T.C.
MERSİN ÜNİVERSİTESİ
SOSYAL BİLİMLER ENSTİTÜSÜ
İŞLETME BİLGİ YÖNETİMİ
ANABİLİM DALI

BAĞLANTISALLIK PROBLEMİNİN CEZALI REGRESYON YÖNTEMLERİ İLE GİDERİLMESİ

Yüksek Lisans Tezi

Hazırlayan
Emel CİĞER

Danışman
Doç. Dr. Evrim Ersin KANGAL

ARALIK-2023, MERSİN

T.C.
MERSİN ÜNİVERSİTESİ
SOSYAL BİLİMLER ENSTİTÜSÜ
İŞLETME BİLGİ YÖNETİMİ
ANABİLİM DALI

BAĞLANTISALLIK PROBLEMİNİN CEZALI REGRESYON YÖNTEMLERİ İLE GİDERİLMESİ

Yüksek Lisans Tezi

Hazırlayan
Emel CİĞER
ORCID No: 0009-0005-3947-2413

Danışman
Doç. Dr. Evrim Ersin KANGAL
ORCID No: 0000- 0001-5906-3143

ARALIK-2023, MERSİN

ONAY

Emel Ciğer tarafından Doç. Dr. Evrim Ersin KANGAL danışmanlığında hazırlanan "Bağlantısallık Probleminin Cezalı Regresyon Yöntemleri ile Giderilmesi" başlıklı çalışma aşağıda imzaları bulunan jüri üyeleri tarafından,
☒ Oy birliği ile
☐ Oy çokluğu ile
Yüksek Lisans Tezi olarak kabul edilmiştir.

Görevi	**Unvanı, Adı ve SOYADI**	**İmza**
Başkan	Doç.Dr.Evrim Ersin KANGAL	……...
Üye	Doç.Dr.Ali Kemal HAVARE	……...
Üye	Dr.Öğr.Üyesi Murat KURTLAR	……...

Yukarıdaki Jüri kararı T.C. Mersin Üniversitesi Sosyal Bilimler Enstitüsü Yönetim Kurulu'nun ……..tarih ve …….sayılı kararıyla onaylanmıştır.

Prof. Dr. Yusuf Gürhan TOPÇU
Sosyal Bilimler Enstitü Müdürü

ETİK BEYAN

Mersin Üniversitesi Lisansüstü Eğitim-Öğretim Yönetmeliğinde belirtilen kurallara uygun olarak hazırladığım bu tez çalışmasında,

- Tez içindeki bütün bilgi ve belgeleri akademik kurallar çerçevesinde elde ettiğimi,
- Görsel, işitsel ve yazılı tüm bilgi ve sonuçları bilimsel ahlâk kurallarına uygun olarak sunduğumu,
- Başkalarının eserlerinden yararlanılması durumunda ilgili eserlere bilimsel normlara uygun olarak atıfta bulunduğumu,
- Atıfta bulunduğum eserlerin tümünü kaynak olarak kullandığımı,
- Kullanılan verilerde herhangi bir tahrifat yapmadığımı,
- Bu tezin herhangi bir bölümünü Mersin Üniversitesi veya başka bir üniversitede başka bir tez çalışması olarak sunmadığımı,
- Tezin tüm telif haklarını Mersin Üniversitesi'ne devrettiğimi

beyan ederim.

ETHICAL DECLARATION

This thesis is prepared in accordance with the rules specified in Mersin University Graduate Education Regulation and I declare to comply with the following conditions:

- I have obtained all the information and the documents of the thesis in accordance with the academic rules.
- I presented all the visual, auditory and written informations and results in accordance with scientific ethics.
- I refer in accordance with the norms of scientific works about the case of exploitation of others' works.
- I used all of the referred works as the references.
- I did not do any tampering in the used data.
- I did not present any part of this thesis as an another thesis at Mersin University or another university.
- I transfer all copyrights of this thesis to Mersin University.

20 Aralık 2023/ 20 December 2023

İmza / Signature

Emel CİĞER

ÖZET

BAĞLANTISALLIK PROBLEMİNİN CEZALI REGRESYON YÖNTEMLERİ İLE GİDERİLMESİ

Gelişen teknoloji ile birlikte yapay zeka uygulamalarına olan ilgi artmış ve bu uygulamalar kurumların, akademik çalışmaların ilgi odağı olmuştur. Makine öğrenmesinde karar ağaçları ve yapay sinir ağları(artificial neural network) algoritmaları sıkça kullanılan yöntemler olsa da araştırma yapılan çalışmanın amacı veya kullanılan veri setlerine uygunluklarından dolayı regresyon modelleri de hala en çok kullanılan yöntemlerdendir. Ancak bazı regresyon modellerinde "Çoklu Doğrusal Bağlantı Problemi" olarak adlandırılan, bağımsız değişkenlerden iki veya daha fazlası arasında doğrusal ya da doğrusala yakın ilişki olması durumu ortaya çıkabilmektedir. Çoklu doğrusal bağlantı problemi(multicollinearity) ile karşılaşılan durumlarda Lasso Regresyon'u ve Ridge Regresyon'u gibi alternatif yöntemler ele alınabilir. Bu tezde Kaggle veri bankasında açık kaynak olarak sunulan öğrencilerin not performanslarının olduğu 1000 kayıttan oluşan bir veri seti kullanılmıştır. Veri setine, Python 3.8.5 yazılım dili kullanılarak sırasıyla Lineer Regresyon, Lasso Regresyon ve Ridge Regresyon makine öğrenmesi modelleri uygulanmıştır. Sonuç olarak, bu çalışmada cezalı regresyon yöntemlerinin denetimli makine öğrenmesine etkisi bir örnek üzerinde denenmiş ve sonuçları tartışılmıştır. Sistem üzerinde ayrı ayrı uygulanan modellerin performans değerleri; Lineer Regresyonda "0,839", Lasso Regresyonda "0,843" ve Ridge Regresyonda "0,846" olarak gerçekleşmiştir.

Anahtar Kelimeler: Lasso, Ridge, Lineer Regresyon, Makine Öğrenmesi.

Danışman: Doç. Dr. Evrim Ersin KANGAL, İşletme Bilgi Yönetimi Anabilim Dalı, Mersin Üniversitesi, Mersin.

ABSTRACT

ELIMINATING THE CONNECTIVITY PROBLEM WITH PENALIZED REGRESSION METHODS

With the developing technology, interest in artificial intelligence applications has increased and has become the center of attention of institutions and academic studies. Although decision trees and artificial neural network algorithms are frequently used methods in machine learning, regression models are still among the most commonly used methods due to their suitability for the purpose of the study or the data sets used. However, in some regression models, there may be a linear or near-linear relationship between two or more of the independent variables, which is called the "Multicollinearity Problem". In cases where multicollinearity is encountered, alternative methods such as Lasso Regression and Ridge Regression can be considered. This thesis uses a dataset of 1000 records of students' grade performance, which is available as open source in the Kaggle database. Linear Regression, Lasso Regression and Ridge Regression machine learning models are applied to the dataset using Python 3.8.5 software language. As a result, in this study, the effect of penalized regression methods on supervised machine learning is tested on an example and the results are discussed. The performance values of the models applied separately on the system were realized as “0.839” in Linear Regression, “0.843” in Lasso Regression and “0.846” in Ridge Regression.

Keywords: Lasso, Ridge, Linear Regression, Machine Learning

Advisor: Assoc. Prof. Evrim Ersin KANGAL, Department of Business Information Management, Mersin University, Mersin.

TEŞEKKÜR

Tez çalışmamın hazırlanması süresince son dakikaya kadar desteğini esirgemeyen danışman hocalarım Sayın Doç. Dr. Evrim Ersin KANGAL'a ve yüksek lisans programında ders aldığım hocalarıma sonsuz teşekkürlerimi sunarım.

İÇİNDEKİLER

TABLOLAR DİZİNİ

ŞEKİLLER DİZİNİ

KISALTMALAR ve SİMGELER

Kısaltma/Simge	Tanım
EKK	En Küçük Kareler
LASSO	Least Absolute Shrinkage and Selection Operator
SVD	Tekil Değer Ayrışımı (Singular Value Decomposition)
PCA	Principal Component Analysis (Temel Bileşenler Analizi)
MSE	Mean Squared Error (Ortalama Karesel Hata)
RMSE	Root Mean Squared Error (Kök Ortalama Kare Hatası)
R2	Düzeltilmiş Kare (doğruluk performansı)
SVM	Support Vector Machines
FP-Growth	Frequent Pattern Growth
FP	Frequent Pattern
HMM	Hidden Markov Model
SOM	Self Organizing Maps

GİRİŞ

Regresyon analizi istatistiksel bir öğrenme metodu olup, bağımsız değişken ile bağımlı değişken arasındaki ilişkileri anlamak ve bir değişkenin diğerine bağlılığını ölçmek bu yöntemin odak noktasını oluşturmaktadır. Algoritmanın kazandığı deneyim algoritmaya hedef değişkenin gelecekteki değerini tahmin etmesine olanak sunacaktır. Modelin kazandığı deneyim ya bir doğru ya da bir eğri ile temsil edilmektedir. Örneğin; basit lineer regresyonda bağımlı değişken ile bağımsız değişken arasındaki ilişki bir doğru denklemi ile ifade edilirken, çoklu regresyon durumunda bağımsız değişkenlerin sayısı 2'den fazla olacağından doğrudan basit doğru denklemi ile ifade edilmesi imkânsızdır. Bu noktada model geçmiş verilerden kazanılan deneyim doğrultusunda bağımsız değişkenlerin katsayılarının alacağı değeri belirlemektedir. Elde edilen bu katsayılar algoritmanın performansını değerlendirmekten sorumlu olan R^2 hakkında bilgi vermektedir. R^2 değeri, 0 ile 1 arasında yer almaktadır. Sıfır durumu başarısızlığı bir ise mükemmel başarıyı temsil etmektedir. Ekonometri, finans, biyoloji, mühendislik ve sosyal bilimler gibi birçok alanda regresyon yaklaşımı ile ilgili birçok çalışma görmek mümkündür.

Çoklu doğrusallık probleminin beraberinde getirdiği bağlantısallık sorunu ilk olarak Frish tarafından gün yüzüne çıkarılmıştır[1]. Sonrasında, Hoerl ve Kennard bu sorunun üstesinden gelmek adına Ridge Regresyon Yaklaşımı olarak bilinen çözüm perspektifi önermiştir[2]. Bu modelleme temel olarak veri kümesindeki hangi özniteliğin karar mekanizmasında etkin rol üstlendiğini belirlemek esasına dayanmaktadır ve sonuç itibari ile karar mekanizması model içerisinde yer alan ayar parametresi yardımı ile belirlenmektedir. Öte yandan, gelişen teknoloji ile birlikte yapay zeka uygulamalarına olan dikkat önemli ölçüde artmış ve böylece kurumlar ve akademik çalışmaların ilgi odağı haline gelmiştir. Her ne kadar Makine Öğrenmesi bağlamında Karar Ağaçları ve Yapay Sinir ağları algoritmaları sıkça kullanılan yöntemler olarak dikkat çekse de üzerinde çalışılan konunun amacı veya kullanılan veri setlerinin karakteristik özellikleri bazı durumlarda regresyon modellerini de başarılı çıkarımlara ulaştırın yaklaşımlar ailesine katmaktadır. Ancak bazı regresyon modellerinde "Çoklu Doğrusal Bağlantı Problemi" olarak adlandırılan, bağımsız değişkenlerden iki veya daha fazlası arasında doğrusal ya da doğrusala yakın ilişki olması durumu ortaya çıkabilmektedir. Çoklu doğrusal bağlantı problemi (ya da terminolojide bilinen adıyla multi-collinearity) ile karşılaşılan durumlarda Lasso Regresyon Yaklaşımı ve Ridge Regresyon Analizi gibi alternatif yöntemler kullanılabilmektedir.

Bu tez çalışmasının araştırma konusu, temel olarak Kaggle veri bankasında açık kaynak olarak sunulan öğrenci performans notlarından oluşan bir veri setinin Lasso ve Ridge Regresyon modelleri aracılığı ile analiz edilmesine dayanmaktadır. Bu tez çalışması beş bölüme ayrılarak hazırlanmıştır. İlk bölümde konu kapsamına giren kavramlar ile ilgili bazı temel bilgilere yer verilmiştir. İkinci bölümde

[1] Friedman, J. H. ve Tukey, J.W. *A projection pursuit algorithm for exploratory data analysis*. IEEE Transactions on computers, 100(9), 881-890. 1974.
[2] Hoerl, A.E. Kennard, R.W. Baldwin, K.F. Ridge regression: some simulations. *Communications in Statistics 4*, 105–123. 1975.

ise bu araştırmaya ışık tutan ve literatürde dikkat çeken önemli bazı araştırmalar hakkında kısa bilgiler sunulmuştur. Üçüncü bölümde tez araştırmasında kullanılan kuramsal materyaller ile uygulanacak analiz yaklaşımının ana hatlarına yer verilmiştir. Dördüncü bölümde ise çalışmanın önemli bulguları paylaşılarak elde edilen sonuçların dikkat çeken noktalarını ön plana çıkaran analizler yapılmıştır. Son bölüm ise tartışmalara ayrılmıştır.

1. KURAMSAL TEMELLER

1.1. Çoklu Regresyon

Doğrusal regresyon, bağımlı ve bağımsız değişkenler arasındaki ilişkiyi tanımlamak amacıyla kullanılan bir doğrusal model yaklaşımıdır. Verilerin çoklu regresyon açısından değerlendirilebilmesi için özelliklerinin birbirleri arasında bir bağlantısızlığın olması esastır. Matematiksel olarak

$$\sum_{i=0}^{n} \beta_i X_i = 0 \tag{1}$$

ifade edilir. Burada n örneklem sayısını, β_i model parametresi ve X_i bağımlı değişkenleri tanımlamaktadır. Bu eşitlik, model parametrelerinin hepsinin sıfıra eşit olması durumda sağlanmaktadır. Bu durumda tüm bağımlı değişkenler arasında bir bağlantısallık probleminin önüne geçilmiş olacaktır. Böylece bu sistem i. örneklem için bağımlı ile bağımsız değişken arasındaki matematiksel modeli, aşağıdaki gibi yazmak mümkün olacaktır:

$$y_i = \beta_0 + \beta_1 X_{1i} + \cdots + \beta_n X_{ni} + e_i \tag{2}$$

burada e_i hata payını temsil etmektedir. Bu eşitliği tüm örneklem için uyguladığımızda

$$\begin{aligned} y_0 &= \beta_0 + \beta_1 X_{10} + \cdots + \beta_n X_{n0} + e_0 \\ y_1 &= \beta_0 + \beta_1 X_{11} + \cdots + \beta_n X_{n1} + e_1 \\ &\vdots \\ y_n &= \beta_0 + \beta_1 X_{1n} + \cdots + \beta_n X_{nn} + e_n \end{aligned} \tag{3}$$

elde edilir. Eşitliği bir matris formatına yazacak olursak:

$$\begin{bmatrix} y_0 \\ y_1 \\ \vdots \\ y_n \end{bmatrix} = \begin{bmatrix} 1 & X_{10} & \dots & X_{n0} \\ 1 & X_{11} & \dots & X_{n1} \\ \vdots & \vdots & & \vdots \\ 1 & X_{1n} & \dots & X_{nn} \end{bmatrix} \begin{bmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_n \end{bmatrix} + \begin{bmatrix} e_0 \\ e_1 \\ \vdots \\ e_n \end{bmatrix} \tag{4}$$

Bu durumda denklem 2'de yer alan eşitlik kapalı matris formunda gösterilecek olursa;

$$y_i = X_i \beta_i + e_i \tag{5}$$

olarak yazılır. Hata değeri tek başına bırakılır ve toplam hata hesaplanırsa[3]

W.,1974);

[3] Friedman, J. H. ve Tukey, J.W. *A projection pursuit algorithm for exploratory data analysis.* 1974.

$$f(\beta) = \sum_i^n (y_i - X_i\beta_i)(y_i - X_i\beta_i)^T = \sum_i^n (y_i - X_i\beta_i)\left(y_i^T - \beta_i^T X_i^T\right) \quad (6)$$

olarak elde edilir. Bir fonksiyonun yerel minimum noktasını hesaplamak için ilgili fonksiyonun parametresine göre türevini sıfır yapan noktayı hesaplamak gerekmektedir. Bu durumda 6 numaralı denklemin model parametresine göre türevi hesaplandığında;

$$\frac{df(b)}{db} = -\sum_i^n X_i\left(y_i^T - b_i^T X_i^T\right) = 0$$
$$\sum_i^n \left(X_i y_i^T - X_i b_i^T X_i^T\right) = 0 \quad (7)$$

Denklem 7'nin sıfır olması için toplam sembolün içerisinin sıfır olması gerekmektedir. Bu durumda;

$$X_i y_i^T - X_i b_i^T X_i^T = 0$$
$$X_i^T \rightarrow X_i y_i^T = X_i b_i^T X_i^T \leftarrow X_i$$
$$X_i^T X_i y_i^T X_i = X_i^T X_i b_i^T X_i^T X_i$$
$$\left(X_i^T X_i\right)^{-1} \rightarrow X_i^T X_i y_i^T X_i = X_i^T X_i b_i^T X_i^T X_i$$
$$y_i^T X_i = b_i^T X_i^T X_i \leftarrow \left(X_i^T X_i\right)^{-1}$$
$$y_i^T X_i \left(X_i^T X_i\right)^{-1} = b_i^T \quad (8)$$

Eşitliğin her iki tarafının eşleniği alındığında aşağıdaki model parametresinin sonucu elde edilir.

$$b_{EKK} = \left(X_i^T X_i\right)^{-1}\left(X_i^T y\right) \quad (10)$$

Bu sonuç aslında en küçük kareler tahmininin sonucu olarak değerlendirilir. Uygulamada esas alınan nokta bağımlı değişkenler arasında herhangi bir bağlantısızlığın var olmasıdır. Fakat doğada böyle mükemmel sistemlerin varlığından bahsetmek mümkün değildir. Bu sorunun üstesinden gelmek için Ridge ve Lasso gibi bir çok farklı yaklaşımlar literatürde yer almaktadır.

1.2. Ridge Yaklaşımı

Bağlantısallık problemi pek çok araştırmacının dikkatini çekmiştir. İlk olarak Hoerl ve Kennard tarafından önerilen Ridge regresyon çözüm yöntemi en başarılı yaklaşım olarak ön plana çıkmaktadır[4].

[4] Hoerl, A.E. Kennard, R.W. ve Baldwin, K.F. Ridge regression: some simulations. *Communications in Statistics 4*, 105–123. 1975.

Daha sonra, McDonald ve Galarneau[5] Ridge parametresini belirlemek için iki yöntem kullanmış ve sonuçlarını en küçük kareler yaklaşımı bağlamında irdelemiştir. Benzer bir bakış açısıyla, Lawless ve Wang[6] Ridge tahmin edicisinin performans testini en küçük kareler ve iki hata kareler ortalamasına göre değerlendirmiştir. Ridge regresyon için tanımlanan amaç fonksiyonu[7].

$$f_R(\beta, \beta^T) = \sum_i^n (y_i - X_i\beta_i)\left(y_i^T - \beta_i^T X_i^T\right) + \alpha\beta_i\beta_i^T \tag{11}$$

burada α denge parametresi olarak tanımlanmaktadır. Çoklu regresyondaki sürecin aynısını çalıştırmak için denklem 11'i, β'a göre değişimini minimum yapan nokta, aşağıdaki formüllerde gösterildiği gibi I birim matris olarak tanımlanır.

$$\frac{\partial f_R(\beta)}{\partial \beta^T} = -\sum_i^n X_i^T(y_i - X_i\beta_i) + \alpha\beta_i = 0$$

$$\sum_i^n -X_i^T y_i + X_i^T X_i\beta_i + \alpha\beta_i = 0$$

$$\left(X_i^T X_i + \alpha I\right)\beta_i = X_i^T y_i$$

$$\beta_r = (X^T X + \alpha I)^{-1} X^T y \tag{12}$$

Eğer denge parametresini sıfır olarak alırsak 12 nolu denklemin, 10 nolu çoklu regresyon denklemine indirgeneceği açık olarak görülmektedir. Dolayısıyla denge parametresi bağımlı değişkenler arasında bir baskı görevi üstlenerek kararı etkileyecek ve böylece bağımsız değişkenin katkısını sıfıra çekerek boyut indirgemesine neden olacaktır. Bu süreç veri biliminde, boyut indirgemesi olarak değerlendirilir. Fakat Ridge yaklaşımı katsayıları sıfıra doğru indirgenmeye zorlamasına rağmen, hiçbir katsayıyı sıfıra indirgeyemeyeceği durumların varlığı söz konusudur. Bunun temel nedeni, l_2 ceza teriminin tüm ağılıkları aynı derecede 0'a doğru küçültmesinden kaynaklanmaktadır.

1.3. Lasso Yaklaşımı

LASSO regresyon yaklaşımı, çoklu bağlantı sorunlarına çözüm getirmek amacıyla 1996 yılında Tibshirani tarafından önerilmiş olup Ridge regresyon yaklaşımıyla benzer bir mantığa sahiptir. LASSO regresyonunda da regresyon katsayılarına bir ceza uygulanarak, bu katsayıların sıfıra yaklaştırılması amaçlanmaktadır. Lasso modeli için bağımlı değişken ile bağımsız değişkenler arasındaki hatayı minimize eden matematiksel amaç fonksiyon aşağıdaki biçimde tanımlanır(Tibshirani, 1996):

$$f_L\left(b_l, b_l^T\right) = (y - Xb_l)\left(y^T - b_l^T X^T\right) + \alpha|b_l| \tag{13}$$

[5] McDonald, G.C. ve Galarneau, D.I. A Monte Carlo evaluation ofsome ridge-type estimators. *Journal of the American Statistical Association 70 (*350), 407–412. 1975.
[6] Wiley, J. *Applied linear regression.* United States of America. 83. 2005.
[7] Hoerl, A.E. Kennard, R.W. ve Baldwin, K.F. *Ridge regression: some simulations.* 1975.

burada α sıkışma miktarını belirleyen ayar parametresi veya ceza terimi olup, son terim ceza değişkeni olarak tanımlanmaktadır. Temel amaç bu hata fonksiyonunu minimum yapan b_l katsayının karşılık geldiği değeri hesaplamaktır. Bunu yapmak için denklem 13'de yer alan eşitliğin b_l'e göre türevini sıfır yapan noktayı bulmak gerekmektedir.

$$\frac{\partial f}{\partial b_l} = -X\left(y^T - b_l^T X^T\right) + \alpha sgn(b_l)=0 \tag{14}$$

Eşitliğin her iki tarafının eşleniği alıp, sağdan X^T ve soldan X ile çarpılır akabinde bir takım matematiksel işlemlere tabi tutulduğunda

$$b_l = b_{EKK} - \alpha sgn(b_l)(X^T X)^{-1} \tag{15}$$

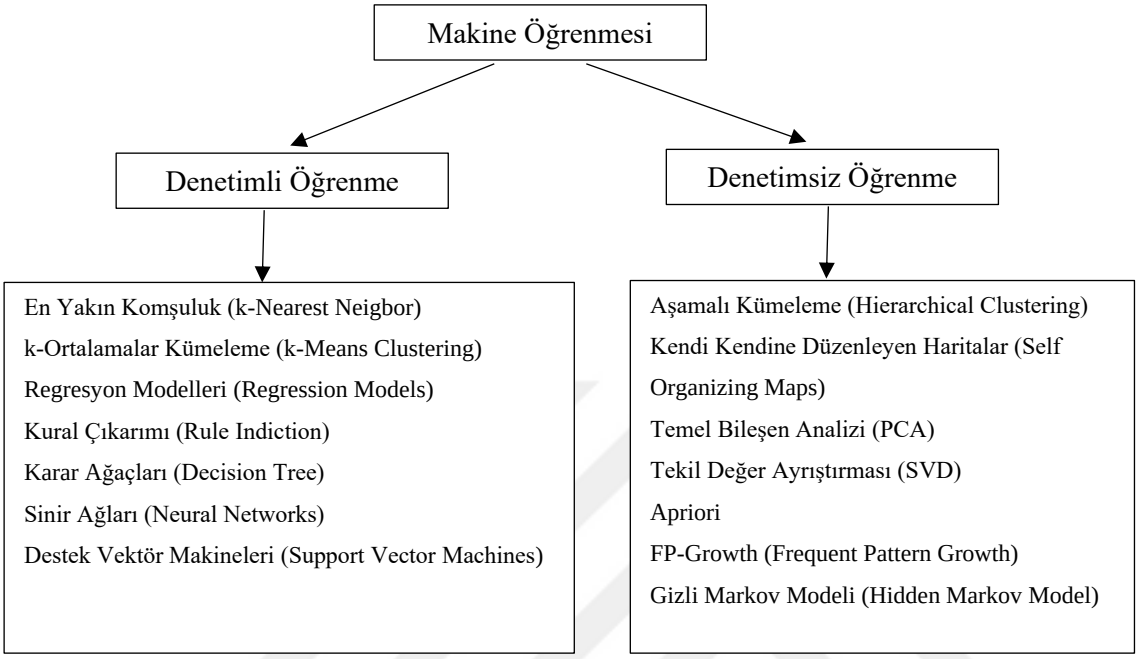
b_{EKK} EKK tahmin edicisi olarak tanımlanmaktadır. Eğer burada α katsayısı sıfır olarak alınırsa Lasso yaklaşımı EKK yaklaşımına indirgenir. Ridge yaklaşımına benzer şekilde baskı parametresini sıfıra yaklaştırdığımızda Lasso modelinin EKK modeline indirgendiği açık olarak görülmektedir. Lasso tahmin edicisine bakıldığında katsayıların karelerini almak yerine mutlak değer olarak alındığı için bazı özellikler baskılanmaya tabi tutularak karar mekanizmasından çıkarılmasına yardımcı olabilmektedir. Bu açıdan bakıldığında öz nitelik seçimi için Lasso yaklaşımının son derece önemli rol oynadığı görülmektedir.

1.4. Makine Öğrenme Yaklaşımı ve Temelleri

Gelişen dünyada veri hacminin hızla artması beraberinde verilerin işlenmesinde insanların karar vermede tek başlarına yetersiz kalmasına neden olmaktadır. Yapay zekâ algoritmaları ile donatılmış bilgisayar yazılımları ile bu sorunun üstesinden gelinmeye çalışılmaktadır. Şekil 1.1'de, yapay zekanın kapsamı gösterilmektedir.

Şekil 1.1. Yapay zeka

Şekil 1.1.'den anlaşılacağı üzere makine öğrenmesi yapay zekânın bir alt dalı olup büyük miktarda veri içerisinde yer alan gizli örüntüleri tanıyabilen algoritmalar geliştirmek üzerine odaklanır. Algoritmalar denetimli ve denetimsiz olmak üzere iki faklı fay hattı üzerinde ilerlemektedir.



Şekil 1.2 Makine Öğrenmesi Yöntemleri

Denetimli öğrenmede, algoritma etiketli veri setleri üzerinde eğitilir ve belirli bir çıktıyı tahmin etmeyi amaçlarken denetimsiz öğrenme ise etiketsiz veri setlerini inceleyerek veri içindeki yapıları keşfetmeye çalışır. Makine öğrenmesi günümüzde birçok endüstride yaygın olarak kullanılmaktadır. Finans, sağlık, perakende ve ulaşım gibi birçok sektör, veri analizi ve tahmin yetenekleri sayesinde makine öğrenmesinden faydalanır. Örneğin, finans sektöründe, kredi riski değerlendirmesi ve hisse senedi tahminleri gibi konularda makine öğrenmesi kullanılarak daha doğru sonuçlar elde edilebilir. Ayrıca makine öğrenmesindeki gelişmeler, otonom araçlar, sesli asistanlar, yüz tanıma sistemleri ve öneri sistemleri gibi birçok uygulama alanında çığır açmıştır. Gelecekte, makine öğrenmesinin daha da gelişmesiyle birlikte, daha akıllı ve daha etkili sistemlerin ortaya çıkması beklenmektedir.

1.4.1. Denetimli Yöntemler

Denetimli öğrenme yöntemlerinde, eğitim için kullanılacak olan veri bir "etiket" bilgisi içermektedir. Yani sonuçları belli olan veriler ile sistem eğitilerek bir model oluşturulur. Bu sayede sisteme etiket bilgisi belli olmayan yeni veriler girildiğinde, sistem bunu öğrendiği model ile etiket bilgisini tahmin eder.

1.4.1.1. En Yakın Komşuluk (k-Nearest Neigbor)

En yakın komşu karar kuralı, henüz sınıflandırılmamış bir noktaya, önceden sınıflandırılmış bir dizi noktanın uzaklık olarak en yakınının sınıflandırılmasını atayarak çalışır[8]. Hem sınıflama hem de regresyon ayağında kullanılabilen çok yönlü bir algoritmadır. k-En Yakın Komşu algoritmasında en önemli hususlardan biri önceden belirlenen optimal k sınıf değerini bulmaktır.

1.4.1.2. k-Ortalamalar Kümeleme (k-Means Clustering)

Veri kümeleme için yaygın olarak kullanılan ve bilinen bir algoritmadır. Temel amacı; verileri nitelik veya özelliklerine göre k adet sınıfa ayırmaktır ve bu sınıflandırmayı, verilerin en yakın küme merkezleri etrafına yerleştirilmesi ile gerçekleştirir[9]. K-means algoritması, özellikle veri analizi ve desen tanıma gibi alanlarda kullanılarak verilerin anlamlı gruplara ayrılmasına olanak tanır.

1.4.1.3. Regresyon Modelleri (Regression Models)

Regresyon modelleri, bir bağımlı değişkenin diğer bağımsız değişkenler tarafından nasıl etkilendiğini modellemek için kullanılır. Regresyon, bir bağımlı değişken ile ilişkilendirilen bağımsız değişkenlerin bir fonksiyonu olarak ifade edilir ve bu ilişkinin fonksiyonel şekli regresyon modelleri ile incelenir. Kullanılan regresyon modeli, verinin yapısına bağlı olarak değişir ve yanlış modelin seçilmesi, hatalı sonuçlara yol açabilir[10]. En çok kullanılan doğrusal regresyon ve lojistik regresyon olmak üzere temel bileşenler regresyonu, ridge regresyon, lasso regresyon gibi modelleri ve uygulamaları vardır.

1.4.1.4. Kural Çıkarımı (Rule Indiction)

Kurallara dayalı çıkarım, veri setlerindeki desenleri belirleyen ve bu desenlere dayanarak kararlar alan bir makine öğrenmesi yaklaşımıdır. Bu yöntem, genellikle "Eğer-Şart" yapılarına sahip kurallar seti kullanarak sonuçlar elde eder. Kural çıkarımı genellikle tıp, finans, ve pazarlama gibi alanlarda kullanılır. Örneğin, bir sağlık sistemine uygulanan kural tabanlı bir model, hastalıkların teşhisi veya tedavi planlaması konularında yardımcı olabilir.

1.4.1.5. Karar Ağaçları (Decision Tree)

Karar ağacı öğrenimi, ayrık değerli hedef fonksiyonların yaklaştırılması için kullanılan bir yöntemdir. Bu yöntemde, karar ağaçları bir dizi test gerçekleştirir ve en iyi diziye ulaşarak hedefi tahmin etmeye çalışır. Her test, dallara ayrılan ve test bir yaprak düğümde sonlanana kadar devam eden bir yapı

[8] Cover, T. ve Hart, P. Nearest neighbor pattern classification. *IEEE Transactions on Information Theory, 13*(1), 21-27. 1967.
[9] Dinçer, Ş.E. *Veri madenciliğinde K-means algoritması ve tıp alanında uygulanması.* Yayımlanmamış Yüksek Lisans Tezi, Kocaeli Üniversitesi. 2006.
[10] Arzu, A.R.I. ve Önder, H. *Farklı veri yapılarında kullanılabilecek regresyon yöntemleri. Anadolu Tarım Bilimleri Dergisi, 28*(3), 168-174. 2013.

oluşturur. Bu ağaç yapısındaki yol, kökten hedef yaprağa giden kuralı temsil eder ve kurallar genellikle eğer-o zaman formatında ifade edilir[11].

Karar ağaçları geniş bir uygulama yelpazesine sahiptir. Örneğin, müşteri segmentasyonu, risk değerlendirmesi ve pazar analizi gibi konularda kullanılabilir.

1.4.1.6. Sinir Ağları (Neural Networks)

Sinir ağları algoritmasındaki temel amaç, belirli bir görev için uygun bir iç yapı geliştirmeye olanak sağlayacak güçlü bir sinaptik modifikasyon kuralı bulmaktır. Bu görevde, giriş birimlerinin durum vektörleri için çıkış birimlerinin istenen durum vektörleriyle eşleştirilmesi belirtilir. Giriş birimleri doğrudan çıkış birimlerine bağlı olduğunda, bağlantıların göreceli güçlerini aşamalı olarak ayarlayan öğrenme kuralları ile gerçek ve istenen çıkış vektörleri arasındaki farkı azaltma çabası bulunmaktadır[12]. Bu, sinir ağındaki bağlantıların uygun şekilde güçlendirilmesi veya zayıflatılması yoluyla, ağın istenilen görevi daha etkili bir şekilde gerçekleştirmesine olanak tanıyabilir. Bu tür öğrenme kuralları, sinir ağlarının adaptasyon yeteneğini ve performansını artırmak için kullanılabilecek temel araçlardan biri olarak öne çıkıyor. Sinir ağları, görüntü tanıma, ses işleme, doğal dil işleme ve oyun stratejileri gibi birçok alanda kullanılır. Örneğin, bir görüntü tanıma sistemine uygulanan sinir ağları, nesne tespiti veya yüz tanıma gibi görevlerde başarılı olabilir.

1.4.1.7. Destek Vektör Makineleri (Support Vector Machines veya SVM)

Destek Vektör Makineleri, özellikle yüksek boyutlu sınıflandırma ve regresyon görevlerinde kullanılan bir makine öğrenimi algoritmasıdır. Veri noktalarını iki sınıfa ayırmak ve bu ayrımı en iyi şekilde gerçekleştiren bir hiperdüzlemi bulmak üzerine odaklanır. Hiper düzlemi en iyi destekleyen noktalara destek vektörleri denir. Giriş verileri, doğrusal olmayan bir şekilde özellik uzayına eşlenir. Bu uzayda, veriler arasında bir karar yüzeyi oluşturularak sınıflandırma yapılır. SVM'in özelliği, geniş bir genelleme yeteneği sunmasıdır. Yani, öğrenme sürecindeki hataların, daha önce görülmemiş verilere karşı etkili bir sınıflandırma yapabilme becerisiyle başa çıkabilmesidir. Bu, SVM'in eğitim verilerindeki karmaşıklıkları anlama ve yeni verileri etkili bir şekilde sınıflandırma yeteneğiyle ilgilidir[13]. SVM, özellikle görüntü sınıflandırma, metin analizi, biyoinformatik ve finansal tahmin gibi birçok alanda başarıyla kullanılır.

---

[11] Bounsaythip, C. ve Rinta-Runsala, E. Overview of data mining for customer behavior modeling. *VTT Information Technology Research Report, Version, 1,* 1-53. 2001.

[12] Rumelhart, D.E. Hinton, G. E. ve Williams, R.J. Learning representations by back-propagating errors. *Nature, 323*(6088), 533–536. 1986.

[13] Cortes, C. ve Vapnik, V. Support-vector networks. *Machine learning, 20*(3), 273-297. 1995.

1.4.2. Denetimsiz Öğrenme

Denetimsiz öğrenme yönteminlerinde verilerin etiketi yoktur. Oluşturulan sistem tarafından, veri setindeki benzerlikler veya ilişkiler bulunarak, ortak özelliklere göre grupların oluşturulması amaçlanmaktadır.

1.4.2.1. Aşamalı Kümeleme (Hierarchical Clustering)

Aşamalı Kümeleme, veri noktalarını hiyerarşik bir yapıda birleştiren bir kümeleme yöntemidir. Veri noktaları benzerlikleri temel alarak gruplara ayrılır ve bu gruplar alt gruplara bölünerek bir ağaç yapısı oluşturulur. Her kümeleme bir önceki kümelemeden gelen kümelerin birleştirilmesi ve alfa artışı, hiyerarşik bir kümeleme şeması üretmektedir, Dolayısıyla bir yanda Hiyerarşik Kümeleme Şemaları, diğer yanda ultrametrik eşitsizliği karşılayan metrikler arasında tam bir uygunluk vardır[14]. Aglomeratif (birleştirici) ve ayrıştırıcı (bölücü) yöntemler olarak iki temel türü vardır. Bu yöntemler, veri setindeki benzerlik veya uzaklıklara dayanarak kümeleme işlemini gerçekleştirirler. Genetik analiz, biyomedikal görüntüleme ve sosyal ağ analizi gibi birçok farklı alanda kullanılır. Aglomeratif yöntemler genellikle büyük veri setlerinde etkilidir ve veri setinin yapısını daha iyi anlamak için kullanışlıdır.

1.4.2.2. Kendi Kendine Düzenleyen Haritalar (Self Organizing Maps)

SOM; yüksek boyutlu bir dağılımın düzenli bir şekilde düşük boyutlu bir sisteme eşlenmesi işlemidir. Bu işlem, yüksek boyutlu veri öğeleri arasındaki karmaşık ve doğrusal olmayan istatistiksel ilişkileri, düşük boyutlu bir ekranda basit geometrik ilişkilere dönüştürmeyi amaçlar (Kohonen, T., 1998). Bu yöntem, verinin karmaşıklığını azaltarak düşük boyutlu bir temsil oluşturur. Veri madenciliği, görüntü işleme ve metin madenciliği gibi birçok uygulama alanında kullanılır. Özellikle büyük ve karmaşık veri setlerinde desenlerin keşfedilmesi için etkilidir.

1.4.2.3. Temel Bileşen Analizi (PCA)

Bu yöntem çok boyutlu veri setlerini daha az boyutlu bir alt uzaya dönüştürerek veri setindeki değişkenliği maksimize etmeyi amaçlayan bir boyut azaltma tekniğidir. Temel Bileşen Analizi'nin (PCA) esasen titiz dağılımsal veya model varsayımlarına bağlı olmadan çalışabililen, esnek ve geniş bir veri yelpazesinde kullanılabilen tanımlayıcı bir araçtır[15]. PCA, özellikle büyük boyutlu veri setlerinde, özellikler arasındaki ilişkileri anlamak, veri setini görselleştirmek veya makine öğrenimi modelleri için giriş boyutunu azaltmak amacıyla yaygın olarak kullanılır. PCA, özellikle görüntü işleme, biyoinformatik, finansal analiz ve sinyal işleme gibi alanlarda kullanılır.

[14] Johnson, S.C. Hierarchical clustering schemes. *Psychometrika, 32(*3), 241-254. 1967.
[15] Jolliffe, I. Principal component analysis. Principal Component *Analysis for Special Types of Data*,338-339. 2002.

1.4.2.4. Tekil Değer Ayrıştırması (SVD)

Tekil Değer Ayrıştırması (SVD), matrislerin üç temel bileşenine (sol singular vektörler, tekil değerler ve sağ singular vektörler) ayrıştırılması ve temsil edilmesi için kullanılan matematiksel bir yöntemdir. Her bir bileşen, matrisin farklı özelliklerini temsil eder. SVD ile en fazla varyasyonun olduğu yerler belirlendikten sonra, elde edilen temel bileşenler kullanılarak orijinal veri noktalarının daha az boyut kullanarak en iyi yaklaşımını bulmak mümkündür[16]. Denetimsiz öğrenme kapsamında değerlendirilir çünkü genellikle veri setlerindeki yapıları anlamak ve veriyi sıkıştırmak için kullanılır.

1.4.2.5. Apriori

Apriori, veri madenciliği ve özellikle birliktelik kuralları analizi için kullanılan bir algoritmadır. Bu algoritma, bir veri setindeki öğeler arasındaki ilişkileri ve kuralları belirlemek amacıyla kullanılır. Bu algoritma iki aşamalıdır. İlk aşamada, belirli bir yüzdeye kadar olan işlemlerde sıkça bir araya gelen öğelerden oluşan, sık kullanılan öğe kümeleri tespit edilir. İkinci aşamada ise, ilk adımda bulunan sık kullanılan öğe kümelerinden ilişkilendirme kuralları oluşturulur[17]. Apriori, perakende sektöründe müşteri alışveriş alışkanlıklarını anlama, pazarlama stratejilerini geliştirme ve ürün yerleştirmesi gibi birçok alanda kullanılır. Ayrıca, büyük veri setlerinden anlamlı kuralların çıkarılmasında da etkilidir.

1.4.2.6. FP-Growth (Frequent Pattern Growth)

FP-Growth (Frequent Pattern Growth), birliktelik kuralları analizi ve veri madenciliği alanında sıklıkla kullanılan bir algoritmadır. Minimum desteğin düşük olduğu durumlarda aday kümelerin üretilmesi genellikle zaman alıcı ve tekrarlayan bir işlem olabilir. Ancak FP büyüme algoritması, aday kümeleri oluşturma ihtiyacını ortadan kaldırır. Bunun yerine, sık öğe kümesini sağlayan veritabanı, sık bir model ağacına (FP ağacına) sıkıştırılır ve FP ağacı kullanılarak madencilik işlemi gerçekleştirilir. Bu yaklaşım, madencilik sürecini daha hızlı ve verimli hale getirir, ayrıca minimum desteğin az olduğu durumlarda bile etkili sonuçlar sağlar[18]. Bu algoritma, özellikle büyük veri setlerinde sık görülen örüntüleri ve birliktelik kurallarını çıkartmak için etkilidir. FP-Growth, Apriori algoritmasının bazı dezavantajlarını aşarak daha hızlı bir performans sunar. Özellikle büyük veri setleri üzerinde çalışırken Apriori'ye göre daha etkili olabilir.

1.4.2.7. Gizli Markov Modeli (Hidden Markov Model veya HMM)

Gizli Markov modellemesinin istatistiksel yöntemlerinin son yıllarda daha popüler hale gelmesinin iki güçlü nedeni vardır. Birincisi, modeller matematiksel yapı bakımından çok zengin olduğundan çok çeşitli uygulamalarda kullanım için temel oluşturabilir. İkincisi ise modeller doğru

[16] Baker, K. Singular value decomposition tutorial. *The Ohio State University,* 24, 511. 2005.
[17] Borgelt, C. ve Kruse, R. Induction of association rules: Apriori implementation. *In Compstat: Proceedings in Computational Statistics* 395-400. 2002.
[18] Liu, Y. ve Guan, Y. Fp-growth algorithm for application in research of market basket analysis. *In 2008 IEEE International Conference on Computational Cybernetics.* 269-272. 2008.

şekilde uygulandığında, birçok önemli uygulama için pratikte çok iyi çalışmaktadır[19]. Bu yöntem özellikle zamanla değişen gizli durumları modellenmesi gereken sistemlerde kullanılan bir olasılık modelleme yöntemidir. HMM, bir dizi gözlemin ardındaki gizli durumları ve bu durumlar arasındaki geçiş olasılıklarını modelleyerek olayların olasılıklı bir sıralamasını tahmin etmeye yönelik bir çerçeve sunar. Gizli Markov Modeli, zaman içinde değişen sistemlerin modellenmesinde ve çeşitli uygulama alanlarında desenlerin ve ilişkilerin keşfedilmesinde güçlü bir araçtır.

1.5. Veri Tipleri ve Çevrimiçi Kaynaklar

Genel olarak makine öğrenimi modellerinde dört değişken tipinden bahsedilebilir. Bunlar; Sayısal değişkenler, kategorik değişkenler, bağımlı değişkenler ve bağımsız değişkenler. Sayısal değişkenler doğrudan sayılar ile ifade edilebilen değişkenleri kapsar. Fiyat, boy, yaş, miktar vb. şeklinde örneklendirilebilir. Kategorik değişkenler ise cinsiyet, eğitim durumu gibi değişkenler şeklinde ifade edilebilir. Bu değişken türlerinin biri hedef değişken olurken kalan değişkenler ise bağımsız değişken olmaktadır. Böylece bağımlı değişken ile bağımsız değişken arasındaki ilişki makine öğrenimi yöntemleri ile ilişkilendirilerek gerekli performans değerleri ölçülmektedir. Burada kritik öneme sahip olan nokta veri setinin hem orijinalliği hem de bilim çevreleri tarafından kabul görülmesidir. Bu noktada makine öğrenme için kullanılacak çok sayıda çevrim içi platform bulunmaktadır. Bunların en bilinenleri aşağıdaki tabloda sunulmuştur.

Tablo 1.1. En çok tercih edilen çevrim içi platformlar

Google’s Data Science Community	https://www.kaggle.com
Google’s Dataset Research	https://datasetsearch.research.google.com
Microsoft Research Open Data	https://www.msropendata.com
U.S. Government’s Open Data	https://data.gov
IBM’s AI and Data Science Community	https://community.ibm.com

[19] Rabiner, L.R. A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE, 77*(2), 257-286. 1989.

2. KAYNAK ARAŞTIRMALARI

Doğrusal regresyon analizindeki karmaşıklık çeşitli nedenlerden dolayı ortaya çıkabilir. Verinin özellikleri her zaman aynı değildir veya analizin amacı her zaman aynı olmayabilir. Tüm bu konulara bağlı olarak doğrusal regresyon yöntemlerine ilişkin istatistiksel araştırmalar on dokuzuncu yüzyılda başlamıştır ve halen oldukça aktiftir[20].

Basit doğrusal regresyon problemlerinde, R2'nin bir özet olarak uygunluğunu belirleyebilmenin yolu, tahmin edicinin karşısında yer alan özet grafiğini incelemektir[21]. Doğrusal Regresyonun tercih edilme sebepleri; doğrusal form nedeniyle model parametreleri kolaylıkla yorumlanabilir, doğrusal model teorileri matematiksel olarak iyi bir şekilde oluşturulmuştur ve en önemlisi birçok modern modelleme aracının yapı taşıdır. Özellikle örneklem boyutu küçük olduğunda temeldeki regresyon fonksiyonuna tatmin edici bir performans gösterir[22].

Regresyon analizi üç şeyi mümkün kılan bir istatistiksel değerlendirme türüdür: Bağımlı değişkenler ile bağımsız değişkenler arasındaki ilişkiler, regresyon analizi yoluyla istatistiksel olarak tanımlanabilir. Bağımlı değişkenlerin değerleri, bağımsız değişkenlerin gözlemlenen değerlerinden tahmin edilebilir. Böylece sonucu etkileyen risk faktörleri kolaylıkla belirlenebilir[23].

Tahmine dayalı regresyonlar yaygın olarak kullanılmaktadır çünkü öngörülebilirlik yıllardır öncelikli hedef olmuştur. Ancak tahmine dayalı regresyonlardaki en önemli sorun gürültü sorunudur. Ne kadar çok değişken ile çalışılırsa algoritmalar daha az tahmin yeteneği göstermektedir. Bu nedenle değişken seçimleri çok önemlidir. Yaklaşık olarak 30 yılda yapılan araştırmalar ele alındığında bu gürültüyü en aza indirmek için kullanılan tahmine dayalı başlıca yöntemlerden biri de mutlak küçültme ve seçme operatörü olan Lasso[24] ve RIDGE[25] öne çıkmaktadır. Lasso değişken seçimi tutarlılığında sahip bir yöntem olduğu için bir çok çalışmada tercih edilmiştir[26].

Lasso yönteminin avantajları, değişken seçimi üzerindeki etkili performansı ve geniş kullanım alanıyla doğrulanmaktadır. Bu yöntemin önemli bir özelliği, bazı katsayıların tam olarak sıfıra eşit olduğu seyrek bir tahmin sağlamasıdır. Yani, Lasso ile elde edilen β tahmini, bazı özelliklerin etkisinin tamamen ortadan kalktığı ve modeldeki sadece belirli özelliklerin korunduğu bir yapı oluşturur[27].

---

[20] Feigelson, E. D. ve Babu, G.J. Linear regression in astronomy. *II. Astrophysical Journal, Part 1 (ISSN 0004-637X), 397,1,* 55-67., 397, 55-67. 1992.

[21] Wiley, J. Applied linear regression. *United States of America.* 83. 2005.

[22] Su, X., Yan, X., ve Tsai, C.L. Linear regression. Wiley Interdisciplinary Reviews: *Computational Statistics, 4*(3), 275-294. 2012.

[23] Schneider, A. Hommel, G. ve Blettner, M. Linear regression analysis: part 14 of a series on evaluation of scientific publications. *Deutsches Ärzteblatt International, 107*(44), 776. 2010.

[24] Tibshirani, R. Regression shrinkage and selection via the Lasso. *J. R. Stat. Soc. Ser. B Stat. Methodol. 58* (1), 267–288. 1996.

[25] Hoerl, A.E. Kennard, R.W. Baldwin, K.F. Ridge regression: some simulations. *Communications in Statistics 4,* 105–123. 1975.

[26] Zou, H. The adaptive Lasso and its oracle properties. J. Amer. *Statist. Assoc. 101* (476), 1418–1429. 2006.

[27] Bunea, F. She, Y. Ombao, H. Gongvatana, A. Devlin, K. ve Cohen, R. *Penalized least squares regression methods and applications to neuroimaging. Neuroimage, 55*(4), 1519-1527. 2011.

Lasso regresyonu, elde edilen katsayı tahminlerinin seyrek olması nedeniyle Ridge regresyonuna göre önemli bir avantaja sahiptir. Çapraz doğrulamayla "0" katsayıya düşen değişkenlerin sistemden çıkarılmasıyla değişken sayısı azalır[28]. Bu çalışmalar neticesinde tahmine dayalı regresyonlar ile ilgili çıkarımsal bilgilere rastlanılmaktadır[29].

Ridge ve genelleştirilmiş ters tahmincilerin avantajlarından biri, hesaplamaların kolaylığıdır. Bu yöntemlerde X'X ve X'Y matrisleri bir kez hesaplanır ve ardından ölçeklendirilerek korelasyon matrisi oluşturulur. Sırt regresyonu için, genellikle (X'X + kI)'nın tersinin alınması gibi, her bir λ (lambda) değeri için bir kez yapılacak basit inversiyon işlemleri, sırt izinin nerede stabilize olduğunu belirlemek için yeterlidir[30].

Ridge ve Lasso'nun etkinliği, kullanılan veri setinin özelliklerine ve hedefine bağlıdır. Literatürde yapılan pek çok çalışma, bu yöntemlerin özellikle yüksek boyutlu (high-dimensional) veri setlerinde ve çoklu doğrusal bağlantı sorunuyla karşılaşıldığında faydalı olduğunu göstermektedir[31].

Ridge, bir istatistiksel öğrenme yöntemidir ve yaygın olarak finansal piyasa analizlerinde kullanılmaktadır. Ridge regresyonu, L2 düzenlemesi olarak da bilinir ve modelin genel karmaşıklığını kontrol eder. Ridge regresyonu, genellikle çoklu doğrusal bağlantı (multicollinearity) sorunuyla başa çıkmak için kullanılır. Bu durumda, bağımsız değişkenler arasında yüksek bir korelasyon olduğunda, Ridge regresyonu modelin kararlılığını artırabilir.

Lasso regresyonu ise L1 düzenlemesi olarak adlandırılır ve katsayıları sıfıra yaklaştırarak aynı zamanda değişken seçimi (variable selection) yapar. Bu özellik, modeldeki önemsiz değişkenleri elemine ederek daha basitleştirilmiş ve yorumlanabilir modeller elde etmeye yardımcı olabilir.

Bu iki yöntem de modelin aşırı uyuma (overfitting) eğilimini kontrol etmek ve tahmin performansını artırmak amacıyla geliştirilmiştir. Bununla birlikte, Ridge ve Lasso'nun en önemli dezavantajı parametrelerin doğru şekilde seçilmesinin zor olabilmesidir. Ridge ve Lasso'nun kullanılmasıyla elde edilen modellerin daha az yorumlanabilir olması da bir diğer dikkate değer husustur.

---

28 James, G. Witten, D. Hastie, T. ve Tibshirani, R. *An introduction to statistical learning* (Vol. 112, p. 18). New York: springer. 2013.

29 Kostakis, A. Magdalinos, T. ve Stamatogiannis, M.P. Robust econometric inference for stock return predictability. *Rev. Financ. Stud. 28* (5), 1506–1553. 2014. Lee, J.H. Predictive quantile regression with persistent covariates: *IVX-QR approach. J. Econometrics 192 (*1), 105–118. 2016.

30 Marquardt, D.W. ve Snee, R.D. Ridge regression in practice. *The American Statistician, 29*(1), 3-20. 1975.

31 Hoerl, A.E. Kennard, R.W. ve Baldwin, K.F. Ridge regression: some simulations. *Communications in Statistics 4,* 105–123. 1975. McDonald, G.C. ve Galarneau, D.I. *A Monte Carlo evaluation ofsome ridge-type estimators.* 1975. Hocking, R.R. Speed, F.M. ve Lynn, M.J. A class of biased estimators in linear regression. *Technometrics 18* (4), 425–437. 1976. Pasha, G.R. ve Shah, M.A. (2004). Application of ridge regression to multicollinear data. *Journal of Research (Science) 15* (1), 97–106. Dorugade, A.V. Kashid, D.N. Alternative method for choosing ridge parameter for regression. *International Journal of Applied Mathematical Sciences 4* (9), 447–456. 2010.

Sonuç olarak, Ridge ve Lasso regresyonları, çoklu bağlantı sorunuyla başa çıkmak için en önemli yöntemlerdir. Bu yöntemlerle ilgili olarak literatürde pek çok çalışma bulunmaktadır ve uygulama alanlarında da başarıyla kullanılmaktadır.

3. MATERYAL VE METOT

Bu çalışmanın amacı; makine öğrenmesi kullanılarak Lineer regresyon, Lasso regresyon ve Ridge regresyon modellerinin uygulamaları yapılarak analizlerinin çıkarılması, sonuçlarının incelenerek karşılaştırılma yapılmasıdır.

3.1. Kodlama Materyalleri

Yapılan çalışmada; makine öğrenmesi ve analizler için Python 3.7 yazılım programı ve derleyicileri kullanılmıştır. Python, yazılım geliştirmede, web uygulamaları tasarlamada, veri madenciliği ve makine öğrenmesinde yaygın olarak kullanılan nesne tabanlı, yüksek seviyeli bir programlama dilidir. Yazılım geliştiriciler tarafından; farklı platformlarda çalıştırılabildiği, öğrenmesi kolay ve etkili olduğu için Python son yıllarda çok tercih edilen bir yazılım dili olmuştur. Ayrıca Python ücretsiz bir yazılımdır, hemen her türlü sistemle entegre edilebilir ve geliştirme hızını arttırır.

Veri bilimciler, Python'un makine öğrenimi kitaplıklarını kullanarak çeşitli sınıflandırıcılar oluştururlar ve bu sınıflandırıcıları, görüntü, metin, ağ trafiği, konuşma ve yüz tanıma gibi farklı alanlarda kullanırlar. Ayrıca, veri bilimciler, derin öğrenme gibi gelişmiş makine öğrenimi tekniklerini uygulamak için de Python'u tercih ederler. Bu çalışmada da kullanılacağı gibi veri bilimi ile çalışan kişiler çalışmalarında pythondan aşağıdaki görevler için yararlanırlar:

- Veri temizleme yani yanlış verileri düzeltme veya kaldırma,
- Verileri çeşitli özelliklerine göre seçme ve ayıklama,
- Veri etiketleme,
- Verileri analiz ederek farklı sonuçlar çıkarma,
- Çeşitli grafikleri kullanarak verileri görselleştirmek.

Bu tez çalışmasında, Python içinde kullanılan bazı kütüphaneler aşağıda sıralanmıştır;

- import seaborn
- numpy
- plotly.express
- pandas
- sklearn.linear_model
- warnings
- matplotlib.pyplot
- sklear

3.2. Öğrenme Metodolojisi

Bu çalışmada izlenecek prosedürler aşağıda sıralanmıştır:

- Öncelikle çeşitli sorgulama ve analizler ile veri setinde bulunan bilgilerin anlaşılması,
- Yapılacak analizler için verinin uygun hale getirilmesi,

- Veri setinin bir kısmı eğitim için, bir kısmı da test için kullanılmak üzere ayrılarak sırasıyla Lineer Regresyon, Lasso regresyon ve Ridge regresyon için eğitilmesi;
- Her bir regresyonun yeni girilen verilere göre bağımlı değişkeni tahmin etme doğruluk oranlarının bulunması;
- Lasso ve Ridge' de kullanılan ceza katsayılarının sistemin doğruluk oranını nasıl etkilediğinin gözlemlenmesi olacaktır.

3.3. Veri Setinin İncelemesi

Bu çalışmada kullanılan veri seti kamuya açık olup, içerisinde veri bilimi çalışmaları yapmak için 50.000'den fazla veri kümesi bulunan www.kaggle.com sitesinden alınmıştır. Kullanılan veri seti; Amerika'da lise öğrencilerinin çeşitli derslerde kazandıkları notlardan oluşmaktadır. Bu veri; 1.000 satır ve 8 değişkenden ("gender (Cinsiyet)", "race/ethnicity (Irk/Etnik)", "parental level of education (Ebeveyn eğitim düzeyi)", "lunch (Öğle yemeği)", "test preparation course" (Sınava hazırlık kursu)", "math score (Matematik puanı), "reading score (Okuma puanı)", " writing score (Yazma puanı)" oluşan, lise öğrencilerinin bazı derslerden aldıkları notlar ve bu notları etkileyen çeşitli kişisel, sosyal ve ekonomik faktörleri içeren bir veri setidir.

İçeriğine örnek olması amacıyla veri setinin ilk elli satırı Şekil 3.1'de gösterilmiştir.

gender	race/ethnicity	parental level of education	lunch	test preparation course	math score	reading score	writing score
female	group B	bachelor's degree	standard	none	72	72	74
female	group C	some college	standard	completed	69	90	88
female	group B	master's degree	standard	none	90	95	93
male	group A	associate's degree	free/reduced	none	47	57	44
male	group C	some college	standard	none	76	78	75
female	group B	associate's degree	standard	none	71	83	78
female	group B	some college	standard	completed	88	95	92
male	group B	some college	free/reduced	none	40	43	39
male	group D	high school	free/reduced	completed	64	64	67
female	group B	high school	free/reduced	none	38	60	50
male	group C	associate's degree	standard	none	58	54	52
male	group D	associate's degree	standard	none	40	52	43
female	group B	high school	standard	none	65	81	73
male	group A	some college	standard	completed	78	72	70
female	group A	master's degree	standard	none	50	53	58
female	group C	some high school	standard	none	69	75	78
male	group C	high school	standard	none	88	89	86
female	group B	some high school	free/reduced	none	18	32	28
male	group C	master's degree	free/reduced	completed	46	42	46
female	group C	associate's degree	free/reduced	none	54	58	61
male	group D	high school	standard	none	66	69	63
female	group B	some college	free/reduced	completed	65	75	70
male	group D	some college	standard	none	44	54	53
female	group C	some high school	standard	none	69	73	73
male	group D	bachelor's degree	free/reduced	completed	74	71	80
male	group A	master's degree	free/reduced	none	73	74	72
male	group B	some college	standard	none	69	54	55
female	group C	bachelor's degree	standard	none	67	69	75
male	group C	high school	standard	none	70	70	65
female	group D	master's degree	standard	none	62	70	75
female	group D	some college	standard	none	69	74	74
female	group B	some college	standard	none	63	65	61
female	group E	master's degree	free/reduced	none	56	72	65
male	group D	some college	standard	none	40	42	38
male	group E	some college	standard	none	97	87	82
male	group E	associate's degree	standard	completed	81	81	79
female	group D	associate's degree	standard	none	74	81	83
female	group D	some high school	free/reduced	none	50	64	59
female	group D	associate's degree	free/reduced	completed	75	90	88
male	group B	associate's degree	free/reduced	none	57	56	57
male	group C	associate's degree	free/reduced	none	55	61	54
female	group C	associate's degree	standard	none	58	73	68
female	group B	associate's degree	standard	none	53	58	65
male	group B	some college	free/reduced	completed	59	65	66
female	group E	associate's degree	free/reduced	none	50	56	54
male	group B	associate's degree	standard	none	65	54	57
female	group A	associate's degree	standard	completed	55	65	62
female	group C	high school	standard	none	66	71	76
female	group D	associate's degree	free/reduced	completed	57	74	76
male	group C	high school	standard	completed	82	84	82

Şekil 3.1. Veri setinin ilk 50 satırı

Kullanılan veri setinde herhangi eksik bir veri bulunmamaktadır. Bunu gösteren her bir değişken için verilerin tam olduğu bilgisi Şekil 3.2'de yer almaktadır.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1000 entries, 0 to 999
Data columns (total 8 columns):
 #   Column                       Non-Null Count  Dtype
---  ------                       --------------  -----
 0   gender                       1000 non-null   object
 1   race/ethnicity               1000 non-null   object
 2   parental level of education  1000 non-null   object
 3   lunch                        1000 non-null   object
 4   test preparation course      1000 non-null   object
 5   math score                   1000 non-null   int64
 6   reading score                1000 non-null   int64
 7   writing score                1000 non-null   int64
dtypes: int64(3), object(5)
```

Şekil 3.2 Veri setindeki toplam veri

Veri setinin daha iyi anlaşılması için her değişkenin Türkçe anlamı Tablo 3.1'de birer örnek veri ile verilmiştir.

Tablo 3.1 Öğrencilerin sınavlardaki performansı veri seti açıklaması

Değişken	Anlamı	Örnek veri
gender	Cinsiyet	female
race/ethnicity	Irk / etnik	group B
parental level o feducation	Ebeveyn eğitim düzeyi	master's degree
Lunch	Öğle yemeği düzeyi	free/reduced
test preparati on course	Kursta test hazırlığı yamamlamış mı?	none
math score	Matematik Sınav Sonucu	72
reading score	Okuma Sınav Sonucu	72
writing score	Yazma Sınav Sonucu	74

Tablo 3.2. Öğrencilerin Sınavlardaki Performansı veri seti açıklaması

4. BULGULAR VE TARTIŞMALAR

4.1. Veri İşleme

Veri setindeki değişkenlerin 5 tanesi kategorik değişken, 3 tanesi ise sayısal değişkenden meydana gelmektedir. Tablo 4.1'de kategorik ve sayısal değişkenlerin hangileri olduğu listelenmiştir.

Tablo 4.1 Veri setinde kullanılan değişkenlerin türleri

Değişken	Türü
gender	Kategorik
race/ethnicity	Kategorik
parental level o feducation	Kategorik
lunch	Kategorik
test preparati on course	Kategorik
math score	Sayısal
reading score	Sayısal
writing score	Sayısal

Çalışmada yapılacak işlemler ve analiz sonuçlarının daha iyi anlaşılması için veri setindeki değişken başlıkları Türkçe'ye çevrilmiştir.

```
data.rename(columns={'gender':'Cinsiyet','race/ethnicity': 'Irk', 'parental level of education': 'Ebeveyn_Egitim',
                     'lunch':'Ogle_Yemegi','test preparation course':'Kursta_Test_Hazirligi','math score':'Matematik_Notu',
                     'reading score':'Okuma_Notu','writing score':'Yazma_Notu'},inplace=True)
df=data
df.head()
```

Şekil 4.1 Değişken Başlıkları Türkçeye Çevrilmesi Kodu

	Cinsiyet	Irk	Ebeveyn_Egitim	Ogle_Yemegi	Kursta_Test_Hazirligi	Matematik_Notu	Okuma_Notu	Yazma_Notu
0	female	group B	bachelor's degree	standard	none	72	72	74
1	female	group C	some college	standard	completed	69	90	88
2	female	group B	master's degree	standard	none	90	95	93
3	male	group A	associate's degree	free/reduced	none	47	57	44
4	male	group C	some college	standard	none	76	78	75

Şekil 4.2. Değişken başlıkları Türkçeye çevrilmesi ile veri setinin ilk 5 satırı

Türkçeye çevrilen değişken başlıkları Tablo 4.2'de gösterilmiştir.

Tablo 4.2. Yeni Türkçe Değişken başlıkları

Başlıklar	**Türkçe Karşılığı**
gender	Cinsiyet
race/ethnicity	Irk
parental level o feducation	Ebeveyn_Egitim
lunch	Ogle_Yemegi
test preparati on course	Kursta_Test_Hazirligi
math score	Matematik_Notu
reading score	Okuma_Notu
writing score	Yazma_Notu

Cinsiyet **:** Öğrencinin cinsiyet bilgisini içermektedir.

Irk **:** Öğrencinin etnik kökenini, hangi ulustan olduğu bilgisini içermektedir. Veri setinde "Grup" isimleri olarak belirtilmiştir.

Ebeveyn_Egitim **:** Öğrencinin ebeveyninin eğitim düzeyi bilgisini içermektedir.

Ogle_Yemegi **:** Öğrencinin ne kadar öğle yemeği yediği bilgisini vermektedir.

Kursta_Test_Hazirligi: Öğrencinin aldığı kursta yaptığı test hazırlığını tamamlamış olma bilgisidir.

Matematik_Notu **:** Öğrencinin matematik dersinden aldığı notu belirtmektedir.

Okuma_Notu **:** Öğrencinin okuma dersinden aldığı notu belirtmektedir.

Yazma_Notu **:** Öğrencinin yazma dersinden aldığı notu belirtmektedir.

Kullanılan veri setindeki değişkenlerin benzersiz kaç farklı değer aldığı bilgisi Tablo 4.3'de gösterilmektedir.

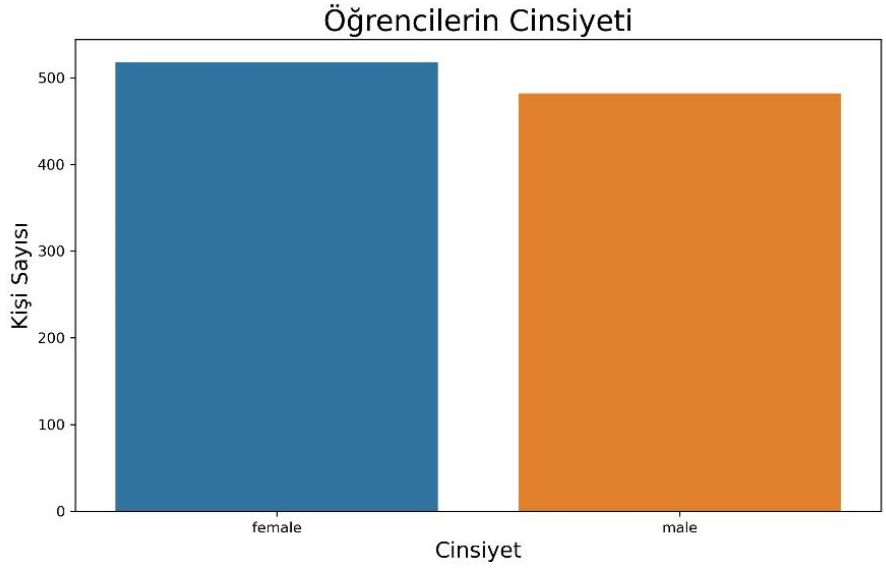
Tablo 4.3. Değişkenlerin benzersiz değer sayıları

Değişken Adı	**Benzersiz Değer Sayısı**
Cinsiyet	2
Irk	5
Ebeveyn_Egitim	6
Ogle_Yemegi	2
Kursta_Test_Hazirligi	2
Matematik_Notu	81
Okuma_Notu	72
Yazma_Notu	77

Veri setinde cinsiyet kadın ve erkek olarak 2 farklı değer alırken, öğrencilerin milletlerini gösteren ırk değişkeni 5 farklı gruba ayrılmıştır. Öğrencilerin ebeveynlerinin öğrenim durumunu gösteren değişken 6 farklı kategoride toplanmıştır. Öğrencilerin öğle yemeklerini ne ölçüde yedikleri iki kategoriye ayrılmış ve kursta test hazırlığı yapıp yapmadıkları da iki kategoride bildirtilmiştir. Öğrencilerin notlarını etkileyen bu değişkenlerin sonucunda matematikten 81, Okuma'dan 72 ve Yazma'dan 77 farklı not kaydedilmiştir.

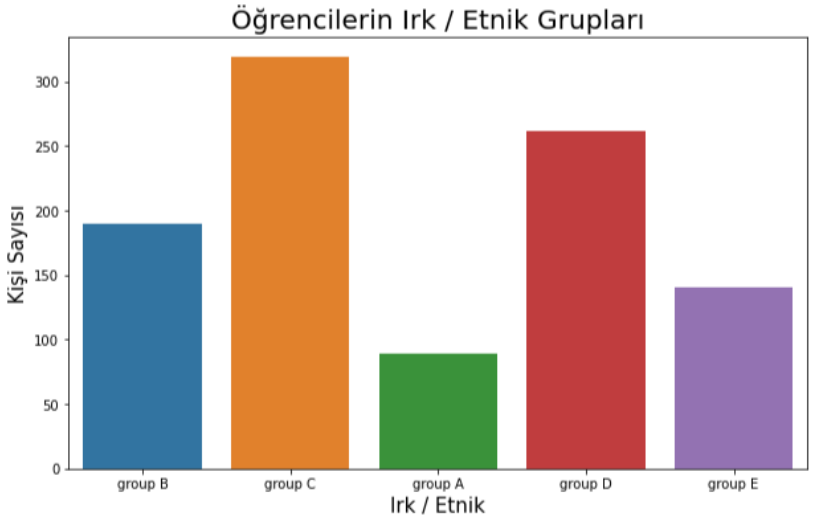
4.2. Veri Görselleştirme

Çalışmada kullanılacak veri içindeki değişkenlerin benzersiz her farklı başlığının kaçar değer aldığını gösteren grafikler aşağıda bulunmaktadır.



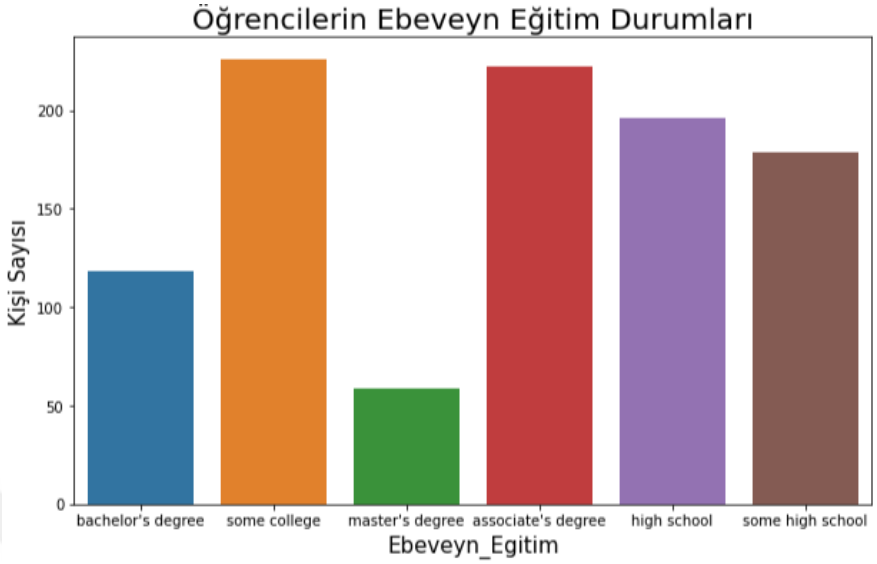
Şekil 4.3. Cinsiyet Dağılımı

Araştırmada 518 kadın, 482 erkek öğrenci üzerinde çalışılmıştır.



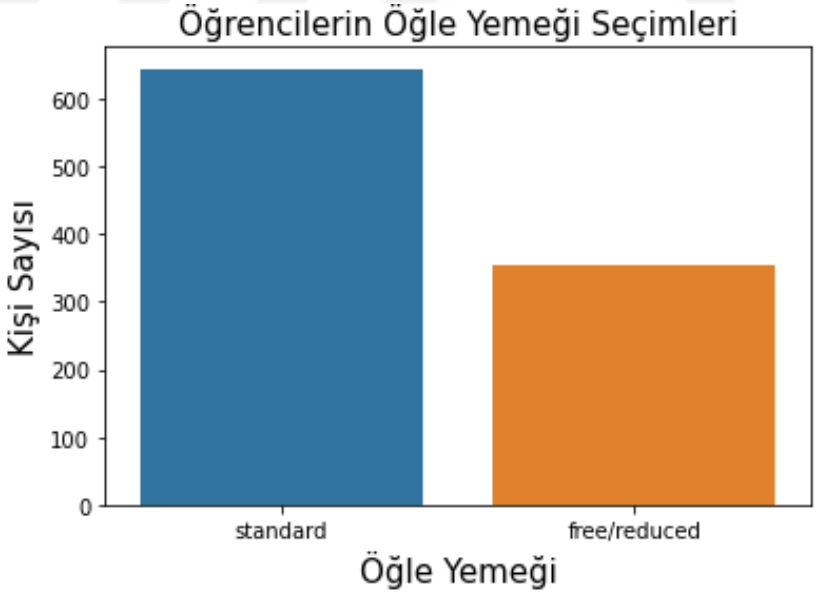
Şekil 4.4. Etnik Grup Dağılımı

Bu öğrencilerin; 89'u Grup A, 190'ı Grup B, 319'u Grup C, 262'si Grup D ve 140'ı Grup E ırkındandır.



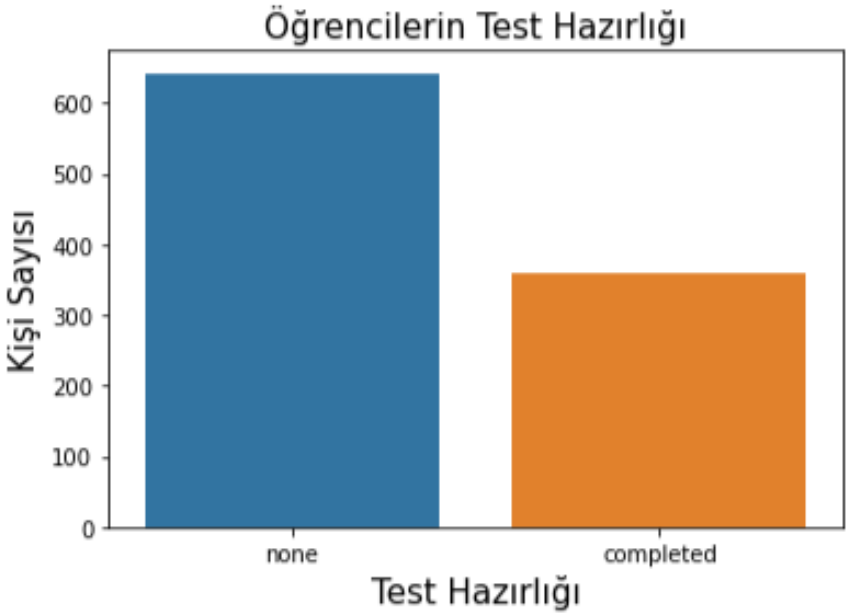
Şekil 4.5. Ebeveyn Eğitim Dağılımı

Öğrencilerin ebeveynlerinin; 118'i lisans, 226'sı kolej, 59'u lisans, 222'si ön lisans, 196'sı lise, 179'u lise dengi okul mezunudur.



Şekil 4.6. Öğle Yemeği Seçim Dağılımı

Araştırmadaki öğrencilerin; 645'i standart öğle yemeği yerken, 355'i ücretsiz/azaltılmış öğle yemeği yemektedir.



Şekil 4.7. Test Hazırlığı Dağılımı

Öğrencilerin; 358 öğrenci kurstaki test hazırlıklarını tamamlamışken 642'si test hazırlığı yapmamıştır.

4.3. Veri Dönüştürme

Makine öğrenmesi uygulamalarında kullanılacak verinin numerik yani sayısal değişkenlere sahip olması gerekmektedir. Bu nedenle yapılacak çalışmada makine öğrenmesini gerçekleştirebilmek için veri setindeki kategorik değişkenlerin sayısal değişkenlere çevrilmesi sağlanmıştır.

```
from sklearn import preprocessing
cevir=preprocessing.LabelEncoder()
df[('Cinsiyet')]=cevir.fit_transform(df[('Cinsiyet')])
df[('Irk')]=cevir.fit_transform(df[('Irk')])
df[('Ebeveyn_Egitim')]=cevir.fit_transform(df[('Ebeveyn_Egitim')])
df[('Ogle_Yemegi')]=cevir.fit_transform(df[('Ogle_Yemegi')])
df[('Kursta_Test_Hazirligi')]=cevir.fit_transform(df[('Kursta_Test_Hazirligi')])
df
```

Şekil 4.8. Veri setindeki kategorik verileri sayısal verilere çeviren kod

	Cinsiyet	Irk	Ebeveyn_Egitim	Ogle_Yemegi	Kursta_Test_Hazirligi	Matematik_Notu	Okuma_Notu	Yazma_Notu
0	0	1	1	1	1	72	72	74
1	0	2	4	1	0	69	90	88
2	0	1	3	1	1	90	95	93
3	1	0	0	0	1	47	57	44
4	1	2	4	1	1	76	78	75
...	...	...	...	...	...	...	...	...
995	0	4	3	1	0	88	99	95
996	1	2	2	0	1	62	55	55
997	0	2	2	0	0	59	71	65
998	0	3	4	1	0	68	78	77
999	0	3	4	0	1	77	86	86

1000 rows × 8 columns

Şekil 4.9. Veri setinin sayısal değişkenler ile beraber görünümü

Sadece sayısal değerlere sahip olan veri setindeki her değişkenin içindeki verinin dağılımı Şekil 4.11'de farklı bir şekilde gösterilmiştir.

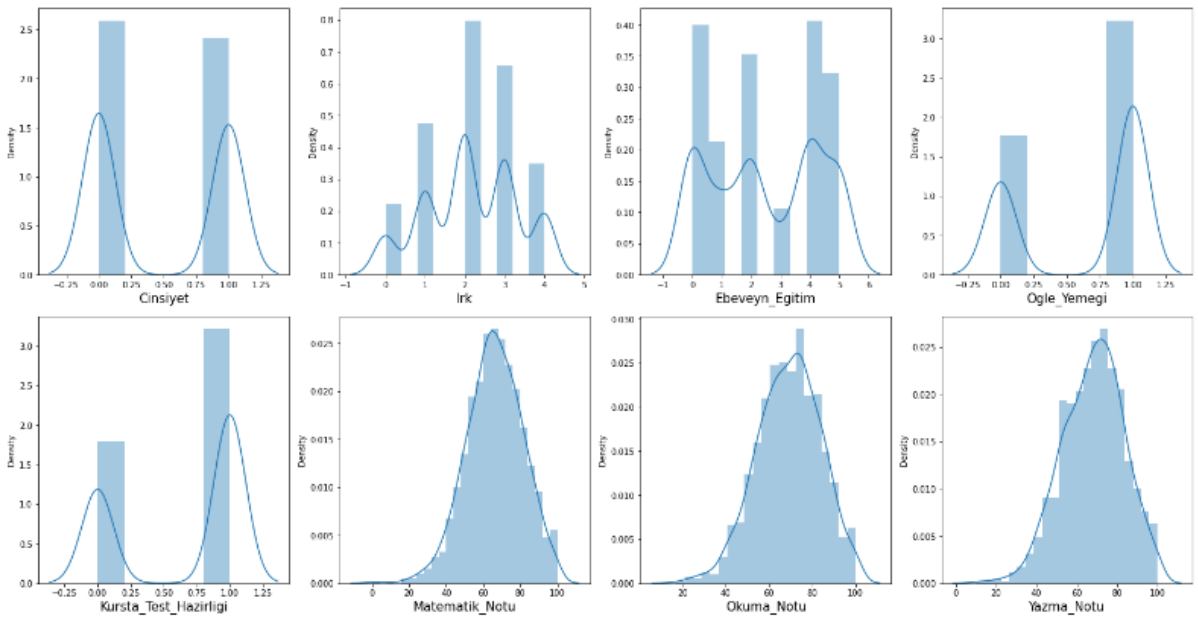
```
plt.figure(figsize = (20, 15))
plotnumber = 1

for column in df:
    if plotnumber <= 8:
        ax = plt.subplot(3, 4, plotnumber)
        sns.distplot(df[column])
        plt.xlabel(column, fontsize = 15)

    plotnumber += 1

plt.tight_layout()
plt.show()
```

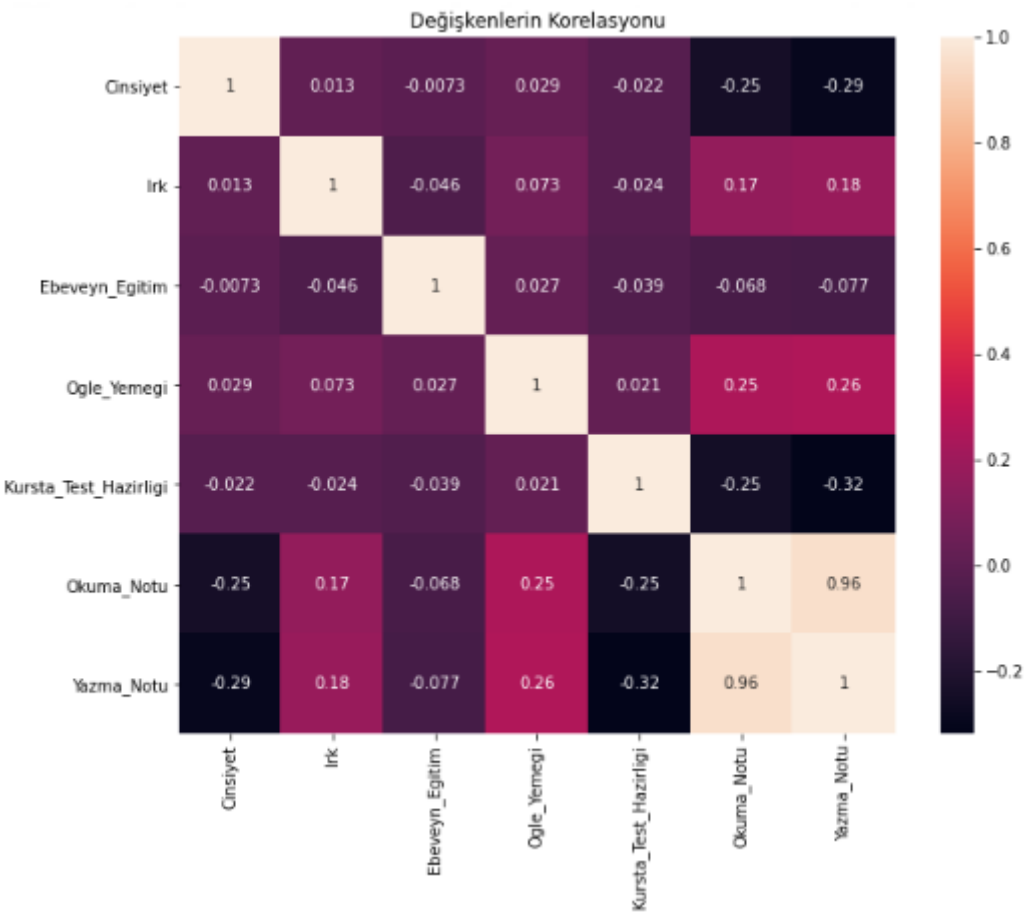
Şekil 4.10. Sütunlara göre veri dağılımını garfikleyen kod



Şekil 4.11. Sütunlara göre veri dağılımı

Sütunlara göre veri dağılımına Şekil 4.11'de bakıldığında çok aykırı bir veri dağılımı olmadığı, verinin genel anlamda dengeli dağılım gösterdiği gözlenmektedir. Bu durum yapılacak analizlerin sağlıklı olması için önem arz etmektedir.

Makine öğrenmesinde bir diğer önemli nokta da değişkenlerin birbirleri ile olan korelasyonu yani aralarındaki doğrusal ilişki olup olmadığıdır. Bunu görmek için veri setindeki değişkenlerin olduğu korelasyon matrisi oluşturuldu. Korelasyon matrisinde her özelliğin diğer tüm özelliklerle arasındaki korelasyon değerleri görülmektedir. Korelasyon değeri [-1, -0.5] aralığında olan özellikler arasında negatif, [0.5, 1] aralığında ise pozitif yönlü doğrusal bir ilişki olduğu söylenebilir. Bu durumda birbirleri ile ilişkili verilerden sadece bir tanesinin makine öğrenmesi için kullanılması diğerlerinin sistem dışı bırakılması doğru olacaktır. Bu aralık dışında kalan değerler ise o değişkenler arasında ilişki olmadığını göstermektedir.



Şekil 4.12. Değişkenlerin korelasyonu grafiği

Veri seti için oluşturulan korelasyon matrisine bakıldığında neredeyse tüm değişkenlerin birbirleriyle çok düşük korelasyona sahip olduğu görülmektedir. Sadece "Okuma Notu" ve "Yazma Notu" birbiriyle yüksek düzeyde ilişkili görülmektedir. Her iki değişken de çalışma için önemli olduğundan bu korelasyona izin verilmektedir.

4.4. Veri Üzerinde Makine Öğrenmesi Uygulamaları

Buraya kadar yapılan işlemler ile kullanılacak veri makine öğrenmesi için uygun duruma getirilmiştir. Bu bölümde veri seti üzerinde, daha önce de belirtildiği gibi makine öğrenmesi ile sistem eğitilerek denetimli öğrenme algoritmalarından Lineer Regresyon, Lasso ve Ridge Regresyonları uygulanacaktır. Veri setinde hedef değişken "Matematik Puanı", diğer değişkenler ise "Matematik Puanı"nı bulmak için kullanacak bağımsız değişkenlerdir.

```
X=df[['Cinsiyet','Irk','Ebeveyn_Egitim','Ogle_Yemegi','Kursta_Test_Hazirligi','Okuma_Notu','Yazma_Notu']]
Y=df['Matematik_Notu']
```

Şekil 4.13. Bağımlı ve bağımsız değişkeni belirleyen kod

Amaç; denetimli makine öğrenimi modellerini kullanarak yazılan program ile değişkenlerin aldığı değerlere göre öğrencilerin "Matematik puanı"nı tahmin edilmesini sağlamaktır. Bunun için veri setinin %80'i sistemin eğitilmesi için, kalan %20'si ise test için kullanılacaktır.

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, Y, test_size=0.2, random_state=101)
print(X_train.shape)
print(X_test.shape)
print(y_train.shape)
print(y_test.shape)
```

```
(800, 7)
(200, 7)
(800,)
(200,)
```

Şekil 4.14. Sistemi eğitim ve test için belirleyen kod

4.4.1. Lineer Regresyon Uygulaması

Veri seti üzerinde ilk olarak Lineer Regresyon modeli, makine öğrenmesi sistemi üzerinde uygulanmıştır.

```
#Doğrusal regresyon modeli oluşturma
from sklearn.linear_model import LinearRegression
model = LinearRegression()
model.fit(X_train,y_train)
```

```
LinearRegression()
```

Şekil 4.15. Lineer regresyon uygulaması kodu

Lineer Regresyon sonucunda gerçek veriler ile tahmin edilen veriler arasında nasıl bir ilişki olduğunu göstermek amacıyla sonuçlar regresyon grafiği üzerinde Şekil 4.17'de gösterilmektedir.

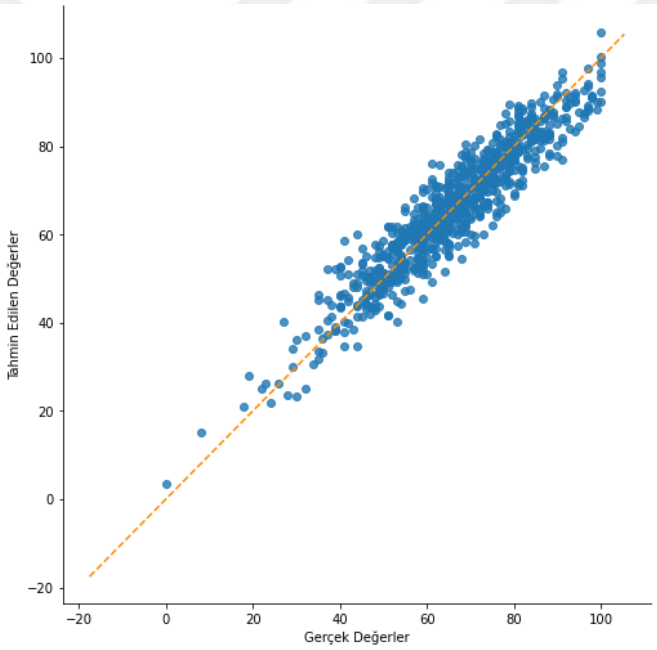
```python
# Doğrusallık Varsayımı

def calculate_residuals(model, features, label):
    predictions = model.predict(features)
    df_results = pd.DataFrame({'Gerçek Değerler': label, 'Tahmini Değerler': predictions})
    df_results['Residuals'] = abs(df_results['Gerçek Değerler']) - abs(df_results['Tahmini Değerler'])
    return df_results

def linear_assumption(model, features, label):
    df_results = calculate_residuals(model, features, label)
    sns.lmplot(x='Gerçek Değerler', y='Tahmini Değerler', data=df_results, fit_reg=False, size=7)
    line_coords = np.arange(df_results.min().min(), df_results.max().max())
    plt.plot(line_coords, line_coords, color='darkorange', linestyle='--')
    plt.title('Gerçek Değerler ve Tahmin Edilen Değerler')
    plt.savefig("Liner_Regresyon_Grafik.jpg",dpi=300)
    plt.show()

linear_assumption(model,X_train,y_train)
```

Şekil 4.16. Sonuçların dağılımını bulan ve grafikleyen kod



Şekil 4.17. Lineer regresyon grafiği

"Gerçek Değerler" ile "Tahmin Edilen Değerler" arasındaki Lineer Regresyon grafiğine bakıldığında görülüyor ki; değerler pozitif yönde doğru çizgisinin üzerinde ve çevresinde sıralanmaktadır. Bu olay, Lineer Regresyon sonucu elde edilen değerler arasında doğrusal bir ilişki olduğunu göstermektedir.

Lineer Regresyon sonucunda "artıkların" nasıl bir dağılım gösterdiği sistemin doğru çalıştığını göstermesi açısından önemlidir. Burada bahsedilen "artıklar" veya "kalıntılar", gerçek hedef değerleri ile modelin tahmin ettiği değerler arasındaki farklardır. Daha matematiksel bir ifadeyle, her bir gözlem için:

Artık (Kalıntı) = Gerçek Değer – Tahmini Değer

İstenen; tahmin değerlerinin gerçek değerlere mümkün olduğunca yakın olmasıdır. Ancak bazen model mükemmel tahminler yapamaz. Bu nedenle, artıklar modelin hatasını ölçer. Eğer artıklar küçük değerlere sahipse, bu modelin iyi çalıştığını göstermektedir. Eğer artıklar büyük değerlere sahipse, modelin üzerinde çeşitli işlemler yapılarak iyileştirilmesi gerekebilmektedir.

Uygulanan Lineer Regresyon modelinde artıkların nasıl dağıldığını göstermek için Anderson-Darling testi kullanılmaktadır.

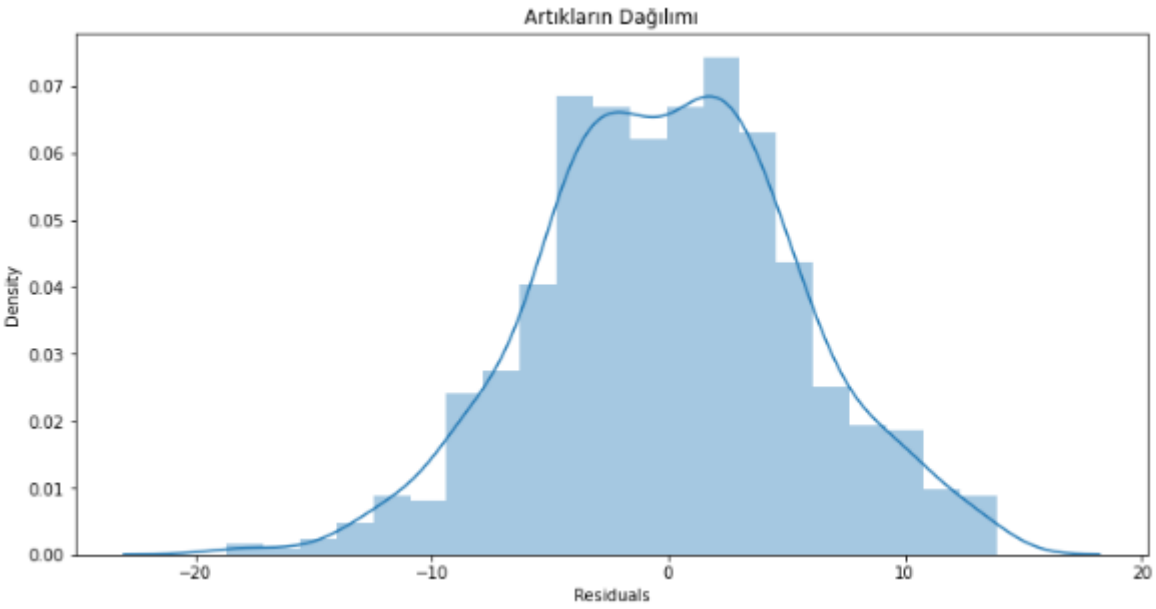
```python
# Hata terimleri normal dağılım gösterir

def normal_errors_assumption(model, features, label, p_value_thresh=0.05):
    from statsmodels.stats.diagnostic import normal_ad
    df_results = calculate_residuals(model, features, label)
    print('Normal Dağılım için Anderson-Darling testini kullanıldı')
    p_value = normal_ad(df_results['Residuals'])[1]
    print('Testten elde edilen p-değeri: ', p_value)
    if p_value < p_value_thresh:
        print('P-değeri 0,05 ten küçük olduğu için artıklar normal dağılmamıştır.')
    else:
        print('P-değeri 0,05 ten büyük olduğu için artıklar normal dağılım göstermektedir.')

    # Artık dağılımının çizilmesi
    plt.subplots(figsize=(12, 6))
    plt.title('Artıkların Dağılımı')
    sns.distplot(df_results['Residuals'])
    plt.savefig("Artıkların_Dağılımı_Grafik.jpg",dpi=300)
    plt.show()

normal_errors_assumption(model,X_train,y_train)
```

Şekil 4.18. Artıkların dağılımını hesaplayan ve grafikleyen kod



Şekil 4.19. Artıkların dağılımı grafiği

Test sonucunda elde edilen p değeri “0,4448110390787107” olarak bulunmaktadır. Bu değer 0,05’den büyük olduğundan artıkların normal dağılım gösterdiği söylenebilmektedir. Bu şekilde normal dağılım varsayımının sağlanması, regresyon analizinin çeşitli istatistiksel özelliklerinin doğru bir şekilde yorumlanabilmesi için önemlidir.

OLS Regression Results

Dep. Variable:	Matematik_Notu	R-squared:	0.872
Model:	OLS	Adj. R-squared:	0.871
Method:	Least Squares	F-statistic:	771.6
Date:	Thu, 07 Dec 2023	Prob (F-statistic):	0.00
Time:	10:33:19	Log-Likelihood:	-2501.3
No. Observations:	800	AIC:	5019.
Df Residuals:	792	BIC:	5056.
Df Model:	7		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	-12.6991	1.279	-9.932	0.000	-15.209	-10.189
Cinsiyet	13.5603	0.425	31.892	0.000	12.726	14.395
Irk	0.8781	0.174	5.048	0.000	0.537	1.220
Ebeveyn_Egitim	0.0383	0.108	0.355	0.722	-0.173	0.250
Ogle_Yemegi	3.6344	0.437	8.310	0.000	2.776	4.493
Kursta_Test_Hazirligi	3.0781	0.453	6.792	0.000	2.188	3.968
Okuma_Notu	0.3524	0.048	7.384	0.000	0.259	0.446
Yazma_Notu	0.6092	0.048	12.579	0.000	0.514	0.704

Omnibus:	0.521	Durbin-Watson:	1.940
Prob(Omnibus):	0.771	Jarque-Bera (JB):	0.487
Skew:	-0.060	Prob(JB):	0.784
Kurtosis:	3.007	Cond. No.	664.

Şekil 4.20. Lineer regresyon sonuçları

Lineer Regresyon modeline göre eğitim ve test verileri üzerindeki sonuçların doğruluk oranlarını gösteren R-kare değerleri ile ortalama karesel hata bilgileri “Tablo 4.4” de gösterilmektedir. R-kare, regresyon modelinin veri setini ne kadar iyi açıkladığını ölçen bir istatistiksel metriktir.

Tablo 4.4. Lineer Regresyon Doğruluk Sonuçları

Ölçülen Değer	Performans Sonucu
Lineer Regression Eğitim Seti Üzerinde R-kare	% 87,2
Lineer Regression Test Seti Üzerinde R-kare	% 83,9
Lineer Regression Ortalama Karesel Hata	31,6

4.4.2. Lasso Regresyon Uygulaması

Genellikle değişken seçimi (feature selection) yani gereksiz veya düşük etkili özelliklerin belirlenip modelden çıkarılması amacıyla kullanılan Lasso regresyonu (Least Absolute Shrinkage and

Selection Operator - En Küçük Mutlak Küçülme ve Seçim Operatörü) Lineer Regresyonun bir türüdür. Temelde, modele toplam karesel hatanın (least squares error) bir terimi olan ceza terimi (penalty term) alpha eklenir. En doğru alpha değeri bulunduğunda ve model bu alpha değerine göre eğitildiğinde en yüksek doğruluk oranına ulaşılmış olmaktadır.

Lasso regresyonunda kullanılacak doğru Alpha değerini bulmak amacıyla -2 ile 2 aralığında 0,002 artan farklı alpha değerleri için Lasso regresyon modeli eğitilerek, her bir modelin test verileri üzerindeki performansı ölçülmüş, elde edilen skorlar daha sonra analiz veya model seçimi için kullanılmak üzere "scores" listesine eklenmiştir.

```
#Lasso Regresyon modeli oluşturma
from numpy import arange
from sklearn.linear_model import Ridge
from sklearn.linear_model import Lasso
from sklearn.metrics import mean_squared_error

scores=[]
for i in arange(-2,2,0.002):
    ls = Lasso(alpha=10**i)
    print(10**i)

    ls.fit(X_train, y_train)
    y_pred = ls.predict(X_test)
    scores.append(ls.score(X_test, y_test))
```

Şekil 4.21. Lasso Regresyon modelini oluşturan ve alpha değerlerini bulan kod

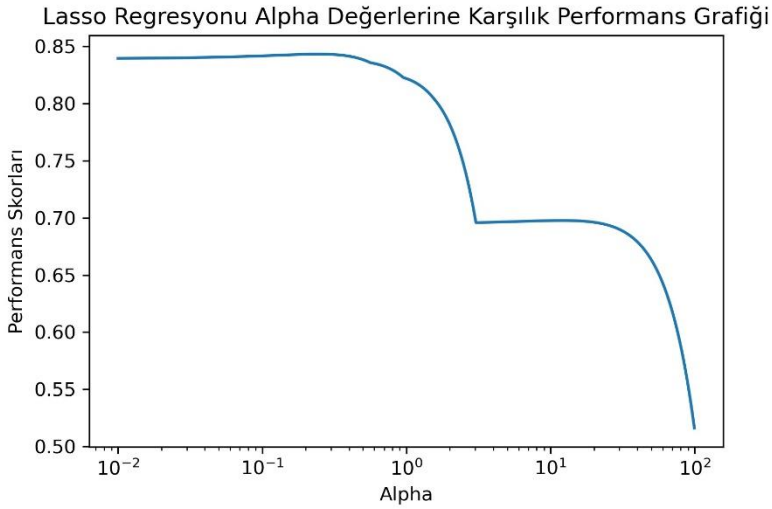
```
0.010185913880541169
0.010232929922807542
0.010280162981264735
0.010327614057613975
0.010375284158180127
0.010423174293933041
0.010471285480508996
0.01051961873823223
0.010568175092136586
0.010616955571987247
0.010665961212302578
0.010715193052376065
0.010764652136298349
0.01081433951297938
0.010864256236170655
0.010914403364487566
0.01096478196143185
```

Şekil 4.22. Sistem tarafından bulunan alpha değeri örnekleri

Farklı alpha değerlerine karşılık gelen Lasso regresyon modelinin performans skorları aşağıda bir çizgi grafiği ile gösterilmektedir. Bu görselleştirme, Lasso regresyon modelinin alfa değerine bağlı olarak nasıl performans gösterdiğini anlamak için kullanışlı olabilmektedir.

```
plt.plot(10**arange(-2,2,0.002),scores)
plt.xlabel('Alpha')
plt.ylabel('Performans Skorları')
plt.title('Lasso Regresyonu Alpha Değerlerine Karşılık Performans Grafiği')
plt.xscale('log')
plt.savefig("Lasso_Lamda_Sabit_Grafigi.jpg",dpi=300)
```

Şekil 4.23. Alpha değerlerine karşılık performans grafiği çizen kod



Şekil 4.24. Lasso regresyonu Alpha değerlerine karşılık performans grafiği

Şekil 4.24, x ekseni üzerinde logaritmik ölçekle alpha değerlerini ve y ekseni üzerinde modelin performansını temsil eden değerler içermektedir. Grafiğin maksimum noktası veya en yüksek skor, optimal alpha değerini temsil etmektedir. Bu değer, modelin en iyi genelleme yeteneğine sahip olduğu düzenleme seviyesini belirtmektedir. Oluşturulan grafiğe göre kullanılan veri için en uygun Lasso Regresyonu alpha değerinin 0 – 1 aralığında görülmektedir.

Lasso regresyon modeli, 10 katlı çapraz doğrulama (cross-validation) ile eğitirek en iyi alpha (düzenleme parametresi) değerinin bulunmasını sağlanmıştır. Çapraz doğrulama, modelin genelleme yeteneğini artırmak ve aşırı uyumu azaltmak için kullanılmaktadır. En iyi alpha değeri, çapraz doğrulama süreci içindeki performans ölçümlerine dayanarak belirlenmektedir. Yapılan çapraz doğrulama sonucu bulunan en iyi alpha değeri; 0.18801939531250014' dir.

Çapraz doğrulama sonucunda belirlenen en uygun alpha değeri ile eğitilen yeni bir Lasso regresyon modelini oluştulmuştur. Bu tip bir model, genellikle belirli bir düzenleme seviyesinde daha iyi bir performans gösterebilmekte ve modelin genelleme yeteneğini artırabilmektedir.

Veri setindeki değişkenlerin, oluşturulan Lasso modelinde önem sırasını görmek için aşağıda bulunan "Farklı Değişkenler için Lasso Katsayıları Grafiği" oluşturulmuştur. Grafikte farklı alpha değerleri için Lasso regresyon modelinin her değişken (özellik) için bulunan etki katsayılarını göstermektedir. Bu katsayılar, Lasso modelin her bir özellik için tahmin ettiği ağırlıklardır. Bu katsayılar her özelliğin model için önemini göstermektedir.

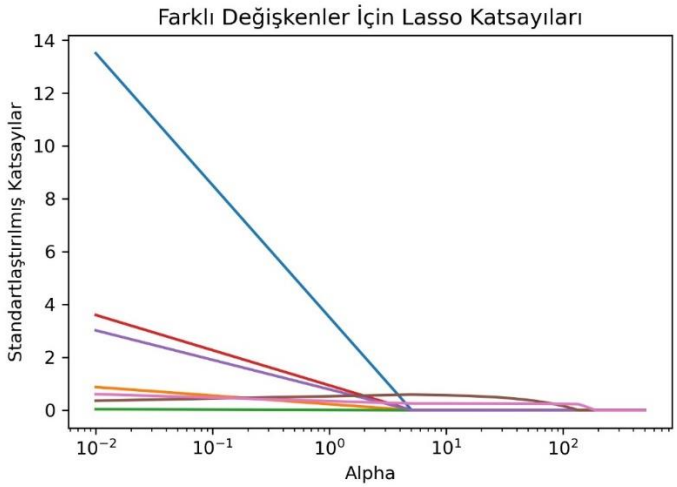
```
#alfa parametrelerinin rolü
alphas = np.linspace(0.01,500,100)
lasso = Lasso(max_iter=10000)
coefs = []

for a in alphas:
    lasso.set_params(alpha=a)
    lasso.fit(X_train, y_train)
    coefs.append(lasso.coef_)

ax = plt.gca()

ax.plot(alphas, coefs)
ax.set_xscale('log')
plt.axis('tight')
plt.xlabel('Alpha')
plt.ylabel('Standartlaştırılmış Katsayılar')
plt.title('Farklı Değişkenler İçin Lasso Katsayıları');
plt.savefig("Lasso_Katsayılar_Grafigi.jpg",dpi=300)
```

Şekil 4.25. Farklı Değişkenler için Lasso Katsayılarını bulan ve grafikleştiren kod



Şekil 4.26. Farklı değişkenler için Lasso katsayıları grafiği

Lasso regresyon modelinin değişken katsayılarını ve bu katsayıların karşılık geldiği özellikler gösterildiği Tablo 4.5'de gösterilmektedir.

Tablo 4.5. Lasso regresyonda değişkenlerin önem katsayıları tablosu

Değişken (Özellik)	**Katsayı**
Cinsiyet	12,639
Irk	0,786
Ebeveyn_Egitim	0,000
Ogle_Yemegi	2,981
Kursta_Test_Hazirligi	2,033
Okuma_Notu	0,390
Yazma_Notu	0,560

Tablo 4.5'e göre veri setinde bulunan değişkenlerden amaç değişkenimiz olan "Matematik_Notu" nu en çok "Cinsiyet" faktörünün etkilediği; "Ogle_Yemegi" değişkeni yani yeterli beslenme ve derse hazırlığı gösteren "Kursta_Test_Hazirligi"nın da "Matematik_Notu" üzerinde etkili görülmektedir. Öğrencilerin milliyetini gösteren "Irk", "Okuma_Notu" ve "Yazma_Notu" nun sisteme etkileri ise çok az olduğu ve "Ebeveyn_Egitim" nin ise sisteme hiçbir etkisi olmadığı Lasso Regresyon sonucunda belirlenmiş olmaktadır.

Lasso regresyon modelinin genelleme yeteneğini, doğruluğunu ve tahmin performansını değerlendirmek için eğitim seti üzerinde R-kare, test seti üzerinde R-kare ve test seti üzerinde ortalama karesel hatayı (Mean Squared Error, MSE) hesaplanmıştır. Hesaplanan sonuçlar "Tablo 4.6"da gösterilmiştir.

Tablo 4.6. Lasso regresyon modeli performans sonuçları

Ölçülen Değer	**Performans Sonucu**
Lasso Regresyon Eğitim Seti Üzerinde R-kare	% 87
Lasso Regresyon Test Seti Üzerinde R-kare	% 84,3
Lasso Regresyon Ortalama Karesel Hata	30.85

4.4.3. Ridge Regresyon Uygulaması

Ridge regresyonu, Lineer regresyona, L2 (modele katsayıların karesiyle orantılı olan bir ceza terimi ekler) düzenlemesi ekleyerek modelin genelleştirmesini artırmayı amaçlar. Bir çeşit Lineer Regresyonun türüdür ve ağırlıkların büyüklüğünü kontrol ederek aşırı öğrenmeyi önlemeye ve çoklu doğrusallık sorunu ile başa çıkmaya yardımcı olur.

Veri seti Ridge regresyon modeline göre yeniden eğitilerek yeni sonuçlarına ulaşılacaktır. Bunun için Ridge regresyonunda kullanılacak doğru alpha değerini bulmak amacıyla, her farklı alpha değeri için model tekrar eğitilerek, modelin test verileri üzerindeki performansı, MSE ve RMSE değerleri ölçülmüştür. Elde edilen skorlar "sonuc" listesine eklenmiştir. Bu sonuçlara göre doğru alpha değeri belirlenecektir.

```
#Ridge Regresyon modeli oluşturma

alphas = [0.01, 0.1, 1,5,10,15,20,25,30,50,60,70,80,90,100]

sonuc=[]
for i in alphas:
    ls = Ridge(alpha=i).fit(X_train, y_train)
    score = ls.score(X_test, y_test)
    pred_y = ls.predict(X_test)
    mse = mean_squared_error(y_test, pred_y)

    print("Alpha:{0:.6f}, R2:{1:.3f}, MSE:{2:.2f}, RMSE:{3:.2f}"
    .format(i, score, mse, np.sqrt(mse)))
    sonuc.append(mean_squared_error(pred_y, y_test))
```

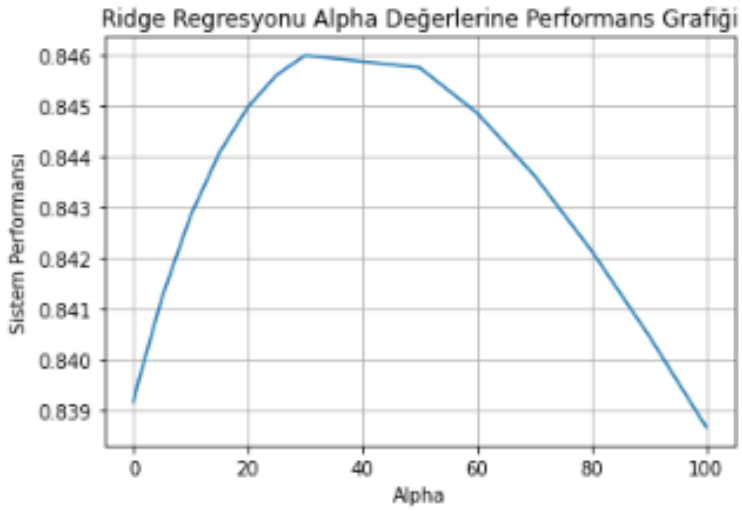
Şekil 4.27. Ridge Regresyon modeli oluşturma ve alpha değerlerini deneme kodu

```
Alpha:0.010000, R2:0.839, MSE:31.60, RMSE:5.62
Alpha:0.100000, R2:0.839, MSE:31.59, RMSE:5.62
Alpha:1.000000, R2:0.840, MSE:31.51, RMSE:5.61
Alpha:5.000000, R2:0.841, MSE:31.20, RMSE:5.59
Alpha:10.000000, R2:0.843, MSE:30.88, RMSE:5.56
Alpha:15.000000, R2:0.844, MSE:30.64, RMSE:5.54
Alpha:20.000000, R2:0.845, MSE:30.46, RMSE:5.52
Alpha:25.000000, R2:0.846, MSE:30.33, RMSE:5.51
Alpha:30.000000, R2:0.846, MSE:30.26, RMSE:5.50
Alpha:50.000000, R2:0.846, MSE:30.31, RMSE:5.51
Alpha:60.000000, R2:0.845, MSE:30.48, RMSE:5.52
Alpha:70.000000, R2:0.844, MSE:30.72, RMSE:5.54
Alpha:80.000000, R2:0.842, MSE:31.01, RMSE:5.57
Alpha:90.000000, R2:0.840, MSE:31.34, RMSE:5.60
Alpha:100.000000, R2:0.839, MSE:31.70, RMSE:5.63
```

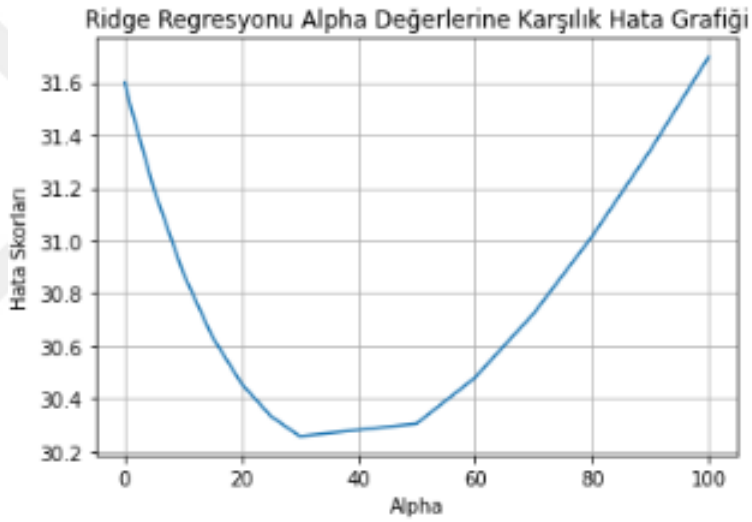
Şekil 4.28. Ridge model üzerinde denenen Alpha değeri sonuçları

Şekil 4.28'e göre seçilen alpha değerleri Ridge regresyon modelinde denenmiş ve her alpha değeri için R2 (doğruluk performansı), MSE (Mean Squared Error – Ortalama Karesel Hata), RMSE (Root Mean Squared Error - Kök Ortalama Kare Hatası) hesaplanmıştır. Şekil 4.28'de görüldüğü gibi alpha değeri bir noktaya kadar arttıkça model performansı da artarken Ortalama Karesel Hata ve Kök Ortalama Kare Hatası düşmektedir. Alpha değeri "30" olduğunda model en yüksek performansı vermekte ve en düşük hatalar gözlenmektedir. Bu noktadan sonra alpha değeri büyüdükçe modelin performansı düşmekte, hata değerleri artmaktadır.

Ridge regresyonunda değişen alpha değerleri ile sistem performansı artarken hata değerlerinin düşmesini ve sistem performansı düşerken hata değerlrinin artmasını daha iyi gözlemlemek için iki durum da Şekil 4.29 ve Şekil 4.30 da gösterilmiştir.



Şekil 4.29. Ridge regresyonu Alpha-Performans grafiği



Şekil 4.30. Ridge regresyonu Alpha-Hata grafiği

Grafiklerde de görüldüğü gibi; Şekil 4.29'de Sistem Performansının maksimum olduğu alpha değerinde Şekil 4.30'de Hata Skorları minimum noktadadır.

Optimum alpha sayısı ile eğitilen Ridge regresyonuna göre veri setindeki değişkenlerin model üzerindeki katsayıları Tablo 4.7.'de görülmektedir. Grafikte her değişkenin (öznitelik) farklı katsayılarla Ridge regresyon modelini etkilediği görülmektedir. Bu katsayılar her değişkenin model için önemini göstermektedir.

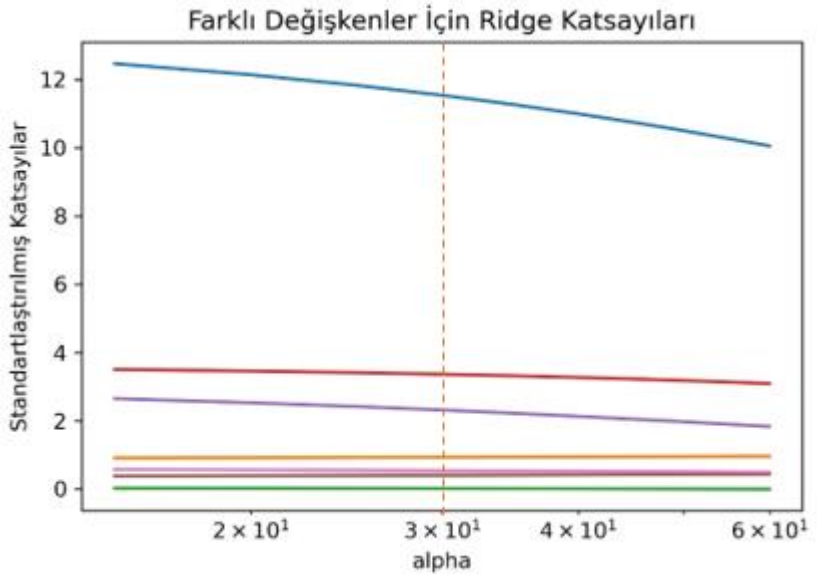
```
#alfa parametrelerinin rolü
alphas = [5,10,15,20,25,30,32,34,36,38,40]
ridge = Ridge()
coef = []

for a in alphas:
    ridge.set_params(alpha=a)
    ridge.fit(X_train, y_train)
    coef.append(ridge.coef_)

ax = plt.gca()

ax.plot(alphas, coef)
ax.set_xscale('log')
plt.grid(10)
plt.axis('tight')
plt.xlabel('alpha')
plt.ylabel('Standartlaştırılmış Katsayılar')
plt.title('Farklı Değişkenler İçin Ridge Katsayıları');
plt.savefig("Farklı Değişkenler İçin Ridge Katsayıları.jpg",dpi=300)
```

Şekil 4.31. Ridge regresyonda değişkenlerin katsayılarını gösteren ve grafikleyen kod



Şekil 4.32. Ridge regresyonda değişkenlerin katsayıları grafiği

Tablo 4.7. Ridge Regresyon Değişken Önem Katsayıları Tablosu

Değişken (Özellik)	**Katsayı**
Cinsiyet	11,534
Irk	0,931
Ebeveyn_Egitim	0,011
Ogle_Yemegi	3,361
Kursta_Test_Hazirligi	2,313
Okuma_Notu	0,400
Yazma_Notu	0,540

Tablo 4.7.'de görüldüğü gibi optimum alpha değerinde Ridge regresyona göre "Matematik_Notu"nu Lasso regresyonda olduğu gibi en çok "Cinsiyet" değişkeninin etkilediği görülmektedir. Yine "Ogle_Yemegi" ve "Kursta_Test_Hazirligi" faktörlerinin de "Matematik_Notu" üzerinde etkisi olduğu görülmektedir. Ancak Lasso regresyonun aksine, Ridge regresyonun bir özelliği olarak değişkenlerin etkileri sıfırlanamamaktadır. Bu özellik, Şekil 4.32'de Ridge regresyonda değişken katsayılaları grafiğinde görülebilmektedir. Bu nedenle "Ebeveyn_Egitimi"nin siteme etkisi sıfıra çok yakın olmasına rağmen Ridge regresyon değişkenin sistemden çıkarılmasına izin vermemektedir.

Ridge regresyon modelinin doğruluğunu ve tahmin performansını görmek için eğitim seti ile test seti üzerinde R-kare ve test seti üzerinde ortalama karesel hata (Mean Squared Error, MSE) hesaplanarak sonuçlar "Tablo 4.8"de gösterilmiştir.

```
ridgecv = RidgeCV(alphas = 30, scoring = 'neg_mean_squared_error')
ridgecv.fit(X_train, y_train)

ridge = Ridge(alpha = ridgecv.alpha_)
ridge.fit(X_train, y_train)

print('Ridge Regression:')
print("Alpha =", ridgecv.alpha_)
print("MSE =", mean_squared_error(y_test, ridge.predict(X_test)))
print("Rİdge Regression Eğitim Doğruluk Sonucu = ",ridge.score(X_train,y_train))
print("Ridge Regression Test Doğruluk Sonucu = ",ridge.score(X_test, y_test))
```

Şekil 4.33. Ridge regresyon sonuç yazdırma kodu

Tablo 4.8. Ridge Regresyon Modeli Performans Sonuçları

Ölçülen Değer	**Performans Sonucu**
Ridge Regresyon Eğitim Seti Üzerinde R-kare	% 86,8
Ridge Regresyon Test Seti Üzerinde R-kare	% 84,6
Ridge Regresyon Ortalama Karesel Hata	30.25

4.4.4. Uygulanan Makine Öğrenmesi Modellerin Karşılaştırılması

Bu çalışmada kullanılan denetimli makine öğrenmesi modellerinin performans raporları ve Lasso ve Ridge regresyonda kullanılan optimum alpha değerleri Tablo 4.9'da karşılaştırmalı olarak gösterilmektedir.

Tablo 4.9. Çalışmada uygulanan modellerin performans tablosu

	Eğitim Seti Üzerinde R-kare	**Test Seti Üzerinde R-kare**	**Ortalama Karesel Hata**	**Optimum Alpha**
Lineer Regresyon	0,872	0,839	31,6	
Lasso Regresyon	0,87	0,843	30,85	0.188
Ridge Regresyon	0,868	0,846	30,25	30

Tablo 4.9'da Lineer Regresyon dan sonra uygulanan ceza alpha uygulamaları ile çalışan Ridge ve Lasso regresyonları sonucunda, test verileri üzerindeki doğruluk oranında bir artış sağlandığı ve ortalama karesel hata da bir azalma olduğu görülmektedir.

Tablo 4.10. Lasso ve Ridge modellerinde değişkenlerin modele katkıları

Değişken (Özellik)	**Lasso Regresyon Katsayıları**	**Ridge Regresyon Katsayıları**
Cinsiyet	12,639	11,534
Irk	0,786	0,931
Ebeveyn_Egitim	0,000	0,011
Ogle_Yemegi	2,981	3,361
Kursta_Test_Hazirligi	2,033	2,313
Okuma_Notu	0,390	0,400
Yazma_Notu	0,560	0,540

Ayrıca Tablo 4.10'da görüldüğü gibi Lasso Regresyon modeline göre "Ebeveyn_Eğitimi" değişkeninin modele hiçbir etkisi olmadığı saptanmıştır. Ancak Ridge regresyonda yapısı gereği modelin değişken katsayısını sıfırlama potansiyeli olmadığından sisteme 0,011 oranında bir katkısı gözlemlenmiştir.

SONUÇ VE ÖNERİLER

Tablo 4.9'dan elde edilen veriler doğrultusunda eğitim sırasında Lasso yaklaşımı Ridge yaklaşımına göre daha baskınken test noktasında Ridge yaklaşımı daha etkili olmaya başlamaktadır. Diğer taraftan hata payı açısından bakıldığında Lasso yaklaşımı Ridge yaklaşımına göre daha yüksek hata oranı içermektedir. Fakat Ridge yaklaşımının ayar parametresinin optimizasyonu hata minimizasyon yöntemi ile belirlenirken Lasso yaklaşımı için R^2 parametresi ile optimize edilmiştir. Ridge yaklaşımında hata payı üzerinden gidilmesindeki temel neden ceza teriminin kare şeklinde hata fonksiyonunda yer almasıdır. Bu durum katsayıların baskılanmasını engellemekte ve buna paralel olarak ayar parametresinin yüksek değerlere ulaşmasına neden olmaktadır. Diğer taftan Lasso yaklaşımında ise hata fonksiyonunda ayar parametresi birinci dereceden bağımlı olduğu için katsayıların bir birlerinin baskılanmasına neden olmaktadır. Sonuç olarak bu veri seti için bakıldığında Lasso yaklaşımının Ridgeye göre daha uygun olduğu açık olarak görülmektedir. Bunun temel nedeni ise Alpha ayar parametresinin Tablo 4.9'dan anlaşılacağı üzere daha düşük değerde anlamlı R2 değerine ulaştığı açık olarak görülmektedir. Sonuç olarak, Ridge ve Lasso regresyon modellerinin benzer performans gösterdiği ancak belirli durumlarda bir modelin diğerine göre daha iyi sonuçlar elde ettiği gözlemlenmiştir. Model seçiminin, uygulamanın gereksinimlerine ve veri setinin özelliklerine bağlı olarak değerlendirilmesi gerektiği bu çalışmada net olarak görülmektedir.

Tablo 4.11. Lasso ve Ridge değişken katsayıları arasındaki yüzdelik değişim

Değişken (Özellik)	Lasso Regresyon Katsayıları	Ridge Regresyon Katsayıları	Değişim
Cinsiyet	12,639	11,534	-9%
Irk	0,786	0,931	18%
Ebeveyn_Egitim	0,000	0,011	Tanımsız
Ogle_Yemegi	2,981	3,361	13%
Kursta_Test_Hazirligi	2,033	2,313	14%
Okuma_Notu	0,390	0,400	2%
Yazma_Notu	0,560	0,540	-4%

Tablo 4.11'de değişken katsayılarının Ridge regresyon sonuçlarının Lasso regresyon sonuçlarına göre yüzdelik değişimi hesaplanmıştır. Bu tabloda görüldüğü gibi "Irk", "Ogle_Yemegi", "Kursta_Test_Hazirligi" ve "Okuma_Notu" değişkenlerinin modele etkisini gösteren katsayılarda pozitif yönde yüzdelik bir değişim gözlemlernmiş yani bu değişkenlerin modele katkısı Ridge Regresyonu'nda daha fazla olduğu hesaplanmıştır. Bunun tersine, "Cinsiyet" ve "Yazma_Notu"'nda negatif yönde bir değişim meydana gelmiş ve bu değişkenlerin modele katkısı Lasso Regresyonunda

daha fazla olmuştur. “Ebeveyn_Egitim” değişkenin modele katkısı ise Lasso Regresyonunda sıfırlandığından Lasso ve Ridge arasındaki yüzdelik değişim hesaplanamamıştır. Ancak bu değişkenin çok az da olsa Ridge Regresyonda bir katkısı hesaplanmıştır.

KAYNAKLAR

Hoerl A.E. Kennard, R.W. "Ridge Regression: Applications to Non-Orthogonal Problems", *Technometrics*, 12, 1, 69-82.1970a.

Hoerl, AE. Kennard, R W. "Ridge Regression: Biased Estimation for Non-Orthogonal Problems", *Technometrics, 12*, 1, 55-67. 1970b.

Arzu, A.R.I. ve Onder, H. "Farkli Veri Yapilarinda Kullanilabilecek Regresyon Yöntemleri". *Anadolu Tarım Bilimleri Dergisi,* 28(3), 168-174. 2013.

Baker, K. "Singular Value Decomposition Tutorial". *The Ohio State University, 24,* 511. 2005.

Borgelt, C. ve Kruse, R. "Induction of Association Rules: Apriori implementation". *In Compstat: Proceedings in Computational Statistics*, 395-400. 2002.

Bounsaythip, C. ve Rinta-Runsala, E. "Overview of Data Mining For Customer Behavior Modeling. *VTT Information Technology Research Report, Version, 1,* 1-53. 2001.

Bunea, F. She, Y. Ombao, H. Gongvatana, A. Devlin, K. ve Cohen, R. (2011). Penalized least squares regression methods and applications to neuroimaging. Neuroimage, 55(4), 1519-1527.

Cortes, C. ve Vapnik, V. "Support-Vector Networks". *Machine learning, 20*(3), 273-297. 1995.

Cover, T. ve Hart, P. "Nearest Neighbor Pattern Classification". *IEEE Transactions on Information Theory, 13*(1), 21-27. 1967.

Dinçer, Ş.E. Veri Madenciliğinde K-Means Algoritması ve Tıp Alanında Uygulanması. Yayımlanmamış Yüksek Lisans Tezi, Kocaeli Üniversitesi, 2006.

Dorugade, A.V. Kashid, D.N. "Alternative Method For Choosing Ridge Parameter For Regression. International". *Journal of Applied Mathematical Sciences 4* (9), 447–456. 2010.

Feigelson, E.D. ve Babu, G.J. "Linear Regression In Astronomy". *II. Astrophysical Journal, Part 1 (ISSN 0004-637X), 397,* 1, 55-67, 397, 55-67. 1992.

Friedman, J.H. ve Tukey, J.W. "A Projection Pursuit Algorithm For Exploratory Data Analysis". *IEEE Transactions on Computers*, 100(9), 881-890. 1974.

Golub, G.H. ve Heath, Wahba, M.G. "Generalized Cross-Validation As A Method For Choosing A Good Ridge Parameter". *Technometrics, 21*,2, 215-223. 1979.

Hocking, R.R. Speed, F.M. ve Lynn, M.J. "A Class of Biased Estimators In Linear Regression". *Technometrics 18* (4), 425–437. 1976.

Hoerl, A.E. Kennard, R.W. ve Baldwin, K.F. "Ridge Regression: Some Simulations". *Communications in Statistics 4*, 105–123. 1975.

James, G. Witten, D. Hastie, T. ve Tibshirani, R. (2013). An İntroduction to Statistical Learning. New York. 2013.

Johnson, S.C. "Hierarchical Clustering Schemes". *Psychometrika, 32*(3), 241-254. 1967.

Jolliffe, I. "Principal Component Analysis". *Principal Component Analysis for Special Types of Data*, 338-339. 2002.

Kohonen, T. "The Self-Organizing Map". *Neurocomputing, 21*(1-3), 1-6. 1998.

Kostakis, A. Magdalinos, T. ve Stamatogiannis, M.P. "Robust Econometric İnference For Stock Return Predictability". *Rev. Financ. Stud. 28* (5), 1506–1553. 2014.

Lawless, J.F. Wang, P. 1976. A simulation study of ridge and other regression estimators. Communications in Statistics – Theory and Methods 14, 1589–1604.

Lee, J.H. "Predictive Quantile Regression With Persistent Covariates: *IVX-QR Approach". J. Econometrics 192* (1), 105–118. 2016.

Liu, Y. ve Guan, Y. "Fp-Growth Algorithm For Application In Research Of Market Basket Analysis". In 2008 IEEE International Conference on Computational Cybernetics, 269-272. 2008.

Marquardt, D.W. ve Snee, R.D. "Ridge Regression İn Practice". *The American Statistician, 29*(1), 3-20. 1975.

McDonald, G.C. Galarneau, D.I. " A Monte Carlo Evaluation Ofsome Ridge-Type Estimators". *Journal of the American Statistical Association 70* (350), 407–412. 1975.

Pasha, G.R. Shah, M.A. "Application of Ridge Regression to Multicollinear Data". *Journal of Research, 15* (1), 97–106. 2004.

R. Frisch, Statistical Confluence Analysis By Means of Complete Regression Systems, Norway, Institute of Economics Oslo. 1934.

Rabiner, L.R. "A tutorial on Hidden Markov Models And Selected Applications In Speech Recognition". *Proceedings of the IEEE, 77*(2), 257-286. 1989.

Rumelhart, D.E. Hinton, G.E. ve Williams, R.J. Learning Representations By Back-Propagating Errors. *Nature, 323*(6088), 533–536. 1986.

Krishna Meghana, S. Predicting Math Score-Linear (88%),Ridge, Lasso Reg. www.kaggle.com. 2023.

Schneider, A. Hommel, G. ve Blettner, M. "Linear Regression Analysis: Part 14 of A Series on Evaluation of Scientific Publications". *Deutsches Ärzteblatt International, 107*(44), 776. 2010.

Su, X. Yan, X. ve Tsai, C.L. "Linear Regression". *Wiley Interdisciplinary Reviews: Computational Statistics, 4*(3), 275-294. 2012.

Tibshirani, R. "Regression Shrinkage and Selection Via the Lasso. *J. R. Stat. Soc. Ser. B Stat. Methodol. 58* (1), 267–288. 1996.

Wiley, J. "Applied Linear Regression". *United States of America, 83*. 2005.

Zou, H. "The Adaptive Lasso and İts Oracle Properties". *J. Amer. Statist. Assoc. 101* (476), 1418–1429. 2006.

EKLER

EK-1 Python Kodları

Çalışmada kullanılan Python kodları aşağıda listelenmiştir.

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import plotly.express as px
import warnings
warnings.filterwarnings('ignore')
data = pd.read_csv("C:\\Users\\ERKA\\Desktop\\Master\\Tez\\sinav\\StudentsPerformance.csv")
data.head(50)
data.shape
data.info()
data.rename(columns={'gender':'Cinsiyet','race/ethnicity': 'Irk', 'parental level of education':
'Ebeveyn_Egitim','lunch':'Ogle_Yemegi','test preparation course':'Kursta_Test_Hazirligi','math
score':'Matematik_Notu', 'reading score':'Okuma_Notu','writing score':'Yazma_Notu'},inplace=True)
df=data
df.head()
df.info()
df['Cinsiyet'].value_counts()
plt.figure(figsize=(10,6))
sns.countplot(df["Cinsiyet"])
plt.title("Öğrencilerin Cinsiyeti",fontsize=20)
plt.xlabel("Cinsiyet",fontsize=15)
plt.ylabel("Kişi Sayısı",fontsize=15)
plt.savefig("cinsiyet_grafik.jpg",dpi=300)
plt.show()
df['Irk'].value_counts()
plt.figure(figsize=(10,6))
sns.countplot(df["Irk"])
plt.title("Öğrencilerin Irk / Etnik Grupları",fontsize=20)
plt.xlabel("Irk / Etnik Grup",fontsize=15)
plt.ylabel("Kişi Sayısı",fontsize=15)
plt.savefig("ırk_grafik.jpg",dpi=300)
plt.show()
df['Ebeveyn_Egitim'].value_counts()
plt.figure(figsize=(10,6))
sns.countplot(df["Ebeveyn_Egitim"])
plt.title("Öğrencilerin Ebeveyn Eğitim Durumları",fontsize=20)
plt.xlabel("Ebeveyn_Egitim",fontsize=15)
plt.ylabel("Kişi Sayısı",fontsize=15)
plt.savefig("Ebeveyn_Egitim_grafik.jpg",dpi=300)
plt.show()
df['Ogle_Yemegi'].value_counts()
plt.figure(figsize=(6,4))
sns.countplot(df["Ogle_Yemegi"])
plt.title("Öğrencilerin Öğle Yemeği Seçimleri",fontsize=15)
plt.xlabel("Öğle Yemeği",fontsize=15)
plt.ylabel("Kişi Sayısı",fontsize=15)
```

```
plt.savefig("Ogle_Yemegi.jpg",dpi=300)
plt.show()
df['Kursta_Test_Hazirligi'].value_counts()
plt.figure(figsize=(6,4))
sns.countplot(df["Kursta_Test_Hazirligi"])
plt.title("Öğrencilerin Test Hazırlığı",fontsize=15)
plt.xlabel("Test Hazırlığı",fontsize=15)
plt.ylabel("Kişi Sayısı",fontsize=15)
plt.savefig("Kursta_Test_Hazirligi.jpg",dpi=300)
plt.show()
df['Matematik_Notu'].value_counts()
df['Okuma_Notu'].value_counts()
df['Yazma_Notu'].value_counts()
from sklearn import preprocessing
cevir=preprocessing.LabelEncoder()
df[('Cinsiyet')]=cevir.fit_transform(df[('Cinsiyet')])
df[('Irk')]=cevir.fit_transform(df[('Irk')])
df[('Ebeveyn_Egitim')]=cevir.fit_transform(df[('Ebeveyn_Egitim')])
df[('Ogle_Yemegi')]=cevir.fit_transform(df[('Ogle_Yemegi')])
df[('Kursta_Test_Hazirligi')]=cevir.fit_transform(df[('Kursta_Test_Hazirligi')])
df
X=df[['Cinsiyet','Irk','Ebeveyn_Egitim','Ogle_Yemegi','Kursta_Test_Hazirligi','Okuma_Notu','Yazma_
Notu']]
Y=df['Matematik_Notu']
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, Y, test_size=0.2, random_state=101)
print(X_train.shape)
print(X_test.shape)
print(y_train.shape)
print(y_test.shape)
from sklearn.linear_model import LinearRegression
model = LinearRegression()
model.fit(X_train,y_train)
def calculate_residuals(model, features, label):
    predictions = model.predict(features)
    df_results = pd.DataFrame({'Actual': label, 'Predicted': predictions})
    df_results['Residuals'] = abs(df_results['Actual']) - abs(df_results['Predicted'])
    return df_results
def linear_assumption(model, features, label):
    df_results = calculate_residuals(model, features, label)
    sns.lmplot(x='Actual', y='Predicted', data=df_results, fit_reg=False, size=7)
    line_coords = np.arange(df_results.min().min(), df_results.max().max())
    plt.plot(line_coords, line_coords, color='darkorange', linestyle='--')
    plt.title('Actual vs. Predicted Values')
    plt.show()
linear_assumption(model,X_train,y_train)
def normal_errors_assumption(model, features, label, p_value_thresh=0.05):
    from statsmodels.stats.diagnostic import normal_ad
    df_results = calculate_residuals(model, features, label)
    print('Using the Anderson-Darling test for Normal Distribution')
    p_value = normal_ad(df_results['Residuals'])[1]
    print('p-value from the test: ', p_value)
    if p_value < p_value_thresh:
        print('Since p-value less than 0.05, Residuals are not normally distributed.')
    else:
```

```
        print('Since p-value greater than 0.05, Residuals are normally distributed.')

    plt.subplots(figsize=(12, 6))
    plt.title('Distribution of Residuals')
    sns.distplot(df_results['Residuals'])
    plt.show()

normal_errors_assumption(model,X_train,y_train)
def multicollinearity_assumption(model, features, label, feature_names=None):

    plt.figure(figsize = (10,8))
    sns.heatmap(pd.DataFrame(features, columns=feature_names).corr(), annot=True)
    plt.title('Correlation of Variables')
    plt.show()

multicollinearity_assumption(model,X_train,y_train)
from statsmodels.stats.stattools import durbin_watson #Using Durbin-Watson test
df_results = calculate_residuals(model,X_train,y_train)
durbin_watson(df_results['Residuals'])
df_results = calculate_residuals(model,X_train,y_train)
plt.subplots(figsize=(12, 6))
ax = plt.subplot(111)
plt.scatter(x=df_results.index, y=df_results.Residuals, alpha=0.5)
plt.plot(np.repeat(0, df_results.index.max()), color='darkorange', linestyle='--')
ax.spines['right'].set_visible(False)
ax.spines['top'].set_visible(False)
plt.title('Residuals')
plt.show()
from sklearn.linear_model import LinearRegression
model = LinearRegression()
model.fit(X_train,y_train)
import statsmodels.api as sm
X_train_Sm= sm.add_constant(X_train)
X_train_Sm= sm.add_constant(X_train)
ls=sm.OLS(y_train,X_train_Sm).fit()
print(ls.summary())
plt.savefig("OLS_Regresyon_Sonuçları.jpg",dpi=300)
print("Linear Regression score : ",model.score(X_train,y_train))
print("Linear Regression score : ",model.score(X_test, y_test))
from sklearn.metrics import mean_squared_error
predictions = model.predict(X_test)
print("Mean Square Error: ",mean_squared_error(y_test,predictions))
from numpy import arange
from sklearn.linear_model import Ridge
from sklearn.linear_model import Lasso
from sklearn.metrics import mean_squared_error
scores=[]
for i in arange(-2,2,0.002):
    ls = Lasso(alpha=10**i)
    print(10**i)
    ls.fit(X_train, y_train)
    y_pred = ls.predict(X_test)
    scores.append(ls.score(X_test, y_test))
plt.plot(10**arange(-2,2,0.002),scores)
plt.xlabel('Alpha')
```

```
plt.ylabel('Performans Skorları')
plt.title('Lasso Regresyonu Alpha Değerlerine Karşılık Performans Grafiği')
plt.xscale('log')
plt.savefig("Lasso_Lamda_Sabit_Grafigi.jpg",dpi=300)
from sklearn.linear_model import LassoCV
model = LassoCV(cv=10, random_state=0, max_iter=10000)
model.fit(X_train, y_train)
model.alpha_
lasso_best = Lasso(alpha=model.alpha_)
lasso_best.fit(X_train, y_train)
alphas = np.linspace(0.01,500,100)
lasso = Lasso(max_iter=10000)
coefs = []
for a in alphas:
    lasso.set_params(alpha=a)
    lasso.fit(X_train, y_train)
    coefs.append(lasso.coef_)

ax = plt.gca()
ax.plot(alphas, coefs)
ax.set_xscale('log')
plt.axis('tight')
plt.xlabel('Alpha')
plt.ylabel('Standartlaştırılmış Katsayılar')
plt.title('Farklı Değişkenler İçin Lasso Katsayıları');
plt.savefig("Lasso_Katsayılar_Grafigi.jpg",dpi=300)
print(list(zip(lasso_best.coef_, X)))
print('R squared training set', round(lasso_best.score(X_train, y_train)*100, 2))
print('R squared test set', round(lasso_best.score(X_test, y_test)*100, 2))
print('Mean Square Error', round(mean_squared_error(y_test, lasso_best.predict(X_test)), 2))

alphas = [0.01, 0.1, 1,5,10,15,20,25,30,50,60,70,80,90,100]
sonuc=[]
for i in alphas:
    ls = Ridge(alpha=i).fit(X_train, y_train)
    score = ls.score(X_test, y_test)
    pred_y = ls.predict(X_test)
    mse = mean_squared_error(y_test, pred_y)
    print("Alpha:{0:.6f}, R2:{1:.3f}, MSE:{2:.2f}, RMSE:{3:.2f}"
    .format(i, score, mse, np.sqrt(mse)))
    sonuc.append(mean_squared_error(pred_y, y_test))

plt.plot( alphas,sonuc)
plt.grid("30")
plt.xlabel('Alpha')
plt.ylabel('Hata Skorları')
plt.title('Ridge Regresyonu Alpha Değerlerine Karşılık Hata Grafiği')
plt.savefig("Ridge Regresyonu Alpha Değerlerine Karşılık Hata Grafiği.jpg",dpi=300)

plt.plot( alphas,sonuc)
plt.grid("on")
plt.xlabel('Alpha')
plt.ylabel('Sistem Performansı')
plt.title('Ridge Regresyonu Alpha Değerlerine Performans Grafiği')
plt.savefig("Ridge Regresyonu Alpha Değerlerine Performans Grafiği.jpg",dpi=300)
```

```
alphas = [15,20,25,30,35,40,45,50,60]
ridge = Ridge()
coef = []

for a in alphas:
    ridge.set_params(alpha=a)
    ridge.fit(X_train, y_train)
    coef.append(ridge.coef_)

ax = plt.gca()
ax.plot(alphas, coef)
ax.set_xscale('log')
plt.grid('on')
plt.axis('tight')
plt.xlabel('alpha')
plt.ylabel('Standartlaştırılmış Katsayılar')
plt.title('Farklı Değişkenler İçin Ridge Katsayıları');
plt.savefig("Farklı Değişkenler İçin Ridge Katsayıları.jpg",dpi=300)

ridge.set_params(alpha=30)
sonuc=ridge.fit(X_train, y_train)
print(list(zip(sonuc.coef_, X)))

from sklearn.linear_model import RidgeCV
ridgecv = RidgeCV(alphas = 30, scoring = 'neg_mean_squared_error')
ridgecv.fit(X_train, y_train)

ridge = Ridge(alpha = ridgecv.alpha_)
ridge.fit(X_train, y_train)

print('Ridge Regression:')
print("Alpha =", ridgecv.alpha_)
print("MSE =", mean_squared_error(y_test, ridge.predict(X_test)))
print("Rİdge Regression Eğitim Doğruluk Sonucu = ",ridge.score(X_train,y_train))
print("Ridge Regression Test Doğruluk Sonucu = ",ridge.score(X_test, y_test))
```

BENZERLİK RAPORU ÖZET SAYFASI

BAĞLANTISALLIK PROBLEMİNİN CEZALI REGRESYON YÖNTEMLERİ İLE GİDERİLMESİ

ORİJİNALLİK RAPORU

%6	%5	%1	%3
BENZERLİK ENDEKSİ	İNTERNET KAYNAKLARI	YAYINLAR	ÖĞRENCİ ÖDEVLERİ

BİRİNCİL KAYNAKLAR

1	acikbilim.yok.gov.tr İnternet Kaynağı	%1
2	www.veribilimiokulu.com İnternet Kaynağı	%1
3	Submitted to Mersin Ãniversitesi Öğrenci Ödevi	%1
4	Submitted to The Scientific & Technological Research Council of Turkey (TUBITAK) Öğrenci Ödevi	<%1
5	Submitted to Selçuk Üniversitesi Öğrenci Ödevi	<%1
6	openaccess.hacettepe.edu.tr:8080 İnternet Kaynağı	<%1
7	ARI, Arzu and ÖNDER, Hasan. "Farklı veri yapılarında kullanılabilecek regresyon yöntemleri", Ondokuz Mayıs Üniversitesi / University of Ondokuz Mayıs, 2013. Yayın	<%1

ÖZGEÇMİŞ

Adı ve Soyadı **:** Emel CİĞER

Doğum Tarihi **:**

E-mail **:**

Öğrenim Durumu **:**

Derece	**Anabilim Dalı/ Bölüm/Program**	**Üniversite**	**Yıl**
Ön Lisans	Bilgisayar Programcılığı	Mersin Üniversitesi	1998
Lisans	İşletme	Anadolu Üniversitesi	2004
Yüksek Lisans	İşletme Bilgi Yönetimi Anabilim Dalı	Mersin Üniversitesi	2023

Görevler **:**

Görev Unvanı	**Görev Yeri**	**Yıl**
Bilgi İşlem Sorumlusu	İstanbul Hava Yolları A.Ş.	1998-2000
Ofis Yöneticisi	Dominet A.Ş.	2000-2002
GPRS / Kurumsal / VİP Müşteri Hizmetleri	Vodafone Telekomünikasyon A.Ş	2002-2003
Öğrenci İşleri Daire Başkanlığı Personeli	Mersin Üniversitesi / ÖİDB	2003-2005
Genel Müdür	WeNET İnternet Hizmetleri İletişim ve Teknoloji San. Tic. Ltd. Şti.	2014-2017

ESERLER

- Mehmet Saltı, Emel Ciğer, Evrim Ersin Kangal ve Bilgin Zengin, *"Data-driven predictive modeling of Hubble parameter",* Physica Scripta 97 (2022) 085011.