

# Analytics

Silke Meiner, Rafaela Neff

20 3 2021

## Contents

<b>Abstract</b>	<b>1</b>
<b>Business Understanding</b>	<b>2</b>
<b>Data Sources and Data Understanding</b>	<b>2</b>
Data Understanding . . . . .	3
Data Visualisation, looped back after first modelling . . . . .	5
<b>Data Preparation</b>	<b>6</b>
<b>Modeling</b>	<b>6</b>
Logistic Regression . . . . .	6
Explainability . . . . .	8
Artificial Neural Networks . . . . .	10
Normalization . . . . .	10
<b>Final Assessment</b>	<b>11</b>

## Abstract

We present and compare machine learnt classification algorithms for diagnosing breast tumor cells as benign or malignant. The algorithms were trained on tabular data consisting of features extracted from microscopic images of fine-needle aspirates / biopsies.

We get reasonably good results from Logistic regression and were able to improve these results through single and multi-layer neural networks. The final neural network with an accuracy of 98%, sensitivity of 1 and specificity 1 is being audited at the German Gesundheitsministerium to become a state approved diagnostic tool.

The layout of this paper follows the CRISP-DM model.

## Business Understanding

Breast cancer is the most common cancer for women and ranks highest for cancer-related deaths in women in Germany: In 2016 there were 68,950 women and 710 men suffering from Breast Cancer (ICD-10 C50). In 2020 18,570 women and 1 men have died of breast cancer. The situation is similar in many other countries.

Tumors can build in the human body as some cells grow more than they normally should. If this growth is not limiting itself and destroys body tissue and hinders body functions the tumor is labeled malignant and called cancer.

Tumors are classified in a binary fashion as either malignant or benign. Their difference in microscopic imagery are shown in figure 1. A typical first step when diagnosing a tumor is to do a fine needle aspirate (FNA) of a breast mass and looking at the cells through a microscope, describing characteristics of the cell nuclei. Further treatment differs according to the tumor diagnosis as benign or malignant.

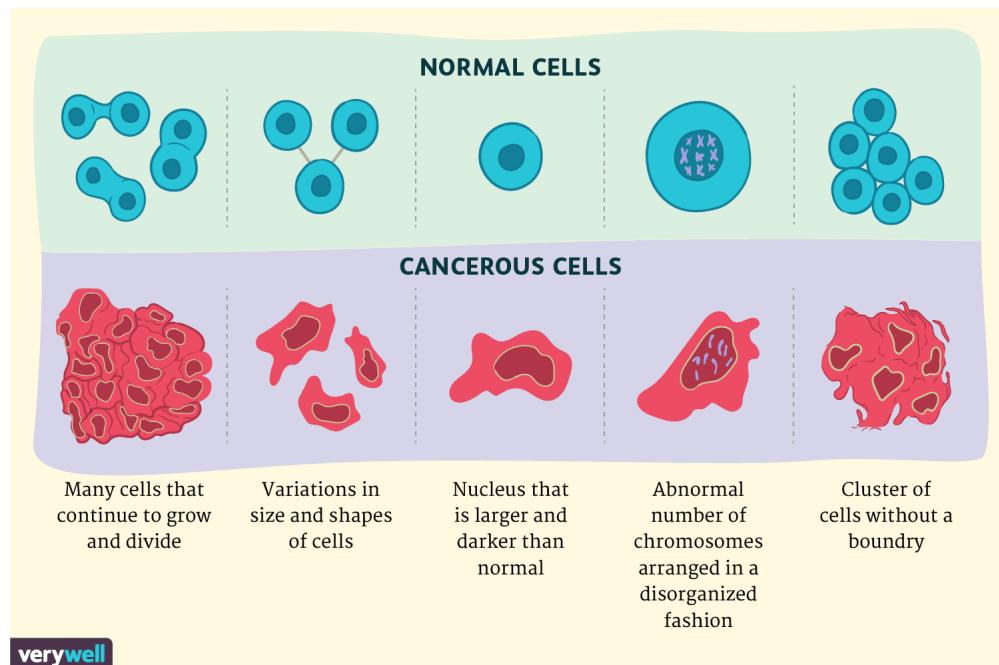


Figure 1: benign / normal and malignant / cancerous cells

In medical diagnostics, based on human or artificial intelligence, correctness is very important. It is described in terms of sensitivity and specificity. Both are measured in percentage. Sensitivity is the true positive rate, the rate at which malignant tumors are correctly identified. Specificity is the true negative rate, the rate at which a patient's concerns can correctly be relieved.

Sources :

- RKI
- VerywellHealth [b]
- VerywellHealth [a]

## Data Sources and Data Understanding

The data for this project was collected in 1995 by the University of Wisconsin and made available to us through Prof. Dr. Nick Street of the University of Iowa.

The data can be downloaded from the University of California, Irvine, <https://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+%28Diagnostic%29>

## Data Understanding

Our data set is mid sized with 569 observations for 30 numeric feature variables. Each observation has an additional binary diagnosis as benign or malignant. The data set is slightly unbalanced with 357 (63%) benign and 212 (37%) malignant cases. We made malignant the positive class.

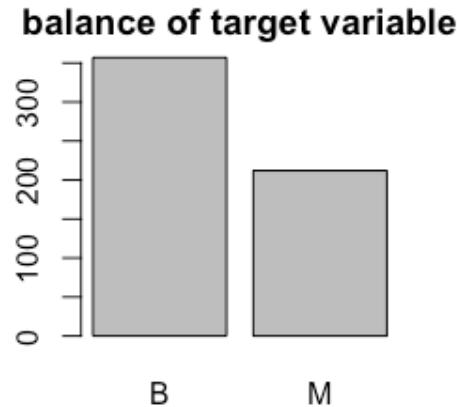


Figure 2: classes in the binary classification task, B: benign, M: malignant / cancerous

In figure 3 we present the correlation of features and find some features / predictor variables highly correlated.

We point out three groups of correlated features:

The group of most highly (and positively) correlated features correspond to the description of malignant / cancerous cells given in 1: They describe cells with a large and irregular perimeter, radius and area:

- perimeter\_mean, perimeter\_worst
- radius\_mean, radius\_worst
- area\_mean, area\_worst

The second most highly (and positively) correlated group of features are the standard errors for the cells perimeter, radius and area, underlying the group of most highly correlated features.

- perimeter\_se
- radius\_se
- area\_se

A third group of high to mild positive correlation is for features related to concavity and compactness.

- concave.points\_mean, concave.points\_worst
- concavity\_mean, concavity\_worst
- compactness\_mean, compactness\_worst

In figure 1 convexity is a feature in the benign / normal cells. Its opposite, concavity, can easily be associated with the malignant / cancerous cells.

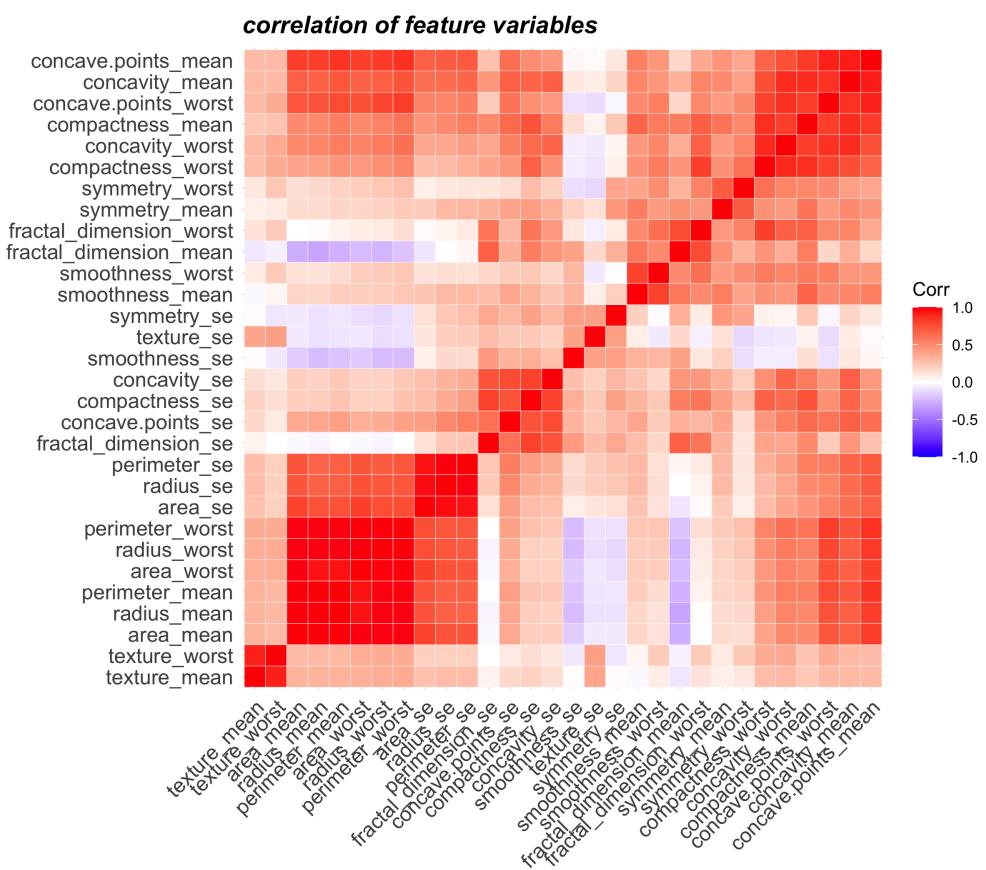


Figure 3: correlation of feature variables

When there is (positive) correlation of feature variables with the target variable, it is most prominent for features from the first and third group of correlated features. Positive correlation of target and features is maximal for concave.points\_worst with a value of 0.794.

In figure 4 we present two density plots of our features. Both features have a high (positive) correlation to the target. Distributions are well distinguishable for the two classes which is in advantage for the classification task.

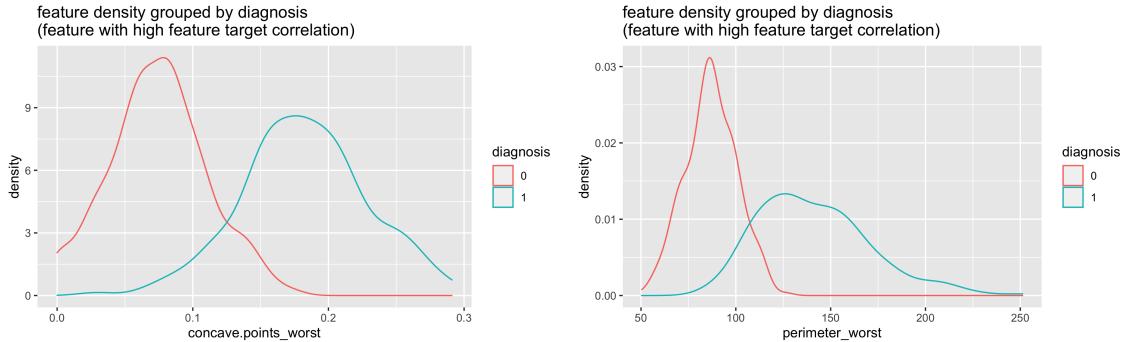


Figure 4: Density plots for two exemplary features

For more density plots, please see our data-understanding.Rmd notebook.

The features in this data set seem hand crafted and contain expert domain knowledge.

## Data Visualisation, looped back after first modelling

This visualisation of our data by application of PCA was developed after a first loop in the CRISP model. In sequential reading this serves as a general visualisation now and will also be relevant later on.

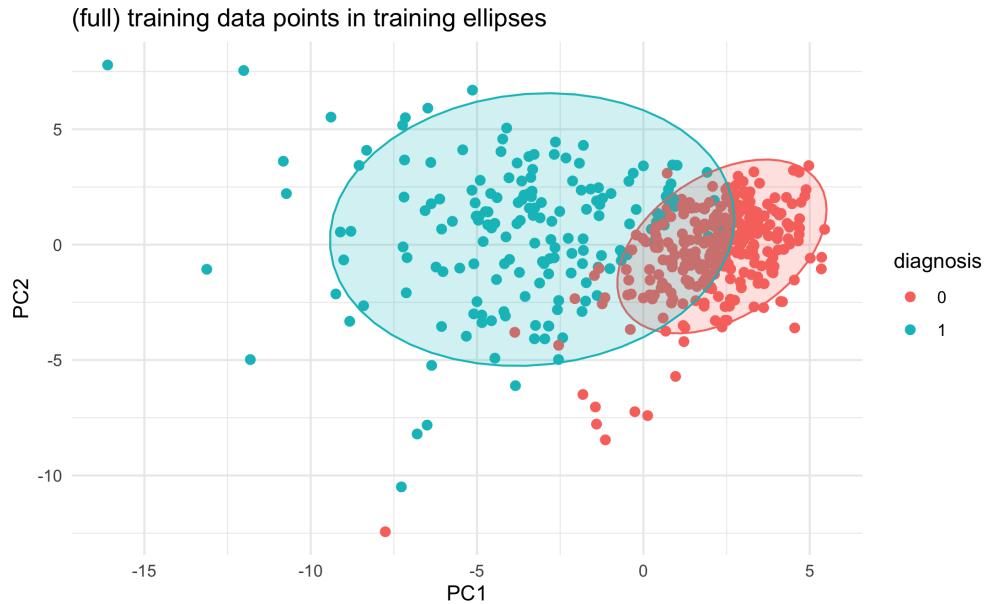


Figure 5: PCA for the full training data

## Data Preparation

Since there were no missing data in our set we did not impute anything. Data was used as delivered.

Data was separated into training and test sets, each with a separate file. The data was split 80/20 and with stratification wrt to the diagnosis.

For further details, please look into our notebook `data-preparation.Rmd`.

## Modeling

To solve the classification task we applied two machine learning methods: Logistic regression and neural networks.

In classification tasks we generally have predictor variables and a target variable. In binary classification the target variable can take one of two distinct values, interpreted as the two classes on option.

The methodological similarities of both machine learning models are in the training and evaluation of the model. The training requires a training set of observed data points including the true values of the target variable. For evaluation a test set of observed data points including the true values of the target variable is required. Training and test set need to be disjoint. On the test set the algorithm predicts classes for the data points and predictions are compared with the targets. Counting correct and not correct classifications and setting them in relation results in accuracy, sensitivity and specificity as measures of success.

## Logistic Regression

### Mathematical Background

Given: Some numeric data, in tabular form of  $n$  rows and  $p + 1$  variables.  $p$  variables are predictors and the remaining variable is the target. The target takes one of two values, interpreted as two classes, with one class defined as the positive class.

Desired: The class the data point belongs to with a probability distribution over the two classes (stating the probabilities that the data point belongs to each class). Alternatively, the following result would be of equal value: if or if not the data point belongs to the positive class and the associated probability (single value in  $[0,1]$ ).

Linear regression uses a linear combination of the variables to predict another numeric target variable. logistic regression does linear regression for the log-odds of the desired probabilities.

$$\text{log-odds} = \beta_0 + \beta_1 x_1 + \cdots + \beta_p x_p$$

The log-odds are transformed into (conditional) probabilities for the positive class through the logistic function  $\sigma(\cdot)$ , also called sigmoid.

$$p(\mathbf{x}) = \sigma(\text{log-odds}) = \sigma(\beta_0 + \beta_1 \mathbf{x}_1 + \cdots + \beta_p \mathbf{x}_p)$$

with  $\mathbf{x} = (\mathbf{x}_1, \dots, \mathbf{x}_p)$ ,  $p(\mathbf{x}) = \mathbf{P}(\text{target} = \text{positive class} | \text{predictors} = \mathbf{x})$  and  $\sigma(a) = \frac{1}{1+e^{-a}}$ .

If the probability for the positive class  $p(\mathbf{x})$  exceeds a threshold, the data point is classified as the positive class.

The performance of the model is determined by its coefficients / weights  $\beta$ . Finding the best / suitable coefficients is done in training. For logistic regression there are several training methods performed by statistical software like R.

In the beginning of running a logistic regression we ran into a ‘problem’ hinting at perfect separation of classes in our data set. We decided to ignore this and try to get the best possible results from logistic regression. Another option would have been: change to support vector machines which could directly exploit the separability in our data.

Sources:

Log Regression (christophm)

for ignoring the problem of complete separation (UCLA)

Finding coefficients for logistic regression: (StackExchange)

## Modelling in R

First attempts to run a logistic regression are documented in log-reg-01.Rmd and log-reg-02.Rmd in the github repository’s code folder. The following presentation is based on log-reg-03.Rmd and uses the caret library in R.

We compared different preprocessing options for the logistic regression:

- no preprocessing
- center and scaleing
- min-max-normalization to the interval [0,1]
- PCA (including center and scaling)

Using 10 fold cross validation which was repeated 50 times (with new and different separation of the training data into 10 subsets / folds one of which would be used for validation) we obtained the following plot 6 for accuracies for the different preocessing options.

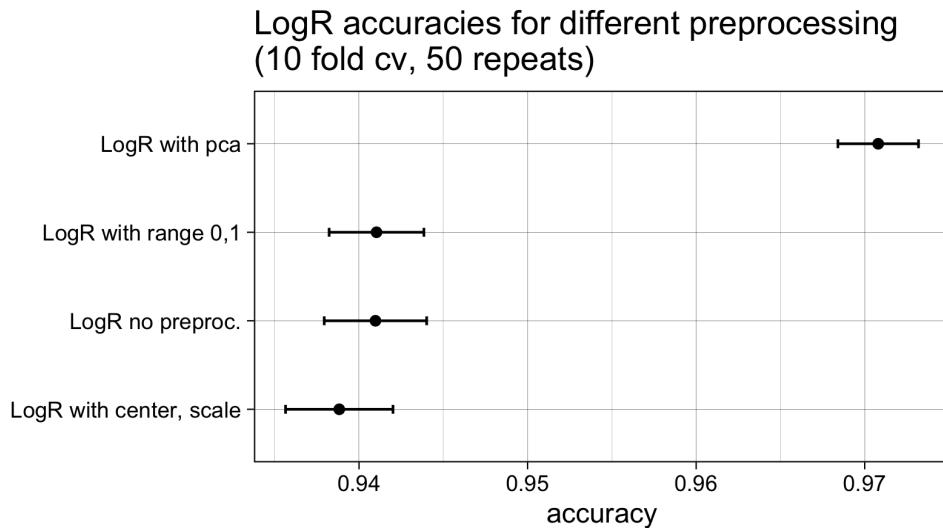


Figure 6: accuracy for different pre-processing options

Principal component analysis (pca) shows by far the best result.

We also tested if removing correlated variables improves accuracy. This can happen because removing highly correlated variables can be seen as removing noise making it easier for the model to pick up relevant structures in the data.

When looking at a group of highly correlated features all but one features of the group were removed.

We tested removing features with 99% correlation and with 97% correlation against the full feature set. We tested this for the logistic regression with PCA pre-processing and for the logistic regression with no pre-processing (cf figure 7) arriving at the following results:

- Removing highly correlated variables seems to improve accuracy for the logistic regression algorithm on data that has not been pre-processed.
- Removing highly correlated variables may or may not have an effect or a positive effect on accuracy for the logistic regression algorithm on data that has been pre-processed with pca.

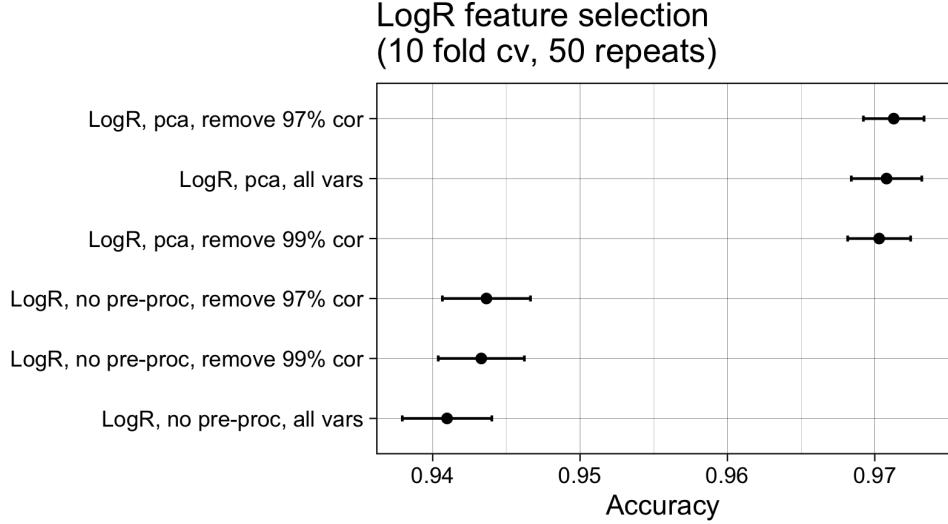


Figure 7: accuracy for different reduced feature number

This makes sense, as the PCA is a transformation of the data that creates new variables that explain the data's variance, in descending order. High correlation with previous principal components will not help with explaining remaining variance.

As the over all best model we keep the best model from Logistic Regression with a PCA as pre processing and the full set of variables (mainly, we don't have a good enough reason to reduce features and throw away information, especially when the PCA reduces information in a reasonable, statistically approved way).

We arrived at our best model: logistic regression with PCA pre-processing and the whole feature set. We then trained the model again on the full training data. summary of the model is shown in figure 8. The model did converge after 10 iterations, some of the principal components are highly significant below 0.1%.

From cross validation during training we obtained a mean accuracy of 96.5% and mean sensitivity and specificity of 98.2% (same for both). In general, these are very acceptable results. In the medical domain we should aim for better results, especially better sensitivity.

## Explainability

Logistic regression is a well explainable model. The  $\beta$  coefficients show the change in the log-odds for each feature variable under the condition of all else equal. In our case with PCA preprocessing we get coefficients that show the change in the log-odds for each principal component. Since principal components are linear combinations of the original features the all else equal condition will not generally hold / be less realistic.

We state the results of the variable importance function (varImp) in R caret in figure ??

We try to apply black box interpretation models like variable shuffling.

```

Deviance Residuals:
    Min      1Q  Median      3Q     Max
-1.7308 -0.0251 -0.0010  0.0001  3.4270

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept) -0.196944  0.740646 -0.266  0.79031
PC1         -3.814377  0.821663 -4.642 3.45e-06 ***
PC2          2.310947  0.549902  4.202 2.64e-05 ***
PC3          0.400714  0.412670  0.971  0.33153
PC4          1.185675  0.399186  2.970  0.00298 **
PC5          -1.956653  0.627340 -3.119  0.00181 **
PC6          0.002311  0.374948  0.006  0.99508
PC7          -1.364886  1.056143 -1.292  0.19624
PC8          1.152518  0.880345  1.309  0.19048
PC9          -3.142731  1.096291 -2.867  0.00415 **
PC10         0.959892  1.262080  0.761  0.44692
PC11         1.916270  0.930464  2.059  0.03945 *
PC12         1.113943  0.795467  1.400  0.16140
PC13         1.238862  1.329158  0.932  0.35130
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

Null deviance: 602.315  on 455  degrees of freedom
Residual deviance: 45.352  on 442  degrees of freedom
AIC: 73.352

```

Number of Fisher Scoring iterations: 10

Figure 8: Final model for logistic regression

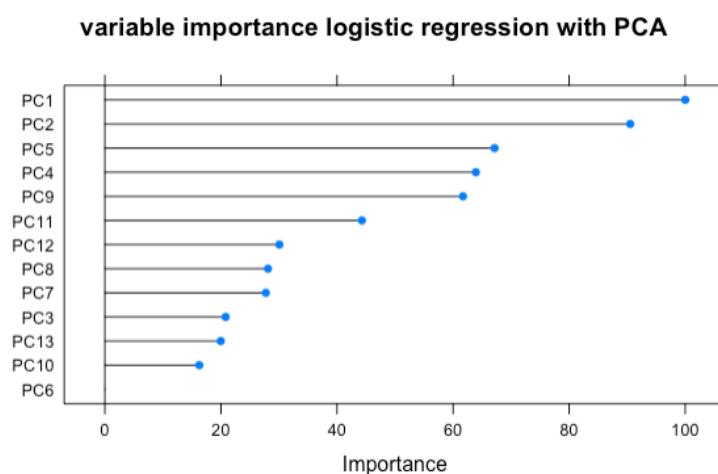


Figure 9: variable importance for logistic regression with PCA

# Artificial Neural Networks

## Model overview

Artificial Neural Networks are forecasting methods based on simple mathematical models of the brain. They allow complex nonlinear relationships between the response variable and its predictors. (<https://otexts.com/>)

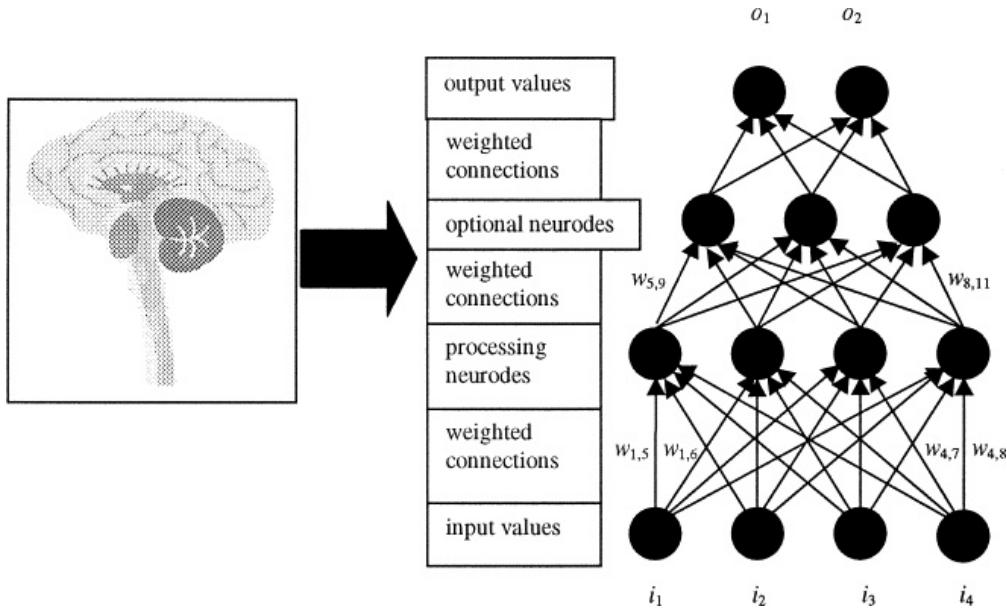


Figure 10: Sample artificial neural network architecture (not all weights are shown) - (<https://www.sciencedirect.com>)

Connections between neurons are weighted to represent the connection strength. It's the artificial pendant of the synapses in the brain. Positive weights are used to excite neurons in the network and negative weights are used to inhibit other neurons.

**Architecture:** Topology of the network

**Activities:** How do ANNs Neurons respond to one another to produce a certain behavior.

**Learning Rule:** How should weights and connections change regarding the input, output, and error-rate.

**Deep Neural Network:** Network with many hidden layers.

ANN structure can be described as its Architecture, Activities, and Learning Rule.

## Implementation

We've implemented two different kinds of Neural Network. One wth R's nnet-Package (Venables and Ripley [2002]) and one using the R-Package for h2o (Venables and Ripley [2020]).

Nnet is supporting one hidden layer only and comes with poor explainability. Since the results on the test-data were poor showing an accuracy below 90% we've choosen h2o over nnet and will evaluate further on h2o throughout this report.

## Normalization

Depending on the implementation and the specific model different standardization is used throughout this project in order to ensure that all features have the same chance of importance fitting the model.

The nnet Implementation uses it's own user-defined-function (UDF) standardizing via min-max-normalization using this formula:

$$\frac{x_i - \min(x_i)}{\max(x_i) - \min(x_i)}$$

The h2o-Implementation uses the Standard Deviation implicitly, standardazing via

$$\frac{x - \text{mean}}{\text{stddev}}$$

## Proposed Model architecture

We arrived at the selected Hyperparameters by performing a grid-search.

**hidden\_dropout\_ratios:** (Activation type: **RectifierWithDropout**) Specify the hidden layer dropout ratio to improve generalization. Specify one value per hidden layer. The range is  $\geq 0$  to  $<1$ , and the default is 0.5.

The amount of dropout on the input layer can be specified for all activation functions, but hidden layer dropout is only supported is set to WithDropout. The default hidden dropout is 50%, so you don't need to specify anything but the activation type to get good results, but you can set the hidden dropout values for each layer separately.

Performing a grid-search we've chosen the model with the highest auc-value which isn't 1. This decision is based on preventing overfitting. The False-Negatives were examined manually to get a grip on the real-world-value of our model. What we are trying to prevent is our model telling that a person doesn't have breast-cancer, when in fact he or she does.

Considering all this we did arrive at the following architecture:

```
Model Details:
=====
H2OBiNomialModel: deeplearning
Model Key: Grid_DeepLearning_RTMP_sid_a746_15_model_R_1615916806869_10538_model_15
Status of Neuron Layers: predicting diagnosis, 2-class classification, bernoulli distribution, CrossEntropy loss, 23.502 weights/biases, 286,9 KB, 289.170 training samples, mini-batch size 1
layer units      type dropout    l1     l2 mean_rate rms momentum mean_weight weight_rms mean_bias
1   1   30      Input 20.00 %  NA    NA    NA    NA    NA    NA    NA    NA
2   2   100 RectifierDropout 50.00 % 0.000100 0.000000  0.005773 0.004502 0.000000  0.018782 0.133632 0.233092
3   3   100 RectifierDropout 30.00 % 0.000100 0.000000  0.007395 0.005760 0.000000 -0.021943 0.086610 0.598928
4   4   100 RectifierDropout 20.00 % 0.000100 0.000000  0.061793 0.171625 0.000000 -0.023894 0.081698 0.632382
5   5   2       Softmax    NA 0.000100 0.000000  0.040476 0.107380 0.000000  0.024835 0.479295 0.001147
bias_rms
1   NA
2  0.070721
3  0.115045
4  0.274273
5  0.096048
```

Figure 11: Summary of the choosen Artificial Neural Network

## Criteria

**Explainability** We did choose the R-Package h2o over nnet for it's great explainability.

**Expert Plots** A deeper understanding of a fittet model can be gained with a variety of plots. For an ANN a Partial Dependence Plot or the Individual Conditional Expectation Plot of a given column can be visualized.

## Final Assessment

We compare the logistic regression and the neural networks wrt accuracy and sensitivity and specificity on the test set.

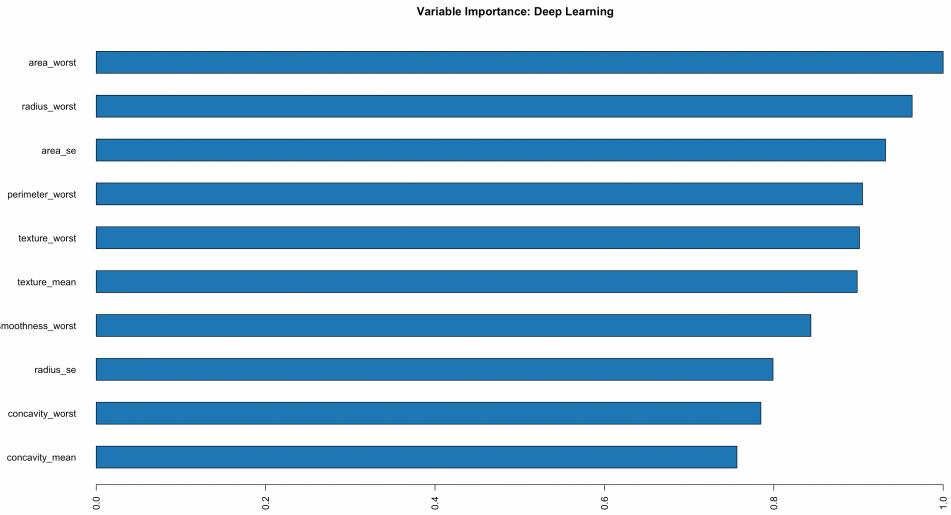


Figure 12: Variable Importance of the choosen Artificial Neural Network

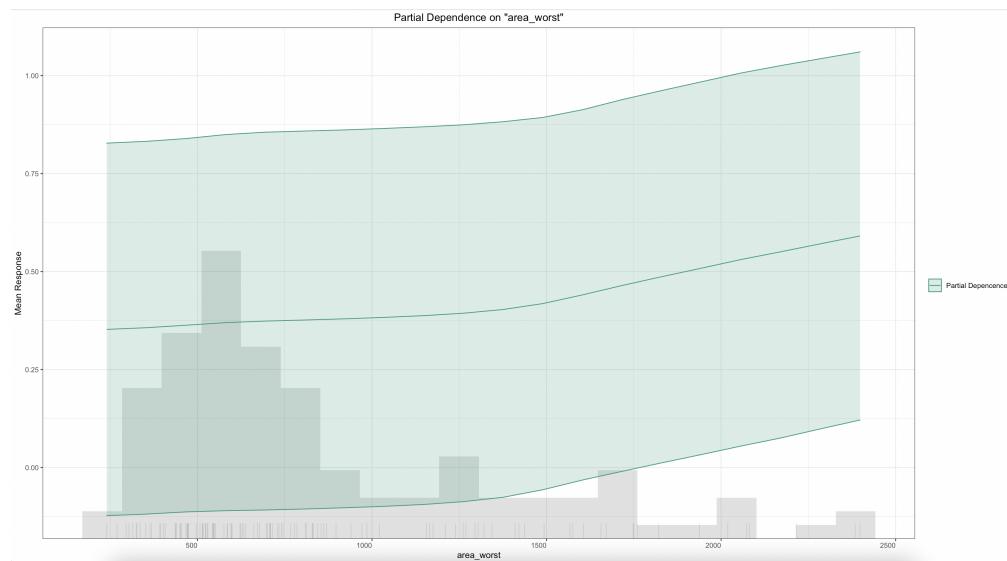


Figure 13: Partial Dependence Plot wrt area\_worts

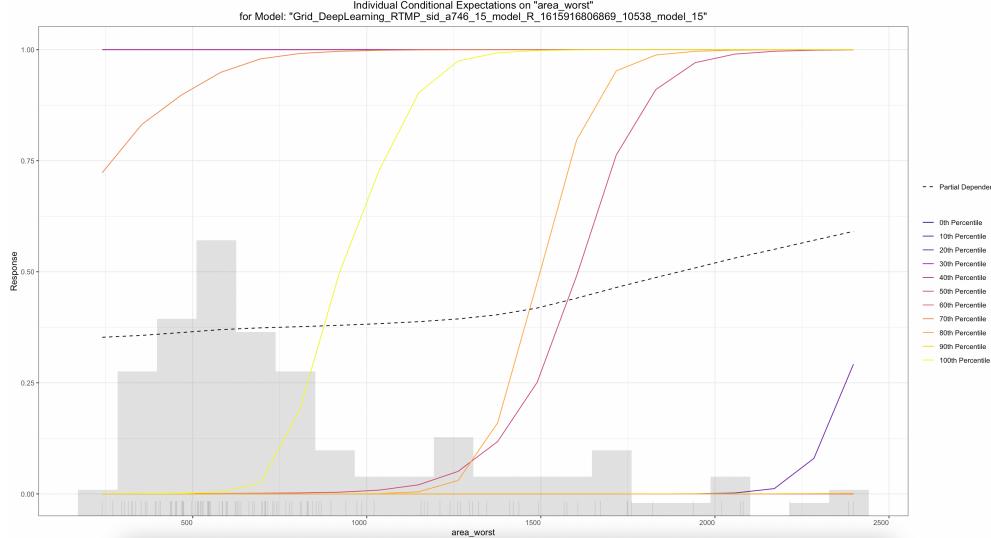


Figure 14: Conditional Expectation Plot wrt area\_worst

Model	Accuracy	Sensitivity/Recall	Specificity
Logistic regression	.965	.929	.986
Neural Network	.982	.976	.986

The neural network performs better in accuracy with only 2 misclassifications on the test set whereas the logistic regression misclassifies in 4 cases. The difference in errors is due to the better sensitivity of the neural network: only 1 malignant tumor was wrongly classified, whereas the logistic regression did err in 3 cases.

please, see the notebooks (of specific model) for more details.

## References

- christophm. Logistic regression. URL <https://christophm.github.io/interpretable-ml-book/logistic.html>.
- <https://otexts.com/>. Neural network models. URL <https://otexts.com/fpp2/nnetar.html>.
- <https://www.sciencedirect.com>. Artificial neural network. URL <https://www.sciencedirect.com/topics/computer-science/artificial-neural-network>.
- RKI. Breast cancer, icd-10 c50. URL [https://www.krebsdaten.de/Krebs/EN/Content/Cancer\\_sites/Breast\\_cancer/breast\\_cancer\\_node.html](https://www.krebsdaten.de/Krebs/EN/Content/Cancer_sites/Breast_cancer/breast_cancer_node.html).
- StackExchange. Logistic regression. URL <https://stats.stackexchange.com/questions/344309/why-using-newtons-method-for-logistic-regression-optimization-is-called-iterati>. Why using Newton's method for logistic regression optimization is called iterative re-weighted least squares?
- UCLA. Logistic regression. URL <https://stats.idre.ucla.edu/other/mult-pkg/faq/general/faqwhat-is-complete-or-quasi-complete-separation-in-logistic-regression-and-what-are-some-strategies-to-deal-with-the-issue/>. for ignoring the problem of complete separation.
- W. N. Venables and B. D. Ripley. *Modern Applied Statistics with S*. Springer, New York, fourth edition, 2002. URL <https://www.stats.ox.ac.uk/pub/MASS4/>. ISBN 0-387-95457-0.

```

H2OBinomialMetrics: deeplearning

MSE: 0.01749678
RMSE: 0.1322754
LogLoss: 0.1206374
Mean Per-Class Error: 0.01894702
AUC: 0.9919517
AUCPR: 0.9909359
Gini: 0.9839034

Confusion Matrix (vertical: actual; across: predicted) for F1-optimal threshold:
      0   1   Error   Rate
0    70  1 0.014085 =1/71
1     1 41 0.023810 =1/42
Totals 71 42 0.017699 =2/113

Maximum Metrics: Maximum metrics at their respective thresholds
               metric threshold      value idx
1             max f1  0.920597  0.976190  41
2             max f2  0.920597  0.976190  41
3             max f0points5 0.992213  0.990099  39
4             max accuracy 0.992213  0.982301  39
5             max precision 1.000000  1.000000  0
6             max recall  0.000074  1.000000  64
7             max specificity 1.000000  1.000000  0
8             max absolute_mcc 0.992213  0.962439  39
9   max min_per_class_accuracy 0.920597  0.976190  41
10  max mean_per_class_accuracy 0.920597  0.981053  41
11             max tns  1.000000 71.000000  0
12             max fns  1.000000 41.000000  0
13             max fps  0.000000 71.000000 112
14             max tps  0.000074 42.000000  64
15             max tnr  1.000000  1.000000  0
16             max fnr  1.000000  0.976190  0
17             max fpr  0.000000  1.000000 112
18             max tpr  0.000074  1.000000  64

```

Figure 15: Metrices of the choosen Artificial Neural Network

W. N. Venables and B. D. Ripley. *h2o: R Interface for H2O*. *R*. package version 3.30.0.6. edition, 2020.  
URL <https://github.com/h2oai/h2o-3>.

VerywellHealth. Cancer cells vs. normal cells: How are they different? a. URL <https://www.verywellhealth.com/cancer-cells-vs-normal-cells-2248794>.

VerywellHealth. Differences between a malignant and benign tumor. b. URL <https://www.verywellhealth.com/what-does-malignant-and-benign-mean-514240>.