T.C.

BİLECİK ŞEYH EDEBALİ ÜNİVERSİTESİ

İKTİSADİ İDARİ BİLİMLER FAKÜLTESİ

YÖNETİM BİLİŞİM SİSTEMLERİ

VERİ MADENCİLİĞİ

PROJE ADI:

KRONİK BÖBREK HASTALIĞI TAHMİNİ

HAZIRLAYAN:

AD-SOYAD: ERSEVEN KARATAŞ

İçindekiler

Giriş

Bu çalışmada, kronik böbrek hastalığının tahmin edilmesi amacıyla bir veri madenciliği uygulaması gerçekleştirilmiştir. Günümüzde sağlık alanında erken teşhis hem hastaların yaşam kalitesini artırmakta hem de sağlık sistemleri üzerindeki maliyetleri azaltmaktadır. Bu bağlamda, klinik ve demografik veriler kullanılarak KBH riskinin tahmin edilmesi önemli bir problem olarak karşımıza çıkmaktadır.

Bu çalışmada, böbrek fonksiyonlarını ve hastaya ait bazı klinik özellikleri içeren bir veri seti kullanılarak sınıflandırma problemi ele alınmıştır. Veri madenciliği sürecinde K-En Yakın Komşu (KNN) ve Naive Bayes algoritmaları uygulanmış ve modellerin performansları karşılaştırılmıştır.

Problemin Tanımı ve Çalışmanın Amacı

Bu çalışmanın temel problemi, hastalara ait klinik ölçümler kullanılarak bireyin kronik böbrek hastası olup olmadığının tahmin edilmesidir. Problem bir **ikili sınıflandırma problemi** olup hedef değişken KBH olarak belirlenmiştir.

Çalışmanın amacı, verilen klinik değişkenler doğrultusunda farklı sınıflandırma algoritmalarının performanslarını karşılaştırmaktır. Bu sayede, sağlık alanında karar destek sistemlerine katkı sağlayabilecek bir analiz ortaya konulması hedeflenmektedir.

Veri Setinin Tanımı ve Anlaşılması

Bu çalışmada kullanılan veri seti toplam 5000 gözlem ve 11 değişkenden oluşmaktadır. Veri seti, böbrek fonksiyonları, hastaya ait klinik ölçümler ve bazı demografik bilgileri içermektedir.

3.1 Değişkenler

- **Kreatinin:** Kandaki kreatinin seviyesi
- **BUN:** Blood Urea Nitrogen değeri
- **GFR:** Glomerüler Filtrasyon Hızı
- **İdrar_çıkışı:** Günlük idrar çıkışı
- **Diyabet:** Diyabet durumu (0: Yok, 1: Var)
- **Hipertansiyon:** Hipertansiyon durumu (0: Yok, 1: Var)
- **Yaş:** Hastanın yaşı
- **İdrarda_Protein:** İdrarda protein miktarı
- **Günlük Su tüketimi:** Günlük su tüketimi
- **İlaç Türü:** Kullanılan ilaç türü

- **KBH:** Kronik böbrek hastalığı durumu (0: Yok, 1: Var)

Hedef değişken KBH olup, modelleme aşamasında bu değişkenin tahmin edilmesi amaçlanmıştır.

```
> head(df)
  Creatinine       BUN       GFR Urine_Output Diabetes Hypertension      Age
1  0.7888031  8.386869 102.16179    1632.6494        0            0 27.68207
2  3.4139698 53.688796  50.07126     935.5405        1            0 33.12221
3  0.6476449  7.466540  89.45183    1774.5538        1            1 55.83228
4  0.7955081 12.516821  99.87218    2360.6030        0            0 32.39190
5  0.8690101 19.855960  86.11018    1987.7509        0            1 66.68951
6  0.7122430 18.360351  95.79075    1742.2611        0            0 48.93320
  Protein_in_Urine Water_Intake    Medication CKD_Status
1        106.70020     1.570370          None          0
2        410.00836     3.425287 ACE Inhibitor          1
3        123.33692     1.123301      Diuretic          0
4        116.09887     3.086846 ACE Inhibitor          0
5         55.66876     2.174980           ARB          0
6        122.53111     2.735247           ARB          0
```

Tablo 1

Veri Ön İşleme ve Hazırlık Süreci

Veri madenciliği sürecinin sağlıklı ilerleyebilmesi için veri seti üzerinde çeşitli ön işleme adımları uygulanmıştır.

Öncelikle veri seti incelenmiş ve bazı gözlemlerde **İlaç Türü** değişkeninde eksik değerler olduğu tespit edilmiştir. Bu eksik değerler uygun yöntemlerle ele alınmıştır ve ismi **X_ilacı** olarak değiştirilmiştir.

Sayısal değişkenler arasında ölçek farklılıkları bulunduğundan, özellikle KNN algoritmasının mesafeye dayalı çalışması nedeniyle normalizasyon işlemi uygulanmıştır. Normalizasyon işlemi ise bir R paketi olan **ClusterSim** paketi ile yapılmıştır. Bu sayede değişkenlerin modele etkisi dengelenmiştir. Tablo 5'te görüldüğü gibi normalizasyon işlemi yapılmış ve tablo 6'da sonuçlar verilmiştir.

Son olarak veri seti, modelleme sürecinde kullanılmak üzere eğitim (%70) ve test (%30) olmak üzere iki alt kümeye ayrılmıştır. Daha sonra eğitim (%80) ve test (%20), eğitim (%90) ve test (%10), eğitim (%60) ve test (%40) olacak şekilde denenmiş ve en ideal olan eğitim ve test verileri üzerinde çalışılmıştır.

Tablo 2 ve tablo 3'te görüldüğü üzere hedef değişkenimiz ve kategorik değişkenlerimiz faktöre dönüştürülerek KNN ve NAİVE BAYES algoritmaları için işleme hazırlanmıştır. Tablo 4 de ise en son dönüştürülmüş sonuçlar bulunmaktadır.

```r
# hedef değişken ve kategorik değisken faktör donüşümü
df$KBH <- factor(df$KBH, levels = c(0, 1))
df$Diyabet <- factor(df$Diyabet, levels = c(0, 1))
df$Hipertansiyon <- factor(df$Hipertansiyon, levels = c(0, 1))
df$Ilac_Turu <- factor(df$Ilac_Turu)
```

Tablo 2

```r
# Sayisal degiskenler numaric dönüşümü
df$Kreatinin <- as.numeric(df$Kreatinin)
df$BUN <- as.numeric(df$BUN)
df$GFR <- as.numeric(df$GFR)
df$Idrar_Cikisi <- as.numeric(df$Idrar_Cikisi)
df$Yas <- as.numeric(df$Yas)
df$Idrarda_Protein <- as.numeric(df$Idrarda_Protein)
df$Gunluk_Su_Tuketimi <- as.numeric(df$Gunluk_Su_Tuketimi)
```

Tablo 3

```
> str(df)
'data.frame':   5000 obs. of  11 variables:
 $ Kreatinin         : num  0.789 3.414 0.648 0.796 0.869 ...
 $ BUN               : num  8.39 53.69 7.47 12.52 19.86 ...
 $ GFR               : num  102.2 50.1 89.5 99.9 86.1 ...
 $ Idrar_Cikisi      : num  1633 936 1775 2361 1988 ...
 $ Diyabet           : Factor w/ 2 levels "0","1": 1 2 2 1 1 1 1 2 1 2 ...
 $ Hipertansiyon     : Factor w/ 2 levels "0","1": 1 1 2 1 2 1 2 2 1 1 ...
 $ Yas               : num  27.7 33.1 55.8 32.4 66.7 ...
 $ Idrarda_Protein   : num  106.7 410 123.3 116.1 55.7 ...
 $ Gunluk_Su_Tuketimi: num  1.57 3.43 1.12 3.09 2.17 ...
 $ Ilac_Turu         : Factor w/ 4 levels "ACE Inhibitor",..: 4 1 3 1 2 2 2 1 4 4 ...
 $ KBH               : Factor w/ 2 levels "0","1": 1 2 1 1 1 1 2 2 2 1 ...
>
```

Tablo 4

```
sayisal_df <- df[, sayisal_sutunlar]


sayisal_norm <- data.Normalization(sayisal_df, type = "n4")

df_norm <- df

df_norm[, sayisal_sutunlar] <- sayisal_norm

str(df_norm)
```

Tablo 5

```
> str(df)
'data.frame':   5000 obs. of  11 variables:
 $ Kreatinin         : num  0.789 3.414 0.648 0.796 0.869 ...
 $ BUN               : num  8.39 53.69 7.47 12.52 19.86 ...
 $ GFR               : num  102.2 50.1 89.5 99.9 86.1 ...
 $ Idrar_Cikisi      : num  1633 936 1775 2361 1988 ...
 $ Diyabet           : int  0 1 1 0 0 0 0 1 0 1 ...
 $ Hipertansiyon     : int  0 0 1 0 1 0 1 1 0 0 ...
 $ Yas               : num  27.7 33.1 55.8 32.4 66.7 ...
 $ Idrarda_Protein   : num  106.7 410 123.3 116.1 55.7 ...
 $ Gunluk_Su_Tuketimi: num  1.57 3.43 1.12 3.09 2.17 ...
 $ Ilac_Turu         : chr  "X_Ilaci" "ACE Inhibitor" "Diuretic" "ACE Inhibitor" ...
 $ KBH               : int  0 1 0 0 0 0 1 1 1 0 ...
```

Tablo 6

Modelleme ve Kullanılan Algoritmalar

Bu çalışmada sınıflandırma problemi için iki farklı algoritma kullanılmıştır: K-En Yakın Komşu (KNN) ve Naive Bayes.

5.1 K-En Yakın Komşu (KNN) Algoritması

KNN algoritması, bir gözlemin sınıfını belirlemek için en yakın komşularının sınıflarını dikkate alan mesafeye dayalı bir sınıflandırma yöntemidir. Algoritma, benzer özelliklere sahip gözlemlerin genellikle aynı sınıfa ait olduğu varsayımına dayanır.

Bu çalışmada KNN algoritması, sayısal değişkenlerin ağırlıklı olduğu veri setleri için uygun olması nedeniyle tercih edilmiştir. Komşu sayısı (k) değeri deneysel olarak belirlenmiş ve model eğitim verisi üzerinde test edilmiştir.

Kullanılan R paketleri:

- Class
- Caret

5.2 Naive Bayes Algoritması

Naive Bayes algoritması, Bayes teoremine dayanan olasılıksal bir sınıflandırma yöntemidir. Algoritma, değişkenlerin birbirinden bağımsız olduğu varsayımını yaparak sınıflandırma işlemini gerçekleştirir.

Bu çalışmada Naive Bayes algoritması, basit yapısı ve hızlı çalışması nedeniyle tercih edilmiştir. Özellikle sağlık verilerinde yaygın olarak kullanılan bu yöntem, KBH durumunun tahmin edilmesinde alternatif bir yaklaşım sunmaktadır.

Kullanılan R paketleri:

- e1071
- Caret

Model Performansının Değerlendirilmesi

6.1 KNN Modeli Performans Değerlendirmesi

KNN algoritması için elde edilen sonuçlar incelendiğinde, modelin tüm veri bölme oranlarında oldukça yüksek doğruluk değerleri ürettiği görülmektedir. %60–%40 veri bölünmesinde doğruluk oranı **0.993** olarak elde edilirken, %70–%30 oranında bu değer **0.9953** seviyesine ulaşmıştır. %80–%20 bölünmesinde doğruluk değeri **0.993** olarak sabit kalmış, %90–%10 oranında ise doğruluk **0.988** olarak ölçülmüştür. Bu sonuçlar k = 10 iken elde edilmiştir. Diğer k değerleri denenmiş en performanslı sonuç k = 10 iken alınmıştır.

Eğitim verisi oranının %70 seviyesinde en yüksek doğruluk değerinin elde edilmesi, KNN algoritmasının yeterli sayıda gözlem ile komşuluk ilişkilerini en iyi şekilde öğrendiğini göstermektedir. %90–%10 bölünmesinde doğruluk değerinde gözlemlenen düşüş, test verisinin çok az olması nedeniyle modelin genelleme yeteneğinin sınırlanmış olabileceğini düşündürmektedir.

```r
# 3.KNN MODELI
# KNN icin sadece sayisal verileri ve hedefi seciyoruz
X <- df_norm[, sayisal_sutunlar]
y <- df_norm$KBH

knn_calistir <- function(X, y, train_orani, k = 10) {
  set.seed(42)
  indeks <- sample(1:nrow(X), size = train_orani * nrow(X))

  X_train <- X[indeks, ]
  X_test  <- X[-indeks, ]
  y_train <- y[indeks]
  y_test  <- y[-indeks]

  knn_tahmin <- knn(
    train = X_train,
    test  = X_test,
    cl    = y_train,
    k     = k
  )

  accuracy <- mean(knn_tahmin == y_test)
  return(accuracy)
}

# Farkli oranlarda KNN calistirma
acc_knn_60 <- knn_calistir(X, y, 0.60, k = 10)
acc_knn_70 <- knn_calistir(X, y, 0.70, k = 10)
acc_knn_80 <- knn_calistir(X, y, 0.80, k = 10)
acc_knn_90 <- knn_calistir(X, y, 0.90, k = 10)

acc_knn_60
acc_knn_70
acc_knn_80
acc_knn_90


knn_sonuclar <- data.frame(
  Egitim_Orani = c("60%", "70%", "80%", "90%"),
  K_Degeri     = c(10, 10, 10, 10),
  Basari_Orani = c(acc_knn_60, acc_knn_70, acc_knn_80, acc_knn_90)
)

print(knn_sonuclar)
```

Tablo 7

```
> print(knn_sonuclar)
  Egitim_Orani K_Degeri Basari_Orani
1          60%       10    0.9920000
2          70%       10    0.9946667
3          80%       10    0.9920000
4          90%       10    0.9900000
```

Tablo 8

6.2 Naive Bayes Modeli Performans Değerlendirmesi

Naive Bayes algoritması için elde edilen sonuçlar, modelin farklı veri bölme oranlarında **istikrarlı ancak KNN'ye kıyasla daha düşük** doğruluk değerleri ürettiğini göstermektedir. %60–%40 bölünmesinde doğruluk **0.982**, %70–%30 oranında **0.9813**, %80–%20 oranında **0.983** olarak hesaplanmıştır. %90–%10 veri bölünmesinde ise doğruluk değeri **0.978** olarak elde edilmiştir.

Naive Bayes algoritmasının performansının eğitim verisi oranı arttıkça belirgin bir artış göstermemesi, algoritmanın varsayımsal bağımsızlık yapısından kaynaklanmaktadır. Buna rağmen, tüm oranlarda %97'nin üzerinde doğruluk sağlaması, modelin sağlık verileri gibi gerçek dünya problemlerinde güvenilir bir alternatif sunduğunu ortaya koymaktadır.

```r
#naive bayes
library(e1071)
library(caret)

str(df_norm)

nb_calistir <- function(df, train_orani) {

  set.seed(42)

  indeks <- sample(
    1:nrow(df),
    size = train_orani * nrow(df)
  )

  train_df <- df[indeks, ]
  test_df  <- df[-indeks, ]

  nb_model <- naiveBayes(
    BYH ~ .,
    data = train_df
  )

  tahmin <- predict(nb_model, test_df)

  accuracy <- mean(tahmin == test_df$BYH)

  return(accuracy)
}

acc_nb_60 <- nb_calistir(df_norm, 0.60)
acc_nb_60

acc_nb_70 <- nb_calistir(df_norm, 0.70)
acc_nb_70

acc_nb_80 <- nb_calistir(df_norm, 0.80)
acc_nb_80
acc_nb_90 <- nb_calistir(df_norm, 0.90)
acc_nb_90

nb_sonuclar <- data.frame(
  Egitim_Orani = c("60%", "70%", "80%","90%"),
  Test_Orani   = c("40%", "30%", "20%","90%"),
  Accuracy     = c(acc_nb_60, acc_nb_70, acc_nb_80,acc_nb_90)
)

nb_sonuclar
```

Tablo 9

```
> nb_sonuclar
  Egitim_Orani Test_Orani  Accuracy
1          60%        40% 0.9820000
2          70%        30% 0.9813333
3          80%        20% 0.9830000
4          %90        %10 0.9780000
>
```

Tablo 10

6.3 KNN ve Naive Bayes Karşılaştırmalı Değerlendirme

Her iki algoritma birlikte değerlendirildiğinde, KNN modelinin tüm eğitim–test oranlarında Naive Bayes modelinden daha yüksek doğruluk değerleri ürettiği görülmektedir. Bu durum, veri setinde yer alan sayısal değişkenlerin KNN algoritmasının mesafeye dayalı yapısına daha uygun olduğunu göstermektedir.

```
karsilastirma_tablosu <- data.frame(
  Egitim_Orani = c("60%", "70%", "80%", "90%"),
  Test_Orani   = c("40%", "30%", "20%", "10%"),
  KNN_Basari   = c(acc_knn_60, acc_knn_70, acc_knn_80, acc_knn_90),
  NB_Basari    = c(acc_nb_60, acc_nb_70, acc_nb_80, acc_nb_90)
)
```

Tablo 11

```
> print(karsilastirma_tablosu)
  Egitim_Orani Test_Orani KNN_Basari NB_Basari
1          60%        40%  0.9930000 0.9820000
2          70%        30%  0.9953333 0.9813333
3          80%        20%  0.9930000 0.9830000
4          90%        10%  0.9880000 0.9780000
>
```

Tablo 12

Sonuçlar ve Değerlendirme

Bu çalışmada, kronik böbrek hastalığının tahmin edilmesi amacıyla klinik ve demografik verilerden oluşan bir veri seti kullanılarak veri madenciliği süreci gerçekleştirilmiştir. Çalışma kapsamında K-En Yakın Komşu (KNN) ve Naive Bayes algoritmaları uygulanmış, farklı eğitim–test veri oranları kullanılarak modellerin performansları karşılaştırılmıştır.

Elde edilen bulgular, her iki algoritmanın da yüksek doğruluk değerleri ürettiğini göstermektedir. Ancak tüm veri bölme oranları dikkate alındığında, KNN algoritmasının Naive Bayes algoritmasına kıyasla daha yüksek sınıflandırma başarısı sağladığı görülmüştür. Özellikle %70–%30 eğitim–test bölünmesinde KNN modelinin en yüksek doğruluk değerine ulaşması, bu oranın modelin genelleme yeteneği açısından en dengeli yapı olduğunu göstermektedir.

Naive Bayes algoritması ise daha basit yapısı ve düşük hesaplama maliyeti ile dikkat çekmektedir. Eğitim verisi oranındaki değişimlere rağmen istikrarlı performans sergilemesi, bu algoritmanın sağlık verileri gibi gerçek dünya problemlerinde güvenilir bir alternatif sunduğunu ortaya koymaktadır. Ancak bağımsızlık varsayımı nedeniyle KNN algoritmasına kıyasla sınıflandırma başarısının daha düşük kaldığı gözlemlenmiştir.

Genel olarak değerlendirildiğinde, bu çalışmada kullanılan veri seti için **KNN algoritmasının daha uygun bir sınıflandırma yöntemi olduğu** sonucuna varılmıştır. Bununla birlikte, Naive Bayes algoritması da hesaplama maliyeti ve uygulama kolaylığı açısından önemli avantajlar sunmaktadır. Çalışma sonuçları, geliştirilen modellerin sağlık alanında karar destek sistemleri kapsamında erken teşhis süreçlerine katkı sağlayabileceğini göstermektedir.

Gelecek çalışmalarda, farklı sınıflandırma algoritmalarının kullanılması, daha geniş veri setleri ile model performansının artırılması mümkündür.

Kaynakça

Kaggle. (2025). Kidney Function Health Dataset.
https://www.kaggle.com/datasets/miadul/kidney-function-health-dataset

Torun, N. (2025). Veri Madenciliği ders notları.

Bilecik Şeyh Edebali Üniversitesi, Yönetim Bilişim Sistemleri Bölümü.

https://medium.com/machine-learning-türkiye/knn-k-en-yakın-komşu-7a037f056116