**İTÜ END620 - UYGULAMALI ÇOK DEĞİŞKENLİ İSTATİSTİKSEL ANALİZ DERSİ**

**UYGULAMA ÖDEVİ**

**(2018-2019 BAHAR)**

KONU:

UCI Wine Veri Setinin İçbağımlı Yöntemlerle Analizi

SEÇİLEN YÖNTEMLER:

Temel Bileşen Analizi (PCA) ve

Doğrusal Ayırma Analizi (LDA)

TAKDİM:

Doç. Dr. Umut ASAN

ÖDEVİ HAZIRLAYAN:

Uğur ÜRESİN, 507172109

İçindekiler

[1. GİRİŞ 1](#_Toc6955506)

[2. PROBLEMİN TANIMI VE ÇALIŞMANIN AMACI 2](#_Toc6955507)

[3. ÖNERİLEN MODEL 3](#_Toc6955508)

[3.1. Kavramlar, Değişkenler ve Ölçekleri 3](#_Toc6955509)

[4. YÖNTEM 4](#_Toc6955510)

[4.1. Kavramlar, Değişkenler ve Ölçekleri 4](#_Toc6955511)

[4.2 Temel Hipotezler 5](#_Toc6955512)

[4.3. Verilerin Gözden Geçirilmesi 5](#_Toc6955513)

[4.3.1. Dışa Düşen Veriler 6](#_Toc6955514)

[4.3.2. Eksik Veriler 6](#_Toc6955515)

[4.4 Varsayımların Gözden Geçirilmesi 7](#_Toc6955516)

[4.4.1. Normal Dağılıma Uygunluk 8](#_Toc6955517)

[4.4.2. Normallik için Dönüştürme 9](#_Toc6955518)

[4.4.3. Bartlett ve KMO Testleri 10](#_Toc6955519)

[5. R UYGULAMALI ÇOK DEĞİŞKENLİ İSTATİSTİKSEL ANALİZ 12](#_Toc6955520)

[6. BULGULAR VE DEĞERLENDİRME 16](#_Toc6955521)

[7. SONUÇ VE ÖNERİLER 17](#_Toc6955522)

[8. KAYNAKLAR 18](#_Toc6955523)

[9. EKLER 19](#_Toc6955524)

# 1. GİRİŞ

Çok değişkenli analiz, geniş kapsamlı olası araştırma durumlarını kapsayan veri analizi için sürekli genişleyen bir teknikler setidir. Gelişen tekniklerin yanı sıra, daha köklü olan çok değişkenli analiz yöntemleri aşağıdakileri listelenmiştir (Hair, 2010).

* Temel bileşenler ve ortak faktör analizi
* Çoklu regresyon ve çoklu korelasyon
* Çoklu diskriminant analizi ve lojistik regresyon
* Kanonik korelasyon analizi
* Çok değişkenli varyans analizi ve kovaryans
* Birleşik analiz (Conjoint analysis)
* Kümeleme analizi
* Algısal haritalama (Çok boyutlu ölçeklendirme olarak da bilinir)
* Uyum analizi (Correspondence analysis)
* Yapısal eşitlik modellemesi ve doğrulayıcı faktör analizi (Confirmatory factor analysis)

Bu çalışmada ‘Temel bileşenler analizi’ (PCA) ve ‘Doğrusal ayırma analizi’ (LDA) üzerinde durulacaktır. Üst başlıkta, her iki yöntem de içbağımlı bir yöntem olan faktör analizinin alt kırılımlarıdır. Bu çalışmada bu yöntemlerin teorik ayrıntılarına girilmeyecektir ancak seçilen bir veri seti üzerinde karşılaştırılmalı olarak sınamaları ve uygulamaları yapılacaktır.

Hem “Temel Bileşen Analizi” (PCA) hem de “Ortak faktör analizi” dahil olmak üzere faktör analizi, çok sayıda değişken arasındaki ilişkiyi analiz etmek ve bu değişkenleri ortak temel boyutları açısından açıklamak için kullanılabilecek istatistiksel bir yaklaşımdır. (Hair, 2010).

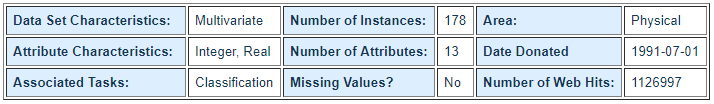
Diskriminant analizi, bir değişken türetmeyi içerir. Ayrımcı değişken, "a priori" olarak tanımlanan gruplardaki nesneler (kişiler, firmalar, vb.) arasında en iyi ayrımcılığa neden olacak iki (veya daha fazla) bağımsız değişkenin doğrusal birleşimidir. Ayrımcılık, gruplar arasındaki farkları en üst düzeye çıkarmak için değişkenin her bağımsız değişken için ağırlıklarının hesaplanmasıyla sağlanır (Hair, 2010).

‘Doğrusal ayırma analizi’ (LDA), daha yüksek özelliklere sahip veriler için boyutsallık azaltma çözümünde kullanılır. Örüntü sınıflandırma ve makine öğrenme uygulamaları için ön işleme adımlarından biridir. LDA aynı zamanda, özellik (‘feature’) çıkarma için kullanılır. Çoklu sınıflar arasındaki ayrımı en üst düzeye çıkaran doğrusal dönüşümü sağlar.

# 2. PROBLEMİN TANIMI VE ÇALIŞMANIN AMACI

Veriler daha az özellikle açıklanabiliyorsa, verinin temelini oluşturan işlem hakkında daha iyi bir fikir ediniriz ve bu anlamlı bilgi çıkarmaya (knowledge extraction) izin verir. Çoğu öğrenme algoritmasında, karmaşıklık, giriş boyutlarının (d) yanı sıra veri numunesinin boyutuna (N) da bağlıdır ve düşük bellek ve hesaplama maliyeti için, problemin boyutluluğunu azaltmakla ilgileniriz. D'nin azaltılması, test sırasında çıkarım algoritmasının karmaşıklığını da azaltır. Veriler bilgi kaybı olmadan birkaç boyutta gösterilebiliyorsa, yapı ve aykırı değerler için görsel olarak çizilebilir ve analiz edilebilir (Alpaydın, 2014).

Bu çalışmada farazi bir şarap şirketini için çalışan bir makine öğrenmesi (machine learning) ekibinin, ‘Wine’ veri setini (<https://archive.ics.uci.edu/ml/datasets/wine> adresinden indirilmiştir) kullanarak, şirket için katma değer yaratacak bir makine öğrenmesi uygulaması yapmak istediğini varsayalım.



Tablo 2.1: Wine verisinin özellikleri (<https://archive.ics.uci.edu/ml/datasets/wine> adresinden alınmıştır)

Söz konusu farazi ekip, Wine veri setindeki bilgilerden bir katma değer yaratmak için, şarapların tüm bilgilerini ve müşteri segmentleri hakkındaki bilgileri kullanarak bir sınıflandırma modeli hazırlayabilir. Böylece işletme, her yeni şarap için ürünü doğru müşteri segmentine sunabilir.

Diyelim ki bu şirketin pazarlama ekibi, öngörülen müşteri segmentlerinin doğru yerinde olup olmadığını görebilmek için tahmin bölgeleri ve sınıflandırma modelinin sınırları konusunda açık ve net bir görsel istiyor olsun. Mevcut verideki 13 bağımsız değişken ile böyle bir görselleştirme yapmak imkansızdır. Nitekim, buradaki amaç, değişkenliği en çok açıklayan yeni iki bağımsız değişken elde etmek ve bazı boyut azalatma tekniklerini uygulayarak tahmin bölgelerini ve tahmin sınırını görselleştirmektir. Bu amaç doğrultusunda ‘Temel Bileşenler Analizi’ (PCA) yöntemi uygulanacaktır.

Benzer şekilde hem bağımsız değişkenler arasındaki örüntüyü ortaya çıkarmak hem de verideki bilgilerden yola çıkarak tahmin bölgelerini belirlemek için ‘Doğrusal ayırma analizi’ (LDA) yöntemi uygulanacak, (PCA) yönteminden elde edilen bulgularla beraber yorumlanacaktır.

# 3. ÖNERİLEN MODEL

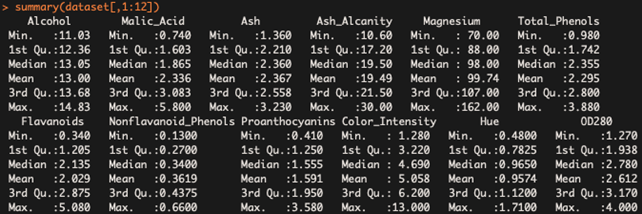
Bu çalışmada kullanılan Wine veri setinde 13 bağımsız değişken bulunmaktadır. Bağımsız değişkenlerin ardındaki örüntüyü görselleştirmek imkansızdır. Ancak bu örüntüyü oluşturan temel iki bileşen (faktör) bulunursa, bu örüntüyü resmetmek mümkün hale gelecektir. Bir başka deyişle, çok sayıdaki bağımsız değişken içbağımlı olarak yöntemlerle (PCA ve LDA) incelenecek ve değişkenlerin altında yatan anlamlı bilgiye ulaşılmaya çalışılacaktır.

3.1. Kavramlar, Değişkenler ve Ölçekleri

Bu çalışmada kullanılan ‘Wine’ veri setine ait bağımsız değişkenler aşağıdaki gibidir.

* Alcohol
* Malic acid
* Ash
* Alcalinity of ash
* Magnesium
* Total phenols
* Flavanoids
* Nonflavanoid phenols
* Proanthocyanins
* Color intensity
* Hue
* OD280 (OD280/OD315 of diluted wines)
* Proline

Verideki bağımsız değişkenlerin isimleri ve temel istatistikleri Şekil 3.1’de verilmiştir. Veride, bağımlı değişken olarak 3 seviyeli müşteri segmenti (‘Customer\_Segment’) bulunmaktadır. Yani bağımlı değişken, müşterinin üç segmentine karşılık gelmektedir. Bağımsız değişkenlerin tamamı nümerik olup, bağımlı değişken 3 seviyeli kategorik bir değişkendir.



Şekil 3.1: Wine verisindeki bağımsız değişkenler ve temel istatistikleri

# 4. YÖNTEM

Temel bileşen analizi (PCA), çok değişkenli istatistiksel analizde anahtar araçlardan biridir. Her biri diğer bileşenler tarafından açıklanmayan verilerden azami miktarda varyasyon içeren bileşenler oluşturmayı amaçlamaktadır. Böylece kullanıcı, verilerdeki bilgilerin, çoğu zaman pratikte olduğu gibi birkaç temel bileşene özetlenebileceğini ummaktadır. Temel bileşenler belirlendikten sonra, ilgili bilgileri taşıdıkları için orijinal veriler yerine tüm ileri analizler yapılabilir (Serneels ve Verdonck, 2008).

Temel bileşen analizi (PCA), çıktı bilgisini kullanmadığından gözetimsiz (‘unsupervised’) bir yöntemdir. Bu yöntemde maksimize edilecek kriter ise varyanstır (Alpaydın, 2014).

4.1. Kavramlar, Değişkenler ve Ölçekleri

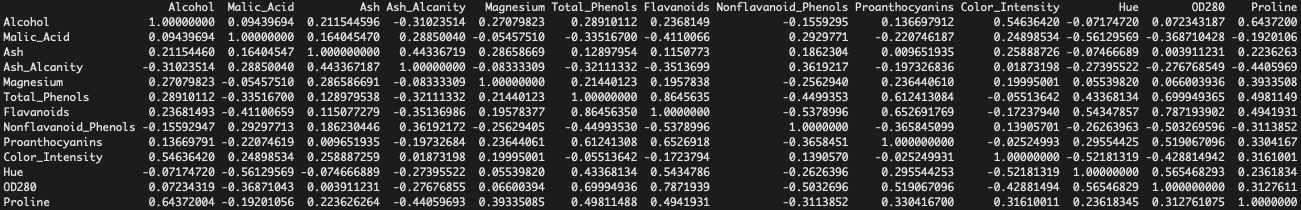
* Korelasyon Matrisi: Gözlenen değişkenlerden üretilen korelasyon matrisine gözlenen korelasyon matrisi (observed correlation matrix), faktörlerden üretilen korelasyon matrisine üretilmis korelasyon matrisi (reproduced correlation matrix) adı verilir. Gözlenen ve üretilmiş korelasyon matrislerinin arasındakı fark ise, hata korelasyon matrisi (residual correlation matrix) olarak isimlendirilir.
* Özdeğer (Eigenvalue): Özdeğer, her bir faktörün faktör yüklerinin kareleri toplamı, her bir faktör tarafından açıklanan varyansın oranının hesaplanmasında ve önemli faktör sayısına karar vermede kullanılan bir katsayıdır. Özdeğer yükseldikçe, faktörün açıkladığı varyans da yükselir.
* Faktör Yük Değeri (Factor Loading): Faktör yük değeri, maddelerin faktörlerle olan iliskisini açiklayan bir katsayidir. Maddelerin yer aldıkları faktördeki yük değerlerinin yüksek olması beklenir. Bir faktörle yüksek düzeyde ilişki veren maddelerin oluşturdugu bir küme var ise bu bulgu, o maddelerin birlikte bir kavramı-yapıyı-faktörü ölçtüğü anlamına gelir. Bir değişkenin 0.3'lük faktör yükü, faktör tarafindan açıklanan varyansın %9 olduğunu gösterir. Bu düzeydeki varyans dikkat çekicidir ve genel olarak, işaretine bakılmaksızın 0.60 ve üstü yük değeri yüksek; 0.30-0.59 arası yük değeri orta düzeyde büyüklükler olarak tanımlanabilir ve değişken çıkartmada dikkate alinir. Faktör yük değerleri, bir korelasyon değeri olarak istatistiksel anlamlılık bakımından da incelenebilir. Ancak, düşük korelasyon miktarlarının da, örneklem arttıkça anlamlı çıkma olasılığının artacağı unutulmamalıdır (Kline, 1994). Faktör yük değeri, bazen faktör katsayısı (factor coefficient) olarak isimlendirilir.

4.2 Temel Hipotezler

PCA yaklaşımındaki ana hipotezlerden biri, verilerin geçici olarak i.i.d. (**ing. independent and identically distributed)** olasılık teorisi ve istatistikte, her bir rastgele değişkenin diğerleri ile aynı olasılık dağılımına sahip olması ve hepsi birbirinden bağımsız olması durumunda, rastgele değişkenlerin toplanması bağımsızdır ve özdeş olarak dağılır) olduğudur.

İkinci ana hipotez, gözlem modelinin kesin olması yani modelde gürültü (Noise) olmamasıdır (Djafari, 2006).

Bağımsız değişkenlerin birbirleri ile olan korelasyonları Şekil 4.1’de verilmiştir.

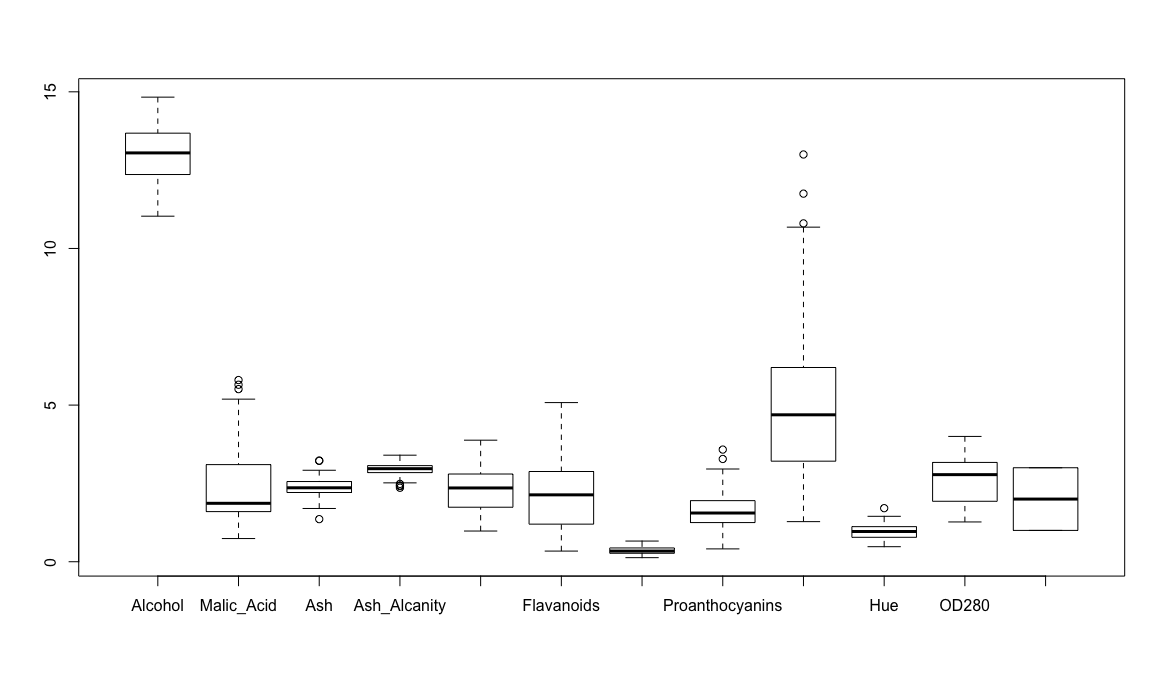


Şekil 4.1: Wine verisindeki bağımsız değişkenlere ait korelasyon matrisi

Veriler arasındaki korelasyonların çok güçlü olmadıkları görülmektedir. Dolayısıyla eldeki bilgiler ışığında verilerin bağımsız oldukları varsayımı yapılabilir.

4.3. Verilerin Gözden Geçirilmesi

Veri setindeki bağımsız değişkenlere ait kutu diagramları (box-plot) Şekil 4.2’de verilmiştir.



Şekil 4.2: Wine verisindeki bağımsız değişkenlere ait kutu diagramları

(Nümerik değerleri çok büyük olduğu için 5. ve 13. Değişkenler yer almamaktadır)

Tüm bağımsız değişkenlerin farklı ‘Customer Segment’ seviyeleri için ayrı ayrı kutu diagramları ise Ek-A1’de verilmiştir. Bu diagramlara bakılırsa, neredeyse tüm bağımsız değişkenlerde müşteri segmentine göre bir farklılaşma olduğu görsel olarak da tespit edilmiştir.

4.3.1. Dışa Düşen Veriler

Veri setindeki bağımsız değişkenlere ait kutu diagramları (box-plot) Ek-A1’de verilmiştir. Hemen her bağımsız değişkende dışa düşen veriler vardır. Dışa düşen verilerin sayısı özellikle Malic Acid (Bkz. Şekil A1.2), Magnesium (Bkz. Şekil A1.5) ve Proanthocyanins (Bkz. Şekil A1.9) değişkenlerinde çok sayıda dışa düşen veri bulunmaktadır.

Hem varyans hem de varyans-kovaryans matrisinin aykırı değerlere duyarlı olduğu bilinmektedir. Bu nedenle, aynı sonuç bir bütün olarak PCA için de geçerlidir: bu bir robust olmayan yöntemdir. Dışa düşen veriler, ana bileşenin (faktörlerin) yakınmasının (fitting) bozulmasına ve sonuçların kötü yorumlanmasına neden olabilir. Dışa düşen veriler aynı zamanda maskeleme etkisine (‘masking effect’) de neden olabilir: Dışa düşen verilerden dolayı model, temel bileşenlere dayanarak, dışa düşen veri tespit edilemeyecek şekilde çarpıtılır (Serneels ve Verdonck, 2008).

4.3.2. Eksik Veriler

Bu çalışmada kullanılan ‘Wine’ veri setinde eksik veri bulunmamaktadır. Eksik veri bulunan veri setlerini düzeltmek için kullanılan yöntemlerden bazıları aşağıda verilmiştir.

* Tüm gözlemi Silme: Herhangi bir katılımcıdan eksik değerleri olan tüm verilerin silinmesidir. Veri seti yeterince büyükse, büyük miktarda istatistiksel güç kaybı olmadan eksik değer içeren gözlemler veri setinden düşürülebilir.
* Değerlerin telafi edilmesi: Bazen katılımcılarla iletişim kurulabilir ve eksik değerleri doldurmaları istenebilir.
* Akıllı tahmin: Bir gözleme ait boş olan değerin veya değerlerin, aynı gözlemdeki diğer değerlerden çıkarılması veya makul bir güvenirlik ile tahmin edilmesidir. Örneğin, bir veri setinde kişilerin hem eğitim seviyesi (İlk, Orta, Lise, Üniversite) hem de meslekleri olsun. Eğer mesleği mühendislik olan bir katılımcının eğitim seviyesi hanesi eksik ise doğrudan ‘Üniversite’ yazılabilir..
* İkame değer kullanma: Veri setindeki eksik bırakılmış sayısal veriler diğer gözlemlerdeki değerlerden faydalanılarak yapay olarak elde edilebilir. Genel olarak eksik girişler için, ilgili değişken için diğer gözlemlere ait değerlerin aritmetik ortalaması ya da medyan değeri girilebilir. Bir çok uygulamada, dışa düşen değerlerden etkilenmemek için medyan daha çok tercih edilir.

4.4 Varsayımların Gözden Geçirilmesi

Faktör analizinin altında yatan kritik varsayımlar istatistiksel olmaktan ziyade kavramsaldır. Araştırmacı, her zaman çok değişkenli bir teknik için istatistiksel gereksinimin karşılanmasıyla ilgilenmektedir, ancak faktör analizinde, öncelikli endişeler, analizde yer alan değişkenlerin karakteristik özellikleri ve kompozisyonları ile ilgili istatistiksel nitelikleri kadar önemlidir.

Faktör analizinin altında yatan kavramsal varsayımlar, seçilen değişkenler ve seçilen örneklem ile ilgilidir. Faktör analizinin temel bir varsayımı, bazı temel yapıların seçilen değişkenler setinde mevcut olduğudur. İlişkilendirilmiş değişkenlerin varlığı ve takip eden faktörlerin tanımı, istatistiksel gereklilikleri yerine getirseler bile ilişki düzeyini garanti etmez. Gözlemlenen kalıpların kavramsal olarak geçerli olmasını ve faktör analizi ile çalışmak için uygun olmasını sağlamak araştırmacının sorumluluğundadır, çünkü tekniğin değişkenler arasındaki korelasyondan başka uygunluğu belirleme aracı yoktur. Örneğin, bağımlı ve bağımsız değişkenlerin tek bir faktör analizinde karıştırılması ve ardından bağımlılık ilişkilerini desteklemek için türetilmiş faktörlerin kullanılması uygun değildir.

Araştırmacı ayrıca, örneklemin temel faktör yapısına göre homojen olmasını sağlamalıdır. Cinsiyet nedeniyle farklılık gösterdiği bilinen bir madde kümesi için bir erkek ve kadın örneğine faktör analizi yapılması uygun değildir. İki alt örnek (erkekler ve kadınlar) birleştirildiğinde, ortaya çıkan korelasyonlar ve faktör yapısı, her grubun benzersiz yapısının zayıf bir temsili olacaktır. Bu nedenle, numunede farklı gruplar beklendiğinde, ayrı faktör analizleri yapılmalı ve sonuçlar, birleştirilmiş numunenin sonuçlarına yansıtılmayan farklılıkları tanımlamak için karşılaştırılmalıdır.

İstatistiki açıdan bakıldığında normallik, varyansların eşitliği (‘homoscedasticity’) ve doğrusallıktan ayrılma, yalnızca gözlemlenen korelasyonları azalttığı ölçüde uygulanır. Faktörlerin önemine istatistiksel bir test uygulanırsa sadece normallik gereklidir, ancak bu testler nadiren kullanılır. Aslında, bir dereceye kadar çoklu doğrusallık arzu edilir, çünkü amaç birbiriyle ilişkili değişken kümelerini tanımlamaktır (Hair, 2010).

Uygulamada, faktör analizi için aşağıda verilen varsayımlar sınanır.

1. Örneklem büyüklüğü 150’den fazla olmalı ve her değişken için 5-10 arası gözlem bulunmalıdır (Burada 179 gözlem ve 13 değişken var. Bu da değişken başına yaklaşık 14 gözlem demek).
2. Bartlett ve KMO testleri anlamlı çıkmalıdır (Bkz. Bölüm 4.4.3)
3. Doğrusallık: Genellikle faktör analizi yapılan faktörler birbiri ile ilişkili ve örneklem sayısı da en az 150 olunca, doğrusallık kendiliğinden sağlanır denilebilir.
4. Outlier’lar atılır ve işlem tekrar yapılır.

4.4.1. Normal Dağılıma Uygunluk

Verilerin normal dağılımlarını test etmek için aşağıdaki testler kullanılabilir:

* Shapiro-Wilk Testi
* Jarque-Bera Testi

Normal dağılmayan değişkenin veya değişkenlerin bulunması için aşağıdaki testler kullanılabilir:

* Anderson-Darling Test
* Cramer-Von Mises Test
* Kolmogorov-Smirnov Test
* Pearson chi-square Test
* Shapiro-Francia Test

Parametrik testlerin yanısıra, bazı görselleştirme teknikleri de verilerin normal dağılıp dağılmadığını tespit etmek için kullanılmaktadır. Bunlardan en çok kullanılan ikisi yoğunluk dağılım grafiği (‘density plot’) ve QQ-diagramıdır (QQ-plot).

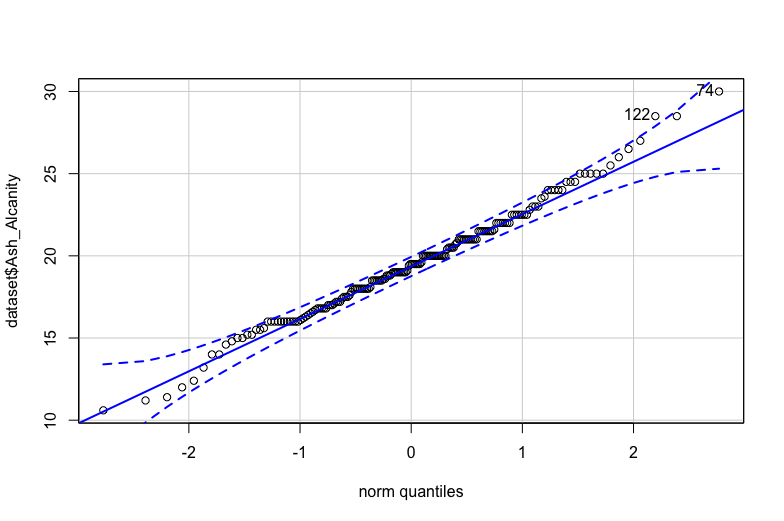
Bu çalışmada, değişkenlerin normal dağılıp dağılmadığını control etmek için hem parametrik bir test olan Shapiro-Wilks testi uygulanmış hem de görsel bir sınama için QQ-diagramları çizdirilmiştir (Tüm QQ-diagramları Ek-A2’de verilmiştir).

Bağımsız değişkenlerin Shapiro-Wilks testinden elde edilen p-değerleri Tablo 4.1’de verilmiştir.

|  |  |  |
| --- | --- | --- |
| **Değişken** | **p-value** | **α** |
| Alcohol | 0.02004798 | 0,05 |
| Malic acid | 2,95E-04 | 0,05 |
| Ash | 0.03868278 | 0,05 |
| Alcalinity of ash | 0.2638688 | 0,05 |
| Magnesium | 6,35E-01 | 0,05 |
| Total phenols | 0.004395305 | 0,05 |
| Flavanoids | 1,68E-05 | 0,05 |
| Nonflavanoid phenols | 0.00010551 | 0,05 |
| Proanthocyanins | 0.01445402 | 0,05 |
| Color intensity | 9,23E-02 | 0,05 |
| Hue | 0.0174255 | 0,05 |
| OD280 | 2,32E-06 | 0,05 |
| Proline | 1,74E-07 | 0,05 |

Tablo 4.1: Shapiro-Wilk testinden elde edilen p-değerleri

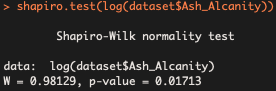
Görüldüğü üzere, yalnızca 4. değişkene (‘Alcalinity of ash’) ait p-değeri kritik değerden (α=0.05) aşmıştır. Bu değişkene ait QQ-diagramına bakacak olursak:

****

Şekil 4.1: 4 nolu bağımsız değişkene ait QQ diagramı

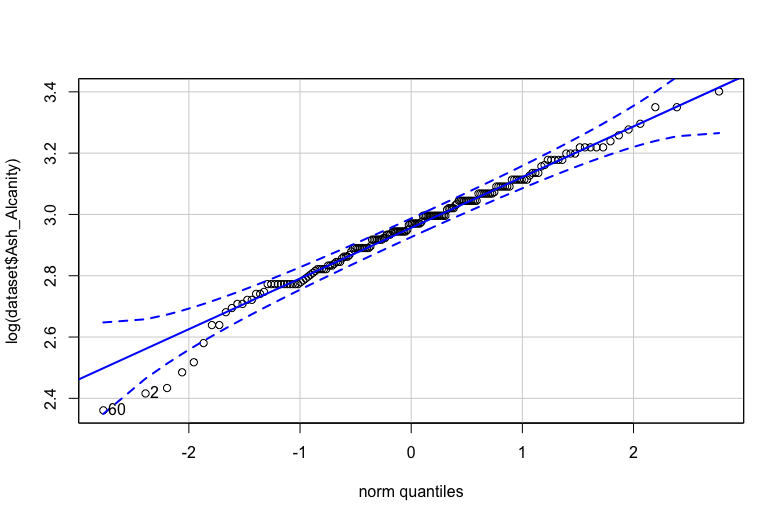
4.4.2. Normallik için Dönüştürme

Şekilde 4.1’den görüldüğü üzere, 4. değişkenin (‘Alcalinity of ash’) yoğunluklu olarak, normal dağılım için referans verilen bölgenin içinde kaldığı, referans bölge dışına sapmanın çok az olduğu söylenebilir. Yine de bu değişkene logaritmik dönüşüm uygulanarak (tüm değerlerin logaritması alınarak) normal dağılıma yaklaştırılmıştır. Yeni durumda, Shapiro-Wilks testinden p-değeri 0.01713 olarak elde edilmiştir (Bkz Şekil 4.2).



Şekil 4.2: Dönüştürülmüş 4 nolu bağımsız değişkene ait Shapiro-Wilk testi sonuçları

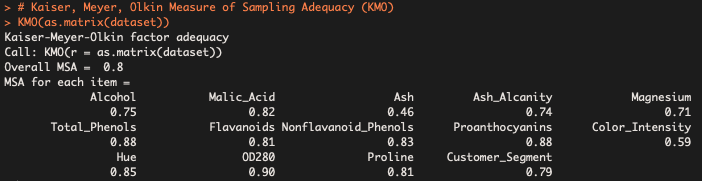
Dönüştürülmüş verinin QQ-diagramı Şekil 4.3’te verilmiştir.



Şekil 4.3: Dönüştürülmüş 4 nolu bağımsız değişkene ait QQ diagramı

* + 1. Bartlett ve KMO Testleri

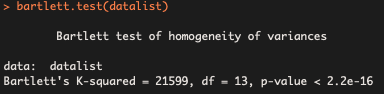
Kaiser- Meyer- Olkin (KMO) and Bartlett's Test, değişkenler arası ilişkinin gücünü ölçümler. KMO, tatmin edici bir faktör analizi yapılabilmesi için örneklemin 0,5’den büyük olması gerektiğini öngörür. Şayet, 0,5’den düşük değişkenler var ise bunlar analize dahil edilmez. Araştırmacı, değişken başına en az 10-15 katılımcıya sahip olmalıdır. Fiedel, tatminkar bir analizin yapılabilmesi için 300 katılımcıyı yeterli görmektedir. Kaisen gibi araştırmacılar ise en az 0,5 olması gerektiğini, 0,7- 0,8 aralığının kabul edilebilir diğer yandan 0,9 ve üzerinin çok ideal olduğunu belirtmişlerdir. KMO ölçümü 0 ile 1 arasında değişir. 1’e yaklaştıkça daha iyidir. Minimum 0,6 olması önerilir. KMO testinin sonuçları Şekil 4.4’te verilmiştir.



Şekil 4.4: KMO testi sonuçları

Görüldüğü üzere bir değişken dışında (Ash) KMO=0.5 değerinin üstündedir. Ash değişkenine ait değer KMO=0.5 referans değerine çok yakın olduğu için veri setinden çıkarılmamıştır.

Barlett's test de aynı KMO gibi değişkenler arası ilişki gücünü ölçümler. Bu test H0’ı test eder. Barlett’s testte (Barlet’s test of sphericity) sig <0.05 olmalıdır. 0,5’in altındaysa H0 reddedilir bu da korelasyon matrisinin özdeş bir matris olmadığı anlamına gelir. Eğer tabloda Bartlett değeri p<0.001 ise bu ilerleyebileceğimizi gösterir. Barlett testinin sonuçları ise Şekil 4.5’te verilmiştir.



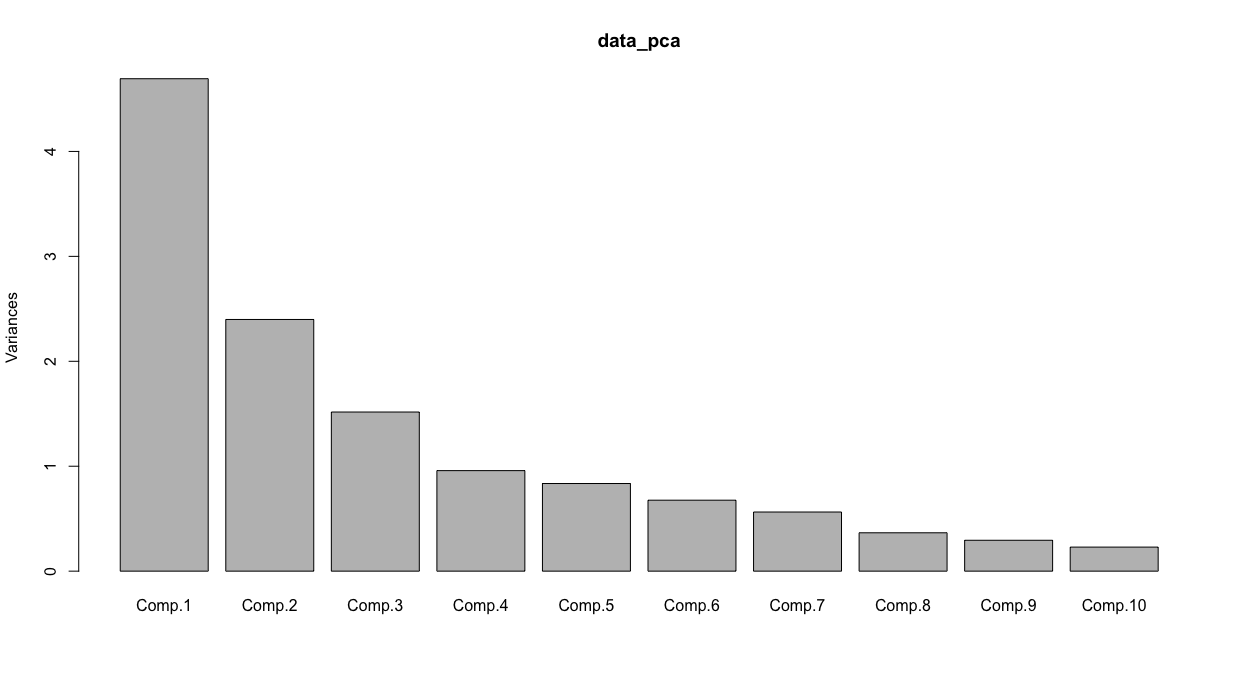
Şekil 4.5: Barlett testi sonuçları

Görüldüğü üzere p-değeri muazzam bir şekilde kritik değerden (α=0.01) düşüktür. Dolayısıyla veri seti bu haliyle faktör analizlerine uygundur.

# 5. R UYGULAMALI ÇOK DEĞİŞKENLİ İSTATİSTİKSEL ANALİZ

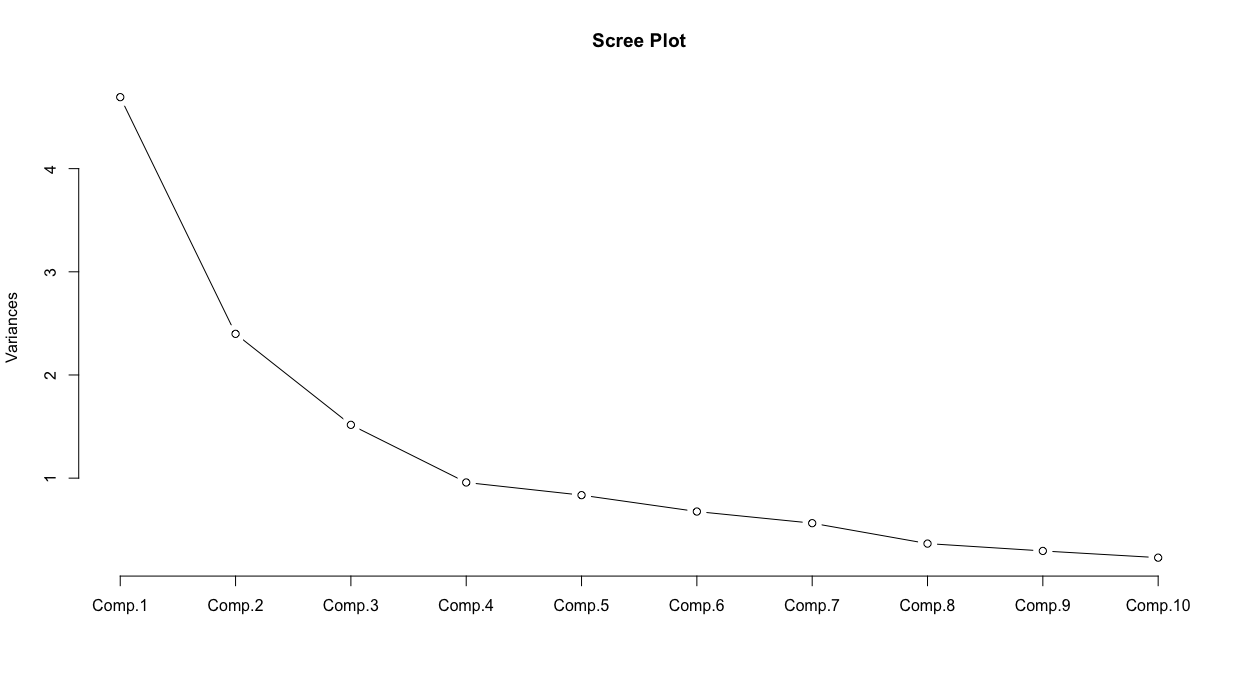
Bu çalışmada, hem PCA hem de LDA için R programlama dili kullanıldı. Öncelikle verinin normalitesi kontrol edildi. Bölüm 4.4.2’de verildiği üzere 4 nolu bağımsız değişken (‘Ash\_Alcanity’) logaritması alınarak normal dağılıma benzetildi. Daha sonra veri setinin %80’i ‘traning set’ (modeli oluşturmak için kullanılacak veriler) olarak; kalan %20’si ise ‘test set’ (oluşturulan modelinm sınanması için kullanılacak veriler) olarak bölündü. Sonuç olarak 178 satır verinin 142’si ‘training set’, kalan 38’i ise ‘test\_set’ olarak bölünmüş oldu. Daha sonra bağımsız değişkenlere ait değerler ölçeklendirilip merkezlendi. (Burada, merkezin değeri sütun merkezlemenin nasıl yapıldığını belirler. Merkez, X sütun sayısına eşit uzunlukta bir sayısal vektör ise, her X sütunu ondan çıkartılan merkezden karşılık gelen değere sahiptir).

Training ve test setleri oluşturulduktan sonra, traning set üzerinden temel bileşenler hesaplandı. Her bir bileşene ait değişkenlik miktarı (varyans) Şekil 5.1’de verilmiştir.



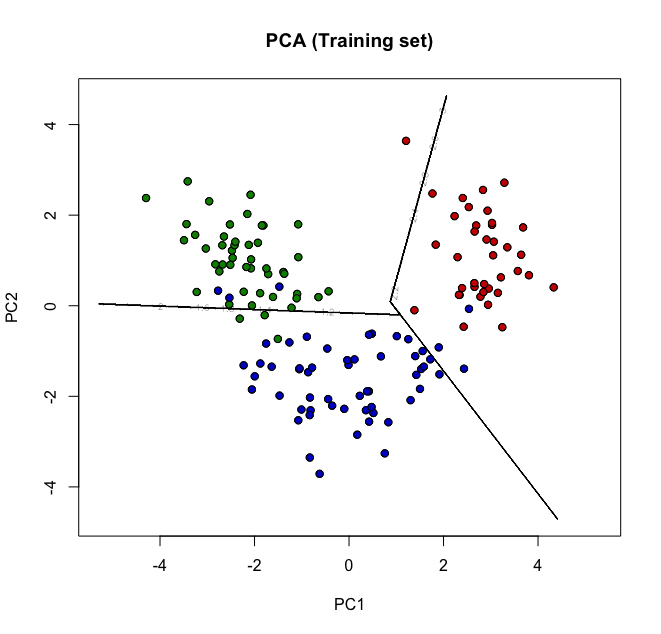
Şekil 5.1: Temel bileşenler varyans grafiği

Buradaki kritik soru kaç temel bileşenin yeterli olacağıdır. Bunu tespit etmenin yollarından biri referans bir varyans ölçütünü sağlayan ‘asgari faktör sayısına’ kadar boyutu indirgemektir. Şayet araştırmacı, modeli oluşturan bileşenleri bir başka deyişle problemin işleyişini çok iyi biliyorsa, faktör seçimi noktasında sezgisel de davranabilir. Buna rağmen, araştırmacı yine de elde ettiği sonuçları sınamalı ve gerekirse analizi başka bir faktör sayısı ile tekrarlamalıdır. Bir diğer yol ise görselleştirmelerden faydalanmaktır. En sık kullanılan görselleştirme ‘Scree’ diagramı (‘Scree plot’) idir. Training seti için scree plot Şekil 5.2’de verilmiştir.



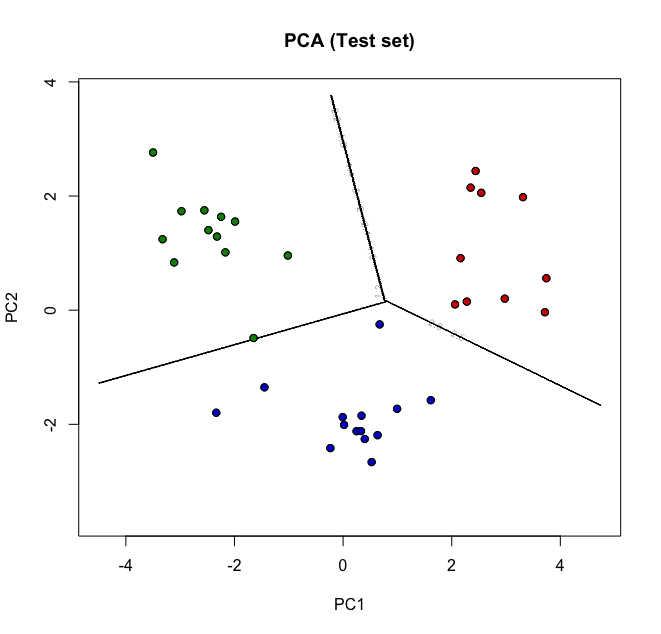
Şekil 5.2: Temel bileşenler scree-diagramı

ScreePlot özdeğerin grafik olarak gösterimidir. Her faktörün açıkladığı varyans miktarının gösterir. Kırılma noktaları arasındaki değişim dikkate alınarak yorumlanır. İlk faktör en yüksek varyansı açıklar. En son faktör ise en düşük varyansı açıklar. Scree test grafiği yatay şekil aldığı noktaya kadar olan faktörler elde edilecek maksimum faktör sayısı olarak kabul edilir. Burada 2 faktör alınmıştır ki problemin amacına da uygundur (Görsellik için en fazla 3 faktör olabilirdi zira 3. boyuttan sonra görselleştirme mümkün değil’). Faktör sayısı belirlendikten sonra, training seti ile PCA uygulanarak ‘Customer Segment’lerin ne kadar ayrıldığına bakılır. PCA ile elde edilen faktörlerin training setini nasıl ayırdığı Şekil 5.3’te verilmiştir (Yeşil, Mavi, Kırmızı noktalar sırasıyla 1., 2. ve 3. Segmentleri temsil etmektedir).



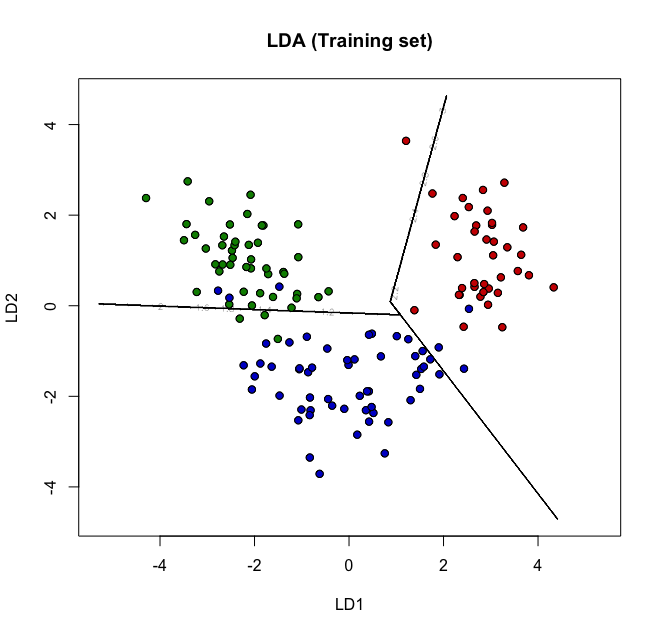
Şekil 5.3: PCA ile elde edilen faktörlerin training setini ayırması

Aynı şekilde PCA ile elde edilen faktörlerin test setini nasıl ayırdığı ise Şekil 5.4’te verilmiştir.

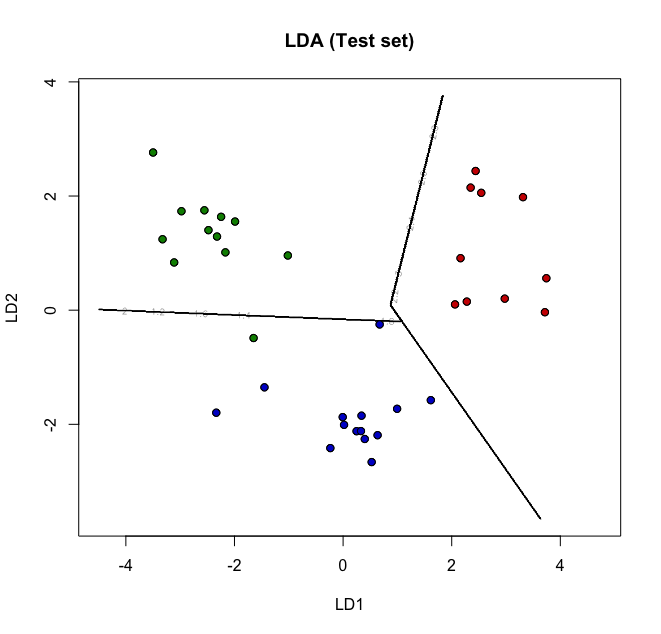


Şekil 5.4: PCA ile elde edilen faktörlerin training setini ayırması

PCA’deki veri hazırlama işlemlerinin tamamı LDA için de geçerlidir. Dolayısıyla hazırlanan training ve test setleri farklı isimlere atanıp LDA analizine tabii tutuldu. Sonuçlar Şekil 5.5 ve 5.6’da gösterilmiştir.



Şekil 5.5: LDA ile elde edilen faktörlerin training setini ayırması



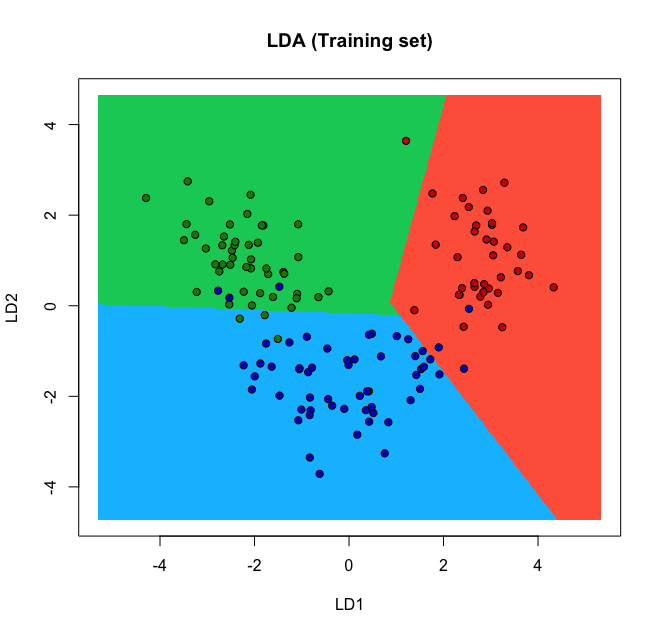
Şekil 5.5: LDA ile elde edilen faktörlerin test setini ayırması

# 6. BULGULAR VE DEĞERLENDİRME

Bu çalışmada içbağımlı teknikler olaran ‘Temel bileşen analizi’ (PCA) ve ‘Doğrusal ayırma analizi’ (LDA) yöntemleri kullanılarak, Wine verisi içerisindeki örüntü ortaya çıkarılmış ve çok boyutlu veri seti iki temel boyuta indirgenerek, çıktının (‘Customer Segmentation’) doğru bir biçimde yapılıp yapılmadığına dair görselin elde edilmesi sağlanmıştır. PCA ve LDA yöntemleri ile test verisine ait sonuçlar sırasıyla Şekil 6.1 ve Şekil 6.2’de verilmiştir.



Şekil 6.1: PCA ile elde edilen faktörlerin training setini ayırması



Şekil 6.2: LDA ile elde edilen faktörlerin training setini ayırması

# 7. SONUÇ VE ÖNERİLER

Bölüm 4.4’te verilen önkoşulların (varsayımlar) hepsi sınanmış sadece dışa düşen değerlerin (‘outlier’) atılması şartı gerçekleştirilmemiştir. Veri mevcut haliyle beklenen sonuçları vermemiş olsaydı, sınama aşamasına geri dönüp dışa düşen verileri veri setinden çıkarmak gerekebilirdi.

Görsellere bakılarak, iki yöntemin de test performansı %100 olmakla beraber (test setinin hata matrisine göre), training setindeki görselleştirme açısından LDA ile PCA arasında göre çok küçük bir fark olduğu söylenebilir. PCA segment 1 ve 3’ü kendi aralarında %100 bir şekilde ayırırken (Bkz. Şekil 6.1), LDA, segment 1 ve 3’ü kendi aralarında %100 ayıramamıştır. Her iki yöntem de segment 1 ile 2’yi kendi arasında ve 2 ile 3’ü kendi arasında %100 ayıramamıştır ancak genele bakıldığında hem PCA hem de LDA çok başarılı sonuçlar vermiştir.

Araştırmacılar, benzer veri setleri ile çalışırken, alternatif olarak ‘Karesel ayrıma analizi’ (Quadratic Discriminant Analysis, QDA) ve ‘Ayarlanmış ayırma analizi’ (Regularized Discriminant Analysis, RDA) yöntemlerine de başvurulabilir ve bu yöntemlerin tümünü kıyaslayabilirler.

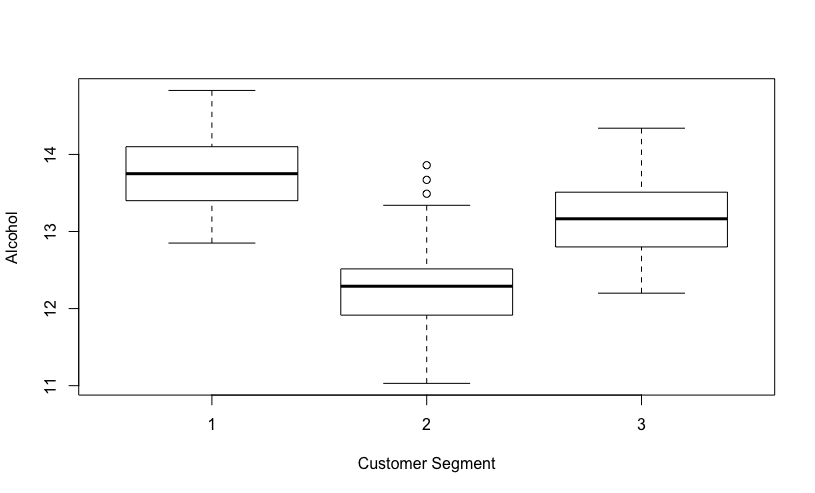
# 8. KAYNAKLAR

* Hair J., Black W., Babin B. and Anderson R. (2010) Multivariate Data Analysis: Global Edition. 7th edn. Upper Saddle River: Pearson Higher Education.
* Alpaydın E. (2014) Introduction to Machine Learning. 3th edn. London: The MIT Press
* Djafari A. (2006) Bayesian Source Separation: Beyond PCA and ICA. ESANN'2006 proceedings - European Symposium on Artificial Neural Networks.
* Serneels S., Verdonck T. (2008) Principal component analysis for data containing outliers and missing elements. Computational Statistics & Data Analysis 52.
* Hastie T., Tibshirani R. & Friedman J. (2009). The Elements of Statistical Learning.2nd ed. New York: Springer
* Web-1: <https://www.rdocumentation.org/packages/graphics/versions/3.5.3> (23.04.2019 itibariyle)

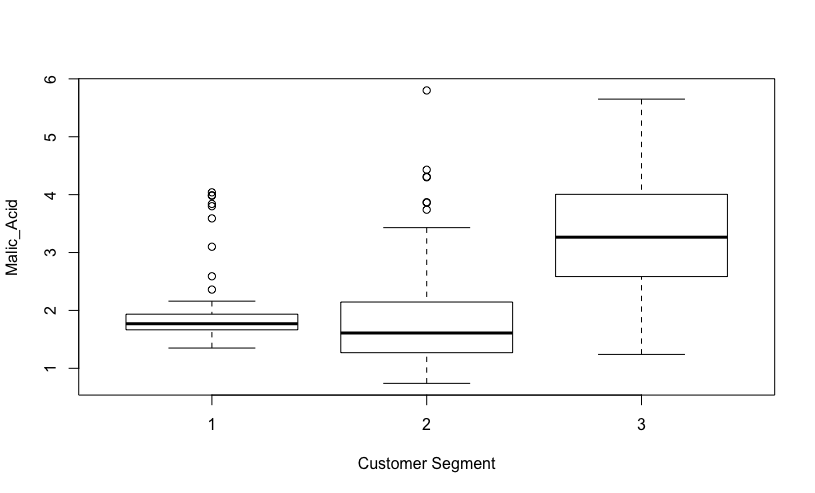
# 9. EKLER

**Ek-A. Veriye ait görselleştirmeler**

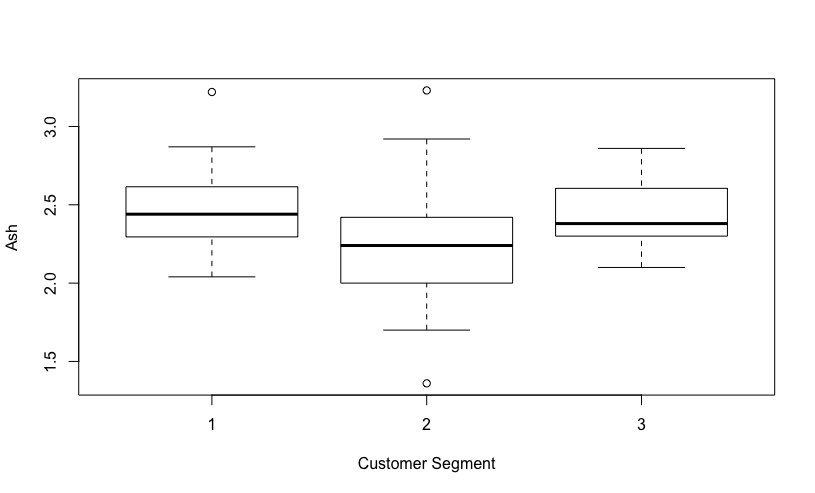
**A1. Verilerin kutu diagramları**



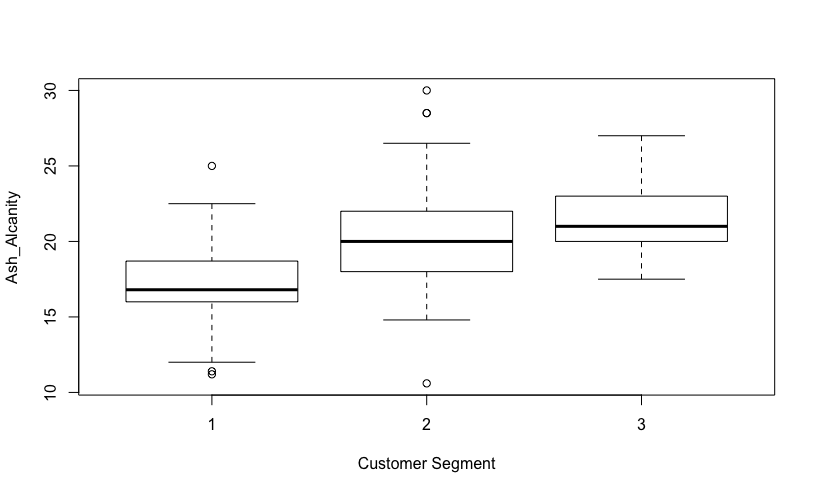
Şekil A1.1: 1 nolu bağımsız değişkene ait kutu diagramı



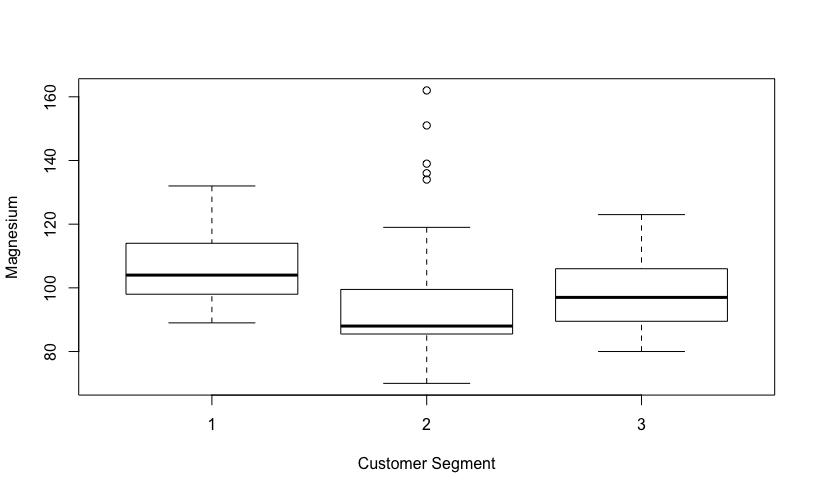
Şekil A1.2: 2 nolu bağımsız değişkene ait kutu diagramı



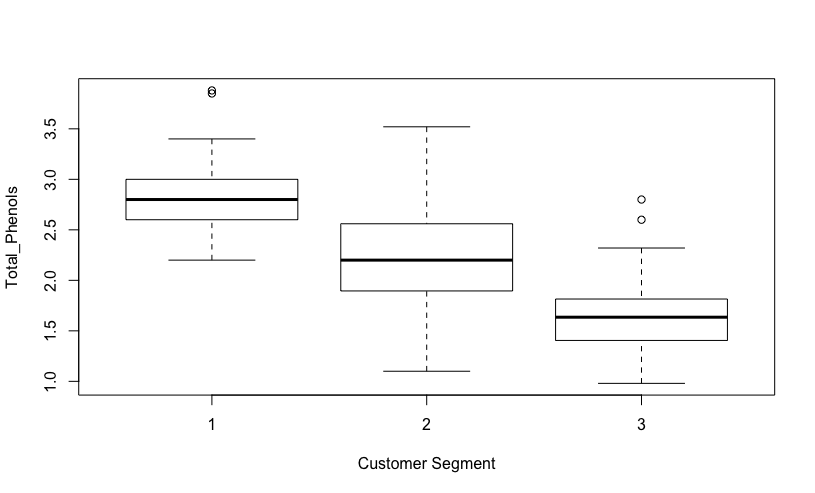
Şekil A1.3: 3 nolu bağımsız değişkene ait kutu diagramı



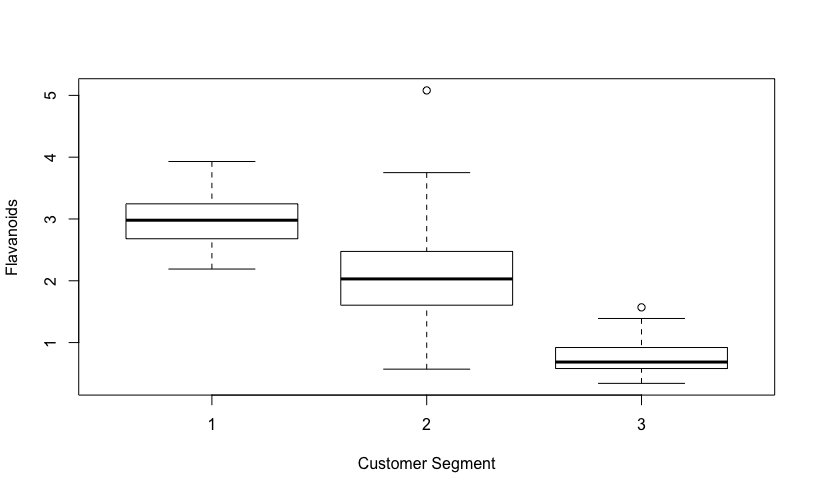
Şekil A1.4: 4 nolu bağımsız değişkene ait kutu diagramı



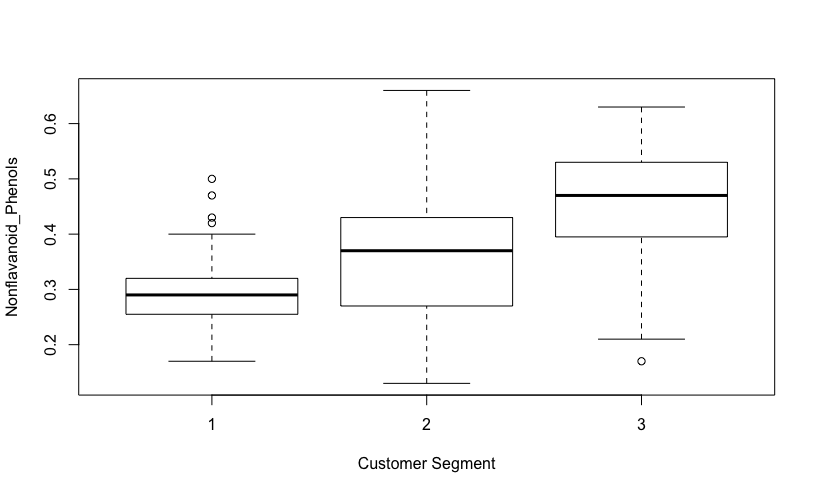
Şekil A1.5: 5 nolu bağımsız değişkene ait kutu diagramı



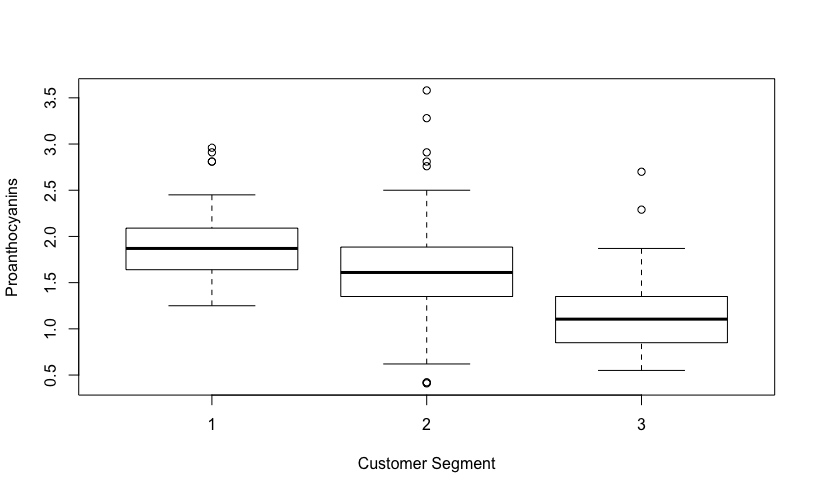
Şekil A1.6: 6 nolu bağımsız değişkene ait kutu diagramı

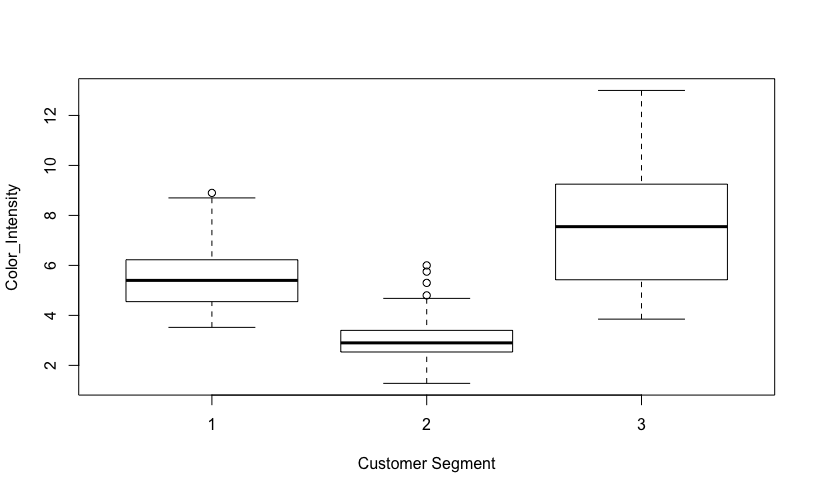


Şekil A1.7: 7 nolu bağımsız değişkene ait kutu diagramı

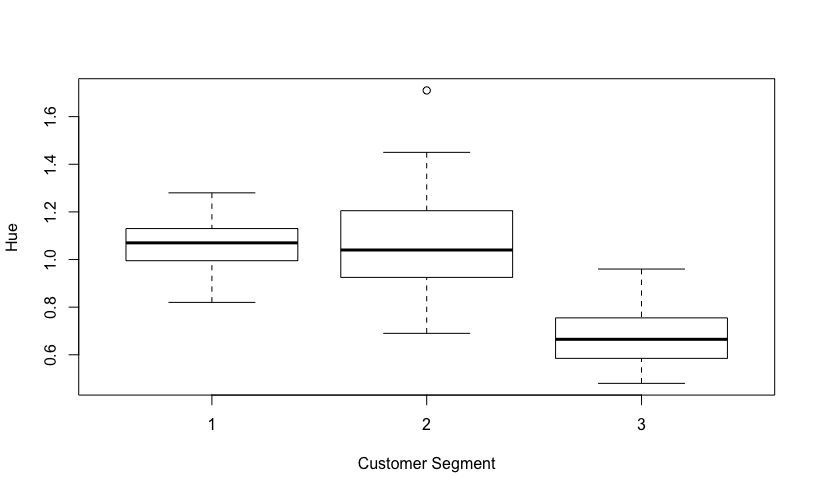


Şekil A1.8: 8 nolu bağımsız değişkene ait kutu diagramı

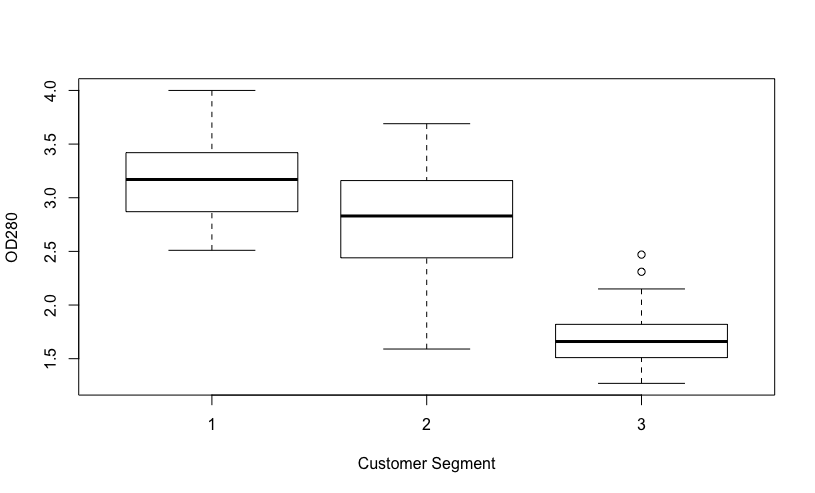


Şekil A1.9: 9 nolu bağımsız değişkene ait kutu diagramı

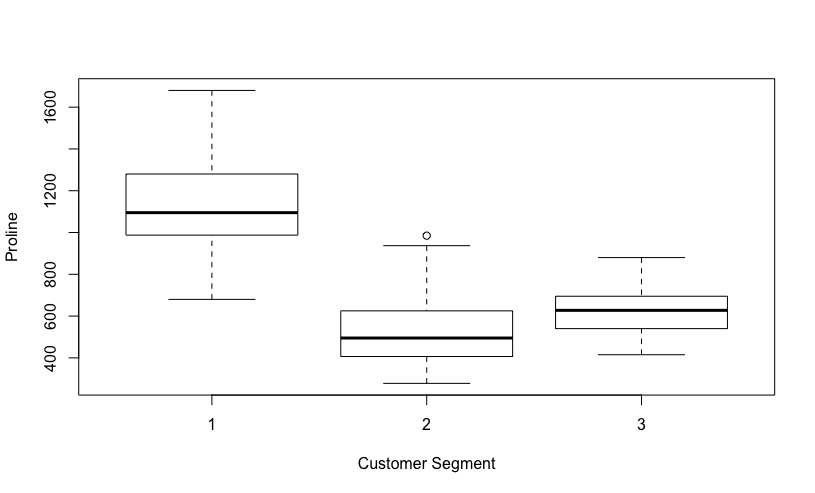
Şekil A1.10: 10 nolu bağımsız değişkene ait kutu diagramı



Şekil A1.11: 11 nolu bağımsız değişkene ait kutu diagramı

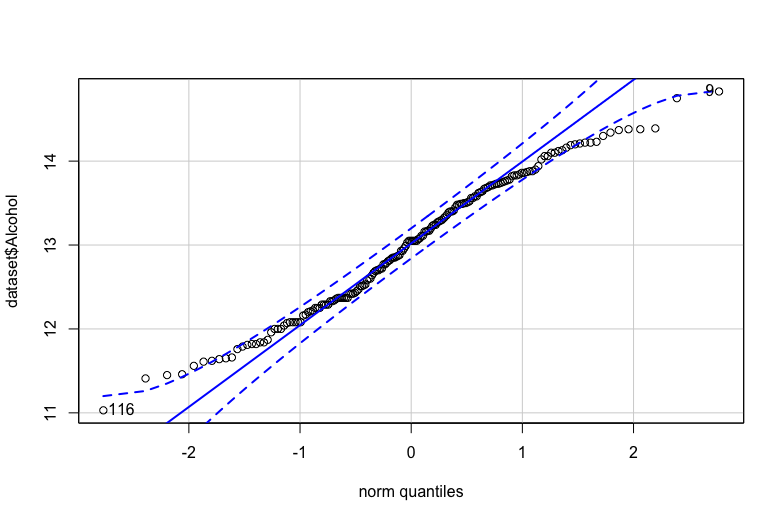


Şekil A1.12: 12 nolu bağımsız değişkene ait kutu diagramı

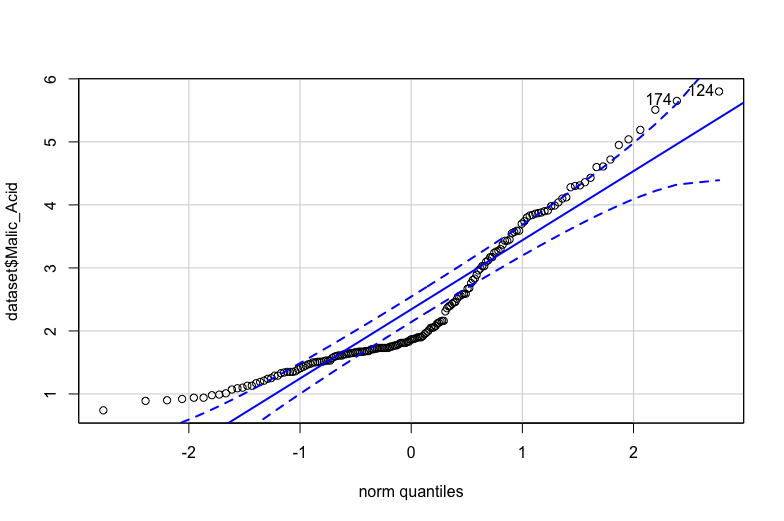


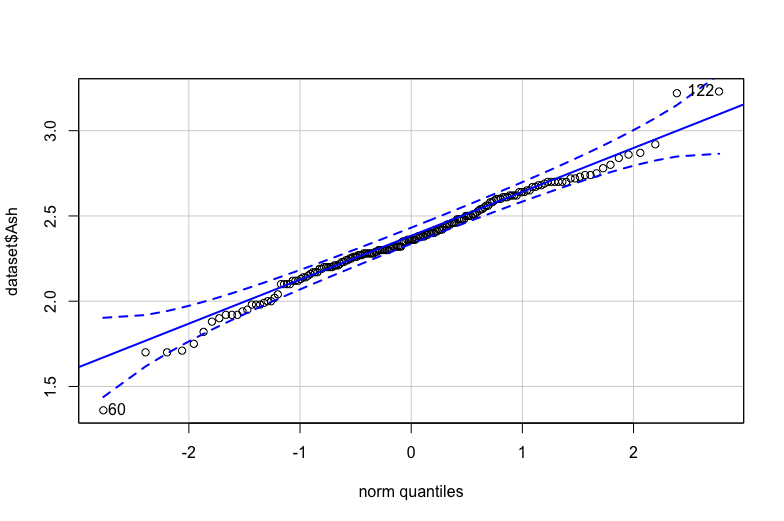
Şekil A1.13: 13 nolu bağımsız değişkene ait kutu diagramı

**A2. Verilerin QQ diagramları**

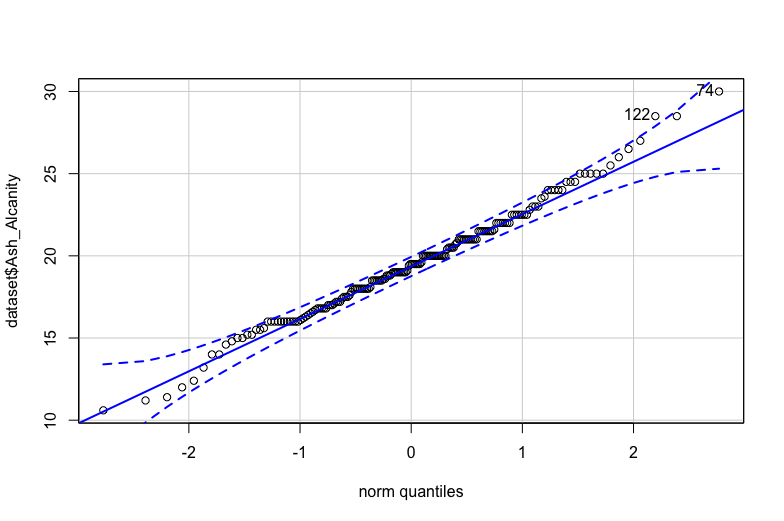
****

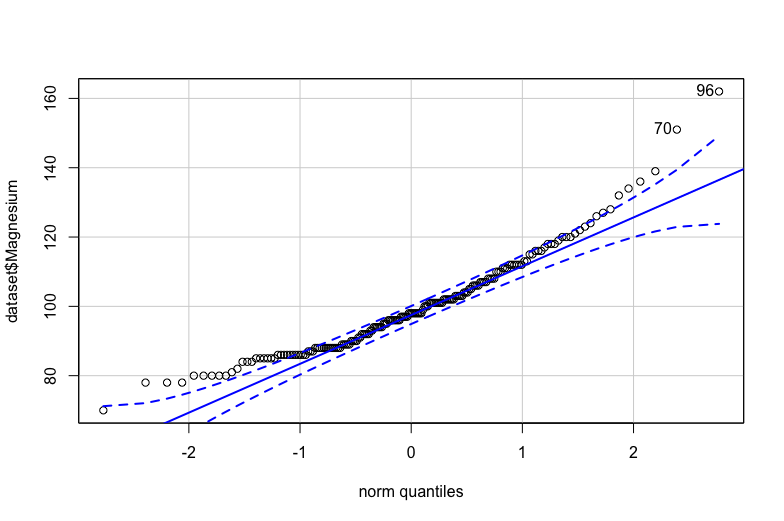
Şekil A2.1: 1 nolu bağımsız değişkene ait QQ diagramı

****

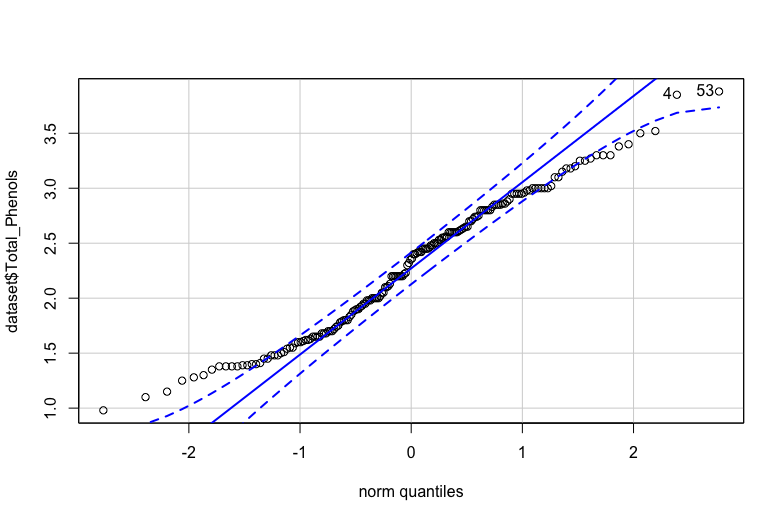
Şekil A2.2: 2 nolu bağımsız değişkene ait QQ diagramı****

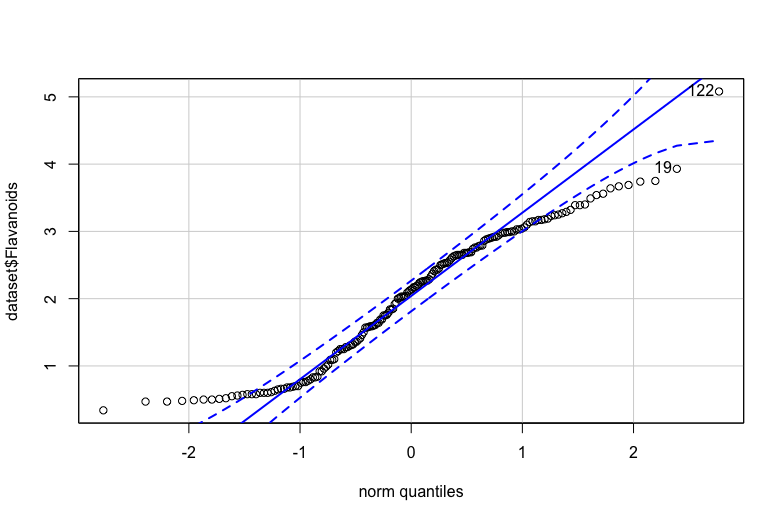
Şekil A2.3: 3 nolu bağımsız değişkene ait QQ diagramı

****

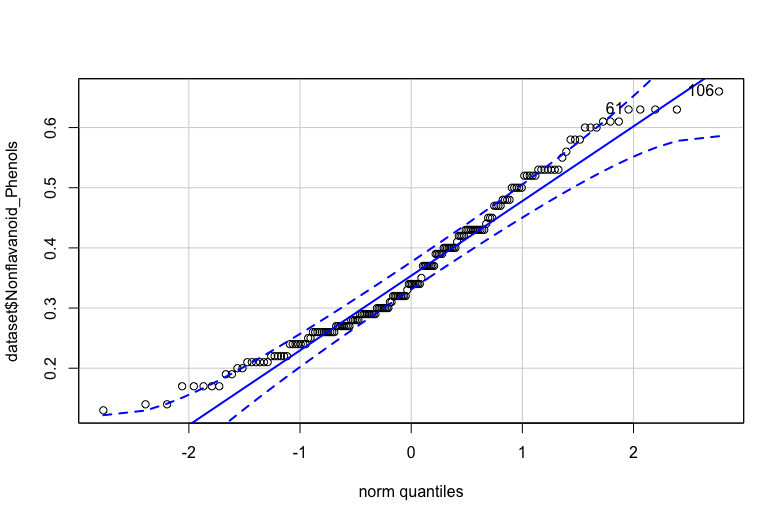
Şekil A2.4: 4 nolu bağımsız değişkene ait QQ diagramı****

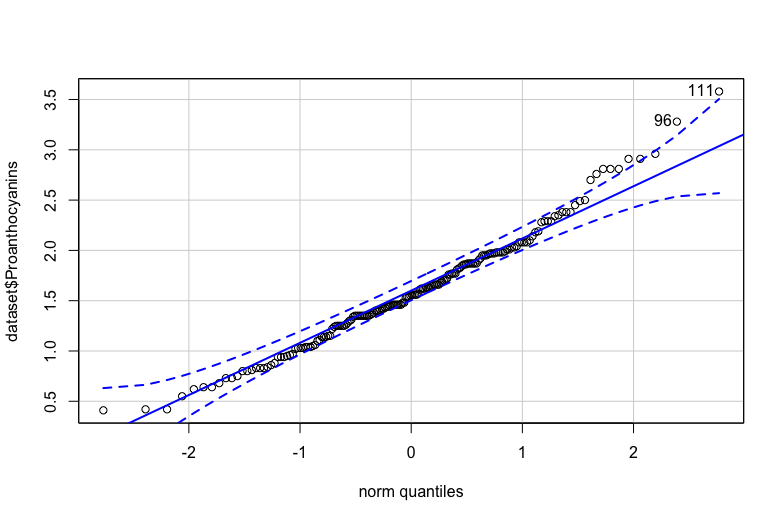
Şekil A2.5: 5 nolu bağımsız değişkene ait QQ diagramı

****

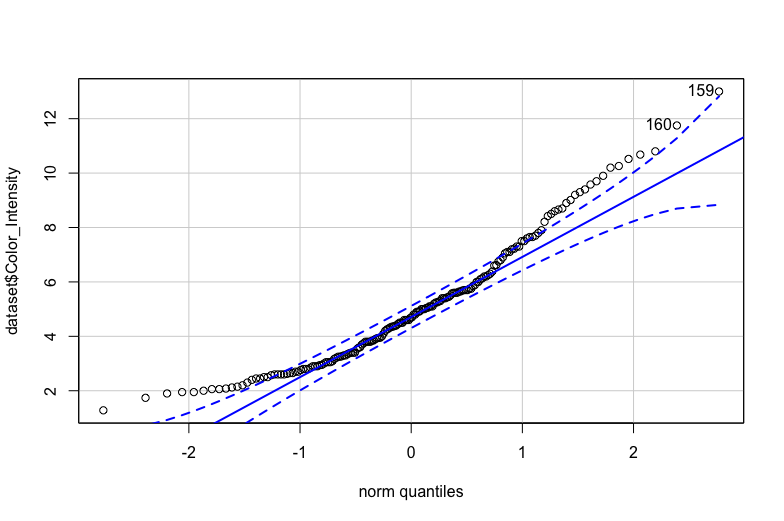
Şekil A2.6: 6 nolu bağımsız değişkene ait QQ diagramı****

Şekil A2.7: 7 nolu bağımsız değişkene ait QQ diagramı

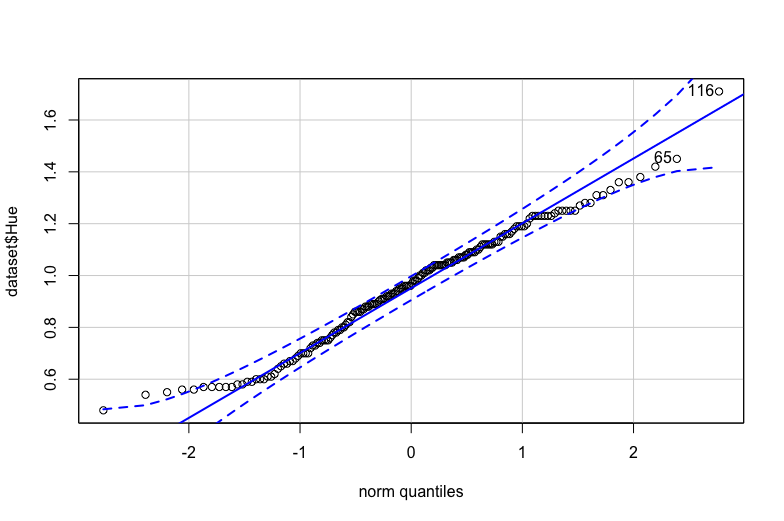
****

Şekil A2.8: 8 nolu bağımsız değişkene ait QQ diagramı****

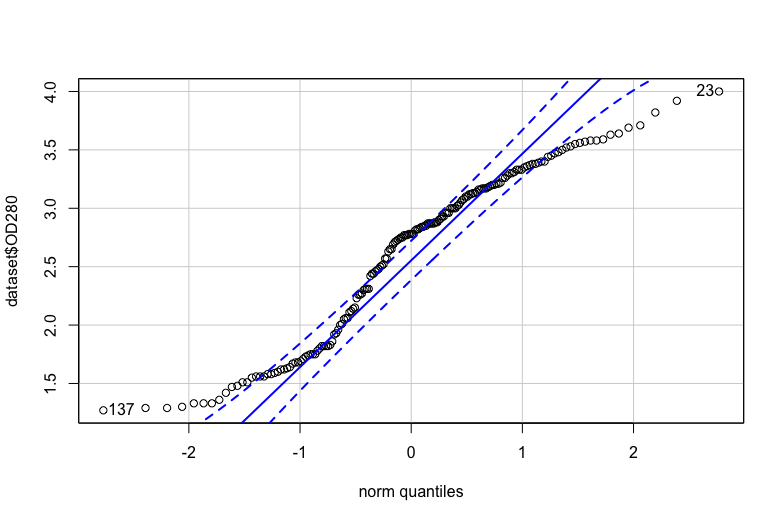
Şekil A2.9: 9 nolu bağımsız değişkene ait QQ diagramı

****

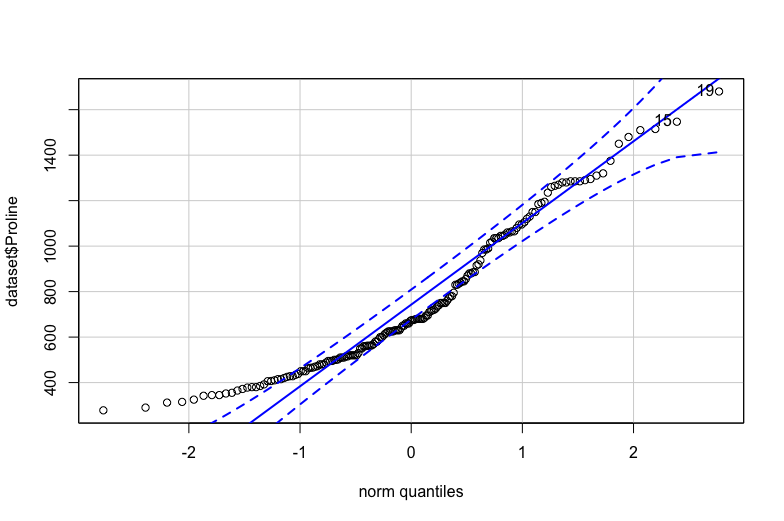
Şekil A2.10: 10 nolu bağımsız değişkene ait QQ diagramı

****

Şekil A2.11: 11 nolu bağımsız değişkene ait QQ diagramı

****

Şekil A2.12: 12 nolu bağımsız değişkene ait QQ diagramı

****

Şekil A2.13: 13 nolu bağımsız değişkene ait QQ diagramı

**Ek-B. Çalışmada kullanılan R kodları**

#########################################################

#ITU END620 - UYGULAMALI COK DEGISKENLI ISTATISTIK ANALIZ

#UYGULAMA ODEVI (2018-2019 BAHAR DONEMI)

#HAZIRLAYAN UGUR URESIN (507172109)

#########################################################

#ATTENTION! PLEASE MODIFY BELOW DIRECTORY ACCORDING TO YOUR W.DIRECTORY!

########################################################################

setwd("/Users/ugur/Desktop/R/myprojects/END620/data")

## Dataset importing

dataset = read.csv('Wine.csv')

## Initial data checks

summary(dataset[,1:14])

str(dataset)

## Visual data checks

# Combined Scatter Plot

independents <- data.frame(dataset$Alcohol,

dataset$Malic\_Acid,

dataset$Ash,

dataset$Ash\_Alcanity,

dataset$Magnesium,

dataset$Total\_Phenols,

dataset$Flavanoids,

dataset$Nonflavanoid\_Phenols,

dataset$Color\_Intensity,

dataset$Hue,

dataset$OD280,

dataset$Proline)

plot(independents,dataset$Customer\_Segment)

# Box-PLOTS

boxplot(dataset[,-c(5,13)])

boxplot(Alcohol ~ Customer\_Segment,

data=dataset,

xlab="Customer Segment",

ylab="Alcohol")

boxplot(Malic\_Acid ~ Customer\_Segment,

data=dataset,

xlab="Customer Segment",

ylab="Malic\_Acid")

boxplot(Ash ~ Customer\_Segment,

data=dataset,

xlab="Customer Segment",

ylab="Ash")

boxplot(Ash\_Alcanity ~ Customer\_Segment,

data=dataset,

xlab="Customer Segment",

ylab="Ash\_Alcanity")

boxplot(Magnesium ~ Customer\_Segment,

data=dataset,

xlab="Customer Segment",

ylab="Magnesium")

boxplot(Total\_Phenols ~ Customer\_Segment,

data=dataset,

xlab="Customer Segment",

ylab="Total\_Phenols")

boxplot(Flavanoids ~ Customer\_Segment,

data=dataset,

xlab="Customer Segment",

ylab="Flavanoids")

boxplot(Nonflavanoid\_Phenols ~ Customer\_Segment,

data=dataset,

xlab="Customer Segment",

ylab="Nonflavanoid\_Phenols")

boxplot(Proanthocyanins ~ Customer\_Segment,

data=dataset,

xlab="Customer Segment",

ylab="Proanthocyanins")

boxplot(Color\_Intensity ~ Customer\_Segment,

data=dataset,

xlab="Customer Segment",

ylab="Color\_Intensity")

boxplot(Hue ~ Customer\_Segment,

data=dataset,

xlab="Customer Segment",

ylab="Hue")

boxplot(OD280 ~ Customer\_Segment,

data=dataset,

xlab="Customer Segment",

ylab="OD280")

boxplot(Proline~ Customer\_Segment,

data=dataset,

xlab="Customer Segment",

ylab="Proline")

# QQ-Plots

library(car)

qqPlot(dataset$Alcohol)

qqPlot(dataset$Malic\_Acid)

qqPlot(dataset$Ash)

qqPlot(dataset$Ash\_Alcanity)

qqPlot(dataset$Magnesium)

qqPlot(dataset$Total\_Phenols)

qqPlot(dataset$Flavanoids)

qqPlot(dataset$Nonflavanoid\_Phenols)

qqPlot(dataset$Proanthocyanins)

qqPlot(dataset$Color\_Intensity)

qqPlot(dataset$Hue)

qqPlot(dataset$OD280)

qqPlot(dataset$Proline)

# Shapiro-Wilk test

norm01 <- shapiro.test(dataset$Alcohol)

norm02 <- shapiro.test(dataset$Malic\_Acid)

norm03 <- shapiro.test(dataset$Ash)

norm04 <- shapiro.test(dataset$Ash\_Alcanity)

norm05 <- shapiro.test(dataset$Magnesium)

norm06 <- shapiro.test(dataset$Total\_Phenols)

norm07 <- shapiro.test(dataset$Flavanoids)

norm08 <- shapiro.test(dataset$Nonflavanoid\_Phenols)

norm09 <- shapiro.test(dataset$Proanthocyanins)

norm10 <- shapiro.test(dataset$Color\_Intensity)

norm11 <- shapiro.test(dataset$Hue)

norm12 <- shapiro.test(dataset$OD280)

norm13 <- shapiro.test(dataset$Proline)

#

norm01$p.value

norm02$p.value

norm03$p.value

norm04$p.value

norm05$p.value

norm06$p.value

norm07$p.value

norm08$p.value

norm09$p.value

norm10$p.value

norm11$p.value

norm12$p.value

norm13$p.value

# Normality Check for log(Variable\_#4))

shapiro.test(log(dataset$Ash\_Alcanity))

qqPlot(log(dataset$Ash\_Alcanity))

#

# LOG-Transformation

dataset$Ash\_Alcanity <- log(dataset$Ash\_Alcanity)

shapiro.test(dataset$Ash\_Alcanity)

# Correlation Check

pairs.panels <- pairs.panels(dataset[-14])

pwc <- corr.test(dataset[-14], use="pairwise", adjust="holm")

cormat0 <- pwc$r

cormat0

# Bartletts TEST

datalist <- list(dataset$Alcohol,

dataset$Malic\_Acid,

dataset$Ash,

dataset$Ash\_Alcanity,

dataset$Magnesium,

dataset$Total\_Phenols,

dataset$Flavanoids,

dataset$Nonflavanoid\_Phenols,

dataset$Proanthocyanins,

dataset$Color\_Intensity,

dataset$Hue,

dataset$OD280,

dataset$Proline,

dataset$Customer\_Segment)

bartlett.test(datalist)

# Kaiser, Meyer, Olkin Measure of Sampling Adequacy (KMO)

KMO(as.matrix(dataset))

# PCA (PRINCIPAL COMPONENT ANALYSIS)

####################################

# Splitting the dataset into the Training set and Test set

# install.packages('caTools')

library(caTools)

set.seed(123)

split = sample.split(dataset$Customer\_Segment, SplitRatio = 0.8) #80% percent for training

training\_set = subset(dataset, split == TRUE)

test\_set = subset(dataset, split == FALSE)

# Feature Scaling

training\_set[-14] = scale(training\_set[-14])

test\_set[-14] = scale(test\_set[-14])

# Pre-Checks

data\_pca <- princomp(training\_set[-14], scores=T, cor=T, covmat=NULL)

plot(data\_pca) #PC barchart

screeplot(data\_pca, type="line", main="Scree Plot") #Scree plot

sd\_comps <- data\_pca$sdev #Standart deviations

barplot(data\_pca$sdev/sum(data\_sce1\_pca$sdev)) #PC barchart in st. dev.

cum\_facs <- cumsum(data\_pca$sdev/sum(data\_pca$sdev)) #Cumulative sum of pca ratios

fac\_cutoff = 0.33

as.numeric(which(cum\_facs>=fac\_cutoff)[1]) -> xfactors #Finds num of factors > fac\_cutoff

xfactors

# Applying PCA

# install.packages('caret')

library(caret)

# install.packages('e1071')

library(e1071)

pca = preProcess(x = training\_set[-14], method = 'pca', pcaComp = xfactors)

training\_set = predict(pca, training\_set)

training\_set = training\_set[c(2, 3, 1)]

test\_set = predict(pca, test\_set)

test\_set = test\_set[c(2, 3, 1)]

# Fitting SVM to the Training set

# install.packages('e1071')

library(e1071)

classifier = svm(formula = Customer\_Segment ~ .,

data = training\_set,

type = 'C-classification',

kernel = 'linear')

# Predicting the Test set results

y\_pred = predict(classifier, newdata = test\_set[-3])

y\_pred

# Making the Confusion Matrix

cm = table(test\_set[, 3], y\_pred)

cm

# Visualising the Training set results

library(ElemStatLearn)

set = training\_set

X1 = seq(min(set[, 1]) - 1, max(set[, 1]) + 1, by = 0.01)

X2 = seq(min(set[, 2]) - 1, max(set[, 2]) + 1, by = 0.01)

grid\_set = expand.grid(X1, X2)

colnames(grid\_set) = c('PC1', 'PC2')

y\_grid = predict(classifier, newdata = grid\_set)

plot(set[, -3],

main = 'PCA (Training set)',

xlab = 'PC1', ylab = 'PC2',

xlim = range(X1), ylim = range(X2))

contour(X1, X2, matrix(as.numeric(y\_grid), length(X1), length(X2)), add = TRUE)

points(grid\_set, pch = '.', col = ifelse(y\_grid == 2, 'deepskyblue', ifelse(y\_grid == 1, 'springgreen3', 'tomato')))

points(set, pch = 21, bg = ifelse(set[, 3] == 2, 'blue3', ifelse(set[, 3] == 1, 'green4', 'red3')))

# Visualising the Test set results

library(ElemStatLearn)

set = test\_set

X1 = seq(min(set[, 1]) - 1, max(set[, 1]) + 1, by = 0.01)

X2 = seq(min(set[, 2]) - 1, max(set[, 2]) + 1, by = 0.01)

grid\_set = expand.grid(X1, X2)

colnames(grid\_set) = c('PC1', 'PC2')

y\_grid = predict(classifier, newdata = grid\_set)

plot(set[, -3], main = 'PCA (Test set)',

xlab = 'PC1', ylab = 'PC2',

xlim = range(X1), ylim = range(X2))

contour(X1, X2, matrix(as.numeric(y\_grid), length(X1), length(X2)), add = TRUE)

points(grid\_set, pch = '.', col = ifelse(y\_grid == 2, 'deepskyblue', ifelse(y\_grid == 1, 'springgreen3', 'tomato')))

points(set, pch = 21, bg = ifelse(set[, 3] == 2, 'blue3', ifelse(set[, 3] == 1, 'green4', 'red3')))

# LDA (LINEAR DISCRIMINANT ANALYSIS)

####################################

# Splitting the dataset into the Training set and Test set

# install.packages('caTools')

#library(caTools)

#set.seed(123)

#split = sample.split(dataset$Customer\_Segment, SplitRatio = 0.8)

#training\_set = subset(dataset, split == TRUE)

#test\_set = subset(dataset, split == FALSE)

# Feature Scaling

#training\_set[-14] = scale(training\_set[-14])

#test\_set[-14] = scale(test\_set[-14])

# Applying LDA

library(MASS)

lda = lda(formula = Customer\_Segment ~ ., data = training\_set)

training\_set2 = as.data.frame(predict(lda, training\_set))

training\_set2 = training\_set2[c(5, 6, 1)]

test\_set2 = as.data.frame(predict(lda, test\_set))

test\_set2 = test\_set2[c(5, 6, 1)]

# Fitting SVM to the Training set

# install.packages('e1071')

library(e1071)

classifier2 = svm(formula = class ~ .,

data = training\_set2,

type = 'C-classification',

kernel = 'linear')

# Predicting the Test set results

y\_pred2 = predict(classifier2, newdata = test\_set2[-3])

# Making the Confusion Matrix

cm2 = table(test\_set2[, 3], y\_pred2)

cm2

# Visualising the Training set results

library(ElemStatLearn)

set = training\_set

X1 = seq(min(set[, 1]) - 1, max(set[, 1]) + 1, by = 0.01)

X2 = seq(min(set[, 2]) - 1, max(set[, 2]) + 1, by = 0.01)

grid\_set = expand.grid(X1, X2)

colnames(grid\_set) = c('x.LD1', 'x.LD2')

y\_grid = predict(classifier2, newdata = grid\_set)

plot(set[, -3],

main = 'LDA (Training set)',

xlab = 'LD1', ylab = 'LD2',

xlim = range(X1), ylim = range(X2))

contour(X1, X2, matrix(as.numeric(y\_grid), length(X1), length(X2)), add = TRUE)

points(set, pch = 21, bg = ifelse(set[, 3] == 2, 'blue3', ifelse(set[, 3] == 1, 'green4', 'red3')))

points(grid\_set, pch = '.', col = ifelse(y\_grid == 2, 'deepskyblue', ifelse(y\_grid == 1, 'springgreen3', 'tomato')))

points(set, pch = 21, bg = ifelse(set[, 3] == 2, 'blue3', ifelse(set[, 3] == 1, 'green4', 'red3')))

# Visualising the Test set results

library(ElemStatLearn)

set = test\_set

X1 = seq(min(set[, 1]) - 1, max(set[, 1]) + 1, by = 0.01)

X2 = seq(min(set[, 2]) - 1, max(set[, 2]) + 1, by = 0.01)

grid\_set = expand.grid(X1, X2)

colnames(grid\_set) = c('x.LD1', 'x.LD2')

y\_grid = predict(classifier2, newdata = grid\_set)

plot(set[, -3], main = 'LDA (Test set)',

xlab = 'LD1', ylab = 'LD2',

xlim = range(X1), ylim = range(X2))

contour(X1, X2, matrix(as.numeric(y\_grid), length(X1), length(X2)), add = TRUE)

points(set, pch = 21, bg = ifelse(set[, 3] == 2, 'blue3', ifelse(set[, 3] == 1, 'green4', 'red3')))

points(grid\_set, pch = '.', col = ifelse(y\_grid == 2, 'deepskyblue', ifelse(y\_grid == 1, 'springgreen3', 'tomato')))

points(set, pch = 21, bg = ifelse(set[, 3] == 2, 'blue3', ifelse(set[, 3] == 1, 'green4', 'red3')))

#

# END OF THE CODE#

# UGUR URESIN , uresinugur35@gmail.com#

#