the above analysis of cell radius, area, perimeter, smoothness, con-

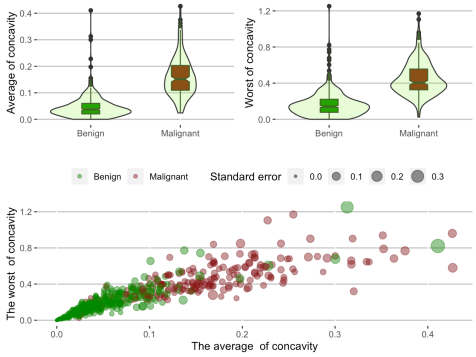

Figure 3.5: Concavity extent

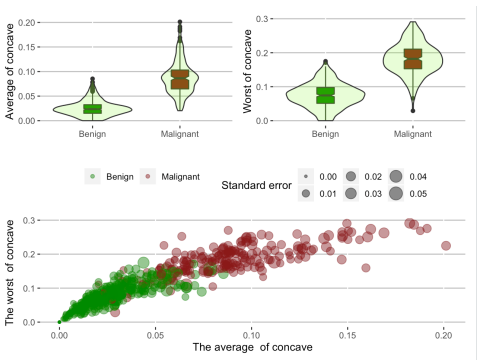

Figure 3.6: Concave points

cavity extent, concave points, compactness and texture, malignant tumour cells have higher *radius*, *area*, *perimeter* and *compactness* and more *concave.points* than benign tumour cells. A comparison of these indicators with normal values can be used to make a guess about the tumour type. For the *smoothness* and *concavityextent* and *texture* indicators, although to some extent the hypotheses are right, the volatility is not obvious, especially for texture. We need more observations to measure their volatility. But in general, the above analysis can support the hypotheses made in this project.

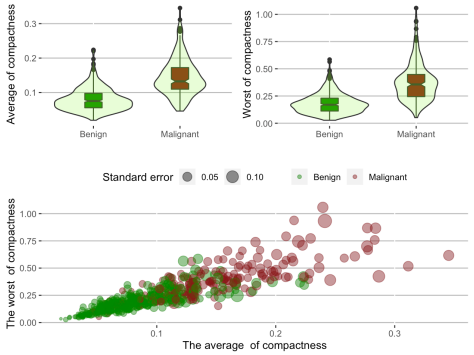

Figure 3.7: Compactness

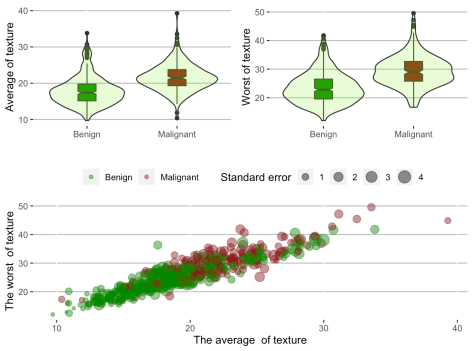

Figure 3.8: Texture

Analysing the inter-correlations between different features can help us have a better understanding of the data set. The correlation analysis shows that *radius*, *perimeter* and *area* of the cell are positively correlated with each other, which is understandable as these three features are all correlated with the cell size. While from Figure 3.9, based on the average value, we can find that these three features related to the cell size also have a significant positive correlation with the *concavity*, which means the larger the cell is, the greater the severity of concavity would be. This supports that although the volatility is high when illustrating the concavity extent, this feature can be used as an indicator to some extent. For *smoothness*, its average values are positively correlated with compactness, symmetry and the cell's fractal dimension,

which means cells which are compacted together are more likely to be smoother. As compactness can be a good indicator for us, we can also use smoothness as an indicator as well. However, for texture, it is not related to any other features, we still need more observations for determining whether this feature can help us distinguish benign and malignant tumour cells.

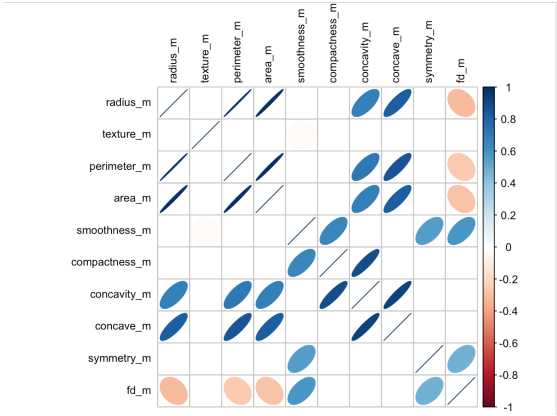

Figure 3.9: Correlation for mean values

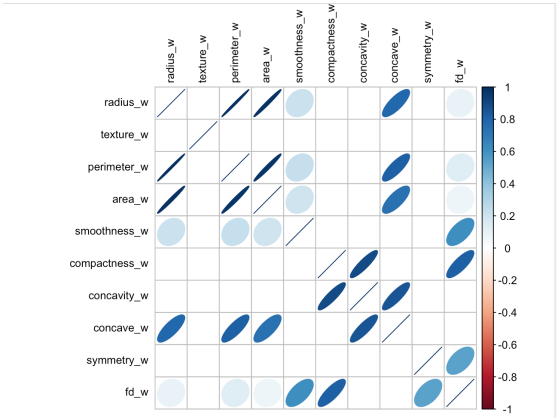

Figure 3.10: Correlation for worst values

4. Experiments with Data Mining Methods

4.1 Performance Metrics

The performance metrics should be chosen based on the characteristics of any specific data set. Here we have a binary classification problem and the aim is predicting the type of the tumour which can be either benign or malignant. Breast cancer is a serious health disease and a wrong diagnosis can create huge and unrecoverable impacts on people's lives. Thus, the accuracy of the predictions is of great importance. There are many different ways to comment on the accuracy of a model for binary classification problems and we will now outline them in this section.

Confusion Matrix

The confusion matrix is one of the most suitable methods for the performance metrics of binary classification problems. The confusion matrix gives information about the error rate, success rate (accuracy), precision, sensitivity and selectivity. We can calculate all of these performance metrics with the number of true predictions of malignant and benign and false prediction of malignant and benign. All of these predictions can be observed from the confusion matrix (displayed on the next page), the number of true and positive predictions are shown in the confusion matrix. When implementing data mining methods your goal would be to maximise or minimise some of these metrics. For example, analysts would usually like the success rate, which could also be thought of as the classification accuracy of the model, to be maximised as it can give a fast understanding of how well the model performed. This can be calculated by dividing the number of correct predictions by the total number of predictions made. However, if there are not approximately equal number of samples from each classification in the training data set, it is not reasonable to use it. The training data is not very imbalanced so there is no need to worry about this issue.

Considering the type of data at our disposal, it would be most sensible to maximise recall as detecting malignant tumours is of greater importance. This is due to the fact that they are more dangerous and wrong diagnosis could end in fatality. Furthermore, another important evaluation here is the precision, which shows what percentage of malignant diagnoses are correct, this is important for similar reasons as recall. Moving forward, recall, precision and selectivity are chosen as the main concerns in this project and we shall combine these to calculate various other statistics in which

their use is outlined in the next subsections. The measurement rates formulas are below.

	Predicted Benign	Predicted Malginant
Actual Benign	TN	FP
Actual Malignant	FN	TP

Table 4.1: The Confusion Matrix

$$Error\ Rate = \frac{FP + FN}{TP + TN + FP + FN}$$

$$Success\ Rate\ (Accuracy) = \frac{TP + TN}{TP + TN + FP + FN}$$

$$Recall\ (Sensitivity) = \frac{TP}{TP + FN}$$

$$Selectivity\ (Specifity) = \frac{TN}{TN + FP}$$

$$Precision = \frac{TP}{TP + FP}$$

Error Rate: The percentage of false diagnoses to the total diagnoses.

Success Rate: The percentage of true diagnoses to the total diagnoses.

Precision: The percentage of true malignant diagnoses to the total malignant diagnoses.

Recall: The percentage of people who have malignant tumours and are diagnosed correctly.

Selectivity: The percentage of people who have benign tumours and are diagnosed correctly.

F1-Score

F1-Score combines recall and precision rates, the most important rates for this problem, so the F1-Score is an important statistic to choose the best model and we would hope to see it maximised. The formula for F1-Score can be observed at from the below equation.

$$\text{F1 Score} = \frac{(2 * Precision * Recall)}{(Precision + Recall)}$$

ROC (Receiver Operating Characteristics) Curve

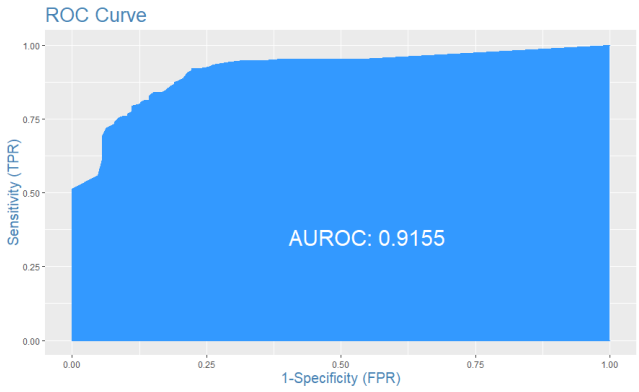

Figure 4.1: The ROC Curve Example

The ROC Curve is a beneficial method to check and visualize the performance of a classification model. ROC Curve combines recall and selectivity rates thus the ROC curve is plotted with Sensitivity against the Specificity. As it can be seen from the Figure 4.1 Sensitivity is on the y-axis and Specificity is on the x-axis. The area under the ROC curve is used to evaluate the efficiency of the model. The model is better at distinguishing between the malignant and benign tumours when it has higher AUROC which is the area under the curve in the ROC Curve Figure 4.1. It gives the optimum results that we can get with a model. A way of finding the optimum rate of that model would be the changes on the cut-off/threshold in the probability percentages for defining a tumour as a malignant. For example in a normal way of prediction we assign the tumour type to malignant if it has 50 percent or more probability. If the probability is less than 50 percent then the model classifies it as benign but it is not always efficient to do it like this. Hence, the ROC Curve gives the optimal efficiency of the model when changing the threshold of 50 percent to a lower or larger threshold to increase the sensitivity and specificity.

Kappa Statistic

The Kappa Statistic takes into account the chance probability which may be found by a classifier making random predictions about observations in our data set. A high kappa equal to 1 would indicate that there is no way our classifier could achieve these results by chance, which would increase our confidence in our model. A kappa equal to 0 would mean that our classifier is equivalent to making a guess about each observations classification, and would indicate that our model is ineffective.

$$\kappa = \frac{(Success\ Rate - Chance\ Probability)}{(1 - Chance\ Probability)}$$

$$Chance\ Probability = (TN + FN) * (FP + TN) + (FP + TP) * (FN + TP)$$

4.2 Decision Tree

Decision tree, as the name shows, is a tree-like model of decisions. It is an important predictive modeling approach which covers both classification and regression. This method is suitable for medical analysis as the result can be interpreted easily and the importance of variables which can influence the medical result can be visualised and compared. This means that by applying this model, not only can we predict whether the breast cancer is benign or malignant, but also we can know the set of variables which are more influential for the predictive result. Meanwhile, this model can be calculated efficiently by the computer, which means we can save much computational time.

However, it is inefficient for us to show the entire tree due to the large number of variables. We need to choose the optimal tree size to make the results clearly. As for decision tree model, the calculation time is acceptable, so application of dimensional reduction methods such as PCA are not necessary. In order to find the optimal tree size without deteriorating the accuracy, we can apply model parameter tuning by changing the complexity parameter *cp* value with k-fold cross validation. We also tried to apply model parameter tuning on a neural network, although this method can help us determine the parameters such as number of neurons with more accurate predictive result, it cost us longer than 6 hours for calculation. This however can work quite well for decision tree model due to it's simplicity.

Cross validation is a resampling procedure, which can help us evaluate the models. The parameter k in cross validation refers to the number of groups that the original dataset is split into. Here we choose k equals 10 as this value can more likely give us a low variance through the finding of many experiments. The complexity parameter can exactly help us select the optimal tree size, as when comparing the value of *cp*, if the cost of adding another variable to our decision tree is higher, then we just stop. Here we choose $Giniindex$ as the cost function, as the equation shows here, where pi is the probability of an object being classified to a particular class.

$$Gini = 1 - \sum_{i=1}^{n} (p_i)^2$$

Firstly, we try to tune the model by setting the *cp* between 0.001 to 1. Figure 4.2, 4.3, 4.6 shows that the worst value of concave is the most important variable here. The accuracy is the highest for the *cp* value between 0.01 to 0.8, as this range is still large, we can try and tune this model again.

Then, we try to tune the model again by setting the *cp* between 0.001 to 0.8. Figure 4.4, 4.5, 4.7 shows that the variable $area\_worst$ is the most important one in this case. These two decision trees very similar as the they have an overlap for *cp* values

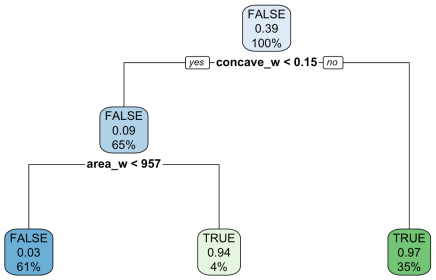

Figure 4.2: Decision tree

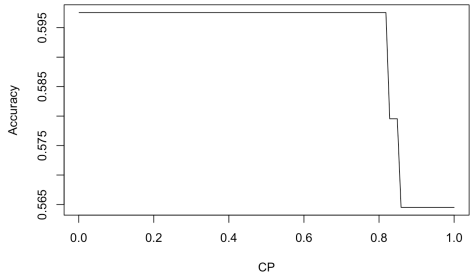

Figure 4.3: Accuracy vs. cp

which can lead to the highest accuracy in each case. The construction of decision trees is depending on whether it can distinguish benign from malignant to the largest extent.

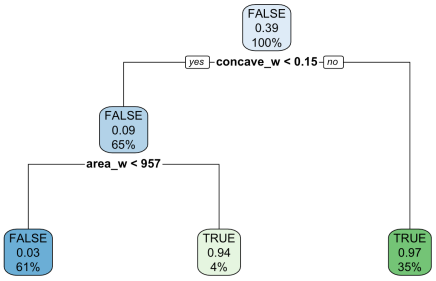

Figure 4.4: Decision tree

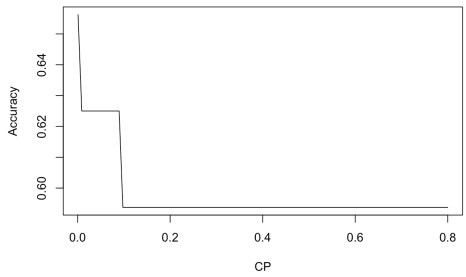

Figure 4.5: Accuracy vs. cp

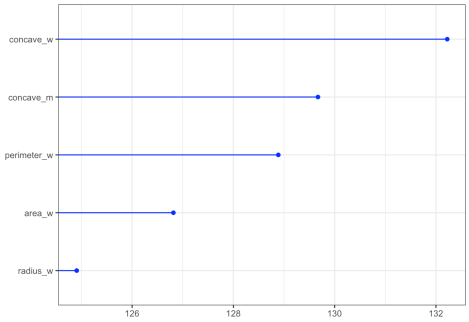

Figure 4.6: Importance (first tune)

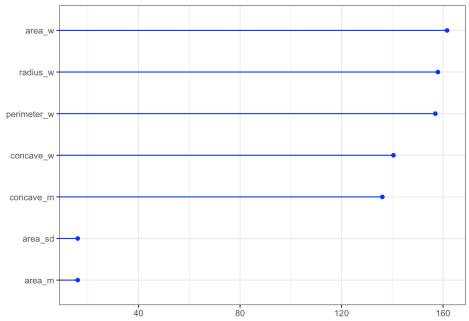

Figure 4.7: Importance (second tune)

At this stage, by comparing the decision tree model with the first parameter configuration and the second configuration, we can find that the error rate of classification with the second configuration is lower.

The evaluation parameter including recall,precision,kappa-statistic and F1 score all supports that the decision tree model can be enhanced after with the second set

of parameters. Especially, a high recall is very important for medical diagnosis, a prediction with high false negatives may lead to severe results. So appropriate model tuning is important for improving the accuracy of a decision tree model. We should mainly consider the results from the second configuration.

Table	Misclassification	Recall	Precision	κ	F1 Score
first tune	0.1579	0.8182	0.7419	0.6499	0.7782
second tune	0.0877	0.8364	0.8654	0.7961	0.8506

4.3 Gradient Boosted Trees

Gradient Boosting is simply a generalisation of the AdaBoost algorithm that we were introduced to in the lectures for MA429. In AdaBoost we used weak learners (stumps) to make a prediction about the classification of each observation. In each round of AdaBoost we were updating weights that increased the importance of certain training examples that were misclassified and decreased the weights of training examples it got right. Adjusting the weights in each round is essentially optimising a loss function, and in AdaBoost the loss function being minimised is the exponential loss function [8]. We outline the algorithm below.

Algorithm 1: Gradient Tree Boosting Algorithm

1. Initialise model with $\mathrm{F}_0(x) = \arg\min_{\gamma} \sum_{i=1}^{n} L(y_i, \gamma)$

2. for *m = 1 to M:* **do**

i) Compute $r_{im} = -\left[\frac{\partial L(y_i, F(x_i))}{\partial F(x_i)}\right]_{F(x)=F_{m-1}(x)}$ for $i = 1, ..., n$

ii) Fit a regression tree to the r_{im} values and create terminal regions R_{jm} for $i = 1, ..., J_m$

iii) For $j = 1, ..., J_m$ compute $\gamma_{jm} = \arg\min_{\gamma} \sum_{x_i \in R_{ij}} L(y_i, F_{m-1}(x_i) + \gamma)$

iv) Update $F_m(x) = F_{m-1}(x) + \nu \sum_{j=1}^{J_m} \gamma_m I(x_{jm})$

end

The loss function that we are using is the logistic loss.

For an interpretation or walk-through of this algorithm applied in the binary classification setting please see the appendix.

XGBoost

In our experiment we shall be using the XGBoost (extreme gradient boosting) flavour of gradient boost [9]. Traditional gradient boost has a weak point, being that we are allowed to build weak learners that are trees instead of the stumps we had in AdaBoost. The trees therefore can grow to large depths and overfit the training data easily. Thus, XGBoost adds a regularisation parameters λ and Γ to stop the model from overfitting. Specifically what we mean by this is that step iii) of the loop in Algorithm 1 is changed to factor in a penalty term when calculating the output values, as well as Γ encouraging tree pruning. Another positive of XGBoost is simply that it's faster due to the XGBoost framework having a scalable, distributed implementation whereby you can run analysis on data in parallel on multiple machines or CPU cores.

Hyperparameters for XGBoost

η	Γ	λ	nrounds	max_depth	min_child_weight	subsample	colsample_bytree
0.1	1	0	90	9	2	0.8738	0.9149

The majority of these hyperparameters were found after extensive tuning. To justify a few of them, we chose 0.1 for the learning rate η as it seemed the model learned too fast when we had it at the default of 0.3, dropping it slightly to 0.1 made the model gather more nuances in the data. We arrived at setting nrounds to 90 as the log-loss errors seem to converge around this number of iterations, i.e. there wasn't much change in the models predictive performance after a threshold of 20 iterations past the 90. We chose 1 for the value of γ as when trying to find the optimal number of rounds using the XGBoost package's built in cross validation function, the log-loss (error) dropped off very fast for both training and test sets. The convention for this is to add in some regularisation if the log-loss error for both train and test set does not fall at a similar rate. The reason for this is that it's a sign of overfitting.

We have a similar story also for *min_child_weight*, when we had the default value of 1, the log-loss would demonstrate characteristics of overfitting so we added in this form conservatism to avoid this. With all of the previously mentioned fixed, we used a random grid search to hypertune the rest of the parameters. More details about XGBoost's hyperparamters can be found in the documentation. [10]

Results

Firstly, let's see which features XGBoost found particularly useful for splitting up the data when building trees. We can see in Figure 4.8 that overwhelmingly *radius_w* was the best at splitting the data. This followed by *concave_w*, *perimeter_w*, *area_w* and two *texture* measurements all having significant contributions. After a brief inspection of the dataset, this is no surprise. If you simply sort the columns of these features by magnitude you can see that at either extreme we clearly have some

obvious clustering of a certain tumour type. For example, if you look at *radius*, the first 131 examples in ascending order are all benign and looking at the top there is a sea of malignant classifications! To be precise the top 124 are all malignant. When you consider the size of the dataset at 569 instances this is pretty significant. However, this is not at all surprising when you consider the nature of malignant tumours and their ability to invade neighbouring cells. You would expect them to have their worst radius generally higher than benign tumours.

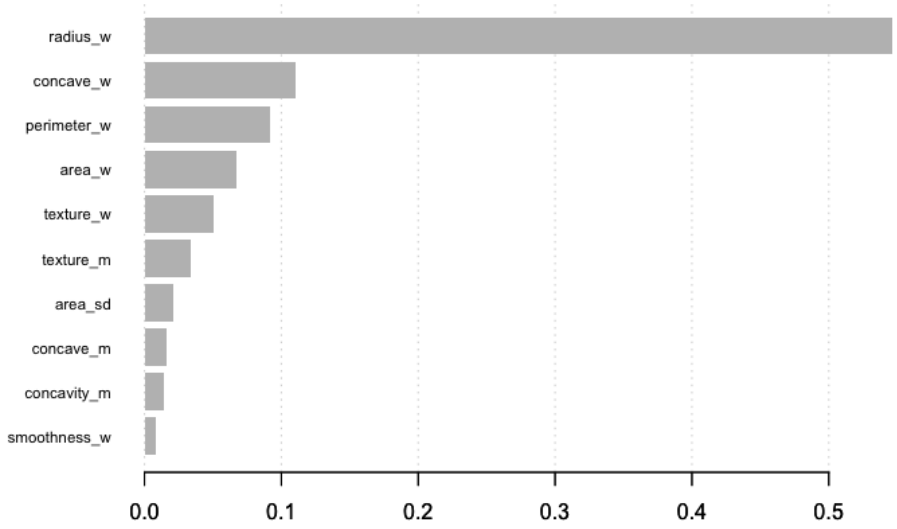

Figure 4.8: Feature Importance in XGBoost

What resonates with our hypotheses about the data is that there is still the odd benign or malignant tumour amongst what it appears to be a clustering of classifications. The tumour that often broke the pattern was then followed by another string of the opposite classification. Highlighting the fact that we do see these kind of outliers and using single indicators for diagnosis should be approached with caution. A similar theme can be seen for the other features XGBoost found important. An insight to takeaway here is that, if a tumour seems to be large in size and of irregular shape then is it absolutely imperative that the patient be taken for further checks.

With our initial configuration of the model we managed to achieve an accuracy of 95.8% on on our one test set. After tuning the parameters of our model we managed to see an improvement with the mean cross-validated prediction accuracy of 97.2%. The results with respect to our evaluation metrics can be seen in the table below.

Recall	Specificity	Precision	κ	F1 Score
0.9773	0.9636	0.9636	0.9409	0.9704

4.4 Logistic Regression

Logistic regression is one of the most popular methods in binary classification problems and is very useful for seeing the significance of certain features. The logistic regression model finds coefficients for each feature from the training dataset. The predictions are the probabilities of being class 1 or class 2. The classes can be numerical, factorial or boolean. Based on the probabilities and the thresholds, we assign the predictions to one of the classes. Normally the threshold is 0.5 but it can be changed to increase the accuracy of the predictions.

The Brief Mathematical Explanation

In this project we want to predict the type of the tumour. In the test data we have the features/variables to make a prediction about the type of the tumour. Let us call the variables x and the predictions y. If we want to find the probability of the tumour to be malignant with the features x we need to calculate $P(y = "M")$.

We have used the coefficients which we find from the logistic regression model to make predictions about the output with the new variables $x_1, x_2...x_n$. The coefficients are shown as β. The first β is the intercept and the rest are coefficients of the variables. The probability of a tumour being malignant is calculated in the following equation. The probability calculation for the prediction is done based on the given information x. If the probability is bigger than the threshold then we predict the tumour as malignant if it is less than the threshold, we predict the tumour as benign.

$$p(X) = \frac{e^{\beta_0+\beta_1\times x_1...+\beta_n\times x_n}}{1+e^{\beta_0+\beta_1\times x_1...+\beta_n\times x_n}}$$

Feature Selection and Dimension Reduction

When the model is applied with 30 variables the model gives an "algorithm did not converge" warning and the p values of all the variables are 1 or 0.999 which possibly means the model is over-fitting so we look to apply feature selection and dimension reduction methods to apply the models. In the examples of the application of logistic regression to this breast cancer data set, the mean columns of the features were taken and the dimension reduction is applied. We now outline some of the different feature selection methods that have been tried.

One of the most popular feature selection method is the recursive feature elimination method. The method eliminates the variables recursively, until reaching a pre-specified number of variables. In the R package of the recursive feature elimination function we can decide on the iteration number instead of specified number of

variables. The number of iterations to eliminate the variables is decided as ten. The results of the ten iteration the model selected 12 variables from the data set. The selected variables are:

area_w	concave_w	perimeter_w	radius_w
texture_w	concave_m	area_sd	texture_m
concavity_w	smoothness_w	concavity_m	area_m

In addition to this, we also tried applying principal component analysis to reduce the dimension of the variables. First of all, the columns are normalized to prepare the data set for PCA. Firstly, PCA was applied to all the features in the data set. PCA results show the each principal component capturing a certain amount of variance and are a linear combination of all of the variables. In our dataset, the first 5-6 principal components can account for the majority of the variance in the results and it is enough to use these components in the prediction models instead of 30 variables. In every component, there is a weight for each variable and the variable values are multiplied with that weight and then these multiplications are summed up like in the equation below to find the component value.

$$z_i = w_{i1} \times x_1 + w_{i2} \times x_2 + ... + w_{ip} \times x_p, \qquad i = 1, 2, ..., q$$

Normally the PCA produces 30 principal components for a data set which have 30 variables but the applier of the model can decide on which components to use to make dimension reduction because as it mentioned, most of the variance is captured by only a few of the components. PCA is also applied to partial variables which are separated as mean columns, standard deviation columns and worst case columns. In each of these sets, the first three principal components are taken and explain at least 80% of the variance. For example the first principal component is the one which explains the variance with the best combination of the variables and in the first PCA model with 30 variables, PC_1 explains the 45.84% of the variance.

Results

Logistic regression is applied to the following: total variables, separated mean variables, separated standard deviation variables, separated worst case variables. Based on the results, the model applied again to the features with high importance. If the p-value of the feature are smaller than 0.05, the feature is called significantly important. Secondly, the features, which have found from the recursive feature elimination, were used to apply the logistic regression. The results are in the Table 4.2. It can be observed that the models which are created with the features that are deemed significantly important have similar accuracy levels to the raw

data sets. For example for the data set with the mean variables has 0.8906 recall and in the model which only contained the mean features with high significance ($texture\_m, smoothness\_m, area\_m$), has 0.875 recall level which is very similar. The results are measured based on the optimal cut-off (optimal thresholds) so these accuracy levels are the optimum for each model. The model with worst case variables and the model with the features which we gathered from recursive feature elimination method give the best results in that part. The significant variables between mean variables are: $texture\_m, smoothness\_m, area\_m$. The features which is significantly important for every data-set can be observed from Table 4.2.

	Success Rate	Precision	Recall	Specificity	Kappa	F1-Score	ROC
30 FEATURE	0.9590	0.9384	0.9531	0.9626	0.9128	0.9457	0.9572
10 MEAN FEATURE	0.9473	0.9661	0.8906	0.9813	0.8858	0.9268	0.9783
texture_m, smoothness_m, area_m	0.9415	0.9655	0.875	0.9813	0.8727	0.9180	0.9759
10 STD DEV FEATURE	0.9239	0.9473	0.8437	0.9719	0.8340	0.8925	0.9558
radius_sd, area_sd, compactness_sd, fd_sd	0.9298	0.9482	0.8593	0.9719	0.8472	0.9016	0.9629
10 WORST CASE FEATURE	0.9766	0.9838	0.9531	0.9906	0.9497	0.9682	0.9892
texture_w,smoothness_w, concave_w	0.9181	0.9464	0.8281	0.9719	0.8207	0.8833	0.9691
10 FEATURE RFE	0.9649	1	0.9047	1	0.9230	0.95	0.9941

Table 4.2

The logistic regression, which was applied to all the variables, overfits and in the results the significance level of all the variables are 1 or 0.999 which means the variables are not effective with regards to prediction. However, this is not true, it happens because of the overfitting so it is decided to apply the Principal Component Analysis. The first model is for the first 10 principal components of the all 30 principal component which it was gotten from the all variables. The results can be observed from the Table 4.3. The rates are measured based on the optimal cut-offs. The 30 principal components results are not good because again there is the problem of overfitting. When the logistic regression was applied to 10 principal components of 30 variables, it gave a better result. The best models are mean and worst case variables with 3 first components. The first 3 components of the mean variables PCA model explain 89.294% of the variance and the first 3 components of the worst case variables PCA model explain 86.543% of the variance. As it can be observed from the Table 4.3 the rates for less components give better results because the main purpose is getting the components which capture significant amounts of the total variance. The additional components which don't have significant impact may misguide the model and cause and could be seen as noise in the data.

The best logistic regression model is the for the first 3 principal components of the

	Success Rate	Precision	Recall	Specificity	Kappa	F1-Score	ROC
30PC/30PC Total	0.8011	0.6944	0.8064	0.7981	0.5842	0.7462	0.8023
10PC/30PC Total	0.8947	0.8666	0.8387	0.9266	0.7706	0.8524	0.9362
10PC/10PC Mean	0.9239	0.8888	0.9032	0.9357	0.8360	0.896	0.9590
3PC/10PC Mean	0.9590	0.8985	1	0.9357	0.9135	0.9465	0.9880
10PC/10PC Std Dev.	0.5087	0.3589	0.4516	0.5412	-0.0067	0.4	0.5
3PC/10PC Std Dev.	0.5672	0.4117	0.4516	0.6330	0.08291	0.4307	0.4926
10PC/10PC Worst Case	0.9649	0.9516	0.9516	0.9724	0.9240	0.9516	0.9852
3PC/10PC Worst Case	0.9824	0.9538	1	0.9724	0.9624	0.9763	0.9969

Table 4.3

worst case variables. The iteration number is 8 and the optimal cut-off is 0.39. When the model is compared with the other logistic regression application examples for that data set, the 0.9824 is a good accomplishment because based on the research, the accuracy rates are 0.9380 or 0.9647. What we have done differently is the application of PCA to the different partials of the data set. The best logistic regression model can be observed from the below table.

Success Rate	Recall	Specificity	Precision	κ	F1 Score
0.9824	1	0.9724	0.9538	0.9624	0.9763

5. Summary

5.1 Summary of Results

In this project, the Decision Tree, Gradient Boosted Trees and Logistic Regression are applied on the breast cancer data set. We also applied feature reduction methods and various parameter tuning techniques which showed promise in improving our model with respect to the performance metrics. The *radius_w*, *concave_w*, *perimeter_w*, *radius_w*, *concave_m*, *area_sd* are the most important features for all the models. The *texture_m*, *smoothness_w* and *concavity_m* are similar between Gradient Boosted Trees and Logistic Regression also, the *area* is similar between Decision Trees and Logistic Regression. Notice that many of these are the worst values of our main measurements. Our best predictor was the logistic regression that only contained only the worst case features.

The comparison of the results can be observed in the table below.

Model	Success Rate	Recall	Precision	κ	F1 Score
Decision Trees	0.9123	0.8364	0.8654	0.7961	0.8506
Gradient Boosted Trees	0.972	0.9773	0.9636	0.9409	0.9704
Logistic Regression	0.9824	1	0.9538	0.9624	0.9763
Logistic Regression(RFE)	0.9649	0.9047	1	0.9230	0.95

In the above table there are two logistic regression results because the RFE has similar parameters with the other models and the result is worth a comparison. When decision trees, gradient boosted trees and logistic regression with RFE are compared the best model changes depend on what performance metric we are concerned about. In this data-set, we are most concerned about recall and precision rates which are combined with the help of the F1 Score. The F1-Score of Gradient Boosted Trees is better than the Decision Trees and LR with RFE, however if we add the other model of logistic regression it is hard to make a comparison because the features are different.

There is no best obvious result because the main purpose of this project is applying each model in the best improved way. There are different ways to improve the results so we try to choose the best ones for our models. The Decision Trees model accuracy rate is improved from 0.8421 to 0.9123 with tuning which is a significant difference to mention, all the other rates are also improved. The recall and precision are improved

too which are the most important rates for this data set. We can see the effect of that changes on F1-Score from 0.7782 to 0.8506. The Gradient Boosted Trees improves the accuracy from 0.958 to 0.972 with tuning.

In addition to this, a neural network model was tried but it was not effective as it had poor running time and was most likely overkill for this dataset. The accuracy rate is not always the most important thing for the selection of the model, we must also consider running time of our methods. In our case there is not a remarkable difference we can not choose one of the model based on this parameter, this however would be a different story if we had larger and more richer data. All of the models have a good consistency level which we can understand from the kappa statistic. The F1 Scores and recall rates are high as we aimed at the beginning of the project. We wanted to predict mainly the number of malignant tumours correctly and we have succeeded in this respect with only few examples managing to be misclassified by some of our models.

When considering our hypotheses, we did not consider how important the worst values for our measurements would be. We have shown that in many of our models the worst features had the most power when predicting the classifications of different input data. Something that we were apprehensive about was that even though certain features can be seen as a good indicator for diagnosis there are always certain outliers that manage to defy the indicator. This resonated well with some of our best models with only few test examples slipping through the net each time.

5.2 Ethical Implications

Performing analysis on such a UCI dataset does not have any obvious ethical implications, whilst an argument can be made that all data can be handled maliciously we believe that anyone putting any analytical effort into this data would be doing work for the greater good. It is crucial for us to pay attention to breast cancer as this is the most common cancer among women and the number of deaths it causes is substantial. Therefore it is important to be prudent when coming to conclusions about the results of any model, as any misinformation could directly result in deaths.

This dataset was created from 1995 and personal information has been removed therefore there are no ways to link any of this information to anyone and use it against them. We are unsure about the geographical features of the patients who have been tested, as the creators of this dataset are all from the University of Wisconsin, it is highly possible that the patients who have been recorded in this dataset are all from the United States. This is obviously a very small sample and it would be interesting to see data recorded in different parts of the world as it could show that certain features are more important for different types of people.

6. Conclusion

In conclusion, we have demonstrated with various models that we are able to indicate whether a tumour is either malignant or benign with convincing accuracy. When considering the models that we used, the Decision Tree may have been too simple to capture nuances in the dataset whilst XGBoost was most likely overkill for such a small dataset. When comparing XGBoost for classification to Logistic Regression, they are similar in many ways and it's no surprise that they gave fairly similar results. To be realistic, Logistic Regression is probably the best choice for such a dataset. It is much simpler to implement than XGBoost and it was shown to be more powerful and it is also easier to interpret the results. We have shown that worst recordings of our main measurements showed the most predictive power in our model, which was surprising given that we initially thought that the mean would carry more weight.

We noticed that no matter what we tried, a small number of observations would always seem to beat any model that we applied. Future work would look at ways to stop these small number of instances managing to fool the models. After much research online, nobody seemed to have a model that could achieve 100% accuracy on any reasonable split of the dataset. This would indicate that future interested parties should look at collecting larger and richer datasets with even more descriptive features added. We concede that data in this field might be expensive to collect, however we must insist that this is the best possible way find a better model. If an interested party were to obtain a deeper, richer dataset then perhaps XGBoost given it's complexity and parallelised features might be a better choice for data analytics in comparison to Logistic Regression.

6. Bibliography

[1] Uci machine learning repository: breast cancer wisconsin (diagnostic) data set. *UCI Machine Learning Repository: Breast Cancer Wisconsin (Diagnostic) Data Set*. http://archive.ics.uci.edu/ml/datasets/breast cancer wisconsin (diagnostic).

[2] Lisa Fayed. Differences between a malignant and benign tumor. *Verywell Health*, Jan 2020.

[3] How common is breast cancer?: Breast cancer statistics. *American Cancer Society*. https://www.cancer.org/cancer/breast-cancer/about/how-common-is-breast-cancer.html.

[4] Nick Street, William Wolberg, and O Mangasarian. Nuclear feature extraction for breast tumor diagnosis. *Proc. Soc. Photo-Opt. Inst. Eng.*, 1993, 01 1999.

[5] Kristin Bennett. Decision tree construction via linear programming. *Proceedings of the 4th Midwest Artificial Intelligence and Cognitive Science Society Conference, Utica, Illinois*, 01 1992.

[6] Jeremy J Erasmus, John E Connolly, H Page McAdams, and Victor L Roggli. Solitary pulmonary nodules: Part i. morphologic evaluation for differentiation of benign and malignant lesions. *Radiographics*, 20(1):43–58, 2000.

[7] Rangaraj M Rangayyan, Nema M El-Faramawy, JE Leo Desautels, and Onsy Abdel Alim. Measures of acutance and shape for classification of breast tumors. *IEEE Transactions on medical imaging*, 16(6):799–810, 1997.

[8] Jerome Friedman, Trevor Hastie, Robert Tibshirani, et al. Additive logistic regression: a statistical view of boosting (with discussion and a rejoinder by the authors). *The annals of statistics*, 28(2):337–407, 2000.

[9] Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pages 785–794, 2016.

[10] xgboost documentation — xgboost 1.1.0-snapshot documentation. https://xgboost.readthedocs.io/en/latest/.

A. Feature Information

Variable	Explanation	Value
id	id number	8670~91132050
label	diagnosis (m = malignant, b = benign)	m, b
radius_m	the average of radius (mean of distances from center to points on the perimeter)	6.981-28.110
texture_m	the average of texture (standard deviation of gray-scale values)	9.71-39.28
perimeter_m	the average of perimeter	43.79-188.50
area_m	the average of area	143.5-2501.0
smoothness_m	the average of smoothness (local variation in radius lengths)	0.05263-0.16340
compactness_m	the average of compactness (perimeter^2 / area - 1.0)	0.01938-0.34540
concavity_m	the average of concavity (severity of concave portions of the contour)	0.00000-0.42680
concave_m	the average of concave points (number of concave portions of the contour)	0.00000-0.20120
symmetry_m	the average of symmetry	0.1060-0.3040
fd_m	the average of fractal dimension ("coastline approximation" - 1)	0.04996-0.09744
radius_sd	the standard error of radius (mean of distances from center to points on the perimeter)	0.1115-2.8730
texture_sd	the standard error of texture (standard deviation of gray-scale values)	0.3602-4.8850
perimeter_sd	the standard error of perimeter	0.757-21.980
area_sd	the standard error of area	6.802-542.200
smoothness_sd	the standard error of smoothness (local variation in radius lengths)	0.001713-0.031130
compactness_sd	the standard error of compactness (perimeter^2 / area - 1.0)	0.002252-0.135400
concavity_sd	the standard error of concavity (severity of concave portions of the contour)	0.00000-0.39600
concave_sd	the standard error of concave points (number of concave portions of the contour)	0.000000-0.052790
symmetry_sd	the standard error of symmetry	0.007882-0.078950
fd_sd	the standard error of fractal dimension ("coastline approximation" - 1)	0.0008948-0.0298400
radius_w	the worst of radius (mean of distances from center to points on the perimeter)	7.93-36.04
texture_w	the worst of texture (standard deviation of gray-scale values)	12.02-49.54
perimeter_w	the worst of perimeter	50.41-251.20
area_w	the worst of area	185.2-4254.0
smoothness_w	*the worst of smoothness (local variation in radius lengths)*	0.07117-0.22260
compactness_w	the worst of compactness (perimeter^2 / area - 1.0)	0.02729-1.05800
concavity_w	the worst of concavity (severity of concave portions of the contour)	0.0000-1.2520
concave_w	the worst of concave points (number of concave portions of the contour)	0.00000-0.29100
symmetry_w	the worst of symmetry	0.1565-0.6638
fd_w	the worst of fractal dimension ("coastline approximation" - 1)	0.05504-0.20750

Figure A.1: Variable explanation

B. XGBoost for Classification: A walkthrough

Here we refer to **Algorithm 1** and explain how it differs for XGBoost. Let's now interpret what is going on here in the binary classification setting. In step 1, we must initialise the model with an initial prediction F_0. Let's define the logit function or log(odds):

$$\log(odds) = \log\left(\frac{p}{1-p}\right)$$

where in our case p would simply be the number of malignant tumours divided by the total number of diagnoses in our training set. The loss function is defined as:

$$L(y_i, \gamma) = -y_i \log(odds) + \log\left(1 + e^{\log(odds)}\right)$$

where we can think of the $\log(odds)$ as γ. We then sum up all of the losses and acquire the value of γ that minimises this sum, which can be found by setting the derivative equal to zero and solving for γ. We now have found F_0, this can be interpreted as a single leaf storing the probability of whether or not one single individual will have a malignant diagnosis. In step 2, this is where we start building a tree. As this is for binary classification we only need one tree per round. Let's start the first iteration of the loop, we calculate r_{im} which are called the *pseudo residuals* whereby we take the derivative of loss function with respect to the $\log(odds)$. After calculation you should be left with

$$r_{im} = y_i - \frac{e^{\log(odds)}}{1 + e^{\log(odds)}}$$

where i represents the training example and m represents the tree that we are building. Once we have calculated all of the pseudo residuals, we can then move to part ii) of the loop and we fit a regression tree to the newly found pseudo residuals. The terminal regions R_{jm} refer to the the leaves of the tree, this is something that can be limited by pre-setting a hyperparameter beforehand. In part iii) of the loop we calculate the output value γ_{jm} for each R_{jm}. We calculate the output values by minimising the loss function of all examples that fall into one leaf, typically what is done here to save time is that instead of taking the derivative to find γ we approximate the loss function as the second-order taylor polynomial.

$$L(y_i, F_{m-1}(x_i)+\gamma) \approx L(y_i, F_{m-i}(x_i)+\frac{\mathrm{d}}{\mathrm{dF}()}(y_i, F_{m-1}(x_i))\gamma+\frac{1}{2}\frac{\mathrm{d}^2}{\mathrm{dF}()^2}(y_i, F_{m-1}(x_i))\gamma^2$$

If you minimise this instead of the original loss function and solve for γ, this comes out to

$$\gamma_{jm} = \frac{\sum_{k=1}^{n} r_{kj}}{\sum_{k=1}^{n} p_k(1-p_k)}$$

we state this without proof as it takes quite a lot of work to simplify γ down to this single term. Where the numerator is the sum of the residuals of the training examples grouped into each terminal region and the denominator is sum of most recent predicted probabilities for each training example. In the first round for instance the value of p_k will all be equal to our initial prediction F_0. Lastly, in iv) we collect all of our findings to give our new prediction F_m. This is where we can start to see a similarity with AdaBoost, we now combine weak learners and scale our future learners with a learning rate ν. The convention is to have a small value for ν say between 0.1 and 0.3. We have now completed a loop. We continue in the same fashion adding weak learners scaled by ν until either the maximum number of trees M has been reached, or the residuals converge. Another convention is that M typically tends to be greater than 100. Once you have your final prediction F_m, to predict the classification of a new data item you must convert the output to a probability. This is done by passing the output $F_m(x)$ to the logistic function e.g.

$$\text{Probability of Malignancy} = \frac{e^{F_m(x)}}{1 + e^{F_m(x)}}$$

If we use a threshold of say 0.5 then we could classify x as malignant if the value comes out as > 0.5.

XGBoost

What is different about XGBoost then? Traditional gradient boost has a weak point, being that we are allowed to build weak learners that are trees instead of the stumps we had in AdaBoost. The trees therefore can grow to large depths and overfit the training data easily. Thus, XGBoost adds a regularisation parameter λ to stop the model from overfitting. Specifically what we mean by this is that step iii) of the loop in Algorithm 1 is changed to factor in a penalty term when calculating the output values.

$$\gamma_{jm} = \arg\min_{\gamma} \sum_{x_i \in R_{ij}} L(y_i, F_{m-1}(x_i) + \gamma) + \Gamma T + \frac{1}{2}\lambda\Gamma^2$$

T is number of terminal nodes in the tree and Γ is a predefined hyperparameter that is to encourage tree pruning. Now to talk about the tree building process in detail, we can either greedily compute the tree splits or take advantage of the XGBoost

approximation algorithm. Using either strategy, we compute the similarity scores of the residuals in the terminal regions.

$$\text{Similarity Score} = \frac{\left(\sum_{k=1}^{n} r_{kj}\right)^2}{\sum_{k=1}^{n} p_k(1-p_k) + \lambda}$$

When the similarity score is high it means that we either have few residuals or the grouped examples are close together in terms of their residuals. We then calculate the *Gain*.

$$Gain = Similarity_{left} + Similarity_{right} - Similarity_{parent}$$

We choose to split at the threshold that gives the largest gain. Something to note is that XGBoost has a parameter called *minchildweight* where if the denominator of the Similarity Score is less than this parameter it does not allow such leaves. Lastly, to prune the trees we calculate $Gain - \Gamma$ and if this value is negative we delete the branch. This is where the regularisation parameter comes in as it can be shown that for $\lambda > 0$ it reduces the sensitivity to terminal regions with few training examples. Now that the tree is built we just need to calculate the output value for each leaf. This is just a minor change to traditional algorithm where we add lambda to the denominator.

$$\gamma_{jm} = \frac{\sum_{k=1}^{n} r_{kj}}{\sum_{k=1}^{n} p_k(1-p_k) + \lambda}$$

The rest of the algorithm is the same, the only change is a formality where the learning rate is referred to as *eta*, a value typically set to 0.3. We also briefly touched on the greedy and approximation algorithms XGBoost can use for tree building. The difference is that the greedy algorithm forms branches without considering the long term effect on the residuals whereas the approximation algorithm considers the long run. It can do this by only considering quantiles as thresholds as well as utilising distributed computing to make calculations fast. We shall omit the details of this, the only thing we need to know here is that when the dataset becomes large and complicated the greedy approach becomes very slow and the approximation algorithm is superior.