

**YANIT YÜZEYİ MODELLERİNE
MARS YAKLAŞIMI**

Betül KAN
Doktora Tezi

İstatistik Anabilim Dalı

Ocak - 2011

JÜRİ VE ENSTİTÜ ONAYI

Betül KAN' ın "**Yanıt Yüzeyi Modellerine MARS Yaklaşımı**" başlıklı **İstatistik** Anabilim Dalındaki Doktora Tezi 24.12.2010 tarihinde, aşağıdaki jüri tarafından Anadolu Üniversitesi Lisansüstü Eğitim-Öğretim ve Sınav Yönetmeliğinin ilgili maddeleri uyarınca değerlendirilerek kabul edilmiştir.

	Adı Soyadı	İmza
Üye (Tez Danışmanı)	: Doç. Dr. Berna YAZICI	
Üye	: Prof. Dr. Aydın ERAR	
Üye	: Doç. Dr. İlker ERCAN	
Üye	: Doç. Dr. Meral ÇETİN	
Üye	: Yard. Doç. Dr. Atilla ASLANARGUN	

Anadolu Üniversitesi Fen Bilimleri Enstitüsü Yönetim Kurulu'nun tarih ve sayılı kararıyla onaylanmıştır.

Enstitü Müdürü

ÖZET

Doktora Tezi

YANIT YÜZEYİ MODELLERİNE MARS YAKLAŞIMI

Betül KAN

Anadolu Üniversitesi

Fen Bilimleri Enstitüsü

İstatistik Anabilim Dalı

Danışman: Doç. Dr. Berna YAZICI

2010, 129 sayfa

Bu tez çalışmasında, yanıt yüzeyi metodolojisinde ikinci dereceden tasarımlarda modelleme aşamasında MARS (Çok Değişkenli Uyarlamalı Regresyon Splaynları) yaklaşımı önerilmiştir. Bu yaklaşımın, hangi faktöriyel tasarımlara ve hangi veri yapılarına uygulanabileceği üzerinde durulmuş, modelleme aşamasının nasıl yapılacağı açıklanmıştır. Çalışmanın uygulama kısmında 3^2 deneysel düzenden gelen toprak ve toz örneklerinden yararlanarak Eskişehir merkezli yapılan saha çalışmasında ağır metal kirlilikleri regresyon ağaçlarıyla sınıflandırılmış, yanıt yüzeyi modelleri ve MARS ile modellenmeye çalışılmıştır. Sonuçlar istatistiksel testlerle ve kriterlerle değerlendirilmiştir. R Yazılımı kullanılarak tek ve iki değişkenli durum için regresyon ağacına dayalı iki ayrı program yazılmıştır. MARS yöntemi kullanılarak ağır metal kirlilik verisini modellemek için R Yazılımı ile tek ve iki değişkenli durum için iki ayrı program yazılmıştır.

Anahtar Kelimeler: Yanıt Yüzeyi Metodolojisi, Çok Değişkenli Uyarlamalı Regresyon Splaynları (MARS), Regresyon Ağacı

ABSTRACT

PhD Dissertation

MARS APPROACH TO RESPONSE SURFACE MODELS

Betül KAN

Anadolu University

Graduate School of Sciences

Statistics Program

Supervisor: Assoc. Prof. Berna YAZICI

2010, 129 pages

In this study, the approximation of the multivariate adaptive regression splines (MARS) is proposed at the stage of modeling in second order design in response surface methodology. It is mentioned that what kind of factorial designs and datasets can be used for this approximation and also how the modeling stage can be made is explained. In the application part of the study, the pollution of heavy metals coming from 3^2 design for both soil and road dust datasets which are collected from a specific area in Eskisehir are classified by the means of regression trees and is modeled by the use of response surface methodology and MARS. The results are evaluated by statistical tests and by some criteria. Two programs, based on regression trees for a univariate case and a bivariate case, are generated using R Software. Two different programs for a univariate case and a bivariate case to model the pollution of heavy metal data by MARS are generated in R Software.

Key Words : Response Surface Methodology, Multivariate Adaptive Regression Splines, Regression Tree

TEŞEKKÜR

Bu çalışmanın her adımında sonsuz özverisini, sabrını, güvenini ve her anlamda desteğini hiç esirgemeyen, gece gündüz benimle beraber çalışan sevgili danışmanım Doç.Dr. Berna YAZICI' ya akademik anlamdaki çalışmalarına yön vermesi, doktora eğitimim için beni yurtdışına gitmem konusunda cesaretlendirmesi ve bu anlamda bu tez konusunun oluşmasında en büyük katkıyı sağlaması açısından binlerce teşekkür ederim.

Yükseklisans ve doktora çalışmalarım boyunca, değerli fikirlerine son derece önem verdiğim ve manevi desteğini benden hiç esirgemeyen Bölüm Başkanımız Sayın Prof.Dr. Embiya AĞAOĞLU' na teşekkürlerimi sunarım.

Değerli hocalarım Prof.Dr. Ali Fuat YÜZER'e ve Prof.Dr. Aladdin ŞAMİLOV'a lisans, yüksekisans ve doktora eğitimim boyunca verdikleri destek ve bana duydukları güven için en içten teşekkürlerimi sunarım.

Çalışmalarım boyunca, tezin değerlendirilmesi konusundaki katkılarından dolayı Sayın jüri üyeleri hocalarıma ve programlama aşamasında beni sabırla dinleyen, fikirlerinden ilham aldığım değerli hocam Dr. Muzaffer DOĞAN' a çok teşekkür ederim.

Uygulama aşamasında kullanılan verilerle ilgili olarak yardımlarından dolayı Yard.Doç.Dr. Semra MALKOÇ' a ve haritalama konusundaki yardımlarından dolayı Yard.Doç.Dr. Metin ALTAN' a en derin teşekkürlerimi sunarım.

Her zaman benimle olan çok sevgili aileme ve dostlarıma gösterdikleri sabır ve destek için ayrıca teşekkür ederim.

Betül KAN

Aralık 2010

İÇİNDEKİLER

	<u>Sayfa</u>
ÖZET	i
ABSTRACT	ii
TEŞEKKÜR	iii
İÇİNDEKİLER	iv
ŞEKİLLER DİZİNİ	vii
ÇİZELGELER DİZİNİ	viii
1. GİRİŞ	1
2. GENEL BİLGİLER	7
2.1. Temel Kavramlar	7
2.1.1. Vektörler ve Vektör Uzayları	7
2.1.2. Doğrusal Toplam ve Germe	7
2.1.3. Doğrusal Bağımlılık ve Doğrusal Bağımsızlık	8
2.1.4. Tabanlar	8
2.1.5. Parçalı Doğrusal Regresyon Modeli	14
2.1.6. Splaynlar ve Matematiksel Gösterimi	16
2.2. Yanıt Yüzeyi Modellerinin Yapısı	19
2.2.1. 2^k Faktöriyel Tasarımları	23
2.2.2. 3^k Faktöriyel Tasarımları	27
2.2.3. Merkezi Bileşik Tasarım	28
2.3. Bir Yanıt Yüzeyinin Analizi için Temel Metotlar	30
2.3.1. En Hızlı Tırmanış Metodu (Steepest Ascent Method)	32
2.3.2. Kanonik Analiz	33
2.4. Model Seçim Kriterleri	35
2.4.1. Çapraz Geçerlilik Kriteri	35
2.4.2. Genelleştirilmiş Çapraz Geçerlilik Kriteri	36

2.4.3. C_p Kriteri	36
2.4.4. Akaike Bilgi Kriteri	37
3. MARS	38
3.1. Yinelemeli Bölüntüleme (Recursive Partitioning)	38
3.1.1. Altbölgelerin Yinelemeli Bölünmesi (Recursive Splitting)	38
3.1.2. Yinelemeli Bölüntülemenin Olumsuzlukları	43
3.2. Regresyon Splaynları	47
3.3. Uyarlamalı Regresyon Splaynları	54
3.3.1. Model Seçimi	56
3.3.2. Değişken Önemliliği	58
3.4. MARS-İleriye Dönük Algoritma.	58
3.5. MARS-Geriye Dönük Algoritma.	61
4. UYGULAMA	63
4.1. Analizde Kullanılan Verilerin Yapısı.	63
4.2. YYM ve Regresyon Ağaçları	63
4.2.1. Toprak Verisi Regresyon Ağacı Analizi	64
4.2.2. Toz Verisi Regresyon Ağacı Analizi	65
4.3. YYM ve MARS Analizi	67
4.3.1. Toprak Verisi Cd Kirliliği için MARS Analizi	69
4.3.2. Toprak Verisi Zn Kirliliği için MARS Analizi	75
4.3.3. Toprak Verisi Pb Kirliliği için MARS Analizi	80
5. SONUÇLAR	86
KAYNAKLAR	91
Ek-1 Yanıt değişkeninin, tek değişken kullanarak regresyon ağacının oluşturmasında kullanılan R kodları	96
Ek-2 Yanıt değişkeninin, iki değişken kullanarak regresyon ağacının oluşturmasında kullanılan R kodları	101

Ek-3 Cd yanıt deęiřkeninin, tek deęiřken kullanarak MARS modelinin oluřturmasında kullanılan R kodları	115
Ek-4 Cd yanıt deęiřkeninin, iki deęiřken kullanarak MARS modelinin oluřturmasında kullanılan R kodları	117
Ek-5 MARS Modelleri- ANOVA	120
Ek-6 Veri seti	123

ŞEKİLLER DİZİNİ

2.1. Doğrusal model ve uygun taban	9
2.2. $S_{u,M(\{u_0,u_K\})}$ uzayında bir fonksiyon, $M = 0$	10
2.3. $S_{u,M(\{u_0,u_K\})}$ uzayında bir fonksiyon, $M = 1$	11
2.4. Parçalı doğrusal regresyon modeli ve uygun taban	14
2.5. Çok parçalı regresyon modeli ve uygun taban	16
2.6. a-Budanmış doğru, b-B-splayn taban, c-Radyal Taban, d-Demmler-Reinsch taban	16
2.7. x_0 ile x_{n-1} arası düğümler ve splayn fonksiyonu	19
2.8. $k = 2$ ve $k = 3$ için merkezi bileşik tasarım	29
2.9. İkinci dereceden modeller için yüzeyler	31
3.1. Karesel regresyon spline fonksiyonları	49
4.1. Toprak örneği- Cd için regresyon ağacı	64
4.2. Toz örneği- Zn için regresyon ağacı	65
4.3. Regresyon ağacı ile istasyonlara göre kirlilik haritası	67
4.4. Cd için yanıt yüzeyi ve kontör grafikleri	71
4.5. Cd için ikinci ve üçüncü MARS modelleri grafikleri	74
4.6. Zn için yanıt yüzeyi ve kontör grafikleri	77
4.7. Zn için MARS grafiği	80
4.8. Pb için yanıt yüzeyi ve kontör grafikleri	83
4.9. Pb için MARS grafiği	84

ÇİZELGELER DİZİNİ

4.1. 3^2 Faktöriyel tasarımı	68
4.2. Cd yanıt yüzeyi modeli için etkilerin parçalanması	69
4.3. Cd yanıt yüzeyi için ANOVA	70
4.4. Cd için MARS modelleri	72
4.5. Cd MARS modeli için değişken önemliliği	75
4.6. Zn yanıt yüzeyi modeli için etkilerin parçalanması	76
4.7. Zn yanıt yüzeyi için ANOVA	77
4.8. Zn için MARS modelleri	78
4.9. Zn MARS modeli için değişken önemliliği	80
4.10. Pb yanıt yüzeyi modeli için etkilerin parçalanması	81
4.11. Pb yanıt yüzeyi için ANOVA	82
4.12. Pb için MARS modelleri	83
4.13. Pb MARS modeli için değişken önemliliği	85
5.1. Kirlilik modelleri için karşılaştırma	88

1 GİRİŞ

1980' lerde Avrupa'daki endüstriyel çalışmalar doğrultusunda yoğun bir şekilde kalite iyileştirmeleriyle ilgilenilmeye başlanmıştır. Özellikle deney tasarımları ve istatistiksel kalite kontrol çalışmaları hız kazanmıştır. Deney tasarımları yöntemleriyle daha yüksek güvenilirliğe sahip olan ve müşteri gereksinimlerini karşılayan ürünlerin ortaya çıktığı görülmüştür. Bu anlamda yanıt yüzeyi metodu (YYM) için, deney tasarımının özel bir dalı olarak, tarımsal, kimyasal, endüstriyel ve biyolojik araştırmalarda, süreçlerin iyileştirilmesinde, geliştirilmesinde veya optimizasyonunda kullanılan istatistiksel bir tekniktir denilebilir. İlk olarak YYM alanında çalışmaya Box ve Draper' ın 1951 yılında başladıkları bilinir. Sonraki yıllarda yanıt yüzeylerinin araştırılmasında ilgili desen seçiminde etken olabilecek farklı nedenler üzerinde çalışmışlardır [1]. Box ve Draper, YYM'da yanlış model tanımlanmış olsa bile YYM' nin etkinliği üzerinde durmuşlardır. Hill ve Hunter [2], yanıt yüzeyi metodu ile ilgili olarak kimya endüstrisindeki çalışmalara öncülük etmişlerdir. Yanıt yüzeyi çalışmalarını biyometri açısından inceleyen Mead ve Pike [3] ise, bu metodun başlangıcının 1930' larda kullanılan yanıt eğrilerine dayandığını ileri sürmüştür. Myers, Carter ve Khuri [4] fizik, mühendislik, gıda bilimindeki uygulamalarını ortaya koymuştur. Jensen ve Myers [5], tahminlemedeki doğruluğa ilişkin ölçüleri belirlemek üzere yüzeydeki noktaların tespitini yapmıştır. Hardy ve Nychka [6] çalışmalarında, ince tabakalı splaynların yanıt yüzeyleri uyumu için non-parametrik bir metot olduğundan bahsetmiştir. Burada, ince tabakalı splaynları R- splaynların özel bir durumu olarak incelemişlerdir. R-splaynlar bir yokluk uzayı (null space) polinomu ve radyal (radial) tabanlı fonksiyonların splayn uyumu olarak öne sürülmüştür. Bu makalelerinde, pürüzlülük ceza modifikasyonunu tartışmışlardır.

Hardy, ve ark. [7] çalışmalarında, ince tabakalı splaynların genişlemesi olarak bilinen R-splaynlarla yanıt yüzey uyumunda nonparametrik bir metot üzerine odaklanır. Bu araştırmalarında, 12 potansiyel bağımsız değişkenle çalışılmış özel bir uygulama yapmışlardır. Değişkenlerin fonksiyonel form-

larının, bu fonksiyonel formlar arasındaki etkileşimler ile beraber düzenlendiklerinde, uyumu yapılabilecek modellerin sayısının o andaki standartlara bağlı olan küçük veri setlerini bile etkilediğini düşünmüşlerdir. Pek çok model seçim tekniği önerilmesine rağmen, hangisinin daha üstün olduğuna dair bir fikir birliği yoktur. Bu çalışmalarında, model ve değişken seçimini yapmak için, iki çeşit simulasyon yapılmıştır. İlkinin amacı, bütün mümkün alt kümeler arasından R-splaynları kullanarak değişken seti seçiminde dört strateji arasındaki farkları test etmektir. İkincisinin amacı ise üç modelleme tekniğini karşılaştırmaktır. Bunlar, tüm altküme değişken seçimi metodolojisini kullanarak iki aşamalı bir R-splayn yaklaşımı, standart tüm altküme doğrusal regresyonları ve sınıflama regresyon ağaçlarıdır. Özetle, R-splaynları kullanarak model ve değişken seçimlerini incelemişlerdir. Öncelikle önemli açıklayıcı değişkenler R-splaynların özel bir tipi kullanılarak belirlenmiştir. Daha sonra, bu seçilmiş değişkenler uygun R-splayn modellerinin uyumu için kullanılmıştır. Ren [8], yanıt yüzeyi metodolojisini doğrusal olmayan metotlar üzerinden incelemiştir. Buna göre, çoklu doğrusal regresyona dayalı yanıt yüzeyi modellerini aromatik kimyasalların toksitlikleri için geliştirmişlerdir. Bu çalışmalarında, aromatik kimyasallarının yanıt yüzeyi modellerini doğrusal olmayan altı tane modelleme metodu kullanarak oluşturmuşlardır. Bütün modellerin çapraz geçerlilik (CV) kriteri ile geçerliliği sınanmış ve dışarıda bırakılan veri seti için tahminin doğruluğu (prediction accuracy) test edilmiştir. Sonuçlar, LOESS (locally weighted regression scatter plot smoothing), MARS (multivariate adaptive regression splines), NN (Neural Networks), PPR (projection pursuit regression) metotlarıyla belirlenmiş ve model uyumları karşılaştırılmıştır. Sonuçların benzer uygulanabilirlikleri bulunmaktadır. O' Connell ve Wolfinger [9] çalışmalarında, spatial regresyon modelleri, ikinci dereceden polinomial yanıt yüzeylerine bir tamamlayıcı alternatif olarak geliştirilmiştir. Bu modeller, tasarım uzayında bilinmeyen yanıt değişkeni tahminlerini ve tasarım değişkenlerinin etkilerini doğru ve düzgün (smooth) veri yaklaşımlarıyla sağlar. Tahminlenmiş yanıt yüzeyleri, modellerin kovaryans yapılarıyla elde edilir. Heiligers [10] çalışmasında, taban fonksiyonlarıyla parametrelendirilmiş d. dereceden poli-

nomiyal regresyon için E-optimal tasarım problemine çözüm elde edilmiştir. Buna göre bir kapalı aralıkta sonlu taşıyıcılığı olan bir olasılık ölçümü olarak tanımlanan tahmini bütün tasarımlar arasında katsayısı matrisinin sadece ve sadece en küçük özdeğerini maksimize eden bu tasarım E-optimaldir denir. Butler [11], çalışmasında G-optimal tasarımların özelliklerini düzeltme splayn-larıyla inceler. G-optimallik, parametre tahmincisinin varyansını maksimum yapan tasarımın minimumunu seçen performans kriteridir.

Tarpey ve Holcomb [12], çalışmalarında polinomiyal regresyon ile splayn modellerin uyumu arasındaki ilişkiyi incelemişlerdir.

Bir başka çalışmada Craven [13], splayn tabanlı yaklaşımla yanıt yüzeyinin çok boyutlu kesikli bir gruptan çok boyutlu sürekli bir gruba nasıl yapılandırılacabileceği üzerinde durulmuş ve bu durum çok kriterli Bayes karar problemi olarak yorumlanmıştır.

Chen [14], çok boyutlu dinamik stokastik programlamada bir deneysel tasarım ve regresyon splaynları uygulaması yapmıştır.

Crino [15], simülasyona dayalı tasarım optimizasyon problemlerinde yanıt yüzeyi metodolojisini çok değişkenli uyarlamalı splayn regresyonu ile beraber kullanmıştır.

YYM özellikle ürün geliştirilmesinde önemli bir araçtır. Üç faktörün üç farklı dozu denildiğinde, bütün kombinasyonların oluşturduğu deneme sayısı $3^3 = 27$ ' dir. Ancak bu durum bazen maliyet ve zaman açısından bütün kombinasyonların denenmesine izin vermeyebilir. Böyle bir durumda YYM kullanılarak, deneme sayısı 15'e indirilip sonuçlar test edilebilir [16]. YYM' de denemeye alınmayan faktör kombinasyonları için yanıt tahmin edecek uygun bir fonksiyonun oluşturulması ve yanıt (bağımlı) değişken fonksiyonunu maksimum veya minimum yapacak bağımsız değişken kombinasyonlarının bulunması üzerinde durulur. Yanıt yüzeyini oluşturma amacı, tüm faktör uzayında belirli özellikleri sağlayan bir bölgeyi ve bu bölgeye ait optimum noktayı tahmin etmektedir.

Çok boyutlu veri setleriyle uğraşmada yeni bir metot olarak " Çok Değişkenli Uyarlanabilir Regresyon Splaynları (MARS)" karşımıza çıkar. Buradaki yak-

laşım, bilinmeyen modelin splayn taban fonksiyonları çarpımı olarak ifade edilebilmesidir. Splayn taban fonksiyonlarının sayısı ve parametre sayısı otomatik olarak veri tarafından belirlenir. Bu süreçte temel olarak, ardışık bölüntüleme yaklaşımından faydalanılır. Ancak ardışık bölüntülemeden farklı olarak bu metot sürekli modeller ve sürekli türevler üretir. Bu yaklaşım, etkileşimleri de içerir veya toplamsal ilişkilerin modellendiği durumlarda daha esnektir.

Az sayıda gözlemin incelendiği YYM' de polinomial modellemeye ve bunların çok değişkenli durumlarına ihtiyaç duyulabilir. Bu çalışmada YYM' de MARS kullanılmasına dayalı bir yaklaşım üzerinde durulmuştur.

Bu çalışmanın ikinci bölümünde, konuya ilişkin temel kavramlara yer verilmiştir. Buna göre, ilk olarak vektörler ve vektör uzayları tanımları üzerinde durulmuş, sonrasında doğrusal bağımlılık ve doğrusal germe tanımları incelenmiştir. Ardından, tabanlar ve bunun yardımıyla oluşturulan özel regresyon modelleri tanıtılmıştır. Son olarak splaynın matematiksel terim olarak ifade sine yer verilmiştir.

Daha sonra, yanıt yüzeyi metodolojisinin yapısı ve yanıt yüzeyi modellerinin parametrelerinin tahminlenmesinde kullanılan yanıt yüzeyi tasarımlarından bazıları verilmiştir. YYM' nin birinci dereceden tasarımlarından " 2^k Faktör Tasarımı" ve " 3^k Faktör Tasarımı" ile ikinci dereceden tasarımlarından "Merkezi Bileşik Tasarım" açıklanmıştır. Sözü geçen tasarımların açıklanmasından sonra, bu modellerin analizi için kullanılan tekniklere değinilmiştir. Bu amaçla, gradyant artma ve kanonik analiz yöntemleri üzerinde durulmuştur. Bu bölümün sonunda, model seçimi için kullanılan önemli bazı kriterler incelenmiştir.

Üçüncü bölümde, yinelemeli bölüntüleme ve regresyon splaynları üzerinde durulmuştur. Ardından MARS detaylı bir şekilde incelenmiştir. MARS algoritmasının model seçim aşaması açıklanmıştır ve son kısımda MARS' ın ileriye ve geriye dönük algoritması incelenmiştir.

Bu bağlamda ilk olarak, MARS deneysel verinin koordinatlarının belirlenmesinde kullanılmıştır [17]. Bu çalışmada, " 3^k Faktöriyel Tasarım" a alternatif olarak geliştirilen "Merkezi Birleşik Tasarım" üzerinde durulmuştur.

Bu tasarım matrisi kullanılarak elde edilen MARS modelinin ürettiği taban fonksiyonları incelenmiştir. MARS' ın bu taban fonksiyonları için ürettiği düğüm noktaları, değişken önemliliği gibi ekstra bilgiler, modeli yorumlamada kullanılmaya çalışılmıştır. "2⁵ Faktöriyel Tasarım" ın faktörleri ve düzeyleri ile, MARS modelinin ürettiği taban fonksiyonları ve düğüm noktaları incelenmiştir [18]. Çalışmada regresyon ağaçları (CART) olarak da bilinen yinelemeli bölüntüleme yöntemi ile tüm faktörler ve bunların birbirleriyle etkileşimleri ortaya konmuştur. MARS modeli ile 2⁵ faktöriyel tasarıma ait etkilerin (ikili, üçlü, dörtlü ve beşli terimler dahil) olduğu birinci dereceden yanıt yüzeyi modeli üzerinde durulmuştur. Bu çalışma doğrultusunda, MARS modeli, birinci dereceden yanıt yüzeyi modelindeki faktör sayısını azalttığı için araştırmacılara bir tercih nedeni olarak önerilmiştir. Başka bir çalışmada, ince tabakalı splaynlar, kübik splaynlar ve MARS ile oluşturulan çoklu regresyon modellerinin, bazı kriterlerle karşılaştırılması R yazılımında hazırlanan bir programla yapılmış, modellerin serbestlik dereceleri üzerinde durulmuştur [19]. Ayrıca bir başka çalışmada, 3² faktöriyel deney verisi kullanılarak regresyon ağaçları oluşturulmuştur [20]. Bu çalışmada, sekiz ağır metalin ikinci dereceden yanıt yüzeyleri ve regresyon ağaçları incelenmiştir. Toz ve toprak kirlilik araştırması için tasarlanan deneysel düzenden hareketle elde edilen onaltı modele ilişkin sonuçlar yorumlanmıştır.

Tezin uygulama bölümünde, önerilen yaklaşımın kullanılabilmesi ve istatistiksel açıdan klasik yöntemle karşılaştırılabilmesi için söz konusu ağır metal verileri kullanılmıştır. Bunların sebep olduğu kirlilik araştırması için verilere, ikinci dereceden yanıt yüzeyi ve MARS modelleri uydurulmuştur. Her iki yöntemle elde edilen modellerin karşılaştırılması, modeldeki faktör sayıları, modellerin standart hataları, düzeltilmiş belirlilik sayıları (R_{adj}^2) ve geliştirilmiş çapraz geçerlilik (GCV) kriteri ile yapılmıştır. Bunlara ek olarak Cd, Zn ve Pb için oluşturulan yanıt yüzeylerinin grafikleri ve kontür haritaları verilmiştir. Bu grafikler MARS'ın ürettiği etkileşim grafikleriyle karşılaştırılmış ve sonuçlar yorumlanmıştır. MARS' ın ürettiği değişken önemlilikleri verilmiştir. Her iki yöntemle kurulan modeller ile toprak verisi kullanılarak yapılan ağır

metal kirliliği araştırmasında, MARS modellerinin daha detaylı bilgiler vermesi ve modele etkileşim terimlerini de katarak yanıt yüzeyleri analizine katkı sağlamış, gerçek yapıyı daha iyi yansıtmış, deneysel düzenden gelen verinin geleneksel yolla modellenmesi yerine temsili bir model olarak kullanılması önerilmiştir.

Sürece ilişkin algoritma, iki parçalı halde, hem tek değişken hem de iki değişken olması durumu için geliştirilmiş ve Ek-1-4 arasında verilmiştir. Ek-1-2’ de yer alan algoritmalar bir bağımlı değişkenin tek ve iki bağımsız değişkenle olan ilişkisinin yinelemeli bölüntüleme algoritmasıyla oluşturulmasına yöneliktir. Ek-3-4’ de Cd yanıt değişkeninin tek ve iki değişkenle kurulabilecek MARS modelleri oluşturulmaya çalışılmıştır. Ek-5’ te Cd, Zn ve Pb için kurulan MARS modellerine ilişkin ANOVA sonuçları verilmiştir. Ek-6’ da uygulamada kullanılan veri setleri verilmiştir.

Bu tez çalışmasında önerilen yaklaşım, az sayıda gözlemin incelendiği YYM’ de polinomiyal modellemeye ve bu çalışmaların çok değişkenli durumlarına ihtiyaç duyulabileceği varsayımıyla, YYM’ de MARS kullanılmasına ilişkindir. Buna göre:

- 3^2 faktöriyel tasarım gibi az veri noktasıyla çalışılan düzenlerde modelleme aşamasındaki yetersizlik giderilmiştir.
- Bu anlamda MARS modeli, YYM’ de temsili bir model olarak önerilmiştir.
- MARS analizine taban oluşturması açısından, R Yazılımı kullanarak yinelemeli bölüntüme algoritmasına dayalı iki program yazılmıştır.
- Uygulamada kullanılan Cd yanıt değişkeni için R Yazılımı kullanılarak, MARS modellemesi yapan iki program yazılmıştır.

2 GENEL BİLGİLER

2.1 Temel Kavramlar

Bu bölümde konuyla ilgili bilinmesi gereken kavramlara kısaca değinilmiştir.

2.1.1 Vektörler ve Vektör Uzayları

Tanım 2.1.1 *Bir vektör uzayı, üzerinde iki işlem tanımlanmış*

$$\begin{aligned} \mathbf{x} + \mathbf{y} &\in V, & \text{eğer } \mathbf{x}, \mathbf{y} &\in V \\ \mathbf{x} \in V, \alpha &\in R & \text{eğer } \alpha \mathbf{x} &\in V \end{aligned} \quad (2.1)$$

ve aşağıdaki aksiyomları sağlayan kümedir:

Aksiyom 1. Her bir $\mathbf{x}, \mathbf{y}$ vektör çifti için $\mathbf{x} + \mathbf{y} = \mathbf{y} + \mathbf{x}$ kuralı sağlanır.

Aksiyom 2. $\mathbf{x}, \mathbf{y}$ ve $\mathbf{z}$ gibi herhangi üç vektör için $(\mathbf{x} + \mathbf{y}) + \mathbf{z} = \mathbf{x} + (\mathbf{y} + \mathbf{z})$

Aksiyom 3. Sıfır vektörünü içerir ve $\mathbf{x} + \mathbf{0} = \mathbf{x}$ 'dır.

Aksiyom 4. Her bir $\mathbf{x}$ vektörüne karşılık $\mathbf{x} + (-\mathbf{x}) = \mathbf{0}$ olacak biçimde bir tek $-\mathbf{x}$ vektörü vardır.

Aksiyom 5. Her bir gerçel sayısı ve $\mathbf{x}$ ve $\mathbf{y}$ gibi her vektör çifti için $r(\mathbf{x} + \mathbf{y}) = r\mathbf{x} + r\mathbf{y}$ 'dir.

Aksiyom 6. Her bir $\mathbf{x}$ vektörü ve r ve s gibi her gerçel sayı çifti için $(r+s)\mathbf{x} = r\mathbf{x} + s\mathbf{x}$ 'dir.

Aksiyom 7. Her bir $\mathbf{x}$ vektörü ve r ve s gibi her gerçel sayı çifti için $(rs)\mathbf{x} = r(s\mathbf{x})$ 'dır.

Aksiyom 8. Her $\mathbf{x}$ vektörü için $1\mathbf{x} = \mathbf{x}$ 'dir.

Bunun dışında, vektör uzayının sıfır vektörünü içermesi gerekir. Bir vektör uzayının elemanlarına vektörler denir.

2.1.2 Doğrusal Toplam ve Germe

Tanım 2.1.2 *V vektör uzayının $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n$ vektörleri verilsin. $\alpha_1, \alpha_2, \dots, \alpha_n$ gerçel sayılar olmak üzere, $\alpha_1\mathbf{x}_1 + \dots + \alpha_n\mathbf{x}_n$ toplamı bir vektördür ve buna $\mathbf{x}_1, \dots, \mathbf{x}_n$ nin doğrusal toplamı(bileşkesi) denir.*

Tanım 2.1.3 $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n$ vektörleri sabit seçilmiş olsun. $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n$ nın bütün doğrusal toplamlarından oluşan vektörlerin $\mathbf{x} = \alpha_1 \mathbf{x}_1 + \dots + \alpha_k \mathbf{x}_n$ kümesine $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n$ vektörlerin doğrusal toplamlar kümesi veya doğrusal spanı denir.

Genellikle $span(\mathbf{x})$ ile gösterilir [21].

2.1.3 Doğrusal Bağımlılık ve Doğrusal Bağımsızlık

Tanım 2.1.4 E , bir vektörler kümesi, $\mathbf{x}_1, \dots, \mathbf{x}_k$ de E de farklı vektörler ve $\alpha_1, \dots, \alpha_k$ hepsi aynı anda sıfır olmayan sayılar olmak üzere eğer

$$\alpha_1 \mathbf{x}_1 + \dots + \alpha_k \mathbf{x}_k = \mathbf{0} \quad (2.2)$$

koşulu sağlanıyorsa, E kümesine doğrusal bağımlıdır denir.

E kümesine $\mathbf{x}_1, \dots, \mathbf{x}_k$ doğrusal bağımlı vektörlerinden oluşan bir kümedir denir. Bir E kümesi doğrusal bağımlı değilse doğrusal bağımsızdır. Eğer bir E vektör kümesi sıfır vektörünü içeriyorsa doğrusal bağımlıdır [21].

2.1.4 Tabanlar

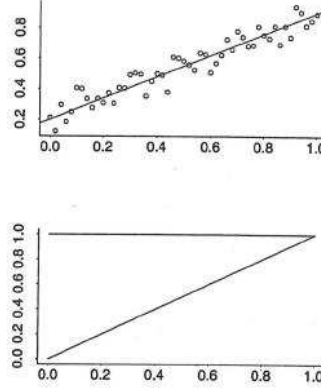
Tanım 2.1.5 V , bir vektör uzayı ve E de bu uzayda bir vektör kümesi olsun. Eğer E doğrusal bağımsız ise ve V 'yi geriyorsa, E 'ye V 'nin bir tabanı denir.

Yukarıdaki tanım dahilinde bir basit doğrusal regresyon modelini göz önüne alındığında,

$$y_i = \beta_0 + \beta_1 \mathbf{x}_i + e_i \quad (2.3)$$

denklemini bir doğruyu temsil eder. Bu model için uygun taban $\mathbf{1}$ ve $\mathbf{x}$ fonksiyonları ile temsil edilebilir. Bir başka deyişle $\beta_0 + \beta_1 \mathbf{x}$ ifadesi $\mathbf{1}$ ve $\mathbf{x}$ taban fonksiyonlarının doğrusal bir kombinasyonudur.

Buna ilişkin grafik Şekil 2.1 ile verilmiştir.



Şekil 2.1: Doğrusal model ve uygun taban

Eşitlik 2.3’de eşitliğin sağ tarafı bu fonksiyonların bir doğrusal kombinasyonudur. Regresyon için kurulacak $\mathbf{X}$ matrisinin sütunları, uygun olan taban fonksiyonları $\mathbf{1}, \mathbf{x}$ kullanılarak (2.4) ile elde edilir.

$$\mathbf{X} = \begin{bmatrix} 1 & x_1 \\ 1 & x_2 \\ \dots & \dots \\ 1 & x_n \end{bmatrix} \quad (2.4)$$

Benzer şekilde, $\{\mathbf{1}, \mathbf{x}, \mathbf{x}^2, \dots, \mathbf{x}^n\}$ kümesi çoklu polinomiyal regresyon için alınabilecek bir tabandır. Buna ilişkin $\mathbf{X}$ matrisi (2.5) ile verilir.

$$\mathbf{X} = \begin{bmatrix} 1 & x_1 & \dots & x_1^n \\ 1 & x_2 & \dots & x_2^n \\ \dots & \dots & \dots & \dots \\ 1 & x_n & \dots & x_n^n \end{bmatrix} \quad (2.5)$$

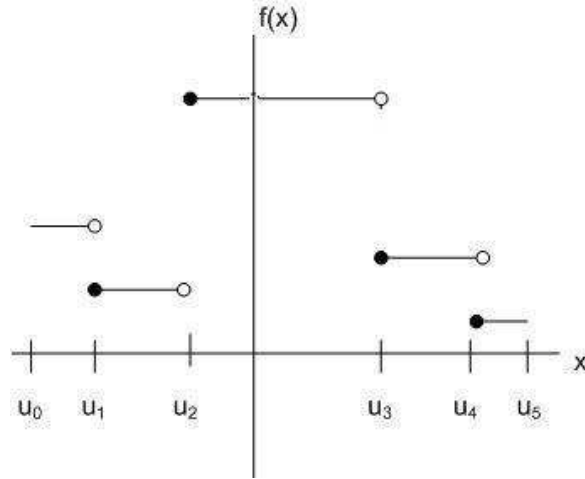
Burada, $\beta_0, \beta_1, \dots, \beta_n$ katsayıları $\mathbf{X}$ model matrisinde sırasıyla $\{\mathbf{1}, \mathbf{x}, \mathbf{x}^2, \dots, \mathbf{x}^n\}$ taban fonksiyonları ile temsil edilirler. Çoklu model, basit doğrusal modelin doğrusal olmayan (değişkenlerde) duruma genişletilmesidir [21].

Tanım 2.1.6 $M \in N_0$ ve $u_0 < u_1 < \dots < u_K$ olsun. $u = \{u_j\}_{j=0,\dots,K}$. Splayn uzayı $S_{u,M}([u_0, u_K])$ ' yi aşağıdaki gibi ifade ederiz:

$$S_{u,M}([u_0, u_K]) = \{f : [u_0, u_K] \rightarrow \mathbb{R}\}$$

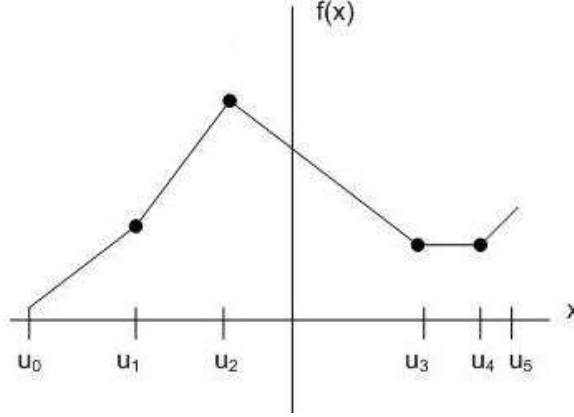
olmak üzere $p_0, \dots, p_{K-1}$, M . ya da daha küçük dereceden polinomlar olsunlar; öyle ki $f(x) = p_i(x)$, $x \in [u_i, u_{i+1})$ ($i = 0, \dots, K-1$). Eğer $M-1 \geq 0$ ise o zaman f , $[u_0, u_K]$ üzerinde $M-1$ kere sürekli diferensiyellenebilir. Burada u düğüm vektörü ve M de, $S_{u,M}([u_0, u_K])$ splayn uzayının derecesidir.

$S_{u,0}([u_0, u_K])$ parçalı sabit fonksiyonlardır. Buna ilişkin grafik Şekil 2.2 ile verilmiştir. Burada her bir $[u_i, u_{i+1})$ aralığında çizilmiş sabit fonksiyonlar gösterilmiştir.



Şekil 2.2: $S_{u,M}([u_0, u_K])$ uzayında bir fonksiyon, $M = 0$

$S_{u,1}([u_0, u_K])$ içindeki fonksiyonlar, $[u_0, u_K]$ üzerinde parçalı ve sürekli. Buna ilişkin grafik Şekil 2.3 ile verilmiştir.



Şekil 2.3: $S_{u,M}([u_0, u_K])$ uzayında bir fonksiyon, $M = 1$

Açıkça görülüyor ki, $S_{u,M}([u_0, u_K])$ bir doğrusal vektör uzayıdır. Bir sonraki teoremden bu doğrusal vektör uzayının bir tabanından bahsedilecektir.

Teorem 2.1.7 $M \in \mathbb{N}_0$ ve $u_0 < u_1 < \dots < u_K$ olsun. Fonksiyonlar kümesi,

$$\{1, x, \dots, x^M\} \cup \{(x - u_j)_+^M : j = 1, \dots, K - 1\} \quad (2.6)$$

$S_{u,M}([u_0, u_K])$ 'nin bir tabanıdır. Her $f \in S_{u,M}([u_0, u_K])$ için $a_0, a_1, \dots, a_M, b_1, \dots, b_{K-1} \in \mathbb{R}$ vardır; öyle ki,

$$f(x) = \sum_{i=0}^M a_i x^i + \sum_{j=1}^{K-1} b_j (x - u_j)_+^M \quad (2.7)$$

$x \in [u_0, u_K]$ 'de, doğrusal toplamlar şeklinde yazılabilir.

Kanıt

Görülüyor ki, $S_{u,M}([u_0, u_K])$ vektör uzayının boyutu $(M+1)+(K-1)=M+K$ 'dır. Eşitlik 2.6' deki fonksiyonların $S_{u,M}([u_0, u_K])$ tarafından kapsandığını gösterelim. $k = 0, \dots, M - 1$ için,

$$\frac{\partial^k}{\partial x^k} \Big|_{x=u_j} = M \cdot (M - 1) \cdots (M - k + 1) ((x - u_j)^{M-k}) \Big|_{x=u_j} = 0 \quad (2.8)$$

(j=1,...,K-1)

yazılabilir. Bir sonraki adımda, (2.6)'daki fonksiyonların doğrusal bağımsız

olduklarını gösterelim. Varsayalım ki, $a_0, a_1, \dots, b_1, \dots, b_{K-1} \in \mathfrak{R}$ keyfi olsun ve $(x \in [u_0, u_K])$ için,

$$\sum_{i=0}^M a_i x^i + \sum_{j=1}^{K-1} (x - u_j)_+^M = 0 \quad (2.9)$$

olur.

$x \in [u_0, u_1]$ için $(x - u_j)_+^M = 0, (j = 1, \dots, K - 1)$ olur. Dolayısıyla (2.9) ifadesinden $(x \in [u_0, u_1])$ için

$$\sum_{i=0}^M a_i x^i = 0 \quad (2.10)$$

elde edilir. Çünkü $1, x, x^2, \dots, x^M$ ifadeleri, $a_0 = a_1 = \dots = a_M = 0$ olacak şekilde en az $M+1$ tane farklı noktayı içeren her küme de doğrusal bağımsızdır. Buradan, (2.9) düşünüldüğünde $x \in [u_0, u_K]$ için,

$$\sum_{j=1}^{K-1} (x - u_j)_+^M = 0 \quad (2.11)$$

yazılır. Eğer $x = \frac{u_j + u_{j+1}}{2}$, $(j = 1, \dots, K - 1)$ alınır ve (2.11)'de yerine yazılırsa, $b_j = 0$ elde edilir $(j = 1, \dots, K - 1)$. Çünkü

$$\left(\frac{u_j + u_{j+1}}{2} - u_K\right)_+^M = 0 \quad (2.12)$$

dır $(k > j)$. O halde gösterilmesi gereken $f \in S_{u,M}([u_0, u_K])$ için, (2.7)'de $a_0, a_1, \dots, b_1, \dots, b_{K-1} \in \mathfrak{R}$ olduğudur. Burdan matematiksel induksiyon ile, her bir $k \in \{0, \dots, K - 1\}$ için, $x \in [u_0, u_{K+1})$ aralığında,

$$f(x) = \sum_{i=0}^M a_i x^i + \sum_{j=1}^{K-1} b_j (x - u_j)_+^M \quad (2.13)$$

olacak şekilde $a_0, a_1, \dots, b_1, \dots, b_{K-1} \in \mathfrak{R}$ olduğunu göstermek gerekir. Bu durum, $k = 0$ için sağlanır; çünkü f , $[u_0, u_1]$ aralığında M . ya da daha küçük dereceden bir polinomdur. Varsayalım ki, (2.14) ifadesi, $k < K - 1$ için sağlansın. O zaman, g gibi bir fonksiyon tanımlansın:

$$g(x) = f(x) - \sum_{i=0}^M a_i x^i - \sum_{j=1}^{K-1} b_j (x - u_j)_+^M \quad (2.14)$$

ve $x \in [u_0, u_{k+1})$ için,

$$g(x) = 0 \quad (2.15)$$

sağlanır. $f \in S_{u,M}([u_0, u_K])$ olduğundan, bu fonksiyon, u_{k+1} noktasında $M-1$ kere sürekli diferensiyellenebilir. Bundan dolayı, $i = 0, \dots, M-1$ için,

$$\frac{\partial^i g(u_{k+1})}{\partial x^i} = 0 \quad (2.16)$$

dır. Çünkü $f \in S_{u,M}([u_0, u_K])$ olduğundan, g , M . ya da daha küçük dereceden bir polinomdur. Dolayısıyla, $x \in [u_{k+1}, u_{k+2})$ için öyle $c_0, \dots, c_M \in \Re$ sayıları bulunur ki,

$$g(x) = \sum_{i=0}^M c_i (x - u_{k+1})^i$$

olur. Çünkü, $\frac{\partial^j g(u_{k+1})}{\partial x^j} = \sum_{i=j}^M c_i \cdot i \cdot (i-1) \cdots (i-j+1) \cdot (x - u_{k+1})^{i-j} \big|_{x=u_{k+1}} = j! c_j$, dir. Eşitlik 2.16 ifadesinden $c_0 = \dots = c_{M-1} = 0$ olur. Burada $x \in [u_{k+1}, u_{k+2})$ olduğu için,

$$g(x) - c_M (x - u_{k+1})^M = 0$$

olur. Bu durum (2.22) ile beraber kullanıldığında, $k+1$ için (2.14) ifadesinde $b_{k+1} = C_M$ olarak bulunur [22].

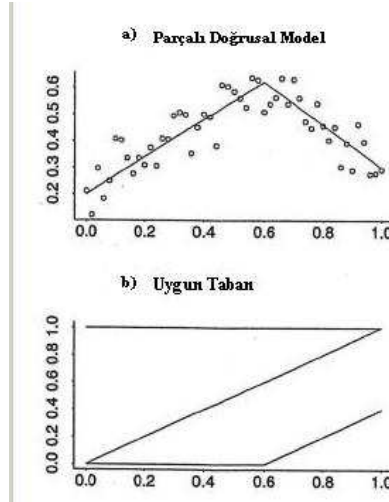
2.1.5 Parçalı Doğrusal Regresyon Modeli

Bu kısım MARS' ın alt yapısında kullanılan taban fonksiyonlarının regresyon analizinde nasıl kullanılacağını daha iyi kavramak açısından verilmiştir. Bu bölümde özel taban fonksiyonları kullanıldığında tasarım matrisinin nasıl düzenlendiğine ve ilgili taban fonksiyonların nasıl çizildiğine ilişkin bilgi verilmiştir.

Parçalı doğrusal regresyon modeli (Broken stick regression model), farklı eğimli iki doğrunun bir t düğüm noktasında birleşmesiyle oluşur. Bu durumda oluşturulabilecek taban fonksiyonları kümesi, t noktasının solunda sıfır değerini alan bir fonksiyon ancak sağ tarafında pozitif eğimli başka bir fonksiyon olabilir. Bu taban fonksiyonu oluşturmanın matematiksel ifadesi aşağıda verilmiştir.

$$(x - t)_+$$

Bu ifade ile verilen fonksiyon, ayrıca, "budanmış bir doğru" veya "a truncated line" olarak da bilinir. Buna ilişkin grafik seçilmiş t değerleri için Şekil 2.4b ile verilmiştir.



Şekil 2.4: Parçalı doğrusal regresyon modeli ve uygun taban

Burada her u için u_+ , eğer u pozitifse u , değilse 0 değerini alır. Bu duruma ilişkin parçalı doğrusal regresyon modeli (2.17) ile ilgili grafik Şekil 2.4a ile verilmiştir.

$$y_i = \beta_0 + \beta_1 x_i + \beta_{11}(x_i - t)_+ e_i \quad (2.17)$$

Eşitlik 2.17 ile verilen modele ilişkin $\mathbf{X}$ matrisi (2.18) ile verilmiştir.

$$\mathbf{X} = \begin{bmatrix} 1 & x_1 & (x_1 - t)_+ \\ 1 & x_2 & (x_2 - t)_+ \\ \dots & \dots & \dots \\ 1 & x_n & (x_n - t)_+ \end{bmatrix} \quad (2.18)$$

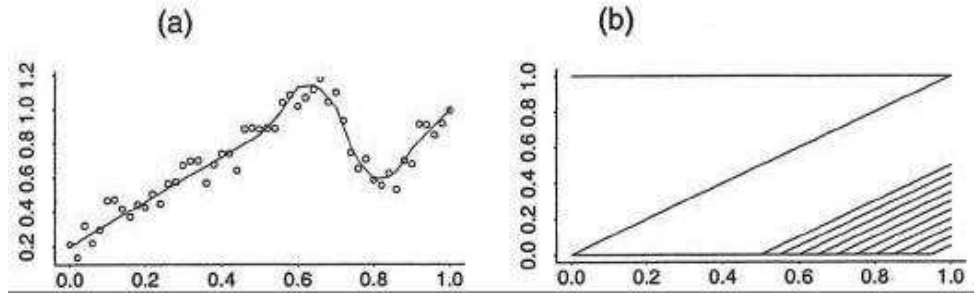
Buraya kadar açıklanan model yapıları daha karışık olarak düşünülebilir.

Buna ilişkin regresyon modeli için kurulacak $\mathbf{X}$ matrisi (2.19) ile verilir:

$$\mathbf{X} = \begin{bmatrix} 1 & x_1 & (x_1 - t_1)_+ & (x_1 - t_2)_+ & \dots & (x_1 - t_n)_+ \\ 1 & x_2 & (x_2 - t_1)_+ & (x_2 - t_2)_+ & \dots & (x_2 - t_n)_+ \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 1 & x_n & (x_n - t_1)_+ & (x_n - t_2)_+ & \dots & (x_n - t_n)_+ \end{bmatrix} \quad (2.19)$$

Bu regresyon modeli, çok parçalı regresyon modeli (Whip regression model) olarak bilinir [21]. $(x - t)_+$ yapısındaki fonksiyonların tabana eklenmesi, $\mathbf{X}$ matrisine $(x_i - t)_+$ yapısındaki sütun vektörü değerleri olarak yansır. Eşitlik 2.19' da seçilen t_i , $i = 1, \dots, n$ değerleri ile regresyon fonksiyonunun tahmininde kullanılabilir. Uygun $(x - t)_+$ fonksiyonuna ilişkin t değeri genellikle "düğüm" olarak bilinir. "Düğüm" kelimesi burada iki doğrusal fonksiyonu birleştirmesi anlamında kullanılmıştır. İlgili modelin grafiği ve uygun taban grafiği Şekil 2.5a ve b ile verilmiştir.

Doğrusal splayn taban fonksiyonu $(x - t)_+$ olarak da bilinir. Bu tür fonksiyonların bir kümesi "Doğrusal Splayn Taban" olarak adlandırılır. Bu tür doğrusal splayn taban fonksiyonlarının, $1, x, (x - t_1)_+, \dots, (x - t_K)_+$ herhangi bir doğrusal kombinasyonu, $t_1, \dots, t_K$ düğümleri ile bir parçalı doğrusal fonksiyondur.

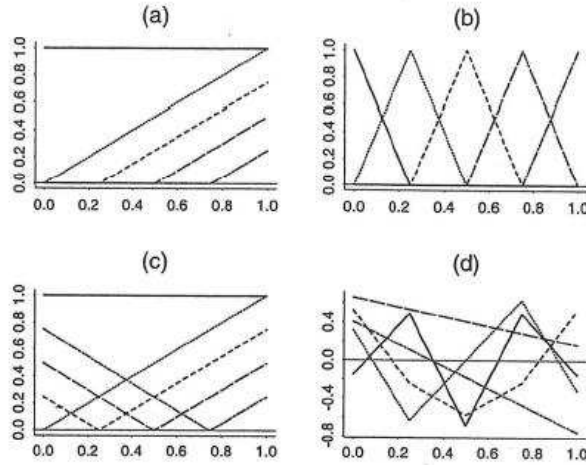


Şekil 2.5: Çok parçalı regresyon modeli ve uygun taban

Bu şekilde oluşturulan fonksiyonun splayn olarak adlandırıldığı daha önceki bölümde ifade edilmiştir ve ilgili splayn modeli (2.20) ile verilmiştir:

$$f(x) = \beta_0 + \beta_1 x + \sum_{k=1}^K b_k (x - t_k)_+ \quad (2.20)$$

Sıkça kullanılan tabanların grafikleri Şekil 2.6 ile gösterilmiştir [21].



Şekil 2.6: a-Budanmış doğru, b-B-splayn taban, c-Radyal Taban, d-Demmler-Reinsch taban

2.1.6 Splaynlar ve Matematiksel Gösterimi

Splaynlar genellikle nümerik analizde, interpolasyonda, düzeltme (smoothing) veya geometrik tasarımlarda kullanılan fonksiyonlardır. Bir grup veriye polinomiyal bir eğri uydurma veya pek çok parçadan oluşan polinomların bileşiminden oluşan temel bir fonksiyon olarak tanımlanabilir. Amacı, tanımlanan

bir aralıkta, bağımsız değişkenlerle yanıt değişkeni arasındaki ilişkiyi, gözlem değerlerinden yararlanarak, alt aralıklar ve her alt aralıkta tanımlanan polinomlar yardımıyla modellemek ve pürüzsüz bir eğri elde etmektir.

Tanım 2.1.8 $[a, b]$ üzerinde tanımlı, k . veya büyük dereceden, $k > 0$, λ_i artan reel sayı dizisine sahip, $i = 0, 1, 2, \dots, g + 1$, ($\lambda_0 = a, \lambda_{g+1} = b$), ve aşağıdaki koşulları sağlayan $s(x)$ fonksiyonuna splayn denir.

(i) $\forall s(x), [\lambda_i, \lambda_{i+1}]$ aralığında derecesi en çok k veya k' ya eşit olan bir polinomdur.

$s_{[\lambda_i, \lambda_{i+1}]} \in P_k \quad i = 0, 1, 2, \dots, g ; \quad P_k: k$ 'inci dereceden polinomlar ailesi

(ii) $s(x)$ fonksiyonu ve onun $k - 1$ 'inci mertebeye kadar olan türevleri $[a, b]$ üzerinde süreklidir.

$s(x) \in C^{k-1}[a, b]$

Özellik 1 $[a, b]$ üzerindeki alt aralıklarda derecesi k ve/veya daha küçük her polinom, $[a, b]$ üzerinde bir splayn fonksiyonudur. Genellikle, k 'inci dereceden splayn, $[a, b]$ aralığında farklı polinomlarla verilir.

Özellik 2 k' inci dereceden splayn fonksiyonun k' inci dereceden türevi basamak fonksiyonudur. $k - 1$ ve daha düşük dereceden türevleri ise süreklidir. Bu tanımlamaya göre k' inci dereceden bir polinom k' inci dereceden bir splaynın özel bir durumudur.

Özellik 3 $\lambda_0, \lambda_1, \dots, \lambda_{g+1}$ artan ve sırayla dizilmiş noktalaradır.

$s(x)$, k' inci dereceden bir splayn fonksiyonu olsun. λ_i düğüm noktalarında $s(x)$ fonksiyonunun $k - 1$ 'inci dereceden türevi süreklidir. $P_{k,i}, [\lambda_i, \lambda_{i+1}]$ aralığında k' inci dereceden polinom olsun. Öyleyse (2.21) elde edilir.

$$P_{k,i}^r(\lambda_i) = P_{k,i+1}^r(\lambda_i) \quad (2.21)$$

Buradan, k' inci dereceden $g(x)$ polinomu için (2.22) yazılır:

$$g(x) = P_{k,i+1}(x) - P_{k,i}(x) \quad (2.22)$$

$g(x)$ ' in $k - 1$ inci dereceden türevlerinin $x = \lambda_i$ de kökleri olduğundan (2.23) elde edilir:

$$g(x) = P_{k,i+1}(x) - P_{k,i}(x) = c_i(x - \lambda_i)^k \quad (2.23)$$

Burada, c_i sabittir. Bu denklem yeniden, (2.24) şeklinde yazılabilir:

$$P_{k,i+1}(x) = P_{k,i}(x) + c_i(x - \lambda_i)^k \quad (2.24)$$

k . dereceden $P_{k,0}(x)$ polinomu $(\lambda_{-1}, \lambda_0)$ aralığında verildiği zaman $P_{k,i}(x)$, (λ_0, λ_1) aralığında (2.25) bağıntısını gerçekler:

$$P_{k,1}(x) = P_{k,0}(x) + c_0(x - \lambda_0)^k \quad (2.25)$$

Buradan, $P_{k,2}(x)$ polinomu, (λ_1, λ_2) aralığında (2.26)şeklini alır:

$$P_{k,2}(x) = P_{k,0}(x) + c_0(x - \lambda_0)^k + c_1(x - \lambda_1)^k \quad (2.26)$$

Benzer olarak $P_{k,i+1}(x)$, $(\lambda_i, \lambda_{i+1})$ aralığında (2.27)şeklinde yazılır:

$$P_{k,i+1}(x) = P_{k,0}(x) + \sum_{j=0}^i c_j(x - \lambda_j)^k \quad \lambda_i \leq x \leq \lambda_{i+1} \quad (2.27)$$

$P_{k,i+1}(x)$ ' i $s(x)$ splayn gösteriminde yeniden yazarsak (2.28) olarak elde edilir:

$$s(x) = P_{k,0}(x) + \sum_{j=0}^i c_j(x - \lambda_j)^k \quad \lambda_i \leq x \leq \lambda_{i+1} \quad (2.28)$$

Parçalı kuvvet fonksiyonu kullanarak (2.29) ile, x 'in tüm noktaları üzerinde (2.29) ifadesi bulunur:

$$s(x) = P_{k,0}(x) + \sum_{i=0}^n c_i(x - \lambda_i)_+^k \quad (2.29)$$

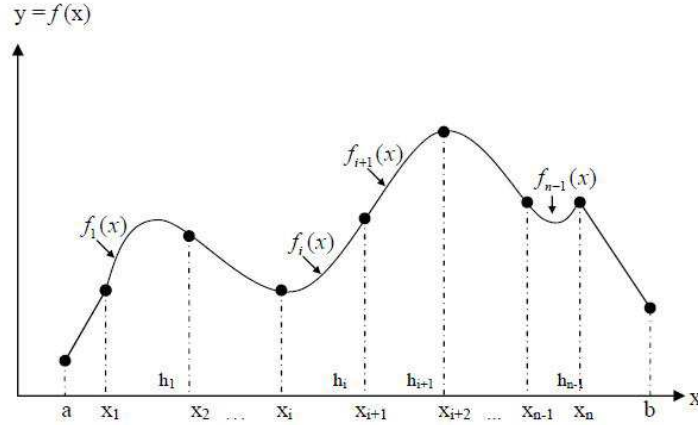
Eşitlik 2.29' un c_i katsayıları bilinmemektedir. Dolayısıyla bulunması gerekmektedir. k ' ıncı dereceden bir splayn fonksiyonunun k ' ıncı dereceden türevi bir basamak fonksiyon olduğundan, $s^k(x)$ düğüm noktalarında süreksiz olur. λ_i düğüm noktalarının iki tarafında splayn fonksiyonlarının k ' ıncı dereceden türevlerinin farklarına bakalım. Eşitlik 2.28' yi kullanarak ve türev alarak (2.30) olarak elde edilir:

$$s^k(\lambda_i + 0) - s^k(\lambda_i - 0) = k!c_i \quad (2.30)$$

Buradan c_i leri çözersek (2.31) elde edilir [23].

$$c_i = \frac{1}{k!}[s^k(\lambda_i + 0) - s^k(\lambda_i - 0)] \quad (2.31)$$

Şekil 2.7' de splayn fonksiyonu ve düğümler arası ilişkileri verilmiştir.



Şekil 2.7: x_0 ile x_{n-1} arası düğümler ve splayn fonksiyonu

2.2 Yanıt Yüzeyi Modellerinin Yapısı

Bir süreçte bağımsız değişkenlerle yanıt değişkeni arasındaki ilişki, yanıtı en büyükleyen veya en küçükleyen bağımsız değişkenlerin düzeylerine bakarak bulunabilir. Yanıt değişkeni Y ile k tane bağımsız değişken arasındaki ilişki (2.32) ifadesiyle verilir:

$$Y = f(\xi_1, \xi_2, \dots, \xi_k) + \epsilon \quad (2.32)$$

Burada, $\xi_1, \xi_2, \dots, \xi_k$ bağımsız değişkenler ve ϵ deneysel hata olarak formülde yer almaktadır. Gerçek yanıt fonksiyonu f bilinmediği durumlarda bu yapının, bir polinomial model vasıtasıyla bağımsız değişkenlerin oluşturduğu bir bölgenin içerisinde olduğu kabul edilir.

Yanıt yüzeyi modelleri için, y yanıt(bağımlı) değişkeninin, bağımsız değişkenlerin bir fonksiyonu olduğu ve deneysel hatanın sıfır ortalamalı ve σ^2 sabit varyanslı olduğu varsayılmaktadır. Bir başka varsayım, modelde bulunan bağımsız değişkenlerin arasında ilişki olmamasıdır. Yanıt değişkeninin ortalaması (2.33) ile verilmiştir:

$$E[Y] = f(\xi_1, \xi_2, \dots, \xi_k) \quad (2.33)$$

Buna ortalama tepki fonksiyonu da denir. X ' ler için deney tekrarlandıkça ölçülen tepki ölçüm hatalarından dolayı değişim gösterecektir. Yanıt yüzeyi modelleri iki veya daha fazla değişkenin bağımlı değişken üzerindeki etkilerinin

ortaya konulduğu çalışmalardır. YYM deneysel hatadan etkilenen tepki fonksiyonunun özelliklerini inceler. Birinci dereceden (doğrusal) bir model (2.34) ile,

$$Y = \beta_0 + \beta_1\xi_1 + \beta_2\xi_2 + \dots + \beta_k\xi_k + \epsilon \quad (2.34)$$

ve ikinci dereceden bir model (2.35) ile yazılabilir:

$$Y = \beta_0 + \sum_{i=1}^k \beta_i\xi_i + \sum_{i=1}^k \beta_{ii}\xi_i^2 + \sum_{i<j}^n \beta_{ij}\xi_i\xi_j + \epsilon \quad (2.35)$$

İstatistiksel analiz aşamasına geçilmeden önce bağımsız değişkenlerin matematiksel işlemleri kolaylaştırmak için "kodlanmış değişkenler" olan $x_1, x_2, \dots, x_k$ 'ya dönüştürülmesi gerekebilir. Buna göre (2.32) ile verilen ilişki (2.36) ile gösterilebilir:

$$E[Y] = f(x_1, x_2, \dots, x_k) \quad (2.36)$$

Bu eşitlik bağımsız değişken sayısı $k=1$ olduğunda bir eğri, $k>1$ olduğunda ise bir hiper yüzeyi temsil eder. Kodlanmış değişkenler (2.37) ifadesiyle belirtilmiştir:

$$\bar{\xi}_i = \frac{1}{N} \sum_{u=1}^N \xi_{ui} \quad (2.37)$$

$$x_{ui} = \frac{\xi_{ui} - \bar{\xi}_i}{S_i}$$

$$S_i = \left\{ \sum_{u=1}^N \frac{(\xi_{ui} - \bar{\xi}_i)^2}{N} \right\}^{1/2}$$

Buradaki S_i tasarım noktalarının yayılışının bir ölçüsüdür. Bu eşitliklerde, parametreler deneme kombinasyonlarıyla tahmin edilmektedir. Parametrelerin ifade ettikleri aşağıda belirtilmiştir:

- β_0 : Bağımsız değişkenlerin sıfır olduğu durumda yanıt değişkeninin aldığı değer(intercept).
- β_i : i . bağımsız değişkenin doğrusal etkisi.
- β_{ii} : i . bağımsız değişkenin karesel etkisi.
- β_{ij} : i . ve j . bağımsız değişkenlerinin etkileşim etkisi ($i < j$).

dir. Uygulamalarda çoğu zaman, gerçek yanıt değişkeninin temsil edildiği bu eşitliğin yapısı bilinmediğinden, çeşitli yaklaşımlarla bu fonksiyon temsil edilmeye çalışılır. Başka bir deyişle, bu eşitliğin hangi derecedeki faktör kombinasyonlarının modele dahil edileceği bilinemez. Ancak doğrusal veya karesel yaklaşımlarla gerçek fonksiyonun tahmin edilmesi gerekir. Bunun için, i) Birinci dereceden yanıt yüzeyleri tasarımları ve ii) İkinci dereceden bazı yanıt yüzeyleri tasarımları aşağıda açıklanmaya çalışılacaktır:

i. Birinci Dereceden Yanıt Yüzeyi Tasarımlarının Yapısı

Birinci dereceden yanıt yüzeyi k tane bağımsız değişkeni içeren model formuyla aşağıda verilmiştir:

$$\mathbf{Y} = \beta_0 + \sum_{i=1}^k \beta_i X_i + \epsilon$$

Bu formu matris ve vektörlerle ifade etmek için (2.38) kullanılır:

$$\mathbf{Y} = \mathbf{X}\beta + \epsilon \quad (2.38)$$

Eşitlik 2.38 de Y yanıt değişkeni N denemeden oluşan bir vektör, β bilinmeyen parametreler vektörünü $\beta = (\beta_0, \beta_1, \dots, \beta_k)'_{(k+1) \times 1}$, $\epsilon = (\epsilon_0, \epsilon_1, \dots, \epsilon_k)'_{N \times 1}$ hata vektörünü ve $\mathbf{X}_{N \times (k+1)}$ bağımsız değişkenlerinin düzey kombinasyonlarının oluşturduğu matrisi temsil etmektedir. $\mathbf{X}$ matrisinin ilk sütunu 1 değerlerinden oluşan $N \times 1$ boyutlu bir vektördür. Kalan sütunlar ise, u 'inci denemede (experimental run), i 'inci değişkenin u_i 'inci elemanı olan ϵ_{ui} değeridir. Dolayısıyla kalan sütunlar için oluşan $N \times k$ boyutlu matris D -tasarım matrisi olarak bilinir. O halde $\mathbf{X}$ matrisini $\mathbf{X} = [\mathbf{1}, \mathbf{D}]$ olarak yazabiliriz. $\mathbf{X}$, tam rank-

lı bir matris olmak üzere, β bilinmeyen parametreler vektörünün en küçük kareler tahmin edicisi $\hat{\beta}$ olarak gösterilir.

Bağımsız değişkenler kodlanmış değişkenler olarak $\mathbf{X}$ matrisinde yer alırlar. Modelin gerçek ortalamayı veren kısmı, (2.38) formülünden yola çıkarak (2.39) ile elde edilir:

$$\hat{\mathbf{Y}} = [\mathbf{1}, \mathbf{X}']\hat{\beta} \quad (2.39)$$

Eşitlik 2.39'deki model yanıt değişkenini temsil etmeye yeterli ise, bir $\mathbf{x} = (x_1, x_2, \dots, x_k)'$ noktasındaki kestirilmiş yanıt değişkeni değeri $\hat{\mathbf{Y}}(x)$, (2.40) ifadesiyle hesaplanabilir:

$$\hat{\mathbf{Y}}(x) = [\mathbf{1}, \mathbf{x}']\hat{\beta} \quad (2.40)$$

Birinci dereceden modeller arasından en iyi modeli seçerken seçim kriteri olarak yanıt değişkeninin varyansının minimum olanı benimsenir [24]. Bunun için (2.41) ifadesine dikkat edilmelidir;

$$0 \leq \mathbf{x}'(\mathbf{D}'\mathbf{D})^{-1}\mathbf{x} \leq \|\mathbf{x}\|^2 \cdot \|(\mathbf{D}'\mathbf{D})^{-1}\| \quad (2.41)$$

Burada,

$$\|\mathbf{x}\| = (\mathbf{x}'\mathbf{x})^{1/2}$$

$$\|(\mathbf{D}'\mathbf{D})^{-1}\| = \left\| \sum_{i=1}^k \sum_{j=1}^k (d^{ij})^2 \right\|^{1/2}$$

ifadeleri Öklid ve Frobenius normlarını göstermektedir. Burada, $\|(\mathbf{D}'\mathbf{D})^{-1}\|$ ifadesi $(\mathbf{D}'\mathbf{D})^{-1}$ matrisinin Frobenius normunu göstermektedir. d^{ij} , ifadesi $i, j = 1, 2, \dots, k$, $(\mathbf{D}'\mathbf{D})^{-1}$ matrisinin $\{ij\}$ ' inci elemanıdır. Eşitsizlik incelendiğinde $\mathbf{x}'(\mathbf{D}'\mathbf{D})^{-1}\mathbf{x}$ ifadesini minimize etmek için $(\mathbf{D}'\mathbf{D})^{-1}$ normunu en küçük yapan, $1 \leq u \leq N$ için $x_{u1}, x_{u2}, \dots, x_{uk}$ koordinatlarında $\mathbf{D}$ tasarım matrisinin seçilmesi gerekmektedir.

$\mathbf{D}$ matrisinin $N \times k$ boyutlu olduğu bilinmektedir. Bu matrisin i . sütununu belirtmek için d_i ifadesini kullanalım. O halde, $\mathbf{D} = [d_1; d_2; \dots; d_k]$ yazılabilir.

Ayrıca R bölgesinde, x_i değerleri üzerinde, (2.42) ile verilen kısıtının olduğu varsayalım.

$$d_i' d_i \leq c_i^2 \quad (2.42)$$

Buradaki, c_i , ($i = 1, 2, \dots, k$) sabit bir değerdir. Eşitsizlik 2.42 ile verilen bir kısıt, tasarımın i . koordinat eksenini doğrultusundaki yayılımının c_i^2 ile sınırladığı anlamına gelmektedir. Bu durum (2.43) olarak gösterilebilir:

$$d_{ii} \leq c_i^2 \quad (2.43)$$

Burada, d_{ii} ifadesi $\mathbf{D}'\mathbf{D}$ matrisinin i . köşegen elemanıdır. Ayrıca

$$d^{ii} \geq \frac{1}{d_{ii}} \quad (2.44)$$

olduğu gösterilmiştir [24]. Bir başka deyişle d^{ii} , $(\mathbf{D}'\mathbf{D})^{-1}$ matrisinin köşegen elemanı iken, d_{ii} ifadesi $\mathbf{D}'\mathbf{D}$ matrisinin köşegen elemanıdır ($i = 1, 2, \dots, k$). Böylece (2.43) ve (2.44) ifadelerinden, R bölgesindeki X_i değerleri için

$$d^{ii} \geq \frac{1}{c_i^2} \quad (2.45)$$

olduğu görülebilir. $\text{var}(b_i) = \sigma^2 d^{ii}$ olduğu için, $\text{var}(b_i) \geq \sigma^2/c_i^2$ yazılabilir. Eşitlik 2.44' deki durum, $d_{ij} = 0$ olduğunda, diğer bir deyişle, ($i \neq j$) için $d'_i d_j = 0$ olduğunda sağlanır. Bu durumda, d^{ii} , $d_{ii} = c_i^2$, ($i = 1, 2, \dots, k$) olduğunda (2.45)'deki $1/c_i^2$ değerini alır. Dolayısıyla tasarım, (2.46) ve (2.47) ile verilen koşulları sağlar:

$$d'_i d_j = 0 \quad , \quad i \neq j \quad (2.46)$$

$$d'_i d_i = c_i^2 \quad , \quad i = 1, 2, \dots, k \quad (2.47)$$

Görülüyor ki, d^{ii} minimum değer olarak, $1/c_i^2$ ifadesini almaktadır. Eğer bu koşullar sağlanıyorsa, R bölgesinde, $\|(\mathbf{D}'\mathbf{D})^{-1}\|$ ' in minimum değerine ulaşılabilir.

2.2.1 2^k Faktöriyel Tasarımları

2^k faktöriyel tasarımlarında, her bir faktörün 2 düzeyi vardır. Bu tasarımlarda, k faktörün düzeylerinin bütün mümkün kombinasyonları dikkate alınarak bir tasarım matrisi $\mathbf{D}$ elde edilir. Bu tasarımda her bir faktör en düşük

düzeyindeyken -1 , en yüksek düzeyindeyken $+1$ değeriyle kodlanabilen iki düzeyde ölçülür. Tasarım matrisi kodlama değişkenleri ile 2^k satır ve k sütundan oluşur. Bu matrisin her bir satırı faktör düzeylerinin bir kombinasyonunu gösterir. 2^k düzeni için istatistiksel model k ana etki, $\binom{k}{2}$ iki faktör etkileşimlerini, $\binom{k}{3}$ üç faktör etkileşimlerini ve bir k faktör etkileşimlerini içerir. Böylece bir 2^k düzen için tam model $2^k - 1$ etkiden oluşur.

Kodlanmış değişkenlerin kullanıldığı bir tasarım matrisi için 2^3 tasarımı $\mathbf{D}$ matrisi aşağıdaki gibi oluşturulur:

$$\mathbf{D} = \begin{pmatrix} -1 & -1 & -1 \\ 1 & -1 & -1 \\ -1 & 1 & -1 \\ 1 & 1 & -1 \\ -1 & -1 & 1 \\ 1 & -1 & 1 \\ -1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix} = [d_1 : d_2 : d_3]$$

$i \neq j$ için $\mathbf{D}$ matrisinin d_i ve d_j sütunları için $d^i d_j = 0$ ' dir. Bu nedenle 2^3 tasarımı diktir. $i = j$ için ise $d^i d_j = 8$ ' dir (i=1,2,3). Bu durumda,

$$\mathbf{D}'\mathbf{D} = \begin{pmatrix} 8 & 0 & 0 \\ 0 & 8 & 0 \\ 0 & 0 & 8 \end{pmatrix}$$

$$(\mathbf{D}'\mathbf{D})^{-1} = \begin{pmatrix} 1/8 & 0 & 0 \\ 0 & 1/8 & 0 \\ 0 & 0 & 1/8 \end{pmatrix}$$

elde edilir. Bu durumu 2^k için genelleştirirsek,

$$\mathbf{D}'\mathbf{D} = \begin{pmatrix} 2^k & 0 & \dots & 0 \\ 0 & 2^k & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & 2^k \end{pmatrix}$$

$$(\mathbf{D}'\mathbf{D})^{-1} = \begin{pmatrix} 2^{-k} & 0 & \dots & 0 \\ 0 & 2^{-k} & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & 2^{-k} \end{pmatrix}$$

elde edilir. Buradan $2^k I_k = \mathbf{D}'\mathbf{D}$ yazılabilir. Bu durumda 2^k faktöriyel tasarım-
dan elde edilecek bilinmeyen model katsayıları (2.48) ile bulunur:

$$\beta = \frac{1}{2^k} \mathbf{X}'\mathbf{Y} \quad (2.48)$$

Buna göre $\hat{\beta}_0, \dots, \hat{\beta}_k$ katsayıları 2^k uygulama kombinasyonları altında elde edilmiş sabitlerdir.

Bunun yanında kesikli fraksiyonel tasarımlar da bu başlık altında incelenebilir, ancak bu tezin konusu dışında kaldığından bahsedilmeyecektir. Araştırmacılar genellikle yanıt yüzeyinin şekli hakkında bilgiye sahip olmadıklarından ilk aşamada yanıt yüzeyine birinci dereceden bir model uyarlarlar, ancak yüzeyde eğrisellik durumunda veya uyum eksikliği bulunması nedeniyle modelle yüksek dereceli terimler eklenir ve model ikinci dereceye çıkartılır.

ii. İkinci Dereceden Yanıt Yüzeyi Tasarımlarının Yapısı

İkinci derece yanıt yüzeyi modelleri faktörlerin karesel etkilerinin de modelde yer aldığı tasarımlardır. Eşitlik 2.49 ile, 2 faktörlü 3 düzeyli bir ikinci dereceden yanıt yüzeyi modeli verilmiştir:

$$Y = \beta_0 + \sum_{i=1}^k \beta_i X_i + \sum_{i=1}^k \beta_{ii} X_i^2 + \sum_{i=1}^k \sum_{j=2}^{k-1} \beta_{ij} X_i X_j + \epsilon \quad (2.49)$$

Burada, $X_1, X_2, \dots, X_k$ bağımsız değişkenler olmak üzere, $\mathbf{Y}$ bu değişkenlerden etkilenen yanıt değişkeni, $\beta_0, \dots, \beta_k, i, j = 1, 2, \dots, k$ bilinmeyen parametre kestirimleri ve ϵ rasgele hata terimine karşılık gelir. İkinci dereceden bir yanıt yüzeyi modelini tahmin etmek için deney tasarımları en azından her bir faktörün 3 düzeyini içermelidir. Bu durum 3^k faktör tasarımlarının kullanımını gerektirir. Box ve Wilson [25] tarafından, ikinci dereceden modelleri tahmin etmek için 3^k faktör tasarımlarına bir alternatif olarak merkezi bileşik tasarımlar (composite central design) önerilmiştir. İkinci dereceden bir yanıt yüzeyi modelinin elde edilmesi ve bu modelin incelenmesi için gerekli adımlar şu şekildedir:

1. Tahminlenmiş ikinci dereceden yanıt yüzeyini, yeterli bir model ile temsil etmek.
2. Durağan nokta koordinatlarını belirlerken (2.49) ile elde edilen modeli kullanmak. Bu noktada kestirilmiş yanıt yüzeyinin eğimi sıfırdır. Durağan (stationary) nokta deney bölgesinin içinde bulunursa, aşama 3'e geçilir. Eğer durağan nokta deney bölgesi dışında ise, durağan noktanın bulunduğu yönde başka deneylerin uygulanması gerekmektedir.
3. Durağan noktanın bir maksimum, minimum veya eyer (saddle) noktası olup olmadığını belirlemek.
4. Durağan nokta civarındaki, yanıt yüzeyinin şeklini belirlemek. Buradaki, durağan nokta bir maksimum (veya minimum) veya bir eyer noktası olabilir. Eyer noktasının varlığı durumunda, tahminlenmiş yüzeyin yüksekliği, durağan noktadan belli yönlere hareket edildiğinde artar. Genellikle bir eyer noktasının varlığı, maksimum içeren iki ayrı bölgenin varlığını göstermektedir.

İkinci dereceden tasarımlar 3^k faktöriyel tasarım, Box and Behnken tasarımlar, Merkezi bileşik tasarımlar, Silindirik döndürülebilir tasarımlar, Simetrik olmayan döndürülebilir tasarımlar, Hake tasarımlar, Karışık tasarımlar, vs... gibi sınıflandırılabilir. Bir sonraki bölümde 3^k tasarımlar ve merkezi bileşik tasarımlar detaylarıyla anlatılacaktır. 2^k Faktöriyel Tasarımlar ve Merkezi bileşik tasarımlar için MARS yaklaşımı önceki çalışmalarımızda başarıyla uygulanmıştır [17,18]. Bu tezin kapsamında, önceki çalışma sonuçları da dikkate alınarak ikinci dereceden yanıt yüzeyi modellerine MARS yaklaşımı önerilmiştir.

2.2.2 3^k Faktöriyel Tasarımları

3^k faktöriyel tasarımları her bir faktörün 3 düzeyinin olduğu deneysel düzenlerdir. Bu tasarımlarda faktörün düşük, orta ve yüksek düzeyleri olarak karşılaşılır. Bu düzeyler kodlanarak düşük= -1, orta= 0 ve yüksek= 1 gibi sayısal değerler olarak hesaplamalara katılırlar. Bu düzey kodlamaları her bir faktör için yapılır. Burada, k değeri analizi yapılacak faktör sayısını göstermektedir. Faktör düzeylerinden yararlanarak yanıt değişkeni üzerinde bir modelleme yapılmaya çalışılır. 2^k faktöriyel tasarımlarından farklı olarak regresyon modelinde fazladan bir faktörün olması, tasarım faktörlerinin karesel etkilerinin de modelde bulunmasına sebep olur. Buna göre elde edilen ikinci dereceden model Eşitlik 2.49 ile önceki bölümde gösterilmiştir. Kodlanmış değişkenlerle 3^2 faktöriyel tasarım matrisi aşağıda verilmiştir:

$$\mathbf{D} = \begin{pmatrix} -1 & -1 \\ -1 & 0 \\ -1 & 1 \\ 0 & -1 \\ 0 & 0 \\ 0 & 1 \\ 1 & -1 \\ 1 & 0 \\ 1 & 1 \end{pmatrix}$$

Bunlara ek olarak 3^3 faktöriyel tasarımının da polinomiyal regresyona

uyumu incelenebilir. Benzer şekilde deęişkenler tasarım matrisinin sütunlarına yerleştirilir. Ancak bu tez kapsamında daha detaylı incelenmeyecektir.

2.2.3 Merkezi Bileşik Tasarım

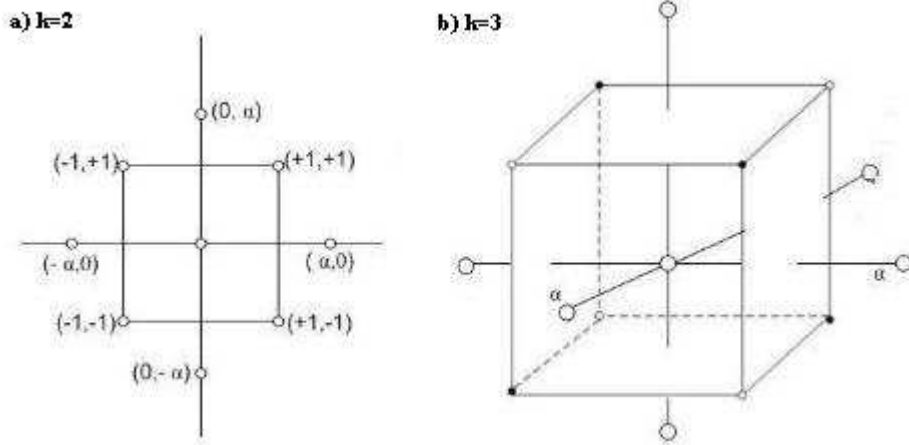
Bu tasarım, ikinci dereceden yüzeylerin katsayılarını tahmin etmek için ilave noktalarla büyütülmüştür; 3^k faktöriyel tasarımlara alternatif bir tasarım olarak geliştirilmiştir ve aşağıdaki kısımlardan oluşmaktadır:

1. Her bir nokta $(x_1, x_2, \dots, x_k) = (\pm 1, \pm 1, \dots, \pm 1)$ biçiminde düzenlenir.
2. Merkez noktaları $(x_1, x_2, \dots, x_k) = (0, 0, \dots, 0)$ olarak tanımlanır.
3. Eksen noktaları, her bir deęişkenin tasarım merkezinden α uzaklığında tanımlanır.

Eksen kısmındaki noktalar aşağıdaki gibi gösterilir:

$$\begin{pmatrix} x_1 & x_2 & \dots & x_k \\ -\alpha & 0 & \dots & 0 \\ \alpha & 0 & \dots & 0 \\ 0 & -\alpha & \dots & 0 \\ 0 & \alpha & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & -\alpha \\ 0 & 0 & \dots & \alpha \end{pmatrix}$$

Bu tasarım $N = 2^k + 2k + n_0$ noktaya sahiptir [24]. Burada n_0 merkez noktası sayısıdır. Faktör noktaları etkileşim terimlerinin tahminine, eksen noktaları ikinci dereceden terimlerin tahminine yardımcı olur. Merkez noktaları da, hatanın bir tahminine olanak sağlar. Merkezi bileşik tasarıma ait grafikler, $k=2$ ve $k=3$ için Şekil 2.8a-b ile verilmiştir [26].



Şekil 2.8: $k = 2$ ve $k = 3$ için merkezi bileşik tasarım

Şekil 2.8 ile verilen tasarımlardan sol paneldeki, $k=2$ için çizilen merkezi bileşik tasarımdır. Burada, tasarımın faktör noktaları eksenler ve dikdörtgenin köşeleri üzerinde yer alır. Merkez noktası eksenlerin kesişim noktasında bulunur. $k=3$ için çizilen merkezi bileşik tasarıma ait grafik ise sağ panelde yer

almıştır. Buna göre faktör noktaları $(\pm 1, \pm 1, \pm 1)$ kübün köşeleri üzerinde ve eksen noktaları (α uzaklığında) ise eksenler üzerinde yer alırlar. Merkez noktası ise kübün merkezinde yer alır.

2.3 Bir Yanıt Yüzeyinin Analizi için Temel Metotlar

Matris notasyonuyla bir yanıt değişkeninin durağan noktalarının belirlenmesinde k değişkenli kestirilmiş yanıt yüzeyi üzerinden gidilir. Durağan noktaların bulunması için (2.50) göz önüne alınır:

$$\hat{Y}(\mathbf{x}) = \mathbf{b}_0 + \mathbf{x}'\mathbf{b} + \mathbf{x}'\mathbf{B}\mathbf{x} \quad (2.50)$$

Burada $\mathbf{b}_0$ sabit terimi göstermek üzere, diğer terimler (2.51) ile verilir:

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \dots \\ x_k \end{bmatrix} \quad \mathbf{b} = \begin{bmatrix} b_1 \\ b_2 \\ \dots \\ b_k \end{bmatrix} \quad \mathbf{B} = \begin{bmatrix} b_{11} & \frac{b_{12}}{2} & \dots & \frac{b_{1k}}{2} \\ & b_{22} & \dots & \frac{b_{2k}}{2} \\ & & \dots & \\ & & & \frac{b_{k-1,k}}{2} \\ sim & & & b_{kk} \end{bmatrix} \quad (2.51)$$

$\mathbf{B}$, simetrik matrisi elemanları denklemdeki ikinci dereceden terimlerinin tahminlerinin bulunduğu bir matristir. $\mathbf{b}$ vektörü ise denklemdeki birinci dereceden terimlerin tahminleridir. $\hat{Y}(\mathbf{x})$ ' nin sırasıyla $x_1, x_2, \dots, x_k$ ' ya göre parçalı (kısmi) türevleri, (2.52) ile verilir:

$$\begin{aligned} \frac{\partial \hat{Y}(\mathbf{x})}{\partial x_1} &= b_1 + 2b_{11}x_1 + \sum_{j=2}^k b_{1j}x_j \\ \frac{\partial \hat{Y}(\mathbf{x})}{\partial x_2} &= b_2 + 2b_{22}x_2 + \sum_{j \neq 2}^k b_{2j}x_j \\ &\dots \\ &\dots \\ &\dots \\ \frac{\partial \hat{Y}(\mathbf{x})}{\partial x_k} &= b_k + 2b_{kk}x_k + \sum_{j=1}^{k-1} b_{kj}x_j \end{aligned} \quad (2.52)$$

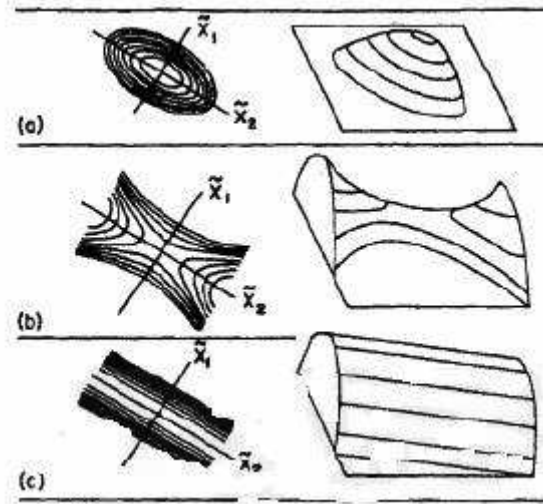
Bütün kısmi türevler, $\mathbf{b} + 2\mathbf{B}\mathbf{x}$ şeklinde de ifade edebilir. Kısmi türevlerin her biri sıfıra eşitlenerek ve x_i değerlerine göre çözülerek, $k \times 1$ boyutlu $\mathbf{x}_0$ vektörünün elemanları yani bu durağan noktaların koordinatları; başka bir deyişle (2.53) bulunmuş olur:

$$\mathbf{x}_0 = \frac{-\mathbf{B}^{-1}\mathbf{b}}{2} \quad (2.53)$$

Burada $\mathbf{B}$, $k \times k$ boyutlu matrisin tersiyle işlem yapılmaktadır.

Durağan noktanın, maksimum, minimum veya eyer noktası mı olduğunun araştırılması gerekir. Bütün bu durumlar Şekil 2.9 üzerinde incelenebilir:

- a) Yanıt değerinde maksimumu söz konudur.
- b) Yanıt değerinde eyer noktası söz konusudur.
- c) Yanıt değeri için bir şey söylenemez.



Şekil 2.9: İkinci dereceden modeller için yüzeyler

İkinci dereceden iki değişkenli yanıt fonksiyonunun kestirimi (2.54) ile verilmiştir:

$$\hat{Y}(x) = b_0 + b_1x_1 + b_2x_2 + b_{11}x_1^2 + b_{22}x_2^2 + b_{12}x_1^2x_2^2 \quad (2.54)$$

Burada, $b_0, b_1, b_2, b_{11}, b_{22}, b_{12}$ katsayıları modeldeki bilinmeyen parametre tahminleridir ve kısmi regresyon katsayıları olarak bilinirler. İlgilenilen yanıt yüzeyinin şekli bu parametrelere bağlı olarak, parabol, daire veya bir hiper düzlem oluşturabilir. Bununla beraber ikinci dereceden modellerde yanıt fonksiyonunun maksimumunun ve minimumunun araştırılması önemlidir. Buna göre (2.54) ifadesinin değişkenlere göre kısmi türevleri alınıp, sıfıra eşitlenirse (2.55) elde edilir:

$$\frac{d\hat{Y}}{dx_1} = b_1 + 2b_{11} + b_{12}x_2 = 0 \quad \frac{d\hat{Y}}{dx_2} = b_2 + 2b_{22} + b_{12}x_1 = 0 \quad (2.55)$$

Bu denklemlerin çözümünden $x_{1,0}$ ve $x_{2,0}$ noktaları elde edilir. Bu noktalar modelde yerine konursa $\hat{Y}_0$ yanıt değerinin maksimum veya minimum değerine ulaşılabilir.

2.3.1 En Hızlı Tırmanış Metodu (Steepest Ascent Method)

Tanım 2.3.1 *En hızlı çıkış yöntemi olarak da bilinir. En hızlı çıkış doğrultusu, $\hat{Y}$ 'nin en hızlı şekilde arttığı doğrultu olup, bu doğrultu kestirilmiş yanıt yüzeyinin normaline paraleldir. İlgili bölgenin merkezi boyunca kestirilmiş yüzeyin normali olan doğru, en hızlı çıkış yolu olarak bilinir. Deneyler en hızlı çıkış yolu boyunca yanıt değerinde artış durana kadar devam edilir. Daha sonra yeni bir çıkış yoluyla yöntem devam ettirildiğinde yanıt değerinin optimum komşuluğuna gelinir. Başka bir deyişle, verilen bir fonksiyonun belirlenmiş bir noktadaki artışının(azalış) en yüksek oranının yönüdür. Gradyant yönünde amaç fonksiyonunun en büyük değerini veren noktaya hareket ederken en iyi adım büyüklüğünü sağlamaya çalışılması da denilebilir.*

Varsayalım ki f , P noktasında türevlenebilir olsun ve f 'in P noktasındaki gradyanı $\nabla f_P \neq 0$ 'dır.

Önerme 1 *i. $D_u f$ yönlü türevinin P 'deki en büyük değeri $\|\nabla f_P\|$ ' dir ve u birim vektörü ∇f_P 'nin yönünde olduğunda ortaya çıkar.*

ii. $D_u f$ yönlü türevinin P 'deki en küçük değeri $\|-\nabla f_P\|$ ' dir ve u birim vektörü $-\nabla f_P$ 'nin yönünde olduğunda ortaya çıkar.

Eğer $\mathbf{u}$ herhangi bir birim vektör ise, o zaman **Kanıt.** $D_u f = \nabla f_P \cdot \mathbf{u} = \|\nabla f_P\| \|\mathbf{u}\| \cos\theta = \|\nabla f_P\| \cos\theta$ ■ dır. Burada, θ , ∇f_P ve $\mathbf{u}$ arasındaki açıdır. Ancak $\theta = 0$ iken $\cos\theta = 1$ ' dir öyle ki $\mathbf{u}$ ∇f_P yönünü gösterir. Sonuç olarak, $D_u f$ ' nin olası en büyük değeri $D_u f = \|\nabla f_P\| (1) = \|\nabla f_P\|$ ' dir. Ayrıca $\cos\theta$ ' nın en küçük değeri $\theta = \pi$ iken, -1 ' dir. Bu değer birim vektör $\mathbf{u} = -\nabla f_P$ ' e işaret eder ve

$$D_u f = \|\nabla f_P\| (-1) = -\|\nabla f_P\|$$

yönlü türevini verir [27].

2.3.2 Kanonik Analiz

Bu analizin amacı aslında uyumu yapılmış bir ikinci dereceden eşitliğin yeniden ifade edilmesidir. Yanıt değişkeninin durağan nokta x_0 orijinli yeni bir koordinat sistemine dönüşümü ve bu sistemin eksenlerinin döndürülmesinden oluşur. Bu eksenlerin tüm çapraz terimlerini yok edecek şekilde döndürülmesi ile elde edilir. Bu yaklaşım Box ve Draper tarafından A kanonik formu olarak adlandırılmıştır [1]. Eğer istenirse, orijinin değiştirilmesiyle birinci derece terimlerin elde edilmesi mümkün olabilir. Bu yaklaşımı da benzer şekilde B kanonik formu olarak adlandırmışlardır.

Varsayalım ki uyumu yapılmış model ikinci dereceden bir model olsun. $\mathbf{B}$, $k \times k$ boyutlu matrisi elemanları denklemdaki ikinci dereceden terimlerin tahminlerinin bulunduğu simetrik bir matristir. Bu matrisin özdeğerleri, $\lambda_1, \lambda_2, \dots, \lambda_k$ ve bu özdeğerlere karşılık gelen özvektörler de $\mathbf{m}_1, \mathbf{m}_2, \dots, \mathbf{m}_k$ olsun. Buradan (2.56), ($m_i \neq 0$), yazılabilir:

$$\mathbf{B}\mathbf{m}_i = \mathbf{m}_i\lambda_i \quad i = 1, 2, \dots, k \quad (2.56)$$

$\mathbf{B}$ matrisi (2.57) ile temsil edilir:

$$\mathbf{B} = \begin{bmatrix} b_{11} & \frac{b_{12}}{2} & \dots & \frac{b_{1k}}{2} \\ & b_{22} & \dots & \frac{b_{2k}}{2} \\ & & \dots & \\ & & & \frac{b_{k-1,k}}{2} \\ sim & & & b_{kk} \end{bmatrix} \quad (2.57)$$

Her bir özvektör $\mathbf{m}_i' \mathbf{m}_i = \mathbf{1}$ olacak şekilde ve $\mathbf{m}_i$, $k \times k$ boyutlu $\mathbf{M}$ matrisinin i ' inci sütunu iken, $\mathbf{M}$ ortonormal bir matris olur. Dolayısıyla k tane (2.56) ifadesi, (2.58) gibi benzer formda yazılabilir:

$$\mathbf{B}\mathbf{M} = \mathbf{M}\mathbf{\Lambda} \quad (2.58)$$

Eşitlik 2.58' de $\mathbf{\Lambda}$, i ' inci köşegen elemanı λ_i olan köşegen bir matristir. Eşitliğin sol taraftan $\mathbf{M}' (= \mathbf{M}^{-1})$ ile çarpımıyla (2.59) elde edilir:

$$\mathbf{M}'\mathbf{B}\mathbf{M} = \mathbf{\Lambda} \quad (2.59)$$

$\mathbf{M}\mathbf{M}' = \mathbf{I}$ olduğu dikkate alınır, o zaman genel model olarak (2.60) elde edilir.

$$\hat{y} = b_0 + \mathbf{x}'\mathbf{b} + \mathbf{x}'\mathbf{B}\mathbf{x} = b_0 + (\mathbf{x}'\mathbf{M})(\mathbf{M}'\mathbf{b}) + (\mathbf{x}'\mathbf{M})\mathbf{M}'\mathbf{B}\mathbf{M}(\mathbf{M}'\mathbf{x}) \quad (2.60)$$

Buradan, $\mathbf{X} = \mathbf{M}'\mathbf{x}$ ve $\mathbf{\Theta} = \mathbf{M}'\mathbf{b}$ ($\mathbf{x} = \mathbf{M}\mathbf{X}$ veya $\mathbf{b} = \mathbf{M}\mathbf{\Theta}$) olarak alınır, (2.61) yazılabilir:

$$\hat{y} = b_0 + \mathbf{X}'\mathbf{\Theta} + \mathbf{X}'\mathbf{\Lambda}\mathbf{X} \quad (2.61)$$

Eşitlik 2.61'i açık formuyla yeniden yazarsak,

$$\hat{y} = b_0 + \Theta_1 X_1 + \dots + \Theta_k X_k + \lambda_1 X_1^2 + \dots + \lambda_k X_k^2 \quad (2.62)$$

elde edilir. Görüldüğü gibi $\mathbf{A}$ kanonik form ile çapraz terimler yok edilmiştir. Eşitlik 2.62'nin $X_1, X_2, \dots, X_k$ ' ya göre kısmi türevleri alındığında durağan nokta (2.63) ile bulunur:

$$X_{is} = \frac{-\Theta_i}{2\lambda_i} \quad (2.63)$$

λ_i özdeğerinin büyüklüğü ve işareti uyumu yapılmış olan ikinci derece yüzeyin

tipini belirlemektedir. Θ_i değeri döndürülmüş koordinat eksenleri yönünde, orijinal orijin olan $\mathbf{x} = \mathbf{0}$ noktasındaki yüzeyin eğimini vermektedir. X_{is} değerleri, kanonik eksenler boyunca durağan nokta S' ye olan uzaklığı ölçer.

2.4 Model Seçim Kriterleri

Çoklu doğrusal regresyonda model seçimi için belirlenmiş pek çok kriter vardır. Bunlardan bazıları "Çapraz Geçerlilik Kriteri (Cross Validation)", "Genelleştirilmiş Çapraz Geçerlilik Kriteri (Generalized Cross Validation)", AIC (Akaike Information Criteria) ve " C_p Kriteri (Mallow's C_p)" olarak sıralanabilir [21]. Bir sonraki alt bölümde bunların tanımlarına ilişkin bilgiler verilecektir.

2.4.1 Çapraz Geçerlilik Kriteri

Regresyon eğrisinin uyumu konusunda en çok kullanılan kriterlerden biri "Artık Kareler Toplamı (AKT)" olarak bilinir. Parametrik regresyonda gözlem değeri y_i tahmincisinin bir parçası olarak kullanıldığından bu durum, AKT'nın model seçimi için uygun olmadığı sonucunu çıkarır. Ancak çapraz geçerlilik kriteri bu problemin üstesinden gelir. λ düzeltme parametresiyle x noktasındaki nonparametrik regresyon tahmini $\hat{f}(x; \lambda)$ olsun. Buna göre, AKT (2.64) ile elde edilir:

$$AKT(\lambda) = \sum_{i=1}^n \left\{ y_i - \hat{f}(x_i; \lambda) \right\}^2 \quad (2.64)$$

Çapraz geçerlilik (CV) kriteri ise (2.65) ile verilir:

$$CV(\lambda) = \sum_{i=1}^n \left\{ y_i - \hat{f}_{-i}(x_i; \lambda) \right\}^2 \quad (2.65)$$

Burada $\hat{f}_{-i}$ veri setinden (x_i, y_i) gözlem çiftinin çıkartılmasıyla elde edilen nonparametrik regresyon tahmincisidir. Birini dışarda bırakma stratejisi AKT(λ) kriterinin yetersizliğini ortadan kaldırır. λ' nın belirlediği CV, $\lambda > 0$ olduğu

sürece $CV(\lambda)$ değerini minimum yapandır.

2.4.2 Genelleştirilmiş Çapraz Geçerlilik Kriteri

$CV(\lambda)$ için yazılan algoritmalar 1980'lerin ortalarında geliştirilmiştir. Ancak, çapraz geçerlilik formülünün hesabındaki zorluk nedeniyle, genelleştirilmiş çapraz geçerlilik (GCV) kriteri daha çok kullanılmaya başlanmıştır. Bu kriter, (2.66) ile verilmiştir:

$$GCV(\lambda) = \sum_{i=1}^n \left(\frac{\{(\mathbf{I} - \mathbf{S}_\lambda)\mathbf{y}\}_i}{1 - n^{-1}tr(\mathbf{S}_\lambda)} \right)^2 = \frac{AKT(\lambda)}{\{1 - n^{-1}tr(\mathbf{S}_\lambda)\}^2} \quad (2.66)$$

Burada S_λ düzleştirme matrisi olarak yer alır. Genelleştirilmiş çapraz geçerlilik hem hata terimini hem de model karmaşasını hesaba katar. GCV, düzeltme splaynları kapsamında ilk olarak Craven ve Wahba tarafından önerilmiştir [13]. MARS algoritması, Craven ve Wahba tarafından önerilen bu kriterden yararlanarak model seçiminde modifiye edilmiş bir GCV hesabı yapar.

2.4.3 C_p Kriteri

Çoklu regresyon modelleri seçiminde kullanılan bir başka kriter C_p değeri olarak karşımıza çıkar. Bu kriter (2.67) ile verilmiştir:

$$C_p = AKT(p) + 2\hat{\sigma}_\epsilon^2 p \quad (2.67)$$

Burada $\hat{\sigma}_\epsilon^2$, (2.68) ile verilir:

$$\hat{\sigma}_\epsilon^2 = \frac{AKT(\lambda)}{df_{artk}(\lambda)} \quad (2.68)$$

dir. Burada df_{artk} , bu terime ait serbestlik derecesidir. $C_p(\lambda)$ yaklaşık olarak $GCV(\lambda)$ 'ya eşittir. Bu durum (2.69) ile verilmiştir:

$$GCV(\lambda) \simeq AKT(\lambda) + 2\hat{\sigma}_\epsilon^2(\lambda)df_{it}(\lambda) \quad (2.69)$$

Burada $d_{fit} = iz(S_\lambda)$ ' dır. Bu iki kriter arasındaki temel fark, GCV, σ_ϵ^2 ' yi $AKT(\lambda)$ ' yi kullanarak tahminlerken, $C_p(\lambda)$, σ_ϵ^2 ' nin bir tahminine ihtiyaç duyar.

2.4.4 Akaike Bilgi Kriteri

GCV ve C_p ' den farklı olarak Akaike Bilgi Kriteri (AIC) kriteri, AKT ve $df_{artk}(\lambda)$ arasında dengeyi sağlamaya çalışır [28]. Bu durum (2.70) ile ifade edilir:

$$AIC(\lambda) \equiv \log(AKT(\lambda)) + 2df_{fit}(\lambda)/n \quad (2.70)$$

Literatürde modifiye edilmiş AIC kriterlerine rastlamak mümkündür.

3 MARS

3.1 Yinelemeli Bölüntüleme (Recursive Partitioning)

Yinelemeli bölüntüleme regresyon modelleme metodolojisinin asıl kaynağı 1960' larda Morgan ve Sonquist tarafından geliştirilen AID(Automatic Interaction Detection) adında bir programa dayanır [29]. Bunun üzerine yapılan çalışmalar Breiman tarafından 1984' te geliştirilmiştir [30]. Bir sonraki alt bölümde yinelemeli bölüntüleme konusu ikiye ayrılarak açıklanmaya çalışılacaktır. Yinelemeli bölüntüleme, altbölgelerin yinelemeli olarak bölünmesiyle açıklanır [31].

3.1.1 Altbölgelerin Yinelemeli Bölünmesi (Recursive Splitting)

y yanıt değişkeninin $\mathbf{x} = (x_1, \dots, x_k)$, k adet tahminleyici değişkenin vektörü ile bilinmeyen ilişkisi (3.1) ile verilsin:

$$y = f(x_1, \dots, x_k) + \epsilon \quad (3.1)$$

Varsayalım ki, N tane $\{y_j, x_j\}_{j=1}^N$ örneği olsun. $\{R_i\}_{i=1}^S$, $D \subset \mathbb{R}^k$ nin S tane ayrık altbölgelerin bir kümesi olsun. Yinelemeli bölüntüleme, verilen $\{R_i\}_{i=1}^S$ altbölgeleriyle, bilinmeyen $f(\mathbf{x})$ fonksiyonunu, $\mathbf{x}$ değerinde (3.2) ile tahmin eder:

$$\hat{f}(\mathbf{x}) = \hat{f}_i(\mathbf{x}) \quad (3.2)$$

Burada, $\mathbf{x} \in R_i$ için, $\hat{f}_i(\mathbf{x})$ fonksiyonu, D' nin R_i alt bölgesi üzerindeki bilinmeyen $f(\mathbf{x})$ fonksiyonunu tahmin eder. Yinelemeli bölüntülemeye, $\hat{f}_i(\mathbf{x})$ sabit bir fonksiyon olarak alınır ve bu durum (3.3) ile gösterilir [29, 30].

$$\hat{f}_i(\mathbf{x}) = c_i \quad \forall \quad \mathbf{x} \in R_i \quad (3.3)$$

Burada, her c_i hata kareler toplamını minimum (Lack of fit) yapan i .nci bileşen olarak düşünülür. Buna göre,

$$Lof \left[\hat{f}_i(\mathbf{x}) \right] = \min_{c_i} \sum_{\mathbf{x} \in R_i} (y_i - c_i)^2 \quad (3.4)$$

D tanım bölgesinin altbölgeleri ayrık olduklarından her bir c_i , $\{x_i\}_{i=1}^N$, ye ait olan y_i ' lerin örnek ortalaması olacaktır. Yinelemeli bölüntülemeye ilk olarak tanım bölgesi D tek bir altbölge olarak kabul edilir. Buna göre $R_1 = D$ yazılabilir. İleriye dönük algoritmanın ilk adımı, altbölge sayısı M kullanıcı tarafından seçilen, çok sayıda $M \geq S$ için $\{R_i\}_{i=2}^M$ ayrık altbölgelere ayırmaktır. Burada $i = 2, \dots, M$ olduğuna dikkat edilmelidir. İkinci olarak, geriye dönük algoritma ilk adımı geriye çevirir; modeldeki altbölge sayısını ve model uygunluğunu hesaplayan bir ölçüt kullanarak modelden $M - S$ kadar fazla altbölgeyi budar. İki algoritma kullanmanın amacı veriyi c_i sabit fonksiyonları ile beraber en iyi $\{R_i\}_{i=1}^S$ altbölgeler kümesi elde edecek şekilde, tanım bölgesinin her bir altbölgesi üzerinde tahmin eden $f(\mathbf{x})$ fonksiyonunu seçmektir. Varsayalım ki, $k = 3$ tane açıklayıcı değişken ve ileriye dönük adım altbölge sayısı maksimum $M = 5$ olsun. $v = 1, \dots, k$ açıklayıcı değişkenleri indeksleyen ve $p = 1, \dots, n$ de R_i altbölgesindeki bir açıklayıcı değişken olan x_v ' nin sıralı değerlerini indeksleyen simge olsun. Bu durumda m altbölgeyi yinelemeli bölüntüleme modeli için ileriye dönük adım ölçütü olarak (3.5) ile verilir:

$$Lof_m = \sum_{i=1}^m Lof[\hat{f}_i(\mathbf{x})] \quad (3.5)$$

Burada dikkat edilmelidir ki, $x_{v,p}$ ile belirtilmek istenen v . açıklayıcı değişkenin p . sıralı örnek değeridir. Tek başına x_v ifadesi ise v . açıklayıcı değişkenin sabit değerleridir. İleriye dönük adım yinelemeli bölüntüleme algoritmasının başında R_1 tanım bölgesi D ' nin kendisidir ve $f(\mathbf{x})$ için tek altbölge tahmini (3.6) ile verilir:

$$\hat{f}(\mathbf{x}) = \hat{f}_1(\mathbf{x}) = c_1 = \frac{1}{N} \sum_{j=1}^N y_j \quad (3.6)$$

Tek altbölge yinelemeli bölüntüleme modelinin ileriye dönük adım uyum ölçütü (3.7) ile verilir:

$$Lof_1 = \sum_{j=1}^N (y_j - c_1)^2 \quad (3.7)$$

Başlangıç bölüntüsü, $m = 2$, ileriye dönük adım algoritması için, R_1 altbölgesini en iyi iki ayrık bölgeye ayıran t^* bölüntü noktasını seçer. t^* değerinin

bulunması için metot şu şekilde ilerler:

Her örnek değeri $x_{v,p}$ ' yi ($v = 1, \dots, k; p = 1, \dots, n$) potansiyel bir bölüntü noktası olarak alır ve bunların arasından $m = 2$ altbölge modeli için Lof değerini minimum yapanı hesaplar. Başka bir deyişle, $t = x_{1,15}$ değeri x_1 açıklayıcı değişkeni için potansiyel bir bölüntü noktası olsun. Bu durumda, ana altbölge olan R_1 , t ' nin solundaki alanda, $x_1 < t$, $R_{1,sol}$ altbölgesini oluşturur. t ' nin sağındaki alan ise, $x_1 \geq t$, $R_{1,sag}$ altbölgesini oluşturur. R_1 ' in $t = x_{1,15}$ boyunca uygun şekilde bölünmesinden sonra, model $m = 2$ altbölge için Lof_m değeri hesaplanır. Buna göre,

$$Lof_2 = \min_{c_{sol}} \sum_{x_j \in R_{1,sol}} (y_j - c_{sol})^2 + \min_{c_{sag}} \sum_{x_j \in R_{1,sag}} (y_j - c_{sag})^2 \quad (3.8)$$

elde edilir. v ve p indeksleri kullanılarak R_1 dahilinde olan her açıklayıcı değişken için sırasıyla bütün mümkün bölüntü noktalarını araştırır (Burada R_1 , D tanım bölgesine eşittir).

Bölüntü noktası $t^* = x_{2,25}$, R_1 altbölgesini ayıran nokta olduğunda, $m = 2$ altbölge için ileriye dönük adım Lof_m ölçütünü minimum yapan nokta anlaşılır. $x_{2,25}$ noktası önceki ana bölgeyi iki yeni ayrık altbölgeye bölmek ve eskisini ortadan kaldırma işlemi boyunca kullanılır. Bu eski bölge R_{1*} olarak adlandırılır. İlk olarak, ana altbölge R_{1*} ' daki alan, t^* ' in solunda, $x_2 < t^*$, olacak şekilde oluşturulur ve buna eş altbölge R_2 denir. Buna karşın t^* ' in sağındaki alan tekrar R_1 olarak kalır. R_1 ve R_2 ayrık yeni bölgelerin yaratılması ve eski ana bölge olan R_{1*} 'i n ortadan kaldırılması, D tanım bölgesinin, bir bir alt bölgelere bölünmesiyle, ileriye dönük adım süreci tamamlanana kadar sürer gider. Dolayısıyla, $f(\mathbf{x})$ 'in iki altbölge yinelemeli bölüntüsünün tahmini (3.9) ile verilir:

$$\hat{f}(\mathbf{x}) = c_i \quad \mathbf{x} \in R_i, \quad i = 1, 2 \quad (3.9)$$

$$\mathbf{x} \in \begin{cases} R_1 & , x_2 \geq x_{2,25} \\ R_2 & , x_2 < x_{2,25} \end{cases} \quad (3.10)$$

Burada belirtmelidir ki, (3.2) ifadesi bölüntüleme boyunca bozulmamıştır. Sadece D' yi bölen ayırık altbölgelerin sayısı değişmiştir.

İleriye dönük adım algoritmanın yinelemeleri, $m = 3, \dots, M = 5$ 'e kadar yinelemeyi tekrar eder. Bundan sonra araştırma, mevcut $m - 1$ altbölge modelinden altbölgelerin sadece ve sadece biri için en iyi bölüntüleme (Lof_m değerinin en küçüklenmesi) oluşturur. Önceki gibi, her yinelemenin bölüntü noktası olan t^* benzer şekilde seçilir. Sonra her bir açıklayıcı değişken için modelin varolan $\{R_i\}_{i=1}^{m-1}$ altbölgeleri içinde bütün mümkün potansiyel noktaların hesabı yapılır. Yinelemeli bölüntüleme, tanım bölgesi olan D , $M = 5$ parçaya ayrılana kadar devam eder. Bu durumda $\{R_i\}_{i=1}^5$ altbölgeleri oluşmuş olur. İleriye dönük adım algoritmanın tamamlanması ile geriye dönük adım algoritması hem modelin uygunluğunu hem de altbölge sayısını hesaplayan bir kriter kullanarak düzeltme yapar. Geriye dönük adım algoritmanın da tamamlanmasıyla yinemelemeli bölüntüleme sonuçlanır.

Her bir R_i altbölgelerinin araştırılmasında taban fonksiyonları kümesinin bir açılımı kullanılır. Aday bölüntüleme noktasının seçimi, $\hat{f}(\mathbf{x})$ için özel bir fonksiyonel form yaratır. Bunun için algoritmada g fonksiyonu kullanılmıştır.

Algoritmada görülen basamak fonksiyonları önemlidir. H basamak fonksiyonu (3.11) ile gösterilir:

$$H(\eta) = \begin{cases} 1 & , \eta \geq 0 \\ 0 & , dd \end{cases} \quad (3.11)$$

Friedman 1988' de yinelemeli bölüntüleme için, taban fonksiyonları kümesini kullanan ileriye dönük-adımsal bir algoritma geliştirmiştir [32]. Buna göre:

- (1) $R_1 = V, B_1(\mathbf{x}) = 1$
- (2) For $R_m, m = 2$ to M do (Her altbölge için)
- (3) $lof^* = \infty, i^* = 0, v^* = 0, t^* = 0$
- (4) For $R_i, i = 1$ to $m - 1$ (Herbir kurulmuş altbölge için)
- (5) For $v = 1$ to p do (Herbir tahminleyici için)
- (6) For $t = x_{v,k=1}$ to $x_{v,k=n}$ (Her bir değer için)
- (7) $g = \sum_{d \neq j} c_d B_d(\mathbf{x}) + c_m B_i(\mathbf{x}) H[t - x_v] + c_i B_i(\mathbf{x}) H[x_v - t]$

(8) $lof = Lof_m$
 (9) If $lof < lof^*$ then $lof^* = lof; i^* = i; v^* = v; t^* = t$ end if
 (10) end for
 (11) end for
 (12) end for
 (13) $R_m \leftarrow \{R_{i^*} : (t^* - x_{v^*}) > 0\}$
 (14) $R_{i^*} \leftarrow \{R_{i^*} : (x_{v^*} - t^*) \geq 0\}$
 (15) end for
 (16) end for

İleriye dönük adım yineleme bölüntüleme algoritması ilk olarak R_1 altbölgesinin D tanım bölgesine eşitlenmesiyle başlatılır(1). Bir sonraki döngüde(2) $\{R_m\}_{m=2}^M$ altbölgeleri iteratif olarak oluşturulur. Daha sonra, yapay değişkenler lof^* sürecini, j^* altbölgesini, v^* açıklayıcı değişkeninin ve t^* bölüntüleme noktasının hesabı için başlangıç değerleri belirlenir ve bu değerler kurulmuş olan bir $\{R_i\}_{i=1}^{m-1}$ altbölgesinin bir sonraki altbölüntüsünü tayin etmede kullanılır(3). Bundan sonraki iç üç döngü (4-5-6) bir sonraki bölüntüleme noktasının bulunmasını, iteratif olarak bütün kurulan altbölgeler(4), bütün açıklayıcı değişkenler(5) ve j . nci altbölgede(6) olan açıklayıcı değişkenlerin bütün değerleri arasından seçerek yapar. Verilen bir R_i altbölgesi içindeki x_v açıklayıcı değişkeni için bir bölüntüleme noktası t , g fonksiyonu (7) parametre vektörü $\mathbf{c} = (c_1, c_2, \dots, c_m)$ ile, $f(\mathbf{x})$ 'in ileriye dönük adım m . nci iterasyonunda bir yinelemeli bölüntüleme model tahmini için adaydır. (7) ifadesindeki ilk terim R_i altbölgesi hariç bütün altbölgeleri içine alır. Sonraki iki terim ise (7) ifadesiyle

$$c_m B_i(\mathbf{x}) H[t - x_v] + c_i B_i(\mathbf{x}) H[x_v - t]$$

olarak karşımıza çıkar. Burada yapılan aslında R_i ana bölgesinin iki ayrık eş altbölgeye, basamak fonksiyonları olan $H[t - x_v]$ ve $H[x_v - t]$ leri kullanarak her bir $\mathbf{x}$ in yerinin t bölüntüleme noktasına göre ayrılmasıdır. Daha sonra, Lof_m (8) değeri, veriye göre g fonksiyonunun ileriye dönük adım ölçüsünü hesaplayan bir değerdir. En iyi bölüntüleme, açıklayıcı değişken ve altbölge

araştırması(9), m . nci iterasyonda kurulan bir $\{R_i\}_{i=1}^{m-1}$ altbölgesinin en iyi bölüntülemesini elde edene kadar devam eder. m . nci iterasyonun bitmesiyle, yani R_i ana bölgesinin iki eş altbölgeye x_v ' nin bölüntü noktası t ' ye bağlı olarak ayrılmasıyla(13-14)(ve ortadan kaldırılmasıyla) biter. (İterasyonlar tanım bölgesi D ' nin M tane ayrık $\{R_i\}_{i=1}^M$ altbölgeye bölüntülenmesiyle devam eder).

Her bir taban fonksiyon $B_i(\mathbf{x})$, D ' nin R_i . nci altbölgesiyle olan ilişkisini tanımlar ve bu bölüntü noktaları R_i . nci altbölgeyi tanımlayan taban fonksiyonları çarpımının bir sonucudur. Bunu bir örnekle açıklamak gerekirse, $D \in \mathbb{R}^2$ ve R_5 bölgesi sırasıyla $H[x_1 - t_1^*]$, $H[t_2^* - x_2]$, $H[x_2 - t_3^*]$ ve $H[t_4^* - x_1]$ basamak fonksiyonlarından oluşan bir altbölge olsun. Burada $\{t_i^*\}_{i=1}^4$ değerleri sırasıyla 0, 1, 0, 1' dir. O halde taban fonksiyonu olan $B_5(\mathbf{x})$ (3.12) ile gösterilir:

$$B_5(\mathbf{x}) = H[x_1 - 0] \times H[1 - x_2] \times H[x_2 - 0] \times H[1 - x_1] \quad (3.12)$$

Eşitlik 3.12 ifadesi $\mathbb{R}^2$ ' de bir birim kareyi temsil eder. Taban fonksiyon B_5 , (3.13) ile yeniden düzenlenebilir:

$$B_5(\mathbf{x}) = \begin{cases} 1 & , 0 \leq x_1 \leq 1, 0 \leq x_2 \leq 1 \\ 0 & , \text{diğer durumlarda} \end{cases} \quad (3.13)$$

Yinelemeli bölüntülemede, $\{R_i\}_{i=1}^S$ altbölgeleri ayrıktır. Her bir veri noktası $\mathbf{x}$ sadece tek bir R_i altbölgesine aittir.

Yinelemeli bölüntüleme, regresyon modellemede oldukça hızlı hesaplama yapabilen kullanışlı bir metottur. Özellikle $\hat{f}_i(\mathbf{x})$ bir sabit fonksiyon c_i olarak alındığında hızlı sonuç verir [31].

3.1.2 Yinelemeli Bölüntülemenin Olumsuzlukları

Yinelemeli bölüntülemenin dezavantajları, süreksizlik durumu, değişken etkileşimleri ve model yorumlarındaki eksiklikleridir.

Yinelemeli bölüntüleme algoritmasının çok boyutlu durumlar için uygulama zorlukları bulunmaktadır. İteratif bölünme ve ana bölgeyi eş altbölgelere ayırırken ortadan kaldırma durumları, doğrusal ve toplamsal fonksiyonu tahminlemede zorluklar çıkartır. $H[\eta]$ Basamak fonksiyonlarının süreksizliği, ileriye dönük adım algoritma(satır 7)' deki her doğrusal regresyon için uygulandığında süreklilikten kayba sebep olmaktadır.

Friedman [32], yinelemeli bölüntülemenin doğrusal ve toplamsal fonksiyonun tahmininde yarattığı zorluklar nedeniyle bazı önerilerde bulunmuştur. Buna göre eş bölgeler oluşturulurken ana bölge ortadan kaldırılmamalıdır. Dolayısıyla, gelecek iterasyonlarda hem ana bölge hem de eş bölgeler daha sonraki bölüntülemeler için kullanılabilir. Bunun doğal bir sonucu olarak, kalan ana bölgeler tanımın altbölgeleriyle örtüşür. Ayrıca, her bir ana bölge pek çok eş altbölgenin kümelerinden oluşabilir. Bu modifikasyonla, yinelemeli bölüntüleme, başlangıç bölgesi R_1 ' in farklı açıklayıcı değişkenler tarafından tekrarlı bölüntülemelerle doğrusal modeller üretebilir. Birden fazla açıklayıcı değişkenli toplamsal modellerin elde edilmesi yinelemeli bölüntülemenin farklı açıklayıcı değişkenler kullanmasından kaynaklanmaktadır.

Bu şekilde çalışan yinelemeli bölüntüleme algoritması esnekliği daha fazla olan modeller sınıfı oluşturur. Ancak bu modifikasyon hala $H[\eta]$ basamak fonksiyonunun süreksizliği durumunu değiştirmemektedir. Bunun için Friedman, algoritmadaki(satır 7) basamak fonksiyonunu $H[\eta]$, doğrusal regresyon splaynlarıyla ($q=1$ dereceden) değiştirmeyi önermiştir. Ancak bu regresyon splaynları soldan(-) ve sağdan(+) budanmış splayn formları kullanır. r_m bileşenleri doğrultunun yönünü (sola veya sağa) gösteren R_m . altbölgeyle birleşmiş 2 boyutu ifade etsin. Öyle ki, bunlardan ilki özel bir açıklayıcı değişkeni diğeri ise kendi ana bölgesinden altbölge oluşturmak için kullanılan bölüntüleme noktasını gösterebilir. Sağ ve sol budanmış splaynlar R_m ve R_{m+1} altbölgelerini oluşturur. Bu altbölgeler R_i ana bölgesinden x_v ' nin içinde bir bölüntüleme noktası olan t ile elde edilmiş R_m . ve R_{m+1} . altbölgelerdir. Buna göre, bu altbölgeler (3.14) ile tanımlanır:

$$T_{i,r_m}(\mathbf{x}) = [(t - x_v)_+]^{q=1} = (t - x_v)_+ \quad (3.14)$$

$$T_{i,r_{m+1}}(\mathbf{x}) = [(x_v - t)_+]^{q=1} = (x_v - t)_+$$

$r_m = (-v, t)$ ve $r_{m+1} = (+v, t)$ ve $m > i$ ' dir. Fazladan yazılan alt indisler i, m veya $i, m+1$ çoklu açıklayıcı değişken içerisindeki etkileşim terimlerine müade edildiğinde budanmış splayn fonksiyonların çarpımlarını ortaya koymak için kullanılır. Budanmış splayn fonksiyonlar argümanları açıklayıcı değişkenlerin bir vektörüdür.

Eşitlik 3.14' te kullanılan ifadelerdeki modelleme yaklaşımı sürekli bir $\hat{f}(\mathbf{x})$ yaklaşımı verir, öyle ki, $\hat{f}(\mathbf{x})$ ' in modeldeki her açıklayıcı değişkenin bölüntüleme noktalarında ilk kısmi türevi süreksizdir. Doğrusal budanmış splayn (3.14) için kullanılan ifade esneklikten yana çok az bir kazanç ve hesaplama olarak fazlaca bir iş yükü gerektigini gösterir. Doğrusal splayn fonksiyonları regresyon modelinin hızlı güncellemesine müade eder.

Yinelemeli bölüntülemedeki bu zorluklar, yapılan modifikasyonlarla MARS algoritmasının oluşturulmasına yaramıştır.

MARS algoritması, D tanım bölgesi üzerinde $\{R_i\}_{i=1}^S$ altbölgeleriyle örtüşen doğrusal $q = 1$ budanmış splayn modelleri oluşturur. Bir MARS modelinin her bir örtüşen altbölgesi doğrusal budanmış splaynların sıralı bir dizisinden gelen tahminleyici değişkenlerin bölüntüleme noktasıyla tanımlanır.

Taban fonksiyonun çarpımı $K_m(\mathbf{x})$, R_m altbölgesi ile ilişkili budanmış splaynların bir sıralı dizisi olarak tanımlansın. Her taban fonksiyonu çarpımının ilk terimi $T_{0,r_1}(\mathbf{x}) = 1$ ' dir. Buna R_1 ile ilişkili başlangıç fonksiyonu denir. Her bir eklenen budanmış splayn bir ana bölgenin bir eş altbölgeye iteratif bölüntülenmesidir. Varsayalım ki, R_T ana bölgesi için sıralı budanmış splaynlar dizisi $(1, 3, 7)$ olsun. Buna göre, R_m altbölgesi yaratmak için ana bölge $T_{T,r_m}(\mathbf{x})$ kullanılarak (3.15) elde edilir:

$$K_m(\mathbf{x}) = T_{0,r_1}(\mathbf{x}) \times T_{1,r_3}(\mathbf{x}) \times T_{3,r_7}(\mathbf{x}) \times T_{7,r_m}(\mathbf{x}) \quad (3.15)$$

Burada $m > 7$ ' dir. Eşitlik 3.15 ifadesindeki ilk üç terimin çarpımı $K_7(\mathbf{x})$ olarak hesaplanır. $K_m(\mathbf{x})$ ' i hesaplamak için çarpım taban fonksiyonu içindeki

herbir budanmış splaynın $\mathbf{x}'$ deki değerinin hesaplanmasını gerektirir. Eğer herhangi bir budanmış splayn hesabının $\mathbf{x}'$ deki değeri sıfır ise o zaman $K_m(\mathbf{x})'$ in de $\mathbf{x}'$ deki değeri sıfırdır. Başka bir deyişle, $K_m(\mathbf{x})'$ nin $\mathbf{x}'$ deki değeri, budanmış splaynların çarpımının $\mathbf{x}'$ deki değeridir. Varsayalım ki, $R_5 \in \mathfrak{R}^3$ için sıralanmış budanmış splaynlar $(1, 2, 5)$ olsun. Buna göre $r_2 = (2, 3)$ ve $r_5 = (-3, 1)$ olarak belirlensin. R_5 ile ilgili çarpım taban fonksiyonu (3.16) ile verilir:

$$K_5(\mathbf{x}) = T_{0,r_1}(\mathbf{x}) \times T_{1,r_2}(\mathbf{x}) \times T_{2,r_5}(\mathbf{x}) \quad (3.16)$$

$$= 1 \times (x_2 - 3)_+ \times (1 - x_3) \quad (3.17)$$

$$= \begin{cases} (x_2 - 3)(1 - x_3) & x_2 > 3, x_3 < 1 \\ 0 & \text{diğer durumlarda} \end{cases} \quad (3.18)$$

Eğer, $\mathbf{x} = \{5, 4, .5\} \in R_5$ ise o zaman $K_5(\mathbf{x}) = 0.5$ ve eğer $\mathbf{x} = \{4, 3.5, 6\} \notin R_5$ ise ve $K_5(\mathbf{x}) = 0$ dır.

Tahminleyici değişkenlerin etkileşiminin düzeyi ilişkili olduğu R_i içinde, $K_i(\mathbf{x})$ çarpım taban fonksiyonu içindeki budanmış splaynların ($T_{0,r_1}(\mathbf{x})$ hariç) sayısıdır. Tek terimli çarpım taban fonksiyonu, kendi tahminleyici değişkeniyle budanmış olan bir doğrusal ilişkiyi temsil ederken, iki terimli çarpım tabanı ikili etkileşimleri temsil etmektedir. Bir MARS modelinde etkileşim terimlerinin sayısı ve derecesi MARS algoritmasındaki etkileşim terimleri için belirlenen maksimum sayıya ve veri setine bağlıdır. MARS bilinmeyen $f(\mathbf{x})$ fonksiyonunu, (3.19) ile tahmin eder:

$$\hat{f}(\mathbf{x}) = \sum_{i=1}^S c_i K_i(\mathbf{x}) \quad (3.19)$$

Burada $\hat{f}(\mathbf{x})$ ilgili $\{R_i\}_{i=1}^S$ altbölgelerindeki çarpım taban fonksiyonları olan $\{K_i(\mathbf{x})\}_{i=1}^S$ lerin toplamsal bir fonksiyonudur. Bir çarpım taban fonksiyonu verildiğinde, bölüntüleme noktaları, ayrıca modelin parametreleri de olmak üzere, sabittirler. Eşitlik 3.19 ile verilen MARS modelinin katsayıları $\{c_i\}_{i=1}^S$ en küçük kareler regresyonu ile bulunur.

Yinelemeli bölüntüleme gibi, MARS'daki ileriye dönük algoritmanın

amacı, iteratif olarak katsayı değerleri vektörünü, altbölgeleri $\{R_i\}_{i=1}^M$ ($M \geq S$) belirlemeye çalışırken düzeltmek, ve bu bölgelerdeki çarpım taban fonksiyonları $f(\mathbf{x})$ 'i veriye yaklaştırmaktır. Yine yinelemeli bölüntülemedeki gibi ileriye dönük adım prosedürünü geriye dönük bir düzeltme süreciyle dengelemek gerekir. Böylelikle modeldeki fazladan $(M - S)$ tane altbölge çıkartılır [31].

3.2 Regresyon Splaynları

Regresyon splayn modelinin geliştirilmesi, MARS' ı açıklamada yeni bir metot olarak ortaya çıkmıştır. Silverman [33] splayn fonksiyonlarını parametrik ve nonparametrik regresyon metodolojisi arasında etkili bir yaklaşım olarak kabul etmiştir. Buna göre, $\mathbf{x} \in D \subset \mathbb{R}^1$ tanım bölgesi üzerinde, bilinmeyen c_l sabitleriyle tanımlanan q . dereceden(order) polinomiyal bir fonksiyon (3.20) ile verilmiştir:

$$p_q(\mathbf{x}) = \sum_{l=0}^q c_l \mathbf{x}^l \quad \mathbf{x} \in D \quad (3.20)$$

Eşitlik 3.20' teki polinomlar pürüzsüzdür. Ancak, veriye polinomlarla global olarak bir model uydurmada yüksek dereceden terimler yüzünden etkilerini kaybedebilirler. Bu durumda, D tanım bölgesini daha küçük R_i altbölgelerine farklı polinomlar kullanarak bölmek, her bir altbölgede daha düşük dereceden terimlerin olmasını sağlar ve yüksek dereceli terim sıkıntısından kurtulmuş olunur.

Varsayalım ki $[a, b] = D \subset \mathbb{R}^1$ ve $\Delta_S = \{t_1, \dots, t_{S-1}\}$ kümesi $[a, b]$ ' de sıralı bir bölüntü olsun. Bu bölüntü S tane ayrık altbölgeye $a = t_0 < t_1 < \dots < t_S = b$ ayrılsın. Her bir altbölge $R_i = [t_{i-1}, t_i]$, $i = 1, \dots, S$ ile gösterilsin. Varsayalım ki $C^q[D]$ kümesi D içindeki $q - 1$. türevleri de sürekli olan bütün sürekli fonksiyonların kümesi olsun.

$s_{\Delta_S}^q(\mathbf{x})$ fonksiyonu, S parçalı q . dereceden polinomiyal fonksiyonlar kümesinin, öyle ki q . dereceden fonksiyonunun ve ilk $q - 1$ türevleri bölüntü noktalarında çakışan, bir splayn fonksiyonu (3.21) olarak tanımlansın:

$$s_{\Delta_s}^q(\mathbf{x}) = \sum_{i=1}^S p_{q,i}(\mathbf{x}) I[\mathbf{x} \in R_i] \quad (3.21)$$

Burada, $s_{\Delta_s}^q(\mathbf{x}) \in C^q[D]$ kısıt olarak verilmiştir.

Splaynları regresyon içine koymak için pek çok yaklaşım önerilmiştir [34]. Bunlardan bir tanesi Eşitlik 3.1 modelini parçalı splaynlarla yazmaktır. Bu durumda (3.22) ile verilen parçalı regresyon splayn modeli elde edilir.

$$y = s_{\Delta_s}^q(\mathbf{x}) + \epsilon \quad (3.22)$$

Burada ϵ ' nun ortalamasının sıfır, varyansının σ_ϵ^2 ve $s_{\Delta_s}^q(\mathbf{x})$ den bağımsız olduğu varsayılır. Bu durumda gerçek $f(\mathbf{x})$ 'in tahmincisi olarak $s_{\Delta_s}^q(\mathbf{x})$ kullanılır.

Smith [35], Δ_S bölüntü noktaları kümesi verildiğinde, daha faydalı bir regresyon splayn modeli için (+) fonksiyonları kullanılmasını önermiştir. (+) fonksiyonu (3.23) ile verilir:

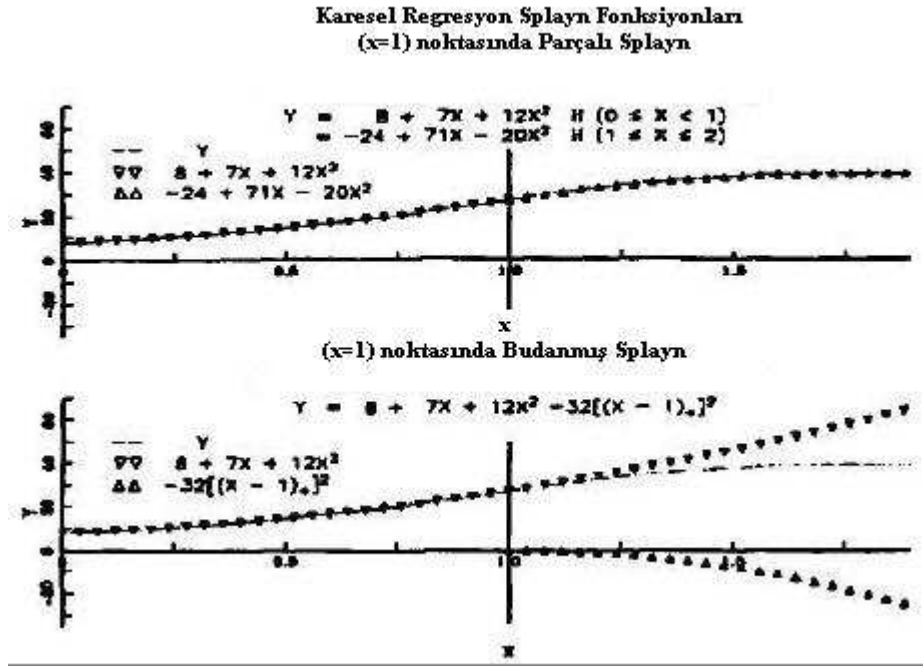
$$v_+ = \begin{cases} v & \text{eger } v > 0 \\ 0 & \text{eger } v \leq 0 \end{cases} \quad (3.23)$$

Varsayalım ki yine $[a, b] = D \subset \mathbb{R}^l$ olsun. Ancak, $\Delta_{S_0} = \{t_1, \dots, t_{S-1}\}$ bölüntüsü, $[a, b]$ nin sıralı bir bölüntüsü olsun ve S içindeki altbölgeleri $R_i = [t_{i-1}, t_S]$, $i = 1, \dots, S$ olarak tanımlansın. Tanım bölgesinin her altbölgesindeki polinomiyal terimlerinin derecesi l ve bir splayn modelindeki $(i + 1)$.nci altbölgedeki l . polinomiyal terim için ilgili katsayı c_{il} ile gösterilsin. (+) fonksiyonunun kullanımı sonucu, bir budanmış regresyon splaynı parçalı regresyon splayn modeline eşit olur. Bu durumda, (3.24) yazılabilir:

$$y = \sum_{l=0}^q c_l \mathbf{x}^l + \sum_{i=1}^{S-1} c_{iq} [(\mathbf{x} - t_i)_+]^q + \epsilon \quad q \geq 1 \quad (3.24)$$

Burada, ϵ' 'nün ortalamasının sıfır, varyansının $\sigma_{\epsilon'}^2$, $s_{\Delta_s}^q(\mathbf{x})$ 'den bağımsız ve q' 'nün 1'e eşit veya daha büyük olduğu varsayılır. Bölüntü noktaları kümesi Δ_{S_0} sıralı olduğundan, sıfırdan farklı değer alan budanmış splayn fonksiyonları sayısı her t_i bölüntü noktası boyunca sağa gittikçe bir artar.

Bu durumu Şekil 3.1 ile incelemek mümkündür.



Şekil 3.1: Karesel regresyon spline fonksiyonları

Şekil 3.1' de $q = 2$ dereceden bir tane parçalı (üstteki) ve bir tane budanmış splayn (alttaki) fonksiyonları verilmiştir. Bunlar $x = 1$ bölüntü noktasıyla ikiye ayrılırlar. Üstteki çizimde y eğrisi $0 \leq x \leq 2$ aralığında iki ayrı ikinci dereceden polinomiyal fonksiyonla gösterilmiştir. Diğer çizimde ise bir kısmı $\nabla\nabla$ ile $[0, 1)$ bölgesinde ikinci dereceden polinomiyal bir fonksiyon ve diğer kısmı da $\Delta\Delta$ sembolüyle $[1, 2]$ bölgesinde başka bir ikinci dereceden polinomla ifade edilen bir eğri vardır. Alt grafikteki y eğrisi de $[0, 1)$ bölgesinde ikinci dereceden bir polinomiyal fonksiyon olarak, $\nabla\nabla$ ile gösterilmiştir. Ancak ikinci kısımda y eğrisi, $[1, 2]$ bölgesinde iki tane ikinci dereceden polinomiyal fonksiyonun örtüşmesi ile oluşturulmuştur: $\nabla\nabla$ ile gösterilen ikinci dereceden polinomiyal bir fonksiyon ve $\Delta\Delta$ ile gösterilen budanmış ikinci dereceden

polinomiyal bir fonksiyon. Her iki parçalı ve budanmış splayn fonksiyonları, sırasıyla (3.21) ve (3.24), y eğrisini tanımlar.

Buradaki önemli nokta, daha önce de belirtildiği gibi bölüntü noktalarını $\{t_i\}_{i=1}^{S-1}$ yerlerinin ve değerlerinin belirlenmesidir. Bu belirlendikten sonra q . dereceden budanmış regresyon splayn modeli, (3.24), bu bölüntü noktalarıyla katsayıları sıradan en küçük kareler yöntemiyle bulunan doğrusal bir modeldir. Buradaki en büyük zorluk q . dereceden regresyon splayn modelini oluşturacak bölüntü noktalarının sayısı ve yeridir.

Şu ana kadar regresyon splayn modeli $\mathfrak{R}^1$ ' de tanımlandı. k , adet açıklayıcı değişken için bu durum tek değişkenli splayn fonksiyonların çarpımlarıyla elde edilebilir ($k>1$). Ancak bunu yapmak boyut sıkıntısı getireceğinden, MARS bu durumdan kurtulmak için modifiye edilmiş bir yinelemeli bölüntüleme yöntemi kullanarak tanım bölgesi için bölüntüler elde etmeye çalışır [31].

Uygulamalarda amaç bir yanıt değişkeninin bir veya daha fazla tahminleyici değişkenin $x_1, \dots, x_n$ verilen değerlerine, $\{y_i, x_{1i}, \dots, x_{ni}\}_1^N$ olan bağımlılığını modellemektir [36]. Veriyi üreten sistemin yapısı (3.25) ile verilmiştir:

$$y = f(x_1, \dots, x_n) + \epsilon \quad (3.25)$$

Burada $(x_1, \dots, x_n) \in D \subset R_n$ tanım bölgesi üzerinde gösterilebilir. Buradaki ϵ ortalaması sıfır olan stokastik bir bileşen ve modelde y 'ye ilişkin gözlenemeyen veya kontrol edilemeyen nitelikleri temsil eder. Regresyon analizinin amacı veriyi bir $\hat{f}(x_1, \dots, x_n)$ fonksiyonu ile temsil edebilmektir öyle ki; tanım bölgesi D üzerindeki $f(x_1, \dots, x_n)$ için uygun bir yaklaşık olarak da kullanılabilsin. Buradaki uygun ifadesiyle farklı yaklaşımlar kullanılabileceği anlatılmak istenmektedir. Bu anlamda uygulamalarda yaklaşımın doğruluğu ifadesi önem kazanmaktadır. Doğruluk yetersizliği (Lack of Accuracy) olarak ifade edilebilecek bu tanım (3.26) ve (3.27) ile verilmiştir:

$$I = \int_D w(\mathbf{x}) \Delta [\hat{f}(\mathbf{x}), f(\mathbf{x})] d\mathbf{x} \quad (3.26)$$

$$E = \frac{1}{N} \sum_{i=1}^N w(\mathbf{x}_i) \Delta [\hat{f}(\mathbf{x}), f(\mathbf{x})] d\mathbf{x} \quad (3.27)$$

Burada, $\mathbf{x} = (x_1, \dots, x_n)$, Δ bir fark ölçüsü ve $w(\mathbf{x})$ ise bir ağırlık fonksiyonudur. Eşitlik 3.26 ile verilen tanım bölgesi üzerinde doğruluğun ortalama bir yaklaşımı (the average accuracy of the approximation) iken, E beklenen değer ile ifade edilen tasarım noktaları $\mathbf{x}_1, \dots, \mathbf{x}_N$ üzerinde doğruluğun ortalama (average accuracy) bir ölçüsüdür.

Bunun yanında regresyon modellemede parametrik ve nonparametrik metodolojilerden bahsedilmesi de gerekmektedir. Buna göre global parametrik modelleme ve nonparametrik modellemeden bahsedilebilir. Global parametrik modelleme çoğunlukla en küçük kareler metoduyla parametrik bir $g(\mathbf{x} | \{\hat{a}_j\}_1^p)$ fonksiyonu kullanılarak (3.28) ve (3.29) elde edilir:

$$\hat{f} = g(\mathbf{x} | \{\hat{a}_j\}_1^p) \quad (3.28)$$

$$\{\hat{a}_j\}_1^p = \underset{\{a_j\}_1^p}{\operatorname{argmin}} \sum_{i=1}^N [y_i - g(\mathbf{x} | \{a_j\}_1^p)]^2 \quad (3.29)$$

Çoğunlukla kullanılan doğrusal parametrik fonksiyon, $p \leq n$, (3.30) ile gösterilir.

$$g(\mathbf{x} | \{\hat{a}_j\}_0^p) = a_0 + \sum_{i=1}^p a_i x_i \quad (3.30)$$

Bu parametrik yaklaşım esnekliğe izin vermez ve bu yöntemle doğru tahminler yapmak ancak tahminleyici fonksiyon gerçek fonksiyona yakın olduğunda mümkün olur. Bunun yanında parametrik modellerin yorumlaması ve hesaplaması kolaydır.

Küçük boyutlu çalışmalarda ($n \leq 3$), parametrik modelleme üç şekilde geliştirilebilir: parçalı ve lokal uyum (fitting) ve pürüzlülük ceza yaklaşımı metodları. Parçalı parametrik uyumdaki temel fikir, bilinmeyen f fonksiyonuna her biri D tanım bölgesinin farklı bir alt bölgesinde tanımlı pek çok basit fonksiyonla (genellikle düşük dereceli polinomlar olmak üzere) yaklaşımdır. Bu yaklaşımdaki kısıt her yerde süreklilik şeklinde ve bazen de düşük dereceli

türevlerde de süreklilik şeklindedir. Düzgünlük (smoothness) ve yaklaşımın esnekliği arasındaki denge veya uyum iki şeyle kontrol edilir. Bunlardan biri alt bölgelerin sayısı ve düşük dereceli türevlerin getirdiği alt bölge sınırlarındaki süreksizliktir. En çok tercih edilen parçalı polinomial uyum şekli splaynlardır. Parametrik fonksiyonların q dereceli (degree) ve bunların $q - 1$ dereceli türevlerinin sürekli olduğu ($q = 3$ durumu) polinomlar en çok tercih edilenlerdir. Bu süreç, q . mertebeden (order) splayn yaklaşımlarının gerdiği(oluşturduğu) uzayı oluşturan taban fonksiyonları oluşturmak ve verinin taban fonksiyonlarıyla olan açılımının katsayılarını sıradan en küçük karelerle tahminlemektir. Örneğin; doğru parçası üzerindeki K tane düğüm(knots) ile betimlenen $K + 1$ tane bölgeye sahip tek değişkenli durumda ($n = 1$), tek taban (3.31)' deki fonksiyonlarla gösterilir. [36]

$$1, \{x^j\}_1^q, \{(x - t_k)_+\}_1^K \quad (3.31)$$

Burada $\{t_k\}_1^K$ lar düğüm yerlerini göstermektedir($+$: bir ifadenin negatif değerleri için sıfır değerini temsil etmektedir). Bu ifade budanmış kuvvet tabanını temsil eder ve $K + q + 1$ boyutta q dereceli splayn fonksiyonları uzayını geren(oluşturan) tabanlardan biri olarak bilinir. Splaynların detaylı bir açıklaması Boor 1978' de verilmiştir [37].

Parçalı parametrik modellemenin büyük boyutlardaki uzantısı pratikte zor ama teoride aynıdır ($n > 2$). Bu zorluklar boyut sorunuyla ilgilidir. Splayn yaklaşımı durumunda, alt bölgeler genellikle n tane değişken üzerindeki $K + 1$ aralığın (K tane düğümün) tensör çarpımları olarak oluşturulur. Uygun global taban, (3.31) ifadesindeki gibi her değişkenle birleştirilmiş $K + q + 1$ tane olan taban fonksiyonları üzerinde bir tensör çarpımıdır. Bu da, ($K + q + 1$) n tane katsayının veriden tahminlenmesi demektir. Lokal parametrik yaklaşımlar (smoothers) (3.32) ile gösterilen bir form alırlar:

$$f(\hat{\mathbf{x}}) = g(\mathbf{x}|\hat{a}_j(\mathbf{x})_1^p) \quad (3.32)$$

Burada, g parametrik basit bir fonksiyondur. Global parametrik yaklaşımlardan farklı olarak, burada parametrelerin değerleri genellikle hesaplanan her

bir $\mathbf{x}$ noktasında farklılık gösterir ve lokal olarak ağırlıklı en küçük karelerle uyum (3.33) ile yapılır.

$$\{\hat{a}_j(\mathbf{x})\}_1^p = \operatorname{argmin} \sum_{i=1}^N w(\mathbf{x}, \mathbf{x}_i) [y_i - g(\mathbf{x}_i | \{a_j\}_1^p)]^2 \quad (3.33)$$

Burada ağırlık fonksiyonu $w(\mathbf{x}, \mathbf{x}')$, $\mathbf{x}'$ noktaları üzerindeki ağırlığı $\mathbf{x}'$ e yakın olacak şekilde yerleştiren bir fonksiyondur. Bu yaklaşımlarda en önemli problem ağırlık fonksiyonunun ve parametrik fonksiyon g 'nin belirlenmesidir. Bu açıdan en çok $g(\mathbf{x}|a) = a$ üzerinde çalışılmıştır [38–40]. Lokal parametrik metotları uygulanmanın çok boyutlu durumlardaki zorluğu Eşitlik 3.33 ile verilen uygun ağırlık fonksiyonunun seçiminde yatar. Bu da (3.25) ile verilen bilinmeyen f fonksiyonuna bağlıdır.

Pürüzlülük ceza yaklaşımı (3.34) ile tanımlanır [36].

$$f(\hat{\mathbf{x}}) = \operatorname{argmin}_g \left\{ \sum_{i=1}^N [y_i - g(\mathbf{x}_i)]^2 + \lambda R(g) \right\} \quad (3.34)$$

Burada $R(g)$, $g(\mathbf{x})$ fonksiyonundaki pürüzlülüğün büyümesiyle artan bir fonksiyoneldir. Minimizasyon $R(g)$ nın tanımlandığı bütün g ' ler üzerinden yapılır. λ parametresi, g 'nin pürüzlülüğü ile veriye uyumu arasındaki dengeyi temsil eder. En çok kullanılan pürüzlülük cezası, karesel Laplace integrali, (3.35) ile verilmiştir:

$$R(g) = \sum_{k=1}^n \sum_{l=1}^n \int \left| \frac{\partial^2 g}{\partial x_k \partial x_l} \right|^2 d\mathbf{x} \quad (3.35)$$

Bu ifade Laplace düzeltme (ince tabakalı) splayn yaklaşımlarına $n \leq 3$ için bir gösterimdir. $n > 3$ için genel ince tabakalı splayn cezasının 2' den daha büyük mertebeden türevleri içeren daha karmaşık bir yapısı vardır [41].

Bunların dışında az boyuttaki nonparametrik modelleme yaklaşımları, çok boyuttaki modellemenin yetersizliğini giderebilmektedir [36]. Bu durum (3.36) ile ifade edilebilir.

$$\hat{f}(\mathbf{x}) = \sum_{j=1}^J \hat{g}_j(\mathbf{z}_j) \quad (3.36)$$

Burada her bir $\mathbf{z}_j$, $\{x_1, \dots, x_n\}$ den oluşan önceden seçilmiş küçük bir kümeyi temsil etmektedir. $\{\mathbf{z}_j\}_1^J$ değişken kümesi seçildikten sonra, uygun $\{\hat{g}_j(\mathbf{z}_j)\}_1^J$ fonksiyon tahminleri nonparametrik metotlarla elde edilir. Bu durum (3.37) ile verilir:

$$\{\hat{g}_j(\mathbf{z}_j)\}_1^J = \underset{\{g_j\}}{\operatorname{argmin}} \sum_{i=1}^N \left[y_i - \sum_{j=1}^J g_j(\mathbf{z}_{ij}) \right]^2 \quad (3.37)$$

Burada pürüzsüzlük kısıtları $\hat{g}_j$ fonksiyonları üzerinden kurulur. Parçalı polinomiyaller (splaynlar) durumunda, uygun bir taban her bir $\mathbf{z}_j$ için oluşturulur ve çözüm tüm taban fonksiyonlarının bir birleşimi üzerinden en küçük kareler metoduyla elde edilir [42]. Ceza terimlerinin eklenmesiyle (3.36) yeniden elde edilir ve (3.38) gösterilebilir:

$$\hat{f}(\mathbf{x}) = \underset{\{g_j\}}{\operatorname{argmin}} \left\{ \sum_{i=1}^N \left[y_i - \sum_{j=1}^J g_j(\mathbf{z}_{ij}) \right]^2 + \sum_{j=1}^J \lambda_j R(g_j) \right\} \quad (3.38)$$

Herhangi bir az boyutlu nonparametrik fonksiyon kestiricisi backfitting algoritmasıyla bağlantılı olarak (3.38)'daki ifadeyi çözmek için kullanılabilir [30, 43, 44]. Bu süreç iteratif olarak $\hat{g}_j(\mathbf{z}_j)$ 'leri yeniden tahminler.

$$\hat{g}_j(\mathbf{z}_j) \leftarrow \underset{g_j}{\operatorname{argmin}} \sum_{i=1}^N \left[\left(y_i - \sum_{k \neq j} g_k(\mathbf{z}_{ik}) \right) - g_j(\mathbf{z}_{ij}) \right]^2 \quad (3.39)$$

Bu tahminleme (3.39)'da verildiği gibi yakınsaklık sağlanana kadar devam eder.

3.3 Uyarlamalı Regresyon Splaynları

Çok boyutlu çalışmalarda, fonksiyonların tahminleri uyarlamalı bir hesaplama yapılır. İstatistikte kullanılan uyarlamalı algoritmalar iki konuda yapılmıştır. Bunlardan ilki yinelemeli bölüntüleme [29, 30] ve "project pursuit" algoritmasıdır [43]. MARS yaklaşımının kolay anlaşılması açısından yinelemeli bölüntüleme regresyonun (Recursive Partitioning Regression) bilinmesi önerilir.

Aslında RPR adımsal regresyon (stepwise regression) süreci gibi görülebilir. Yinelemeli bölüntüleme regresyon modeli (3.40) ile verilebilir [36].

Eğer, $x \in R_m$ ise,

$$\hat{f} = g_m(x | \{a_j\}_1^p) \quad (3.40)$$

olur. Burada, $\{R_m\}_1^M$ ler D tanım bölgesinin bir bölüntüsünü (partition) temsil eden ayrık alt bölgelerdir. g_m fonksiyonları basit parametrik fonksiyonlar olarak alınır. En çok kullanılanı, (3.41) ile verilen sabit fonksiyondur [29, 30].

$$g_m(x | a_m) = a_m \quad (3.41)$$

Temel olarak, bölgelerin geometrik kavramını (3.40) ve (3.41)' deki ifadelerle eşit olacak şekilde yer değiştirerek, bölgeleri toplama ve çarpma işlemleriyle ayırarak bir model oluşturmaktır. Buna göre, (3.40) ve (3.41)' deki modele yaklaşık yapıda bir model uydurmak için (3.42)' deki taban fonksiyonları kümesi kullanılır.

$$\hat{f} = \sum_{m=1}^M a_m B_m(x) \quad (3.42)$$

B_m taban fonksiyonları Eşitlik 3.43 ile gösterilen yapıdadır.

$$B_m(x) = I[x \in R_m] \quad (3.43)$$

Burada, I gösterge fonksiyonu olarak bilinir. Buna göre, I , belirttiği argüman doğrultusunda doğruysa 1 değilse 0 değerini alan fonksiyondur. $\{a_m\}_1^M$ ler veriye en iyi uyumu sağlayan değerlerdir. $\{R_m\}_1^M$ ler (3.40) ve (3.41)' de verilen değişken uzayının oluşturduğu aynı alt bölgelerdir. Bu bölgeler ayrık olduğundan, (3.42) ve (3.43) ifadeleri (3.40) ve (3.41)'ye eşittir. Yinelemeli bölüntülemedeki amaç sadece, veriye en iyi uyacak katsayı değerlerini bulmak değildir, ayrıca taban fonksiyonlarının (subregion) da iyi bir kümesini üretebilmektir. Bilinmeyen f fonksiyonu taban fonksiyonları $B_k(x)$ lerin doğrusal bir bileşkesi olarak (3.44) gibi yazılabilir.

$$\hat{f}(x) = a_0 + \sum_{m=1}^M a_m B_m(x) \quad (3.44)$$

Burada, $a_0, a_1, \dots, a_k$ lar modelin katsayılarıdır. Her bir taban fonksiyon, budanmış(truncated) kuvvet splayn fonksiyonu olarak (3.45)' deki gibi yazılabilir.

$$B_m(x) = \prod_{k=1}^{K_m} H[s_{km}(x_{v_{k,m}} - t_{k,m})] \quad (3.45)$$

Burada, K_m ; etkileşimin derecesini, s_{km} ; (+1) veya (-1) değerlerini alır(Dalların sağa veya sola açılımını), $x_{v_{k,m}}$; bağımsız değişkenin değerini, $t_{k,m}$; düğüm değerini temsil eder. Eşitlik 3.45 ifadesindeki basamak fonksiyonları (3.14)' deki gibi splaynlarla yer değiştirerek süreksizlik durumu ortadan kalkar.

3.3.1 Model Seçimi

İleriye dönük adım MARS algoritması maksimum taban fonksiyonları sayısına, M_{max} , ulaşana kadar devam eder. Burada önemli olan bu sayıyı optimal olandan yeterince daha büyük seçmek ve fazlalık olan taban fonksiyonlarını silmektir. Bu süreçte M_{max} potansiyel olarak seçilebilecek/silinebilecek bütün değişkenlerin sayısını temsil etmektedir. Bu stratejinin olmasının nedeni aslında ileriye dönük adım algoritmasından gelmektedir. Her iterasyonda yeni iki taban fonksiyonu üretir ve bunu bir önceki iterasyonlarda oluşturulan fonksiyonları kullanarak yapar. Sonuçta, önceki taban fonksiyonlarının katkısı çok önemli olmayabilir. Bunların asıl katkısı diğer taban fonksiyonlarını üretmeye yardımcı olmalarıdır. Mümkün olan daha yüksek dereceden taban fonksiyonları çarpımlarını elde etmek için yeterli durum ileriye dönük adım algoritmasındaki mücade edilen taban fonksiyonu sayısına bağlıdır.

Bu model seçimini yapabilmek için bir kritere ihtiyaç duyulur. Geriye dönük adım algoritmada kullanılan bu kriteri minimum yapan model, son model olarak belirlenir. MARS sürecinde çapraz geçerlilik kriteri, (3.46) ile verilir:

$$CV(M) = \frac{1}{N} \sum_{i=1}^N [y_i - \hat{f}_{M/i}(x_i)]^2 \quad (3.46)$$

Burada kriter, taban fonksiyonları sayısına M' ye bağlıdır. Bu eşitlikteki $\hat{f}_{M/i}$ ifadesi geriye dönük adım algoritmadaki silme süresinde göz önüne alınan ve i .nci gözlem silindikten sonra hesaplanan M tane taban fonksiyonu modelidir. Genelleştirilmiş çapraz geçerlilik kriteri ilk olarak Craven ve Wahba [13] tarafından önerilmiştir. Eşitlik 3.47, MARS' da kullanılan bu kriterin modifiye edilmiş halini gösterir [36].

$$GCV(M) = \frac{1}{N} \sum_{i=1}^N [y_i - \hat{f}_M(\mathbf{x}_i)]^2 / [1 - \frac{C(M)}{N}]^2 \quad (3.47)$$

Oluşturulan maksimum modelin budanmasından sonra hata kareler toplamının en küçükleyen model araştırılır. Budama algoritması genelleştirilmiş çapraz geçerlilik algoritması ile yapılır. Burada paydaki ifade uyum eksikliğini ve paydadaki ise model karmaşıklığına karşılık gelen bir ceza ifadesini temsil eder. veriye uygun modelleme yapıldığında, açılımın katsayı sayısını içerir.

$C(M)$ ifadesi gerçekte modeldeki efektif parametrelerin sayısıdır. Bu parametre, uydurulan modeldeki terim sayısı ile düğüm noktalarının optimal pozisyonlarının seçiminde kullanılan parametre sayısının toplamıdır. Bazı simülasyonlar göstermiştir ki; parçalı doğrusal regresyonda bir düğüm noktası seçmenin 3 serbestlik derecesi ile cezalandırılması gerekir [45]. Friedman' nın burada yaptığı katkı (3.48) ile verilir:

$$\tilde{C}(M) = M.(d/2 + 1) + 1 \quad (3.48)$$

Eşitlik 3.48 ile tahminlenmiş modelin doğrusal serbestlik derecesi hesaplanmaktadır. Buradaki M , $\hat{f}_M(\mathbf{x})$ modelindeki sabit olmayan taban fonksiyonu sayısıdır. d ifadesi ise her iterasyonda oluşturulan modelin karmaşıklığına getirilen cezadır ve düzeltme parametresi gibi düşünülebilir. Her bir taban fonksiyonu için katkısı $d/2$ ' dir. Simülasyon çalışmaları göstermiştir ki, modifiye edilmiş GCV_m kriteri için d değeri 3 olarak kabul edilmiştir [46].

Bir başka önemli nokta, düğümler arasında kalacak gözlem değerleri sayısına

karar vermektir. Friedman, bunun için (3.49)' u önermiştir [36].

$$L^* = -\log_2\left(\frac{-1}{nN_m} \ln(1 - \alpha)\right) \quad (3.49)$$

Burada N_m , taban fonksiyonu B_m içindeki sıfırdan farklı gözlemlerin sayısıdır ($B_m > 0$). n , bağımsız değişken sayısı ve nN_m ise, bu şekildeki gözlemlerin uygun sayısıdır, çünkü döngü içerisinde değişmektedir ($\alpha = 0.05$).

3.3.2 Değişken Önemliliği

Bu bölümde, kavram olarak yeni ancak MARS algoritmasının ürettiği "Değişken Önemlilik" tanımlamasından bahsedilecektir. Değişken önemlilikleri hesaplanırken MARS, önemliliği araştırılan değişken dışındaki bütün değişkenleri dışarda tutar ve tekrar uyum iyiliğini araştırır ve buna göre her bir değişken için skor değerleri belirler. Bu değerler modelde yer alan değişkenlerin diğerlerine göre göreceli katkılarıdır.

3.4 MARS-İleriye Dönük Algoritma

Algoritmada $R_1 = V$ başlangıç olarak belirlenir. Ancak, MARS R_m ve R_{m+1} olmak üzere iki tane yeni altbölge oluşturmaya çalışırken ana bölge R_{i*} ' ı her bölüntülemeye dikkate alır. Ayrıca, MARS her bir tahminleyici değişken için, her sıralı budanmış splayn'nin birden fazla bölüntüye sahip olmasını engeller. Çünkü bu durum doğrusal olmayan splayn fonksiyonu yaratır. MARS, bir sonraki altbölge R_i nin en iyi bölüntülenmesi için bu kısıttan yararlanır ve çarpım taban fonksiyonu $K_i(\mathbf{x})$ içinde herhangi bir tahminleyici değişken için zaten var olan bir bölüntüleme noktasını dışarda bırakır.

Yinelemeli bölüntüleme ve MARS arasındaki en dikkat çekici fark, MARS algoritmasının modeli oluşturmasındaki farklılıktan ileri gelir. Friedman [32]' de verildiği gibi, (3.15) ile verilen çarpım taban fonksiyonu $\{K_i(\mathbf{x})\}_{i=1}^m$ ve (3.14) ile verilen budanmış splaynlar $T_{i,r_m}(\mathbf{x})$ ve $T_{i,r_{m+1}}(\mathbf{x})$ ' i taban fonksiyonları $\{B_i(\mathbf{x})\}_{i=1}^m$ ve basamak fonksiyonları $H[t - x_v]$ ve $H[x_v - t]$ ile yer değiştirir.

Bu basamak fonksiyonlarından daha önce (3.11)' de olduğu gibi ileriye dönük adım yinelemeli bölüntüleme algoritmasında bahsedilmiştir (Satır 7);

- (1) $R_1 = V, T_{0,r_1}(\mathbf{x}) = 1$
- (2) For R_m için, $m = 2$ to M do
- (3) $lof^* = \infty, i^* = 0, v^* = 0, t^* = 0$
- (4) For R_i için, $i = 1$ to $m - 1$
- (5) For $x_v \in R_i$ için $v = 1$ to p do (öyle ki, $v \neq K_i(\mathbf{x})$)
- (6) For $x_{v,k} \in R_i$ için $t = x_{v,k=1}$ to $x_{v,k=n}$ (Her veri noktası)
- (7) $g = (\sum_d c_d K_d(\mathbf{x})) + c_m K_i(\mathbf{x}) T_{i,r_m}(\mathbf{x}) + c_{m+1} K_i(\mathbf{x}) T_{i,r_{m+1}}(\mathbf{x})$
- (8) $lof = Lof_m$
- (9) If $lof < lof^*$ then $lof^* = lof; i^* = i; v^* = v; t^* = t$ end if
- (10) end for
- (11) end for
- (12) end for
- (13) $R_m \leftarrow \{R_{i^*} : (t^* - x_{v^*}) > 0\}$
- (14) $R_{i^*} \leftarrow \{R_{i^*} : (x_{v^*} - t^*) \geq 0\}$
- (15) $m \leftarrow m + 2$
- (16) end for
- (17) end

İleriye dönük algoritmayı daha iyi anlamak için, tahminleyici değişken sayısının 3 ve ileriye dönük maksimum bölüntü sayısının $M = 5$ olduğunu varsayalım. MARS algoritması yinelemeli bölüntüleme algoritmasına daha önce bahsedilen modifikasyonlar haricinde benzer bir işleyiştir. MARS algoritmasının başlaması için tanım bölgesinin ilk altbölge olarak $V = R_1$ ataması gerekir. $f(\mathbf{x})$ ' in tek bölge MARS tahmini, yinmeli bölüntüleme gibi

$$\hat{f}(\mathbf{x}) = c_1 K_1(\mathbf{x}) = c_1 T_{0,r_1}(\mathbf{x}) = c_1 = \frac{1}{N} \sum_{i=1}^N y_i \quad (3.50)$$

ile gösterilir. Yine varsayalım ki ilk iterasyondaki araştırma R_1 ' in $t^* = x_{2,25}$ noktasında bölüntülenmesine karar versin. Bu şekilde devam ederek $f(\mathbf{x})$ ' in üç altbölge MARS tahmini ikinci adımda (ilk bölüntüleme $t^* = x_{2,25}$ nok-

tasında), $T_{0,r_1}(\mathbf{x}) = 1$ iken (3.51), (3.52) ve (3.53) ile

$$\hat{f}(\mathbf{x}) = c_1 K_1(\mathbf{x}) + c_2 K_2(\mathbf{x}) + c_3 K_3(\mathbf{x}) \quad (3.51)$$

$$= c_1 T_{0,r_1}(\mathbf{x}) + c_2 T_{0,r_1}(\mathbf{x}) T_{0,r_2}(\mathbf{x}) + c_3 T_{0,r_1}(\mathbf{x}) T_{0,r_3}(\mathbf{x}) \quad (3.52)$$

$$= c_1 + c_2 (t^* - x_2)_+ + c_3 (x_2 - t^*)_+ \quad (3.53)$$

verilir. Burada,

$$\mathbf{x} \in \begin{cases} R_1 & \text{eğer} & \mathbf{x} \in V \\ R_2 & \text{eğer} & x_2 < x_{2,25} \quad \text{ve} \quad \mathbf{x} \in R_1 \\ R_3 & \text{eğer} & x_2 \geq x_{2,25} \quad \text{ve} \quad \mathbf{x} \in R_1 \end{cases}$$

olur. MARS ileriye dönük adım algoritmasının bir sonraki iterasyonunda, R_1, R_2 ve R_3 altbölgelerindeki en iyi bölüntüleme noktasını araştırır ve yinelemeli bölüntülemeye olduğu gibi, tek bir istisna ile, üç altbölge içindeki her bir tahminleyici değişken için bütün potansiyel bölüntüleme noktalarının hesabından sonra bunu yapar. Farklı olarak, x_2 üzerindeki bir başka bölüntünün R_2 veya R_3 içinde olması engellenir. Çünkü bu durumda derecesi $q > 1$ olan budanmış splayn fonksiyonu üretilmiş olur.

$M = 5$ iken, MARS ileriye dönük adım algoritması tanım bölgesi V içinde ikinci bir bölüntülemeden sonra tamamlanmış olacaktır. $f(\mathbf{x})$ ' in son MARS tahmini (3.51) ile verilen bütün terimleri ve ek olarak ikinci bölüntülemeyle gelen iki terimi daha kapsayacaktır. Bu durumda model eğer ikinci bölüntüleme R_1 ' in içinde olursa 5 tane tekli çarpım taban splayn fonksiyonlarını ($T_{0,r_1}(\mathbf{x})$ hariç) içerecektir. Eğer ikinci bölüntüleme R_2 veya R_3 de olursa model, üç tane tek terimli çarpım splayn fonksiyonu ve 2 tane çift çarpım taban splayn fonksiyonları içerecektir.

Geriye dönük düzeltme sürecinden sonra MARS modeli (3.19) ile verildiği gibi $K_1(\mathbf{x})$ çarpım taban splayn fonksiyonunun katsayısı olan c_1 ve MARS'ın

geriye dönük adım altbölge eleme sürecinden gelen çarpım taban fonksiyonları ve katsayıları kalan terimlerinden oluşur. Friedman(1988), tahminleyici değişkenlerin ve onların ilişkilerini ayrı ayrı görebilmek için $f(\mathbf{x})$ ' in MARS tahmininin varyans analizi parçalanmasını (3.54) ile göstermiştir.

$$\hat{f}(\mathbf{x}) = c_1 + \sum_{Z=1} c_i K_i(\mathbf{x}) + \sum_{Z=2} c_i K_i(\mathbf{x}) + \dots \quad (3.54)$$

Burada Z çarpım taban fonksiyonu, $T_{0,r_1}(\mathbf{x})$ dışında kalan $\{K_i(\mathbf{x})\}_{i=1}^S$ 'deki budanmış splaynların sayısını gösterir. Bu metot ilgilenilen değişkenler tarafından $\hat{f}(\mathbf{x})$ ' e olan bütün ve her türlü katkıları içine alır. $Z = 1$ indeksli çarpım taban fonksiyonları budanmış doğrusal etkileri ve $Z = 2$ indeksli çarpım taban fonksiyonları ise budanmış çift etkileşimleri gösterir [31].

Buraya kadar anlatılanları kısaca özetlemek gerekirse, ileriye dönük algoritma ile mümkün olan tüm temel fonksiyonlar oluşturulur. Modelin karmaşıklığı maksimum seviyeye ulaşınca kadar bu işlem devam eder.

3.5 MARS-Geriye Dönük Algoritma

İleriye dönük algoritma ile elde edilen modelin budanması işlemi geriye dönük algoritma ile yapılır. Buna göre algoritmanın adımları aşağıda sıralanmıştır:

- (1) $J^* = \{1, 2, \dots, M_{max}\} ; K^* \leftarrow J^*$
- (2) $Lof^* \leftarrow \min_{\{a_j | j \in J^*\}} \text{LOF} (\sum_{j \in J^*} a_j B_j(\mathbf{x}))$
- (3) For $M = M_{max}$ to 2 do: $b \leftarrow \infty ; L \leftarrow K^*$
- (4) For $m = 2$ to M do: $K \leftarrow -\{m\}$
- (5) $Lof \leftarrow \min_{\{a_k | k \in K\}} \text{LOF} (\sum_{k \in K} a_k B_k(\mathbf{x}))$
- (6) if $Lof < b$, then $b \leftarrow Lof ; K^* \leftarrow K$ end if
- (7) if $Lof < Lof^*$, then $lof^* \leftarrow ; J^* \leftarrow K$ end if
- (8) end for
- (9) end for
- (10) end

Burada ilk satırda döngü, ileriye dönük algoritmada oluşturulan taban

fonksiyon kümesiyle oluşturulmuş modelle başlanır. En dıştaki For döngüsüyle, döngünün her tekrarında her bir taban fonksiyon silinmektedir. İçerdeki For döngüsü ise bunun hangisi olduğunu seçmektedir. Bu taban fonksiyonun silinmesiyle model uyumunda en az etkili olan taban fonksiyonu ortadan kalkar ve modelin uyumu artar. Burada dikkat edilmesi gereken başka bir nokta $B(\mathbf{x}) = 1$ sabit fonksiyonunun silinme durumu yoktur. Bu algoritma ile $M_{max} - 1$ tane model oluşturulur. Her bir modelin taban fonksiyon sayısı bir önceki modeldekinden bir tane daha azdır. En iyi modele bu şekilde ilerleyen bir algoritma ile ulaşılır.

4 UYGULAMA

4.1 Analizde Kullanılan Verilerin Yapısı

Uygulamada kullanılan veriler Eskişehir'deki tramvay hatlarından elde edilmiştir. Çalışmada, Anadolu Üniversitesi Çevre Sorunları Uygulama ve Araştırma Merkezi'nce tramvay ve araçların çevreye kattığı kirlilik miktarları araştırılmak üzere toplanan toz ve toprak verilerinden yararlanılmıştır. Toz ve topraktan elde edilen ağır metal ölçümleri; Cd (Kadmiyum), Cr (Krom), Cu (Bakır), Fe (Demir), Mn (Manganez), Ni (Nikel), Pb (Kurşun) ve Zn (Çinko) olmak üzere sınıflandırılmıştır. Toprak parçaları örnekleri yaklaşık olarak 25gr kadar toplanmış, bunlar mekanik bir akik tezgahında $200\mu m$ 'den küçük parçalar kalıncaya dek tutulmuştur. Örnek noktalarının koordinatları GPS cihazıyla kaydedilmiştir. Bu toprak örnekleri şehir içi yoldan, üst topraktaki 0-10cm'lik katmanlardan toplanmıştır. Bütün toprak örnekleri 3 saat boyunca 105 derecede kurutulmuş, 0.5mm kalınlıkta naylon bir elekten geçirilmiş ve reaksiyon teknelerine gönderilmiştir. Bir sonraki bölümde bu veri seti kullanılarak yapılan regresyona dayalı sınıflandırma anlatılacaktır.

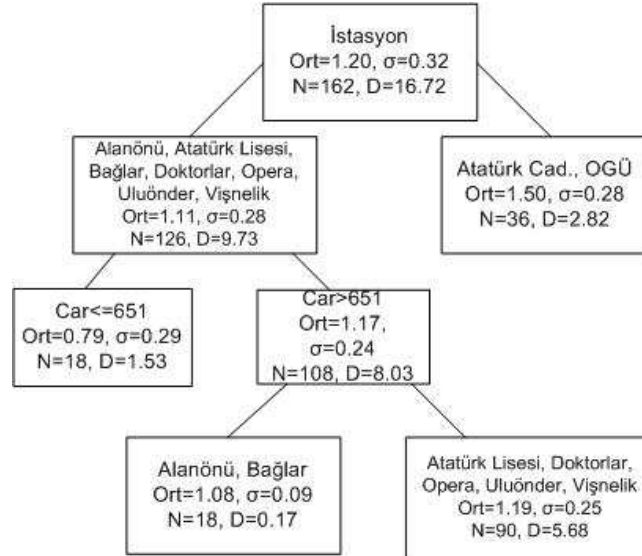
4.2 YYM ve Regresyon Ağaçları

Ağır metallerin tümü toz ve toprak kirleticileri olarak düşünülmüş, bağımlı değişkenler olarak alınmış ve tramvay ve araçların belirtilen hatlarda yaptığı kirlenme sonuçları kaydedilmiştir. Bu amaçla ilk olarak, belirlenmiş tramvay ve araç güzergahlarından; Opera, Çarşı, OGÜ, Atatürk Lisesi, Atatürk Caddesi, Bağlar, Alanönü, Vişnelik ve Doktorlar caddesinden elde edilen toz ve toprak örneklerinden ağır metal ölçümleri alınmıştır. Toz ve toprak örneklerinden elde edilen ağır metallerin hepsi kullanılmak üzere, toplam 16 adet regresyon ağacı oluşturulmuştur. Analizlerde kullanılan veri sayısı her bir ağır metal incelemesi için 162 adettir. Burada amaçlanan hedef, toz ve toprak kirliliğine en çok katkısı olan hatları ortaya çıkarmaktır. Bu çalışmada toprak verisi kullanılarak yapılan regresyon ağaçları analizi Bölüm 4.2.1 ile toz verisi kullanılarak yapılan regresyon ağaçları analizi ise Bölüm 4.2.2' de verilmiştir.

Bölüm 4.2’ de oluşturulan tüm regresyon ağaçları modelleri R Yazılımı kullanılarak elde edilmiştir [20].

4.2.1 Toprak Verisi Regresyon Ağacı Analizi

Bu bölümde regresyon ağaçları, toprak kirliliğinden etkilenen hatları ortaya çıkarmak için kullanılmıştır. Toprak kirliliğini modellemek için Bölüm 4.1’de adı geçen sekiz ağır metal kullanılmıştır. Her bir ağır metalin, tramvay ve araç sürekli değişkenleri ile istasyon kategorik değişkeni arasındaki ilişkisi yinelemeli bölüntüleme algoritması ile incelenmiştir. Bu algoritmanın bölüntüleme için kullandığı ölçüt veri setindeki her bir değerın olası bölüntü noktası olması düşüncesiyle, her bir model için hesaplanan sapma miktarıyla, $\sum(y - \bar{y})^2$, ölçülmüştür. Buna göre, Cd için elde edilen regresyon ağacı Şekil 4.1 ile verilmiştir.



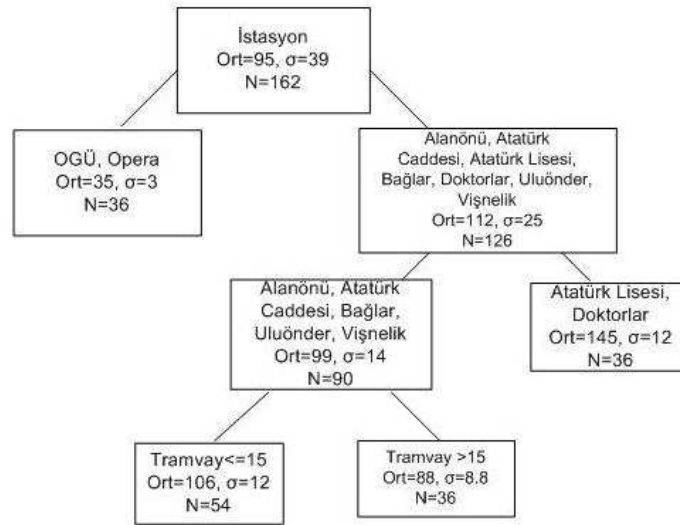
Şekil 4.1: Toprak örneği- Cd için regresyon ağacı

Burada, regresyon ağacı en yukarıdan en aşağıya doğru açıklanır. En tepedeki değişken algoritmanın bulduğu en önemli değişken olarak düşünülür, bu yüzden ilk bölüntüleme bu değişkenden başlamıştır. Buna göre, istasyon değişkeni 162 birimi, 126 ve 36 olmak üzere iki parçaya ayırmıştır. Sağdaki parçaya ait sapma değeri, $D = 2.82$ olarak bulunmuştur. Soldaki parçanın

sapma değeri, $D = 9.73$ olarak bulunmuştur. İlk bölüntüleme sol parçanın tekrar bölünmesiyle devam etmiştir. İkinci bölüntümede araç değişkeni 651 düğüm noktasından ikiye ayrılmıştır. Başka bir deyişle araç değişkeni için 651 değeri önemli bulunmuştur. Bu durum araç sayısının 651' den fazla olduğu durumlarda Cd kirliliğinin daha çok ortaya çıktığı yönünde yorumlanır. Son bölüntüleme de ise araç değişkeninin 651'den fazla olduğu istasyonlar arasında yeniden bölüntüleme yapılmıştır. Buna göre, Atatürk Lisesi, Doktorlar caddesi, Opera, Uluönder ve Vişnelik istasyonları Cd kirliliğine maruz kalmaktadır [20].

4.2.2 Toz Verisi Regresyon Ağacı Analizi

Benzer şekilde, toz örneklerinden elde edilen ağır metaller regresyon ağaçlarıyla modellenmiştir. Çalışmada toz kirliliğini modellemek için Bölüm 4.1' de belirtilen sekiz ağır metal kullanılmıştır. Her bir ağır metalin tramvay ve araç sürekli değişkenleri ile istasyon kategorik değişkeni arasındaki ilişkisi yinelemeli bölüntüleme algoritması ile incelenmiştir. Buna göre, Zn kirliliğine ilişkin yinelemeli bölüntüleme sonuçları Şekil 4.2 ile verilmiştir.



Şekil 4.2: Toz örneği- Zn için regresyon ağacı

Benzer şekilde, ağacın tepesinde bulunan değişkenin en önemli değişken

olarak kabul edildiğinden bahsedilmişti. İstasyon değişkeni, tramvay ve araç değişkenlerinden daha fazla değişkenlik gösterdiği için ilk bölüntülemeye yer almıştır. Buna göre, sol parçanın ortaya çıkardığı sonuç oldukça az sayıda veri içermektedir. Burada, OGÜ ve Opera istasyonlarına ilişkin Zn kirliliği ortalama değeri, $\bar{y} = 35$ olarak hesaplanmıştır. Sağ parçadaki ortalama değer, $\bar{y} = 112$ soldakine göre daha yüksektir. Bu durumda bu parçanın yeniden bölünmesi gerektiği düşünülebilir. Ancak daha önce de belirtildiği gibi bölüntünün, her bir veri noktasının olası düğüm noktası gibi varsayıldığı ve buna göre modelin sapmasından yola çıkılarak düğüm noktasına karar verilen bir yapıda olduğu unutulmamalıdır. Dolayısıyla, bölüntüler yanıt değişkeninin ortalama değerinin büyük olmasıyla ilgili olarak değil modelin sapmasıyla ilgili olarak yapılmaktadır. Ancak burada, sapma değerlerinin çok büyük olmaları yüzünden ilgili şekilde bunlara yer verilmemiştir. İlk bölüntüleme sağı parçanın tekrar ikiye bölünmesi durumu gözlenmiştir. Buna göre, Atatürk Lisesi ve Doktorlar istasyonlarındaki ortalama Zn kirliliği, $\bar{y} = 145$ olarak bulunmuştur. Ancak diğer taraftan ikinci bölüntüdeki sol parçanın yeniden ikiye bölündüğü gözlenmiştir. Bu durum, bölüntü değişkenin tramvay olması açısından önemlidir. Buna göre, tramvay sayısının 15 den az olduğu istasyonlarda bile Zn kirliliğine ait ortalama olarak $\bar{y} = 106$ olarak bulunmuştur.

Bu çalışmaların sonunda elde edilen sonuçlar Şekil 4.3' teki gibi haritalandırılmıştır.

Şekil 4.3 ile toprak ve toz örneklerinden elde edilen ağır metallerin, regresyon ağaçları yardımıyla yapılan modellemeler sonucunda hangi istasyonlar üzerinde kirlilik yarattıkları gösterilmiştir.

Bölüm 4.2.1' de verilen toprak çalışmasına ve Bölüm 4.2.2' de verilen toz çalışmasına ilişkin Cr, Cu, Fe, Mn, Ni, Pb ve Zn değişkenlerinin regresyon ağaçları da oluşturulmuştur. Benzer açıklamaların yer alacağı düşünülerek, diğer ağır metallere ilişkin regresyon ağaçları sonuçları bu tezin kapsamında verilmemiştir.

Modelleme için bağımlı değişken olarak ağır metallerden Cd, Zn ve Pb toprak kirleticileri olarak düşünülmüş, tramvay ve araçların belirtilen hatlarda yaptığı kirlenme sonuçları kaydedilmiştir. Diğer ağır metallerin incelemeye dahil edilmemesinin sebebi örneğin, Fe (Demir) için, Eskişehir'in bölge olarak bu ağır metal kirliliğine maruz kaldığının zaten bilinmesidir. Buna göre, Cd, Zn ve Pb ağır metalleri üzerinde, tramvay ve araç değişkenlerinin etkisini araştırmak için, 3^2 faktöriyel tasarım oluşturulmuştur.

Bu deneysel tasarım için kullanılan veriler Çizelge 4.1' de verilmiştir.

Çizelge 4.1: 3^2 Faktöriyel tasarımı	
Faktör x_1 -Tramvay	Faktör x_2 -Araç
12	174
18	852
24	1530

Çizelge 4.1' de görüldüğü gibi, iki faktörün üç düzeyli kombinasyonu ile elde edilen tasarım matrisinde, x_1 ; tramvay için kodlanan, x_2 ; araç için kodlanan terimlerdir. Burada, ilk sütunda tramvaya ait düzeyler 12, 18 ve 24 olarak verilmiştir. Burada ilgili istasyonlardan geçen tramvay sayıları verilmiştir. Benzer şekilde araç değişkenine ait sayısal değerler ikinci sütunda verilmiştir. Araştırmada 2 faktör, bunların 3' er düzeyi ve her biri düzeyin 5 tekrarı olduğundan, her bir ağır metal için toplam 45' şer veri kullanılmıştır.

Her bir ağır metal için 3^2 deneysel düzenden veriler toplanmıştır. Daha sonra tramvay ve araç değişkenlerin, Cd, Zn ve Pb bağımlı değişkenleri üzerindeki etkileri, birinci ve ikinci dereceden yanıt yüzeyi modelleriyle incelenmiştir. Ardından kurulan MARS modellerinin ürettiği taban fonksiyonları ile modellemeler yeniden yapılmıştır. ANOVA sonuçları ve yanıt yüzeyleri modelleri MARS modelleriyle karşılaştırılmıştır. Esas olarak elde edilen modellerin içerdiği faktör sayısı, modellere ilişkin düzeltilmiş belirlilik katsayısı R_{adj}^2 ve modelin standart hatası hesaplanmıştır. Bunlara ek olarak etkileşim terimlerin yüzey grafikleri ve kontör grafikleri oluşturulmuştur. Çizilen grafikler MARS

modelleriyle oluşturulan grafiklerle karşılaştırılmıştır ve sonuçlar yorumlanmıştır. Bu bölümde R Yazılımı 2.10, SAS 9.0, MINITAB 14 ve MARS 2.0 programları birlikte kullanılmıştır.

4.3.1 Toprak Verisi Cd Kirliliği için MARS Analizi

Çalışmadaki amacımız Cd kirliliğine ilişkin yanıt yüzeylerini, klasik yolla ve MARS ile oluşturabilmektir. İlk olarak Cd kirliliği için toplanan veriler her biri beş tekrarlı olmak üzere 3^2 deneysel düzenden elde edilmiştir. 3^2 deneysel tasarımdan gelen 9 gözlem, hepsi beş tekrarlı olmak üzere toplam 45 gözlem bu analiz için kullanılmıştır. x_1 ve x_2 olarak kodlanmış tramvay ve araç faktörleri için varyans analizi, etkileri birbirinden ayırarak detaylı olarak incelenmiş; istatistiksel açıdan anlamlı olan faktörler, etkileşim terimleri ve bunların doğrusal, karesel ve karşılıklı ayrı ayrı bileşenlerinin anlamlılıklarına bakılmıştır. Bulunan sonuçlar R Yazılımı kullanılarak hazırlanmış ve Çizelge 4.2’ de verilmiştir.

Çizelge 4.2: Cd yanıt yüzeyi modeli için etkilerin parçalanması

Faktör	Sd	KT	Ortalama Kareler	F	P	
x_1	2	0.04773	0.02387	3.8792	0.02981	*
x_1	1	0.03713	0.03713	6.0349	0.01898	*
x_1^2	1	0.0106	0.0106	1.7235	0.19756	
x_2	2	0.48709	0.24355	39.5863	8.11E-10	***
x_2	1	0.48503	0.48503	78.8367	1.34E-10	***
x_2^2	1	0.00207	0.00207	0.3359	0.5658	
x_1x_2	4	1.59105	0.39776	64.6525	6.19E-16	***
x_1x_2	1	0.12002	0.12002	19.5084	8.78E-05	***
$x_1^2x_2$	1	1.18105	1.18105	191.9681	5.34E-16	***
$x_1x_2^2$	1	0.0015	0.0015	0.2438	0.62447	
$x_1^2x_2^2$	1	0.28848	0.28848	46.8898	5.20E-08	***
Artıklar	36	0.22148	0.00615			

Anlamlılık Düzeyleri:0 '***' 0.001 '**' 0.01 '*' 0.05 '.'

Çizelge 4.2’de etkilerin ve etkileşim teriminin doğrusal ve karesel bileşen-

leri tek tek incelenmiş ve MARS modeli ile karşılaştırma yapmak için kullanılmıştır. İlk sütununda 3^2 faktöriyel deneyde kullanılan faktörler, bunların doğrusal etkileri (x_1, x_2) , karesel etkileri (x_1^2, x_2^2) ve etkileşim terimlerinin karşılıklı bileşenleri $(x_1x_2, x_1^2x_2, x_1x_2^2)$ bileşenlerine ayrılarak verilmiştir. Satırlarda faktörlerin etkileri, sırasıyla bunların doğrusal ve karesel bileşenlerinin etkileri alt satırında italik olarak verilmiştir. Buna göre, ilk satırda x_1 faktörü, onun altındaki satırda x_1 faktörünün doğrusal etkisi ve x_1 faktörünün karesel etkisi italik olarak verilmiş ve ayrı ayrı incelenmiştir. Benzer şekilde, x_2 faktörü ve onun altındaki satırlarda, bu faktöre ait doğrusal ve karesel etkiler italik olarak verilmiştir. Bu inceleme aynı şekilde etkileşim terimi x_1x_2 için de yapılmış, tüm bileşenleri alt satırlarda italik olarak verilmiştir. İkinci sütunda, bu terimlere ilişkin serbestlik dereceleri (Sd) verilmiştir. İlerleyen sütunlarda sırasıyla ortalama kareler, F değerleri, ve bunlara ilişkin olasılıklar yer almıştır. Açıkça görülmektedir ki, x_1 ve x_2 faktörleri arasındaki etkileşim terimi (x_1x_2) , anlamlı bulunmuştur. Bunların arasından etkileşim teriminin bileşenlerinden, doğrusal ile doğrusal etkisi (x_1x_2) , karesel ile doğrusal etkisi $(x_1^2x_2)$ ve karesel ile karesel etkisi, $(x_1^2x_2^2)$, anlamlı bulunmuştur, ancak $(x_1x_2^2)$ teriminin doğrusal ile karesel etkisi anlamlı bulunmamıştır. SAS 9.0 kullanılarak elde edilen ANOVA sonuçları Çizelge 4.3 ile verilmiştir.

Çizelge 4.3: Cd yanıt yüzeyi için ANOVA

Değişim Kaynağı	Sd	KT	F	Pr>F
Doğrusal	2	0.012670	0.16	0.8509
Karesel	2	0.522156	6.68	0.0032
Etkileşim	1	0.288480	7.38	0.0098
Regresyon Toplam	5	0.823306	4.21	0.0037
Uyum Eksikliği	3	1.302567	70.57	<0.0001
Saf Hata	36	0.221483		
Hata Toplam	39	1.524050		
Toplam	44	2.34736		

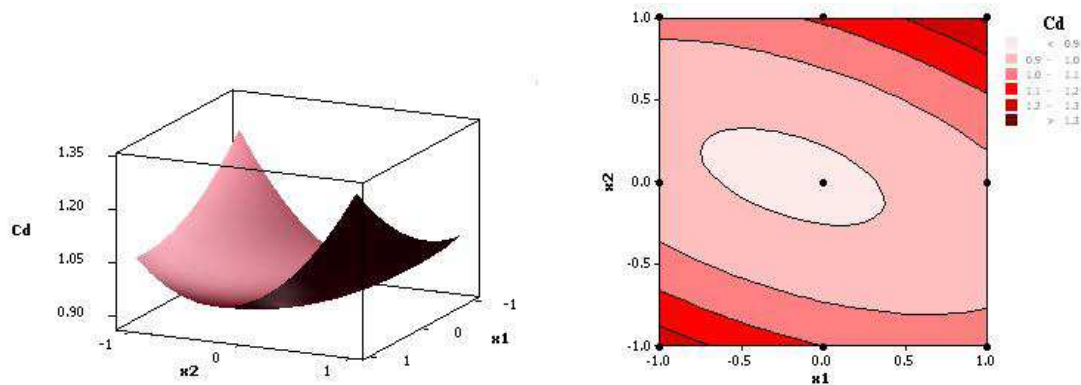
Çizelge 4.3' de karesel ve etkileşim terimlerinin istatistiksel olarak anlamlı oldukları görülmüştür.

Elde edilen ikinci dereceden yanıt yüzeyi Eşitlik 4.1 ile verilmiştir:

$$E_{cd}(y) = 0.887 + 0.018x_1 + 0.008x_2 + 0.06x_1^2 + 0.22x_2^2 + 0.120x_1x_2 \quad (4.1)$$

Eşitlik 4.1 ile verilen ikinci dereceden yanıt yüzeyi modeline ait standart hata $S_2=0.18$ olarak bulunmuştur. Elde edilen $F_{hesap} = 4.21 > F_{0.05,5,39} = 2.455$ olduğundan model anlamlıdır. Artıkların çoğunun mühendislik uygulamalarında pratik olarak önemli olmadığı ifade edilmiştir. İkinci dereceden yanıt yüzeyine ilişkin durağan nokta araştırması Bölüm 2.3.2' de anlatıldığı gibi kanonik analiz ile incelenmiştir. Buna göre, x_1 ve x_2 ' nin bir fonksiyonu olarak $\hat{y}$ ' nin kısmi türevleri alınarak durağan nokta $x_s = (-0.186, 0.032)$ bulunur. Eşitlik 4.1 ile verilen modele ait özdeğerler, $(\lambda_1, \lambda_2) = (0.24, 0.04)$ olarak bulunmuştur. Her iki özdeğer sıfırdan büyük olduğundan durağan nokta bir minimum noktasıdır. Burada dikkat edilmesi gereken β_{11} ve β_{22} toplamının λ_1 ve λ_2 ' ye eşit olmasıdır. Durağan noktaya karşılık gelen tahminlenmiş Cd değeri ise $\hat{y}_{cd} = 0.886$ olarak bulunmuştur.

İkinci dereceden modele ilişkin yanıt yüzeyi grafiği ve buna ilişkin kontör haritası Şekil 4.4 ile verilmiştir.



Şekil 4.4: Cd için yanıt yüzeyi ve kontör grafikleri

Kontör grafiğine göre yanıt yüzeyi bir çukur göstermektedir. Buna göre örneğin, $\hat{y} = 0.967$ ve $x_1 = 0$ iken, yüzey için elde edilen model $0.967 =$

$0.887 + 0.0084x_2 + 0.22x_2^2$ ' dir. Buradan x_2 için denklemin kökleri elde edilirse, iki kök bulunur. Bunlar $x_{21} = 0.586$ ve $x_{22} = -0.622$ ' dir. Bu koordinatları veren iki nokta $(0, 0.586)$ ve $(0, -0.622)$ ise, $y = 0.967$ için çizilen yanıt kontörü üzerindedir. Her bir kontör çizgisinin tahmin edilmiş yanıt değerinin yüksekliğini gösterdiği kontör grafiği Şekil 4.4' te verilmiştir.

Benzer şekilde, $\hat{y} = 1.099$ ve $x_1 = 0$ için yanıt yüzeyi modeli $1.099 = 0.887 + 0.008x_2 + 0.22x_2^2$ olarak elde edilir. Buradan denklemin kökleri $x_{21} = 0.96$ ve $x_{22} = -1.00$ olarak bulunur. Bu koordinatları veren iki nokta $(0, 0.96)$ ve $(0, -1.00)$ ise, $\hat{y} = 1.099$ için çizilen kontör üzerindedir.

MARS analizi için değişkenler ortogonal hale getirildikten sonra, tasarım matrisi oluşturulur. Bunun için $(x_1) = (-1, 0, +1)$ kodlamasının karesel hali olan $(x_1^2) = (+1, -2, +1)$ kullanılmıştır. Modele sırasıyla, x_1 , x_2 , x_1^2 ve x_2^2 terimleri alınmıştır. Etkileşim terimi için belirlenen maksimum derece 2 olarak verilmiştir. MARS modelleri Çizelge 4.4 ile verilmiştir.

Çizelge 4.4: Cd için MARS modelleri

Taban Fonksiyonları	Model	Birim	Etkileşim	R_{adj}^2	GCV
		Sayısı**			
Y=1.075	1	0	Hayır	0.00	0.055
TF1=max(0,x ₂ -0)	2	0	Evet	0.817	0.051
TF2=max(0,0-x ₂)					
TF3=max(0,x ₁ +1)*TF2					
TF4=max(0,x ₁ +1)					
TF5=max(0,x ₁ +1)*TF1					
**:					
Y=0.62+0.52*TF1+0.75*TF2-0.54*TF3+0.29*TF4-0.30*TF5					
TF1=max(0,x ₂ -0)	3	1	Evet	0.802	0.027*
TF2=max(0,0-x ₂)					
TF3=max(0,x ₁ +1)*TF2					
TF4=max(0,x ₁ +1)					
TF6=max(0,0-x ₁)*TF1					
Y=0.675+0.70*TF2-0.52*TF3+0.280*TF4+0.52*TF6					

Düğümmler arası minimum birim sayısı; *: En uygun model

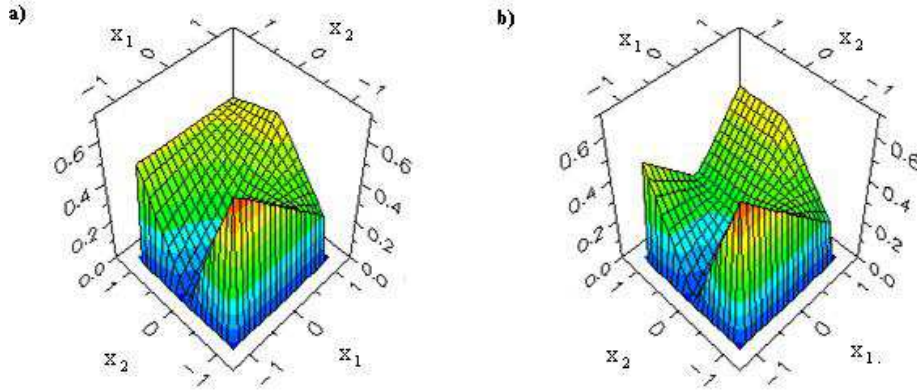
Çizelge 4.4' ün ilk sütununda MARS ile kurulan modellerin sayısı, sonraki sütunda her bir model için belirlenen düğüm sayısı, daha sonra modelde etkileşim teriminin olup olmadığı, MARS modelin ürettiği taban fonksiyonları (TF), düzeltilmiş belirlilik katsayısı R_{adj}^2 ve GCV değeri yer almaktadır. Modelde etkileşim terimleri yer almadığında buna toplamsal model denilmektedir [36]. Taban fonksiyonları ve bunların yansımaları sırasıyla (X-düğüm) veya (düğüm-X) olarak tanımlanmıştır. İlk model, başka bir deyişle toplamsal model R_{adj}^2 ve GCV değerlerine göre iyi sonuçlar vermemiştir. Birinci modelde sadece sabit katsayının modelde olması, diğer değişkenlerin modelde yer alması modelin geliştirilmesi gerektiğini ortaya koyar. Modele etkileşim terimleri eklenince, başka bir deyişle ikinci modelde, R_{adj}^2 ve GCV değerleri sırasıyla 0.817 ve 0.051 olarak bulunmuştur. Modelde bulunan bütün taban fonksiyonları anlamlı çıkmıştır. Model için hesaplanan $F_{0.05,5,39}$ değeri 40.36 (p=0) olarak bulunmuştur. Modeldeki değişkenlerin anlamlılıkları detaylı bir şekilde Ek-5' te verilmiştir. Aynı modelde, ilk taban fonksiyonu TF1 ve ikinci taban fonksiyonu TF2, x_2 değişkeni kullanılarak hesaplanmış birbirinin yansıması olan fonksiyonlardır. MARS, TF1 ve TF2' nin yanıt değişkeni üzerindeki etkisini modelde birleştirmiştir. x_2 değişkeninin (0) olduğu nokta bu değişkenin düğüm noktasıdır. MARS, TF1 ve TF2' nin yanıt değişkeni üzerindeki etkisini $(+0.529 * TF1 + 0.753 * TF2)$ ifadesiyle modele dahil etmiştir. TF4 taban fonksiyonu, x_1 değişkenini kullanarak (+1) düğüm noktasıyla modelde yer almıştır. Bu modelde, standart taban fonksiyonları yanıt değişkenini pozitif yönde (0.52, 0.75, 0.29) etkilemiştir. Modelde, TF3 ve TF5 taban fonksiyonları sırasıyla TF2 ve TF1 taban fonksiyonlarıyla çarpımı olarak yer almışlardır. Bunların yanıt değişkeni üzerindeki etkisi ise negatif yönlü (-0.54, -0.30) olmuştur. Üçüncü modelde de etkileşim terimlerinin modelde bulunmasına izin verilmiştir. Buna göre TF1 ve TF2 taban fonksiyonları, x_2 değişkenini kullanarak oluşturulmuş yansıma fonksiyonlarıdır. Bu taban fonksiyonları için (0) düğüm noktası olarak seçilmiştir. Benzer şekilde TF3 ve TF4 taban fonksiyonları, ikinci modeldeki gibi davranmışlardır. Ayrıca TF3 ve TF6 taban fonksiyonları etkileşim terimleri olarak üçüncü modelde yer almışlardır. Üçüncü modeldeki

TF6'nın modelde bulunması modeli GCV anlamında geliştirmiştir.

Kurulan MARS modelleri arasında gözlemler arası birim sayısı sıfırdan bire yükselirken ve etkileşim terimlerinin modelde olması veya tersi durumuna göre bir karşılaştırma yapıldığında, üçüncü model en küçük GCV değeri ile 0.027 ve R_{adj}^2 değeri 0.802 ile en uygun model olarak bulunmuştur. Bu model ile Eşitlik 4.1 ile verilen ikinci dereceden yanıt yüzeyi, içerdikleri değişken sayısı, R_{adj}^2 değeri ve modelin standart hatası anlamında karşılaştırılmıştır. MARS modeline göre hesaplanan F değeri, $F_{0.05,4,40} = 45.53(p = 0)$ olarak bulunmuştur. Bu değer oldukça yüksek olduğu için kurulan model anlamlıdır.

Cd kirliliği üzerine elde edilen MARS modeline ilişkin yüzey grafikleri Şekil 4.5 (a-b) ile gösterilmiştir.

Şekil 4.5' de ikinci model ve üçüncü model için etkileşim halinde olan faktörler çizilmiştir. Şekil 4.5a' da, ikinci MARS modeli için maksimum yanıt değeri $(x_1, x_2) = (-1, -1)$ koordinatında gözlenirken, Şekil 4.5b' de verilen üçüncü model için maksimum yanıt değeri $(x_1, x_2) = (-1, -1)$ koordinatında elde edilmiştir.



Şekil 4.5: Cd için ikinci ve üçüncü MARS modelleri grafikleri

Şekil 4.5b ile verilen grafikte, diğerinden farklı olarak, x_1 değişkenin $(0, 1)$ noktasında Cd yanıt değişkeni üzerindeki etkisi azalmıştır.

MARS ile oluşturulan en iyi modele ilişkin değişkenlerin göreceli önemlilikleri en yüksekten en düşüğe göre Çizelge 4.5' de düzenlenmiştir. Değişken önemlilikleri hesaplanırken, önemliliği araştırılan değişken dışındaki bütün değişkenleri dışarda tutup tekrar uyum iyiliği araştırılır ve buna göre skor

değerleri belirlenir. Bu değerler modelde yer alan değişkenlerin diğerlerine göre göreceli katkılarıdır. Cd kirliliği için MARS modelinin oluşturduğu değişken önemliliği Çizelge 4.5 ile verilmiştir. Buna göre, x_2 değişkeninin önem düzeyi %100 çıkmıştır. Bir başka ifadeyle Cd kirliliğine en çok katkısı olan değişken, MARS analizi sonucu x_2 , araç olarak bulunmuştur.

Çizelge 4.5: Cd MARS modeli için değişken önemliliği

Değişken	Önemlilik
x_2	100.000
x_1	96.906

Bunun yanında x_1 değişkeninin yanıt değişkenine olan katkısı %96.9 seviyesinde önemli çıkmıştır. MARS analizi sonucunda, değişkenler arasında önemlilik skorlarının hesaplanması bu anlamda çalışmaya yenilik getirmiştir. Bir başka deyişle, Cd kirliliği üzerinde araç değişkeninin daha önemli bulunması, bunun yanıt yüzeyleri modelleri ile araştırılabilecek bir durum olmaması ancak MARS ile mümkün olması, çalışmaya sonuçları yorumlama anlamında katkı sağlamıştır.

4.3.2 Toprak Verisi Zn Kirliliği için MARS Analizi

Zn kirliliği için yine her biri 5 tekrarlı bir 3^2 deneysel tasarımdan gelen 9 gözlem her biri beş tekrarlı olmak üzere toplam 45 gözlemle çalışılmıştır. Yanıt değerine etkisi istatistiksel olarak anlamlı olan faktörler, etkileşim terimi ve bunların karşılıklı bileşenlerinin ayrı ayrı etkileri incelenmiş, sonuçlar Çizelge 4.6' da verilmiştir.

Burada faktörler, faktörlerin doğrusal ve karesel etkileri ile etkileşim terimi ve etkileşim teriminin doğrusal ve karesel etkileri her biri bileşenlerine ayrılarak itelik olarak verilerek sırasıyla hesaplanmıştır. Buna göre tüm faktörler, etkileşim terimi ve bunların doğrusal ve karesel etkilerinin bütün bileşenleri istatistiksel olarak anlamlı bulunmuştur.

Çizelge 4.6: Zn yanıt yüzeyi modeli için etkilerin parçalanması

Faktör	Sd	KT	Ortalama Kareler	F	P	
x_1	2	42554	21277	3024.42	2.20E-16	***
x_1	1	33285	33285	4731.31	2.20E-16	***
x_1^2	1	9269	9269	1317.53	2.20E-16	***
x_2	2	18368	9184	1305.43	2.20E-16	***
x_2	1	1879	1879	267.13	2.20E-16	***
x_2^2	1	16488	16488	2343.73	2.20E-16	***
x_1x_2	4	11758	2939	417.82	2.20E-16	***
x_1x_2	1	2926	2926	415.86	2.20E-16	***
$x_1^2x_2$	1	1635	1635	232.37	2.20E-16	***
$x_1x_2^2$	1	5965	5965	847.84	2.20E-16	***
$x_1^2x_2^2$	1	1233	1233	175.23	2.12E-15	***
Artıklar	36	253	7			

Anlamlılık Düzeyleri:0 '***' 0.001 '**' 0.01 '*' 0.05 '.'

SAS 9.0 kullanılarak elde edilen ANOVA sonuçları Çizelge 4.7 ile verilmiştir. Çizelge 4.7' de yanıt yüzeyi modelindeki karesel ve etkileşim terimlerinin istatistiksel olarak anlamlı oldukları görülmüştür.

Elde edilen ikinci dereceden yanıt yüzeyi Eşitlik 4.2 ile verilmiştir:

$$E_{zn}(y) = 112.69 + 17.58x_1 - 23.44x_2 - 57.69x_1^2 - 13.71x_2^2 + 7.85x_1x_2 \quad (4.2)$$

Eşitlik 4.2 ile verilen ikinci dereceden yanıt yüzeyi modeline ait standart hata $S_2=33.51$ olarak bulunmuştur. Elde edilen $F_{hesap} = 44.98 > F_{0.05,5,39} = 2.455$ olduğundan model anlamlıdır.

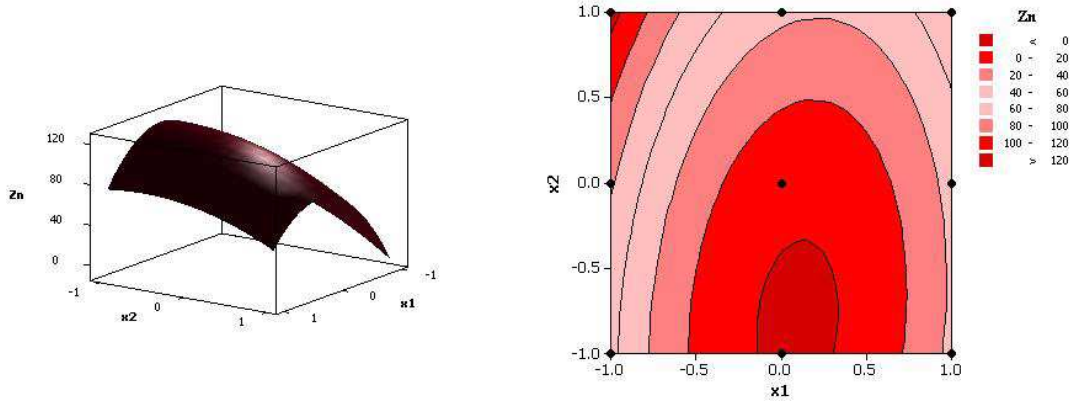
İkinci dereceden yanıt yüzeyine ilişkin durağan nokta araştırması Bölüm 2.3.2' de anlatıldığı gibi kanonik analiz ile incelenmiştir. Buna göre, Eşitlik 4.2 ile verilen modele ait özdeğerler, $(\lambda_1, \lambda_2) = (-13.36, -58.04)$ ve her ikisi de sıfırdan küçük olduğundan durağan nokta bir maksimum noktasıdır, $x_s = (0.096, -0.827)$. Durağan noktaya karşılık gelen tahminlenmiş Zn değeri ise $\hat{y} = 123.235$ olarak bulunmuştur.

Zn kirliliği yanıt yüzeyi modeline ait yanıt değerleri eğrileri Şekil 4.6 ile gös-

Çizelge 4.7: Zn yanıt yüzeyi için ANOVA

Değişim Kaynağı	Sd	KT	F	Pr>F
Doğrusal	2	25757	46.6	0.0001
Karesel	2	35164	63.62	0.0001
Etkileşim	1	1232.73	4.46	0.0411
Regresyon Toplam	5	62154.1	44.98	0.0001
Uyum Eksikliği	3	10525	498.69	<0.0001
Saf Hata	36	253.26		
Hata Toplam	39	10778		
Toplam	44	72932.3		

terilmiştir. Kontör çizgilerinden, yüzeye ilişkin şeklin tepecik olduğu söylenebilir. Buna göre örneğin, $\hat{y} = 122.426$ ve $x_1 = 0$ için yanıt yüzeyi modeli $122.42 = 112.69 - 23.44x_2 - 13.71x_2^2$ olarak elde edilir. Buradan denklemin kökleri bulunursa, $x_{21} = -0.71$ ve $x_{22} = 0.99$ elde edilir. Bu koordinatları veren iki nokta $(0, -0.71)$ ve $(0, 0.99)$ ise, $\hat{y} = 122.42$ için çizilen kontör üzerindedir.

**Şekil 4.6:** Zn için yanıt yüzeyi ve kontör grafikleri

Çizelge 4.8' de üç MARS modeli kurulmuştur. Bunlardan ilki toplamsal model olarak incelenmiştir.

Çizelge 4.8: Zn için MARS modelleri

Taban Fonksiyonları	Model	Birim Sayısı**	Etkileşim	R_{adj}^2	GCV
TF1= $\max(0, x_1-0)$ TF2= $\max(0, 0-x_1)$ TF3= $\max(0, x_2-0)$ Y=117.55-40.11*TF1-75.27*TF2-42.02*TF3	1	0	Hayır	0.813	680.710
TF1= $\max(0, x_1-0)$ TF2= $\max(0, 0-x_1)$ TF3= $\max(0, x_2-0)$ TF5= $\max(0, x_1^2+2)$ *TF3 TF6= $\max(0, x_1+1)$ *TF3 TF7= $\max(0, x_2^2-0)$ *TF1 Y=131.56-77.48*TF1-89.73*TF2-98.71*TF3+19.36*TF5+14.68*TF6+6.55*TF7	2	0	Evet	0.994	94.720
TF1= $\max(0, x_1-0)$ TF2= $\max(0, 0-x_1)$ TF3= $\max(0, x_2-0)$ TF4= $\max(0, 0-x_2)$ TF5= $\max(0, x_1-0)$ *TF3 TF6= $\max(0, 0-x_1)$ *TF3 BF7= $\max(0, x_1-0)$ *TF4 Y=131.56-77.48*TF1-89.73*TF2-84.02*TF3+92.44*TF5+43.40*TF6+19.66*TF7	3	1	Evet	0.994	41.573*

** : Döğümler arası minimum birim sayısı*; En uygun model

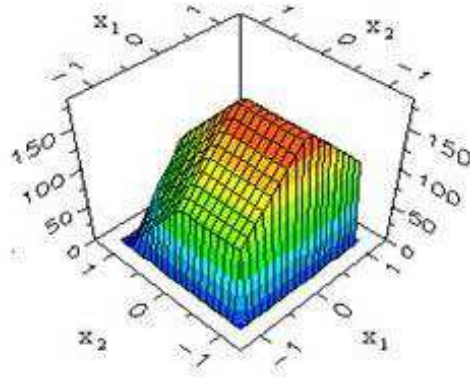
Toplamsal modelin R_{adj}^2 değeri 0.813 ve GCV değeri 680.710 olarak bulunmuştur. Toplamsal modelde, Zn yanıt değışkeni TF1, TF2 ve TF3 taban fonksiyonlarının bir doğrusal bileşkesi olarak ortaya çıkmıştır. Burada TF1 ve TF2 taban fonksiyonları için, x_1 değışkeni kullanılmış ve en iyi döğüm noktası olarak (0) olarak seçilmiştir. Buradaki TF1 ve TF2 taban fonksiyonları birbirinin yansıması olan fonksiyonlardır. Aynı modelde TF3 taban fonksiyonu, x_2 değışkeni ile hesaplanmış ve en iyi döğüm noktası (0) olarak seçilmiştir. Modeldeki tüm taban fonksiyonları istatistiksel olarak anlamlı bulunmuştur.

Modele taban fonksiyonlarının ve modelin anlamlılığı Ek-5' de detaylı incelenmiştir. Etkileşim terimleri modele dahil edildiğinde, başka bir deyişle ikinci modelde, TF1 ve TF2 taban fonksiyonları x_1 değişkeni ve buna karşılık gelen en iyi düğüm noktasını (0) olarak seçmiştir. Aynı modelde TF3 taban fonksiyonu, x_2 değişkenini ve (0) düğüm noktasını seçmiştir. Bunların dışında modeldeki etkileşim terimleri TF5, TF6 ve TF7 olarak görülmektedir. Bunlardan TF5 ve TF6 taban fonksiyonları TF3 ile, TF7 taban fonksiyonu TF1 fonksiyonlarının çarpımıyla elde edilmiştir.

Bunlara ek olarak ve daha önce de belirtildiği gibi yine TF1 ve TF2 taban fonksiyonları üçüncü modelde birbirinin yansıması olan fonksiyonlar olarak gözlenmiştir. TF5, TF6 ve TF7 terimleri modelde etkileşim terimleri olarak üçüncü modelde yer alırlar. Benzer şekilde TF1 ile TF2 ve TF3 ile TF4 taban fonksiyonları da birbirinin yansıması olan fonksiyonlar olarak modelde yer alırlar. TF5, TF6 ve TF7 taban fonksiyonları da x_1 ve x_2 değişkenlerinin modifiye edilmiş halleri olarak modelde yer alırlar. Çizelge 4.8' de üçüncü modelde TF4 taban fonksiyonunun TF3 taban fonksiyonunun yansıması olarak modelde bulunması, modeli GCV anlamında geliştirmiştir. GCV değeri ile karşılaştırma yapıldığında en iyi model GCV değeri 41.573 olan üçüncü model olarak elde edilmiştir. Üçüncü modelde altı tane taban fonksiyonu olup bunlar yanıt değişkenini doğrusal olmayan bir yapıda etkilemiştir.

Zn kirliliğini modellemek üzere MARS' ın ürettiği üçüncü modele ilişkin grafikler Şekil 4.7 ile verilmiş ve yanıt değişkenine olan katkıları gösterilmiştir.

Şekil 4.7 de x_1 ile x_2 değişkenlerinin yanıt değeri ile ilişkisi gösterilmiştir. Buna göre, yanıt değişkeni maksimum değerini $(x_1, x_2) = (0, 0)$ ve $(x_1, x_2) = (0, -1)$ koordinatlarında almaktadır.



Şekil 4.7: Zn için MARS grafiği

MARS ile oluşturulan en iyi modele ilişkin değişkenlerin göreceli önemlilikleri en yüksekten en düşüğe göre Çizelge 4.9’ da düzenlenmiştir. Bu değerler modelde yer alan değişkenlerin diğerlerine göre göreceli katkılarıdır. Zn kirliliği için MARS modelinin oluşturduğu değişken önemliliği Çizelge 4.9 ile verilmiştir.

Çizelge 4.9: Zn MARS modeli için değişken önemliliği

Değişken	Önemlilik
x_1	100.00
x_2	81.83

Buna göre, x_1 değişkeninin önem düzeyi %100 çıkmıştır. Bir başka ifadeyle Zn kirliliğine en çok katkısı olan değişken, MARS analizi sonucu x_1 , tramvay olarak bulunmuştur. Bunun yanında x_2 değişkeninin yanıt değişkenine olan katkısı %81.83 seviyesinde önemli çıkmıştır.

4.3.3 Toprak Verisi Pb Kirliliği için MARS Analizi

Pb kirliliğine ilişkin toplanan verilerden her biri beş tekrarlı 3^2 deneysel düzeni oluşturulmuştur. 3^2 deneysel düzenden gelen 9 gözlem, her biri 5 tekrarlı olmak üzere toplam 45 gözlemle çalışılmıştır. Yanıt değişkenine etkisi olan faktörler, etkileşim terimleri ve bunların karşılıklı bileşenleri ayrı ayrı incelenmiş, sonuçlar varyans analizi tablosunda Çizelge 4.10 ile verilmiştir. Çizelge 4.10’ de görüldüğü gibi faktörlerin ve etkileşim terimlerinin doğrusal

ve karesel etkileri bileşenlerine göre ayrı ayrı hesaplanmıştır. Ancak bunlar arasında sadece x_1x_2 etkileşim terimi istatistiksel olarak anlamlı bulunmamış, diğer bütün etkiler istatistiksel olarak anlamlı bulunmuştur.

Çizelge 4.10: Pb yanıt yüzeyi modeli için etkilerin parçalanması

Faktör	Sd	Kareler	Ortalama Kareler	F	P(>F)	
Toplamı						
x_1	2	9899.3	4949.6	295.727	<2.2e-16	***
x_1	1	681.6	681.6	40.727	2.15E-07	***
x_1^2	1	9217.6	9217.6	550.728	<2.2e-16	***
x_2	2	8880.6	4440.3	265.297	<2.2e-16	***
x_2	1	6841.3	6841.3	408.747	<2.2e-16	***
x_2^2	1	2039.4	2039.4	121.847	4.15E-13	***
x_1x_2	4	5789.1	1447.3	86.471	<2.2e-16	***
x_1x_2	1	477.9	477.9	28.552	5.24E-06	***
$x_1^2x_2$	1	4166.7	4166.7	248.949	<2.2e-16	***
$x_1x_2^2$	1	1144.6	1144.6	68.385	7.66E-10	***
$x_1^2x_2^2$	1	0	0	5.85E-07	0.999	***
Artıklar	36	602.5	16.7			

Anlamlılık Düzeyleri:0 '***' 0.001 '**' 0.01 '*' 0.05 '.'

SAS programı kullanılarak elde edilen ANOVA sonuçları Çizelge 4.11 ile verilmiştir.

Çizelge 4.11: Pb yanıt yüzeyi için ANOVA

Değişim Kaynağı	Sd	Kareler Toplamı	F	Pr>F
Doğrusal	2	11257	34.34	0.0001
Karesel	2	7522.90	22.95	0.0001
Etkileşim	1	0.00	0.00	0.9998
Regresyon Toplam	5	18780	22.92	0.0001
Uyum Eksikliği	3	5789.14	115.30	<0.0001
Saf Hata	36	602.537		
Hata Toplam	39	6391.68		
Toplam	44	25171.6		

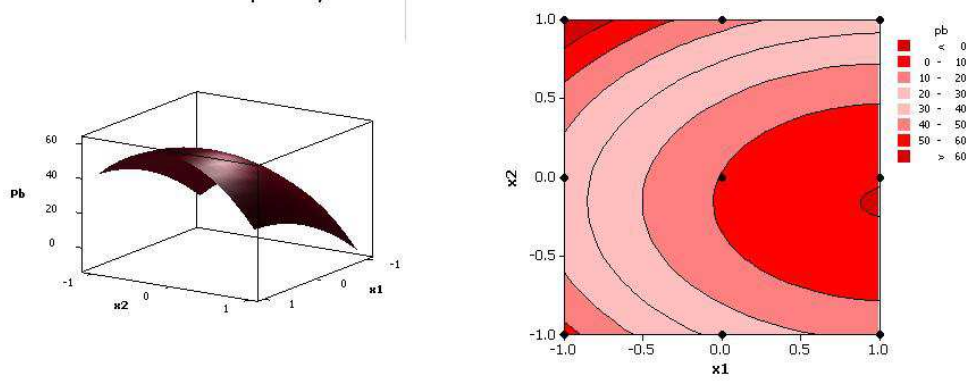
Elde edilen ikinci dereceden yanıt yüzeyi Eşitlik 4.3 ile verilmiştir:

$$E_{pb}(y) = 50.313 + 17.528x_1 - 8.244x_2 - 8.256x_1^2 - 26.155x_2^2 \quad (4.3)$$

Eşitlik 4.3 ile verilen ikinci dereceden yanıt yüzeyi modeline ait standart hata $S_2=12.80$ olarak bulunmuştur. Elde edilen $F_{hesap} = 22.92 > F_{0.05,5,39} = 2.455$ olduğundan model anlamlıdır. İkinci dereceden yanıt yüzeyine ilişkin durağan nokta araştırması Bölüm 2.3.2' de anlatıldığı gibi kanonik analiz ile incelenmiştir. Buna göre, Eşitlik 4.3 ile verilen modele ait özdeğerler, $(\lambda_1, \lambda_2) = (-8.25, -26.15)$ ve her ikisi de sıfırdan küçük olduğundan durağan nokta bir maksimum noktasıdır, $x_s = (1.06, -0.15)$. Durağan noktaya karşılık gelen tahminlenmiş Pb değeri ise $\hat{y} = 60.26$ olarak bulunmuştur.

Burada, Pb yanıt değişkenini etkileyen x_1 ve x_2 değişkenlerinin oluşturduğu yüzey gösterilmiştir. Buna göre örneğin, $\hat{y} = 41.67$ ve $x_1 = 0$ için elde edilen yanıt yüzeyi modeli $41.67 = 50.313 - 8.24x_2 - 26.15x_2^2$ olarak bulunmuştur. Bu denklem çözülürse, kökler $x_{21} = 0.43$ ve $x_{22} = -0.75$ olarak elde edilir.

Bu koordinatları veren iki nokta $(0, 0.43)$ ve $(0, -0.75)$ ise, $\hat{y} = 41.67$ için çizilen kontör üzerindedir. Bu modele ilişkin yanıt yüzeyi grafiği Şekil 4.8 ile verilmiştir.



Şekil 4.8: Pb için yanıt yüzeyi ve kontör grafikleri

Pb yanıt değişkenini modellemek üzere kurulan MARS modelleri Çizelge 4.12 ile verilmiştir. İlk olarak toplamsal model incelenmiş, daha sonra modele etkileşim terimleri eklenerek ikinci bir model oluşturulmuştur. İkinci modeli geliştirme adına bir katkısı olmadığından Çizelge 4.12' ye üçüncü bir model konulmamıştır.

Çizelge 4.12: Pb için MARS modelleri

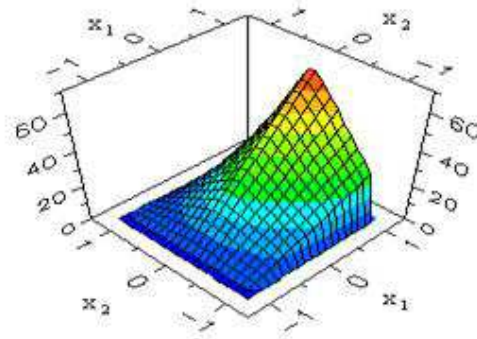
Taban Fonksiyonları	Model Birim Sayısı**	Etkileşim	R_{adj}^2	GCV
TF1= $\max(0, x_1+1)$ TF2= $\max(0, x_2-0)$ Y= $18.326+17.52*TF1-25.44*TF2$	1	0	Hayır	0.605 369.005
TF1= $\max(0, x_1+1)$ TF2= $\max(0, x_2-0)*TF1$ TF3= $\max(0, 0-x_2)*TF1$ Y= $9.84+34.65*TF1-30.64*TF2-20.74*TF3$	2	0	Evet	0.841 211.806*

**: Düşümler arası minimum birim sayısı*; En uygun model

Birinci modelde etkileşim terimi bulunmamaktadır ve bu modelde düzeltilmiş belirlilik katsayısı $R_{adj}^2 = 0.605$ olarak bulunmuştur. Bununla beraber ikinci modelde R_{adj}^2 katsayısı 0.841 olarak bulunmuştur. Düşüm sayısı artırılarak

üçüncü model daha incelenmiştir ancak katkı sağlamadığından bu çizelgede yer almamıştır. Etkileşim terimlerinin ilavesi bu anlamda modeli geliştirmiştir. MARS' ın ürettiği en iyi model en küçük GCV değeri 211.806 olan ikinci model olarak seçilir. Birinci modelde TF1 ve TF2 taban fonksiyonları hem x_1 hem de x_2 faktörlerini içerir. Bu taban fonksiyonları için bulunan düğüm noktaları sırasıyla (+1) ve (0) olarak bulunmuştur. İkinci modelde ise TF1 tek başına bir taban fonksiyonu iken TF2 ve TF3 etkileşimi ifade eden taban fonksiyonları olarak modelde yer almıştır. GCV değerinin en küçük olduğu model, ikinci modeldir. Ancak dikkat edilmesi gereken nokta, burada ikinci modelden başka bir etkileşim modelinin olmaması, durumu toplamsal model ile etkileşimli model arasında seçim yapma aşamasına getirir. Böyle bir durumda Friedman [36], etkileşimli modelin sadece, modelin R_{adj}^2 değerinin diğerinden oldukça büyük ise kabul edileceğini ileri sürmüştür. Buna göre, etkileşimli model ile toplamsal modelin R_{adj}^2 değerleri arasındaki fark $0.841-0.605=0.236$ ' dır. Bu değer yeterince büyük olduğu varsayımıyla etkileşimli model toplamsal modele tercih edilmiştir.

Şekil 4.9' de Pb kirliliği yanıt yüzeyini en iyi modelleyen ikinci MARS modelinin grafiği çizilmiştir. Burada, yanıt değişkeninin, x_1 ve x_2 değişkenleriyle olan ilişkisi gösterilmiştir.



Şekil 4.9: Pb için MARS grafiği

Buna göre, yanıt değişkeninin maksimum değer aldığı nokta, $(x_1, x_2) = (+1, 0)$ koordinatı olarak bulunmuştur.

MARS ile oluşturulan en iyi modele ilişkin değişkenlerin göreceli önemlilik-

leri en yüksekten en düşüğe göre Çizelge 4.13’ te düzenlenmiştir. Bu değerler modelde yer alan değişkenlerin diğerlerine göre göreceli katkılarıdır. Pb kirliliği için MARS modelinin oluşturduğu değişken önemliliği Çizelge 4.13 ile verilmiştir.

Buna göre, x_1 değişkeninin önem düzeyi %100 çıkmıştır. Bir başka ifadeyle Pb kirliliğine en çok katkısı olan değişken, MARS analizi sonucu x_1 , tramvay değişkeni olarak bulunmuştur.

Çizelge 4.13: Pb MARS modeli için değişken önemliliği

Değişken	Önemlilik
x_1	100.00
x_2	84.61

Bunun yanında x_2 değişkeninin yanıt değişkenine olan katkısı %84.61 seviyesinde önemli çıkmıştır. Pb kirliliğine en çok katkısı olan faktör tramvay olarak kaydedilmiştir.

5 SONUÇLAR

Bu çalışmada yanıt yüzeyi metodolojisiinde, birinci ve ikinci dereceden polinomiyal modeller üzerinde durulmuştur. Uygulamada, ikinci dereceden tasarımlar arasından 3^2 faktöriyel tasarım için MARS yaklaşımı önerilmiştir. Sürekli bağımsız değişkenler kullanılarak yanıt değişkenine olan etkiler iki yaklaşımla incelenmiştir. Klasik yolla elde edilen yanıt yüzeyleri modellerinin yetersiz kalması problemi vurgulanmış, yeni bir yaklaşım olarak Friedman' ın [36] 1991' de ortaya koyduğu MARS algoritmasının ürettiği taban fonksiyonlarından oluşan modellerle bu problem başarıyla ortadan kaldırılmıştır. MARS ile model oluşturmada, ikinci dereceden klasik yanıt yüzeyi modelinin tüm terimlerini elde edilebilecek şekilde, terimler yeniden düzenlenmiştir. Bu anlamda, karesel ve etkileşim terimlerinin modele dahil edilmesi, terimler arasındaki ortogonal ilişki göz önüne alınarak yapılmıştır. Terimlerin doğrusal, karesel ve ikili ilişkileri ve önemlilikleri ayrı ayrı incelenmiştir. Bu yüzden, MARS analizine dahil edilmesi gereken terimler üzerinde durulmuş ve MARS modeliyle elde terimlerin anlamlılıkları karşılaştırılmıştır. MARS' ın getirdiği değişken önemliliği incelemesi, klasik yanıt yüzeyi modeli analizinde yeni bir kavram olarak önerilmiştir.

Bunlara ek olarak, yinelemeli bölüntüleme algoritmasına dayalı iki program yazılmıştır. Bunların ilki, tek bağımsız değişkenin bağımlı değişken üzerindeki etkisini araştırmak üzere yapılmıştır. İkincisi iki bağımsız değişkenin bağımlı değişken üzerindeki etkisini araştırmak üzere yapılmıştır. Uygulamada kullanılan Cd ağır metal verisi üzerinden x_1 değişkeninin bu değişken üzerindeki etkisini araştırmak üzere MARS modellemesi yapan bir program yazılmıştır. Benzer şekilde, Cd verisi üzerinden, x_1 ve x_2 değişkenlerinin etkisini araştırmak üzere MARS modellemesi yapan bir program yazılmıştır. Bu programlar R Yazılımında hazırlanmıştır.

Elde edilen sonuçlar maddeler halinde açıklanmıştır:

- i. Bilinmeyen fonksiyonun tahmini
- ii. Uyum İyiliği
- iii. Grafik Uyumu
- iv. Algoritmalar

i. Bilinmeyen fonksiyonun tahmini

Bilinmeyen fonksiyonel yapı iki tip yaklaşımla açıklanmaya çalışılmıştır: Bunlardan ilki polinomial yanıt yüzeyleri modelleri, ikincisi MARS yaklaşımıdır. Klasik yanıt yüzeyi metodolojisi bilinmeyen yanıt yüzeyinin şeklini tahmin etmek için kullanılmıştır. Cd, Zn ve Pb kirlilikleri için ikinci dereceden yanıt modelleri başarıyla uygulanmıştır (Eş. 4.1, 4.2, 4.3). Bunlar Çizelge 4.4, 4.8 ve 4.12' de verilen MARS modelleriyle karşılaştırıldığında ise, etkileşim terimleriyle ortaya çıkan MARS modellerinin yanıt değişkenini nasıl etkilediği ve hangi faktörlerin önemli olduğu soruları cevaplandırılmıştır. MARS modelleri bilinmeyen yanıt yüzeyleri modellerini taban fonksiyonları kullanarak başarılı bir şekilde tahminlemiştir.

ii. Uyum İyiliği

MARS incelenen Cd, Zn ve Pb kirlilik modellerini ikinci dereceden yanıt yüzeyi modellerine göre daha iyi analiz etmiştir. Çizelge 5.1' deki R_{adj}^2 değerleri incelendiğinde MARS modellerinin daha iyi sonuçlar verdiği Çizelge 5.1' de görülür. Çizelge 4.4, 4.8 ve 4.12' de verilen MARS modellerinin ikinci dereceden yanıt yüzeyi modellerine göre daha karmaşık görünmelerine karşın, MARS' ın kullandığı değişken sayısı incelendiğinde bu durum hâlâ uygun kabul edilebilir. Bununla beraber RMSE (Root Mean Square Error) değeri her bir kirlilik modeli için bu iki tür yaklaşımla hesaplanmış ve Çizelge 5.1' de özetlenmiştir.

Çizelge 5.1' de verilen karşılaştırma tablosu dikkatle incelenirse, ilk satırda kirlilik değişkenleri Cd, Zn ve Pb sıralanmıştır. Bir sonraki satırda her bir

ağır metal için sırasıyla; R_{adj}^2 , modeldeki parametre sayısı ve RMSE değeri yer alır. Buna göre, Cd için oluşturulan klasik yanıt yüzeyi modelinin düzeltilmiş belirlilik katsayısı $R_{adj}^2 = 0.267$ iken MARS ile bulunan model için aynı değer $R_{adj}^2 = 0.802$ olarak hesaplanmıştır. Cd yanıt yüzeyi modelinde doğrusal etkilerin %0.05 düzeyinde anlamlı bulunmasına rağmen modelden çıkartılması uygun değildir. Bu durumda ikinci dereceden tüm etkiler anlamlı kabul edilir ve modelde kullanılan değişken sayısı 5 olarak kalır. Aynı ağır metal için üretilen MARS modeli ise 4 taban fonksiyon kullanmıştır.

Çizelge 5.1: Kirlilik modelleri için karşılaştırma

Kirlilik	Cd			Zn			Pb		
	R_{adj}^2	Ts	RMSE	R_{adj}^2	Ts	RMSE	R_{adj}^2	Ts	RMSE
YYM	0.267	5	0.197	0.714	5	12.80	0.833	5	33.51
MARS	0.802	4	0.103	0.994	3	3.118	0.841	6	9.535

Ts:Terim sayısı

Benzer şekilde yorumları sırasıyla Zn ve Pb kirlilik modelleri için de yapmak mümkündür. Buna göre, Zn yanıt yüzeyi modeli için hesaplanan R_{adj}^2 değeri 0.714 olarak bulunurken, aynı yanıt değişkenine ait MARS modelinin düzeltilmiş belirlilik katsayısı 0.994 çıkmıştır. Zn yanıt yüzeyi modeli için tüm etkiler anlamlı çıkmıştır, dolayısıyla modelde kullanılan değişken sayısı 5' dir. Aynı yanıt değişkenine ait MARS modeli ise 3 tane taban fonksiyonu üretmiştir. Her iki yaklaşım için modellerin hesaplanan RMSE değerleri sırasıyla 12.80 ve 3.12 olarak bulunmuştur. Pb yanıt değişkenine ait yanıt yüzeyi modelinin düzeltilmiş belirlilik katsayısı 0.833 olarak bulunurken, MARS aynı kriteri 0.841 olarak bulmuştur. Aynı değişkene ait yanıt yüzeyi modelindeki tüm etkiler anlamlı olduğundan modelde 5 değişken vardır. MARS ise bu değişkene ait model üretirken 6 tane taban fonksiyonu kullanmıştır. Buna rağmen, RMSE değerleri incelendiğinde MARS modelinin daha iyi sonuç verdiği görülmüştür.

iii. Grafik Uyumu

Grafiklerle kirlilik modellemesinde yanıt değişkeniyle tahmincilerin ilişkisini göstermek üzere Şekil 4.4-5 çizilmiştir. Cd kirlilik modeli için verilen yüzey grafiği Şekil 4.4 ile verilirken uygun MARS grafiği Şekil 4.5 ile verilmiştir. Her iki grafikte de tramvay ve araç değişkenlerinin Cd kirlilik tahminleri gösterilmiştir. Tramvayın ve aracın OGÜ ve Atatürk Lisesi civarında hem düşük değerlerinde $(-1, -1)$ hem de yüksek değerlerinde $(+1, +1)$, Cd kirliliğinde etkili bölgeler olarak tespit edilmiştir. Zn kirliliğine ilişkin yanıt yüzeyi grafiği ise Şekil 4.6 ile verilirken uygun MARS modeli grafiği Şekil 4.7 ile verilmiştir. Burada Zn kirliliğine en çok katkısı olan noktalar $(0, 0)$ ve $(0, -1)$ başka bir deyişle Alanönü ve Vişnelik olarak tespit edilmiştir. Pb kirliliği için yanıt yüzeyi grafiği Şekil 4.8 ile verilirken uygun MARS grafiği Şekil 4.9 ile verilmiştir. Pb kirliliğine en çok katkısı olan nokta $(+1, 0)$ başka bir deyişle Atatürk Bulvarı olarak tespit edilmiştir. Grafik incelemesinde karşımıza çıkan ortak değerler MARS ve yanıt yüzeyi grafiklerinin birbirini dengelediğini göstermiştir. Bununla beraber MARS yaklaşımının yanıt yüzeyi modelleri analizine olan katkısı, analizde önemli olan değişkenlerin üretilmesidir.

iv. Algoritma

Tek ve iki bağımsız değişkenin bağımlı değişken üzerindeki etkisini araştırmak üzere yinelemeli bölüntüleme algoritmasına dayalı iki program yazılmıştır. Bunlar sırasıyla Ek-1 ve Ek-2' de verilmiştir. Bununla beraber, Cd yanıt değişkenine ilişkin tek ve iki değişkenli MARS modelleri oluşturmada iki ayrı program yazılmış ve ilgili R kodları sırasıyla Ek-3 ve Ek-4' te verilmiştir. Bu programların yazımında *R*.2.10.0 sürümü kullanılmıştır. Ek-6' da kullanılan ham verilere yer verilmiştir. Programda kullanılan formüller Friedman' nın 1991' deki makalesinden alınmıştır [36].

Her iki yöntemle kurulan modeller ile toprak verisi kullanılarak yapılan ağır metal kirliliği araştırmasında, MARS modellerinin daha detaylı bilgiler

vermesi ve modele etkileşim terimlerini de katarak yanıt yüzeyleri analizine katkı sağlaması, gerçek yapıyı daha iyi yansıtmaları ve modelin açıklayıcılık gücünü artırması açısından, deneysel düzenden gelen verinin geleneksel yolla modellenmesi yerine temsili bir model olarak kullanılması önerilmiştir.

KAYNAKLAR

- [1] Box, G.E.P., ve Draper, N.R., *Empirical Model-Building and Response Surfaces*, Wiley - Sons, 1987.
- [2] Hill, W.J., Hunter, W.G., "A review of response surface methodology: A Literature Review", *Technometrics*, **8**, 571-590, 1966.
- [3] Mead, R and Pike, D J, "A review of response surface methodology from a biometric viewpoint", *Biometrics*, **31**, 803-851, 1975.
- [4] Myers, R H, Khuri, A I and Carter, W H, "Response surface methodology :1966-1988", *Technometrics*, **31**, 137-157, 1989.
- [5] Giovannitti-Jensen, R.H. Myers, "Graphical assessment of the prediction capability of response surface designs", *Technometrics*, **31**, 159-171, 1989.
- [6] Hardy, S.W. ve Nychka, D.W., "R-splines for response surface modeling", *North Carolina State University Mimeo Series*, **2526**, Raleigh, NC, 2000.
- [7] Hardy, S.W., Nychka, D.W., Boos, D.D. ve Haaland, R. D., "Variable selection for response surface modeling using R-splines", *Technometrics*, 2000.
- [8] Ren, S., "Modeling the toxicity of aromatic compounds to tetrahymena pyriformis:the response surface methodology with nonlinear methods", *J. Chem.Inf.Comput. Sci.*,**43**,1679-1687, 2003.
- [9] O'Connell, M.A. ve Wolfinger, R.D., "Spatial regression, response surfaces, and process optimization", *Journal of Computational and Graphical Statistics*, **6**, 224-241, 1997.
- [10] Heiligers, B., "E-Optimal designs for polynomial spline regression", *Journal of Statistical Planning and Inference*, **75**, 159-172, 1998.

- [11] Butler, N.A., "Properties of optimal designs under smoothing spline models", *Journal of Statistical Planning and Inference*, **92**, 259-266, 2001.
- [12] Tarpey, T., ve Holcomb, J., "Spline bottles", *The American Statistician*, **54(2)**, 2000.
- [13] Craven, P. ve Wahba, G., "Smoothing noisy data with spline functions estimating the correct degree of smoothing by the method of generalized cross-validation", *Numerical Mathematics*, **31**,317-403, 1979.
- [14] Chen, V.C.P., Ruppert, D. ve Shoemaker C.A., "Applying experimental and regression splines to a high-dimensional continuous state stochastic dynamic programming",*Journal of Operations Research*, **47(1)** , 1999.
- [15] Crino,S.T., *Combining Multivariate Adaptive Regression Splines with A Response Surface Methodology for Simulation Based Design Optimization*, PhD.Thesis, University of Virginia, 2006.
- [16] Walker, C., E. ve Parkhurst, A.M., "Response surface analysis of bake-lab data with a personal computer", *Cereals Food World*, **29(662)**, 1984.
- [17] Kan, B.ve Yazıcı, B., "Determining the coordinates of an experimental data set based on Multivariate Adaptive Regression Splines", *JSM 2009 Proceedings Book*, 3098-3104,Washington, DC, USA, 2009.
- [18] Kan, B., Yazıcı, B., "Assesment of fuel consumption using factorial experiments, regression trees and MARS", *Proceedings Book-Recent Advances in Applied Mathematics*, 196-201, 14th WSEAS MATH'09, Tenerife, Canary Islands, Spain, 2009.
- [19] Alpu, Ö., Yazıcı, B. ve Kan, B., "Analysis of Meteorological Data Using Nonparametric Multiple Regression", *JSM 2010 Bildiler Kitabı*, Washington, DC, USA, 2009.

- [20] Kan, B., Yazıcı, B., Malkoç, S., Altan, M. ve Koparal S., "A regression tree application on soil and road dust pollution", *JSM 2010 Bildirer Kitabı*, Vancouver, Canada, 2010.
- [21] Wand, M.P., Ruppert, D. ve Carroll, R.J., *Semiparametric Regression*, Cambridge Series in Statistics and Probabilistic Mathematics, 2003.
- [22] Laszlo, G.i, Kohler, M., Krzyzak, A. ve Walk, H., *A Distribution-Free Theory of Nonparametric Regression*, Springer Series in Statistics, 2002.
- [23] Dierckx, P., *Curve and Surface Fitting with Splines*, Clarendon-Oxford, 1993.
- [24] Cornell, J.A., ve Khuri, A.I., *Response Surfaces- Designs and Analyses*, Second Ed. Revised and Expanded, New York, 1996.
- [25] Box, G.E.P. ve Wilson, K.B., "On the experimental attainment of optimum conditions", *J.Roy., Statist. Soc.*, **B13**, 1-38, 1951.
- [26] Raymond H.M ve Montgomery C. D., *Response surface methodology - process and product optimization using designed experiments-* 2nd Ed., New York: J. Wiley, 2002.
- [27] Bal, H., *Optimizasyon Teknikleri*, Gazi Üniversitesi, 1995.
- [28] Wood, N.S., *Generalized Additive Models- An Introduction with R*, Chapman Hall /CRC, NY, 2006.
- [29] Morgan, J. N. and Sonquist, J. A., "Problems in the Analysis of Survey Data, and a Proposal", *Journal of the American Statistical Association*, **58**, 415-434, 1963.
- [30] Breiman, L., Friedman, J., Olshen, R., and Stone, C., *Classification and Regression Trees*, Wadsworth, Belmont, CA., 1984.
- [31] Stevens J.G., "Nonlinear modeling of time series using multivariate adaptive regression splines", *Journal of American Statistical Association*, **86**, 864-877, 1991.

- [32] Friedman, H. Jerome, "Multivariate adaptive regression splines", Dept. of Statistics, Stanford University, *Report 102*, 1988.
- [33] Silverman, B. W., *Some Aspects of the Spline Smoothing Approach to Non-Parametric Regression Curve Fitting*, Journal of the Royal Statistical Society (Ser B), **47**, **1**, 1-52, 1985.
- [34] Wegman, E. J. and Wright, I. W., "Splines in statistics", *Journal of the American Statistical Association*, **78**, **382**, 351-365, 1983.
- [35] Smith, P. L., "Splines as a Useful and Convenient Statistical Tool", *The American Statistician*, **33**(**2**), 57-62, 1979.
- [36] Friedman, H. Jerome, "Multivariate adaptive regression splines", *The Annals of Statistics*, **19**(**1**), 1991.
- [37] Boor, Carl de, *A Practical Guide to Splines*, Springer-Verlag, 1978.
- [38] Parzen, E., "On estimation of a probability density function and mode", *Ann. Math. Statist.*, **33**, 1065-1076, 1962.
- [39] Shepard, D., "A two-dimensional interpolation function for irregularly spaced data", *Proc. ACM Nat. Conf.*, 517-524, 1964.
- [40] Bozzini, M. Lenarduzzi, L., "Local smoothing for scattered noisy data", *Inten. Ser., Numer. Math.*, **75**, 51-60, Basel, 1985.
- [41] Wahba, G., *Spline Models for Observational Data*, 1990.
- [42] Stone, C. J. and Koo, Cha-Yong, "Additive splines in statistics", *Proceedings, Annual Meeting of Amer. Statist. Assoc.*, Statist. Comp. Section, August, 45-48, 1985.
- [43] Friedman, J. H. and Stuetzle, W., "Projection pursuit regression", *Journal of the American Statistical Association*, **76**(**376**), 817-823, 1981.
- [44] Buja, A., Hastie, T., Tibshirani, R., "Linear smoothers and additive models", *The Annals of Statistics*, **17**(**2**), 453-555, 1989.

- [45] Hastie, T., Tibshirani, R., Friedman, J.H., *The Elements of Statistical Learning*, 2nd Ed., Springer, 2009.
- [46] Friedman, H.Jerome, Roosen, C.B., "An Introduction to multivariate adaptive regression splines", *Statistical Methods in Medical Research*, 4,197-217, 1995.

Ek-1 Yanıt değişkeninin, tek değişken kullanarak regresyon ağacının oluşturmasında kullanılan R kodları

```
y=cd ; x=A ;n=length(y); reps=length(y) ;library(rpart)
left.ort=rep(0,reps);bof=rep(0,reps);right.ort=rep(0,reps);
left.sum=rep(0,reps); right.sum=rep(0,reps);total.sum=rep(0,reps);
left.dev=rep(0,reps);right.dev=rep(0,reps) ;imp=rep(0,reps) ;
cp=rep(0,reps);BOF= sum( (y- mean(y)) ^2 ) ;BOF ;
mse=sum(lm(y~x)$residuals^2)/n ;
dev= deviance(lm(y~x) ) ; summary(rpart(y~x))
G=function(x,y) {
# sıralı x,y lileri için döngüye girer.
left.x=matrix(0, reps, reps) ; left.y=matrix(0, reps, reps)
right.x=matrix(0, reps, reps) ; right.y=matrix(0, reps, reps)
# Her bir x değeri candidate partition değeri olarak düşünülür.
#0lası bölünmelerin matrisleri sol/sağ olarak ayrılır.
for(t in 1:reps ){
len= length(x[x < x[t]])
if (len==0) left.x[t,1:reps]= x
else {
left.x[t,1:len]=x[ x<x[t] ]
left.y[t,1:len]= y[x<x[t] ]
left.ort[t]= sum( (y[x<x[t]]-mean(y[x<x[t]])) ^2)
left.sum=rowSums(left.y)
}}
for(t in 1:reps ){
len= length(x[x>=x[t]])
if (len==0)right.x[t,1:reps]= x
else{
right.x[t,1:len]=x[x>=x[t]]
right.y[t,1:len]=y[x>=x[t]]
```

```

right.ort[t]= sum( (y[x>=x[t]]-mean(y[x>=x[t]])) ^2)
right.sum=rowSums(right.y)
}}

return(list(left.x=left.x, left.y=left.y,
  right.x=right.x, right.y=right.y , left.dev=left.ort,
  right.dev=right.ort, left.sum=left.sum, right.sum=right.sum)) }

G=G(x,y)

## F fonksiyonu NODE 2 ve NODE3  çıkartır.
F=function(left.x,left.y,right.x,right.y)
{ for (i in 1:reps){
left.x=G$left.x ; left.y=G$left.y ;
  right.x=G$right.x ; right.y=G$right.y
left.dev=G$left.dev; right.dev=G$right.dev
left.sum=rowSums(left.y)
right.sum=rowSums(right.y)
total.dev=left.dev[1:length(left.sum)] + right.dev[1:length(right.sum)]
total.ss=sum((y-mean(y))^2)
cp=round(((total.ss-(total.dev)) / BOF , 6)
imp=round( (total.ss-(total.dev)) / total.ss ,6)
bof.result=data.frame(left.dev,right.dev,imp,cp,total.dev,total.ss) }
# İlk partition için seçilen noktanın x vektöründeki sırasını verir
sira.0=which.min(bof.result[,5])
# İlk partition point olarak seçilen knot
p.point.1=x[order(bof.result[,5])[1] ] ;
sira.1=which(sort(x)==p.point.1)
sira.2=sira.1-1
candi=(sort(x)[sira.1]+sort(x)[sira.2])/2
xx=x[x<candi] ## NODE 3
yy=y[x<candi] ## NODE 3
left.mean=mean(y[x<candi]) ;
left.mse=bof.result[sira.0,][1]/length(yy)

```

```

right.mean=mean(y[x>=candi]);
right.mse=bof.result[sira.0,][2]/length(y[x>=candi])
xx.r=x[x>=candi] ## NODE 2
yy.r=y[x>=candi] ## NODE 2
reps=length(yy)
leaves23=rbind(bof.result[sira.0,])
leaves23[,9:12]=cbind(left.mean, right.mean, left.mse, right.mse)
colnames(leaves23)[9:12]=c("left.mean", "right.mean",
"left.mse", "right.mse")
# Kaçınıcı sıradaki x değerlerinin en küçük
# bof değerini verdiğini gösterir.
# En küçük bof değerini veren x lerin, vektördeki sırası
order ((bof.result[,5]))
i=1
while(order(bof.result[,5])[i]<6 | order(bof.result[,5])[i]>n-5)
i=i+1
x[order(bof.result[,5])[i]]
knot= x[order(bof.result[,5])[i]]
return(list(bof.result=bof.result, sira.0=sira.0,
sira.1=sira.1,sira.2=sira.2, p.point.1=p.point.1,i=i ,
knot=knot, candi=candi, xx=xx, yy=yy, xx.r=xx.r,
yy.r=yy.r, cp=cp,leaves23=leaves23)) }
F=F(left.x,left.y,right.x,right.y) ;F
rpart(y~x)
tree=F$leaves23
decide=function(F){
if(dim(data.frame(F$xx.r, F$yy.r))[1] > dim(data.frame(F$xx,F$yy))[1])
{d=data.frame(F$xx.r,F$yy.r) ; d} else{d=data.frame(F$xx,F$yy) ;d}
return(list(d=d))
} ; R=decide (F) ;R ; dim(R$d)
x=as.data.frame(R$d)[,1] ; y=as.data.frame(R$d)[,2]

```

```

n=length(y) ;reps=length(y)
left.ort=rep(0,reps) ; bof=rep(0, reps );
right.ort=rep(0,reps) ; left.sum=rep(0,reps) ;
  right.sum=rep(0,reps) ;total.sum=rep(0,reps) ;
  left.dev=rep(0,reps); right.dev=rep(0,reps) ;
  imp=rep(0,reps) ; cp=rep(0,reps);
G=function(x,y) {
# sıralı x,y lileri için döngüye girer.
left.x=matrix(0, reps, reps) ; left.y=matrix(0, reps, reps)
right.x=matrix(0, reps, reps) ; right.y=matrix(0, reps, reps)
# Her bir x değeri candidate partition değeri olarak düşünülür.
# Olası bölünmelerin matrisleri sol/sağ olarak ayrılır.
for(t in 1:reps ){
len= length(x[x < x[t]])
if (len==0) left.x[t,1:reps]= x
else {
left.x[t,1:len]=x[ x<x[t] ]
left.y[t,1:len]= y[x<x[t] ]
left.ort[t]= sum( (y[x<x[t]]-mean(y[x<x[t]])) ^2)
left.sum=rowSums(left.y)
}}
for(t in 1:reps ){
len= length(x[x>=x[t]])
if (len==0)right.x[t,1:reps]= x
else{
right.x[t,1:len]=x[x>=x[t]]
right.y[t,1:len]=y[x>=x[t]]
right.ort[t]= sum( (y[x>=x[t]]-mean(y[x>=x[t]])) ^2)
right.sum=rowSums(right.y)
}}
return(list(left.x=left.x, left.y=left.y,

```

```

right.x=right.x, right.y=right.y , left.dev=left.ort,
right.dev=right.ort, left.sum=left.sum, right.sum=right.sum)) }
G=G(x,y)
F=function(left.x,left.y,right.x,right.y)
{ for (i in 1:reps){
left.x=G$left.x ; left.y=G$left.y ;
      right.x=G$right.x ; right.y=G$right.y
left.dev=G$left.dev; right.dev=G$right.dev
left.sum=rowSums(left.y)
right.sum=rowSums(right.y)
total.dev=left.dev[1:length(left.sum)] + right.dev[1:length(right.sum)]
total.ss=sum((y-mean(y))^2)
cp=round((total.ss-(total.dev)) / BOF , 6)
imp=round( (total.ss-(total.dev)) / total.ss ,6)
bof.result=data.frame(left.dev,right.dev,imp,cp,total.dev,total.ss) }
#İkinci partition için seçilen noktanın x vektöründeki sırasını verir")
sira.0=which.min(bof.result[,5])
# İkinci partition point olarak seçilen knot
p.point.1=x[order(bof.result[,5])[1] ] ;
sira.1=which(sort(x)==p.point.1)
sira.2=sira.1-1
candi=(sort(x)[sira.1]+sort(x)[sira.2])/2
xx=x[x<candi] ## NODE 5
yy=y[x<candi] ## NODE 5
left.mean=mean(y[x<candi]) ;
left.mse=bof.result[sira.0,][1]/length(yy)
right.mean=mean(y[x>=candi]);
right.mse=bof.result[sira.0,][2]/length(y[x>=candi])
xx.r=x[x>=candi] ## NODE 4
yy.r=y[x>=candi] ## NODE 4
reps=length(yy)

```

```

leaves23=rbind(bof.result[sira.0,])
leaves23[,9:12]=cbind(left.mean, right.mean,
left.mse, right.mse)
colnames(leaves23)[9:12]=c("left.mean", "right.mean",
"left.mse", "right.mse")
# Kaçınıcı sıradaki x değerlerinin en küçük bof değerini verdiğini gösterir.
# En küçük bof değerini veren x lerin, vektördeki sırası
order ((bof.result[,5]))
i=1
while(order(bof.result[,5])[i]<6 | order(bof.result[,5])[i]>n-5)
i=i+1
x[order(bof.result[,5])[i]]
knot= x[order(bof.result[,5])[i]]
return(list(bof.result=bof.result, sira.0=sira.0, sira.1=sira.1,
sira.2=sira.2, p.point.1=p.point.1,i=i , knot=knot, candi=candi,
xx=xx, yy=yy, xx.r=xx.r, yy.r=yy.r, cp=cp,leaves23=leaves23)) }
F=F(left.x,left.y,right.x,right.y) ;F
tree=rbind(tree,F$leaves23); tree
for(i in 1:dim(tree)[1] )
if (tree[i,8]<0.01) agac=tree[-i,] ; agac
#SONUC
left.dev right.dev imp    cp total.dev total.ss
1 1.9957 0.3171 0.0146 0.0146 2.3129 2.3473
left.mean right.mean left.mse right.mse
1.0554 1.11406 0.06652 0.02114
#Check if correct
rpart(cd~x) ; summary(rpart(cd~x))

```

Ek-2 Yanıt değişkeninin, iki değişken kullanarak regresyon ağacının oluşturmasında kullanılan R kodları

```

y=cd ; x=A ;z=B; reps=length(y);n=length(y);library(rpart)
bof=rep(0, reps );
x.left.ort=rep(0,reps) ; x.right.ort=rep(0,reps)
x.left.sum=rep(0,reps) ; x.right.sum=rep(0,reps)
z.left.ort=rep(0,reps) ; z.right.ort=rep(0,reps)
z.left.sum=rep(0,reps) ; z.right.sum=rep(0,reps)
total.sum=rep(0,reps) ;left.dev=rep(0,reps);
right.dev=rep(0,reps) ;
ximp=rep(0,reps) ; xcp=rep(0,reps);
zimp=rep(0,reps) ; zcp=rep(0,reps);
BOF= sum( (y- mean(y)) ^2 ) ;BOF ;
mse=sum(lm(y~x+z)$residuals^2)/n ;
dev= deviance(lm(y~x+z) ) ; summary(rpart(y~x+z))
G=function(x,y,z) {
left.x=matrix(0, reps, reps) ; left.y=matrix(0, reps, reps)
right.x=matrix(0, reps, reps) ; right.y=matrix(0, reps, reps)
left.z=matrix(0,reps,reps) ; zleft.y=matrix(0, reps, reps)
right.z=matrix(0,reps,reps) ; zright.y=matrix(0, reps, reps)
# Her bir x değeri candidate partition değeri olarak düşünülür.
# Olası bölünmelerin matrisleri sol/sağ olarak ayrılır.
for(t in 1:reps ){
len= length(x[x < x[t]])
if (len==0) left.x[t,1:reps]= x
else {
left.x[t,1:len]=x[ x<x[t] ]
left.y[t,1:len]= y[x<x[t] ]
x.left.ort[t]= sum( (y[x<x[t]]-mean(y[x<x[t]])) ^2)
x.left.sum=rowSums(left.y)
}}
for(t in 1:reps ){
len= length(x[x>=x[t]])

```

```

if (len==0)right.x[t,1:reps]= x
else{
right.x[t,1:len]=x[x>=x[t]]
right.y[t,1:len]=y[x>=x[t]]
x.right.ort[t]= sum( (y[x>=x[t]]-mean(y[x>=x[t]])) ^2)
x.right.sum=rowSums(right.y)
}}

# x lerin ayırdığı y değerleri kümesi, sol
xsum.left=x.left.sum

# x lerin ayırdığı y değerleri kümesi, sağ
xsum.right=x.right.sum
for(t in 1:reps ){
len= length(z[z < z[t]])
if (len==0) left.z[t,1:reps]= z
else {
left.z[t,1:len]=z[ z<z[t] ]
zleft.y[t,1:len]= y[z<z[t] ]
z.left.ort[t]= sum( (y[z<z[t]]-mean(y[z<z[t]])) ^2)
# z lerin ayırdığı y değerlerinin kümesi, sol
z.left.sum=rowSums(zleft.y)
}}

for(t in 1:reps ){
len= length(z[z>=z[t]])
if (len==0)right.z[t,1:reps]= z
else{
right.z[t,1:len]=z[z>=z[t]]
zright.y[t,1:len]=y[z>=z[t]]
z.right.ort[t]= sum( (y[z>=z[t]]-mean(y[z>=z[t]])) ^2)
# z lerin ayırdığı y değerlerinin kümesi, sağ
z.right.sum=rowSums(zright.y)
}}

```

```

return(list(left.x=left.x,left.y=left.y,right.x=right.x,
right.y=right.y,left.z=left.z,right.z=right.z, zleft.y=zleft.y,
zright.y=zright.y,xdev.left=x.left.ort,xdev.right= x.right.ort,
zdev.left=z.left.ort,zdev.right=z.right.ort,zsum.left=z.left.sum,
zsum.right=z.right.sum,xsum.left=xsum.left,xsum.right=xsum.right)))}
G=G(x,y,z)

## F fonksiyonu NODE 2 ve NODE3  çıkartır.
F=function(
left.x,left.y,right.x,right.y,left.z,zleft.y,right.z,zright.y)
{ for (i in 1:reps){
left.x=G$left.x ; left.y=G$left.y ;
      right.x=G$right.x ; right.y=G$right.y
xdev.left=G$xdev.left; xdev.right=G$xdev.right
left.z=G$left.z ; right.z=G$right.z ;
      zleft.y=G$zleft.y; zright.y=G$zright.y
zdev.left=G$zdev.left; zdev.right=G$zdev.right
xsum.left=rowSums(left.y) ;xsum.right=rowSums(right.y)
      zsum.left=rowSums(zleft.y) ; zsum.right=rowSums(zright.y)
xtotal.dev=xdev.left[1:length(xsum.left)] +
xdev.right[1:length(xsum.right)]
xtotal.ss=sum((y-mean(y))^2)
ztotal.dev=zdev.left[1:length(zsum.left)] +
zdev.right[1:length(zsum.right)]
#ztotal.ss=sum((y-mean(y))^2)
xcp=round((xtotal.ss-(xtotal.dev)) / BOF , 6);
ximp=round( (xtotal.ss-(xtotal.dev)) / xtotal.ss ,6)
zcp=round((xtotal.ss-(ztotal.dev)) / BOF , 6);
zimp=round( (xtotal.ss-(ztotal.dev)) / xtotal.ss ,6)
bof.result=data.frame(xdev.left,xdev.right,
ximp,xcp,xtotal.dev,xtotal.ss);
gof.result=data.frame(zdev.left,zdev.right,

```

```

zimp,zcp,ztotal.dev,xtotal.ss)
}
# İlk partition için seçilen noktanın
# x vektöründeki sırasını verir
sira.0=which.min(bof.result[,5])
z.sira.0=which.min(gof.result[,5])
# İlk partition point olarak seçilen knot
p.point.1=x[order(bof.result[,5])[1] ] ;
sira.1=which(sort(x)==p.point.1)
sira.2=sira.1-1
z.point.1=z[order(gof.result[,5])[1]]
z.sira.1=which(sort(z)==z.point.1)
z.sira.2=z.sira.1-1
candi=((sort(x)[sira.1]+sort(x)[sira.2])/2)[1]
z.candi=((sort(z)[z.sira.1]+sort(z)[z.sira.2])/2)[1]
if (min(gof.result[,5]) <min(bof.result[,5])) {candi=z.candi
bak=2
leaves23=rbind(gof.result[z.sira.0,])
zz=z[z<candi] ## NODE 3
yz=y[z<candi] ## NODE 3
zleft.mean=mean(y[z<candi]) ;
zleft.mse=gof.result[z.sira.0,][1]/length(yz)
zz.r=z[z>=candi] ## NODE 2
yz.r=y[z>=candi] ## NODE 2
zright.mean=mean(y[z>=candi]);
zright.mse=gof.result[z.sira.0,][2]/length(y[z>=candi])
reps=length(yz.r)
leaves23[,7:10]=cbind(zleft.mean,zright.mean,zleft.mse,zright.mse)
colnames(leaves23)[7:10]=c("left.mean","right.mean",
"left.mse","right.mse")
print(gof.result) ;print( data.frame(zz.r , yz.r)) ;

```

```

print(data.frame( sira.0,sira.1,sira.2, p.point.1)) ;
print(leaves23) }
else{candi=candi
bak=1
leaves23=rbind(bof.result[sira.0,])
xx=x[x<candi] ## NODE 3
yx=y[x<candi] ## NODE 3
xleft.mean=mean(y[x<candi]) ;
xleft.mse=bof.result[sira.0,][1]/length(yx)
xx.r=x[x>=candi] ## NODE 2
yx.r=y[x>=candi] ## NODE 2
xright.mean=mean(y[x>=candi]);
xright.mse=bof.result[sira.0,][2]/length(y[x>=candi])
reps=length(yx.r)
leaves23[,7:10]=cbind(xleft.mean,xright.mean,xleft.mse,xright.mse)
colnames(leaves23)[7:10]=c("left.mean","right.mean",
"left.mse","right.mse")
print(bof.result) ; print(data.frame( xx.r,yx.r));
print( data.frame(sira.0,sira.1,sira.2,p.point.1));
print(leaves23) }
# Kaçınıcı sıradaki x değerlerinin en küçük bof
# değerini verdiğini gösterir.
# En küçük bof değerini veren x lerin, vektördeki sırası
order ((bof.result[,5]))
i=1
while(order(bof.result[,5])[i]<6 | order(bof.result[,5])[i]>n-5)
i=i+1
x[order(bof.result[,5])[i]] ; knot= x[order(bof.result[,5])[i]]
zz=z[z<candi] ## NODE 3
yz=y[z<candi] ## NODE 3
zz.r=z[z>=candi] ## NODE 2

```

```

yz.r=y[z>=candi] ## NODE 2
xx=x[x<candi] ## NODE 3
yx=y[x<candi] ## NODE 3
xx.r=x[x>=candi] ## NODE 2
yx.r=y[x>=candi] ## NODE 2
criterial1=data.frame( imp.z=zimp[z.sira.0] , imp.x=ximp[z.sira.0] ,
  cp.z=zcp[z.sira.0] , cp.x=xcp[z.sira.0])
return(list(leaves23=leaves23,zz=zz, zz.r= zz.r,yz=yz,yz.r=yz.r,
xx=xx,yx=yx,xx.r=xx.r,yx.r=yx.r, candi=candi ,
bak=bak , criterial1=criterial1))
}
F=F(left.x,left.y,right.x,right.y,left.z,right.z,zleft.y,zright.y);F
tree=F$leaves23
decide=function(F){
  if(F$bak==1){ if(F$bak==1 &
dim(data.frame(F$xx.r, F$yx.r))[1] > dim(data.frame(F$xx,F$yx))[1])
{d=data.frame(F$xx.r,F$yx.r, F$zz.r) ; d
} else{d=data.frame(F$xx,F$yx, F$zz) ;d}
  } else{
    if (F$bak==2){
      if( dim(data.frame(F$zz.r,F$yz.r))[1]>dim(data.frame(F$zz, F$yz)) [1])
      {d=data.frame(F$zz.r,F$yz.r,F$xx.r);d
      }else{d=data.frame(F$zz,F$yz,F$xx);d}
    }
  }
  return(list(d=d))
} ; R=decide (F) ;R ; dim(R$d)
# NODE 4 ve NODE 5 oluřturulur.
z=as.data.frame(R$d)[,1] ; y=as.data.frame(R$d)[,2] ;
x=as.data.frame(R$d)[,3] ; n=length(y) ;reps=length(y)
criterial1=F$criterial1 ;rpart(y~x+z)
reps=length(y) ;library(rpart)

```

```

bof=rep(0, reps );
x.left.ort=rep(0,reps) ; x.right.ort=rep(0,reps) ;
x.left.sum=rep(0,reps) ; x.right.sum=rep(0,reps)
z.left.ort=rep(0,reps) ; z.right.ort=rep(0,reps) ;
z.left.sum=rep(0,reps) ; z.right.sum=rep(0,reps)
total.sum=rep(0,reps) ; left.dev=rep(0,reps);right.dev=rep(0,reps);
ximp=rep(0,reps) ; xcp=rep(0,reps);
zimp=rep(0,reps) ; zcp=rep(0,reps);
G=function(x,y,z) {
left.x=matrix(0, reps, reps) ; left.y=matrix(0, reps, reps)
right.x=matrix(0, reps, reps) ; right.y=matrix(0, reps, reps)
left.z=matrix(0,reps,reps) ; zleft.y=matrix(0, reps, reps)
right.z=matrix(0,reps,reps) ; zright.y=matrix(0, reps, reps)
# Her bir x değeri candidate partition değeri olarak düşünülür.
# Olası bölünmelerin matrisleri sol/sağ olarak ayrılır.
for(t in 1:reps ){
len= length(x[x < x[t]])
if (len==0) left.x[t,1:reps]= x
else {
left.x[t,1:len]=x[ x<x[t] ]
left.y[t,1:len]= y[x<x[t] ]
x.left.ort[t]= sum( (y[x<x[t]]-mean(y[x<x[t]])) )^2)
x.left.sum=rowSums(left.y)
}}
for(t in 1:reps ){
len= length(x[x>=x[t]])
if (len==0)right.x[t,1:reps]= x
else{
right.x[t,1:len]=x[x>=x[t]]
right.y[t,1:len]=y[x>=x[t]]
x.right.ort[t]= sum( (y[x>=x[t]]-mean(y[x>=x[t]])) )^2)

```

```

x.right.sum=rowSums(right.y)
}}

# x lerin ayırdığı y değerleri kümesi, sol
xsum.left=x.left.sum

# x lerin ayırdığı y değerleri kümesi, sağ
xsum.right=x.right.sum

for(t in 1:reps ){
len= length(z[z < z[t]])
if (len==0) left.z[t,1:reps]= z
else {
left.z[t,1:len]=z[ z<z[t] ]
zleft.y[t,1:len]= y[z<z[t] ]
z.left.ort[t]= sum( (y[z<z[t]]-mean(y[z<z[t]])) ^2)
# z lerin ayırdığı y değerlerinin kümesi, sol
z.left.sum=rowSums(zleft.y)
}}

for(t in 1:reps ){
len= length(z[z>=z[t]])
if (len==0)right.z[t,1:reps]= z
else{
right.z[t,1:len]=z[z>=z[t]]
zright.y[t,1:len]=y[z>=z[t]]
z.right.ort[t]= sum( (y[z>=z[t]]-mean(y[z>=z[t]])) ^2)
# z lerin ayırdığı y değerlerinin kümesi, sağ
z.right.sum=rowSums(zright.y)
}}

return(list(left.x=left.x, left.y=left.y , right.x=right.x,
right.y=right.y , left.z=left.z, right.z=right.z, zleft.y=zleft.y,
zright.y=zright.y, xdev.left=x.left.ort, xdev.right= x.right.ort,
zdev.left=z.left.ort, zdev.right=z.right.ort,zsum.left=z.left.sum,
zsum.right=z.right.sum,xsum.left=xsum.left,xsum.right=xsum.right))

```

```

}
G=G(x,y,z)
## F fonksiyonu NODE 4 ve NODE 5  çıkartır.
F=function(
left.x,left.y,right.x,right.y,left.z,zleft.y,right.z,zright.y)
{ for (i in 1:reps){
left.x=G$left.x ; left.y=G$left.y ;
      right.x=G$right.x ; right.y=G$right.y
xdev.left=G$xdev.left; xdev.right=G$xdev.right
left.z=G$left.z ; right.z=G$right.z ;
      zleft.y=G$zleft.y; zright.y=G$zright.y
zdev.left=G$zdev.left; zdev.right=G$zdev.right
xsum.left=rowSums(left.y) ;xsum.right=rowSums(right.y)
zsum.left=rowSums(zleft.y) ; zsum.right=rowSums(zright.y)
xtotal.dev=xdev.left[1:length(xsum.left)] +
xdev.right[1:length(xsum.right)]
xtotal.ss=sum((y-mean(y))^2)
ztotal.dev=zdev.left[1:length(zsum.left)] +
zdev.right[1:length(zsum.right)]
#ztotal.ss=sum((y-mean(y))^2)
xcp=round((xtotal.ss-(xtotal.dev)) / BOF , 6);
ximp=round( (xtotal.ss-(xtotal.dev)) / xtotal.ss ,6)
zcp=round((xtotal.ss-(ztotal.dev)) / BOF , 6);
zimp=round( (xtotal.ss-(ztotal.dev)) / xtotal.ss ,6)
bof.result=data.frame(xdev.left, xdev.right ,
ximp, xcp , xtotal.dev, xtotal.ss);
gof.result=data.frame(zdev.left, zdev.right ,
zimp, zcp , ztotal.dev, xtotal.ss )
}
# İkinci partition için seçilen noktanın
# x vektöründeki sırasını verir

```

```

sira.0=which.min(bof.result[,5])
z.sira.0=which.min(gof.result[,5])
# İkinci partition point olarak seçilen knot
p.point.1=x[order(bof.result[,5])[1] ] ;
sira.1=which(sort(x)==p.point.1)
sira.2=sira.1-1
z.point.1=z[order(gof.result[,5])[1]]
z.sira.1=which(sort(z)==z.point.1)
z.sira.2=z.sira.1-1
candi=((sort(x)[sira.1]+sort(x)[sira.2])/2)[1]
z.candi=((sort(z)[z.sira.1]+sort(z)[z.sira.2])/2)[1]
if(min(gof.result[,5]) <min(bof.result[,5])){candi=z.candi
bak=2
leaves23=rbind(gof.result[z.sira.0,])
zz=z[z<candi] ## NODE 5
yz=y[z<candi] ## NODE 5
zleft.mean=mean(y[z<candi]) ;
zleft.mse=gof.result[z.sira.0,][1]/length(yz)
zz.r=z[z>=candi] ## NODE 4
yz.r=y[z>=candi] ## NODE 4
zright.mean=mean(y[z>=candi]);
zright.mse=gof.result[z.sira.0,][2]/length(y[z>=candi])
reps=length(yz.r)
leaves23[,7:10]=cbind(zleft.mean,zright.mean,zleft.mse,zright.mse)
colnames(leaves23)[7:10]=c("left.mean","right.mean",
"left.mse","right.mse")
print(gof.result) ;print( data.frame(zz.r , yz.r)) ;
print(data.frame( sira.0,sira.1,sira.2, p.point.1) ) ;
print(leaves23) }
else{candi=candi
bak=1

```

```

leaves23=rbind(bof.result[sira.0,])
xx=x[x<candi] ## NODE 5
yx=y[x<candi] ## NODE 5
xleft.mean=mean(y[x<candi]) ;
xleft.mse=bof.result[sira.0,][1]/length(yx)
xright.mean=mean(y[x>=candi]);
xright.mse=bof.result[sira.0,][2]/length(y[x>=candi])
xx.r=x[x>=candi] ## NODE 4
yx.r=y[x>=candi] ## NODE 4
reps=length(yx.r)
leaves23[,7:10]=cbind(xleft.mean,xright.mean,xleft.mse,xright.mse)
colnames(leaves23)[7:10]=c("left.mean", "right.mean",
  "left.mse", "right.mse")
print(bof.result) ; print(data.frame( xx.r,yx.r));
print( data.frame(sira.0,sira.1,sira.2,p.point.1)) ;
print(leaves23)  }

# Kaçınıcı sıradaki x değerlerinin en küçük
# bof değerini verdiğini gösterir.
# En küçük bof değerini veren x lerin, vektördeki sırası ;
order ((bof.result[,5]))
i=1
  while(order(bof.result[,5])[i]<6 | order(bof.result[,5])[i]>n-5)
i=i+1
x[order(bof.result[,5])[i]] ; knot= x[order(bof.result[,5])[i]]
zz=z[z<candi] ## NODE 5
yz=y[z<candi] ## NODE 5
zz.r=z[z>=candi] ## NODE 4
yz.r=y[z>=candi] ## NODE 4
xx=x[x<candi] ## NODE 5
yx=y[x<candi] ## NODE 5
xx.r=x[x>=candi] ## NODE 4

```

```

yx.r=y[x>=candi] ## NODE 4
criteria2=data.frame( imp.z=zimp[z.sira.0], imp.x=ximp[z.sira.0],
cp.z=zcp[z.sira.0] , cp.x=xcp[z.sira.0])
return(list(leaves23=leaves23,zz=zz, zz.r= zz.r,yz=yz,
yz.r=yz.r,xx=xx,yx=yx,xx.r=xx.r,yx.r=yx.r,
candi=candi,bak=bak,criteria2=criteria2))
}

F=F(left.x,left.y,right.x,right.y,left.z,right.z,zleft.y,zright.y);F
decide=function(F){
if(F$bak==1){ if(F$bak==1 &
dim(data.frame(F$xx.r, F$yx.r))[1] > dim(data.frame(F$xx,F$yx))[1])
{d=data.frame(F$xx.r,F$yx.r, F$zz.r) ; d
} else{d=data.frame(F$xx,F$yx, F$zz) ;d}
} else{
if (F$bak==2){
if( dim(data.frame(F$zz.r,F$yz.r)) [1] > dim(data.frame(F$zz,F$yz))[1])
{d=data.frame(F$zz.r,F$yz.r,F$xx.r) ;d
} else{d=data.frame(F$zz,F$yz,F$xx) ;d}
}}
return(list(d=d))
} ; R=decide (F) ;R ; dim(R$d)
tree=rbind(tree, F$leaves23) ;tree
cp.change=(tree[1,6]- (tree[2,1]+tree[2,2]+tree[1,2]))/(2*tree[1,6])
agac=function(tree){
if(dim(tree)[1]==2) {tree[1,4]=cp.change ;tree} else { tree}
return(tree)
} ; agac(tree)
#SONUC

```

	zdev.left	zdev.right	zimp	zcp	ztotal.dev	xtotal.ss
7	2.017797	0.1793333	0.063998	0.1037537	2.197130	2.347356
2	0.755055	0.9258737	0.166948	0.1435100	1.680929	2.017797

```
left.mean right.mean left.mse right.mse
1.034100 1.1566667 0.06725989 0.01195556
1.140067 0.9281333 0.05033700 0.06172492
#Check if correct
rpart(cd~A+B);summary(rpart(cd~A+B))
```

NOT: if/else komutları, sayfa marjinleri yüzünden iki satıra bölünerek yazılmıştır. Bu bakımdan, programı R-Software konsoluna yazarken tek satırda yazılması tavsiye edilir, aksi hali sorun çıkartabilmektedir.

Ek-3 Cd yanıt değişkeninin, tek değişken kullanarak MARS modelinin oluşturmada kullanılan R kodları

```
cd=(1.195 0.990 1.195 1.395 0.597 0.990 1.399 0.597 1.395 1.199 0.996 0.597
1.195 0.999 0.996 0.598 1.200 0.990 0.996 0.999 1.198 1.199 1.395 1.200 1.200
1.198 1.198 0.990 1.192 1.395 0.988 1.192 1.192 1.199 0.790 0.990 1.198 0.990
0.996 0.992 1.584 0.993 0.992 0.598 0.996)
z=(0 -1 0 -1 0 -1 1 0 -1 1 1 0 0 -1 1 0 1 -1 1 -1 1 1 1 0 1 -1 -1 1 1 -1
-1 0 0 0 -1 0 -1 1 0 0 -1 )
x= (1 1 1 -1 -1 1 1 -1 -1 1 0 -1 1 0 0 -1 -1 1 0 0 1 1 -1 -1 -1 -1 -1 0 0 -1 1 0 1
0 1 0 1 0 0 0 -1 1 0 -1 0)

single.mars=function(y,z) {
R.m<-function(z,t) ifelse(t>=z, t-z, 0)
R.j<-function(t,z) ifelse(z>=t, z-t, 0)
coef.1=lm(cd~R.m(z,0)+R.j(0,z))$coef
coef.2=lm(cd~R.m(z,-1)+R.j(-1,z))$coef
coef.3=lm(cd~R.m(z,1)+R.j(1,z))$coef
coef.mat.z=data.frame(coef.1,coef.2,coef.3);coef.mat.z
for(i in 1:3) {
  if(any(is.na(coef.mat.z[i,]))){
    # print(which((is.na(coef.mat.z[i,]))))
coef.matz=coef.mat.z[which(any(!is.na(coef.mat.z[i,])))]
y.fitz= coef.matz[1,1]+coef.matz[2,1]*R.m(z,0)+coef.matz[3,1]*R.j(0,z)
} }

fit.1=lm(cd~R.m(z,0)+R.j(0,z))
fit.2=lm(cd~R.m(z,-1)+R.j(-1,z))
fit.3=lm(cd~R.m(z,1)+R.j(1,z))
rss.1=sum((y-predict(fit.1))^2)
rss.2=sum((y-predict(fit.2))^2)
rss.3=sum((y-predict(fit.3))^2)
```

```

rss=c(rss.1, rss.2, rss.3)
for (i in 1:3){
  Penalty=2
  nTerms=length(coef.1)
  nCases=length(cd)
  nKnots =(nTerms-1) / 2
  cost = nTerms + Penalty * nKnots
  gcv=(rss/nCases)/(1-cost/nCases)**2
  mygcv=min(gcv)
  myrss=min(rss)
}
fit=function(mygcv,gcv){
  if (mygcv==gcv[1]) { model=fit.1; r.sq=summary(fit.1)$r.sq
} else {if (mygcv==gcv[2] ) {model=fit.2; r.sq=summary(fit.2)$r.sq
      } else{if(mygcv==gcv[3]) {model=fit.3; r.sq=summary(fit.1)$r.sq }}}
  return(list(model=model,r.sq=r.sq))}
return(list(fit(mygcv,gcv), gcv=min(gcv), rss=min(rss)))
}
#SONUC
single.mars(cd,z)
[[1]]
[[1]]$model
Call:
lm(formula = cd ~ R.m(z, 0) + R.j(0, z))
Coefficients:
(Intercept)      R.m(z, 0)      R.j(0, z)
      0.9281        0.2119        0.2285
[[1]]$r.sq
[1] 0.2075075
$gcv
[1] 0.05231987

```

```
$rss
```

```
[1] 1.860262
```

Ek-4 Cd yanıt değişkeninin, iki değişken kullanarak MARS modelinin oluşturmada kullanılan R kodları

```
y=cd ; x=A ; z=B
multiple.mars=function(y,z,x){
  R.m<-function(z,t) ifelse(t>=z, t-z, 0)
  R.j<-function(t,z) ifelse(z>=t, z-t, 0)
  coef.1=lm(cd~R.m(z,0)+R.j(0,z))$coef
  coef.2=lm(cd~R.m(z,-1)+R.j(-1,z))$coef
  coef.3=lm(cd~R.m(z,1)+R.j(1,z))$coef
  coef.4= lm(cd~R.m(x,0)+R.j(0,x))$coef
  coef.5= lm(cd~R.m(x,-1)+R.j(-1,x))$coef
  coef.6= lm(cd~R.m(x,1)+R.j(1,x))$coef
  coef.mat.z=data.frame(coef.1,coef.2,coef.3);coef.mat.z
  coef.mat.x=data.frame(coef.4, coef.5,coef.6) ;coef.mat.x
  for(i in 1:3) {
    if(any(is.na(coef.mat.z[i,]))) {
      # print(which((is.na(coef.mat.z[i,]))))
    }
    coef.matz=coef.mat.z[which(any(!is.na(coef.mat.z[i,])))]
    y.fitz= coef.matz[1,1]+coef.matz[2,1]*R.m(z,0)+coef.matz[3,1]*R.j(0,z)
  }
  for(i in 1:3) {
    if(any(is.na(coef.mat.x[i,]))) {
      # print(which((is.na(coef.mat.x[i,]))))
    }
    coef.matx=coef.mat.x[which(any(!is.na(coef.mat.x[i,])))]
    y.fitx=coef.matx[1,1]+coef.matx[2,1]*R.m(x,0)+coef.matx[3,1]*R.j(0,x)
  }
  fit.1=lm(cd~R.m(z,0)+R.j(0,z));fit.2=lm(cd~R.m(z,-1)+R.j(-1,z))
```

```

fit.3=lm(cd~R.m(z,1)+R.j(1,z));fit.4=lm(cd~R.m(x,0)+R.j(0,x))
fit.5=lm(cd~R.m(x,-1)+R.j(-1,x));fit.6=lm(cd~R.m(x,1)+R.j(1,x))
rss.1=sum(fit.1$res^2);rss.2=sum(fit.2$res^2)
rss.3=sum(fit.3$res^2);rss.4=sum(fit.4$res^2)
rss.5=sum(fit.5$res^2);rss.6=sum(fit.6$res^2)
rss=c(rss.1, rss.2, rss.3, rss.4, rss.5, rss.6)
for (i in 1:6){
  Penalty=2
  nTerms=length(coef.1)
  nCases=length(cd)
  nKnots =(nTerms-1) / 2
  cost = nTerms + Penalty * nKnots
  gcv=(rss/nCases)/(1-cost/nCases)**2
  mygcv=min(gcv)
  myrss=min(rss)
}
fit=function(mygcv,gcv){
  if (mygcv==gcv[1]) { model=fit.1; r.sq=summary(fit.1)$r.sq
} else {if (mygcv==gcv[2] ) {model=fit.2; r.sq=summary(fit.2)$r.sq
} else{if(mygcv==gcv[3]) {model=fit.3; r.sq=summary(fit.1)$r.sq
} else{ if(mygcv==gcv[4]) {model=fit.4;r.sq=summary(fit.4)$r.sq
} else{ if(mygcv==gcv[5]) {model=fit.5;r.sq=summary(fit.5)$r.sq
} else { if(mygcv==gcv[6]) {model=fit.6;r.sq=summary(fit.6)$r.sq
}}}}}}
return(list(model=model,r.sq=r.sq))}
return(list(fit(mygcv,gcv), gcv=min(gcv), rss=min(rss)))
}
#SONUC
  multiple.mars(cd,z,x)
[[1]]
[[1]]$model

```

```

Call:
lm(formula = cd ~ R.m(z, 0) + R.j(0, z))

Coefficients:
(Intercept)      R.m(z, 0)      R.j(0, z)
      0.9281         0.2119         0.2285

[[1]]$r.sq
[1] 0.2075075

$gcv
[1] 0.05231987

$rss
[1] 1.860262

```

Ek-5 MARS Modelleri- ANOVA

1- Cd- MARS Model 2 - Birim sayısı 0 - Etkileşim Var

N: 45.00 R²: 0.83806
Ortalama= 1.07496 ADJ R²: 0.81730

PARAMETRE	TAHMIN	S.H.	T-ORANI	P-DEGERI
Sabit	0.62873	0.04031	15.59935	0.00000
Taban Fonksiyon 1	0.52933	0.05700	9.28655	0.00000
Taban Fonksiyon 2	0.75293	0.05700	13.20935	0.00000
Taban Fonksiyon 3	-0.54100	0.04415	-12.25312	0.00000
Taban Fonksiyon 4	0.29940	0.03122	9.58995	0.00000
Taban Fonksiyon 5	-0.30080	0.04415	-6.81282	0.00000

F-hesap = 40.36568

P-olasılığı = 0.00000 HATA K.T. = 0.38013

S.D. = [5, 39] REGRESYON K.T.= 1.96722

2- Cd-MARS Model 3 - Birim Sayısı 1 - Etkileşim Var

N: 45.00 R²: 0.81993
Ortalama: 1.07496 ADJ R²: 0.80192

PARAMETRE	TAHMIN	S.H.	T-ORANI	P-Değeri
Sabit	0.67497	0.03885	17.37200	0.00000
Taban Fonksiyon 2	0.70670	0.05719	12.35664	0.00000
Taban Fonksiyon 3	-0.52166	0.04256	-12.25629	0.00000
Taban Fonksiyon 4	0.28006	0.02747	10.19356	0.00000

Taban Fonksiyon 6 0.52423 0.06019 8.70922 0.00000

F-Değeri = 45.53282

P-Olasılığı = 0.00000 HATA K.T. = 0.42270

S.D. = [4, 40] REGRESYON K.T. = 1.92466

3- Zn- MARS Model 1- Birim sayısı 0- Etkileşim Yok

N: 45.00

R²: 0.82557

Ortalama: 65.08958

ADJ R²: 0.81281

PARAMETRE	TAHMIN	S.H.	T-ORANI	P-Değeri
Sabit	117.55832	4.91257	23.93012	0.00000
Taban Fonksiyon 1	-40.11579	6.43206	-6.23685	0.00000
Taban Fonksiyon 2	-75.27046	6.43206	-11.70239	0.00000
Taban Fonksiyon 3	-42.01996	5.57033	-7.54354	0.00000

F-Değeri = 64.68305

P-Olasılığı = 0.00000 HATA K.T. = 12721.70416

S.D. = [3, 41] REGRESYON K.T. = 60210.63039

4- Zn- MARS Model 2 - Birim sayısı 0 -Etkileşim Var

N: 45.00

R²: 0.99493

Ortalama: 65.08958

ADJ R²: 0.99413

PARAMETRE	TAHMIN	S.H.	T-ORANI	P-Değeri
-----------	--------	------	---------	----------

Sabit	131.56030	0.98614	133.40905	0.00000
Taban Fonksiyon 1	-77.48789	1.70805	-45.36634	0.00000
Taban Fonksiyon 2	-89.73809	1.39462	-64.34611	0.00000
Taban Fonksiyon 3	-98.71499	2.14925	-45.93004	0.00000
Taban Fonksiyon 5	19.36400	0.71642	27.02900	0.00000
Taban Fonksiyon 6	14.68910	1.30454	11.25995	0.00000
Taban Fonksiyon 7	6.55587	0.65743	9.97199	0.00000

F-Değeri = 1243.60836

P-Olasılığı = 0.00000

HATA K.T. = 369.54107

S.D. = [6, 38]

REGRESYON K.T. = 72562.79349

5- Zn- MARS Model 3 - Birim sayısı 1 -Etkileşim Var

N: 45.00

R²: 0.99493

Ortalama: 65.08958

ADJ R²: 0.99413

PARAMETRE	TAHMIN	S.H.	T-ORANI	P-Değeri
Sabitt	131.56030	0.98614	133.40905	0.00000
Taban Fonksiyonu 1	-77.48789	1.70805	-45.36634	0.00000
Taban Fonksiyonu 2	-89.73809	1.39462	-64.34611	0.00000
Taban Fonksiyonu 3	-84.02589	1.70805	-49.19410	0.00000
Taban Fonksiyonu 5	92.44868	2.60909	35.43335	0.00000
Taban Fonksiyonu 6	43.40289	2.41554	17.96816	0.00000
Taban Fonksiyonu 7	19.66759	1.97228	9.97199	0.00000

F-Değeri = 1243.60836

P-Olasılığı = 0.00000

HATA K.T. = 369.54107

S.D. = [6, 38]

REGRESYON K.T.= 72562.79349

6- Pb- MARS Model 1 - Birim sayısı 0 -Etkileşim Yok

N: 45.00

R²: 0.62341

Ortalama: 27.37253

ADJ R²: 0.60548

PARAMETRE	TAHMIN	S.H.	T-ORANI	P-Değeri
Sabit	18.32563	3.87899	4.72433	0.00003
Taban Fonksiyonu 1	17.52866	2.74286	6.39066	0.00000
Taban Fonksiyonu 2	25.44530	4.75077	-5.35604	0.00000

F-Değeri = 34.76381

P-Olasılığı = 0.00000

HATA K.T.= 9479.32335

S.D.= [2, 42]

REGRESYON K.T. = 15692.25760

7- Pb- MARS Model 2 - Birim sayısı 0 -Etkileşim Var

N: 45.00

R²: 0.85190

Ortalama: 27.37253

ADJ R²: 0.84107

PARAMETRE	TAHMIN	S.H.	T-RATIO	P-Değeri
Sabit	9.84387	2.24751	4.37990	0.00008
Taban Fonksiyonu 1	34.65768	2.33568	14.83837	0.00000
Taban Fonksiyonu 2	-30.64019	2.69701	-11.36080	0.00000
Taban Fonksiyonu 3	-20.74683	2.69701	-7.69253	0.00000

F-Değeri = 78.61464

P-Olasılığı = 0.00000

HATA K.T.= 3727.85808

S.D.= [3, 41]

REGRESYON K.T. = 21443.72287

Ek-6 Toz ve Toprak Veri seti

Tramvay Taşıt Cd-toz Pb-toz Zn-toz Cd-toprak Pb-toprak Zn-toprak

İstasyon

12	1212	1.392	33.579	118.735	1.394	17.650	47.401	Uluönder
12	1212	1.392	35.170	120.326	1.394	17.247	47.600	Uluönder
12	1212	1.392	35.766	119.928	1.394	17.650	48.397	Uluönder
12	1212	1.077	30.946	111.287	1.376	19.657	51.711	Uluönder
12	1212	1.077	29.565	110.300	1.180	20.443	51.907	Uluönder
12	1212	1.879	30.551	108.721	1.376	21.426	52.497	Uluönder
12	1212	1.198	33.520	106.451	1.393	18.478	50.358	Uluönder
12	1212	1.398	34.119	107.250	1.393	19.274	51.354	Uluönder
12	1212	1.398	33.520	106.451	1.194	19.473	51.154	Uluönder
12	1212	0.996	25.256	91.000	0.992	16.234	44.436	Uluönder
12	1212	0.996	22.667	89.407	0.992	15.837	44.039	Uluönder
12	1212	0.996	23.862	90.601	0.992	16.234	44.436	Uluönder
12	1212	0.996	22.477	111.554	0.983	14.315	42.649	Uluönder
12	1212	0.996	23.473	110.558	0.983	14.315	42.256	Uluönder
12	1212	0.996	22.676	110.159	0.983	15.101	42.453	Uluönder
12	1212	1.000	23.557	97.161	1.000	14.167	42.000	Uluönder
12	1212	1.000	21.558	97.161	1.000	13.967	42.200	Uluönder
12	1212	1.000	22.957	97.161	1.000	13.967	42.000	Uluönder
18	420	0.995	22.191	87.744	0.995	13.099	40.191	Bağlar
18	420	0.796	23.186	88.142	0.995	12.303	39.395	Bağlar
18	420	0.995	25.574	88.739	0.796	13.695	39.395	Bağlar
18	420	0.992	17.236	88.527	0.797	11.919	39.243	Bağlar
18	420	0.992	18.228	89.321	0.598	10.724	39.641	Bağlar
18	420	1.191	17.434	89.718	0.598	11.122	39.442	Bağlar
18	420	0.990	20.153	88.462	1.377	12.553	37.365	Bağlar
18	420	0.990	20.153	87.671	1.377	14.913	38.151	Bağlar
18	420	0.990	19.559	88.264	1.377	13.340	37.955	Bağlar

18 420 0.994 15.079 100.616 0.595 10.278 37.874 Bağlar
 18 420 0.994 15.278 102.406 0.595 11.468 38.469 Bağlar
 18 420 0.994 14.483 101.810 0.595 11.270 37.874 Bağlar
 18 420 0.797 20.294 85.492 0.598 10.936 41.085 Bağlar
 18 420 0.797 18.301 84.496 0.598 11.335 40.088 Bağlar
 18 420 0.797 20.094 85.492 0.598 11.534 41.284 Bağlar
 18 420 0.799 23.548 94.667 0.597 10.713 39.801 Bağlar
 18 420 0.799 22.749 92.471 0.597 11.509 39.403 Bağlar
 18 420 0.799 23.747 91.472 0.597 10.514 38.408 Bağlar
 12 1206 1.198 30.923 152.586 1.395 9.929 45.627 Doktorlar
 12 1206 0.999 31.523 151.987 1.395 9.331 46.822 Doktorlar
 12 1206 1.198 31.123 151.588 1.395 11.523 45.627 Doktorlar
 12 1206 1.186 25.461 145.455 1.194 11.511 45.780 Doktorlar
 12 1206 1.186 26.252 146.245 1.194 11.511 44.188 Doktorlar
 12 1206 0.988 26.449 145.455 1.194 11.113 45.780 Doktorlar
 12 1206 1.194 29.414 146.041 1.198 10.749 50.120 Doktorlar
 12 1206 1.194 32.398 147.433 1.198 10.350 49.720 Doktorlar
 12 1206 1.194 30.608 147.035 1.198 11.149 50.120 Doktorlar
 12 1206 1.797 28.726 169.163 0.994 10.110 46.142 Doktorlar
 12 1206 1.598 30.124 170.162 0.994 11.303 45.744 Doktorlar
 12 1206 1.598 27.528 171.560 0.994 10.707 45.943 Doktorlar
 12 1206 1.187 31.217 153.481 1.192 10.301 49.285 Doktorlar
 12 1206 0.989 30.426 150.712 1.192 10.102 48.490 Doktorlar
 12 1206 0.989 30.030 151.503 1.192 12.288 49.086 Doktorlar
 12 1206 1.190 25.562 146.230 0.996 10.728 46.433 Doktorlar
 12 1206 1.190 24.967 143.849 0.996 9.333 47.828 Doktorlar
 12 1206 1.190 26.753 145.833 0.996 9.732 45.835 Doktorlar
 12 984 0.993 38.482 101.052 1.189 64.982 120.912 Alanönü
 12 984 0.993 36.695 99.067 0.991 65.576 120.515 Alanönü
 12 984 0.993 36.695 99.861 0.991 66.766 120.714 Alanönü
 12 984 0.998 29.508 105.988 1.184 49.118 129.293 Alanönü

12 984 0.998 30.506 106.587 1.184 49.513 132.452 Alanönü
 12 984 0.998 29.508 106.387 1.184 50.105 130.478 Alanönü
 12 984 0.994 32.975 119.308 0.987 42.020 129.319 Alanönü
 12 984 0.994 35.362 119.507 1.185 42.218 130.701 Alanönü
 12 984 0.994 35.560 118.115 1.185 41.626 128.332 Alanönü
 12 984 0.997 36.118 99.422 0.988 50.925 134.703 Alanönü
 12 984 0.997 36.717 98.425 1.185 51.320 133.518 Alanönü
 12 984 0.997 38.112 99.023 1.185 51.715 135.690 Alanönü
 12 984 0.793 37.841 93.000 0.992 45.800 120.437 Alanönü
 12 984 0.793 36.453 91.414 0.992 47.586 122.222 Alanönü
 12 984 0.991 41.014 93.992 0.992 46.594 121.230 Alanönü
 12 984 0.990 50.838 98.377 0.990 41.139 132.621 Alanönü
 12 984 0.990 49.254 96.991 0.990 41.337 133.610 Alanönü
 12 984 0.990 49.650 98.773 0.990 42.326 134.402 Alanönü
 12 1284 1.183 38.209 138.577 1.390 14.261 58.368 Atatürk Lisesi
 12 1284 1.183 38.012 137.197 1.390 16.644 58.567 Atatürk Lisesi
 12 1284 1.183 41.560 138.183 1.390 15.055 59.162 Atatürk Lisesi
 12 1284 1.505 42.343 140.594 1.177 20.956 60.416 Atatürk Lisesi
 12 1284 1.703 43.729 139.604 1.177 21.937 60.220 Atatürk Lisesi
 12 1284 1.505 43.729 139.802 1.177 23.310 60.220 Atatürk Lisesi
 12 1284 1.182 39.557 131.968 1.000 15.963 60.188 Atatürk Lisesi
 12 1284 1.182 39.557 131.968 1.000 14.764 61.388 Atatürk Lisesi
 12 1284 1.182 39.361 129.604 1.000 14.764 60.188 Atatürk Lisesi
 12 1284 0.999 41.509 129.619 0.995 21.862 65.486 Atatürk Lisesi
 12 1284 0.999 41.309 128.820 1.194 20.667 64.490 Atatürk Lisesi
 12 1284 1.198 41.109 126.623 0.995 21.862 65.884 Atatürk Lisesi
 12 1284 1.189 39.404 157.551 0.993 15.660 63.369 Atatürk Lisesi
 12 1284 1.189 39.801 158.740 0.993 16.455 62.972 Atatürk Lisesi
 12 1284 1.189 39.404 157.352 1.192 16.653 62.773 Atatürk Lisesi
 12 1284 0.986 32.987 132.945 1.199 21.749 62.310 Atatürk Lisesi
 12 1284 0.986 34.566 133.340 1.199 21.949 61.711 Atatürk Lisesi

12 1284 0.986 32.987 133.340 1.399 21.150 62.710 Atatürk Lisesi
 12 174 0.990 54.984 36.810 1.734 7.484 38.543 OGÜ
 12 174 0.792 57.557 36.414 1.542 9.218 37.965 OGÜ
 12 174 0.792 57.557 35.820 1.734 8.447 38.736 OGÜ
 12 174 0.828 55.582 33.174 1.576 8.046 43.744 OGÜ
 12 174 0.828 47.636 32.777 1.576 8.834 42.956 OGÜ
 12 174 0.828 47.636 32.976 1.576 7.258 43.350 OGÜ
 12 174 0.800 35.959 28.794 1.767 9.197 43.598 OGÜ
 12 174 0.800 35.360 28.994 1.767 8.412 43.401 OGÜ
 12 174 0.800 35.560 29.194 1.571 8.412 42.616 OGÜ
 12 174 0.793 42.623 40.230 1.393 10.514 42.388 OGÜ
 12 174 0.793 42.226 40.032 1.393 9.718 42.388 OGÜ
 12 174 0.793 48.857 39.834 1.393 9.320 42.189 OGÜ
 12 174 0.599 40.070 36.113 1.386 8.086 46.931 OGÜ
 12 174 0.798 41.068 36.313 1.584 6.898 47.129 OGÜ
 12 174 0.599 40.270 36.512 1.584 9.076 46.337 OGÜ
 12 174 0.797 41.011 40.247 1.395 10.132 42.057 OGÜ
 12 174 0.797 41.210 39.052 1.395 9.933 41.658 OGÜ
 12 174 0.797 40.413 39.251 1.395 8.737 42.057 OGÜ
 18 1100 1.393 44.336 76.204 2.183 74.004 54.387 Atatürk Blv Çilek Ö.
 18 1100 1.393 42.943 74.811 2.183 71.821 54.784 Atatürk Blv Çilek Ö.
 18 1100 1.194 46.525 75.209 2.183 74.600 55.181 Atatürk Blv Çilek Ö.
 18 1100 0.994 51.448 98.986 1.198 69.029 50.299 Atatürk Blv Çilek Ö.
 18 1100 0.994 53.038 98.589 1.198 69.428 50.100 Atatürk Blv Çilek Ö.
 18 1100 0.994 49.858 98.191 1.397 69.428 49.501 Atatürk Blv Çilek Ö.
 18 1100 1.599 68.326 71.757 1.793 70.897 64.355 Atatürk Blv Çilek Ö.
 18 1100 1.599 70.125 71.757 1.594 70.897 64.555 Atatürk Blv Çilek Ö.
 18 1100 1.599 69.725 70.758 1.594 70.499 64.355 Atatürk Blv Çilek Ö.
 18 1100 0.789 54.186 98.580 1.194 70.588 55.334 Atatürk Blv Çilek Ö.
 18 1100 0.789 53.398 98.580 1.393 73.574 56.728 Atatürk Blv Çilek Ö.
 18 1100 0.789 54.778 99.369 1.194 78.995 55.334 Atatürk Blv Çilek Ö.

18 1100 0.793 48.926 81.665 1.198 71.185 56.498 Atatürk Blv Çilek Ö.
 18 1100 0.793 49.521 82.061 1.198 97.391 55.300 Atatürk Blv Çilek Ö.
 18 1100 0.793 48.332 82.061 1.198 71.384 55.500 Atatürk Blv Çilek Ö.
 18 1100 0.786 71.929 85.726 1.195 76.660 52.789 Atatürk Blv Çilek Ö.
 18 1100 0.786 69.767 85.332 1.195 76.062 53.187 Atatürk Blv Çilek Ö.
 18 1100 0.786 69.767 84.742 1.195 76.062 53.586 Atatürk Blv Çilek Ö.
 12 852 1.993 26.703 145.251 1.182 44.790 156.882 Vişnelik
 12 852 1.793 27.900 145.650 1.182 46.774 153.114 Vişnelik
 12 852 1.793 26.703 144.254 1.182 45.584 154.899 Vişnelik
 12 852 1.194 28.811 101.452 1.381 41.584 127.022 Vişnelik
 12 852 1.194 28.015 101.850 1.381 41.387 129.389 Vişnelik
 12 852 1.194 28.811 100.656 1.381 43.951 128.008 Vişnelik
 12 852 1.193 22.233 103.579 1.792 41.194 127.664 Vişnelik
 12 852 1.193 22.631 104.573 1.992 41.990 129.058 Vişnelik
 12 852 1.193 21.438 104.374 1.792 38.605 127.863 Vişnelik
 12 852 0.794 28.268 99.821 0.993 40.666 130.832 Vişnelik
 12 852 0.794 30.451 100.020 0.993 43.842 133.214 Vişnelik
 12 852 0.794 28.070 98.631 0.993 42.850 132.222 Vişnelik
 12 852 1.197 31.485 98.943 0.996 49.578 135.286 Vişnelik
 12 852 1.197 32.083 98.943 0.996 48.582 137.478 Vişnelik
 12 852 1.197 29.689 98.544 0.996 49.977 137.278 Vişnelik
 12 852 0.796 22.255 106.070 1.199 40.942 133.120 Vişnelik
 12 852 0.796 21.857 107.463 0.999 41.142 130.722 Vişnelik
 12 852 0.796 21.658 108.060 0.999 41.342 132.920 Vişnelik
 12 1530 1.360 16.571 36.371 1.188 46.437 71.201 Opera
 12 1530 1.360 16.769 37.161 1.188 48.631 71.400 Opera
 12 1530 1.360 16.966 36.568 1.188 48.232 71.201 Opera
 12 1530 1.946 18.909 33.938 1.181 25.145 81.197 Opera
 12 1530 1.946 18.712 33.938 1.181 25.740 79.810 Opera
 12 1530 1.946 16.542 33.149 1.181 24.551 78.818 Opera
 12 1530 1.992 13.313 32.072 1.985 23.384 68.069 Opera

12	1530	1.793	13.313	31.076	1.985	24.178	67.871	Opera
12	1530	1.992	13.712	31.673	1.985	25.170	67.672	Opera
12	1530	0.798	13.928	30.913	1.000	31.960	74.973	Opera
12	1530	0.798	13.329	31.512	1.000	31.360	75.173	Opera
12	1530	0.798	14.127	30.714	1.000	31.960	75.773	Opera
12	1530	0.992	14.247	32.923	0.790	24.666	73.503	Opera
12	1530	0.992	14.049	34.113	0.790	25.061	74.096	Opera
12	1530	0.992	14.049	33.320	0.988	24.468	72.910	Opera
12	1530	0.993	18.711	35.508	0.990	28.866	74.228	Opera
12	1530	0.993	18.711	33.521	0.990	28.669	73.436	Opera
12	1530	0.993	18.115	35.905	0.990	29.658	74.030	Opera