

T.C.
SAKARYA ÜNİVERSİTESİ
BİLGİSAYAR VE BİLİŞİM BİLİMLERİ FAKÜLTESİ

BSM 401 BİLGİSAYAR MÜHENDİSLİĞİ TASARIMI

SHARED MEMORY
IPC
(Interprocess Communication)

B221210376 - Ersin ALÇİN

Bölüm : **BİLGİSAYAR MÜHENDİSLİĞİ**
Danışman : **Dr. Öğr.Üyesi ALİ GÜLBAĞ**

2022-2023 Yaz Dönemi

ÖNSÖZ

Bilgi teknolojilerinin hızla geliştiğı günümüzde, veri analizi ve örüntü tanıma gibi konular önemli bir yer tutmaktadır. Özellikle büyük veri kümesinin işlenmesi, boyut azaltma ve anlamlı bilgilerin çıkarılması, veri bilimcilerin ve mühendislerin en büyük zorluklarından biri haline gelmiştir. Bu noktada, PCA (Principal Component Analysis - Temel Bileşen Analizi) gibi çok değerli bir istatistiksel yöntem, veri kümesinin boyutunu azaltarak veri içerisindeki temel yapının anlaşılmasına yardımcı olabilir.

İÇİNDEKİLER

SİMGELER VE KISALTMALAR LİSTESİ.....	iv
BÖLÜM 1. GİRİŞ.....	1
1.1. PCA Nedir?.....	1
1.2. PCA'nın Temel İlkeleri.....	2
1.2.1. PCA'nın Uygulama Alanları.....	2
1.2.2. PCA Analizi Aşamaları.....	2
1.2.3. PCA'nın Avantajları.....	3
1.2.4. PCA'nın Dezavantajları.....	4
BÖLÜM 2. SİSTEMATİK YAKLAŞIM	5
2.1. Formülasyon.....	5
2.1.1. Veri Hazırlığı:	5
2.1.2. Kovaryans Matrisi Hesaplaması:.....	5
2.1.3. Özdeğer ve Özvektör Analizi:	6
2.1.4. Özdeğerleri Sıralama ve Temel Bileşen Seçimi	6
2.1.5. Yeri Veri Seti Oluşturma.....	6
2.2. Yazılım Mimarisi.....	6
2.2.1. Katmanlı Mimari	6
2.2.2. Kullanılan Teknolojiler.....	7
2.2.3. Adım Adım İş Akışı.....	7
2.2.4. Örnek Senaryo.....	7
BÖLÜM 3. .NET Core MVC ile PCA Analizini Gerçekleştirme Süreci ve Veri Boyutunu Azaltma 9	9
3.1. Deney Düzenegi	9
3.2. Sanal Laboratuvar Karakteristiği.....	10
3.3. Test Aşaması.....	11
BÖLÜM 4. VERİ GÜVENLİĞİ DEĞERLENDİRMESİ.....	14
4.1. Veri İşleme ve Depolama Güvenliği	14
4.2. Veri Anonimleştirme ve Maskelendirme	14
BÖLÜM 5. SONUÇLAR VE ÖNERİLER.....	15

SİMGELER VE KISALTMALAR LİSTESİ

MVC	: Model-View-Controller
PCA	: Principal Component Analysis (Temel Bileşenler Analizi)

ŞEKİLLER LİSTESİ

Şekil 2.1.	Veri standartlaştırma formülü.....	18
Şekil 2.2.	Kovaryans Matrisi Hesaplama.....	19
Şekil 2.3.	Özdeğer ve Özvektör Analizi.....	27
Şekil 2.4.	Yeni Veri Seti.....	28
Şekil 3.1.	CSV yükleme.....	28
Şekil 3.2.	Veri seti analizi.....	28
Şekil 3.3.	Accord.NET PCA Analizi.....	28
Şekil 3.4.	PCA Analizi Sonuç Ekranı.....	28

TABLÖLAR LİSTESİ

Tablo 3.1.	E-ticaret ürün bilgileri.....	4
Tablo 3.2.	Öğrenci bilgileri.....	19

ÖZET

Anahtar kelimeler: .NET Core MVC, PCA, Accord.NET, Veri analizi

Bu tasarım çalışması, .NET Core MVC çerçevesi kullanılarak PCA (Principal Component Analysis - Temel Bileşen Analizi) yönteminin tasarımını öne çıkarır. Ana hedefi, PCA'nın veri boyutunu azaltma, temel yapıyı anlama ve anlamlı bilgileri çıkarma yeteneklerini göstermektir. Aynı zamanda Accord.NET kütüphanesinin etkili kullanımını sergileyerek, veri analizi projelerindeki başarıya odaklanır.

Bu tasarım çalışması, Accord.NET kütüphanesinin .NET Core MVC uygulamalarına entegrasyonunu adım adım gösterir. İlk olarak, PCA'nın temel teorik kavramları hızlı bir şekilde açıklanır. Ardından, veri ön işleme adımları sonrasında PCA analizinin nasıl gerçekleştirileceği detaylandırılır. Elde edilen temel bileşenler ve varyans oranları, veri setinin özünün daha iyi anlaşılmasını sağlar.

Sonuç olarak, bu tasarım çalışması, veri analitiği ve örüntü tanıma alanındaki profesyonellerin, .NET Core MVC ile PCA analizi tasarlamayı ve Accord.NET kütüphanesini bu süreçte nasıl kullanacaklarını öğrenmelerine yardımcı olmayı amaçlar. Bu tasarımın amacı, kullanıcıların veri analizi projelerinde PCA'nın gücünden tam anlamıyla yararlanmalarına olanak sağlamaktır.

BÖLÜM 1. GİRİŞ

Veri analizi, günümüzde birçok alanda hayati bir rol oynamaktadır. Büyük veri setleriyle çalışan bilim insanları, mühendisler ve analistler, bu verilerdeki anlamlı bilgileri çıkarmak ve örüntüleri anlamak için çeşitli istatistiksel ve matematiksel yöntemlere başvurmaktadır. Bu yöntemlerden biri de PCA (Principal Component Analysis - Temel Bileşen Analizi) olarak adlandırılan güçlü bir tekniktir.

1.1. PCA Nedir?

Türkçesi “Temel Bileşenler Analizi” olan PCA tanıma, sınıflandırma, görüntü sıkıştırma alanlarında kullanılan yararlı bir istatistiksel tekniktir. Temel amacı yüksek boyutlu verilerde en yüksek varyans ile veri setini tutmak ancak bunu yaparken boyut indirgemeyi sağlamak olan bir tekniktir. Fazla boyutlu verilerdeki genel özellikleri bularak boyut sayısının azaltılmasını, verinin sıkıştırılmasını sağlar. Boyut azalmasıyla bazı özelliklerin kaybedileceği kesindir; fakat amaçlanan, bu kaybolan özelliklerin popülasyon hakkında çok az bilgi içeriyor olmasıdır. Bu yöntem, yüksek korelasyonlu değişkenleri bir araya getirerek, verilerdeki en çok varyasyonu oluşturan “temel bileşenler” olarak adlandırılan daha az sayıda yapay değişken kümesi oluşturur.

PCA verideki gerekli bilgileri ortaya çıkarmada oldukça etkili bir yöntemdir. PCA’nın arkasında yatan temel mantık çok boyutlu bir veriyi, verideki temel özellikleri yakalayarak daha az sayıda değişkenle göstermektir.

1.2. PCA'nın Temel İlkeleri

PCA, veri setinin içerdği değişkenler arasındaki ilişkileri anlamamıza yardımcı olurken aynı zamanda boyut azaltma sağlar. Özünde, PCA, veri noktalarını yeni bir koordinat sistemi içerisinde temsil ederek verinin varyansını en yüksek olan yönleri bulmaya çalışır. Bu yöntem, verilerdeki temel özellikleri yakalamamıza ve anlamamıza yardımcı olurken, azaltılmış boyutlardaki veri ile çalışmayı daha etkili hale getirir.

1.2.1. PCA'nın Uygulama Alanları

PCA, birçok farklı alanda kullanım bulur. Örneğin, görüntü işleme alanında, görüntülerdeki özellikleri sıkıştırmak veya benzerlikleri bulmak için kullanılabilir. Ayrıca biyomedikal verilerin analizinde, pazarlama stratejilerinin geliştirilmesinde, ekonomik analizlerde ve daha birçok alanda kullanılabilir. PCA analizi, verilerdeki önemli varyansları belirleyerek veri setinin temel yapılarını açığa çıkarabilir.

Bu yazıda, PCA analizinin temel prensiplerini ve önemini anlamaya odaklanacağız. PCA'nın nasıl çalıştığını, hangi durumlarda kullanılabileceğini ve veri analizi projelerindeki potansiyel etkilerini inceleyerek, bu güçlü analitik aracının nasıl kullanıldığını keşfedeceksiniz.

1.2.2. PCA Analizi Aşamaları

1. Veri Hazırlığı: Analiz yapmak istediğiniz veriyi hazırlayın ve gerekiyorsa standartlaştırın.
2. Kovaryans Matrisi Hesaplaması: Veri değişkenleri arasındaki ilişkileri ifade eden kovaryans matrisini hesaplayın.
3. Özdeğer ve Özvektör Hesaplamaları: Kovaryans matrisinin özdeğerlerini ve özvektörlerini hesaplayın.

4. **Özdeğerleri Sıralama:** Özdeğerleri büyükten küçüğe sıralayarak temel bileşenleri belirleyin.
5. **Temel Bileşenleri Seçme:** Özdeğerleri sıraladıktan sonra, belirli bir varyansı korumak veya belirli sayıda temel bileşen seçmek için eşik değerleri belirleyin.
6. **Yeni Veri Seti Oluşturma:** Seçtiğiniz temel bileşenleri kullanarak yeni bir veri seti oluşturun.
7. **Analiz ve Yorumlama:** Yeni veri setini kullanarak analizlerinizi gerçekleştirin, temel bileşenleri yorumlayın ve sonuçları değerlendirin.
8. **Sonuçların Sunumu:** Elde ettiğiniz sonuçları açıklayıcı bir şekilde sunarak PCA analizi sonuçlarını aktarın.

1.2.3. PCA'nın Avantajları

1. **Boyut Azaltma:** Veri setlerinin boyutunu azaltarak, daha az değişken içeren ve daha anlaşılır veri setleri oluşturabilir. Bu, veri analizini daha etkili hale getirir ve hesaplama yükünü azaltır.
2. **Veri Basitleştirme:** Temel bileşen analizi, veri setlerini daha az değişken içeren bir uzayda temsil ederken ana yapıyı korur. Bu, gereksiz gürültüyü filtreleyebilir ve veriyi daha anlamlı bir hale getirebilir.
3. **Veri Görselleştirme:** Temel bileşenler, veri değişkenlerinin kombinasyonları olarak ifade edildiği için, temel bileşen analizi veriler arasındaki ilişkileri görselleştirmeyi kolaylaştırır. Özellikle yüksek boyutlu veri setlerinde, bu görselleştirmeler veri anlamını artırır.
4. **Veri Özünü Kavrama:** PCA, veri setlerindeki temel yapıları belirlemek ve değişkenler arasındaki ilişkileri anlamak için kullanılır. Bu, verilerin gerçek özünü kavramada yardımcı olabilir ve daha anlamlı yorumlar yapmanıza imkan tanır.
5. **Gürültü Azaltma:** PCA, veri setlerindeki gürültüyü azaltmada etkili olabilir. Yüksek varyanslı bileşenler ana özellikleri temsil ederken, düşük varyanslı bileşenler gürültüyü içerebilir. Bu sayede daha temiz ve anlamlı bir veri temsili elde edebilirsiniz.

1.2.4. PCA'nın Dezavantajları

1. **Bilgi Kaybı:** PCA, boyut azaltma amacıyla kullanıldığında, veri setinin orijinal yapısında bazı bilgi kaybına neden olabilir. Yüksek varyansı taşıyan bileşenler ana yapıyı korurken, düşük varyanslı bileşenler bazen önemli ayrıntıları kaybetmenize yol açabilir.
2. **Anlam Karmaşıklığı:** Temel bileşenler, orijinal değişkenlerin kombinasyonları olarak ifade edildiği için, bu bileşenlerin gerçek dünyadaki anlamlarını yorumlamak bazen zor olabilir. Bu, analistlerin sonuçları anlamak için daha fazla çaba harcamasına neden olabilir.
3. **Veri Dönüşümü:** PCA, veri setini yeni bir koordinat sistemine dönüştürdüğü için, elde edilen temel bileşenlerin fiziksel anlamı olmayabilir. Bu, bazı durumlarda yorumlamanın zorlaşmasına yol açabilir.
4. **Veri Dağılımı Sorunları:** PCA, veri setindeki değişkenlerin dağılımlarını dikkate almaz. Bu nedenle, veri setlerindeki dağılım bozukluğu veya aşırı aykırı değerler sonuçları etkileyebilir.
5. **Ölçeklendirme Etkisi:** PCA, değişkenler arasındaki ölçek farklılıklarına duyarlı olabilir. Bu nedenle, veri setinde ölçek uyumsuzluğu varsa sonuçlar etkilenebilir.

BÖLÜM 2. SİSTEMATİK YAKLAŞIM

Sistematiik yaklaşım, karmaşık sorunları daha anlaşılır ve düzenli adımlarla çözmeyi amaçlayan bir yöntemdir. Özellikle analiz, tasarım veya problem çözme süreçlerinde kullanılan bu yaklaşım, adımların mantıklı bir sıra içinde ilerlemesini sağlayarak sonuca daha etkili bir şekilde ulaşmayı hedefler. Sistematiik yaklaşım, sürecin her aşamasının belirlenmiş bir plana dayanmasını sağlar, böylece her adım önceki ve sonraki adımlarla uyumlu bir şekilde ilerler. Bu yaklaşım, karmaşıklığı yönetmek, analiz süreçlerini organize etmek ve sonuçları daha anlaşılır bir şekilde sunmak isteyen birçok alanda başarıyla kullanılmaktadır. Bu yazıda, PCA analizini daha sistematiik bir perspektifle ele alacak ve adım adım nasıl hesaplanacağını inceleyeceğiz.

2.1. Formülasyon

2.1.1. Veri Hazırlığı:

Başlangıç olarak, "m" örnek sayısı ve "n" değişken sayısı olan veri matrisini XX şeklinde temsil ediyoruz. Veriyi standartlaştırarak ölçeklendirebiliriz:

$$X_{ij} = \frac{X_{ij} - \mu_j}{\sigma_j} \quad (2.1)$$

Burada X_{ij} veri matrisinin i-inci örneğinin j-inci değişkeni, μ_j değişkenin ortalama değeri ve σ_j standart sapmasıdır.

2.1.2. Kovaryans Matrisi Hesaplaması:

Kovaryans matrisini C hesaplamak için veri matrisinin transpozunu kullanarak aşağıdaki formülü kullanabiliriz:

$$C = \frac{1}{m - 1} X^T X \quad (2.2)$$

2.1.3. Özdeğer ve Özvektör Analizi:

Özdeğerler ve özvektörler, kovaryans matrisinin özdeğer ve özvektör analizi ile elde edilir:

$$Cv = \lambda v \quad (2.3)$$

Bu denklemde, v özvektörü, λ özdeğeri temsil eder. Bu denklemi çözerek özdeğerleri ve özvektörleri elde ederiz.

2.1.4. Özdeğerleri Sıralama ve Temel Bileşen Seçimi

Özdeğerleri büyükten küçüğe sıralayarak, en büyük özdeğerlere karşılık gelen özvektörleri temel bileşenler olarak seçeriz. Bu, en fazla varyansı yakalayarak boyut azaltmayı amaçlar.

2.1.5. Yeri Veri Seti Oluşturma

Seçtiğimiz özvektörleri kullanarak yeni bir veri matrisi Y oluştururuz:

$$Y = XV \quad (2.4)$$

Burada V seçilen özvektörleri içeren matristir.

Bu adımlar, PCA analizini matematiksel formüllerle adım adım açıklar. Bu formüller, PCA'nın temel prensiplerini ve veri setinin boyutunu nasıl azaltacağını anlamamıza yardımcı olur.

2.2. Yazılım Mimarisi

2.2.1. Katmanlı Mimari

PCA analizinin .NET Core MVC uygulaması için katmanlı bir mimari kullanmak önemlidir. Bu, kodun modüler, sürdürülebilir ve genişletilebilir olmasını sağlar.

Tipik olarak üç katmanlı bir yapı kullanılır:

- **Sunum Katmanı (Presentation Layer):** Bu katman kullanıcı arayüzü ile ilgilenir. MVC yapısının Controller kısmını temsil eder. Kullanıcıdan gelen istekleri işler, veri işleme katmanıyla iletişim kurar ve sonuçları görüntüler.
- **İş Mantığı Katmanı (Business Logic Layer):** Bu katman, PCA analizi ile ilgili iş mantığını içerir. Veri analizi, boyut azaltma ve PCA algoritmalarının

uygulanması gibi işlemleri gerçekleştirir. Sunum katmanıyla iletişim halinde olup gerektiğinde veritabanı katmanıyla etkileşime geçer.

- Veri Erişim Katmanı (Data Access Layer): Bu katman, veri tabanı veya veri kaynaklarıyla iletişimi sağlar. PCA analizi için gereken verileri alır veya verileri kaydeder. Veritabanı bağlantıları, sorgular ve veri işleme bu katmanda yapılır.

2.2.2. Kullanılan Teknolojiler

- .NET Core MVC: Kullanıcı arayüzünü oluşturmak, istekleri işlemek ve sonuçları sunmak için kullanılır.
- Accord.NET: PCA analizi algoritmalarını içeren bir kütüphanedir. Bu kütüphane, veri analizi için gerekli olan matematiksel işlemleri kolaylaştırır.
- Veritabanı (Opsiyonel): Eğer PCA analizi için veri tabanı kullanılıyorsa, Entity Framework gibi bir ORM (Object-Relational Mapping) aracı kullanılabilir.

2.2.3. Adım Adım İş Akışı

- Kullanıcı, web tarayıcısı üzerinden MVC uygulamasına istek gönderir.
- İstek, Controller tarafından alınır ve gerekli veriler İş Mantığı Katmanına iletilir.
- İş Mantığı Katmanı, Accord.NET kütüphanesini kullanarak PCA analizi gerçekleştirir.
- Analiz sonuçları, Controller tarafından alınır ve kullanıcı arayüzüne gönderilir.
- Kullanıcı arayüzü, sonuçları kullanıcıya görüntüler.

2.2.4. Örnek Senaryo

Örneğin, bir e-ticaret platformunda ürünlerin özelliklerini (fiyat, boyut, ağırlık vb.) içeren bir veri seti üzerinde PCA analizi yapmak istediğinizi düşünelim. Bu analizle benzer ürün gruplarını belirlemek veya ürünleri boyutlarına göre kategorilere ayırmak hedefleniyor olabilir.

- Sunum Katmanı: Kullanıcının ürün özelliklerini seçebildiği bir arayüz sunar.
- İş Mantığı Katmanı: Seçilen ürün özellikleri veri seti olarak alınır. Accord.NET kullanılarak PCA analizi gerçekleştirilir ve sonuçlar elde edilir.
- Veri Erişim Katmanı: Gerekirse, ürün verileri veritabanından çekilir veya kaydedilir.

Bu mimari, .NET Core MVC ve Accord.NET kullanarak PCA analizi yapmanın temel adımlarını içerir. Katmanlı yapı, kodun düzenli ve sürdürülebilir olmasını sağlar. Accord.NET ise matematiksel işlemleri kolaylaştırarak analiz işlemini hızlandırır.

BÖLÜM 3. .NET Core MVC ile PCA Analizini Gerçekleştirme Süreci ve Veri Boyutunu Azaltma

Günümüzün hızla gelişen dijital dünyasında, veri analizi ve boyut azaltma gibi veri madenciliği teknikleri giderek daha fazla önem kazanmaktadır. Bu bağlamda, .NET Core MVC ile gerçekleştirilen PCA analizi, deney düzeneği ve sanal laboratuvar aracılığıyla, veri bilimine ilgi duyan herkesin pratik deneyim kazanmasını sağlayan önemli bir öğrenme aracıdır.

3.1. Deney Düzeneği

Bu deney düzeneği, veri madenciliği alanında boyut azaltma tekniği olan PCA analizini .NET Core MVC kullanarak nasıl gerçekleştireceğinizi anlamak ve öğrenmek amacıyla tasarlanmıştır. Deney düzeneği, aşağıdaki bileşenleri içerir:

1. **Veri Hazırlığı:** Deney düzeneği, önceden belirlenmiş bir veri setini temsil eder. Bu veri seti, örnek olarak e-ticaret ürünlerinin fiyat, boyut, ağırlık gibi özelliklerini içerebilir.
2. **.NET Core MVC:** Deney düzeneği, .NET Core MVC çerçevesi kullanılarak bir web uygulaması olarak tasarlanır. Bu uygulama, kullanıcıların veriyi yüklemesine, analiz parametrelerini belirlemesine ve sonuçları görüntülemesine olanak tanır.
3. **Veri Analizi ve Boyut Azaltma:** MVC uygulaması, kullanıcının yüklediği veriyi alır ve Accord.NET gibi kütüphaneleri kullanarak PCA analizini gerçekleştirir. Bu adım, veri boyutunu azaltarak verinin özünü yakalamayı amaçlar.
4. **Sonuç Sunumu:** Analiz sonuçları, MVC uygulaması tarafından kullanıcıya sunulur. Bu sonuçlar, bileşenler arası ilişkileri veya verinin farklı boyutlarda nasıl temsil edildiğini gösteren grafikler, tablolar veya görseller olabilir.
5. **Öğrenme ve Deneyim:** Deney düzeneği, kullanıcıların PCA analizini pratiğe dökme sürecini anlamalarını ve öğrenmelerini desteklemek üzere tasarlanmıştır. Kullanıcılar, gerçek veri setleri üzerinde denemeler yaparak ve sonuçları görselleştirerek bu teknikle ilgili deneyim kazanırlar.

Tablo 3.1. E-ticaret ürünlerinin fiyat, boyut ve ağırlıklarını gösteren tablo

Ürün ID	Fiyat (\$)	Boyut (cm)	Ağırlık (kg)
1	50	10	2
2	30	8	1
3	80	15	3
4	120	25	4
5	70	12	2.5
6	90	18	3
7	55	11	2.2
8	40	9	1.5
9	65	14	2.7
10	75	13	2.3
11	85	16	3.2
12	110	22	3.8
13	95	17	3.1
14	60	10.5	2.1
15	105	20	3.5

3.2. Sanal Laboratuvar Karakteristiği

1. Pratik Uygulama: Kullanıcılar, gerçek veri setleri üzerinde PCA analizi yaparak, boyut azaltma ve veri madenciliği konularını pratik olarak anlayabilirler.
2. Güvenli Ortam: Sanal laboratuvar, deneylerin güvenli bir dijital ortamda yapılmasını sağlar. Yanlış yapılan deneyler gerçek dünyada potansiyel riskler taşımadan deneyimlenebilir.
3. Öğrenme Fırsatları: Kullanıcılar, PCA analizi sürecini adım adım deneyimleyerek, veri ön işleme, analiz sonuçlarının yorumlanması gibi becerileri geliştirirler.
4. Görsel Temsil: Sanal laboratuvar, analiz sonuçlarını grafikler, görseller ve tablolar aracılığıyla kullanıcıya sunarak, verinin boyut azaltma sonuçlarını daha iyi anlamalarını sağlar.

5. Hatalı Senaryolar: Kullanıcılar, farklı parametrelerle deneyler yaparak hatalı senaryoları deneyimleyebilir ve analizin sonuçları üzerindeki etkilerini anlayabilirler.
6. Özelleştirme ve Tekrar: Kullanıcılar, farklı veri setleri ve analiz parametreleriyle deneyler yaparak sonuçların nasıl değişebileceğini gözlemleyebilirler.

3.3. Test Aşaması

PCA analizi gerçekleştirilerek, farklı özelliklerin boyut azaltma ve veri madenciliği üzerindeki etkileri test edilecektir. Analiz sonuçları, veri setinin özünü yakalamada ne kadar etkili olduğunu ve farklı özellikler arasındaki ilişkileri göstermeye yardımcı olacaktır.

Tablo 3.2. Öğrencilerin fiziksel özellikleri, akademik başarıları ve fiziksel aktivite alışkanlıkları gibi değişkenleri içeren tablo

Kişi ID	Yaş	Boy (cm)	Kilo (kg)	Matematik Puanı	Fiziksel Aktivite Saati
1	25	180	75	85	3
2	32	165	60	70	1
3	45	175	80	60	2
4	22	160	50	95	4
5	28	172	70	78	1
6	38	185	90	65	2
7	19	155	45	92	5
8	51	170	75	58	1
9	29	175	65	72	3
10	34	182	80	80	2
11	42	163	55	70	1
12	26	168	68	88	4
13	31	178	73	77	2
14	20	160	48	95	5
15	48	175	82	60	1

MVC projesi içerisinde, kullanıcıların veriyi yükleyebileceği bir alana veri seti yüklenir. Kullanıcının yüklediği veri, Controller tarafında işlenecektir. Veri işleme aşamasında Accord.NET kütüphanesi kullanılarak PCA analizi yapılacaktır.

Şekil 3.1. Veri setinin yükleneceği web sayfası(.csv file)

```
[HttpPost]
public IActionResult PerformPCA(IFormFile csvFile)
{
    using (var streamReader = new StreamReader(csvFile.OpenReadStream()))
    {
        List<double[]> records = new List<double[]>();
        List<string> columnNames = new List<string>(); // Sütun isimlerini tutacak liste
        int columnCount = -1;

        while (!streamReader.EndOfStream)
        {
            var line = streamReader.ReadLine();
            var values = line.Split(',');

            if (columnCount == -1)
            {
                columnCount = values.Length;
                columnNames.AddRange(values); // İlk satırdaki değerleri sütun isimleri olarak ekliyoruz
            }
            else if (values.Length != columnCount)
            {
                |
            }

            var record = new double[columnCount];

            for (int i = 0; i < columnCount; i++)
            {
                if (double.TryParse(values[i], NumberStyles.Any, CultureInfo.InvariantCulture, out double value))
                {
                    record[i] = value;
                }
                else
                {
                    |
                }
            }

            records.Add(record);
        }
    }
}
```

Şekil 3.2. Veri setinin yüklendikten sonra verileri PCA analizi için uygun hale getiren kod bölümü

```

double[][] rawData = records.ToArray();

PrincipalComponentAnalysis pca = new PrincipalComponentAnalysis();
pca.Learn(rawData);

double[][] pcaResult = pca.Transform(rawData);

int numComponents = pca.Components.Count();
int numInstances = pcaResult.GetLength(0);
double[][] reducedData = new double[numInstances][];

for (int i = 0; i < numInstances; i++)
{
    reducedData[i] = pcaResult[i].Take(numComponents).ToArray();
}

ViewBag.ReducedData = reducedData;
ViewBag.ColumnNames = columnNames; // Sütun isimlerini ViewBag'e ekliyoruz
ViewBag.records = records;
return View("PcaResult");
}

```

Şekil 3.3. PCA analizinin Accord.NET kütüphanesi kullanarak yapılmasını içeren kod bölümü

PCA Result

Kişi ID	Yaş	Boy (cm)	Kilo (kg)	Matematik Puanı	Fiziksel Aktivite Saati
1	25	180	75	85	3
2	32	165	60	70	1
3	45	175	80	60	2
4	22	160	50	95	4
5	28	172	70	78	1
6	38	185	90	65	2
7	19	155	45	92	5
8	51	170	75	58	1
9	29	175	65	72	3
10	34	182	80	80	2
11	42	163	55	70	1
12	26	168	68	88	4
13	31	178	73	77	2
14	20	160	48	95	5
15	48	175	82	60	1

Şekil 3.4. PCA analizi sonuç ekranı : Kırmızı renkler indirgenmesi gereken verileri göstermektedir.

BÖLÜM 4. VERİ GÜVENLİĞİ DEĞERLENDİRMESİ

Veri güvenliği, modern yazılım projeleri için olmazsa olmaz bir gerekliliktir, özellikle kişisel verilerin işlendiği ve depolandığı projelerde. PCA analizi yapmak üzere .NET Core MVC kullanıldığında, projenin veri güvenliği kritik bir rol oynar. Veri güvenliği değerlendirmesi, projenin baştan sona nasıl tasarlandığını, verilerin nasıl işlendiğini ve saklandığını detaylı bir şekilde ele alır.

4.1. Veri İşleme ve Depolama Güvenliği

Projeye yüklenen veri setleri, güvenli bir şekilde kabul edilmeli ve işlenmelidir. Dosya yüklemesi sırasında dosya türü, boyutu ve içeriği dikkatlice kontrol edilmelidir. Dosyaların doğrulanması ve temizlenmesi, zararlı içeriklerin sisteme sızmasını önler. Ayrıca, veri tabanı veya depolama mekanizması seçimi, veri şifrelemesi, günlük kayıtları ve izleme gibi önlemlerle desteklenmelidir.

4.2. Veri Anonimleştirme ve Maskelendirme

Projede kullanılan veri seti, özel veya hassas veriler içeriyorsa, anonimleştirme veya maskelendirme yöntemleri kullanılmalıdır. Anonimleştirme, veriyi gerçek kullanıcılarla ilişkilendirmeyi zorlaştırırken, maskelendirme belirli verilerin gizlenmesini sağlar. Bu yöntemler, kullanıcı gizliliğini koruma konusunda önemlidir.

BÖLÜM 5. SONUÇLAR VE ÖNERİLER

Sonuç olarak, bu proje, .NET Core MVC ile PCA analizini gerçekleştirmenin mümkün olduğunu ve veri güvenliği odaklı bir yaklaşımla başarılı bir şekilde uygulandığını göstermiştir. Projenin öneriler doğrultusunda geliştirilmesi, daha geniş veri setlerinde ve gerçek dünya uygulamalarında kullanılabilir hale getirilmesini sağlayabilir. Bu projenin, veri analizi ve güvenliği konularında ilgili bireyler ve topluluklar için önemli bir kaynak olacağına inanıyoruz.

KAYNAKLAR

- [1] Accord.NET Resmi Dökümantasyonu: <https://accord-framework.net/docs/Index.html>
- [2] <https://builtin.com/data-science/step-step-explanation-principal-component-analysis> PCA Analizi ve Uygulamaları
- [3] <http://www.zafercomert.com/IcerikDetay.aspx?zcms=78> Temel Bileşenler Analizi
- [4] <https://www.veribilimiokulu.com/makine-ogrenmesine-cok-degiskenli-istatistiksel-yaklasimlar-temel-bilesenler-analizi/> Temel Bileşenler Analizi

ÖZGEÇMİŞ

Eray ALÇİN, 10 Haziran 1997 tarihinde Bolu'nun Kıbrıscık ilçesinde doğdu. İlk ve orta eğitimini Pendik'te tamamladı. Lise eğitimini Tuzla Teknik ve Endüstri Meslek Lisesi'nde Bilişim Teknolojileri bölümünün web tasarım alanında aldı. 2015 yılında liseden mezun oldu. 2016 yılında Sakarya Üniversitesi Bilgisayar Programcılığı Bölümü'nü kazandı. 2018 yılında Dikey Geçiş Sınavı'na girerek Gebze Teknik Üniversitesi Bilgisayar Mühendisliği bölümünü kazandı. Hazırlık dahil 4 yıl boyunca okuduktan sonra Sakarya Üniversitesi Bilgisayar Mühendisliği Bölümü'ne yatay geçiş yaptı ve şu an son sınıf öğrencisi olarak eğitimine devam etmektedir.

BSM 401 BİLGİSAYAR MÜHENDİSLİĞİ TASARIMI DEĞERLENDİRME VE SÖZLÜ SINAV TUTANAĞI

KONU : PCA Analizinin .Net Core MVC ile Gerçekleştirilmesi

ÖĞRENCİLER (Öğrenci No/AD/SOYAD): B221210376 / Eray / ALÇİN

Değerlendirme Konusu	İstenenler	Not Aralığı	Not
Yazılı Çalışma			
Çalışma klavuza uygun olarak hazırlanmış mı?	x	0-5	
Teknik Yönden			
Problemin tanımı yapılmış mı?	x	0-5	
Geliştirilecek yazılımın/donanımın mimarisini içeren blok şeması (yazılımlar için veri akış şeması (dfd) da olabilir) çizilerek açıklanmış mı?			
Blok şemadaki birimler arasındaki bilgi akışına ait model/gösterim var mı?			
Yazılımın gereksinim listesi oluşturulmuş mu?			
Kullanılan/kullanılması düşünülen araçlar/teknolojiler anlatılmış mı?			
Donanımların programlanması/konfigürasyonu için yazılım gereksinimleri belirtilmiş mi?			
UML ile modelleme yapılmış mı?			
Veritabanları kullanılmış ise kavramsal model çıkarılmış mı? (Varlık ilişki modeli, noSQL kavramsal modelleri v.b.)			
Projeye yönelik iş-zaman çizelgesi çıkarılarak maliyet analizi yapılmış mı?			
Donanım bileşenlerinin maliyet analizi (prototip-adetli seri üretim vb.) çıkarılmış mı?			
Donanım için gerekli enerji analizi (minimum-uyku-aktif-maksimum) yapılmış mı?			
Grup çalışmalarında grup üyelerinin görev tanımları verilmiş mi (iş-zaman çizelgesinde belirtilebilir)?			
Sürüm denetim sistemi (Version Control System; Git, Subversion v.s.) kullanılmış mı?			
Sistemin genel testi için uygulanan metotlar ve iyileştirme süreçlerinin dökümü verilmiş mi?			
Yazılımın sızma testi yapılmış mı?			
Performans testi yapılmış mı?			
Tasarımın uygulamasında ortaya çıkan uyumsuzluklar ve aksaklıklar belirtilerek çözüm yöntemleri tartışılmış mı?			
Yapılan işlerin zorluk derecesi?	x	0-25	
Sözlü Sınav			
Yapılan sunum başarılı mı?	x	0-5	
Soruları yanıtlama yetkinliği?	x	0-20	
Devam Durumu			
Öğrenci dönem içerisindeki raporlarını düzenli olarak hazırladı mı?	x	0-5	
Diğer Maddeler			
Toplam			

DANIŞMAN:

DANIŞMAN İMZASI: