UPPSALA
UNIVERSITET

Machine learning techniques for binary classification of microarray data with correlation-based gene selection

By Patrik Svensson

Master thesis, 15 hp

Department of Statistics
Uppsala University

Supervisor: Inger Persson

June, 2016

ABSTRACT

Microarray analysis has made it possible to predict clinical outcomes or diagnosing patients with the help of biological data such as biomarkers or gene expressions. The data from microarrays are however characterized by high dimensionality and sparsity so that traditional statistical methods are difficult to use and machine learning algorithms are therefore applied for classification and prediction. In this thesis, five different machine learning algorithms were applied on four different microarray datasets from cancer studies and evaluated in terms of cross-validation performance and classification accuracy. A correlation-based gene selection method was also applied in order to reduce the amount of genes with the aim of improving accuracy of the algorithms. The findings of the thesis imply that the algorithm s elastic net and nearest shrunken centroid perform best on datasets with no gene selection, while support vector machine and random forest perform well on the reduced datasets with gene selection. However, no machine learning algorithm can be said to consistently outperform any of the other and the nature of the dataset seem to be a more important influence on the performance of the algorithm. The correlation-based gene selection method did however improve prediction accuracy of all the models by removing irrelevant genes.

TABLE OF CONTENTS

1. INTRODUCTION

Development in biotechnology has made it possible to accumulate and store vast amounts of biological information with the help of innovative tools such as microarrays. Microarrays enable scientists to measure and extract information on so called gene expressions, which are a process where genes synthesize into functional products such as proteins. Gene expressions can be used to study the effect of treatments or to discover diseases by comparing a healthy gene expression to the expression of those genes that are infected or changed by the treatment. In the field of genomics and bioinformatics the rise of technology such as microarrays has been the driving force behind improvements in important areas such as disease diagnosis, evaluation of treatment response in patient and cancer research (Cruz & Wishart, 2006).

One tissue sample from a microarray can contain up to tens of thousands of different gene expressions and as such the task of analyzing the data to find any meaningful patterns can be quite overwhelming if done by traditional statistical methods (Tan & Gilbert, 2003). There is a difficulty in performing statistical analysis on microarray data since there is a systematic bias in the output of the microarrays, which is characterized by sparsity and high dimensionality (Fan & Ren, 2006). The problem of high dimensionality is that the amount of genes are often much larger than the amount of samples and sparsity refers to the situation that many of these genes are irrelevant for an analysis.

As a result of the growing complexity in both sheer volume of samples and genes, computationally intensive statistical techniques such as machine learning algorithms have been found to be increasingly popular for analyzing microarrays. (Tan & Gilbert, 2003). In fields such as genomics and bioinformatics machine learning algorithms are often employed for prediction purposes in order to identify tissues of tumors or as a tool to diagnose patients with different types of cancer. The goal of machine learning in this context is to learn what kind of samples or patients which belong to a certain group and then classify new samples or patients into those groups. Machine learning has been used with success in learning to classify patients who have cancer and patients who do not have cancer by learning from the microarray profile (Cruz & Wishart, 2006).

As of today, there are a lot of machine learning algorithms available to choose from with some machine learning algorithms being more prominent than others while at the same time new ones are being developed and gaining popularity. Support vector machines and random forests are two examples of established machine learning algorithms that have been enjoying an immense popularity. They have been used in many kinds of studies and their popularity means that they are easily available in many software packages.

There are however a wide range of machine learning algorithms available which have not been utilized as much in the literature such as nearest shrunken centroid (Tibshirani, 2002) which was developed with microarray analysis in mind. Traditional classification methods like logistic regression have also found utility in high-dimensional context when paired with regularization techniques as done by Zou & Hastie (2005). As such it would be interesting to compare the popular machine learning methods and see if there is any major difference between them.

The problem of high-dimensionality and sparsity can be solved by different variable reduction techniques. Variable reduction in microarray analysis is usually denoted as gene selection since the variables in are gene expression coefficients (Guyon & Elisseeff, 2003). Data from microarrays has a high probability of containing irrelevant and redundant variables which adds noise to the algorithms and is the cause of poor performance and prediction accuracy. In order to improve the quality of the analysis it is therefore often desirable to reduce the features in the dataset. (Guyon & Elisseeff, 2003).

Reducing the variables can be done by applying a variable selection method in the data preprocessing stage which selects the variables (or genes) that are deemed most significant in order to improve the model accuracy. Like in the case of machine learning algorithms, there are many different methods of removing irrelevant variables where the trade-off is between complexity and runtime.

1.1 Objective

The objective of the thesis is to evaluate the performance of different machine learning algorithms when discriminating between healthy and sick cancer patients, and to apply a gene selection method on the microarray data in order to reduce the amount of genes to see if there is an improvement in the performance of the algorithms.

The chosen machine learning algorithms have been selected by their dissimilarity in order to get an overview of the performance of different type of algorithms. More specifically the algorithms chosen are support vector machines, random forest, gradient boosting machine, nearest shrunken centroids and logistic regression with elastic net. Support vector machine and random forest are two of the more established algorithms, while nearest shrunken centroid, elastic net and boosting have not been explored as much and are therefore interesting.

1.2 Disposition

The introduction will be followed by a methodology chapter where the chosen machine learning algorithms and the evaluation measures will be described. After the methodology chapter the data chapter will present the datasets used in the study and the effect of the gene selection method. Then follows a chapter with the results from fitting the algorithms on the datasets and finally a concluding chapter which answers the objectives of the study.

2. LITERATURE OVERVIEW

There has been some research in comparing machine learning algorithms with a range of methods and with different ways of evaluating the algorithms. Most of the comparative studies follow the same formula; two or more datasets are chosen from already published studies on microarray analysis and then machine algorithms are applied in order to evaluate the performance according to a chosen metric. In most cases the metric for evaluating performance is the prediction accuracy.

Ben-Dor et al. (2000) is one of the first examples of a comparative study of machine learning algorithms. Classification rates on samples from gene expression data was evaluated with different methods of evaluation. The study included three different algorithms on three different datasets. The study concluded that the algorithm's performance is influenced by the characteristics of the dataset and that datasets with many irrelevant features may contribute to a poor classification. All of the methods perform similarly however in terms of classification accuracy.

In a similar study to that of Ben-Dor et al. (2000), Dudoit et al. (2002) also compared three algorithms on three different gene expression datasets in a study also focused on classification of cancer tumors. The study found that two of the datasets were easy for the classifiers to handle while the third proved to be more difficult yet again concluding that the characteristics of the datasets affect classification performance. In terms of algorithms they found that in linear discrimination and nearest neighbor perform slightly better than the decision tree classifiers.

Sung Bae & Hong-Hee (2003) evaluated the performance of five different classification algorithms on three different DNA microarrays with cancer outcomes. They found that ensemble methods, which combine different classifiers to form one classifier, perform best in terms of classification accuracy.

In a rather ambitious undertaking, Lee et al. (2005) compared 21 different machine learning algorithms with each other on seven different gene expression datasets. The general conclusion from the study was that the method of gene selection had the greatest effect on classification performance and that classical methods such as linear discriminant perform well when applied on datasets with gene selection. However, in terms of overall performance the support vector machines perform best with or without gene selection with the random forest algorithm close behind.

However, the conclusions from Lee et al. (2005) stood in contrast with Li et al. (2004) who found that the gene selection methods are not important and the conclusions are rather in line with the earlier studies of Ben-Dor et al. (2000) and Dudoit et al. (2002) in which the characteristics of the datasets had the greatest impact on performance. The chosen machine learning algorithm was considered the most important factor for achieving a high classification rate. Most of the algorithms perform well, but the support vector machine was regarded as the best algorithm.

Pirooznia et al. (2008) analyzed eight different public datasets from microarray studies with eight different machine learning algorithms. Some feature selection methods were also included and the performance of the models were tested before and after feature selection. The study concluded that the nature of the dataset influences the accuracy of the algorithms and that noise in the data

is a problem. Regarding feature selection it had a beneficial effect on the prediction accuracy and all models perform better with the chosen feature selection methods but no algorithm was the clear winner. Support vector machine and random forests consistently perform very well on each of the datasets.

Önskog et al. (2010) studied the effects of normalization and gene selection on microarray datasets and then compared the performance on eight different datasets. The gene selection methods were all filter-based and they found that there was a positive relationship between gene selection with t-statistics and the performance of machine learning techniques. This study also confirmed that the performance of machine learning algorithms differs between datasets but they found that support vector machines perform consistently well.

In a more recent study Raza & Hasan (2015) compared ten different machine learning algorithms on a single prostate cancer dataset in order find the best performing algorithm and they also used t-statistics as the chosen method of feature selection. They found that the Bayes Net perform the best while popular algorithms such as support vector machine and random forests did not perform as well.

The main conclusion from the literature overview is that the performances of the machine learning algorithms are greatly influenced by the nature or characteristics of the dataset on which the algorithm is applied. This shows that in order to evaluate machine learning algorithms it is advisable to test the chosen algorithms on more than one dataset since it is hard to generalize the performance on a single dataset.

3. METHODOLOGY

In this chapter the chosen gene selection method is described together with a brief overview of the chosen machine learning algorithms. In the end of the chapter the method for evaluation of the algorithms will also be described.

3.1 Correlation-based feature selection

The goal of a feature selection method is to reduce the available features (variables) and select only the most important features for predicting a class. There are in general two approaches for reducing features; the wrapper approach and the filter approach (John et al. 1994). Wrapper methods apply a statistical learning algorithm on the data which is then evaluated and the model chooses a subset of features which maximizes the performance of the algorithm. Filter methods on the other hand evaluate subsets of features on the characteristics of the data with some chosen criterion such as t-statistics, correlation between features and other univariate scoring methods. (Inza et al. 2004, Kuhn et al. 2013) Filter methods are usually chosen for high-dimensional data since they are more computationally efficient (Yu & Liu, 2003).

The correlation-based feature selection (CFS) developed by Hall (1999) is a filter method designed to be fast and efficient. The method is based on the hypothesis "Good feature subsets contain features highly correlated with the class, yet uncorrelated with each other." (Hall & Smith, 1997). The heuristic in which the goodness-of-fit is measured is based on the Pearson's correlation where all of the variables have been standardized.

Let *n* be the number of components, r_{xc} be the correlation between the summed components and an outside variable, r_{xi} the average correlation between components and an outside variable and r_{xj} the average inter-correlation between components. (Hall & Smith, 1997). Then the equation

$$r_{xc} = \frac{n r_{xi}}{\sqrt{n + n(n-1) r_{xj}}}$$

gives the merit r_{xc} of which the subsets are evaluated. The "outer variable" in this case is the class of the response variable which was stated in the hypothesis. Those subsets with the highest merit are then reduced from the dataset. The CFS filter method can be used with both backward elimination and forward selection with backward elimination.

3.2 Random forest

Random forest (RF) is an ensemble learning method that uses decisions trees for classification. It combines multiple decisions trees to form a final classifier. By generating an ensemble of multiple decision trees that are uncorrelated and then averaging the results, the slight differences in the trees will occur due to the variations between each decision tree.

The algorithm of random forest follows three steps (adapted from Hastie et al., 2002):

1. Draw a bootstrap sample from the training data.
2. Grow a random-forest tree to the bootstrapped data. At each node randomly sample the predictors and choose best split among the chosen.
3. Predict new data by aggregating the prediction of the splits. For classification this means the majority of the votes for splitting.

The final classifier of a random forest can be several hundreds of decision trees which has voted for a class with different amount of depth in the tree. The randomness in the resampling of each tree ensures that the variance is low and that the bias does not increase. Random forest also has a variable importance function in the algorithm which can be extracted from the model. Variable importance is seen as the variable which is most commonly used when splitting the decision trees. The logic is that a variable that is chosen when splitting must have a greater impact on the classification than variables that are not chosen. (Hastie et al., 2005)

3.3 Elastic net

The elastic net (Enet) is a regularization method combines the penalties of ridge regression and the lasso and is useful for high-dimensional situations when the amount predictors are much bigger then observations. Following the definition of Zou & Hastie (2008), consider the standard linear model:

$$Y = \beta_0 + \beta_1 X_1 + \ldots + \beta_N X_N$$

The estimated least square coefficients of the ridge regression model is then defined as

$$\hat{\beta}(ridge) = arg\,min\|Y - X\beta\|^2 + \lambda\|\beta\|^2$$

where the penalty $\|\beta\|^2$ is the sum of the squared betas. The ridge regression penalty restricts the estimated coefficients and penalizes them if β_i takes on large values. The lasso is defined as

$$\hat{\beta}(lasso) = arg\,min\|Y - X\beta\|^2 + \lambda\|\beta\|_1$$

where the penalty $\|\beta\|_1$ is the sum of the betas . With the lasso penalty it is possible to shrink the least square coefficients to zero if λ is large enough and such coefficients are removed from the final estimated model. (Zou & Hastie, 2008) This variable selection feature is an important part of the lasso since it removes highly correlated variables and automatically creates a parsimonious model.

However, the drawbacks of the lasso is that is unable to handle high-dimensional data since it cannot select more variables than observations. The lasso also performs poorly and has poor selection of the variables when there are many pairwise correlations. As a remedy for these problems of the lasso Zou & Hastie (2005) proposed a regularization method called elastic net. The aim of the method was enable the lasso to select more variables than observations if there was a need for it and to handle groups of correlated variables (Zou & Hastie, 2008). The naïve elastic net is defined as

$$L(\lambda_1, \lambda_2, \beta) = \|Y - X\beta\|^2 + \lambda_2\|\beta\|^2 + \lambda_1\|\beta\|_1$$

Where $\|\beta\|^2$ and $\|\beta\|_1$ is the linear combination of the ridge and lasso penalties and is called the elastic net penalty. However, the double shrinkage in the naïve elastic net introduces bias and to correct this the corrected elastic net is rescaled by multiplying the estimated coefficients by $(1+\lambda_2)$. The final elastic net is then defined as

$$\hat{\beta}(elastic\,net) = (1+\lambda_2)\hat{\beta}(naive)$$

This correction preserves the variable selection of the naïve elastic net via the lasso but also undoes the bias of the double shrinkage and leads to better predictions. (Zou & Hastie, 2008)

3.4 Nearest shrunken centroids

Nearest shrunken centroids was put forward by Tibshirani et al. (2002) and was specifically developed for high-dimensional problems such as the analysis of data from microarrays. The overall aim of the method is to shrink the centroids of each class towards the overall centroid in order to find the genes that are most useful for predicting each class (Tibishirani et al., 2002). This is accomplished by calculating the t-statistic d_{ik} for gene *i* by normalizing each feature by the within-class standard deviation.

Let x_{ij} be the expression for genes $i=1,2,\ldots,p$ and samples $j=1,2,\ldots,n$. Let $x_{ik}=\sum x_{ij}/n_k$ be the centroid for class *k* and $x_i=\sum x_i/n$ be the overall centroid. The t-statistic d_{ik} is then given by

$$d_{ik} = \frac{\acute{x}_{ik} - \acute{x}_i}{m_k * s_i}$$

Where s_i is the within class standard deviation for gene *I*, and x_{ik} is the *i*th component of the centroid for classes $K=1,2,\ldots,K$ and $m_k=\sqrt{(1/n_k - 1/n)}$. As such the denominator becomes the standard error for the numerator (Tibshirani et al, 2002).

The shrunken centroids are then given by:

$$\acute{x}'_{ik} = \acute{x} + m_k s_i d'_{ik}$$

which in this context is called *soft thresholding* (Tibshirani et al, 2002) where the absolute value of d_{ik} is reduced by a value Δ and if the value less than zero it is set to zero. This has the consequence that if Δ is very large, many of the genes will be reduced to zero and eliminated from the class as they are redundant. The value of Δ is chosen by iterative methods.

3.5 Support vector machine

Support vector machines (SVM) discriminate between the two groups by creating a line or hyperplane with the largest possible margin to the nearest data points from both groups (Yu and Kim, 2012). SVM builds upon the concept that many problems are linearly solvable if the dimensionality is high enough. SVM works by optimizing the set (x_i, y_i) according to (Hsu. Et al, 2003):

$$min_{f_0}\left(\frac{1}{2}w^T w + C\sum \xi_i\right)$$

$$y_i\left(w^T \phi(x_i) + b\right) \geq 1 - \xi_i$$

Where C is the penalty of the error term, ξ is the error term and W is a vector of weights. The vector x_i is projected into the high dimensional space by the function ϕ:

$$K(x_i, x_j) = \phi(x_i)^T \phi(x_j)$$

Where K is the so called kernel function. The kernel function is to be chosen and decides the dimension. SVM was intended for linear relationships but the kernel function makes it possible to choose other types of kernels such as radial or polynomial. For high-dimensional problems the linear kernel is recommended by Hsu et al. (2003).

3.6 Gradient boosting machine

A boosting algorithm is a classifier built upon the concept that many weak learners will form a strong learner (Hastie et al., 2005). Just like random forest, a gradient boosting machine is an ensemble method which builds many classification trees in order to form one classifier. There are however some fundamental differences in how these methods arrive at the final classifier. While random forests average many trees to build a final classifier, the boosting machines sequentially fits new models to the ensemble with some weight (Natekin & Knoll, 2013). The gradient boosting machine algorithm has five steps (adapted from Friedman, 2001):

Algorithm 1: Gradient boosting machine

1. Initialize the start function f_0 with a constant and for $t = 1$ to N:
2. Calculate the negative gradient $g_t(x)$
3. Fit the base-learner model $h(x, \theta_t)$
4. Find the gradient descent p_t

$$p_t = argmin\sum \psi\left[y_i, f_{t-1}(x_i) + ph(x_i, \theta_t)\right]$$

5. Update the function estimate in step 1 with the gradient and the base-learner

$$f_t \Leftarrow f_{t-1} + p_t h(x, \theta_t)$$

6. End of algorithm.

The base learner $h(x,\theta_t)$ as seen in the first step of algorithm 1 can be any chosen statistical model such as linear regression or classification trees. The loss function $\psi(y,f)$ in step four of the algorithm can be arbitrary chosen but is usually chosen by the characteristics of the response variable. In the binary classification case a binomial loss function is used. (Natekin & Knoll, 2013). The gradient boosting machine tries to approximate function f_0 by minimizing the loss function $\psi(y,f)$. In order to avoid overfitting a weight is added to each iteration and random sub-sampling is used on the data where the base learner $h(x,\theta_t)$ is trained on (Hastie et al., 2008).

3.7 Model evaluation

For evaluating the performance of the algorithms cross-validation, ROC curves and metrics such as accuracy will be used. The notation used for the following formulas are the following:

- TP: True positive
- FP: False positive
- TN: True negative
- FN: False negative

A true positive is when an observation is classified as a positive when it is positive and a false positive is an observation that is classified as a positive when it is actually negative. The same relationship applies for the true negative and false positive. These are all outcomes when applying a model to the dataset. To get the full number of positives (*Np*) in the dataset you add the TP and FN and to get the full number of negatives (*Nn*) you add TN and FP. (James et al., 2013).

In order to assess the accuracy of a binary classification model the measures precision sensitivity and specificity are often employed. The formulae for the measures can be found in Table 1.

Table 1: Formulae for prediction accuracy measures

MEASURE	FORMULA
ACCURACY	$\frac{TP+TN}{Nn+Np}$
PRECISION	$\frac{TP}{TP+FP}$
SENSITIVITY	$\frac{TP}{Np}$
SPECIFICITY	$\frac{TN}{Nn}$

Accuracy is the model's ability to correctly identify the observations while the precision measures the model's ability to distinguish between positive and negative observations. The sensitivity measures how many positive classifications are found of all the available positive classifications while the specificity has the same interpretation for the negative observations. (James et al., 2013).

ROC Curves and AUC

The ROC Curve is a way to visualize the performance of a binary classifier. The ROC curve plots the true positive rate (sensitivity) against the false positive rate (1 - specificity) for different cut-off points (James et al., 2013).

If the ROC is a way to visualize the performance of a classifier then the Area Under the Curve (AUC) is a way to summarize the performance of a model with just one value. The value is the area under the ROC curve and is a ratio between 0 and 1 where a value of 1 is a perfect classifier while a value close to 0.5 is a bad model since that is equivalent to a random classification (James et al., 2013).

K-fold cross validation

Cross-validation is a training technique used for evaluating the performance of a machine learning algorithm. By splitting the dataset *k* times into random groups without replacement which are about equal in size, *k*-1 groups are then used for training. The last group is held out for testing the model. In k-fold cross validation this process is repeated k-times where each fold is the process where the model is evaluated on each group. Each iteration is then assessed by a chosen performance metric such as AUC or accuracy. Cross-validation is known to balance the bias and variance when being used and is known to be computationally efficient compared to other validation techniques and in particular when the amount of features are greater than the amount of samples. (James et al, 2013)

3.8 Study design

Each dataset is split into two subsets; a training set and a test set where the training set is made up of 70% of the available observations and the test set consists of the last 30%. The observations are then randomly put into each set as to avoid any kind of systematic bias. The amount of data split was chosen arbitrary; in the literature there are examples of 20/80 splits, 2/3 splits and even 50/50 splits for training and testing. The chosen split is decided by the researcher with the amount of samples available taken into consideration.

Each machine learning algorithm is then fitted on the training data with ten cross-validation folds which is repeated five times. The performance metrics on the training data given by the cross-validation is therefore the average of 50 different models. The trained algorithm is then applied to predict the untouched test data which gives us a single performance metric for the fitted test models which is used for evaluation of the performance.

The performance metrics are then compared between the training and testing to see if there is any difference between the two. The performance metric AUC is used to compare the training and test sets and prediction accuracy is used to compare the filtered and full datasets. This process is done two times; once on the dataset will all of the features and then on the datasets with the CFS filter applied. The results are then compared.

4. DATA

There are many datasets available from published studies with microarray data and some of the datasets have been used in other articles exploring machine learning algorithms. Alon (1999) is one of those well-known datasets and has been utilized by many researchers. The datasets chosen to participate in this thesis have all been published and the datasets are characterized by having more features than samples, the features are gene expressions from microarrays, the response variable is binary and all of the features are continuous variables. The properties of the datasets can be seen in table 2 where *n* is the number of samples and p is the number of variables.

Table 2: Overview of the datasets

DATASET	AUTHOR	N	P	CLASSES	POSITIVE CLASS RATIO
BREAST CANCER I	Gravier (2010)	168	2905	2	0.51
COLON CANCER	Alon (1999)	62	2000	2	0.55
BREAST CANCER II	West (2001)	49	7129	2	0.96
CNS DISORDER	Pomeroy (2002)	60	7128	2	0.54

Gravier (2010) examined small, invasive ductal carcinomas without axillary lymph node involvement to see if it could predict the metastasis of small node-negative breast carcinoma. The study involved 168 patients which was followed by five years and the event of interest was if the patients developed metastasis. 111 patients were classified as good (with no event experienced) while 57 patients experienced the event and were classified as poor.

West (2001) also analyzed invasive ductal carcinomas and in form of breast tumors to see if it possible to discriminate tumors on the basis of estrogen receptor status. The chosen 49 tumors are classified as receptor-positive or receptor-negative depending on whether they tested positive or negative for both estrogen and progesterone.

Alon (1999) created a classifier for colon tumors which was able to discriminate between colon tumors from normal colon tissues. Of the 62 samples 40 are classified as colon tumors and 22 are colon tissue.

Pomeroy (2002) classified clinical outcomes of embryonal tumors in the central nervous system. They classified patients after their treatment outcome which was death or survival. 60 patients participated in the study and was classified as survivors while 39 was classified as failures (death.)

4.1 Preprocessing

The values of the variables have been transformed with the log2-scale and then scaled between 0 and 1. This is not required for all of the machine learning algorithms, however, some algorithms perform better when using scaled values. Hsu et al. (2003) note that scaling is beneficial for support vector machine for two reasons; to avoid large numbers getting greater weights than

smaller numbers and to make the numerical calculations more efficient. For consistency purposes in this thesis, all of the models are trained on scaled data. The reason for log-transforming the variables is to create a similar order of magnitude.

The data was scaled between [0, 1] as per recommendation of Hsu et al. (2003) by the min-max normalization technique where the normalized value is given by:

$$x_{norm} = \frac{x - min}{max - min}$$

4.2 Gene selection

As explained in the methodology chapter the gene selection was achieved by applying the correlation-based feature selection filter by Hall (1999). Table 3 shows how many genes were chosen by the filter.

Table 3: Dataset after applying correlation-based feature selection (CFS)

DATASET	GENES BEFORE	GENES AFTER	GENES REMOVED (%)
BREAST CANCER I	2905	34	98.8
COLON CANCER	2000	27	98.6
BREAST CANCER II	7129	36	99.5
CNS DISORDER	7128	39	99.5

5. RESULTS

In the following chapter the performance of the models are presented. First the results from the full datasets with all features enabled and then the results from the CFS datasets. At the end of the chapter the performance between the two are compared.

5.1 Full datasets

5.1.1 Breast cancer I

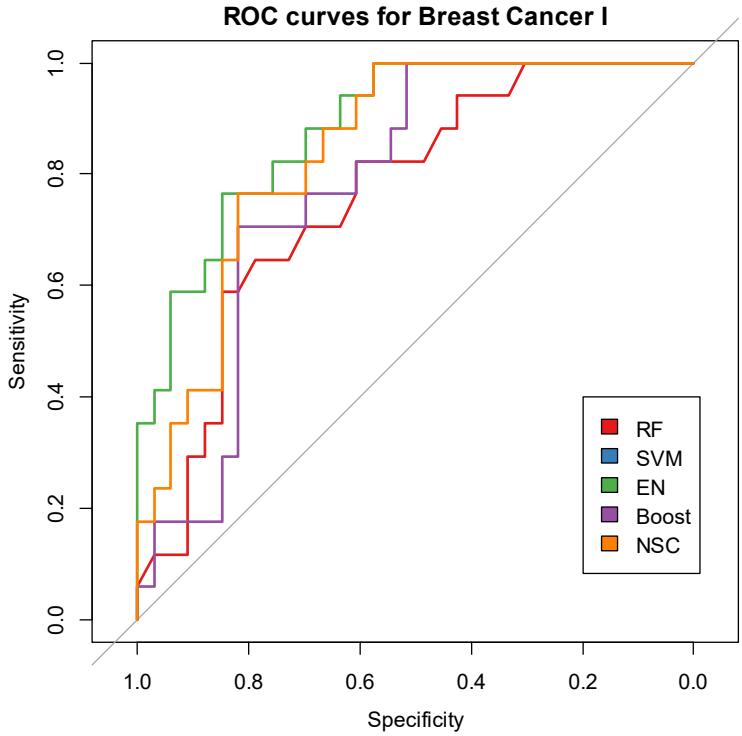

Figure 1: ROC curves for the full Breast Cancer I dataset

Table 4: ROC performance on the full Breast Cancer I dataset

	AUC		SENS		SPEC	
Model	Train	Test	Train	Test	Train	Test
Random forest	**0,83**	0,76	**0,93**	**0,91**	0,29	0,24
SVM	0,80	**0,88**	0,92	0,89	0,40	0,53
Boost	0,80	0,78	0.90	0,82	0,50	0,29
Elastic net	0,80	**0,88**	0.91	**0,91**	0,51	0,59
NSC	0,78	0,84	0.84	0,85	**0,60**	**0,65**
Average	0,80	0,83	0,90	0,87	0,46	0,46

Table 4 shows the performance metrics for both the training and test set for Breast cancer I. Looking at the average AUC it can be seen that the algorithms in general perform better on the test data. Enet and SVM perform best on the test data with a shared AUC of 0.88. Both algorithms also have relatively low scores on the training data so the algorithms are relatively underfitting when learning the training set. Random forest on the other hand scored the highest on the training data with an AUC of 0.83 but lowest on the test data with 0.76 and as such the algorithm overfits on the training data.

The plotted ROC curves for the fitted test models can be seen in figure 1 and the models are all behaving similarly for different thresholds. SVM cannot be seen in the figure since they share the same values as the elastic net. The average values for the sensitivity and specificity implies that the algorithms are better on detecting the patients who have cancer and have difficulties recognizing patients that did not have cancer. On the test data random forest was in particular bad on predicting the true negatives with a specificity of 0.24 and the nearest shrunken centroid have the best specificity of 0.65. The random forest and elastic net perform best on detecting cancer positives with a value of 0.91.

Table 5: Prediction accuracy on the Breast cancer I dataset

Model	Accuracy	95% CI	
Random forest	0,68	0,53	0,80
SVM	0,76	0,62	0,87
Boost	0,64	0,49	0,77
Elastic net	**0,80**	**0,66**	**0,90**
NSC	0,78	0,64	0,89

In table 5 the prediction accuracy on the test set can be seen. Elastic net perform the best with an accuracy of 80% correctly identified patients while the boosting algorithm have the worst accuracy.

5.1.2 Colon cancer

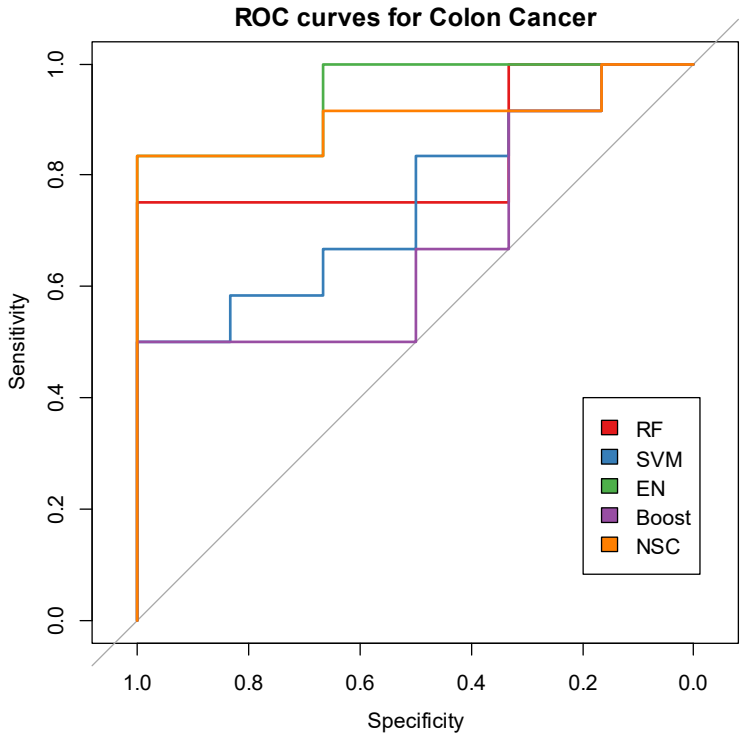

Figure 2: ROC curves for the full Colon cancer dataset

Table 6: ROC performance on the full Colon cancer dataset

	AUC		SENS		SPEC	
Model	Train	Test	Train	Test	Train	Test
Random forest	0.88	0,83	0.57	0,33	0.89	**0,83**
SVM	0,87	0,75	0,72	0,33	0,89	**0,83**
Boost	0.89	0,68	0.67	0,33	0.87	0,67
Elastic net	0.90	**0,94**	0.69	0,83	**0.94**	**0,83**
NSC	**0.90**	0,90	**0.83**	**1**	0.77	0,5
Average	0,89	0,82	0,70	0,57	0,87	0,73

Table 6 shows the ROC values for the colon cancer dataset with all features. The algorithms perform very similar on the training data with every value at the high end of 0.8. On the test data however some of the models do not perform as similarly. The boosting algorithm and SVM perform much worse on the test data implying that they overfitted on the training data. Elastic net improves on the test data and hits the highest AUC value of 0.94. NSC perform consistently over both the training and test data with not much deviance from its training value.

Turning the attention to the sensitivity and specificity it can be seen from the average values that the true positives (tissues with cancer) were in general harder to identify with two obvious exceptions. On the test data the NSC manages to accurately predict all of the true positives with the elastic net managing a high value of 0.83. The rest of the algorithms did not perform as well with only 0.33. The NSC however perform worst on predicting the true negatives (tissue without cancer) on the test set with elastic net, random forest and SVM performing the best. The plotted ROC curves for the algorithms can be seen in figure 2.

Table 7: Accuracy on the testing dataset for colon cancer

Model	Accuracy	95% CI	
Random forest	0,67	0,41	0,87
SVM	0,67	0,41	0,87
Boost	0,56	0,31	0,75
Elastic net	**0,83**	**0,59**	**0,96**
NSC	0,67	0,41	0,87

From table 7 it can be seen that the most accuracy algorithm was elastic net with 83% correct predictions. Random forest, SVM and NSC all perform similarly while boosting was the worst.

5.1.3 Breast cancer II

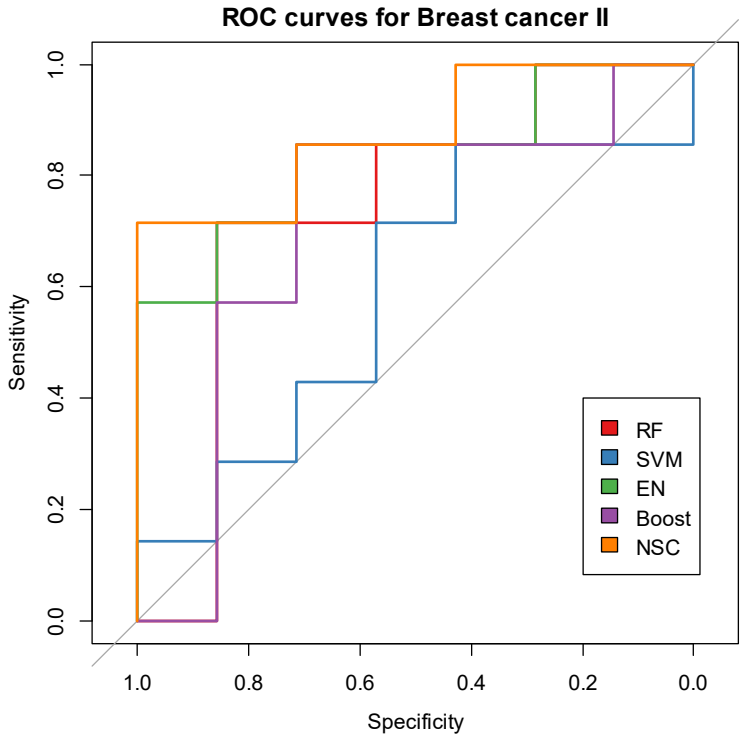

Figure 3: ROC curves for the full Breast Cancer II dataset

Table 8: ROC performance on the full Breast cancer II dataset

	AUC		SENS		SPEC	
Model	Train	Test	Train	Test	Train	Test
Random forest	0.94	0,74	0.84	0,43	0.79	**0,86**
SVM	0,79	0,59	0,63	0,57	0,74	0,57
Boost	0.92	0,71	0.83	0,43	**0.80**	**0,86**
Elastic net	**0.98**	0,84	**0.99**	**0,86**	0.55	0,57
NSC	0.97	**0,88**	0.96	0,71	0.76	0,71
Average	0,92	0,75	0,85	0,6	0,73	0,71

The average values in table 8 show that the training data fits on average with a higher AUC than the test data with four of the models all reaching values above 0.90. No model reaches values above 0.90 in on the test set so all of the algorithms overfitted on the training data. SVM does not perform well on either of the datasets and is the worst algorithm. NSC fits the best model on the test data with an AUC of 0.88.

From the ROC curves in figure 3 the SVM stands out as the worst performer. With values in the 0.50s for both sensitivity and specificity it is close to being as good as a random guess and this is visualized by the line of SVM hugging the grey line. In this particular dataset no class is standing out as more difficult than the other. The average values of the sensitivity and specificity are close to each other. Looking closer at the individual algorithms it can be seen that there is a trade-off between a high specificity and a lower sensitivity for the highest performing algorithms.

Table 9: Accuracy on the testing dataset for Breast Cancer II

Model	Accuracy	95% CI	
Random forest	0,64	0,35	0,87
SVM	0,57	0,29	0,82
Boost	0,64	0,35	0,87
Elastic net	**0,71**	**0,42**	**0,92**
NSC	**0,71**	**0,42**	**0,92**

Elastic net and NSC have the highest accuracy according to Table 9 which follows the results in table 8 where they also had the highest AUC. SVM does not perform as good while Random forest and boosting reached the same accuracy in the middle.

5.1.4 CNS Disorder

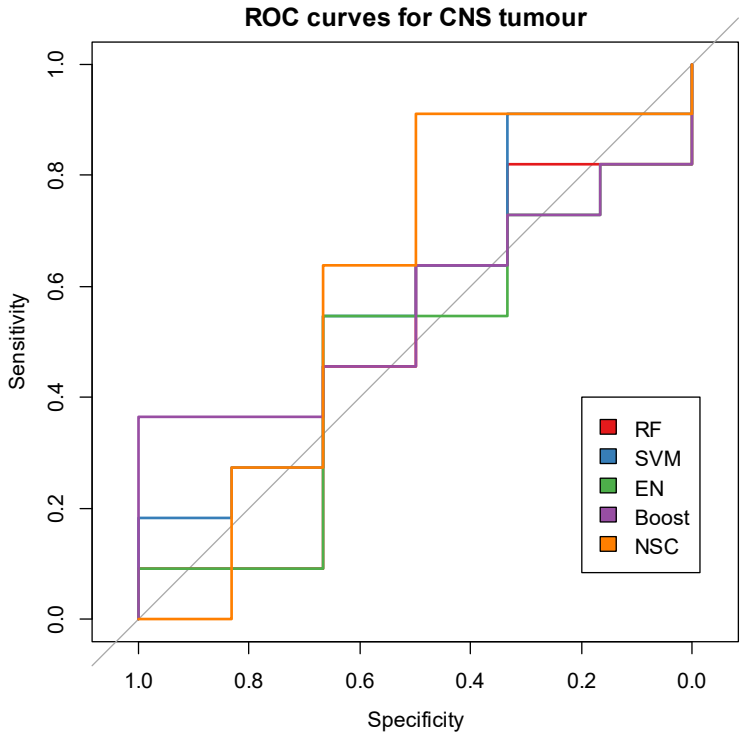

Figure 4: ROC curves for the full CNS tumor dataset

Table 10: ROC performance on the full CNS tumor dataset

	AUC		SENS		SPEC	
Model	Train	Test	Train	Test	Train	Test
Random forest	0,71	0,49	0,29	0,17	0,88	0,91
SVM	0,72	0,58	0,17	0	**0,91**	**1**
Boost	0,73	0,56	0,39	0,17	0,84	0,72
Elastic net	**0,77**	0,47	0,47	**0,33**	0,83	0,91
NSC	0,72	**0,61**	**0,65**	**0,33**	0,64	0,36
Average	0,73	0,54	0,39	0,2	0,82	0,78

The performance metrics for the CNS tumor dataset can be found in table 10. This was a difficult dataset for the algorithms with an average AUC on the test set of only 0.54. The fitted training algorithms perform a bit better with an average AUC of 0.73 but the data must be noisy and the models overfitted since they fail to perform on the test data. The patients who had cancer were the hardest to identify with the highest sensitivity of 0.33 and SVM does not manage to identify even one on the test data. The SVM does however manage to identify all of the true negatives (patients without cancer) and in general the algorithms manage to identify the true negatives more often than the true positives. This is not good at all; by intuition it is better to be able to identify patients who has cancer than identifying patients who does not have cancer.

The plotted ROC curves in figure 4 visualizes the dismal performance all of the algorithms where they follow the middle line and as such can be said to perform as good as a random guess.

Table 11: Prediction accuracy on the testing dataset for CNS tumor

Model	Accuracy	95% CI	
Random forest	0,65	0,38	0,86
SVM	0,65	0,38	0,86
Boost	0,53	0,28	0,77
Elastic net	**0,71**	**0,44**	**0,87**
NSC	0,35	0,14	0,62

The predicted accuracy in table 11 is interesting. NSC which has the highest AUC scores for the test data is actually the worst in terms of accuracy and Elastic net which has the lowest AUC has the highest accuracy. In this case it seems that the specificity plays an important role. Since all of the algorithms except NSC has good results on the specificity it means that they are better at identifying patients without cancer and correctly classifies most of them while NSC has low scores in both patient groups. This is also an example that a single performance metric is not always satisfactory in order to decide the best algorithm.

5.2 CFS datasets

5.2.1 Breast cancer I

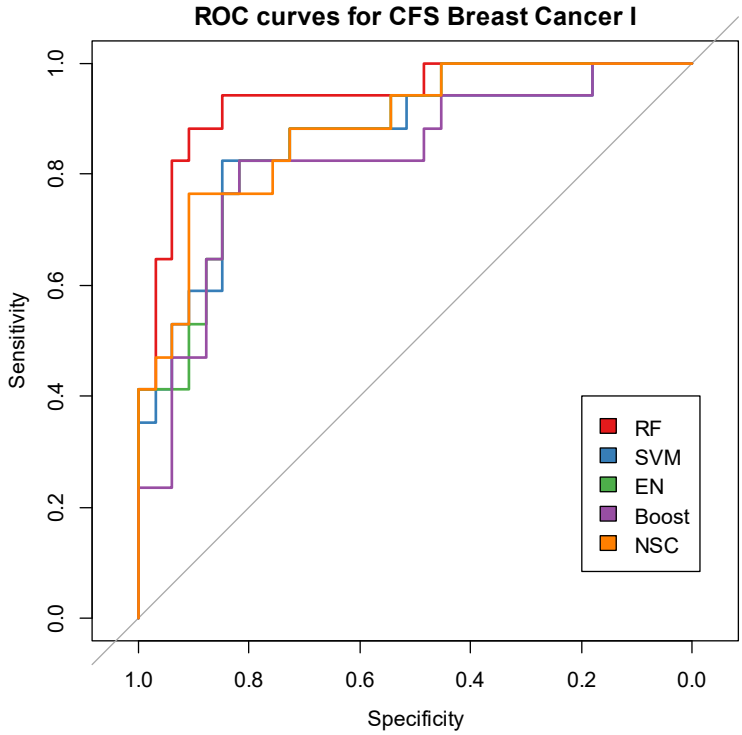

Figure 5: ROC curves for the CFS Breast Cancer I dataset

Table 12: ROC performance on the CFS Breast cancer I dataset

	AUC		SENS		SPEC	
Model	Train	Test	Train	Test	Train	Test
Random forest	**0,94**	**0,94**	0,95	**0,97**	0,61	0,47
SVM	0,82	0,86	0,91	0,91	0,52	0,59
Boost	0,92	0,83	0,90	0,85	**0,75**	**0,65**
Elastic net	0,91	0,87	0,94	0,94	0,67	0,41
NSC	0,92	0,88	**0,96**	**0,97**	0,61	0,41
Average	0,90	0,88	0,93	0,93	0,63	0,51

In table 12 it can be noted that there are only slight differences between the average values of the training set and the testing set. Random forest performs well on the reduced dataset and has the same performance on both the training and testing showing that the model captures the characteristics of the data. The worst model in terms of AUC was the boosting machine but all of the models perform reasonably well. Looking at the sensitivity it can be seen that the algorithms perform well in identifying the patients which had cancer but have a more difficult time identifying the negative patient outcomes.

Table 13: Accuracy on the CFS testing dataset for Breast Cancer I

Model	Accuracy	95% CI	
Random forest	**0,80**	0,66	0,90
SVM	**0,80**	0,66	0,90
Boost	0,78	0,64	0,89
Elastic net	0,76	0,62	0,87
NSC	0,78	0,64	0,89

The random forest and SVM recorded the highest accuracy when testing on the held out data as can be seen in table 13. All of the performances are however close to each other and no one really stands out as the best algorithm in terms of accuracy.

5.2.2 Colon cancer

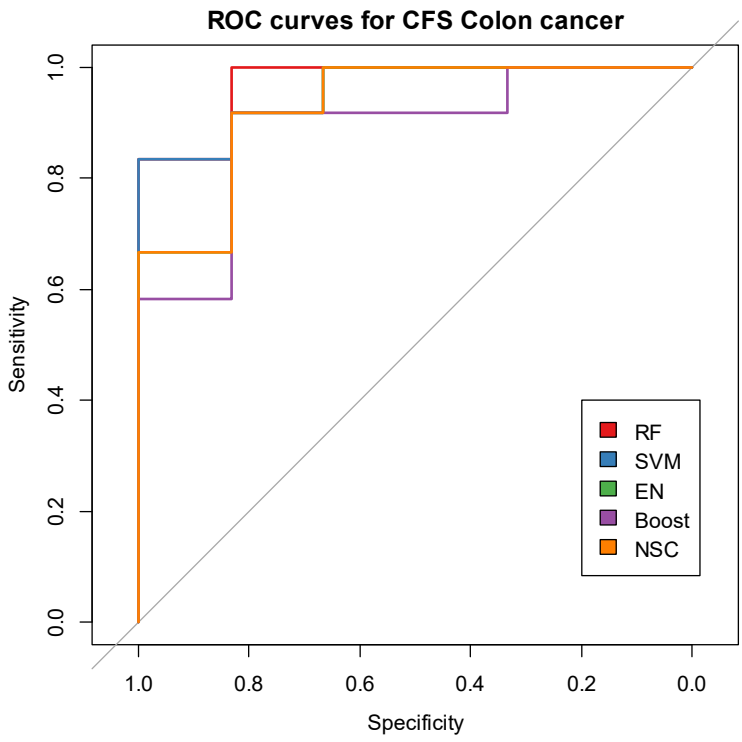

Figure 6: ROC curves for the CFS colon cancer dataset

Table 14: ROC performance on the CFS Colon cancer dataset

	AUC		SENS		SPEC	
Model	Train	Test	Train	Test	Train	Test
Random forest	**0,98**	0,93	**0,91**	**0,83**	**0,92**	0,92
SVM	0,94	**0,96**	0,77	0,5	**0,92**	**1**
Boost	0,91	0,89	0,72	0,67	0,90	0,92
Elastic net	0,96	0,93	0,87	**0,83**	0,90	0,92
NSC	0,96	0,93	0,87	**0,83**	0,89	0,8
Average	0,94	0,93	0,83	0,73	0,91	0,92

For the reduced colon cancer dataset the random forest perform best on the training set with an AUC of 0.98 and all of the models achieved an AUC above 0.90. The performances are similar on the testing set but with SVM now being the best model. The ROC curves in figure 6 visualize the very good performance of the models.

In this case the tissues with cancer proved harder to identify on the testing set in general with a lower average sensitivity but the performance on the true negatives are similar. On the testing set SVM manage a perfect score on the specificity but is by far the worst model in regards to the sensitivity showing a tradeoff between sensitivity and specificity.

Table 15: Accuracy on the CFS testing dataset for colon cancer

Model	Accuracy	95% CI	
Random forest	**0,89**	**0,65**	**0,99**
SVM	0,83	0,59	0,96
Boost	0,83	0,59	0,96
Elastic net	**0,89**	**0,65**	**0,99**
NSC	**0,89**	**0,65**	**0,99**

The three models random forest, elastic net and nearest shrunken centroids shares the highest prediction accuracy according to table 15.

5.2.3 Breast cancer II

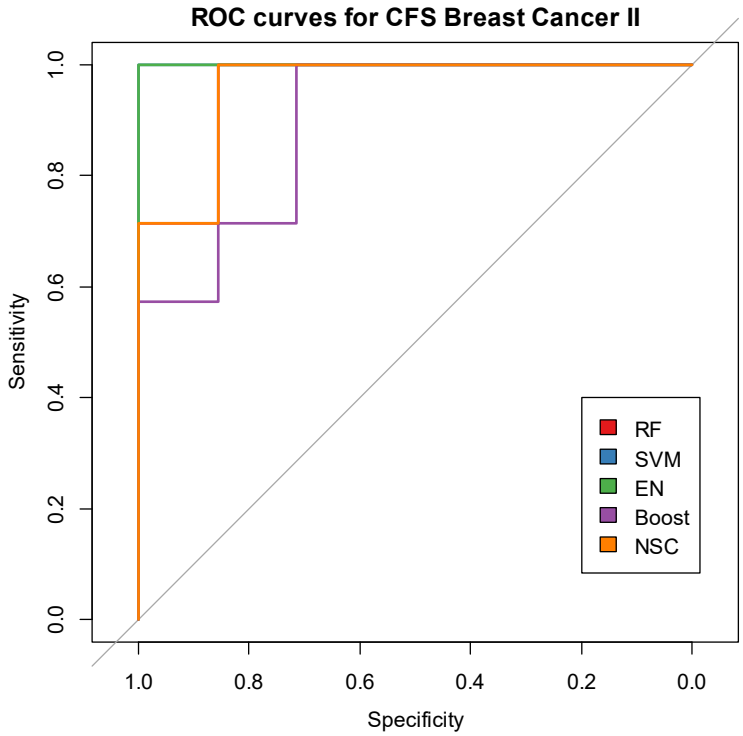

Figure 7: ROC curves for the CFS Breast Cancer II dataset

Table 16: ROC performance on the CFS Breast cancer II dataset

	AUC		SENS		SPEC	
Model	Train	Test	Train	Test	Train	Test
Random forest	0,98	0,96	0,91	0,86	**0,91**	**0,86**
SVM	0,95	**1**	0,89	**1**	0,80	**0,86**
Boost	0,91	0,90	0,80	0,71	0,80	0,71
Elastic net	0,98	**1**	0,94	**1**	0,90	0,71
NSC	**0,99**	0,96	**0,95**	**1**	0,89	0,71
Average	0,96	0,96	0,90	0,91	0,86	0,77

For the breast cancer II dataset the performance of the training set and testing set are similar and the algorithms perform very well. In table 16 it can be seen that the average AUC is 0.96 for both testing and training with SVM and Enet managing a perfect AUC of 1. Regarding the sensitivity it can be seen that the cancer samples are the easiest class to identify and three of the models managed a perfect score on the testing set. The values for the specificity is also relatively high but non-cancer samples are in general harder to identify.

Table 17: Accuracy on the CFS testing dataset for breast cancer II

Model	Accuracy	95% CI	
Random forest	0,86	0,57	0,98
SVM	**0,93**	**0,66**	**1**
Boost	0,71	0,42	0,92
Elastic net	0,86	0,52	0,98
NSC	0,86	0,57	0,98

It can be seen in table 17 that the support vector machine has the highest accuracy while boosting machines has the worst. The rest of the models all achieved the same score of 86%.

5.2.4 CNS disorder

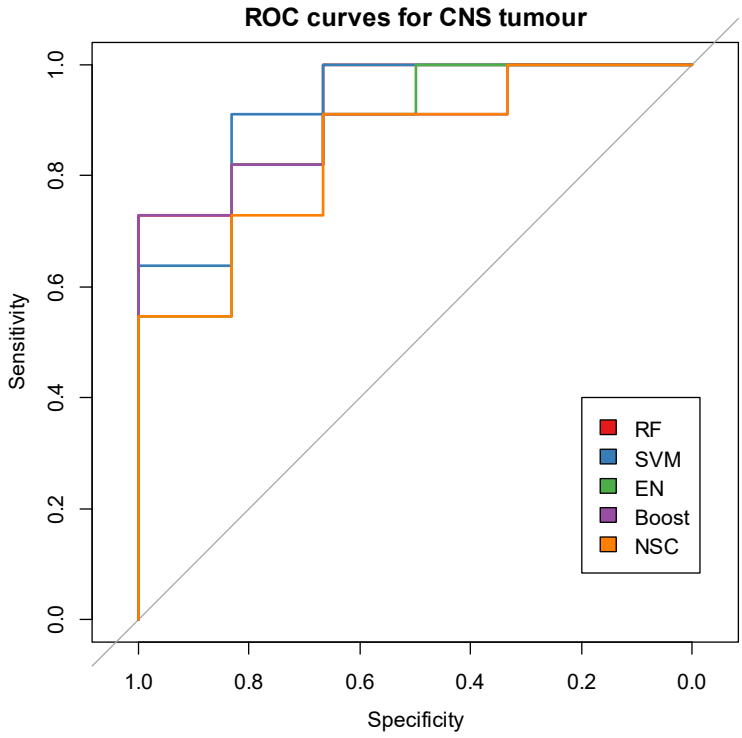

Figure 8: ROC curves for the CFS CNS tumor dataset

Table 18: ROC performance on the CFS CNS tumor dataset

	AUC		SENS		SPEC	
Model	Train	Test	Train	Test	Train	Test
Random forest	**0,98**	**0,92**	0,61	0,67	**0,97**	**1**
SVM	0,88	**0,92**	0,72	**0,83**	0,86	0,82
Boost	0,86	0,89	0,55	0,67	0,86	0,82
Elastic net	0,90	0,88	0,73	0,67	0,85	0,91
NSC	0,89	0,85	**0,76**	0,67	0,89	0,73
Average	0,90	0,89	0,67	0,70	0,87	0,85

The performance metrics for the CNS tumor can be seen in table 18. Yet again the differences in the average AUC for the training set and the testing set is small implying no overfitting in general. The best algorithms are the random forest and support vector machine which scored an AUC of 0.92 on the testing set. SVM stands out in the sensitivity benchmark with 0.83 while every other model only reaches 0.67. The SVM is also quite balanced reaching 0.82 on the specificity while the other models perform better on the specificity. The patients without cancer turned out to be easier to identify in general on this dataset with the random forest managing a perfect result.

Table 19: Accuracy on the CFS testing dataset for CNS tumor

Model	Accuracy	95% CI	
Random forest	**0,88**	**0,64**	**0,99**
SVM	0,82	0,57	0,96
Boost	0,76	0,50	0,93
Elastic net	0,82	0,57	0,96
NSC	0,71	0,44	0,90

In terms of prediction the random forest scores the highest accuracy of 0.88 which can be seen in table 19. RF is followed by SVM and Elastic net while NSC is the worst performing algorithm at 0.71.

5.3 Comparisons between full dataset and CFS datasets

Table 20: Comparison between algorithm performance on Breast Cancer I dataset.

		AUC			Accuracy	
Model	Full	CFS	Diff (%)	Full	CFS	Diff (%)
Random forest	0,76	**0,94**	**23,2**	68	**80**	17,6
SVM	**0,88**	0,86	-3,1	76	**80**	5,2
Boost	0,78	0,83	5,9	64	78	**21,9**
Elastic net	**0,88**	0,87	-1,5	**80**	76	-5,2
NSC	0,84	0,88	4,9	78	78	0
Average	0,83	0,88	5,88	73	78	7,9

From table 20 it can be seen that on average the algorithms perform better on the CFS filtered dataset with an average increase in AUC by close to 6% and an average increase in accuracy up to 8%. The algorithm with the best total increase in performance is the random forest which improves the AUC by 23% and the accuracy by 18%. SVM and elastic net does benefit as much and in the case of elastic net the performance actually gets worse on the CFS dataset. NSC improves AUC a little but achieves the same accuracy on both datasets. The boosting algorithm improves the accuracy by 22%.

Table 21: Comparison between algorithm performance on Colon cancer I dataset.

	AUC			**Accuracy**		
Model	Full	CFS	Diff (%)	Full	CFS	Diff (%)
Random forest	0,83	0,93	11,9	67	89	32,8
SVM	0,75	**0,96**	27,8	67	83	23,9
Boost	0,68	0,89	**30,6**	56	83	**48,2**
Elastic net	**0,94**	0,93	-1,4	**83**	89	7,2
NSC	0,90	0,93	3,1	67	89	32,8
Average	0,82	0,93	14	66	87	29

For the colon cancer dataset the performance metrics can be found in table 21. The performance is on average better with 14% increase in AUC and 29% increase in accuracy on average. SVM and boosting algorithm benefitted in both metrics and boosting in particular improves the accuracy by almost 50%. The elastic net however does not benefit as much and in terms of AUC actually gets worse. It can be noted that the accuracy of all the models are improved.

Table 22: Comparison between algorithm performances on Breast Cancer II dataset.

	AUC			**Accuracy**		
Model	Full	CFS	Diff (%)	Full	CFS	Diff (%)
Random forest	0,73	0,96	30,6	64	86	34,4
SVM	0,59	**1**	**69**	57	**93**	**63,2**
Boost	0,71	0,90	25,7	64	71	10,9
Elastic net	0,84	**1**	19,5	**71**	86	21,1
NSC	**0,88**	0,96	9,3	**71**	86	21,1
Average	0,75	0,96	30,8	65	84	30,1

The performance metrics on Table 22 on the Breast cancer II dataset shows that the performance improves on average by 30% in both AUC and accuracy. Notable is that both the elastic net and the SVM achieves a perfect AUC on the CFS dataset. All of the algorithms did benefit on this dataset with the average AUC being 0.96 and the average accuracy 0.84. The SVM scores the highest accuracy with 93%.

Table 23: Comparison between algorithm performances on CNS tumor dataset.

	AUC			Accuracy		
Model	Full	CFS	Diff (%)	Full	CFS	Diff (%)
Random forest	0,48	**0,92**	**90,6**	65	**88**	35,4
SVM	0,58	**0,92**	60,5	65	82	26,2
Boost	0,56	0,89	59,5	53	76	43,4
Elastic net	0,47	0,88	87,1	**71**	82	15,5
NSC	**0,61**	0,85	40	35	71	**103**
Average	0,54	0,89	66	0,58	0,80	44

In table 23 the differences in performance on the CNS tumor dataset are found. This particular dataset has the worst performance of all the algorithms on the full dataset. As seen in the table the average improvement is 66% when reducing the features. The accuracy improves all along the board with an average of 44% increase. The random forest and elastic net especially liked the feature reduction and improved the AUC by 90%. The highest improvement is seen from NSC which improved accuracy by over 100%. Overall, the dataset became more manageable when reducing the features with CFS. Random forest achieves the highest accuracy with 0.88. In conclusion the algorithms on this dataset really improves when removing irrelevant genes.

Table 24: The best algorithm in each performance metric per dataset

	AUC		Accuracy	
Dataset	Full	CFS	FULL	CFS
Breast cancer I	EN, SVM	RF	EN	RF, SVM
Colon cancer	EN	SVM	EN	EN, NSC, RF
Breast cancer II	NSC	EN, SVM	EN, NSC	SVM
CNS tumor	NSC	RF, SVM	EN	RF

The best performers of each dataset can be found in table 24. For the full dataset the elastic net perform best in terms of accuracy and also perform very well on the AUC metric with the nearest shrunken centroid also performing well. On the reduced datasets SVM and RF perform well. It is important to note that in most of the scenarios the algorithms are quite close to each other. Both NSC and Elastic net have variable reduction built into their algorithms which means that they reduce the high dimensionality in the full datasets while the other algorithms use all of the information.

6. CONCLUSION

The objective of this thesis was to compare different machine learning algorithms to see if there is an algorithm that performs better than the others. The findings imply that this is not the case. It would seem that the algorithms perform different when applied on different datasets. It is therefore hard to generalize about which algorithm is the best since there is no clear answer. It seems that in order to choose a good algorithm it is therefore advisable to try several alternatives before deciding which one to use for a specific dataset. It is also important for a machine learning researcher to consider what the goal of the analysis is since some of the models are hard to interpret. Random forests could potentially be interpreted in some meaningful way since they are built upon classification trees which are conceptually easy to grasp. Nearest shrunken centroids and boosting methods are however notoriously hard to interpret since the only meaningful outcome is the prediction accuracy.

The most surprising finding is that the logistic regression with elastic net performs better than the random forest and support vector machine, which have been proven to be one of the more flexible and better performing methods in many of the earlier studies. Random forests and support vector machines have been proven to generally perform consistently well on every dataset which is given to them which is one of the reasons for their popularity.

This might be explained by the fact that the elastic net automatically reduces variables from the model in its algorithm and ends up with a more parsimonious model in those cases where it performs the best. Judging from the results the elastic net performs best on the full dataset but takes a step back when the dataset is reduced by the gene selection method.

In general, all of the algorithms are prone to overfitting on the full dataset with some discrepancy on the AUC values for the training and the testing data. On the CFS datasets the AUC are quite similar and overfitting not as much of a problem. The curious case is the CNS tumor dataset where every algorithm perform poorly with no reason as to why. The full dataset must have been very noisy and it is hard to perform any sort of remedy. The CFS gene selection method does solve the problems and the accuracy becomes acceptable afterwards.

The final part of the objective of the thesis is to see if it is possible to improve the performance by removing irrelevant genes with a gene selection method. With the help of CFS the number of features is greatly reduced with 99% in each dataset. This results in performance gains and a higher prediction accuracy for almost every algorithm on each of the datasets.

It is however important to note that gene filter methods such as CFS do not take in consideration which genes that are actually important from a medical or biological perspective. It is therefore difficult to say if the chosen genes are important or rather that the genes have been chosen by the heuristic method in order to make a good classification. So yet again, in order to utilize gene selection methods a researcher has to think about the goals of the research; is it achieve a model with the highest accuracy or is to extract information about important genes? If it’s the latter, a filter-based method such as CFS is seemingly not recommended.

7. REFERENCES

Alon, U., Barkai, N., Notterman, D. A., Gish, K., Ybarra, S., Mack, D., & Levine, A. J. (1999). Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays. *Proceedings of the National Academy of Sciences*, *96*(12), 6745-6750.

Ben-Dor, A., Bruhn, L., Friedman, N., Nachman, I., Schummer, M., & Yakhini, Z. (2000). Tissue classification with gene expression profiles. *Journal of computational biology*, *7*(3-4), 559-583.

Breiman, L. (2001). Random forests. *Machine learning*, *45*(1), 5-32.

Cai, Z., Xu, D., Zhang, Q., Zhang, J., Ngai, S. M., & Shao, J. (2015). Classification of lung cancer using ensemble-based feature selection and machine learning methods. *Molecular BioSystems*, *11*(3), 791-800.

Cho, S. B., & Won, H. H. (2003, January). Machine learning in DNA microarray analysis for cancer classification. In *Proceedings of the First Asia-Pacific bioinformatics conference on Bioinformatics 2003-Volume 19* (pp. 189-198). Australian Computer Society, Inc..

Cruz, J. A., & Wishart, D. S. (2006). Applications of machine learning in cancer prediction and prognosis. *Cancer informatics*, *2*.

Dudoit, S., Fridlyand, J., & Speed, T. P. (2002). Comparison of discrimination methods for the classification of tumors using gene expression data. *Journal of the American statistical association*, *97*(457), 77-87.

Fan, J., & Ren, Y. (2006). Statistical analysis of DNA microarray data in cancer research. *Clinical Cancer Research*, *12*(15), 4469-4473.

Friedman, J. H. (2001). Greedy function approximation: a gradient boosting machine. *Annals of statistics*, 1189-1232.

Gravier, E., Pierron, G., Vincent-Salomon, A., Gruel, N., Raynal, V., Savignoni, A., ... & Fourquet, A. (2010). A prognostic DNA signature for T1T2 node-negative breast cancer patients. *Genes, Chromosomes and Cancer*, *49*(12), 1125-1134.

Guyon, I., & Elisseeff, A. (2003). An introduction to variable and feature selection. *Journal of machine learning research*, *3*(Mar), 1157-1182.

Hall, M. A. (1999). *Correlation-based feature selection for machine learning* (Doctoral dissertation, The University of Waikato).

Hastie, T., Tibshirani, R., Friedman, J., & Franklin, J. (2005). The elements of statistical learning: data mining, inference and prediction. *The Mathematical Intelligencer*, *27*(2), 83-85.

Hsu, C. W., Chang, C. C., & Lin, C. J. (2003). A practical guide to support vector classification.

Inza, I., Larrañaga, P., Blanco, R., & Cerrolaza, A. J. (2004). *Artificial intelligence in medicine: Filter versus wrapper gene selection approaches in DNA microarray domains* Elsevier. doi:10.1016/j.artmed.2004.01.007

James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). *An introduction to statistical learning* (Vol. 112). New York: springer.

Kuhn, M., & Johnson, K. (2013). *Applied predictive modeling* (pp. 389-400). New York: Springer.

Lee, J. W., Lee, J. B., Park, M., & Song, S. H. (2005). An extensive comparison of recent classification tools applied to microarray data. *Computational Statistics & Data Analysis*, *48*(4), 869-885.

Li, T., Zhang, C., & Ogihara, M. (2004). A comparative study of feature selection and multiclass classification methods for tissue classification based on gene expression. *Bioinformatics*, *20*(15), 2429-2437.

Natekin, A., & Knoll, A. (2013). Gradient boosting machines, a tutorial. *Frontiers in neurorobotics*, *7*.

Pirooznia, Mehdi, et al. "A comparative study of different machine learning methods on microarray gene expression data." *BMC genomics* 9.1 (2008): 1.

Pardo, M., & Sberveglieri, G. (2008). Random forests and nearest shrunken centroids for the classification of sensor array data. *Sensors and Actuators B: Chemical*, *131*(1), 93-99.

Pomeroy, S. L., Tamayo, P., Gaasenbeek, M., Sturla, L. M., Angelo, M., McLaughlin, M. E., ... & Allen, J. C. (2002). Prediction of central nervous system embryonal tumor outcome based on gene expression. *Nature*, *415*(6870), 436-442.

Raza, K., & Hasan, A. N. (2015). A comprehensive evaluation of machine learning techniques for cancer class prediction based on microarray data. *International journal of bioinformatics research and applications*, *11*(5), 397-416.

Tan, A. C., & Gilbert, D. (2003). Ensemble machine learning on gene expression data for cancer classification.

Tibshirani, R., Hastie, T., Narasimhan, B., & Chu, G. (2002). Diagnosis of multiple cancer types by shrunken centroids of gene expression. *Proceedings of the National Academy of Sciences*, *99*(10), 6567-6572.

Yu, L., & Liu, H. (2003, August). Feature selection for high-dimensional data: A fast correlation-based filter solution. In *ICML* (Vol. 3, pp. 856-863).

West, M., Blanchette, C., Dressman, H., Huang, E., Ishida, S., Spang, R., ... & Nevins, J. R. (2001). Predicting the clinical status of human breast cancer by using gene expression profiles. *Proceedings of the National Academy of Sciences*, *98*(20), 11462-11467.

Zou, H., & Hastie, T. (2005). Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, *67*(2), 301-320.

Önskog, J., Freyhult, E., Landfors, M., Rydén, P., & Hvidsten, T. R. (2011). Classification of microarrays; synergistic effects between normalization, gene selection and machine learning. *BMC bioinformatics*, *12*(1), 1.