VRIJE UNIVERSITEIT AMSTERDAM

RESEARCH PAPER

Master Business Analytics

Predicting malignant tumor cells in breasts

Author:
Larissa Westerdijk

Supervisor:
Prof. dr. Sandjai Bhulai

10th July, 2018

Predicting malignant tumor cells
in breasts

Larissa Westerdijk

Research paper

Vrije Universiteit Amsterdam
Faculty of Science
Business Analytics
De Boelelaan 1105
1081 HV Amsterdam

July 2018

Contents

Preface

As a compulsory part of the Master's degree program Business Analytics at the VU Amsterdam, this research paper is produced. This thesis is written based on a research project regarding a specific problem statement. The input of this specific research involves the use of data. The purpose of this report is to demonstrate the ability to describe a problem in a clear manner for the benefit of an expert manager. In this research problem, the field of data mining has been addressed, which includes the main goal of predicting a certain event.

I would like to thank my supervisor Prof. dr. Sandjai Bhulai for guiding and supporting me during this research.

Abstract

Predicting malignant tumor cells in breasts

by Larissa Westerdijk

Breast cancer is the most common cancer among women in the Netherlands in 2016. The accuracy of visually diagnosed breast fine needle aspirates, however, is about 90%. It is, therefore, necessary to minimize this subjectivity with digitized images of the fine needle aspirates and machine learning techniques. In this research, the presence of malignant tumor cells in breasts will be predicted for the Wisconsin Diagnostic Breast Cancer data. Several machine learning methods, like logistic regression, random forest, support vector machine, and neural network, will be used to model this. The performance of these models will be tested using the accuracy, as well as the AUC from the ROC curve, the sensitivity, and the specificity. Finally, after optimizing the four individual models, an ensemble will be created to achieve an even more robust predictive model. The final accuracy results are 0.9735, 0.9735, 0.9823, 0.9735, and 0.9823 for the logistic regression, random forest, support vector machine, neural network, and ensemble models respectively. Especially the number of false negatives need to be decreased, which is a recommendation for future research.

1 Introduction

Breast cancer is the most common cancer among women in the Netherlands. Moreover, the IKNL (Integraal Kankercentrum Nederland) shows it is the most common cancer after skin cancer occurring in this little country [13]. In the Netherlands, 109,663 people are diagnosed with cancer every year, where 14,890 out of those people have breast cancer. This is equal to 13.6% of the total Dutch cancer population. Each year the number of people who die because of breast cancer is about 3,175. These are serious numbers that need to be decreased. To accomplish this, it is essential to correctly detect and diagnose the patients as early as possible. A visual diagnosis is executed by an assigned doctor and has an accuracy of about 90%. This percentage, however, must and can be optimized with the use of data about the breast cells.

In order to improve this accuracy, the diagnosis can be supported by the use of digitized images made of the breast cells and machine learning techniques. Computers are used to predict malignant (cancerous) tumor nuclei and therefore detect them. This process starts with taking fine needle aspirates of the breast mass of women. This substance is stained under a microscope to highlight the nuclei. The next step involves imaging a portion of the fine needle aspirates in which the cells are well-differentiated and digitalizing these images. Subsequently, a computer system divides the individual cells in each of the images by accurately specifying the boundary of each cell, which is shown in Appendix A. It is necessary to convert all potentially important size, shape, and texture characteristics of each cell out of the image into several features. Examples of these features are the radius and perimeter of the cell nucleus. Once the data is ready, it can be used to build the predictive models to differentiate malignant (cancerous) from benign (non-cancerous) cases.

The objective of this paper is to develop several machine learning models with a better performance than the visual diagnosis of the doctors, and therefore simultaneously minimize this subjectivity. These models are made to predict the presence of malignant tumor cells in breasts. This research focuses on the predictive models, processing the data by defining the features with computer-based analytical techniques is outside the scope of this paper.

The remainder of this paper is divided into seven parts, which is covered in Chapters 2 to 8. In the first part, the related work of this research is addressed. The second part is about the description and analysis of the data. The third part provides the methodologies used for building and evaluating the models, such as the predictive models, the evaluation measures, and the experimental setup. In the fourth part, the results of the performing models are demonstrated, which includes the chosen features, the parameter settings, and the performance values. This is followed by a short evaluation of the wrongly predicted cases. Finally, the paper ends with part six and seven covering the conclusion and discussion respectively.

2 Related work

There has been a lot of research on medical diagnosis of breast cancer. Many of those consider models that can predict whether the tumor is benign or malignant. For predicting such a classification problem many techniques are available. Besides this, different kinds of explanatory variables can be used for predicting the presence of malignant tumor cells. A lot of researches, however, have used one of the three different Wisconsin breast cancer datasets (Wisconsin Breast Cancer (WBC), Wisconsin Diagnosis Breast Cancer (WDBC), and Wisconsin Prognosis Breast Cancer (WPBC)).

All three datasets are used in the paper of Salama et al. [16], where each dataset has different features trying to predict the outcome. For each of the datasets, a multi-classifier was made using a selection of the classifiers decision tree, Multi-Layer Perceptron, Naive Bayes, Sequential Minimal Optimization, and Instance Based for K-Nearest neighbor. The highest obtained accuracies are 97.28%, 97.72%, and 77.32% for the datasets WBC, WDBC, and WPBC, respectively.

Several studies used the WBC dataset for detecting breast cancer. In the study of Karabatak et al. [11] breast cancer is detected based on association rules and a neural network. The association rules are used for eliminating unnecessary data and thus reducing the feature dimension. The neural network classifies each record using those remaining features. The final model resulted in an accuracy of 97.4%. Abbass [1] achieved an accuracy of 98.1% $\pm$ 0.5 using an evolutionary artificial neural network approach. This approach is based on the Pareto-differential evolution algorithm that is augmented with local search. In Marcano-Cedeño et al. [14], an Artificial metaplasticity Multilayer Perceptron algorithm is applied, obtaining a classification accuracy of 99.26%. Akay [2] reached the highest accuracy of 99.51% using an SVM model combined with feature selection.

Various other studies used the WDBC dataset, which is also used in this paper. Wolberg et al. [18] applied two models, logistic regression and Multisurface Method-Tree. These resulted in 10-fold cross-validated classification accuracies of 96.2% and 97.5%, respectively. In the research of Mu et al. [15] support vector machines, radial basis function networks, and self-organizing maps are applied to detect breast cancer. The performance of different combinations of the classifiers is compared based on 10-fold cross-validation. The average performance accuracy is over 98%.

3 Data

In this research, the Wisconsin Diagnosis Breast Cancer dataset is analyzed. This section contains the description and analysis of the data.

3.1 Description

The dataset consists of features obtained from a digitized image of a fine needle aspirate of a breast mass. These features describe characteristics of the cell nuclei present in the image. The ten real-valued features are computed for each cell, and are defined as follows.

1. *Radius*. The average distance from the center of the nucleus to each of the boundary points.
2. *Texture*. The standard deviation of the gray-scale values. A gray-scale value represents the intensity of the shades of gray in each pixel of the image.
3. *Perimeter*. The total distance of the boundary of the cell nucleus.
4. *Area*. The number of pixels on the interior of the boundary and adding one-half of the pixels on the perimeter, to correct for the error caused by digitization.
5. *Smoothness*. The difference between the length of a radius length and the mean length of the two radius lines surrounding it, hence the local variation in radius lengths.
6. *Compactness*. The perimeter and area are combined using the formula $\frac{perimeter^2}{area} - 1$ to obtain a measure of compactness of the cell nuclei.
7. *Concavity*. The severity of concave portions of the contour. A high concavity means that the boundary of the cell nucleus has indentations, and thus is rather rough than smooth.
8. *Concave points*. The number of concave portions of the contour of the cell nucleus.
9. *Symmetry*. The symmetry is determined by first finding the longest line from boundary point to boundary point through the center of the nucleus. Subsequently, the relative length differences between the lines perpendicular to the longest line to the boundary in both directions are measured. Attention should be given to nuclei where the longest line cuts through the boundary because of concavity.
10. *Fractal dimension*. The fractal dimension is approximated by the 'coastline approximation'. The perimeter of the nucleus can be measured using different lengths of measuring sticks. As this length increases, the total length of the measured 'coastline' decreases due to lower precision of the

measurement. The theoretical fractal dimension is then determined by dividing the logarithm of the observed perimeter L(s) by the logarithm of the measuring stick length s. Plotting log(L(s)) against log(s) and determining the negative value of the slope results in an approximation of the fractal dimension D [7]. Finally, the desired feature is determined by the calculation D - 1.

The size of the nucleus is expressed by the features radius and area. The shape is expressed by the features smoothness, concavity, compactness, concave points, symmetry, and fractal dimension. The perimeter expresses both the size and shape of the nucleus. A higher value of shape features corresponds to a less regular contour and, therefore, to a higher probability of malignancy. For each of the features the mean value, worst value (mean of the three largest values), and standard error are computed for each image, resulting in 30 features of 569 images. When referring to the mean, worst, or standard error value of, for example, the feature radius, this paper will mention radius, radius worst, or radius SE respectively. The mean and standard error of a feature x are determined as follows.

$$Mean_x = \frac{1}{N}\sum_{i=0}^{N} x_i,$$
$$Standard\ error_x = \frac{sd}{\sqrt{N}},$$

where N refers to the number of observations of the sample, x_i refers to the i^{th} feature value of the sample, and sd refers to the standard deviation.

All features are recorded with four significant digits. Together with the ID number and the diagnosis the final dataset consists of 32 attributes.

3.2 Analysis

In order to know how to handle the data in the predictive models, some analyses have been performed. The first observation of these data is the fact that it has no missing values. Hence, no suitable method is needed to deal with such a problem.

Some general analysis shows that 357 women have a benign tumor, while the other 212 women have a malignant tumor. This means 37.26% of the data is our main focus. This distribution of the data shows that the number of benign samples is not excessively more than the number of malignant samples. Because the dataset is not unbalanced, there is no need to scale the dataset.

Each of the features has been analyzed to detect unusual values. This includes calculating the minimum, maximum, mean, and median values. Plotting the data in boxplots makes it easy to detect outliers. Most of the boxplots look like the boxplot in Figure 3.1, which means some of the data points do not belong in the box. This does not necessarily mean this feature has outliers that need much attention. In the case of the boxplot in Figure 3.2 you could argue whether the two highest points are outliers. Since each malignant tumor cell is unique and might differ extremely from others in real life, here is assumed the dataset does not have any outliers. Based on this assumption no instances are removed from

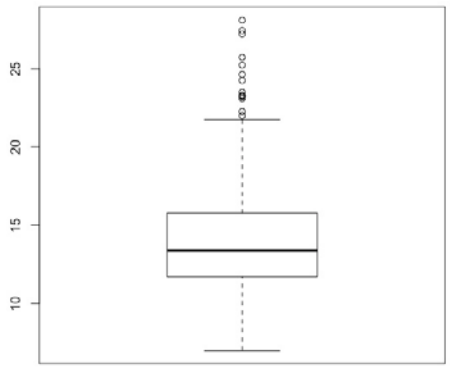

FIGURE 3.1: Boxplot radius mean

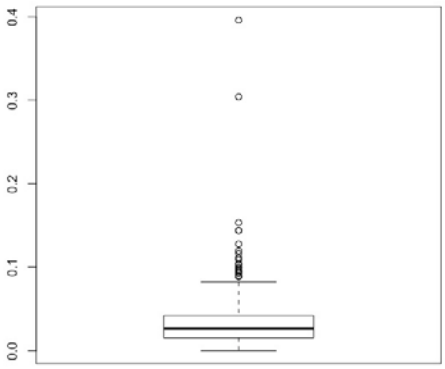

FIGURE 3.2: Boxplot concavity SE

the dataset.

An important part of the analysis is examining the correlation between the features. The correlation measures the strength and direction of a linear relationship between two features. Using two highly correlated features in a model can reduce the performance. Also, since they have a strong relation, one of the features is redundant. A correlation between two features does not necessarily mean those two features have a causal relation. However, if two features have a causal relation, they must be correlated. Here, only correlation has been tested on the data. The two features radius and perimeter have a correlation of 0.9979, which means they have a very strong positive linear relationship. The correlation between compactness and concavity is equal to 0.8831, which means they also have a strong positive linear relationship. Finally, the features symmetry and perimeter have a correlation of 0.1830, which means they have no linear relationship since a weak positive linear relationship corresponds to a minimum correlation of 0.3. These differences in correlation are clearly visible in Figures 3.3, 3.4, and 3.5, where the relation between compactness and concavity, perimeter and symmetry, and radius and perimeter are shown respectively.

The relationship plots in Figure 3.5 show the obvious linear relationships, which is also reflected in the correlation value of 0.9979 between radius and perimeter. A less obvious but still visible straight line is the one between compactness and concavity, shown in Figure 3.3. This difference in the level of linearity can also be distinguished by the correlation of 0.8831, which is less than 0.9979. The thicker-lined plot indicates a lower level of linearity. The plot that does not show any linear relationship is the one in Figure 3.4, which is in accordance with the low correlation value of 0.1830 between perimeter and symmetry.

Aside from the level of linearity, all correlation values show a positive number. This means all relations are positive, and increasing the value of, for example, the perimeter will correspond to an increasing value of the area.

Also the relationships between mean, SE, and worst are plotted for each of the

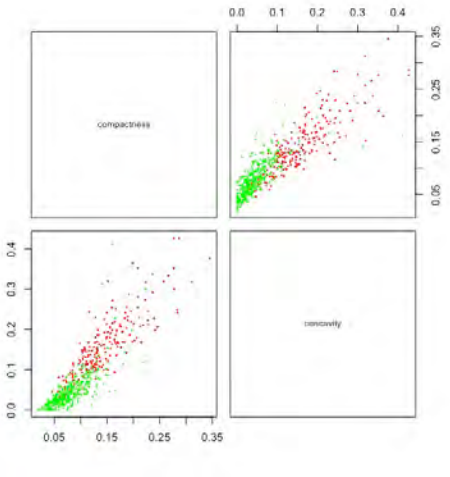

FIGURE 3.3: Relationship between compactness and concavity

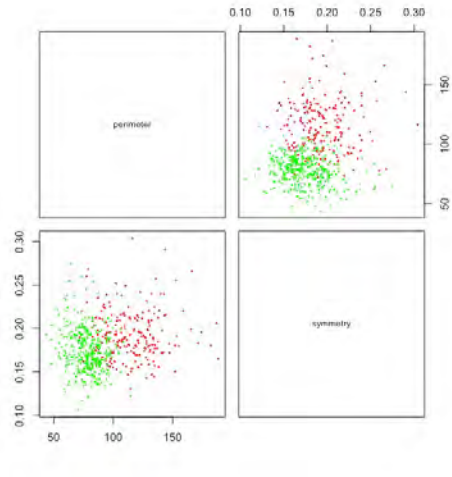

FIGURE 3.4: Relationship between perimeter and symmetry

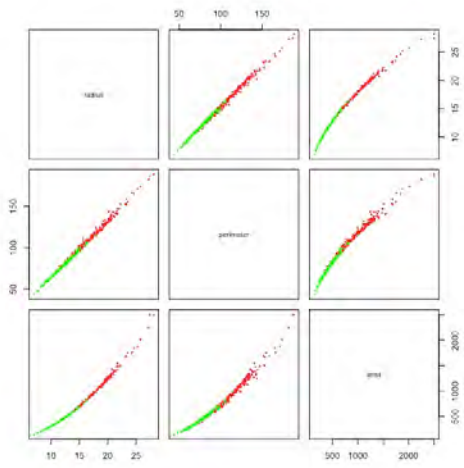

FIGURE 3.5: Relationship between radius, perimeter and area

features individually. Further investigation has been done in cases of obvious linear relationships. It is concluded that whenever two features have a big correlation, then the same applies to the 'worst' version of these features. In addition, also the level of linearity is high between the worst version of one of those features and the mean version of the other concerning feature.

All the combinations of features with high level of linearity need to be kept in mind during the feature selection.

Besides the relationships shown in the plots in Figures 3.3, 3.4, and 3.5, these plots show which cases are benign (green) and which are malignant (red). None of the plots have a clear division to divide the two groups. Therefore, more advanced models will be used to optimize such divisions. This is confirmed by examining the minimum and maximum values of each feature. In Table 3.1 these values are shown for the feature radius. The minimum value of the malignant cases is smaller than the maximum value of the benign cases, which concludes overlap between the benign and malignant cases. If all the benign values would have been below a certain number and the malignant values above this number, then the cases would have been easily separated and consequently easily predicted.

	Minimum	Maximum
Benign	6.981	17.85
Malignant	10.95	28.11

TABLE 3.1: Values of radius

Overall the data is very clean, which means no preprocessing of the data has been executed. Also no features have been engineered.

4 Methodology

This section explains the methods that are used in order to receive the desired outcome. The aim of this research is to predict whether a tumor cell is malignant or benign. The diagnosis is the dependent variable (the target), which leaves the remaining features as possible explanatory variables. Since the target variable is categorical, this problem is considered as a classification problem. A possible class label for this target variable is either 'benign' or 'malignant'. Classification is a supervised learning method, which means that the training dataset consists of the correctly identified observations of the target variable.

The data analysis in Section 3.2 showed that the dataset has no missing values, has no outrageous outliers, and is not extremely unbalanced. Therefore, there is no need to tackle these in a data preparation. Consequently, all the 569 original observations are used to create the classifiers.

The models that are used for this classification are Logistic Regression (LR), Random Forest (RF), Support Vector Machine (SVM), Neural Network (NN),

and an ensemble. These models are explained in detail in Section 4.1, after which Sections 4.2 and 4.3 explain the evaluation measures and the experimental setup, respectively.

4.1 Predictive models

All models that are to be explained below are suitable to solve the classification problem in question.

4.1.1 Logistic regression

Regression is a statistical process that estimates the relationship between the dependent (target) variable and the independent (explanatory) variables. Linear regression is the most basic type of regression, in which case the relation between the dependent and independent variable(s) is linear. Logistic regression models can be thought of as extensions of linear regression models, and hence, linear regression is explained first. The explanation has the same approach as de Gunst [8].

In general, there are n observed data points $y_1, ..., y_n$, which represent the realizations of the independent random variables $Y_1, ..., Y_n$. Besides this, there are m explanatory variables for which the right coefficients vector β needs to be found. The vector x_i of length m+1 contains the intercept along with the explanatory variables of the i-th observation. The linear regression model can be described as follows.

$$\text{(i)} \quad Y_i \sim \mathcal{N}(\mu_i, \sigma^2),$$

$$\text{(ii)} \quad \eta_i = x_i^T \beta,$$

$$\text{(iii)} \quad \eta_i = g(\mu_i) = \mu_i,$$

for i = 1,...,n, with $\beta = (\beta_0, ..., \beta_m)^T$ the coefficient vector for the intercept (β_0) and the explanatory variables. This representation seems quite difficult for such a relatively easy model, but it ensures that the alteration to the logistic regression model is clear.

As is shown above, the model consists of three components, the random component (i), the systematic component (ii), and the link function (iii). The random component specifies the distribution of the target Y_i. The systematic component is the vector η_i, which consists of the predictors for each observation. These predictors are formed by multiplying the values of the explanatory variables with the coefficient vector. The link function, denoted by g, is the link between the random and the systematic component. More precisely, it specifies the relation between the two by $g(\mathbb{E}Y_i) = \eta_i$. Since the relation is linear η_i equals μ_i.

In logistic regression (LR), however, the dependent variable is categorical. The approach of logistic regression is part of a whole class of models, called generalized linear models (GLMs). They are called generalized *linear* models because the systematic component remains the same, and hence the η_i are still linear. The random component and the link function need to be adjusted, though, to

transform the linear regression model into the logistic regression model. In the linear regression model η_i and Y_i can take any real number, meaning $\eta_i, Y_i \in \mathbb{R}$. In logistic regression the target variables Y_i can only take two values 0 or 1. This means the following for the binomial random variables, $0 < \mathbb{E}Y_i < 1$. And, therefore, the link function should ensure that it maps the interval (0,1) onto the real line. Satisfying this condition results in the logistic regression model below.

$$\text{(i)} \quad n_i Y_i \sim \text{Bin}(n_i, \mu_i),$$

$$\text{(ii)} \quad \eta_i = x_i^T \beta,$$

$$\text{(iii)} \quad \eta_i = g(\mu_i) = \log \tfrac{\mu_i}{1-\mu_i},$$

for i = 1,...,n.

Relating the theory to this particular problem, the binary dependent variable can take one of the two values 'benign' or 'malignant'. Then the algorithm tries to fit the best regression coefficient vector β to the given data.

4.1.2 Random forest

Random forest (RF) originates from the decision tree, since a random forest is the result of various weak decision tree predictors added together. A random forest on its own is, therefore, an ensemble of decision trees. This model is able to describe nonlinear relations in data, which is needed for this specific problem as is shown in Section 3.2. The input of a random forest model can take both numerical and categorical variables. Probably the biggest advantage of this model is the fact that it can be used for both regression and classification problems. Here, the algorithm will be explained focused on classification.

Decision tree

Since a random forest consists of multiple decision trees, first, a short and general description of a decision tree will be given. The basis of a decision tree is a system based on a set of rules. Given the training dataset including the target and features, the algorithm designs a set of rules. This same set of rules will be used to perform the predictions on the test dataset. These rules can be visually represented as a tree. The top of the tree is the first decision node, called the root node. Each of the decision nodes corresponds to a feature. The bottom of the tree consists of several so-called leaf nodes. Each leaf node corresponds to a class label, which is 'benign' or 'malignant' for this specific problem. The process of making these rules, and thus the decision tree, can be explained by several steps. Here, the ID3 algorithm is used to explain how to generate this decision tree from the dataset.

1. Calculate the entropy of each feature, using dataset S.
2. Split this dataset S into subsets using the feature with the highest information gain.
3. Make a decision node containing this feature.
4. Recurse on the subsets using the remaining features until you find a leaf node in all the branches of the tree.

The entropy of data X, $H(X)$, is a measure of the irregularity of the data. High entropy means X is from a uniform distribution, and low entropy means X is from a varied distribution with peaks and valleys. The information gain, $IG(X|Y)$, measures the decrease in entropy if you know the value of a feature. These two measures are calculated with the following equations, where $p(x)$ is the proportion of the number of elements equal to x to the number of elements in the set X.

Entropy $H(X)$ is equal to Equation (4.1)

$$H(X) = \sum_{x \in X} -p(x) \log_2 p(x) \tag{4.1}$$

Conditional entropy $H(X|Y)$ is equal to Equation (4.2)

$$H(X|Y) = \sum_{y \in Y} p_{Y=y} \times H(X|Y = y) \tag{4.2}$$

Information gain $IG(X|Y)$ is equal to Equation (4.3)

$$IG(X|Y) = H(X) - H(X|Y) \tag{4.3}$$

The ID3 algorithm does not guarantee an optimal solution, since it can get stuck in a local optimum. Also, a decision tree tends to overfit the training data. These two disadvantages of the algorithm are not a big concern when using the decision tree for the random forest. Using many of these weak learners together in the random forest will make the final classifier stronger.

Random forest

Random forests are trained on different parts of the same training dataset. This is what tackles the issue of overfitting that is common for decision trees. The approach used to generate the random forest can be described in the following steps.

1. Randomly sample n cases with replacement from the training data, where the total number of training cases is n.
2. At each node:
   (a) Randomly select k out of a total of m features, where k < m.
   (b) Split the node on the feature with the best split.
   (c) At the next node, iterate step (a) and (b).
3. Each tree is grown to the largest extent possible.
4. Iterate steps 1 to 3 until N number of trees are generated.

After creating the random forest, the model is used for the predictions. This process starts with using the input features of the test dataset in each of the decision trees in the forest. The classification of each individual tree counts as one vote for the corresponding class. The final prediction is the classification having the most votes.

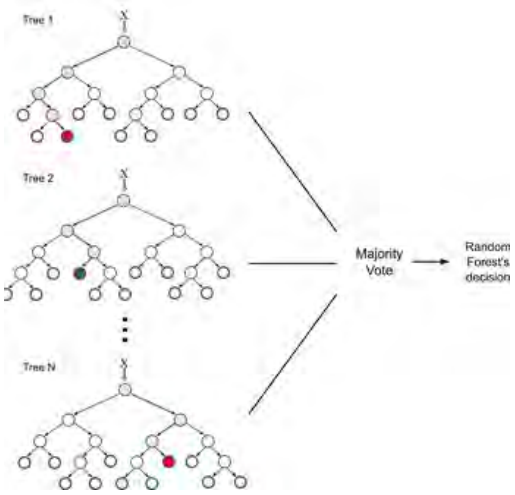

FIGURE 4.1: Visualization of random forest [3]

Important to ensure is the correlation between the N generated trees. The higher these correlations, the higher the error rate of the random forest. Therefore, the trees should be as uncorrelated as possible. By lowering the value of k, the inter-tree correlation but also the strength of each individual tree goes down. This is why the optimal value of k needs to be discovered. This optimal value usually lies in an optimal range.

A random forest can deal with missing values. Also, the algorithm can be used for feature engineering. This means the algorithm can identify the most important features out of all the available features.

One big difference exists between the decision tree and random forest algorithm. This difference is the reason why not all the individual trees in the random forest end up being the same. Creating an individual tree for a random forest is a little different than for a decision tree. Instead of finding the root node and other decision nodes by using the information gain, these nodes will be found randomly as is shown in the steps above.

4.1.3 Support vector machine

A support vector machine (SVM) is a linear classifier which is based on margin maximization. For this particular problem, the data needs to be classified into two groups, 'benign' and 'malignant'. The SVM accomplishes this classification by constructing a hyperplane that separates the data space into two areas, called classes. Depending on the number of input features, this hyperplane is constructed in a certain dimensional space. An SVM can be linear or nonlinear, with separable and non-separable cases. The extension from the linear to the nonlinear case is achieved by the use of the kernel trick.

Figure 4.2 shows a linear SVM which is separable. All negative samples are on one side of the optimal hyperplane and all positive samples are on the other side. Both maximal margin hyperplanes are optimized to be as far away from each other as possible and still separate the classes. The data points touching the maximal margin hyperplanes are called the support vectors. Because of

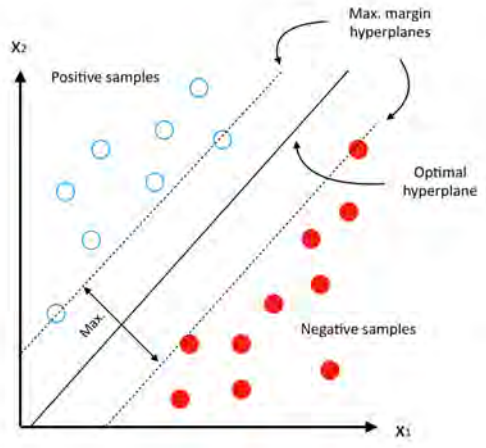

FIGURE 4.2: Maximum margin hyperplane

demonstration purposes, this construction is relatively easy. However, most advanced problems are not linearly separable. In those cases, a kernel can be used, which is the radial basis function (RBF) kernel in this research. Linearly non-separable data in the original dimension might be linearly separable in a higher dimension.

This means a transformation or mapping is needed. The dataset is given in dimension N and it is, therefore, necessary to find a transformation $\phi : \mathbb{R}^N \to \mathbb{R}^M$ such that the transformed data is linearly separable in $\mathbb{R}^M$, where $M > N$. The decision boundary is then linear in the dimension M, which is achieved by making use of the higher-dimensional space, and is non-linear if you transform the boundary back to dimension N. However, the transformation into a higher dimension can easily lead to serious computational and memory problems. This is why kernel functions are used. Such a kernel function $k(\mathbf{x}, \mathbf{z})$ is a function $k : \mathbb{R}^N \times \mathbb{R}^N \to \mathbb{R}$ with the aim to indicate the similarity between two inputs $\mathbf{x}$ and $\mathbf{z}$. It implicitly computes the similarity between the two input vectors in $\mathbb{R}^M$ without explicitly transforming vectors $\mathbf{x}$ and $\mathbf{z}$ to this higher dimension. Now no extra memory is needed and only minimal extra computation time is needed to compute all pairwise $k(\mathbf{x}, \mathbf{z})$. This is called the kernel trick. [12]

In this research, the RBF kernel is used and is as follows.

$$k(\mathbf{x}, \mathbf{z}) = \langle \phi(\mathbf{x}), \phi(\mathbf{z}) \rangle = e^{-\frac{\|\mathbf{x}-\mathbf{z}\|^2}{2\sigma^2}} \tag{4.4}$$

The kernel parameter sigma σ needs to be tuned to get an optimal performance from the final model. For the given training set $\{(x_i, y_i) \mid x_i \in \mathbb{R}^N,\ y_i \in \{-1, 1\},\ i = 1, ..., n\}$ of a binary classification problem, each hyperplane should satisfy the following equation.

$$y_i((w \cdot x_i) + b) \geq 1 - \xi_i, \tag{4.5}$$

where w is the corresponding weight, b is the intercept term, and $\xi_i \geq 0$ is a slack variable.

In order to get the optimal hyperplane in dimension M that will separate the

space in two regions, the following formula should be minimized subject to Equation (4.5).

$$\frac{1}{2} \parallel w \parallel^2 \; + \; C \sum_{i=1}^{n} \xi_i, \tag{4.6}$$

where C > 0 is a constant that determines the balance between complexity and classification accuracy and should be tuned. [17]

4.1.4 Artificial neural network

An artificial neural network (ANN) has its name due to the fact that it has certain performance characteristics in common with biological neuron networks. The neurons are the elements where the information processing occurs. The needed signals are transported between the connecting neurons, where each connection has an associated weight. This weight will be multiplied with the signal being transmitted. Once the signal reaches the neuron, an activation function is applied to the input to determine the output. The activation function used in this research is sigmoid, which gives a value between [0,1]. In mathematical form, this transformation in each neuron is as follows.

$$o = S\left(\sum_j w_j i_j + b\right), \tag{4.7}$$

where o is the output of the neuron, w_j are the weights of the neuron, i_j are the inputs of the neuron, and b is a possible bias term. The sigmoid activation function S is as follows.

$$S(x) = \frac{e^x}{e^x + 1} \tag{4.8}$$

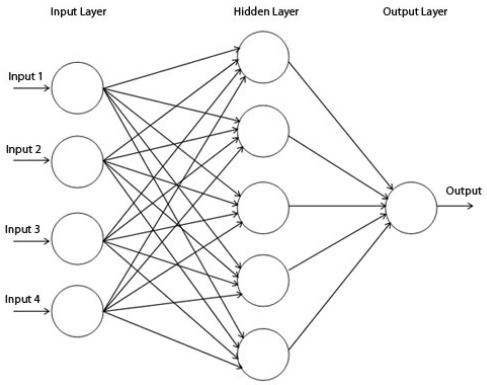

FIGURE 4.3: Architecture of a neural network

An example of the architecture of an artificial neural network is shown in Figure 4.3. Neural networks exist in many different kinds. In general, there are three different classes of network architectures, single-layer feed-forward neural networks, multi-layer feed-forward neural networks, and recurrent neural networks. The architecture that has been used for this problem is the single-layer feed-forward neural network, since a higher level of complexity is not needed for this specific problem. This means the neural network has an input

layer, one hidden layer, and an output layer, as is shown in Figure 4.3. Feed-forward means that the signals are only transmitted in forward direction, not in forward and backward direction as in a recurrent neural network.

The objective during training a neural network model is to assign suitable weights to the connections, in order to obtain a minimum error between the output value and target value. All weights are randomly initialized and must be optimized in such a way that the model leads to accurate classifications. Backpropagation can be used for determining the optimal weights. The number of neurons in the hidden layer should also be optimized. The number of input neurons and output neurons, however, are fixed. The first is equal to the number of features used and for this specific problem the latter is equal to one, since the output is the diagnosis of the tumor cell. The other parameter that should be determined is the decay. The weight decay is important for properly updating the new weights. Weight updating is applied as shown in the following equation.

$$w_i \leftarrow w_i - \eta \frac{\partial E}{\partial w_i} - \eta \lambda w_i, \tag{4.9}$$

where η is the learning rate, E is the error function, and λ is the decay parameter. This error function $E(\mathbf{w})$ needs to be minimized.

4.1.5 Ensemble

An ensemble in the field of data mining is an approach that combines multiple predictive models with the aim of improving the predictive performance. Multiple types of ensemble methods exist, the easiest method of which is majority voting in case of a classification problem. Here the method stacking has been used. Stacking begins by training multiple individual models, which are typically different types of models. These models are used to make predictions on another part of the data. The ensemble model is trained using those predictions and it learns how to best combine the predictions of the individual models. Preferably, the predictions resulting from the individual models should have a correlation lower than 0.75, when a stacking method is applied. The main idea of an ideally low correlation is that highly correlated poor classifiers can override the answers of the better classifiers, which should be avoided.

The chosen ensemble method in this research is a random forest model. Since all the inter-correlations are lower than 0.75, the input of this combiner are the output predictions of all the individual models LR, RF, SVM, and NN. To make the final model even more robust, some of the original input features can be used as input in the ensemble model.

A final ensemble model resulting from combining several predictors generally shows a better performance than the individual models, which is achieved by mixing the strengths and diversity of the individual models together.

4.2 Evaluation measures

To evaluate the performance of the different models, evaluation measures are needed. These measures are based on the confusion matrix, which uses the

abbreviations below. Here, it is important to note that correctly predicting the malignant cases is the aim of this research. Hence, predicting malignant is positive. When a binary prediction is made, four different types of outcomes are possible.

- *TP (True Positive)*: predicting malignant, while it is malignant
- *FP (False Positive)*: predicting malignant, while it is benign
- *TN (True Negative)*: predicting benign, while it is benign
- *FN (False Negative)*: predicting benign, while it is malignant

Each of the predicted cases belongs to one of the four categories above. The total number of occurrences in the categories are used to calculate the following evaluation measures.

1. *Accuracy*: The proportion of correctly identified cases. The corresponding equation is as follows.

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \tag{4.10}$$

2. *Sensitivity*: The proportion of correctly identified positives with respect to all positives, also called the true positive rate (TPR). The corresponding equation is as follows.

$$Sensitivity = \frac{TP}{P} = \frac{TP}{TP + FN} \tag{4.11}$$

3. *Specificity*: The proportion of correctly identified negatives with respect to all negatives, also called the true negative rate. The corresponding equation is as follows.

$$Specificity = \frac{TN}{N} = \frac{TN}{TN + FP} \tag{4.12}$$

4. *AUC*: The area under the curve, which is referring to the receiver operating characteristic (ROC) curve. It is created by plotting the sensitivity with respect to the false positive rate (FPR). The total area under the resulting plotted line is the performance measure AUC. This false positive rate is given by the following equation.

$$FPR = 1 - specificity = \frac{FP}{TN + FP} \tag{4.13}$$

   The ROC curve is used to visualize the performance of a binary classifier. It is the result of plotting the TPR on the y-axis versus the FPR on the x-axis for every possible classification threshold (between 0 and 1). At each classification threshold a proportion of the positives and a proportion of the negatives is correctly predicted. By plotting these proportion values for the different thresholds, the ROC curve is created in the graph. The ideal situation is when the classifier can separate the two classes completely, resulting in a ROC curve like the green one in Figure 4.4. This optimal predictor will have an AUC value of 1. Obviously, in many real life

situations this does not occur. A more realistic level of the ROC curve is towards the blue curve, shown in this Figure 4.4. Subsequently, the AUC value is calculated by taking the part of the area that is under the curve, which can be at minimum 0 and maximum 1. The larger the area under the ROC curve the better the classifier. The dashed line on the diagonal represents the ROC curve of a random predictor, with a corresponding AUC value of 0.5.

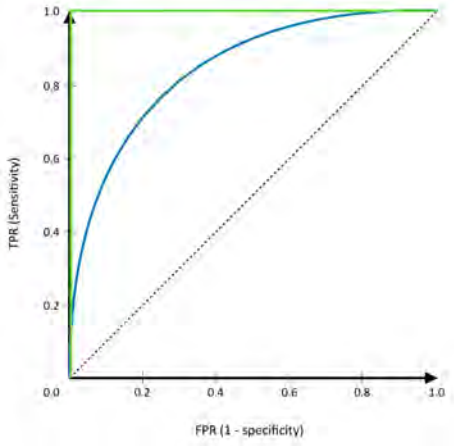

FIGURE 4.4: ROC curves

A note about this figure is that the smoothness of the curve is for demonstration purposes. The ROC results of the created models will look less smooth.

In this research, the focus is on improving the accuracy. This value needs to be maximized to optimize the models, which is the objective in the experimental setup. However, another important value to monitor for this specific problem is the number of false negatives. Predicting the nucleus as being benign, while it actually is malignant is obviously the worst that can happen. Therefore, the false negatives need to be as low as possible. In terms of the evaluation measures above, this means the sensitivity needs to be as high as possible.

4.3 Experimental setup

The models are implemented in R, a software package and programming language designed for statistical and data analytical purposes.

In order to do a fair evaluation of a model with certain chosen features and parameters, the data is divided into different sets. This division has split the whole dataset in a training, validation, and test set of 60%, 20%, and 20% respectively. Each of the four individual models, LR, RF, SVM, and NN, are trained on the training dataset. These trained models are validated on the validation set, which will give the prediction accuracy for this particular set of data. Since the ensemble model is based on the predictions of the other models, this model cannot be trained on the training dataset. The predictions of the other models, together with some features are added together in a new dataset. Let us call this dataset, dataset S. Set S now has the same number of instances as

the validation dataset. The ensemble model is trained on dataset S and has, therefore, 66.67% less data to train on. Finally, all five models are tested on the test dataset. These performance measures are compared and based on those comparisons can be concluded which model is the best.

10-Fold cross validation is used during training each of the models. This validation procedure randomly divides the dataset into ten equally sized sets. Each of the ten sets is used as a test set. The remaining nine sets are used for training and creating the classifier. The obtained ten estimates are averaged. This method is repeated 10 times for each model being trained. The final 10-fold cross validation accuracy is the average value of the 10 individual accuracies. The accuracy resulting from the 10-fold cross validation should approximately be the same as the accuracy resulting from testing on the validation set. More specifically, these values can definitely not differ more than 10% from each other, since this would indicate that the training or validation set is biased and is not a good representative for the whole dataset. If they do not approximately have the same value, the model might be overfitting or underfitting.

Besides splitting up the dataset, preprocessing of the data, feature selection, and parameter selection are necessary steps to optimize the performances of the models. The last step of this research is to compare the different models using the evaluation measures resulting from testing on the test set.

4.3.1 Preprocessing data

For each of the models is investigated whether preprocessing the data will result in a better performance. The performance of some models increases when all features are in a similar range. This can be achieved by normalizing or standardizing the data. Normalizing a dataset means all numeric features will be scaled into the range [0,1]. This transformation on a certain feature X is done by the following formula.

$$X_{new} = \frac{X - X_{min}}{X_{max} - X_{min}} \tag{4.14}$$

Standardizing a dataset means transforming the features to have a mean of 0 and a variance of 1. This transformation is done by the following formula, where μ is the feature mean and σ is the feature standard deviation.

$$X_{new} = \frac{X - \mu}{\sigma} \tag{4.15}$$

By trial and error it has been decided to use data standardization for the RF, SVM, and NN models. The data of both the LR and ensemble models do not need any preprocessing.

4.3.2 Feature selection

Selecting adequate features is of great relevance for the overall classifier performance. Different methods have been used to discover the feature importance and subsequently do a suitable feature selection for each of the models. These methods should be performed solely on the training dataset, since performing

them on the whole dataset leads to prediction bias. The final selection of features chosen for each model is based on trial and error of different combinations of the output of these methods. The methods are explained below.

Medical background

A first approach to determine the important features is by considering the medical side. The visual characteristics of a malignant cell nucleus compared to a benign cell nucleus are obviously different. In literature the most distinguishing visual features of malignant cells and nuclei are irregular size, irregular shape, lower level of cells sticking to each other, cells compacted within smaller area, haphazardly arranged cells, much darker color of the nucleus, and high ratio of nucleus compared to the whole cell [4] [6]. The dataset only consists of information about the size and shape. Since both are mentioned as important in the literature, all features are considered significant based on the medical analysis.

Correlation

Testing the correlation between the features is important, because including highly correlated features might worsen the final model. Particularly because many of the features in this specific problem are similar, features can be redundant.

Recursive feature elimination

Recursive feature elimination (RFE) is a backward feature elimination that results in a list of features, which should be used as predictors. It uses a random forest algorithm to test different combinations of features. The evaluation measure for these tests is the accuracy. The method starts with all the features (potential predictors) in the first test model. The test results in the corresponding accuracy and the feature importance ranking. The lowest ranked features are removed from the next models to be tested. This continues with all the subsequent models. The subset of features that results in the highest accuracy is the output of the RFE method.

If one feature is eliminated after each model test, this has a corresponding feature ranking in each step. The features that are ranked the highest, and therefore eliminated last, are individually not necessarily the most relevant. Only together this subset of features is optimal in some sense [9]. Therefore, removing several features at a time might be preferential.

Genetic algorithm

Genetic algorithm (GA) also has a list of features as result, which should be used as predictors. However, the exact list might be different and is generated differently. The approach of the genetic algorithm is inspired by Darwin's evolutionary principles of natural selection. The aim of this heuristic is to optimize a population of individuals. In each iteration, the fitness of each individual is determined, after which the genetically fittest individuals are designated to be the ones producing the next generation. Eventually, after a specified number of iterations the 'fittest' individuals remain. This concept can also be applied to

non-evolutionary purposes, like feature selection in machine learning.

In case of usage for feature selection, the individuals are subsets of features. These are indicated as binary vectors, 1 meaning the corresponding feature is used and 0 meaning the corresponding feature is not used [5]. The fitness values of the individuals are calculated using an appropriate performance measure, which in this case is the classification accuracy. The algorithm selects two subsets of features, randomly picks a point to split the corresponding binary vectors, adds the first part of one vector to the second part of the other vector (and vice versa), and finally randomly mutates the resulting binary vectors according to predetermined probabilities of crossover and mutation.

4.3.3 Parameter selection

Also parameter tuning is needed for each of the five models in order to optimize the performances. In other words, several combinations of parameter values are tested to find the optimal parameter settings. R has two options to systematically search for the optimal parameters, which are as follows.

1. *Tunelength*. This method automatically tries several various parameter values. This is called random search. The tunelength indicates the number of different randomly generated values for each model parameter.

2. *Tunegrid*. This method needs human input about which parameter values for each parameter need to be tested. This is called grid search.

The result of both methods is the combination of parameter values that have the best performance. Here, the method tunelength is used for RF and SVM and tunegrid is used for NN. The setting for tunegrid is a certain set of parameter values. The final settings used for tuning are given below.

- Logistic regression: no parameters to be tuned
- Random forest: tunelength = 9
ntree = 250
- Support vector machine: tunelength = 9
kernel = svmRadialSigma
- Neural network: decay = {0.5, 0.1, 0.01, 0.001}
size = {6, 7, 8, 9, 10, 11, 12}
- Ensemble: no parameters are tuned

For random forest this means that the parameter ntree, which is the number of trees to grow, is set to a fixed value of 250. We chose this value, because experiments have shown that more trees do not significantly improve the performance. The parameter mtry is the number of variables randomly sampled as candidates at each split. The final value of mtry is determined by taking one of the nine randomly chosen values that has the best performance.

The support vector machine is used with the Gaussian Radial Basis kernel function and has the two parameters sigma and cost. Nine random values for cost and six random values for sigma are drawn and all combinations are tested.

That is the result of choosing a tunelength of nine in combination with kernel 'svmRadialSigma'. The parameter sigma is the inverse kernel width. The parameter cost is the cost of constraints violation, which is the C-constant of the regulation term in the Lagrange formulation.

Since this problem does not have a high complexity, a neural network with one hidden layer is sufficient. Therefore, this parameter is fixed to one. The remaining parameters are determined using tunegrid. The parameter decay, which is the term in the weight update rule that causes the weight to decay in proportion to its size and avoids overfitting, has been tried with the values 0.5, 0.1, 0.01, and 0.001. The parameter size, which is the number of nodes in the hidden layer, has been tried with the values 6, 7, 8, 9, 10, 11, and 12.

The parameters that should be determined for the ensemble are the same as for the random forest model, ntree and mtry. However, these parameters are set to a fixed value of 500 and 2, for ntree and mtry respectively. The number of variables randomly sampled as candidates at each split is equal to 2, because this is the default value when seven features are used. The number of trees should be chosen carefully, since a high performance of the individual models might lead to overfitting when the number of trees is very high. However, taking 50 trees caused a lower accuracy than taking 500 trees. Therefore, this higher number of trees is chosen.

These tuning settings lead to the optimal parameter values. Those optimal parameter values are used in the final models, which will be listed in Section 5.2.

5 Results

In this section, the obtained results of the used models are presented. These results will be evaluated in Section 6.

5.1 Chosen features

After performing several feature selection methods and trying several subsets of features, the final classifiers are constructed using the following subsets of features.

- Logistic regression: perimeter + concavePoints + radiusSE + fractalDimensionSE + textureWorst + areaWorst + smoothnessWorst + compactnessWorst

- Random forest: areaWorst + concavePointsWorst + perimeterWorst + radiusWorst + concavePoints + textureWorst + texture + concavityWorst + areaSE + concavity + area + smoothnessWorst + radius + perimeter + symmetryWorst + compactnessWorst + perimeterSE + radiusSE + compactness + smoothness + concavitySE + fractalDimensionWorst
- Support vector machine: all original features are used
- Neural network: texture + perimeter + concavePoints + radiusSE + perimeterSE + fractalDimensionSE + textureWorst + areaWorst + smoothnessWorst + compactnessWorst
- Ensemble: LRprediction + SVMprediction + RFprediction + NNprediction + smoothnessWorst + texture + areaWorst

The support vector machine model does not need any prior feature selection. The performance of the model does not significantly improve when selected features are used.

The final ensemble model is constructed with three original features next to the predictions of the four individual models. One negative consequence of using three extra features is a lower AUC value. However, the three most important features are still used because this should make the model more robust.

5.2 Parameter settings

The final parameter values are the result from either the tunelength or tunegrid procedure within R. These values are as follows.

- Logistic regression: -
- Random forest: ntree = 250
mtry = 2
- Support vector machine: kernel = RBF
sigma = 0.023178803
cost = 2
(number of support vectors = 75)
- Neural network: size = 9
decay = 0.01
- Ensemble: ntree = 500
mtry = 2

5.3 Performances

The five models are evaluated on the test set using the selected features and the optimal parameter values mentioned in Sections 5.1 and 5.2. The validation set and test set performance accuracies are compared with the 10-fold cross validation accuracies in Section 5.3.1 to check the overall performance of the models. In addition, the performances between the models are compared in Section 5.3.2.

5.3.1 Reliability of models

In Table 5.1 the 10-fold cross validation, validation set, and test set accuracies are presented. It is important for each of the models to have these three values close to each other, as explained in Section 4.3. From these results can be concluded that the values are significantly close enough, and therefore reliable. The blue colored cells in the table represent the best performance value for all three the cross validation, validation set, and test set. Thus, the support vector machine model has the best cross validation performance, the random forest has the best performance on the validation set, and both the support vector machine and ensemble model have the best performance on the test set.

Model	10-fold CV	Validation set	Test set
LR	0.9664	0.9646	0.9735
RF	0.9589	0.9823	0.9735
SVM	0.9772	0.9735	0.9823
NN	0.9717	0.9735	0.9735
Ensemble	0.9734	-	0.9823

TABLE 5.1: Accuracies of 10-fold cross validation & validation set & test set

The reliability of the model performances can also be tested with confidence intervals. A confidence interval can determine whether the achieved performance value can be trusted, since it shows which values the mean accuracy will be between in 95% of the cases. The confidence intervals are originating from the 10-fold cross validation. Since the 10-fold cross validation is performed 10 times, it produces 100 accuracy values. The confidence intervals are determined using bootstrapping. Based on these 100 values, 1000 times a sample of 100 accuracies is made with replacement. For each sample the mean is calculated, resulting in 1000 times a mean accuracy value. Then, the 2.5% and 97.5% quantiles are taken from these 1000 values, which results in the 95% confidence intervals shown below.

- Logistic regression: [0.9606, 0.9721]
- Random forest: [0.9529, 0.9648]
- Support vector machine: [0.9728, 0.9815]
- Neural network: [0.9656, 0.9773]
- Ensemble: [0.9654, 0.9814]

From these intervals, it can be concluded that the 10-fold cross validation accuracies are reliable, since the intervals are narrow. Hence, the cross validation accuracy is accurate. This, together with the cross validation accuracy, validation set accuracy, and test set accuracy being significantly close to each other, means the overall results are accurate.

5.3.2 Model comparison

The five models are compared based on the accuracy, sensitivity, specificity, and AUC. The accuracy has been used for optimizing the models as good as possible. After determining the final parameters and features, all the models are tested on the test set. The corresponding performance values are shown in Table 5.2.

Model	Accuracy	Sensitivity	Specificity	AUC
LR	0.9735	0.9524	0.9859	0.996
RF	0.9735	0.9286	1	0.9908
SVM	0.9823	0.9524	1	0.996
NN	0.9735	0.9286	1	0.9946
Ensemble	0.9823	0.9524	1	0.9856

TABLE 5.2: Model performances on test set

Again, the blue colored cells indicate for each performance measure which model performs best. All values are close to each other, and many are even the same. This becomes more clear when visualizing it in a bar chart like Figure 5.1. A reason for these similar performance values, which should be taken into account, is the number of instances in the test set. A test set containing only 113 instances and models performing quite similar will often result in the same performance values. More variation would occur if the data set was larger. A performance accuracy of approximately 0.9823 means in this case that 2 out of 113 are misclassified. Having a dataset of 500 instances could have 8 or 9 misclassified to approximately get the same accuracy.

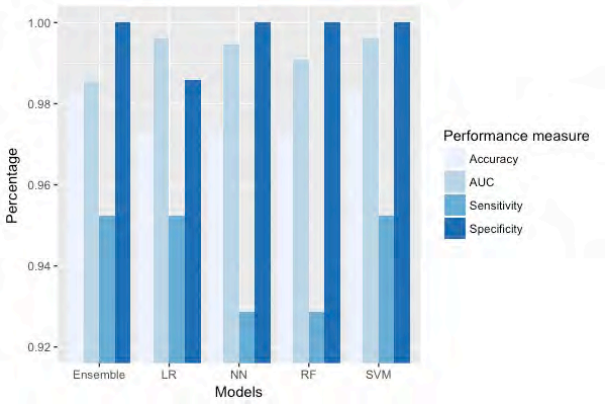

FIGURE 5.1: Model performances on test set

There is not one model with overall distinctive performances. One might argue that the models support vector machine and ensemble are the most remarkable

ones, since they have the highest performance measure in ¼th and ¾th of the time respectively. However, all performances are close to each other. Also the performances of testing on the validation set shows similar performance values. These values are presented in Appendix A. Moreover, those results even show that different models have the best performance values. The accuracy of the random forest model resulting from testing on the validation set is, for example, higher than the accuracy of the support vector machine. However, this is the other way around for the accuracy from testing on the test set.

Other visualizations of the performance values of the validation and test set are presented in Appendix A.

6 Evaluation

Since the models are almost similar in their performances, it is interesting to evaluate the wrongly predicted cases in this section. The accuracies resulting from testing on the test set either have a value of 0.9735 or 0.9823, which corresponds to 3 or 2 cases misclassified, respectively. For evaluation purposes, the cases in the test set are numbered and the misclassified cases per classification model are presented below.

- Logistic regression: {18, 28, 88}
- Random forest: {18, 28, 78}
- Support vector machine: {18, 28}
- Neural network: {18, 28, 30}
- Ensemble: {18, 28}

Clearly, cases 18 and 28 are hard to predict, since all the models predict these two wrong. Both of them are false negatives: they are actually a malignant cell nucleus, but are predicted as benign. The worst situation is saying that a tumor is benign, when it is actually malignant. These false negatives should, therefore, be minimized.

Combining all four individual models together in the ensemble obviously can prevent the occurrence of wrongly predicting the cases 30, 78, and 88. This does not apply to the cases 18 and 28, which is why these should be further analyzed in future research. It could be examined whether these two are hard to predict because of possible abnormal feature values or because of some other reason. What are the characteristics? What is the 'distribution' of the wrongly predicted cases? Is it possible to adjust the data in order to predict them correctly? This

specific investigation to identify the wrong predictions is not in the scope of this research, but is definitely needed in future research to optimize the models.

7 Conclusion

From the results it can be concluded that all five models obtain very promising performances in classifying the possible breast cancer. All models are optimized based on the accuracy, hence the final model should at least have the highest accuracy. Selecting the best-suited model for this specific problem also depends on the sensitivity value, because it is important to have a low number of false positives. The tumor cell nuclei are best predicted by the support vector machine and the ensemble. Both have the highest performance values for accuracy, sensitivity, and specificity. However, the support vector machine is the model which also has the highest value for the AUC. Therefore, the support vector machine model is recommended to use for this specific problem.

Since all the models easily have performance values over 90%, it can be concluded that the features have a high predictive power. This not only might be a reason why all models have such high but also significantly similar performances. Thus, during further research into this problem, all five models are suitable to optimize towards extremely high performance values.

All five models outperform the visual diagnosis of a human being, which is about 90%. Especially the support vector machine and the ensemble models can be very useful for determining and detecting malignant tumor cells in this dataset. These efficient models make very accurate decisions, which is why these decisions are encouraged to support the final decision of the doctors next to the visual diagnosis. Experience and expertise are always important to maintain in the decision-making process. Despite the high performances of the models, they should not replace the doctors but only support their final decision.

8 Discussion

Even though the performance of the models outperforms the visual diagnosis, there are still improvements to be made. As is highlighted in Section 2 'Related work', other researches have similar or even better performances.

The most important improvement to be made is decreasing the number of false negatives. The models have little difficulties with correctly predicting the malignant cells (positives). The number of false negatives is highest compared to the false positives, while these are most important to predict correctly. Therefore, the models should be further developed for a higher malignant cells detection. One approach to achieve a decrease in the number of false negatives could be to optimize the models according to the sensitivity instead of the accuracy, since decreasing the number of false negatives results in an increasing

sensitivity.

A different approach to improve the models is to analyze the incorrectly predicted cases, as already mentioned in Section 6. Examining the wrong predictions could clarify why this part of the data is so hard to predict. Do they have some specific connection? Do they have the same characteristics? If those questions can be answered with yes, that means the models can be adjusted to handle these exceptions.

There are also other issues that might have affected the performances. The five models are suitable to be compared, because all classifiers are trained with the same training set and tested with the same test set. However, comparing the five models with each other also has its disadvantages for this specific problem. One issue to take into account is the size of the total dataset, which is relatively small. Since the dataset needed to be divided into training, validation, and test sets, the test set does not contain a lot of instances. Therefore, there is a strong probability that the models will have the same accuracy. Hence, in future research it would be advisable to collect more data to make a more accurate distinction between the model performances.
A larger dataset also has a positive effect on the training performances of the models. Especially the ensemble has little data to train with, since it should be trained on the predictions of the other four individual models.

Another issue is the dependence of the final performance values on the specific training and test set. One test set might be easier to predict than the other. Perhaps testing on another test set would have resulted in higher performances. Those different performances might have resulted in another model having the highest values, which was actually the case with testing on the validation set. The fact that the outcomes are close to each other, and the different outcomes of testing on the test and validation set, means there is not one model which is obviously performing best. Again, the distinctions between the model performances might be bigger and clearer when larger data sets were used.

One last discussion point is the AUC value of the ensemble. This value is lower than the AUC value of all the other models, while it is expected to be at least as high as the best performing model. The reason for this is unclear and not part of this research. Therefore, it is advised to address in future research in order to improve the model performances.

Appendix A

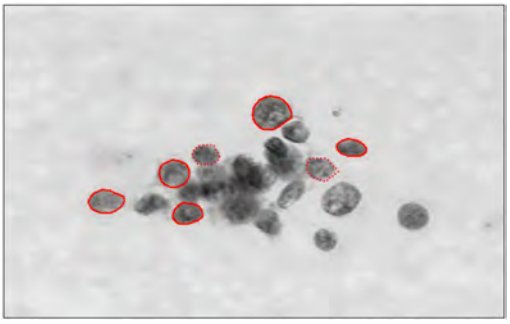

FIGURE A.1: Specifying the boundary of each cell in digitized image [10]

Model	Accuracy	Sensitivity	Specificity	AUC
LR	0.9646	0.9286	0.9859	0.9953
RF	0.9823	0.9762	0.9859	0.9926
SVM	0.9735	0.9762	0.9718	0.992
NN	0.9735	0.9524	0.9859	0.979

TABLE A.2: Model performances on validation set

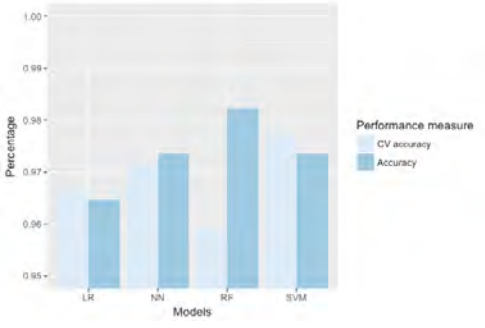

FIGURE A.3: Accuracies of 10-fold cross validation & validation set

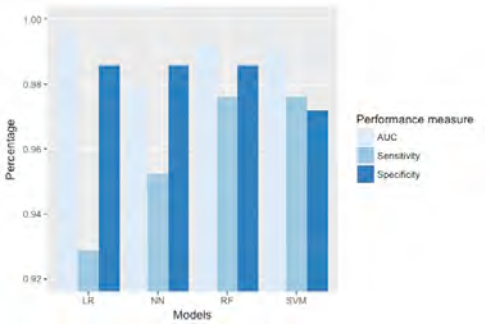

FIGURE A.4: Model performances on validation set

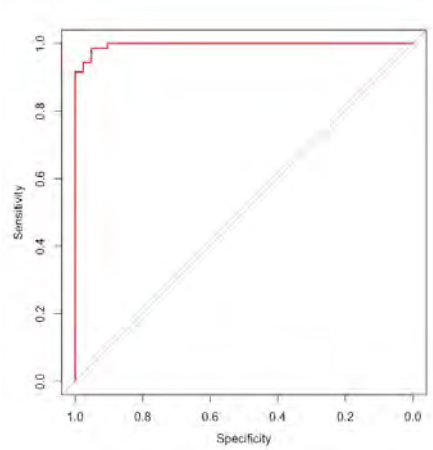

FIGURE A.5: ROC curve of logistic regression model

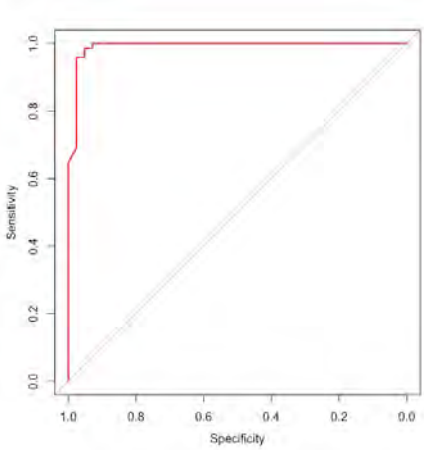

FIGURE A.6: ROC curve of random forest model

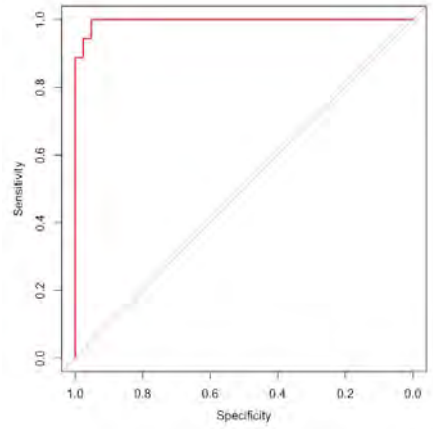

FIGURE A.7: ROC curve of support vector machine model

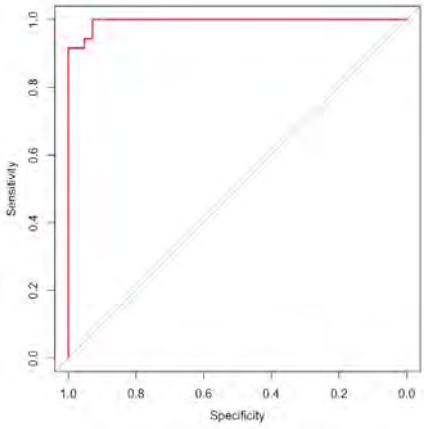

FIGURE A.8: ROC curve of neural network model

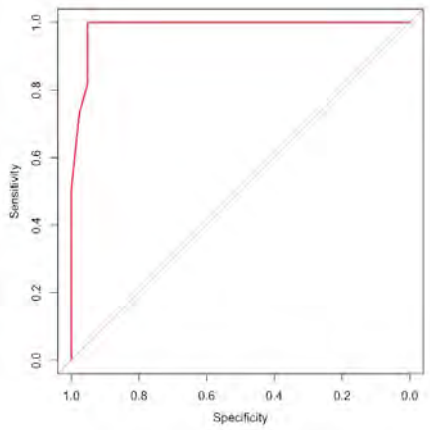

FIGURE A.9: ROC curve of ensemble model

References

[1] Hussein A Abbass. "An evolutionary artificial neural networks approach for breast cancer diagnosis". In: *Artificial intelligence in Medicine* 25.3 (2002), pp. 265–281.

[2] Mehmet Fatih Akay. "Support vector machines combined with feature selection for breast cancer diagnosis". In: *Expert systems with applications* 36.2 (2009), pp. 3240–3247.

[3] David Carrasco. *Random Forest - Modeling the Titanic voyage with R*. May 2017. URL: https://blog.datatons.com/2017/05/16/random-forest-titanic-voyage/.

[4] *Characteristics of Benign and Malignant Tumors*. 2018. URL: http://www.healthhype.com/characteristics-of-benign-and-malignant-tumors.html.

[5] K.M. Faraoun and A. Rabhi. "Data dimensionality reduction based on genetic selection of feature subsets". In: (2006).

[6] *Features Of Malignant Cells*. URL: http://ozradonc.wikidot.com/rb:features-of-malignant-cells.

[7] *Fractal Dimension*. 2003. URL: http://paulbourke.net/fractals/fracdim/.

[8] M C M de Gunst. *Statistical models*. 2013.

[9] Isabelle Guyon et al. "Gene Selection for Cancer Classification using Support Vector Machines". In: *Machine Learning* 46.1 (Jan. 2002), pp. 389–422. ISSN: 1573-0565. DOI: 10.1023/A:1012487302797. URL: https://doi.org/10.1023/A:1012487302797.

[10] *Image showing Xcyt in use*. URL: http://pages.cs.wisc.edu/~street/saves/xcyt1.gif.

[11] Murat Karabatak and M Cevdet Ince. "An expert system for detection of breast cancer based on association rules and neural network". In: *Expert systems with Applications* 36.2 (2009), pp. 3465–3469.

[12] Eric Kim. "Everything You Wanted to Know about the Kernel Trick". In: (2015). URL: http://www.eric-kim.net/eric-kim-net/posts/1/kernel_trick_blog_ekim_12_20_2017.pdf.

[13] *KWF en borstkanker - KWF Kankerbestrijding*. 2018. URL: https://www.kwf.nl/kanker/borstkanker/pages/default.aspx.

[14] Alexis Marcano-Cedeño, Joel Quintanilla-Domnguez, and Diego Andina. "WBCD breast cancer database classification applying artificial metaplasticity neural network". In: *Expert Systems with Applications* 38.8 (2011), pp. 9573–9579.

[15] Tingting Mu and Asoke K Nandi. "Breast cancer detection from FNA using SVM with different parameter tuning systems and SOM–RBF classifier". In: *Journal of the Franklin Institute* 344.3 (2007), pp. 285–311.

[16] Gouda I Salama, M Abdelhalim, and Magdy Abd-elghany Zeid. "Breast cancer diagnosis on three different datasets using multi-classifiers". In: *Breast Cancer (WDBC)* 32.569 (2012), p. 2.

[17] Yongchao Wang and Juanying Xie. "Granular Computing Combined with Support Vector Machines for Diagnosing Erythemato-Squamous Diseases". In: *Health Information Science*. Ed. by Siuly Siuly et al. Cham: Springer International Publishing, 2017, pp. 56–68. ISBN: 978-3-319-69182-4.

[18] William H Wolberg et al. "Computer-derived nuclear features distinguish malignant from benign breast cytology". In: *Human Pathology* 26.7 (1995), pp. 792–796.