Veri Biliminde Sınıflandırma Algoritmaları

1. Giriş

Bir veri kümesinde tanımlı olan sınıflar arasında veriyi dağıtmaya *sınıflandırma* denir. Sınıflandırma algoritmaları, eğitim kümesinden dağılımın şeklini öğrenir ve sınıfı belli olmayan test verileri geldiğinde doğru şekilde sınıflandırmaya çalışır.[1] Sınıflandırmada kullanılan algoritmalardan bazıları şunlardır:

- K En Yakın Komşu
- Çekirdek (Kernel) Regresyon
- Karar Ağaçları
- Destek Vektör Makinesi
- Naive Bayes Sınıflandırıcısı

Bu çalışmada sınıflandırma yöntemi olarak *lojistik regresyon* ve *karar ağaçları* incelenecektir.

2. Veri Kümesi

İncelenecek veri kümesi, bir Portekiz bankacılık kurumunun telefon görüşmelerine dayalı olan doğrudan pazarlama kampanyaları ile ilgilidir.[2] Çalışma, Mayıs 2008'den Kasım 2010'a kadar yapılan 41188 telefon görüşmesini içerir. Müşterilerin vadeli mevduatı kabul edip etmeyeceği bilgisine ulaşmak için çoğu kez aynı müşteriyle birden fazla kez irtibat kurma gereği doğmuştur.[3] Veri kümesi dengesizdir, sadece 4640 görüşme (%11.26) başarıyla sonuçlanmıştır.

```
data <- read.csv("/Users/meryemcamir/Desktop/dataset/bank-additional-full.csv", header = TRUE, sep = ";
```

İlk olarak masaüstündeki "dataset" klasöründe bulunan `bank-additional-full.csv` isimli dosya programa okutulur ve oluşan veri çerçevesi **"data"** adındaki değişkene atanır. `header = TRUE` veri çerçevesindeki ilk satırın, sütunların adıyla oluşturulan bir başlık olmasını sağlar.

```
30;blue-collar;married;basic.9y;no;yes;no;cellular;may;fri;487;2;999;0;nonexistent;-1.8;92.893;-46.2;1.3
```

Veri kümesinden alınan yukarıdaki örnekte görüldüğü üzere dosyanın her bir satırındaki değerler "**;**" karakteri ile birbirinden ayrılmıştır. `sep = ";"` ile ayrım yerlerinin program tarafından tanınması sağlanır.

Aşağıda `head` fonksiyonu aracılığıyla veri çerçevesinin baştaki satırlarından oluşturulmuş bir önizleme görülmektedir:

```
head(data)
```

```
##   age       job marital   education default housing loan   contact month
## 1  56 housemaid married    basic.4y      no      no   no telephone   may
## 2  57  services married high.school unknown      no   no telephone   may
## 3  37  services married high.school      no     yes   no telephone   may
## 4  40    admin. married    basic.6y      no      no   no telephone   may
## 5  56  services married high.school      no      no  yes telephone   may
## 6  45  services married    basic.9y unknown      no   no telephone   may
##   day_of_week duration campaign pdays previous    poutcome emp.var.rate
## 1         mon      261        1   999        0 nonexistent          1.1
## 2         mon      149        1   999        0 nonexistent          1.1
## 3         mon      226        1   999        0 nonexistent          1.1
## 4         mon      151        1   999        0 nonexistent          1.1
## 5         mon      307        1   999        0 nonexistent          1.1
## 6         mon      198        1   999        0 nonexistent          1.1
##   cons.price.idx cons.conf.idx euribor3m nr.employed  y
```

```
## 1         93.994         -36.4     4.857        5191 no
## 2         93.994         -36.4     4.857        5191 no
## 3         93.994         -36.4     4.857        5191 no
## 4         93.994         -36.4     4.857        5191 no
## 5         93.994         -36.4     4.857        5191 no
## 6         93.994         -36.4     4.857        5191 no
```

2.1. Değişkenler

Müşteri Bilgileri

age: Müşterilerin yaşlarından oluşmuştur. 17 ile 98 arasında değerler alır ve numerik tiptedir.

job: Müşterilerin mesleklerini belirtir. Kategoriler: 'admin.', 'blue-collar', 'entrepreneur', 'housemaid', 'management', 'retired', 'selfemployed', 'services', 'student', 'technician', 'unemployed', 'unknown'.

marital: Müşterilerin medeni halini gösterir. Kategoriler: 'divorced'*, 'married', 'single', 'unknown'.

education: Müşterilerin eğitim durumlarını gösterir. Kategoriler: 'basic.4y', 'basic.6y', 'basic.9y', 'high.school', 'illiterate', 'professional.course', 'university.degree', 'unknown'.

default: Müşterinin kredi borcu olup olmadığını gösterir. Kategoriler: 'no', 'yes', 'unknown'.

housing: Müşterinin konut kredisi sahibi olup olmadığını belirtir. Kategoriler: 'no', 'yes', 'unknown'.

loan: Müşterinin bireysel kredi kullanıp kullanmadığını gösterir. Kategoriler: 'no', 'yes', 'unknown'.

Son Telefon Görüşmesine İlişkin Bilgiler

contact: Müşteri ile kurulan iletişimin cep telefonu üzerinden mi yoksa karasal hat üzerinden mi gerçekleştiğini gösterir. Kategoriler: 'cellular', 'telephone'.

month: Müşteri ile son iletişime geçilen ayı gösterir. Kategoriler: 'jan', 'feb', 'mar', 'apr', 'may', 'jun', 'jul', 'aug', 'sep', 'oct', 'nov', 'dec'.

day_of_week: Müşteri ile son iletişime geçilen hafta içi gününü belirtir. Kategoriler: 'mon', 'tue', 'wed', 'thu', 'fri'.

duration: Son iletişimin süresini saniye cinsinden gösterir. 0 ile 4918 arasında değerler alır. Numerik tipte bir değişkendir.

Diğer Özellikler

campaign: Bu kampanya süresince müşteri ile kurulan irtibat sayısını gösterir, numerik tiptedir. 1 ile 56 arasında değerler alır.

pdays: Müşteri ile bir önceki kampanya esnasındaki son görüşmeden itibaren geçen gün sayısını gösterir. Numeriktir. Müşteri ile daha öncesinde herhangi bir iletişime geçilmemişse *999* değerini alır.

previous: Bu kampanyadan önce müşteri ile yapılan görüşme sayısını belirtir. Numeriktir ve 0 ile 7 arasında değişir.

poutcome: Önceki kampanyanın başarıya ulaşıp ulaşmadığını gösterir. Kategoriler: 'failure', 'nonexistent', 'success'.

Sosyal ve Ekonomik Özellikler

emp.var.rate: İstihdam varyasyon oranını gösterir ve numeriktir.

cons.price.idx: Aylık tüketici fiyatları endeksini (TÜFE) gösterir. Numeriktir.

cons.conf.idx: Aylık tüketici güven endeksini gösterir. Numeriktir.

euribor3m: AB'deki bankalar arası geriye dönük üç aylık faiz oranının günlük ortalama karşılığını belirtir. Numeriktir.

nr.employed: Müşterinin iş yerindeki çalışan sayısını (üç aylık gösterge) belirtir. Numeriktir.

Sonuç Değişkeni

y: Müşteri vadeli mevduat açtırmış mı? Kategoriler: 'yes', 'no'.

* *'divorced' kategorisinin içerisinde hem boşanmış olanlar hem de eşleri vefat edenler yer almaktadır.*

3. Lojistik Regresyon

3.1. Giriş

Regresyon analizi, herhangi bir değişkenin bir veya birden fazla değişkenle arasındaki ilişkiyi ölçmek için kullanılan analiz yöntemidir.

Bir vakada iki ana değişken vardır: bağımlı değişken ve bağımsız değişken. Bağımlı değişken sayısı tektir ancak bağımsız değişken sayısı birden fazla olabilir.

Bağımsız değişken (açıklayıcı değişken) bağımlı değişkeni etkilediği düşünülen sebep değişkenidir.

Bağımlı değişken (cevap değişkeni) vakadaki bağımsız değişkenlerden etkilenen, test edilen ve ölçülen değişkendir.

Burada incelenen veri kümesindeki ilk yirmi değişken bağımsız değişkenlerdir. Yirmi birinci değişken olan y, bağımlı değişkendir. Amaç bağımsız değişkenlerin ışığı altında y değişkeninin yanıtına ulaşmaktır.

Lineer regresyon, veri kümesinde normal dağılım ister. Ayrıca hem bağımlı değişken hem de bağımsız değişkenler nicel olmalıdır. Örneğin, yaş ile kan basıncı arasında bir ilişki saptanacaksa hem yaş hem de kan basıncı sayısal olarak belirtilmelidir.[4] Ancak gerçek hayatta çoğu zaman nitel değişkenlerle karşılaşılır.

Lojistik regresyon ise cevap değişkeninin kategorik ve çoklu; bağımsız değişkenlerin numerik veya kategorik olabildiği bir regresyon çeşididir. Bağımlı değişkenin kategori sayısına göre uygulanacak yöntem farklıdır.

Binomial	Multinomial	Ordinal
2 kategori etkili - etkisizevet - hayıriyileşti - iyileşmedi	2+ kategori*sırasız* çalışıyor - çalışmıyor - emekli	2+ kategori*sıralı* çok etkili - orta derecede etkili - az etkili

Lojistik regresyon, 1958'de İngiliz istatistikçi David Cox tarafından geliştirilmiştir. Tıp, sosyal bilimler, mühendislik, makine öğrenimi gibi birçok alanda lojistik regresyondan yararlanılır. Örneğin, yaralılarda ölüm oranının tahmin edilmesinde kullanılan "Travma ve Yaralanma Şiddeti Skoru" lojistik regresyon kullanılarak geliştirilmiştir. Bir seçmenin yaşı, gelir düzeyi, cinsiyeti, ikamet ettiği yer, önceki seçimlerdeki oyları gibi kategorilere bakılarak bir sonraki seçimde sağ kesime mi yoksa sol kesime mi yönelik oy kullanacağının tahmin edilmesi de bir başka kullanım alanı örneğidir. Pazarlama alanındaki kullanımına ise bir müşterinin bir ürünü veya hizmeti satın alacağının ya da üyeliğini durduracağının öngörülmesi örnek olarak verilebilir.

3.2. Matematiksel Açıklama

Standart lojistik fonksiyon $t \in \mathbb{R}, \quad f(t) \in (0,1)$ için

$$f(t) = \frac{e^t}{e^t + 1} = \frac{1}{1 + e^{-t}}$$

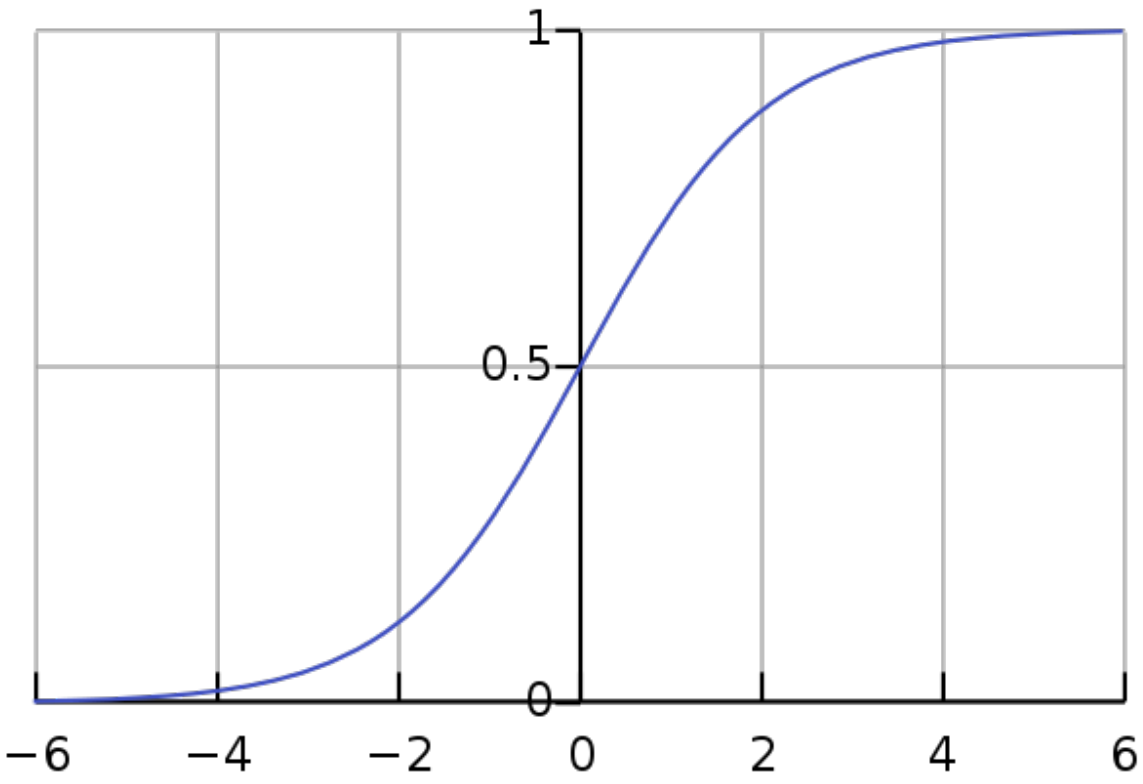


Figure 1:

şeklinde tanımlanır.

Kolaylık açısından t'nin tek bir bağımsız değişkene ait lineer bir fonksiyon olduğu kabul edilsin:

$$t = \beta_0 + \beta^T x \ .$$

Artık lojistik fonksiyon

$$p(x) = \frac{e^{\beta_0+\beta^T x}}{1+e^{\beta_0+\beta^T x}} \text{ veya } \ p(x) = \frac{1}{1+e^{-(\beta_0+\beta^T x)}}$$

şeklinde yazılabilir.

Lojistik regresyon modeli, K adet sınıfın toplamları bire eşit olan $(0,1)$ aralığındaki sonsal olasılıklarını x 'e bağlı lineer fonksiyonlar aracılığıyla modelleme isteğinden doğmuştur.[5]

Eğer iki adet sınıf $(K=2)$ varsa sonsal olasılık:

$$Pr(G=1|X=x) = \frac{e^{\beta_0+\beta^T x}}{1+e^{\beta_0+\beta^T x}} \ ,$$

$$Pr(G=2|X=x) = \frac{1}{1+e^{\beta_0+\beta^T x}} \ .$$

Bu iki denklemin oranının doğal logaritması alınırsa

$$ln(\frac{p}{1-p}) = ln(\frac{Pr(G=1|X=x)}{Pr(G=2|X=x)}) = \beta_0 + \beta^T x$$

lineer denklemi elde edilir.

Burada $\frac{p}{1-p}$ **odds** olarak adlandırılır. Yani bir şeyin başarılı olması veya meydana gelmesi olasılığının meydana gelmeme olasılığına oranıdır. Odds'un logaritmasının alınmasıyla oluşturulan $ln(\frac{p}{1-p})$ ise **lojit** olarak isimlendirilir.[6]

Lojistik regresyon modelleri genellikle G'nin X'e bağlı koşullu olasılığı ($Pr(G|X)$) kullanılarak maksimum olabilirlik *(maximum likelihood)* yöntemi ile uygun hale getirilir.

$p_k(x_i;\theta) = Pr(G=k|X=x_i;\theta)$ iken N adet gözlem için olabilirlik fonksiyonunun logaritması:

$$\ell(\theta) = \sum_{i=1}^{N} log p_{g_i}(x_i;\theta) \ .$$

Denklemlerde sade bir görüntü oluşturmak açısından sınıf sayısı 2 olarak kabul edilsin.

$g_i = 1$ iken $y_i = 1$, $\quad p_1(x;\theta) = p(x;\theta)$,

$g_i = 2$ iken $y_i = 0$, $\quad p_2(x;\theta) = 1 - p(x;\theta)$ olsun. Bu durumda olabilirlik fonksiyonunun logaritması

$$\ell(\beta) = \sum_{i=1}^{N} \{y_i \ log \ p(x_i;\beta) + (1-y_i) \ log \ (1-p(x_i;\beta))\}$$

şeklinde yazılabilir.[5] Bu fonksiyonun maksimize edilmesi, parametreye göre türevinin alınıp sıfıra eşitlenmesiyle gerçekleşir. Maksimizasyon sonucunda en uygun modele ulaşılır.

3.3. Model Oluşturma

```
egit <- read.csv("/Users/meryemcamir/Desktop/dataset/bank-additional.csv", header=TRUE, sep=";")
```

Modelleme için kullanılacak olan bu veri kümesi, **"bank-additional-full.csv"** isimli kümedeki örneklerin rastgele seçilmesiyle oluşturulmuştur ve örneklerin %10'unu içerir.[3]

Analize başlamadan önce veri kümesinde eksik değerler olup olmadığı kontrol edilmelidir. Eksik bir değere sahip gözlem varsa oluşturulacak modelin doğruluğu açısından ilgili satır, veri kümesinden çıkarılacaktır.

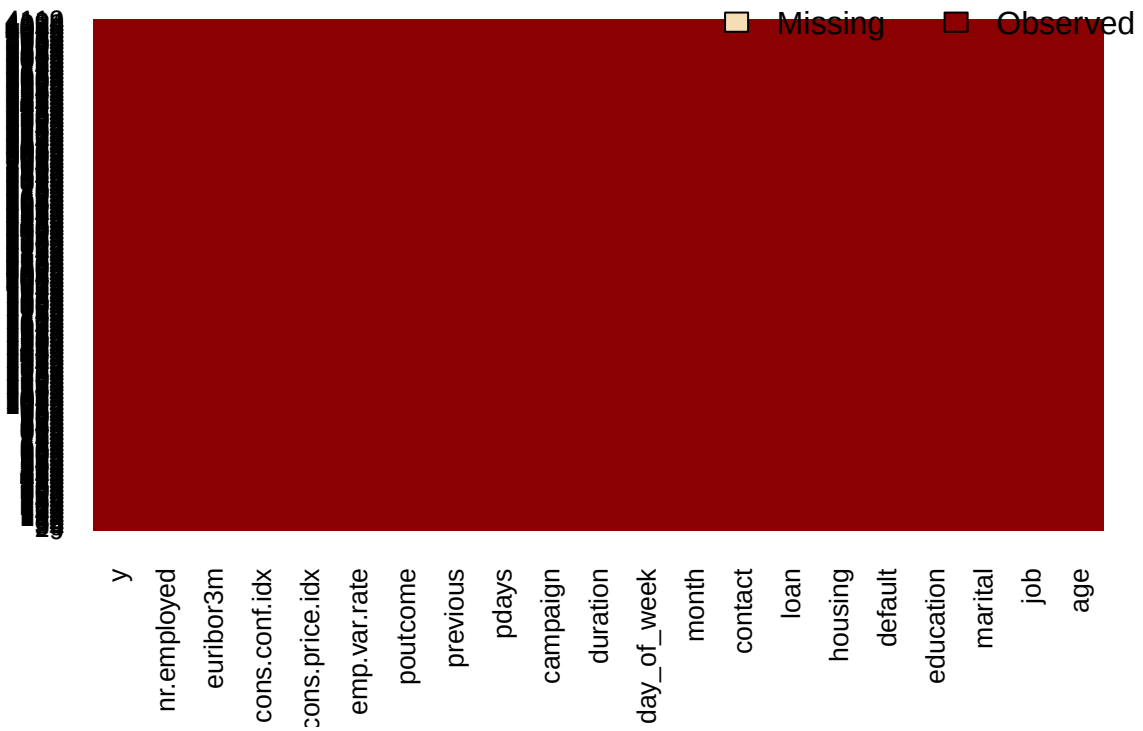
```
#Veri kümesinde eksik değer (NA) var mı? Varsa true, yoksa false döndürür.

anyNA(egit)
```

```
## [1] FALSE
```

```
#Eksik değerlerin görsel olarak incelenmesi

library(Amelia)
missmap(egit, main = "Eksik Degerler - Gozlenen Degerler")
```



Yukarıda görüldüğü üzere veri kümesinde herhangi bir eksik değer yoktur. Artık model oluşturulabilir.

Genelleştirilmiş lineer modeller oluşturmak için `glm` fonksiyonu kullanılır. İncelenen verideki sonuç değişkeni iki seviyeli (y : 'no', 'yes') olduğu için `glm` fonksiyonu `binomial(link='logit')` ailesiyle çağrılır. `summary()` ile modele ilişkin sonuçlar görüntülenir.

Aşağıda veri kümesinde bulunan bütün açıklayıcı değişkenlerin sonuç değişkeniyle ilişkisini gösteren bir model oluşturulmuştur.

```
model <- glm(y ~ ., family = binomial(link = "logit"), data = egit)
summary(model)
```

```
##
## Call:
## glm(formula = y ~ ., family = binomial(link = "logit"), data = egit)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -4.9382  -0.2840  -0.1750  -0.1135   2.8049
##
## Coefficients: (1 not defined because of singularities)
##                                Estimate Std. Error z value Pr(>|z|)
## (Intercept)                  -1.447e+02  1.228e+02  -1.179 0.238448
## age                           8.668e-03  8.137e-03   1.065 0.286774
## jobblue-collar               -2.269e-01  2.706e-01  -0.839 0.401739
## jobentrepreneur              -7.740e-01  4.988e-01  -1.552 0.120738
## jobhousemaid                  2.477e-01  4.462e-01   0.555 0.578854
## jobmanagement                -3.136e-01  2.854e-01  -1.099 0.271806
## jobretired                   -2.069e-01  3.464e-01  -0.597 0.550354
## jobself-employed             -7.305e-01  4.163e-01  -1.755 0.079333 .
```

```
## jobservices                    1.251e-01  2.821e-01   0.443 0.657473
## jobstudent                    -7.367e-02  3.982e-01  -0.185 0.853200
## jobtechnician                  2.209e-01  2.243e-01   0.985 0.324818
## jobunemployed                  3.362e-01  3.912e-01   0.859 0.390076
## jobunknown                    -4.624e-01  7.253e-01  -0.637 0.523827
## maritalmarried                 2.782e-01  2.457e-01   1.132 0.257485
## maritalsingle                  3.277e-01  2.797e-01   1.171 0.241478
## maritalunknown                 2.769e-01  1.145e+00   0.242 0.808802
## educationbasic.6y              3.219e-01  4.077e-01   0.789 0.429827
## educationbasic.9y              2.012e-01  3.214e-01   0.626 0.531321
## educationhigh.school           1.515e-01  3.066e-01   0.494 0.621138
## educationilliterate           -1.144e+01  5.354e+02  -0.021 0.982947
## educationprofessional.course   1.169e-01  3.355e-01   0.348 0.727615
## educationuniversity.degree     3.189e-01  3.088e-01   1.033 0.301752
## educationunknown               2.426e-01  3.879e-01   0.626 0.531612
## defaultunknown                 1.453e-01  2.099e-01   0.692 0.488834
## defaultyes                    -8.818e+00  5.354e+02  -0.016 0.986860
## housingunknown                -5.596e-01  5.220e-01  -1.072 0.283691
## housingyes                    -6.731e-02  1.376e-01  -0.489 0.624697
## loanunknown                           NA         NA      NA       NA
## loanyes                       -1.144e-01  1.870e-01  -0.612 0.540546
## contacttelephone              -9.556e-01  2.784e-01  -3.432 0.000599 ***
## monthaug                       4.943e-01  4.140e-01   1.194 0.232428
## monthdec                       8.877e-01  6.760e-01   1.313 0.189088
## monthjul                       9.392e-02  3.617e-01   0.260 0.795096
## monthjun                       5.517e-01  4.343e-01   1.270 0.203955
## monthmar                       2.474e+00  5.181e-01   4.775  1.8e-06 ***
## monthmay                      -3.228e-01  3.007e-01  -1.074 0.283009
## monthnov                      -3.241e-01  4.188e-01  -0.774 0.439028
## monthoct                       2.817e-01  5.276e-01   0.534 0.593467
## monthsep                       1.540e-01  5.966e-01   0.258 0.796248
## day_of_weekmon                 1.425e-01  2.130e-01   0.669 0.503616
## day_of_weekthu                 1.023e-01  2.153e-01   0.475 0.634629
## day_of_weektue                 4.485e-02  2.189e-01   0.205 0.837627
## day_of_weekwed                 3.126e-01  2.222e-01   1.407 0.159487
## duration                       5.260e-03  2.608e-04  20.167  < 2e-16 ***
## campaign                      -9.988e-02  4.591e-02  -2.175 0.029599 *
## pdays                         -4.959e-04  6.450e-04  -0.769 0.441998
## previous                       1.213e-01  1.714e-01   0.708 0.479198
## poutcomenonexistent            5.792e-01  2.950e-01   1.964 0.049574 *
## poutcomesuccess                1.327e+00  6.353e-01   2.089 0.036692 *
## emp.var.rate                  -8.653e-01  4.651e-01  -1.861 0.062782 .
## cons.price.idx                 1.409e+00  8.108e-01   1.738 0.082244 .
## cons.conf.idx                  6.442e-02  2.633e-02   2.446 0.014432 *
## euribor3m                     -1.610e-01  4.172e-01  -0.386 0.699575
## nr.employed                    2.156e-03  9.923e-03   0.217 0.828036
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 2845.8  on 4118  degrees of freedom
## Residual deviance: 1597.7  on 4066  degrees of freedom
## AIC: 1703.7
```

```
##
## Number of Fisher Scoring iterations: 12
```

3.4. Model Sonuçlarının Yorumlanması

Sapma *(deviance)* modelin ne kadar iyi uyduğunun bir ölçüsüdür. Yüksek sayılar başarısız uyuma işaret eder. **Null deviance** sonuç değişkeninin sadece kesen içeren bir model tarafından ne kadar iyi tahmin edilebildiğini gösterir. Modele değişkenler eklenmesiyle bu sayı düşer ve model daha uygun hale gelir.

AIC birden fazla modelin olduğu durumlarda bir modelin diğerlerine göre üstünlüğünü belirlemeye yardımcı olur. Düşük AIC'ye sahip modeller daha iyidir.

Fisher puanlama algoritması Newton–Raphson metodu üzerinden oluşturulmuş bir formdur ve maksimum olabilirlik problemlerinin sayısal çözümü için kullanılır.[9]

Std. Error ile güven aralıkları belirlenebilir.

$$\text{Güven Aralığı} = \bar{X} \pm Z^* \frac{\sigma}{\sqrt{n}}$$

$\bar{X}$: Nokta tahminidir. *"Estimate"* kolonundaki değerler değişkenlerin katsayılarının nokta tahminlerini verir. Z^* : Talep edilen güven aralığı için Z değeridir. $\frac{\sigma}{\sqrt{n}}$: Standart hatadır.

Örneğin, *age* değişkeninin katsayısı için %95 güven aralığı hesaplansın. %95 güven aralığı için $Z^* = 1.96 \approx 2$ kabul edilebilir. $0.008668 \pm 2(0.008137)$ işlemi sonucunda *age* değişkeninin katsayısı için %95 güven aralığı $(-0.0076\ ,\ 0.0249)$ olarak bulunur.

- Yaşın müşterilerin kararı (evet/hayır) üzerindeki etkisi yok denecek kadar azdır. Yaştaki 1 birimlik artış lojiti 0.0087 artırır.
- Mesleğin sonuç üzerindeki etkisi incelenirken *"admin"* referans olarak belirlenmiştir. Meslek değişkeninin müşterilerin kararı üzerindeki etkisi oldukça zayıftır. İstatistiksel olarak sadece serbest meslek sahipleri *(jobself-employed)* önemli sayılabilir. Referans grubu ile kıyaslandığında temizlik görevlileri, hizmet sektöründekiler, teknisyenler ve işsizler sonuç üzerinde pozitif etkiye; diğer meslek dalları ise negatif etkiye sahiptir.
- Medeni durumun sonuç değişkeni üzerindeki etkisi incelenirken *"divorced"* referans olarak belirlenmiştir. Medeni durum istatistiksel olarak önemli değildir. Evliler, bekarlar ve medeni hali bilinmeyenler boşanmış olanlara göre zayıf derecede pozitif etkiye sahiptir.
- Eğitimin müşteri cevabı üzerindeki etkisine bakıldığında *"basic.4y"* referans olarak kabul edilmiştir. Diğer tüm koşullar eşitken okur yazar olmayan kesimin hayır cevabı verme olasılığı, ilkokul dördüncü sınıf mezunlarına göre daha fazladır *(katsayı: -11.44)*. Diğer eğitim seviyeleri, dört senelik eğitim seviyesine göre zayıf pozitif etkiye sahiptir. Ancak eğitim istatistiksel açıdan önemli bir değişken değildir.
- Kredi borcunun müşteri kararı üzerindeki etkisi analiz edilirken *"no"* kategorisi referans olarak belirlenmiştir.Kredi borcu olanların kredi borcu olmayanlara göre vadeli mevduatı reddetme olasılığı daha fazladır. İstatistiki olarak önemli değildir.
- Konut kredisi sahipliğinin sonuç üzerindeki etkisine bakılırken konut kredisi sahibi olmama *("no")* durumu referans olarak belirlenmiştir. Konut kredisi sahibi olanların vadeli mevduatı kabul etmeme olasılığı konut kredisi sahibi olmayanlara göre eşit koşullar altında zayıf bir etkiyle daha fazladır. İstatistiksel açıdan önemli değildir.
- Bireysel kredi kullanımının sonuç üzerindeki etkisi incelenirken bireysel kredi kullanmama durumu referans olarak belirlenmiştir. Bireysel kredi kullanan müşterilerin hayır cevabı verme olasılığı kredi kullanmayan müşterilere göre zayıf bir etkiyle daha fazladır. Sonuçlarda *"loanunknown"* etiketine ait değerlerde `NA` yazısı yer almaktadır. `Coefficients: (1 not defined because of singularities)`

uyarısı ile birlikte buradan, açıklayıcı değişkenlerden birinin diğer değişkenlerin lineer kombinasyonuyla ifade edilebileceği sonucu çıkarılabilir. Başka bir ifadeyle değişkenler arasında lineerlik mevcuttur.

- İletişim çeşidinin sonuç üzerindeki etkisinin kesinliği istatistiksel açıdan kuvvetlidir. Cep telefonu üzerinden yapılan görüşmeler referans alınmıştır. Referans ile karşılaştırıldığında karasal hat üzerinden yapılan görüşmelerde yanıtın olumsuz olma ihtimali daha fazladır.
- Ayların etkisi incelenirken nisan referans olarak seçilmiştir. Mart ayı hakkındaki sonuçların kesinliği oldukça fazladır ve referans ile karşılaştırıldığında sonuç değişkenine etkisi de diğer aylara göre daha fazladır. Ayrıca mart ayı referansa göre pozitif etkilidir.
- Günlerin etkisine bakıldığında cuma referans olarak seçilmiştir. Pazartesi, salı, çarşamba ve perşembe günleri cuma gününe göre sonuç üzerinde zayıf pozitif bir etkiye sahiptir. Gün değişkeni istatistiksel açıdan önemli sayılmaz.
- Müşteri ile yapılan telefon konuşması süresinin sonuç üzerindeki etkisinin kesinliği epey yüksektir. Süredeki 1 saniyelik artış lojiti 0.00526 artırır.
- Kampanya süresince müşteri ile kurulan iletişim sayısının artışı müşterinin evet deme olasılığını zayıf olarak da olsa olumsuz yönde etkilemektedir. İstatistiksel açıdan *"campaign"* önemlidir.
- Müşteri ile bir önceki kampanyadaki en son görüşmenin üzerinden geçen gün sayısının sonuca etkisi yok denecek kadar azdır. Günün 1 artması lojiti 0.0004959 düşürür. İstatistiksel açıdan pek önemli değildir.
- Kampanyadan önce müşteriyle yapılan toplam görüşme sayısının sayıca fazla olması sonuç üzerinde zayıf pozitif bir etki yaratır. Bu değişken de istatistiksel açıdan pek önemli değildir.
- Önceki kampanyanın başarısızlığa uğraması referans kabul edilmiştir. Buna göre diğer koşulların eşit olması durumunda bir önceki kampanyaya olumlu katılım gösteren kişilerin olumsuz katılımcılara göre "evet" deme olasılığı daha yüksektir. Bir önceki kampanyaya gösterdiği eğilim hakkında bilgi sahibi olunmayan müşteriler de olumsuz yanıt verenlere göre sonuç üzerinde pozitif etkiye sahiptir. Bu değişkenin sonuç üzerindeki etkisinin kesinliği istatistiksel olarak kuvvetlidir.
- İstihdam varyasyon oranındaki artış sonucu negatif etkiler. İstatistiksel olarak önemli bir değişkendir.
- Tüketici fiyatları endeksinin artışı sonucu pozitif etkiler. Etkisinin kesinliği istatistiksel olarak kuvvetlidir.
- Tüketici güven ekdeksinin artışı sonuç üzerinde zayıf pozitif bir etki yaratır. İstatistiksel olarak önemli bir değişkendir.
- Hesaplanan faiz oranının yükselmesi sonuç üzerinde zayıf negatif bir etki yaratır. İstatistiksel olarak önemli bir değişken değildir.
- Çalışan sayısındaki artış sonuç üzerinde oldukça zayıf pozitif etkiye sahiptir. İstatistiksel olarak önemli değildir.

İstatistiksel olarak anlamlı değişkenler kullanılarak başka bir model oluşturulsun:

```
model2 <- glm(y ~ contact + month + duration + campaign +
          poutcome + emp.var.rate + cons.price.idx + cons.conf.idx,
          family = binomial(link = "logit"), data = egit)
summary(model2)
```

```
##
## Call:
## glm(formula = y ~ contact + month + duration + campaign + poutcome +
##     emp.var.rate + cons.price.idx + cons.conf.idx, family = binomial(link = "logit"),
##     data = egit)
##
## Deviance Residuals:
```

```
##     Min       1Q   Median       3Q      Max
## -5.0031  -0.2842  -0.1774  -0.1227   2.7880
##
## Coefficients:
##                       Estimate Std. Error z value Pr(>|z|)
## (Intercept)         -1.341e+02  1.594e+01  -8.416  < 2e-16 ***
## contacttelephone    -9.395e-01  2.497e-01  -3.762 0.000168 ***
## monthaug             6.136e-01  3.574e-01   1.717 0.086014 .
## monthdec             9.200e-01  6.070e-01   1.516 0.129620
## monthjul             1.086e-01  3.435e-01   0.316 0.751995
## monthjun             5.432e-01  3.080e-01   1.764 0.077785 .
## monthmar             2.465e+00  4.237e-01   5.817 5.99e-09 ***
## monthmay            -3.308e-01  2.725e-01  -1.214 0.224762
## monthnov            -3.970e-01  3.362e-01  -1.181 0.237637
## monthoct             3.081e-01  4.228e-01   0.729 0.466180
## monthsep             2.079e-01  4.335e-01   0.480 0.631561
## duration             5.171e-03  2.548e-04  20.291  < 2e-16 ***
## campaign            -1.019e-01  4.535e-02  -2.248 0.024570 *
## poutcomenonexistent  4.002e-01  1.977e-01   2.025 0.042901 *
## poutcomesuccess      1.810e+00  2.677e-01   6.761 1.37e-11 ***
## emp.var.rate        -9.338e-01  7.486e-02 -12.473  < 2e-16 ***
## cons.price.idx       1.411e+00  1.724e-01   8.184 2.74e-16 ***
## cons.conf.idx        5.744e-02  1.743e-02   3.295 0.000985 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 2845.8  on 4118  degrees of freedom
## Residual deviance: 1620.0  on 4101  degrees of freedom
## AIC: 1656
##
## Number of Fisher Scoring iterations: 7
```

Yukarıda görüldüğü üzere bu modelden istatistiki açıdan daha kesin sonuçlar elde edilmiştir. Değişkenlere ait hesaplanmayan değerler de yoktur. Ayrıca AIC değerinin daha düşük olması ve serbestlik derecesindeki 17 kayıpla sapmanın 1225.8 azaltılması da bu modelin önceki modele göre daha uygun bir model olduğunu gösterir.

3.5. Model Ne Kadar Başarılı?

Modelin başarısını test etmek için ilk olarak test kümesi oluşturulmalıdır. Bu işlem, büyük veri kümesinden eğitim için kullanılan küçük veri kümesinin çıkarılmasıyla yapılacaktır.`dplyr` kütüphanesinde bulunan `setdiff(data, egit)`, "data" kümesinde görünüp "egit" kümesinde görünmeyen satırları çalıştırır. Böylece *"test"* adında yeni bir veri kümesi oluşturulur.

```
library(dplyr)
test <- setdiff(data,egit)
pdata <- predict(model2, newdata = test, type = "response")
head(pdata)
```

```
##           1           2           3           4           5           6
## 0.014647372 0.008261313 0.012252234 0.008346476 0.018507857 0.010618277
```

Görüldüğü üzere buradaki `predict()` kurulan modele göre "test" kümesindeki y değerlerinin olasılıklarını

(0,1) aralığında olacak şekilde tahmin eder. y değişkeni, sıfıra karşılık gelen "no" ve bire karşılık gelen "yes" sınıflarına sahip olduğu için tahmin edilen bu sayısal değerler de iki sınıfta toplanmalıdır. Bu yüzden olasılığı 0.5'ten küçük olanlar "no", 0.5'ten büyük olanlar ise "yes" olarak kategorilendirilecektir. Böylece gerçek sınıflar ile tahmin edilen sınıflar arasındaki örtüşme kontrol edilebilir.

Bu örtüşme **hata matrisi** aracılığıyla incelenebilir.

```
library(caret)
confusionMatrix(data = as.vector(ifelse(pdata > 0.5,"yes","no")), reference = test$y, positive = "yes")
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    no   yes
##        no  31943  2406
##        yes   926  1782
##
##                Accuracy : 0.9101
##                  95% CI : (0.9071, 0.913)
##     No Information Rate : 0.887
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.4698
##  Mcnemar's Test P-Value : < 2.2e-16
##
##             Sensitivity : 0.42550
##             Specificity : 0.97183
##          Pos Pred Value : 0.65805
##          Neg Pred Value : 0.92995
##              Prevalence : 0.11302
##          Detection Rate : 0.04809
##    Detection Prevalence : 0.07308
##       Balanced Accuracy : 0.69866
##
##        'Positive' Class : yes
##
```

Hata matrisi gerçek değerler ile tahmini değerler arasında oluşturulmuş 2x2 boyutlarında bir matristir. Dört bileşenden oluşur:[10]

- **Doğru negatif (DN):** Gerçek karşılığı "hayır" olup "hayır" olarak tahmin edilenlerdir. 31943 doğru negatif vardır.
- **Doğru pozitif (DP):** Gerçek karşılığı "evet" olup "evet" olarak tahmin edilenlerdir. 1782 doğru pozitif vardır.
- **Yanlış negatif (YN):** Gerçek karşılığı "evet" olmasına rağmen "hayır" olarak tahmin edilen 2406 değer vardır.
- **Yanlış pozitif (YP):** Gerçek karşılığı "hayır" olmasına rağmen "evet" olarak hesaplanan 926 değer vardır.

37057 durumun 34349 tanesi "hayır", 2708 tanesi "evet" olarak tahmin edilmiştir. Ancak aslında 32869 tane "hayır", 4188 tane "evet" cevabı bulunmaktadır. Ayrıca 37057 durumun 33725 tanesi isabetli tahmindir.

Sonuçlar incelendiğinde dengesiz bir dağılım olduğu görülür. "Evet" olduğu halde yanlış tahmin edilen "evet" sayısı doğru tahmin edilenlerden fazladır ($2406 > 1782$). Daha dengeli bir dağılım için olasılığı 0.25'ten büyük olanlar "yes", küçük olanlar ise "no" olarak sınıflandırılsın.

```r
confusionMatrix(data = as.vector(ifelse(pdata > 0.25,"yes","no")), reference = test$y, positive = "yes")
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    no   yes
##        no  30686  1330
##        yes  2183  2858
##
##                Accuracy : 0.9052
##                  95% CI : (0.9022, 0.9082)
##     No Information Rate : 0.887
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.5657
##  Mcnemar's Test P-Value : < 2.2e-16
##
##             Sensitivity : 0.68243
##             Specificity : 0.93358
##          Pos Pred Value : 0.56695
##          Neg Pred Value : 0.95846
##              Prevalence : 0.11302
##          Detection Rate : 0.07712
##    Detection Prevalence : 0.13603
##       Balanced Accuracy : 0.80801
##
##        'Positive' Class : yes
##
```

`confusionMatrix()` işlemi sonucunda ortaya çıkan önemli bazı terimler şunlardır:

- **Accuracy:** Sınıflayıcının doğruluk oranıdır. $\frac{DP+DN}{Toplam}$ ile hesaplanır.
- **Kappa:** Cohen'in Kappa katsayısı iki değerleyici arasındaki uyumun güvenilirliğini ölçer. En büyük değer olan 1'e yaklaştıkça iki değerleyici arasındaki uyuşma artar.
- **Sensitivity:** Sınıflayıcının hassasiyetlik oranını belirtir. Pozitif değerlerin sınıflayıcı tarafından ne kadar iyi tanınabildiğini gösterir. $\frac{DP}{Pozitif} = \frac{DP}{YN+DP}$ ile hesaplanır.
- **Specificity:** Negatif örneklerin sınıflayıcı tarafından ne kadar iyi tanınabildiğini gösterir. $\frac{DN}{Negatif} = \frac{DN}{DN+YP}$ ile hesaplanır.
- **Pos Pred Value:** Gerçek pozitiflerin bütün pozitif tahmin edilenlere oranıdır.
- **Neg Pred Value:** Gerçek negatiflerin bütün negatif tahmin edilenlere oranıdır.
- **Prevalence:** Pozitif örneklerin yaygınlığını gösterir, gerçek pozitif örneklerin toplam örnek sayısına oranıdır. Yani test kümesindeki "evet" oranını belirtir.
- **Detection Rate:** Doğru pozitif tespitlerin toplam örnek sayısına oranıdır. $\frac{DP}{DP+DN+YN+YP}$ ile hesaplanır.
- **Detection Prevalence:** Pozitif olarak tahmin edilenlerin toplam örnek sayısına oranıdır. $\frac{DP+YP}{DP+DN+YN+YP}$ ile hesaplanır.
- **Balanced Accuracy:** $\frac{sensitivity\ +\ specificity}{2}$ formülü ile hesaplanır. Veri kümesinin dengesiz olduğu durumlarda dengelenmiş doğruluğu dikkate almak daha iyi olabilir.[11]

Buna göre yukarıdaki iki hata matrisi sonucu karşılaştırıldığında olasılığı 0.25'ten büyük olanları "yes" olarak kategorilendirmenin bu örnekte daha uygun olacağı söylenebilir.

4. Karar Ağaçları

4.1. Giriş

Karar ağaçları anlaşılmasının ve yorumlanmasının kolay olması, veri tabanları ile entegrasyonunun kolaylığı, güvenilirliklerinin iyi olması gibi sebeplerden dolayı popüler tekniklerden biridir.[7]

Karar ağaçları, veri madenciliğinde hem sınıflandırmada hem de regresyonda kullanılabilecek bir tahmin yöntemi iken yöneylem araştırmalarında hiyerarşik bir karar modelini ve sonuçlarını ifade eder. Bir karar ağacı sınıflandırma görevlerinde kullanılıyorsa *sınıflandırma ağacı*, regresyon görevlerinde kullanılıyorsa *regresyon ağacı* adını alır.[8] Bu çalışmada sınıflandırma ağaçlarından bahsedilecektir.

Sınıflandırma ağaçları bir nesneyi veya durumu niteliklerine göre önceden tanımlanmış olan sınıflara ayırmak için kullanılır. Örneğin, bu çalışmada müşterilerin kararlarını yaş, meslek, konuşma süresi vb. özellikler yardımıyla evet/hayır sınıflarında toplamak için sınıflandırma ağaçları kullanılacaktır.

Bir sınıflandırma ağacı iç düğümlerden *(test düğümleri)* ve yapraklardan *(karar düğümleri)* oluşur. Her bir iç düğüm, bağımsız bir değişkeni temsil eder ve örnek uzayı iki veya daha fazla alt uzaya böler. Her bir yaprak sonuç çıktısının değerini gösterir.[8]

Karar ağacı tekniği kullanılarak sınıflandırma, *öğrenme* ve *sınıflama* olmak üzere iki basamaklı bir işlemdir. Ağacın öğrenilmesi sırasında üzerinde eğitim yapılan kümenin çeşitli özelliklere göre alt kümelere bölünmesi öz yineli olarak devam eder ve tekrarlama işleminin tahmin üzerinde bir etkisi kalmayana dek sürer. Bu işlem *öz yineli parçalama* olarak isimlendirilir. Öğrenilen model, sınıflama kuralları veya karar ağacı olarak gösterilir. Sınıflama aşamasında ise test kümesi, belirlenen sınıflama kuralarının veya karar ağacının doğruluğunun tespiti için kullanılır. Kabul edilebilir oranda doğruluk mevcutsa eğitimde belirlenen kurallar yeni verilerin sınıflandırılması amacıyla kullanılabilir.

4.2. Model Oluşturma

Karar ağacı oluşturmak için `party` kütüphanesindeki `ctree()` kullanılabilir.. Oluşan grafik oldukça büyük olduğu için düzgün görüntülenebilmesi amacıyla boyutları ayarlanmış olan `.png` uzantılı dosyaya çizdirilmiştir.

```
library(party)
claTree <- ctree(y ~ ., data = egit)
png("clatree.png", res=80, height=1000, width=2000)
plot(claTree, type = "simple")
dev.off()
```

4.3. Model Sonuçlarının Yorumlanması

- Müşteri ile kurulan iletişimin süresi 616 saniyenin üzerine çıktığında sonucun başarılı olma olasılığı 0.52'dir.
- 616 saniyeden kısa süren konuşmalarda müşterinin iş yerindeki çalışan sayısı 5076.2'den fazlaysa ve müşteri mart ya da ekim aylarında aranmışsa sonucun başarılı olma ihtimali 0.54'tür. Nisan ayında yapılan görüşmelerde ise sonuç büyük olasılıkla (0.9) olumsuz olmaktadır.
- Çalışan sayısının 5076.2'den fazla olduğu; görüşmenin 310 saniye veya daha kısa sürdüğü; iletişim kurulan ayın ağustos, aralık, temmuz, haziran, mayıs veya kasım olduğu; önceki kampanyanın başarıya ulaşıp ulaşmadığının bilinmediği durumlarda sonuç olumsuz olmuştur. Modele katılan 4119 örnekten 2326 tanesi bu kategoriye girmektedir.

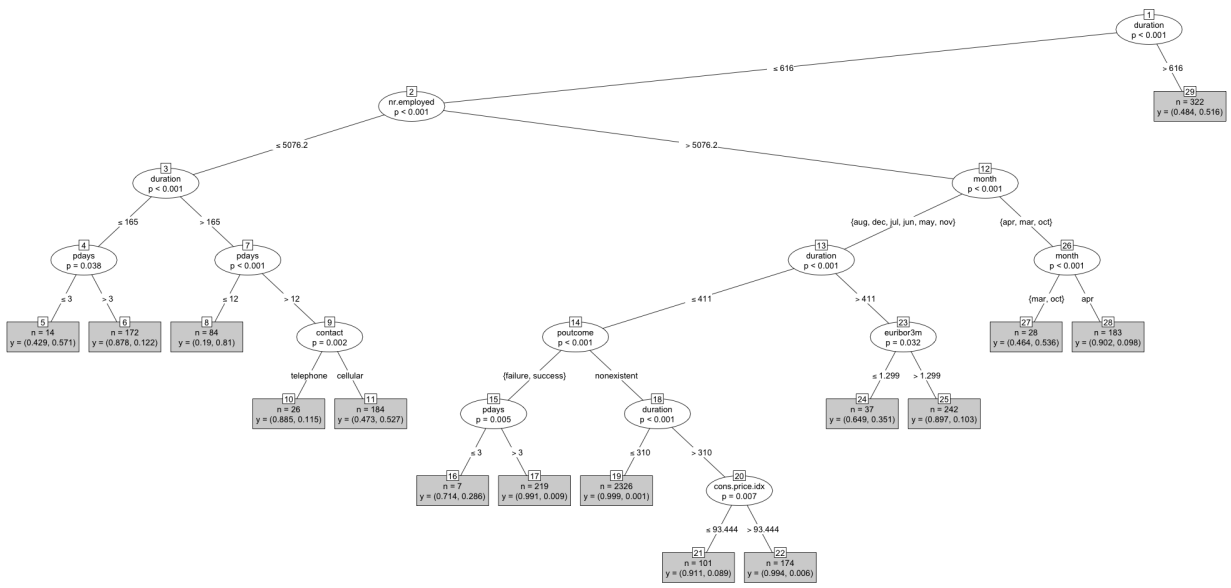


Figure 2:

4.4. Model Ne Kadar Başarılı?

`predict()` ile `claTree` isimli modele uygun olarak "test" kümesindeki bağımlı değişken olan y'nin sınıfları tahmin edilir ve "hesaplanan" isimli değişkene atanır. `confusionMatrix()`, sonuç değişkeninin hesaplanan ve gerçek karşılıklarının sayıca karşılaştırılması için bir hata matrisi oluşturur. `positive = "yes"` sonuç değişkenindeki *"yes"* sınıfının yapılacak işlemlerde pozitif kısım olarak hesaplatılmasını sağlatır.

```
hesaplanan <- predict(claTree, newdata  = test)
confusionMatrix(hesaplanan, test$y, positive = "yes")
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    no   yes
##        no  30284  1154
##        yes  2585  3034
##
##                Accuracy : 0.8991
##                  95% CI : (0.896, 0.9022)
##     No Information Rate : 0.887
##     P-Value [Acc > NIR] : 3.807e-14
##
##                   Kappa : 0.562
##  Mcnemar's Test P-Value : < 2.2e-16
##
##             Sensitivity : 0.72445
##             Specificity : 0.92135
##          Pos Pred Value : 0.53995
##          Neg Pred Value : 0.96329
##              Prevalence : 0.11302
##          Detection Rate : 0.08187
```

```
##    Detection Prevalence : 0.15163
##       Balanced Accuracy : 0.82290
##
##        'Positive' Class : yes
##
```

- 30284 **doğru negatif** vardır.
- 3034 **doğru pozitif** vardır.
- 1154 **yanlış negatif** vardır.
- 2585 **yanlış pozitif** vardır.

Modele göre 37057 durumun 31438 tanesi "hayır", 5619 tanesi "evet" olarak tahmin edilmiştir. Gerçekte 32869 tane "hayır", 4188 tane "evet" cevabı bulunmaktadır.

5. Sonuç

Portekiz bankacılık kurumunun pazarlama kampanyalarında müşteriler, vadeli mevduatı seçip seçmeme konusunda sınıflandırılmak istenmiştir. Bunun için lojistik regresyon ve karar ağaçları yöntemleri kullanılmıştır. Doğruluk, dengelenmiş doğruluk, duyarlık gibi sonuçlar karşılaştırıldığında her iki modelin de benzer sonuçlar sergilediği görülmektedir. Modelin, oluşan görsele bakılarak kolay yorumlanabilirliği konusunda karar ağaçları öne çıkmaktadır. Regresyonda ise değişkenlerdeki her bir sınıfın etkisi detaylı olarak gözlenebilmektedir.

Kaynakça

1. Alpaydin, Ethem. (2010). *Introduction to machine learning* (s.9). MIT Press.
2. Moro, S., Cortez, P. & Rita, P. (2014). *A data-driven approach to predict the success of bank telemarketing.* Decision Support Systems, Elsevier, 62:22-31
3. Dua, D. and Karra Taniskidou, E. (2017). UCI Machine Learning Repository [http://archive.ics.uci.edu/ml]. Irvine, CA: University of California, School of Information and Computer Science.
4. Sümbüloğlu, K. (2014). *Lojistik regresyon analizi.* http://78.189.53.61/- /bs/ess/k_sumbuloglu.pdf
5. Hastie, T., Tibshirani, R. & Friedman, J. (2009). *The elements of statistical learning: data mining, inference, and prediction* (2. Baskı, ss. 102, 119-120). New York: Springer.
6. Odds, odd Ratios, probabilities and the logit. (2009, 27 Eylül). https://www.colorado.edu/economics/morey/7818/probth
7. Çalış, A., Kayapınar, S. & Çetinyokuş, T. (2014). Veri madenciliğinde karar ağacı algoritmaları ile bilgisayar ve internet güvenliği üzerine bir uygulama. *Endüstri Mühendisliği Dergisi*, *25*(3-4), 2-19.
8. Rokach, L. & Maimon, O. (2014). *Data mining with decision trees: theory and applications* (2. Baskı, ss. 10, 12-13). Singapur: World Scientific.
9. Lillis, D. (b.t.). *Generalized linear models in R, part 2: understanding model fit in logistic regression output.* https://www.theanalysisfactor.com/r-glm-model-fit/
10. Fawcett, T. (2006, Haziran). An introduction to ROC analysis. *Pattern Recognition Letters*,*27*(8), 861-874.
11. Brodersen, K. H., Ong, C. S., Stephan, K. E. & Buhmann, J. M. (2010). The balanced accuracy and its posterior distribution. *Proceedings of the 20th International Conference on Pattern Recognition.*