TripleSec

true

2021-08-31

Introduction

In a day and age, where more data per patient is collected and has to be evaluated, health care providers are starting to rely on clinical decision support systems (CDSS). The systems support health care providers by sifting through the patient data and suggesting further steps of treatment depending on evidence based clinical guidelines [1].

The Triple Sec Algorithm is a clinical support algorithm for risk assessment of Alzheimer's disease (AD), which is the most common cause of a decline in mental ability, dementia [2].

Dementia is a general term, which is used to classify a group of neurodegenerative disease patterns [3]. These diseases are classified by a general decline in mental abilities caused by changes of the brain. The cause for the changes varies between the diseases, though symptoms usually include problems with language, mood changes, disorientation and memory loss [3], which are often mistaken for normal aging due to the fact that most neurodegenerative diseases, including AD, are more common in patients over 65 years of age [4].

Alzheimer's disease is a highly complex disease, which causes changes in the brain on a cellular and structural level. These changes can occur years before the first symptoms are visible [3]. The changes in the brain are irreversible, nor is the progression of AD stoppable, though the treatments available today could temporarily improve symptoms [5]. To improve the patients life quality as much and for as long as possible, an early diagnosis is critical. Studies have also shown that early diagnosis, and therefore early treatment, increases the chances of the efficiency of the treatment [6].

According to current statistics over 50 million people are living with dementia today and a lot of them have not been diagnosed. Statistics estimate that someone in the world develops dementia every 3.2 seconds, leading to about 10 million new dementia cases each year [7]. Due to the growing and aging population demiological studies suggest the number of worldwide dementia patients will increase to over 130 million in 2050 [4].

Due to the number of cases there is an urgent need for early and reliable diagnosis. The problem with this is that the cause of Alzheimer's disease, thus the reason why the patients brain changes on a cellular and structural level is unknown. An additional problem is that the changes can only scarcely be detected without invasive and costly procedures due to the lacking consistency between different clinical, laboratory and imaging findings. For the task of selecting affected individuals with confidence, we propose the created Triple Seq algorithm, which identifies patients dependent on user given risk thresholds by linking several predictors to the diagnostic procedure which are optimal to determine the given risk level. Thus the patients are categorized into risk categories dependent on their neurological symptoms. These categories, high risk, low risk and indiscriminate, represent the risk of the patient progressing to Alzheimer's disease, dependent on the given biomarker values in the completed examinations. The categories are defined by cutoff values of the

biomarkers. These cutoff values are calculated dependent on user defined risk-threshold and ratio of patients. The algorithm allows healthcare providers to justify more invasive or more expensive treatment options for high risk converters [8].

Alzheimer

"A degenerative disease of the brain characterized by the insidious onset of dementia. Impairment of memory, judgment, attention span, and problem solving skills are followed by severe apraxias and a global loss of cognitive abilities. [...]" - ICD 10 definition of Alzheimer [9]

Dementia is a general term, which is used to classify a group of neurodegenerative symptoms concerning language usage, mood changes, disorientation and memory loss [2,3]. The disease patterns lead to a significant impairment of everyday activities, causing patients to rely on others [3,4]. The diseases which evoke dementia are caused by brain changes on the cellular level leading to structural changes of the brain. These transformations are often caused by accumulations of misfolded, disease-specific proteins, though there are histopathological characteristic overlaps in many diseases [3]. For example about one third of Alzheimer's patients have transactivation response dna-binding protein deposits additional to the AD typical beta-amyloid beta and tau protein deposits [3].

Current statistics estimate that over 50 million people are living with dementia today , and also that someone in the world develops dementia every 3.2 seconds, which leads to an additional 10 million new dementia cases each year [7]. Due to the growing and aging population demiological studies suggest the number of worldwide dementia patients will increase to over 130 million in 2050 [4].

Alzheimer's disease (AD) is by far the most common form of dementia accounting for 60-80% of the cases [2]. Even though the disease affects so many people, the cause for the deposits of misfolded proteins is still unclear. There are various hypotheses ranging from genetic mutations to chronic inflammations [3,5]. The biggest risk factor is age, as Alzheimer's is more common in patients over 65 years of age, which makes the diagnosis that much harder as the first signs of AD, such as short term memory loss and subtle problems with attentiveness, planning, abstract thinking, orientation and flexibility, are often mistaken for normal aging [10]. This intermediate stage between normal aging and an onset of dementia is called mild cognitive impairment (MCI) [3,4]. The course of progression of AD is usually categorized in three stages, early or mild, middle or moderate, and late or severe, with a continual decrease in cognitive and functional capabilities ending in the patients death [4].

Diagnosis

The course of Alzheimer's dementia with the depositions of protein fibrils in the affected nerve cells, typically beta-amyloid and tau protein deposits, begins 10-15 years before the first symptoms become visible. Due to the fact that the cause of Alzheimer's disease, thus the reason why the patients brain changes on a cellular and then structural level is unknown, it is difficult to detect the changes early on. Therfore healthcare professionals have to act quickly when the first symptoms become visible and the patient enters the stage of MCI. During the stage of MCI, though the symptoms are often mistaken for normal aging thefore often not being recognized by the patient nor by their treating physician. In general it is difficult to make a reliable diagnosis early on and be able to treat symptoms quickly and correctly. The only reliable and definitive ways to diagnosis the patients brain for protein fibrils, though this is not possible while the patient is still alive. The changes can only scarcely be detected otherwise due to the lacking consistency between different

clinical, laboratory and imaging findings therefore never creating a good base for them to justify their need for invasive tests.

The health care professionals are left with an exclusion procedure by doing these tests and ruling out other diseases which could cause the symptoms.

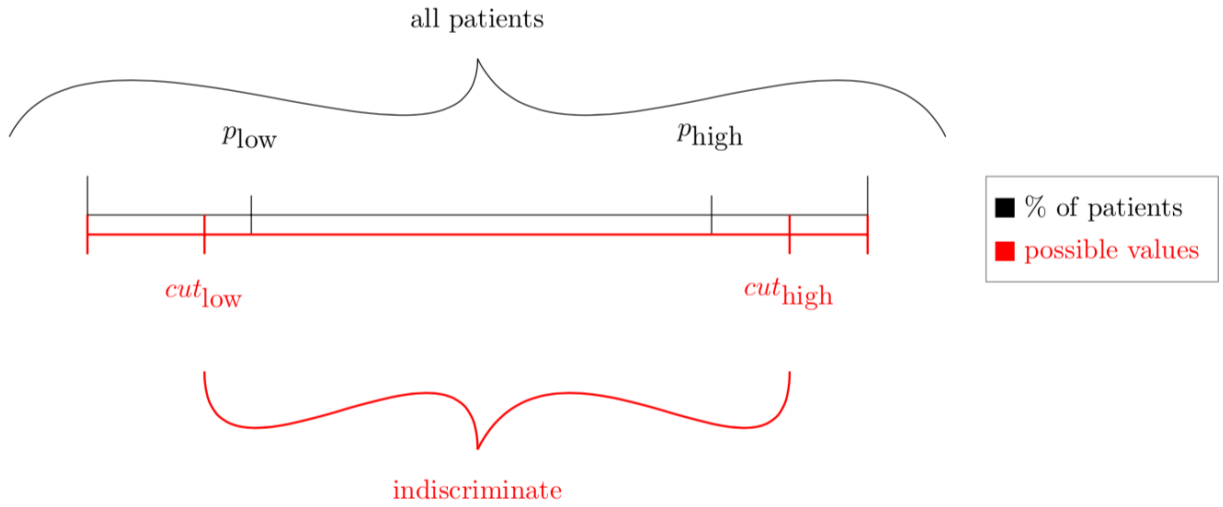
Triple Sec

For the task of selecting patients that are affected by Alzheimer's disease with confidence we propose the created Triple Seq algorithm, which is a clinical support algorithm. It identifies patients by creating links between the preformed neurological and image test by means of a user given risk thresholds. This is done by linking several predictors to the diagnostic procedure, which are optimal to determine the given risk level.

The biomarker values resulting from the diagnostic procedures are the base for the designated risk levels which are categorized into three different categories. These categories are high risk, low risk and indiscriminate. The biomarker values are classified based on cutoff values which are calculated dependent on user defined risk-threshold and ratio of patients. The algorithm allows healthcare providers to justify more costly or invasive treatment options for high risk converters [8].

Algorithm flow

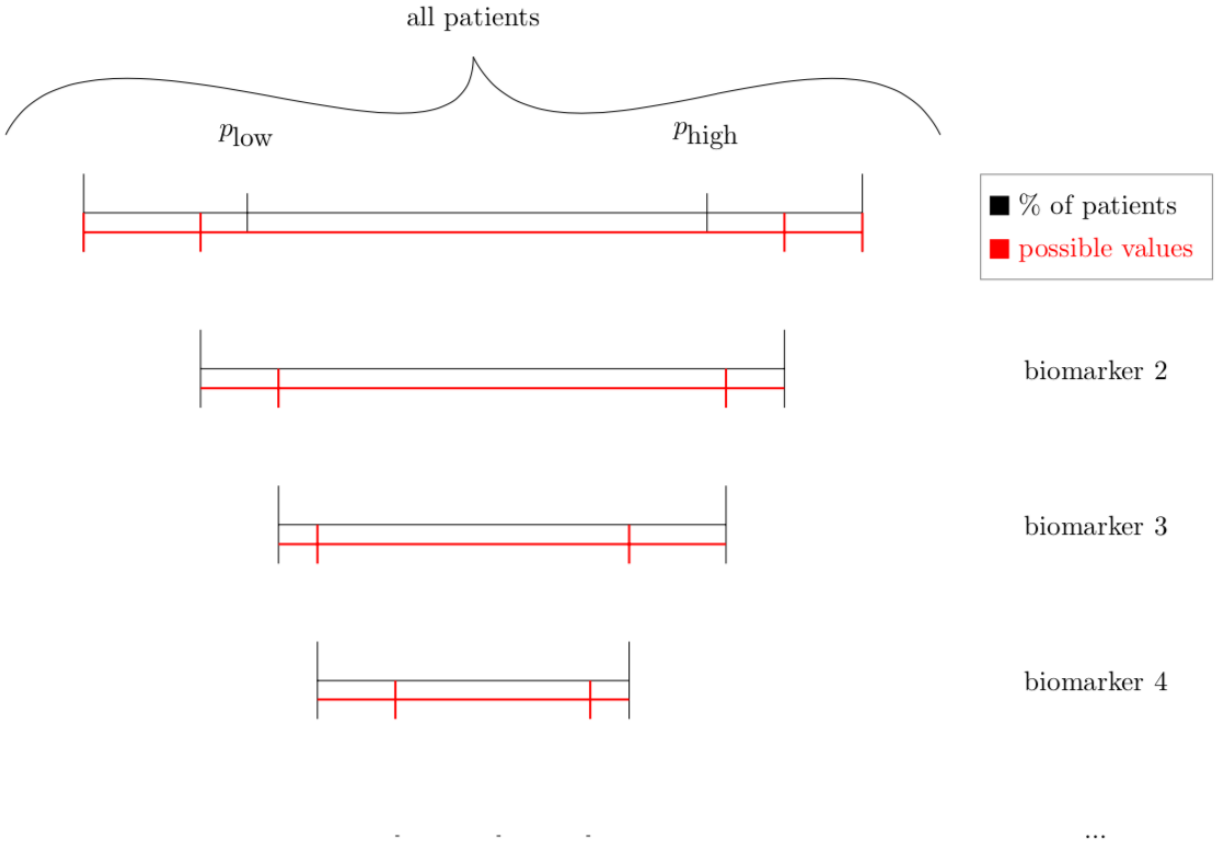
The TripleSec algorithm categorizes the patients into three categories depended upon their biomarker values. The user provides the algorithm with a data table including the patients which are supposed to be assessed and the biomarkers as well as two risk-threshold values, pHigh and pLow. The algorithm proceeds to iterate through all patients using all but one patient to calculate the cutoff values and assesing the risk of conversion of the last patient. The calculation of the cutoff values is based upon the user given risk-threshold, pHigh and pLow. They represent the percentage of patients which are supposed to have a certain biomarker larger or equal to x (pHigh) or smaller or equal to x (pLow). During each iteration the biomarker value is decreased (pHigh) or increased (pLow) by 5% of the highest (pHigh) or lowest (pLow) value. The biomarker value which is the first value upon which the percentage of patients crosses the risk-threshold is used as the cutoff value (cutHigh, cutLow) for that particular biomarker.



This cutoff value calculation is repeated for each biomarker given from the user in the data table. After each

biomarker all patients that can be classified as high risk or low risk are excluded from the next biomarker calculation. Patients are categorized as high or low risk by their biomarker values, if their value is higher or equal to cutHigh they are classified as high risk and if their value below or equal to cutLow they are classified as low risk.

This allows the algorithm to focus on the indiscriminate ("unkown risk") section and classifying more patients during each iteration.



Preinstallation

TripleSec depends on one package: readr.

Please ensure to install the readr package before using the TripleSec package to ensure the data file used as inputData can be processed.

Installation via GitHub

```
require(readr)
install_github("zaynabhammoud/TripleSec")
library(TripleSec)
```

Available Functions

pruneTree calculate cutOff table for given data

This function divides the given input data frame into training and test data. All patients but one are used as training data and are used in the function calCutOff, which returns a matrix containing the calculated cutoff values. Using this matrix and the test patient, pruneTree then calls predictTree. This function asseses the risk of conversion for the test patient.

The function pruneTree returns a variable result with a link to a table containing the average cutoff values for each feature and a data table with the patients and feature and their risk assessment, once it has iterated through all input data and using each patient as a test patient. The risk assessment is an additional column for each biomarker which is added in the function with the values 1 for a high conversion risk and 0 otherwise.

The function requires the following arguments:

- **inputData** - The prepared input data. The data can be prepared via the function `prepareData(data, firstFeature, featureList)`
- **pHigh** - The high risk threshold
- **pLow** - The low risk threshold

Example

```
inputData <- prepareData(data, firstFeature, featureList)
pHigh <- 0.8
pLow <- 0.2
result <- pruneTree(inputData, pHigh, pLow)
```

calCutOff Sequential training of the algorithm

This function is used to calculate the cutoff values for low risk and high risk groups.

The values are calculated by the proportion between converters with a value and all patients having this certain value corresponding with the given pHigh and pLow values. For each feature cutoff calculation all patients that have not been categorized into the high or low risk categories will be used.

The function calCutOff returns a variable fit, which contains links to the calculated cutoff table, filtered data and effectiveness of the cutoff values.

The function requires the following arguments:

- **inputData** - The prepared input data for training of the algorithm, which is provided by pruneTree.
- **pHigh** - The high risk threshold.
- **pLow** - The low risk threshold.

predictTree Sequential prediction of a patient conversion risk

This function calls the recursive function predictTreeRec. The function returns a variable with a link to a patient data table containing their risk assessments and a link to the average cutoff value matrix.

The function predictTree returns a variable result with its links to the cutoff matrix and predicted patient information.

The function requires the following arguments:

- **cutOff values** - The calculated cutoff values, which are provided by the function calCutOff.
- **inputData** - The inputData/the patient which is supposed to be predicted (provided by pruneTree).

predictTreeRec Sequential prediction of a patient conversion risk

This recursive function calculates the cutoff value for each parameter dependent upon the patient ratio, it also verifies if the model is effective (cut_high > cut_low).

The function predictTreeRec returns a variable result with its links to the cutoff matrix and predicted patient information.

The function requires the following arguments:

- **cutOff values** - The calculated cutoff values, which are provided by the function calCutOff.
- **inputData** - The inputData/the patient which is supposed to be predicted (provided by pruneTree).

prepareData Filters and transforms given data

This function transforms the given data into a data frame containing only the columns necessary for the TripleSec algorithm, therefore for the risk assessment.

The function requires the following arguments:

- **data** - The data which is given by the user.
- **firstFeature** - The name of column containing the feature data which is most relevant.
- **featureList** - A list contaning all column names of the features which the algorithm is supposed to work with.

Example

```
inputData <- prepareData(dataFile.csv, "adasCog", c("fdgPet, p_tau"))
```

Testing

The TripleSec algorithm was tested with a csv data file containing data of 144 individuals.

RID	conversion	TimeToConversion	Ab1to42	p_tau	fdgPet	hippoVolume	ApoE4	AVLT	sex	age	edu	adasCog
4170	false	24	166	33	1.5357	4578	positive	55	female	65.6	18	3
4636	false	24	129	24	1.2511	2998	positive	47	female	77.1	16	3
188	false	36	109	55	1.1421	3185	positive	47	male	86.1	16	4
718	false	36	133	40	1.462	3506	negative	46	female	80.4	18	4
1217	true	12	172	41	1.1922	3062	positive	35	female	67.8	16	4
4767	false	24	157	16	1.4827	3054	positive	56	female	66.4	18	5
994	false	48	133	61	1.1327	3490	positive	31	female	55.1	18	5
1030	false	48	162	51	1.0518	4000	negative	41	male	67.4	20	5
1034	false	60	150	18	1.1155	3898	negative	29	male	75.1	18	5
291	false	72	111	33	1.3814	3545	positive	31	male	79.4	16	5
1421	false	36	142	49	1.1992	2648	positive	37	female	74.4	12	6
112	false	96	126	27	1.2006	4128	positive	50	male	70.6	18	6
1120	false	24	136	19	1.1108	1990	negative	40	female	77.9	20	7
4029	false	24	169	36	1.3859	4656	positive	45	male	61.1	20	7
4232	false	24	156	22	1.1731	3712	negative	23	male	74.2	16	7
285	false	72	129	38	1.3424	3873	positive	49	male	65.6	10	7
378	false	72	129	53	1.1362	3066	positive	35	female	69	16	7
4857	true	6	154	56	1.1112	3344	positive	30	male	68.3	16	7
101	true	24	142	19	1.2278	3264	positive	24	male	73.6	18	7
892	true	24	121	36	1.1652	3069	positive	44	female	72.8	12	7
544	false	24	109	15	1.0578	2223	negative	51	female	76.7	12	8
4311	false	24	180	37	1.255	2749	negative	44	female	72.5	18	8
4510	false	24	126	79	1.297	3820	positive	41	female	66.4	12	8
4611	false	24	124	40	1.207	3806	positive	34	male	66.9	18	8
...	...	...	...	...	...	...	...	...	...	...	...	...

Using the function prepareData with the csv file, firstFeature equaling “adasCog” and the featureList being c(“fdgPet”, “p_tau”) the csv file was transformed into a data frame which then was used as inputData for pruneTree.

```
library(readr)
library(TripleSec)
dataAD <- read_csv("predictAD.csv")
#>
#> -- Column specification --------------------------------------------------------
#> cols(
#>   RID = col_double(),
#>   conversion = col_logical(),
#>   TimeToConversion = col_double(),
#>   Ab1to42 = col_double(),
#>   p_tau = col_double(),
#>   fdgPet = col_double(),
#>   hippoVolume = col_double(),
#>   ApoE4 = col_character(),
#>   AVLT = col_double(),
#>   sex = col_character(),
#>   age = col_double(),
#>   edu = col_double(),
#>   adasCog = col_double()
#> )
firstFeature <- "adasCog"
featureList <- c("fdgPet", "p_tau")
inputData <- prepareData(dataAD, firstFeature, featureList)
pHigh <- 0.8
pLow <- 0.2
result <- pruneTree(inputData, pHigh, pLow)
print(result$cutOff)
#>           adasCog    fdgPet     p_tau
#> cut_high 15.53264 1.5920333 102.18785
#> cut_low   7.94375 0.9156108  24.04688
print(result$data)
#>          ID adasCog riskClass fdgPet riskClass p_tau riskClass
#>   [1,] 4170       3         0 1.5357         1    33         1
#>   [2,] 4636       3         0 1.2511         1    24         1
#>   [3,]  188       4         0 1.1421         1    55         1
#>   [4,]  718       4         0 1.4620         1    40         1
#>   [5,] 1217       4         0 1.1922         1    41         1
#>   [6,] 4767       5         0 1.4827         1    16         0
#>   [7,]  994       5         0 1.1327         1    61         1
#>   [8,] 1030       5         0 1.0518         1    51         1
#>   [9,] 1034       5         0 1.1155         1    18         0
```

```
#>  [10,]  291       5          0 1.3814          1    33          1
#>  [11,] 1421       6          0 1.1992          1    49          1
#>  [12,]  112       6          0 1.2006          1    27          1
#>  [13,] 1120       7          0 1.1108          1    19          0
#>  [14,] 4029       7          0 1.3859          1    36          1
#>  [15,] 4232       7          0 1.1731          1    22          0
#>  [16,]  285       7          0 1.3424          1    38          1
#>  [17,]  378       7          0 1.1362          1    53          1
#>  [18,] 4857       7          0 1.1112          1    56          1
#>  [19,]  101       7          0 1.2278          1    19          0
#>  [20,]  892       7          0 1.1652          1    36          1
#>  [21,]  544       8          0 1.0578          1    15          0
#>  [22,] 4311       8          0 1.2550          1    37          1
#>  [23,] 4510       8          0 1.2970          1    79          1
#>  [24,] 4611       8          0 1.2070          1    40          1
#>  [25,] 4746       8          0 1.1833          1    31          1
#>  [26,] 4782       8          0 1.2061          1    20          0
#>  [27,]  608       8          0 0.8747          0    21          0
#>  [28,] 1419       8          0 1.2171          1    38          1
#>  [29,]  800       8          0 1.3726          1    22          0
#>  [30,] 4203       8          0 1.1874          1    47          1
#>  [31,]  906       8          0 1.0801          1    30          1
#>  [32,]  231       8          0 1.0658          1    30          1
#>  [33,] 4713       9          1 1.3541          1    29          1
#>  [34,] 4736       9          1 1.1592          1    97          1
#>  [35,]  481       9          1 1.2971          1    31          1
#>  [36,]  621       9          1 1.5989          1    46          1
#>  [37,] 4197       9          1 1.2333          1    28          1
#>  [38,]  362       9          1 1.2156          1    70          1
#>  [39,]  723       9          1 0.9861          1    24          0
#>  [40,] 4015       9          1 1.1575          1    68          1
#>  [41,] 4715       9          1 1.0710          1    82          1
#>  [42,] 4631      10          1 1.4113          1    62          1
#>  [43,] 1043      10          1 1.3289          1    24          1
#>  [44,] 1265      10          1 1.1094          1    30          1
#>  [45,] 4030      10          1 1.0384          1    84          1
#>  [46,]   51      10          1 1.3698          1    30          1
#>  [47,] 4414      10          1 1.2569          1    61          1
#>  [48,]  204      10          1 1.3138          1    30          1
#>  [49,]  222      10          1 1.1466          1    45          1
#>  [50,]  258      10          1 1.1946          1    24          0
#>  [51,]  904      10          1 1.1699          1    41          1
#>  [52,] 4584      10          1 1.3040          1    60          1
#>  [53,] 4596      10          1 1.0793          1    59          1
```

```
#>  [54,] 4300      11          1 1.1088          1    38          1
#>  [55,] 4346      11          1 1.2348          1    72          1
#>  [56,] 4363      11          1 1.2872          1    54          1
#>  [57,] 4521      11          1 1.1473          1   108          1
#>  [58,] 4538      11          1 1.3420          1    35          1
#>  [59,]  950      11          1 1.1330          1    27          1
#>  [60,]  961      11          1 1.0682          1    60          1
#>  [61,] 1380      11          1 1.1898          1    43          1
#>  [62,]  150      11          1 1.2597          1    53          1
#>  [63,] 4928      11          1 1.2441          1    73          1
#>  [64,] 4035      11          1 1.1004          1    42          1
#>  [65,]  861      11          1 1.0638          1    35          1
#>  [66,]  997      11          1 1.2235          1    45          1
#>  [67,]  293      11          1 1.1770          1    39          1
#>  [68,]  978      11          1 1.3289          1    37          1
#>  [69,] 4042      11          1 1.2573          1    73          1
#>  [70,] 4250      11          1 1.1017          1    45          1
#>  [71,] 4406      11          1 1.2152          1    40          1
#>  [72,] 4675      11          1 1.1836          1    65          1
#>  [73,] 4287      12          1 1.3624          1    93          1
#>  [74,] 4582      12          1 1.0301          1    60          1
#>  [75,] 4187      12          1 1.4753          1    20          0
#>  [76,] 1073      12          1 1.1331          1    70          1
#>  [77,] 1224      12          1 1.3889          1    45          1
#>  [78,]  673      12          1 1.2689          1    55          1
#>  [79,]  424      12          1 1.1517          1    27          1
#>  [80,]  626      12          1 1.4735          1    30          1
#>  [81,] 4888      12          1 1.2558          1    38          1
#>  [82,] 4114      12          1 1.4673          1   163          1
#>  [83,] 1393      12          1 1.1601          1    33          1
#>  [84,]  932      13          1 1.1359          1    27          1
#>  [85,] 4302      13          1 1.1480          1    49          1
#>  [86,] 4562      13          1 1.2706          1    42          1
#>  [87,] 4757      13          1 1.3557          1    90          1
#>  [88,]  783      13          1 1.1229          1    26          1
#>  [89,]  511      13          1 1.0102          1    43          1
#>  [90,] 1010      13          1 1.1547          1    58          1
#>  [91,] 4263      13          1 1.2195          1    40          1
#>  [92,] 4324      13          1 1.1134          1    33          1
#>  [93,] 4244      14          1 1.1811          1    15          0
#>  [94,] 4294      14          1 1.3358          1    41          1
#>  [95,] 4462      14          1 1.1092          1    41          1
#>  [96,] 1351      14          1 1.2744          1    69          1
#>  [97,]  552      14          1 1.1587          1    49          1
```

```
#>   [98,] 1033      14         1 1.2251         1    26         1
#>   [99,] 4502      14         1 0.9147         0    83         1
#>  [100,]  344      14         1 1.1226         1    39         1
#>  [101,]  394      14         1 1.0087         1    46         1
#>  [102,] 1423      14         1 1.1161         1    47         1
#>  [103,] 4668      14         1 1.0501         1    47         1
#>  [104,] 1295      14         1 1.2418         1    45         1
#>  [105,]  748      15         1 1.0231         1    16         0
#>  [106,]  314      15         1 1.0952         1    41         1
#>  [107,]  256      15         1 0.9930         1    54         1
#>  [108,] 4689      15         1 1.2386         1    64         1
#>  [109,] 4458      15         1 1.0285         1   170         1
#>  [110,] 4058      15         1 1.1720         1    37         1
#>  [111,] 1315      16         1 1.1160         1    35         1
#>  [112,]  566      16         1 1.1669         1    68         1
#>  [113,]  925      16         1 1.0751         1    14         0
#>  [114,]   57      16         1 1.1151         1    44         1
#>  [115,] 4162      16         1 1.1489         1    38         1
#>  [116,] 4171      16         1 1.0613         1    55         1
#>  [117,]  361      17         1 1.2282         1    88         1
#>  [118,] 4515      17         1 1.1093         1    48         1
#>  [119,]  930      17         1 1.1256         1    48         1
#>  [120,] 4402      17         1 1.1282         1    83         1
#>  [121,] 4712      17         1 1.2897         1    70         1
#>  [122,] 4507      17         1 1.4010         1    64         1
#>  [123,] 4807      18         1 1.1616         1   134         1
#>  [124,]   33      18         1 1.1057         1    41         1
#>  [125,] 4057      18         1 1.1047         1    94         1
#>  [126,] 4189      18         1 1.2607         1    47         1
#>  [127,] 4096      18         1 1.1668         1    36         1
#>  [128,] 4167      18         1 0.9335         1    59         1
#>  [129,] 4595      18         1 0.9387         1    90         1
#>  [130,] 4796      18         1 1.1130         1    81         1
#>  [131,] 4240      19         1 1.1784         1    66         1
#>  [132,] 4079      19         1 1.1004         1    52         1
#>  [133,] 4531      20         1 1.1289         1    28         1
#>  [134,] 1130      20         1 1.1768         1    51         1
#>  [135,]  567      21         1 1.0296         1    21         0
#>  [136,] 4430      21         1 1.0707         1    61         1
#>  [137,] 4918      21         1 1.2463         1    71         1
#>  [138,] 1394      21         1 1.0162         1    30         1
#>  [139,] 4542      22         1 1.1498         1    38         1
#>  [140,] 4243      22         1 1.1096         1    81         1
#>  [141,]  941      23         1 0.9720         1    40         1
```

```
#> [142,] 4943      25         1 1.2673         1    51         1
#> [143,] 4131      25         1 1.0658         1    48         1
#> [144,] 4366      26         1 0.9778         1    74         1
```

References

1. CDC. How to Implement Clinical Decision Support Systems [Internet]. Centers for Disease Control and Prevention. 2021 [cited 12. August 2021]. available on: https://www.cdc.gov/dhdsp/pubs/guides/best-practices/clinical-decision-support.htm

2. Dementia vs. Alzheimer's Disease: What is the Difference? [Internet]. Alzheimer's Disease and Dementia. [cited 12. August 2021]. available on: https://alz.org/alzheimers-dementia/difference-between-dementia-and-alzheimer-s

3. Sturm D, Biesalski A-S, Höffken O, Herausgeber. Neurologische Pathophysiologie: Ursachen und Mechanismen neurologischer Erkrankungen [Internet]. Berlin, Heidelberg: Springer Berlin Heidelberg; 2019 [cited 29. August 2021]. available on: http://link.springer.com/10.1007/978-3-662-56784-5

4. Frost C. Molecular Dynamics Simulation: Applications from Alzheimer's Disease to Photopharmacology. :208

5. Alzheimer-Krankheit. In: Wikipedia [Internet]. 2021 [cited 19. August 2021]. available at: https://de.wikipedia.org/w/index.php?title=Alzheimer-Krankheit&oldid=214843556

6. Alzheimer verstehen [Internet]. [cited 12. August 2021]. available on: https://www.alzheimer.de/

7. ADI - Dementia statistics [Internet]. [cited 12. August 2021]. available on: https://www.alzint.org/about/dementia-facts-figures/dementia-statistics/

8. Early probability-driven prediction of disease progression: a new algorithm applied to Alzheimers disease. :13.

9. 2021 ICD-10-CM Codes G30*: Alzheimer's disease [Internet]. [cited 30. August 2021]. available at: https://www.icd10data.com/ICD10CM/Codes/G00-G99/G30-G32/G30-

10. What is Alzheimer's? [Internet]. Alzheimer's Disease and Dementia. [cited 19. August 2021]. available at: https://alz.org/alzheimers-dementia/what-is-alzheimers

11. Ren J. Automated Detection of Early-Stage Alzheimer's Disease. :94.

12. Sperling RA, Aisen PS, Beckett LA, Bennett DA, Craft S, Fagan AM, u. a. Toward defining the preclinical stages of Alzheimer's disease: Recommendations from the National Institute on Aging-Alzheimer's Association workgroups on diagnostic guidelines for Alzheimer's disease. Alzheimers Dement. May 2011;7(3):280–92.

13. HealthITSecurity. Understanding the Basics of Clinical Decision Support Systems [Internet]. HealthIT-Analytics. 2017 [cited 29. August 2021]. available on: https://healthitanalytics.com/features/understanding-the-basics-of-clinical-decision-support-systems