Yapay Sinir Ağları ile Sınıflandırma Problemi

ELİF EKMEKCİ

2023-02-07

ÖRNEK 1: Yapay Sinir Ağları ile Sınıflandırma Probleminin Çözülmesi

Bu örnekte yapay sinir ağları çalışırken kaggle sitesinden yararlandığımız "Predict Diabetes" veri seti üzerine çalışmalar gerçekleştirilmiştir.

VERİ SETİ AÇIKLAMASI;

Bu veri seti aslen Ulusal Diyabet ve Sindirim ve Böbrek Hastalıkları Enstitüsü'nden alınmıştır.

Veri setinin amacı, bir hastanın diyabet hastası olup olmadığını veri kümesinde yer alan belirli tanılama ölçümlerine dayanarak tanısal olarak tahmin etmektir.

Yaygın olarak diyabet olarak adlandırılan diabetes mellitus (DM), uzun bir süre boyunca yüksek kan şekeri seviyelerinin olduğu bir grup metabolik bozukluktur. Tip 1 diyabet, pankreasın yeterli insülin üretememesinden kaynaklanır. Tip 2 diyabet, hücrelerin insüline düzgün yanıt vermediği bir durum olan insülin direnci ile başlar. 2015 itibariyle, dünya çapında tahmini 415 milyon insanda diyabet vardı ve vakaların yaklaşık % 90'ını tip 2 diyabet oluşturuyordu. Bu, yetişkin nüfusun% 8,3'ünü temsil etmektedir.

(Kaynak: https://https://www.kaggle.com/datasets/whenamancodes/predict-diabities)

DEĞİŞKENLER:

Değişkenler	Açıklaması
Pregnancies	Gebelik sayısını ifade eder
Glucose	Kandaki glikoz seviyesini ifade eder
BloodPressure	Kan basıncı ölçümünü ifade eder
SkinThickness	Cildin kalınlığını ifade eder
Insulin	Kandaki insülin seviyesini ifade eder
BMI	Vücut kitle indeksini ifade eder
DiabetesPedigreeFunction	Diyabet yüzdesini ifade eder
Age	Yaşını ifade eder
Outcome	Nihai sonucu ifade etmek için 1 Evet ve 0 Hayır

1.ADIM: Kullanılan Paketlerin Yüklenmesi ve Aktifleştirilmesi

KULLANILAN PAKETLER

```
library(readr)
library(dplyr)
library(corrr)
library(ggcorrplot)
library(PerformanceAnalytics)
library(GGally)
library(tidyr)
library(ggplot2)
library(neuralnet)
```

2.ADIM: Veri Yükleme ve Düzenleme

- Öncelikle verimizi yükleyelim ve diabetes_data isimi ile tanımlayalım:

```
set.seed(121519016) # Rastgele sayı üretme fonksiyonudur.
#Rastgele seçimleri başlatmak için kullanır.
diabetes_data <- read_csv("/Users/elif/Desktop/diabetes.csv")
head(diabetes_data)
```

```
## # A tibble: 6 x 9
##   Pregnancies Glucose BloodPressure SkinThickness Insulin   BMI DiabetesPedigre~
##         <dbl>   <dbl>         <dbl>         <dbl>   <dbl> <dbl>            <dbl>
## 1           6     148            72            35       0  33.6            0.627
## 2           1      85            66            29       0  26.6            0.351
## 3           8     183            64             0       0  23.3            0.672
## 4           1      89            66            23      94  28.1            0.167
## 5           0     137            40            35     168  43.1            2.29
## 6           5     116            74             0       0  25.6            0.201
## # ... with 2 more variables: Age <dbl>, Outcome <dbl>
```

3.ADIM: Veri Keşfi

- Veri türlerini glimpse() ile kontrol edelim:

```
glimpse(diabetes_data)
```

```
## Rows: 768
## Columns: 9
## $ Pregnancies              <dbl> 6, 1, 8, 1, 0, 5, 3, 10, 2, 8, 4, 10, 10, 1, ~
## $ Glucose                  <dbl> 148, 85, 183, 89, 137, 116, 78, 115, 197, 125~
## $ BloodPressure            <dbl> 72, 66, 64, 66, 40, 74, 50, 0, 70, 96, 92, 74~
## $ SkinThickness            <dbl> 35, 29, 0, 23, 35, 0, 32, 0, 45, 0, 0, 0, 0, ~
## $ Insulin                  <dbl> 0, 0, 0, 94, 168, 0, 88, 0, 543, 0, 0, 0, 0, ~
## $ BMI                      <dbl> 33.6, 26.6, 23.3, 28.1, 43.1, 25.6, 31.0, 35.~
## $ DiabetesPedigreeFunction <dbl> 0.627, 0.351, 0.672, 0.167, 2.288, 0.201, 0.2~
## $ Age                      <dbl> 50, 31, 32, 21, 33, 30, 26, 29, 53, 54, 30, 3~
## $ Outcome                  <dbl> 1, 0, 1, 0, 1, 0, 1, 0, 1, 1, 0, 1, 0, 1, 1, ~
```

- Verimizin özet istatistiklerini inceleyelim:

```
summary(diabetes_data)
```

```
##   Pregnancies        Glucose      BloodPressure    SkinThickness
##  Min.   : 0.000   Min.   :  0.0   Min.   :  0.00   Min.   : 0.00
##  1st Qu.: 1.000   1st Qu.: 99.0   1st Qu.: 62.00   1st Qu.: 0.00
##  Median : 3.000   Median :117.0   Median : 72.00   Median :23.00
##  Mean   : 3.845   Mean   :120.9   Mean   : 69.11   Mean   :20.54
##  3rd Qu.: 6.000   3rd Qu.:140.2   3rd Qu.: 80.00   3rd Qu.:32.00
##  Max.   :17.000   Max.   :199.0   Max.   :122.00   Max.   :99.00
##     Insulin           BMI        DiabetesPedigreeFunction      Age
##  Min.   :  0.0   Min.   : 0.00   Min.   :0.0780           Min.   :21.00
##  1st Qu.:  0.0   1st Qu.:27.30   1st Qu.:0.2437           1st Qu.:24.00
##  Median : 30.5   Median :32.00   Median :0.3725           Median :29.00
##  Mean   : 79.8   Mean   :31.99   Mean   :0.4719           Mean   :33.24
##  3rd Qu.:127.2   3rd Qu.:36.60   3rd Qu.:0.6262           3rd Qu.:41.00
##  Max.   :846.0   Max.   :67.10   Max.   :2.4200           Max.   :81.00
##     Outcome
##  Min.   :0.000
##  1st Qu.:0.000
##  Median :0.000
##  Mean   :0.349
##  3rd Qu.:1.000
##  Max.   :1.000
```

Age değişkenini inceleyelim:

- Minimum değerin 21.00 ve maksimum değerin ise 81.00 olduğunu görüyoruz.
- İlk çeyreği (1st Qu.) 24.00'dır.Bu da tüm kayıtların %25'inin Age değerinin 24.00'ın altında olduğunu gösterir.
- Benzer şekilde üçüncü çeyrekte (3rd Qu.) 41.00 değeri tüm kayıtların %75'inin Age değerinin 41.00 'nin altında olduğunu gösterir.
- Age değerinin ortalaması ise bize aritmetik ortalamayı gösterir ve 33.24 olarak hesaplandığı görülür.

Verimizde missing gozlemler var mi yok mu apply komutuyla kontrol etmeliyiz;

```
apply(diabetes_data,2,function(x) sum(is.na(x)))
```

```
##              Pregnancies                  Glucose            BloodPressure
##                        0                        0                        0
##            SkinThickness                  Insulin                      BMI
##                        0                        0                        0
## DiabetesPedigreeFunction                      Age                  Outcome
##                        0                        0                        0
```

Eksik gözlemlere bakıldığında eksik veri görülmemiştir. Bu işlemi yaparken apply fonksinoyu kullanılmıştır.(Her değişkenin altında yazan değer,o değişkende kaç eksik gözlem olduğunu gösterir.)

- Korelasyon değerleri araştıralım:

Öncelikle tahmin edilecek olan hedef değişkenin açıklayıcı değişkenlerle olan ilişkisine bakalım:

```
diabetes_data %>% correlate() %>% focus(Outcome)
```

```
## # A tibble: 8 x 2
##   term                     Outcome
##   <chr>                      <dbl>
## 1 Pregnancies               0.222
## 2 Glucose                   0.467
## 3 BloodPressure             0.0651
## 4 SkinThickness             0.0748
## 5 Insulin                   0.131
## 6 BMI                       0.293
## 7 DiabetesPedigreeFunction  0.174
## 8 Age                       0.238
```

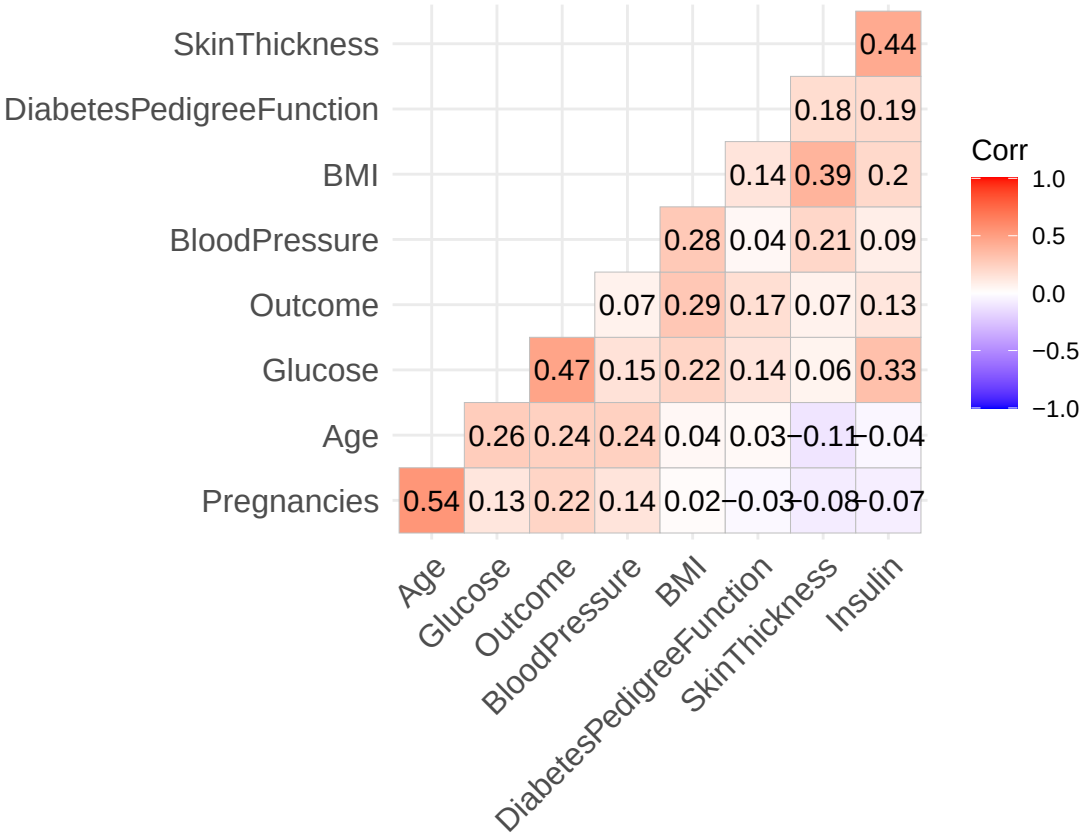
Korelasyon değerleri incelendiğinde:

- Pregnancies,Glucose,BMI ve Age değişkenleri ile hedef değişken Outcome arasında pozitif bir korelasyon vardır.
- BloodPressure,SkinThickness,Insulin ve DiabetesPedigreeFunction değişkenleri ile hedef değişken Outcome arasında düşük bir pozitif korelasyon vardır.
- Değişkenler ile hedef değişken Outcome arasında negatif bir korelasyon yoktur.

Değişkenler arasındaki korelasyonu görsel olarak incelemek istersek:

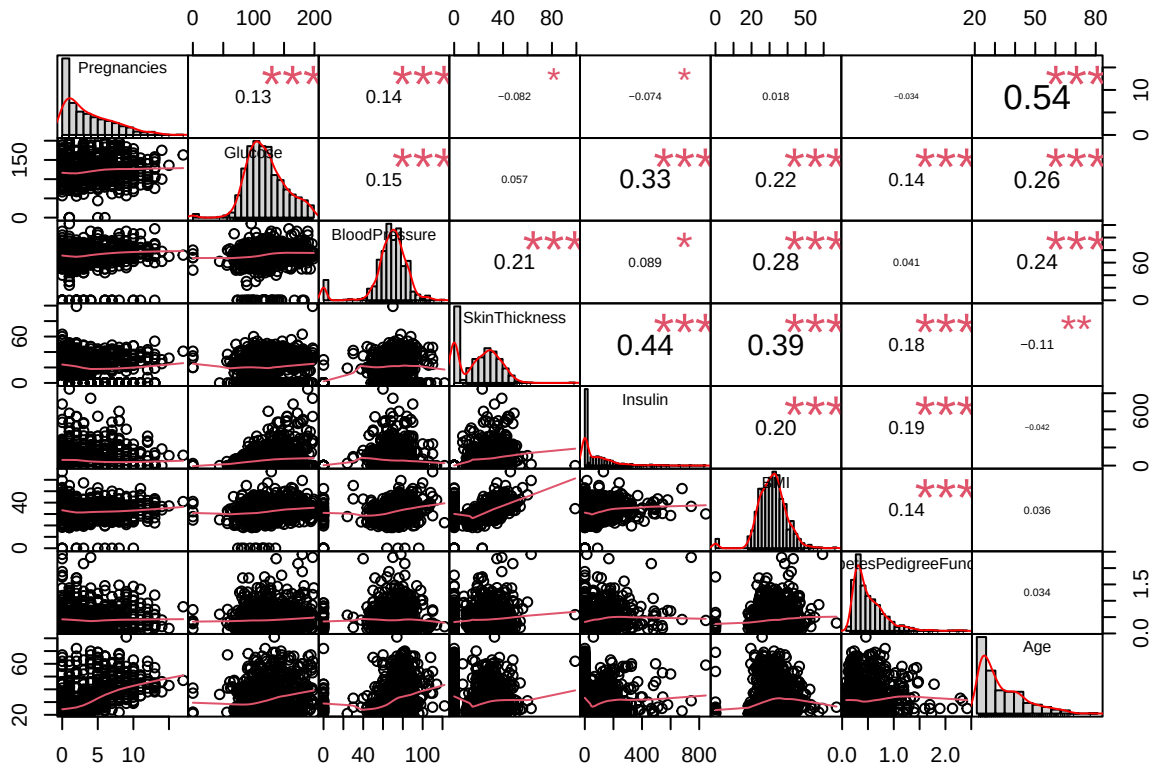
```
diabet_cor <- cor(diabetes_data, use="complete.obs")

ggcorrplot(diabet_cor,
           hc.order = TRUE,
           type = "lower",
           lab = TRUE)
```



- İkinci olarak, açıklayıcı değişkenler arasındaki doğrusal ilişkinin ölçütleri olan ikili korelasyon değerlerini gözden geçirelim:

```
chart.Correlation(diabetes_data[,-9],histogram = TRUE,pch = 19)
```



Grafiği incelediğimizde,

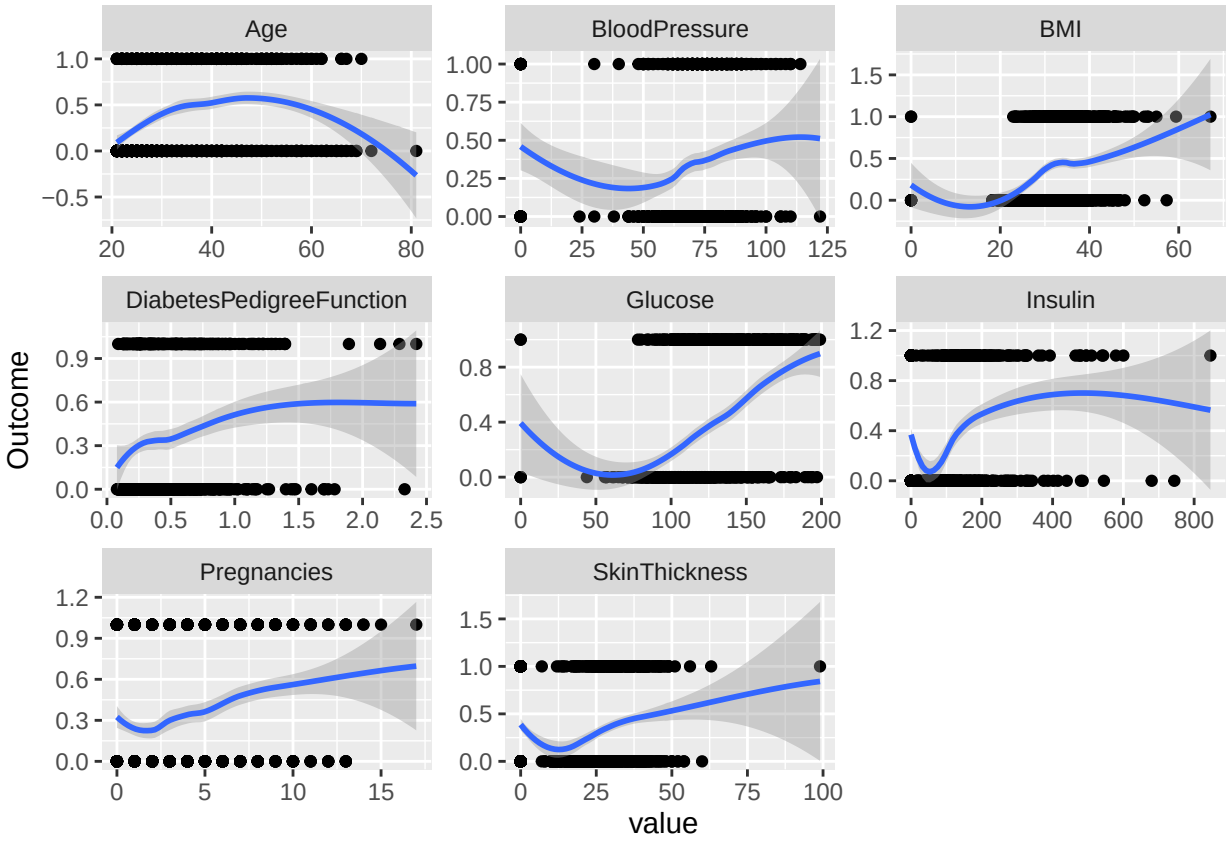
- En yüksek pozitif korelasyonlar:
  - 0.54 ile Pregnancies ve Age değişkenleri arasında
  - 0.44 ile SkinThickness ve Insulin değişkenleri arasında
  - 0.39 ile SkinThickness ve BMI değişkenleri arasında
- Negatif korelasyonlar:
  - -0.11 ile SkinThickness ve Age değişkenleri arasında
  - -0.082 ile Pregnancies ve SkinThickness değişkenleri arasındadır.

Değişkenlerin dağılımlarını inceleyelim

Hedef değişken Outcome'ın açıklayıcı değişkenlere karşı dağılım grafiklerini oluşturalım ve yorumlayalım:

```
diabetes_data %>%
gather(-Outcome, key = "var", value = "value") %>%
filter(var != "chas") %>%
ggplot(aes(x = value, y = Outcome)) +
```

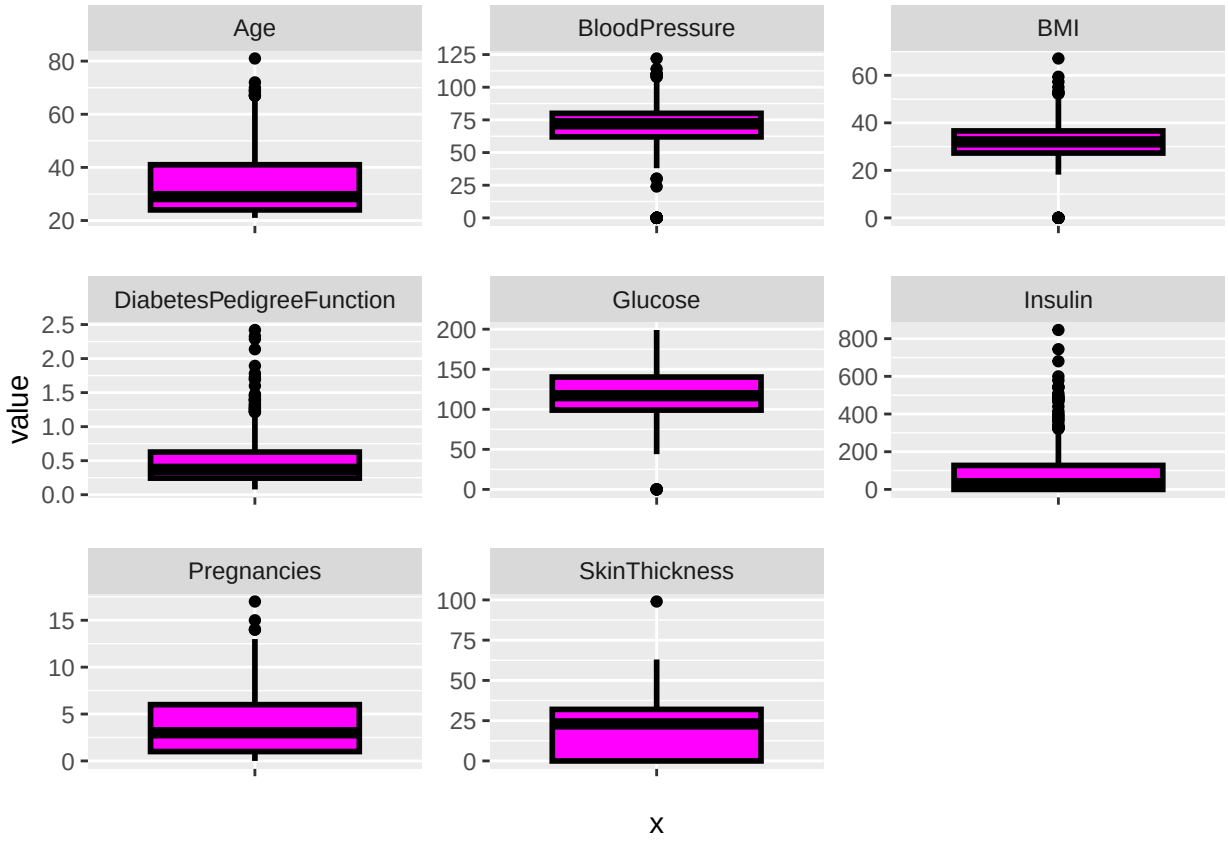
```
geom_point() +
stat_smooth() +
facet_wrap(~ var, scales = "free") +
theme_get()
```



Hedef değişken Outocome ile açıklayıcı değişkenler arasında doğrusal olmayan bir yapı olduğu görülmektedir.

- Açıklayıcı değişkenler için boxplot çizdirelim ve yorumlayalım:

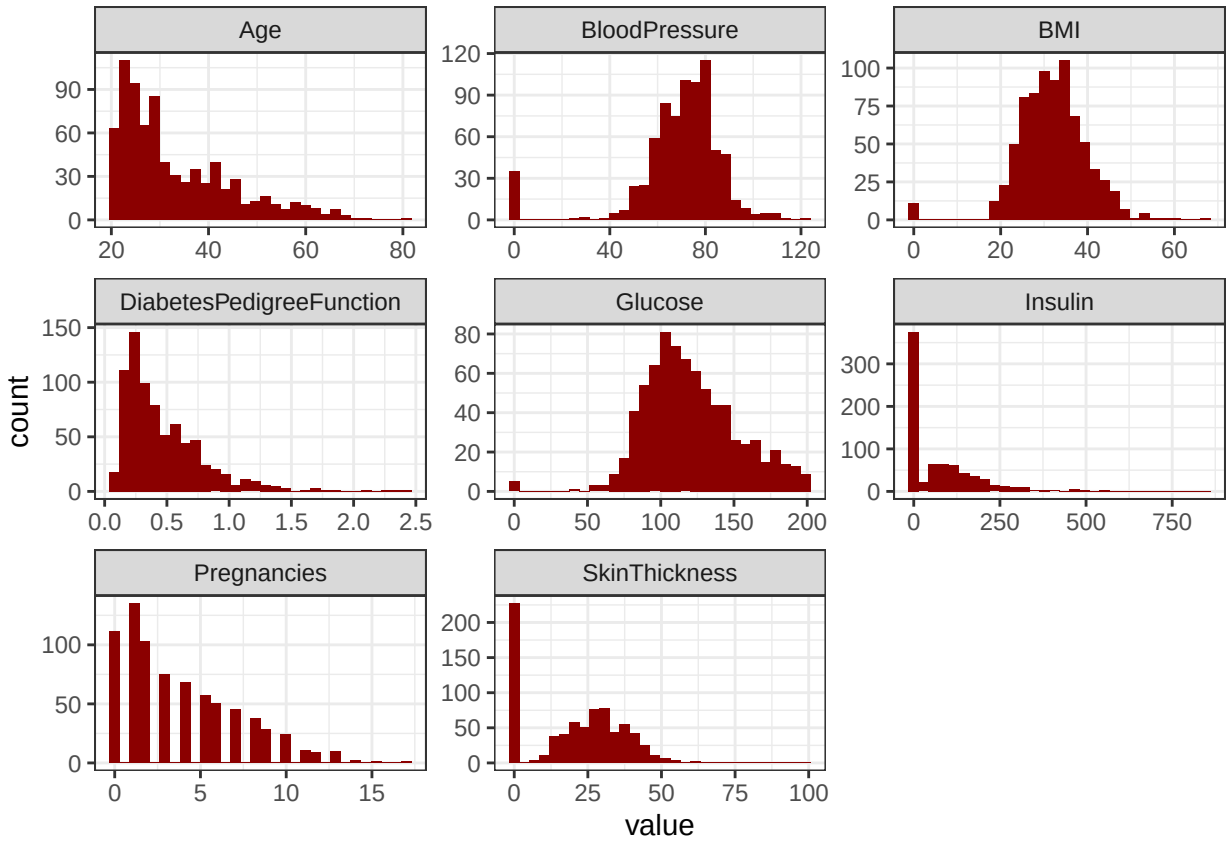
```
diabetes_data %>%
gather(-Outcome, key = "var", value = "value") %>%
filter(var != "chas") %>%
ggplot(aes(x = '',y = value)) +
geom_boxplot(fill = '#FF00FF', color="black", size=1) +
facet_wrap(~ var, scales = "free") +
theme_get()
```



Kutu çizimleri incelendiğinde, verilerde bazı aykırı değerlerin olduğu görülür.

- Son olarak açıklayıcı değişkenlerin histogramını çizdirirsek:

```
diabetes_data %>%
gather(-Outcome, key = "var", value = "value") %>%
filter(var != "chas") %>%
ggplot(aes(x = value)) +
geom_histogram(fill="darkred") +
facet_wrap(~ var, scales = "free") +
theme_bw()
```



- Pregnancies ve SkinThickness değişkenlerinin aralarında hiçbir veri olmadan ayrılmış iki farklı tepe noktası vardır ve bu durum karışım dağılımının (mixture distribution) varlığına işaret eder.
- Ayrıca burada çoğu değişkenin dağılımlarının çarpık olduğu gözlemlenmiştir.

4.ADIM: Veriyi Hazırlama

```
ddata <- read_csv("/Users/elif/Desktop/diabetes.csv") %>%
  na.omit() %>%
  mutate(Outcome = ifelse(Outcome == 0,0,1),
         Outcome = factor(Outcome))
```

5.ADIM: Veriyi Normalleştirme

```
scale01 <- function(x){
(x - min(x)) / (max(x) - min(x))
}
ddata_Scaled <- ddata  %>%
mutate(Pregnancies = scale01(Pregnancies),
Glucose = scale01(Glucose),
BloodPressure = scale01(BloodPressure),
SkinThickness = scale01(SkinThickness),
Insulin = scale01(Insulin),
```

```
BMI = scale01(BMI),
DiabetesPedigreeFunction = scale01(DiabetesPedigreeFunction),
Age= scale01(Age),
Outcome = as.numeric(Outcome)-1)
head(ddata_Scaled)
```

```
## # A tibble: 6 x 9
##   Pregnancies Glucose BloodPressure SkinThickness Insulin   BMI DiabetesPedigre~
##         <dbl>   <dbl>         <dbl>         <dbl>   <dbl> <dbl>            <dbl>
## 1      0.353   0.744         0.590         0.354   0     0.501           0.234
## 2      0.0588  0.427         0.541         0.293   0     0.396           0.117
## 3      0.471   0.920         0.525         0       0     0.347           0.254
## 4      0.0588  0.447         0.541         0.232   0.111 0.419           0.0380
## 5      0       0.688         0.328         0.354   0.199 0.642           0.944
## 6      0.294   0.583         0.607         0       0     0.382           0.0525
## # ... with 2 more variables: Age <dbl>, Outcome <dbl>
```

Neuralnet paketi ile kullanılan sınıflandırma amaçlı YSA yanıt özelliğinin, bu örnekte Outcome, bir Boolean (yalnızca TRUE ve FALSE değişkenlerini alabilen) özelliği olarak girilmesini gerektirir. Bu özellik bu doğrultuda değiştirilir.

```
ddata_adj <- ddata_Scaled %>%
mutate(Outcome = as.integer(Outcome) - 1,
Outcome = ifelse(Outcome == 1, TRUE, FALSE))
head(ddata_adj)
```

```
## # A tibble: 6 x 9
##   Pregnancies Glucose BloodPressure SkinThickness Insulin   BMI DiabetesPedigre~
##         <dbl>   <dbl>         <dbl>         <dbl>   <dbl> <dbl>            <dbl>
## 1      0.353   0.744         0.590         0.354   0     0.501           0.234
## 2      0.0588  0.427         0.541         0.293   0     0.396           0.117
## 3      0.471   0.920         0.525         0       0     0.347           0.254
## 4      0.0588  0.447         0.541         0.232   0.111 0.419           0.0380
## 5      0       0.688         0.328         0.354   0.199 0.642           0.944
## 6      0.294   0.583         0.607         0       0     0.382           0.0525
## # ... with 2 more variables: Age <dbl>, Outcome <lgl>
```

6.ADIM: Yapay Sinir Ağı Modeli Oluşturma

Tek gizli tabaka ve tek gizli tabaka birim içeren YSA yapısı oluşturulsun. Neuralnet paketi varsayılan (default) olarak başlangıç ağırlıklarını rastgele belirlemektedir. Yeniden üretebilirliği sağlamak adına set.seed() komutu kullanarak başlangıç ağırlıkları sabitlenir. Eklenen üç argüman ise;

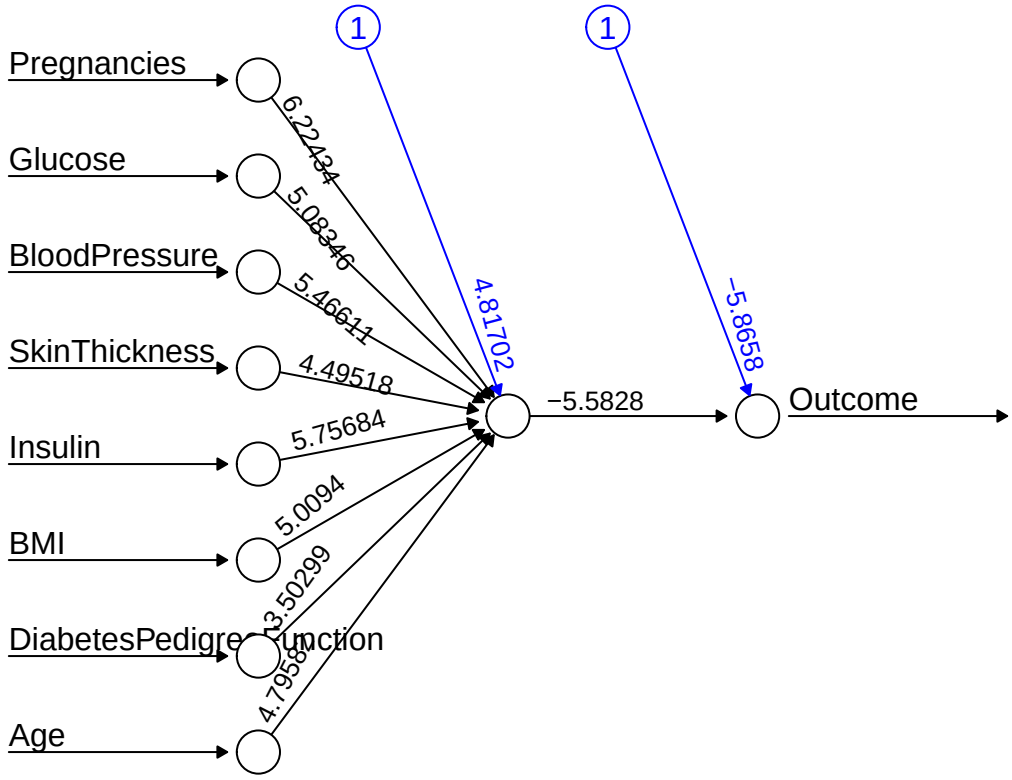
> *linear.output*,problemin bir sınıflandırma problemi olduğunun göstergesidir ve modelin Outcome yanıt değişkeninin sınıfının 1 olması olasılığı şeklinde yorumlanacak çıktıları ürettiği anlamını taşır.

> *err.fct'yi* "ce",bir sınıflama problemi için kullanımı daha uygun ve regresyon amaçlı kullanılan YSA'larda hesaplanan SSE'den farklı olan çapraz entropi hata metriğinin kullanıldığını işaret eder.

TRUE olarak ayarlanan *olabilirlik* argümanı ise,*AIC* ve *BIC* ölçümlerinin hesaplanabilmesine olanak sağlar.

```
set.seed(121519016)
ddata_NN1 <- neuralnet(Outcome ~ Pregnancies + Glucose + BloodPressure
                       + SkinThickness + Insulin + BMI +
                         DiabetesPedigreeFunction + Age,
data = ddata_adj,
linear.output = FALSE,
err.fct = 'ce',
likelihood = TRUE)
```

```
plot(ddata_NN1,rep = "best")
```



Yapay Sinir Ağı için çizdirdiğimiz plota baktığımızda Error: 0.00819 ve Steps: 64 olarak çıkmıştır.

Bu grafikte görülen hata,ddata veri setindeki gözlemlerin her biri için tahmin edilen ve gözlemlenen çıktı arasındaki farkların bir ölçüsü olan çapraz entropi hatasıdır. AIC,BIC ve hata ölçümlerini hesaplayalım:

```
ddata_NN1_Train_Error <- ddata_NN1$result.matrix[1,1]
paste("CE Error: ", round(ddata_NN1_Train_Error, 3))
```

```
## [1] "CE Error:  0.008"
```

```
ddata_NN1_AIC <- ddata_NN1$result.matrix[4,1]
paste("AIC: ", round(ddata_NN1_AIC,3))
```

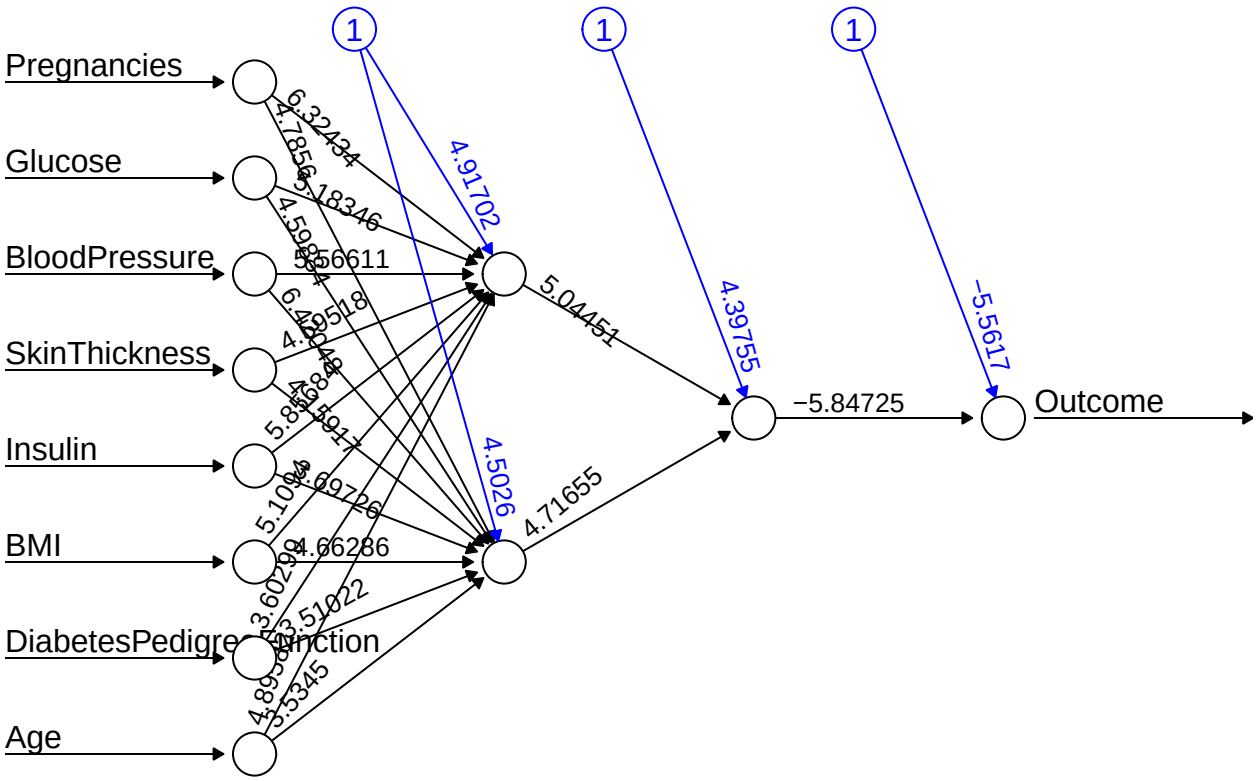
```
## [1] "AIC:  22.016"
```

```
ddata_NN1_BIC <- ddata_NN1$result.matrix[5,1]
paste("BIC: ", round(ddata_NN1_BIC, 3))
```

```
## [1] "BIC:  73.098"
```

SINIFLANDIRMA HİPERPARAMETRELERİ

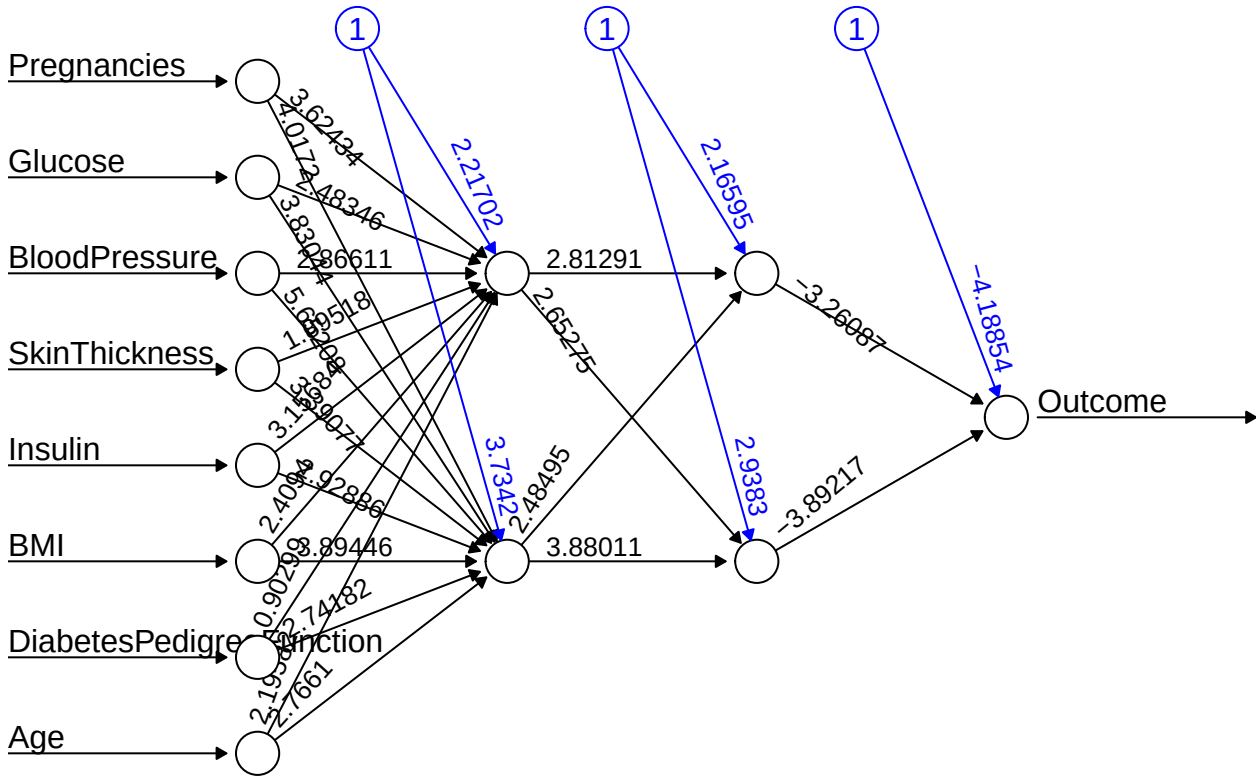
```
set.seed(121519016)
#2-Gizli Katmanlar, Katman-1 2-nöron, Katman-2, 1-nöron
ddata_NN2 <- neuralnet(Outcome ~ Pregnancies + Glucose
                       + BloodPressure + SkinThickness + Insulin + BMI +
                         DiabetesPedigreeFunction + Age,
data = ddata_adj,
linear.output = FALSE,
err.fct = 'ce',
likelihood =
TRUE, hidden = c(2,1))
plot(ddata_NN2, rep = 'best')
```



,

Yapay Sinir Ağı için çizdirdiğimiz plota baktığımızda Error: 0.008521 ve Steps: 53 olarak çıkmıştır.

```
# 2-Gizli Katmanlar, Katman-1 2-nöron, Katman-2, 2-nöron
set.seed(121519016)
ddata_NN3 <- neuralnet(Outcome ~ Pregnancies + Glucose +
                         BloodPressure + SkinThickness +
                         Insulin + BMI + DiabetesPedigreeFunction + Age,
data = ddata_adj,
linear.output = FALSE,
err.fct = 'ce',
likelihood = TRUE,
hidden = c(2,2))
plot(ddata_NN3, rep = 'best')
```



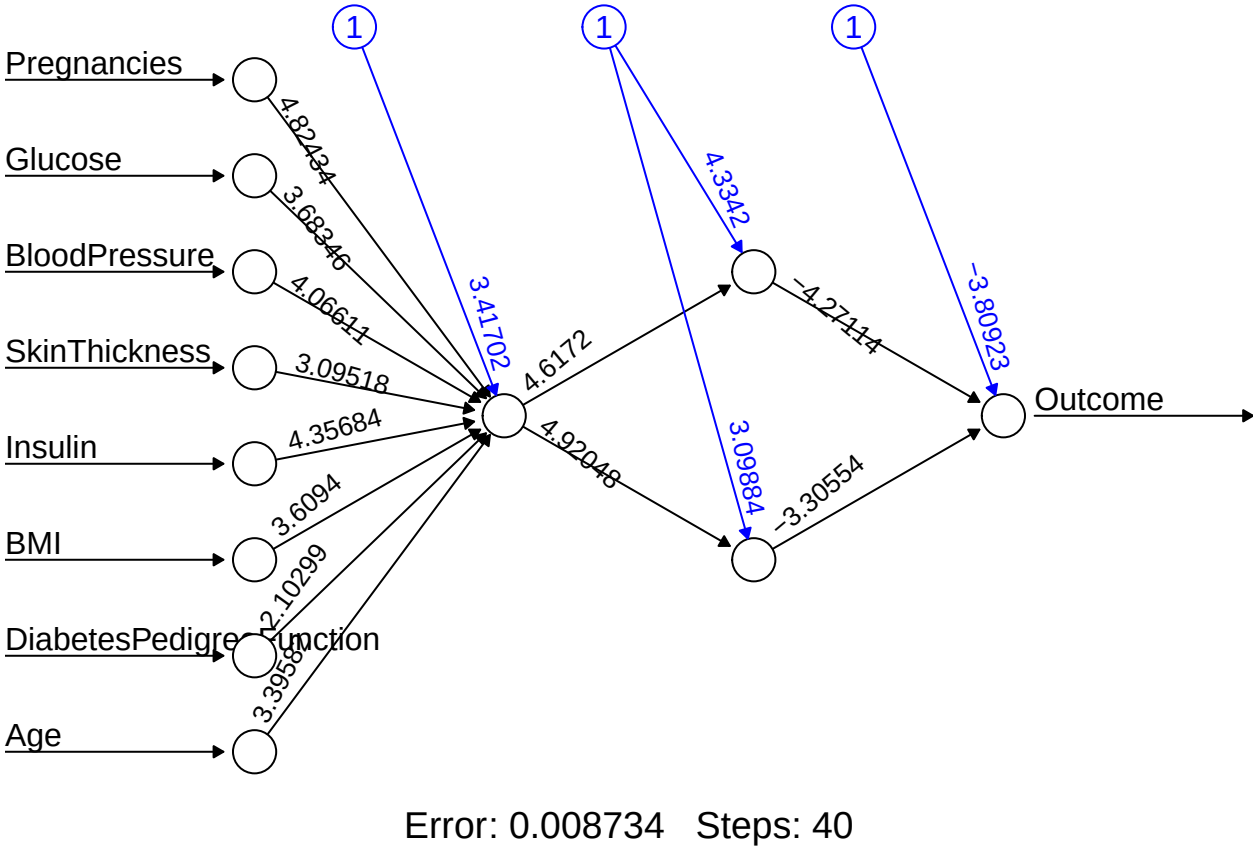
Error: 0.009135 Steps: 34

,

Yapay Sinir Ağı için çizdirdiğimiz plota baktığımızda Error: 0.009135 ve Steps: 34 olarak çıkmıştır.

```
# 2-Gizli Katmanlar, Katman-1 1-nöron, Katman-2, 2-nöron
set.seed(121519016)
ddata_NN4 <- neuralnet(Outcome ~ Pregnancies + Glucose +
                         BloodPressure + SkinThickness +
                         Insulin + BMI + DiabetesPedigreeFunction + Age,
linear.output = FALSE,
data = ddata_adj,
```

```
err.fct = 'ce',
likelihood = TRUE,
hidden = c(1,2))
plot(ddata_NN4, rep = 'best')
```



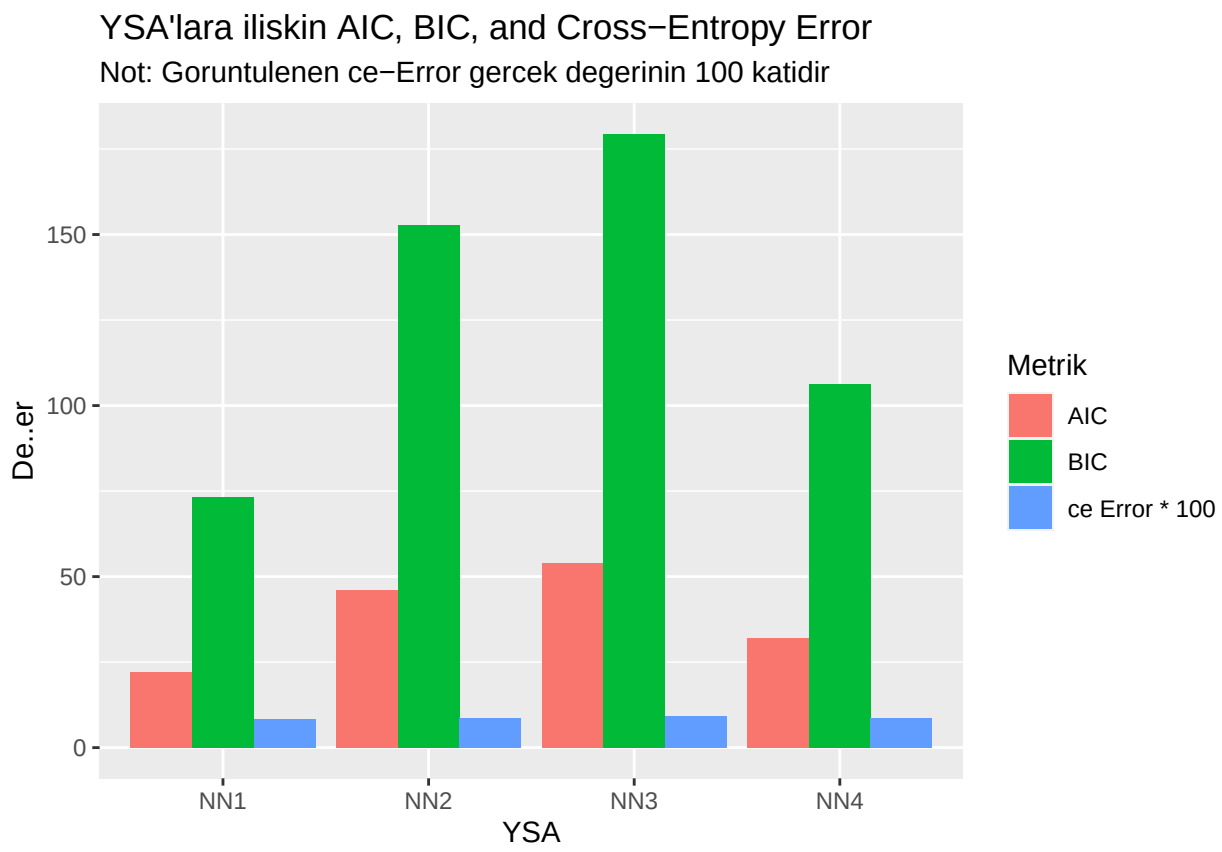
,

Yapay Sinir Ağı için çizdirdiğimiz plota baktığımızda Error: 0.008734 ve Steps: 40 olarak çıkmıştır.

Sonuçları bar grafiği üzerinde gösterelim:

```
Class_NN_ICs <- tibble('YSA' = rep(c("NN1", "NN2", "NN3", "NN4"), each = 3),
'Metrik' = rep(c('AIC', 'BIC', 'ce Error * 100'), length.out = 12),
'Değer' = c(ddata_NN1$result.matrix[4,1], ddata_NN1$result.matrix[5,1],
1000*ddata_NN1$result.matrix[1,1], ddata_NN2$result.matrix[4,1],
ddata_NN2$result.matrix[5,1], 1000*ddata_NN2$result.matrix[1,1],
ddata_NN3$result.matrix[4,1], ddata_NN3$result.matrix[5,1],
1000*ddata_NN3$result.matrix[1,1], ddata_NN4$result.matrix[4,1],
ddata_NN4$result.matrix[5,1], 1000*ddata_NN4$result.matrix[1,1]))
```

```
Class_NN_ICs %>%
ggplot(aes(YSA, Değer, fill = Metrik)) +
geom_col(position = 'dodge') +
ggtitle("YSA'lara iliskin AIC, BIC, and Cross-Entropy Error",
        "Not: Goruntulenen ce-Error gercek degerinin 100 katidir")
```



Grafik üzerinden modelleri kıyaslarsak en büyük AIC ve BIC değerine sahip olan modelimiz NN3 olmuştur.En az AIC ve BIC değerine ise NN1 modeli sahiptir.Dolayısıyla NN1 modeli incelediğimiz dört model arasında en iyi sonucu veren model olacaktır.