"PREDICTING STROKE OCCURRENCE WITH ARTIFICIAL NEURAL NETWORKS"

- Input data "healthcare-dataset-stroke-data.csv" yang didapat dari Kaggle ke R Studio

```
> #Input Data
> healthcare.dataset.stroke.data <- read.csv("C:/Users/naura/Downloads/archive (1)/healthcare-dataset-stroke-data.csv")
>   View(healthcare.dataset.stroke.data)
> data = healthcare.dataset.stroke.data
```

	id	gender	age	hypertension	heart_disease	ever_married	work_type	Residence_type	avg_glucose_level	bmi	smoking_status	strok
1	9046	Male	67.00	0	1	Yes	Private	Urban	228.69	36.6	formerly smoked	
2	51676	Female	61.00	0	0	Yes	Self-employed	Rural	202.21	N/A	never smoked	
3	31112	Male	80.00	0	1	Yes	Private	Rural	105.92	32.5	never smoked	
4	60182	Female	49.00	0	0	Yes	Private	Urban	171.23	34.4	smokes	
5	1665	Female	79.00	1	0	Yes	Self-employed	Rural	174.12	24	never smoked	
6	56669	Male	81.00	0	0	Yes	Private	Urban	186.21	29	formerly smoked	
7	53882	Male	74.00	1	1	Yes	Private	Rural	70.09	27.4	never smoked	
8	10434	Female	69.00	0	0	No	Private	Urban	94.39	22.8	never smoked	
9	27419	Female	59.00	0	0	Yes	Private	Rural	76.15	N/A	Unknown	
10	60491	Female	78.00	0	0	Yes	Private	Urban	58.57	24.2	Unknown	
11	12109	Female	81.00	1	0	Yes	Private	Rural	80.43	29.7	never smoked	
12	12095	Female	61.00	0	1	Yes	Govt_job	Rural	120.46	36.8	smokes	
13	12175	Female	54.00	0	0	Yes	Private	Urban	104.51	27.3	smokes	
14	8213	Male	78.00	0	1	Yes	Private	Urban	219.84	N/A	Unknown	
15	5317	Female	79.00	0	1	Yes	Private	Urban	214.09	28.2	never smoked	
16	58202	Female	50.00	1	0	Yes	Self-employed	Rural	167.41	30.9	never smoked	
17	56112	Male	64.00	0	1	Yes	Private	Urban	191.61	37.5	smokes	
18	34120	Male	75.00	1	0	Yes	Private	Urban	221.29	25.8	smokes	
19	27458	Female	60.00	0	0	No	Private	Urban	89.22	37.8	never smoked	
20	25226	Male	57.00	0	1	No	Govt_job	Urban	217.08	N/A	Unknown	
21	70630	Female	71.00	0	0	Yes	Govt_job	Rural	193.94	22.4	smokes	
22	13861	Female	52.00	1	0	Yes	Self-employed	Urban	233.29	48.9	never smoked	
23	68794	Female	79.00	0	0	Yes	Self-employed	Urban	228.70	26.6	never smoked	
24	64778	Male	82.00	0	1	Yes	Private	Rural	208.30	32.5	Unknown	
25	4219	Male	71.00	0	0	Yes	Private	Urban	102.87	27.2	formerly smoked	

Showing 1 to 26 of 5,110 entries, 12 total columns

- Memilih variable-variabel yang akan digunakan dalam analisis data

```
> #Memilih variabel yang akan digunakan
> library(dplyr)
> data <- dplyr::select(data, age, hypertension, heart_disease, Residence_type, avg_glucose_level, smoking_status, strok
e)
> View(data)
```

	age	hypertension	heart_disease	Residence_type	avg_glucose_level	smoking_status	stroke
1	67.00	0	1	Urban	228.69	formerly smoked	1
2	61.00	0	0	Rural	202.21	never smoked	1
3	80.00	0	1	Rural	105.92	never smoked	1
4	49.00	0	0	Urban	171.23	smokes	1
5	79.00	1	0	Rural	174.12	never smoked	1
6	81.00	0	0	Urban	186.21	formerly smoked	1
7	74.00	1	1	Rural	70.09	never smoked	1
8	69.00	0	0	Urban	94.39	never smoked	1
9	59.00	0	0	Rural	76.15	Unknown	1
10	78.00	0	0	Urban	58.57	Unknown	1
11	81.00	1	0	Rural	80.43	never smoked	1
12	61.00	0	1	Rural	120.46	smokes	1
13	54.00	0	0	Urban	104.51	smokes	1
14	78.00	0	1	Urban	219.84	Unknown	1
15	79.00	0	1	Urban	214.09	never smoked	1
16	50.00	1	0	Rural	167.41	never smoked	1
17	64.00	0	1	Urban	191.61	smokes	1
18	75.00	1	0	Urban	221.29	smokes	1
19	60.00	0	0	Urban	89.22	never smoked	1
20	57.00	0	1	Urban	217.08	Unknown	1
21	71.00	0	0	Rural	193.94	smokes	1
22	52.00	1	0	Urban	233.29	never smoked	1
23	79.00	0	0	Urban	228.70	never smoked	1
24	82.00	0	1	Rural	208.30	Unknown	1
25	71.00	0	0	Urban	102.87	formerly smoked	1
26	80.00	0	0	Rural	104.12	never smoked	1

Showing 1 to 27 of 5,110 entries, 7 total columns

- Dikarenakan ada variable-variabel yang kategorikal, maka kita ubah dulu menjadi variabel numerik.

```
> #Mengubah variabel kategorikal menjadi variabel numerik
> data[,4] = sapply(data[,4], switch, "Rural"=0, "Urban"=1)
> data[,6] = sapply(data[,6], switch, "formerly smoked"=0, "never smoked"=1, "smokes"=2, "Unknown"=3)
```

	age	hypertension	heart_disease	Residence_type	avg_glucose_level	smoking_status	stroke
1	67.00	0	1	1	228.69	0	1
2	61.00	0	0	0	202.21	1	1
3	80.00	0	1	0	105.92	1	1
4	49.00	0	0	1	171.23	2	1
5	79.00	1	0	0	174.12	1	1
6	81.00	0	0	1	186.21	0	1
7	74.00	1	1	0	70.09	1	1
8	69.00	0	0	1	94.39	1	1
9	59.00	0	0	0	76.15	3	1
10	78.00	0	0	1	58.57	3	1
11	81.00	1	0	0	80.43	1	1
12	61.00	0	1	0	120.46	2	1
13	54.00	0	0	1	104.51	2	1
14	78.00	0	1	1	219.84	3	1
15	79.00	0	1	1	214.09	1	1
16	50.00	1	0	0	167.41	1	1
17	64.00	0	1	1	191.61	2	1
18	75.00	1	0	1	221.29	2	1
19	60.00	0	0	1	89.22	1	1
20	57.00	0	1	1	217.08	3	1
21	71.00	0	0	0	193.94	2	1
22	52.00	1	0	1	233.29	1	1
23	79.00	0	0	1	228.70	1	1
24	82.00	0	1	0	208.30	3	1
25	71.00	0	0	1	102.87	0	1
26	80.00	0	0	0	104.12	1	1

Showing 1 to 27 of 5,110 entries, 7 total columns

- Dikarenakan skala data dari variabel-variabel prediktornya berbeda-beda, maka kita akan melakukan normalisasi dengan mengubah skala semua data menjadi 0 sampai 1 menggunakan MinMax Scaller.

```
> #Mengubah skala data dengan menjadi 0 sampai 1 pada tiap variabel menggunakan MinMax Scaller
> for (i in names(data[,-7])) {
+     data[i] <- (data[i] - min(data[i]))/(max(data[i]) - min(data[i]))
+ }
> View(data)
```

	age	hypertension	heart_disease	Residence_type	avg_glucose_level	smoking_status	stroke
1	0.81689453	0	1	1	0.801264888	0.0000000	1
2	0.74365234	0	0	0	0.679023174	0.3333333	1
3	0.97558594	0	1	0	0.234512049	0.3333333	1
4	0.59716797	0	0	1	0.536007756	0.6666667	1
5	0.96337891	1	0	0	0.549349091	0.3333333	1
6	0.98779297	0	0	1	0.605161112	0.0000000	1
7	0.90234375	1	1	0	0.069107192	0.3333333	1
8	0.84130859	0	0	1	0.181285200	0.3333333	1
9	0.71923828	0	0	0	0.097082449	1.0000000	1
10	0.95117188	0	0	1	0.015926507	1.0000000	1
11	0.98779297	1	0	0	0.116840550	0.3333333	1
12	0.74365234	0	1	0	0.301634198	0.6666667	1
13	0.65820312	0	0	1	0.228002954	0.6666667	1
14	0.95117188	0	1	1	0.760409934	1.0000000	1
15	0.96337891	0	1	1	0.733865756	0.3333333	1
16	0.60937500	1	0	0	0.518373188	0.3333333	1
17	0.78027344	0	1	1	0.630089558	0.6666667	1
18	0.91455078	1	0	1	0.767103684	0.6666667	1
19	0.73144531	0	0	1	0.157418521	0.3333333	1
20	0.69482422	0	1	1	0.747668729	1.0000000	1
21	0.86572266	0	0	0	0.640845721	0.6666667	1
22	0.63378906	1	0	1	0.822500231	0.3333333	1
23	0.96337891	0	0	1	0.801311052	0.3333333	1
24	1.00000000	0	1	0	0.707136922	1.0000000	1
25	0.86572266	0	0	1	0.220432093	0.0000000	1
26	0.97558594	0	0	0	0.226202567	0.3333333	1

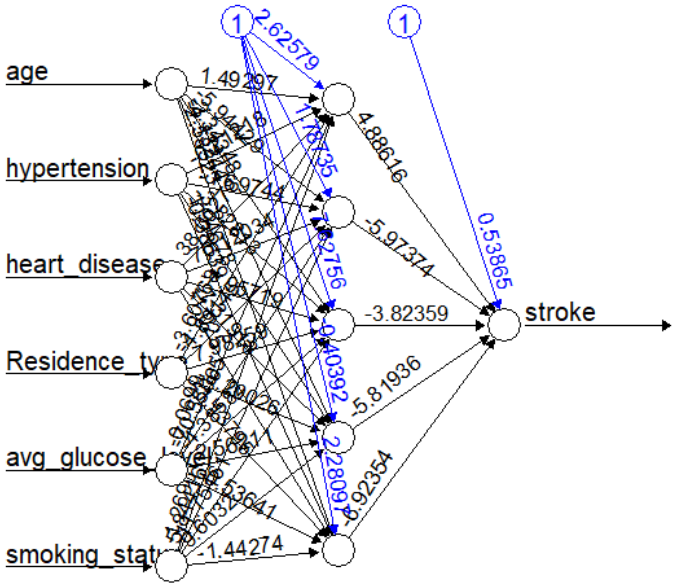
Showing 1 to 27 of 5,110 entries, 7 total columns

- Setelah dataset yang kita pakai sudah siap, maka kita akan membaginya ke data training dan data testing dengan proporsi 75% : 25%

```
> #Membagi data training dan data testing
> set.seed(222)
> index = sample(2, nrow(data), replace = TRUE, prob = c(0.75, 0.25))
> datatraining = data[index==1,]
> datatesting = data[index==2,]
```

- Langsung saja kita akan membuat model NN yang pertama, dengan 5 hidden layer dan activation function sigmoid.

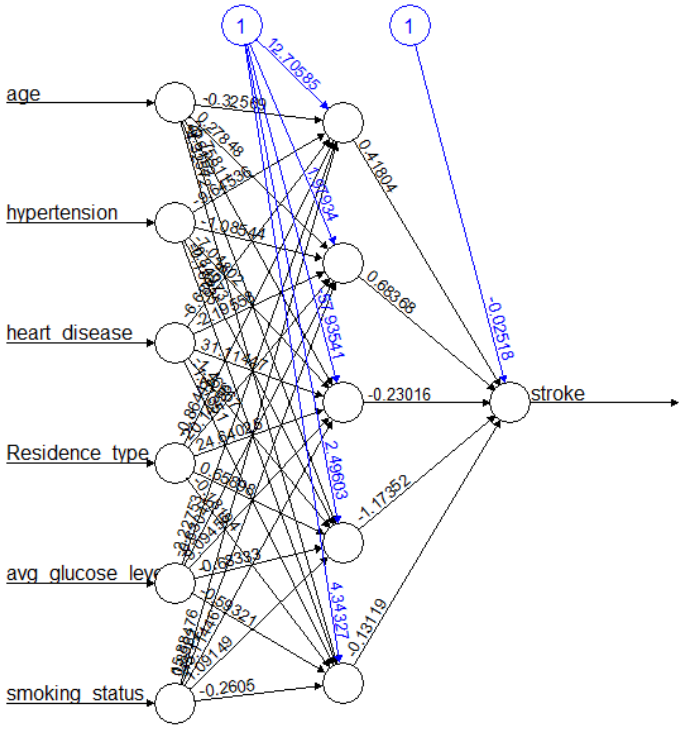
```
> #Membuat model Neural Network dengan act fct = sigmoid
> library(neuralnet)
> set.seed(333)
> NNsigmoid = neuralnet(stroke~., data = datatraining, hidden = 5, act.fct = 'logistic', linear.output = FALSE)
> plot(NNsigmoid)
```



Error: 78.871853 Steps: 8614

- Membuat model NN yang kedua, dengan 5 hidden layer dan activation function tanh.

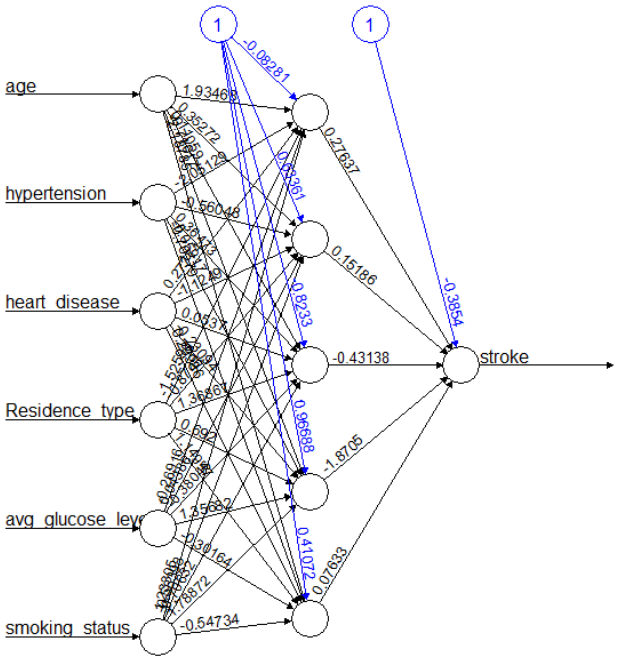
```
> #Membuat Model Neural Network dengan act fct = tanh
> set.seed(333)
> NNtanh = neuralnet(stroke~., data = datatraining, hidden = 5, act.fct = 'tanh', linear.output = FALSE)
> plot(NNtanh)
```



Error: 79.30519 Steps: 61105

- Membuat model NN yang ketiga, dengan 5 hidden layer dan activation function ReLu.

```
> #Membuat Model Neural Network dengan act fct = ReLu
> library(sigmoid)
> set.seed(333)
> NNrelu = neuralnet(stroke~., data = datatraining, hidden = 5, act.fct = relu, linear.output = FALSE)
> plot(NNrelu)
```



- Melakukan prediksi menggunakan model NN yang pertama.

```
> #Melakukan prediksi menggunakan act fct = sigmoid
> output1 = compute(NNsigmoid, datatesting[,-1])
> head(output1$net.result)
           [,1]
1  0.1015987782
5  0.2081488148
6  0.0550352538
15 0.0504484581
21 0.0005722997
26 0.0015758048
> head(datatraining[1,])
        age hypertension heart_disease Residence_type avg_glucose_level smoking_status stroke
2 0.7436523            0             0              0         0.6790232      0.3333333      1
> results1 <- data.frame(DataAsli=datatesting$stroke, Prediksi=output1$net.result)
> results1
```

-Melakukan prediksi menggunakan model NN yang kedua.

```
> #Melakukan prediksi menggunakan act fct = tanh
> output2 = compute(NNtanh, datatesting[,-1])
> head(output2$net.result)
          [,1]
1  0.069553651
5  0.199893276
6  0.020546858
15 0.114609942
21 0.004118941
26 0.004213283
> head(datatraining[1,])
        age hypertension heart_disease Residence_type avg_glucose_level smoking_status stroke
2 0.7436523            0             0              0         0.6790232      0.3333333      1
> results2 = data.frame(DataAsli=datatesting$stroke, Prediksi =output2$net.result)
```

- Melakukan prediksi menggunakan model NN yang ketiga.

```
> #Melakukan prediksi menggunakan act fct = relu
> output3 = compute(NNrelu, datatesting[,-1])
> head(output3$net.result)
   [,1]
1     0
5     0
6     0
15    0
21    0
26    0
> head(datatraining[1,])
        age hypertension heart_disease Residence_type avg_glucose_level smoking_status stroke
2 0.7436523            0             0              0         0.6790232      0.3333333      1
> results3 = data.frame(DataAsli=datatesting$stroke, Prediksi=output3$net.result)
```

- Melihat akurasi prediksi model pertama dengan data aktual menggunakan confusion matrix.

```
> #Menampilkan data asli
> actual1 <- round(datatesting$stroke, digits = 0)
> #Menampilkan data prediksi
> prediction1 <- round(output1$net.result, digits = 0)
> #Mengidentifikasi data asli dan prediksi berdasarkan model ANN menggunakan confusion matrix
> mtab = table(actual1, prediction1)
> library(caret)
> confusionMatrix(mtab)
Confusion Matrix and Statistics

       prediction1
actual1    0    1
      0 1191   11
      1   59    0

               Accuracy : 0.9445
                 95% CI : (0.9304, 0.9565)
    No Information Rate : 0.9913
    P-Value [Acc > NIR] : 1

                  Kappa : -0.0149

 Mcnemar's Test P-Value : 1.937e-08

            Sensitivity : 0.9528
            Specificity : 0.0000
         Pos Pred Value : 0.9908
         Neg Pred Value : 0.0000
             Prevalence : 0.9913
         Detection Rate : 0.9445
   Detection Prevalence : 0.9532
      Balanced Accuracy : 0.4764

       'Positive' Class : 0
```

-Pasien yang benar diprediksi tidak mengalami stroke (0) sebanyak 1191 orang dan prediksi yang salah sebanyak 11 orang.

-Kemudian pasien yang benar diprediksi mengalami stroke (1) sebanyak 0 orang atau tidak ada pasien yang mengalami stroke, dan prediksi salah sebesar 59 orang.

-Tingkat akurasi yang diperoleh dari hasil prediksi data testing sebesar 0.9445 atau 94.45%.

- Melihat akurasi prediksi model kedua dengan data aktual menggunakan confusion matrix.

```
> #Menampilkan data asli
> actual2 = round(datatesting$stroke, digits = 0)
> #Menampilkan data prediksi
> prediction2 = round(output2$net.result, digits = 0)
> #Mengidentifikasi data asli dan prediksi berdasarkan model ANN 2 menggunakan confusion matrix
> mtab2 = table(actual2, prediction2)
> confusionMatrix(mtab2)
```

-Pasien yang benar diprediksi tidak mengalami stroke (0) sebanyak 1051 orang dan prediksi yang salah sebanyak 151 orang.

-Kemudian pasien yang benar diprediksi mengalami stroke (1) sebanyak 59 orang atau tidak ada pasien yang mengalami stroke, dan prediksi salah sebesar 0 orang.

- Melihat akurasi prediksi model ketiga dengan data aktual menggunakan confusion matrix.

```
> #Menampilkan data asli
> actual3 = round(datatesting$stroke, digits = 0)
> #Menampilkan data prediksi
> prediction3 = round(output3$net.result, digits = 0)
> #Mengidentifikasi data asli dan prediksi berdasarkan model ANN 3 menggunakan confusion matrix
> mtab3 = table(actual3, prediction3)
> mtab3
       prediction3
actual3    0
      0 1202
      1   59
```

- Pasien yang benar diprediksi tidak mengalami stroke (0) sebanyak 1201 orang dan prediksi yang salah sebanyak 0 orang.

- Kemudian pasien yang benar diprediksi mengalami stroke (1) sebanyak 59 orang atau tidak ada pasien yang mengalami stroke, dan prediksi salah sebesar 0 orang.

KESIMPULAN

Model yang paling sesuai dari 3 model dengan activation function yang berbeda adalah model 1, yang menggunakan act function sigmoid. Model ini mempunyai tingkat akurasi yang besar dari hasil prediksi model ke data testing.