MİMAR SİNAN
GÜZEL SANATLAR
ÜNİVERSİTESİ

ÇOK DEĞİŞKENLİ İSTATİSTİKSEL YÖNTEMLER İLE KRONİK BÖBREK RAHATSIZLIĞI İNCELEMESİ

ZEHRA BETÜL GÜNDOĞDU

Tez Danışmanı

DR. ÖĞR. ÜYESİ BİLGE ÖZLÜER BAŞER

Temmuz 2024

ÖNSÖZ

Mimar Sinan Güzel Sanatlar Üniversitesi, Fen Edebiyat Fakültesi, İstatistik bölümü bitirme tezi kapsamında "Çok Değişkenli İstatistiksel Yöntemler ile Kronik Böbrek Rahatsızlığı İncelemesi" adlı çalışma hazırlanmıştır. Bu çalışma ile Kronik Böbrek Rahatsızlığını daha iyi anlamak ve hastalığın risk faktörlerini belirlemek amacıyla gerçekleştirilmiştir.

Tez çalışmasının aşamalarında rehberlik yapan ve katkılarını esirgemeyen danışman hocam **Dr. Öğr. Üyesi Bilge ÖZLÜER BAŞER**'e şükranlarımı arz ederim.

Lisans eğitimim boyunca bilgi ve tecrübeleriyle bana destek olan Fen Edebiyat Fakültesi İstatistik bölümündeki tüm hocalarıma da sonsuz şükranlarımı arz ederim.

Son olarak yardımlarını hiçbir zaman esirgemeyen ve yanımda olduklarını her az hissettiren aileme de sonsuz teşekkür ederim.

Zehra Betül GÜNDOĞDU

İÇİNDEKİLER

TABLO LİSTESİ

GRAFİK LİSTESİ

ÖZET

Bu çalışma, kronik böbrek hastalığını anlamak ve hastalığın risk faktörlerini belirlemek amacıyla çok değişkenli istatistiksel analiz yöntemlerini kullanmaktadır. Veri seti, kronik böbrek hastalığı olan hastalara ait sağlık verileri içermekte ve toplamda 26 sütun ve 280 gözlemden oluşmaktadır. Veriler, Kaggle platformundan elde edilmiştir ve hastaların çeşitli sağlık parametrelerini ve tıbbi geçmişlerini içermektedir.

Analiz sürecine başlamadan önce, veri ön işleme adımları uygulanmıştır. Bu adımlar arasında, analiz için gerekli olmayan benzersiz hasta kimlik numarası "id" kolonunun çıkarılması ve eksik veri içeren gözlemlerin temizlenmesi bulunmaktadır. Bu işlemler sonucunda, veri seti 25 sütun ve 107 gözlemden oluşmaktadır.

Veri setindeki değişkenlerin normal dağılıma uygunluğunu test etmek için Shapiro-Wilk normallik testi uygulanmıştır. Test sonuçlarına göre, yalnızca yaş değişkeni normal dağılıma uygun bulunmuştur. Diğer değişkenler için logaritmik dönüşüm, Box-Cox dönüşümü ve Yeo-Johnson dönüşümü gibi veri dönüşüm teknikleri uygulanmıştır. Bu dönüşümler sonucunda, bazı değişkenler normal dağılıma uygun hale getirilmiştir.

Varyans homojenliği testi (Levene testi) uygulanmış ve gruplar arasındaki varyansların homojen olduğu sonucuna varılmıştır. ANOVA ve MANOVA testleri kullanılarak gruplar arasındaki farklar incelenmiş ve bu testler sonucunda, kronik böbrek hastalığı durumu ile bazı bağımlı değişkenler arasında anlamlı ilişkiler bulunmuştur.

Diskriminant analizi ve lojistik regresyon analizi kullanılarak, kronik böbrek hastalığının sınıflandırılması gerçekleştirilmiştir. Diskriminant analizi, hasta olan ve olmayan bireyleri diskriminant bileşenlerine göre başarılı bir şekilde ayırmıştır. Lojistik regresyon modeli ise %97 doğruluk oranıyla hastalık sınıflandırmasını yapmıştır. Bu modelde, bazı değişkenler kronik böbrek hastalığı olasılığını artırırken bazıları azaltmaktadır.

Kümeleme analizi, hasta olan ve olmayan bireyleri gruplar halinde ayırmak için kullanılmıştır. K-means kümeleme algoritması ile yapılan analizlerde, her kümede bireylerin hastalık var/yok durumlarına göre dağılımı gösterilmiştir. Ek olarak, temel bileşenler analizi ve faktör analizi ile verilerin boyut indirgemesi yapılmış ve bu analizlerin sonuçları grafiksel olarak gösterilmiştir.

Son olarak, Random Forest modeli kullanılarak değişkenlerin önem dereceleri belirlenmiştir. Bu model, test verisi üzerinde %100 doğruluk oranına sahiptir ve değişken önem sıralamasında kan üresi ve kırmızı kan hücresi sayısının en yüksek öneme sahip olduğu bulunmuştur.

Genel olarak, bu çalışma kronik böbrek hastalığının erken teşhisi ve risk faktörlerinin belirlenmesi için önemli bilgiler sunmaktadır. Çeşitli istatistiksel yöntemlerin kullanılmasıyla elde edilen sonuçlar, hastalığın risk faktörlerinin daha iyi anlaşılmasına katkı sağlamaktadır.

1.GİRİŞ

Kronik böbrek rahatsızlığı, dünya genelinde milyonlarca insanı etkileyen ve yaşam kalitesini ciddi şekilde düşüren bir sağlık sorunudur.

Bu çalışmanın amacı, çok değişkenli istatistiksel yöntemler kullanarak kronik böbrek rahatsızlığını daha iyi anlamak ve hastalığın risk faktörlerini belirlemektir. Bu doğrultuda, Kaggle platformundan elde edilen kronik böbrek hastalığı veri seti üzerinde çeşitli istatistiksel analizler gerçekleştirilmiştir. Çalışmada kullanılan veri seti, kronik böbrek hastalığı olan hastalara ait sağlık verilerini içermekte olup, bu veriler çeşitli sağlık parametrelerini ve tıbbi geçmişleri kapsamaktadır.

Çalışmanın ilk bölümünde, veri ön işleme adımları ve veri setinin temel istatistikleri incelenmiştir. Ardından, değişkenlerin normal dağılıma uygunluğunu değerlendirmek için Shapiro-Wilk normallik testi ve dönüşüm teknikleri kullanılmıştır. Levene testi ve ANOVA testleri ile gruplar arasındaki farklar analiz edilmiştir. Sonrasında, MANOVA ile bağımsız değişkenlerin bağımlı değişkenler üzerindeki etkileri incelenmiş ve anlamlı ilişkiler belirlenmiştir.

Diskriminant analizi ve lojistik regresyon analizi kullanılarak, hastalık durumu sınıflandırılmış ve sınıflandırma modellerinin doğruluğu değerlendirilmiştir. K-means kümeleme algoritması ile veri seti kümeleme analizine tabi tutulmuş ve hastalık durumuna göre grupların farklılıkları görsel olarak gösterilmiştir. Faktör analizi ve Random Forest modeli kullanılarak değişkenlerin hastalık durumu üzerindeki etkileri ve önem dereceleri belirlenmiştir.

Bu çalışma, kronik böbrek rahatsızlığı için önemli bilgiler sunmakta ve çeşitli istatistiksel yöntemlerin kullanımının hastalığın risk faktörlerinin belirlenmesinde nasıl katkı sağladığını göstermektedir

2.ÇOK DEĞİŞKENLİ İSTATİSTİKSEL YÖNTEMLER

Çok değişkenli istatistiksel analizler, araştırılan olay ve ilgili olduğu değişkenleri dikkate alarak veya çok sayıdaki değişkeni daha az sayıda doğrusal faktörlerine indirgeyerek, değişkenler arasındaki karmaşık ilişkileri incelemek ve çözümlere ulaşmak için geliştirilmiş yöntemler bütünüdür. (Özdamar,2002:1) Ayrıca tek değişkenli analiz yöntemleri için varsayımların gerçekleşmemesi durumunda çok değişkenli analizler gerçekçi bir yaklaşımdır. Bu nedenle çok değişkenli istatistiksel analizler veri çözümlemede önemli bir yere sahiptir.

Çok değişkenli analiz yöntemleri, iki veya daha fazla değişkenin birbirleriyle olan ilişkilerini incelemek için kullanılır.

Çok Değişkenli Analiz Türleri

1. **ANOVA**
    - Bir bağımsız değişkenin, bir veya daha fazla grubu olan bir bağımlı değişken üzerindeki etkisini analiz etmek için kullanılır.
    - Gruplar arasındaki ortalama farklarını test etmek için kullanılır.
2. **MANOVA (Çoklu Varyans Analizi)**:
    - Bir veya daha fazla bağımsız değişkenin, birden fazla bağımlı değişken üzerindeki etkisini analiz etmek için kullanılır.
    - Tek yönlü ANOVA'nın çoklu bağımlı değişkenli versiyonudur.
3. **Diskriminant Analizi**:
    - Gruplar arasındaki farkları belirlemek ve bu gruplara yeni gözlemleri sınıflandırmak için kullanılır
4. **Lojistik**:
    - Bir veya daha fazla bağımsız değişken kullanarak bir bağımlı değişkenin belirli bir kategorisine ait olma olasılığını tahmin etmek için kullanılanılır.
    - Bağımlı değişkenin iki veya daha fazla kategorik değer aldığı durumlarda kullanılır.
5. **Faktör Analizi**:
    - Büyük veri setlerindeki değişkenlerin, daha az sayıda gizli faktörle temsil edilmesini sağlar.
    - Faktör yükleri, her bir değişkenin faktörlerle ne kadar ilişkili olduğunu gösterir.

6. **Kümeleme Analizi (Cluster Analysis)**:
    - Veri setini benzer özelliklere sahip gruplara ayırmak için kullanılır.
    - Hiyerarşik Kümeleme ve K-Means Kümeleme gibi yöntemler vardır.
7. **Temel Bileşenler Analizi (PCA)**:
    - Veri setindeki değişkenlerin, daha az sayıda bileşenle (özelliklerle) temsil edilmesini sağlar.
    - Ana bileşenler, veri setindeki en fazla varyansı açıklayan doğrusal kombinasyonlardır.

2.1.ANOVA

ANOVA, birden fazla grubun ortalamalarını karşılaştırarak bu gruplar arasında anlamlı fark olup olmadığını belirlemek için kullanılan istatistiksel bir yöntemdir. Temel amacı, gruplar arasındaki varyansın, grup içi varyanstan anlamlı derecede büyük olup olmadığını belirlemektir.

2.1.1ANOVA'nın Temel Kavramları ve Formülü

1. **Gruplar Arası Varyans:**
    - Grupların ortalamaları arasındaki farkların toplam varyansıdır.

      Formül:

$$GAKT = \sum_{i=1}^{k} n_i(\bar{X} - X)^2$$

2. **Grup İçi Varyans**:
    - Her grubun kendi içindeki varyansıdır.

      Formül:

$$GIKT = \sum_{i=1}^{k} \sum_{j=1}^{n_i} (X_{ij} - \bar{X}_i)^2$$

3. **Toplam Varyans**:
    - Tüm veri setinin toplam varyansıdır.

Formül:

$$GNKT=GAKT+GİKT$$

4. **F Test İstatistiği**:
   - Gruplar arası varyans ile grup içi varyansın oranını hesaplar.

   Formül:

$$F=\frac{GAKO}{GIKO} \sim F_{\propto,k-1,n-k-1}$$

$$GAKO = \frac{GAKT}{GA_{sd}} \qquad\qquad GIKO = \frac{GIKT}{GI_{sd}}$$

GAKO: Gruplar arası kareler ortalaması

GIKO: Grup içi kareler ortalaması

2.1.2ANOVA'nın Adımları

1. **Hipotezlerin Belirlenmesi**:
   - H_0 Hipotezi: Gruplar arasında ortalama farkı yoktur.($\mu_1= \mu_2= \mu_3=\ldots=\mu_k$)
   - H_1 Hipotez: En az bir grup diğerlerinden farklıdır.
2. **F Değerinin Hesaplanması**:
   - Gruplar arası ve grup içi varyans hesaplanır.
   - F değeri hesaplanır.
3. **p-değerinin Hesaplanması**:
   - F dağılımı kullanılarak p-değeri hesaplanır.
   - p-değeri, belirli bir anlamlılık düzeyi ile karşılaştırılır.
4. **Karar Verme**:
   - p-değeri < 0.05 ise H_0 hipotez reddedilir, gruplar arasında anlamlı fark vardır.
   - p-değeri ≥ 0.05 ise H_0 hipotez reddedilemez, gruplar arasında anlamlı fark yoktur.

   (Anlamlılık düzeyi 0.05 olarak varsayıldığında)

2.2.MANOVA

MANOVA, bir veya daha fazla bağımsız değişkenin birden fazla bağımlı değişken üzerindeki etkisini analiz etmek için kullanılan bir istatistiksel yöntemdir. Birden fazla bağımlı değişkenin aynı anda değerlendirilmesine olanak tanır. Gruplar arasındaki farkları belirlemek ve bu farkların hangi bağımlı değişkenlerde olduğunu anlamak için kullanılır.

2.2.1.Varsayımlar

1. İki ya da daha fazla bağımlı değişken olmalı.
2. İki ya da daha fazla gruplu bir bağımsız değişken olmalı.
3. Gözlemler bağımsız olmalı.
4. Örneklem büyüklüğü yeterli olmalı.
5. Tek veya çok değişkenli aykırı değerler olmamalı.
6. Veriler, çoklu normal dağılım göstermeli.
7. Bağımsız değişkenin her grubu için her bir bağımlı değişken çifti arasında doğrusal bir ilişki olmalı.
8. Varyans-kovaryans matrisleri homojen olmalı.
9. Çoklu doğrusallık olmamalı.

Arslan (2024)

2.2.2.Hipotez Aşaması

1. **Hipotezlerin Belirlenmesi**:
   - **H_0 Hipotezi**: Bağımsız değişkenlerin bağımlı değişkenler üzerinde anlamlı bir etkisi yoktur. Yani, gruplar arasında bağımlı değişkenlerin ortalamaları arasında fark yoktur.
   - **H_1 Hipotez:** Bağımsız değişkenlerin bağımlı değişkenler üzerinde anlamlı bir etkisi vardır. Yani, en az bir bağımlı değişkenin ortalaması gruplar arasında farklıdır.
2. **Veri Hazırlığı**:
   - Bağımsız değişkenler ve bağımlı değişkenler tanımlanır ve veri seti oluşturulur.
3. **Kovaryans Matrisi Hesaplama**:
   - Bağımlı değişkenler arasındaki kovaryanslar hesaplanır. Bu, bağımlı değişkenlerin birbiriyle ilişkilerini anlamak için önemlidir.

4. **Test İstatistiklerinin Hesaplanması**:

- **Wilks' Lambda**: Hipotez matrisi ve hata matrisi oranı.
- **Pillai's Trace**: Bağımsız değişkenlerin bağımlı değişkenlerin varyansının ne kadarını açıkladığını gösterir.
- **Hotelling-Lawley Trace**: Hipotez matrisi ile hata matrisi oranının izini kullanır.
- **Roy's Greatest Root**: Bağımsız değişkenlerin bağımlı değişkenler üzerindeki en büyük tek varyans bileşenini ölçer.

(Şen, 2016)

5. **F Değerinin ve p-değerinin Hesaplanması**:

- Her bir test istatistiği için F değeri ve p-değeri hesaplanır. p-değeri, belirli bir anlamlılık düzeyine göre değerlendirilir (genellikle 0.05).

6. **Karar Verme**:

- p-değeri < 0.05 ise H_0 hipotez reddedilir. Bu durumda, bağımsız değişkenlerin bağımlı değişkenler üzerinde anlamlı bir etkisi vardır.
- p-değeri ≥ 0.05 ise H_0 hipotez reddedilemez. Bu durumda, bağımsız değişkenlerin bağımlı değişkenler üzerinde anlamlı bir etkisi yoktur.

2.2.3. Normallik Dönüşümleri

Bağımlı değişkenlerin her bir grup için çok değişkenli normal dağılım göstermesi gerekir. Normalliği sağlamak için aşağıdaki dönüşümler uygulanabilir.

- **Logaritmik Dönüşüm:**

Veri setindeki değerlerin logaritmasının alınması ile dönüşüm işlemi gerçekleştirilir. Veri setindeki her bir değer (x) işlem sırasında 10 veya 2'lik tabanda log(x) değerine dönüştürülür. (Dayanıklı, 2021)

- **Boxcox Dönüşümü:**

İsmini iki istatistikçi George Box ve Sir David Roxbee Cox'tan alan Box-Cox dönüşümü lambda (λ) parametresini kullanarak veri seti üzerinde birden çok test uygulayıp, normal

dağılıma en yakın sonucu üretir. Lambda (λ) parametresi -5 ile 5 arasında değer alır. (Dayanıklı, 2021)

Zaman serilerine de uyarlanabilir.

Box-cox dönüşümünün uygulanabilmesi için veri setindeki değerler pozitif olmalıdır.

- **Yeo-Johnson Dönüşümü:**

Box-Cox gibi adını yaratıcılarından alan bir diğer dönüşüm metodu Yeo-Johnson, veri setindeki değerlerin pozitif olma şartı olmadan kullanılabildiği bir dönüşüm metodudur. (Dayanıklı, 2021)

2.2.4.Temel İstatistiklerini Belirlemek

1. **Roy'un En Büyük Karakteristik Kök Yöntemi**

Eğer bağımlı değişkenler arası ilişki çok yüksek ise kullanılır. Varsayımlar sağlanmış ise en güçlü testtir.

BW^{-1} matrisinin öz değerleri bulunur.

$|BW^{-1} - \lambda I|=0$ bulunan p tane öz değerinin en büyüğü alınır.

$$Q = \frac{\lambda}{1+\lambda} \sim F_{(2m+2;2\tilde{n}+2)}$$

$$\tilde{n} = (n-k-p-1)/2$$

$Q<F_{(2m+2;2\tilde{n}+2)}$ → H_0 Reddedilir

Q'nun 1'e yaklaşmasında H_0 reddedilir, 0'a yaklaşmasında H_0 reddedilemez.

2. **Wilks'in Olabilirlik Oran Yöntemi**

En çok kullanılan istatistiktir. Küçük lambda değerleri gruplar arası dağılımın büyük olduğunu işaret eder.

$$\Lambda = \frac{|W|}{|W+B|}$$

W=Hata Varyans Kovaryans Matrisi

B=Hipotez Varyans Kovaryans Matrisi

3. Hotelling İz Yöntemi

Tüm karakteristik kökler arası genel farklılığı hesaba katar.

$$T_o^2 = \sum_{r=1}^{p} \lambda_r$$

$$\text{HL} = nT_o^2 \sim \chi^2_{p(k-1)}$$

$$\text{HL} > \chi^2_{p(k-1)} \rightarrow \quad H_0 \text{ Reddedilir}$$

4. Pillai's Trace

Tüm karakteristik kökler arası genel farklılığı hesaba katar. Varyans homojenliği ihlal edildiğinde, örneklem sayısı küçük olduğunda ve eşit olmayan gruplar olduğunda daha dirençli olduğu için tercih edilir.

$$\theta_1 = max(\lambda_i)$$

Bu istatistiklerin güç değerleri farklı durumlara göre değişir. Eğer farklılık ilk değişkenden kaynaklanıyorsa güç sıralaması Roy's statistic > Hotelling's trace > Wilks's lambda > Pillai's trace. Eğer farklılık birden fazla değişkenden kaynaklanıyorsa güç sıralaması ters yöndedir.

2.3.DİSKRİMİNANT ANALİZİ

Diskriminant analizi, üzerinde ölçüm yapılan bir bireyi sonlu sayıda bilinen farklı kitleden birine atanmasını gerçekleştiren istatistiksel bir tekniktir. (Atakan & Karabulut, 2024)

2.3.1.Varsayımlar

- Bütün bağımsız değişkenlerin normal dağılıma sahip olmalıdır.

- Veride uç değer bulunmamalıdır.

- Her kategorinin kendi içinde, bütün bağımsız değişken çiftleri arasında doğrusal bir ilişki olmalıdır.

- Bağımsız değişkenler arasında çoklu doğrusallık olmamalıdır.

(Şavkay, 2024)

2.3.2. Diskriminant Analizi Adımları

- **Veri Hazırlığı**:
  - Bağımlı ve bağımsız değişkenlerin belirlenmesi.
  - Verilerin normallik, varyans homojenliği ve bağımsızlık gibi varsayımları karşılayıp karşılamadığının kontrol edilmesi.

- **Diskriminant Fonksiyonunun Oluşturulması**:
  - Bağımsız değişkenlerin doğrusal kombinasyonu ile diskriminant fonksiyonu oluşturulur.

$D = a + w1X1 + w2X2 + \cdots + wnXn$

D Diskriminant fonksiyonu, a sabit terim, w bağımsız değişkenlerin katsayısıdır.

- **Modelin Eğitilmesi**:
  - Veriler kullanılarak diskriminant fonksiyonu oluşturulur ve bağımsız değişkenlerin katsayıları belirlenir.

- **Modelin Değerlendirilmesi**:
  - Sınıflandırma doğruluğu, confusion matrix (karışıklık matrisi) ve doğruluk, hassasiyet, özgüllük gibi performans ölçütleri ile değerlendirilir.

- **Yeni Gözlemlerin Sınıflandırılması**:
  - Yeni veri noktaları kullanılarak hangi sınıfa ait oldukları tahmin edilir.

2.3.3.Fisher'in Fonksiyonu

1936 yılında R. A. Fischer tarafından geliştirilen bir sınıflama metodudur. Basit olmasına rağmen kompleks problemlerde iyi sonuçlar üreten bir modeldir. Bu yazımda bu modeli basit bir örnek üzerinden anlatmaya çalışacağım.

Lineer Diskriminant Analizi, iyi sınıf (hedefler) arasında en iyi şekilde ayıran değişkenleri lineer bir kombinasyonunu aramaya dayanır. Fisher aşağıdaki score fonksiyonunu tanımlar. (Uzun, 2024)

$$Z = \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_k x_k$$

$S(\beta) = \frac{\beta^T \mu_1 - \beta^T \mu_2}{\beta^T C \beta}$: Score Fonksiyonu

$$S(\beta) = \frac{\overline{Z_1} - \overline{Z_2}}{Z'nin\ gruptaki\ varyansı}$$

Score fonksiyonuna göre, problem score'u maksimize eden lineer katsayıları tahmin etmedir.

$\beta = C^{-1}(\mu_1 - \mu_2)$: Model Katsayıları

$C = \frac{1}{n_1+n_2}(n_1 C_1 + n_2 C_2)$: Toplanmış Kovaryans Matrisi

β : Lineer model katsayıları

C_1, C_2 : Kovaryans matrisi

μ_1, μ_2 : Ortalama vektörler

En iyi diskriminantı belirleme yolu iki grup arasında Mahalanobis mesafesini hesaplamaktır. Mahalanobis mesafesinin üçten küçük olması yanlış sınıflandırma olasılığını oldukça küçük olduğunu anlamına gelir.

$$\Delta^2 = \beta^T(\mu_1 - \mu_2)$$

Δ : İki grup arasındaki Mahalanobis farkı

2.4. LOJİSTİK REGRESYON ANALİZİ

Lojistik regresyon analizi, bağımlı değişkenin kategorik olduğu durumlarda, bir veya daha fazla bağımsız değişkenin bu bağımlı değişken üzerindeki etkisini modellemek için kullanılan bir istatistiksel yöntemdir. (Terzi, Y.)

2.4.1. Lojistik Regresyon Yöntemleri

Lojistik regresyonda üç temel yöntem bulunmaktadır: ikili (binary), sıralı (ordinal) ve nominal lojistik regresyon yöntemleri.

2.4.1.1. İkili Lojistik Regresyon (Binary)

İkili lojistik regresyon, bağımlı değişkenin iki kategorik cevaptan oluştuğu durumlarda uygulanan bir analiz yöntemidir. Açıklayıcı değişkenler faktör veya sürekli değişkenler olabilir. Faktör değişkenler, isimsel ölçekli kategorik değişkenler iken, sürekli değişkenler, ölçülebilir değerlerdir.

Lojistik regresyonda "odds oranı" kullanılır. Odds oranı (OR), bir olayın gerçekleşme olasılığının gerçekleşmeme olasılığına oranı olarak tanımlanır. Odds, başarı veya gerçekleşme olasılığı olan **P** ile başarısızlık veya gerçekleşmeme olasılığı olan **1−P** oranı olarak ifade edilir. Odds değeri **0** ile +∞ arasında değer alır ve şu şekilde hesaplanır:

$$Odds = \frac{P}{1-P}$$

Odds Oranı (OR), iki farklı odds'un birbirine oranıdır ve iki değişken arasındaki ilişkinin özet bir ölçüsüdür. Lojistik regresyonda OR, bağımsız değişkenin katsayısının üssü alınarak hesaplanır:

$$OR = \exp(\beta)$$

Bu, bağımsız değişkenin bir birim artması durumunda, olayın gerçekleşme olasılığının nasıl değiştiğini gösterir. Yani, eğer OR>1 ise bağımsız değişkenin artışı, olayın gerçekleşme olasılığını artırır; eğer OR<1 ise, bağımsız değişkenin artışı, olayın gerçekleşme olasılığını azaltır. (Terzi, Y.)

2.4.1.2. Lojit Fonsiyonu

Lojit fonksiyonu, bir olasılığın odds oranının doğal logaritması olarak tanımlanır. Odds oranı, bir olayın gerçekleşme olasılığının gerçekleşmeme

olasılığına oranıdır. Lojit fonksiyonu, bu oranı simetrik bir hale getirmek için kullanılır.

Lojit fonksiyonu, olasılık **P** ve odds oranı **OR** kullanılarak aşağıdaki gibi tanımlanır:

$$Odds = \frac{P}{1-P}$$

$$Lojit(P) = \ln\left(\frac{P}{1-P}\right)$$

Lojit fonksiyonu, odds oranının doğal logaritmasını alarak asimetrik dağılımı simetrik bir hale dönüştürür. Bu, olasılıklar arasındaki farkları daha anlaşılır kılar ve lojistik regresyon modellerinde kullanılır.

(Terzi, Y.)

Lojistik Regresyon Modeli

$$z = \beta_0 + \beta_1 X_1 + \cdots + \beta_p X_p$$

$$P = \frac{e^z}{1+e^z} = \frac{1}{1+e^{-1}}$$

$$Lojit(P) = ln\left(\frac{P}{1-P}\right) = \ln(e^z) = z = \beta_0 + \beta_1 X_1 + \cdots + \beta_p X_p$$

$$\frac{P}{1-P} = e^{\beta_0+\beta_1 X_1+\cdots+\beta_p X_p} = OR$$

P: İncelenen olayın gözlenme olasılığı

(Terzi, Y.)

2.4.2.Varsayımlar

- Analizde kullanılacak bağımsız değişkenler, modele uygun ve anlamlı olmalıdır.
- Analize uygun olmayan ve modele katkı sağlamayan bağımsız değişkenler dışlanmalıdır.
- Aynı birey üzerinde sadece bir kez gözlem yapılmalıdır. Tekrarlayan ölçümler modelin geçerliliğini azaltır.

- Ölçüm hataları küçük olmalı ve kayıp (eksik) veri olmamalıdır. Hatalar, katsayıların tahmininde yanlılığa ve modelin yetersizliğine neden olabilir.
- Bağımsız değişkenler arasında yüksek korelasyon olmamalıdır. Çoklu bağlantı, modelin tahmin performansını olumsuz etkiler.
- Aşırı değerler, modelin doğruluğunu ve güvenilirliğini etkileyebilir. Bu nedenle, aşırı değerlerin tespiti ve analizi yapılmalıdır.
- Analizin güvenilir sonuçlar verebilmesi için yeterli büyüklükte bir örneklem kullanılmalıdır.
- Bağımlı değişkenin beklenen varyansı ile gözlenen varyansı arasında büyük bir fark varsa, model yetersiz olabilir ve yeniden tanımlanması gerekebilir. Sonuç değişkeni ikili (binary) değerler aldığından, hata terimi sıfır ortalamalı ve $P(1-P)$ varyanslı Binom dağılımına sahiptir.

(Terzi, Y.)

2.4.3.Lojistik Regresyon Analizi Adımları

1. **Modele dahil edilecek bağımsız değişkenler belirlenir.** Bu aşamada önsel bilgilerden ya da istatistiksel tekniklerden yararlanılabilir.
2. **Modelin parametreleri tahmin edilir.** Ardından modelin genel anlamlılığı olabilirlik oranı testi ile değerlendirilir. Eğer model anlamlı bulunmazsa analize son verilir. Model anlamlı bulunduğunda ise diğer aşamaya geçilir.
3. **Tahmin edilen model parametrelerinin tek tek anlamlılığı incelenir.** Bu amaçla olabilirlik oranı testi veya Wald istatistiği kullanılabilir. Her katsayının anlamlılığı incelendikten sonra, odds oranları değerlendirilerek açıklayıcı değişkenlerin bağımlı değişken üzerindeki etkileri yorumlanır.
4. Tahmin edilen model parametreleri kullanılarak, **her bir gözlemin hangi gruptan geldiği tahmin edilir.**
5. **Modelin uyum iyiliğini değerlendirmek** amacıyla doğru sınıflandırma yüzdesi ve Hosmer-Lemeshow testi gibi ölçütler kullanılır. Modelin uyum iyiliği kabul edilebilir düzeyde ise bu aşamadaki grup tahminleri kullanılabilir. Aksi halde, 2. aşamaya geri dönülerek modele girecek değişkenler yeniden gözden geçirilir ve işlemler tekrar edilir.

(Terzi, Y.)

2.4.4.Hosmer-Lemeshow İstatistiği

Hosmer-Lemeshow istatistiği, lojistik regresyon modelinin genel uyumunu test eden bir istatistiktir ve Model Ki-kare istatistiği olarak da bilinir. Bu istatistik, lojistik regresyon modelinin uygunluğunu değerlendirmek için kullanılır. Yokluk hipotezi (H_0) şu şekilde kuruludur:

H_0: Sabit terim dışındaki tüm katsayılar sıfırdır.

Hosmer-Lemeshow istatistiği, olabilirlik oranı testidir ve modelde bağımsız değişkenlerin olmadığı -2 log L_0 istatistiği ile modelde bağımsız değişkenlerin yer aldığı -2 log L istatistiği arasındaki fark alınarak hesaplanır. Bu istatistik, incelenen modelin parametre sayısı ile sabit terimli modelin parametreleri arasındaki fark kadar serbestlik derecesi olan ki-kare dağılımına uyar. Modelin anlamlı olması arzu edilir ($p < 0.05$). (Terzi, Y.)

2.4.5. -2 log L İstatistiği

-2 log L istatistiği, sapma ki-kare istatistiği olarak bilinir. Bu istatistiğin anlamlı olmaması, lojistik regresyonda istenen durumu gösterir. -2 log L istatistiği, analize bağımsız değişken ilave edildiğinde modelin hatasını gösterir. Bu nedenle, -2 log L istatistiği, bağımlı değişkendeki açıklanmayan varyansın anlamlılığını gösterir. Log olabilirlik (log likelihood) değeri 0 ile 1 arasında değerler alır ve bu oran, bağımlı değişkenin bağımsız değişkenler tarafından tahmin edilme olasılığını gösterir. 1'den küçük sayıların logaritması 0 ile -∞ arasındadır. LogL istatistiği, maksimum olabilirlik algoritması ile tahmin edilmektedir ve -2 log L istatistiği yaklaşık olarak ki-kare dağılımına uyar. Lojistik regresyon analizinde -2 log L istatistiği, regresyon analizindeki hata kareler toplamına (HKT) benzer. (Terzi, Y.)

2.4.6.Cox-Snell R^2 ve Nagelkerke R^2

R^2, bağımlı değişkenin açıklanan varyansının yüzdesini gösterir. Ancak lojistik regresyonda bağımlı değişkenin varyansı, bu değişkenin olasılık dağılımına (frekans dağılımı) bağlıdır. İki gruplu bir bağımlı değişkenin varyansı, grup frekansları eşit olduğu zaman ($0.5 * 0.5 = 0.25$) maksimum olur. Bu açıdan, regresyon analizindeki R^2 ile lojistik regresyondaki R^2 farklıdır.

Lojistik regresyonda, bağımlı ve bağımsız değişkenler arasındaki ilişkinin gücünün ölçülmesinde kullanılan iki sözde R^2 istatistiği vardır. Bu istatistikler, Cox-Snell R^2 ve Nagelkerke R^2 istatistikleridir (Kalaycı, 2005). Bu değerlerin 0.2'den büyük olması istenir.

(Terzi, Y.)

2.4.7.Modelin Uygunluğunun Test Edilmesi

Oluşturulan birçok modelin uygunluğunu test etmek, değerlendirmek ve bu modeller arasından en uygun olanı seçmek için tüm gözlem değerlerini temsil edecek bir istatistiksel değere ihtiyaç duyulur. Modelin uygunluğunu test etmek için Pearson ki-kare istatistiği, sapma ölçüsü ve sözde R^2 değeri yaygın olarak kullanılır. Ancak, sözde R^2 değeri kategorik bağımlı değişkenin yer aldığı modeller için kesin sonuçlar vermez (Long, 1997). Ayrıca, gözlem sayısının az olduğu çalışmalarda sapma ölçüsü yetersiz kalabilir.

Model Ki-kare anlamlı ($p < 0.05$), Cox-Snell R^2 ve Nagelkerke R^2 0.2'den büyükse modelin anlamlı olduğu söylenebilir. Hosmer-Lemeshow testinde ise $p > 0.05$ ise modelin veriye uyumunun iyi olduğuna karar verilir.

(Terzi, Y.)

2.5.KÜMELEME

Kümelenme analizi, bir veri setinin içinde farklı grupların olup olmadığını belirlemek ve eğer varsa bu grupları tespit etmek için kullanılan çok değişkenli istatistiksel bir yöntemdir.

Kümelenme analizi, çok boyutlu uzaydaki verilerin özetlenmesi ve tanımlanmasında rehberlik eden bir araştırma yöntemi olarak kullanılır. Bu yöntem, heterojen gruplardaki farklı gözlem yapıları ile homojen gruplardaki benzer gözlemleri uygun yöntemlerle sınıflandırır ve gruplar.

Diğer çok değişkenli istatistiksel yöntemlerden farklı olarak, kümelenme analizi normallik, doğrusallık ve homojenlik varsayımlarına dayanmaz. Bu yöntem, prensipte kalarak, veri setindeki uzaklık değerlerinin normalliğini dikkate almaz ve bu sayede daha esnek bir analiz imkânı sunar.

2.5.1. Uzaklık Ölçütleri

a. Minkowski uzaklığı

$$d_{\lambda}(x_i, x_j) = [\sum_{k=1}^{p} (x_{ik} - x_{jk})^{\lambda}]^{1/\lambda}$$

b. Manhattan City Blok Uzaklığı ($\lambda = 1$)

$$d_1(x_i, x_j) = [\sum_{k=1}^{p} (x_{ik} - x_{jk})]$$

c. Öklit Uzaklığı ($\lambda = 2$)

$$d_2(x_i, x_j) = [\sum_{k=1}^{p} (x_{ik} - x_{jk})^2]^{1/2}$$

d. Ölçekli Öklit Uzaklığı

$$d_2(x_i, x_j) = [\sum_{k=1}^{p} {W_k}^2 (x_{ik} - x_{jk})^2]^{1/2}$$

e. Mahallanobis Uzaklığı

$$D^2 = (x_i - x_j)' S^{-1} (x_i - x_j)$$

f. Hotelling T^2

$$T^2 = \frac{n_1 n_2}{n} (\overline{x_i} - \overline{x_j})' S^{-1} (\overline{x_i} - \overline{x_j})$$

2.5.2.Kümeleme Yöntemleri

a. Sıralı Yöntemler (Hiyerarşik)

Veri noktalarını hiyerarşik bir yapı içinde gruplandırmak için kullanılan bir yöntemdir. Bu yöntem, veri noktalarının benzerliklerine veya

farklılıklarına dayanarak küçük kümeleri birleştirir veya büyük kümeleri bölerek daha küçük kümeler oluşturur. Aşağıda, hiyerarşik kümelenme yönteminin adımları sıralı olarak açıklanmıştır:

Hiyerarşik Kümelenme Adımları

1- Her bir birey (gözlem) kendi başına bir küme oluşturur. Yani, başlangıçta n tane birey n tane kümeye sahiptir.

2- İki en yakın gözlemi veya kümeyi bulmak için uzaklık matrisi hesaplanır. Bu uzaklık, genellikle Euclidean uzaklığı veya başka bir mesafe metriği kullanılarak hesaplanır.

3- En yakın gözlemler veya kümeler birleştirilir ve yeni bir küme oluşturulur. Bu işlem sonucunda, toplam küme sayısı bir azalır.

4- Yeni oluşan kümeler arasındaki uzaklıkları içeren uzaklık matrisi tekrar hesaplanır. Bu işlem, kümeler arasındaki mesafeyi günceller ve yeni mesafeleri belirler.

5- Adım 2 ve 3, tüm gözlemler tek bir kümede birleşene kadar tekrarlanır. Her birleştirme adımında küme sayısı bir azalır ve uzaklık matrisi güncellenir.

6- Tüm gözlemler tek bir kümede birleştiğinde, hiyerarşik ağaç (dendrogram) tamamlanır. Bu ağaç, veri noktalarının birleşme sırasını ve mesafelerini gösterir.

b. Sıralı Olmayan Yöntemler (Hiyerarşik olmayan, K-means)

K-Ortalamalar kümeleme yöntemi, veri setini K adet önceden belirlenmiş sayıda küme ile gruplar. Her kümenin merkezi hesaplanır ve veri noktaları en yakın merkez noktaya atanır. Bu işlem, küme merkezleri normal hale gelene kadar tekrarlanır.

K-Ortalamalar Kümelenme Adımları

1- Küme sayısı *K* önceden belirlenir.

2- İlk olarak K tane rastgele merkez noktası seçilir. Bu merkezler, başlangıç kümelerinin merkezleri olarak kullanılır.

3- Her veri noktası, en yakın merkez noktaya atanır. Bu işlem, tüm veri noktaları en yakın merkez noktaya atanana kadar devam eder.

4- Her küme için yeni merkez noktası, o kümedeki veri noktalarının ortalaması alınarak hesaplanır.

5- Adım 3 ve 4, küme merkezleri stabil hale gelene kadar tekrarlanır. Küme merkezleri değişmediğinde veya değişiklikler çok küçük olduğunda işlem sonlandırılır.

2.6.TEMEL BİLEŞEN ANALİZİ

Regresyon analizinde değişkenler arasındaki ilişki çoklu bağlantı sorununa neden olabilir. Bu tür sorunları gidermek amacıyla Temel Bileşenler Analizi uygulanur.

Temel Bileşenler Analizi, başlangıçtaki ilişkili p değişkenden daha az sayıda ve birbirleriyle ilişkisiz yeni değişkenler türetmeyi amaçlar. Bu yeni değişkenler, başlangıçtaki değişkenlerin doğrusal kombinasyonlarıdır ve temel bileşenler olarak adlandırılır.

Temel Bileşenler Analizinin amaçları genel olarak şunlardır:

- Boyut indirgemek
- Değişkenler arasındaki ilişki yapısını ortadan kaldırmak
- Diğer istatistiksel analizler için veri hazırlamak

(Bulut, 2024)

2.6.1.Temel Bileşen Analizinin Gereklilik Testi

$$H_0: R = I$$

$$H_1: R \neq I$$

$$-\left[(n-1)-\frac{1}{6}(2p+5)\right]\log|R| \quad \sim \quad \chi^2_{\frac{p(p-1)}{2}}$$

Veya

$$-\left[(n-1)-\frac{1}{6}(2p+11)\right]\log|R| \quad \sim \quad \chi^2_{\frac{p(p-1)}{2}}$$

$$Hesaplanan\ Değer > \chi^2_{\frac{p(p-1)}{2}}$$

H_0 Red → TBA uygulanır.

2.6.2. Boyut İndirgemede Ampirik Kural

Boyut indirgeme, yüksek boyutlu veri setlerini daha düşük boyutlu bir uzaya indirgerken veri setindeki önemli bilgiyi korumayı amaçlar. Ampirik kurallar, boyut indirgeme sürecinde hangi bileşenlerin seçileceğine karar vermede yardımcı olur. Temel Bileşenler Analizi yöntemlerinde sıkça kullanılan ampirik kurallar arasında Andersan özdeğer artışı, korelasyon matrisinin spektral dekompozisyonu ve scree plot bulunmaktadır.

$\sum_{i=1}^{p}\frac{\lambda_i}{p} \geq \frac{2}{3}$ kuralı (Ampirik Kuralı)

H_0: $\lambda_{m+1} = \lambda_{m+2} = \cdots = \lambda_{m+g} = 0$

H_1: $\lambda_{m+i} \neq \lambda_{m+l}$, $l: 1, \ldots, g$

m tane temel birleşen, g tanesi önemsiz testtir.

1.Yol:

$$2log\ \lambda = \left(\mathrm{n} - \frac{2p+11}{6}\right)(p-m)\log\left(\frac{a_0}{g_0}\right) \sim \chi^2_{(p-m+2)(n-m-1)/2}$$

a_0: İhmal edilecek özdeğerlerin aritmetik ortalaması

g_0: İhmal edilecek öz değerin geometrik ortalaması

2.Yol: Andersan Özdeğer Artışı

Andersan özdeğer artışı, özdeğerlerin büyüklük sırasına göre dizildiğinde belirgin bir artış veya düşüş noktası aramayı içerir. Bu noktalar, önemli bileşenleri belirlemeye yardımcı olur.

$$(n-1)\sum_{j=m+1}^{m+g} log\,\lambda_j + (n-1)g log\frac{\sum_{j=m+1}^{m+g}\lambda_j}{g} \sim \chi_{\frac{g+(g+1)}{g}}{}^2 - 1$$

3.Yol: Korelasyon Matrisinin Spektral Dekompozisyonu

Spektral dekompozisyon, bir matrisin özdeğerler ve özvektörler kullanılarak ayrıştırılmasıdır. TBA'da, veri setinin kovaryans veya korelasyon matrisi ile analiz edilir.

$$R = R_1 + R_2 + \cdots + R_p$$

$$\sum_{j=1}^{m} R_j = R_1 + R_2 + \cdots + R_p = R_h$$

$$\sum_{j=m+1}^{p} R_j = R_{m+1} + R_{m+2} + \cdots + R_p = R_g = R - R_h$$

$$H_0: |R_g| = 0$$

$$H_1: |R_g| \neq 0$$

$$U_g = \frac{|R|g^g}{(\prod_{j=1}^{m}\lambda_j)(p - \sum_{j=1}^{m}\lambda_j)^g}$$

$$-\left[(n-1) - \frac{1}{6}(2p+5) - \frac{2m}{3}\right] log U_g \;>\; \chi^2_{\frac{g(g+1)}{2}-1}$$

Yukarıdaki eşitlik durumunda H_0 hipotezini reddederiz.

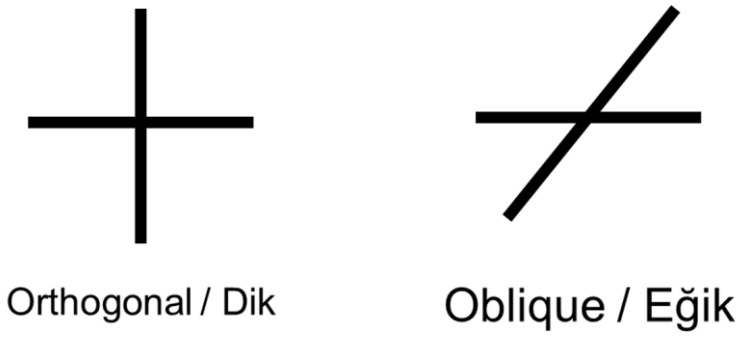
H_0 Red durumunda H_0 Reddedilemez durumuna kadar m artırılır.

4.Yol: Scree Plot

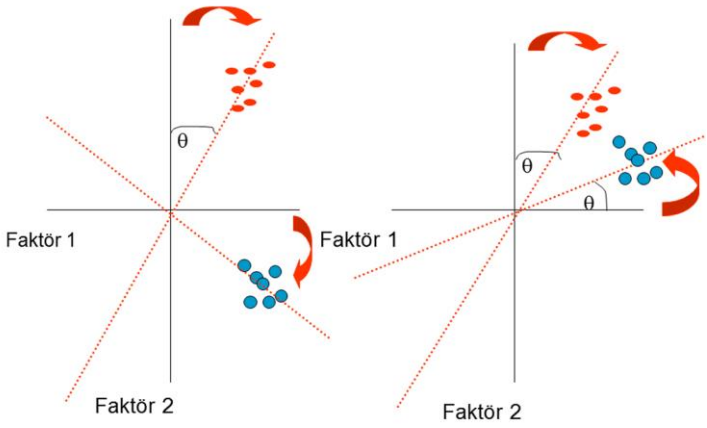
Scree plot, özdeğerlerin bileşen sayısına karşı grafiğe döküldüğü bir yöntemdir. Grafikte özdeğerlerin belirgin bir şekilde azalmaya başladığı noktayı belirlemek için kullanılır.

2.7.FAKTÖR ANALİZİ

Faktör analizi, birbirleriyle ilişkili olan p sayıda değişkeni bir araya getirerek, az sayıda ilişkisiz ve kavramsal olarak anlamlı yeni değişkenler türetmeyi amaçlayan birçok değişkenli istatistik yöntemidir. (Terzi, 2019). Veri setini küçülterek daha anlaşılır ve yorumlanabilir hale getirmeyi hedefler. Her faktör için en az 5 değişken, her değişken için en az 10 gözlem olmalıdır.



Şekil.1.: Dik ve Eğik Döndürme Grafiği 1



Şekil.2.: Dik ve Eğik Döndürme Grafiği 2

Dik Döndürme (Orthogonel Rotation)

Quartimex: Faktör yükleri matrisinin matris satırlarını birim yapmaya çalışır.

Varimax: Faktör yükleri matrisinin matris sütunlarını birim yapmaya çalışır.

Equamax: Faktör yükleri matrisinin matris satır ve sütunlarını birim yapmaya çalışır.

Eğik Döndürme (Oblique Rotation)

- Verilen noktaların eksenler üzerindeki izdüşümleri eksenlere paralel doğrularla bulunuyorsa, bu yük değerlerine örüntü yükleri (pattern loading) denir.

- Noktaların eksenler üzerindeki izdüşümleri eksenlere dik doğrularla bulunuyorsa, bu yük değerlerine yapı yükleri (structure loading) denir.
- Eğik döndürmede, bu tanımlanan döndürmeden sonra bulunan eksenlere "temel eksen" (principel axes) denir.

$$P=AT$$

2.8.RANDOM FOREST

Random Forest algoritması, birden çok karar ağacı oluşturarak her bir ağacı farklı gözlem örnekleri üzerinde eğiterek sınıflandırma ve regresyon problemlerini çözer. Kullanım kolaylığı sayesinde hem sınıflandırma hem de regresyon problemlerinde yaygın olarak benimsenmiştir. Algoritmanın en beğenilen özelliği, veri kümesini çeşitli modellerle yeniden ve daha derinlemesine keşfetme imkânı sunmasıdır.

3.MATERYAL VE UYGULAMA

3.1.Veri Tanımı

```
data = pd.read_excel('data.xlsx',engine='openpyxl')
data.head()
```

	id	age	bp	sg	al	su	rbc	pc	pcc	ba	...	pcv	wc	rc	htn	dm	cad	appet	pe	ane	classification
0	1	7.0	50.0	1.02	4.0	0.0	NaN	normal	notpresent	notpresent	...	38.0	6000	NaN	no	no	no	good	no	no	ckd
1	2	62.0	80.0	1.01	2.0	3.0	normal	normal	notpresent	notpresent	...	31.0	7500	NaN	no	yes	no	poor	no	yes	ckd
2	4	51.0	80.0	1.01	2.0	0.0	normal	normal	notpresent	notpresent	...	35.0	7300	4.6	no	no	no	good	no	no	ckd
3	6	68.0	70.0	1.01	0.0	0.0	NaN	normal	notpresent	notpresent	...	36.0	NaN	NaN	no	no	no	good	no	no	ckd
4	8	52.0	100.0	1015.00	3.0	0.0	normal	abnormal	present	notpresent	...	33.0	9600	4	yes	yes	no	good	no	yes	ckd

5 rows × 26 columns

Tablo.1.: Veri seti

```
data.shape
```

```
(280, 26)
```

Bu veri seti, kronik böbrek hastalığı olan hastalarla ilgili tıbbi verileri içermektedir. Veri seti, toplamda 26 sütun ve 280 gözlemden oluşmaktadır. Veriler, hastaların çeşitli sağlık parametrelerini ve tıbbi geçmişlerini içermektedir. Veri kaggle üzerinden alınmıştır ve kaynakçada belirtilmiştir.

3.2.Proje Tanımı

Çok Değişkenli İstatistiksel Yöntemlerle Kronik Böbrek Rahatsızlığını daha iyi anlamak ve hastalığın risk faktörlerini belirlemek amacıyla Python Jupyter Notebook üzerinden analizler gerçekleştirilmiştir.

id: Benzersiz hasta kimlik numarası	**sod:** Sodyum (mEq/L)
age: Yaş (yıl olarak)	**pot:** Potasyum (mEq/L)
bp: Kan basıncı (mm/Hg)	**hemo:** Hemoglobin (g/dL)
sg: Spesifik gravitasyon (idrar yoğunluğu)	**pcv:** Paketlenmiş hücre hacmi (vol %)
al: Albümin seviyeleri	**wc:** Beyaz kan hücresi sayısı (10^3/mL)
su: Şeker seviyeleri	**rc:** Kırmızı kan hücresi sayısı (10^6/mL)
rbc: Kırmızı kan hücresi sayısı (normal/abnormal)	**htn:** Hipertansiyon (yes/no)
pc: Hücre yığını (normal/abnormal)	**dm:** Diyabet (yes/no)
pcc: Piyüri (irin hücrelerinin varlığı)	**cad:** Koroner arter hastalığı (yes/no)
ba: Bakteri varlığı	**appet:** İştah (good/poor)
bgr: Kan glukoz rastgele (mg/dL)	**pe:** Periferik ödem (yes/no)
bu: Kan üre (mg/dL)	**ane:** Anemi (yes/no)
sc: Serum kreatinin (mg/dL)	**class:** Hastalığın sınıflandırılması (ckd/normal)

Tablo.2.: Değişken tanımları

3.3.Uygulama

3.3.1.Veri Ön İşleme

Veri setimizde her hastaya ait benzersiz bir kimlik numarası "id" bulunmaktadır. Ancak analiz sürecinde bu kimlik numarası gerekli olmadığından, veri setinden çıkarılmıştır. Bu işlem, veri analizi ve modelleme sürecini kolaylaştırmak amacıyla yapılmıştır.

```
data.drop('id', axis = 1, inplace = True)
```

Kayıp Gözlemler

Veri setimiz başlangıçta toplamda 26 sütun ve 280 gözlemden oluşmaktaydı. Ancak, bu veri setinde bazı kayıp gözlemler mevcuttu. Analiz sürecinde kayıp gözlemleri içeren satırların çıkarılması, veri kalitesini artırmak ve analiz sonuçlarının güvenilirliğini sağlamak için gerekli hale geldi.

```
data_duzen = data.dropna() # Silme fonksiyonu
print(data_duzen)
```

```
data_duzen.shape
```

```
(107, 25)
```

İşlemler sonucunda veri seti, toplamda 25 sütun ve 107 gözlemden oluşmaktadır. Bu sayede, kronik böbrek hastalığına dair yapılacak analizlerin doğruluğu ve güvenilirliği artırılmıştır.

3.3.2.Temel İstatistikler

```
data_duzen.describe()
```

	age	blood_pressure	specific_gravity	albumin	sugar	blood_glucose_random	blood_urea	serum_creatinine	sodium	potassium	haemoglobin	packed_cell_volume	white_blood_cell_count	red_blood_cell_cou
count	107.000000	107.000000	107.000000	107.000000	107.000000	107.000000	107.000000	107.000000	107.000000	107.000000	107.000000	107.000000	107.000000	107.0000
mean	49.682243	73.084112	507.288972	0.794393	0.233645	130.130841	51.635514	1.971963	138.869159	4.812150	13.601869	41.962617	8511.214953	4.8831
std	16.377964	10.764303	513.439898	1.419130	0.759586	54.841123	45.669525	2.600101	7.287990	4.175122	2.880644	9.153745	3324.872734	1.0000
min	6.000000	50.000000	1.010000	0.000000	0.000000	70.000000	10.000000	0.400000	114.000000	2.900000	3.100000	9.000000	4300.000000	2.1000
25%	38.000000	60.000000	1.020000	0.000000	0.000000	99.000000	27.000000	0.700000	135.000000	3.800000	12.550000	38.000000	6550.000000	4.5000
50%	52.000000	70.000000	1.020000	0.000000	0.000000	118.000000	39.000000	1.000000	139.000000	4.600000	14.200000	44.000000	7500.000000	5.0000
75%	61.500000	80.000000	1025.000000	1.000000	0.000000	131.000000	49.500000	1.250000	144.000000	4.900000	15.550000	49.000000	9750.000000	5.6000
max	83.000000	100.000000	1025.000000	4.000000	4.000000	380.000000	309.000000	13.300000	150.000000	47.000000	17.800000	54.000000	26400.000000	8.0000

Tablo.3.: Veri setine ait değişkenlerin temel istatistikleri

```
class_group = data_duzen.groupby('class').size().reset_index(name='count')
print(class_group)
```

```
    class  count
0     ckd     29
1  notckd     78
```

Veri setimizdeki yaş değişkeni, hastaların yaşlarını göstermektedir ve geniş bir yaş aralığını kapsamaktadır. Yaş dağılımının minimum 6, maksimum 83 ve ortalama yaş ölçüsü yaklaşık 50'dir.

Veri setindeki bağımsız değişkenimiz olan Kronik Böbrek Rahatsızlığı durumudur. 29 kişi Kronik Böbrek Rahatsızlığına sahipken, 78 kişi sağlıklıdır.

3.3.3.Normallik Testi (Shapiro-Wilk Testi)

H_0: Veriler normal dağılım göstermektedir.

H_1: Veriler normal dağılım göstermemektedir.

```
shapiro_results = columns_name.apply(lambda x: shapiro(x.dropna()), result_type='expand')

for column, result in shapiro_results.items():
    print(f"{column}: W-statistic = {result[0]}, p-value = {result[1]}")
```

```
age: W-statistic = 0.9789282083511353, p-value = 0.08707195520401001
blood_pressure: W-statistic = 0.8759690523147583, p-value = 5.608043451843514e-08
specific_gravity: W-statistic = 0.6383794546127319, p-value = 7.448638331564866e-15
albumin: W-statistic = 0.5868314504623413, p-value = 7.018968163575914e-16
sugar: W-statistic = 0.34864652156829834, p-value = 1.1424854476929615e-19
blood_glucose_random: W-statistic = 0.760567843914032, p-value = 6.424133794369347e-12
blood_urea: W-statistic = 0.6632692813873291, p-value = 2.544220835799891e-14
serum_creatinine: W-statistic = 0.571090579032898, p-value = 3.5619249503436546e-16
sodium: W-statistic = 0.932136595249176, p-value = 3.698692307807505e-05
potassium: W-statistic = 0.17598295211791992, p-value = 8.476643389010648e-22
haemoglobin: W-statistic = 0.9213405847549438, p-value = 8.85990903043421e-06
packed_cell_volume: W-statistic = 0.8972711563110352, p-value = 5.138522283232305e-07
white_blood_cell_count: W-statistic = 0.8237409591674805, p-value = 5.705718675308447e-10
red_blood_cell_count: W-statistic = 0.9707428216934204, p-value = 0.018320225179195404
```

Sonuçlara göre, sadece yaş değişkeni p-değeri 0.05'ten büyük olduğundan normal dağılıma uymaktadır. H_0 Hipotezi reddedilemez.

Diğer tüm değişkenler için p-değeri 0.05'in altındadır, bu da bu değişkenlerin normal dağılıma uymadığını göstermektedir. H_0 Hipotezi reddedilir.

Bu sonuçlara istinaden normal dağılıma uymayan değişkenler için Logaritmik dönüşüm yaparak normalleştirmeye çalışmak önemlidir. Bu da analizimizin doğruluğunu arttırmakta önemli rol oynamaktadır.

3.3.3.1.Logaritmik Dönüşüm

H_0: Veriler normal dağılım göstermektedir.

H_1: Veriler normal dağılım göstermemektedir.

```
import numpy as np

# Logaritmik dönüşüm yapılacak değişkenler
log_transform_vars = ['age','blood_pressure', 'specific_gravity', 'albumin', 'sugar',
                      'blood_glucose_random', 'blood_urea', 'serum_creatinine',
                      'sodium', 'potassium', 'haemoglobin', 'packed_cell_volume',
                      'white_blood_cell_count', 'red_blood_cell_count']

# Logaritmik dönüşüm
for var in log_transform_vars:
    df.loc[:, var] = np.log(df[var] + 1)  # 0 olan değerlerin önüne geçmek için 1 ekleyerek dönüşüm yapılır
```

```
from scipy.stats import shapiro

# Dönüşüm sonrası Shapiro-Wilk testi
shapiro_results = {}

for var in log_transform_vars:
    stat, p_value = shapiro(df[var].dropna())
    shapiro_results[var] = {'W-statistic': stat, 'p-value': p_value}

# Sonuç
for var, result in shapiro_results.items():
    print(f"{var}: W-statistic = {result['W-statistic']}, p-value = {result['p-value']}")
```

```
age: W-statistic = 0.8799936771392822, p-value = 8.36691711469939e-08
blood_pressure: W-statistic = 0.8761228322982788, p-value = 5.693619087310253e-08
specific_gravity: W-statistic = 0.6372233629226685, p-value = 7.0460774530527875e-15
albumin: W-statistic = 0.5774292945861816, p-value = 4.670333736274911e-16
sugar: W-statistic = 0.38023489713668823, p-value = 3.111801507142264e-19
blood_glucose_random: W-statistic = 0.9185633659362793, p-value = 6.238060450414196e-06
blood_urea: W-statistic = 0.9695444703102112, p-value = 0.014661029912531376
serum_creatinine: W-statistic = 0.8348392844200134, p-value = 1.3950308686005997e-09
sodium: W-statistic = 0.9108830094337463, p-value = 2.4413336632278515e-06
potassium: W-statistic = 0.6557786464691162, p-value = 1.746281579355271e-14
haemoglobin: W-statistic = 0.7690730690956116, p-value = 1.1173405950470894e-11
packed_cell_volume: W-statistic = 0.7451223731040955, p-value = 2.4313400685122e-12
white_blood_cell_count: W-statistic = 0.9750410318374634, p-value = 0.04128501936793327
red_blood_cell_count: W-statistic = 0.9165180921554565, p-value = 4.837193500861758e-06
```

Tüm değişkenler için p-değeri 0.05'in altındadır, bu da bu değişkenlerin normal dağılıma uymadığını göstermektedir. H_0 Hipotezi reddedilir.

Alınan sonuçlara göre, Logaritmik dönüşüm sonrası normalleştirme adımlarından biri olan Box-Cox dönüşümü uygularız.

3.3.3.2.Box-Cox Dönüşümü

H_0: Veriler normal dağılım göstermektedir.

H_1: Veriler normal dağılım göstermemektedir.

```python
import pandas as pd
import numpy as np
from scipy.stats import boxcox, yeojohnson, shapiro
from sklearn.preprocessing import PowerTransformer

# Box-Cox dönüşümünü yapılacak değişkenler
boxcox_vars = ['age','blood_pressure', 'specific_gravity', 'albumin', 'sugar',
               'blood_glucose_random', 'blood_urea', 'serum_creatinine',
               'sodium', 'potassium', 'haemoglobin', 'packed_cell_volume',
               'white_blood_cell_count', 'red_blood_cell_count']

# Dönüşüm uygulama
boxcox_transformed = {}
lmbdas = {}

for var in boxcox_vars:
    shifted_var = df[var] - df[var].min() + 1
    transformed, lmbda = boxcox(shifted_var.dropna())
    boxcox_transformed[var] = transformed
    lmbdas[var] = lmbda

# Box-Cox dönüşümü Shapiro-Wilk testi
shapiro_results_boxcox = {}

for var in boxcox_vars:
    stat, p_value = shapiro(boxcox_transformed[var])
    shapiro_results_boxcox[var] = {'W-statistic': stat, 'p-value': p_value}

# Sonuç
for var, result in shapiro_results_boxcox.items():
    print(f"{var}: W-statistic = {result['W-statistic']}, p-value = {result['p-value']}")
```

```
age: W-statistic = 0.9774375557899475, p-value = 0.0653688907623291
blood_pressure: W-statistic = 0.8774024844169617, p-value = 6.461002755031586e-08
specific_gravity: W-statistic = 0.6376445889472961, p-value = 7.190067124855388e-15
albumin: W-statistic = 0.5527240037918091, p-value = 1.6511841898093617e-16
sugar: W-statistic = 0.365169882774353, p-value = 1.920581808931946e-19
blood_glucose_random: W-statistic = 0.9778725504875183, p-value = 0.0710720345377922
blood_urea: W-statistic = 0.9784350991249084, p-value = 0.0791940912604332
serum_creatinine: W-statistic = 0.9539532661437988, p-value = 0.0009690001606941223
sodium: W-statistic = 0.9651666879653931, p-value = 0.006602797191590071
potassium: W-statistic = 0.9226840734481812, p-value = 1.0523666787776165e-05
haemoglobin: W-statistic = 0.9674619436264038, p-value = 0.009998559020459652
packed_cell_volume: W-statistic = 0.9483073949813843, p-value = 0.0003938291920349002
white_blood_cell_count: W-statistic = 0.9910241365432739, p-value = 0.705795168876648
red_blood_cell_count: W-statistic = 0.9763712882995605, p-value = 0.053262464702129364
```

Box-Cox dönüşümü normallik testi sonuçlarına göre, yaş, rastgele kan glukozu, kan üre, beyaz kan hücresi sayısı ve kırmızı kan hücresi sayısı değişkenleri p-değeri 0.05'ten büyük olduğundan normal dağılıma uymaktadır.

Box-Cox dönüşümü sonrası diğer değişkenler için normalleştirme adımlarından biri olan Yeo-Johnson dönüşümünü uygularız.

3.3.3.3. Yeo-Johnson Dönüşümü

H_0: Veriler normal dağılım göstermektedir.

H_1: Veriler normal dağılım göstermemektedir.

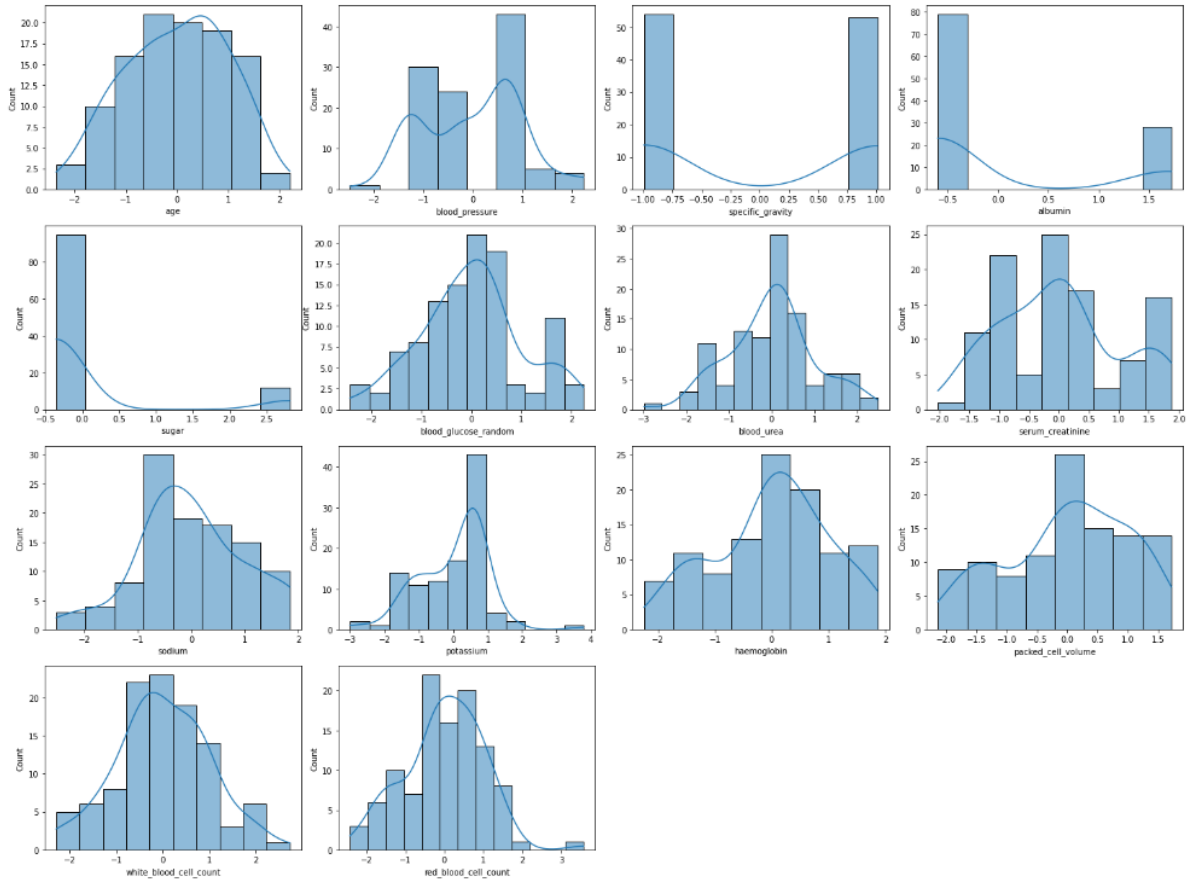
```
#Yeo-Johnson dönüşümü
yeojohnson_transformed = {}
pt = PowerTransformer(method='yeo-johnson')

for var in boxcox_vars:
    transformed = pt.fit_transform(df[var].values.reshape(-1, 1)).flatten()
    yeojohnson_transformed[var] = transformed

# Dönüşüm sonrası Shapiro-Wilk testi
shapiro_results_yeojohnson = {}

for var in boxcox_vars:
    stat, p_value = shapiro(yeojohnson_transformed[var])
    shapiro_results_yeojohnson[var] = {'W-statistic': stat, 'p-value': p_value}

for var, result in shapiro_results_yeojohnson.items():
    print(f"{var}: W-statistic = {result['W-statistic']}, p-value = {result['p-value']}")
```

```
age: W-statistic = 0.9799825549125671, p-value = 0.10661809146404266
blood_pressure: W-statistic = 0.8774587512016296, p-value = 6.497150195627 91e-08
specific_gravity: W-statistic = 0.637508749961853, p-value = 7.143278718914955e-15
albumin: W-statistic = 0.5527240633964539, p-value = 1.6511971600013665e-16
sugar: W-statistic = 0.3651696443557739, p-value = 1.920581808931946e-19
blood_glucose_random: W-statistic = 0.977777898311615, p-value = 0.06979023665189743
blood_urea: W-statistic = 0.9785634875297546, p-value = 0.08117378503084183
serum_creatinine: W-statistic = 0.9529549479484558, p-value = 0.0008238396258093417
sodium: W-statistic = 0.9651257395744324, p-value = 0.0065545677207410336
potassium: W-statistic = 0.9246078133583069, p-value = 1.35010286612669e-05
haemoglobin: W-statistic = 0.9686462879180908, p-value = 0.012421149760484695
packed_cell_volume: W-statistic = 0.9498800039291382, p-value = 0.0005039615207351744
white_blood_cell_count: W-statistic = 0.9910626411437988, p-value = 0.7090852856636047
red_blood_cell_count: W-statistic = 0.9763184785842896, p-value = 0.052725110203027725
```

Yeo-Johnson dönüşümü normallik testi sonuçlarına göre, rastgele kan glukozu, kan üre, beyaz kan hücresi sayısı ve kırmızı kan hücresi sayısı değişkenleri p-değeri 0.05'den büyük olduğundan normal dağılıma uymaktadır. H_0 hipotezi reddedilir.

Box-Cox dönüşümü ile aynı sonuçları vermektedir.

Normalleştirme sonuçlarına istinaden, değişkenlerin grafik aşağıdaki gibidir:



Grafik.3.: Değişkenlere ait histogram grafikleri

Age (yaş): Grafikte dağılım, normal dağılım göstermemektedir.

Blood Pressure (kan basıncı): Normal dağılım göstermemektedir. Sola çarpıklık vardır.

Specific Gravity (özgül ağırlık): Normal dağılım göstermemektedir.

Albumin: Normal dağılım göstermemektedir. Sola çarpıklık vardır

Sugar (şeker): Normal dağılım göstermemektedir. Sağa çarpıklık vardır.

Blood Glucose Random (rastgele kan şekeri): Normal dağılım göstermemektedir.

Blood Urea (kan üre): Normal dağılım göstermektedir.

Serum Creatinine (serum kreatinin): Normal dağılım göstermemektedir. Hafif sağa çarpıklık vardır.

Sodium (sodyum): Yaklaşık olarak normal dağılım göstermektedir. Hafif sola çarpıklık vardır.

Potassium (potasyum): Yaklaşık olarak normal dağılım göstermektedir. Hafif sola çarpıklık vardır.

Haemoglobin (hemoglobin): Yaklaşık olarak normal dağılım göstermektedir. Hafif sola çarpıklık vardır.

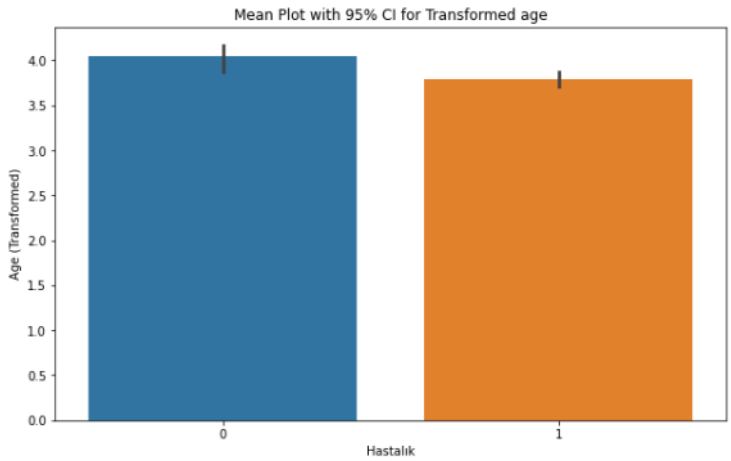
Packed Cell Volume (pıhtılaşmış hücre hacmi): Veriler, yaklaşık normal bir dağılım göstermektedir.

White Blood Cell Count (beyaz kan hücresi sayısı): Normal dağılım göstermemektedir.

Red Blood Cell Count (kırmızı kan hücresi sayısı): Normal dağılım göstermemektedir.

3.3.3.4. Dönüşüm sonrası İstatistiksel Özellikler

```
for var in boxcox_vars:
    df[f'{var}_transformed'] = yeojohnson_transformed[var]

# Dönüştürülmüş age değişkeni için barplot oluşturma
plt.figure(figsize=(10, 6))
sns.barplot(x='class', y='age', data=df, ci=95)
plt.xlabel('Hastalık')
plt.ylabel('Age (Transformed)')
plt.title('Mean Plot with 95% CI for Transformed age')
plt.show()
```



Grafik.4.: Yaş değişkenine göre hastalık durumu grafiği

Hasta Olmayan (0):

- Dönüştürülmüş yaş değişkeninin ortalaması yaklaşık olarak 4.0'dır.
- Güven aralığı dar olup, bu grup içerisindeki yaş değişkenliğinin düşük olduğunu gösterir.

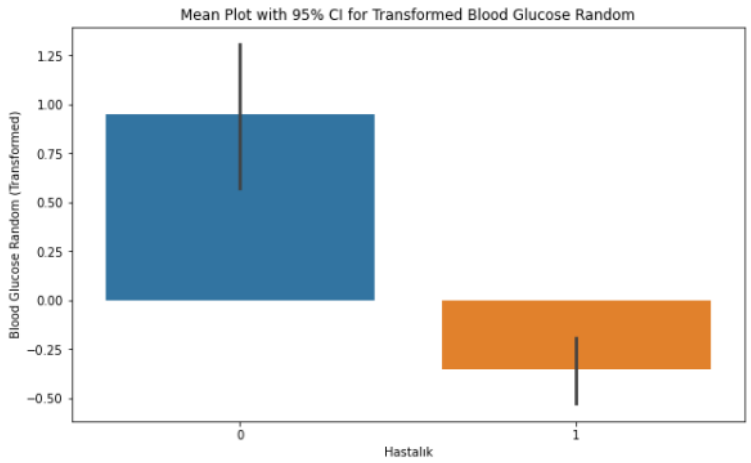
Hasta Olan (1):

- Dönüştürülmüş yaş değişkeninin ortalaması yaklaşık olarak 3.7'dir.

- Güven aralığı yine dar olup, bu grup içerisindeki yaş değişkenliğinin de düşük olduğunu gösterir.

Sonuç olarak, kronik böbrek hastalığı olan bireylerde dönüştürülmüş yaş ortalamasının, hastalık olmayan bireylere göre biraz daha düşük olduğunu ve bu farkın %95 güvenle istatistiksel olarak anlamlı olduğu söylenebilir. Ancak, iki grup arasındaki farkın büyük olmadığı ve güven aralıklarının örtüştüğü göz önünde bulundurulmalıdır. Bu, yaş değişkeninin kronik böbrek hastalığı üzerinde çok büyük bir etkisini olmayabilir.

```
# Dönüştürülmüş blood_glucose_random değişkeni için barplot oluşturma
plt.figure(figsize=(10, 6))
sns.barplot(x='class', y='blood_glucose_random_transformed', data=df, ci=95)
plt.xlabel('Hastalık')
plt.ylabel('Blood Glucose Random (Transformed)')
plt.title('Mean Plot with 95% CI for Transformed Blood Glucose Random')
plt.show()
```



Grafik.5.: Rasgele Kan Glukozu değişkenine göre hastalık durumu grafiği

Hasta Olmayan (0):

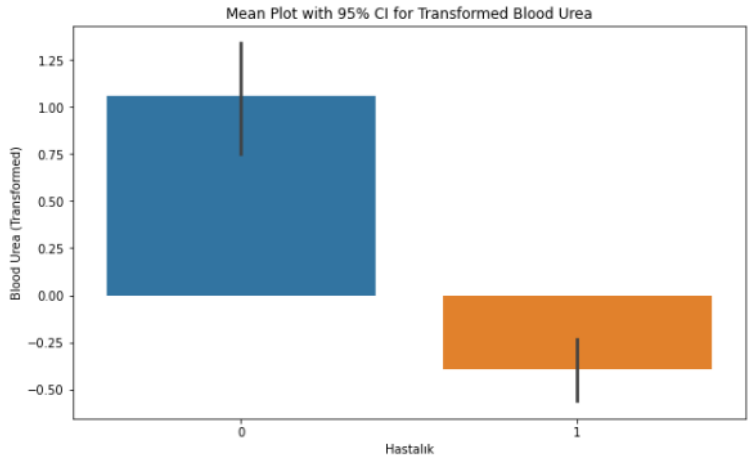
- Rastgele kan glukozu seviyesinin ortalaması daha yüksektir.
- Güven aralığı geniştir, bu da hasta olmayan bireylerde rastgele kan glukozu seviyesinin daha yüksek ve daha değişken olduğunu gösterir.

Hasta Olan (1):

- Rastgele kan glukozu seviyesinin ortalaması daha düşüktür.
- Güven aralığı daha dardır, bu da hasta olan bireylerde rastgele kan glukozu seviyesinin daha düşük ve daha az değişken olduğunu gösterir.

Sonuç olarak, kronik böbrek hastalığı olan bireylerde rastgele kan glukozu seviyesinin daha düşük olduğunu ve bu farkın %95 güvenle istatistiksel olarak anlamlı olduğu söylenebilir.

```
plt.figure(figsize=(10, 6))
sns.barplot(x='class', y='blood_urea_transformed', data=df, ci=95)
plt.xlabel('Hastalık')
plt.ylabel('Blood Urea (Transformed)')
plt.title('Mean Plot with 95% CI for Transformed Blood Urea')
plt.show()
```



Grafik.6.: Kan Üresi değişkenine göre hastalık durumu grafiği

Hasta Olmayan (0):

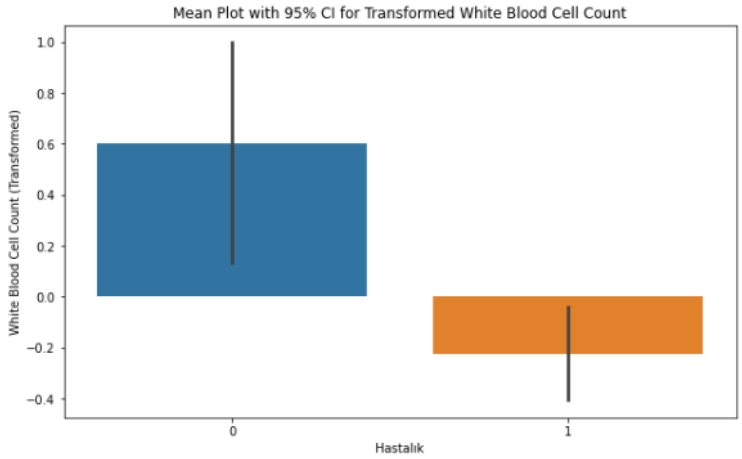
- Dönüştürülmüş kan üre seviyesinin ortalaması pozitif ve yaklaşık 1.0'dır.
- Güven aralığı geniştir, bu da hasta olmayan bireylerde kan üre seviyesinin daha yüksek ve daha değişken olduğunu gösterir.

Hasta Olan (1):

- Dönüştürülmüş kan üre seviyesinin ortalaması negatif ve yaklaşık -0.25'dir.
- Güven aralığı dardır, bu da hasta olan bireylerde kan üre seviyesinin daha düşük ve daha az değişken olduğunu gösterir.

Sonuç olarak, kronik böbrek hastalığı olan bireylerde dönüştürülmüş kan üre seviyesinin daha düşük olduğunu ve bu farkın %95 güvenle istatistiksel olarak anlamlı olduğu söylenebilir. Hasta olmayan grupta kan üre seviyesinin daha yüksek ve değişken olması, bu seviyelerin hastalıkla bağlantılı olmadığını gösteriyor olabilir. Ancak, hasta olan grupta kan üre seviyesinin daha düşük ve tutarlı olması, bu durumun hastalıkla ilişkili olduğunu göstermektedir.

```
plt.figure(figsize=(10, 6))
sns.barplot(x='class', y='white_blood_cell_count_transformed', data=df, ci=95)
plt.xlabel('Hastalık')
plt.ylabel('White Blood Cell Count (Transformed)')
plt.title('Mean Plot with 95% CI for Transformed White Blood Cell Count')
plt.show()
```



Grafik.7.: Beyaz Kan Hücre Sayısı değişkenine göre hastalık durumu grafiği

Hasta Olmayan Grup (0):

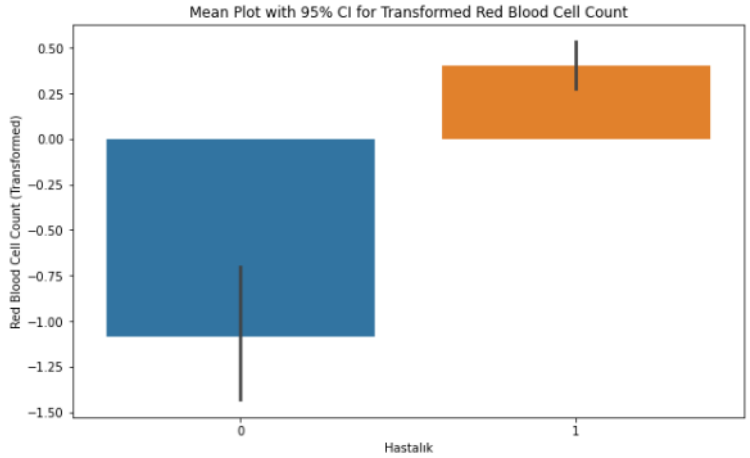
- Dönüştürülmüş beyaz kan hücresi sayısının ortalaması pozitif ve yaklaşık 0.6'dır.
- Güven aralığı geniştir, bu da hasta olmayan bireylerde beyaz kan hücresi sayısının daha yüksek ve daha değişken olduğunu gösterir.

Hasta Olan Grup (1):

- Dönüştürülmüş beyaz kan hücresi sayısının ortalaması negatif ve yaklaşık -0.2'dir.
- Güven aralığı daha dardır, bu da hasta olan bireylerde beyaz kan hücresi sayısının daha düşük ve daha az değişken olduğunu gösterir.

Sonuç olarak, kronik böbrek hastalığı olan bireylerde dönüştürülmüş beyaz kan hücresi sayısının daha düşük olduğunu ve bu farkın %95 güvenle istatistiksel olarak anlamlı olduğu söylenebilir. Hasta olmayan grupta beyaz kan hücresi sayısının daha yüksek ve değişken olması, bu bireylerde bu seviyelerin hastalıkla ilişkili olmadığını gösterebilir. Buna karşılık, hasta olan grupta beyaz kan hücresi sayısının daha düşük ve tutarlı olması, hastalıkla bağlantılı olabileceğini göstermektedir.

```
for var in boxcox_vars:
    df[f'{var}_transformed'] = yeojohnson_transformed[var]

plt.figure(figsize=(10, 6))
sns.barplot(x='class', y='red_blood_cell_count_transformed', data=df, ci=95)
plt.xlabel('Hastalık')
plt.ylabel('Red Blood Cell Count (Transformed)')
plt.title('Mean Plot with 95% CI for Transformed Red Blood Cell Count')
plt.show()
```



Grafik.8.: Kırmızı Kan Hücre Sayısı değişkenine göre hastalık durumu grafiği

Hasta Olmayan Grup (0):

- Dönüştürülmüş kırmızı kan hücresi sayısının ortalaması negatif ve yaklaşık -0.75'tir.
- Güven aralığı geniştir, bu da hasta olmayan bireylerde kırmızı kan hücresi sayısının daha düşük ve daha değişken olduğunu gösterir.

Hasta Olan Grup (1):

- Dönüştürülmüş kırmızı kan hücresi sayısının ortalaması pozitif ve yaklaşık 0.25'tir.
- Güven aralığı dardır, bu da hasta olan bireylerde kırmızı kan hücresi sayısının daha yüksek ve daha tutarlı olduğunu gösterir.

Sonuç olarak, hasta olmayan grupta kırmızı kan hücresi sayısının daha düşük ve değişken olduğunu, hasta olan grupta ise bu sayının daha yüksek ve tutarlı olduğunu göstermektedir. Buna göre, kırmızı kan hücresi sayısının kronik böbrek hastalığı ile ilişkili olabileceğini işaret etmektedir. Bu durum %95 güven aralığında istatistiksel olarak anlamlı olduğu söylenebilir.

3.3.4.Varyans Homojenliği Testi (Levene Testi)

H_0: Gruplar arasında homojendir.

H_1: Gruplar arasında homojen değildir.

```python
import pandas as pd
import numpy as np
from sklearn.preprocessing import PowerTransformer
import matplotlib.pyplot as plt
import seaborn as sns
from scipy.stats import levene

# Homojenlik testi
columns_to_test_transformed = ['age_transformed', 'blood_urea_transformed',
                               'blood_glucose_random_transformed','white_blood_cell_count_transformed',
                               'red_blood_cell_count_transformed']

homogeneity_test = levene(*[df[col] for col in columns_to_test_transformed])

print("Homojenlik Testi İstatistik Değeri:", homogeneity_test.statistic)
print("Homojenlik Testi p-değeri:", homogeneity_test.pvalue)
```

```
Homojenlik Testi İstatistik Değeri: 0.3142574006935452
Homojenlik Testi p-değeri: 0.8684755754171054
```

P-değeri 0.868 ve 0.05 anlamlılık seviyesinden büyük olduğu için, H_0 hipotezi reddedilemez. Bu durumda, yaş, kan üresi, rastgele kan glukozu, beyaz kan hücre sayısı ve kırmızı kan hücre sayısı grupların homojen olduğu sonucuna varılır.

3.3.5.ANOVA

$H_0: \mu_0 = \mu_1$ (Gruplar arasında istatistiksel olarak anlamlı bir fark yoktur.)

$H_1: \mu_0 \neq \mu_1$ (Gruplar arasında istatistiksel olarak anlamlı bir fark vardır.)

```python
#ANOVA testi
from scipy import stats

# Değişken
degisken = ['age_transformed', 'blood_urea_transformed',
                               'blood_glucose_random_transformed','white_blood_cell_count_transformed',
                               'red_blood_cell_count_transformed']

# Her değişken için ANOVA 'class grubuna göre'
for var in degisken:
    groups = df.groupby('class')[var].apply(list)

    # Farklı gruplar için ANOVA testi
    f_val, p_val = stats.f_oneway(*groups)

    print(f"\nDeğişken: {var}")
    print(f"ANOVA test F-değeri: {f_val}")
    print(f"ANOVA test p-değeri: {p_val}")

    # Hipotez
    if p_val < 0.05:
        print("Gruplar arasında istatistiksel olarak anlamlı bir fark vardır.")
    else:
        print("Gruplar arasında istatistiksel olarak anlamlı bir fark yoktur.")
```

```
Değişken: age_transformed
ANOVA test F-değeri: 17.232460224878768
ANOVA test p-değeri: 6.743780369442737e-05
Gruplar arasında istatistiksel olarak anlamlı bir fark vardır.

Değişken: blood_urea_transformed
ANOVA test F-değeri: 74.80149053411725
ANOVA test p-değeri: 6.47169983958861e-14
Gruplar arasında istatistiksel olarak anlamlı bir fark vardır.

Değişken: blood_glucose_random_transformed
ANOVA test F-değeri: 52.55216662094112
ANOVA test p-değeri: 7.389550062303392e-11
Gruplar arasında istatistiksel olarak anlamlı bir fark vardır.

Değişken: white_blood_cell_count_transformed
ANOVA test F-değeri: 16.22018874557863
ANOVA test p-değeri: 0.00010681402713388788
Gruplar arasında istatistiksel olarak anlamlı bir fark vardır.

Değişken: red_blood_cell_count_transformed
ANOVA test F-değeri: 82.12277283178982
ANOVA test p-değeri: 7.760441668722O6e-15
Gruplar arasında istatistiksel olarak anlamlı bir fark vardır.
```

ANOVA test sonuçlarına bakıldığında, p-değerleri 0.05 anlamlılık seviyesinin oldukça altında. Bu durum, her bir değişken için gruplar arasında istatistiksel olarak anlamlı farklar olduğunu söylenebilir. Buna göre H_0 hipotezi reddedilir.

3.3.6.MANOVA

H_0 : Kronik böbrek hastalığı durumu, yaş, rastgele kan glukozu, kan üre, kırmızı kan hücresi sayısı ve beyaz kan hücresi sayısı gibi bağımlı değişkenler üzerinde anlamlı bir etkiye sahip değildir. $(H_0: \mu_0 = \mu_1)$

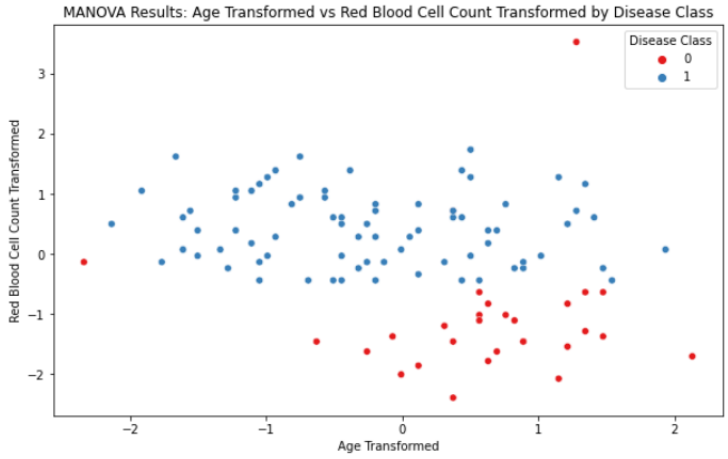
H_1: Kronik böbrek hastalığı durumu, yaş, rastgele kan glukozu, kan üre, kırmızı kan hücresi sayısı ve beyaz kan hücresi sayısı gibi bağımlı değişkenler üzerinde anlamlı bir etkiye sahiptir. $(H_1: \mu_0 \neq \mu_1)$

```python
import pandas as pd
import numpy as np
from statsmodels.multivariate.manova import MANOVA
import matplotlib.pyplot as plt
import seaborn as sns

# 'class' sütununu 'disease_class' olarak yeniden adlandırdık kütüphane ile karıştığından dolayı
df_yeni = df.rename(columns={'class': 'disease_class'})

# MANOVA analizi
manova = MANOVA.from_formula('age_transformed + blood_glucose_random_transformed + blood_urea_transformed + red_blood_cell_count_
print(manova.mv_test())
```

```
                 Multivariate linear model
============================================================

------------------------------------------------------------
       Intercept        Value  Num DF  Den DF  F Value Pr > F
------------------------------------------------------------
          Wilks' lambda 0.3416 5.0000 101.0000 38.9292 0.0000
         Pillai's trace 0.6584 5.0000 101.0000 38.9292 0.0000
 Hotelling-Lawley trace 1.9272 5.0000 101.0000 38.9292 0.0000
    Roy's greatest root 1.9272 5.0000 101.0000 38.9292 0.0000
------------------------------------------------------------

------------------------------------------------------------
     disease_class      Value  Num DF  Den DF  F Value Pr > F
------------------------------------------------------------
          Wilks' lambda 0.2744 5.0000 101.0000 53.4029 0.0000
         Pillai's trace 0.7256 5.0000 101.0000 53.4029 0.0000
 Hotelling-Lawley trace 2.6437 5.0000 101.0000 53.4029 0.0000
    Roy's greatest root 2.6437 5.0000 101.0000 53.4029 0.0000
============================================================
```

Wilks' Lambda, Pillai's trace, Hotelling-Lawley trace ve Roy's greatest root testlerine göre p-değerleri 0.00 olarak hesaplanmıştır. Bu, bağımsız değişken Kronik Böbrek Rahatsızlığı ile bağımlı değişkenler (yaş, rastgele kan glukozu, kan üre, kırmızı kan hücresi sayısı ve beyaz kan hücresi sayısı) arasında anlamlı bir ilişki olduğunu göstermektedir. Buna göre, H_0 hipotezi reddedilir.

```python
# görselleştirme
plt.figure(figsize=(10, 6))
sns.scatterplot(x='age_transformed', y='red_blood_cell_count_transformed', hue='disease_class', data=df_yeni, palette='Set1')
plt.xlabel('Age Transformed')
plt.ylabel('Red Blood Cell Count Transformed')
plt.title('MANOVA Results: Age Transformed vs Red Blood Cell Count Transformed by Disease Class')
plt.legend(title='Disease Class')
plt.show()
```



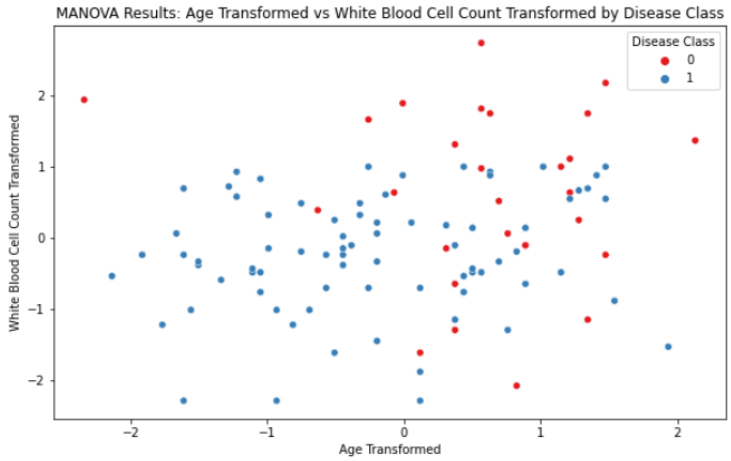
Grafik.9.: Hastalık durumuna göre Yaş ve Kırmızı Kan Hücre Sayısı değişkenine göre Manova sonuçları

Mavi noktalar hasta olmayan bireyleri temsil eder. Buna göre, yaş ve kırmızı kan hücresi sayısı arasında geniş bir dağılım vardır. Kırmızı kan hücresi sayısının çoğunlukla pozitif değerlerde olduğu ve yaşın daha geniş bir aralıkta dağıldığı görülmektedir.

Kırmızı noktalar ise hasta olan bireyleri temsil eder. Buna göre, kırmızı kan hücresi sayısının çoğunlukla negatif değerlerde olduğu ve yaşın daha dar bir aralıkta dağıldığı görülmektedir. Bu grafik, hastalık sınıfı ile yaş ve kırmızı kan hücresi sayısı arasında anlamlı bir ilişki olduğunu ve hasta olan bireylerin genellikle daha düşük yaş ve kırmızı kan hücresi sayısına sahip olduğunu göstermektedir.

Genel olarak, kronik böbrek hastalığı olan bireylerde kırmızı kan hücresi sayısının genellikle daha düşük olduğunu ve hastalık durumu ile kırmızı kan hücresi sayısı arasındaki ilişkinin dikkate değer olduğunu göstermektedir

```
# görselleştirme
plt.figure(figsize=(10, 6))
sns.scatterplot(x='age_transformed', y='white_blood_cell_count_transformed', hue='disease_class', data=df_yeni, palette='Set1')
plt.xlabel('Age Transformed')
plt.ylabel('White Blood Cell Count Transformed')
plt.title('MANOVA Results: Age Transformed vs White Blood Cell Count Transformed by Disease Class')
plt.legend(title='Disease Class')
plt.show()
```



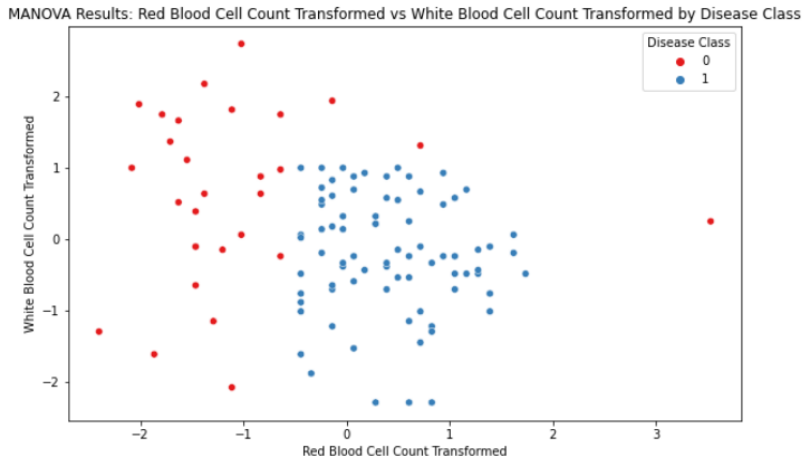
Grafik.10.: Hastalık durumuna göre Yaş ve Beyaz Kan Hücre Sayısı değişkenine göre Manova sonuçları

Mavi noktalar hasta olmayan bireyleri temsil eder. Buna göre, beyaz kan hücresi sayısının daha geniş bir aralıkta dağıldığı ve yaşın pozitif ve negatif değerler arasında değişkenlik gösterdiği görülmektedir. Beyaz kan hücresi sayısı genellikle daha yüksek değerlere sahiptir.

Kırmızı noktalar hasta olan bireyleri temsil eder. Buna göre, beyaz kan hücresi sayısının daha düşük ve daha dar bir aralıkta dağıldığı, yaşın ise daha geniş bir aralıkta değişkenlik gösterdiği görülmektedir. Beyaz kan hücresi sayısı genellikle daha düşük değerlere sahiptir.

Genel olarak, beyaz kan hücresi sayısının kronik böbrek hastalığı olan bireylerde genellikle daha düşük olduğunu ve hastalık durumu ile beyaz kan hücresi sayısı arasındaki ilişkinin dikkate değer olduğunu göstermektedir. Yaşın dağılımı ise her iki grup için de benzerlik göstermektedir.

```
# görselleştirme
plt.figure(figsize=(10, 6))
sns.scatterplot(x='red_blood_cell_count_transformed', y='white_blood_cell_count_transformed', hue='disease_class', data=df_yeni,
plt.xlabel('Red Blood Cell Count Transformed')
plt.ylabel('White Blood Cell Count Transformed')
plt.title('MANOVA Results: Red Blood Cell Count Transformed vs White Blood Cell Count Transformed by Disease Class')
plt.legend(title='Disease Class')
plt.show()
```



Grafik.11.: Hastalık durumuna göre Kırmızı Kan Hücre Sayısı ve Beyaz Kan Hücre Sayısı değişkenine göre Manova sonuçları

Kırmızı noktalar hasta olmayan bireyler, kırmızı kan hücresi sayısının genellikle negatif değerlere sahip olduğu, beyaz kan hücresi sayısının ise geniş bir aralıkta dağıldığı görülmektedir.

Mavi noktalar hasta olan bireyler, kırmızı kan hücresi sayısının genellikle pozitif değerlere sahip olduğu ve beyaz kan hücresi sayısının daha dar bir aralıkta dağıldığı görülmektedir.

Genel olarak, kırmızı kan hücresi sayısının ve beyaz kan hücresi sayısının kronik böbrek hastalığı durumu ile ilişkili olduğunu göstermektedir. Hasta olan bireylerde kırmızı kan hücresi sayısı genellikle daha yüksekken, beyaz kan hücresi sayısı daha dar bir aralıkta değişmektedir. Hasta olmayan bireylerde ise kırmızı kan hücresi sayısı daha düşük ve beyaz kan hücresi sayısı daha değişkenlik göstermektedir. Bu, her iki kan hücresi türünün de hastalık durumunu belirlemede önemli olabileceğini göstermektedir.

```
# görselleştirme
plt.figure(figsize=(10, 6))
sns.scatterplot(x='blood_urea_transformed', y='blood_glucose_random_transformed', hue='disease_class', data=df_yeni, palette='Set
plt.xlabel('Blood Urea Transformed')
plt.ylabel('Blood Glucose Random Transformed')
plt.title('MANOVA Results: Blood Urea Transformed vs Blood Glucose Random Transformed by Disease Class')
plt.legend(title='Disease Class')
plt.show()
```

Grafik.12.: Hastalık durumuna göre Kan Üresi ve Rasgele Kan Glukozu değişkenine göre Manova sonuçları

Mavi noktalar hasta olmayan bireyler, kan üre seviyesinin genellikle negatif değerlere sahip olduğu ve rastgele kan glukozu seviyesinin daha dar bir aralıkta dağıldığı gözlenmektedir. Kan üre seviyesi ile rastgele kan glukozu seviyesi arasında belirgin bir ilişki gözlenmemektedir.

Kırmızı noktalar hasta olan bireyler, kan üre seviyesinin genellikle pozitif değerlere sahip olduğu ve rastgele kan glukozu seviyesinin geniş bir aralıkta dağıldığı gözlenmektedir. Kan üre seviyesi ile rastgele kan glukozu seviyesi arasında pozitif bir ilişki olduğu görülmektedir.

Genel olarak, kan üre seviyesi ile rastgele kan glukozu seviyesinin kronik böbrek hastalığı durumu ile ilişkili olduğunu göstermektedir. Hasta olan bireylerde her iki değişken de genellikle daha yüksek değerlere sahipken, hasta olmayan bireylerde daha düşük değerler gözlenmektedir.

3.3.7.Diskriminant Analizi

```
for var in boxcox_vars:
    df[f'{var}_transformed'] = yeojohnson_transformed[var]

# bağımsız ve bağımlı değişken
X = df[[f'{var}_transformed' for var in boxcox_vars]]
y = df['class']

# Eğitim ve test
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)

# Modelinin oluşturulması
lda = LinearDiscriminantAnalysis()
lda.fit(X_train, y_train)

# Tahmin
y_pred = lda.predict(X_test)

# performansının değerlendirilmesi
accuracy = accuracy_score(y_test, y_pred)
conf_matrix = confusion_matrix(y_test, y_pred)
class_report = classification_report(y_test, y_pred)

print(f'Doğruluk Oranı: {accuracy}')
print('Karmaşıklık Matrisi:')
print(conf_matrix)
print('Sınıflandırma Raporu:')
print(class_report)
```

```
Doğruluk Oranı: 0.9696969696969697
Karmaşıklık Matrisi:
[[ 9  1]
 [ 0 23]]
Sınıflandırma Raporu:
              precision    recall  f1-score   support

           0       1.00      0.90      0.95        10
           1       0.96      1.00      0.98        23

    accuracy                           0.97        33
   macro avg       0.98      0.95      0.96        33
weighted avg       0.97      0.97      0.97        33
```

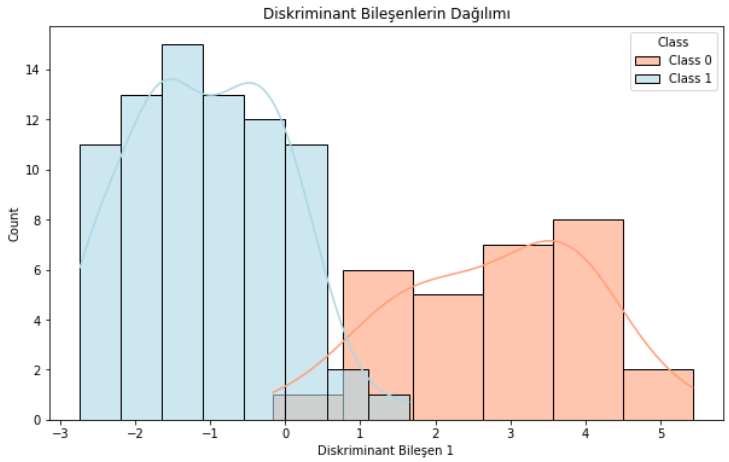
Veriler, %70 eğitim ve %30 test olacak şekilde bölünmüştür.

Modelin doğruluk oranı yaklaşık olarak %97'dir. Bu, modelin test verisindeki gözlemlerin %97'sini doğru bir şekilde sınıflandırdığını gösterir.

Karmaşıklık matrisinde satırlar gerçek sınıfları, sütunlar ise tahmin edilen sınıfları temsil eder. Buna göre, 9 doğru negatif, 23 doğru pozitif, 1 yanlış pozitif ve 0 yanlış negatif sınıflandırma vardır.

Modelin kronik böbrek hastalığını sınıflandırmada oldukça başarılı olduğunu göstermektedir. Hasta olan (1) için duyarlılık %100 olup, hasta olmayan (0) için duyarlılık %90'dır. Precision (kesinlik) ve F1 skoru da her iki sınıf için yüksek değerlerdedir. Bu, modelin hem doğru pozitifleri hem de doğru negatifleri tespit etmede etkili olduğunu göstermektedir.

```python
# Görselleştirme
X_lda = lda.transform(X)

# Histogram
plt.figure(figsize=(10, 6))
for class_value, color in zip(np.unique(y), colors):
    sns.histplot(X_lda[y == class_value, 0], kde=True, label=f'Class {class_value}', color=color, alpha=0.6)
plt.xlabel('Diskriminant Bileşen 1')
plt.title('Diskriminant Bileşenlerin Dağılımı')
plt.legend(title='Class')
plt.show()
```



Grafik.13.: Diskriminant Bileşenlerinin Dağılımı

Yukarıdaki grafik, Lojistik Regresyon modeli kullanılarak elde edilen diskriminant bileşenlerin dağılımını göstermektedir. Bu bileşenler, sınıflar arasındaki ayrımı en iyi şekilde temsil eden doğrusal kombinasyonlardır.

- X Ekseni: Birinci diskriminant bileşeninin değerlerini gösterir.
- Y Ekseni (Count): Her sınıf için diskriminant bileşen değerlerinin frekansını gösterir.

Histogramda hasta olmayanlar turuncu renkle gösterilmiştir. Bu sınıfta diskriminant bileşeni değerleri genellikle -3 ile 0 arasında yoğunlaşmıştır. Bu bireylerin diskriminant bileşeni değerleri daha düşük değerlere sahiptir.

Histogramda hasta olanlar ise açık mavi renkle gösterilmiştir. Bu sınıfta diskriminant bileşeni değerleri genellikle 0 ile 5 arasında yoğunlaşmıştır. Bu bireylerin diskriminant bileşeni değerleri daha yüksek değerlere sahiptir.

Genel olarak incelediğimizde iki sınıf arasında belirgin bir ayrım gözlemlenmektedir. Hasta olan bireyler (1) daha yüksek diskriminant bileşeni değerlerine sahipken, hasta olmayan bireyler (0) daha düşük değerlere sahiptir.

Grafikteki eğriler, veri dağılımının yoğunluklarını gösterir. Hasta olmayan bireylerin yoğunluğu -1 ve -2 civarında iken, hasta olan bireylerin yoğunluğu 2 ve 3 civarında yoğundur.

Lineer Diskriminant Analizi modelinin hasta olan ve olmayan bireyleri diskriminant bileşenlerini kullanarak başarılı bir şekilde ayırt edebildiğini göstermektedir. Grafik, modelin diskriminant bileşenlerinin her iki sınıfı nasıl ayırdığını görsel olarak temsil etmektedir. Bu tür bir ayrım, modelin sınıflandırma performansını ve doğruluğunu göstermektedir.

3.3.8.Lojistik Regresyon Analizi

```
# degiskenler
boxcox_vars = ['blood_glucose_random', 'blood_urea', 'white_blood_cell_count',
               'red_blood_cell_count', 'age']


# bağımsız ve bağımlı değişken
X = df[[f'{var}_transformed' for var in boxcox_vars]]
y = df['class']

# Eğitim ve test
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)

# Model oluşturulması ve eğitilmesi
logreg = LogisticRegression(max_iter=1000)
logreg.fit(X_train, y_train)

# Tahmin
y_pred = logreg.predict(X_test)

# Model performansı değerlendirilmesi
accuracy = accuracy_score(y_test, y_pred)
conf_matrix = confusion_matrix(y_test, y_pred)
class_report = classification_report(y_test, y_pred)

print(f'Doğruluk Oranı: {accuracy}')
print('Karmaşıklık Matrisi:')
print(conf_matrix)
print('Sınıflandırma Raporu:')
print(class_report)
```

```
Doğruluk Oranı: 0.9696969696969697
Karmaşıklık Matrisi:
[[ 9  1]
 [ 0 23]]
Sınıflandırma Raporu:
              precision    recall  f1-score   support

           0       1.00      0.90      0.95        10
           1       0.96      1.00      0.98        23

    accuracy                           0.97        33
   macro avg       0.98      0.95      0.96        33
weighted avg       0.97      0.97      0.97        33
```

Veriler, %70 eğitim ve %30 test olacak şekilde bölünmüştür.

Modelin doğruluk oranı yaklaşık olarak %97'dir. Bu, modelin test verisindeki gözlemlerin %97'sini doğru bir şekilde sınıflandırdığını gösterir.

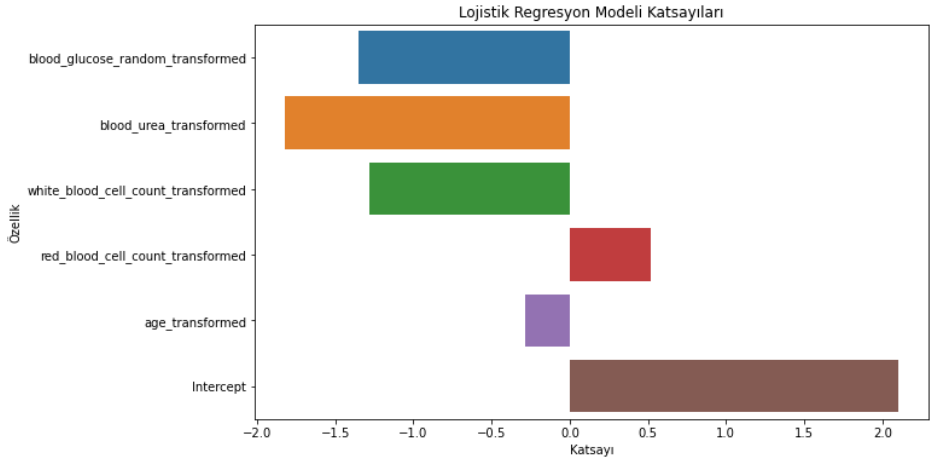
Karmaşıklık matrisinde satırlar gerçek sınıfları, sütunlar ise tahmin edilen sınıfları temsil eder. Buna göre, 9 doğru negatif, 23 doğru pozitif, 1 yanlış pozitif ve 0 yanlış negatif sınıflandırma vardır.

Modelin kronik böbrek hastalığını sınıflandırmada oldukça başarılı olduğunu göstermektedir. Hasta olan (1) için duyarlılık %100 olup, hasta olmayan (0) için duyarlılık

%90'dır. Precision (kesinlik) ve F1 skoru da her iki sınıf için yüksek değerlerdedir. Bu, modelin hem doğru pozitifleri hem de doğru negatifleri tespit etmede etkili olduğunu göstermektedir.

```
intercept = logreg.intercept_[0]
coefficients = logreg.coef_[0]
features = [f'{var}_transformed' for var in boxcox_vars]
features.append('Intercept')
coefficients = list(coefficients)
coefficients.append(intercept)
coeff_df = pd.DataFrame({'Feature': features, 'Coefficient': coefficients})
print(coeff_df)
```

```
                          Feature  Coefficient
0    blood_glucose_random_transformed    -1.350315
1             blood_urea_transformed    -1.818312
2  white_blood_cell_count_transformed    -1.279972
3    red_blood_cell_count_transformed     0.515061
4                 age_transformed    -0.281962
5                       Intercept     2.099361
```



Grafik.14.: Lojistik Regresyon Modeli Katsayıları

Model Katsayıları denklemi:

$$\mathrm{Ln}\left(\frac{\mathrm{P}(1)}{1}-\mathrm{P}(1)\right) = 2.099 + (-0.281)(\mathrm{age}) + (0.515)(\mathrm{red\_blood\_cell}) + (-1.279)(white\_blood\_cell) + (-1.818)(blood\_urea) + (-1.350)(blood\_glucose\_random)$$

Rasgele kan glukozu değişkeni, negatif bir katsayıya sahip olması, bu değişkenin artmasının kronik böbrek hastalığı olma olasılığını azalttığını gösterir.

Kan üresi değişkeni, negatif bir katsayıya sahip olması, bu değişkenin artmasının kronik böbrek hastalığı olma olasılığını azalttığını gösterir.

Beyaz kan hücre sayısı değişkeni, negatif bir katsayıya sahip olması, bu değişkenin artmasının kronik böbrek hastalığı olma olasılığını azalttığını gösterir.

Kırmızı kan hücresi sayısı değişkeni, pozitif bir katsayıya sahip olması, bu değişkenin artmasının kronik böbrek hastalığı olma olasılığını arttırdığını gösterir.

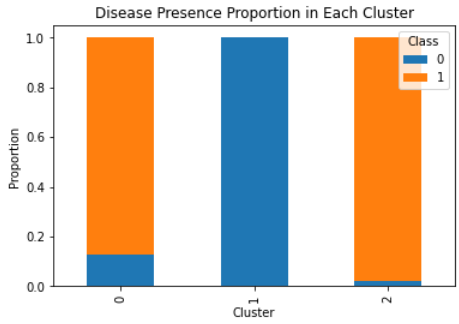
Yaş değişkeni, negatif bir katsayıya sahip olması, bu değişkenin artmasının kronik böbrek hastalığı olma olasılığını azalttığını gösterir.

Intercept: Modelin sabit terimi olup, tüm bağımsız değişkenlerin sıfır olduğu durumda log-odds'un başlangıç değerini temsil eder.

3.3.9.Kümeleme

```
# Her küme için hastalık var/yok oranlarının hesaplanması
cluster_analysis = df.groupby('cluster')['class'].value_counts(normalize=True).unstack().fillna(0)

print("Kümeler için hastalık var/yok oranları:")
print(cluster_analysis)

# Hastalık var/yok oranlarının görselleştirilmesi
cluster_analysis.plot(kind='bar', stacked=True)
plt.xlabel('Cluster')
plt.ylabel('Proportion')
plt.title('Disease Presence Proportion in Each Cluster')
plt.legend(title='Class')
plt.show()
```

```
Kümeler için hastalık var/yok oranları:
class           0         1
cluster
0        0.128205  0.871795
1        1.000000  0.000000
2        0.022222  0.977778
```



Grafik.15.: Hastalık durumuna kümeleme grafiği

Küme 0:

- Hasta olmayan bireylerin oranı (0): %12.82
- Hasta olan bireylerin oranı (1): %87.18
- Bu kümede çoğunlukla hasta olan bireyler bulunmaktadır.

Küme 1:

- Hasta olmayan bireylerin oranı (0): %100

- Hasta olan bireylerin oranı (1): %0
- Bu kümede yalnızca hasta olmayan bireyler bulunmaktadır.

Küme 2:

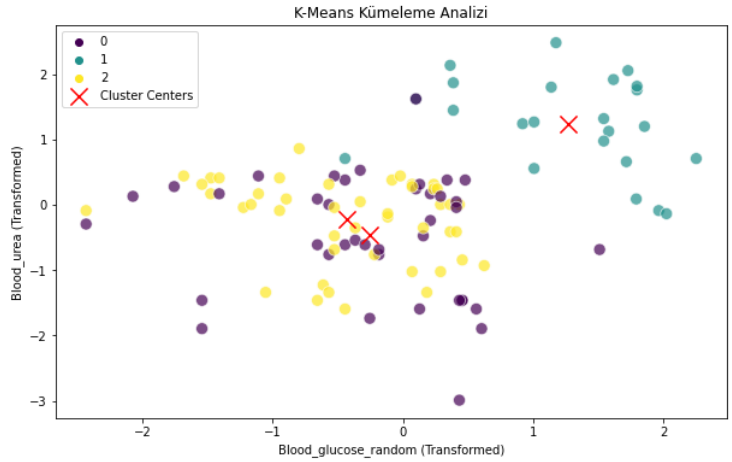
Hasta olmayan bireylerin oranı (0): %2.22

Hasta olan bireylerin oranı (1): %97.78

Bu kümede neredeyse tamamen hasta olan bireyler bulunmaktadır.

Genel olarak, her kümedeki bireylerin hastalık var/yok durumlarına göre dağılımını göstermektedir. Sonuçlarda, küme 0 ve küme 2'de çoğunlukla hasta olan bireylerin bulunduğunu, küme 1'de ise yalnızca hasta olmayan bireylerin bulunduğunu göstermektedir.

```
# değişkenler
boxcox_vars = ['blood_glucose_random', 'blood_urea', 'white_blood_cell_count',
               'red_blood_cell_count', 'age']

# bağımsız değişken
X = df[[f'{var}_transformed' for var in boxcox_vars]]

# K-Means algoritması
kmeans = KMeans(n_clusters=3, n_init=10, random_state=42)
kmeans.fit(X)

# Küme merkezleri ve tahmin edilen kümeler
centers = kmeans.cluster_centers_
labels = kmeans.labels_

df['cluster'] = labels

# görselleştirme
plt.figure(figsize=(10, 6))
sns.scatterplot(x=X.iloc[:, 0], y=X.iloc[:, 1], hue=labels, palette='viridis', s=100, alpha=0.7)
plt.scatter(centers[:, 0], centers[:, 1], c='red', marker='x', s=200, label='Cluster Centers')
plt.xlabel(boxcox_vars[0].capitalize() + ' (Transformed)')
plt.ylabel(boxcox_vars[1].capitalize() + ' (Transformed)')
plt.title('K-Means Kümeleme Analizi')
plt.legend()
plt.show()
```



Grafik.16.: Kan Üresi ve Rasgele Kan Glukozu değişkenlerinin K-Means Kümeleme Analizi Grafiği

Grafikte, K-means kümeleme analizi sonuçları gösterilmektedir. X ekseninde "Rasgele Kan Glukozu", Y ekseninde ise "Kan Üresi" değerleri yer almaktadır. Renkler, her bir veri noktasının ait olduğu kümeyi göstermektedir ve kırmızı çarpılar her kümenin merkezini temsil etmektedir.

Kümeler,

- **Küme 0 (Mor):** Bu kümedeki veri noktaları genellikle daha düşük rasgele kan glukozu ve kan üresi değerlerine sahiptir.
- **Küme 1 (Mavi):** Bu kümedeki veri noktaları genellikle daha yüksek rasgele kan glukozu ve kan üresi değerlerine sahiptir.
- **Küme 2 (Sarı):** Bu kümedeki veri noktaları orta seviyede rasgele kan glukozu ve kan üresi değerlerine sahiptir.

Küme Merkezleri,

Kırmızı çarpılar her bir kümenin merkezini göstermektedir. Bu merkezler, her kümede yer alan veri noktalarının ortalama değerlerine yakın noktaları temsil eder.

Dağılım:

Grafikte, veri noktalarının farklı kümelere nasıl dağıldığını ve hangi bölgelerde yoğunlaştığını görsel olarak gösterilmektedir.

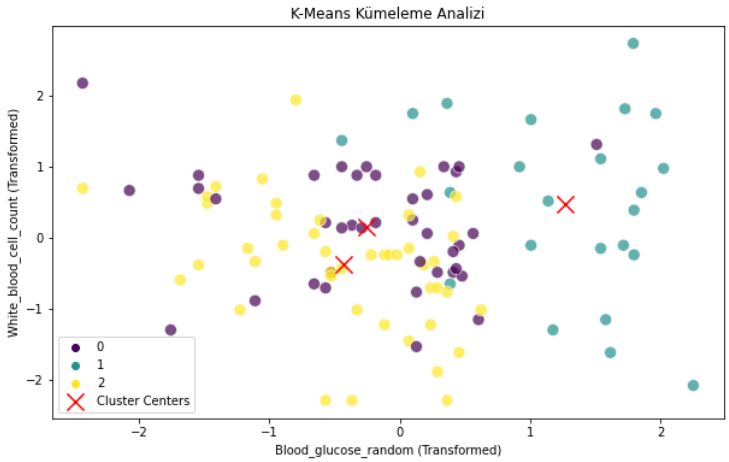
Küme 1 (mavi) daha yüksek rasgele kan glukozu ve kan üresi değerlerine sahipken, Küme 0 (mor) daha düşük değerlere sahiptir. Küme 2 (sarı) ise bu değerlerin orta seviyelerinde yer almaktadır.

Genel olarak,

- **Küme 0 (Mor):** Daha düşük rasgele kan glukozu ve kan üresi değerlerine sahip bireyler. Bu kümede çoğunlukla hasta olmayan bireylerin yer aldığı gözlemlenmiştir.
- **Küme 1 (Mavi):** Daha yüksek rasgele kan glukozu ve kan üresi değerlerine sahip bireyler. Bu kümede çoğunlukla hasta olan bireylerin yer aldığı gözlemlenmiştir.
- **Küme 2 (Sarı):** Orta seviyede rasgele kan glukozu ve kan üresi değerlerine sahip bireyler. Bu kümede de hasta olan bireylerin oranı yüksektir.

Bu grafik, K-means kümeleme algoritmasının verileri nasıl gruplandırdığını ve her bir kümenin merkezini nasıl belirlediğini görsel olarak temsil etmektedir. Hem de, her bir kümenin hastalık durumu ile ilişkili olduğu gözlemlenmiştir.

```python
# değişkenler
boxcox_vars = ['blood_glucose_random', 'blood_urea', 'white_blood_cell_count',
               'red_blood_cell_count', 'age']

# bağımsız değişken
X = df[[f'{var}_transformed' for var in boxcox_vars]]

# K-Means algoritması
kmeans = KMeans(n_clusters=3, n_init=10, random_state=42)
kmeans.fit(X)

# Küme merkezleri ve tahmin edilen kümeler
centers = kmeans.cluster_centers_
labels = kmeans.labels_

df['cluster'] = labels

# görselleştirme
plt.figure(figsize=(10, 6))
sns.scatterplot(x=X.iloc[:, 0], y=X.iloc[:, 2], hue=labels, palette='viridis', s=100, alpha=0.7)
plt.scatter(centers[:, 0], centers[:, 2], c='red', marker='x', s=200, label='Cluster Centers')
plt.xlabel(boxcox_vars[0].capitalize() + ' (Transformed)')
plt.ylabel(boxcox_vars[2].capitalize() + ' (Transformed)')
plt.title('K-Means Kümeleme Analizi')
plt.legend()
plt.show()
```



Grafik.17.: Beyaz Kan Hücre Sayısı ve Rasgele Kan Glukozu değişkenlerinin K-Means Kümeleme Analizi Grafiği

Bu grafikte, K-means kümeleme analizi sonuçları gösterilmektedir. X ekseninde "Rasgele Kan Glukozu", Y ekseninde ise "Beyaz Kan Hücre Sayısı" değerleri yer almaktadır. Renkler, her bir veri noktasının ait olduğu kümeyi göstermekte ve kırmızı çarpılar her kümenin merkezini temsil etmektedir.

Küme Açıklamaları,

- **Küme 0 (Mor):** Bu kümedeki veri noktaları genellikle daha düşük ve orta seviyede Rasgele Kan Glukozu ve Beyaz Kan Hücre Sayısı değerlerine sahiptir.
- **Küme 1 (Mavi):** Bu kümedeki veri noktaları genellikle daha yüksek Rasgele Kan Glukozu ve Beyaz Kan Hücre Sayısı değerlerine sahiptir.
- **Küme 2 (Sarı):** Bu kümedeki veri noktaları genellikle orta seviyede Rasgele Kan Glukozu ve düşük Beyaz Kan Hücre Sayısı değerlerine sahiptir.

Küme Merkezleri, kırmızı çarpılar her bir kümenin merkezini göstermektedir. Bu merkezler, her kümede yer alan veri noktalarının ortalama değerlerine yakın noktaları temsil eder.

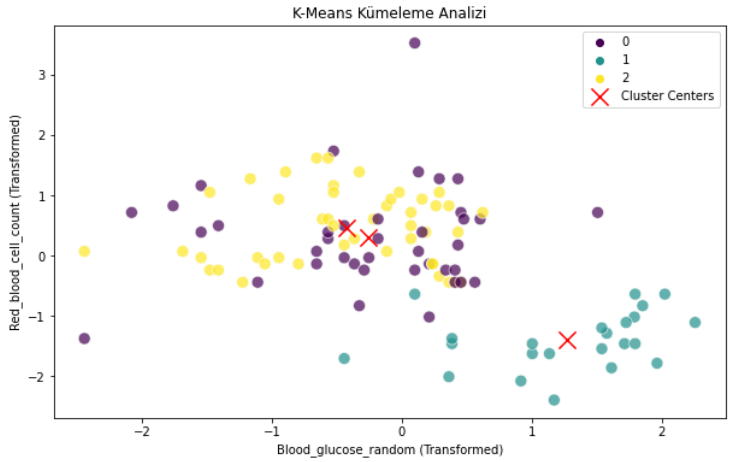
Dağılım, Küme 1 (turkuaz) daha yüksek rasgele kan glukozu ve beyaz kan hücre sayısı değerlerine sahipken, Küme 0 (mor) daha düşük ve orta seviyelerdeki değerlere sahiptir. Küme 2 (sarı) ise daha düşük beyaz kan hücre sayısı ve orta seviyelerde rasgele kan glukozu değerlerine sahiptir.

Genel olarak,

- Küme 0 (Mor): Daha düşük ve orta seviyede rasgele kan glukozu ve beyaz kan hücre sayısı değerlerine sahip bireyler. Bu kümede çeşitli hastalık durumlarına sahip bireyler olabilir.
- Küme 1 (Turkuaz): Daha yüksek rasgele kan glukozu ve beyaz kan hücre sayısı değerlerine sahip bireyler. Bu kümede genellikle hasta olan bireylerin yer aldığı gözlemlenebilir.
- Küme 2 (Sarı): Orta seviyede rasgele kan glukozu ve düşük beyaz kan hücre sayısı değerlerine sahip bireyler. Bu kümede hasta olan bireylerin oranı yüksek olabilir.

Grafik, K-means kümeleme algoritmasının verileri nasıl gruplandırdığını ve her bir kümenin merkezini nasıl belirlediğini görsel olarak temsil etmektedir. Ek olarak, her bir kümenin hastalık durumu ile ilişkili olduğu gözlemlenmiştir.

```
# değişkenler
boxcox_vars = ['blood_glucose_random', 'blood_urea', 'white_blood_cell_count',
               'red_blood_cell_count', 'age']

# bağımsız değişken
X = df[[f'{var}_transformed' for var in boxcox_vars]]

# K-Means algoritması
kmeans = KMeans(n_clusters=3, n_init=10, random_state=42)
kmeans.fit(X)

# Küme merkezleri ve tahmin edilen kümeler
centers = kmeans.cluster_centers_
labels = kmeans.labels_

df['cluster'] = labels

# görselleştirme
plt.figure(figsize=(10, 6))
sns.scatterplot(x=X.iloc[:, 0], y=X.iloc[:, 3], hue=labels, palette='viridis', s=100, alpha=0.7)
plt.scatter(centers[:, 0], centers[:, 3], c='red', marker='x', s=200, label='Cluster Centers')
plt.xlabel(boxcox_vars[0].capitalize() + ' (Transformed)')
plt.ylabel(boxcox_vars[3].capitalize() + ' (Transformed)')
plt.title('K-Means Kümeleme Analizi')
plt.legend()
plt.show()
```



Grafik.18.: Kırmızı Kan Hücre Sayısı ve Rasgele Kan Glukozu değişkenlerinin K-Means Kümeleme Analizi Grafiği

Bu grafikte, K-means kümeleme analizi sonuçları gösterilmektedir. X ekseninde "Rasgele Kan Glukozu", Y ekseninde ise " Kırmızı Kan Hücre Sayısı" değerleri yer almaktadır. Renkler, her bir veri noktasının ait olduğu kümeyi göstermekte ve kırmızı çarpılar her kümenin merkezini temsil etmektedir.

Kümeler,

- **Küme 0 (Mor):** Bu kümedeki veri noktaları genellikle daha düşük ve orta seviyede rasgele kan glukozu ve kırmızı kan hücre sayısı değerlerine sahiptir.
- **Küme 1 (Mavi):** Bu kümedeki veri noktaları genellikle daha yüksek rasgele kan glukozu ve kırmızı kan hücre sayısı değerlerine sahiptir.
- **Küme 2 (Sarı):** Bu kümedeki veri noktaları genellikle orta seviyede rasgele kan glukozu ve düşük kırmızı kan hücre sayısı değerlerine sahiptir.

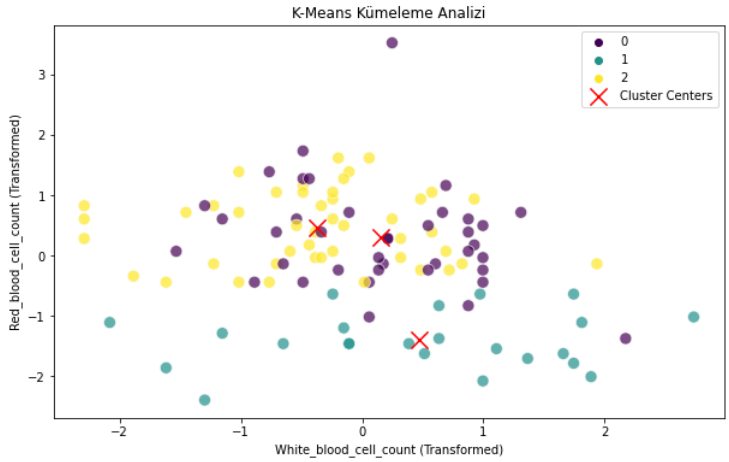
Küme Merkezleri, Kırmızı çarpılar her bir kümenin merkezini göstermektedir. Bu merkezler, her kümede yer alan veri noktalarının ortalama değerlerine yakın noktaları temsil eder.

Dağılım, Küme 1 (Mavi) daha yüksek rasgele kan glukozu ve kırmızı kan hücre sayısı değerlerine sahipken, Küme 0 (mor) daha düşük ve orta seviyelerdeki değerlere sahiptir. Küme 2 (sarı) ise daha düşük kırmızı kan hücre sayısı ve orta seviyelerde rasgele kan glukozu değerlerine sahiptir.

Sonuçlar,

- Küme 0 (Mor): Daha düşük ve orta seviyede rasgele kan glukozu ve kırmızı kan hücre sayısı değerlerine sahip bireyler. Bu kümede çeşitli hastalık durumlarına sahip bireyler olabilir.
- Küme 1 (Yeşil): Daha yüksek rasgele kan glukozu ve kırmızı kan hücre sayısı değerlerine sahip bireyler. Bu kümede genellikle hasta olan bireylerin yer aldığı gözlemlenebilir.
- Küme 2 (Sarı): Orta seviyede rasgele kan glukozu ve düşük kırmızı kan hücre sayısı değerlerine sahip bireyler. Bu kümede hasta olan bireylerin oranı yüksek olabilir.

```
# değişkenler
boxcox_vars = ['blood_glucose_random', 'blood_urea', 'white_blood_cell_count',
               'red_blood_cell_count', 'age']

# bağımsız değişken
X = df[[f'{var}_transformed' for var in boxcox_vars]]

# K-Means algoritması
kmeans = KMeans(n_clusters=3, n_init=10, random_state=42)
kmeans.fit(X)

# Küme merkezleri ve tahmin edilen kümeler
centers = kmeans.cluster_centers_
labels = kmeans.labels_

df['cluster'] = labels

# görselleştirme
plt.figure(figsize=(10, 6))
sns.scatterplot(x=X.iloc[:, 2], y=X.iloc[:, 3], hue=labels, palette='viridis', s=100, alpha=0.7)
plt.scatter(centers[:, 2], centers[:, 3], c='red', marker='x', s=200, label='Cluster Centers')
plt.xlabel(boxcox_vars[2].capitalize() + ' (Transformed)')
plt.ylabel(boxcox_vars[3].capitalize() + ' (Transformed)')
plt.title('K-Means Kümeleme Analizi')
plt.legend()
plt.show()
```



Grafik.19.: Beyaz Kan Hücre Sayısı ve Kırmızı Kan Hücre Sayısı değişkenlerinin K-Means Kümeleme Analizi Grafiği

Grafik, K-means kümeleme analizi sonuçları gösterilmektedir. X ekseninde "Beyaz Kan Hücre Sayısı", Y ekseninde ise " Kırmızı Kan Hücre Sayısı" değerleri yer almaktadır. Renkler, her bir

veri noktasının ait olduğu kümeyi göstermekte ve kırmızı çarpılar her kümenin merkezini temsil etmektedir.

Kümeler,

- **Küme 0 (Mor):** Bu kümedeki veri noktaları genellikle orta seviyede beyaz kan hücre sayısı ve kırmızı kan hücre sayısı değerlerine sahiptir.
- **Küme 1 (Mavi):** Bu kümedeki veri noktaları genellikle düşük beyaz kan hücre sayısı ve kırmızı kan hücre sayısı değerlerine sahiptir.
- **Küme 2 (Sarı):** Bu kümedeki veri noktaları genellikle orta seviyede beyaz kan hücre sayısı ve daha yüksek kırmızı kan hücre sayısı değerlerine sahiptir.

Küme Merkezleri, kırmızı çarpılar her bir kümenin merkezini göstermektedir. Bu merkezler, her kümede yer alan veri noktalarının ortalama değerlerine yakın noktaları temsil eder.

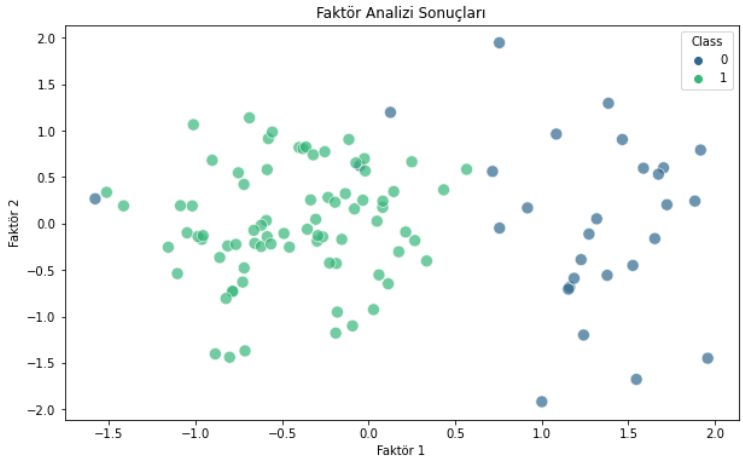
Dağılım, Küme 1 (mavi) daha düşük beyaz kan hücre sayısı ve kırmızı kan hücre sayısı değerlerine sahipken, Küme 0 (mor) orta seviyelerde değerlere sahiptir. Küme 2 (sarı) ise daha yüksek kırmızı kan hücre sayısı ve orta seviyelerde beyaz kan hücre sayısı değerlerine sahiptir.

Sonuçlar,

- **Küme 0 (Mor):** Orta seviyede beyaz kan hücre sayısı ve kırmızı kan hücre sayısı değerlerine sahip bireyler. Bu kümede çeşitli hastalık durumlarına sahip bireyler olabilir.
- **Küme 1 (Mavi):** Daha düşük beyaz kan hücre sayısı ve kırmızı kan hücre sayısı değerlerine sahip bireyler. Bu kümede genellikle hasta olan bireylerin yer aldığı gözlemlenebilir.
- **Küme 2 (Sarı):** Orta seviyede beyaz kan hücre sayısı ve daha yüksek kırmızı kan hücre sayısı değerlerine sahip bireyler. Bu kümede hasta olan bireylerin oranı yüksek olabilir.

3.3.10.Temel Bileşenler Analizi ve Faktör Analizi

```
#TBA Factor Analysis
import pandas as pd
import numpy as np
from sklearn.preprocessing import PowerTransformer
from sklearn.decomposition import FactorAnalysis
import matplotlib.pyplot as plt
import seaborn as sns

degiskenler = ['blood_glucose_random', 'blood_urea', 'white_blood_cell_count',
               'red_blood_cell_count', 'age']

# Faktör analizi için bağımsız değişkenlerin seçimi
X = df[[f'{var}_transformed' for var in degiskenler]]

# Faktör analizi
fa = FactorAnalysis(n_components=2, random_state=42)
X_fa = fa.fit_transform(X)

# Faktör analizi sonuçlarının görselleştirilmesi
plt.figure(figsize=(10, 6))
sns.scatterplot(x=X_fa[:, 0], y=X_fa[:, 1], hue=df['class'], palette='viridis', s=100, alpha=0.7)
plt.xlabel('Faktör 1')
plt.ylabel('Faktör 2')
plt.title('Faktör Analizi Sonuçları')
plt.legend(title='Class')
plt.show()
```



Grafik.20.: Faktör Analizi Scatter Plot

Bu scatter plot, faktör analizi sonucunda elde edilen iki faktörü ve sınıf etiketlerini göstermektedir. Grafikte iki sınıf bulunmaktadır: 0 (mavi noktalar) hasta olmayan ve 1 (yeşil noktalar) hasta olan.

Faktör 1,

- Yatay eksende Faktör 1 yer almakta. Faktör 1'in düşük değerleri (sol taraf) ve yüksek değerleri (sağ taraf) arasında sınıflar arasında bir ayrım gözlemlenmektedir.

Faktör 2,

- Dikey eksende Faktör 2 yer almakta. Faktör 2'nin yüksek (üst taraf) ve düşük (alt taraf) değerleri arasında sınıflar arasında belirgin bir ayrım gözlemlenmektedir.

Sınıf Dağılımı:

- **0 (mavi noktalar) Hasta Olmayan:** Genel olarak Faktör 1'de pozitif değerlere ve Faktör 2'de pozitif ve negatif değerlere yayılmış durumda.
- **1 (yeşil noktalar) Hasta Olan:** Genel olarak Faktör 1'de negatif değerlere ve Faktör 2'de pozitif ve negatif değerlere yayılmış durumda.

Faktör analizi sonucunda, Hasta Olmayan ve Hasta Olan'ın Faktör 1 ve Faktör 2'ye göre ayrıldığı görülmektedir. Özellikle Faktör 1, sınıfları ayırmada güçlü bir etkiye sahip gibi görünmektedir. Hasta Olmayan, Faktör 1'de pozitif değerlere sahipken, Hasta Olan genellikle negatif değerlere sahiptir. Faktör 2 ise her iki sınıf için de yayılım göstermektedir, ancak b faktör, sınıflar arasında belirli bir ayrım oluşturmaz.

```
# Faktör yüklemelerini görüntüleme
loadings = pd.DataFrame(fa.components_.T, columns=['Faktör 1', 'Faktör 2'], index=X.columns)
print(loadings)
```

```
                                 Faktör 1  Faktör 2
blood_glucose_random_transformed    0.530882 -0.272742
blood_urea_transformed              0.566294 -0.119473
white_blood_cell_count_transformed  0.320419  0.622169
red_blood_cell_count_transformed   -0.827308  0.036972
age_transformed                     0.387312  0.083078
```

Faktör 1:

- Yaş, Rasgele Kan Glukozu ve Kan Üresi değişkenleri, Faktör 1 ile yüksek pozitif yüklemelere sahiptir. Rasgele Kan Glukozu ve Kan Üresi değişkenleri Faktör 1 ile güçlü bir ilişkiye sahip olduğunu gösterirken, ancak yaş değişkeni diğer iki değişkene göre daha düşük bir yüklemeye sahiptir.
- Kırmızı Kan Hücre Sayısı değişkeni Faktör 1 ile negatif bir ilişkiye sahiptir.
- Beyaz Kan Hücre Sayısı değişkeni ise Faktör 1 ile düşük pozitif bir yüklemeye sahiptir, ancak diğer değişkenlere göre daha zayıf bir ilişkiye sahiptir.

Faktör 2:

- Beyaz Kan Hücre Sayısı değişkeni, Faktör 2 ile yüksek pozitif yüklemeye sahiptir. Bu, bu değişkenin Faktör 2 ile güçlü bir ilişkiye sahip olduğunu gösterir.
- Rasgele Kan Glukozu ve Kan Üresi değişkenleri, Faktör 2 ile zayıf negatif ilişkilere sahiptir.
- Kırmızı Kan Hücre Sayısı ve Yaş değişkenleri ise Faktör 2 ile pozitif ama düşük yüklemelere sahiptir.

Sonuç

- **Faktör 1:** Rasgele Kan Glukozu ve Kan Üresi değişkenleriyle güçlü pozitif ilişkiler gösterir. Ayrıca Yaş değişkeni de pozitif bir yüklemeye sahiptir. Kırmızı Kan Hücre Sayısı değişkeni ise negatif bir yüklemeye sahiptir.
- **Faktör 2:** Beyaz Kan Hücre Sayı değişkeniyle güçlü pozitif bir ilişki gösterir. Diğer değişkenlerle ise daha zayıf ilişkiler gösterir.

3.3.11.Random Forest

```python
#Random Forest Anaysis
from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score, confusion_matrix, classification_report

# bağımsız ve bağımlı değişken
X = df[[f'{var}_transformed' for var in degiskenler]]
y = df['class']

# Eğitim ve test setlerine ayırma
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)

# model oluşturma ve eğitme
rf = RandomForestClassifier(random_state=42)
rf.fit(X_train, y_train)

# Tahmin yapma
y_pred = rf.predict(X_test)

# Model performansının değerlendirilmesi
accuracy = accuracy_score(y_test, y_pred)
conf_matrix = confusion_matrix(y_test, y_pred)
class_report = classification_report(y_test, y_pred)

print(f'Doğruluk Oranı: {accuracy}')
print('Karmaşıklık Matrisi:')
print(conf_matrix)
print('Sınıflandırma Raporu:')
print(class_report)
```

```
Doğruluk Oranı: 1.0
Karmaşıklık Matrisi:
[[10  0]
 [ 0 23]]
Sınıflandırma Raporu:
              precision    recall  f1-score   support

           0       1.00      1.00      1.00        10
           1       1.00      1.00      1.00        23

    accuracy                           1.00        33
   macro avg       1.00      1.00      1.00        33
weighted avg       1.00      1.00      1.00        33
```

Doğruluk oranı:

- Model, test verisi üzerinde %100 doğruluk oranına sahiptir. Bu, modelin tüm test örneklerini doğru sınıflandırdığını gösterir.

Karmaşıklık Matrisi:

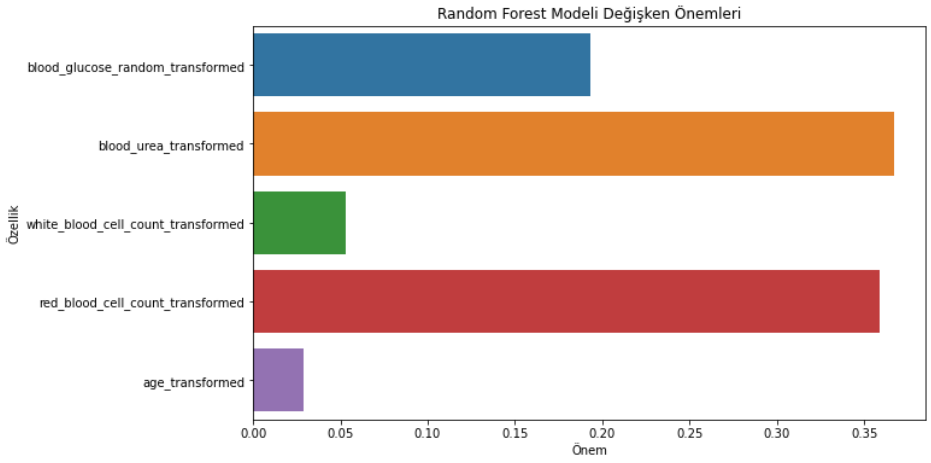
- Matriste, doğru pozitif 10 ve doğru negatif 23 sayıları gösterilmektedir. Yanlış pozitif veya yanlış negatif sınıflandırma yoktur.

- 10 örnek doğru bir şekilde hasta olmayan olarak sınıflandırılmıştır.
- 23 örnek doğru bir şekilde hasta olan olarak sınıflandırılmıştır.

Sınıflandırma Raporu:

- **Precision (Kesinlik):** Her iki sınıf için de 1.00, yani modelin pozitif tahminlerinin %100'ü doğru.
- **Recall (Duyarlılık):** Her iki sınıf için de 1.00, yani model gerçek pozitif örneklerin %100'ünü doğru tespit etmiş.
- **F1-Score:** Her iki sınıf için de 1.00, bu da modelin genel performansının mükemmel olduğunu gösterir.
- **Support:** Her sınıfta kaç örnek olduğu gösterilir (Class 0: 10 örnek, Class 1: 23 örnek).

```
# Model görselleştirme
importances = rf.feature_importances_
features = [f'{var}_transformed' for var in degiskenler]
importances_df = pd.DataFrame({'Feature': features, 'Importance': importances})

plt.figure(figsize=(10, 6))
sns.barplot(x='Importance', y='Feature', data=importances_df)
plt.title('Random Forest Modeli Değişken Önemleri')
plt.xlabel('Önem')
plt.ylabel('Özellik')
plt.show()
```



Grafik.21.: Random Forest Değişkenlerin Önemleri

Grafik, değişkenin modelin karar vermesine olan katkısını görsel olarak ifade eder. Değişken önemleri, modelin hangi özelliklerin sınıflandırma yaparken daha etkili olduğunu anlamamıza yardımcı olur.

- Rasgele Kan Glukozu, Yaklaşık olarak önem derecesi 0.20'dir. Bu değişken, modelde önemli bir yer tutmaktadır ve karar vermede etkili bir rol oynamaktadır.

- Kan Üresi, Yaklaşık olarak önem derecesi 0.30’dur. Bu değişken, modelde en yüksek öneme sahiptir ve modelin karar vermesinde büyük bir etkisi vardır.
- Beyaz Kan Hücre Sayısı, Yaklaşık olarak önem derecesi 0.10’dur. Bu değişken, modelde orta seviyede bir öneme sahiptir ve karar vermede katkıda bulunmaktadır.
- Kırmızı Kan Hücre Sayısı, Yaklaşık olarak önem derecesi 0.35’tir. Bu değişken, modelde çok yüksek bir öneme sahiptir ve modelin karar vermesinde büyük bir rol oynar.
- Yaş, Yaklaşık olarak önem derecesi 0.05’tir. Bu değişken, modelde en düşük öneme sahip olup karar vermede az katkıda bulunmaktadır.

4.SONUÇ

Bu çalışma, çok değişkenli istatistiksel yöntemler kullanılarak kronik böbrek hastalığının risk faktörlerini belirlemeyi ve hastalığı daha iyi anlamayı amaçlamaktadır. Kaggle'dan alınan 280 gözlem ve 26 değişkenden oluşan veri seti üzerinde çeşitli istatistiksel analizler gerçekleştirilmiştir.

Veri Ön İşleme

Analiz sürecinde, veri setinde bulunan benzersiz hasta kimlik numaraları (id) analizden çıkarılmış ve kayıp gözlemler temizlenmiştir. Bu adımlar sonucunda veri seti 25 değişken ve 107 gözlemden oluşmuştur.

Temel İstatistikler ve Normallik Testleri

Veri setinin temel istatistikleri incelenmiş ve değişkenlerin normalliği Shapiro-Wilk testi ile değerlendirilmiştir. Normallik testlerinde p-değerleri 0.05'in altında olan değişkenler Logaritmik, Box-Cox ve Yeo-Johnson dönüşümleri uygulanarak normalleştirilmiştir.

Levene Testi ve ANOVA Testleri

Levene testi ile varyans homojenliği incelenmiş ve ANOVA testi ile gruplar arasındaki farklar değerlendirilmiştir. Varyans homojenliği sağlanırken, ANOVA test sonuçları gruplar arasında istatistiksel olarak anlamlı farklar olduğunu göstermiştir.

MANOVA

Çoklu değişkenli varyans analizi sonuçlarında, bağımsız değişken olarak kronik böbrek hastalığının bağımlı değişkenler (yaş, rastgele kan glukozu, kan üre, kırmızı kan hücresi sayısı ve beyaz kan hücresi sayısı) üzerinde anlamlı bir etkiye sahip olduğunu göstermiştir.

Diskriminant ve Lojistik Regresyon Analizleri

Diskriminant analizi ve lojistik regresyon modeli kullanılarak hastalık durumu sınıflandırılmıştır. Her iki model de yüksek doğruluk oranları (%97) ve duyarlılık göstermiştir. Model performansları, hasta olan ve olmayan bireyleri ayırt etmede oldukça başarılı bulunmuştur.

Kümeleme Analizleri

K-means kümeleme algoritması ile veri seti analiz edilmiş ve kümeleme sonuçları hastalık durumu ile ilişkili bulunmuştur. Kümeleme grafikleri, hastalık durumuna göre grupların farklılıklarını görsel olarak göstermiştir.

Faktör Analizi

Faktör analizi ile değişkenlerin faktör yüklemeleri incelenmiş ve hastalık durumu ile ilişkili önemli değişkenler belirlenmiştir. Faktör analizi sonuçları, özellikle yaş, rasgele kan glukozu ve kan üresi değişkenlerinin hastalıkla güçlü ilişkisi olduğunu ortaya koymuştur.

Random Forest

Random Forest modeli kullanılarak değişkenlerin hastalık durumunu tahmin etme performansı %100 doğruluk oranıyla değerlendirilmiştir. Modelde en önemli değişkenler kırmızı kan hücresi sayısı, kan üresi ve rasgele kan glukozu olarak belirlenmiştir.

Genel Sonuçlar

- **Kırmızı Kan Hücresi Sayısı:** Kronik böbrek hastalığı olan bireylerde genellikle daha düşük değerlere sahiptir ve hastalık durumunun belirlenmesinde önemli bir rol oynar.
- **Beyaz Kan Hücresi Sayısı:** Hasta olan bireylerde daha düşük ve tutarlıdır. Bu değişken de hastalık durumuyla ilişkili bulunmuştur.
- **Kan Üresi:** Hasta olmayan bireylerde daha yüksek ve değişkendir. Hasta olan bireylerde ise daha düşük ve tutarlı seviyelerdedir.
- **Rasgele Kan Glukozu:** Hasta olmayan bireylerde daha yüksek ve değişken seviyelerde iken, hasta olan bireylerde daha düşük ve tutarlıdır.
- **Yaş:** Kronik böbrek hastalığı olan bireylerde dönüştürülmüş yaş ortalaması, hastalık olmayan bireylere göre biraz daha düşüktür.

KAYNAKÇA

CoLearningLounge. (n.d.). *Chronic Kidney Disease Dataset.* Kaggle. Erişim adresi: https://www.kaggle.com/datasets/colearninglounge/chronic-kidney-disease (Erişim Tarihi: 02.07.2024).

Erdem, K. (2024). *Çok Değişkenli İstatistiksel Analiz.* Medium. Erişim adresi: https://kardelennerdem.medium.com/%C3%A7ok-de%C4%9Fi%CC%87%C5%9Fkenli%CC%87-i%CC%87stati%CC%87sti%CC%87ksel-anali%CC%87z-c8e1018f25e4 (Erişim Tarihi:02.07.2024)

Arslan, K. (2024). *Tek Yönlü Çok Değişkenli Varyans Analizi (MANOVA).* Galloglu Blog. Erişim adresi: https://www.galloglu.com/blog/Tek-yonlu-cok-degiskenli-varyans-analizi-(MANOVA) (Erişim Tarihi: 02.07.2024)

Şen, S. (2016). *Tek Yönlü Çok Değişkenli Varyans Analizi (MANOVA).* Sunum. Erişim adresi: https://sedatsen.com/wp-content/uploads/2016/11/11-sunum.pdf (Erişim Tarihi: 02.07.2024)

Dayanıklı, A. S. (2021). *Python Veri Dönüştürme (Data Transformation).* Ravenfo. Erişim adresi: https://ravenfo.com/2021/07/08/python-veri-donusturme-data-transformation/#:~=Yeo%2DJohnson%20d%C3%B6n%C3%BC%C5%9F%C3%BCm%C3%BC,de%C4%9Fere%20sahip%20g%C3%B6zlemler%20oldu%C4%9Funda%20kullan%C4%B1%C5%9Fl%C4%B1d%C4%B1r. (Erişim Tarihi: 04.07.2024)

Atakan, C., & Karabulut, İ. (2024). *Derinliğe Dayalı Diskriminasyon*. Erişim adresi: https://dergipark.org.tr/tr/download/article-file/215054 (Erişim Tarihi: 04.07.2024)

Şavkay, D. (2024). *SPSS ile Diskriminant Analizi.* SPSS Yardımı. Erişim adresi: https://www.spss-yardimi.com/spss-ile-diskriminant-analizi/#:~=Diskriminant%20analizinin%20varsay%C4%B1mlar%C4%B1%20%C5%9Funlard%C4%B1r: (Erişim Tarihi: 04.07.2024)

Uzun, E. (2024). *Lineer Diskriminant Analizi (LDA).* Erdinç Uzun Blog. Erişim adresi: https://erdincuzun.com/makine_ogrenmesi/linear-discriminant-analysis-lda-lineer-diskriminant-analizi/ (Erişim Tarihi: 04.07.2024)

Terzi, Y. tarih yok). Lojistik regresyon analizi. *Ondokuz Mayıs Üniversitesi*. Erişim adresi: https://avys.omu.edu.tr/storage/app/public/yukselt/118305/Lojistik%20regresyon%20analizi.pdf (Erişim Tarihi: 04.07.2024)

Terzi, Y. (tarih yok). *Küme analizi.* Ondokuz Mayıs Üniversitesi. Erişim adresi: https://avys.omu.edu.tr/storage/app/public/yukselt/118305/k%C3%BCme%20analizi.pdf (Erişim Tarihi: 04.07.2024)

Bulut, H. (2024). *Veri Madenciliği Tekniklerinin Tarım Sektöründe Uygulanabilirliği.* Ondokuz Mayıs Üniversitesi. Erişim adresi: https://avys.omu.edu.tr/storage/app/public/hasan.bulut/118007/1.pdf (Erişim Tarihi: 04.07.2024)

Terzi, Y. (2019). *Faktör Analizi.* Ondokuz Mayıs Üniversitesi, Fen-Edebiyat Fakültesi, İstatistik Bölümü, Samsun. Erişim adresi: https://avys.omu.edu.tr/storage/app/public/yukselt/62069/fakt%C3%B6r%20analizi.pdf (Erişim Tarihi: 04.07.2024)