Bitirme Tezi

May 19, 2018

1 Optimizasyon Problemi

Optimizasyon problemi kısaca, bir problemin istenen şartlar altında en iyi sonucunun (minimum veya maksimum olarak) belirlenmesi için kullanılır. Sınıflandırma yapmak için kullanılacak olan Lojistik Regresyon (Logistic Regression), Destek Vektör Makinesi (Support Vector Machine) ve Karar Ağaçları (Decision Trees) teknikleri temelde belli optimizasyon problemlerini baz alarak işlem yapar.

SVM'de datayı ikiye bölen optimal hiperdüzlem bulunmaya çalışılırken optimizasyon probleminden faydalanır. Belirtilen denklemlerden hareketle bazı kısıtlar oluşturarak en iyi hiperdüzlemi bulmak için kurulan optimizasyon problemi şu şekilde ifade edilebilir:

min $||\overrightarrow{w}||^2$

such that

- $\overrightarrow{w}.\overrightarrow{x}\_\{pos\} \geq \gamma + 1$ for + examples
- $\overrightarrow{w}.\overrightarrow{x}\_\{neg\} \leq \gamma - 1$ for - examples

[1]

Lojistik Regresyon'da aynı SVM'de olduğu gibi, verilen veri noktalarını iki parçaya bölen optimal hiperdüzlem bulunmaya çalışılır. Ancak bunu yaparken maksimum olabilirlik (maximum likelihood) modeli kullanılır.

Karar ağaçlarında ise optimal bölünmeleri bulmak, ideal düğüm noktalarını belirlemek için optimizasyon problemi kullanılır. [2]

Lojistik Regresyon'da katsayıların bulunması için maksimize edilen denklem olabilirlik denklemidir:

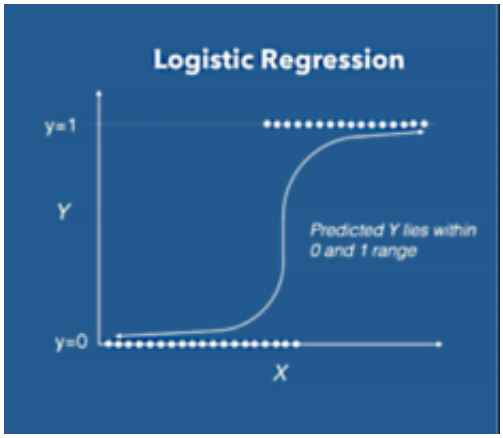
$$\ell(\beta) = \sum_{i=1}^{N} \{y_i \beta^T x_i - log(1 + e^{\beta^T x_i})\}$$

[3] (Hastie ve diğerleri, 2009, s. 120)

Destek Vektör Makinesi'nde optimize edilen denklem ise:

$$C \sum_{i}^{N} \max(0, 1 - y_i f(x_i)) + ||\overrightarrow{w}||^2$$

[4]



2 Lojistik Regresyon

Regresyon analizi, herhangi bir değişkenin bir veya birden fazla değişkenle arasındaki ilişkinin matematiksel fonksiyon şeklinde ifade edilmesidir. Tek bağımsız değişkenin kullanıldığı regresyon "tek değişkenli regresyon analizi", birden fazla bağımsız değişkenin kullanıldığı regresyon analizi de "çok değişkenli regresyon analizi" olarak adlandırılır.

Regresyon analizi; bağımlı ve bağımsız değişkenler arasında bir ilişkinin olup olmadığını, eğer bir ilişki varsa bu ilişkinin gücünü, bağımlı değişkene ait ileriye dönük değerleri tahmin etmenin mümkün olup olmadığını ve tahminin nasıl gerçekleştirileceğini, belirli koşullar altında özel bir değişken veya değişkenler grubunun diğer değişkenler üzerindeki etkisinin nasıl değişeceğini belirlemek için kullanılır.

Lojistik regresyonda bağımsız değişkenin değeri bilinerek bağımlı değişkenin meydana gelme olasılığı tahmin edilmeye çalışılarak analiz gerçekleştirilir. İkili sınıflandırma problemleri olarak adlandırılan problemleri modellemek ve çözmek için kullanılabilir. R'de lojistik regresyon "glm(formül, data, family = binomial("logit"))" şeklinde yapıya sahip tek bir glm() fonksiyonu sayesinde kolayca gerçekleştirilir. [5]

Lojistik regresyon, bir sonucu belirleyen bir veya daha fazla bağımsız değişken bulunan bir veri kümesini analiz etmek için kullanılan sınıflandırma algoritmasıdır. Sonuç, ikili kategorik olarak ölçülür yani yalnızca iki olası sonuç (1 veya 0) vardır. Analiz, bağımsız değişkenler ve bağlantı fonksiyonu arasında doğrusal bir ilişkinin olduğu varsayımıyla gerçekleşir.

Lojistik regresyonun amacı, bağımlı değişken ile ilgili bir dizi bağımsız değişken arasındaki ilişkiyi tanımlamak için en uygun modeli bulmaktır. Modeller kurulurken ilgili bağımlı değişkenlerin varlığının olasılığını tahmin etmek için bir formülün katsayıları üretilir. [6]

2.1 Katsayıların Tahmini:

Bilinmeyen parametrelerin tespiti için doğrusal regresyonda genellikle "En Küçük Kareler Yöntemi" kullanılır. Bu yöntemin amacı, aşağıdaki formülde yer alan β_0 ve β_1 parametrelerini yani bağımsız değişkenin regresyon katsyıları Y'nin yani bağımlı değişkenin gözlenen değerlerinin tahmin edilen değerlerden uzaklaştığı miktarların kareler toplamını minimum yapmaktır. Bu yöntemin doğrusal regresyonda başarılı bir şekilde işlemesine rağmen lojistik regresyon analizinde en küçük kareler yöntemi ile doğru sonuçlar elde edilememektedir. Lojistik regresyon modelinde ise parametrelerin tahmini için çoğunlukla en çok olabilirlik yöntemi kullanılır. Maksimum olasılık şu şekilde işlem yapar: Katsayıları, tahmin edilen olasılıklar ve gözlemlenen olasılıklara mümkün olduğunca yakın bulmaya çalışır. Bu fonksiyon gözlenen verinin olasılığını hesaplar.

Parametrelerin bulunan en çok olabilirlik tahminleri, fonksiyonu maksimum yapan değerlerdir. Böylece, belirlenen tahmin ediciler, gözlenen verilere çok yakın değerlere sahiptir yani 1 veya 0'a en yakındır. [7] [8]: (Kaşko, 2007, s.27)

2.2 Lojistik Regresyon Modelinin Kurulması

Basit doğrusal regresyon modeli;

$$Y = \beta_0 + \beta_1 X + \varepsilon$$

Çoklu doğrusal regresyon modeli ise;

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + ... + \beta_k X_k + \varepsilon$$

ile ifade edilir. Burada;
Y: Bağımlı değişkeni
$X_1, X_2,..., X_k $: Bağımsız değişkenleri
β_0: Bağımsız değişkenler sıfır değerini aldığında bağımlı değişkenin değerini, yani sabiti
$\beta_1, \beta_2, ..., \beta_k$: Bağımsız değişkenlerin regresyon katsayılarını
ε: Hata terimini
k: Bağımsız değişken sayısını göstermektedir.
Basit ve çoklu doğrusal regresyon modelinde, bağımlı değişkenin, verilen bağımsız değişkenlerin değerlerine göre beklenen değeri (ortalama değeri) basit doğrusal regresyon modeli için;

$$E(Y|X) = \beta_0 + \beta_1 X_1$$

çoklu doğrusal regresyon modeli için ise;

$$E(Y|X_1, X_2, ..., X_k) = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + ... + \beta_k X_k$$

Lojistik regresyon modeli, genel doğrusal modellerin binom dağılımlı bağımlı değişkenler için elde edilmiş olan özel bir biçimidir. Lojistik fonksiyonun 0 ile 1 arasında bir değişim aralığına sahip olması lojistik fonksiyonun tercih edilmesindeki önemli nedenlerden biridir.
İncelenen bir olayın olasılığının kendi dışında kalan diğer olayların olasılığına oranına "odds değeri" denilmekte ve şu şekilde hesaplanmaktadır:

$$\text{Odds Değeri} = \frac{p}{1-p} = e^{\alpha+\beta_x}$$

$$\text{odds} = \frac{p}{1-p} = \frac{\text{karakteristik varlığı olasılığı}}{\text{karakteristik yokluğu olasılığı}}$$

Burada P üzerinde durulan olayın olasılığını, 1-P ise üzerinde durulmayan olayın olasılığını göstermektedir. Odds değeri 0 ile $+\infty$ arasında değerler almaktadır. İncelenen iki farklı olayın odds değerlerinin birbirine oranına "odds oranı (odds ratio, OR)" denilmektedir. Odds oranı, incelenen iki olayın gözlenme olasılıklarından birinin diğerine oranla kaç kat daha fazla veya kaç kat daha az olarak ortaya çıkabileceğini göstermekte olup şu eşitlik ile hesaplanmaktadır:

$$OR = \frac{e^{\alpha} + \beta}{e^{\alpha}} = e^{\beta}$$

İncelenen bir olasılığın (P), odds değerinin doğal logaritması lojit fonksiyon olarak adlandırılır.

$$logit(p) = ln\frac{p}{1-p}$$

İncelenen olasılığın odds değeri 0 ile $+\infty$ arasında değer alırken aynı olasılığın lojit değeri $-\infty$ ile $+\infty$ arasında değerler alabilmektedir.

$$= log\frac{p(y=1)}{1-(p=1)} = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + ... + \beta_m X_m$$

Olasılıkların lojit fonksiyonunun kullanılmasının amacı, doğrusal bir model elde edilerek, parametre tahminlerinin yapılmasıdır. İncelenen bir olasılığın (P) lojit değeri doğrusal modele eşitlendiğinde;

$$logit(p) = b_0 + b_1 X_1 + b_2 X_2 + b_3 X_3 + ... + b_k X_k$$

elde edilir. Elde edilen bu eşitlik, basit ve çoklu doğrusal regresyon modellerindeki bağımlı değişkenin beklenen değerini veren ve parametre tahminlerinde kullanılan eşitliğe benzer bir eşitliktir.

Son iki eşitlikten;

$$p(x) = \frac{e^{\beta_0+\beta_1 x_1+\beta_2 x_2+...+\beta_n x_p}}{1+e^{\beta_0+\beta_1 x_1+\beta_2 x_2+...+\beta_k x_k}}$$

elde edilir. Bu eşitliğe "lojistik regresyon modeli" denir. Parametreler değerlerin gözlem olasılığını en yüksek yapacak şekilde maksimum olabilirlik yöntemi ile seçilmiş olur.

P : İncelenen olayın gözlenme olasılığını

β_0 : Bağımsız değişkenler sıfır değerini aldığında bağımlı değişkenin değerini, yani sabiti,

$\beta_1, \beta_2, \ldots, \beta_n$: Bağımsız değişkenlerin regresyon katsayılarını,

$X_1, X_2, \ldots, X_n$: Bağımsız değişkenleri

p : Bağımsız değişken sayısını

e = 2,718 sayısını göstermektedir. [9]

2.3 (*) İşaretinin Anlamı:

Bir regresyon tablosundaki bir yıldız işareti "p <.1" anlamına gelmektedir. İki yıldız işareti "p <.05"; ve üç yıldız işareti "p <.01" anlamına gelmektedir. Regresyon tablosundaki yıldız işaretleri, bir regresyon katsayısının istatistiksel önem düzeyini göstermektedir. Örneğin, p <.05. bir sonuç elde etme olasılığı 0.05 veya daha düşüktür. Başka bir deyişle 100 örnekten 95'inin pozitif, 1 sonucuna sahip olduğunu belirtir. Yani 0.05 in güveni %95, 0.01 in güveni %99'dur. Bu yüzden daha iyi bir sonuç almak için sonraki modelleri yıldızlı değişkenleri alarak yeniden kurarız. Sadece üç yıldızlıları aldığımız model gerçeğe en yakın tahminde bulunmalıdır. [10]

2.4 Lojistik Regresyon Modelinin Uygunluğunun ve Doğruluğunun Değerlendirilmesi:

Lojistik Regresyon için kullanılan değerlendirme kriterleri şu şekildedir:

2.4.1 Akaike Bilgi Kriteri (Akaike Information Criteria, AIC):

AIC, modelin verilere uygunluğunu saptamaya yarayan bir ölçüdür. Model uyumu için önemli bir göstergedir. Daha küçük AIC'ye sahip model daha iyi sonuç verir. Bir modelin AIC metriğine bakmak modelleri karşılaştırmada oldukça kullanışlıdır. Oluşturulan modelin daha doğru sonuç vermesi, regresyon tablosundaki yıldızlı değişkenler seçilerek yeni modellerin kurulmasıyla elde edilebilir. En düşük AIC'ye sahip model nispeten daha iyi olacaktır. [7]

AIC, doğruluğu karmaşıklıkla dengeleyen bir metriktir. Daha yüksek doğrulukta daha iyi sonuçlar alınır, eklenen değişkenler AIC değerini yükselterek kötü sonuçların elde edilmesine yol açabilmektedir. [11]

2.4.2 Hata Matrisi (Confusion Matrix):

Hata matrisi, sınıflandırma modellerini değerlendirmek için kullanılan en önemli metriktir. Gerçek ve tahmin edilen değerleri bir tablo biçiminde vererek karışıklığı önler. Bir hata matrisi, veri içindeki gerçek çıktılarla karşılaştırıldığında sınıflandırma modeliyle yapılan doğru ve yanlış tahminlerin sayısını gösterir.

Tutarlılık, Doğruluk (Accuracy): Modelin tahmin edilen kesinliğini belirler.

```
= (TP + TN)/(TP + TN + FP + FN)
```

Doğru Olumlu Oran (True Positive Rate, TPR): Tüm pozitif değerlerin arasından doğru kaç kere tahmin edildiğini gösterir.

```
= TP/(TP + FN)  , TPR = 1 - FNR, TPR aynı zamanda Hassaslık (Sensitivity) veya Geri Çağırma
```

Yanlış Olumlu Oran (False Positive Rate, FPR): Tüm negatif değerlerden kaç tane negatif değerin yanlış tahmin edildiğini gösterir.

```
= FP/(FP + TN)  , FPR = 1 - TNR,
```

Doğru Olumsuz Oran (True Negative Rate, TNR): Negatif değerlerin hepsinden kaç tanesinin doğru olarak tahmin edildiğini gösterir.

```
= TN/(TN + FP)  , Özgünlük (Specificity) olarak da isimlendirilir.
```

Yanlış Olumsuz Oran (False Negative Rate, FNR): Pozitif değerlerin hepsinden kaç tanesinin yanlış tahmin edildiğini gösterir.

```
= FN/(FN + TP)
```

Kesinlik (Precision): Tahmini pozitif değerlerin hepsinden kaç değerin gerçekten pozitif olduğunu gösterir.

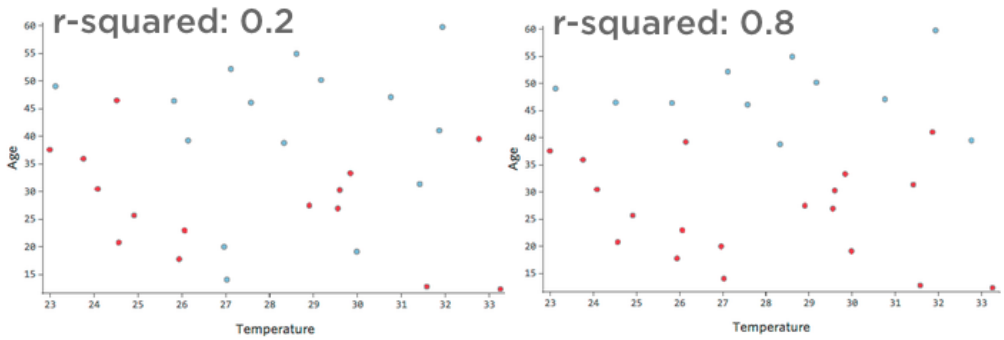
```
= TP/(TP + FP)
```

F Skoru (F Score): F skoru kesinlik ve TPR'nin harmonik ortalamasıdır. 0 ile 1 arasında değer alır. Değeri daha yüksek F skoru, modelin daha iyi olduğunu gösterir.

```
= 2[(kesinlik*TPR)/(kesinlik+TPR)]  [7]
```

Confusion Matrix		Target			
		Positive	Negative		
Model	Positive	a	b	*Positive Predictive Value*	a/(a+b)
	Negative	c	d	*Negative Predictive Value*	d/(c+d)
		Sensitivity	*Specificity*	**Accuracy** = (a+d)/(a+b+c+d)	
		a/(a+c)	d/(b+d)		

	1 (Predicted)	0 (Predicted)
1 (Actual)	True Positive	False Negative
0 (Actual)	False Positive	True Negative



2.4.3 R-Kare (R-Squared, R^2):

Modelin tahmin doğruluğunun nicelleştirilmesi için kullanılan sayısal bir metriktir. 0 ve 1 arasında değer alır. 0, modelin tahmini bir değerinin olmadığı; 1 ise modelin her şeyi mükemmel bir şekilde öngördüğü anlamına gelir.

Örneğin, solda temsil edilen veriler sağdaki verilerden daha az doğru bir modele aittir. Dağılım çizgisi boyunca bir çizgi çizmeye çalıştığımızı varsayarsak; sağ grafikteki kırmızılar maviden neredeyse tamamen ayrı kısımlardayken, sol grafikte bunu yapmak zordur.

İyi bir R-kare'nin sabit tanımı yoktur. Herhangi bir değişken eklendiğinde R-kare hareketi yükselir, bu nedenle mümkün olan en büyük kareyi elde etmek hedef değildir. Bunun yerine, modelin doğruluğunu genellikle içindeki değişkenlerin sayısını R-kare metriği ile dengelenmek istenir. [11]

2.5 Model Uyum Kriterleri:

Lojistik regresyonun model uyumunun değerlendirilmesinde Olabilirlik Oran Ki-kare (G test) ve Ki-kare (Chi-Squared) testleri baz alınmaktadır. G test, Fisher Kesin Olasılık Testi (Fisher's Exact test) ve Ki-kare testi, 2x2 çapraz tablolar için önerilen testlerdendir. G testi, çapraz tabloların hücrelerindeki gözlenen frekansların, beklenenlere göre uyum iyiliğini ya da diğer bir ifadeyle ele alınan iki faktörün birbirinden bağımsız olup olmadığını test etmektedir.

G testi, özel bir hipotezle gözlenen frekanslar ile beklenen frekansları karşılaştırmak amacıyla Ki-kare istatistiğine alternatif bir testtir. Ayrıca, teorik olarak daha sağlam temellere dayanması, Ki-kare dağılımına uygun olması, hücredeki beklenen frekansların 5'ten küçük olduğu durumlarda bile yapılabilmesi ve kolayca hesaplanabilmesi gibi avantajları olan G-testi yukarıda bahsedilen diğer iki test tekniği yerine alternatif olabilmektedir.

Lojistik regresyonda G-testi(-2logL) ile Pearson Ki-kare testi kullanılmaktadır. Lojistik regresyonda ilk aşama modelin dataya olan uyumunun değerlendirilmesi olmalıdır. Buna ilişkin hipotez; H_0: Model dataya uyumludur ya da H_1: Model dataya uyumlu değildir şeklinde kurulabilir. [12]

2.5.1 Pearson Ki-kare Testi:

Ki-kare testi her bir kategorinin gerçek değeri ile bu durumların alabileceği tahmini değerlerin karşılaştırılmasıdır. Gözlenen değer ile beklenen değerin karşılaştırılması olarak da adlandırılabilir. Büyük örneklemlerde çok iyi yaklaşıma sahiptir. Ki-kare testi yapabilmek için hata tablosundaki her hücrede 5'ten küçük değerler yer almamalıdır. Bu nedenden dolayı ki-karenin küçük örneklemlerde tercih edilmemesine neden olmuştur.

$$\chi^2 = \sum \frac{(Gzlenen_{ij} - Model_{ij})^2}{Model_{ij}}$$

2.5.2 Fisher Kesin Olasılık Testi (Fisher's Exact Test):

Küçük örneklemlerde Pearson Ki-kare testinin tercih edilememesinden dolayı alternatif olarak ki-kareye göre daha doğru sonuçlar veren Fisher Kesin Olasılık Testi kullanılır. Bu istatistik özellikle küçük örneklemlerden elde edilen 2x2 tabloları için ideal test aracı olsa da analizlerin daha fazla zaman alma ihtimalinin olmasına rağmen büyük örneklemlerden elde edilen büyük boyuttaki tablolar için de kullanılabilir.

2.5.3 Skor İstatistiği:

Skor istatistiği, temelinde olasılık fonksiyonu yer almayan, p serbestlik dereceli ki-kare dağılımı göstermektedir. Bağımlı değişken ile bağımsız değişken arasında bir ilişki olup olmadığı konusunda fikir yürütülmesini sağlar.

2.5.4 Wald İstatistiği:

Modelde yer alan bağımsız değişkenlere ait katsayıların önemli olup olmadığını değerlendirmede kullanılan bir istatistiktir. T testi ile hesaplanan bu istatistik, örneklem genişliği arttıkça standart normal dağılım gösterir.

$$z = \frac{\beta_j}{\sqrt{var(\beta_j)}}$$

şeklinde hesaplanır. Güven aralığı ise;

$$\beta_j = \pm z_{1-\alpha/2}\sqrt{var(\beta_j)}$$

2.5.5 Yates Düzeltmesi (Yates's Correction):

2x2 hata tablolarında Pearson Ki-kare değeri küçük p değerleri sunarak anlamlı değerler üretmeye eğilimlidir. Bu da I.Tip hata yapılma olasılığının artmasına sebep olur. Yates düzeltmesi bu sorunun ortadan kaldırılması için kulanılır. Aşağıdaki formülün Pearson Ki-kare'den tek farkı gözlenen ile model farkından 0.5 çıkarılmasıdır. Bulunan değer de Pearson Ki-kare değeri gibi yorumlanabilir.

$$\chi^2 = \sum \frac{(|Gzlenen_{ij} - Model_{ij}| - 0.5)^2}{Model_{ij}}$$

2.5.6 Ki-Kare Testinin Varsayımları:

Verilerin tahmini sonuçları hata tablosunun sadece bir hücresine ait olabilir. Yukarıdaki veri analizi istatistikleri 2 kategorik değişken içeren durumlar için kullanılmaktadır. Bu 2 değişkenin kategori sayısına göre tablolar 2x2, 2x3, 3x3... şeklinde adlandırılmaktadır. İki kategorik değişkenin

olduğu durumlarda ki-kare ve diğer istatistikler hesaplanabilir. Eğer veride ikiden fazla kategorik değişken varsa loglinear modeller kullanılabilir. Logaritmik doğrusal (log-linear) fonksiyon modeller iki kategorik değişkenin olduğu veriler için de kullanılabilir.

2.5.7 Logaritmik Doğrusal Model:

Burada ana etki değişkenleri ve bu ana etkilerin etkileşimleri modele bağımsız değişken olarak girmektedir. Analizler frekanslar üzerinden yapıldığı için logaritma alınarak gerçekleştirilmektedir.

2.5.8 Log-doğrusal Model Varsayımları:

Ki-karenin bir uzantısı olan log-doğrusal modeller de ki-karede olduğu gibi frekans sayıları üzerinde bazı şartları gerektirir. Çaprazlık tablosunun her hücresi 5'ten büyük frekansa sahip olmalı. Log-doğrusal modellerin güvenilir sonuçlar vermesi açısından tablodaki her hücrede 1'den küçük hiç değer olmaması ve 5'ten küçük frekansa sahip hücrelerin verinin %20'sini geçmemesi gerekir. [12]

2.6 Varyans Analizi (Analysis of Variance, ANOVA)

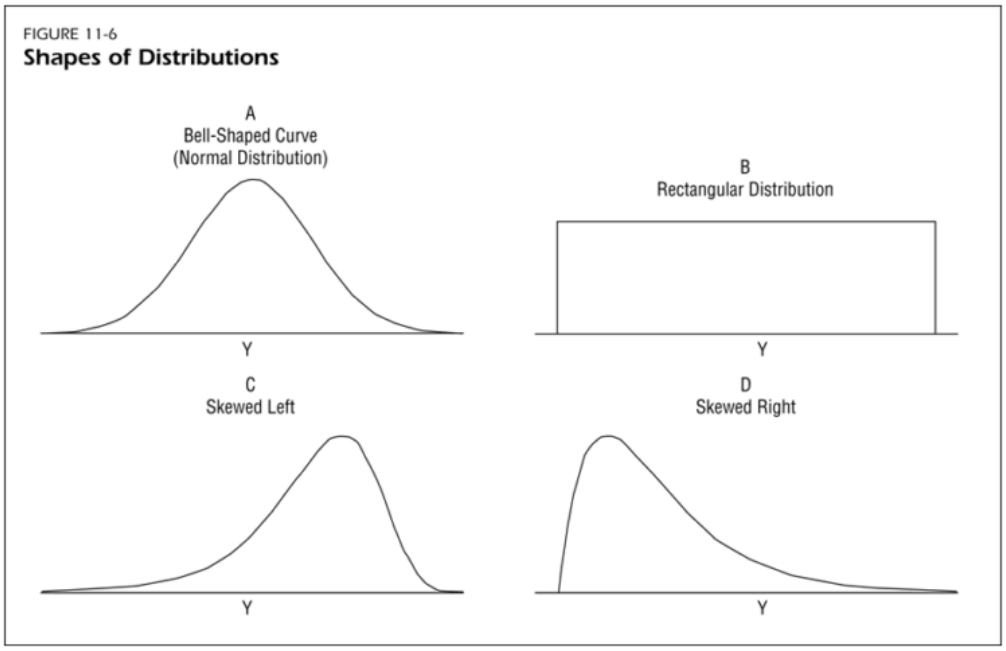
Varyans analizi (ANOVA), bir veri setinde bulunan toplam değişkenliği sistematik faktörler ve rasgele faktörler olarak iki kısma ayıran istatistikte kullanılan bir analiz aracıdı. Sistematik faktörler, veri kümesi üzerinde istatistiksel bir etkiye sahip iken rastgele faktörler sahip değildir. Varyans analizi sayesinde hangi bağımsız değişkenlerin bağımlı değişkene daha büyük etki ettiği tespit edilir. Bir başka deyişle ANOVA modeldeki (*)'lı değişkenleri gösterir. Varyans analizi bağımsız değişkenlerin sayısına göre tek yönlü ve iki yönlü olmak üzere iki çeşittir.

2.6.1 Tek Yönlü ve İki Yönlü Anova:

Tek yönlü ANOVA; bir bağımlı değişkeni etkileyen bağımsız değişkenlerin üzerindeki etkisini değerlendirir. Tüm örneklerin aynı olup olmadığını belirler. İki yönlü ANOVA; tek yönlü ANOVA'nın bir uzantısıdır. İki yönlü ANOVA'da, iki bağımsız değişken vardır. Örneğin, bir şirketin iki bağımsız değişkene dayalı işçi verimliliğini, maaş ve beceri seti olarak karşılaştırması iki yölü ANOVA ile sağlanır. İki faktör arasındaki etkileşimi gözlemlemek için kullanılır. İki faktörün etkisini aynı anda test eder. [13] Veri kümesinin bileşenleri üzerinde yapılan testlerde varyasyon vardır ve ANOVA bu varyasyonun sınıflandırma faktörü tarafından getirilen gruplandırma ile açıklanıp açıklanamayacağını araştırır. [14]

2.6.2 ANOVA'nın R'de Uygulanması:

- anova(model)
- aov1 <- aov(class ~., data = dataset) ~ Tüm değişkenlerle olan ilişkisini gösterir
- summary(aov1)
- aov2 <- aov(class ~ age + bgr, data = dataset) ~ Seçilen değişkenleri inceler
- summary(aov2)



2.7 Çarpık Dağılımlı Veri (Skewed Data):

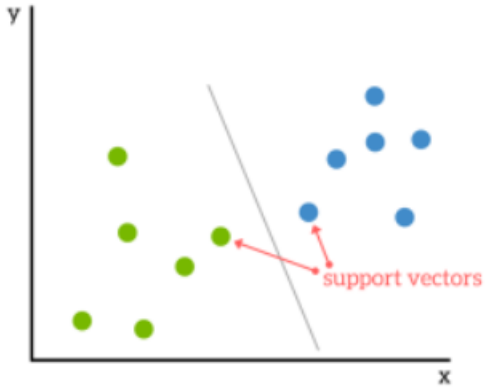
Çarpık dağılımlı veri normal dağılıma sahip olmayan dengesiz veri anlamına gelir. Çarpıklık, verilerin simetri eksikliğini tanımlar. Bir dağılımın asimetri derecesinin ölçüsüdür. Çok değişkenli normallik, regresyonun tüm değişkenlerinin normal olmasını gerektirdiği anlamına gelir. Çarpık dağılmış veri ile normallik varsayımı ihlal edilmiş olur.

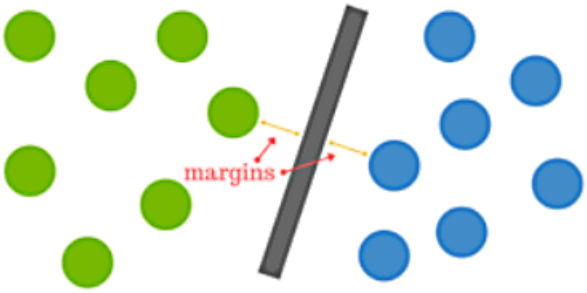
2.7.1 Çarpıklık Regresyon Modelini Nasıl Etkiler?

- Parametre tahminlerinde orantısız etkiye sebep olur. Düzensiz dağılmış gözlemler, parametre tahminleri üzerinde orantısız bir etki yapacaktır.
- Verilerin çarpıklığına bağlı olarak, güven aralığı gibi istatistikler normal dağılımlı hata varsayımına dayandığı için çok geniş veya çok dar olabilir.
- Çarpıklığı hesaba katacak şekilde özel olarak tasarlanmış modeller genellikle daha iyi performans gösterecektir. Bunlar, Gama veya ters Gauss dağılımlarına sahip genelleştirilmiş doğrusal modelleri içerir.

2.7.2 Lojistik Regresyon Modelinde Çarpıklık Durumunda Bağımsız Değişkenin Sonuçları Nelerdir?

Lojistik regresyon bağımsız değişkenler için özel bir dağılım gerektirmez. Bunlar normal, çarpık, kategorik olabilir. Bağımsız değişkenin normal dışı olması lineer gerileme uygulamaktan alıkoymaz, ancak bazı durumlarda sonuçların anlamlı olduğundan emin olmak için teşhislerde daha yakından dikkat etmek zorunda kalınabilir. [15]





3 Destek Vektör Makinesi (SVM):

Destek vektör makinesi bir sınıflandırma algoritmasıdır. SVM'in amacı, lineer programlama gibi gelişmiş optimizasyon metodlarını kullanarak verinin marjinini maksimize eden optimal ayırma hiperdüzlemini bulmaktır. [16]

Etiketli veriler varsa SVM, veri alanını bölümlere ayırarak ve her segment sadece bir tür veri içerecek şekilde çoklu ayırma hiperdüzlemlerini üretmek için kullanılabilir. SVM tekniği, genellikle dağılımı bilinmeyen veriler anlamına gelen düzenli olmayan veriler için kullanışlıdır. [17]

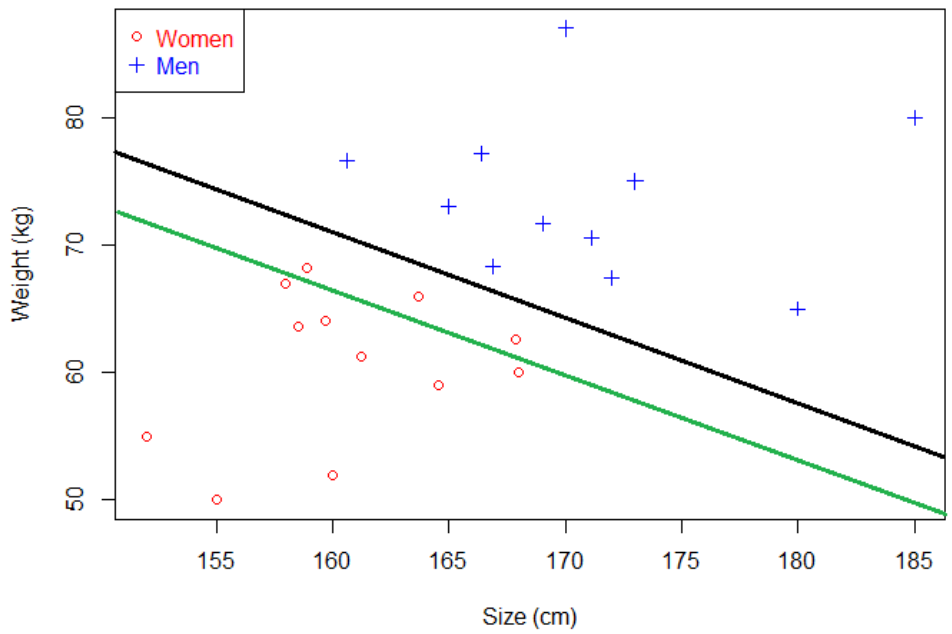
Destek vektörleri, hiperdüzleme en yakın veri noktalarıdır. Bölünen hiperdüzlemin konumunu değiştirecek bir veri kümesinin noktalarıdır. Bu nedenle bir veri kümesinin kritik ögeleri olarak kabul edilebilirler. Hiperdüzlem basitçe, sadece iki özelliğe sahip bir sınıflandırma için veri kümesini doğrusal olarak ayıran ve sınıflandıran bir çizgi olarak düşünülebilir. Hiperdüzlemden elde edilen veri noktalarının daha ötesinde, doğru bir şekilde sınıflandırılmış olduklarından emin olmak için veri noktalarının mümkün olduğunca hiperdüzlemden uzakta olmasını ve bunun doğru tarafında olması istenir.

- İki sınıf veri içinde en iyi nasıl ayrılabilir?

Hiperdüzlem ile en yakın veri noktası arasındaki uzaklık her iki gruptan da marj olarak bilinir. Amaç, hiperdüzlem ile veri seti içindeki herhangi bir nokta arasında mümkün olan en büyük marja sahip bir hiperdüzlem seçmektir ve bu da yeni verilerin doğru şekilde sınıflandırılma olasılığını verir. [18]

Her kategorideki veriden mümkün olduğunca uzakta bir hiperdüzlem seçmeye çalışılır. Örneğin aşağıdaki grafikteki siyah çizgi yeşilden daha doğru sınıflandırma yapar.

[16]



Hata fonksiyonunun şekline göre, SVM modelleri dört ayrı gruba ayrılabilir:

- Sınıflandırma SVM 1 (C-SVM classification)
- Sınıflandırma SVM 2 (nu-SVM classification)
- Regresyon SVM 1 (epsilon-SVM regression)
- Regresyon SVM 2 (nu-SVM regression)

3.1 SVM'in Matematiksel İfadesi:

İlk olarak veriyi ikiye ayırarak test için örneklem olarak kullanılacak veri şu şekilde ifade edilsin:

$$(x_1, y_1), ..., (x_\lambda, y_\lambda)$$

$$x \in R^n, y \in \text{ +1,-1}$$

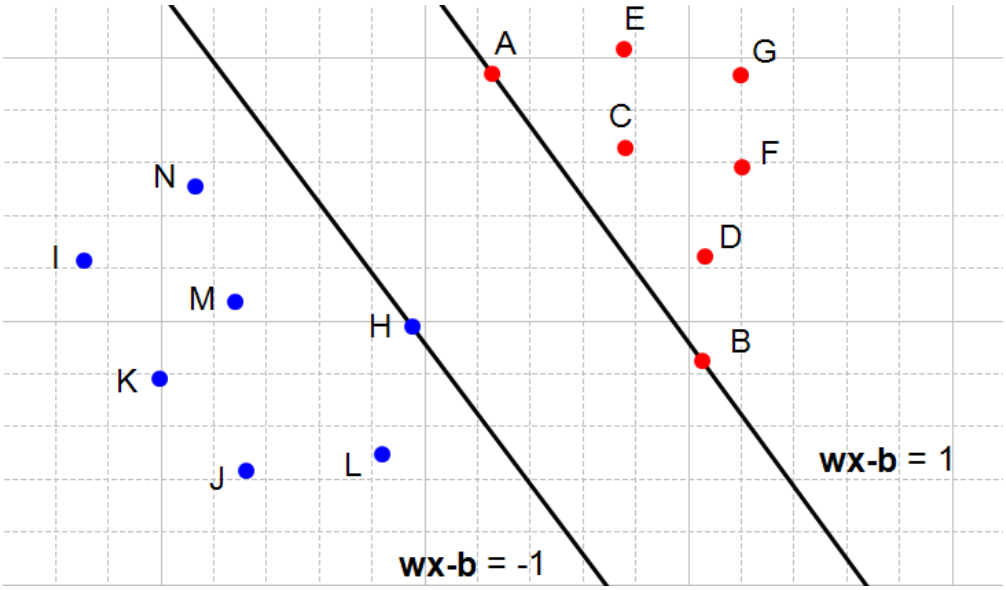
x: noktalar
w: vektörler
Veriyi hatasız ayıran en uygun hiperdüzlem;

$$(w.x) - b = 0$$

Hiperdüzlem ise şu şekilde tanımlanabilir:

$$(w.x_i) - b \geq 1 \text{ if } y_i = 1$$

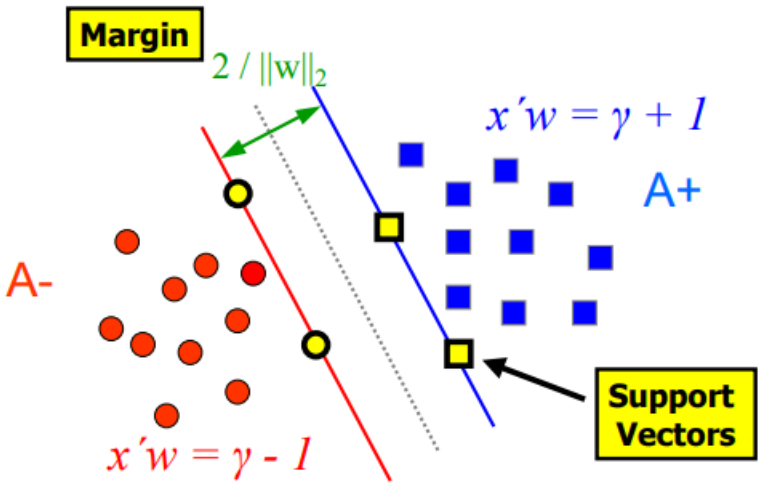
$$(w.x_i) - b \leq 1 \text{ if } y_i = -1$$



- $x_i = A$ noktasında $(w \cdot x_i) - b = 1$
- $x_i = C$ noktasında $(w \cdot x_i) - b \geq 1$
- $x_i = H$ noktasında $(w \cdot x_i) - b = -1$
- $x_i = M$ noktasında $(w \cdot x_i) - b \leq -1$ olur

[19]
Destek vektörleri $a_i \geq 0$ olan ve yukarıdaki denklemleri sağlayan noktalardır.
Kırmızı ve mavi noktaların bulunduğu düzlemler arasındaki uzaklık şu şekilde gösterilir:

$$\text{margin} = \frac{2}{||w||}$$



[1]

4 Karar Ağaçları (Decision Trees):

Sınıflandırma için açık kurallar sağlayan, eksik veriler ve doğrusal olmayan etkiler ile iyi başa çıkabilen veri madenciliğinin en sezgisel ve popüler yöntemlerinden biri karar ağaçlarıdır. Ögeyle ilgili gözlemleri haritalandırarak bir ögenin hedef değerini tahmin eder. Doğrudan pazarlama, müşteri tutma, dolandırıcılık tespiti, tıbbi sorunların tanısı gibi konularda oldukça fazla kullanılır. [20]

Karar ağaçları sıklıkla insan gibi düşünmeyi taklit eder, böylece verileri anlamak ve bazı iyi yorumlar yapmak çok basittir. Karar ağaçları, verilerin yorumlanması için mantığı gerçekten görmenizi sağlar. Karar ağacı, her düğümün bir özelliği temsil ettiği ağaçtır. Her bir bağ bir kararı temsil eder ve her yaprak bir sonucu temsil eder.

Bütün fikir, tüm veriler için bu şekilde bir ağaç oluşturmak ve her yaprakta tek bir sonucu işlemektir. [21]

Karar ağaçları, belirli bir hedef sınıfa olabildiğince saf olan alt kümeleri elde etmek için, eğitim setinin tekrarlı olarak bölünmesi yoluyla çalışır. Ağacın her düğümü, bir özellik üzerinde belirli bir testle bölünen belirli bir T kayıtları kümesine bağlanır. Örneğin, sürekli bir A niteliğinde bölünme test $A \leq x$ tarafından indüklenebilir. T kümeleri daha sonra ağacın sol dalına ve sağa doğru giden iki alt kümede bölünür.

$T\_l = \{t \in T : t(A) \leq x\}$ ve $T\_r = \{t \in T : t(A) > \{x\}\}$

Benzer şekilde, kategorik bir özellik B, ayrılmalarını kendi değerlerine göre uyarmak için kullanılabilir. Örneğin,

$B = b_1, \ldots, b_k$

ise, her bölüm $B = b_i$ tarafından indisle gösterilebilir.

Karar ağacını oluşturmak için özyinelemeli algoritmanın bölünme adımı, her bir özellik için olası tüm bölünmeleri dikkate alır ve seçilen bir kalite ölçüsüne göre en iyi olanı bulmaya çalışır: ayırma kriteri.

$A_1, \ldots, A_n, C$

A_j: öznitelikler, C: hedef sınıf [22]

Sürekli değişkenler için, karelerin toplamını en aza indirmek için her bir bölgeden bir sabit seçilir:

$$\min_{c \in \mathbb{R}} \sum_{x_i \in \mathbb{R}} (y_i - c)^2$$

N_i: $\mathbb{R}_i$'deki gözlem sayısını belirtir. Sonuç olarak;

$$\widehat{c_i} = \frac{1}{N_i} \sum_{x_k \in \mathbb{R}} y_k$$

Benzer şekilde, çıktı kategorik olduğunda i düğümdündeki sınıf k gözlemlerinin oranı:

$$\widehat{p}_{ik} = \frac{1}{N_i} \sum_{x_l \in \mathbb{R}} 1_{y_l \in \mathbb{R}}$$

Daha sonra, büyük çoğunluğa sahip gözlemler sınıflandırılırsa:

$$k(i) := \text{argmax} \widehat{p}_{ik}$$

4.1 Bölünmenin Ne Kadar İyi Olduğunu Belirlemek İçin Kullanılan Ölçümler:

4.1.1 Yanlış Sınıflama Hatası:

$\frac{1}{N_i} \sum_{x_l \in \mathbb{R}} 1_{y_l \neq k(i)} = 1 - \widehat{p}\_\{i, k(i)\}$

4.1.2 Gini İndeksi:

$\sum_{k \neq k'} \widehat{p}_{ik} \widehat{p}_{ik'} = \sum_{k=1}^{K} \widehat{p}_{ik}(1 - \widehat{p}_{ik})$

Her nitelik ikiye bölünerek olası bütün ikiye bölünmelerin sınanmasıdır. Katsayı, 0 (%0) ile 1 (%100) arasında değer alır; 0, kusursuz eşitliği temsil eder, 1 ise tam tersini yani eşitsizliği temsil eder. Gini indeksi, rastgele alınan bir ögenin hangi sıklıkta yanlış tespit edildiğini saptamak için kullanılan bir ölçüttür. Özellikler seçilirken düşük gini indeksi olana öncelik verilmelidir. Gini indeksi, yalnızca ikili bölmeleri (1 veya 0) gerçekleştirir ve gini indeksi arttıkça homojenlik de artar. CART (Sınıflama ve Regresyon Ağacı) ikili bölmeler oluşturmak için gini yöntemini kullanır. [23]

$$gini(S) = 1 - \sum_{j=1}^{n} p_j^2$$

Gini indeksi bütün değişkenlerin sürekli olduğunu kabul eder. Bu sürekliliği bozan en düşük gini indeks değerini veren ayrıma sahip değişken üzerinden bölünme gerçekleştirilir. Eğer bir S veri seti n farklı sınıfta N adet kayıt içeriyorsa p_i, j sınıfının S içindeki göreli sıklığını belirler. [24]

$$gini_{split}(S) = \frac{N_1}{N} gini(S_1) + \frac{N_2}{N} gini(S_2)$$

4.1.3 Entropi (Cross entropy or deviance):

$-\sum_{k=1}^{K} \widehat{p}_{ik} log \widehat{p}_{ik}$

Entropi rastgelelik, belirsizlik ve beklenmeyen bir durumun meydana gelme olasılığıdır. Entropinin yüksek olup olmamasına göre elde edilen bilginin fazlalığı değişir. Temel olarak entropi, belirli bir olayın öngörülebilirliğinden bahseder.

- örnekler aynı sınıfa aitse entropi = 0
- örnekler sınıflar arasında eşit dağılmışsa entropi = 1
- örnekler sınıflar arasında rastgele dağılmışsa 0 <entropi <1 [23]

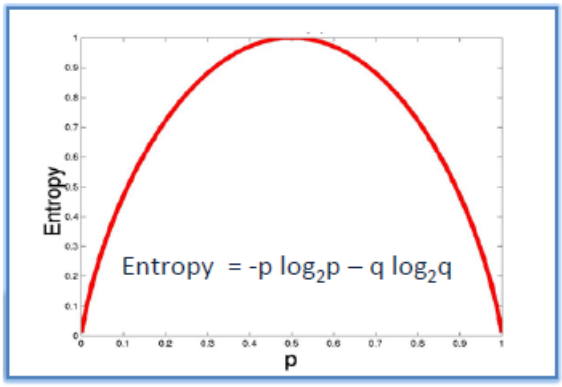
$$H(p_1, p_2, ..., p_s) = -\sum_{i=1}^{s} p_i log(p_i)$$

[25]

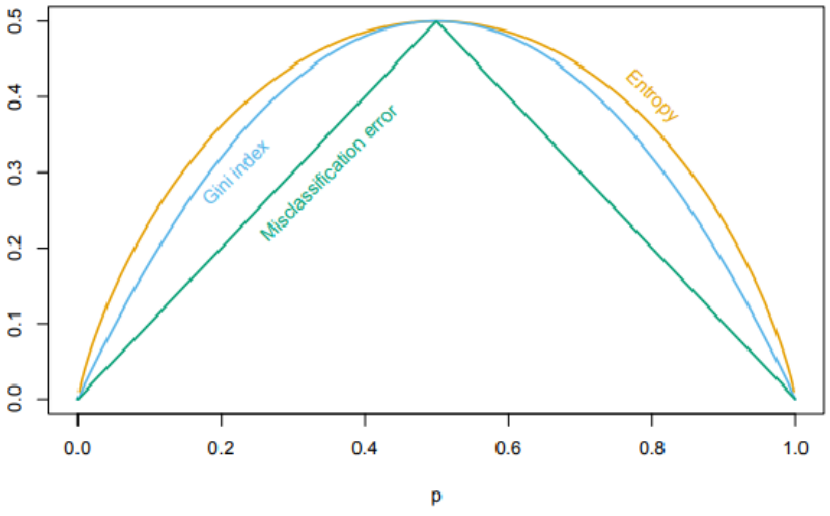
Entropi, verinin ne kadar dağınık olduğunun bir göstergesidir. Veri kümesi homojenliğini ölçmenin bir yoludur. % 100 homojen olan veri alt kümesi için hedef değişkenin entropisi sıfırdır ve mükemmel bir % 50-50 karışımı olan bir alt grubun entropisi 1 olarak kabul edilir. Veri kümesinde hedef değişkenin entropisi 0 ile 1 arasında değişmektedir. [26]

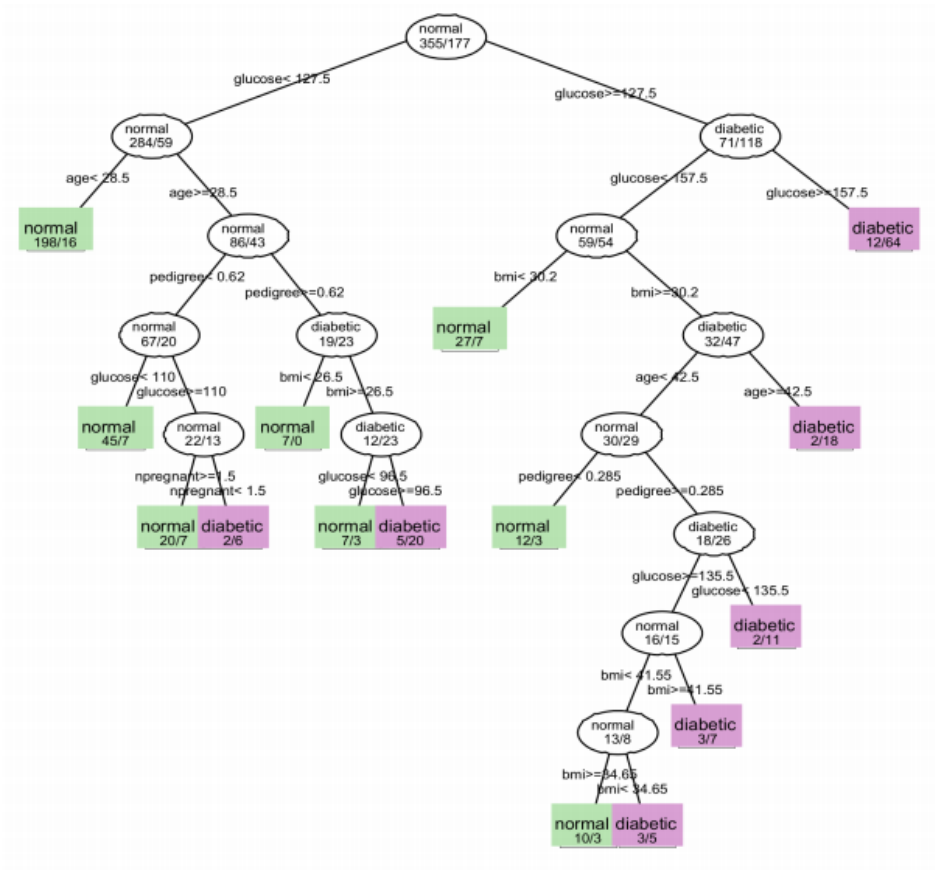
Karar ağacı çizilirken entropi ne kadar az olursa homojenlik o kadar fazla olacağından ana düğüm ve diğer bölünmelere kıyasla en düşük entropiye sahip olan bölünme seçilerek ilerlenir. [27]

[2] (Hastie ve diğerleri, 2009, s. 309)



Entropy = $-0.5 \log_2 0.5 - 0.5 \log_2 0.5 = 1$





Sıralanmamış değerler bölünürken bir q değerinin olası $ 2^{q-1}-1 $ adet ikiye bölünmesi vardır. 0 ve 1 olarak çıktıya sahip hesaplamalarda bu işlem kolaylaşır. Öngörücü sınıflar sonuç sınıfındaki 1'lerin azalan oranına göre sıralanır. Sonrasında sıralı bir tahmin ediciymiş gibi bölme yapılır. Bu bölme işlemi, tüm olası $ 2^{q-1}-1 $ bölünmeler boyunca çapraz entropi veya Gini indeksi terimleri açısından en uygun bölünmeyi gerçekleştirmiş olur. [2] (Hastie ve diğerleri, 2009, s. 310)

- Örnek olarak bir karar ağacı şeması şu şekilde oluşturulur:

[28]

5 Kronik Böbrek Hastalığı Veri Seti (Chronic Kidney Disease Dataset)

Veri; 250 kronik, 150 kronik olmayan 400 örnekten ve 11'i numerik, 14'ü ise nominal olan 25 değişkenden oluşur.

Sütun isimleri;

1. age : age (numerik)
2. bp : blood pressure (numerik)
3. sg : specific gravity
4. al : albumin
5. su : sugar

6. rbc : red blood cells
7. pc : pus cell
8. pcc : pus cell clumps
9. ba : bacteria
10. bgr : blood glucose random (numerik)
11. bu : blood urea (numerik)
12. sc :serum creatinine (numerik)
13. sod :sodium (numerik)
14. pot : potassium (numerik)
15. hemo : hemoglobin (numerik)
16. pcv : packed cell volume (numerik)
17. wc : white blood cell count (numerik)
18. rc : red blood cell count (numerik)
19. htn : hypertension
20. dm : diabetes mellitus
21. cad : coronary artery disease
22. appet : appetite
23. pe : pedal edema
24. ane : anemia
25. class : class

Veri üzerinde lojistik regresyon, destek vektör makinesi ve karar ağacı metodları uygulanarak testler yapılacaktır. Bu testler "class" değişkeninin sonucunu "ckd/notckd" yani kronik/kronik değil olarak tahmin etmeye çalışır. Uygulanan metodların analiz sonucuna göre hata oranının en az olduğu yani en yüksek oranda doğru tahminde bulunan, veriye en uygun metod belirlenir.

6 TESTLER

6.1 Lojistik Regresyon:

```
In [1]: library(caTools)
        library(e1071)
        library(rpart)
```

```
Warning message:
"package 'e1071' was built under R version 3.4.4"
```

Numerik verinin degisik kolonlari arasinda buyukluk farki var. Bazi kolonlar cok buyuk, bazilari cok ufak. Normalize etmek gerekli:

$$z_i = \frac{x_i - \overline{x}}{\sqrt{var(x)}}$$

```
In [2]: kidney <- read.csv("C:\\Users\\Lenova\\Desktop\\ch_kidney_disease.csv",
                           na.strings = "?")
```

```
In [3]: head(kidney)
```

id	age	bp	sg	al	su	rbc	pc	pcc	ba	...	pcv	wbcc	rbcc	ht
1	48	80	1.020	1	0	NA	normal	notpresent	notpresent	...	44	7800	5.2	ye
2	7	50	1.020	4	0	NA	normal	notpresent	notpresent	...	38	6000	NA	no
3	62	80	1.010	2	3	normal	normal	notpresent	notpresent	...	31	7500	NA	no
4	48	70	1.005	4	0	normal	abnormal	present	notpresent	...	32	6700	3.9	ye
5	51	80	1.010	2	0	normal	normal	notpresent	notpresent	...	35	7300	4.6	no
6	60	90	1.015	3	0	NA	NA	notpresent	notpresent	...	39	7800	4.4	ye

```
In [4]: nrow(na.omit(kidney))
```

158

Datada oldukça fazla NA değeri bulunmaktadır. 400 satırdan sadece 158 tanesinin NA değeri içermediğini görüyoruz. Şimdi NA değerleri içeren gözlemleri atarak tekrar dataya bakalım:

```
In [5]: kidney <- na.omit(read.csv("C:\\Users\\Lenova\\Desktop\\ch_kidney_disease.csv",
                                  na.strings = "?"))
        for (name in setdiff(colnames(kidney),c("class"))) {
            kidney[,name] <- as.numeric(kidney[,name])
            kidney[,name] <- (kidney[,name] - mean(kidney[,name]))/sqrt(var(kidney[,name]))
        }

        kidney[,"class"] <- as.numeric(kidney[,"class"]) - 1
```

```
In [6]: nrow(kidney)
```

158

```
In [7]: head(kidney)
```

	id	age	bp	sg	al	su	rbc	pc	p
4	-2.639279	-0.1007779	-0.3624604	-2.70476515	2.2662683	-0.3112437	0.3574321	-2.102409	3.
10	-2.581025	0.2215481	1.4271878	0.02301928	0.8509703	-0.3112437	-2.7800272	-2.102409	3.
12	-2.561607	0.8662002	-0.3624604	-1.79550367	1.5586193	-0.3112437	-2.7800272	-2.102409	3.
15	-2.532481	1.1885262	0.5323637	-1.79550367	1.5586193	2.1475813	0.3574321	-2.102409	3.
21	-2.474228	0.7372698	0.5323637	-0.88624220	0.8509703	-0.3112437	-2.7800272	-2.102409	-0
23	-2.454810	-0.1007779	0.5323637	0.93228075	2.2662683	-0.3112437	0.3574321	-2.102409	-0

```
In [8]: summary(kidney)
```

```
       id                age                bp                sg          
 Min.   :-2.6393   Min.   :-2.80832   Min.   :-2.1521   Min.   :-2.70477  
 1st Qu.:-0.3091   1st Qu.:-0.66485   1st Qu.:-1.2573   1st Qu.: 0.02302  
 Median : 0.2297   Median : 0.06039   Median : 0.5324   Median : 0.02302  
 Mean   : 0.0000   Mean   : 0.00000   Mean   : 0.0000   Mean   : 0.00000  
 3rd Qu.: 0.7855   3rd Qu.: 0.67280   3rd Qu.: 0.5324   3rd Qu.: 0.93228  
 Max.   : 1.2054   Max.   : 2.15550   Max.   : 3.2168   Max.   : 0.93228  
       al                su                rbc                pc          
 Min.   :-0.5643   Min.   :-0.3112   Min.   :-2.7800   Min.   :-2.1024  
 1st Qu.:-0.5643   1st Qu.:-0.3112   1st Qu.: 0.3574   1st Qu.: 0.4726  
```

```
 Median :-0.5643   Median :-0.3112   Median : 0.3574   Median : 0.4726
 Mean   : 0.0000   Mean   : 0.0000   Mean   : 0.0000   Mean   : 0.0000
 3rd Qu.: 0.1433   3rd Qu.:-0.3112   3rd Qu.: 0.3574   3rd Qu.: 0.4726
 Max.   : 2.2663   Max.   : 5.8358   Max.   : 0.3574   Max.   : 0.4726
      pcc                ba                bgr                bu
 Min.   :-0.3108   Min.   :-0.2858   Min.   :-0.944594   Min.   :-0.89831
 1st Qu.:-0.3108   1st Qu.:-0.2858   1st Qu.:-0.528824   1st Qu.:-0.56073
 Median :-0.3108   Median :-0.2858   Median :-0.243945   Median :-0.27589
 Mean   : 0.0000   Mean   : 0.0000   Mean   : 0.000000   Mean   : 0.00000
 3rd Qu.:-0.3108   3rd Qu.:-0.2858   3rd Qu.: 0.006286   3rd Qu.:-0.05962
 Max.   : 3.1970   Max.   : 3.4770   Max.   : 5.522931   Max.   : 5.41032
       sc                sod                pot               hemo
 Min.   :-0.5812   Min.   :-3.71833   Min.   :-0.61464   Min.   :-3.6733
 1st Qu.:-0.4837   1st Qu.:-0.51380   1st Qu.:-0.26945   1st Qu.:-0.3773
 Median :-0.3537   Median : 0.02028   Median :-0.03932   Median : 0.1952
 Mean   : 0.0000   Mean   : 0.00000   Mean   : 0.00000   Mean   : 0.0000
 3rd Qu.:-0.1913   3rd Qu.: 0.68789   3rd Qu.: 0.07574   3rd Qu.: 0.7243
 Max.   : 4.2278   Max.   : 1.48902   Max.   :12.18614   Max.   : 1.4269
      pcv               wbcc               rbcc               htn
 Min.   :-3.6153   Min.   :-1.4954   Min.   :-2.73874   Min.   :-0.522
 1st Qu.:-0.4852   1st Qu.:-0.6239   1st Qu.:-0.38433   1st Qu.:-0.522
 Median : 0.2287   Median :-0.2162   Median : 0.05712   Median :-0.522
 Mean   : 0.0000   Mean   : 0.0000   Mean   : 0.00000   Mean   : 0.000
 3rd Qu.: 0.6680   3rd Qu.: 0.4154   3rd Qu.: 0.69477   3rd Qu.:-0.522
 Max.   : 1.3270   Max.   : 5.7322   Max.   : 3.04918   Max.   : 1.904
       dm                cad              appet               pe
 Min.   :-2.9709   Min.   :-0.2727   Min.   :-0.3768   Min.   :-3.2447
 1st Qu.:-0.4336   1st Qu.:-0.2727   1st Qu.:-0.3768   1st Qu.:-0.3483
 Median :-0.4336   Median :-0.2727   Median :-0.3768   Median :-0.3483
 Mean   : 0.0000   Mean   : 0.0000   Mean   : 0.0000   Mean   : 0.0000
 3rd Qu.:-0.4336   3rd Qu.:-0.2727   3rd Qu.:-0.3768   3rd Qu.:-0.3483
 Max.   : 2.1037   Max.   : 3.6440   Max.   : 2.6763   Max.   : 2.5481
      ane              class
 Min.   :-0.3346   Min.   :0.0000
 1st Qu.:-0.3346   1st Qu.:0.0000
 Median :-0.3346   Median :1.0000
 Mean   : 0.0000   Mean   :0.7278
 3rd Qu.:-0.3346   3rd Qu.:1.0000
 Max.   : 2.9697   Max.   :1.0000
```

In [9]:
```
set.seed(144)
```

In [10]:
```
split <- sample.split(kidney$class, SplitRatio=1/5)
training <-subset(kidney, split == T)
testing  <-subset(kidney, split == F)
```

In [11]:
```
model <- glm(class~ age + bp + bgr + bu + sc + sod + pot + hemo
                    + pcv + wbcc + rbcc, data = na.omit(training), family = binomial("logit"))
```

```
Warning message:
"glm.fit: algorithm did not converge"Warning message:
"glm.fit: fitted probabilities numerically 0 or 1 occurred"
```

```
In [12]: summary(model)
```

```
Call:
glm(formula = class ~ age + bp + bgr + bu + sc + sod + pot +
    hemo + pcv + wbcc + rbcc, family = binomial("logit"), data = na.omit(training))

Deviance Residuals:
       Min          1Q      Median          3Q         Max
-1.011e-05  -2.110e-08   2.110e-08   1.960e-06   8.653e-06

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -5.899e+00  1.738e+05       0        1
age         -6.355e-01  1.500e+05       0        1
bp          -4.924e+00  1.871e+05       0        1
bgr         -4.525e+00  1.123e+05       0        1
bu          -1.865e+01  4.252e+05       0        1
sc          -5.362e+01  6.516e+05       0        1
sod          9.874e+00  2.300e+05       0        1
pot         -2.694e+01  7.565e+05       0        1
hemo         4.445e+00  1.508e+05       0        1
pcv         -9.846e+00  2.407e+05       0        1
wbcc        -1.902e+01  1.078e+05       0        1
rbcc        -5.577e+00  3.715e+05       0        1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 3.8024e+01  on 31  degrees of freedom
Residual deviance: 5.9855e-10  on 20  degrees of freedom
AIC: 24

Number of Fisher Scoring iterations: 25
```

```
In [13]: prediction <- predict(model, newdata = testing, type = "response")
         confMat <- table(testing$class, prediction > 0.5)
```

```
In [14]: confMat
```

```
   FALSE TRUE
  0    33    1
  1     0   92
```

```
In [15]: split <- sample.split(kidney$class, SplitRatio=4/9)
         training2 <-subset(kidney, split == T)
         testing2  <-subset(kidney, split == F)

In [16]: model2 <- glm(class~ age + bp + bgr + bu, data = training2,
                       family = binomial("logit"))

Warning message:
"glm.fit: fitted probabilities numerically 0 or 1 occurred"

In [17]: summary(model2)


Call:
glm(formula = class ~ age + bp + bgr + bu, family = binomial("logit"),
    data = training2)

Deviance Residuals:
     Min        1Q    Median        3Q       Max
-2.58885   0.00000   0.03676   0.27795   1.06172

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)   0.1638     0.7489   0.219   0.8269
age           0.2029     0.4425   0.459   0.6465
bp           -0.9479     0.8239  -1.151   0.2499
bgr          -1.9034     1.2837  -1.483   0.1381
bu          -10.6276     4.6133  -2.304   0.0212 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 81.854  on 69  degrees of freedom
Residual deviance: 19.325  on 65  degrees of freedom
AIC: 29.325

Number of Fisher Scoring iterations: 9



In [18]: prediction2 <- predict(model2, newdata = testing2, type = "response")

In [19]: confMat2 <- table(testing2$class, prediction2 > 0.5)

In [20]: confMat2


    FALSE TRUE
```

```
  0    19    5
  1     0   64
```

Anova ile modeldeki önemli değişkenler görülebilmektedir. Sapması (Deviance) en yüksek olan değişkenler modelde en fazla etkiye sahiptir.

```
In [21]: anova(model2)
```

	Df	Deviance	Resid. Df	Resid. Dev
NULL	NA	NA	69	81.85444
age	1	4.088148	68	77.76629
bp	1	10.790603	67	66.97569
bgr	1	21.274788	66	45.70090
bu	1	26.376077	65	19.32482

Üçüncü modeli anova ile belirlenen en önemli etkiye sahip iki değişkenle kuralım:

```
In [22]: split <- sample.split(kidney$class, SplitRatio=3/7)
         training3 <-subset(kidney, split == T)
         testing3  <-subset(kidney, split == F)

In [23]: model3 <- glm(class ~ bgr + bu, data = training3, family = binomial("logit"))

Warning message:
"glm.fit: fitted probabilities numerically 0 or 1 occurred"

In [24]: summary(model3)


Call:
glm(formula = class ~ bgr + bu, family = binomial("logit"), data = training3)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.5711   0.0000   0.0124   0.1727   1.1382

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  -0.8088     0.9692  -0.834   0.4040
bgr          -6.3415     2.9227  -2.170   0.0300 *
bu          -14.1815     6.6269  -2.140   0.0324 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 77.977  on 66  degrees of freedom
Residual deviance: 17.849  on 64  degrees of freedom
AIC: 23.849
```

```
Number of Fisher Scoring iterations: 10
```

```
In [25]: prediction3 <- predict(model3, newdata = testing3, type = "response")
         confMat3 <- table(testing3$class, prediction3 > 0.5)
```

```
In [26]: confMat3
```

```
    FALSE TRUE
  0    22    3
  1     1   65
```

```
In [27]: chisq.test(confMat)
```

```
Pearson's Chi-squared test with Yates' continuity correction

data:  confMat
X-squared = 116.01, df = 1, p-value < 2.2e-16
```

```
In [28]: chisq.test(confMat2)
```

```
Pearson's Chi-squared test with Yates' continuity correction

data:  confMat2
X-squared = 60.027, df = 1, p-value = 9.358e-15
```

```
In [29]: chisq.test(confMat3)
```

```
Pearson's Chi-squared test with Yates' continuity correction

data:  confMat3
X-squared = 67.301, df = 1, p-value = 2.33e-16
```

Modellerin accuracy ve FNR oranlarını bulalım:

```
In [30]: evaluate <- function(confMat) {
             TN <- confMat[1,1]
             FP <- confMat[1,2]
             FN <- confMat[2,1]
             TP <- confMat[2,2]

             accuracy <- (TP + TN)/(TP + TN + FP + FN)

             FNR <- FN/(FN + TP)

             c(accuracy,FNR) }
```

```
In [31]: evaluate(confMat)
```

1. 0.992063492063492 2. 0

```
In [32]: evaluate(confMat2)
```

1. 0.943181818181818 2. 0

```
In [33]: evaluate(confMat3)
```

1. 0.956043956043956 2. 0.0151515151515152

6.2 Destek Vektör Makinesi (SVM):

```
In [34]: split <- sample.split(kidney$class, SplitRatio=5/9)
         training4 <-subset(kidney, split == T)
         testing4  <-subset(kidney, split == F)
```

```
In [35]: model_svm <- svm(class~ age + bp + bgr + bu + sc
                           + sod + pot + hemo + pcv + wbcc + rbcc, data = training4)
```

```
In [36]: summary(model_svm)
```

```
Call:
svm(formula = class ~ age + bp + bgr + bu + sc + sod + pot + hemo +
    pcv + wbcc + rbcc, data = training4)


Parameters:
   SVM-Type:  eps-regression
 SVM-Kernel:  radial
       cost:  1
      gamma:  0.09090909
    epsilon:  0.1
```

```
Number of Support Vectors:  45
```

```
In [37]: pred <- predict(model_svm, testing4)
         tablo <- table(Hesaplanan = pred>0.5, Gercek = testing4$class)
```

```
In [38]: tablo
```

```
          Gercek
Hesaplanan  0  1
     FALSE 19  0
     TRUE   0 51
```

```
In [39]: chisq.test(tablo)
```

```
Pearson's Chi-squared test with Yates' continuity correction

data:  tablo
X-squared = 65.035, df = 1, p-value = 7.36e-16
```

```
In [40]: evaluate(tablo)
```

1. 1 2. 0

6.3 Karar Ağacı:

```
In [41]: split <- sample.split(kidney$class, SplitRatio=2/7)
         training5 <-subset(kidney, split == T)
         testing5  <-subset(kidney, split == F)
```

```
In [42]: model_dt <- rpart(class~ bgr + bu, data = training5)
```

```
In [43]: summary(model_dt)
```

```
Call:
rpart(formula = class ~ bgr + bu, data = training5)
  n= 45

          CP nsplit rel error    xerror      xstd
1 0.50657895      0 1.0000000 1.0639576 0.1681406
2 0.11111586      1 0.4934211 0.9543570 0.2217721
```

```
3 0.01217532      2 0.3823052 0.7709757 0.1836808
4 0.01000000      3 0.3701299 0.7882931 0.1873774

Variable importance
 bu bgr
 73  27

Node number 1: 45 observations,    complexity param=0.5065789
  mean=0.7333333, MSE=0.1955556
  left son=2 (7 obs) right son=3 (38 obs)
  Primary splits:
      bu  < -0.03325112 to the right, improve=0.5065789, (0 missing)
      bgr < 0.07173144  to the right, improve=0.3284347, (0 missing)
  Surrogate splits:
      bgr < 1.242046    to the right, agree=0.867, adj=0.143, (0 split)

Node number 2: 7 observations
  mean=0, MSE=0

Node number 3: 38 observations,    complexity param=0.1111159
  mean=0.8684211, MSE=0.1142659
  left son=6 (10 obs) right son=7 (28 obs)
  Primary splits:
      bgr < -0.1515522  to the right, improve=0.2251948, (0 missing)
      bu  < -0.4446836  to the right, improve=0.0647138, (0 missing)
  Surrogate splits:
      bu < -0.7822692  to the left,  agree=0.763, adj=0.1, (0 split)

Node number 6: 10 observations
  mean=0.6, MSE=0.24

Node number 7: 28 observations,    complexity param=0.01217532
  mean=0.9642857, MSE=0.03443878
  left son=14 (7 obs) right son=15 (21 obs)
  Primary splits:
      bgr < -0.6289171  to the left,  improve=0.11111110, (0 missing)
      bu  < -0.4446836  to the right, improve=0.04273504, (0 missing)
  Surrogate splits:
      bu < -0.08599887 to the right, agree=0.857, adj=0.429, (0 split)

Node number 14: 7 observations
  mean=0.8571429, MSE=0.122449

Node number 15: 21 observations
  mean=1, MSE=0
```

```
In [44]: predd <- predict(model_dt, testing5)
```

```
In [45]: tabloo <- table(Hesaplanan = predd > 0.5, Gercek = testing5$class)
```

```
In [46]: tabloo
```

```
          Gercek
Hesaplanan  0  1
     FALSE 28  0
     TRUE   3 82
```

```
In [47]: chisq.test(tabloo)
```

```
Pearson's Chi-squared test with Yates' continuity correction

data:  tabloo
X-squared = 93.676, df = 1, p-value < 2.2e-16
```

```
In [48]: evaluate(tabloo)
```

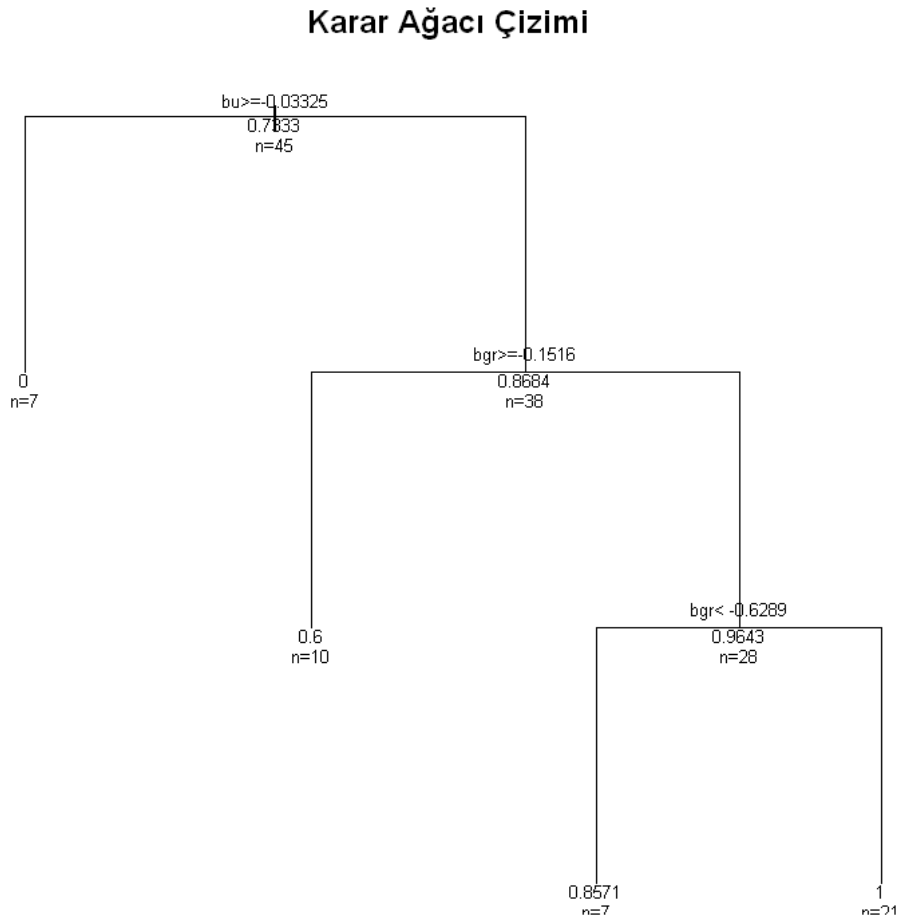
1. 0.973451327433628 2. 0.0352941176470588

```
In [49]: library(rpart)
```

```
In [50]: printcp(model_dt)
```

```
Regression tree:
rpart(formula = class ~ bgr + bu, data = training5)

Variables actually used in tree construction:
[1] bgr bu

Root node error: 8.8/45 = 0.19556

n= 45

        CP nsplit rel error  xerror    xstd
1 0.506579      0   1.00000 1.06396 0.16814
2 0.111116      1   0.49342 0.95436 0.22177
3 0.012175      2   0.38231 0.77098 0.18368
4 0.010000      3   0.37013 0.78829 0.18738
```

```
In [51]: plot(model_dt, uniform=TRUE, main="Karar Ağacı Çizimi ")
         text(model_dt, use.n=TRUE, all=TRUE, cex=.63)
```



7 TEST SONUÇLARININ KARŞILAŞTIRILMASI

- Ki-kare testinin sonuçlarına göre en yüksek değere sahip olan modelin en iyi sonucu verdiğini söyleyebiliriz. Yani bu durumda tüm değişkenlerle ilişkisini göz önüne alarak oluşturulan "model"in veriye yakınsama hatası olmasına rağmen en yüksek ki-kare değerini (116.01) vermesi aslında modelin en yüksek oranda tahminde bulunduğunu göstermektedir. İkinci en yüksek değer 93.676 ile karar ağacına ait olup modelin iyi sonuç verdiğini görmekteyiz.
- Modellerin karşılaştırılması verinin amacına göre değişiklik gösterebilir. Test edilen veride hastanın kronik böbrek hastası olup olmadığı tespit edilmeye çalışılmakta olduğundan, hasta olmayan bir kişiye hasta olduğunu söyleme durumu istenmeyeceği için bunu belirten FNR değerinin en küçük olduğu modeli tercih etmeliyiz. Örneğin başka bir veride

TNR durumuna göre karar verilmesi gerekebilir. Burada önemli olan veride hangi amacın görülmek istendiğidir. Ayrıca FNR ile birlikte tutarlılığın (accuracy) en fazla olduğu modele yönelmeliyiz.

- Modellerin hesaplanan tutarlılık ve FNR değerlerine bakarak değerlendirme yapalım. Öncelikle ki-kare testi sonucunun diğerlerinden daha düşük (65.035) çıkmasına rağmen model %100 doğru tahminde bulunduğundan "doğruluk" 1'dir. Yani lojistik regresyon, SVM ve karar ağacı metodları arasında SVM metodunun dataya en iyi uyum sağlayarak veri noktalarının hepsini doğru tahmin ettiği sonucuna varılmaktadır. Diğer iki metodu karşılaştırırsak ilk olarak ki-kare değerinin lojistik regresyonda daha yüksek olduğunu görmekteyiz. Yani LR, karar ağacına göre daha iyi bir metod gibi gözüküyor. Doğruluk ve FNR değerlerine de bakınca, LR daha yüksek oranda bir tutarlılık gösterdiğinden ve aynı zamanda FNR 0 olduğundan veri için karar ağacından daha iyi olduğunu kanıtlamaktadır.

- LR metodunda kurulan üç farklı modeli karşılaştırırmak için yine kesinik ve FNR değerlerini baz alabiliriz. Öncelikle FNR değeri 0 olan "model" ve "model2" ye baktığımızda doğru tahmin etme oranının "model"de daha yüksek olması sebebiyle tüm değişkenleri içeren ilk model yakınsama hatası vermesine rağmen üçünün arasında en iyi olan modeldir. İkinci ve üçüncü modelde, FNR "model2"de 0, "model3"te 0.015; doğruluk oranları ise "model2"de 0.9431, "model3"te 0.956'dır. Önceliğimiz FNR oranını minimum yapan model olduğundan dolayı "model2"nin "model3"ten daha iyi olduğu sonucuna varabiliriz.

8 KAYNAKÇA

1. Maclin, R. (2009). Advanced Machine Learning [Ders notları]. University of Minnesota-Duluth. Erişim adresi: https://www.d.umn.edu/~rmaclin/cs8751/spring2003/Notes/L15_SVMs.pdf

2. Berhane, F. (2017, 25 Şubat). Logistic regression regularized with optimization [Blog yazısı]. Erişim adresi: https://www.r-bloggers.com/logistic-regression-regularized-with-optimization/

3. HASTIE, T., TIBSHIRANI, R. ve FRIEDMAN, J. (2009). The Elements of Statistical Learning (2nd ed.). Stanford University. Erişim adresi: https://web.stanford.edu/~hastie/Papers/ESLII.pdf

4. Zisserman, A. (2015). Machine Learning [Ders notları]. University of Oxford. Erişim adresi: http://www.robots.ox.ac.uk/~az/lectures/ml/2011/lect4.pdf

5. Alice, M. (2015, 13 Eylül). How to perform a logistic regression in R [Blog yazısı]. Erişim adresi: https://www.r-bloggers.com/how-to-perform-a-logistic-regression-in-r/

6. Analytics Vidha Content Team (2015, 1 Kasım). Simple guide to logistic regression in R [Blog yazısı]. Erişim adresi: https://www.analyticsvidhya.com/blog/2015/11/beginners-guide-on-logistic-regression-in-r

7. Saraswat, M. (t.y.). Practical guide to logistic regression analysis in R [Blog yazısı]. Erişim adresi: https://www.hackerearth.com/practice/machine-learning/machine-learning-algorithms/logistic-regression-analysis-r/tutorial/

8. Kaşko, Y. (2007). Çoklu bağlantı durumunda ikili(binary) lojistik regresyon modelinde gerçekleşen I. tip hata ve testin gücü. Yüksek Lisan Tezi, Ankara Üniversitesi, Ankara.

9. Veri Bilimcisi. "Lojistik Regresyon (Logistic Regression)". Erişim adresi: https://veribilimcisi.com/2017/07/18/lojistik-regresyon/

10. Miller, S. (2014, 13 Ağustos). Reading a regression table: A guide for students [Blog yazısı]. Erişim adresi: http://svmiller.com/blog/2014/08/reading-a-regression-table-a-guide-for-students/

11. Statwing Documentation. "A user-friendly guide to logistic regression". Erişim adresi: http://docs.statwing.com/a-user-friendly-guide-to-logistic-regression/

12. Şen, S. (2016). Kategorik Veri Analizi. Erişim adresi: https://sedatsen.files.wordpress.com/2016/11/10-sunum.pdf

13. Investopedia. "Analysis Of Variance-ANOVA". Erişim adresi: https://www.investopedia.com/terms/a/anova.asp

14. Explorable (2009, 6 Haziran). "ANOVA". Erişim adresi: https://explorable.com/anova https://explorable.com/anova

15. Poyrekar, S. (2015, 20 Mayıs). How does skewness impact regression model? [Tartışma grubu]. Erişim adresi: https://www.quora.com/How-does-skewness-impact-regression-model

16. Kowalczyk, A. (2014, 2 Kasım). SVM - Understanding the math - Part 1 - The margin [Blog yazısı]. Erişim adresi: https://www.svm-tutorial.com/2014/11/svm-understanding-math-part-1/

17. Perceptive Analytics (2017, 19 Nisan). Machine learning using support vector machines [Blog yazısı]. Erişim adresi: https://www.r-bloggers.com/machine-learning-using-support-vector-machines/

18. Bambrick, N. (2016, 7 Temmuz). Support vector machines: A simple explanation [Blog yazısı]. Erişim adresi: https://www.kdnuggets.com/2016/07/support-vector-machines-simple-explanation.html

19. Kowalczyk, A. (2015, 8 Haziran). SVM - Understanding the math - the optimal hyperplane [Blog yazısı]. Erişim adresi: https://www.svm-tutorial.com/2015/06/svm-understanding-math-part-3/

20. DF Team (2017, 7 Temmuz). R decision trees – A tutorial to tree based modeling in R [Blog yazısı]. Erişim adresi: https://data-flair.training/blogs/r-decision-trees/

21. Sanjeevi, M. (2017, 6 Ekim). Chapter 4: Decision trees algorithms [Blog yazısı]. Erişim adresi: https://medium.com/deep-math-machine-learning-ai/chapter-4-decision-trees-algorithms-b93975f7a1f1

22. Simone (2012, 26 Kasım). Mathematics behind classification and regression trees [Tartışma grubu]. Erişim adresi: https://stats.stackexchange.com/questions/44382/mathematics-behind-classification-and-regression-trees

23. Veri Bilimcisi. "Karar Ağaçları (Decision Trees)". Erişim adresi: https://veribilimcisi.com/2018/02/23/karar-agaclari-decision-trees/

24. Akçetin, E. ve Çelik, U. (2014). İstenmeyen elektronik posta (spam) tespitinde karar ağacı algoritmalarının performans kıyaslaması. İnternet Uygulamaları ve Yönetimi, 5(2), 47. Erişim adresi: http://dergipark.gov.tr/download/article-file/402551

25. Sayad, S. (t.y.). Decision tree - classification. Erişim adresi: http://www.saedsayad.com/decision_tree.htm

26. Deshpande, B. (2012, 11 Ocak). A simple explanation of how entropy fuels a decision tree model [Blog yazısı]. Erişim adresi: http://www.simafore.com/blog/bid/94454/A-simple-explanation-of-how-entropy-fuels-a-decision-tree-model

27. Clear Rredictions. "What is a Decision Tree? How does it work?". Erişim adresi: https://clearpredictions.com/Home/DecisionTree

28. Guillot, D. (2016, 6 Nisan). Introduction to Data Mining and Analysis Decision Trees [Ders notları]. University of Delaware. Erişim adresi: http://www.math.udel.edu/~dguillot/teaching/MATH829/lectures-handout/20-MATH829-decision-tree-handout.pdf