



T. C.

**BURSA ULUDAĞ ÜNİVERSİTESİ
SOSYAL BİLİMLER ENSTİTÜSÜ
EKONOMETRİ ANABİLİM DALI
İSTATİSTİK BİLİM DALI**

**TÜRKİYE'DEKİ İLLERİN GÖÇ GÖSTERGELERİNİN
PYTHON KULLANILARAK K-ORTALAMALAR
KÜMELEME YÖNTEMİ İLE ARAŞTIRILMASI**

(YÜKSEK LİSANS TEZİ)

Sedat KARAKAYA

BURSA – 2021



T.C.

BURSA ULUDAĞ ÜNİVERSİTESİ

SOSYAL BİLİMLER ENSTİTÜSÜ

EKONOMETRİ ANABİLİM DALI

İSTATİSTİK BİLİM DALI

**TÜRKİYE’DEKİ İLLERİN GÖÇ GÖSTERGELERİNİN
PYTHON KULLANILARAK K-ORTALAMALAR
KÜMELEME YÖNTEMİ İLE ARAŞTIRILMASI**

(YÜKSEK LİSANS TEZİ)

Sedat KARAKAYA

Danışman:

Prof. Dr. Nuran BAYRAM ARLI

BURSA – 2021

Yemin Metni

Yüksek Lisans tezi olarak sunduğum “Türkiye’deki İllerin Göç Göstergelerinin Python Kullanılarak K-ortalamlar Kümeleme Yöntemi İle Araştırılması” başlıklı çalışmanın bilimsel araştırma, yazma ve etik kurallarına uygun olarak tarafımdan yazıldığına ve tezde yapılan bütün alıntılarının kaynaklarının usulüne uygun olarak gösterildiğine, tezimde intihal ürünü cümle veya paragraflar bulunmadığına şerefim üzerine yemin ederim.

.../.../.....

İmza

Adı Soyadı: Sedat KARAKAYA

Öğrenci No: 701817010

Anabilim Dalı: Ekonometri

Programı: Tezli Yüksek Lisans Programı

Statüsü: Yüksek Lisans

ÖZET

Yazar Adı ve Soyadı : Sedat Karakaya
Üniversite : Bursa Uludağ Üniversitesi
Enstitü : Sosyal Bilimler Enstitüsü
Anabilim Dalı : Ekonometri Anabilim Dalı
Bilim Dalı : İstatistik Bilim Dalı
Tezin Niteliği : Yüksek Lisans Tezi
Sayfa Sayısı : xiii + 95
Mezuniyet Tarihi :
Tez Danışmanı : Prof. Dr. Nuran Bayram Arlı

TÜRKİYE'DEKİ İLLERİN GÖÇ GÖSTERGELERİNİN PYTHON KULLANILARAK K-ORTALAMALAR KÜMELEME YÖNTEMİ İLE ARAŞTIRILMASI

Göç, ekonomik, siyasal, sosyal vb. nedenlerle bireylerin veya toplumların mevcut yaşadıkları yeri terk edip ikamet amacıyla başka yerlere gitmesi olarak tanımlanabilir. Kümeleme analizi ise hem makine öğrenmesinde hem de veri madenciliğinde yaygın olarak kullanılan bir tekniktir. Kümeleme işleminde esas olan kümelerin kendi içinde benzer, kümeler arası ise benzemez yapıda olacak şekilde ayırma işlemini yapmaktır. Bu çalışmanın temel amacı, Türkiye'deki 81 ilin göç istatistikleri dikkate alınarak illerin Kümeleme Analizi ile kümelenebilmesinin ortaya konulmasıdır. Bunun için Türkiye'nin iller bazında 2008-2020 yılları arasındaki TÜİK tarafından yayınlanan göç göstergeleri kullanılmıştır. K-ortalamlar algoritması Python yazılım dili kullanılarak uygulanmıştır. Değişkenler olarak; alınan göç, verilen göç, net göç ve net göç hızı verileri temel alınmıştır. Her bir değişken için ayrı ayrı kümeleme işlemi gerçekleştirilmiştir.

Anahtar Kelimeler: Göç Göstergeleri, Python, K-ortalamlar

ABSTRACT

Name and Surname : Sedat Karakaya
University : Bursa Uludag University
Institution : Social Science Institution
Field : Econometry
Branch : Statistics
Degree Awarded : Master
Page Number : xiii + 95
Degree Date :
Supervisor : Prof. Dr. Nuran Bayram Arlı

INVESTIGATION OF MIGRATION INDICATORS OF PROVINCES IN TURKEY WITH K-MEANS CLUSTERING METHOD USING PYTHON

Migration, economic, political, social, etc. It can be defined as individuals or societies leaving their current place of residence and moving to other places for the purpose of residence. Cluster analysis is a widely used technique in both machine learning and data mining. The main thing in the clustering process is to separate the clusters in such a way that they have a similar structure within themselves and a dissimilar structure between clusters. The main purpose of this study is to reveal the clustering of provinces with Cluster Analysis, taking into account the migration statistics of 81 provinces in Turkey. For this, migration indicators published by TUIK between the years 2008-2020 on the basis of provinces of Turkey were used. K-means algorithm is implemented using Python programming language. As variables; based on migration received, migration given, net migration and net migration rate data. Clustering was performed separately for each variable.

Keywords: Migration Indicators, Python, K-means

ÖNSÖZ

Yüksek Lisans eğitimim boyunca ve bu tez çalışmamın yürütülmesinde desteğini esirgemeyen, yönlendirme ve bilgilendirmeleriyle bana adeta ışık olan, güler yüzlü çok değerli tez danışmanım Prof. Dr. Nuran BAYRAM ARLI 'ya en kalbi duygularıyla teşekkür ediyorum.

Lisans eğitimim üzerinden 10 yıl geçmiş olmasına rağmen yüksek lisans eğitimine başlamam noktasında beni teşvik eden ve bu süreçte maddi, manevi beni her türlü destekleyen değerli eşim Kevser KARAKAYA' ya çok teşekkür ediyorum.

Son olarak tüm hayatım boyunca beni destekleyen, her zaman yanımda olan, bugünlere gelmemde en büyük pay sahipleri olan sevgili annem ve babama minnettarlığımı sunmayı bir borç bilirim.

Bursa – 2021

Sedat KARAKAYA

İÇİNDEKİLER

	Sayfa
TEZ ONAY SAYFASI.....	ii
YEMİN METNİ.....	iii
ÖZET.....	iv
ABSTRACT.....	v
ÖNSÖZ.....	vi
İÇİNDEKİLER.....	vii
TABLOLAR.....	xi
ŞEKİLLER ve GRAFİKLER.....	xii
KISALTMALAR.....	xiii
GİRİŞ.....	1

BİRİNCİ BÖLÜM

MAKİNE ÖĞRENMESİ

1.1. MAKİNE ÖĞRENMESİ NEDİR?.....	3
1.2. MAKİNE ÖĞRENMESİ ÇEŞİTLERİ.....	3
1.2.1. Denetimli Öğrenme.....	4
1.2.2. Denetimsiz Öğrenme.....	4
1.2.3. Yarı Denetimli Öğrenme.....	4
1.2.4. Pekiştirmeli Öğrenme.....	5
1.3. DENETİMLİ ÖĞRENME YÖNTEMLERİ.....	5
1.3.1. Regresyon Analizi.....	5

1.3.2. Lojistik Regresyon.....	8
1.3.3. Yapay Sinir Ağları.....	9
1.3.4. Bayesyen Modeller.....	10
1.3.5. Karar Ağaçları.....	10
1.3.6. Destek Vektör Makineleri.....	13
1.3.7. K-En Yakın Komşu (k-NN).....	14
1.4. DENETİMSİZ ÖĞRENME YÖNTEMLERİ.....	16
1.4.1. Kümeleme Analizi.....	16
1.4.1.1. Hiyerarşik Olmayan Kümeleme Yöntemleri.....	16
1.4.1.2. Hiyerarşik Kümeleme Yöntemleri.....	17
1.4.2. Temel Bileşenler Analizi.....	17
1.5. MODEL DOĞRULAMA YÖNTEMLERİ.....	17
1.5.1. Hold Out.....	18
1.5.2. K-Katlı Çapraz Doğrulama.....	18
1.5.3. Leave One Out.....	19
1.5.4. Yeniden Örnekleme(Bootstrap).....	20
1.6. MODEL BAŞARISINI DEĞERLENDİRME YÖNTEMLERİ....	20
1.6.1. Karışıklık Matrisi.....	20
1.6.2. Roc Eğrisi.....	22

İKİNCİ BÖLÜM

KÜMELEME ANALİZİ

2.1. KÜMELEME ANALİZİ.....	24
2.1.1. Kümeleme Analizi Varsayımları.....	26

2.2. VERİLERİN STANDARTLAŞTIRILMASI.....	27
2.3. UZAKLIK YA DA BENZERLİK MATRİSLERİNİN OLUŞTURULMASI.....	28
2.4. UZAKLIK VE BENZERLİK ÖLÇÜLERİ.....	30
2.4.1. Sayısal Veriler İçin Uzaklık Ve Benzerlik Ölçüleri.....	30
2.4.2. Sıklık Sayıları İçin Uzaklık Ölçüleri.....	33
2.4.3. İkili Veriler İçin Uzaklık Ve Benzerlik Ölçüleri.....	33
2.4.4. Kümeleme Analizi Yöntemleri.....	35
2.4.4.1. Hiyerarşik Kümeleme Yöntemleri.....	36
2.4.4.2. Hiyerarşik Olmayan Kümeleme Yöntemleri.....	38

ÜÇÜNCÜ BÖLÜM

TÜRKİYE'DEKİ İLLERİN GÖÇ GÖSTERGELERİNE GÖRE KÜMELENMESİ

3.1.GÖÇ KAVRAMI ve GÖÇ ÇALIŞMALARI.....	44
3.2. TÜRKİYE'DE İÇ GÖÇ.....	47
3.3. PYTHON PROGRAMLAMA DİLİ.....	56
3.4. TEZİN AMACI ve ÇALIŞMADA KULLANILAN VERİLER....	57
3.5. BULGULAR.....	63
SONUÇ VE ÖNERİLER.....	71

EKLER

EK-1 :KULLANILAN PYTHON KODLARI.....	74
---	-----------

EK-2:2008-2020 TÜRKİYE’NİN İLLER BAZINDA GÖÇ İSTATİSTİKLERİ.....	76
KAYNAKÇA.....	87

TABLolar LİSTESİ

	Sayfa
Tablo 1: İkili Sınıflandırma Karışıklık Matrisi Örneği.....	20
Tablo 2: Veri Tipine Göre Ölçülerin Seçimi.....	29
Tablo 3: İkili Veriler İçin Kontenjans Tablosu.....	34
Tablo 4: 1975-2020 Türkiye Nüfusu ve İç Göç İstatistikleri.....	48
Tablo 5: Net Göç Hızı En Çok Olan İller (2008-2020).....	51
Tablo 6: Net Göç Hızı En Düşük Olan İller (2008-2020).....	54
Tablo 7: Alınan Göç Veri Setindeki Kısaltmalar ve Bazı Temel İstatistikleri.....	58
Tablo 8: Verilen Göç Veri Setindeki Kısaltmalar ve Bazı Temel İstatistikler.....	59
Tablo 9: Net Göç Veri Setindeki Kısaltmalar ve Bazı Temel İstatistikler.....	60
Tablo 10: Net Göç Hızı Veri Setindeki Kısaltmalar ve Bazı Temel İstatistikler.....	61
Tablo 11: İllerin Aldıkları Göç İstatistiklerine Göre Elde Edilen Kümelerin Elemanları.....	65
Tablo 12: İllerin Verdikleri Göç İstatistiklerine Göre Elde Edilen Kümelerin Elemanları.....	66
Tablo 13: İllerin Net Göç İstatistiklerine Göre Elde Edilen Kümelerin Elemanları.....	68
Tablo 14: İllerin Net Göç Hızı İstatistiklerine Göre Elde Edilen Kümelerin Elemanları.....	69

ŞEKİLLER VE GRAFİKLER LİSTESİ

	Sayfa
Şekil 1: Doğrusal Regresyon.....	6
Şekil 2: Polinom Regresyon.....	7
Şekil 3: Lojistik Regresyon Fonksiyonu Eğrisi.....	8
Şekil 4: Yapay Sinir Ağı Örneği.....	9
Şekil 5: Karar Ağacı Örneği.....	11
Şekil 6: Rastgele Orman Örneği.....	12
Şekil 7: Destek Vektör Makineleri Örneği.....	13
Şekil 8: k-NN Örneği.....	15
Şekil 9: Kümeleme Örneği.....	16
Şekil 10: 10 Katlı Çapraz Doğrulama Örneği.....	19
Şekil 11: ROC Eğrisi.....	22
Şekil 12: Kümeleme Öncesi ve Sonrası Verilerin Görünümü.....	25
Şekil 13: İki Gözlem Arası Uzaklık.....	30
Grafik 1: 2020 Yılında En Fazla Göç Alan İller.....	49
Grafik 2: 2020 Yılında En Az Göç Alan İller.....	50
Grafik 3: 2020 Yılında En Fazla Göç Veren İller.....	50
Grafik 4: 2020 Yılında En Az Göç Veren İller.....	51
Grafik 5: Alınan Göç Veri Setine Ait wcss Değerleri.....	64
Grafik 6: Verilen Göç Veri Setine Ait wcss Değerleri.....	66
Grafik 7: Net Göç Veri Setine Ait wcss Değerleri.....	67
Grafik 8: Net Göç Hızı Veri Setine Ait wcss Değerleri.....	69

KISALTMALAR

Kısaltması	Bibliyografik Bilgiler
ag	Alınan Göç
C.	Cilt
DG	Dizi Genişliği
DVM	Destek Vektör Makineleri
FN	False Negative
FP	False Positive
IBM	International Business Machines
KNN	K Nearest Neighbour
ng	Net Göç
ngh	Net Göç Hızı
ROC	Receiving Operating Characteristic
s./ss.	Sayfa/Sayfalar
TÜİK	Türkiye İstatistik Kurumu
TN	True Negative
TP	True Positive
Vb.	Ve Benzeri
vg	Verilen Göç
wcss	Kümeler İçi Kareler Toplamı(Within Clusters Sum Of Square)

GİRİŞ

Göç insanın var olmaya başladığı günden bugüne değin her zaman var olmuş bir sosyal olgudur. İnsanoğlu çok çeşitli sebeplerle mevcut ikamet ettiği yeri terk edip kendisi için daha cazip, çekici, imkânları daha geniş vb. olan yerleşim yerlerine taşınma eğiliminde olmuştur. Bu göç kıtalar arası, ülkeler arası olabileceği gibi iller arası hatta il içinde de bir beldeden diğer beldeye şeklinde de olabilir.

Önüne geçilemeyen bir olgu olan göç her ülkenin kendi içyapısına göre farklılıklar göstermektedir. Türkiye’de her ilin kendine has ekonomik, sosyal, kültürel özelliklerinin de olduğu düşünüldüğünde iç göç kaçınılmaz olmaktadır. Özellikle imkânların fazla olduğu bölgelerde nüfus daha yoğunken, aksine coğrafi özelliklerinden dolayı tarım ve hayvancılık işlerinin geçim kaynağı olduğu istihdam olanaklarının kısıtlı olduğu yerlerde nüfusun daha seyrek olduğu görülmektedir. Yaşanan bu göçler neticesinde hem göçmenler hem de göç edilen yerlerde çeşitli sorunlar baş göstermektedir. Özellikle büyük şehirlerde varoş diye tabir edilen gecekondu mahallelerinin oluşması buna bir örnektir. Öte yandan yine göç veren bölgelerde bazı köylerde hiç insanın kalmamış olması da olumsuz sonuçlardan sadece bir tanesidir.

Göçlerin kendine has sebepleri olduğu gibi aynı zamanda önemli sonuçları da dünya tarihinde yerini almıştır. Örneğin 4. Yüzyılda başlayan Kavimler Göçü neticesinde İlk Çağ kapanmış Orta Çağ başlamıştır.

Türkiye ise jeopolitik konumunun bir sonucu olarak göç yollarının merkezinde yer almıştır. Göç veren, göç alan ve yeri geldiğinde ise bir transit ülke olma özelliğini hep korumuştur. Bunların yanı sıra ülke içindeki göç de 1923-1950 arasında nispeten daha yavaş olmakla beraber bu tarihten sonra çok çeşitli sebeplerden dolayı hızlanmıştır.

Son zamanlarda hem kullanımı hem de popülerliği artan makine öğrenmesi mevcut veriler üzerinde her türlü çalışma, çıkarım, değerlendirme gibi işlemlerin yapılabildiği yapay zekânın bir alt dalıdır. Nispeten daha az verilerdeki işlemlerde klasik

yöntemler yeterli olurken, söz konusu büyük veri olduğunda makine öğrenmesi büyük kolaylıklar sağlamaktadır.

Makine öğrenmesi denize atılan taşın oluşturduğu halkalar misali gün geçtikçe gelişmekte ve büyümektedir. Fen ve sosyal bilimler başta olmak üzere birçok bilim dalında sıklıkla kullanılan bir alandır.

Bu çalışmanın temel amacı, Türkiye'deki 81 ilin göç istatistikleri dikkate alınarak illerin Kümeleme Analizi ile kümelenmesinin ortaya konulmasıdır. Bunu gerçekleştirmek için denetimsiz öğrenme yöntemlerinden biri olan K-ortalamlar algoritması Python yazılım dili kullanılarak uygulanmıştır. Türkiye'deki illerin göç istatistikleri bakımından gruplaşmasının elde edilmesi ileriye yönelik yapılacak çalışmalarda, araştırmacılara ve politika yapıcılara ışık tutacağı düşünülmüştür.

Bu amaç doğrultusunda tez üç temel bölüme ayrılmıştır. Çalışmanın birinci bölümünde makine öğrenmesi, ikinci bölümünde kümeleme analizi, üçüncü bölümünde ise Türkiye'deki illerin göç göstergelerine göre kümelenmesi başlıkları ayrıntılı biçimde ele alınmıştır. Daha sonra ise sonuç ve öneriler kısmı sunulmuştur.

BİRİNCİ BÖLÜM

MAKİNE ÖĞRENMESİ

1.1. MAKİNE ÖĞRENMESİ NEDİR?

Son yıllarda popülerliği artan bir çalışma alanı olan makine öğrenmesi aslında yapay zekânın bir alt dalıdır. Öyle ki geçmişteki verileri kullanarak istatistiksel ve matematiksel yöntemler ile bilişim ve teknolojik gelişmeleri de kullanarak tahmin ve çıkarımlar yapmaya makine öğrenmesi denir. Bunları yaparken ise algoritmaları kullanır. Başka bir deyişle bilgisayarların da insanlar gibi öğrenmesini sağlamaya çalışan bir alan olduğu söylenebilir. Özellikle adı büyük veri ile de anılan bu alan, bu büyük veri içerisindeki potansiyel bilgi ve yapıları ortaya çıkarmakta ve modellemede kullanılır.

1943 yılında Walter Pitts ve Warren McCulloch, A Logical Calculus of Ideas Immanent in Nervous Activity isimli çalışmalarıyla sinir ağlarının matematiksel modelini kurmuşlardır. Ardından 2. Dünya Savaşı'nda matematikçi Alan Turing telgraf şifrelerini kırmış ve makine öğrenmesini öngörmüştür. 1959 yılında Arthur Samuel IBM'de de çalışması vesilesiyle makine öğrenmesi tabirini literatüre kazandırmıştır. O yıllardan bu yıllara kadar makine öğrenmesi her geçen gün hızla büyümekte ve gelişmektedir.

Bilişim, endüstri, bankacılık, finans, siyaset, eğitim vb. birçok alanda makine öğrenmesi algoritmaları kullanılmaktadır. Gelen elektronik postanın gereksiz mi değil mi şeklinde sınıflandırılması, yüz tanıma sistemleri, yılsonu enflasyonunun tahmini, sanayi ürünlerinde iyileştirme, finansal analiz ve tahminleme gibi birkaç örnek verilebilir.

1.2. MAKİNE ÖĞRENMESİ ÇEŞİTLERİ

Temel olarak makine öğrenmesi yöntemleri denetimli(supervised), yarı denetimli (semi-supervised), denetimsiz(unsupervised) ve pekiştirmeli(reinforcement) olmak üzere 4 gruba ayrılabilir.

1.2.1. Denetimli Öğrenme

Denetimli öğrenmede her bir verinin onun içeriğini açıklayan bir etiketi vardır.¹Denetimli öğrenmede verilerin hem girdi hem de çıktı değerleri mevcuttur. Girdi ve çıktı değerleri arasında eşleme yapan bir fonksiyon bulunmaya çalışılır. Daha doğrusu, girdi ve çıktı değerlerinin bulunduğu eğitim verileri iyice analiz edilerek aralarındaki ilişki bulunmaya çalışılır. Daha sonra yeni girilecek değer için çıktı değeri tahmin edilmeye çalışılır.

Esasen buradaki amaç ya da görev regresyon ve sınıflandırmadır. Regresyonda, bağımlı bir değişken ve bağımsız birkaç değişken üzerinden anlamlı tahminler yapılmaya çalışılır. Sınıflandırmada ise makine öğrenmesi ile elde edilen verilerden çıkan sonuçların kategorilere ayrılmaya çalışılır.

1.2.2. Denetimsiz Öğrenme

Denetimsiz öğrenmede, algoritmada kullanılan veri setinde etiket yoktur. Yani girdi verileri bellidir ama çıktı değerleri belli değildir. Algoritma girdilerdeki yapıyı kendi kendine bulmak zorundadır.²Dolayısıyla bu öğrenmede değerlendirme hiç de kolay değildir. Algoritma girdi verilerindeki ilişkileri her yönüyle analiz edip sonuca ulaşmalıdır. Dolayısıyla ne kadar çok veri girişi olursa yapılan analiz bir o kadar güçlü ve doğru olacaktır. K-ortalamalar ve hiyerarşik kümeleme analizi denetimsiz öğrenmeye örnek olarak verilebilir.

1.2.3. Yarı Denetimli Öğrenme

Yarı denetimli öğrenme denetimli öğrenmeye benzerdir. Veri setinde hem etiketli hem de etiketsiz veriler bulunabilir. Kullanılacak olan algoritma bu etiketli veriler arasındaki ilişkileri, çıkarımları anlamak suretiyle etiketlenmemiş verileri etiketlemeye çalışacaktır. Dolayısıyla hem denetimli hem de denetimsiz öğrenmedir denilebilir. Verilerin çoğu etiketsiz olup, az bir kısmı etiketlidir.

¹ Oswald Campaseto, Python 3 For Machine Learning, Dulles:Mercury Learning and Information, 2020, s.143

² Chris Smith, Decision Trees and Random Forests: A Visual Introduction for Beginners, Blue Windmill Media,2017, s.66

1.2.4. Pekiştirmeli Öğrenme

Pekiştirmeli öğrenmede, algoritma belli bir hedefe ulaşmaya çalışır. Bunu da deneme yanılma yoluyla gerçekleştirir. Yapılan her doğru için ödül ve her yanlış için bir ceza alan algoritma en yüksek ödülü alabilmek için sürekli deneme yanılma yolunu kullanır. Pekiştirmeli Öğrenme, Markov karar süreci denilen bir model kullanılmaktadır.³ Yani önce durumu algılar, sonra eyleme geçer ve en sonunda hedefe ulaşır.

1.3. DENETİMLİ ÖĞRENME YÖNTEMLERİ

1.3.1. Regresyon Analizi

Regresyon analizi, ekonometrik çalışmalarda en çok kullanılan araçlardan biridir(Tarı,2010:15). Değişkenlerin arasındaki ilişkinin fonksiyonel şekli ile ilgilenildiğinde bu analiz kullanılmaktadır. Bağımlı veya açıklanan değişken Y ile bağımsız veya açıklayıcı değişken(ler) X arasındaki ilişkiyi ifade etme ve bu ilişkinin derecesinin hesaplanması ile alakalı süreçtir. Bağımsız değişkenlerin, bağımlı değişken üzerinde ne kadar etkisinin olduğunun araştırılması, bağımsız değişken verileri ile bağımlı değişkeninin ortalama değerini tahmin etmek ve gelecekteki alacağı değeri tahmin etmek, bu analizin başlıca amaçları arasında sayılabilir.

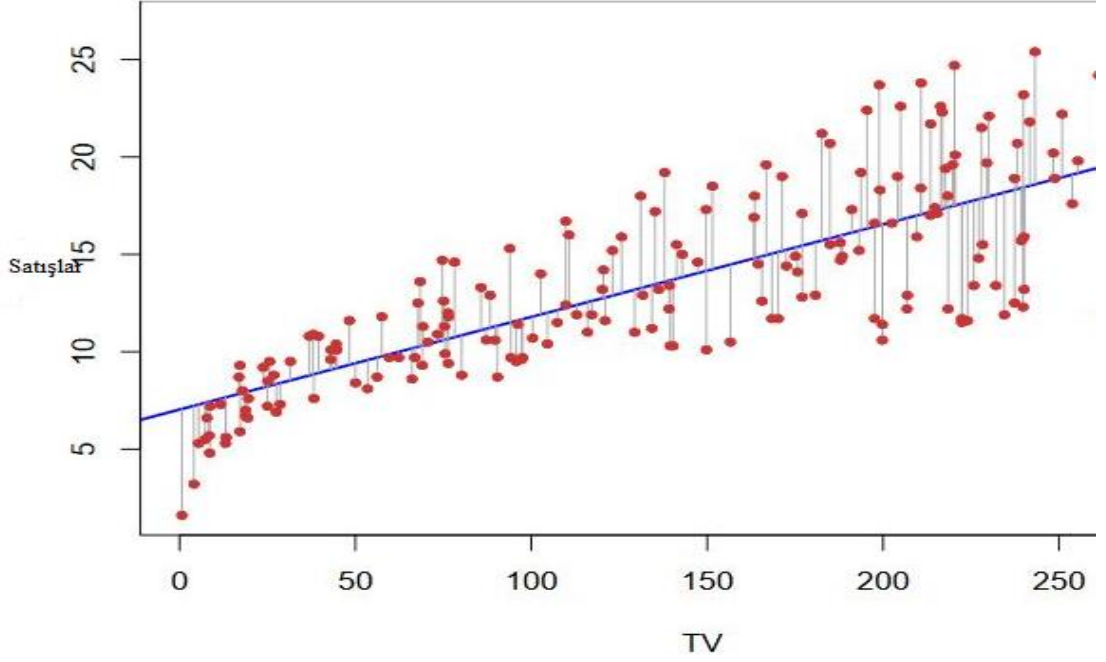
- ***Basit Doğrusal Regresyon***

$$Y = \beta_0 + \beta_1 X + \varepsilon \quad (1.1)$$

Y, bağımlı değişken ,X açıklayıcı değişken, β lar regresyon katsayıları ve ε hata terimi olmak üzere basit doğrusal regresyon denklemdir.Bağımsız değişken ile bağımlı değişkendeki değişimi açıklamayı, bağımsız değişkendeki 1 birimlik değişimin bağımlı değişken üzerindeki etkisini ölçmeyi amaçlar.Bu denklem ε hata terimini içeriyorsa olasılıklı, istatistiksel veya stokastik ilişki olarak, ε hata terimini içermiyorsa kesin veya matematiksel ilişki olarak ifade edilir. Buradaki hata terimi gerçek Y değeri ile tahmin edilen Y değeri arasındaki farktır.Bunun ortaya çıkmasında, modele alınmayan değişkenler, ölçme yanlışlıkları ve matematiksel kalıpların yanlış seçilmesi gibi

³ <https://yz-ai.github.io/blog/pekistirmeli-ogrenme/pekistirmeli-ogrenme-bolum-1>, (15.11.2020)

nedenlerin etkili olduğu değerlendirilebilir. Aşağıdaki şekilde mevcut verilerin modellendiği bir doğrusal regresyon görülmektedir.



Şekil 1: Doğrusal Regresyon

Şekil 1'e bakıldığında, kırmızı noktalar gerçek değerleri, mavi çizgi ise regresyon doğrusunu göstermektedir. Kırmızı noktaların mavi regresyon doğrusuna olan dik izdüşümleri ise gerçek ve tahmin değerleri arasındaki farkı göstermektedir.

• **Çoklu Doğrusal Regresyon**

Gerçek hayata bakılırsa doğrusal regresyonda olduğu gibi bağımlı değişkeni tek bir açıklayıcı değişken ile açıklanması çok gerçekçi bir durum olmaz. Dolayısıyla birden fazla açıklayıcı değişkene ihtiyaç olacaktır. Bağımlı değişkenin birden fazla değişkenle modellenmesi gerektiğinde çoklu doğrusal regresyon modeli kullanılmaktadır. Buradaki esas amaç her bir bağımsız değişkenin bağımlı değişken üzerindeki kısmi etkisinin ölçülmesidir.

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \dots + \beta_i X_i + \varepsilon \quad (1.2)$$

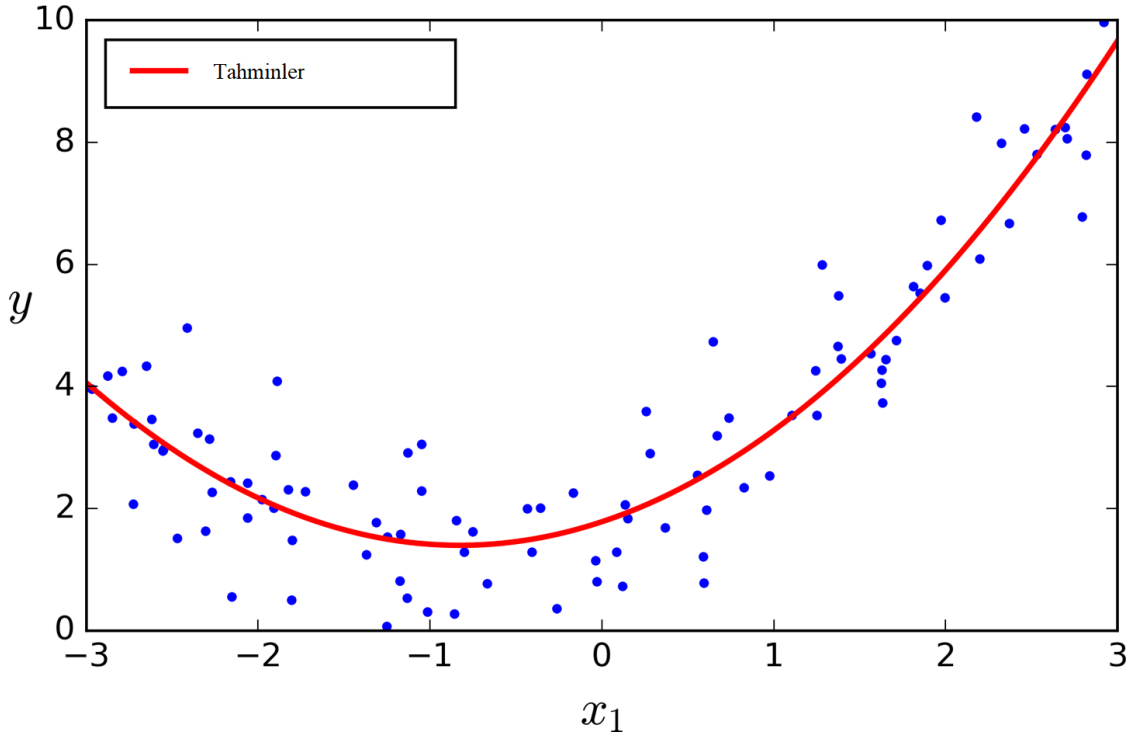
Y bağımlı değişken, X'ler açıklayıcı değişken, β lar regresyon katsayıları ve ε hata terimi olmak üzere çoklu doğrusal regresyon denklemdir.

- **Polinom Regresyon**

Bağımlı değişkenin birden fazla bağımsız değişkenle ve X 'in bir n . Dereceden fonksiyonu olarak modellenmesi gereken durumlarda polinom regresyon kullanılmaktadır.

$$Y = \beta_0 + \beta_1 X + \beta_2 X^2 + \dots + \beta_i X^n + \varepsilon \quad (1.3)$$

Y bağımlı değişken, X açıklayıcı değişken ve ε hata terimi olmak üzere n . Dereceden bir polinom regresyon denklemdir. Aşağıdaki şekilde mevcut verilerin modellendiği bir polinom regresyon örneği görülmektedir.



Şekil 2: Polinom Regresyon ⁴

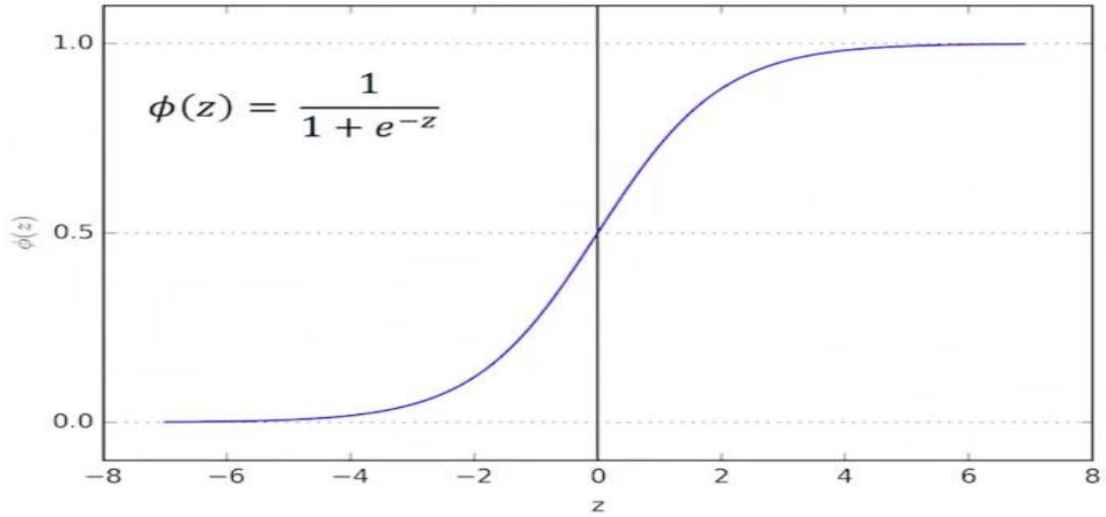
⁴ <https://medium.com/@ekrem.hatipoglu/machine-learning-prediction-algorithms-polynomial-regression-part-4-6c62b4240b53>, (23.11.2020)

1.3.2. Lojistik Regresyon

Lojistik regresyon genellikle bağımlı değişkenin iki veya daha fazla kategoriye sahip olduğu durumlarda kullanılmaktadır. Zaten bağımlı değişkenin iki kategorisi varsa buna ikili lojistik regresyon, ikiden fazla kategoriye sahip ise buna da multinominal lojistik regresyon adı verilir.⁵Matematiksel olarak lojistik regresyon(logit) fonksiyonu 0 ile 1 arasında değerler alır ve şu şekilde ifade edilir:

$$\text{logit}(x) = \frac{1}{1 + e^{-x}} \quad (1.4)$$

X bağımsız değişken ve e Euler sayısı olmak üzere Lojistik regresyon fonksiyonudur.



Şekil 3: Lojistik Regresyon Fonksiyonu Eğrisi

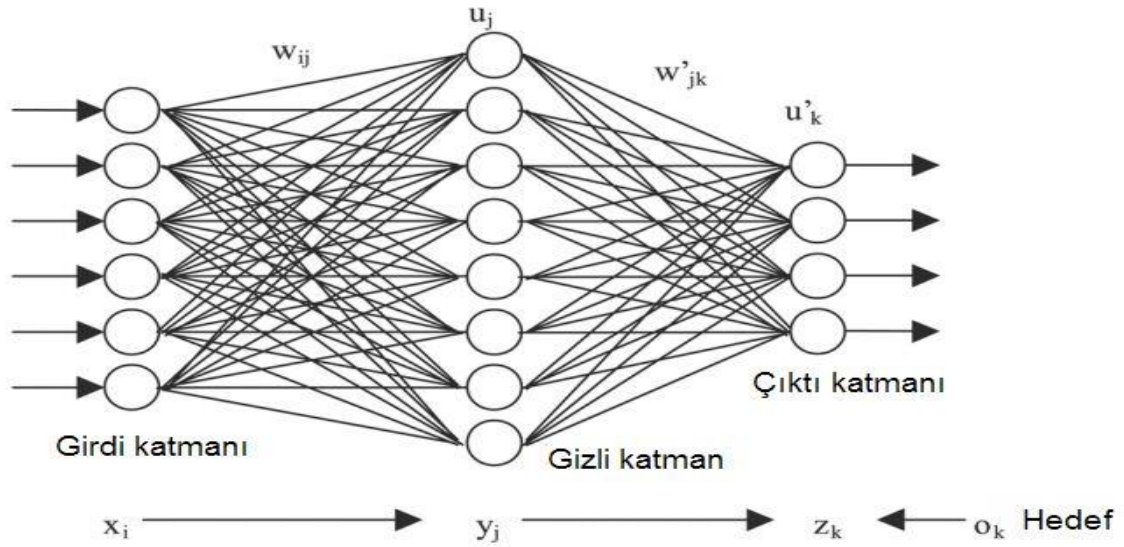
Şekil 3'te lojistik regresyon eğrisi görülmektedir. Lojistik regresyonda, doğrusal regresyonda olduğu gibi karşılanması gereken varsayımlar olmadığı için kullanım alanı daha geniştir. Lojistik regresyonda başarılı tahminler için çok fazla veri gereklidir. Ayrıca verilerdeki sapan değerler çıkarılmalıdır.⁶Doğrusal regresyondan en önemli farkı, doğrusal regresyonda bağımlı değişken sürekli iken lojistik regresyonda bağımlı değişken kategoriktir. Bu nedenle her ne kadar regresyon sözcüğü geçiyor olsa da burada bir sınıflandırma söz konusudur.

⁵ Nuran Bayram, Sosyal Bilimlerde SPSS İle Veri Analizi, 6. Baskı, Bursa: Ezgi Kitabevi, 2017, s.212

⁶ Peter Wlodarczak, Machine Learning and Its Applications, Boca Raton: CRC Press, 2020, s.94

1.3.3. Yapay Sinir Ağları

Yapay sinir ağları, insan vücudundaki sinir sisteminden esinlenilerek tasarlanmıştır. Nasıl ki insan hayatında tecrübe eder, bilgi edinir ve bunları yeni karşılaştığı durumlarda genelleyerek kullanırsa, yapay sinir ağları da aynen bu şekilde işlem yapar. İnsan sinir sistemini adeta taklit eder. Yapay sinir ağlarının kullanım alanları: Kontrol ve sistem tanımlama, görüntü ve ses tanıma, tahmin ve kestirim, arıza analizi, tıp, haberleşme, trafik, üretim yönetimi olarak sayılabilir.⁷İrili ufaklı çok sayıda yapay sinir ağı yapısı mevcuttur. Aşağıdaki şekilde örnek bir yapay sinir ağı görülmektedir.



Şekil 4: Yapay Sinir Ağı Örneği

Şekilden de görüleceği üzere girdi katmanı, sonra gizli katman ve en sonunda çıktı katmanı mevcuttur. Bunların sayıları her ağa göre değişebilir. Yuvarlaklarla resmedilenler nöronlardır ve bunlar arasındaki çizgiler de nöronlar arası yollardır. Girdi ve çıktı katmanlarındaki her bir nöron ortadaki gizli katmanların her biriyle iletişim halindedir. Bu duruma tamamen bağlı bir yapı denir. Geleneksel işlemlerden daha farklı çalışırlar. Yapay sinir ağlarında yapılacak işlem çok sayıda nörona dağıtılarak küçük küçük parçalar halinde yapılır. Aslında basitçe, her nöron almış olduğu girdi verisini

⁷ Harun Pirim, Yapay Zeka, Journal Of Yasar University,2006, s.1-13

işleme sokarak bir çıktı oluşturur. Nöronlar arasında istendiği şekilde ağırlıklandırma yapılabilir.

1.3.4. Bayesyen Modeller

Bayesyen modeller özü Bayes teoremine ilişkilendirilerek oluşturulmuş bir yaklaşımdır. İstatistikte temel olarak biri klasik diğeri de Bayesyen olmak üzere iki yaklaşım vardır. Birçok konu ve kavramların ele alınmasında bu iki yaklaşım birbirine alternatif olmuştur. Bayesyen yaklaşımın işleyişi klasik yaklaşımın aksine, varsayımlar olmadan deneme ve doğrulama adımlarından oluştuğu söylenebilir. Bayes modelinde esas amaç yanlış sınıflandırma ihtimalini en aza indirmektir.

$$\text{Bayes Teoremi: } P(A/B) = \frac{P(B/A)P(A)}{P(B)} \quad (1.5)$$

- **Naive Bayes**

Naive Bayes adından da anlaşılacağı üzere özünde Bayes teoremi ile ilişkilendirilip yapılandırılan basit bir olasılıksal sınıflandırıcıdır. 1700'lerde yaşamış İngiliz matematikçi Thomas Bayes'ten ismini alır. Basit, etkili ve uygulaması kolaydır. Diğer sınıflayıcılara göre daha hızlıdır. Her türlü veriyle kullanılabilirdiği gibi yüksek boyutlu verilerde de gayet iyi sonuçlar vermektedir. Veri az bile olsa yine etkili sonuç vermektedir. Bu yöntemin en iyi yönlerinden biri de tüm özelliklerin birbirinden bağımsız kabul edilmesidir. Böylelikle lojistik regresyondan çok daha etkilidir. Fakat aynı zamanda böyle olması da özellikler arasındaki ilişkinin modellenememesine yol açar. Bu yöntem veri madenciliği, tıp alanında hastalık teşhis etmede, çoklu sınıflandırma problemlerinde sıkça kullanılan bir yöntemdir.

1.3.5. Karar Ağaçları

Karar ağaçları sıklıkla kullanılan bir sınıflandırma algoritmasıdır. Karar düğümleri ve yaprak düğümlerinden oluşan, şekil olarak da gerçek bir ağaca benzeyen bir sınıflandırma yöntemidir. Büyük bir problemi daha küçük parçalara ayırarak işlem yapılır. Ağaçtaki ilk düğüme kök düğüm adı verilir ve son düğümlere ise yaprak düğümleri adı verilir ve bunlar bir sınıf etiketine sahiptir. Gerçek hayattaki ağaçlara

benzediği için anlaşılması ve yorumlanıp değerlendirilmesi daha kolaydır. Ayrıca birçok veri tipiyle kullanılabilir. Hem basit hem de karmaşık problemlerde kullanılabilir. Veriler çok karmaşık olması halinde oluşturulan ağaç da çok karmaşık olabilir ve bu durumda dal ve yaprak düğümleri takip edilemeyebilir. Ayrıca over-fitting (modelin eğitim verisi üzerinde gereğinden fazla çalışıp ezber yapması) de denen ezberleme durumu da oluşabilir. Bu durumu engellemek adına ise sınıflandırmaya katkısı olmayan bölümlerin karar ağacından çıkarılması olarak adlandırılan budama işlemi yapılmalıdır. Budama yapılırken de hangi yapraklarda daha az veri mevcut ise o yapraklar tercih edilmelidir. Aşağıdaki şekilde basit bir karar ağacı örneği görülmektedir.



Şekil 5: Karar Ağacı Örneği

Şekil 5'te görülen basit bir karar ağacında iki adımda karar verilebilir. Her yeni veri girişinde kök düğümden başlamak üzere aşağı doğru ağacın üzerinde gezinir.

Bir karar ağacı yapılandırılırken düğümlerin hangi sırada olacağını belirlemek adına homojenlik ile karar verilir. Homojenliği ölçmek için ise değişik ölçütler kullanılabilir. Bunlardan bazıları Entropi ve Gini indeksi ve bilgi kazancıdır. Entropi ise beklenmeyen bir durumun ortaya çıkma olasılığıdır. Şu şekilde hesaplanabilir:

$$E = \sum_{i=1}^c -p_i \log_2 p_i \quad (1.6)$$

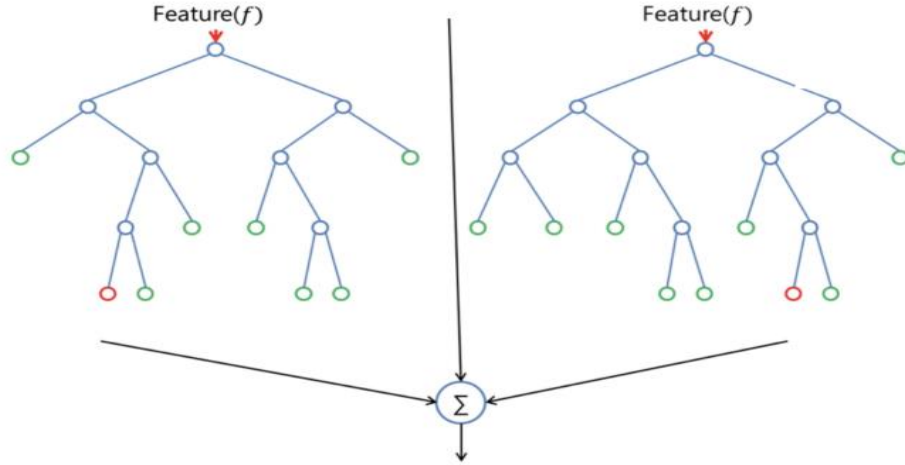
Bu denklemdeki p_i alt düğümde bulunan her bir sınıfın yüzdesidir. C ise sınıfları gösterir. Bilgi kazancı (Information Gain) ise şu formülle hesaplanabilir:

$$Gain_{(S,A)} = Entropy_S - \sum_{v \in value(A)} \frac{|S_v|}{|S|} Entropy_{(S_v)} \quad (1.7)$$

Karar ağacını oluştururken her düğümde bu bilgi kazancı hesaplanır ve en yüksek değerli olan düğüm en üste yazılır ve bu şekilde alta kadar devam edilir.

- **Rastgele Ormanlar**

Rastgele orman öğrenmesi eğitim aşamasında çok sayıda basit karar ağacı yani büyük bir orman üreterek en doğru tahmini yapmak için onları bir araya getirir. Sınıflandırma ve regresyon problemlerinde kullanılabilir. Karar ağaçlarının performansını yükseltmektedir. Ağaç sayısına bağlı olarak yapılan tahminin doğruluğu artıp azalabilir. Aşağıdaki şekilde basit bir rastgele orman örneği görülmektedir.



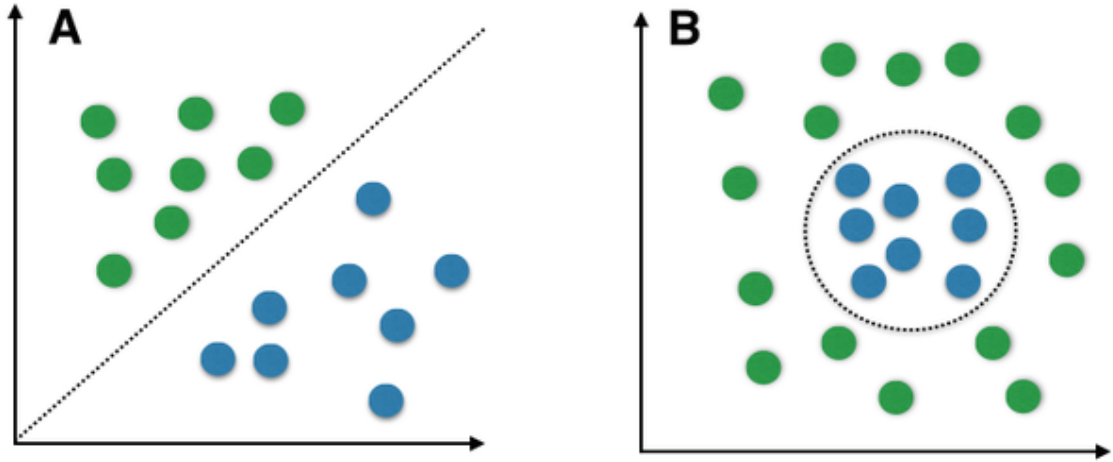
Şekil 6: Rastgele Orman Örneği

Şekilde 6'daki rastgele orman örneğinin iki ayrı ağaca benzediğini görülmektedir.

Karar ağaçları öğrenmesinde girilen eğitim verisine karşı çok hassastır. Bu verideki herhangi bir değişim çok çabuk şekilde karar ağacını etkileyebilir. Ama rastgele ormanlar yöntemi bu karar ağaçlarının ortalamasını kullandığı için eğitim verisine karşı o kadar da hassas değildir. Bunu yaparken ise torbalama (bagging) yöntemini kullanır. Torbalama tekniği varyansı azaltarak performansı iyileştirmeye çalışır ve oluşturduğu ağaçların tahmin sonuçlarının ortalamasını alarak işlem yapar. Rastgele ormanlar yöntemi popüler olarak tıp, bankacılık, ticaret, öneri sistemleri gibi alanlarda sıkça kullanılır.

1.3.6. Destek Vektör Makineleri

Destek vektör makineleri de (DVM) diğer yöntemlerde olduğu gibi sınıflandırma ve regresyon problemlerinde kullanılabilir. DVM temelde veri setindeki iki sınıfa ait örnekleri birbirinden ayıran hiper düzlem çizmek demektir. Burada dikkat edilecek en önemli nokta ise çizilecek olan karar çizgisinin veya hiper düzlemin sınırına en yakın olması gerekir ki bunlara da zaten destek noktaları denir. Temel olarak ise amaç regresyondakine benzer bir şekilde sınıflama hatasını en aza indirecek şekilde bu doğruyu (eğriyi) çizmektir. Aşağıdaki şekilde basit DVM örnekleri görülmektedir.



Şekil 7: Destek Vektör Makineleri Örneği

Yukarıdaki şekilde solda doğrusal DVM ve sağ tarafta ise doğrusal olmayan DVM görülmektedir. DVM 'de genellikle (-1,+1) sınıf etiketleri kullanılır. Şu şekilde ifade edilebilir:

$$X = \{(x_1, y_1), \dots, (x_n, y_n) : x_i \in \mathbb{R}^d, y_i \in \{-1, +1\}\} \quad (1.8)$$

Buradaki x_i ve y_i ifadeleri giriş vektörlerini gösterir. Ayrıca hiper düzlemin denklemi de şu şekilde ifade edilir:

$$w^T x + b = 0 \quad (1.9)$$

Bu denklemde w^T ağırlık vektörünü, x giriş vektörünü ve b ise sapmayı gösterir.

Bunun yanında bu denklem bazı kısıtları da sağlaması gerekir. Bunlar:

$$w^T x_i + b \geq +1 \quad y_i = +1 \quad \text{için} \quad (1.10)$$

$$w^T x_i + b \leq -1 \quad y_i = -1 \quad \text{için} \quad (1.11)$$

Özetlenirse DVM bir optimizasyon algoritmasıdır. Çok yönlü olmasının yanında aynı zamanda basit bir kullanımı da vardır. Birçok bilim alanında sınıflandırma algoritması olarak geniş bir kullanımı vardır.

1.3.7. K-En Yakın Komşu (k-NN)

K-en yakın komşu (k-NN), sınıflandırmanın yanı sıra regresyon için de kullanılan basit ve en temel makine öğrenmesi algoritmalarındandır. k-NN algoritmaları, 1967 yılında T. M. Cover ve P. E. Hart tarafından önerilmiştir.⁸Kolay yorumlanması, düşük hesaplama süresi avantajları arasında söylenebileceği gibi, tembel olması, özellikle uzaklık hesabında tüm veriler saklandığı için büyük verilerde çalışılacaksa eğer çok büyük depolama alanı gerekeceğinden dezavantajlıdır. Bu yüzden büyük veride tercih edilmemelidir. k-NN temelde yeni gelen veri için en yakın k tane noktaya (komşuya) bakılır. Bu noktaların değerine göre yeni gelen veri en yakın komşusunun sınıfına atanır. Yeni verinin k noktalarına olan uzaklığı bazı fonksiyonlarla hesaplanır. Bunlar:

$$\text{Öklid Uzaklık:} \quad \sqrt{\sum_{i=1}^k (x_i - y_i)^2} \quad (1.12)$$

⁸ <https://medium.com/@k.ulgen90/makine-%C3%B6%C4%9Frenimi-b%C3%B6l%C3%BCm-2-6d6d120a18e1>, (05.01.2021)

$$\text{Manhattan Uzaklığı: } \sum_{i=1}^k |x_i - y_i| \quad (1.13)$$

$$\text{Minkowski Uzaklığı: } \left(\sum_{i=1}^k (|x_i - y_i|)^q \right)^{\frac{1}{q}} \quad (1.14)$$

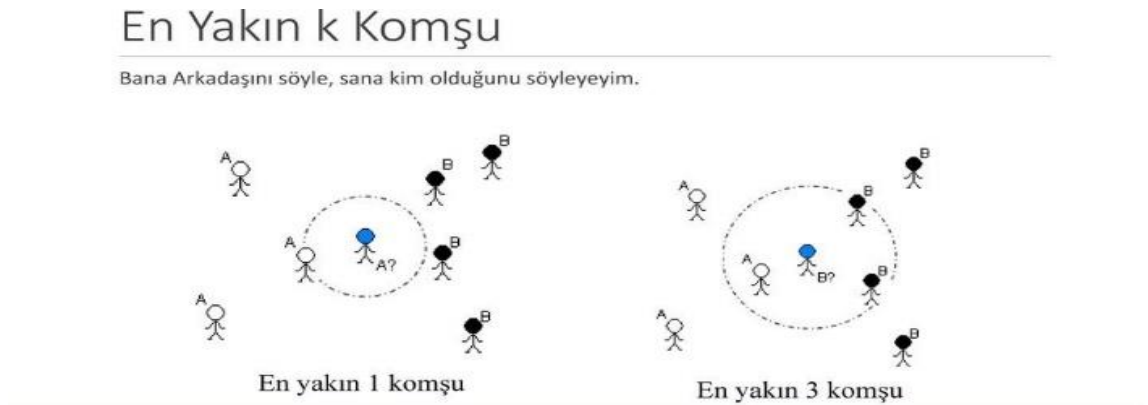
Burada adı geçen uzaklık fonksiyonları değişkenlerimiz sürekli olduğunda kullanılabilir. Değişkenler kategorik olursa şayet şu fonksiyon kullanılmalıdır:

$$\text{Hamming Uzaklığı : } D_H = \sum_{i=1}^k |x_i - y_i|$$

$$x = y \Rightarrow D = 0$$

$$x \neq y \Rightarrow D = 1 \quad (1.15)$$

Aşağıdaki şekilde basit bir k-NN örneği görülmektedir.



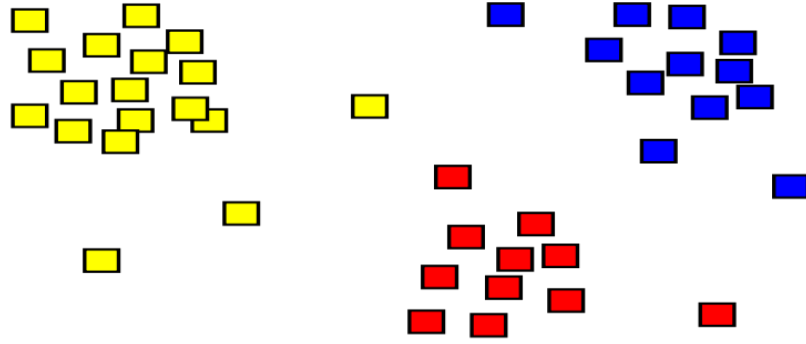
Şekil-8:k-NN Örneği

k-NN işlem adımları şöyle tanımlanabilir: Önce veri incelenir. Sonra uzaklıklar hesaplanır. Ardından en yakın k tane komşu bulunur ve en sonunda da yeni veri etiketlenir.

1.4. DENETİMSİZ ÖĞRENME YÖNTEMLERİ

1.4.1. Kümeleme Analizi

Kümeleme analizinde temel amaç, çeşitli istatistikler kullanılarak mevcut verileri kümelemeye çalışmaktır. Buradaki oluşturulacak olan kümelerin kendi içlerinde homojen yapıda ama kümelerin birbirlerine göre ise heterojen yapıda olması beklenir. Kümeleme yöntemleri genelde iki başlık altında incelenir. Bunlar, hiyerarşik kümeleme yöntemleri ve hiyerarşik olmayan kümeleme yöntemleridir. Aşağıdaki şekilde mevcut ham verilerin kümeleme sonrası nasıl olduğu görülmektedir.



Şekil 9: Kümeleme Örneği ⁹

Şekil 9'daki örneğe bakıldığında kümelerin içinde homojenliğin fazla kümeler arasında ise heterojenliğin fazla olduğunu, aynı zamanda aynı küme içindeki uzaklıkların az kümeler arası uzaklığın ise fazla olduğu görülmektedir.

1.4.1.1. Hiyerarşik Olmayan Kümeleme Yöntemleri

Kümeleme yapılmadan önce oluşturulacak olan küme sayısı araştırmacı tarafından belirlenmiş ise hiyerarşik olmayan kümeleme yöntemi kullanılabilir. İstenen sayıda küme oluşturulduktan sonra küme belirleme kriterine göre verilerin hepsi

⁹ https://tr.wikipedia.org/wiki/K%C3%BCmeleme_analizi#/media/Dosya:Cluster-2.svg, (19.01.2021)

kümelere atanıncaya kadar işleme devam edilir. Hiyerarşik olmayan yöntemlerden en çok kullanılanlar şunlardır:

- K-Ortalama Yöntemi
- Medoid Parçalama Yöntemi
- Yığılma/Yığılma Yöntemi
- Bulanık (Fuzzy) Kümeleme Yöntemi¹⁰

1.4.1.2. Hiyerarşik Kümeleme Yöntemleri

Hiyerarşik kümeleme yöntemleri genellikle daha küçük örneklemeler söz konusu olduğunda kullanılırlar ($n < 300$). Burada araştırmacı oluşturulacak küme sayısına değil, bu kümelerin arasındaki benzerlik veya uzaklığa ya da bunların nasıl ve ne şekilde ayrılıp birleştirileceğine karar vermesi gerekir. Çünkü bu yöntemde kaç küme oluşacağı önceden belli değildir. Bu yöntemde arka arkaya birleştirme ismi de verilebilir. Hiyerarşik yöntemler ise birleştirici ve ayırıcı olmak üzere iki temel grupta incelenebilir.

1.4.2. Temel Bileşenler Analizi

Temel bileşenler analizi esasında, aralarında korelasyon bulunan çok fazla sayıdaki değişkenlerle açıklanmak istenen bir sistemi veya yapıyı aralarında korelasyon olmayan daha az değişkenle açıklamaya çalışmaktır. Böylelikle değişkenler arası bağımlılık yapısı da ortadan kaldırılmaktadır. Bilindiği üzere çok değişkenli istatistikte çok fazla sayıda değişken kullanılmaktadır. Bu durum ise haliyle durumla alakalı bazı değerlendirme ve yorumları yapmayı çok zorlaştırmaktadır. Böyle bir durumda ise temel bileşen analizini kullanmak adeta bir zorunluluk haline almaktadır. Teknik olarak bakacak olursak eğer, elimizdeki veri setinin varyans-kovaryans matrisini de kullanarak değişkenler lineer bağımsız hale getirilerek boyut indirgenmeye çalışılır.

1.5. MODEL DOĞRULAMA YÖNTEMLERİ

Makine öğrenmesi algoritmaları kullanılarak mevcut veri setiyle ilgili birçok farklı model kurulabilir. Ama sadece bunu yaparak modelin başarısını veya hangi model veya algoritmanın daha başarılı olduğunu görülemez. Bunu belirleyebilmek için mutlak surette

¹⁰ Ömay Çokluk, Güçlü Şekercioğlu, Şener Büyüköztürk, Sosyal Bilimler İçin Çok Değişkenli İstatistik SPSS ve LISREL Uygulamaları, 3. Baskı, Ankara: Pegem Akademi, 2018, s.195

model deęerlendirme yöntemlerinin kullanılması gerekmektedir. Esasen bu yöntemler mevcut veri seti üzerinde nasıl ve ne şekilde örnekleme yapılacağıın yoludur. Bu yöntemler şöyle sıralanabilir:

- Hold Out (Dışarıda Tutma)
- K-katlı Çapraz Doğrulama
- Leave One Out (Birini Dışarıda Bırakma)
- Yeniden Örnekleme (Bootstrap)

1.5.1. Hold Out (Dışarıda Tutma)

Bu yöntemde mevcut veri seti, biri eğitim biri de test verisi olmak üzere iki parçaya ayrılır. Yaygın kullanım olarak veri seti 2/3 eğitim veri seti ve 1/3 test veri seti olarak ayrılır. Bazı araştırmacılar da yine bu oranı %70 eğitim ve %30 test verisi olarak da ayırabilmektedirler. Eğitim setiyle makine öğrenmesi gerçekleştirilir ve test verisiyle de bu öğrenmenin hangi oranda gerçekleştięi kontrol edilir. Bu yöntemin en önemli eksik yanı ise gözlem sayılarının az olması durumunda model kurulmasının zorluğu ve ayrıca gözlem sayısı az olduğundan eğitim ve test verilerine ayıracak kadar yeterli gözlem olmamasıdır. Bir dięer olumsuz durum ise eğitim ve test verileri en başta birbirinden ayrıldığı için bütün gözlemlerin kurulacak olan modelde kullanılmıyor olmasıdır. Sonuç olarak bu olumsuz durumlardan dolayı başka yöntemler de geliştirilmiştir.

1.5.2. K-Katlı Çapraz Doğrulama

Bu yöntemde elimizdeki veri seti her biri eşit olmak üzere k adet alt kümeye ayrılır. Sonra bu k adet alt kümelerin her seferinde biri test, dięer $k-1$ tanesi ise eğitim veri seti olarak kullanılmaktadır. Elde edilen k adet hata oranının ortalaması alınarak toplam tahmin hatası elde edilir. Bu yöntemde her veri k kez hesaplamaya girer. Bu durum ise k defa hesaplama gerektirdięi için olumsuz yönü olarak söylenebilir. Burada k sayısını seçerken dikkat etmek gerekir. Çünkü bariz bir şekilde k arttıkça hem varyans hem de hesaplama zamanı artacaktır. Aşağıdaki şekilde 10 katlı çapraz doğrulama örneęi görülmektedir.

	1. Parça	2. Parça	3. Parça	4. Parça	5. Parça	6. Parça	7. Parça	8. Parça	9. Parça	10. Parça
1. Adım	Test	Eğitim	Eğitim	Eğitim	Eğitim	Eğitim	Eğitim	Eğitim	Eğitim	Eğitim
2. Adım	Eğitim	Test	Eğitim	Eğitim	Eğitim	Eğitim	Eğitim	Eğitim	Eğitim	Eğitim
3. Adım	Eğitim	Eğitim	Test	Eğitim	Eğitim	Eğitim	Eğitim	Eğitim	Eğitim	Eğitim
4. Adım	Eğitim	Eğitim	Eğitim	Test	Eğitim	Eğitim	Eğitim	Eğitim	Eğitim	Eğitim
5. Adım	Eğitim	Eğitim	Eğitim	Eğitim	Test	Eğitim	Eğitim	Eğitim	Eğitim	Eğitim
6. Adım	Eğitim	Eğitim	Eğitim	Eğitim	Eğitim	Test	Eğitim	Eğitim	Eğitim	Eğitim
7. Adım	Eğitim	Eğitim	Eğitim	Eğitim	Eğitim	Eğitim	Test	Eğitim	Eğitim	Eğitim
8. Adım	Eğitim	Eğitim	Eğitim	Eğitim	Eğitim	Eğitim	Eğitim	Test	Eğitim	Eğitim
9. Adım	Eğitim	Eğitim	Eğitim	Eğitim	Eğitim	Eğitim	Eğitim	Eğitim	Test	Eğitim
10. Adım	Eğitim	Eğitim	Eğitim	Eğitim	Eğitim	Eğitim	Eğitim	Eğitim	Eğitim	Test

Şekil 10:10 Katlı Çapraz Doğrulama Örneği¹¹

Şekil 10'a bakıldığında k parametresi 10 olarak belirlenen bir çapraz doğrulama örneği görülmektedir. Mevcut veri seti 10 parçaya bölündükten sonra bir parçası test için ayrılıp diğer dokuz parça eğitim verisi olarak kullanılmaktadır. Bu işlem ise 10 adım devam ettirilerek, her adımda farklı bir parça test verisi olmak üzere tamamlanmaktadır. Bu 10 adımda yapılan sınıflandırmaların ortalamaları alınarak en son sonuç elde edilmektedir.

1.5.3. Leave One Out (Birini Dışarıda Bırakma)

K -katlı çapraz doğrulama yönteminin özel bir halidir. Fakat bu yöntemde k sayısı veri setindeki örnek sayısı ile aynıdır. Yani veri seti her birinde sadece bir örnek olacak şekilde n tane alt kümeye ayrılmaktadır. Bir önceki yöntemde olduğu gibi bu alt kümelerden biri test, diğer geriye kalan $n-1$ tanesi ise eğitim için kullanılmaktadır. Her seferinde bulunan hataların ortalaması alınarak toplam hata bulunur. Bu yöntemde çok fazla sayıda veri eğitim setine girdiği için modelin tahmin başarısı yükselir. Fakat veri setindeki tüm örnek sayısı kadar test yapılacağı için hesaplama çok zor ve uzun olacağı gibi büyük veri setlerinde kullanımı ise çok da uygun olmayacaktır.

¹¹ <https://www.ahmetcevahircinar.com.tr/2017/03/25/k-kez-capraz-dogrulama-yontemi/>, (22.01.2021)

1.5.4. Yeniden Örnekleme (Bootstrap)

Bu yöntemde mevcut veri setindeki örnek sayısı kadar rassal olarak eğitim verisi için örnekleme yapılmaktadır. Bu yapılırken seçilen herhangi bir örnek tekrar yerine koyulduğu için bir örnek birden fazla kez eğitim veri setinde yer alabilmektedir. Geriye kalan örnekler ise test veri setini oluşturmaktadır. Bulunan tahmin hatalarının ortalaması modelin tahmin hatasını belirlemektedir. Bu yöntem küçük veri setlerinde uygulanması daha uygun olduğu söylenebilir.

1.6. MODEL BAŞARISINI DEĞERLENDİRME YÖNTEMLERİ

Makine öğrenmesi yaklaşımında kurulan modelin başarısı mutlaka test edilmelidir. Bunun için bazı yöntemler mevcuttur. Karışıklık matrisi(confusion matrix) ile doğruluk(accuracy), hata oranı(error rate), anma(recall), kesinlik(precision), F-ölçütü gibi değerler ile ROC eğrisi gibi yöntemlerden birisi kullanılabilir.

1.6.1. Karışıklık Matrisi

Makine öğrenmesinde sınıflandırma problemlerinde modelin ne kadar doğru ne kadar yanlış tahmin yaptığının değerlendirilmesi gereklidir. Doğruluk, hata oranı, anma ve F ölçütü gibi değerleri karışıklık matrisi kullanılarak hesaplanabilir.

Karışıklık Matrisi	Tahmin		
		1	0
Gerçek	1	TP	FN
	0	FP	TN

Tablo 1: İkili Sınıflandırma Karışıklık Matrisi Örneği

Tablo 1’de görülen karışıklık matrisindeki TP, FP, FN, TN gerçek durum ile tahmin durumu arasındaki değerlendirmeyi ve aradaki ilişkiyi ifade etmek için kullanılır.

TP(True Positive, Doğru Pozitif): Modelde doğru tahin edilen pozitif değerlerin sayısıdır.

TN(True Negative, Doğru Negatif):Modelde doğru tahin edilen negatif değerlerin sayısıdır.

FP(False Positive, Yanlış Pozitif):Modelde yanlış tahmin edilen pozitif değerlerin sayısıdır.

FN(False Negative, Yanlış Negatif):Modelde yanlış tahmin edilen negatif değerlerin sayısıdır.

- ***Doğruluk Oranı***

Doğruluk oranı bir modelin başarısını değerlendirmede en önemli ve geçerli ölçütlerden birisidir. Doğruluk modelde doğru tahmin edilen değerlerin bir oranıdır. İdeal bir sınıflandırma modelinde karışıklık matrisinde bulunan TP ve TN gözelerinin dolu diğer gözelerin ise boş olması beklenir. Doğruluk aşağıdaki denklem ile hesaplanabilir:

$$\text{Doğruluk} = \frac{TP + TN}{TP + TN + FP + FN} \quad (1.16)$$

- ***Tahmin Hatası***

Tahmin hatası modelde algoritmanın yanlış tahmin ettiği değerlerin oranıdır. Tüm olasılık 1'e eşit olduğundan tahmin hatası 1'den doğruluk oranı çıkarılarak bulunabildiği gibi aşağıdaki formül de kullanılabilir:

$$\text{Tahmin hatası} = \frac{FP + FN}{FP + FN + TP + TN} \quad (1.17)$$

- ***Kesinlik (Precision)***

Kesinlik bazı kaynaklarda duyarlılık ismiyle de geçmektedir. Modelin gerçekte pozitif olan değerleri doğru tahmin etme oranını ifade etmektedir. Birazdan bahsedilecek olan anma ölçüsü ile kesinlik ters orantılıdır. Modelden istenen ise kesinlik değerinin yüksek olmasıdır. Şu şekilde hesaplanabilir:

$$\text{Kesinlik} = \frac{TP}{TP + FP} \quad (1.18)$$

- **Anma (Recall)**

Anma ölçütü kesinlik ile ters orantılıdır. Anma oranı yüksek olan modeller daha başarılıdır denilebilir. Şu şekilde hesaplanabilir:

$$Anma = \frac{TP}{TP + FN} \quad (1.19)$$

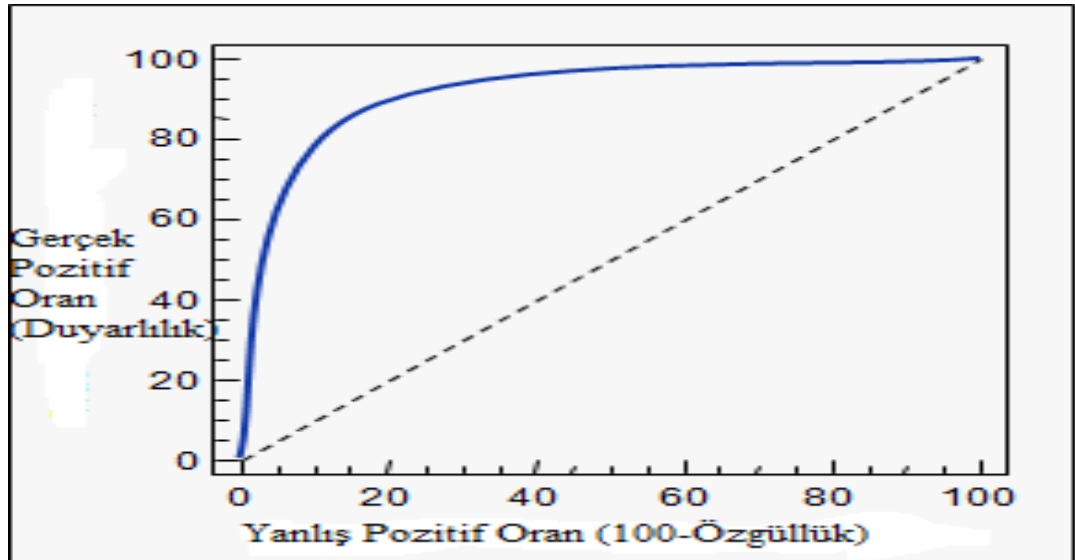
- **F-Ölçütü (Skoru)**

Kesinlik ve anma değerlerinin bir kombinasyonu olan F skoru esas olarak kesinlik ve anma değerlerinin harmonik ortalamasıdır. Model başarısını değerlendirmede kullanılır. Yüksek F skoru istenen bir durumdur. Genellikle sınıflandırma modellerini karşılaştırmakta kullanılır. Şu şekilde hesaplanabilir:

$$F = (2 * Kesinlik * Anma) / (Kesinlik + Anma) \quad (1.20)$$

1.6.2. Roc Eğrisi

Makine öğrenmesinde sınıflandırma modelinin başarısının test edilmesi elzemdir. ROC eğrisi yöntemi modeli değiştirmekte kullanılan en önemli yollardan biridir. ROC eğrisi yanlış pozitif oran ile doğru pozitif oran arasındaki ilişkinin grafiğidir.



Şekil 11: ROC Eğrisi

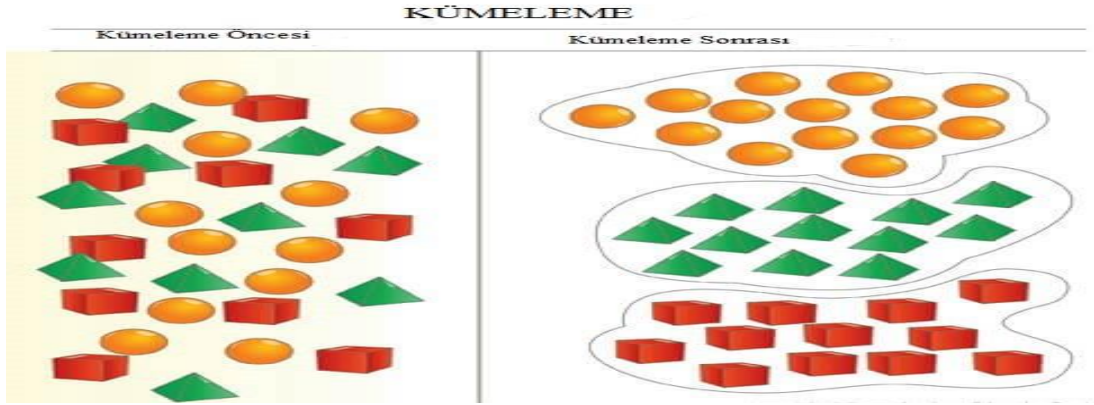
Şekil 11'deki ROC eğrisinde yatay ekseninde yanlış pozitif oranı dikey ekseninde ise doğru pozitif oranı görülmektedir. ROC eğrişi altında kalan alan AUC olarak isimlendirilmektedir. Bu alan ne kadar büyükse model o derece başarılıdır denilebilir. İdeal olan durum bu alanın 1'e eşit olması durumudur. Tam tersi olarak ise bu alan 1'e ne kadar uzak ise modelin o kadar başarısız olduğu söylenebilir.

İKİNCİ BÖLÜM

KÜMELEME ANALİZİ

2.1. KÜMELEME ANALİZİ

İstatistik geçmişten bugüne verilerle ilgilenmiş ve bunlardan çeşitli çıkarımlar yapmıştır. Ölçemediğinizi yönetemezsiniz ve iyileştiremezsiniz(Işığışok, 2004:vii).Bu sözden hareketle istatistikte ölçümün ne kadar önemli olduğunu söylemek mümkündür. Günümüzde geline noktada ise birçok veri vardır ve bunların ölçümü, depolanması ve değerlendirilmesi hiç de kolay değildir. Bu yapılacak işlemleri kolaylaştırmak adına verilerin belli özellik ve değişkenlere göre kümelemek ve sınıflandırmak elzem hale gelmektedir. Kümeleme analizi (cluster analysis) esasen sınıflandırma yapabilmek adına kullanılan birçok işlemin genel adıdır. Mevcut veri setindeki örnekleri, belli değişkenlere bağlı olarak alt gruplara ayırma işlemidir. Birçok bilim dalında sıklıkla kullanılan bir yöntemdir. Buradaki esas amaç verilerin benzer özelliklerine göre kümelenmesi ve araştırmacıya değerlendirme yapmak adına fayda sağlamaktır. İlk öncelik tabii ki de verilerin en mükemmel şekilde doğru kategorilere sınıflandırılmasıdır. Bu yönüyle istatistikteki diskriminant analizine benziyor olabilir ama kümeleme analizi verilerin anlık durumuna göre işlem yaptığı için bir tahmin veya kestirim yapamaz. Makine öğrenmesinde daha önce anlatılan eğitim verisi de kullanılarak sınıflandırma yapılan yaklaşımlardır. Kümeleme analizinde eğitim verisi kullanılmamasından dolayı diğer yaklaşımlardan farklıdır.(Alp ve Öz, 2020:190) Tıp, endüstri, psikoloji, bankacılık... vb. bir çok alanda sıkça kullanılmaktadır. Sınıflandırma yapılırken grup içindeki verilerin homojenliğini, gruplar arasındaki heterojenliği maksimum kılmak hedeflenir. Aşağıdaki şekilde ham halde bulunan mevcut veri setindeki değerlerin kümeleme öncesi ve sonrası durumları gösterilmektedir.



Şekil 12: Kümeleme Öncesi ve Sonrası Verilerin Görünümü

Şekil 12’ de görüldüğü üzere kümeleme öncesinde çok karışık ve anlamsız şekilde görünen ham verilerin, kümeleme sonrasında gayet anlaşılır ve kolay değerlendirme yapılabilir hale geldiği gözlenmektedir.

Özdamar (2004:280) kümeleme analizinin esasen dört farklı amaca hizmet ettiğini vurgulamıştır:

- n sayıda örneği p adet değişkene göre mümkün olduğunca kendi aralarında homojen ve yine kendi aralarında heterojen olacak şekilde alt kümelere ayırmak,
- p sayıda değişkeni, n sayıda örnekteki değerlere göre ortak özellikleri açıkladığı varsayılan alt kümelere ayırmak ve ortak faktör yapıları ortaya koymak,
- Hem örnekleri hem de değişkenleri aynı anda ele alarak ortak n örneği p değişkene göre ortak özellikli alt kümelere ayırmak,
- Örnekleri, p adet değişkene göre belirlenen değerlere göre gösterdikleri biyolojik ve tipolojik sınıflamayı ortaya koymak(taksonomik sınıflandırma yapmak)

Kümeleme analizi yapılırken belli adımlar vardır. Bu adımları şu şekilde sıralanabilir:

- Mevcut örneklerin değişkenlerine ait gözlemlerin oluşturulması yani veri matrisinin elde edilmesi,
- Sonra bu örneklerin birbirleriyle olan uzaklık veya benzerliklerinin hesaplanarak bu matrisin oluşturulması,
- Kümeleme yönteminin belirlenip elde edilen matrislerle bu kümelerin oluşturulması,
- Son olarak ise bu elde edilen kümelerin yorumlanması

Yukarıda sayılan adımlara da bakılacak olursa kümeleme analizinin çok sayıda işlevi bir arada yürüten bir yöntem olduğu söylenebilir. Bunu yaparken ise özellikle hangi değişkenlerin veri matrisine konulacağına çok dikkat etmek gerekir. Özellikle sapan değerlerin buradan çıkarılması gerekmektedir. Bunları göz önüne alınırsa araştırmacının çok dikkat etmesi gerektiğini ve ona bu konuda çok büyük bir görev düştüğü söylenebilir.

2.1.1. Kümeleme Analizi Varsayımları

Birçok istatistiksel yöntemlerdeki varsayımlar olan doğrusallık, normallik ve sabit varyans(homoskedastisite) kümeleme analizinde çoğu zaman göz ardı edilmektedir. Burada üzerinde durulan konu ise çoğu zaman örneklemin anakütleyi iyi bir şekilde temsil etmesi ve çoklu doğrusal bağlantı probleminin çözülmesi olarak görülebilir. Bundan dolayı kümeleme analizinin varsayımları şu şekilde özetlenebilir:¹²

Veri Kalitesi: Birçok istatistiksel yöntemlerde bağımsız değişkenler bağımlı değişkeni açıklarlar. Kümeleme analizinde ise böyle bir şey söz konusu değildir. Değişkenler açıklama amacıyla kullanılmadığı gibi aksine kümeleme analizinde esasen dikkate alınan durum değişkenler arası ilişkilerdir. En önemli nokta ise örnekleme girilen verilerin geçerli olmasıdır.

Anakütlenin Temsili: Bir araştırma yaparken çoğu zaman anakütlenin tamamına sahip olmak pek mümkün değildir. Bu nedenle örneklem üzerinde çalışılması gerekir. Bu kullanılan örneklemin anakütleyi en iyi şekilde temsil etmesi gerekir. Böylelikle araştırmanın sonunda elde edilen sonuç tüm anakütle genellenebilir.

Değişkenler Arası Çoklu Bağlantı: Çoklu bağlantı sorunu aslında değişkenlerin esas etkilerinin anlaşılmasındadır. Bu durum kümeleme analizinde bir nevi dolaylı ağırlıklandırmadır. Eğer bu problem varsa gruptaki değişken sayısı eşitlenebilir veya korelasyonu dengeleyebilecek bir uzaklık ölçüsünün kullanılması tavsiye edilir.(Hair ve ark. 2006)

¹² Ömay Çokluk, Güçlü Şekercioğlu, Şener Büyüköztürk, Sosyal Bilimler İçin Çok Değişkenli İstatistik SPSS ve LISREL Uygulamaları, 3. Baskı, Ankara: Pegem Akademi, 2018, s.153

2.2. VERİLERİN STANDARTLAŞTIRILMASI

İstatistikte değişken veya veriler aynı ölçekle ölçüldüğünde verileri kıyaslamak ve işlem yapmak sorunsuz olacağı için standartlaştırmaya ihtiyaç duyulmayacaktır. Veriler veya değişkenler farklı ölçeklerle ölçüldüğünde ise standartlaştırma şart olmaktadır. Bu gibi durumlarda en sık kullanılan işlem ise verilerin standart puanlara dönüştürülmesidir.

Z Standartlaştırması

Bu yöntemde mevcut verilerin standart sapması 1 ve aritmetik ortalaması 0 olacak şekilde bir işlem yapılır. Bundan dolayı herhangi bir verinin ortalamasının altında mı yoksa üstünde mi olacağını çok çabuk anlaşılır. Buna ek olarak yine bu verinin ortalamadan ne kadar sapsmış olduğu da kolaylıkla tespit edilebilir. Şu formülle hesaplanabilir:

$$z_i = \frac{x_i - \bar{x}}{s} \quad (2.1)$$

-1 ≤ x ≤ 1 Aralığında Standartlaştırma

Genellikle homojen olmayan ve uç değerlerin olduğu veri yapılarında tercih edilen bir yöntemdir. x_{\max} veri setindeki en büyük değer ve x_{\min} en küçük değer ve DG dizi genişliği olmak üzere şu şekilde hesaplanır:

$$s_i = \frac{x_i - \left(\frac{x_{\max} + x_{\min}}{2}\right)}{\frac{DG}{2}} \quad (2.2)$$

0 ≤ x ≤ 1 Aralığında Standartlaştırma

Bu yöntem de yine heterojen yapıda ve uç değerlerin olduğu veri yapılarında tercih edilir. Şu şekilde hesaplanabilir:

$$s_i = \frac{x_i - x_{\min}}{DG} \quad (2.3)$$

En Büyük Değer 1 Olacak Şekilde Standartlaştırma

En büyük değerin 1 olması istendiği durumlarda tercih edilir. Şu şekilde hesaplanabilir:

$$S_i = \frac{x_i}{x_{\max}} \quad (2.4)$$

Dizi Ortalaması 1 Olacak Şekilde Standartlaştırma

Dizinin aritmetik ortalamasının 1 olması istendiğinde tercih edilen yöntemdir. Şu şekilde hesaplanabilir:

$$S_i = \frac{x_i}{\bar{x}} \quad (2.5)$$

Dizinin Standart Sapması 1 Olacak Şekilde Standartlaştırma

Oluşturulacak olan yeni dizinin standart sapması 1 olması istendiğinde tercih edilen yöntemdir. Şu şekilde hesaplanabilir:

$$S_i = \frac{x_i}{s} \quad (2.6)$$

t Standartlaştırması

Oluşturulacak olan yeni dizinin ortalaması 50 standart sapması ise 10 olması istendiğinde tercih edilen yöntemdir. Şu şekilde hesaplanabilir:

$$t = 10 * z + 50 \quad (2.7)$$

2.3. UZAKLIK YA DA BENZERLİK MATRİSLERİNİN

OLUŞTURULMASI

Kümeleme analizinin ikinci adımı olan uzaklık veya benzerlik matrisleri oluşturulurken n tane gözlem ve p tane değişken üzerinden işlemler yapılır. Bunu yaparken ise gözlemler arası uzaklıktan veya benzerliklerden veya benzemezlik

matrislerinden yararlanılır. Burada bulunacak olan uzaklık ne kadar az ise gözlemler birbirine o kadar benzer ve yine uzaklık ne kadar çok ise gözlemler birbirine o kadar benzemez olacaktır. Uzaklık, benzerlik ya da benzemezlik ölçüleri kullanılırken elimizdeki verinin tipi çok önemlidir. Yani her veri tipine aynı ölçü birimi kullanılmaz. Çünkü veriler kesikli, sürekli veya kategorik şeklinde olabilirler.

Veri Tipi	Analizler	
	Uzaklık Ölçüleri	Benzerlik Ölçüleri
Sayısal Veriler	Öklit uzaklık ölçüsü	Pearson korelasyon katsayısı
	Kare öklit uzaklık ölçüsü	Kosinüs benzerlik ölçüsü
	Chebychev uzaklık ölçüsü	
	Manhattan City-blok uzaklık ölçüsü	
	Korelasyon uzaklığı ölçüleri	
Sıklık Sayıları	Ki-kare uzaklık ölçüsü	
	Phi-kare uzaklık ölçüsü	
İkili Veriler	Kare öklit uzaklık ölçüsü	Basit benzerlik ölçüsü
	Öklit uzaklık ölçüsü	Jaccard benzerlik ölçüsü
	Büyüklik farkları uzaklık ölçüsü	Parçalı benzerlik ölçüsü
	Biçim farkları uzaklık ölçüsü	Rogers ve Tanimoto benzerlik ölçüsü
	Değişim uzaklık ölçüsü	Sokal ve Sneath benzerlik ölçüsü 1,2,3,4,5
	Durum uzaklık ölçüsü	Yayılm benzerlik ölçüsü
	Lance ve Williams uzaklık ölçüsü	

Tablo 2: Veri Tipine Göre Ölçülerin Seçimi ¹³

¹³ <https://avys.omu.edu.tr/storage/app/public/vceyhan/125687/K%C3%BCmele%20Analizi.%20PDF.pdf>, (02.02.2021)

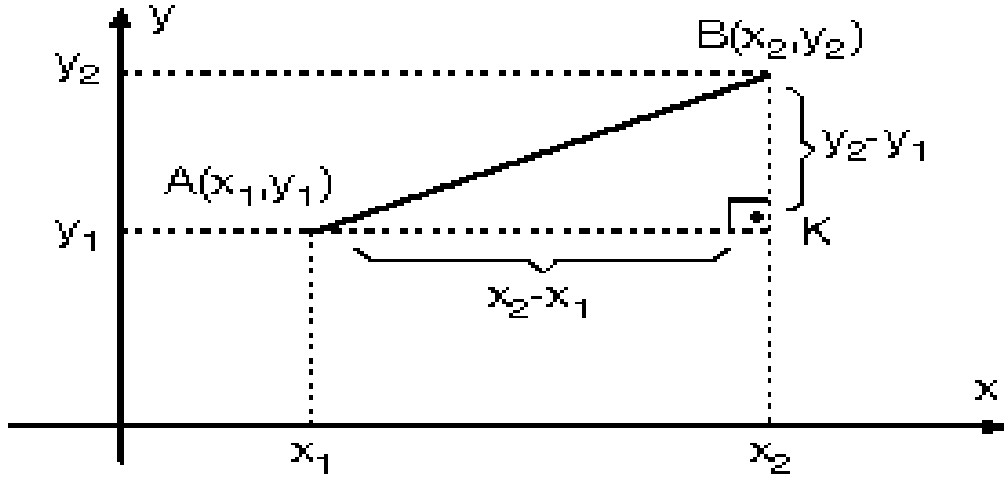
2.4. UZAKLIK VE BENZERLİK ÖLÇÜLERİ

2.4.1. Sayısal Veriler İçin Uzaklık Ve Benzerlik Ölçüleri

Sayısal veriler için uzaklık ölçüleri öklit, kare öklit, Chebychev, Manhattan (City Blok), Minkowski şeklinde ele alınır, benzerlik ölçüleri de Pearson Korelasyon Katsayısı ve Kosinüs Benzerlik Ölçüsü olarak ele alınır. Aşağıda bu ölçülere dair kısa açıklamalar yer almaktadır.

- **Öklit Uzaklık Ölçüsü**

Sık kullanılan ölçülerden biridir. Çok boyutlu uzayda Pisagor teoremi kullanılarak bulunabilir. İki boyutlu uzayda Öklit uzaklığı şu şekilde gösterilebilir:



Şekil 13: İki Gözlem Arası Uzaklık¹⁴

Öklit uzaklığı şu şekilde hesaplanabilir:

$$d(i, j) = \sqrt{(x_{i1} - x_{j1})^2 + (x_{i2} - x_{j2})^2 + \dots + (x_{ip} - x_{jp})^2} \quad (2.8)$$

$d(i, j)$: i ve j birimleri arasındaki uzaklıktır.

x_{ik} : i. Birimin k. Değişkeni ve

¹⁴ <http://www.analitikgeometri.com/ders-1-noktanin-analitik-incelenmesi.html>, (05.02.2021)

$i=1,\dots,n$, $j=1,\dots,n$ ve $k=1,\dots,p$ olmak üzere n adet birim ve p adet deęiřkendir.

Eęer kullanılan deęiřkenler belli bir řekilde aęırlıklandırılmıřsa denklem řu hale gelir:

$$d(i, j) = \sqrt{w_1(x_{i1} - x_{j1})^2 + w_2(x_{i2} - x_{j2})^2 + \dots + w_p(x_{ip} - x_{jp})^2} \quad (2.9)$$

- ***Kare Öklit Uzaklık Ölçüsü***

Öklit uzaklıęının karesinin alınmıř halidir. Öklit uzaklıęındaki gibi karekök alınmadıęı için uç deęerlere karřı daha hassas bir ölçüdür(Çilingirtürk,2011:168).

- ***Chebychev Uzaklık Ölçüsü***

Farkların mutlak deęerinin en büyüęü řeklinde tanımlanmıřtır ve řu řekilde hesaplanır:

$$d_{ij} = \max_k^p |x_{ik} - x_{jk}| \quad (2.10)$$

- ***Manhattan (City Blok) Uzaklık Ölçüsü***

Bu ölçüde birimler arasındaki mutlak uzaklıklar kullanılmaktadır. Genellikle sürekli deęil kesikli veriler için kullanılır. Aykırı deęerlere daha az hassas olan bir uzaklık ölçüsüdür(Timm,2002,s.517).řu řekilde hesaplanabilir:

$$d(i, j) = \sum_{k=1}^p |x_{ik} - x_{jk}| \quad (2.11)$$

- ***Minkowski Uzaklık Ölçüsü***

Daha genel bir uzaklık ölçüsüdür. Bu ölçüde parametre 1 olduęunda Manhattan, 2 olduęunda ise Öklit uzaklık ölçüsü haline gelmektedir. řu řekilde hesaplanabilir:

$$d_{ij} = \left[\sum_{k=1}^p |x_{ik} - x_{jk}|^\lambda \right]^{\frac{1}{\lambda}} ; \quad \lambda \geq 1 \quad (2.12)$$

- **Pearson Korelasyon Katsayısı Benzerlik Ölçüsü**

Bilindiği üzere korelasyon katsayısı değişkenler arasındaki doğrusal ilişkinin yönünü ve şiddetini belirlemeye yarar. Pearson korelasyon katsayısı ise değişkenler arası ilişkinin çok daha fazla olmasıyla ilgilenildiği zamanlarda kullanılmaktadır. k ve l değişkenleri arası Pearson korelasyon katsayısı şu şekilde hesaplanabilir:¹⁵

$$r_{kl} = \frac{\sum_{i=1}^n X_k X_l - \sum_{i=1}^n X_k \sum_{i=1}^n X_l}{\sqrt{\left[\sum_{i=1}^n X_k^2 - \frac{\left(\sum_{i=1}^n X_k \right)^2}{n} \right] \left[\sum_{i=1}^n X_l^2 - \frac{\left(\sum_{i=1}^n X_l \right)^2}{n} \right]}} \quad (2.13)$$

$k, l=1, 2, 3, \dots, p$

Buna bağlı olarak ise korelasyon uzaklığı şu şekilde hesaplanabilir:

$$d_{kl} = 1 - r_{kl} \quad (2.14)$$

Benzerlik ölçüsü ise şu şekildedir:

$$sim(k, l) = 100(1 - d(k, l) / \max(d(k, l))) \quad (2.15)$$

- **Kosinüs Benzerlik Ölçüsü**

Geometride kullanılan kosinüs teoreminde olduğu gibi aralarında α kadar açı olan herhangi iki a ve b vektörleri arasındaki uzaklık şu şekilde hesaplanabilir:

$$d_{ab} = \sqrt{a^2 + b^2 - 2ab \cos \alpha} \quad (2.16)$$

¹⁵ Kazım Özdamar, Paket Programlar İle İstatistiksel Veri Analizi, 5. Baskı, , Ankara: Nobel Akademik Yayıncılık, 2004, ss.286

2.4.2. Sıklık Sayıları İçin Uzaklık Ölçüleri

Sayısal verilerde çoğu zaman bir değişkene ait nicel ölçüm verileri kullanılmaktadır. Ama bazen bunlar yetersiz kalmaktadır. Özellikle bir değişkene ait nesnelerin sıklığı veya frekansı ile ilgili bir çalışma yapılacaksa ki-kare ve phi-kare adında yeni uzaklık ölçüleri geliştirilmiştir. Aşağıda bu ölçülere kısaca yer verilmiştir.

- **Ki-Kare Uzaklık Ölçüsü**

İki gözlem arasındaki ki-kare istatistiğinin karekökü uzaklık ölçüsü olarak adlandırılır. Şu şekilde hesaplanabilir:

$$D = \sqrt{\frac{(GD - BD)^2}{BD}} \quad (2.17)$$

Burada BD, beklenen değerleri, GD ise gözlenen değerleri ifade etmektedir.

- **Phi-Kare Uzaklık Ölçüsü**

Az önce hesaplanan herhangi iki gözlem arasındaki ki-kare uzaklık ölçüsünün toplam gözlem sayısına bölünüp karekökünün alınması ile elde edilir ve aşağıdaki gibi hesaplanabilir:

$$d_{ij} = \sqrt{\left[\frac{ki - kare_{ij}}{n_i + n_j} \right]} \quad (2.18)$$

2.4.3. İkili Veriler İçin Uzaklık Ve Benzerlik Ölçüleri

Günümüzde yapılan birçok araştırmada nicel değişkenlerden daha çok nitel yani kategorik değişkenlerin dikkate alındığı gözlemlenmektedir(Serper,2017, s.459). Buna ek olarak nicel verileri de yine gruplama yoluyla nitel veri durumuna getirmek mümkündür. Bu nitel verileri kategorik bir şekilde ele almak adına ikili veriler oluşturulmuştur. Bunlar çoğu zaman 0 ve 1 ile ifade edilmiştir. Buradaki 1 özelliğin var olmasını ifade ederken, 0 ise o özelliğin olmaması anlamına gelmektedir.2x2'lik bir matris kullanılarak ikili verilerin tablosu gösterilebilir:

		2. Değişken		Toplam
		0	1	
1. Değişken	0	a	b	a+b
	1	c	d	c+d
Toplam		a+c	b+d	N=a+b+c+d

Tablo 3: İkili Veriler İçin Kontenjans Tablosu¹⁶

- ***İkili Öklit Uzaklık Ölçüsü***

Yukarıda verilen kontenjans tablosu kullanılarak Öklit uzaklığı şu şekilde hesaplanabilir:

$$d_{ij} = \sqrt{b+c} \quad (2.19)$$

- ***Büyüklik Farkları Uzaklık Ölçüsü***

Bu ölçü yöntemi şu şekilde hesaplanabilir:

$$d_{ij} = \frac{(b-c)^2}{N^2} \quad 17 \quad (2.20)$$

- ***Biçim Farkları Uzaklık Ölçüsü***

Şu şekilde hesaplanabilir:

$$d_{ij} = \frac{bc}{N^2} \quad 18 \quad (2.21)$$

- ***Basit Benzerlik Ölçüsü***

Şu şekilde hesaplanabilir:

¹⁶ Kazım Özdamar, Paket Programlar İle İstatistiksel Veri Analizi,5. Baskı, , Ankara: Nobel Akademik Yayıncılık, 2004, ss.297

¹⁷ Reha Alpar, Uygulamalı Çok Değişkenli İstatistiksel Yöntemler,Ankara: Detay Yayıncılık,2011. ss.150

¹⁸ a.g.e., ss.151

$$d_{ij} = \frac{a + d}{N} \quad (2.22)$$

Formülden de görüleceği üzere hesaplanan değer örneklerin birbiri ile aynı olma olasılığıdır.

- **Jaccard Benzerlik Ölçüsü**

Jaccard katsayısı benzerlik oranı olarak da ifade edilir ve şu şekilde hesaplanabilir:

$$s_{ij} = \frac{a}{a + b + c} \quad (2.23)$$

Formülden de görüldüğü üzere örneklerdeki 1-1 eşleşmelerinin 0-0 hariç diğer eşleşmeleri oranı hesaplanmıştır. Dolayısıyla 1-1 eşleşmelerinin daha önemli olduğu araştırmalarda kullanılabilir.

- **Parçalı Benzerlik Ölçüsü**

Şu şekilde hesaplanabilir:

$$s_{ij} = \frac{2a}{2a + b + c} \quad (2.24)$$

Görüldüğü gibi Jaccard katsayısından farklı olarak 1-1 eşleşmesinin katsayısı iki katına çıkarılmıştır.

2.4.4. Kümeleme Analizi Yöntemleri

Kümeleme analizinde esasen daha önce de bahsedilen uzaklık ve benzerlik matrisleri kullanılarak kendi içinde homojen olmakla birlikte kendi aralarında heterojen yapıda olacak şekilde verileri gruplamaktır. Bunu yaparken ise iki temel yöntem bulunmaktadır. Bunlar, hiyerarşik kümeleme yöntemleri ve hiyerarşik olmayan kümeleme yöntemleridir.

2.4.4.1. Hiyerarşik Kümeleme Yöntemleri

Hiyerarşik yöntemlerde, uzaklık veya benzerlik matrisleri kullanılarak kümeler art arda olmak üzere ayrılmalar veya birleşmeler şeklinde işlem yapılır. Bu yöntemde yine ayırıcı ve birleştirici yöntemler olmak üzere iki ana gruba ayrılabilir.

Ayırıcı hiyerarşik yöntemlerde, ilk başta tüm birimler tek bir küme olarak kabul edilir. Daha sonra birimler arası uzaklıklar hesaplanarak küme sayısı ikiye çıkarılır. Bu işlem tüm birimler ayrı bir küme oluşturuncaya kadar devam eder. Yani başlangıçta bir olan küme sayısı işlem sonunda n tane olacaktır.

Birleştirici hiyerarşik yöntemlerde ise tam tersine başlangıçta her bir birim bir küme olarak kabul edilir. Yani küme sayısı ilk başta n 'dir. Daha sonra uzaklıklar hesaplanarak birbirine en yakın olan birimler birleştirilerek küme sayısı $n-1$ 'e düşürülür. Bu işlem de yine küme sayısı bir oluncaya dek devam ettirilir. Birimlerin uzaklıkları ve birbirleri ile olan bağlantıları dendogram adı verilen ağaç şeklindeki yapılar ile gösterilir.

Hiyerarşik yöntemlerden en sık kullanılanlar şu şekilde sıralanabilir:

- Tek Bağlantı Yöntemi
- Tam Bağlantı Yöntemi
- Ortalama Bağlantı Yöntemi
- Ward Yöntemi
- Merkezi Yöntem
- ***Tek bağlantı yöntemi***

En çok kullanılan yöntemlerden biridir ve ayrıca en kolay olanıdır. Çeşitli kaynaklarda en yakın komşuluk ismi ile de geçmektedir. Başlangıçta n adet olan küme sayısı, gözlemler arası uzaklıklar hesaplandıktan sonra en yakın olan iki gözlem birleştirilerek bir kümeye konulur. Daha sonra başka bir noktanın uzaklığı kümelenmemiş diğer noktalardan daha az ise bu nokta da az önce oluşturulan kümeye dâhil edilir. Bu işlem tüm gözlemler bir araya gelinceye dek sürdürülür.

Birbirlerine en yakın olan a ve b gibi iki gözlem birleştirildikten sonra, bu küme ile herhangi bir c kümesinin birleştirilmesi için bu c kümesi ile a ve b gözlemlerinin birleştirilmesi için şu şart olmalıdır:¹⁹

$$d_{c(ab)} = \min(d_{ca}, d_{cb}) \quad (2.25)$$

- ***Tam bağlantı yöntemi***

Tam bağlantı yöntemi, bir önceki başlıkta anlatılan tek bağlantı yöntemine oldukça benzemektedir. İstatistiksel literatürde ise Maksimum Yöntem, Çap Yöntemi ve En Uzak Komşu Analizi²⁰ gibi isimlerle de anılmaktadır, fakat tek bağlantı yönteminin aksine veriler arasındaki maksimum uzaklığı dikkate alır. Herhangi bir a kümesinin b ve c kümeleri arasındaki uzaklık şu şekilde tanımlanır:

$$d_{a(bc)} = \max(d_{ab}, d_{ac}) \quad (2.26)$$

Bu bulunacak olan uzaklıklardan hangisi daha fazla ise a kümesi o küme ile birleştirilerek yeni küme oluşturulur.

- ***Ortalama bağlantı yöntemi***

Ortalama bağlantı yönteminde uzaklık matrisi kullanılmaktadır. Herhangi bir a kümesinin hangi b veya c kümesiyle birleştirileceğini saptayabilmek için herhangi bir d kümesi ile b ve c kümeleri arasındaki uzaklık hesaplanır. Bunlar ise küme eleman sayıları ile ağırlıklandırılır. Sonra da yeni oluşacak olan kümenin eleman sayısına bölünerek bulunur. a ile d kümeleri arasındaki bu uzaklık şu şekilde hesaplanabilir:

$$d_{ad} = \frac{(N_b d_{bd} + N_c d_{cd})}{N_a} \quad (2.27)$$

¹⁹ Yusuf Altınok., Veri Madenciliğinde Hiyerarşik Kümeleme Yöntemlerinin Uygulamalı Karşılaştırılması, (Yüksek Lisans Tezi), İstanbul-2019

²⁰ Kazım Özdamar, Paket Programlar İle İstatistiksel Veri Analizi, 5. Baskı, , Ankara: Nobel Akademik Yayıncılık, 2004, ss.300

- **Merkezi yöntem**

Tek bağlantı yöntemine benzer şekilde ilk başta birbirine en yakın nesnelere aynı kümeye alınır. Sonra bu kümedeki değerlerin ortalaması alınarak burası küme merkezi olarak belirlenir. Daha sonra küme elemanlarının yerine bu merkez noktası hesaba katılarak nesnelere arası uzaklıklar tekrar hesaplanarak yeni bir küme merkezi elde edilir ve bu işlem tüm nesnelere aynı kümede toplanıncaya kadar devam edilir.

Her seferinde kümeler ve işlemlerin büyümesi bir dezavantaj olarak görülebilir. Bunun yanında ortalama değerler merkez olarak seçildiği için uç değerlerden daha az etkilenmesi ise avantaj olarak söylenebilir.

- **Ward yöntemi**

Hiyerarşik kümeleme yöntemleri arasında en iyi sonuç veren yöntem olarak kabul edilmektedir. (Hands ve Everitt, 1987; Ferreira ve Hitchcock, 2009). Esasen bu yöntemde kullanılan kriter, en küçük kareler yöntemindeki gibi grup içerisindeki dağılımı en aza indirerek kümelemeyi gerçekleştirmektir. Diğer bir deyişle küme içerisindeki nesnelere varyanslarını en aza indirmektir.

Herhangi iki k_1 ve k_2 kümeleri arasındaki uzaklık U , ve bu k_1 ve k_2 kümelerinde birbirlerine en yakın olan noktalar da x_1 ve x_2 olmak üzere bu kareler toplamı şu şekilde hesaplanabilir:

$$TU_{k_1 \cup k_2} = \sum_{x \in k_1 \cup k_2} U(X, \mu_{k_1 \cup k_2})^2 \quad (2.28)$$

2.4.4.2. Hiyerarşik Olmayan Kümeleme Yöntemleri

Hiyerarşik olmayan kümeleme yöntemlerinde, küme sayısı önceden belirlenip nesnelere bu kümelerin içine atanırlar. Dolayısıyla küme sayısı hakkında bir ön bilgi gerektiği durumlarda kullanılabilir. Bu yöntemde benzerlik matrisinin hesaplanması şart olmadığı gibi, elde edilen verinin yapılan işlem boyunca cihazda saklanması da şart değildir. Bunlar birer avantaj olarak değerlendirilebilir. Buna ek olarak bu yöntem hiyerarşik yöntemlerin aksine büyük verilere ($n > 1000$) de uygulanabilir. Büyük verilere uygulanmasında dolayı da yine uç değerlere olan duyarlılık nispeten daha azdır.

En sık kullanılan hiyerarşik olmayan yöntemler, K-means yöntemi ve en çok olabilirlik yöntemidir. Bunlara ek olarak yine medoid parçalama yöntemi, yığılma/yığılma yöntemi ve bulanık(fuzzy) kümeleme yöntemleri de mevcuttur.

- ***K-Ortalamalar (K-Means) yöntemi***

Sıklıkla kullanılan bir hiyerarşik olmayan yöntemdir. Her bir veri yalnızca bir kümeye ait olacak şekilde kümeleme işlemi gerçekleştirilir. Esas olarak amaç, eldeki n adet verinin, araştırmacının belirleyeceği k adet kümeye ayrılmasıdır. Her zaman olduğu gibi küme içi benzerliklerin en çok, kümeler arası benzerliklerin ise en az olmasıdır.

İşlem ilk olarak rastgele seçilen k adet merkez noktası kümeleri ile başlar. Daha sonra geriye kalan nesnelere kendilerine en yakın olan kümeye atanırlar. Daha sonra k adet küme merkezinin ortalamaları, bu kümelere ait olan nesnelere ile tekrar hesaplanır. Sonra bütün nesnelere yeniden hesaplanan küme merkezlerine atanırlar. Bu yeniden hesaplama işlemleri ilerledikçe küme merkezlerindeki değişimler azalacaktır. Nihayet küme merkezleri artık değişmeyince işlem sonlandırılır.

Kümeler arası uzaklık ise şu şekilde hesaplanabilir:

$$W_n = \frac{1}{n} \sum_{i=1}^n \min_{1 \leq j \leq k} |x_i - a_{j,n}|^2 \quad (2.29)$$

Burada x_i gözlem vektörünü, $a_{j,n}$ ise küme merkezlerini temsil etmektedir.(Tatlıdil, 2002:338)

- ***K-medoids (medoid parçalama) yöntemi***

Medoid, kelime anlamı olarak, bir veri grubu içerisindeki tüm nesnelere olan ortalama uzaklığı minimum olan, o kümenin bir elemanı demektir. K-medoids yöntemi ise çeşitli yapılar içeren bir veri grubu içerisindeki, bu yapıları en iyi şekilde temsil eden nesneyi bulma özelinde işlem yapmaktadır. Kaufman ve Rousseeuw tarafından 1987 yılında öne sürülmüştür. Bu yöntemin çok çeşitli türevleri mevcuttur. Veri madenciliğinde sıklıkla kullanılmaktadır.

K-ortalamlar yöntemindekine benzer olarak k küme sayısı seçilerek işleme başlanır. Daha sonra k adet başlangıç medoidi rastgele olacak şekilde seçilir. Sonra medoid haricinde kalan her birim, kendisine en yakın olan medoide atanır. En yakın medoidler için bütün birimlerin toplamı işleme konarak amaç fonksiyonu hesaplanır(Akın, 2008). Sonra yine rastgele olarak medoid olmayan bir y noktası seçilir. Seçilen x medoidi ile medoid olmayan y noktaları hataların minimize edilmesi yönüyle karşılaştırılır ve eğer y noktasında hata daha az ise x ile y yer değiştirilir. Böylelikle y yeni medoid olmuş olacaktır. x medoid seçimi ile y medoid olmayan nokta seçimleri birbirleri ile yer değiştirmeyinceye kadar bu işlem tekrar edilecektir.

- ***Bulanık (fuzzy) kümeleme yöntemi***

Daha önceden de belirtildiği gibi kümeleme analizi mevcut verilerin benzerlik ve uzaklık matrislerini kullanarak verileri kendi içerisinde homojen kendi aralarında ise heterojen olacak şekilde alt gruplara ayırmaktır. Bulanık kümeleme yöntemi ise K-ortalamlar ve medoid kümeleme yöntemlerinin genelleştirilmiş halidir denilebilir. Diğer kümeleme yöntemlerinin aksine bu yöntemde her nesnenin mutlaka bir kümeye ait olma gibi bir zorunluluk yoktur. Yani aykırı değerler bir kümeye girmeden kalabilirler. Bulanık kümeleme yönteminde birimlerin küme üyelik olasılıkları, tüm olası kümeler arasında sıfır ile bir arasında değişir.²¹Böylelikle bir nesne birden fazla kümeye üye olabilir. Bu duruma ise kümelemenin bulanıklığı adı verilir.

Bulanık kümelemede Bezdek ve Hathaway (1987) tarafından ileri sürülen ve daha sonra ise Kaufman ve Rousseeuw (1990) tarafından geliştirilen bir algoritma kullanılması söz konusudur. Aşağıda verilen C fonksiyonunu minimize etmek amaçlanmaktadır.

$$C = \sum_{v=1}^K \frac{\sum_{i,j=1}^n u_{iv}^2 u_{jv}^2 d(i, j)}{2 \sum_{j=1}^n u_{jv}^2} \quad (2.30)$$

²¹ Kazım Özdamar, Paket Programlar İle İstatistiksel Veri Analizi,5. Baskı, , Ankara: Nobel Akademik Yayıncılık, 2004, ss.345

Burada u_{iv} üyelik fonksiyonlarını gösterirken $u_{iv} \geq 0$ olmak üzere $\sum_{v=1}^k u_{iv} = 1$ sınırlandırması sağlanmalıdır. Ayrıca $d(i, j)$ ise, i ve j birimleri arasındaki uzaklığı ya da farklılığı ifade etmektedir.

Bu işlemlerden sonra ise akla gelen soru uygun küme sayısının kaç olacağıdır. Bulanık kümelemede en uygun küme sayısının hangisinin olduğunu belirlemek için gölge istatistiği (silhouette statistics, s) ve ortalama gölge istatistiği (average silhouette statistics, SC) verilerinden yararlanır.

Herhangi bir A kümesindeki n adet birimden herhangi bir i . Birimin diğer birimlere olan uzaklıkları a olmak üzere:

$$a = \frac{1}{n} \sum_{j=1}^n d_{ij} , \quad n \in A \quad (2.31)$$

Yine bu A kümesi dışında olan ancak i . Birimin en yakın komşu olduğu ve elemanları arasındaki ortalama farklılığın en az olduğu B kümesindeki i . Birimin diğer elemanlara olan uzaklıklarının ortalaması b olmak üzere:

$$b = \frac{1}{n} \sum_{j=1}^n d_{ij} , \quad n \in B , \quad i \in A \quad (2.32)$$

Burada hesaplanan a ve b değerleri kullanılarak i . Birimin gölge istatistiği olan s şu şekilde hesaplanabilir:

$$n = 1 \quad \text{ise} \quad s = 0$$

$$a < b \quad \text{ise} \quad s = 1 - \frac{a}{b}$$

$$b < a \quad \text{ise} \quad s = \frac{b}{a} - 1$$

$$a = b \quad \text{ise} \quad s = 0 \quad (2.33)$$

Ortalama gölge istatistiği tüm birimler için hesaplanmış gölge istatistiklerinin bir ortalamasıdır ve uygun küme sayısı bulunurken en yüksek değerli ortalama gölge istatistiğinden faydalanılır. Bu değer en azından 0,5 olması beklenir.

Uygun küme sayısı bulunurken ortalama gölge istatistiği kullanıldığı gibi aşağıdaki adı geçen katsayılardan da faydalanılabilir:

Dunn Parçalama Katsayısı: Kümelemenin ne kadar bulanık veya ne kadar katı olduğunu belirleyebilmek adına bu katsayı hesaplanır. Aşağıdaki formülle hesaplanabilir:

$$F(u) = \sum_{i=1}^n \sum_{v=1}^k \frac{u_{iv}^2}{n} \quad (2.34)$$

Bu formülü incelenirse eğer hesaplanan katsayının her zaman $\left[\frac{1}{k}, 1\right]$ aralığında olacağını görebiliriz. Herhangi bir birimin tüm kümelere olan üyeliği eşit ise kümeleme tamamen bulanık olur ve bu katsayı $F(u) = \frac{1}{k}$ olacaktır. Tam tersi olarak ise herhangi bir birimin bazı kümelerde üyeliği 1, diğerlerinde 0 olması durumunda ise kümeleme tamamen katı olacak ve bu katsayı da $F(u) = 1$ olacaktır.

Normalize Dunn Katsayısı: Az önce bahsedilen Dunn katsayısı 0 ile 1 arasında değer alacak şekilde standartlaştırılırsa Normalize Dunn Katsayısı elde edilir. Burada bulunacak bir 0 değeri kümelemenin tamamen bulanık olduğunu ve tam tersi 1 değeri ise güçlü bir kümeleme olduğunu gösterir. Şu şekilde hesaplanabilir:

$$F_n(u) = \frac{F(u) - \frac{1}{k}}{1 - \frac{1}{k}} \quad (2.35)$$

Kaufmann Parçalama ve Normalize Kaufmann Parçalama Katsayısı: Dunn katsayısına alternatif olarak geliştirilen Kaufmann katsayıları şu şekilde hesaplanabilir:

$$D(U) = \frac{1}{n} \sum_{k=1}^K \sum_{i=1}^N (h_{ik} - m_{ik})^2 \quad (2.36)$$

$$D_n(U) = \frac{D(U)}{1 - \frac{1}{K}} \quad (2.37)$$

$D(U)$ deęerinde 0 gulü kümelemeyi gsterir. Ve bu eęer sıfırdan uzaklařtıķça bulanık kmelemeyi gsterecektir.

Uygun kme sayısını bulmak iin ise $F_n(u)$ deęerinin byk, $D_n(U)$ deęerinin ise kk olduęu deęerler tercih edilmelidir.

ÜÇÜNCÜ BÖLÜM

TÜRKİYE'DEKİ İLLERİN GÖÇ GÖSTERGELERİNE GÖRE KÜMELENMESİ

3.1. GÖÇ KAVRAMI ve GÖÇ ÇALIŞMALARI

Göç, bireylerin veya toplulukların bir mevcut yaşadıkları yerden ayrılıp başka bir yere yaşamak amacıyla yer değiştirmesi olarak tanımlanabilir. Göç genel olarak iç göç ve dış göç olarak ikiye ayrılır. İç göç, aynı ülke içindeki nüfus hareketliliği olarak adlandırılırken, dış göç ise ülkeler arası nüfus hareketliliği olarak adlandırılır. Göç nedenleri genellikle sosyal, siyasal, ekonomik ve ailevi olmaktadır. Bununla beraber göçün sonuçları arasında ise işsizlik, aile yapısının bozulması, gecekondulaşma ve buna bağlı olarak altyapı yetersizliği gibi birçok sonuçlar gösterilebilir. Göç, çoğu zaman az gelişmiş yerlerden çok gelişmiş, refah seviyesi yüksek yerlere doğru olmaktadır.

Türkiye'de kırdan kente yoğun göçler ve hızlı kentleşme 1950'den sonra başlamıştır.(Tümertekin,1973,Doğanay,1997,Keleş,2002).Tarımda makineleşme, eğitim ve sosyal imkânların azlığı ve tarım arazilerinin miras yoluyla daha küçük parçalara bölünmesi gibi başlıca nedenler sayılabilir. Bununla beraber Türkiye'de sadece kırdan kente değil, aynı zamanda kentten kente göç de yaşanmaktadır. Buna göre Türkiye'de iç göçün yönü ise doğudaki az gelişmiş illerden batıdaki gelişmiş illere doğru olmuştur. Yani doğu-batı yönlü olmuştur.

Türkiye jeopolitik konumu dolayısıyla asırlardır ticaret ve insan hareketliliği için bir köprü durumundadır. Asya ve Avrupa kıtalarını birbirine bağlaması sebebiyle her zaman bir bağlantı noktası olagelmıştır. Dünya sosyo-ekonomik açıdan birbirinden farklı ülkelerden ve bölgelerden oluşmaktadır. Dolayısıyla dünya üzerindeki sürekli bir göç hareketi kaçınılmazdır. Türkiye ise bir köprü ülke olmasından dolayıdır ki bu göç hareketlerinden çok fazla etkilenmektedir.

Türkiye, yeri geldiğinde transit yani geçiş ülkesi, yeri geldiğinde de çok göç vermiş olması sebebiyle bir kaynak ülke ve yine Anadolu'nun misafirperverliği neticesinde bir sığınılacak ülke pozisyonunda olmuştur. Son yıllarda komşu ülkelerde

yaşanan savaşlar neticesi ile de Türkiye net göç alan bir ülke haline gelmiştir(Dünya Bankası).

Ersöz vd. (2017), çalışmalarında Türkiye’de yaşam endeksi değerlerine göre farklılık veya benzerlik gösteren illeri ortaya koymaya çalışmıştır. Bunu K-ortalamlar kümeleme analizi ile incelemiş ve elde edilen sonuçlar diskriminant analizi ile desteklenmiştir. Neticede yaşam endeksi değerlerine göre Türkiye’deki 81 il sınıflandırılmıştır.

Filiz (2008), yaptığı çalışmada 1980 ve sonrasında Türkiye’de göç yollarında doğu-batı ve kırsal-kent doğrultusunda bir değişiklik olup olmadığını araştırmıştır. Buna ilaveten göçlerin çoğu zaman başlıca sebebi olan ekonomik sebeplerin Türkiye’de geçerli olup olmadığını da araştırmıştır. Göçün nedenlerini NUTS2 düzeyinde Gravity modeli ile tahmin etmiştir. Çalışmanın sonundaki bulgulara göre ise Türkiye’nin göç yollarında bir değişim olmadığı tespit edilmiş ve ülkenin batısının hala bir çekim merkezi olduğu görülmüştür. Ekonomik etkenlerin yanında eğitim, sağlık gibi nedenlerin göçün belirleyicisi olduğu görülmüştür.

Erçevik (2013), yaptığı çalışmada Türkiye’de göç edenlerin sayısı, sosyo-ekonomik nitelikleri yerleşim yerleri bazında dağılımları ile göç nedenleri konularını değerlendirerek 2000 yılında Türkiye’deki iç göç hareketlerinin profilini çıkarmaya çalışmıştır. Araştırmasını göç alan ve göç veren bölge bazında ele almıştır. Bu değişkenleri kontenjans tabloları haline getirdikten sonra bu değişkenlerin aralarındaki ilişkileri Log-Lineer Analiz kullanarak incelemiştir. Çalışmasının sonucunda hem göç alan hem de göç veren bölgeler için 6 adet Log-Lineer model oluşturmuştur.

Başar (2015), çalışmasında Türkiye’de iller bazında alınan ve verilen göçlere etki eden değişkenleri belirlemeye çalışmıştır. Bunun için 2008-2013 arasındaki iller bazındaki verilerle yatay kesit regresyon analizi gerçekleştirmiştir. Bu analizleri sonucunda iş gücüne katılımın, kişi başı sanayi elektrik tüketiminin, okul, sağlık kurumu ve ceza infaz kurumu sayısının iç göç üzerinde etkili olduğunu belirlemiştir. Ele aldığı birçok faktörün verilen göçlerden ziyade alınan göçleri açıkladığını görmüş ve buna göre illerin çekicilik gücünün iticilik gücünden çok daha etkili olduğu görülmüştür.

Tatlı (2016), Türkiye’de iller bazında göçü etkileyen faktörlerin mekânsal özellik gösterip göstermediğini incelemiştir. Bunun için göç dağılım haritasını incelemiş ve kuzeydoğu güneybatı yönlü bir kümelenme olduğunu görmüştür. Buna ek olarak kümelenme görülen yerlerde mekânsal etkinin anlamlı olduğu illeri LISA istatistiği ile belirlemiştir. Bunlara uygun mekânsal modeller belirlemiştir. Elde ettiği sonuçların iktisaden beklenen yönde olduğunu görmüştür.

Demirtaş (2017), çalışmasında 2011 yılındaki Türkiye nüfus verilerini kullanarak iç göçler demografik değişkenler açısından yapay sinir ağları ile test etmiştir. Girdi değişkeni olarak yaş, eğitim durumu vb. kullanılırken girdi değişkeni olarak göç etme nedenini kullanmıştır. Yapay sinir ağları yaptığı model çözümlemesinin doğruluk oranını yaklaşık %78 bulmuştur. Göç değişkenleri ile demografik değişkenlerin arasındaki ilişkilerde en tutarlı sonucu yaş değişkeninin verdiği tespit etmiştir.

Tunç (2018), çalışmasında 2008-2016 yıllarında Türkiye’deki iç göçleri Shift Share analizi ile incelemiştir. Gerçek büyümenin en fazla olduğu bölgenin Karadeniz bölgesi olduğunu görmüştür.

Şen (2014), çalışmasında Türkiye’deki iç göçlerin neden ve sonuçlarını incelemiştir. Ülke ilk kurulduğunda az olan göçün ilerleyen yıllarda bir olgu haline geldiğini söylemiştir. Tarımsal yeniliklerle itici hale gelen kırsal ve sanayileşme ile çekici hale gelen kent, yaşanan bu göçü açıklamaktadır.

Şantaş (2019), Türkiye’de iç göçün dağılımını belirlemeye çalışmıştır. 1993-1998-2003-2008-2013 yıllarındaki Türkiye Nüfus ve Sağlık Araştırması verilerini kullanmıştır. Çalışmasının sonucuna göre kent nüfusunun sürekli arttığını, ortaokul ve daha üstü öğrenim düzeyine sahip kadınların %60’ından fazlasının kentlerde yaşadığını tespit etmiştir. Buna ilaveten kentlerde yaşama oranının bekâr, çalışmayan, sağlık sigortası olanlar ile Batı bölgelerde daha yüksek olduğunu tespit etmiştir.

Yılmaz (2015), çalışmasında 1980-2012 yılları arasındaki Türkiye’de kırsal nüfusun değişimini ve illere göre dağılımını incelemiştir. 1980 yılından sonra kırsal nüfusun hem miktar olarak hem de ülke nüfusu içindeki oranının azalma eğiliminde olduğunu tespit etmiştir. Ülkedeki toplam kırsal nüfusun değişimini başta göç olmak

üzere yeni il ve ilçelerin kurulmasıyla değişen belediye sınırları gibi birçok faktörün etkilediğini söylemiştir.

3.2. TÜRKİYE’DE İÇ GÖÇ

Tüm dünyada olduğu gibi Türkiye’de de iç göç ezelden beri hep olagelmıştır. İç göç ülkenin sınırları içerisinde, iller arasında veya illerden daha alt yerleşim yerleri arasında yaşanan insan hareketliliği olarak tanımlanabilir. İç göçün yaşanmasında çok çeşitli sebepler vardır. Bunlar arasında en başta ekonomik olmak üzere demografik, sosyal, kültürel, siyasal ve diğer nedenler şeklinde sıralanabilir.

Türkiye’de iç göç serüvenini genel olarak 4 bölümde incelemek mümkündür. Bunlardan ilki cumhuriyetin ilanından 1945-1950’li yıllara kadar olan zaman dilimidir. Bu dönemde ülkemizde önemli ölçüde bir iç göç hareketliliğinden söz etmek çok da mümkün değildir. Bunun nedenleri olarak ise, ülkenin henüz birçok safhada, en başta ulaşım ve eğitim gibi alanlarda çok da gelişmiş olmaması, halkın çoğunlukla köylerde yaşaması ve devlet politikası olarak da “köylü milletin efendisidir” şeklinde bir yapının etki yaptığı söylenebilir.

İkinci dönem olarak 1945-50’li yıllardan 1980’e kadar olan zaman dilimini söylemek mümkündür. 1945’te 2. Dünya savaşının bitmesi ve yine 1950’de Demokrat Parti’nin iktidara gelmesiyle bir önceki döneme göre iç göç olgusunda farklılıklar yaşanmıştır.

Savaşın sona ermesinin ardından birçok ülkede olduğu gibi özellikle Türkiye’de kırdaki yaşayan toplum gittikçe fakirleşmiş ve 1950’de değişen iktidarla beraber değişen ekonomi politikalarıyla beraber adeta insanlar kır yaşamından kopmuş veya kopmaya teşvik edilmiştir. Bunda tabii ki kır yaşamındaki yetersiz imkânlar, tarımda makineleşme, kırdaki yaşamın zorluğu gibi sebepler sayılabilir.

Üçüncü dönem ise 1980’den 2008’e kadar olan zaman dilimidir. 1980’de gerçekleşen askeri darbe ve sonrasındaki serbest ekonomi politikaları ve ayrıca tüm dünyada hızla yayılan küreselleşme süreçleri ülkedeki iç göçü hızlandırmıştır. Ayrıca özellikle 1980’lerde başlayıp, maalesef hala günümüze kadar da bir türlü bitmeyen doğu ve güneydoğuda patlak veren terör olayları da buralarda yaşayan insanları adeta göç etmeye zorlamıştır.

Ayrıca sanayileşmenin de artmasıyla kentlere olan göçler hızlanmış ve kentlerin yapıları çok değişmiştir. Ülkedeki kentleşme hızı sanayileşme hızının önüne de geçtiğinden özellikle çok göç alan büyük şehirlerde başta altyapı, çarpık kentleşme olmak üzere birçok sorun baş göstermiştir.

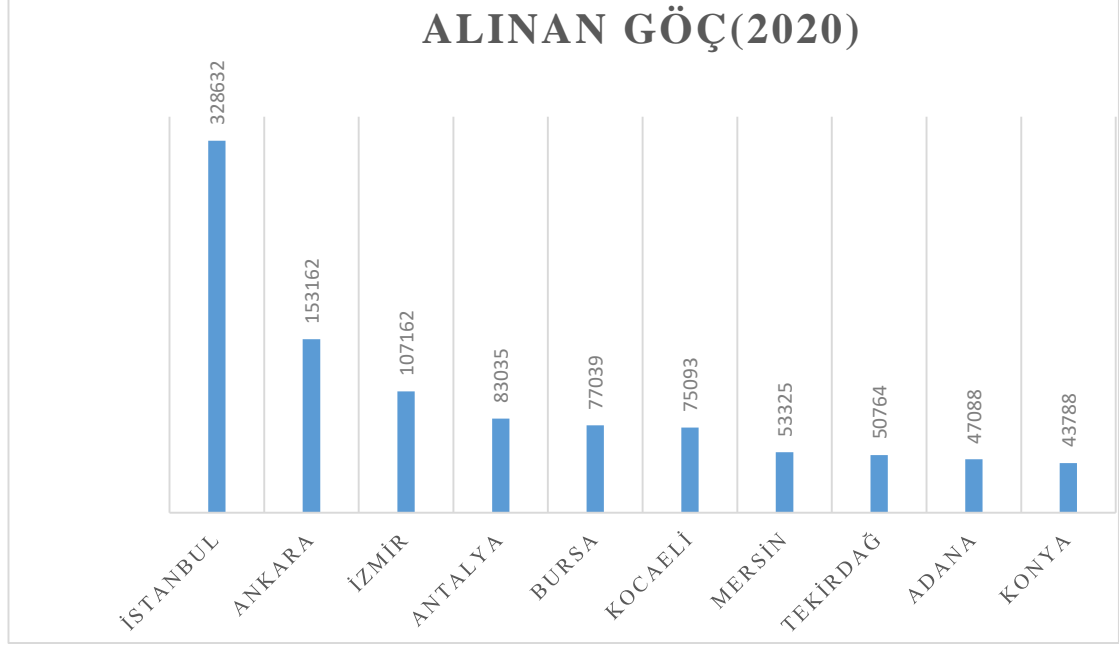
2008’de adrese dayalı nüfus sayımı sisteminde geçilmesiyle birlikte ülkede iç göç olgusunda dördüncü dönem başlamıştır. Bu dönemde de kırdan kente ve kentten kente göç etkili olmuştur. Bunun nedenler arasında özellikle yerleşim yerleri arasındaki imkân farklılıkları veya dengesizlikleri gösterilebilir. Aşağıdaki tabloda Türkiye’nin 1975-2020 yılları arasındaki nüfusu, iç göçü ve bu iç göçün nüfusa oranı gösterilmiştir.

YILLAR	TÜRKİYE NÜFUS	İÇ GÖÇ	İÇ GÖÇ ORANI(%)
1975-1980	38 395 730	2 700 977	7,034576501
1980-1985	44 078 033	2 885 873	6,547190978
1985-1990	49 986 117	4 065 173	8,132604099
1995-2000	60 752 995	4 788 193	7,88141062
2007-2008	71 517 100	2 273 492	3,178948811
2008-2009	72 561 312	2 236 981	3,082883893
2009-2010	73 722 988	2 360 079	3,201279633
2010-2011	74 724 269	2 420 181	3,238815223
2011-2012	75 627 384	2 317 814	3,064781402
2012-2013	76 667 864	2 534 279	3,305529681
2013-2014	77 695 904	2 681 275	3,450986297
2014-2015	78 741 053	2 720 438	3,454916967
2015-2016	79 814 871	2 619 403	3,281848316
2016-2017	80 810 525	2 684 820	3,322364259
2017-2018	82 003 882	3 057 606	3,728611287
2018-2019	83 154 997	2 806 123	3,3745693
2019-2020	83 614 362	2 258 726	2,70136128

Tablo 4: 1975-2020 Arasında Türkiye Nüfusu ve İç Göç İstatistikleri (TÜİK)

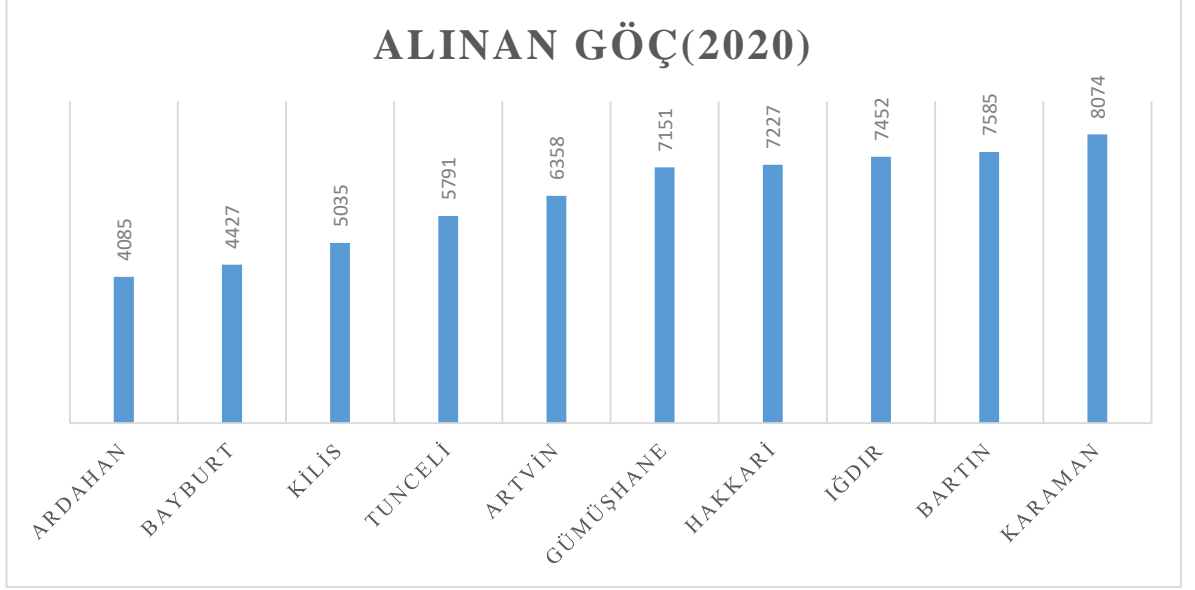
Yukarıdaki tabloya bakıldığında ülke nüfusunun sürekli arttığı görülmektedir. Bununla beraber ülke içi göç edenleri sayısı bazı yıllar artıp bazı yıllar azalmaktadır. Bunun yanında ise özellikle 2008 yılından sonra yıllık olarak yapılan nüfus sayımına göre iç göç oranının her yıl yaklaşık olarak ülke nüfusunun %3’ü kadar olduğu görülmektedir. 2020 yılı itibariyle de 2 milyondan fazla kişinin ülke içinde göç ettiği görülmektedir. Bu oran hiç de az değildir. Burada aslolan ise bu iç göçü engellemek ya

da bitirmek değil kontrollü bir şekilde yönetilebilir hale getirmektir. Aşağıdaki grafiklerde 2020 yılında Türkiye'nin en çok ve en az göç alan ilk 10 ili ile en çok ve en az göç veren ilk 10 ili gösterilmektedir.



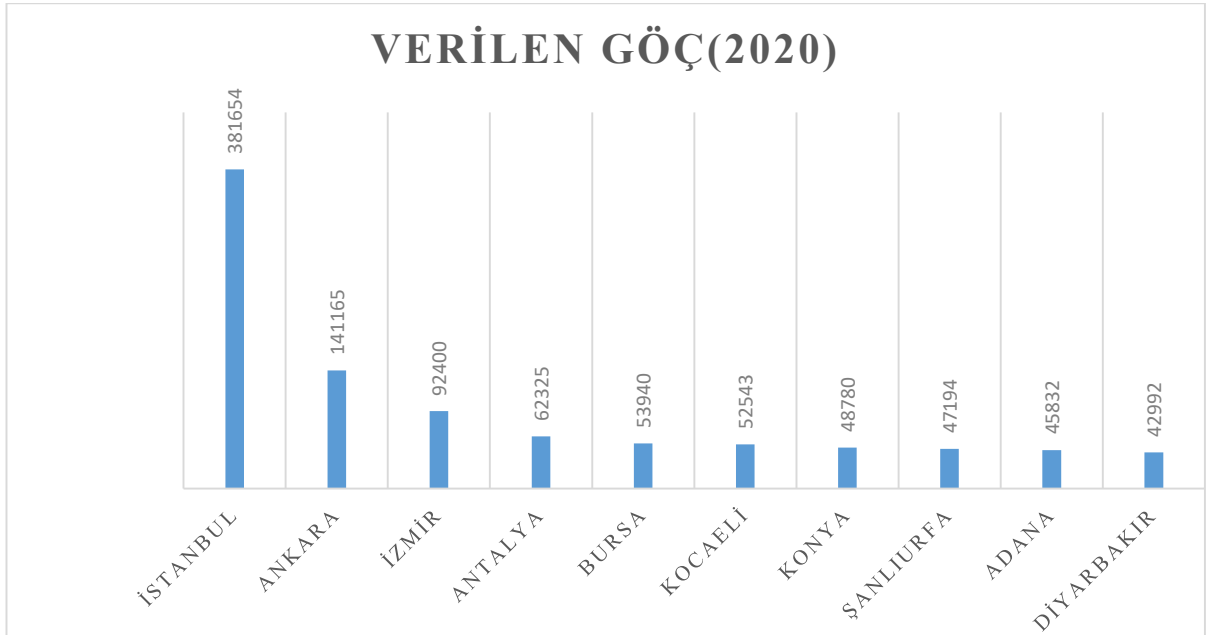
Grafik 1: 2020 Yılında En Fazla Göç Alan İller (TÜİK)

Grafik 1'e bakıldığında 2020 yılı itibariyle ülkede toplam göç eden yaklaşık 2,25 milyon insanın yaklaşık %15'i yani yaklaşık 328000 kişinin İstanbul'a göç ettiği görülmektedir. Grafikteki tüm illere bakacak olursak bunların tamamının ülkenin batısında bulunan daha gelişmiş büyük şehirler olduğunu söylemek mümkündür. Ayrıca bu şehirlerin gerek nüfus olarak gerekse sanayi, istihdam vb. açılardan da gelişmiş olduklarını söyleyebiliriz. Bu ise ekonomik büyüklüğün illeri göç anlamında birer cazibe merkezi haline getirdikleri söylenebilir.



Grafik 2: 2020 Yılında En Az Göç Alan İller(TÜİK)

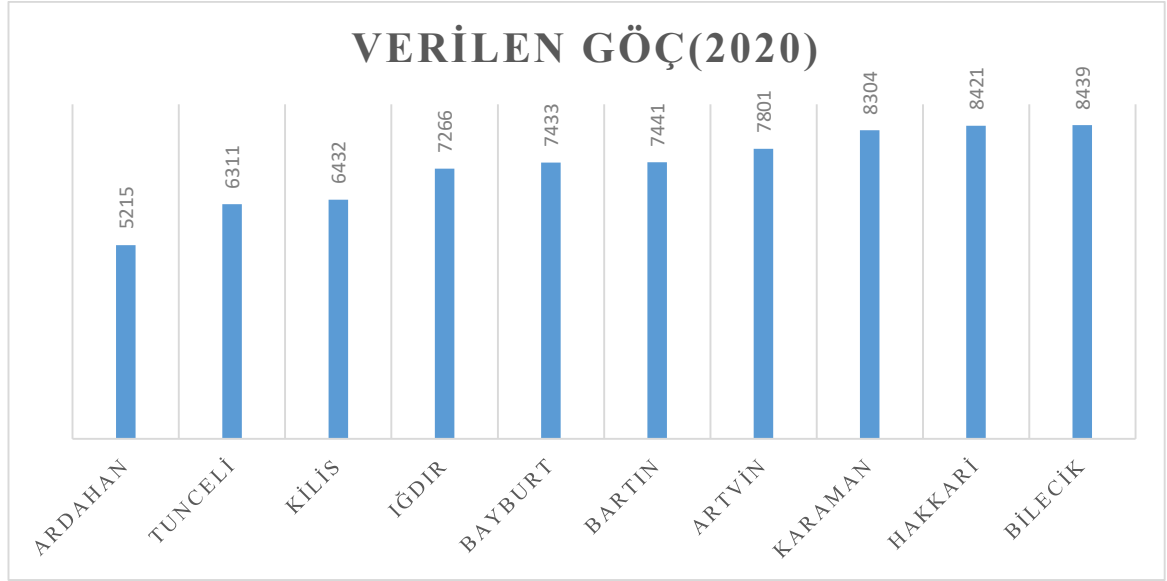
Grafik 2'ye bakılırsa 2020 yılında en az göç alan il 4085 kişi ile Ardahan'dır. En çok göç alan illerin aksine en az göç alan iller çoğunlukla ülkenin doğu bölgelerindeki gelişmemiş ve istihdam, sosyal hayatın vb. çok sınırlı olduğu ve nüfusun da az olduğu iller olduğu söylenebilir.



Grafik 3: 2020 Yılında En Çok Göç Veren İller(TÜİK)

Grafik 3'e bakılırsa 2020 yılı itibariyle en çok göç veren il açık ara farkla 381654 kişiyle İstanbul olmuştur. Bu rakam toplam iç göçün yaklaşık %17'sidir. Ülkenin en

büyük metropol şehri olan İstanbul hem en çok göç alan hem de aynı zamanda en çok göç veren il olmuştur. Grafikteki diğer illere de bakılırsa, ilk 5 sıradaki iller, aynen en çok göç alan iller grafiğindeki illerin aynısıdır. Bu 6 büyükşehir zaten nüfus olarak ülkedeki en büyük ilk 6 şehirdir. Özellikle Adana, Şanlıurfa ve Diyarbakır şehirlerinin burada olmasının nedeni olarak mevsimlik işçi göçlerinin nedeni olarak söylenebilir.



Grafik 4: 2020 Yılında En Az Göç Veren İller(TÜİK)

Grafik 4'e bakılırsa 2020 yılında en az göç veren il 5215 kişiyle Ardahan olmuştur. Ardahan'ın hem en az göç alan hem de en az göç veren il olmasında nüfusunun az olması yadsınamaz bir gerçektir. Yine grafikteki diğer illere bakılırsa, onların da çoğunluğunun ülkenin doğusunda yer aldığını, ayrıca birçok açıdan çok da gelişmemiş iller olduklarını söylemek mümkün olacaktır. Ayrıca bu grafikteki tüm illerin az nüfuslu oldukları söylenebilir.

Aşağıdaki tabloda Türkiye'nin 2008-2020 yılları arasında net göç hızı en yüksek olan 10 ili gösterilmiştir.

2008			2009			2010		
İller	Net Göç	Net Göç Hızı(%)	İller	Net Göç	Net Göç Hızı(%)	İller	Net Göç	Net Göç Hızı(%)
Yalova	10114	52,6	Çankırı	8335	46,1	Tekirdağ	11874	15

Tekirdağ	25161	33,2	Trabzon	10394	13,7	Antalya	25245	12,8
Antalya	36225	19,7	Eskişehir	9121	12,2	Ankara	49405	10,4
Kocaeli	23018	15,6	Tekirdağ	8589	11	Kocaeli	15124	9,7
Bursa	35594	14,3	Yalova	2036	10	Erzincan	1833	8,2
Muğla	10948	13,9	Bolu	2684	9,9	İstanbul	102583	7,8
Eskişehir	9761	13,3	Antalya	17064	8,9	Muğla	6279	7,7
Bartın	2093	11,4	Düzce	2706	8,1	Eskişehir	4882	6,4
Tunceli	793	9,2	Ankara	37079	8	Kayseri	7462	6,1
Aydın	8798	9,2	Kocaeli	12033	7,9	Bursa	15420	5,9
2011			2012			2013		
İller	Net Göç	Net Göç Hızı(%)	İller	Net Göç	Net Göç Hızı(%)	İller	Net Göç	Net Göç Hızı(%)
Tekirdağ	13645	16,6	Çankırı	6441	35,6	Gümüşhane	5515	39,8
Antalya	26856	13,2	Ordu	21645	29,6	Tunceli	2846	33,9
Ankara	54479	11,2	Karabük	5066	22,8	Çankırı	5299	28,2
Gümüşhane	1438	10,9	Çanakkale	8878	18,2	Yalova	4862	22,3
Eskişehir	7137	9,2	Tekirdağ	14113	16,7	Tekirdağ	13632	15,7
İstanbul	121782	9	Gümüşhane	2165	16,1	Çanakkale	6528	13,1
Kocaeli	13244	8,3	Isparta	5401	13,1	Kastamonu	4253	11,6
Muğla	5805	7	Bolu	3459	12,4	Antalya	24530	11,4
Bursa	15985	6,1	Burdur	2976	11,8	Sinop	2216	10,9
Malatya	3410	4,5	Bilecik	2269	11,2	Kocaeli	16417	9,8
2014			2015			2016		
İller	Net Göç	Net Göç Hızı(%)	İller	Net Göç	Net Göç Hızı(%)	İller	Net Göç	Net Göç Hızı(%)
Bayburt	3329	42,2	Tekirdağ	20545	22,1	Bayburt	10328	121,5
Tekirdağ	21728	24,3	Karabük	4450	19	Gümüşhane	8338	49,7
Muğla	18548	21	Gümüşhane	2700	18	Giresun	15092	34,5
Gümüşhane	2740	18,9	Kocaeli	30231	17,1	Tekirdağ	24246	25,2
Yalova	3616	16,1	Antalya	28067	12,3	Bolu	6766	22,8
Bahkesir	17633	14,9	Yalova	2337	10,1	Ordu	15766	21,2
Kocaeli	24637	14,4	Ankara	51047	9,7	Kocaeli	25123	13,8
Çanakkale	6889	13,6	Kastamonu	3509	9,5	Karabük	3303	13,7
Aydın	13446	13	Eskişehir	7743	9,4	Düzce	4839	13,2
Antalya	28426	12,9	Bolu	2280	7,9	Eskişehir	10902	13
2017			2018			2019		
İller	Net Göç	Net Göç Hızı(%)	İller	Net Göç	Net Göç Hızı(%)	İller	Net Göç	Net Göç Hızı(%)
Tekirdağ	20733	20,8	Çankırı	28027	138,5	Bayburt	1249	14,8
Yalova	5035	20,2	Tunceli	5327	62,3	Antalya	31646	12,7
Kocaeli	27538	14,7	Sinop	11129	52	Tekirdağ	10802	10,3
Eskişehir	10261	12	Gümüşhane	7569	47,6	Ankara	57280	10,2

Bilecik	2385	10,8	Rize	14853	43,5	Kocaeli	17302	8,9
Çanakkale	5548	10,5	Artvin	7058	41,4	Eskişehir	7214	8,2
Muğla	9161	9,8	Ordu	24661	32,5	İstanbul	120371	7,8
Düzce	3636	9,7	Giresun	14405	32,2	Muğla	6777	6,9
Uşak	3489	9,6	Sivas	19590	30,8	Yalova	1707	6,3
Şırnak	4486	9	Bartın	4586	23,3	Bursa	17240	5,7
2020								
İller	Net Göç	Net Göç Hızı(‰)						
Tekirdağ	22296	20,8						
Yalova	4380	16						
Muğla	13380	13,5						
Kocaeli	22550	11,4						
Antalya	20710	8,2						
Bursa	23099	7,5						
Balıkesir	9154	7,4						
Ordu	5492	7,2						
Mersin	13083	7						
Kırklareli	2256	6,3						

Tablo 5: Net Göç Hızı En Çok Olan İller (2008-2020)(TÜİK)

Verilen ve alınan göç sayılarına bakmanın yanı sıra illerin net göç hızlarına da bakmak faydalı olacaktır. Tablo 5'e bakıldığında 2008 yılında en çok net göç hızı Yalova'ya aittir ve yine ondan sonra gelen iller Tekirdağ, Antalya, Kocaeli ve Bursa olmuştur. Antalya hariç bu şehirler genel olarak istihdamın fazla olduğu sanayinin gelişmiş olduğu illerdir denilebilir. Yine ilerleyen yıllara bakıldığında Bayburt, Gümüşhane, Şırnak, Uşak, Tunceli gibi illerin de sıralamaya girdiğini görülmektedir. Bunların sebebi olarak ise bu illere yapılan yatırım ve teşviklerin pozitif net göç hızlarının yüksek olmasında etkili olduğu söylenebilir. İstanbul hem alınan hem de verilen göç sıralamasında birinci il olmasına rağmen, söz konusu net göç hızı olduğunda çoğu seneler ilk 10 il arasında olmadığı görülmektedir. Buna sebep bu ilin birçok seneler aldığı göç kadar verdiği göçün de çok olması ve ayrıca bu ilimizin yine ülkemizdeki nüfusu en çok olan il olması da yine çok etkili olmuştur. Ayrıca tablonun tamamına bakıldığında hemen hemen her yıl listedeki sıralamanın ve illerin değiştikleri gözlenmektedir. Bu ise her yıl illerin çok farklı göç durumları yaşadıklarını göstermektedir.

Aşağıdaki tabloda Türkiye'nin 2008-2020 yılları arasındaki net göç hızları en düşük olan 10 ili gösterilmiştir.

2008			2009			2010		
İller	Net Göç	Net Göç Hızı(‰)	İller	Net Göç	Net Göç Hızı(‰)	İller	Net Göç	Net Göç Hızı(‰)
Muş	-15838	-38,4	Ardahan	-3258	-29,7	Çankırı	-6422	-35,2
Erzurum	-24586	-31,2	Mardin	-22012	-29,4	Yozgat	-13437	-27,9
Yozgat	-14765	-30	Tunceli	-2105	-25	Tokat	-15565	-24,9
Ağrı	-15255	-28,2	Muş	-10024	-24,5	Kırıkkale	-6741	-24,1
Kars	-8871	-28	Kars	-6632	-21,4	Kars	-6751	-22,1
Bitlis	-9236	-27,9	Ağrı	-10498	-19,3	Ardahan	-2271	-21,3
Bayburt	-1953	-25,5	Bayburt	-1319	-17,5	Siirt	-5062	-16,7
Ardahan	-2807	-24,7	Çorum	-9027	-16,6	Erzurum	-12417	-16
İğdır	-3797	-20,4	Yozgat	-7841	-16	Çorum	-8398	-15,6
Kırıkkale	-5245	-18,6	İğdır	-2807	-15,2	Niğde	-5064	-14,9
2011			2012			2013		
İller	Net Göç	Net Göç Hızı(‰)	İller	Net Göç	Net Göç Hızı(‰)	İller	Net Göç	Net Göç Hızı(‰)
Van	-48858	-46,7	Ağrı	-15128	-27	Tokat	-20084	-33
Yozgat	-11670	-24,8	Muş	-10732	-25,6	Ağrı	-13752	-24,6
Çankırı	-3740	-20,9	Yozgat	-9879	-21,6	Kars	-7026	-23,1
Kırıkkale	-5292	-19,1	Kars	-6479	-21	Ardahan	-2379	-22,9
Kars	-5722	-18,5	Siirt	-5805	-18,5	Yozgat	-9884	-22
Trabzon	-13588	-17,8	Bitlis	-5888	-17,3	Muş	-9123	-21,9
Ağrı	-9674	-17,3	Hakkâri	-4515	-16	Erzurum	-16599	-21,4
Adıyaman	-10066	-16,8	Kilis	-1760	-14,1	Ordu	-15540	-21
Ardahan	-1765	-16,3	Adıyaman	-8268	-13,8	İğdır	-2897	-15,1
Niğde	-5254	-15,4	Zonguldak	-8408	-13,8	Bitlis	-4860	-14,3
2014			2015			2016		
İller	Net Göç	Net Göç Hızı(‰)	İller	Net Göç	Net Göç Hızı(‰)	İller	Net Göç	Net Göç Hızı(‰)
Çankırı	-8561	-45,6	Kars	-8481	-28,6	Hakkâri	-10122	-37,1
Kars	-9740	-32,3	Ağrı	-15577	-28,1	Ağrı	-16505	-30
Ağrı	-16515	-29,6	Yozgat	-11871	-27,9	Şırnak	-13020	-26,6
Ardahan	-2710	-26,5	Muş	-11103	-26,8	Tunceli	-2209	-26,5
Yozgat	-11522	-26,3	Bayburt	-2131	-26,8	Muş	-10374	-25,2
Muş	-10611	-25,5	Şırnak	-12061	-24,3	Kars	-6381	-21,8

Erzurum	-	17215	-22,3	Ardahan	-2172	-21,6	Mardin	-	17384	-21,6
Kırkkale	-5438	-19,9	Bitlis	-6865	-20	Van	-	20133	-18,1	
Bitlis	-6023	-17,7	Çankırı	-3446	-18,9	Kilis	-2303	-17,5		
Çorum	-8777	-16,5	Siirt	-5661	-17,5	Ardahan	-1716	-17,3		
2017			2018			2019				
İller	Net Göç	Net Göç Hızı(%)	İller	Net Göç	Net Göç Hızı(%)	İller	Net Göç	Net Göç Hızı(%)		
Bayburt	-	10172	-119	Ağrı	-9391	-17,3	Çankırı	-	20894	-101,3
Gümüşhane	-8553	-49	İstanbul	-	210321	-13,9	Gümüşhane	-9319	-55,1	
Ağrı	-	17391	-31,9	Muş	-5047	-12,3	Ordu	-	21254	-27,8
Muş	-9149	-22,4	Van	-10263	-9,1	Rize	-7588	-21,9		
Giresun	-9146	-20,7	Gaziantep	-17502	-8,6	Ağrı	-	11816	-21,8	
Ardahan	-1870	-19,1	Adana	-18978	-8,5	Kırkkale	-5917	-20,7		
Kars	-5531	-19	Ankara	-37365	-6,8	Hakkâri	-5821	-20,5		
Bitlis	-6108	-17,7	Kayseri	-9357	-6,7	Sivas	-	12858	-19,9	
Ordu	-	12194	-16,3	Şanhurfa	-12300	-6	Tunceli	-1662	-19,4	
Van	-	16298	-14,6	Iğdır	-897	-4,5	Artvin	-3273	-19	
2020										
İller	Net Göç	Net Göç Hızı(%)								
Gümüşhane	-	11965	-81							
Bayburt	-3006	-36								
Tokat	-	16906	-27,9							
Karabük	-4602	-18,7								
Ağrı	9727	-18								
Kars	4742	-16,5								
Kırkkale	4424	-15,7								
Çankırı	-2659	-13,7								
Erzurum	9815	-12,9								
Siirt	3914	-11,8								

Tablo 6: Net Göç Hızı En Düşük Olan İller (2008-2020)(TÜİK)

Tablo 6'ya bakıldığında net göç hızı en düşük olan illerin birçoğunun ülkenin doğu, güneydoğu ve kuzeydoğu bölgelerindeki iller olduğunu söylemek mümkündür. En başta da belirtildiği gibi bu illerin iklim koşulları, istihdam eksikliği, yatırımların nispeten daha az olması gibi sebeplerden dolayı daha fazla göç verdikleri söylenebilir. Ayrıca

birkaç istisna hariç Tablo 6'daki illerin tamamı az nüfuslu illerdir. Hemen hemen her yıl listedeki illerin sıraları ve bazen de iller değişmektedir. Bu ise iller bazında her yıl göç durumlarının değişiklik gösterdiği söylenebilir.

3.3. PYTHON PROGRAMLAMA DİLİ

Python, 1990 yılında Guido Van Rossum tarafından geliştirilmeye başlanan bir programlama dilidir. Özellikle veri alanında ve yazılım geliştirme noktasında ön plana çıkmaktadır. Python standart kütüphanesi başta olmak üzere diğer birçok kütüphanesi açık kaynak kodlu olarak ücretsiz indirilebilir. Nesne yönelimli ve yorumlanabilen bir programlama dilidir. Python dilinin Temmuz 2021 itibariyle 3.9.6. sürümü kullanılabilir. Geliştirilmeye başladığından bugüne kadar gelişimini sürdüren Python Linux, MacOS ve Windows dâhil Unix türevlerinde çalışabilen bir dildir. Ayrıca Jupyter Notebook, Anaconda, Google Colab gibi platformlar üzerinden de erişim sağlanarak kullanılabilir.

Python bünyesinde bulunan kütüphaneler “pip install [kütüphane adı]” yazılarak kurulabilmektedir. Ayrıca kurulan kütüphane ise “import [kütüphane adı] as [kullanılan kısaltma]” yazılarak çağrılabilir.

Python programlama dili popülerliğini gittikçe yükseltmektedir. Öyle ki Mart 2021 itibariyle PYPL popülerlik indeksinde 1. Sıraya, Tiobe indeksinde ise 3. Sıraya kadar yükselmiştir.²²Python kodlarını yorumlamak ve değerlendirmek diğer dillere göre daha kolaydır. Diğer dillerde sıklıkla kullanılan noktalama işaretleri ve parantezler gibi kurallar burada olmadığı için kullanımı ve yapısı çok daha sadedir.

Python veri bilimi, veri analizi makine öğrenmesi ve yapay zekâ gibi alanlarda çok gelişmiş kütüphaneleri mevcuttur. Bunların yanı sıra yine birçok kütüphane vardır. Özellikle ileri düzey matematiksel işlemleri de yapmaya yardımcı olan *NumPy* Kütüphanesi vardır. İstatistiksel işlemler ve veri ile ilgili matematiksel işlemler yapabilmek için ideal bir kütüphanedir. Özellikle veri bilimi üzerine çalışanlar ve programlama ile uğraşanlar için ideal bir kütüphane olduğu söylenebilir.

²² <https://www.kdnuggets.com/2021/05/top-programming-languages.html> 16.07.2021

Python içerisinde bir başka kütüphane ise *Matplotlib* kütüphanesidir. Özellikle veri bilimi çalışmalarında matematiksel işlemlerin sonuçlarını iki ya da üç boyutlu olarak görselleştirme sağlar. Basit uygulama ve zengin içeriği sayesinde sıkça tercih edilmektedir.

Bir diğer kütüphane olan *SciKit-Learn* özellikle makine öğrenmesi ve veri bilimi çalışmalarında sıklıkla kullanılmaktadır. Sınıflandırma, regresyon ve kümeleme gibi birçok işlem bu kütüphane yardımıyla kolay ve hızlı bir şekilde yapılabilir. Bu kütüphane *NumPy* ve *Matplotlib* ile birlikte kullanıldığından bu kütüphanelerin kullanımını da bilmekte fayda vardır.

Pandas kütüphanesi ise verileri organize etmede kullanılan çok önemli bir kütüphanedir. Özellikle *numpy* kütüphanesinin yetersiz kaldığı durumlarda *Pandas* kütüphanesi eksiği tamamlamaktadır.

Bu çalışmada Python programlama dilindeki *pandas*, *numpy*, *scikit-learn* ve *matplotlib* kütüphaneleri kullanılmıştır.

Şantaş(2019) Türkiye’de İç Göçün Dağılımı isimli çalışmasında 2018 yılı göç istatistiklerinden alınan göç, verilen göç, net göç ve net göç hızı değişkenlerini kullanmıştır. Özgen ve Erbaş(2018) Türkiye’de İç Göç Hareketlerinin Eğitim Seviyesine Göre Kümelenmesi isimli çalışmasında göç istatistiklerinden alınan göç ve verilen göç değişkenlerini kullanmıştır. Sönmez ve Er (2006) Türkiye’de İllere Göre İç Göç Hareketlerinin Modern Kümeleme Teknikleri İle İncelenmesi isimli çalışmasında göç istatistiklerinden alınan göç ve verilen göç değişkenlerini kullanmışlardır. Literatürde yapılan bu araştırmalardan hareketle bu çalışmada değişkenler olarak alınan göç(ag), verilen göç (vg), net göç (ng) ve net göç hızı (ngh) değişkenleri kullanılmıştır.

3.4. TEZİN AMACI ve ÇALIŞMADA KULLANILAN VERİLER

Daha öncede belirtildiği gibi çalışmanın amacı Türkiye’deki 81 ilin göç istatistikleri dikkate alınarak illerin Kümeleme Analizi ile kümelenmesinin ortaya konulmasıdır. Bunu gerçekleştirmek için denetimsiz öğrenme yöntemlerinden biri olan K-ortalama algoritması Python yazılım dili kullanılarak uygulanmıştır.

Çalışmanın amacını gerçekleştirmek için çalışmada TÜİK tarafından yayınlanan Türkiye’deki illerin 2008-2020 yıllarına ait “Alınan Göç (ag)”, “Verilen Göç (vg)”, “Net Göç (ng)” ve “Net Göç Hızı (ngh)” olmak üzere dört ayrı göç göstere kullanılmıştır. Ele alınan bu dört gösterge için ayrı ayrı kümeleme analizi yapılmıştır. Analizlerin yapılmasında makine öğrenmesi yöntemlerinden de biri olan K-ortalamlar algoritması Python yazılım dili kullanılarak uygulanmıştır.

Aşağıdaki 4 adet tabloda kullanılacak olan niteliklerin bazı temel istatistikleri yer almaktadır. Veri setindeki tüm değerler sayısaldir ve eksik değer bulunmamaktadır. Fakat veriler birbirlerinden çok farklı değerler aldıklarından dolayı bunlar standardize edilecektir.

<i>ag2008</i>		<i>ag2009</i>		<i>ag2010</i>		<i>ag2011</i>	
Ortalama	28067,8	Ortalama	27617,0	Ortalama	29136,8	Ortalama	29878,8
Ortanca	15828,0	Ortanca	15861,0	Ortanca	15311,0	Ortanca	15508,0
Standart Sapma	46011,0	Standart Sapma	47311,6	Standart Sapma	52851,8	Standart Sapma	54269,4
Basıklık	41,5	Basıklık	43,7	Basıklık	46,9	Basıklık	46,7
Çarpıklık	5,9	Çarpıklık	6,1	Çarpıklık	6,4	Çarpıklık	6,4
Aralık	371872,0	Aralık	385366,0	Aralık	435531,0	Aralık	446448,0
En Küçük	2996,0	En Küçük	3101,0	En Küçük	3984,0	En Küçük	3997,0
En Büyük	374868,0	En Büyük	388467,0	En Büyük	439515,0	En Büyük	450445,0
<i>ag2012</i>		<i>ag2013</i>		<i>ag2014</i>		<i>ag2015</i>	
Ortalama	28615,0	Ortalama	31287,4	Ortalama	33102,2	Ortalama	33585,7
Ortanca	15343,0	Ortanca	17875,0	Ortanca	17966,0	Ortanca	18872,0
Standart Sapma	46523,7	Standart Sapma	52596,6	Standart Sapma	54007,1	Standart Sapma	55436,9
Basıklık	44,1	Basıklık	46,2	Basıklık	41,4	Basıklık	42,6
Çarpıklık	6,1	Çarpıklık	6,3	Çarpıklık	5,9	Çarpıklık	6,0
Aralık	380871,0	Aralık	433135,0	Aralık	433675,0	Aralık	447777,0
En Küçük	3664,0	En Küçük	4787,0	En Küçük	5323,0	En Küçük	5630,0
En Büyük	384535,0	En Büyük	437922,0	En Büyük	438998,0	En Büyük	453407,0
<i>ag2016</i>		<i>ag2017</i>		<i>ag2018</i>		<i>ag2019</i>	
Ortalama	32338,3	Ortalama	33145,9	Ortalama	37748,2	Ortalama	34643,5
Ortanca	18990,0	Ortanca	19074,0	Ortanca	26976,0	Ortanca	18126,0
Standart Sapma	46193,7	Standart Sapma	51186,4	Standart Sapma	47397,6	Standart Sapma	60064,7
Basıklık	36,7	Basıklık	40,7	Basıklık	37,4	Basıklık	46,0
Çarpıklık	5,5	Çarpıklık	5,8	Çarpıklık	5,5	Çarpıklık	6,3
Aralık	364760,0	Aralık	411303,0	Aralık	378140,0	Aralık	493512,0

En Küçük	4822,0	En Küçük	5284,0	En Küçük	7342,0	En Küçük	5164,0
En Büyük	369582,0	En Büyük	416587,0	En Büyük	385482,0	En Büyük	498676,0
ag2020							
Ortalama	27885,5						
Ortanca	15347,0						
Standart Sapma	41434,6						
Basıklık	35,7						
Çarpıklık	5,4						
Aralık	324547,0						
En Küçük	4085,0						
En Büyük	328632,0						

Tablo 7: Alınan Göç Veri Setindeki Kısaltmalar ve Bazı Temel İstatistikler

Tablo 7'ye bakıldığında 2008-2020 yılları arasında alınan göç ortalamasının dalgalı bir seyir izlemekle birlikte çok da değişmediği görülmektedir. Bu durum çoğunlukla diğer istatistikler için de geçerlidir.

vg2008		vg2009		vg2010		vg2011	
Ortalama	28067,8	Ortalama	27617,0	Ortalama	29136,8	Ortalama	29878,8
Ortanca	19669,0	Ortanca	18210,0	Ortanca	19438,0	Ortanca	19902,0
Standart Sapma	40721,9	Standart Sapma	40957,4	Standart Sapma	40081,6	Standart Sapma	39832,5
Basıklık	48,9	Basıklık	48,6	Basıklık	44,5	Basıklık	40,5
Çarpıklık	6,4	Çarpıklık	6,4	Çarpıklık	6,1	Çarpıklık	5,7
Aralık	343582,0	Aralık	344566,0	Aralık	332152,0	Aralık	324093,0
En Küçük	4611,0	En Küçük	4420,0	En Küçük	4780,0	En Küçük	4570,0
En Büyük	348193,0	En Büyük	348986,0	En Büyük	336932,0	En Büyük	328663,0
vg2012		vg2013		vg2014		vg2015	
Ortalama	28615,0	Ortalama	31287,4	Ortalama	33102,2	Ortalama	33585,7
Ortanca	18344,0	Ortanca	20209,0	Ortanca	23080,0	Ortanca	23216,0
Standart Sapma	42075,8	Standart Sapma	44266,2	Standart Sapma	49556,3	Standart Sapma	47258,4
Basıklık	45,9	Basıklık	44,9	Basıklık	50,1	Basıklık	47,8
Çarpıklık	6,2	Çarpıklık	6,1	Çarpıklık	6,6	Çarpıklık	6,3
Aralık	349989,0	Aralık	366924,0	Aralık	419079,0	Aralık	396271,0
En Küçük	4085,0	En Küçük	4677,0	En Küçük	5583,0	En Küçük	6593,0
En Büyük	354074,0	En Büyük	371601,0	En Büyük	424662,0	En Büyük	402864,0
vg2016		vg2017		vg2018		vg2019	
Ortalama	32338,3	Ortalama	33145,9	Ortalama	37748,2	Ortalama	34643,5
Ortanca	20691,0	Ortanca	22243,0	Ortanca	22524,0	Ortanca	24821,0

Standart Sapma	51322,1	Standart Sapma	49334,6	Standart Sapma	69441,4	Standart Sapma	44739,5
Basıklık	51,6	Basıklık	49,8	Basıklık	53,8	Basıklık	44,6
Çarpıklık	6,7	Çarpıklık	6,5	Çarpıklık	6,9	Çarpıklık	6,1
Aralık	434236,0	Aralık	416033,0	Aralık	589919,0	Aralık	371277,0
En Küçük	6653,0	En Küçük	6526,0	En Küçük	5884,0	En Küçük	7028,0
En Büyük	440889,0	En Büyük	422559,0	En Büyük	595803,0	En Büyük	378305,0
vg2020							
Ortalama	27885,5						
Ortanca	17604,0						
Standart Sapma	44514,2						
Basıklık	51,3						
Çarpıklık	6,7						
Aralık	376439,0						
En Küçük	5215,0						
En Büyük	381654,0						

Tablo 8: Verilen Göç Veri Setindeki Kısaltmalar ve Bazı Temel İstatistikler

Tablo 8'e bakıldığında alınan göç veri setine benzer olarak verilen göç istatistiklerinin de yıllar geçse de dalgalı bir şekilde seyretse bile çok değişmedikleri gözlenmektedir.

ng2008		ng2009		ng2010		ng2011	
Ortanca	-999,0	Ortanca	-1319,0	Ortanca	-1821,0	Ortanca	-1765,0
Standart Sapma	10986,9	Standart Sapma	8792,3	Standart Sapma	14323,1	Standart Sapma	17149,3
Basıklık	3,1	Basıklık	8,8	Basıklık	34,5	Basıklık	33,5
Çarpıklık	1,4	Çarpıklık	2,3	Çarpıklık	5,3	Çarpıklık	4,7
Aralık	60811,0	Aralık	61493,0	Aralık	118148,0	Aralık	170640,0
En Küçük	-24586,0	En Küçük	-22012,0	En Küçük	-15565,0	En Küçük	-48858,0
En Büyük	36225,0	En Büyük	39481,0	En Büyük	102583,0	En Büyük	121782,0
ng2012		ng2013		ng2014		ng2015	
Ortanca	-233,0	Ortanca	-722,0	Ortanca	-1692,0	Ortanca	-2081,0
Standart Sapma	8069,7	Standart Sapma	10986,6	Standart Sapma	10009,9	Standart Sapma	11715,8
Basıklık	2,8	Basıklık	16,8	Basıklık	3,4	Basıklık	8,2
Çarpıklık	1,1	Çarpıklık	3,1	Çarpıklık	1,5	Çarpıklık	2,4
Aralık	47247,0	Aralık	86405,0	Aralık	57224,0	Aralık	70948,0

En Küçük	-16786,0	En Küçük	-20084,0	En Küçük	-17215,0	En Küçük	-19901,0
En Büyük	30461,0	En Büyük	66321,0	En Büyük	40009,0	En Büyük	51047,0
ng2016		ng2017		ng2018		ng2019	
Ortanca	500,0	Ortanca	-678,0	Ortanca	2191,0	Ortanca	-2175,0
Standart Sapma	11688,3	Standart Sapma	8606,9	Standart Sapma	25355,5	Standart Sapma	17017,5
Basıklık	17,0	Basıklık	3,7	Basıklık	60,8	Basıklık	32,7
Çarpıklık	-2,6	Çarpıklık	1,4	Çarpıklık	-7,3	Çarpıklık	5,1
Aralık	96430,0	Aralık	49433,0	Aralık	238348,0	Aralık	141625,0
En Küçük	-71307,0	En Küçük	-17391,0	En Küçük	-210321,0	En Küçük	-21254,0
En Büyük	25123,0	En Büyük	32042,0	En Büyük	28027,0	En Büyük	120371,0
ng2020							
Ortanca	-520,0						
Standart Sapma	9367,8						
Basıklık	12,9						
Çarpıklık	-1,6						
Aralık	76121,0						
En Küçük	-53022,0						
En Büyük	23099,0						

Tablo 9: Net Göç Veri Setindeki Kısaltmalar ve Bazı Temel İstatistikler

Tablo 9'a bakıldığında yıllar arasındaki net göç istatistikleri birbirlerinden oldukça farklıdır.

ngh2008		ngh2009		ngh2010		ngh2011	
Ortalama	-3,2	Ortalama	-3,5	Ortalama	-5,3	Ortalama	-5,0
Ortanca	-2,5	Ortanca	-2,8	Ortanca	-5,5	Ortanca	-5,4
Standart Sapma	13,8	Standart Sapma	10,8	Standart Sapma	9,3	Standart Sapma	9,6
Basıklık	3,2	Basıklık	4,9	Basıklık	0,8	Basıklık	3,4
Çarpıklık	0,5	Çarpıklık	0,7	Çarpıklık	-0,5	Çarpıklık	-0,9
Aralık	91,0	Aralık	75,8	Aralık	50,2	Aralık	63,2
En Küçük	-38,4	En Küçük	-29,7	En Küçük	-35,2	En Küçük	-46,7
En Büyük	52,6	En Büyük	46,1	En Büyük	15,0	En Büyük	16,6
ngh2012		ngh2013		ngh2014		ngh2015	
Ortalama	-1,3	Ortalama	-1,4	Ortalama	-3,6	Ortalama	-4,3
Ortanca	-1,0	Ortanca	-1,3	Ortanca	-4,3	Ortanca	-4,6
Standart Sapma	11,3	Standart Sapma	12,1	Standart Sapma	13,3	Standart Sapma	11,3
Basıklık	1,1	Basıklık	2,1	Basıklık	2,0	Basıklık	-0,2
Çarpıklık	0,5	Çarpıklık	0,5	Çarpıklık	0,1	Çarpıklık	-0,1

Aralık	62,6	Aralık	72,8	Aralık	87,7	Aralık	50,7
En Küçük	-27,0	En Küçük	-33,0	En Küçük	-45,6	En Küçük	-28,6
En Büyük	35,5	En Büyük	39,8	En Büyük	42,2	En Büyük	22,1
ngh2016		ngh2017		ngh2018		ngh2019	
Ortalama	1,2	Ortalama	-3,7	Ortalama	9,8	Ortalama	-7,5
Ortanca	1,2	Ortanca	-1,4	Ortanca	4,9	Ortanca	-4,6
Standart Sapma	19,0	Standart Sapma	16,8	Standart Sapma	20,6	Standart Sapma	15,3
Basıklık	19,7	Basıklık	27,8	Basıklık	18,6	Basıklık	17,5
Çarpıklık	3,2	Çarpıklık	-4,3	Çarpıklık	3,5	Çarpıklık	-3,2
Aralık	158,6	Aralık	139,8	Aralık	155,8	Aralık	116,1
En Küçük	-37,1	En Küçük	-119,0	En Küçük	-17,3	En Küçük	-101,3
En Büyük	121,5	En Büyük	20,8	En Büyük	138,5	En Büyük	14,8
ngh2020							
Ortalama	-3,3						
Ortanca	-0,9						
Standart Sapma	12,4						
Basıklık	19,1						
Çarpıklık	-3,3						
Aralık	101,9						
En Küçük	-81,0						
En Büyük	20,8						

Tablo 10: Net Göç Hızı Veri Setindeki Kısaltmalar ve Bazı Temel İstatistikler

Tablo 10'a bakıldığında net göç hızı ortalamasının 2016 yılı hariç her zaman negatif olduğu görülmektedir.

Veri setinde kayıp değer olmayıp, bazı değerler birbirlerinden oldukça farklıdır. Bu nedenle verilere standartlaştırma işlemi uygulanacaktır. Bu işlem yapılırken ise veri setindeki tüm değerler $[0,1]$ aralığında olacak şekilde hesaplamalar yapılmıştır. Bunun için x_{\min} veri setindeki minimum değer ve DG ise dizi genişliği olmak üzere aşağıdaki formül kullanılmıştır:

$$s_i = \frac{x_i - x_{\min}}{DG} \quad (3.1)$$

3.5. BULGULAR

Standartlaştırılan veriler üzerinden her biri ayrı ayrı olmak üzere alınan göç(ag), verilen göç (vg), net göç (ng) ve net göç hızı (ngh) istatistikleri kullanılmıştır. Makine öğrenmesinin denetimsiz öğrenme yöntemlerinden biri olan K-ortalamlar algoritması kullanılmış olup işlemler Python yazılım dili üzerinden yapılmıştır. Optimum küme sayısını belirlemek için Elbow metodu kullanılmıştır. Bu metotta küme içi uzaklıkların kareleri toplamının minimum ve kümeler arası uzaklıkların maksimum olması prensibine göre kümeler içi kareler toplamı olan wcss değeri her bir küme sayısı için hesaplanıp bunların içinden en uygun küme sayısı belirlenmeye çalışılmıştır. Ayrıca model başarısını değerlendirmek için Silhouette Score (Siluet İndeksi) hesaplanmıştır. Buna göre Elbow metoduyla hesaplanan değerler aşağıdaki grafiklerde gösterilmiştir:

Elbow (Dirsek) Metodu: Kümeleme analizinde, mevcut bir veri setindeki küme sayısını belirlemede kullanılan bir yöntemdir. Bu yöntemde wcss(Kümeler içi kareler toplamı) değeri noktaların her bir k değerine hesaplanmaktadır. Bu hesaplanan wcss değerine göre yine her k için bir grafik çizilmektedir. Grafikte k değeri arttıkça wcss değerinin başta hızla azaldığı daha sonra bu azalmanın yavaşlaması öngörülmektedir. İşte bu grafikteki kırılma noktası Elbow(dirsek) olarak tanımlanmakta ve bu noktadaki k değeri ise en uygun küme sayısı olarak alınabilmektedir.

Üstünel (2018), K-ortalamlar Algoritmasına Dayalı Kümeleme Analizi Sistemi ve Perakendecilik Sektöründe Uygulaması isimli çalışmasında, her bir k değeri için hataları hesaplamış ve en uygun küme sayısını belirlemek için Elbow metodunu kullanmıştır.

Kılıç vd. (2020), K-ortalamlar Kümeleme Analizi İle Belediyelerin Çevre Hizmetlerinin Değerlendirilmesi isimli çalışmasında yine en uygun küm sayısını belirlemek için Elbow metodunu kullanmıştır.

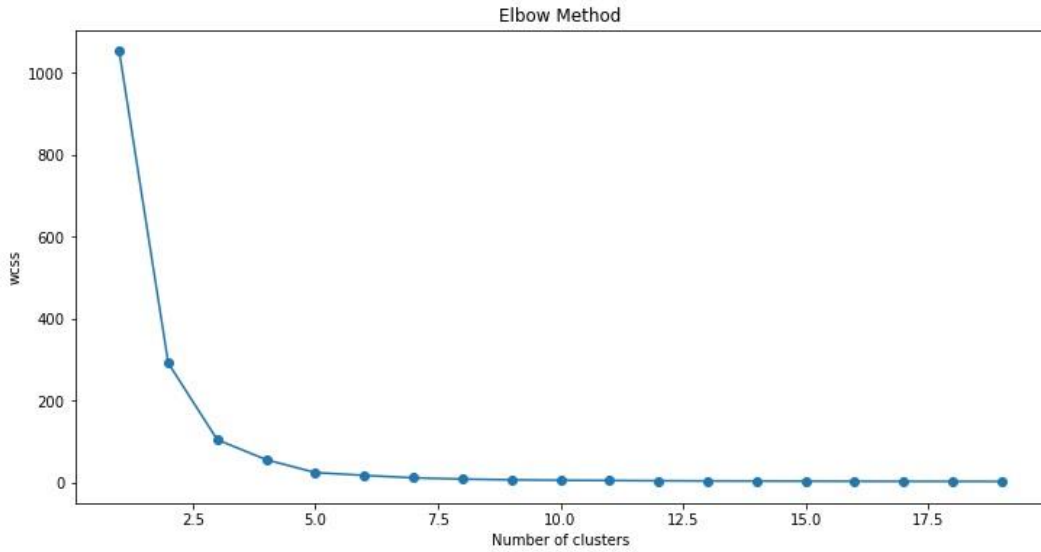
Silva ve Sharma(2020) çalışmalarında Konkani metinlerinin denetimsiz otomatik metin özeti araştırmasını K-ortalamlar algoritmasını kullanarak yapmış ve yine en uygun küme sayısı belirlemede Elbow metodu kullanmışlardır.

Literatüre bakıldığında Elbow metodun kullanıldığı çok sayıda çalışma mevcuttur. Burada bu metodun kullanıldığı çalışmalardan birkaç tane örnek verilmiştir.

Silhouette Score(Siluet İndeksi):Siluet indeksi her nesnenin kendi kümesine uygunluğunu tanımlayacak bir değer olarak Rousseeuw (1987) tarafından önerilmiştir. Siluet indeksi birçok karşılaştırmalı analizlerde iyi sonuçlar vermektedir. Birçok uzaklık ölçüsü ile de çalışabilir. Şu şekilde hesaplanabilir:

$$Sil(i) = \frac{b(i) - a(i)}{\max(a(i), b(i))} \quad (3.2)$$

Denklem 3.2 ‘deki a(i); i. birimin kendi kümesindeki tüm noktalara olan ortalama uzaklıklarını (benzerliğini) göstermekte ve b(i); i. birimin diğer kümelerdeki tüm noktalara olan ortalama uzaklıkların minimumunu göstermektedir. Sil(i) değeri $[-1,1]$ aralığında tanımlanmaktadır. Bu değer 0,5 ‘ten büyük olması yapılan kümeleme işleminin uygun olması anlamına gelmektedir. 1’e ne kadar yakınsa kümelemenin o kadar başarılı olduğu anlamına gelmektedir. Negatif sonuçlar ise herhangi bir değer en uygun kümesine atanmadığında çıkmaktadır.



Grafik 5:Alınan Göç Veri Setine Ait wcss Değerleri

Yukarıdaki grafiğe bakıldığında en uygun küme sayısının belirlemek için Elbow metodu kullanıldığında dirsek şeklindeki kırılmanın k=3 noktasında olduğu görülmektedir. Buna göre k=3 için K-ortalamlar algoritması ile illerin aldıkları göç istatistiklerine göre kümeleme sonuçları aşağıdaki çizelgede sunulmuştur:

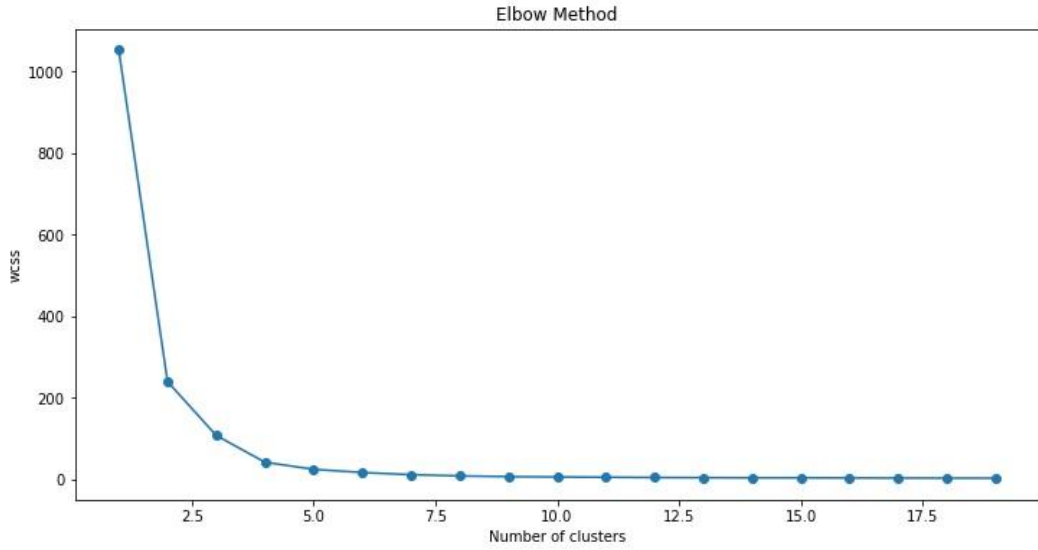
Küme Numarası	İller
1	İstanbul
2	Ankara, Antalya, Bursa, İzmir, Kocaeli
3	Adana, Adıyaman, Afyonkarahisar, Ağrı, Amasya, Artvin, Aydın, Balıkesir, Bilecik, Bingöl, Bitlis, Bolu, Burdur, Çanakkale, Çankırı, Çorum, Denizli, Diyarbakır, Edirne, Elazığ, Erzincan, Erzurum, Eskişehir, Gaziantep, Giresun, Gümüşhane, Hakkâri, Hatay, Isparta, Mersin, Kars, Kastamonu, Kayseri, Kırklareli, Kırşehir, Konya, Kütahya, Malatya, Manisa, Kahramanmaraş, Mardin, Muğla, Muş, Nevşehir, Niğde, Ordu, Rize, Sakarya, Samsun, Siirt, Sinop, Sivas, Tekirdağ, Tokat, Trabzon, Tunceli, Şanlıurfa, Uşak, Van, Yozgat, Zonguldak, Aksaray, Bayburt, Karaman, Kırıkkale, Batman, Şırnak, Bartın, Ardahan, Iğdır, Yalova, Karabük, Kilis, Osmaniye, Düzce

Tablo 11: İllerin Aldıkları Göç İstatistiklerine Göre Elde Edilen

Kümelerin Elemanları

Alınan göç istatistiklerine göre kümeleme işlemi sonrasında bu yapılan işlemin başarısının ölçülmesi için Silhouette Score (Siluet İndeksi) kullanılmıştır. Buna göre Siluet İndeksi = 0,786244 bulunmuş olup modelin oldukça başarılı olduğu söylenebilir.

Tablo 11'e bakıldığında beklenen bir sonuç olarak Türkiye'nin en büyük şehri olan İstanbul tek başına bir küme oluşturmuştur. Bu ilin nüfusunun, iş olanaklarının, sosyo ekonomik gelişmişliğinin, modern kent hayatının cazibesinin vb. sebeplerin böyle bir durum oluşturduğu söylenebilir. Tablo 11'deki ikinci kümeye bakıldığında Ankara, Antalya, İzmir, Bursa ve Kocaeli görünmektedir. İlk dört sıradaki iller Türkiye'nin İstanbul'dan sonraki nüfusu en çok olan illerdir. Dolayısıyla bu iller ekonomik, sosyal, kültürel ve istihdam gibi imkânlarının çok olması sebebiyle böyle bir sonuç ortaya çıkmıştır. Keza yine Kocaeli ili Türkiye'nin sanayide önde gelen illeri arasındadır. Dolayısıyla istihdam yönünden çok göç alan illeri arasında görülmektedir.



Grafik 6: Verilen Göç Veri Setine Ait wcss Değerleri

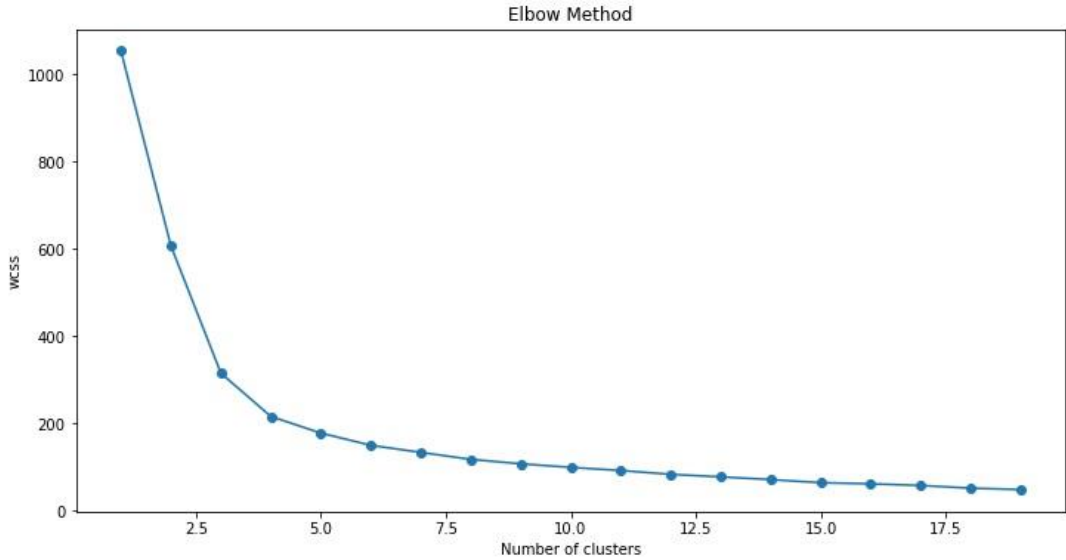
Yukarıdaki grafiğe bakıldığında en uygun küme sayısının belirlemek için Elbow metodu kullanıldığında dirsek şeklindeki kırılmanın $k=3$ noktasında olduğu görülmektedir. Buna göre $k=3$ için K-ortalamlar algoritması ile illerin verdikleri göç istatistiklerine göre kümeleme sonuçları aşağıdaki çizelgede sunulmuştur:

Küme Numarası	İller
1	İstanbul
2	Adana, Ankara, Antalya, Bursa, Diyarbakır, Gaziantep, Mersin, İzmir, Kocaeli, Konya, Samsun, Şanlıurfa, Van
3	Adıyaman, Afyonkarahisar, Ağrı, Amasya, Artvin, Aydın, Balıkesir, Bilecik, Bingöl, Bitlis, Bolu, Burdur, Çanakkale, Çankırı, Çorum, Denizli, Edirne, Elazığ, Erzincan, Erzurum, Eskişehir, Giresun, Gümüşhane, Hakkâri, Hatay, Isparta, Kars, Kastamonu, Kayseri, Kırklareli, Kırşehir, Kütahya, Malatya, Manisa, Kahramanmaraş, Mardin, Muğla, Muş, Nevşehir, Niğde, Ordu, Rize, Sakarya, Siirt, Sinop, Sivas, Tekirdağ, Tokat, Trabzon, Tunceli, Uşak, Yozgat, Zonguldak, Aksaray, Bayburt, Karaman, Kırıkkale, Batman, Şırnak, Bartın, Ardahan, Iğdır, Yalova, Karabük, Kilis, Osmaniye, Düzce

**Tablo 12: İllerin Verdikleri Göç İstatistiklerine Göre Elde Edilen
Kümelerin Elemanları**

Verilen göç istatistiklerine göre kümeleme işlemi sonrasında bu yapılan işlemin başarısının ölçülmesi için Silhouette Score (Siluet İndeksi) kullanılmıştır. Buna göre Siluet İndeksi = 0,648519 bulunmuş olup modelin makul bir seviyede başarılı olduğu söylenebilir.

Tablo 12'ye bakıldığında yine beklenen bir sonuç olarak Türkiye'nin en büyük ili olan İstanbul tek başına bir küme oluşturmuştur. Bu ilin nüfusunun çok fazla olması ve yine bu ilin birçok insan için cazibe merkezi olmasının yanında bir kısım insanlar için de zorlu bir hayat barındırması nedeniyle böyle bir durum ortaya çıktığı değerlendirilmektedir. İkinci kümedeki illerin ise birçoğu çok nüfuslu iller olup aynı zamanda yine bu illerin bir kısmı Türkiye'deki yatırım teşvik kapsamında desteklenecek birinci bölge illeridir. Özellikle Şanlıurfa, Van, Gaziantep gibi illerin de yine bu grupta olmalarının sebebinin batıdaki illerin her anlamda cazibesinin olması ve mevsimlik işçi hareketliliklerinin de neden olduğu değerlendirilmektedir.



Grafik 7:Net Göç Veri Setine Ait wcss Değerleri Grafiği

Yukarıdaki grafiğe bakıldığında en uygun küme sayısının belirlemek için Elbow metodu kullanıldığında dirsek şeklindeki kırılmanın k=5 noktasında olduğu

görülmektedir. Buna göre k=5 için K-ortalamlar algoritması ile illerin net göç istatistiklerine göre kümeleme sonuçları aşağıdaki çizelgede sunulmuştur:

Küme Numarası	İller
1	Adana, Adıyaman, Ağrı, Bitlis, Çorum, Diyarbakır, Erzurum, Hatay, Kars, Kahramanmaraş, Mardin, Muş, Sivas, Tokat, Şanlıurfa, Van, Yozgat, Zonguldak
2	Ankara
3	İstanbul
4	Afyonkarahisar, Amasya, Artvin, Aydın, Balıkesir, Bilecik, Bingöl, Bolu, Burdur, Çanakkale, Çankırı, Denizli, Edirne, Elazığ, Erzincan, Gaziantep, Giresun, Gümüşhane, Hakkâri, Isparta, Mersin, Kastamonu, Kayseri, Kırklareli, Kırşehir, Konya, Kütahya, Malatya, Manisa, Muğla, Nevşehir, Niğde, Ordu, Rize, Sakarya, Samsun, Siirt, Sinop, Trabzon, Tunceli, Uşak, Aksaray, Bayburt, Karaman, Kırıkkale, Batman, Şırnak, Bartın, Ardahan, Iğdır, Yalova, Karabük, Kilis, Osmaniye, Düzce
5	Antalya, Bursa, Eskişehir, İzmir, Kocaeli, Tekirdağ

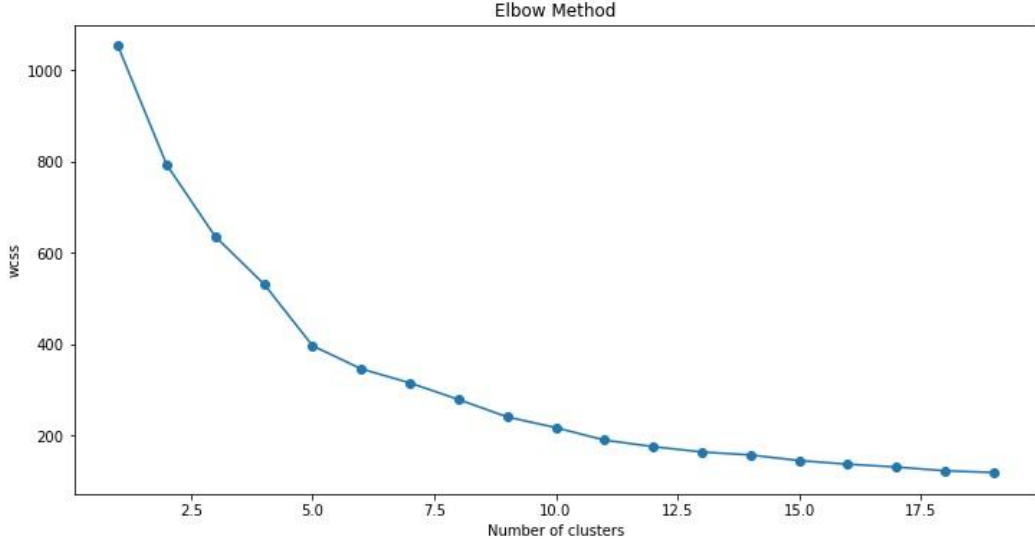
Tablo 13: İllerin Net Göç İstatistiklerine Göre Elde Edilen

Kümelerin Elemanları

Net göç istatistiklerine göre kümeleme işlemi sonrasında bu yapılan işlemin başarısının ölçülmesi için Silhouette Score (Siluet İndeksi) kullanılmıştır. Buna göre Siluet İndeksi = 0,388169 bulunmuş olup modelin makul seviyenin altında kaldığı söylenebilir.

Tablo 13'e bakıldığında Türkiye'nin en büyük iki ili olan İstanbul ve Ankara birer küme oluşturmuştur. 2008-2020 arasındaki göç verilerine bakıldığında bu iki ilin de açık ara farkla en çok net göç alan şehirler olduğu görülmektedir. Beşinci kümedeki iller de yine Türkiye'nin net göç alan diğer büyük illeridir. Birinci kümedeki illere bakıldığında ise bu illerin 2008-2020 arasında en çok net göç veren iller olduğu görülmektedir. Son

olarak dördüncü kümedeki illere bakıldığında ise 2008-2020 yılları göç istatistiklerine göre bunların da yine net göç veren iller olduğu görülmektedir.



Grafik 8:Net Göç Hızı Veri Setine Ait wcss Değerleri Grafiği

Yukarıdaki grafiğe bakıldığında en uygun küme sayısının belirlemek için Elbow metodu kullanıldığında dirsek şeklindeki kırılmanın k=9 noktasında olduğu görülmektedir. Buna göre k=9 için K-ortalamalar algoritması ile illerin verdikleri göç istatistiklerine göre kümeleme sonuçları aşağıdaki çizelgede sunulmuştur:

Küme Numarası	İller
1	Adana, Afyonkarahisar, Amasya, Artvin, Bingöl, Çorum, Diyarbakır, Elazığ, Giresun, Hakkâri, Hatay, Kırşehir, Kütahya, Malatya, Kahramanmaraş, Mardin, Nevşehir, Niğde, Rize, Samsun, Siirt, Sinop, Sivas, Trabzon, Tunceli, Şanlıurfa, Zonguldak, Aksaray, Karaman, Batman, Şırnak, Iğdır, Karabük
2	Bayburt
3	Gümüşhane
4	Çankırı
5	Adıyaman

6	Ankara, Antalya, Bursa, Eskişehir, Kocaeli, Muğla, Tekirdağ, Yalova
7	Ağrı, Bitlis, Erzurum, Kars, Muş, Yozgat, Ardahan
8	Ordu, Tokat, Van, Kırıkkale
9	Aydın, Balıkesir, Bilecik, Bolu, Burdur, Çanakkale, Denizli, Edirne, Erzincan, Gaziantep, Isparta, Mersin, İstanbul, İzmir, Kastamonu, Kayseri, Kırklareli, Konya, Manisa, Sakarya, Uşak, Bartın, Kilis, Osmaniye, Düzce

Tablo 14: İllerin Net Göç Hızı İstatistiklerine Göre Elde Edilen

Kümelerin Elemanları

Net göç hızı istatistiklerine göre kümeleme işlemi sonrasında bu yapılan işlemin başarısının ölçülmesi için Silhouette Score (Siluet İndeksi) kullanılmıştır. Buna göre Siluet İndeksi = 0,211708 bulunmuş olup modelin makul seviyenin çok altında kaldığı söylenebilir.

Tablo 14'e bakıldığında diğer kümelemelerden çok daha farklı bir sonuç olduğu görülmektedir. Küme sayısı dokuz çıkmıştır. Gümüşhane, Bayburt, Çankırı ve Adıyaman illeri tek başına birer küme oluşturmuştur. Bu iller yatırım teşvik kapsamında desteklenecek olan bölgelerden beşinci bölgededirler. Ayrıca 2008-2020 göç verilerine bakıldığında bu illerin nüfuslarının az olması ve göç durumlarının çok oynak olduğu görülmektedir. Yani bir sene yüksek oranda göç alırken diğer senelerde yüksek oranda göç verdiği görülmektedir. Birinci kümedeki illerin ise negatif olarak net göç hızlarının birbirlerine benzer oldukları görülmektedir. Yedinci kümedeki illerin hepsi Türkiye'nin doğusunda yer alan göç veren ve birçok imkânların daha kısıtlı iller oldukları değerlendirilmektedir.

SONUÇ VE ÖNERİLER

Genel olarak göç ekonomik, ailevi, siyasal, kültürel vb. nedenlerle bireylerin veya toplumların mevcut ikamet ettikleri yerleri terk edip başka bir yerleşim yerine gitmeleri olarak tanımlanabilir. Bunun nedenleri arasında ilk etken olarak ekonomik sebepler gelse de bunun yanında siyasal, kültürel, sosyal, ailevi, eğitim vb. birçok neden de vardır. Bu sebepler ise genel olarak itici ve çekici sebepler olmak üzere iki başlıkta incelenebilir. Genel olarak ise kırsal alanların “itici” olma özelliği ve yine kentsel alanların ise “çekici” olma özelliği göç olgusunun gerçekleşmesinde önemli rol oynamaktadır. Kırsalda yaşanan işsizlik, sanayileşmenin yaratmış olduğu tarımdaki olumsuz durumlar itici faktörleri meydana getirirken, kentteki yaşamın cazibeli olması, olanakların çok fazla olması çekici faktörleri meydana getirmiştir. Türkiye’nin kurulduğu günden itibaren içine girmiş olduğu sanayileşme, kalkınma hamleleri ve ekonomik gelişmeler neticesinde iç göç hareketleri yaşanmıştır.

Kümeleme analizi, uzun yıllardan beri makine öğrenmesi ve veri madenciliğinde kullanılan pratik ve etkili yöntemlerden biridir. Esas olarak birbirine benzer değerleri aynı kümede ve birbirine benzemez veya uzak olan değerleri ise farklı kümelere ayırmak prensibine dayanır. Yani küme içi benzerlikleri maksimum kılarken, kümeler arası farklılıkları veya uzaklıkları maksimum kılmak amaçlanmaktadır.

Mevcut veri setleri içerisinde saklı olan, görmesi ve anlaşılması hiç de kolay olmayan çeşitli bağlantıları makine öğrenmesi ve veri madenciliğinin algoritmaları ile bulabilmek ve bunları anlamlı hale getirmek mümkün olmaktadır. Kümeleme analizi ile bu verilerdeki benzer değerlerin bir arada görmek mümkündür.

Yapılan bu çalışmada Türkiye’deki 81 ilin, bu illere ait göç istatistiklerine göre makine öğrenmesi yaklaşımı ile kümelenebilirliği amaçlanmıştır. Makine öğrenmesi denetimsiz öğrenme yöntemlerinden biri olan K-ortalamlar algoritması kullanılmıştır. Alınan göç verilerine göre küme sayısı 3, verilen göç verilerine göre yine küme sayısı 3, net göç verilerine göre küme sayısı 5 ve en son olarak da net göç verilerine göre ise küme sayısı 9 olarak belirlenmiştir. Buna göre, TÜİK tarafından yayınlanan 2008-2020 yılları

arasındaki iç göç istatistikleri kullanılmıştır. Bu istatistiklerdeki alınan göç, verilen göç, net göç ve net göç hızı verilerine göre ayrı ayrı kümeleme işlemleri yapılmıştır.

Genel olarak bakıldığında özellikle büyük ve gelişmiş illerin, çoğu zaman aynı kümelerde yer aldıkları görülmektedir. Özellikle İstanbul, net göç hızı kümelemesi hariç diğer bütün durumlarda başlı başına bir küme oluşturmuştur. Bu veriler ışığında İstanbul'un diğer illerden çok farklı bir yerde olduğu açıktır. Sonuç olarak da tabii ki İstanbul için çok daha farklı ve kapsamlı göç politikaları yapılmasında fayda görülmektedir.

Net göç hızı kümelemesine bakılacak olursa Bayburt, Gümüşhane, Çankırı ve Adıyaman illerinin tek başlarına birer kümede olduğu görülmektedir. Bu illerin nispeten az nüfuslu olmaları ve bir yıl yüksek oranda göç alırken diğer yıl yüksek oranda göç vermelerinin bu sonuca yol açtığı söylenebilir.

Tek tek bakıldığında her ilin kendine has durumları oldukları söylenebilir. Fakat burada yapılan çalışmada 4 değişken üzerinden her biri ayrı ayrı olmak üzere birbirlerine benzer illerin bir arada gösterilmesi amaçlanmıştır.

Kümeleme sonucundaki amaç ise politika yapıcılara Türkiye'deki iç göç stratejileri geliştirirken bir ışık tutmak ve perspektif göstermek olmuştur. Buna göre kullanılan istatistikler ışığında oluşan kümeleme sonuçları ülkede sağlıklı bir göç hareketliliğini sağlamak adına etkili şekilde değerlendirilmelidir. Değişkenler hem ayrı ayrı hem de hep beraber şekilde ele alınmalarında fayda görülmektedir. Her il için özel bir göç stratejisi geliştirmenin yanında benzer iller için de ayrıca bölgesel ve genel stratejilere ihtiyaç olduğu aşikârdır.

Türkiye'de iç göçün daha kontrollü ve yönetilebilir olması adına farklı politikalar uygulayarak özellikle net göç veren illere yönelik farklı teşvik projeleri yapılmasında fayda görülmektedir. İller arası ve yine bölgeler arası farklılık ve dengesizlikler neticesinde yaşanan bu iç göç olgusu devam ettiği sürece iller arasında gelir dengesizlikleri ve gelişmişlik farklılıkları da artmaya devam edeceği değerlendirilmektedir. Bunun doğal bir sonucu olarak göç alan illerde olumsuz durumlar artarken, göç veren illerin de kayıplar yaşayacağı gerçeği gün gibi ortadadır.

Ayrıca, çalışmada gerçekleştirilen kümelemeler neticesinde, illerin aldıkları göçe göre sınıflandırılmasında % 78,6 ve illerin verdikleri göçe göre sınıflandırılmasında % 64,8 oranında doğru sınıflandırıldığı Siluet İndeksi kullanılarak elde edilmiştir. Böylece belirlenen küme sayıları ile birlikte yapılan kümeleme analizinin makul derecede başarılı olduğu değerlendirilmektedir. Öte yandan illerin net göç oranlarına göre sınıflandırılmasında % 38,8 ve illerin net göç hızlarına göre sınıflandırılmasında % 21,1 oranında doğru sınıflandırıldığı görülmektedir. Bu sonuçlara bakıldığında her ne kadar doğru küme sayıları da belirlenmiş olsa bile yapılan kümeleme sonuçlarının çok da başarılı olmadığı değerlendirilmektedir. Buradan hareketle gelecekte bu alanda çalışacak olan araştırmacılar için kullanılacak olan göç istatistiklerinde net göç ve net göç hızlarından ziyade alınan göç ve verilen göç değişkenlerinin kullanmalarının daha uygun olacağı tavsiye edilebilir.

EKLER

EK-1: Kullanılan Python Kodları

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
from sklearn.cluster import KMeans

sonag=pd.read_table("sonag.txt",index_col=0)
sonag

from sklearn.preprocessing import StandardScaler
scaler = StandardScaler()
data_scaled = scaler.fit_transform(sonag)

kmeans = KMeans(n_clusters=2, init='k-means++')

kmeans.fit(data_scaled)

SSE = []
for cluster in range(1,20):
    kmeans = KMeans(n_jobs = -1, n_clusters = cluster, init='k-means++')
    kmeans.fit(data_scaled)
    SSE.append(kmeans.inertia_)
```



```

frame = pd.DataFrame({'Cluster':range(1,20), 'SSE':SSE})
plt.figure(figsize=(12,6))
plt.plot(frame['Cluster'], frame['SSE'], marker='o')
plt.xlabel('Number of clusters')
plt.ylabel('wcss')
plt.title('Elbow Method')

```

```

kmeans = KMeans(n_jobs = -1, n_clusters = 3, random_state=42)
kmeans.fit(data_scaled)
pred = kmeans.predict(data_scaled)

```

```

frame = pd.DataFrame(data_scaled)
frame['cluster'] = pred
frame['cluster'].value_counts()

```

```

print(kmeans.labels_)

```

```

%%time
n_cluster=range(1,21)
kmeans=[KMeans(n_clusters=i).fit(sonag) for i in n_cluster]
scores=[kmeans[i].score(sonag) for i in range(len(kmeans))]

```

```

f=plt.figure(1,figsize=(16,6))

plt.plot(scores)

_=plt.xticks(n_cluster)

from sklearn.metrics import silhouette_score

score=silhouette_score(sonag,kmeans.labels_,metric='euclidean')

print(score)

```

EK-2: 2008-2020 Türkiye'nin İller Bazında Göç İstatistikleri

X	ag2008vg2008	ng2008	ngh2008	ag2009vg2009	ng2009	ngh2009	ag2010vg2010	ng2010	ngh2010	ag2011vg2011	ng2011	ngh2011	ag2012vg2012	ng2012	ngh2012	ag2013vg2013	ng2013	ngh2013	ag2014vg2014	ng2014	ngh2014	ag2015vg2015	ng2015	ngh2015	ag2016vg2016	ng2016	ngh2016	ag2017vg2017	ng2017	ngh2017	ag2018vg2018	ng2018	ngh2018	ag2019vg2019	ng2019	ngh2019	ag2020vg2020	ng2020	ngh2020													
Adana	45493	58316	-12283	-6.3	53685	54109	-424	-0.2	53096	57402	-4306	-2.1	50523	62402	-11879	-5.6	45927	59294	-13367	-6.3	50598	62933	-12335	-5.7	51562	63181	-11619	-5.4	52647	64192	-11545	-5.3	51466	60759	-9293	-4.2	49509	62834	-13325	-6	51660	70638	-18978	-8.5	52290	60900	-8610	-3.8	47088	45832	1256	0.6
Adıyaman	12155	20971	-8816	-15	12091	18295	-6204	-10.5	14150	20135	-5985	-10.1	13873	23939	-10066	-16.8	14323	22591	-8268	-13.8	15715	24077	-8362	-139	16578	24330	-7752	-12.9	18523	24064	-5541	-9.2	18789	21201	-2412	-3.9	18040	22741	-4701	-7.6	22034	22897	-863	-1.4	17619	24821	-7202	-11.4	16163	16936	-773	-1.2
Afyonkarahisar	18270	24861	-6591	-9.4	20223	22256	-2033	-2.9	17451	24119	-6668	-9.5	17219	24063	-6844	-9.7	24167	21738	2429	3.5	22200	26799	-4599	-6.5	21414	27100	-5686	-8	22922	26226	-3304	-4.6	22467	24400	-1933	-2.7	21453	25483	-4030	-5.6	26976	25049	1927	2.7	23818	27006	-3188	-4.4	18439	19434	-995	-1.3
Ağrı	12195	27450	-15255	-28.3	12115	22613	-10498	-19.3	14950	22966	-8016	-14.7	15508	25182	-9674	-17.3	12856	27984	-15128	-27	14725	28477	-13752	-24.6	14480	30995	-16515	-29.6	16597	32174	-15577	-28.1	14306																			

	30811	-16505	-30	15088	32479	-17391	-31.9	20091	29482	-9391	-17.3
	16342	28158	-11816	-21.8	14626	24353	-9727	-18			
Amasya	11748	14268	-2520	-7.8	11533	13187	-1654	-5.1	11729	13726	
	-1997	-5.9	11424	14142	-2718	-8.4	11228	13128	-1900	-5.9	12379
	14658	-2279	-7.1	12970	14817	-1847	-5.7	13957	15092	-1135	-3.5
	15395	13609	1786	5.5	14326	14675	-349	-1.1	16760	15237	1523
	4.5	13329	14813	-1484	-4.4	12241	11347	894	2.7		
Ankara	156760		126198		30562	6.7	168193		131114		37079
	8	182845		133440		49405	10.4	191864		137385	
	54479	11.2	160235		137834		22401	4.5	186642		
	153791		32851	6.5	203621		163612		40009	7.8	
	204048		153001		51047	9.7	177166		159915		17251
	3.2	188100		156058		32042	5.9	184382		221747	
	-37365	-6.8	211744		154464		57280	10.2	153162		
	141165		11997	2.1							
Antalya	92031	55806	36225	19.7	75696	58632	17064	8.9	86907	61662	
	25245	12.8	89731	62875	26856	13.2	83596	62893	20703	9.9	88605
	64075	24530	11.4	93057	64631	28426	12.9	96441	68374	28067	12.3
	79203	73119	6084	2.6	87232	72178	15054	6.4	95920	80349	15571
	6.4	100140		68494	31646	12.7	83035	62325	20710	8.2	
Artvin	6700	8660	-1960	-11.7	6206	7547	-1341	-8.1	7115	7988	-873
	-5.3	7948	7948	0	0	7286	7612	-326	-1.9	10053	8644
	1409	8.4	9551	10187	-636	-3.7	9050	10969	-1919	-11.3	8240
	9283	-1043	-6.2	7788	10146	-2358	-14.1	15934	8876	7058	41.4
	8403	11676	-3273	-19	6358	7801	-1443	-8.5			
Aydın	34375	25577	8798	9.2	28524	27640	884	0.9	29971	29923	48
	0	32077	31056	1021	1	32412	29623	2789	2.8	34688	32338
	2350	2.3	45842	32396	13446	13	42743	37689	5054	4.8	40363
	33119	7244	6.8	41352	35339	6013	5.6	47005	36462	10543	9.7
	40439	36180	4259	3.8	35455	30996	4459	4			
Balıkesir	37407	31129	6278	5.6	33275	31291	1984	1.7	36784	35162	
	1622	1.4	37143	36638	505	0.4	34922	35315	-393	-0.3	38710
	39688	-978	-0.8	57551	39918	17633	14.9	43296	48892	-5596	-4.7
	43475	38566	4909	4.1	43998	40371	3627	3	55238	40028	15210
	12.5	41467	42931	-1464	-1.2	40715	31561	9154	7.4		
Bilecik	8206	8793	-587	-3	8283	8251	32	0.2	8876	8690	186
	0.8	9250	8914	336	1.6	10387	8118	2269	11.2	10231	9202
	1029	4.9	10090	10114	-24	-0.1	10890	9806	1084	5.1	11031
	9911	1120	5.1	11744	9359	2385	10.8	11956	12597	-641	-2.9
	10565	10582	-17	-0.1	8297	8439	-142	-0.6			

Bingöl	8977	10192	-1215	-4.7	6986	10726	-3740	-14.5	8453	10448	-1995
	-7.8	9292	10782	-1490	-5.7	7569	11145	-3576	-13.5	10968	10940
	28	0.1	9400	12852	-3452	-12.9	9906	13122	-3216	-12	10304
	11205	-901	-3.3	11979	11941	38	0.1	15770	12216	3554	12.7
	11266	14879	-3613	-12.8	9497	10633	-1136	-4			
Bitlis	11721	20957	-9236	-27.9	11594	16485	-4891	-14.8	10800	15347	-4547
	-13.7	11419	16610	-5191	-15.3	10998	16886	-5888	-17.3	11314	16174
	-4860	-14.3	11447	17470	-6023	-17.7	11936	18801	-6865	-20	12571
	18074	-5503	-16	12989	19097	-6108	-17.7	17602	17398	204	0.6
	12385	19050	-6665	-19	12050	14223	-2173	-6.2			
Bolu	10049	11032	-983	-3.6	12808	10124	2684	9.9	11874	11926	-52
	-0.2	12545	11386	1159	4.2	13136	9677	3459	12.4	13138	12401
	737	2.6	13264	13065	199	0.7	14681	12401	2280	7.9	18528
	11762	6766	22.8	14839	13643	1196	4	18307	14827	3480	11.2
	15997	15068	929	2.9	11430	12565	-1135	-3.6			
Burdur	9674	10019	-345	-1.4	8144	8659	-515	-2	9439	8904	535
	2.1	8873	9712	-839	-3.3	11473	8497	2976	11.8	11156	10001
	1155	4.5	10235	12271	-2036	-7.9	12537	12247	290	1.1	12337
	11059	1278	4.9	12745	11786	959	3.6	15417	12537	2880	10.7
	12134	13603	-1469	-5.4	9201	11422	-2221	-8.3			
Bursa	82964	47370	35594	14.3	66615	56368	10247	4	72640	57220	15420
	5.9	74243	58258	15985	6	67736	61520	6216	2.3	75518	61744
	13774	5	80717	65027	15690	5.6	84253	64558	19695	7	84458
	63812	20646	7.1	86119	64905	21214	7.2	80940	80086	854	0.3
	85596	68356	17240	5.7	77039	53940	23099	7.5			
Çanakkale	16390	14246	2144	4.5	15412	14802	610	1.3	16578	15420	
	1158	2.4	17278	15623	1655	3.4	23252	14374	8878	18.1	24766
	18238	6528	13.1	27429	20540	6889	13.6	25096	23005	2091	4.1
	25116	20024	5092	9.8	26088	20540	5548	10.5	29473	21068	8405
	15.7	25198	22293	2905	5.4	21055	19022	2033	3.8		
Çankırı	12680	11612	1068	6.1	20166	11831	8335	46.1	11536	17958	
	-6422	-35.2	10582	14322	-3740	-20.9	17050	10609	6441	35.5	22290
	16991	5299	28.1	14519	23080	-8561	-45.6	12702	16148	-3446	-18.9
	13371	13088	283	1.5	13849	13504	345	1.9	40460	12433	28027
	138.5	11381	32275	-20894	-101.3	9955	12614	-2659	-13.7		
Çorum	15828	24072	-8244	-15	13672	22699	-9027	-16.6	14462	22860	-8398
	-15.6	15174	22441	-7267	-13.5	13289	20610	-7321	-13.7	17073	21460
	-4387	-8.2	15875	24652	-8777	-16.5	16286	22636	-6350	-12	16688
	20420	-3732	-7	17214	20880	-3666	-6.9	24363	20212	4151	7.8
	16006	23425	-7419	-13.9	15672	15051	621	1.2			

Denizli	22721	22120	601	0.7	20119	21706	-1587	-1.7	21863	23468	-1605
	-1.7	23454	23853	-399	-0.4	24446	21992	2454	2.6	27088	24039
	3049	3.2	28279	24771	3508	3.6	30185	24961	5224	5.3	28170
	24794	3376	3.4	28855	25466	3389	3.3	28047	28906	-859	-0.8
	26235	27146	-911	-0.9	22534	21370	1164	1.1			
Diyarbakır	31677	47777	-16100	-10.7	32384	43918	-11534	-7.6	34810	44858	
	-10048	-6.6	36622	46834	-10212	-6.5	30789	47575	-16786	-10.5	35466
	47949	-12483	-7.7	36920	48019	-11099	-6.8	36124	56025	-19901	-12
	39177	55604	-16427	-9.8	40493	48983	-8490	-5	43836	50822	-6986
	-4	42329	49730	-7401	-4.2	39101	42992	-3891	-2.2		
Edirne	11202	14249	-3047	-7.7	15183	12910	2273	5.8	14019	14830	-811
	-2.1	14494	14588	-94	-0.2	14813	13675	1138	2.9	16255	17000
	-745	-1.9	16431	17108	-677	-1.7	18779	17458	1321	3.3	16623
	17190	-567	-1.4	19125	15999	3126	7.7	18792	17162	1630	4
	16864	16761	103	0.2	12252	14264	-2012	-4.9			
Elazığ	15294	18813	-3519	-6.4	15900	18001	-2101	-3.8	15551	19438	-3887
	-7	16862	20054	-3192	-5.7	17108	18344	-1236	-2.2	21433	21303
	130	0.2	19724	24162	-4438	-7.8	22170	21699	471	0.8	18990
	20519	-1529	-2.6	20406	22054	-1648	-2.8	30152	24358	5794	9.8
	20343	29248	-8905	-15	15529	21166	-5637	-9.5			
Erzincan	10977	11976	-999	-4.7	11966	11189	777	3.6	14198	12365	
	1833	8.2	11523	14218	-2695	-12.4	12192	10649	1543	7.1	12662
	12884	-222	-1	15450	13641	1809	8.1	14062	16766	-2704	-12.1
	12666	12748	-82	-0.4	15162	14210	952	4.1	19727	17511	2216
	9.4	14496	17555	-3059	-12.9	10907	10998	-91	-0.4		
Erzurum	18999	43585	-24586	-31.2	24830	33681	-8851	-11.4	23294	35711	
	-12417	-16	27719	33599	-5880	-7.5	22551	33234	-10683	-13.6	23398
	39997	-16599	-21.4	27994	45209	-17215	-22.3	29624	41803	-12179	-15.8
	26957	36340	-9383	-12.2	26624	37358	-10734	-14	32501	35431	-2930
	-3.8	26819	40401	-13582	-17.7	20405	30220	-9815	-12.9		
Eskişehir	31731	21970	9761	13.2	32346	23225	9121	12.1	31662	26780	
	4882	6.4	35045	27908	7137	9.2	32363	25299	7064	9	33893
	28329	5564	7	35927	27977	7950	9.8	36041	28298	7743	9.4
	36687	25785	10902	13	38060	27799	10261	12	35995	32946	3049
	3.5	36757	29543	7214	8.2	25376	25533	-157	-0.2		
Gaziantep	37184	36229	955	0.6	36075	34125	1950	1.2	40380	36327	
	4053	2.4	45991	38634	7357	4.2	41672	39410	2262	1.3	41978
	42291	-313	-0.2	46438	44415	2023	1.1	46435	48858	-2423	-1.3
	45726	50003	-4277	-2.2	47431	50485	-3054	-1.5	44704	62206	-17502
	-8.6	52268	53517	-1249	-0.6	42040	41179	861	0.4		

Giresun	20089	18539	1550	3.7	16970	19567	-2597	-6.1	17474	20514	
	-3040	-7.2	17058	19346	-2288	-5.4	18027	17861	166	0.4	23453
	20170	3283	7.8	26857	23620	3237	7.6	22308	26942	-4634	-10.8
	35376	20284	15092	34.5	25529	34675	-9146	-20.7	40358	25953	14405
	32.2	22595	30580	-7985	-17.7	17654	17368	286	0.6		
Gümüşhane	9281	9073	208	1.6	8681	10444	-1763	-13.4	8848	10128	
	-1280	-9.8	10426	8988	1438	10.9	11166	9001	2165	16.1	16677
	11162	5515	39.8	18811	16071	2740	18.9	20437	17737	2700	18
	26681	18343	8338	49.7	18795	27348	-8553	-49	30333	22764	7569
	47.6	16158	25477	-9319	-55.1	7151	19116	-11965	-81		
Hakkari	7801	6981	820	3.2	4314	7949	-3635	-14.1	6013	9225	
	-3212	-12.7	8775	9133	-358	-1.3	5622	10137	-4515	-16	8007
	10293	-2286	-8.3	7187	10897	-3710	-13.3	8216	12983	-4767	-17
	5945	16067	-10122	-37.1	9997	10380	-383	-1.4	12756	10156	2600
	9.1	6898	12719	-5821	-20.5	7227	8421	-1194	-4.2		
Hatay	29459	32624	-3165	-2.2	28547	32296	-3749	-2.6	29752	33730	-3978
	-2.7	29509	37164	-7655	-5.2	27260	35139	-7879	-5.3	29067	39315
	-10248	-6.8	32678	39181	-6503	-4.3	32868	41519	-8651	-5.6	34195
	37627	-3432	-2.2	33745	40623	-6878	-4.4	36499	42655	-6156	-3.8
	35244	42067	-6823	-4.2	35160	29265	5895	3.6			
Isparta	17997	16800	1197	2.9	18633	15822	2811	6.7	15017	18255	-3238
	-7.2	15325	18269	-2944	-7.1	19731	14330	5401	13	18730	19807
	-1077	-2.6	17966	19930	-1964	-4.7	19042	18659	383	0.9	19899
	17240	2659	6.2	18222	18818	-596	-1.4	28789	19584	9205	21.1
	19555	20052	-497	-1.1	12879	17905	-5026	-11.4			
Mersin	46776	50110	-3334	-2.1	48377	49209	-832	-0.5	50430	51739	-1309
	-0.8	51328	54630	-3302	-2	46721	53523	-6802	-4	51468	55482
	-4014	-2.4	55598	54843	755	0.4	56171	58404	-2233	-1.3	54962
	52763	2199	1.2	54492	55393	-901	-0.5	55779	61917	-6138	-3.4
	55429	54452	977	0.5	53325	40242	13083	7			
İstanbul	374868		348193		26675	2.1	388467		348986		
	39481	3.1	439515		336932		102583		7.8	450445	
	328663		121782		9	384535		354074		30461	2.2
	437922		371601		66321	4.7	438998		424662		14336
	1	453407		402864		50543	3.5	369582		440889	
	-71307	-4.8	416587		422559		-5972	-0.4	385482		
	595803		-210321		-13.9	498676		378305		120371	
	7.8	328632		381654		-53022	-3.4				
İzmir	117067		89819	27248	7.2	116390		89517	26873	7	
	111255		99775	11480	2.9	110364		101420		8944	2.3
	105804		95954	9850	2.5	113673		99681	13992	3.5	

	124439	101447	22992	5.6	126238	105389	20849				
	5	122668	98902	23766	5.6	127394	102776	24618			
	5.8	130092	117113	12979	3	128370	106895				
	21475	4.9	107172	92400	14772	3.4					
Kars	8855	17726	-8871	-28	9028	15660	-6632	-21.4	9906	16657	-6751
	-22.1	10654	16376	-5722	-18.5	9706	16185	-6479	-21	11046	18072
	-7026	-23.1	10280	20020	-9740	-32.3	11328	19809	-8481	-28.6	10209
	16590	-6381	-21.8	11588	17119	-5531	-19	14441	16620	-2179	-7.5
	11132	16421	-5289	-18.4	8168	12910	-4742	-16.5			
Kastamonu	14775	14003	772	2.1	13047	14570	-1523	-4.2	13811	15422	
	-1611	-4.4	15023	15482	-459	-1.3	14340	13933	407	1.1	19911
	15658	4253	11.6	17539	18163	-624	-1.7	21601	18092	3509	9.5
	20577	18898	1679	4.5	18729	20052	-1323	-3.5	27175	18397	8778
	23.2	17737	22515	-4778	-12.5	14177	15328	-1151	-3.1		
Kayseri	30021	28621	1400	1.2	31075	28831	2244	1.9	37842	30380	
	7462	6.1	34387	31787	2600	2.1	33917	30143	3774	3	36580
	33789	2791	2.2	40957	33374	7583	5.8	37832	36087	1745	1.3
	35573	35540	33	0	38958	36863	2095	1.5	38690	48047	-9357
	-6.7	40579	39013	1566	1.1	31766	30033	1733	1.2		
Kırklareli	10565	11027	-462	-1.4	10359	11242	-883	-2.6	12844	12088	
	756	2.3	11979	11829	150	0.4	12548	11232	1316	3.9	14120
	13297	823	2.4	14553	13678	875	2.5	16098	13816	2282	6.6
	16882	13175	3707	10.6	16074	14023	2051	5.8	16936	14745	2191
	6.1	15352	14468	884	2.4	13852	11596	2256	6.3		
Kırşehir	9822	12312	-2490	-11.1	10947	12186	-1239	-5.5	9875	12626	
	-2751	-12.3	9598	11870	-2272	-10.2	9733	9898	-165	-0.7	11259
	11346	-87	-0.4	10648	12199	-1551	-6.9	11928	11544	384	1.7
	12188	11362	826	3.6	10840	11329	-489	-2.1	15835	12144	3691
	15.4	10853	13028	-2175	-8.9	9960	10835	-875	-3.6		
Kocaeli	63965	40947	23018	15.6	60432	48399	12033	7.9	64503	49379	
	15124	9.7	63314	50070	13244	8.3	62966	51561	11405	7	71159
	54742	16417	9.8	79697	55060	24637	14.4	87158	56927	30231	17.1
	85602	60479	25123	13.8	87796	60258	27538	14.7	80457	79970	487
	0.3	82977	65675	17302	8.9	75093	52543	22550	11.4		
Konya	45502	56760	-11258	-5.7	46042	51006	-4964	-2.5	47901	56729	-8828
	-4.4	52134	54533	-2399	-1.2	51981	48313	3668	1.8	51080	55006
	-3926	-1.9	55798	55453	345	0.2	56817	57056	-239	-0.1	56500
	54712	1788	0.8	53007	56594	-3587	-1.6	58300	60571	-2271	-1
	54219	58557	-4338	-1.9	43788	48780	-4992	-2.2			

Kütahya	14112	21711	-7599	-13.3	15291	18210	-2919	-5.1	15311	19923
	-4612	-7.8	17547	20655	-3108	-5.5	20550	16702	3848	6.7
	20915	-3040	-5.3	18734	21860	-3126	-5.5	19447	22337	-2890
	18382	20691	-2309	-4	17172	20393	-3221	-5.6	22582	20392
	3.8	21212	21456	-244	-0.4	15347	18688	-3341	-5.8	
Malatya	25500	27830	-2330	-3.2	25806	28208	-2402	-3.3	25320	30931
	-5611	-7.5	32925	29515	3410	4.5	24270	28545	-4275	-5.6
	33194	-7318	-9.6	29285	31476	-2191	-2.8	26925	31368	-4443
	27777	27613	164	0.2	27523	28222	-699	-0.9	36901	35234
	2.1	29880	33272	-3392	-4.2	24198	23329	869	1.1	
Manisa	38301	35458	2843	2.2	28781	34262	-5481	-4.1	32094	35497
	-2.5	33312	36021	-2709	-2	32211	34054	-1843	-1.4	36257
	-732	-0.5	35570	38432	-2862	-2.1	39558	36736	2822	2
	35685	4314	3.1	40257	36021	4236	3	41189	39056	2133
	40857	37164	3693	2.6	37003	29833	7170	5		
Kahramanmaraş	26142	25574	568	0.6	18521	26934	-8413	-8.1	21182	
	28592	-7410	-7.1	21716	31005	-9289	-8.8	19908	29467	-9559
	24560	29322	-4762	-4.4	27619	30903	-3284	-3	26856	32272
	-4.9	27307	28971	-1664	-1.5	26874	30314	-3440	-3	33619
	-42	0	27676	36700	-9024	-7.8	25056	23696	1360	1.2
Mardin	27606	41432	-13826	-18.2	18296	40308	-22012	-29.4	25478	30495
	-6.7	25447	31302	-5855	-7.6	21676	30299	-8623	-11.1	22596
	-6929	-8.8	22207	30796	-8589	-10.8	24255	34592	-10337	-12.9
	38823	-17384	-21.6	28303	30782	-2479	-3.1	30216	31738	-1522
	27395	32544	-5149	-6.1	25373	24744	629	0.7		
Muğla	35907	24959	10948	13.9	28390	26613	1777	2.2	35129	28850
	7.7	35409	29604	5805	6.9	33213	28301	4912	5.8	35246
	4559	5.3	48219	29671	18548	21	42112	37663	4449	4.9
	31520	7556	8.2	42450	33289	9161	9.8	52642	34302	18340
	42987	36210	6777	6.9	41468	28088	13380	13.5		
Muş	10058	25896	-15838	-38.4	10158	20182	-10024	-24.5	11782	17841
	-14.8	12895	18663	-5768	-13.8	9914	20646	-10732	-25.6	11070
	-9123	-21.9	12077	22688	-10611	-25.5	12752	23855	-11103	-26.8
	21683	-10374	-25.2	13094	22243	-9149	-22.4	15742	20789	-5047
	13396	20063	-6667	-16.2	11720	16548	-4828	-11.7		
Nevşehir	9531	10882	-1351	-4.8	8965	10667	-1702	-6	9231	12366
	-3135	-11	8995	11374	-2379	-8.4	10214	10859	-645	-2.3
	12147	-1825	-6.4	10164	13165	-3001	-10.4	11144	11955	-811
	11703	11091	612	2.1	11006	12076	-1070	-3.7	13120	12315
	2.7	12571	11899	672	2.2	9309	8958	351	1.2	

Niğde	15150	13991	1159	3.4	12588	15778	-3190	-9.3	12602	17666	-5064
	-14.9	11769	17023	-5254	-15.4	12359	13514	-1155	-3.4	14731	15493
	-762	-2.2	13785	17946	-4161	-12	13878	15959	-2081	-6	14823
	13589	1234	3.5	13948	15667	-1719	-4.9	22654	15676	6978	19.3
	14055	20769	-6714	-18.3	11349	13588	-2239	-6.2			
Ordu	27719	31458	-3739	-5.2	30335	31296	-961	-1.3	27896	36241	-8345
	-11.5	23963	34472	-10509	-14.6	48240	26595	21645	29.6	30792	46332
	-15540	-21	28555	39937	-11382	-15.6	29835	32600	-2765	-3.8	42995
	27229	15766	21.2	28111	40305	-12194	-16.3	54813	30152	24661	32.5
	25386	46640	-21254	-27.8	26898	21406	5492	7.2			
Rize	13253	13825	-572	-1.8	12428	14575	-2147	-6.7	13070	14819	-1749
	-5.5	14558	14560	-2	0	12315	13856	-1541	-4.7	16842	15859
	983	3	17311	17932	-621	-1.9	15514	18936	-3422	-10.3	15006
	15604	-598	-1.8	14561	16310	-1749	-5.3	31266	16413	14853	43.5
	15946	23534	-7588	-21.9	13143	13290	-147	-0.4			
Sakarya		24321	20887	3434	4	25376	21665	3711	4.3	26248	24627
	1621	1.9	28537	24633	3904	4.4	28457	23787	4670	5.2	30770
	27170	3600	3.9	32701	28311	4390	4.7	35943	28492	7451	7.8
	35228	28965	6263	6.4	35575	28122	7453	7.6	37659	32704	4955
	4.9	35941	31291	4650	4.5	30929	24888	6041	5.8		
Samsun		35404	40633	-5229	-4.2	38874	39581	-707	-0.6	35418	44825
	-9407	-7.5	35103	43408	-8305	-6.6	32249	41561	-9312	-7.4	38644
	42504	-3860	-3.1	41057	44519	-3462	-2.7	41718	44627	-2909	-2.3
	41687	41187	500	0.4	43709	42422	1287	1	53228	48543	4685
	3.5	45973	45020	953	0.7	43054	35442	7612	5.6		
Siirt	11870	12625	-755	-2.5	8475	11797	-3322	-10.9	8911	13973	-5062
	-16.7	10274	14228	-3954	-12.7	8823	14628	-5805	-18.5	10866	13339
	-2473	-7.8	10551	14366	-3815	-11.9	10505	16166	-5661	-17.5	10775
	15068	-4293	-13.2	12163	15406	-3243	-9.9	15017	15726	-709	-2.1
	11016	17252	-6236	-18.7	9044	12958	-3914	-11.8			
Sinop	10299	9472	827	4.1	9562	9558	4	0	11027	9967	1060
	5.2	10038	10618	-580	-2.9	8277	10371	-2094	-10.3	12085	9869
	2216	10.9	10783	11950	-1167	-5.7	10059	11274	-1215	-5.9	10190
	9784	406	2	11015	10086	929	4.5	20922	9793	11129	52
	12926	15642	-2716	-12.4	10292	11070	-778	-3.6			
Sivas	18871	30428	-11557	-18.1	23217	28585	-5368	-8.4	23467	31362	-7895
	-12.2	22884	30131	-7247	-11.5	21492	27441	-5949	-9.5	26354	30244
	-3890	-6.2	29726	34855	-5129	-8.2	25319	33501	-8182	-13.1	22966
	26369	-3403	-5.5	22147	28697	-6550	-10.5	45861	26271	19590	30.8
	23572	36430	-12858	-19.9	18490	23885	-5395	-8.4			

Tekirdağ	47534	22373	25161	33.2	37655	29066	8589	11	41307	29433
	11874	15	42265	28620	13645	16.6	42155	28042	14113	16.7
	31681	13632	15.7	52994	31266	21728	24.3	54482	33937	20545
	56536	32290	24246	25.2	55391	34658	20733	20.8	53895	41010
	12.6	48911	38109	10802	10.3	50764	28468	22296	20.8	
Tokat	29593	35892	-6299	-10.2	32655	34213	-1558	-2.5	25430	40995
	-24.9	29576	36342	-6766	-11.1	34725	31812	2913	4.8	26987
	-20084	-33	35891	38737	-2846	-4.7	34586	40805	-6219	-10.4
	33380	3192	5.3	34735	38746	-4011	-6.6	46391	41326	5065
	39530	39567	-37	-0.1	24840	41746	-16906	-27.9		
Trabzon	25918	27027	-1109	-1.5	36868	26474	10394	13.7	29130	36546
	-7416	-9.7	24858	38446	-13588	-17.8	21864	25478	-3614	-4.8
	29988	-4873	-6.4	31847	29741	2106	2.8	27314	32454	-5140
	28680	26775	1905	2.4	29703	29192	511	0.7	46593	33344
	16.5	29603	37502	-7899	-9.7	23761	24506	-745	-0.9	
Tunceli	6861	6068	793	9.2	4100	6205	-2105	-25	6400	6258
	142	1.9	6187	6419	-232	-2.7	5171	5404	-233	-2.7
	5612	2846	33.9	7286	6810	476	5.5	6996	8133	-1137
	4822	7031	-2209	-26.5	6074	6725	-651	-7.9	11211	5884
	62.3	6517	8179	-1662	-19.4	5791	6311	-520	-6.2	
Şanlıurfa	25510	37282	-11772	-7.5	27190	35154	-7964	-4.9	32555	37555
	-5000	-3	35888	41570	-5682	-3.3	31890	44878	-12988	-7.3
	47429	-14046	-7.8	35670	49030	-13360	-7.2	40135	50580	-10445
	38662	49789	-11127	-5.7	40058	52648	-12590	-6.3	44448	56748
	-6	41841	60885	-19044	-9.1	37720	47194	-9474	-4.5	
Uşak	8666	10222	-1556	-4.6	8693	9644	-951	-2.8	8998	9562
	-1.7	8653	10138	-1485	-4.4	10351	9435	916	2.7	10655
	162	0.5	10190	10895	-705	-2	11044	11001	43	0.1
	10701	1036	2.9	13909	10420	3489	9.6	13691	13011	680
	13117	12363	754	2	8756	10428	-1672	-4.5		
Van	21187	30275	-9088	-9	22866	27175	-4309	-4.2	23231	31312
	-7.8	23415	72273	-48858	-46.7	50003	46639	3364	3.2	32118
	-6389	-6	27587	44435	-16848	-15.4	30492	48061	-17569	-15.9
	49035	-20133	-18.1	31971	48269	-16298	-14.6	34691	44954	-10263
	34372	39575	-5203	-4.6	26621	38114	-11493	-9.9		
Yozgat	15352	30117	-14765	-30	17705	25546	-7841	-16	16181	29618
	-27.8	15251	26921	-11670	-24.7	14646	24525	-9879	-21.6	16688
	-9884	-22	16200	27722	-11522	-26.3	14933	26804	-11871	-27.9
	21825	-4089	-9.7	19074	23498	-4424	-10.5	25715	22524	3191
	18009	25253	-7244	-17.1	14109	17604	-3495	-8.3		

Zonguldak	17639	19530	-1891	-3	16258	20701	-4443	-7.1	15712	23267	
	-7555	-12.1	15822	23658	-7836	-12.7	14279	22687	-8408	-13.8	17006
	25571	-8565	-14.1	18115	24783	-6668	-11.1	18244	25340	-7096	-11.8
	19991	21933	-1942	-3.2	19421	22541	-3120	-5.2	24363	24105	258
	0.4	19654	26251	-6597	-11	15243	19593	-4350	-7.3		
Aksaray	11146	12596	-1450	-3.9	10217	12587	-2370	-6.3	10159	13225	
	-3066	-8.1	10778	13142	-2634	-6.2	10667	12374	-1707	-4.5	11491
	13726	-2235	-5.8	11936	14577	-2641	-6.8	12205	14164	-1959	-5.1
	15234	12294	2940	7.4	14139	13799	340	0.8	16680	14441	2239
	5.4	13689	15672	-1983	-4.8	12208	12311	-103	-0.2		
Bayburt	2996	4949	-1953	-25.5	3101	4420	-1319	-17.5	3984	4780	
	-796	-10.6	3997	4570	-573	-7.4	3664	4085	-421	-5.5	4787
	4677	110	1.5	8912	5583	3329	42.2	7150	9281	-2131	-26.8
	16981	6653	10328	121.5	5648	15820	-10172	-119	9450	7772	1678
	20.6	9131	7882	1249	14.8	4427	7433	-3006	-36		
Karaman	8904	8145	759	3.3	7853	8424	-571	-2.5	7478	8927	
	-1449	-6.2	7593	9365	-1772	-7.5	8191	8066	125	0.5	8747
	9445	-698	-2.9	8673	9788	-1115	-4.6	9195	9865	-670	-2.8
	9519	9813	-294	-1.2	8581	10340	-1759	-7.1	12777	10448	2329
	9.3	10318	11075	-757	-3	8074	8304	-230	-0.9		
Kırıkkale	11360	16605	-5245	-18.6	15861	15565	296	1.1	12488	19229	
	-6741	-24.1	12190	17482	-5292	-19.1	13776	13912	-136	-0.5	15089
	17818	-2729	-9.9	13306	18744	-5438	-19.9	12585	15204	-2619	-9.6
	14776	13300	1476	5.3	13434	14112	-678	-2.4	20575	15076	5499
	19.4	12937	18854	-5917	-20.7	8597	13021	-4424	-15.7		
Batman	16467	19669	-3202	-6.6	18326	16855	1471	3	19561	18839	
	722	1.4	18429	19902	-1473	-2.8	15906	20739	-4833	-9	18816
	20209	-1393	-2.5	18543	21515	-2972	-5.3	18872	23216	-4344	-7.6
	18757	21489	-2732	-4.7	19889	22323	-2434	-4.2	23099	22525	574
	1	20902	23325	-2423	-4	18598	17593	1005	1.6		
Şırnak	13223	15877	-2654	-6.2	8356	13025	-4669	-10.8	11733	13554	-1821
	-4.2	13403	13957	-554	-1.2	11075	14617	-3542	-7.6	13031	13753
	-722	-1.5	12749	16142	-3393	-6.9	11554	23615	-12061	-24.3	11102
	24122	-13020	-26.6	19672	15186	4486	9	19779	15485	4294	8.2
	16365	20724	-4359	-8.2	14461	15008	-547	-1			
Bartın	8418	6325	2093	11.4	7566	7104	462	2.5	6902	7859	-957
	-5.1	6560	7619	-1059	-5.6	7145	7330	-185	-1	7886	8133
	-247	-1.3	7794	8903	-1109	-5.8	8872	8113	759	4	8940
	8108	832	4.3	8759	8144	615	3.2	13399	8813	4586	23.3
	9423	10708	-1285	-6.5	7585	7441	144	0.7			

Ardahan	4640	7447	-2807	-24.7	3775	7033	-3258	-29.7	4586	6857	
	-2271	-21.3	4908	6673	-1765	-16.3	4923	5986	-1063	-9.9	5254
	7633	-2379	-22.9	5323	8033	-2710	-26.5	5630	7802	-2172	-21.6
	5164	6880	-1716	-17.3	5284	7154	-1870	-19.1	7342	6376	966
	9.8	5164	7028	-1864	-19	4085	5215	-1130	-11.7		
Iğdır	5559	9356	-3797	-20.4	5179	7986	-2807	-15.2	6667	8353	-1686
	-9.1	6915	8579	-1664	-8.8	6429	8632	-2203	-11.5	6882	9779
	-2897	-15.1	7407	9453	-2046	-10.6	7321	10589	-3268	-16.8	6721
	9368	-2647	-13.6	7775	9728	-1953	-10	9225	10122	-897	-4.5
	8650	9777	-1127	-5.6	7452	7266	186	0.9			
Yalova	16656	6542	10114	52.6	12269	10233	2036	10.1	10455	10984	-529
	-2.6	10267	9782	485	2.4	11353	9199	2154	10.2	14670	9808
	4862	22.3	14680	11064	3616	16.1	13834	11497	2337	10.1	12825
	10493	2332	9.7	15754	10719	5035	20.2	16232	12905	3327	12.8
	14531	12824	1707	6.3	14661	10281	4380	16			
Karabük	8383	8489	-106	-0.5	8511	9242	-731	-3.3	9036	8906	
	130	0.6	8239	9066	-827	-3.8	13510	8444	5066	22.8	13652
	11676	1976	8.6	13438	13195	243	1.1	17017	12567	4450	19
	16719	13416	3303	13.7	14740	14399	341	1.4	18384	17433	951
	3.8	14571	16485	-1914	-7.7	8148	12750	-4602	-18.7		
Kilis	4998	4611	387	3.2	4662	5677	-1015	-8.3	4813	5572	-759
	-6.1	4692	6420	-1728	-13.8	4315	6075	-1760	-14.1	6566	6161
	405	3.2	5759	7451	-1692	-13.1	6776	6593	183	1.4	5740
	8043	-2303	-17.5	6675	6526	149	1.1	8861	6929	1932	13.6
	6515	8843	-2328	-16.2	5035	6432	-1397	-9.7			
Osmaniye	17464	16338	1126	2.4	15167	16319	-1152	-2.4	16868	16928	
	-60	-0.1	15675	18420	-2745	-5.6	15343	17340	-1997	-4	17338
	17605	-267	-0.5	17880	18214	-334	-0.7	18567	19303	-736	-1.4
	19558	17683	1875	3.6	18842	19008	-166	-0.3	19965	20900	-935
	-1.7	18126	21459	-3333	-6.2	17734	14954	2780	5.1		
Düzce	10723	8913	1810	5.5	11752	9046	2706	8.1	11820	10893	927
	2.7	11253	10679	574	1.7	10432	10579	-147	-0.4	12367	11306
	1061	3	11980	12249	-269	-0.8	13434	12088	1346	3.7	15958
	11119	4839	13.2	15023	11387	3636	9.7	17445	12790	4655	12.1
	15454	14798	656	1.7	11757	10565	1192	3			

KAYNAKÇA

Kitaplar

ALP Selçuk, ÖZ Ersoy, *Makine Öğrenmesinde Sınıflandırma Yöntemleri ve R Uygulamaları*, Nobel Akademik Yayıncılık, 2020

ALPAR Reha, *Uygulamalı Çok Değişkenli İstatistiksel Yöntemler*, Detay Yayıncılık, 2013

BAYRAM Nuran, *Sosyal Bilimlerde SPSS İle Veri Analizi*, Bursa: Ezgi Kitabevi, 2017

CAMPESATO Oswald, *Python 3 For Machine Learning*, Dulles, VA: Mercury Learning and Information, 2020

ÇOKLUK Ömay, ŞEKERCİOĞLU Güçlü, BÜYÜKÖZTÜRK Şener, *Sosyal Bilimler İçin Çok Değişkenli İstatistik SPSS ve LISREL Uygulamaları*, Ankara: Pegem Akademi, 2018

HAIR Joseph, BLACK William, BABIN Barry, ANDERSON Rolph, *Multivariate Data Analysis (7th Edition)*, USA:Pearson, 2006

HASSANIEN Aboul Ella, BHATNAGAR Roheet, DARWISH Ashraf, *Advances In Intelligent Systems And Computing/ Advanced Machine Learning Technologies And Applications*, Singapore:Springer Nature Singapore Pte Ltd., 2020

HENNIG Christina, MEILA Marina, MURTAGH Fionn, ROCCI Roberto, *Handbooks Of Modern Statistical Methods*, Boca Raton, London, New York: CRC Press, 2016

JOLLIFFE Ian, *Principal Component Analysis*, New York: Springer-Verlag, 2002

ÖZDAMAR Kazım, *Paket Programlar İle İstatistiksel Veri Analizi (Çok Değişkenli Analizler)*, Eskişehir: Kaan Kitabevi, 2004

SERPER Özer, *Uygulamalı İstatistik*, 8. Baskı, Bursa: Ezgi Kitabevi, 2017

SMITH Chris, KONING Mark, *Decision Trees and Random Forests: A Visual Introduction For Beginners*, USA: Blue Windmill Media, 2017

TARI Recep, *Ekonometri*, 6. Baskı, Kocaeli: Umuttepe Yayınları, 2010

WIERZCHON Slawomir, KLOPOTEK Mieczyslaw, *Studies In Big Data 34 / Modern Algorithms Of Cluster Analysis*, Cham, Switzerland : Springer International Publishing AG, 2018

WITTEN Ian, FRANK Eibe, HALL Mark, *Data Mining: Practical Machine Learning Tools and Techniques, Third Edition*, San Francisco:Elsevier, 2011

WLODARCZAK Peter, *Machine Learning And Its Applications*, Australia:CRC Press, 2020

Makaleler

ALPTEKİN Nesrin, YEŞİLAYDIN Gözde, “OECD Ülkelerinin Sağlık Göstergelerine Göre Bulanık Kümeleme Analizi İle Sınıflandırılması”, *İşletme Araştırmaları Dergisi*, Cilt:7, Sayı:4, ss.137-155, 2015

BİRANT Derya, “Farklı Bağlantı Yöntemleri İle Hiyerarşik Kümeleme Topluluğu”, *Selçuk Üniversitesi Mühendislik Bilim Ve Teknik Dergisi*, Cilt:7, Sayı:1, ss.154-164, 2019

ÇAĞLAYAN AKAY Ebru, “Ekonometride Yeni Bir Ufuk: Büyük Veri Ve Makine Öğrenmesi”, *Marmara University Social Sciences Research Journal*, Cilt:2, Sayı:7, ss.41-53, İstanbul, 2018

ÇALIŞ Aslı, KAYAPINAR Sema, ÇETİNYOKUŞ Tahsin, “Veri Madenciliğinde Karar Ağacı Algoritmaları İle Bilgisayar Ve İnternet Güvenliği Üzerine Bir Uygulama”, *Gazi Üniversitesi Endüstri Mühendisliği Dergisi*, Cilt:25, Sayı:3-4, ss.2-19, Ankara, 2014

ÇELİK Fatih, “Türkiye’de İç Göçler: 1980-2000”, *Sosyal Bilimler Enstitüsü Dergisi*, Kayseri, Sayı:22, ss.87-109, 2007

- ERSÖZ Filiz, ERSÖZ Taner, UYSAL Fatma Nur, “Türkiye’deki illerin Yaşam Endeksinin Çok Değişkenli İstatistiksel Yöntemlerle İncelenmesi”, *Ekonomi Bilimler Dergisi*, ISSN: 1309-8020, Cilt:9 No:1,ss. 49-65, 2017
- ERSUNGUR Ş. Mustafa, KIZILTAN Alaattin, POLAT Özgür, “Türkiye’de Bölgelerin Sosyo-Ekonomik Gelişmişlik Sıralaması: Temel Bileşenler Analizi”, *İktisadi Ve İdari Bilimler Dergisi*, Cilt:21, Sayı:2, ss.55-66, 2007
- GÜREŞÇİ Ertuğrul, “Türkiye’de Köyden Kente Göç Ve Düşündürdükleri”, *Sosyo Ekonomi Dergisi*, Sayı:15, ss.125-135, 2011
- IŞIĞIÇOK Erkan, “Performans Ölçümü, Yönetimi ve İstatistiksel Analizi”, *Ekonomi ve İstatistik Dergisi*, Sayı: 7, ss.1-23. 2008
- KOLTAN YILMAZ Şebnem, PATIR Said, “Kümeleme Analizi Ve Pazarlamada Kullanımı”, *Akademik Yaklaşımlar Dergisi*, Cilt:2, Sayı:11, ss.91-113, 2011
- ÖZDEMİR Hakan, “Türkiye’de İç Göçler Üzerine Genel Bir Değerlendirme”, *Akademik Bakış Dergisi*, Sayı:30, Celalabat, 2012
- SAĞIROĞLU Ali Zafer, “Türkiye’nin Değişen Göç Karakteri”, *Sosyoloji Divanı Dergisi*, Sayı:6, ss.9-30, 2015
- SARIMAN Güncel, “Veri Madenciliğinde Kümeleme Teknikleri Üzerine Bir Çalışma”, *Süleyman Demirel Üniversitesi Fen Bilimleri Enstitüsü Dergisi*, Cilt:15, Sayı:3, ss.192-202, Isparta, 2011
- SİRKECİ İbrahim, YÜCEŞAHİN M. Murat, “Türkiye’de göç Çalışmaları”, *Göç Dergisi*, Cilt:1, Sayı:1, ss.1-10, 2014
- SÖNMEZ Harun, ER Fikret, “Türkiye’de İllere Göre İç Göç Hareketlerinin Modern Kümeleme Teknikleri İle İncelenmesi”, *Eskişehir Osmangazi Üniversitesi Müh.Mim.Fak.Dergisi*, Eskişehir, C.XX, S.1, 2007
- ŞANTAŞ Gülcan, “Türkiye’de İç Göçün Dağılımı”, *Sosyal Politika Çalışmaları Dergisi*, Sayı:45, ss.893-918, 2019
- ŞEN Mustafa, “Türkiye’de İç Göçlerin Neden Ve Sonuç Kapsamında İncelenmesi”, *Çalışma Ve Toplum Dergisi*, Sayı:1,ss.231-256, 2014

- TEKİN Bilgehan, “Ward, K-means Ve İki Adımlı Kümeleme Analizi Yöntemleri İle Finansal Göstergeler Temelinde Hisse Senedi Tercihini”, *Balıkesir Üniversitesi Sosyal Bilimler Enstitüsü Dergisi*, Cilt:21, Sayı:40, ss.401-436, Balıkesir, 2018
- YILANCI Veli, “Bulanık Kümeleme Analizi İle Türkiye’deki İllerin Sosyoekonomik Açısından Sınıflandırılması”, *Süleyman Demirel Üniversitesi İktisadi Ve İdari Bilimler Fakültesi Dergisi*, Cilt:15, Sayı:3, ss.453-470, Isparta, 2010
- YILMAZ Murat, “Türkiye’de Kırsal Nüfusun Değişimi Ve İllere Göre Dağılımı: (1980-2012)”, *Doğu Coğrafya Dergisi*, Sayı:33, ss.161-188, 2015

Diğer Yayınlar

- AKIŞOĞLU Berkay, *Predicting Economic Growth Using Machine Learning Techniques And Sentiment Analysis*, (Yüksek Lisans Tezi), İstanbul: TED Üniversitesi Lisansüstü Programlar Enstitüsü, 2019
- AKPINAR Nesrin, *Makine Öğrenmesi Teknikleriyle Kredi Başvuru Skor Kartının Oluşturulması*, (Yüksek Lisans Tezi), İstanbul: Yıldız Teknik Üniversitesi Fen Bilimleri Enstitüsü, 2019
- AKTAŞ Sezer, *Dow Jones Endüstri Ortalaması Ve Nasdaq 100 Endekslerinin Makine Öğrenmesi Algoritmaları İle Tahmini*, (Yüksek Lisans Tezi), İstanbul: İstanbul Teknik Üniversitesi Sosyal Bilimler Enstitüsü, 2019
- ALTINOK Yusuf, *Veri Madenciliğinde Hiyerarşik Kümeleme Algoritmalarının Uygulamalı Karşılaştırılması*, (Yüksek Lisans Tezi), İstanbul: Marmara Üniversitesi Sosyal Bilimler Enstitüsü, 2019
- BAŞAR Yavuz, *Türkiye’de İller Bazında İç Göçü Belirleyen Faktörlerin Ekonometrik Analizi:2008-2013*, (Yüksek Lisans Tezi), Trabzon: Karadeniz Teknik Üniversitesi Sosyal Bilimler Üniversitesi, 2015

- CEYLAN Tuğçe, *Perakende Sektöründe makine Öğrenmesine Dayalı Yaklaşımlar*, (Yüksek Lisans Tezi), İstanbul: Yıldız Teknik Üniversitesi Fen Bilimleri Enstitüsü, 2018
- DEMİRTAŞ Özge, *Türkiye'deki İç Göçün Yapay Sinir Ağları İle Demografik Açından İncelenmesi*, (Yüksek Lisans Tezi), Ankara: Gazi Üniversitesi Sosyal Bilimler Enstitüsü, 2017
- DENİZLİ Zahide, *Veri Madenciliği Modelleri Ve Örnek Bir Uygulama*, (Yüksek Lisans Tezi), İzmir: Dokuz Eylül Üniversitesi Sosyal Bilimler Enstitüsü, 2019
- ELASAN Sadi, *Veri Madenciliğinde Farklı Karar Ağaçları Ve K-En Yakın Komşuluk Yöntemlerinin İncelenmesi: Kadın Hastalıkları Ve Doğum Verisinde Bir Uygulama*, (Doktora Tezi), Van: Van Yüzüncü Yıl Üniversitesi Sağlık Bilimleri Enstitüsü, 2019
- ELMAZ Furkan, *A Machine Learning Approach To Biomass Gasification Process*, (Yüksek Lisans Tezi), İzmir: İzmir Katip Çelebi Üniversitesi Fen Bilimleri Enstitüsü, 2020
- ERÇEVİK Hande, *Türkiye'deki İç Göçlerin Nedenlerinin Log Lineer Analiz İle İncelenmesi*, (Yüksek Lisans Tezi), İstanbul: Marmara Üniversitesi Sosyal Bilimler Enstitüsü, 2013
- FİLİZ Enes, *Makine Öğrenmesi Yöntemleri Ve Eğitim Verisi Üzerine Bir Uygulama: Uluslararası Matematik Ve Fen Eğilimleri Araştırması 2015 Türkiye Örneği*, (Doktora Tezi), İstanbul: Yıldız Teknik Üniversitesi Fen Bilimleri Enstitüsü, 2019
- FİLİZ Şükriye Elif, *Internal Migration Patterns Of Turkey*, (Yüksek Lisans Tezi), İstanbul: Marmara Üniversitesi Sosyal Bilimler Enstitüsü, 2008
- GASIM Nicat, *Bayesyen Model İle Doğrusal Regresyon Modellerinin Karşılaştırılması Üzerine Bir Uygulama*, (Yüksek Lisans Tezi), İzmir: Dokuz Eylül Üniversitesi Sosyal Bilimler Enstitüsü, 2013
- GÖKHAN Ali, *The Determinants Of Internal Migration In Turkey*, (Yüksek Lisans Tezi), İstanbul: Sabancı Üniversitesi Sosyal Bilimler Enstitüsü, 2008

- GÖRENTAŞ Muhammed Burak, *Doğu Anadolu Bölgesi İllerinin Sosyo Ekonomik Gelişmişlik Düzeyine Göre Kümeleme Analizi İle Sınıflandırılması*, (Yüksek Lisans Tezi), Van: Van Yüzüncü Yıl Üniversitesi Fen Bilimleri Enstitüsü, 2017
- GÖRGÜN Mertcan, *Makine Öğrenmesi Yöntemleri İle Kalp Hastalığının Teşhis Edilmesi*, (Yüksek Lisans Tezi), İstanbul: İstanbul Aydın Üniversitesi Lisansüstü Eğitim Enstitüsü, 2020
- GÜNEŞ Oğuzhan Mehmet, *Makine Öğrenmesi Yöntemleri Kullanılarak Karaciğer Sirozu Hastalığını Etkileyen Faktörlerin Sınıflandırılması*, (Yüksek Lisans Tezi), Samsun: Ondokuz Mayıs Üniversitesi Lisansüstü Eğitim Enstitüsü, 2020
- İŞİK Hanife, *Sales Forecasting In Fashion Retail Industry With Classical And Machine Learning Methods*, (Yüksek Lisans Tezi), İstanbul: İstanbul Teknik Üniversitesi Sosyal Bilimler Enstitüsü, 2020
- KARAİBRAHİMOĞLU Adnan, *Veri Madenciliğinden Birliktelik Kuralı İle Onkoloji Verilerinin Analiz Edilmesi: Meram Tıp Fakültesi Onkoloji Örneği*, (Doktora Tezi), Konya: Selçuk Üniversitesi Fen Bilimleri Enstitüsü, 2014
- KESKİN Adem Kürşat, *Makine Öğrenmesi Sınıflandırma Algoritmalarının İncelenmesi*, (Yüksek Lisans Tezi), Sinop: Sinop Üniversitesi Fen Bilimleri Enstitüsü, 2018
- KESKİN Mustafa Vahit, *Büyük Veride Makine Öğrenmesi Uygulaması*, (Yüksek Lisans Tezi), İstanbul: Yıldız Teknik Üniversitesi Fen Bilimleri Enstitüsü, 2018
- KOLDERE AKIN Yasemin, *Veri Madenciliğinde Kümeleme Algoritmaları Ve Kümeleme Analizi*, (Doktora Tezi), İstanbul: Marmara Üniversitesi Sosyal Bilimler Enstitüsü, 2008
- NASIRI Mehdi, *Automated Lane Change Decision Making For Autonomous Vehicles Using Machine Learning Techniques*, (Yüksek Lisans Tezi), İstanbul: İstanbul Teknik Üniversitesi Lisansüstü Eğitim Enstitüsü, 2020
- ÖNDER Zeynep Karal, *Türkiye’de İç Göç Ve Kamu Politikaları İlişkisi*, (Doktora Tezi), Eskişehir: Anadolu Üniversitesi Sosyal Bilimler Enstitüsü, 2018

- ÖZDEMİR Ozancan, *Performance Comparison Of Machine Learning Methods And Traditional Time Series Methods For Forecasting*, (Yüksek Lisans Tezi), Ankara: Orta Doğu Teknik Üniversitesi Doğal Ve Uygulamalı Bilimler, 2020
- ÖZTÜRK Sinem, *Hızlı Moda Sektöründe Makine Öğrenmesi Yöntemleri İle Satış Miktarlarının Tahmin Edilmesi*, (Yüksek Lisans Tezi), İstanbul: İstanbul Teknik Üniversitesi Fen Bilimler Enstitüsü, 2020
- POYRAZ İlker, *Makine Öğrenmesi Algoritmaları İle Eczaneler İçin İlaç Talep Tahmini*, (Yüksek Lisans Tezi), İstanbul: İstanbul Aydın Üniversitesi Lisansüstü Eğitim Enstitüsü, 2020
- SAYILAN Ali Burak, *Veri Madenciliğinde bazı Kümeleme Algoritmalarının Karşılaştırılması*, (Yüksek Lisans Tezi), Muğla: Muğla Sıtkı Koçman Üniversitesi Fen Bilimleri Enstitüsü, 2019
- SELVİ Ayşegül, *Bilecik İlinde İlköğretimden Liseye Geçiş Sınavlarında Makine Öğrenmesi Yöntemleri İle Öğrenci Başarısının Tahmini*, (Yüksek Lisans Tezi), Bilecik: Bilecik Şeyh Edebali Üniversitesi Fen Bilimleri Enstitüsü, 2020
- ŞAHİNBAŞ Kevser, *Dengesiz Klinik Veriler İçin Karar Destek Önerisi: Akut Apandisit Örneği*, (Doktora Tezi), İstanbul: İstanbul Üniversitesi Fen Bilimleri Enstitüsü, 2019
- ŞEN Buğra, *Estimating Instant Fuel Consumption By Machine Learning And Improving Fuel Consumption*, (Yüksek Lisans Tezi), İstanbul: Galatasaray Üniversitesi Fen Bilimleri Enstitüsü, 2020
- TATLI Suna, *Mekânsal Ekonometrik Modeller Ve Türkiye'de İç Göçün Belirleyicilerinin Analizi*, (Yüksek Lisans Tezi), İstanbul: İstanbul Üniversitesi Sosyal Bilimler Enstitüsü, 2016
- TUNÇ Harun Bora, *Türkiye'de İç Göçlerin Shift Share Analizi İle İncelenmesi*, (Yüksek Lisans Tezi), Isparta: Süleyman Demirel Üniversitesi Sosyal Bilimler Enstitüsü, 2018

TÜRKOĞLU Batuhan Burç, *Predictive Analysis Of Flight Traffic At New York JFK Airport On Air Pollution Using Machine Learning*, İstanbul: İstanbul Teknik Üniversitesi Çevre Bilimleri Enstitüsü, 2010

TYODEN Bala Mikat, *Clustering And Classification Methods Using Unsupervised, Semi-Supervised And Supervised Algorithms*, (Doktora Tezi), Adana: Çukurova Üniversitesi Doğal Ve Uygulamalı Bilimler Enstitüsü, 2016

UÇAN Ömer, *Öneri Sistemi Modellerinde Kullanılan Makine Öğrenmesi Tekniklerinin Müşteri Satın Alma Tercihleri Doğrultusunda Karşılaştırılması*, (Doktora Tezi), Antalya: Akdeniz Üniversitesi Sosyal Bilimler Enstitüsü, 2020

<https://yz-ai.github.io/blog/pekistirmeli-ogrenme/pekistirmeli-ogrenme-bolum-1>, (15.11.2020)

<https://bookdown.org/ugurdar/dogrusalregresyon/basit-do%C4%9Frusal-regresyon.html>, (18.11.2020)

<https://medium.com/@ekrem.hatipoglu/machine-learning-prediction-algorithms-polynomial-regression-part-4-6c62b4240b53>, (23.11.2020)

<https://medium.com/@k.ulgen90/lojistik-regresyon-makine-%C3%B6%C4%9Frenimi-b%C3%B6l%C3%BCm-7-c6bc685a4084>, (01.12.2020)

Pirim , A . (2006). YAPAY ZEKA . Journal of Yaşar University , 1 (1) , 81-93 . Retrieved from <https://dergipark.org.tr/tr/pub/jyasar/issue/19113/202842>, (02.12.2020)

<https://bilimfili.com/dunyayi-degistirmekte-olan-yapay-sinir-aglari-nedir>, (17.12.2020)

<https://www.getrevue.co/profile/basakesin/issues/makine-ogrenmesi-uzerine-haftalik-notlar-sayi8-karar-agaclari-ve-daha-fazlasi-152086>, (20.12.2020)

<https://devhunteryz.wordpress.com/2018/09/20/rastgele-ormanrandom-forest-algoritmasi/>, (30.12.2020)

https://www.wikiwand.com/tr/Destek_vekt%C3%B6r_makinesi, (02.01.2021)

<https://medium.com/@k.ulgen90/makine-%C3%B6%C4%9Frenimi-b%C3%B6l%C3%BCm-2-6d6d120a18e1>, (05.01.2021)

<https://docplayer.biz.tr/15448102-K-en-yakin-komsu-algoritmasi-ve-bir-uygulama-kredi-riskini-siniflandirma.html>, (10.01.2021)

https://tr.wikipedia.org/wiki/K%C3%BCmeleme_analizi#/media/Dosya:Cluster-2.svg, (19.01.2021)

<https://www.ahmetcevahircinar.com.tr/2017/03/25/k-kez-capraz-dogrulama-yontemi/>, (22.01.2021)

<https://veribilimcisi.com/2017/07/18/roc-egrisi-roc-curve/>, (30.01.2021)

<https://avys.omu.edu.tr/storage/app/public/vceyhan/125687/K%C3%BCmele%20Analizi,%20PDF.pdf>, (02.02.2021)

<http://www.analitikgeometri.com/ders-1-noktanin-analitik-incelesmesi.html>, (05.02.2021)