Analytics

Silke Meiner, Rafaela Neff

10 3 2021

Contents

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 . . . , sensitivity of . . . and specificity . . . is being audited at the German Gesundheitsministerium to become a state approved diagnostic tool.

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.
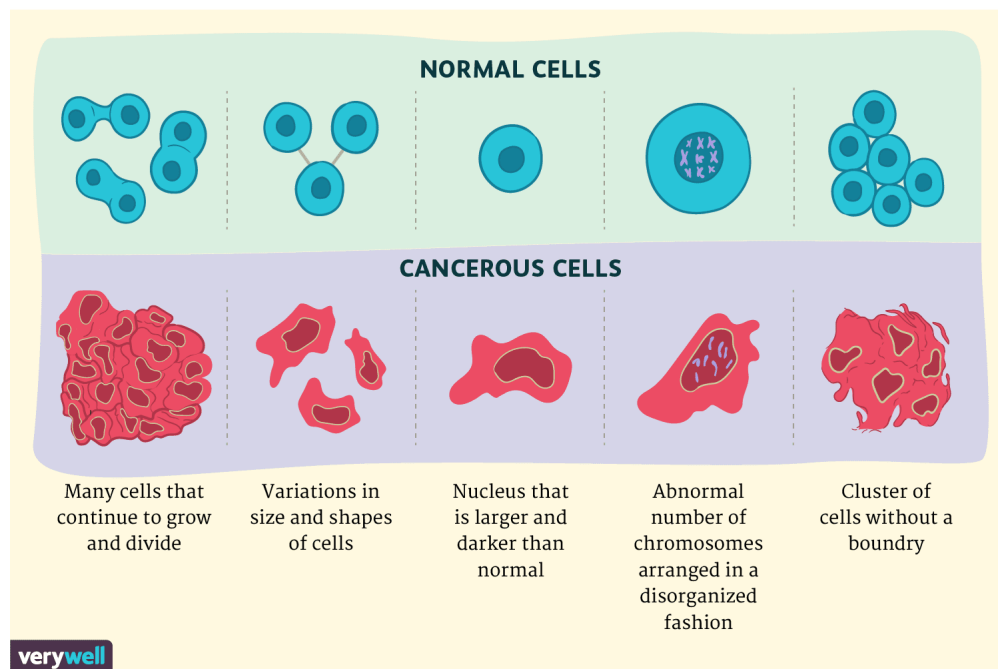

Figure 1: benign / normal and malignant / cancerous cells

Sources : * https://www.krebsdaten.de/Krebs/EN/Content/Cancer/_sites/Breast/_cancer/breast/_cancer/_node.html

- https://www.verywellhealth.com/what-does-malignant-and-benign-mean-514240
- https://www.verywellhealth.com/cancer-cells-vs-normal-cells-2248794

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%) being benign and 212 (37%) malignant cases. We made malignant the positive class.

Some predictor variables are highly correlated.

When there is correlation of feature variables with the target variable, it is mostly positive and not exceeding . . .

Data Visualisation, looped back after first modelling

This visualisation of our data ba 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 become important later on.
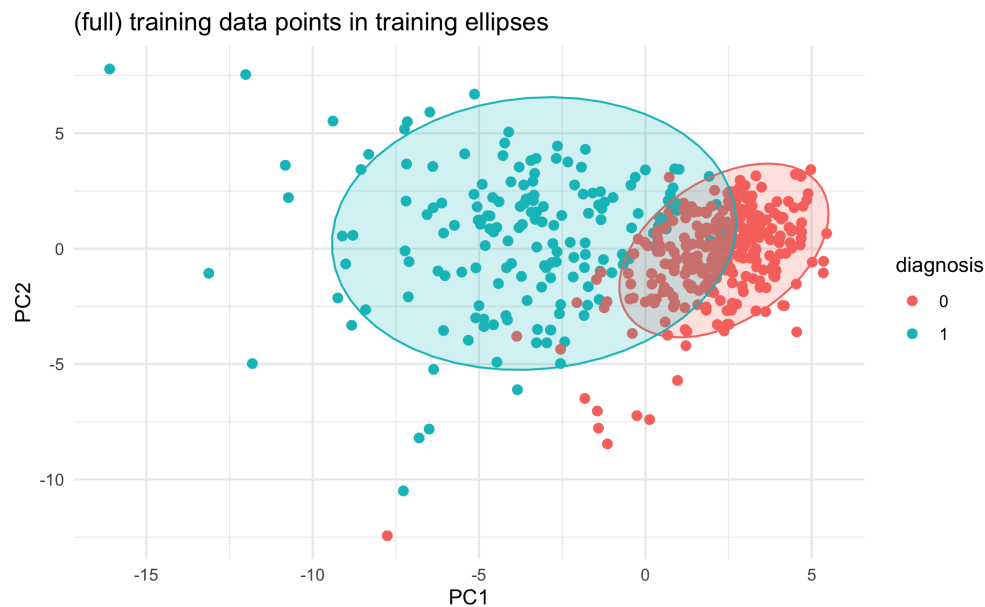
** put images A.png and B.png here **

Figure 2: Caption A-training-full-PCA.png

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 on data preparation.

Modeling

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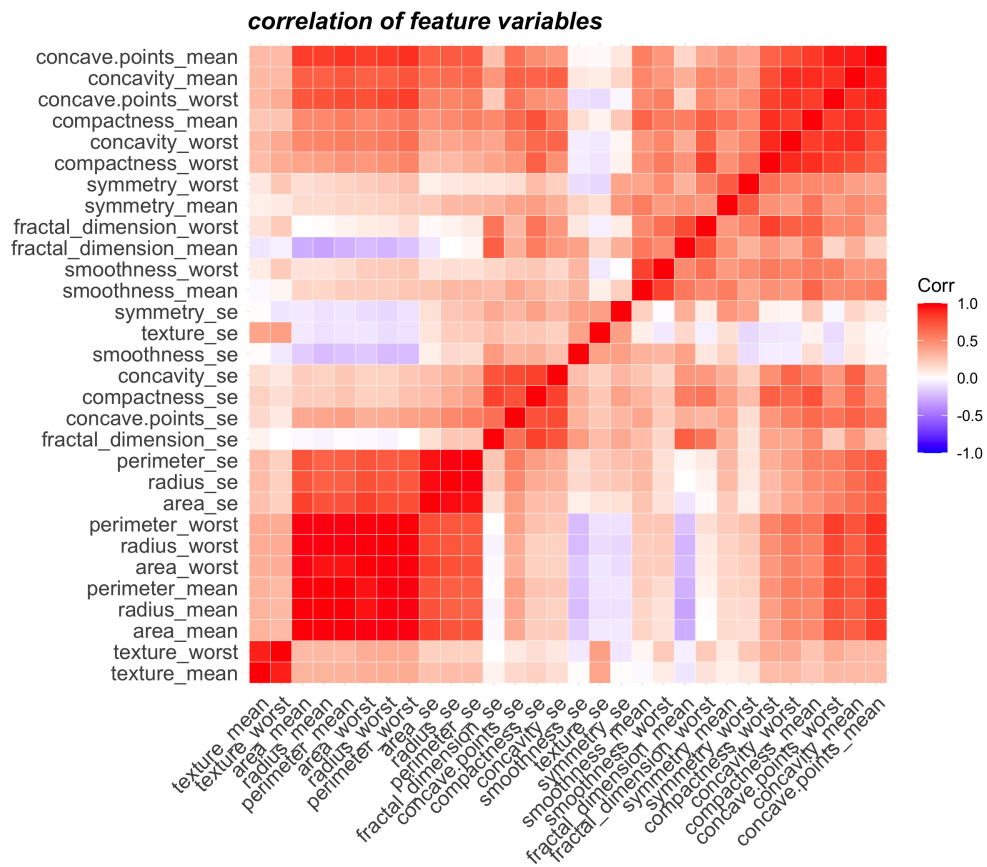
correlation of feature variables

Figure 3: Caption correlation-features.png

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

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).

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.

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

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

$$p(\mathbf{x}) = \sigma(\text{log-odds}) = \sigma(\beta_0 + \beta_1 \mathbf{x_1} + \cdots + \beta_\mathbf{p} \mathbf{x_p})$$

with
$$\mathbf{x} = (\mathbf{x_1}, \ldots, \mathbf{x_p}), \mathbf{p}(\mathbf{x}) = \mathbf{P}\left(\text{target} = \text{positive class}|\text{predictors } = \mathbf{x}\right)$$
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 to perfect separation of classes in our data set. We decided to ignore the problem 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 seperability in our data.

Sources: Log Regression https://christophm.github.io/interpretable-ml-book/logistic.html

for ignoring the problem of complete separation 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/

Finding coefficients for logistic regression: https://stats.stackexchange.com/questions/344309/why-using-newtons-method-for-logistic-regression-optimization-is-called-iterati

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/)
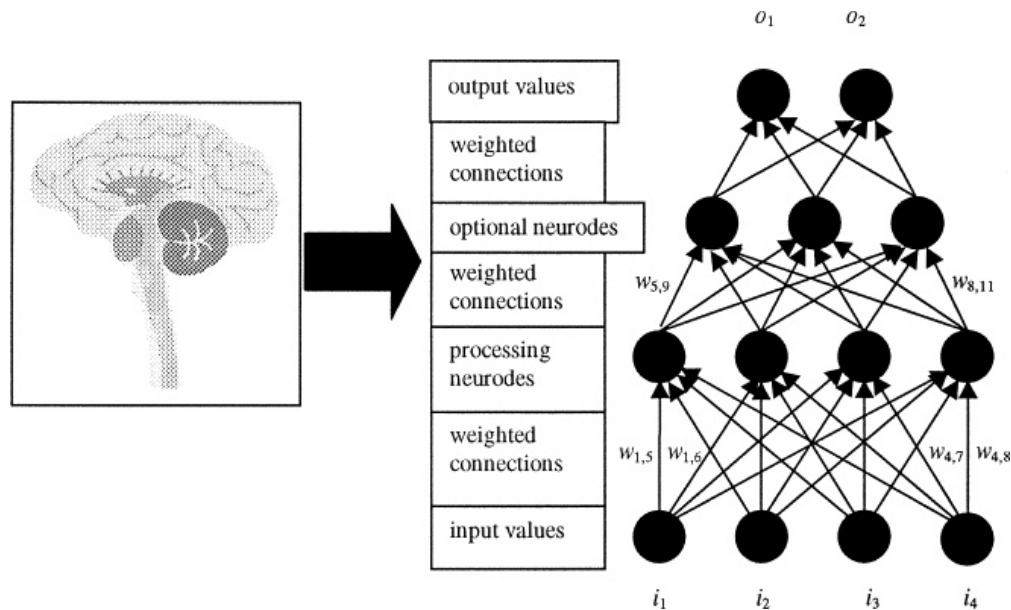

Figure 4: Sample artificial neural network architecture (not all weights are shown) - (https://www.sciencedirect.com)

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-Packe

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 formular:

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

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

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

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 >= 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 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 5: Summary of the choosen Artificial Neural Network

Criteria

Explainibility

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.
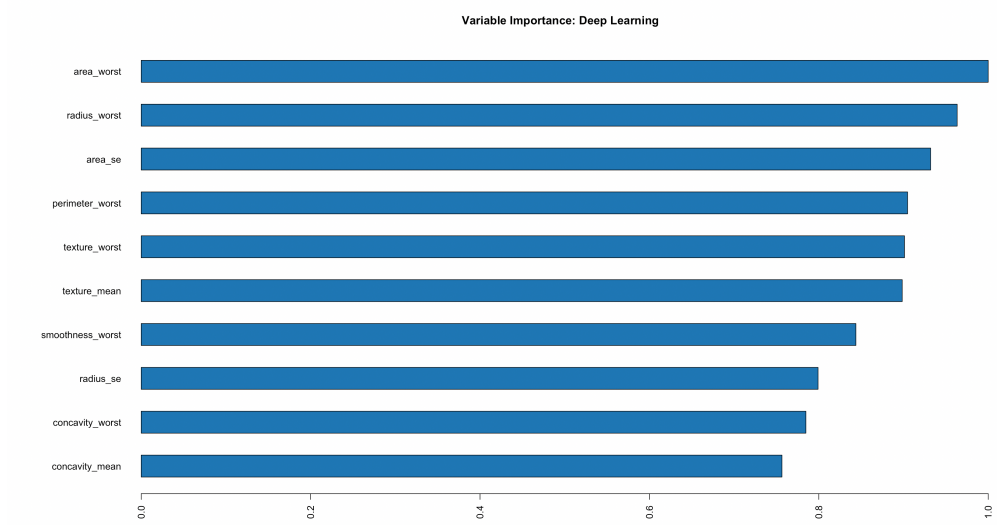
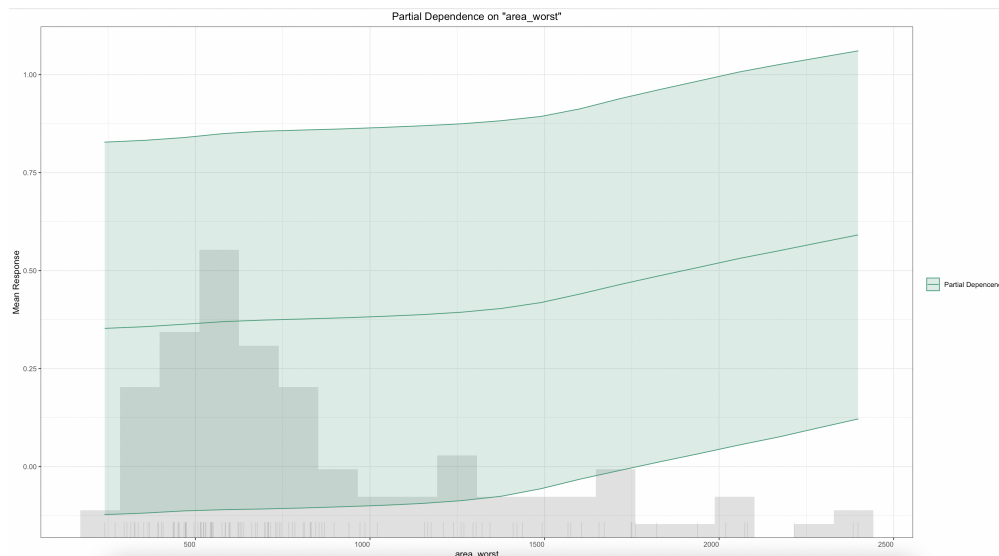Figure 6: Variable Importance of the choosen Artificial Neural Network
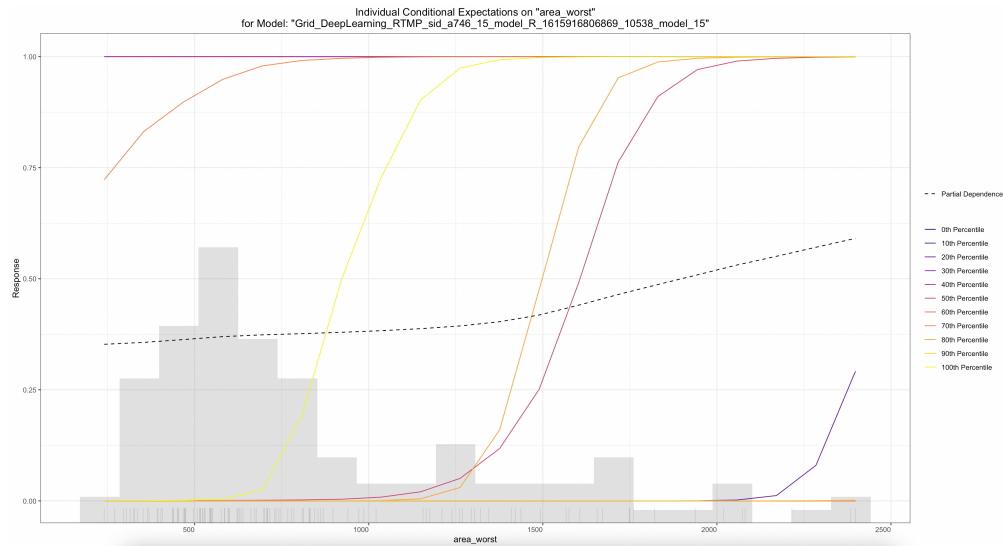

Figure 7: Partial Dependence Plot wrt area_worts

Figure 8: Conditional Expectation Plot wrt area_worst

Final Assessment

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

Model	Accuracy	Sensitivity/Recall	Specificity
Logistic regression	.95	.95	.95
Neural Network	.98	1	1

The neural network performs better in every criterion.

We should show AUC for both methods?!

please, see the notebook for more details.

Deployment

Something on meeting the standards to deploy an algorithm in medical diagnosis. Who decides? What are the criteria.

How to insert References

Venables and Ripley [2002]

https://otexts.com/

References

https://otexts.com/. Neural network models. URL https://otexts.com/fpp2/nnetar.html.

```
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 f0point5  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 9: Metrices of the choosen Artificial Neural Network

https://www.sciencedirect.com. Artificial neural network. URL https://www.sciencedirect.com/topics/computer-science/artificial-neural-network.

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.