

Chapter 10

Discovering Healthcare Data Patterns by Artificial Intelligence Methods



**Dalia Kriksciuniene, Virgilijus Sakalauskas, Ivana Ognjanović,
and Ramo Šendelj**

Abstract The variety of the artificial intelligence and machine learning methods are applied for data analysis in various areas, including the data-rich healthcare domain. However, aiming to improve health care efficiency and use the captured information to improve treatment methods is often hampered by poor quality of medical data collections, as high percent of health data are unstructured and preserved in different systems and formats. In addition, it is not always agreed which methods of artificial intelligence and machine learning perform better in different problem areas, and which computer tools could make their application more convenient and flexible. The chapter provides essential characteristics of methods, traditionally applied in statistics, such as regression analysis, as well as their advanced modifications of logit, probit models, K-means, and Neural networks. The performance of the methods, their analytical power and relevance to the healthcare application domain is illustrated by brief experimental computations for investigation of stroke patient database with the help of several readily available software tools, such as MS Excel, Statistica, Matlab, Google BigQuery ML.

Keywords Artificial intelligence · Data mining · Healthcare data · Machine learning algorithms

D. Kriksciuniene · V. Sakalauskas (✉)
Vilnius University, Vilnius, Lithuania
e-mail: virgilijus.sakalauskas@knf.vu.lt

D. Kriksciuniene
e-mail: dalia.kriksciuniene@knf.vu.lt

I. Ognjanović · R. Šendelj
University of Donja Gorica, Podgorica, Montenegro

© The Author(s) 2022
D. Kriksciuniene and V. Sakalauskas (eds.), *Intelligent Systems for Sustainable Person-Centered Healthcare*, Intelligent Systems Reference Library 205,
https://doi.org/10.1007/978-3-030-79353-1_10

10.1 Introduction

In the age of computer technology, there is no shortage of data for analysis. The main problem is to decide what research method to apply, what insights we can get from this data and what decisions they can propose.

Typically, the analysis of a data begins with the application of classical methods of descriptive statistics and visualisation of data, which can help discover a data pattern or show trends in data change. One of the initial data analysis steps is to determine measures of central tendency (mean, median, mode) or measures of variability (standard deviation, data width, variance, asymmetry factor, excess). These characteristics provide a better understanding of the nature of research object and provide an initial picture of the data, their layout, quality and completeness.

The characteristics of data can be easily discovered by using a variety of computer programs, starting from the widely accepted MS EXCEL to specialized statistical calculation environments such as SPSS, STATISTICA, Matlab or Google BigQuery. For solving more advanced research problems, we will use different software and computer tools that will allow the reader to consider the most appropriate solution for a specific artificial intelligence problem.

The discussion and comparative evaluation the artificial intelligence approaches, and the illustration of their performance by applying different AI methods and tools should help us to reveal advantages of artificial intelligence and machine learning methods in the area of application of health data analysis in different cross-sections. This research topic is very popular and attract attention of many researchers [1–7]. For this purpose, we will take a big real clinical record file and try to analyse it using various research methods.

The database applied for the experimental research is a collection of registered stroke cases of the neurology department of Clinical Centre in Montenegro. The database consists of the structured records of 944 different patients, 58 variables, where 50 of them are coded by scale values of [1, 8, 9] corresponding to “Yes, No, Unspecified” conditions, and 8 variables consisting of the demographic data, admission date and discharge date from hospital. The data is collected between 02/25/2017 and 12/18/2019. The demographic data of stroke patients varies by age (from 13 to 96 years), and gender (485-male, 427-female).

Further, we will introduce several data research methods letting us to examine the structure of the data, find important patterns and disclose the relationships of the most important variables. We will try not only to present various research methods, but also will explain how to clean and transform the original data according to the task requirements, and to use different software tools for specific artificial intelligence and machine learning methods.

The next section will focus on understanding regression and correlation analysis and analysing the dependence strength of our data.

The Sect. 10.3 will examine logit and probit regression application to predict the variable *Vital_Status* of the stroke patient from different individual characteristics, such as *Type of stroke*, *Treatment methods*, *Health modified ranking score before*

stroke, *Age at stroke* and *Gender*. Here we will introduce Google BigQuery Machine Learning capabilities to address this type of challenge.

The Sect. 10.4 describes the unsupervised machine learning method k-Means. This method let us partition data records to the predefined number of clusters. The calculations will be performed by the help of Matlab software.

The Sect. 10.5 explored application of neural networks for the supervised learning case of classification, by applying STATISTICA Data Mining tools.

10.2 Correlation and Regression Analysis

In general, regression analysis is a statistical method that allows the estimation of dependence among two or more quantitative variables in order to predict a dependent variable [10].

The simplest regression dependence is linear: $y = \beta_0 + \beta_1 x$. The coefficients of equation are found by the least squares method, i.e. minimizing differences between the points (x_i, y_i) and the regression curve. The regression analysis methods are very widely used in medical research. Usually, to draw regression line and calculate determination or correlation coefficients we can use MS Excel software, but here we will apply the STATISTICA software package and limit our analysis to providing an example of the simplest regression curve.

We will illustrate the task of finding interdependence among the number of days spent in hospital and the age of the patients at stroke, which varied between 20 and 50 years, in Table 10.1 there is a sample of data set used.

Firstly, we explore a scatterplot for visualisation of data and finding a linear regression equation (Fig. 10.1).

As from Fig. 10.1, the linear regression equation for variables denoting age and days at hospital is: $Days\ At\ Hosp = -4,9906 + 0.4102 \cdot Age$. It enables to estimate forecast number of days to be spent at hospital according to the age at stroke. The relevance of the results, and its suitability for forecasting is judged by the coefficient

Table 10.1 Example of the data records

Days at hospital	Age	Days at hospital	Age	Days at hospital	Age
16	50	3	48	3	48
13	49	24	49	24	49
23	49	34	49	34	49
0	48	19	47	19	47
3	50	1	48	1	48
6	49	8	47	8	47
11	50	0	48	0	48
6	50	47	48	47	48

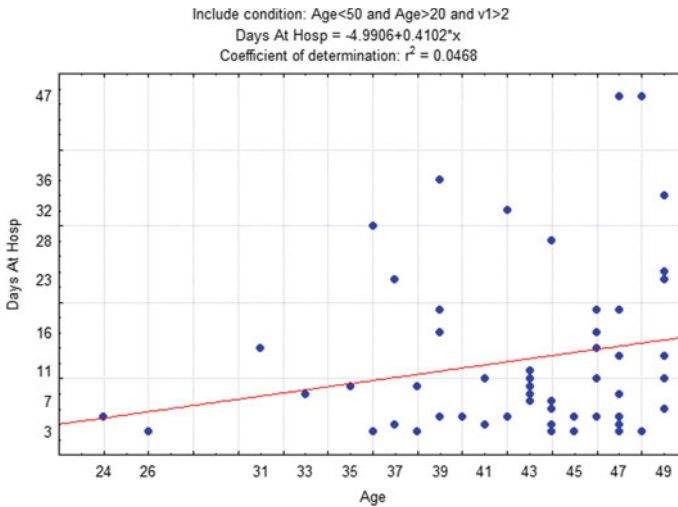


Fig. 10.1 Scatterplot for visualizing linear regression between *Age* and *Days at Hospital*

of determination. In our case (Fig. 10.1) the determination is equal to $r^2 = 0.0468$. The coefficient of determination is interpreted as the proportion of the variance in the dependent variable that is predictable from the independent variable in the range from 0 (no dependence between variables) to 1 (indication of a perfect fit). In the solved example only approximately 5% of variation of the dependent variable *Days At Hosp* can be explained by using the independent variable *Age*.

If the relationship between the variables is not well-fitted to linear (as in Fig. 10.1) we may use the non-linear regression. Then, instead of a line, we explore parabola, exponential, or logarithmic equations, and determine their unknown parameters by the least squares method. If a dependent variable is not well predicted by a single variable, several independent variables can be used to more accurately describe the situation. This type of regression is called group regression. Typically, group linear regression uses no more than 5 or 6 additional variables. Both group and curve regression calculations can be performed using the software tools already mentioned.

10.3 Logit and Probit Models

Traditional regression methods sometimes have difficulty describing a dependent variable that acquires values only from the range $[0,1]$ or values of 0/1 (true/false, success/failure, error/non-error, etc.). In this case, logit or probit regression [11] are appropriated. The main difference among these two models is the different link function. The logit model uses cumulative distribution function of the logistic distribution, and probit invoke the cumulative distribution function of the standard normal distribution. Both functions may take any number as input, and rescale it to fall within

the range of [0; 1]. These regression dependencies are applied for in medical, social science tasks, and are widely used to solve marketing and financial problems.

In order to illustrate the performance of these models we chose an example task, how the condition of blood pressure (0-normal, 1-high) may depend on age, weight, physical activity and stress level of the patient. Other similar examples could be evaluation of prostate enlargement (0-enlarged, 1- normal) from the available health indicators of the patient.

The basic assumptions of the logistic regression model are defined [12]: suppose the dependent variable y acquires a value of 1 with probability p , and it acquires a value of 0 with the probability of $q = 1 - p$. The types of independent variables for building logistic regression model can take any values, i.e. quantitative, qualitative, or categorical. The distribution of the input variables is not restricted for this model either.

In the logistic regression, the relationship between the outcome variable and the descriptive variables is not a linear function, as it was in the case of linear regression. The model of Logistic regression correlates probability value p with the independent variables x_1, x_2, \dots, x_n :

$$p = \frac{1}{1 + e^{-(a+b_1x_1+\dots+b_nx_n)}},$$

where a is a constant, b_i —the regression weights of the independent variables.

This equation takes another form after applying the logistic transformation function (logit) [12].

$$\text{Logit}(p) = \ln\left(\frac{p}{1-p}\right) = a + b_1x_1 + b_2x_2 + \dots + b_nx_n.$$

The link function of the logit regression is expressed by $f(x) = \frac{1}{1+e^{-x}}$, and link function of probit regression is a function of the standard normal distribution $\Phi(x) = \int_{-\infty}^x \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{y^2}{2}\right) dy$, which only slightly differs from the logistic one (Fig. 10.2).

Thus, we can define probabilistic regression as follows:

$$p = \Phi(Z) = \int_{-\infty}^Z \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{x^2}{2}\right) dx,$$

where $Z = a + b_1x_1 + b_2x_2 + \dots + b_nx_n$.

The logistic and probabilistic regressions differ only by the transformation function, which determine differences of the behaviour of these models. The normal distribution function grows faster than the logistic one, therefore it provides a higher sensitivity to probabilistic regression, i.e. dependence on descriptive variables [13].

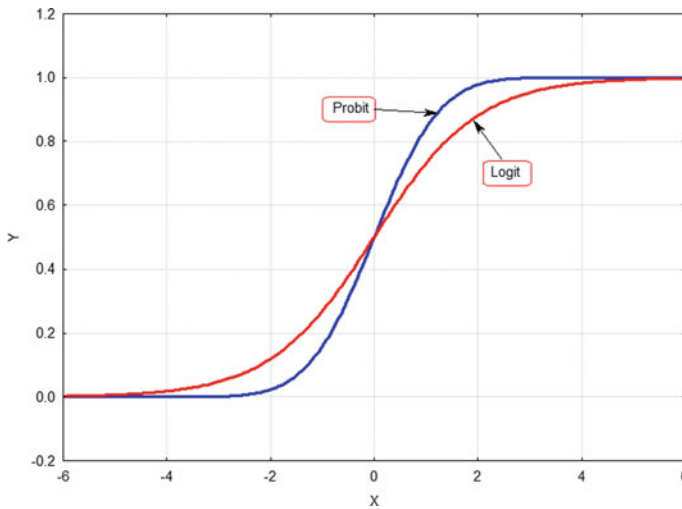


Fig. 10.2 Logit and probit link or transformation functions

The logit and probit regression models belong to the class of the supervised machine learning techniques. It means that the training set with the labelled examples is available for building the model. A supervised learning algorithm analyses the training data learning input/output regularities and produces an inference function, which can be used for estimating output for the new input examples.

We will solve the illustrative example of logit regression with help of Google BigQuery ML (see [8]). BigQuery ML enables to create and execute machine learning models by using standard SQL queries and the ML libraries. BigQuery ML supports not only the linear and logistic regression models, but also provides tools to apply K-means clustering, Matrix factorisation, Time series, Deep Neural Network and other computational intelligence methods.

The 944 data records of patients diagnosed with stroke were used for estimating logit and probit models. We will explore the “Vital status after hospitalisation” (1-Alive, 0-No) as a dependent variable, which is possibly affected by 5 independent variables: (1) Type of stroke, (2) Treatment methods, (3) Health modified ranking score before stroke, (4) Age at stroke, and (5) Gender.

In Table 10.2 the excerpt of data transactions and corresponding variables are presented.

Variable `Stroke_Type` gains value 1 for the diagnosis Ischemic stroke, 2 for Hemorag, 3 for SAH and 4 for unspecified stroke. The variable `Vital_Status` after hospitalisation may take values of Alive marked by 1, or not alive-0. Variable `Treatment_Methods` denote categories of medications, or their combinations, received during the hospital stay, the corresponding values are in Table 10.3.

For example, the code value 13 means combining two types of medication Anti-coagulation and Thrombolysis, 24-Dual Antiplatelet Therapy and medications from the broad group Other. `Health_Status` is evaluated from 0 to good health to 6-very bad

Table 10.2 Example of the data

Vital_status	Stroke_type	Treatment_methods	Health_staus	Age	Gender	Data_frame
1	1	24	0	60	1	T
1	1	2	0	59	2	P
1	1	24	0	59	1	P
1	1	24	0	58	1	P
1	1	24	0	58	2	P
1	3	0	1	58	2	T
0	3	4	0	60	1	T
1	1	24	0	58	1	T
0	1	14	0	59	2	T
1	1	2	0	58	1	T
0	1	24	0	57	1	E
0	1	24	0	57	1	P
1	1	24	0	57	9	T
0	1	14	2	58	2	T
1	2	4	0	57	1	T
0	3	14	1	57	1	T
1	1	1	0	58	1	P
1	4	24	0	57	2	E
0	3	14	1	56	1	E
1	1	1	0	57	9	P
1	1	24	0	57	1	T

health, 9-stands for unknown. Gender code 1 means male, 0-female. The variable Data_Frame ensures the random distribution of the database records to Training-T, Evaluation-E and Prediction-P sets.

The logit regression model can be processed in BigQuery ML, here we need to open Google Cloud platform, BigQuery sandbox, set a new project, create the dataset and upload the data file. Designing the logistic regression model consists of the following steps:

1. Create and train the logistic regression model on training data.
2. Evaluate the model performance with evaluation set of data
3. Predict the output from inputs prediction data

For model creation task we can write a simple SQL query (Fig. 10.3).

Here the 'Logit.Logit' is the name assigned to uploaded table. The achieved performance of logit regression by classifying Vital_Status can be seen from Fig. 10.4.

In Fig. 10.4, the confusion matrix is presented as a table in which predictions are represented in columns and actual status is represented by rows. The performance of the model is explored by applying several characteristics of precision evaluation:

Table 10.3 Codes for treatment methods

Code	Anticoagulation	Dual antiplatelet therapy	Thrombolysis	Others
1	x			
2		x		
3			x	
4				x
12	x	x		
13	x		x	
14	x			x
23		x	x	
24		x		x
34			x	x
123	x	x	x	
134	x		x	x
234		x	x	x
0				

```

Logit_create Edited
1 CREATE OR REPLACE MODEL
2   `Logit.Logit_model`
3 OPTIONS
4   (model_type='LOGISTIC_REG', auto_class_weights=TRUE, input_label_cols=['Vital_Status'])
5 AS
6 SELECT * EXCEPT(Data_Frame) FROM `Logit.Logit` WHERE Data_Frame = 'T'
    
```

Fig. 10.3 Model creation statements

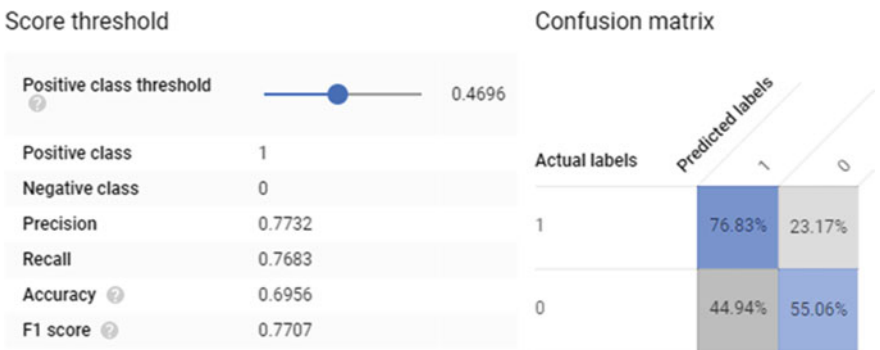


Fig. 10.4 Evaluation of trained logit model

accuracy, recall and precision. The characteristics of *Accuracy* shows what percent of all values are correctly predicted by the model. In our case the general accuracy of prediction is close to 70%. *Recall* calculates the percent of correct predictions of *Vital_Status* for all the true values (=1). It means, that the performance of the model for the Vital Status value “1” is better, than general performance, and equals to 76,83%. The proportion of the instances which were correctly recognized as positive (per total positive predictions) is called the *Precision*. The *F1 score* denotes the harmonic mean of *Precision* and *Recall*. The accuracy of the model may be satisfactory or not sufficient depending on the requirements and complexity of the task solved. In the Fig. 10.4 shows performance of the Logit regression model on training set.

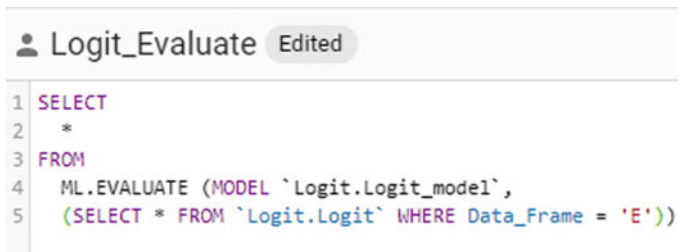
To see the performance of our model on evaluation set we can write an evaluation SQL query on evaluation set (Fig. 10.5).

When the SQL query is executed, BigQuery calculates the accuracy and other model performance characteristics on evaluation set (Fig. 10.6).

As we can see the logit regression model performs on evaluation set even better than on training set. The accuracy and other ratio estimates shows good classification power of *Vital_Status* variable.

The last logit regression modelling step provides model adaptation for prediction set. For this case we need to write prediction SQL query (Fig. 10.7).

The execution of this query let us find the predictions of *Vital_Status* and present the model application results in table (Table 10.4).



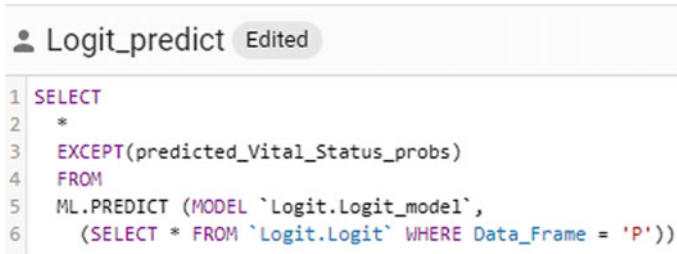
```
1 SELECT
2 *
3 FROM
4 ML.EVALUATE (MODEL `Logit.Logit_model`,
5 (SELECT * FROM `Logit.Logit` WHERE Data_Frame = 'E'))
```

Fig. 10.5 SQL for model evaluation



Row	precision	recall	accuracy	f1_score
1	0.8041958041958042	0.7718120805369127	0.7268722466960352	0.787671232876124

Fig. 10.6 Evaluation results



```

Logit_predict Edited
1 SELECT
2 *
3 EXCEPT(predicted_Vital_Status_probs)
4 FROM
5 ML.PREDICT (MODEL `Logit.Logit_model`,
6 (SELECT * FROM `Logit.Logit` WHERE Data_Frame = 'P'))

```

Fig. 10.7 SQL query for prediction set results

Comparison of the columns “Predicted_Vital _status” and the original values “Vital_status” in Table 10.4 shows that part of the predicted values differ from the original ones, but the overall accuracy calculated for all Prediction set records is equal to 0.684426. This value lets us to conclude the good logistical classification capabilities by applying this method. The presentation of the outcomes in Table 10.4 gives possibility for the advanced further analysis, as the expert analysis of the incorrectly predicted cases may bring insights for adding more input variables, or introduce changes to their coding in order to reduce confusion among the predicted classes and increase the accuracy of model.

10.4 k-Means Clustering

Unlike the Logit and Probit modelling, the Cluster analysis belongs to the class of the unsupervised learning techniques, which enables to find natural groupings and patterns in data, without need of the labelled data set for model training.

K-means clustering is a data partitioning method for assigning records (or objects) to the predefined number of clusters. K-Means treats each observation as an object that has its location in a multidimensional space. The algorithm of k-Means finds a partition in which objects within each cluster are as close to each other as possible, and, at the same time, as far as possible from the objects of the other clusters. Based on the attributes of our data, we can select one of the generally applied distance metric to be used by the k-Means model for calculating distances among the clusters and distances between the instances within cluster.

As k-Means clustering creates a single level of clusters it is suitable for both large amounts of data objects and numerous attributes. Each cluster in a k-Means partition consists of its member objects and has a predefined centre or centroid. K-Means method tries to minimize the sum of the distances between the centroid and each member object of the cluster. The computation procedure depends on the applied distance metrics. By default, k-Means uses the squared Euclidean distance metrics to determine distances. The visualisation of the output of the method plots

Table 10.4 Prediction results for prediction set

Query complete (0.3 s elapsed, 47.1 KB processed)

Row	Predicted_vital_status	Vital_status	Stroke_type	Treatment_methods	Health_status	Age	Gender	Data_frame
101	1	1	1	14	0	63	1	P
102	0	0	3	14	1	61	2	P
103	1	1	1	14	0	61	2	P
104	1	0	1	14	1	61	2	P
105	0	1	3	14	1	53	1	P
106	1	1	3	14	0	50	1	P
107	1	1	3	14	1	48	1	P
108	1	1	3	14	2	44	2	P
109	0	1	1	23	0	86	2	P
110	1	0	1	23	1	76	2	P
111	1	1	1	23	2	74	1	P

the clusters on the two-dimensional space for simplification of the analysis, however the underlying computations deal with the multidimensional settings.

The following steps are performed for k-Means clustering (k-Means Clustering): [14]:

1. Examine k-Means clustering solutions for different selected number of clusters k to determine optimal number of clusters for the data set. Some tools (such as Statistica or Viscosity SOMine) offer estimation of optimal number of clusters;
2. Evaluate clustering solutions by analysing silhouette plots and silhouette values, or based on criteria, such as Davies–Bouldin index values, and Calinski–Harabasz index values;
3. Replicate clustering from different randomly selected centroids and return the final solution with the lowest total sum of distances among all the replicates.

A silhouette value is a standard measure of how close the points of one cluster are to the points of the adjacent clusters. This measure takes values from interval $[-1, 1]$. The value “-1” denotes the points that are probably assigned to the wrong cluster, and silhouette value equal to 1 indicates points that are very distant from the neighbouring clusters. Usually silhouette values are presented graphically by the silhouette plot, which enables to choose the right number of clusters.

The criteria-based method for finding the optimal number of clusters include calculation of Davies–Bouldin (Davies–Bouldin index) or Calinski–Harabasz (Indice de Calinski–Harabasz) index values. Without going into the technical details of calculating these indicators, we will summarize that the optimal number of clusters is indicated by the lowest indicators values.

To illustrate the application of the k-Mean clustering method, we used the extended version of the previously described data file containing various health and personal characteristics of patients diagnosed with stroke. Part of the attributes of this file were explained in Table 10.1, such as the variables of Vital_Status, Type of stroke, Treatment methods, Health modified ranking score before stroke, Age at stroke, Gender, and Days spent in hospital.

For the further study the dataset was expanded by variables expressing other characteristics of the patient history with the assumption that additional knowledge about the patient may increase prediction power of the model. The information whether there was a stroke before, specific Stroke symptoms, and indication of Health complications may enable us to better distribute the patients into meaningful groups and recognise the useful patterns of data. The example of data file records used for k-Means clustering is presented in Table 10.5. According to the concept of multidimensional space associated with clustering computational models we can imagine the file records as points in 9-dimensional space. Contrarily to the supervised methods, all variables serve as inputs.

For this example, we filtered only the record of patients with the Vital_status = 1 (Alive), therefore 642 records we used for research of K-Means clustering. The clustering of patients into predefined number of clusters can be useful in case of meaningful categories (cluster) applied in medical practice, such as separating

Table 10.5 Example data file records for clustering

Days at hospital	Stroke_type	Treatment_methods	Health_status	Age	Gender	Past_stroke	Stroke_symptoms	Health_complications
8	3	0	1	17	2	0	0	0
23	3	4	2	16	1	0	12	0
6	1	24	0	91	1	0	2	0
16	1	24	0	91	1	0	13	0
18	3	4	0	92	1	0	1	0
21	1	24	4	91	2	1	23	0
8	1	2	3	90	2	1	2	0
8	1	24	0	88	1	0	23	0
2	1	24	0	88	2	0	2	0
6	1	24	0	88	2	0	2	3
7	1	24	0	88	1	0	23	4
8	1	4	0	87	1	0	23	2
14	1	4	0	86	2	0	12	0
7	1	24	9	88	2	0	123	0
22	1	24	0	86	2	0	23	0
6	1	12	2	86	2	1	123	0
13	1	4	0	87	2	0	13	4

patients for Rehabilitation, Medication prescription or for appointment of specific health strengthening procedure.

In Table 10.5, *Past_Stroke* value equal to 1 means the repeated stroke case, *Stroke_Symptoms* can take values of: 0-No symptoms, 1-Impaired consciousness, 2-Weakness/paresis, 3-Speech disorder (aphasia) or joint occurrence of several symptoms, e.g. 13 indicates Impaired consciousness and Speech disorder (aphasia). *Health_complications* are divided into four different groups: 1-deep vein thrombosis, 2-other CV complications, 3-pneumonia, 4-other complications. 0 value stands for unspecified complications, and, similarly, the code 13 expresses the double complications of deep vein thrombosis and pneumonia.

The k-Mean clustering can be done by various software, but here we will use MATLAB R2020b version. MATLAB® [15] combines a desktop environment tuned for iterative analysis and design processes with a programming language that expresses matrix and array mathematics directly. Using the predefined Matlab functions we can perform all popular classification, regression, and clustering algorithms for supervised and unsupervised learning.

Matlab enables us to fine tune all parameters of clustering by writing a program code, and to find the optimal number of clusters, as well as to evaluate the clustering solutions by analysing silhouette values, Davies–Bouldin and Calinski–Harabasz index values.

The analysis and clustering of the described data file by using k-Mean algorithm is executed by the Matlab program code presented in Fig. 10.8.

The operator on program line 6 enables us to specify a testing set for the unsupervised learning of k-Means algorithm and to select the attributes. As it is specified by operator 6, for this case we have selected 440 cases starting from record 21 to 460. After initial computation phase we have noticed that variables Days at hospital and Gender have negative influence to the K-Means performance. So, for the following

```

Editor - C:\MATLAB7\work\Kmeans.m
Kmeans.m x +
1 - clc;
2 - clear;
3 - %Virgilijus(c) 2021
4 - load('K_Means1.mat') % Loading the data file
5 - disp(size(M)); %display the number of records and attributes
6 - M=M(21:460,[2 3 4 5 7 8 9]);% selecting testing set and attributes
7 - CH=evalclusters(M,'kmeans','CalinskiHarabasz','KList',[1:5]);%Davies-Bouldin index values
8 - DB=evalclusters(M,'kmeans','DaviesBouldin','KList',[1:5]);%Calinski-Harabasz index values
9 - disp(CH);disp(DB)
10 - [idx,C,sum] = kmeans(M,2,'Distance','cityblock','Display','Iter'); % K-Means function
11 - [silh,h] = silhouette(M,idx,'cityblock'); %Calculation of silhouette value
12 - xlabel('Silhouette Value') %silhouette plot
13 - ylabel('Cluster')
14 - D = mean(silh);
15 - fprintf('Average silhouette values = ');disp(D)
16 - disp(C) % returns the k cluster centroid locations
17 - disp(sum) % sums of distances to centroid centre
18 - %idx(41:50) % Display the clusters for 41:50 records
19 - disp(nnz(idx==1));disp(nnz(idx==2));disp(nnz(idx==3));%the number of cases in clusters

```

Fig. 10.8 Matlab code for k-Means algorithm

CalinskiHarabaszEvaluation with properties:

```

NumObservations: 440
  InspectedK: [1 2 3 4 5]
CriterionValues: [NaN 4.8761e+03 1.1386e+04 9.8071e+03 8.3579e+03]
  OptimalK: 3
    
```

DaviesBouldinEvaluation with properties:

```

NumObservations: 440
  InspectedK: [1 2 3 4 5]
CriterionValues: [NaN 0.0675 0.1944 0.3741 0.5701]
  OptimalK: 2
    
```

Fig. 10.9 Calinski–Harabasz and Davies–Bouldin criterion values

Fig. 10.10 k-Means accuracy verification for 3 clusters

```

      iter  phase      num      sum
      1      1      440     10755
Best total sum of distances = 10755
Average silhouette values = 0.7695
    
```

computation stage we excluded them from our research, and tried to find the clusters only by selecting 2, 3, 4, 5, 7, 8, 9 attributes (see Table 10.4).

In order to find the optimal number of clusters we calculated the Davies–Bouldin and Calinski–Harabasz index values (8 and 9 program lines, Fig. 10.8). The output of 9 line is presented in Fig. 10.9.

The optimal number of 3 clusters was suggested by the Calinski–Harabasz criterion, but Davies–Bouldin criterion advises optimal number of 2 clusters. Therefore we explored both cases of 3 and 2 clusters for evaluation.

The estimation of the K_Mean model to our data was started with $k = 3$ (line 10), it calculated the best total sum of distances to the centroids and average silhouette values. The calculation results are presented on Fig. 10.10.

In Fig. 10.10 the silhouette values equal to 0.7695, which confirm the excellent partition of our cases to 3 clusters. The silhouette plot on Fig. 10.11 visually confirm this assertion. Only very small number of cases have silhouette values less than 0.6 (Fig. 10.11).

Application of k-Means model calculations for 2 clusters show worse performance comparing to the case of 3 clusters (Fig. 10.12).

Although the average of silhouette values of 0.6846 for 2 clusters only differ by small amount from those of 3 clusters. However, the selection of partition of cases to 3 clusters may be more adequate by final expert judgement. After applying the selected K-means clustering model, the clusters can be further explored according to numerous characteristics of the variables included to different clusters. We use Matlab program code to calculate the Number of cases in clusters (line 19) and Sums

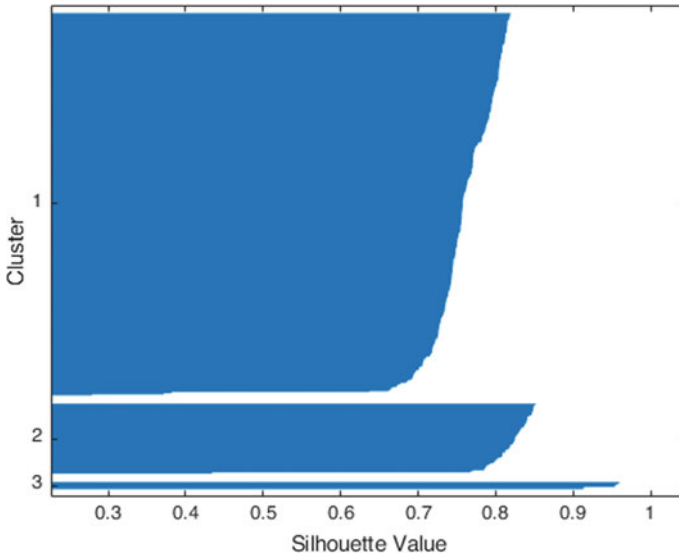


Fig. 10.11 The silhouette plot for 3 clusters

Fig. 10.12 k-Means accuracy verification for 2 clusters

```

iter   phase   num      sum
      1     1    440    19032
Best total sum of distances = 19032
Average silhouette values =    0.6846
    
```

of distances to centroid centre (line 17) in order to characterize the size and similarity of objects within clusters (Table 10.6).

In order to check the membership of a particular patient (or group of patients) to some cluster we may apply different functions of the machine learning environment, such as Matlab: as an example, the command in line 18 (Fig. 10.8) displays the clusters for cases from 41 to 50.

Based on the demonstrated example, we can state that the application of Matlab for machine learning algorithms has a high degree of configuration freedom, allows the researcher to control the parameters of the method and test various computational scenarios. Understanding the background principles of the machine learning

Table 10.6 Cluster information

k-Means clusters (k = 3)	Number of cases in clusters	Sums of distances to centroid centre
1	366	9188
2	67	1266
3	7	301

models and the flexibility of their application in different computational environments enables domain experts and researchers to derive important analytical insights.

10.5 Artificial Neural Network

The Artificial Neural Network (ANN) model is inspired by the biological neural network. It can learn to perform tasks by observing examples, without applying any rules of a particular task.

The ANN model and its modifications is widely used in various application domains, such as language recognition, machine translation models, social network filtering, facial recognition, financial instrument prediction, and many more, where the tasks of classification or time series forecasting are relevant. In medicine, ANN is used to diagnose various diseases and their complications, to evaluate the effects of drugs, to predict the duration of treatment, or to cluster medical anomalies. ANN may link the symptoms of patients with a specific disease, and learn to identify the disease accordingly.

Contrarily to the statistical methods, the ANN is a data-driven approach, therefore the ANN model is trained by available data set for applying it in the testing conditions of the researched domain. For each of the tasks, it is necessary to set up an appropriate neural network. The following methodology should be followed:

1. Preparation of data for the study. These include data collection, organization, normalization, preparing the training and testing sampling.
2. Selection of ANN structure. It is determined by the number of outputs, input variables, hidden layers and the number of neurons of the model. The neuron connection principles, threshold and transmission functions should be determined as well.
3. ANN training. The network training strategy, training algorithm and the training effectiveness needs to be evaluated.
4. Network testing. The evaluation of the created neural network is performed by using an input data set, other than the one used for its training.

All these tasks are highly interrelated and influence the quality of the model. Depending on the available input data set and the task being solved, the appropriate network structure is modelled for applying the most suitable ANN training algorithm. The two most common neural network structures, such as Single-layer perceptron and backpropagation network (multilayer perceptron) are further discussed and explored by presenting the experimental sample.

Single-layer perceptron

Single-layer perceptron is the simplest form of ANN used to classify linearly separated structures. It is a single-layer direct propagation neural network with a threshold transmission function (Fig. 10.13). Rosenblat [16] proved that if such a network is

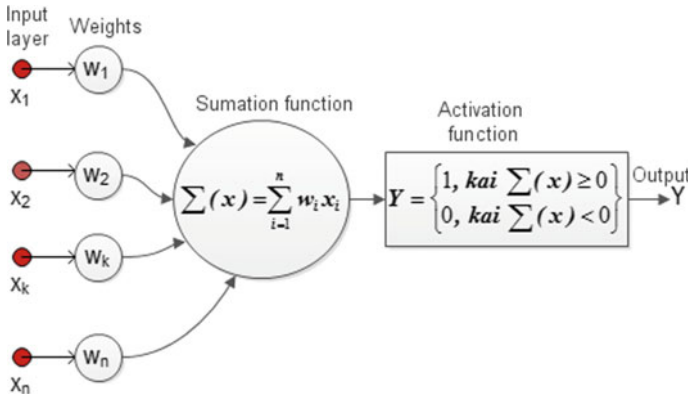


Fig. 10.13 Single-layer perceptron

trained by examples from linearly separable classes, then the perceptron algorithm converges and finds a hyperplane separating those classes.

The solution of the perceptron equation $\sum(x) = 0$ defines a line or hyperplane as the boundary between distinct classes. The solution is obtained by learning the network and choosing the correct network weights. As mentioned, perceptron can only distinguish between linearly separable classes (Fig. 10.14). To describe the structure and training algorithm of perceptron, we will use notations according to Hajek [17].

Perceptron learning is a supervised learning system. Thus the training set consists of pairs $(x^{(p)}, d^{(p)}) \Big|_{p=1}^N$, where $x^{(p)}$ denote the input vector $x^{(p)} = (x_1^{(p)}, x_2^{(p)}, \dots, x_n^{(p)})^T$, and $d^{(p)}$ is the known output a vector (teacher) whose components can acquire only two values: 0 or 1. Let $y^{(p)}$ be the output vector of the neural network.

The error function can be introduced as a vector:

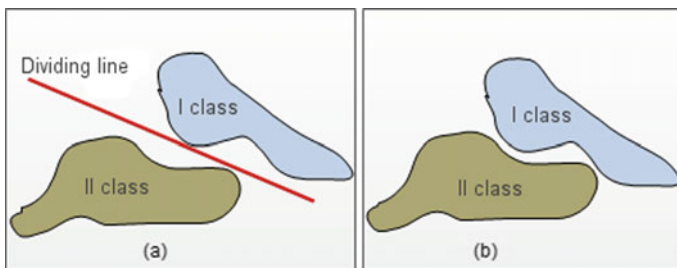


Fig. 10.14 Linearly separable classes (a), Not separable linearly classes (b)

$$J = \sum_{p=1}^N (y^{(p)} - d^{(p)})w^{(p)}x^{(p)}$$

The neural network correctly separates classes when $J = 0$. In all other cases, the separating plane is not found.

In the practical implementation of perceptron training, we change the weights according to the given formula until J becomes as small as possible and no longer changes. If $J = 0$, then our classes were linearly separable and we separated them. If $J \neq 0$, then the classes were not linearly separable and we found the most appropriate separation of those classes.

Several perceptrons can be combined into a more complex network. Such a structure makes it possible to distinguish more complex classes of objects, such as those that can be separated by a plane or a hyper polygon. Figure 10.15 shows a perceptron network with many input and output neurons.

As the perceptron neural network consists of individual perceptron's, each of those can be trained separately according to the algorithm described above. In the 1960s, when perceptron networks became very popular, many researchers thought that any intelligent systems could be constructed with the help of perceptron networks. Unfortunately, it later turned out that far from all systems are so simple. When in 1986 elementary McCulloch-Pitts neural networks was replaced by networks with differentiated activation function and an advanced backpropagation algorithm was described, many complicated systems could be modelled by using such neural networks [18].

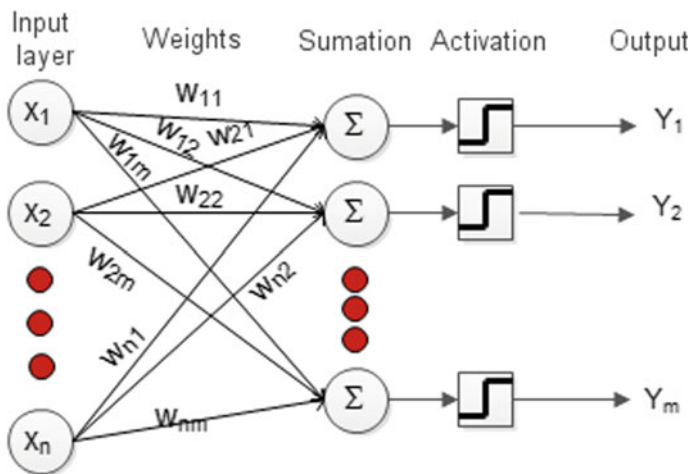


Fig. 10.15 Perceptron network

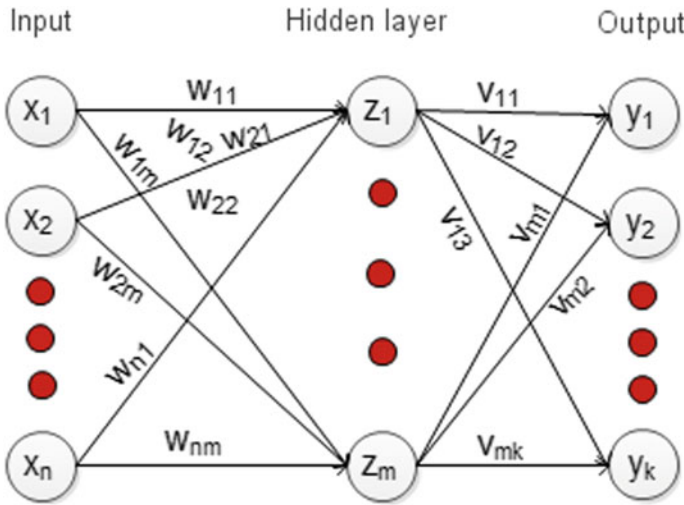


Fig. 10.16 Multilayer perceptron with one hidden layer

Backpropagation networks

The backpropagation network is often referred as the direct propagation multilayer perceptron (Fig. 10.16). His training run with the teacher, having a test set, and the teaching algorithm is called a backpropagation algorithm using a gradient descent method to minimize the total squared error.

The backpropagation algorithm was firstly described in the work of Bryson and Yu-Chi Ho [19], but it did not receive wider recognition until 1986, when Rumelhart et al. [18] published their article. The later period was characterized by a particularly strong development of artificial neural networks and their application.

Using the gradient descent method, it is necessary to differentiate the transfer function with respect to input variables and weights. Thus, nonlinear sigmoidal function or hyperbolic tangent is most commonly used in backpropagation networks. Multilayer perceptron allows the classification of more than just linearly separable classes. Depending on how many neurons are in the hidden layer, we can obtain a separation surface as a convex polygon with approximately as many edges as there are neurons in the second layer.

Once the ANN topology is established, we need to adapt the training algorithm, where a backpropagation algorithm is applied for training of the multilayer perceptron. It consists of two phases: propagation forward and propagation backward.

As the ANN propagates forward, the input variables are transformed layer by layer into output layer variables using fixed weights, thresholds, and transfer functions. In the backpropagation phase, all network weights are recalculated depending on the size of the error signal, which is calculated as the difference between the values of the ANN output variables and the predetermined output vector (teacher).

The opportunity to learn from examples and gain experience has allowed neural networks to be widely used to solve practical problems. Artificial neural networks can help to examine the structure of data, determine its trend, make a forecast, assess risk, or predict impending anomalies. To do this, the neural network must be trained using historical data. The ANN is most commonly used to address classification or clustering challenges because of its greater accuracy and flexibility compared to traditional statistical methods. The most critical challenge for application of the ANN principle in healthcare and other high risk decision domains lays in its “black box” structure: as the model learns from the data set with the labelled output (a teacher) it learns how to estimate the output from the input variables, but it does not provide rules or formulas for clarifying dependencies for decision making. Numerous modifications of the neural network algorithms are proposed in the research works on different conceptual development areas for creating transparency of the ANN performance.

The experimental research

The selected classification task concerns rehabilitation assignment for the patients who have experienced stroke. The experimental analysis was performed for the input data set presented in Table 10.5. As the neural network is a supervised learning algorithm we needed the output variable for training and testing the best performing NN model. Therefore, one more variable of the historical stroke patient database was included, which denotes rehabilitation type prescribed by the expert doctors during the hospitalization. There were four types of rehabilitation therapy (Table 10.7), and the cases with no assignment for rehabilitation were excluded.

Successful solving of this task leads to creating a neural network model which could forecast the output: propose relevant rehabilitation type according to the health characteristics of the patient. The model could also serve to better plan human resources, as different types of rehabilitation required involvement of different specialists and schedule their time.

The analysis enables to solve what kind of rehabilitation is most likely to be prescribed according to the nine input variables (Table 10.5) serving as health characteristics of the patient.

Several experiments were performed for exploring ANN models performance. In the first step, the neural network models were generated from the data set by applying different algorithms, and three best performing models were retained for

Table 10.7 Rehabilitation types

Code	Code value	Rehabilitation prescribed
1	RWt	Working therapy
2	RPt	Physical Therapy
3	RSp	Speaking exercise
4	RSw	Swallowing exercise
0	RNo	No rehabilitation

Model Summary Report (DataRehabNN Rh)							
Index	Profile	Train Perf.	Select Perf.	Test Perf.	Inputs	Hidden(1)	Hidden(2)
1	Single Layer NN 9:39-4:1	0,84	0,85	0,79	9	0	0
2	MLP 10:45-12:11-4:1	0,95	0,81	0,76	10	12	11
3	RBF 9:44-17-4:1	0,83	0,88	0,81	9	17	0

Fig. 10.17 Multilayer perceptron with one hidden layer

further analysis. The second step had to explore the accuracy of the models in solving the classification task by analysis their general performance, and the confusion among the output classes. The third step had to reveal the importance of different variables for building the neural network model. The last step had to investigate classification behaviour at different value ranges of the variables. The last two steps had to provide solution for the “black box” nature of the ANN models. In the healthcare problems the situation of the “black box” is mainly not acceptable, as it means that the ANN model may just advice the output without providing rules or explanatory insights, therefore many modifications and solutions of the ANN algorithms are being explored for converting ANN to “grey box” or “white box”.

The STATISICA for Windows software was used to design the neural network. The data set was randomly split to three subsets used for training (70%), selecting evaluation set (15%), and testing (15%).

Three models were retained (Fig. 10.17), we can see that different algorithms, such as Single Layer perceptron NN, MLP (multilayer perceptron), RBF (Radial basis function), had similar performance. The MLP model was the most accurate in the training stage (0,95), whereas the RBF was slightly better in the testing stage, which may indicate good performance for the unknown new data set. The general classification precision of the models in different stages varied between 0,79 and 0,95 (Fig. 10.17), which indicates good possibility to propose most suitable rehabilitation type. The structure of the neural network models retained is described by their profile data (Fig. 10.17), which denotes number of input variables (9), number of neurons in each hidden layer, and one output variable with the four classification outcomes (Table 10.7).

As the model aims to correctly select the output value, namely the rehabilitation type, we may explore the performance of different models while assigning particular output values. In Fig. 10.18 the confusion matrix reveals, that the Single Layer NN model had quite significant confusion among classes: it could not assign the rehabilitation types of RWt and RSp to any of the classes, while most of the cases of RSw were wrongly assigned to RPt. Similar confusion problems were demonstrated by the RBF model. Despite similar general accuracy of the models, the best ability to recognize different output classes was shown by MLP model.

The confusion problem may be determined by different number of cases with various output, used for training the models. In our case the biggest number of cases had the output variable value RPt; or it may be determined the significance of different variables which may be explored by sensitivity analysis of the designed

	Confusion Matrix - RehabA(1,2,3) (DataRehabNN Rh)			
	RWt	RPt	RSw	RSp
RWt.1-Single Layer NN	0	0	0	0
RPt.1-Single Layer NN	37	520	31	23
RSw.1-Single Layer NN	2	16	12	1
RSp.1-Single Layer NN	0	0	0	0
RWt.2- MLP	12	4	0	0
RPt.2- MLP	23	507	10	13
RSw.2- MLP	4	17	32	4
RSp.2- MLP	0	8	1	7
RWt.3- RBF	0	0	0	0
RPt.3- RBF	39	536	43	24
RSw.3- RBF	0	0	0	0
RSp.3- RBF	0	0	0	0

Fig. 10.18 Confusion matrix

	Sensitivity Analysis - Models 1, 2 and 3 (DataRehabNN Rh)					
	Days at Hospital	Stroke_Type	Treatment_methods	Age	Stroke_Symptoms	Health_complications
Ratio.1:Single Layer NN	1,00	1,00	0,00	1,00	1,00	1,00
Rank.1:Single Layer NN	6	2	9	3	8	1
Ratio.2: MLP	1,01	1,11	1,11	1,26	1,23	1,05
Rank.2: MLP	8	4	3	1	2	5
Ratio.3: RBF	1,00	1,02	1,02	1,03	1,03	1,01
Rank.3: RBF	7	4	3	1	2	5

Fig. 10.19 Sensitivity analysis of the most influential variables

neural network models. In Fig. 10.19 the variables are ranked by calculating ratio of their significance.

In Fig. 10.19 the sensitivity analysis revealed different importance of the variables for generating ANN models. The most influential variables are shown: for the Single Layer NN the *Health complications* were ranked 1st, but for the MLP and RBF models the *Age* and *Stroke Symptoms* were ranked correspondingly 1st and 2nd. The sensitivity analysis may advice the areas for more detailed investigation and improving precision of the models. It can be achieved by enhancing richness of data in the areas related to the most significant influences and identifying most vulnerable areas of inaccurate performance.

The performance of the MLP model in recognizing values of the output variables denotes strongest reliability of the model for rehabilitation of type RPt (94,6% correct), while the model is not useful for the RWt (30,77% correct) and RSp (29,17% correct). It can be noticed, that relatively small number of cases with different outputs is not the determining factor, as the RSw (43 cases) accuracy is 74,42% whereas RWt had similar number of cases (39) with much lower performance (30,77%) (Fig. 10.20).

Application of the ANN algorithms and models in healthcare has broad potential due to their computational power, and as the regression, classification or time series-related tasks are important in the healthcare processes related to diagnosis, treatment, rehabilitation and many others. However, the experimental research has demonstrated necessity to apply various approaches not only for building models and analysing

	Classification (MLP) (DataRehabNN_Rh)			
	RehabA.RWt	RehabA.RPt	RehabA.RSw	RehabA.RSp
Total	39	536	43	24
Correct	12	507	32	7
Wrong	27	29	11	17
Unknown	0	0	0	0
Correct(%)	30,77	94,6	74,42	29,17
Wrong(%)	69,23	5,4	25,58	70,83
Unknown(%)	0,00	0,0	0,00	0,00

Fig. 10.20 Sensitivity analysis of the most influential variables

their general accuracy, but for their in-depth analysis of performance, influences and possible sources of vulnerabilities and inaccuracies.

10.6 Conclusion

The chapter provides essential characteristics of methods, traditionally applied for data processing, such as regression analysis, as well as their modifications towards the area of artificial intelligence methods, such as logit, probit models, K-means, Neural networks. The healthcare domain uses variety of data sources and measurement scales, as well as different target requirements for output information. It implies that different methods have to be considered for solving tasks, while the in-depth analysis of the generated solution models may bring to adoption or rejection of different models due to their imbalanced reliability in different classes, segments of cases. The performance of the methods, their analytical power and relevance to the healthcare application domain is illustrated by brief experimental computations for investigation of stroke patient database. Various software tools, such as STATISTICA, Matlab, Google BigQuery ML were applied for analysis, ensuring broad variety of analytical tools for in-depth analysis of generated solutions and deriving new insights for their improvement. The regression analysis, characteristics and the experimental examples of their applications reveal advantages, disadvantages, and causes of irrelevant application of the methods. The analytical tools not only enhance transparency of the artificial intelligence data driven models, but may indicate areas of improving data quality, or initiate potential sources for supplementing enriched data related to the most influential variables characterizing persons and various aspects of healthcare.

Acknowledgements This publication is based upon work from COST Action “**European Network for cost containment and improved quality of health care-CostCares**” (CA15222), supported by COST (European Cooperation in Science and Technology)

COST (European Cooperation in Science and Technology) is a funding agency for research and innovation networks. Our Actions help connect research initiatives across Europe and enable scientists to grow their ideas by sharing them with their peers. This boosts their research, career and innovation.

<https://www.cost.eu>



References

1. Buch V.H., Ahmed, I., Maruthappu, M.: Artificial intelligence in medicine: current trends and future possibilities. *Br. J. Gen. Pract.* **68**, 143–144 (2018)
2. Hosny, A., Aerts, H.J.W.L.: Artificial intelligence for global health. *Science* **366**, 955–9556 (2019)
3. Neill D.B.: Using artificial intelligence to improve hospital inpatient care. *IEEE Intell. Syst.* **28**, 92–95
4. Panch, T., Pearson-Stuttard, J., Greaves, F., Atun R.: Artificial intelligence: opportunities and risks for public health. *Lancet Dig. Health* **1**, e13–e14 (2019)
5. Reddy, S.: Use of artificial intelligence in healthcare delivery, In: *eHealth-Making Health Care Smarter*, pp. 81–97. IntechOpen (2018)
6. Sanders S.F., Terwiesch, M., Gordon, W.J., et al.: How Artificial Intelligence Is Changing Health Care Delivery. <https://catalyst.nejm.org/health-care-aisystems-changing-delivery/>. Accessed 10 Jan 2021
7. Triantafyllidis A.K., Tsanas, A.: Applications of machine learning in real-life digital health interventions: review of the literature. *J. Med. Internet Res.* **21**, e12286 (2019)
8. BigQuery: <https://cloud.google.com/bigquery-ml/docs/introduction> Accessed 10 Jan 2021
9. Davies–Bouldin index: https://en.wikipedia.org/wiki/Davies%E2%80%93Bouldin_index. Accessed 10 Jan 2021
10. Nisbet R., Elder J., Miner G.: *Statistical Analysis & Data Mining Applications*. Elsevier, Canada (2009)
11. Statistica: <http://www.statsoft.com/textbook/>. Accessed 10 Jan 2021
12. Krikščiūnienė, D., Sakalauskas, V.: *Intelektiniai modeliai marketingo sistemoje*: Monograph in Lithuanian. Vilnius, Vilniaus universiteto leidykla, 384 p (2014)
13. Pfaffenberger Roger, C.: *Patterson Jamer H. Statistical methods for business and economics.*-Richard D. Irvin, INC., 828 s. (1981)
14. k-Means Clustering: <https://se.mathworks.com/help/stats/k-means-clustering.html>. Accessed 10 Jan 2021
15. Matlab: https://se.mathworks.com/products/matlab.html?s_tid=hp_products_matlab. Accessed 10 Jan 2021
16. Rosenblatt, F.: The perceptron: A probabilistic model for information storage and organization in the brain, cornell aeronautical laboratory, *Psychological Review*, vol. **65**, No. 6, 386–408 (1958). <https://doi.org/10.1037/h0042519>
17. Hajek M.: “Neural Networks”. (2005). <http://www.cs.unp.ac.za/notes/NeuralNetworks2005.pdf>
18. Rumelhart, D.E., Hinton, GE., Williams, R.J.: Learning representations by back propagating errors, *Nature* **323**, 533–536 (1986)
19. Bryson, A.E.Jr., Yu-Chi Ho.: *Applied optimal control*. Blaisdell Publishing Co. (1969)
20. Indice de Calinski-Harabasz: https://fr.wikipedia.org/wiki/Indice_de_Calinski-Harabasz. Accessed 10 Jan 2021

Open Access This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.

