Bank Marketing Data Analysis

Umut Aykanat

GİRİŞ

Bu proje, Portekiz'de bir bankanın doğrudan pazarlama kampanyaları sırasında toplanan müşteri verilerini analiz etmeyi amaçlamaktadır. Veri seti, müşterilerin demografik özellikleri (yaş, medeni durum, eğitim düzeyi), finansal durumları (bakiye, kredi geçmişi, konut ve kişisel kredi durumu) ve banka ile iletişim bilgilerini içermektedir.

Projenin temel hedefi, müşterilerin **vadeli mevduat (term deposit)** teklifini kabul edip etmemelerini etkileyen faktörleri incelemektir. Böylece pazarlama kampanyalarının başarısını artırabilecek değişkenler belirlenebilir.

Veri setinde toplam 17 değişken bulunmaktadır. Bunlardan 16'sı açıklayıcı (bağımsız) değişken, biri ise hedef (bağımlı) değişkendir:
- **Bağımlı değişken:** y – Müşteri vadeli mevduat teklifini kabul etti mi? ("yes" veya "no")
- **Bağımsız değişkenler:** yaş, meslek, eğitim durumu, kredi durumu, iletişim tipi, kampanya süresi vb.

Bu analiz kapsamında; - Değişkenlerin yapısı incelenmiş,
- Kategorik değişkenler görselleştirilmiş,
- Temel özet istatistikler sunulmuş,

Sonuç olarak proje, pazarlama stratejilerinin daha veriye dayalı ve hedef odaklı hale gelmesine katkı sağlayabilecek içgörüler sunmaktadır.

```
data <- read.csv("/Users/umutaykanat/Desktop/portfolio/banking data/train.csv", sep = ";", header = TRUE)
head(data)
```

1
2
3
4
5
6

6 rows | 1-1 of 18 columns

Veri Seti Hakkında Betimleyici Bilgiler

```
str(data)
```

```
## 'data.frame':    45211 obs. of  17 variables:
##  $ age      : int  58 44 33 47 33 35 28 42 58 43 ...
##  $ job      : chr  "management" "technician" "entrepreneur" "blue-collar" ...
##  $ marital  : chr  "married" "single" "married" "married" ...
##  $ education: chr  "tertiary" "secondary" "secondary" "unknown" ...
##  $ default  : chr  "no" "no" "no" "no" ...
##  $ balance  : int  2143 29 2 1506 1 231 447 2 121 593 ...
##  $ housing  : chr  "yes" "yes" "yes" "yes" ...
##  $ loan     : chr  "no" "no" "yes" "no" ...
##  $ contact  : chr  "unknown" "unknown" "unknown" "unknown" ...
##  $ day      : int  5 5 5 5 5 5 5 5 5 5 ...
##  $ month    : chr  "may" "may" "may" "may" ...
##  $ duration : int  261 151 76 92 198 139 217 380 50 55 ...
##  $ campaign : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ pdays    : int  -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 ...
##  $ previous : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ poutcome : chr  "unknown" "unknown" "unknown" "unknown" ...
##  $ y        : chr  "no" "no" "no" "no" ...
```

Verimiz 45211 gözlem ve 17 değişkenden oluşuyor. Değişkenleri tanıyalım: **age**: Müşterinin yaşı **job**: İş türü **marital**: Medeni durumu **education**: Eğitim düzeyi **default**: Kredi temerrüt durumu **balance**: Ortalama yıllık bakiye(€) **housing**: Konut kredisi durumu **loan**: Kişisel kredi durumu **contact**: İletişim Türü **day**: Ayın son iletişim günü **month**: Son iletişim ayı **duration**: Son iletişim süresi **campaign**: Kampanya süresince yapılan iletişim sayısı) **pdays**: Önceki kampanyadan sonra geçen gün sayısı **previous**: Bu kampanyadan önce yapılan iletişim sayısı **poutcome**: Önceki pazarlama kampanyasının sonucu **y**: Müşteri vadeli mevduata abone oldu mu?

Veri Tipi Dönüşümü

Kategorik değişkenleri factor olarak belirleyelim

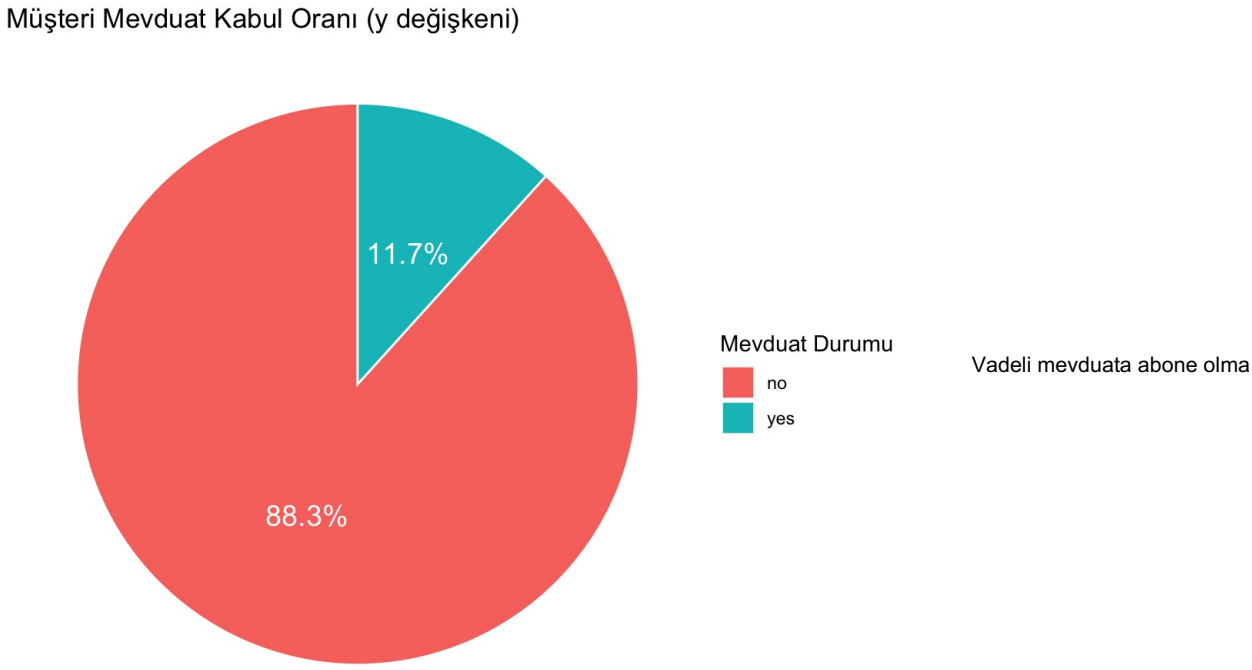
```
# Tüm karakter tipindeki değişkenleri faktöre çevir
data[sapply(data, is.character)] <- lapply(data[sapply(data, is.character)], as.factor)
str(data)
```

```
## 'data.frame':    45211 obs. of  17 variables:
##  $ age      : int  58 44 33 47 33 35 28 42 58 43 ...
##  $ job      : Factor w/ 12 levels "admin.","blue-collar",..: 5 10 3 2 12 5 5 3 6 10 ...
##  $ marital  : Factor w/ 3 levels "divorced","married",..: 2 3 2 2 3 2 3 1 2 3 ...
##  $ education: Factor w/ 4 levels "primary","secondary",..: 3 2 2 4 4 3 3 3 1 2 ...
##  $ default  : Factor w/ 2 levels "no","yes": 1 1 1 1 1 1 1 2 1 1 ...
##  $ balance  : int  2143 29 2 1506 1 231 447 2 121 593 ...
##  $ housing  : Factor w/ 2 levels "no","yes": 2 2 2 2 1 2 2 2 2 2 ...
##  $ loan     : Factor w/ 2 levels "no","yes": 1 1 2 1 1 1 2 1 1 1 ...
##  $ contact  : Factor w/ 3 levels "cellular","telephone",..: 3 3 3 3 3 3 3 3 3 3 ...
##  $ day      : int  5 5 5 5 5 5 5 5 5 5 ...
##  $ month    : Factor w/ 12 levels "apr","aug","dec",..: 9 9 9 9 9 9 9 9 9 9 ...
##  $ duration : int  261 151 76 92 198 139 217 380 50 55 ...
##  $ campaign : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ pdays    : int  -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 ...
##  $ previous : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ poutcome : Factor w/ 4 levels "failure","other",..: 4 4 4 4 4 4 4 4 4 4 ...
##  $ y        : Factor w/ 2 levels "no","yes": 1 1 1 1 1 1 1 1 1 1 ...
```

Keşifçi Veri Analizi ve Görselleştirme

Öncelikle hedef değişkenin dağılımına bakalım

```
library(ggplot2)
# Y değişkeninin oranlarını tablo haline getirelim
y_data <- as.data.frame(table(data$y))
colnames(y_data) <- c("y", "count")
# Pasta grafiği
ggplot(y_data, aes(x = "", y = count, fill = y)) +
  geom_col(width = 1, color = "white") +
  coord_polar(theta = "y") +
  labs(title = "Müşteri Mevduat Kabul Oranı (y değişkeni)",
       fill = "Mevduat Durumu") +
  theme_void() +
  geom_text(aes(label = paste0(round(count / sum(count) * 100, 1), "%")),
            position = position_stack(vjust = 0.5),
            color = "white",
            size = 5)
```



durumunun tüm müşteriler bazında dağılımı %11.7 evet iken %88.3 hayırdır. Hedef değişkenin dağılımında eşitsizlik olduğu göz önünde bulundurulmalıdır.

Her değişkenin alt kategorilerine ait müşteri mevduat kabul oranlarını inceleyelim. Bu oranlar tekil incelendiğinde anlamlıdır. Örneğin kişinin yalnızca medeni durumuna göre analiz yapılırsa mevduata abone olma durumu %6.1 iken olmama durumu %54.1'dir.

Değişkenlerin kategorilerinin kendi aralarında müşteri mevduat kabul oranlarını inceleyelim:

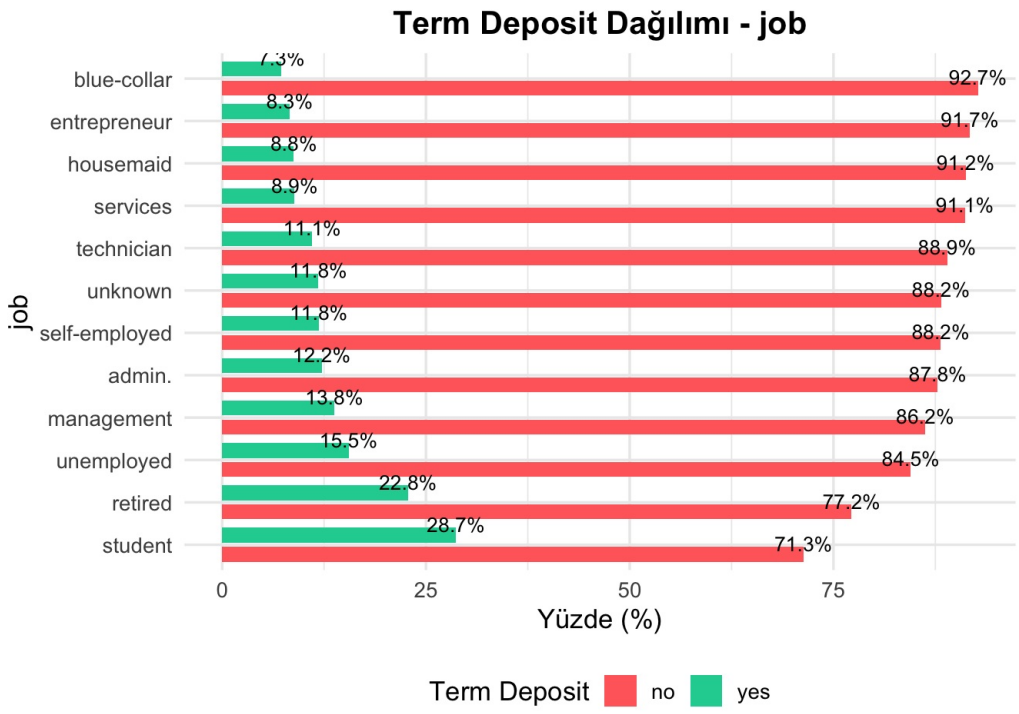
```
library(dplyr)
```

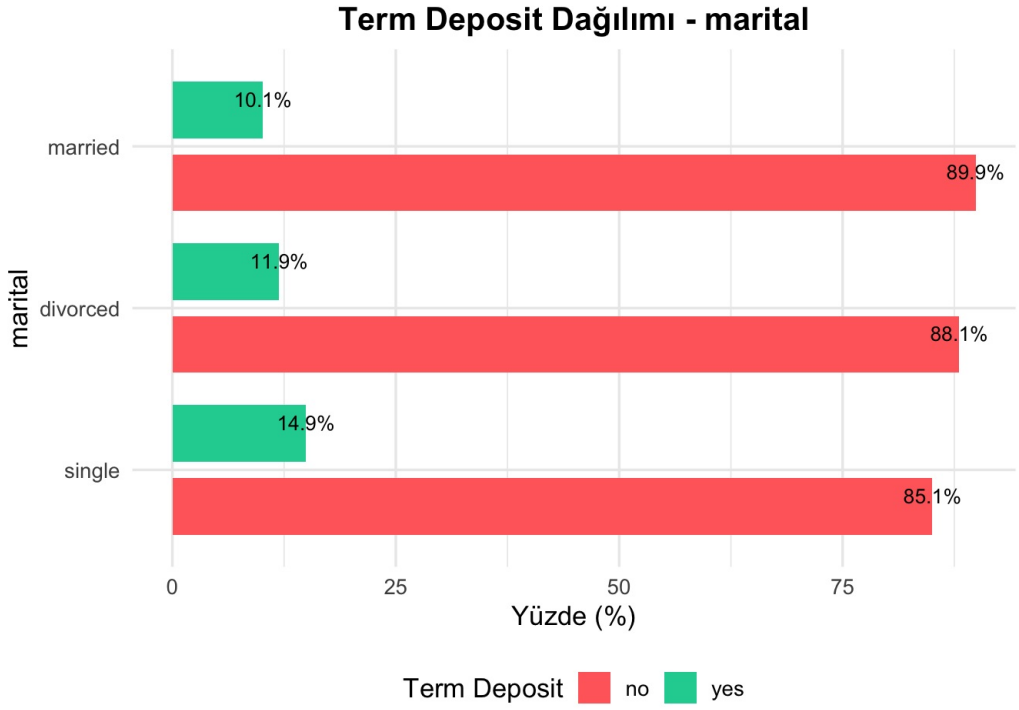
```
##
## Attaching package: 'dplyr'
```

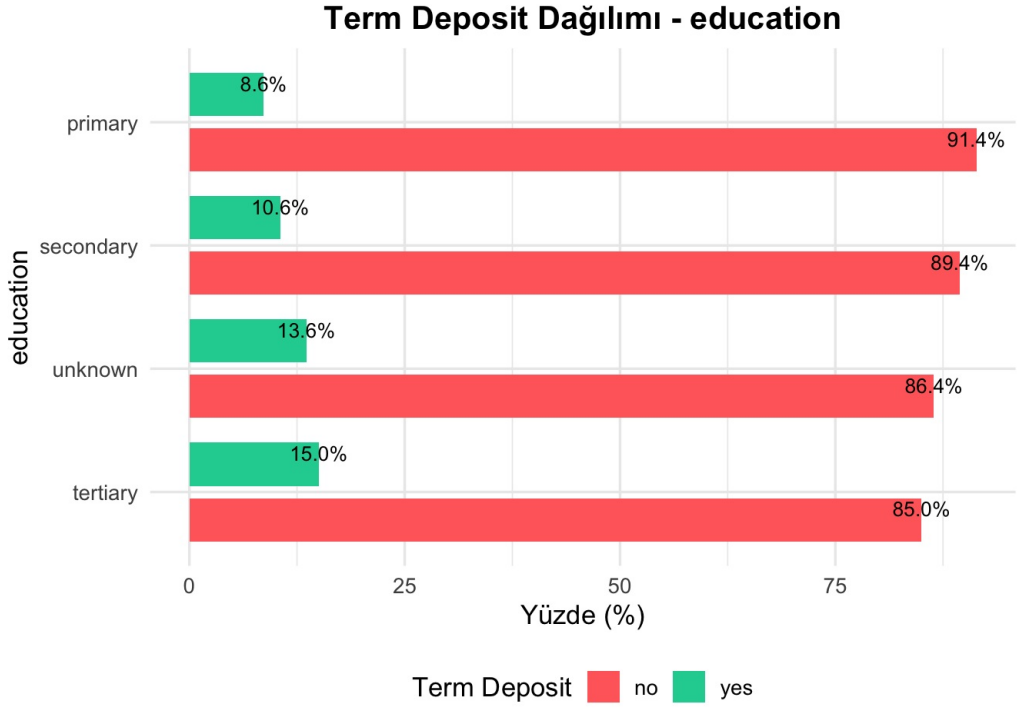
```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

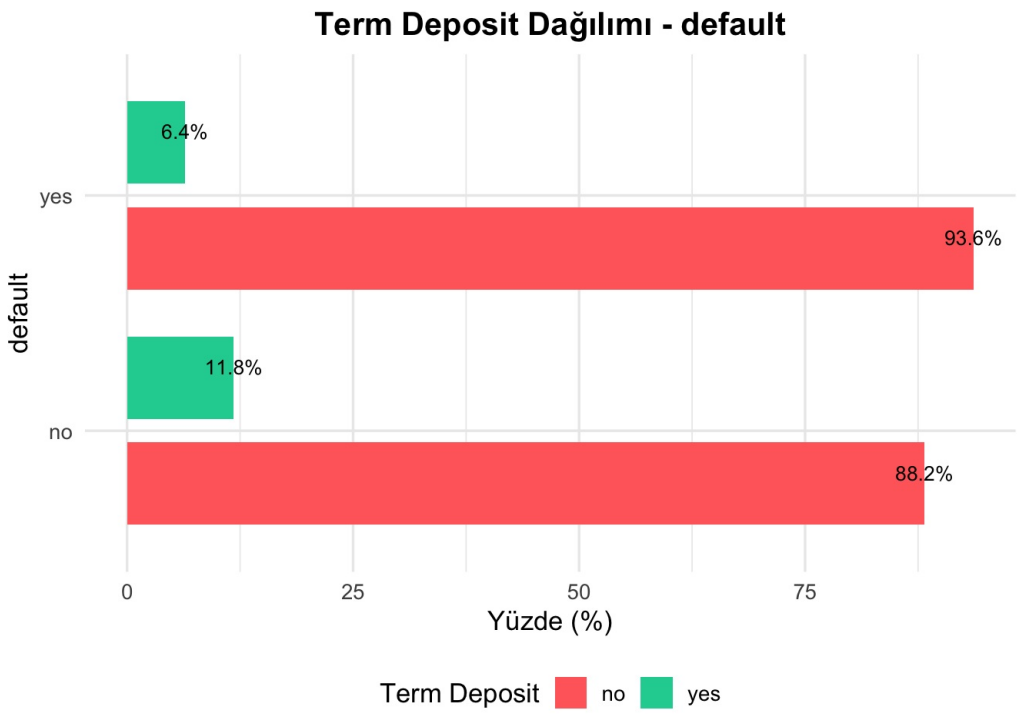
```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

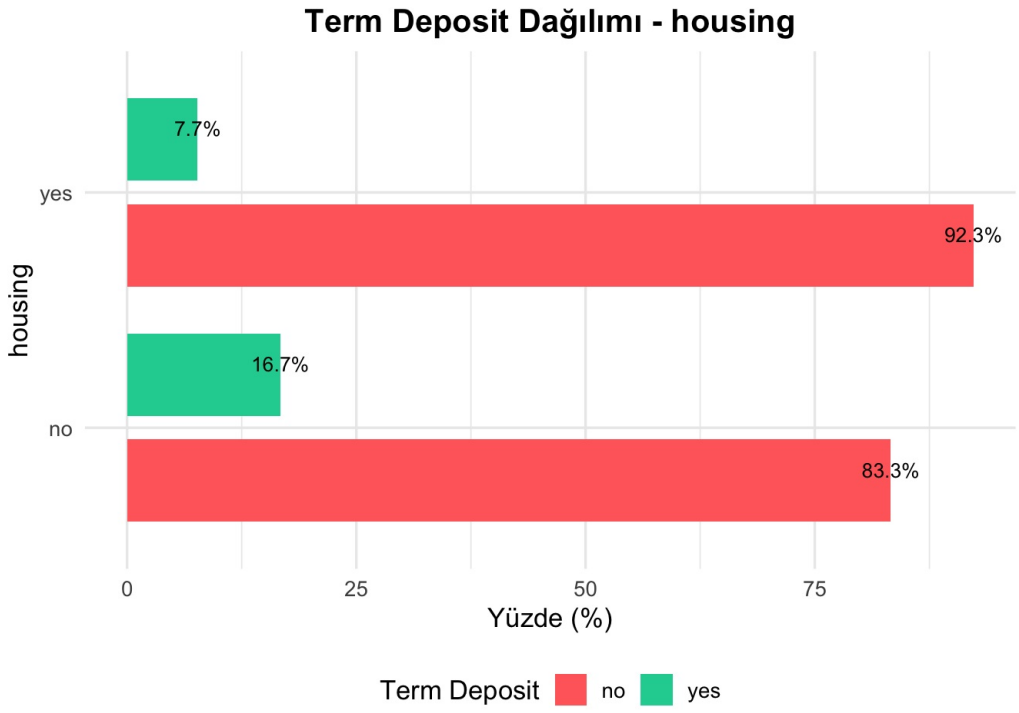
```r
library(ggplot2)
library(forcats)
library(rlang)

cat_vars <- names(data)[sapply(data, is.factor)]
custom_colors <- c("no" = "#ff6b6b", "yes" = "#1dd1a1")

for (var in cat_vars) {
  p_data <- data %>%
    group_by(across(all_of(var)), y) %>%      # kategori ve y'ye göre say
    summarise(count = n(), .groups = "drop") %>%
    group_by(across(all_of(var))) %>%         # kategori bazında topla
    mutate(percent = count / sum(count) * 100) %>%
    ungroup()

  # label sütunu (% formatında)
  p_data <- p_data %>% mutate(label = sprintf("%.1f%%", percent))

  p <- ggplot(p_data, aes(x = fct_reorder(as.factor(!!sym(var)), percent, .fun = max),
                          y = percent, fill = y, label = label)) +
    geom_col(position = position_dodge(width = 0.9), width = 0.7) +
    geom_text(position = position_dodge(width = 0.9),
              vjust = -0.25, size = 3.5, color = "black") +
    coord_flip() +
    scale_fill_manual(values = custom_colors) +
    labs(title = paste("Term Deposit Dağılımı -", var),
         x = var, y = "Yüzde (%)", fill = "Term Deposit") +
    theme_minimal(base_size = 13) +
    theme(plot.title = element_text(face = "bold", hjust = 0.5),
          legend.position = "bottom")

  print(p)
}
```

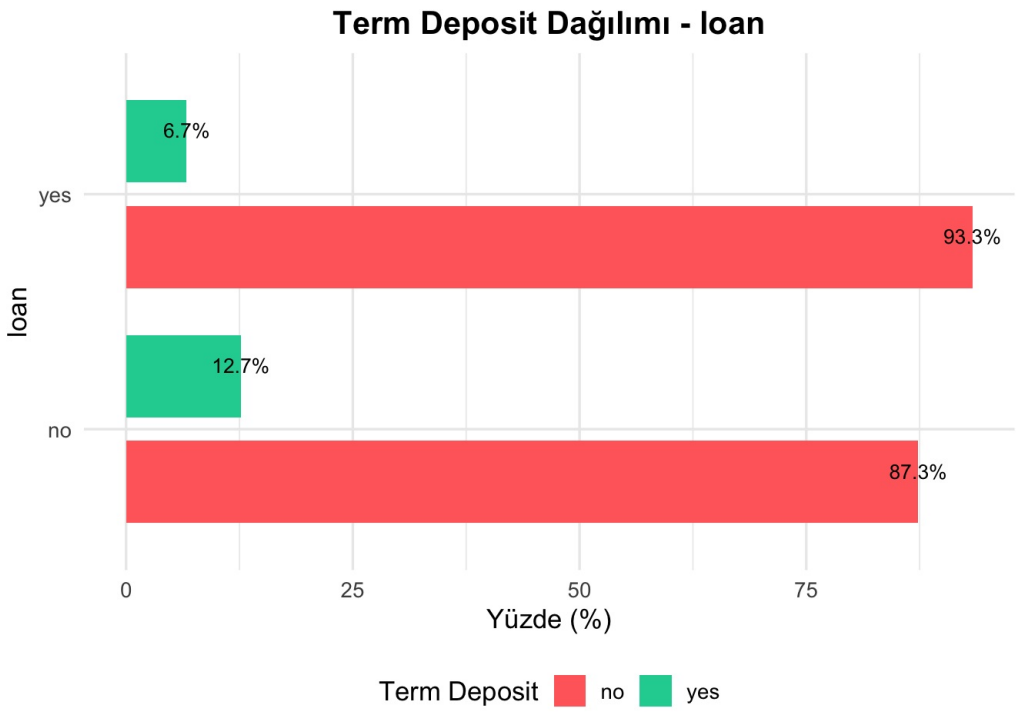

Term Deposit Dağılımı - job
blue-collar
entrepreneur
housemaid
services
technician
unknown
self-employed
admin.
management
unemployed
retired
student
job
7.3%
92.7%
8.3%
91.7%
8.8%
91.2%
8.9%
91.1%
11.1%
88.9%
11.8%
88.2%
11.8%
88.2%
12.2%
87.8%
13.8%
86.2%
15.5%
84.5%
22.8%
77.2%
28.7%
71.3%
0
25
50
75
Yüzde (%)
Term Deposit
no
yes

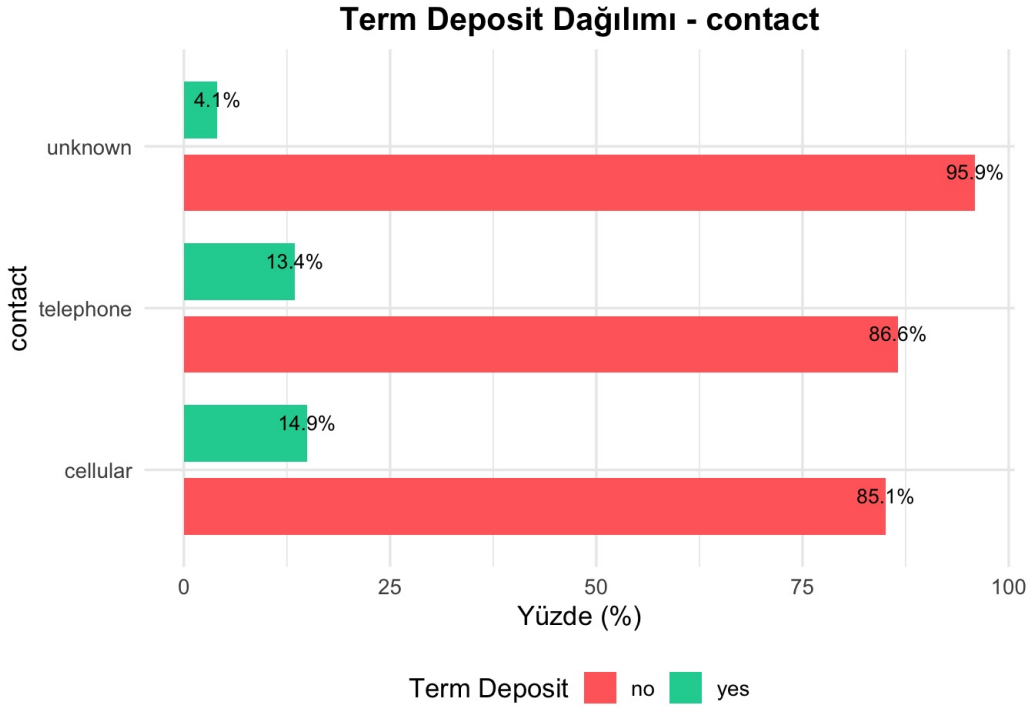

Term Deposit Dağılımı - marital
married
divorced
single
marital
10.1%
89.9%
11.9%
88.1%
14.9%
85.1%
0
25
50
75
Yüzde (%)
Term Deposit
no
yes

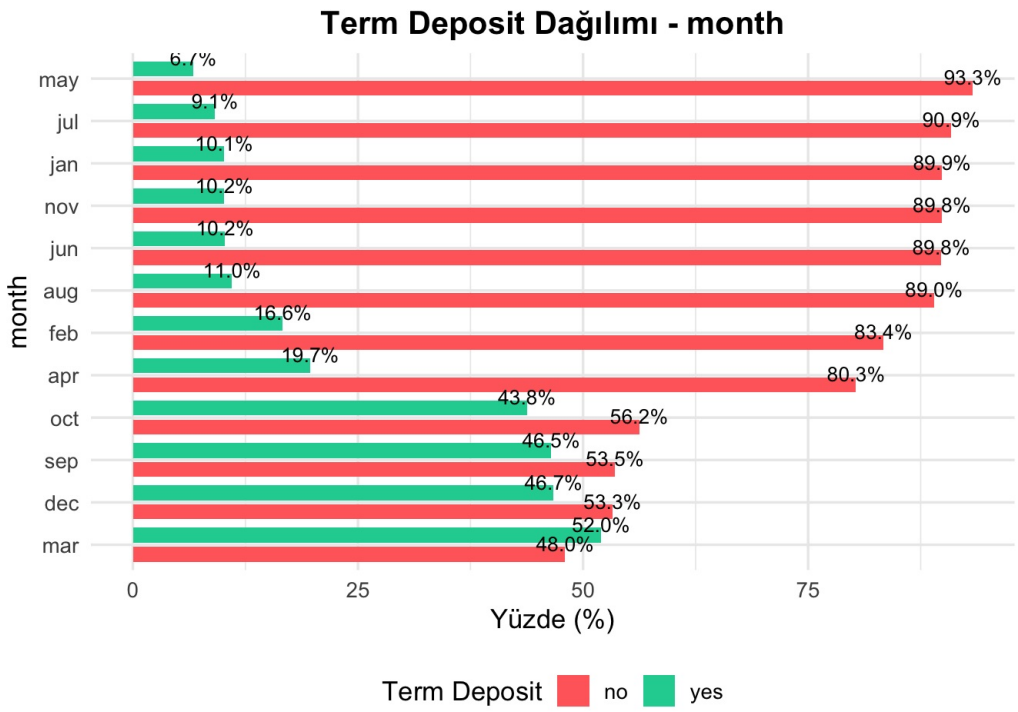

Term Deposit Dağılımı - education
primary
secondary
unknown
tertiary
education
8.6%
91.4%
10.6%
89.4%
13.6%
86.4%
15.0%
85.0%
0
25
50
75
Yüzde (%)
Term Deposit
no
yes

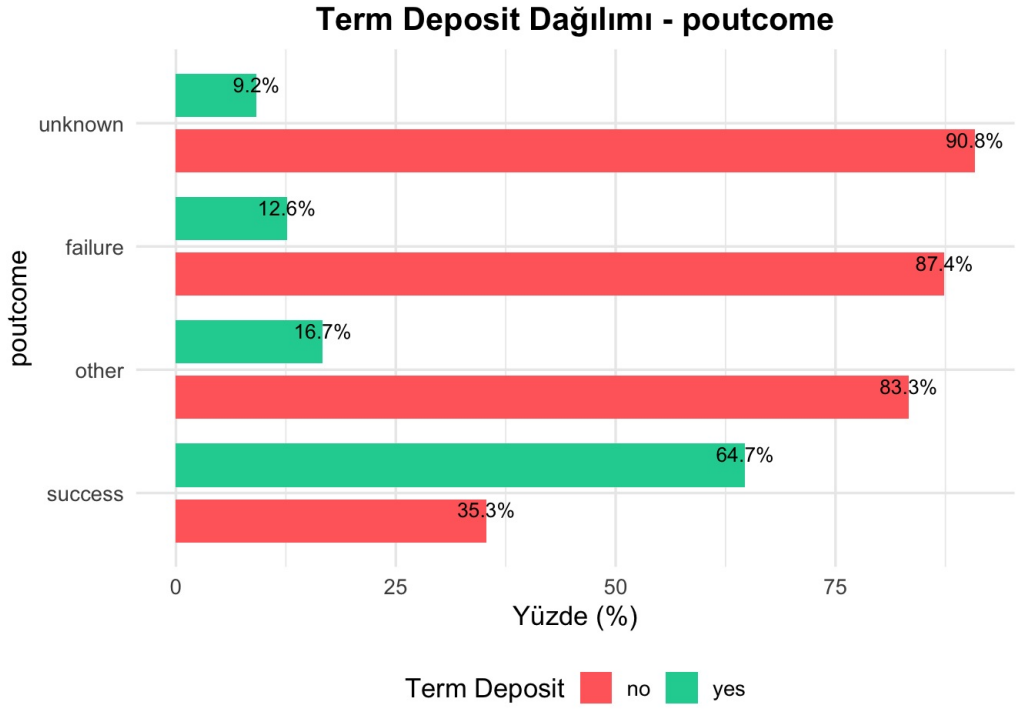

Term Deposit Dağılımı - default
6.4%
yes
93.6%
default
11.8%
no
88.2%
0
25
50
75
Yüzde (%)
Term Deposit
no
yes

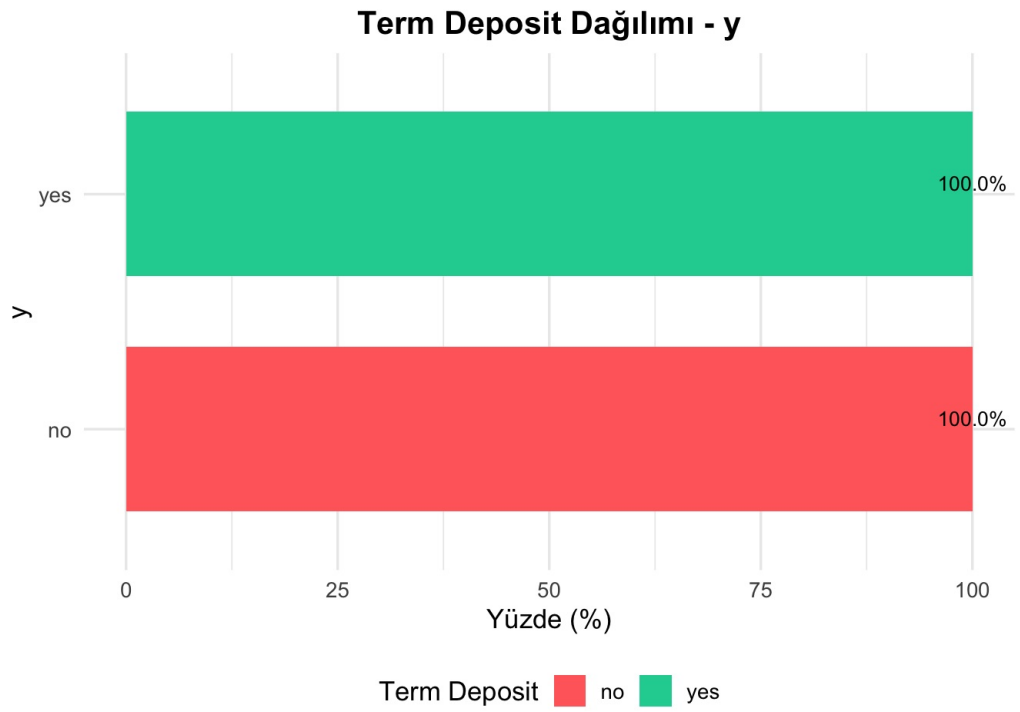

Term Deposit Dağılımı - housing
7.7%
yes
92.3%
housing
16.7%
no
83.3%
0
25
50
75
Yüzde (%)
Term Deposit
no
yes


Term Deposit Dağılımı - loan
loan
yes
no
6.7%
93.3%
12.7%
87.3%
0
25
50
75
Yüzde (%)
Term Deposit
no
yes


Term Deposit Dağılımı - contact
contact
unknown
telephone
cellular
4.1%
95.9%
13.4%
86.6%
14.9%
85.1%
0
25
50
75
100
Yüzde (%)
Term Deposit
no
yes


Term Deposit Dağılımı - month
may
jul
jan
nov
jun
aug
feb
apr
oct
sep
dec
mar
month
6.7%
93.3%
9.1%
90.9%
10.1%
89.9%
10.2%
89.8%
10.2%
89.8%
11.0%
89.0%
16.6%
83.4%
19.7%
80.3%
43.8%
56.2%
46.5%
53.5%
46.7%
53.3%
52.0%
48.0%
0
25
50
75
Yüzde (%)
Term Deposit
no
yes


Term Deposit Dağılımı - poutcome
unknown
failure
other
success
poutcome
9.2%
90.8%
12.6%
87.4%
16.7%
83.3%
64.7%
35.3%
0
25
50
75
Yüzde (%)
Term Deposit
no
yes



Kayıp gözlem kontrolü:

```
sum(is.na(data)) # kayıp gözlem görünmüyor.
```

```
## [1] 0
```

Bu aşamada y değişkenini tahmin etmek üzere regresyon modeli kurup tahmin yapacağız

Bağımlı değişkenin dağılımı:

```
table(data$y)
```

```
##
##    no   yes
## 39922  5289
```

Bağımlı değişkenimizin alt kategorilerini incelediğimizde ret sayısının kabul sayısından yaklaşık 7.5 kat fazla olduğu görülmektedir. Tahmin modelini eğitirken bu dengesizliğin yaratacağı yanlılığın önüne geçmek adına eğitim kümesine eşit sayıda kabul ve ret içeren gözlemleri dahil etmek gerekmektedir. Bu işlem yalnızca eğitim kümesinde gerçekleştirilecek ve test kümesine müdahale edilmeyecektir.

Eğitim setine yeterli sayıda kabul ve ret alabilmek adına düzenlemeler yapalım:

```
library(dplyr)
dataYes <- data %>% filter(y == "yes")
dataNo <- data %>% filter(y == "no")
nrow(dataYes) ; nrow(dataNo)
```

```
## [1] 5289
```

```
## [1] 39922
```

```
set.seed(111)
dataNoIndex <- sample(1:nrow(dataNo), size = 0.8*nrow(dataYes))
set.seed(111)
dataYesIndex <- sample(1:nrow(dataYes), size = 0.8*nrow(dataYes))

trainYes <- dataYes[dataYesIndex, ]
trainNo <- dataNo[dataNoIndex, ]

# Şimdi rbind kullanarak alt gruplardan ayrı ayrı aldığımız örneklemleri satır bazında birleştirelim.
trainset<-rbind(trainYes,trainNo)
table(trainset$y) # eşit sayıda no ve yes içeren gözlemi train sete dahil etmiş olduk.
```

```
##
##   no  yes
## 4231 4231
```

Aynı işlemleri test seti için de yapalım

```
testYes <- dataYes[-dataYesIndex,]
testNo <-dataNo[-dataNoIndex,]

testset<-rbind(testYes, testNo)
table(testset$y)
```

```
##
##    no   yes
## 35691  1058
```

GLMNET yöntemi ile ilk modeli kuralım

```
library(glmnet)
```

```
## Loading required package: Matrix
```

```
## Loaded glmnet 4.1-8
```

```
model_glm <- glm(y ~. , family = "binomial", data = trainset)
summary(model_glm)
```

```
##
## Call:
## glm(formula = y ~ ., family = "binomial", data = trainset)
##
## Coefficients:
##                       Estimate Std. Error z value Pr(>|z|)
## (Intercept)        -8.686e-01  3.069e-01  -2.831  0.00464 **
## age                -2.018e-03  3.648e-03  -0.553  0.58010
## jobblue-collar     -5.411e-01  1.213e-01  -4.462 8.11e-06 ***
## jobentrepreneur    -2.643e-01  2.010e-01  -1.315  0.18861
## jobhousemaid       -5.992e-01  2.277e-01  -2.632  0.00848 **
## jobmanagement      -1.707e-01  1.260e-01  -1.355  0.17554
## jobretired          3.544e-01  1.707e-01   2.076  0.03793 *
## jobself-employed   -5.323e-01  1.822e-01  -2.922  0.00348 **
## jobservices        -3.794e-01  1.419e-01  -2.674  0.00750 **
## jobstudent          5.454e-01  2.030e-01   2.686  0.00723 **
## jobtechnician      -1.382e-01  1.169e-01  -1.183  0.23698
## jobunemployed      -2.677e-01  1.908e-01  -1.403  0.16063
## jobunknown          1.416e-02  4.041e-01   0.035  0.97205
## maritalmarried     -7.655e-02  9.857e-02  -0.777  0.43740
## maritalsingle       2.030e-01  1.132e-01   1.792  0.07306 .
## educationsecondary  1.143e-01  1.076e-01   1.062  0.28831
## educationtertiary   2.810e-01  1.267e-01   2.217  0.02664 *
## educationunknown    4.340e-02  1.779e-01   0.244  0.80726
## defaultyes         -1.935e-01  2.579e-01  -0.750  0.45312
## balance             1.963e-05  1.018e-05   1.929  0.05369 .
## housingyes         -8.207e-01  7.213e-02 -11.378  < 2e-16 ***
## loanyes            -6.908e-01  9.605e-02  -7.191 6.41e-13 ***
## contacttelephone   -2.571e-01  1.309e-01  -1.963  0.04960 *
## contactunknown     -1.685e+00  1.131e-01 -14.893  < 2e-16 ***
## day                 6.910e-03  4.111e-03   1.681  0.09280 .
## monthaug           -8.964e-01  1.290e-01  -6.947 3.74e-12 ***
## monthdec            1.114e+00  4.326e-01   2.576  0.01001 *
## monthfeb            7.448e-02  1.491e-01   0.499  0.61743
## monthjan           -1.351e+00  1.939e-01  -6.966 3.26e-12 ***
## monthjul           -1.115e+00  1.290e-01  -8.645  < 2e-16 ***
## monthjun            2.319e-01  1.539e-01   1.507  0.13171
## monthmar            1.599e+00  2.366e-01   6.756 1.41e-11 ***
## monthmay           -6.390e-01  1.236e-01  -5.172 2.32e-07 ***
## monthnov           -8.765e-01  1.401e-01  -6.257 3.92e-10 ***
## monthoct            1.165e+00  2.082e-01   5.593 2.23e-08 ***
## monthsep            1.449e+00  2.736e-01   5.297 1.17e-07 ***
## duration            5.865e-03  1.519e-04  38.623  < 2e-16 ***
## campaign           -8.714e-02  1.563e-02  -5.577 2.45e-08 ***
## pdays              -1.308e-05  4.907e-04  -0.027  0.97873
## previous            4.243e-02  1.828e-02   2.321  0.02029 *
## poutcomeother       3.996e-01  1.558e-01   2.566  0.01030 *
## poutcomesuccess     2.237e+00  1.620e-01  13.813  < 2e-16 ***
## poutcomeunknown    -2.845e-02  1.621e-01  -0.176  0.86065
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 11730.8  on 8461  degrees of freedom
## Residual deviance:  6686.7  on 8419  degrees of freedom
## AIC: 6772.7
##
## Number of Fisher Scoring iterations: 6
```

Çıktıda modelin anlamlı (p-value < alpha) olduğu görünmektedir. NullDeviance(Boş modelin sapması) = 11730.8 elde edilirken, modele değişkenler eklendiğinde ResidualDeviance(Artıkların sapması) = 6686.7 olarak daha iyi bir model elde edilmiştir. Yani **bağımsız değişkenlerin eklenmesi modeli iyileştirmiştir.** Her bir değişkenin modele katkılarını ve değişkenlerin anlamlı olup olmadıklarını basitçe çıktıdaki * sayısından yorumlayabiliriz.
Modele katkısı olmayan veya katkısı az olan değişkenlerin model çıkarılması model karmaşıklığını düşürmek adına fayda sağlayacaktır. İlerleyen aşamalarda bu işlemlere yer verilecektir.
Elde edilen model katsayıları hakkında daha detaylı yorumlar yapabilmek adına exp(β) katsayıları elde edilmelidir.

```
exp(coef(model_glm))
```

```
##        (Intercept)                 age     jobblue-collar    jobentrepreneur
##          0.4195240           0.9979837           0.5820974           0.7677620
##       jobhousemaid      jobmanagement         jobretired   jobself-employed
##          0.5492265           0.8430387           1.4252747           0.5872415
##        jobservices         jobstudent      jobtechnician      jobunemployed
##          0.6843038           1.7253149           0.8708816           0.7651186
##         jobunknown     maritalmarried      maritalsingle educationsecondary
##          1.0142613           0.9263087           1.2250589           1.1210819
##  educationtertiary   educationunknown         defaultyes             balance
##          1.3244210           1.0443585           0.8240944           1.0000196
##         housingyes             loanyes   contacttelephone     contactunknown
##          0.4401284           0.5011905           0.7733229           0.1854430
##                day             monthaug             monthdec             monthfeb
##          1.0069337           0.4080273           3.0468524           1.0773202
##            monthjan             monthjul             monthjun             monthmar
##          0.2589776           0.3279464           1.2610269           4.9468424
##            monthmay             monthnov             monthoct             monthsep
##          0.5278018           0.4162537           3.2049005           4.2607833
##            duration             campaign                 pdays             previous
##          1.0058819           0.9165520           0.9999869           1.0433427
##      poutcomeother    poutcomesuccess    poutcomeunknown
##          1.4912731           9.3662918           0.9719526
```

Bu değerler odds oranıdır. Katsayı 1'in altındaysa azaltıcı, 1'e eşitse etkisi aynı(fark yok), 1'den fazlaysa etkisi fazladır yorumu yapılabilir.

Pseudo_r2 değeri:

```
library(pscl)
```

```
## Classes and Methods for R originally developed in the
## Political Science Computational Laboratory
## Department of Political Science
## Stanford University (2002-2015),
## by and under the direction of Simon Jackman.
## hurdle and zeroinfl functions by Achim Zeileis.
```

```
pseudo_r2 = pscl::pR2(model_glm)["McFadden"]
```

```
## fitting null model for pseudo-r2
```

```
pseudo_r2
```

```
##  McFadden
## 0.4299881
```

0.40'tan büyük Pseudo_r2 değeri modelin anlamlı tahminler yaptığını işaret eder. Kurduğumuz model anlamlı tahminler yapabilimiştir!

Hangi değişken modele daha fazla katkı sağlamış?

Bu yorumu yapabilmek için Resid. Dev kısmındaki değeri en çok hangi değişkenlerin düşürdüğü incelenir.

```
anova(model_glm)
```

NULL
age
job
marital
education
default
balance
housing
loan
contact

1-10 of 17 rows | 1-1 of 6 columns Previous **1** 2 Next

Değerler incelendiğinde modelin summary'sinde de olduğu gibi anlamlı çıkan değişkenlerin modele katkısının daha yüksek olduğunu görüyoruz. Tüm değişkenler incelendiğinde complexity'ide azaltmak için etkisi az olan değişkenleri modelden çıkarabiliriz. Modelin karmaşıklıktan uzak olması istediğimiz bir şeydir.

Model Tahminleri

```
predict1<-predict(model_glm,testset,type="response")
head(predict1)
```

```
##         8         9        10        11        13        16
## 0.9098142 0.9982538 0.3324927 0.9985782 0.9653114 0.4211302
```

```
library(devtools)
```

```
## Loading required package: usethis
```

```
cm<-InformationValue::confusionMatrix(testset$y, predictedScores = predict1)
cm
```

0
1

2 rows | 1-1 of 3 columns

Sütun isimlerinde yer alan No ve Yes değerleri test setinde yer alan gerçek değerleri, satır isimlerinde yer alan **0** değeri model tahmininde **No** olarak belirlenen tahminleri, **1** değeri ise **Yes** olarak belirlenen tahminleri ifade etmektedir.
confusion matrix incelendiğinde **model_glm** doğru negatifleri(30238 tahmin) tespit etmekte oldukça başarılı görünüyor. Başka bir deyişle model_glm mevduata abone olmama durumunu doğru tahmin etmekte başarılı ancak mevduata abone olma durumunu doğru tahmin etmekte iyileştirmeler yapılabilir. Bu değerlendirmeleri **accuracy(doğruluk) ve errorrate(hata oranı)** metriklerinde de inceleyelim:

Accuracy toplam gözlem sayısısnın doğru atamalara oranlanması ile edle edilen bir metricdir.

```
accuracy1<-(cm[2,2]+cm[1,1])/sum(cm)
accuracy1
```

```
## [1] 0.845955
```

accuracy metriği doğru tahminlerin tüm tahminlere oranlamasıyla elde edillir. Tahminlerde %84 doğruluk oranı sağlanmış olmasına karşın yanlış negatiflerin çok olması durumu iyileştirilebilir.

Bir de **Error Rate**(hata oranına) bakalım. Hata oranı yanlış atamaların toplam gözlem sayısına oranlanmasıyla elde edilir.

```
errorRate1<-(cm[1,2]+cm[2,1])/sum(cm)
errorRate1
```

```
## [1] 0.154045
```

Hata oranı %15 olarak elde edildi. Yani model1 ile bir tahminde bulunulduğunda %85 doğru %15 yanlış bir sonuç elde edilecektir.

GLM MODEL İYİLEŞTİRMESİ

Optimal Cut_off Value

```
# install.packages("InformationValue")
library(InformationValue)

optCutoff<-InformationValue::optimalCutoff(testset$y,predictedScores=predict1)
optCutoff
```

```
## [1] 0.009999996
```

Esik degeri cok kucuk bulunmustur 0.0099 Burda **optCutt_off** point belirlenirken dikkate alınan metric *accurcy* dir. Fakat probleme göre veriye göre bu durum değişiklik gösterebilir. Yani accuracy'ye göre **cutoff_point** belirlememiz bazen yanıltıcı olabilir. Testte iyi sonuçlar verirken gerçek

hayat probleminde elde edilen cutoff_pointe göre sonuçlar gerçeği yansıtmayabilir. Ve cutoff-point'in belirlenmesi bazen araştırıcıya da bırakılabilir. Şimdi bu optimal cutoff noktasına göre confusion matrixi yeniden oluşturursak;

```
cmOpt<-InformationValue::confusionMatrix(testset$y,predictedScores = predict1,
                                         threshold =optCutoff )
cmOpt       # threshold değerini optcutoff yaptık.
```

0
1

2 rows | 1-1 of 3 columns

```
accuracyopt<-(cmOpt[2,2]+cmOpt[1,1])/sum(cmOpt)
accuracyopt; cm; cmOpt
```

```
## [1] 0.041661
```

0
1

2 rows | 1-1 of 3 columns

0
1

2 rows | 1-1 of 3 columns

Bu noktada cutoff_pointe gore atandıgında **positive-positive** oranı artmış fakat **negative-negative** oranı düşmüştür. Accuracy değerinin de düştüğü görülmektedir. Bu noktada amaca gore doğru eşik değeri belirlenmelidir. Araştırma bazında hangi metriğin kritik rol aldığına bağlı olarak cutt_off değeri ayarlanabilir. Örneğin doğru-pozitifleri yakalamanın önemli olduğu hastalık teşhisi gibi durumlar olabilir. Bu durum test verisi uzerindende inceleme yaparak karar verilebilir.

Daha önce belirtildiği gibi yanlış negatiflerin(Tip I hata) sebebini inceleyelim:

```
summary(predict1)
```

```
##      Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
## 0.0001522 0.0695097 0.1622704 0.2603437 0.3630364 1.0000000
```

model_glm kullanılarak elde edilen tahminlerin medyan değeri 0.16 gibi düşük bir değer gelmiştir. Default(varsayılan) olarak 0.5 odds oranı ile çalıştığımız için bu sonuç ortaya çıkmış olabilir. Optimal cut_off value belirleyerek bu konuda iyileştirme yapılabilir. İlerleyen aşamalarda bu konuya değinilecektir.

İKİNCİ MODELİN KURULMASI

Daha az bağımsız değişkenle daha iyi bir model kurmak pek çok açıdan avantajlı bir durumdur. Daha basit bir model kurmak adına değişken sayısı azaltılabilir. Anlamlı değişkenlerle modeli tekrar kurup daha iyi bir model elde etmeye çalışalım:

```
new_train_set <- trainset %>% select(job,marital,education,default,balance,housing,loan,contact,day,month,duratio
n,campaign,pdays,previous,poutcome,y)
head(new_train_set)
```

979
175
2841
699
2117

3747

6 rows | 1-1 of 17 columns

```
model_glm2 <- glm(y ~.,
              family = "binomial", data = new_train_set)
summary(model_glm2)
```

```
##
## Call:
## glm(formula = y ~ ., family = "binomial", data = new_train_set)
##
## Coefficients:
##                       Estimate Std. Error z value Pr(>|z|)
## (Intercept)        -9.633e-01  2.548e-01  -3.781 0.000156 ***
## jobblue-collar     -5.387e-01  1.212e-01  -4.445 8.78e-06 ***
## jobentrepreneur    -2.665e-01  2.009e-01  -1.326 0.184738
## jobhousemaid       -6.093e-01  2.271e-01  -2.684 0.007282 **
## jobmanagement      -1.737e-01  1.259e-01  -1.380 0.167735
## jobretired          3.145e-01  1.549e-01   2.031 0.042270 *
## jobself-employed   -5.331e-01  1.822e-01  -2.926 0.003431 **
## jobservices        -3.789e-01  1.419e-01  -2.671 0.007568 **
## jobstudent          5.648e-01  2.000e-01   2.824 0.004743 **
## jobtechnician      -1.398e-01  1.169e-01  -1.196 0.231568
## jobunemployed      -2.679e-01  1.908e-01  -1.404 0.160372
## jobunknown          5.612e-03  4.040e-01   0.014 0.988918
## maritalmarried     -7.255e-02  9.831e-02  -0.738 0.460528
## maritalsingle       2.245e-01  1.064e-01   2.110 0.034858 *
## educationsecondary  1.200e-01  1.072e-01   1.120 0.262822
## educationtertiary   2.893e-01  1.259e-01   2.298 0.021556 *
## educationunknown    4.207e-02  1.779e-01   0.236 0.813089
## defaultyes         -1.913e-01  2.580e-01  -0.742 0.458307
## balance             1.913e-05  1.013e-05   1.887 0.059124 .
## housingyes         -8.176e-01  7.191e-02 -11.371  < 2e-16 ***
## loanyes            -6.892e-01  9.601e-02  -7.178 7.07e-13 ***
## contacttelephone   -2.675e-01  1.296e-01  -2.064 0.039025 *
## contactunknown     -1.686e+00  1.131e-01 -14.898  < 2e-16 ***
## day                 6.940e-03  4.110e-03   1.688 0.091345 .
## monthaug           -8.980e-01  1.290e-01  -6.962 3.36e-12 ***
## monthdec            1.111e+00  4.324e-01   2.570 0.010160 *
## monthfeb            7.374e-02  1.491e-01   0.495 0.620827
## monthjan           -1.352e+00  1.940e-01  -6.971 3.16e-12 ***
## monthjul           -1.113e+00  1.289e-01  -8.637  < 2e-16 ***
## monthjun            2.332e-01  1.538e-01   1.516 0.129462
## monthmar            1.598e+00  2.366e-01   6.754 1.44e-11 ***
## monthmay           -6.372e-01  1.235e-01  -5.159 2.48e-07 ***
## monthnov           -8.780e-01  1.400e-01  -6.270 3.62e-10 ***
## monthoct            1.163e+00  2.083e-01   5.585 2.34e-08 ***
## monthsep            1.450e+00  2.737e-01   5.297 1.17e-07 ***
## duration            5.865e-03  1.518e-04  38.625  < 2e-16 ***
## campaign           -8.706e-02  1.562e-02  -5.572 2.51e-08 ***
## pdays              -1.993e-05  4.903e-04  -0.041 0.967570
## previous            4.245e-02  1.828e-02   2.322 0.020250 *
## poutcomeother       4.015e-01  1.557e-01   2.578 0.009924 **
## poutcomesuccess     2.235e+00  1.619e-01  13.803  < 2e-16 ***
## poutcomeunknown    -2.944e-02  1.620e-01  -0.182 0.855796
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 11731  on 8461  degrees of freedom
## Residual deviance:  6687  on 8420  degrees of freedom
## AIC: 6771
##
## Number of Fisher Scoring iterations: 6
```

```
pseudo_r2 = pscl::pR2(model_glm)["McFadden"]
```

```
## fitting null model for pseudo-r2
```

```
pseudo_r2_2 = pscl::pR2(model_glm2)["McFadden"]
```

```
## fitting null model for pseudo-r2
```

```
pseudo_r2 ; pseudo_r2_2
```

```
##  McFadden
## 0.4299881
```

```
## McFadden
## 0.429962
```

```
predict2 <- predict(model_glm2, testset ,type="response")
head(predict2)
```

```
##         8         9        10        11        13        16
## 0.9094505 0.9982244 0.3306641 0.9985548 0.9649953 0.4241437
```

Accuracy

```
cm2 <- InformationValue::confusionMatrix(testset$y, predictedScores = predict2)
cm2
```

0
1

2 rows | 1-1 of 3 columns

```
accuracy2  <-(cm2[2,2]+cm2[1,1])/sum(cm2)
accuracy2
```

```
## [1] 0.8460911
```

Recall

```
recall2<-(cm2[2,2])/(cm2[1,2]+cm2[2,2])
recall2
```

```
## [1] 0.8043478
```

Recall değeri 0.8043478 bulunmuştur. Modelin pozitif örneklerin %80.43478'ini doğru bir şekilde tespit ettiği anlamına gelir.

Precision

Kesinlik (Precision) Positive olarak tahminlediğimiz değerlerin gerçekten kaç adedinin Positive olduğunu göstermektedir.

```
precision2 <- (cm2[2,2])/(cm2[2,1]+cm2[2,2])
precision2
```

```
## [1] 0.1350794
```

Yani model *müşteri vadeli mevduat aboneliği kabul olanları* %13.50794 kesinlikle tahmin edebiliyor şeklinde yorumlanabilir.

Sensivity

Sensivity(Duyarlılık) aslında Recall ile aynı förmülüzasyona sahiptir.

```
sensivity2<-(cm2[2,2])/(cm2[1,2]+cm2[2,2])
sensivity2
```

```
## [1] 0.8043478
```

Positive classları tahmin ederken ne kadar hassaslıkla tahminde bulunuluğunu gösterir. model2'nin confusion matrix'i için bu değeri %80.43478 olarak elde edildi.

Specifity

Negative classı ne kadar iyi tahmin edebildiğimizi gösteren bir metriktir. Bu durumda *mevduata abone olmama durumunu* tahmin etmedeki model başarısını gösterecektir.

```
specificity2<-(cm2[1,1])/(cm2[2,1]+cm2[1,1])
specificity2
```

```
## [1] 0.8473285
```

Accuracy değeriyle yakın bir değer(0.8473285) elde ettik. Metrikler üzerindeki gözlemlerimiz sonucunda modelimizin **doğru negatifleri bulmada** başarılı bir model olduğunu görebiliyoruz.

F1 Scoru

F1'in yüksek olması positive classların tahmininde modelin iyi olduğunu belirten bir ölçüt olarak kullanılabilir. Bizim modelimiz için iyi sonuç vermesini beklemeyiz.

```
f1_score2<-2*((precision2*recall2)/(precision2+recall2))
f1_score2
```

```
## [1] 0.2313129
```

Modelin mevduata abone olma durumunu tahmin etme F1 scoru %23 olarak elde edilmistir. Bir kez daha modelin *mevduata abone olmama durumunu* tahmin etmede başarılı olduğunu vurgulayabiliriz.

ROC CURVE

Eğrinin altında kalan alan anlamını taşımaktadır. Alan büyüklüğü 0-1 arasındadır ve 1'e yakın değerler tercih edilir. Yani eğri altında kalan alan büyüdükçe model iyileşir.

```
#install.packages("pROC")
library(pROC)
```

```
## Type 'citation("pROC")' for a citation.
```

```
##
## Attaching package: 'pROC'
```

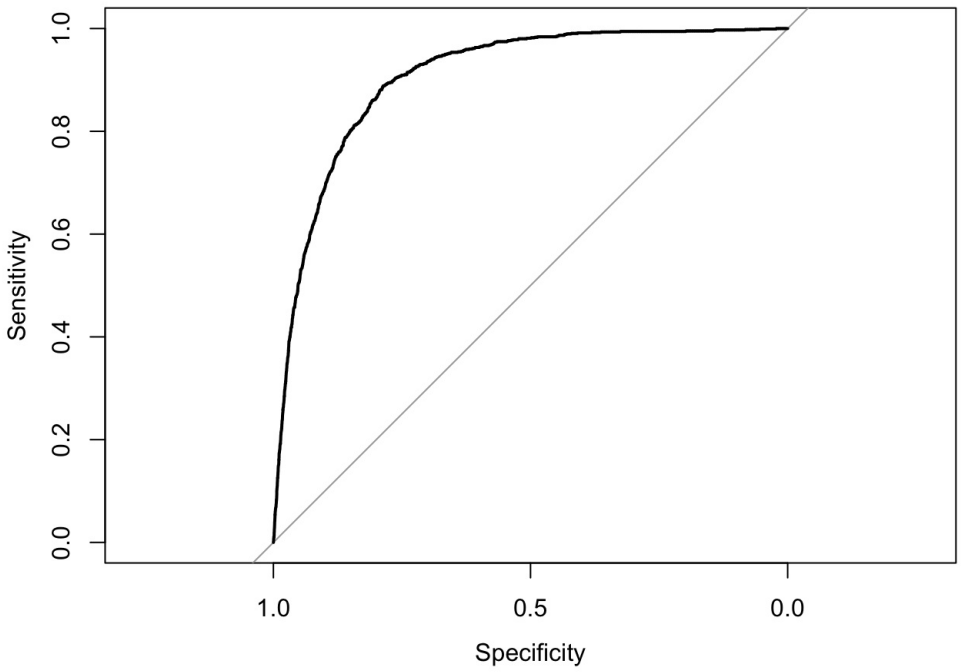
```
## The following objects are masked from 'package:stats':
##
##     cov, smooth, var
```

```
rocModel2<-roc(testset$y~predict2)
```

```
## Setting levels: control = no, case = yes
```

```
## Setting direction: controls < cases
```

```
plot(rocModel2)
```



Yukarıdaki görselde eğri altında kalan alanı gözlemleyebiliriz. Bir de sayısal değer olarak bakalım:

```
rocModel2
```

```
##
## Call:
## roc.formula(formula = testset$y ~ predict2)
##
## Data: predict2 in 35691 controls (testset$y no) < 1058 cases (testset$y yes).
## Area under the curve: 0.905
```

Area under the curve(eğri altında kalan alan) %90.5 olarak elde edilmiştir ve gayet iyi bir sonuçtur.

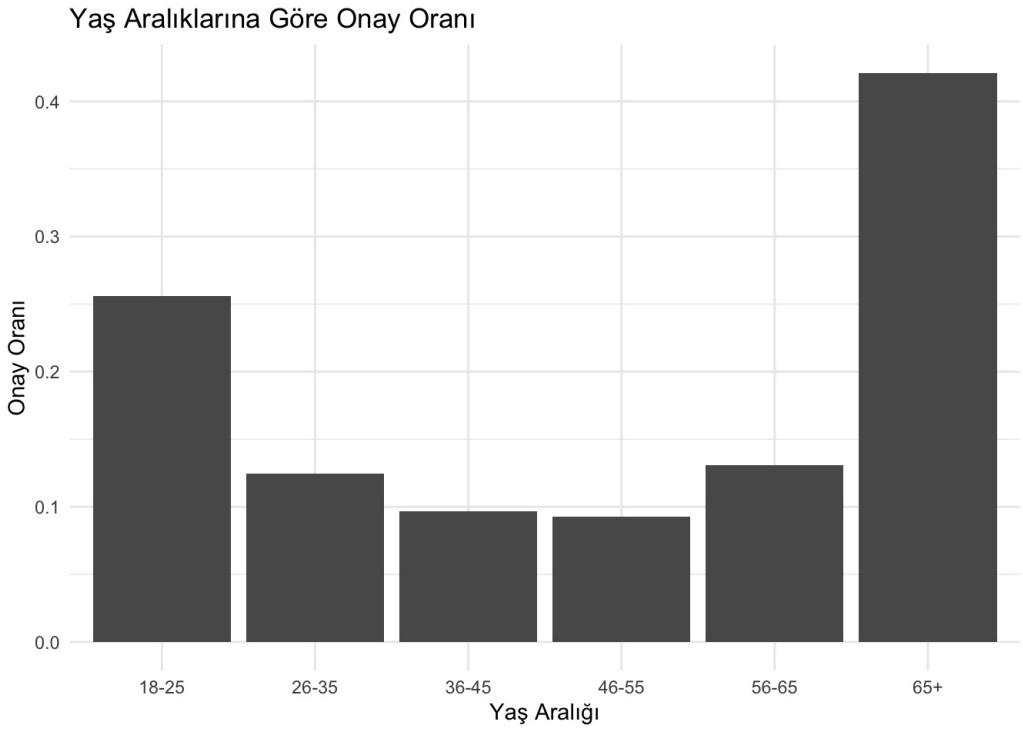
Sonuç: Elde Edilen Tahmin Denklemi ve Yorumlanması

İlk olarak 1. model(model_glm) veri setine uygun olarak *binomial logistic regresyon* kullanılarak elde edilmiş ve katsayıları yorumlanmıştır. Daha sonrasında 2. model(model_glm_2) elde edilirken complexity'i düşürmek hedeflenerek anlamsız değişkenler modelden çıkartılmış ve confusionmatrix üzerinden metriklerle karşılaştırma yapılıp model yorumlanmıştır. Seçilen model(model_glm_2)'yi iyileştirmek için cut_off value belirlenmesi gibi çeşitli yollara gidilmiş ve uygun adımlar uygulanmıştır.

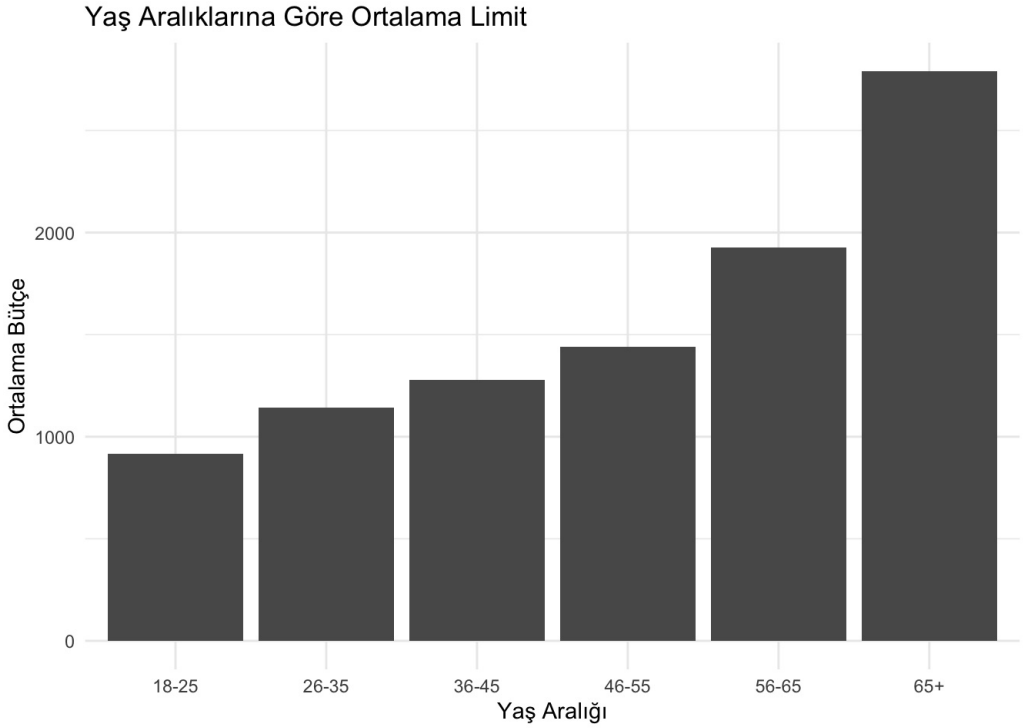
```
library(dplyr)

data <- data %>%
  mutate(age_group = cut(
    age,
    breaks = c(18, 25, 35, 45, 55, 65, 100),
    labels = c("18-25", "26-35", "36-45", "46-55", "56-65", "65+"),
    right = FALSE
  ))
```

```
summary_data <- data %>%
  group_by(age_group) %>%
  summarise(yas_bazinda_onay_orani = mean(y=="yes"),
            yas_bazinda_ortalama_butce = mean(balance))
```

```
library(ggplot2)
ggplot(summary_data, aes(x = age_group, y = yas_bazinda_onay_orani)) +
  geom_col() +
  labs(
    title = "Yaş Aralıklarına Göre Onay Oranı",
    x = "Yaş Aralığı",
    y = "Onay Oranı"
  ) +
  theme_minimal()
```




```
ggplot(summary_data, aes(x = age_group, y = yas_bazinda_ortalama_butce)) +
  geom_col() +
  labs(
    title = "Yaş Aralıklarına Göre Ortalama Limit",
    x = "Yaş Aralığı",
    y = "Ortalama Bütçe"
  ) +
  theme_minimal()
```



Bu iki grafikten anlaşılacağı üzre ortalama bütçe değişkeni onay oranı üzerinde kesin bir etkiye sahip değildir. Yaş aralığı arttıkça ortalama bütçe artmasına rağmen onay oranları orta yaş gruplarında düşük yüzdelerde seyir etmektedir.

Değişkenler Arasındaki Korelasyon Katsayıları

```
str(data)
```

```
## 'data.frame':    45211 obs. of  18 variables:
##  $ age      : int  58 44 33 47 33 35 28 42 58 43 ...
##  $ job      : Factor w/ 12 levels "admin.","blue-collar",..: 5 10 3 2 12 5 5 3 6 10 ...
##  $ marital  : Factor w/ 3 levels "divorced","married",..: 2 3 2 2 3 2 3 1 2 3 ...
##  $ education: Factor w/ 4 levels "primary","secondary",..: 3 2 2 4 4 3 3 3 1 2 ...
##  $ default  : Factor w/ 2 levels "no","yes": 1 1 1 1 1 1 1 2 1 1 ...
##  $ balance  : int  2143 29 2 1506 1 231 447 2 121 593 ...
##  $ housing  : Factor w/ 2 levels "no","yes": 2 2 2 2 1 2 2 2 2 2 ...
##  $ loan     : Factor w/ 2 levels "no","yes": 1 1 2 1 1 1 2 1 1 1 ...
##  $ contact  : Factor w/ 3 levels "cellular","telephone",..: 3 3 3 3 3 3 3 3 3 3 ...
##  $ day      : int  5 5 5 5 5 5 5 5 5 5 ...
##  $ month    : Factor w/ 12 levels "apr","aug","dec",..: 9 9 9 9 9 9 9 9 9 9 ...
##  $ duration : int  261 151 76 92 198 139 217 380 50 55 ...
##  $ campaign : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ pdays    : int  -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 ...
##  $ previous : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ poutcome : Factor w/ 4 levels "failure","other",..: 4 4 4 4 4 4 4 4 4 4 ...
##  $ y        : Factor w/ 2 levels "no","yes": 1 1 1 1 1 1 1 1 1 1 ...
##  $ age_group: Factor w/ 6 levels "18-25","26-35",..: 5 3 2 4 2 3 2 3 5 3 ...
```

```
num_vars = names(data)[sapply(data, is.numeric)]

cor_mat <- cor(data[num_vars])


cor_list <- as.data.frame(as.table(cor_mat))

cor_list <- cor_list %>%
  dplyr::filter(Var1 != Var2) %>%
  dplyr::distinct()

cor_sorted <- cor_list %>%
  dplyr::arrange(desc(Freq))

head(cor_sorted)
```

1	
2	
3	
4	
5	
6	

6 rows | 1-1 of 4 columns

Modelin Test Seti Tahminleri

Bu adımda modelden, modele daha önce göstermediğimiz test seti gözlemlerini tahmin etmesini isteyeceğiz. Model kurulurken modeli eğitmek için eğitim-test seti olarak bölmüştük. Buradaki test set modelin hiç görmediği değerlerden oluşmaktadır. Karıştırılmamalıdır.

```
test_data <- read.csv("/Users/umutaykanat/Desktop/portfolio/banking data/test.csv", sep = ";", header = TRUE)
head(test_data)
```

1	
2	
3	
4	
5	
6	

6 rows | 1-1 of 18 columns

```
predict_test <- predict(model_glm,test_data,type="response")
head(predict_test)
```

```
##          1          2          3          4          5          6
## 0.59996667 0.13136447 0.43198681 0.04651941 0.03256433 0.59860116
```

```
library(devtools)
cm<-InformationValue::confusionMatrix(test_data$y, predictedScores = predict_test)
cm
```

0
1

2 rows | 1-1 of 3 columns

```
accuracy_test<-(cm[2,2]+cm[1,1])/sum(cm)
accuracy_test
```

```
## [1] 0.8378677
```

Test setine ilişkin tahminlerin accuray oranı **%83.7'dir.** Modelin test set üzerinde başarılı tahminler yaptığını söyleyebiliriz.

TEŞEKKÜRLER

Umut Aykanat