T.C.
İSTANBUL ÜNİVERSİTESİ
MÜHENDİSLİK FAKÜLTESİ

BİTİRME PROJESİ

DIMENSIONALITY REDUCTION TECHNIQUES

ELEKTRİK ELEKTRONİK MÜHENDİSLİĞİ

ALİ ÖZTAŞ
1357130053

DANIŞMAN
Yrd. Doç. Dr.Mahmut Öztürk

Haziran, 2017

İSTANBUL

ÖNSÖZ

Bu ödevde Dimensionality Reduction Techniques (Boyutluluk Azaltma Teknikleri) kullanım alanları ve örnekleri anlatılacaktır.

Bu çalışmada yardımını esirgemeyen hocam Yrd.Doç. Dr. Mahmut Öztürk'e teşekkür ederim.

Haziran 2017 Ali Öztaş

İÇİNDEKİLER

Sayfa No

1. GİRİŞ

Özellik uzayının boyutluluğunu azaltmak veri madenciliği alanında uzun süredir kullanılmaktadır. Örnek olarak PCA tekniği [1] çok yüksek boyutluluğa sahip problemler için bir pre process basamağı olarak kullanılmaktadır. Linear Discriminant Analysis(LDA) tekniği de PCA ile ortak özellikleri taşır, boyutluluğu bire düşürmeye çalışır. PCA için önemsiz olmayan komponent sayısını bulmak için gereken en uygun boyut sayısı birçok kez araştırılmıştır.[9]

Ödevimde bu tekniklerin sınıflandırma gibi başka kullanım alanlarının da olduğu ve bu alandaki kullanım nedenleri ve etkileri araştırılmıştır. Programlar python programlama dili kullanılarak yazılmıştır.

2. GENEL KISIMLAR

Boyutluluk azaltma tekniklerini kaba olarak aşağıdaki gibi sınıflandırabiliriz:

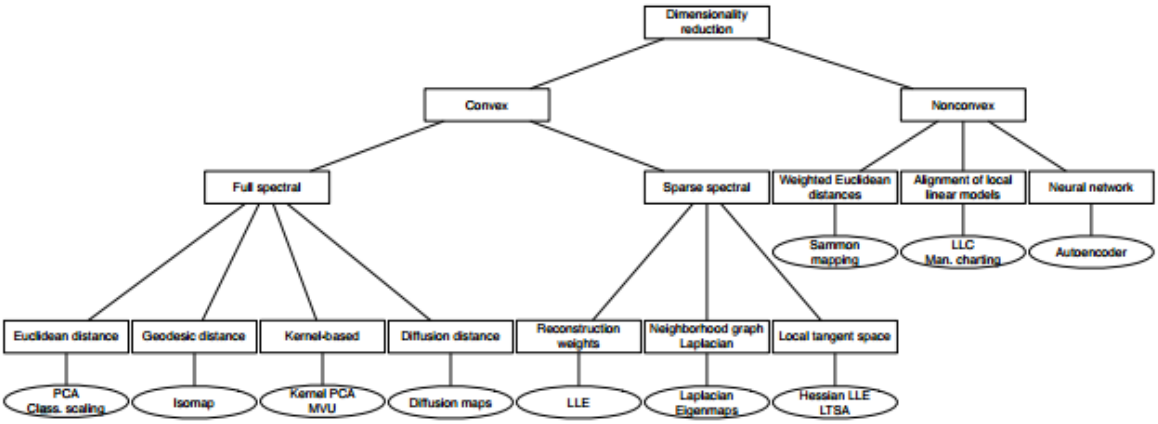
- *Zor boyutluluk azaltma problemleri,*bu problemlerde boyutluluk aralığı çok fazladır.(100-100.000)Daha çok yasaklayıcı küçültmeler kullanılır.Bileşenler genellikle bir büyüklüğün uzayın farklı noktalarında veya farklı zamanlarda ölçülen değerlerinden oluşur.Bu sınıfta içinde resimler de içeren desen tanıma ve sınıflama problemleri bulunur.Konuşma tanıma da bu sınıfın içindedir.
- *Kolay boyutluluk azaltma problemleri,*bu problemlerde data çok yüksek boyutlu değildir ve küçültme çok kısıtlayıcı değildir.Bileşenler genelde farklı değişkenlerin gözlemlenen veya ölçülen değerleridir ve incelenmesi rahattır.Sosyal bilimler,psikoloji gibi çoğu istatistiksel analizler bu sınıfın içindedir.
- *Görüntüleme problemleri,*bu problemlerde data çok yüksek boyuta sahip değildir fakat görüntülemek için 2 veya 3'e indirmek gereklidir.5 boyuta kadar olan datasetleri göstermek için görüntüleme teknikleri vardır,renkler,yönler,kabartmalar,stereografi veya başka aletler kullanılır fakat basit bir grafiğin çekiciliğine sahip değillerdir.Chernoff yüzleri (Chernoff,1973) daha da fazla boyuta izin verir fakat datayı incelemek ve görüntülemek oldukça zordur.

Zaman değişkeni de işin içine katılırsa iki kategori daha eklenebilir: *static boyutluluk azaltma* ve *zamana bağlı boyutluluk azaltma*.Vektör zaman serilerinde kullanışlıdırlar örnek olarak videolardan kesitler ve devamlı konuşmalar verilebilir.

Boyutluluk azaltma yöntemleri şöyle tanıtılabilir,$n \times D$ bir matris $\mathbf{X}$ ile temsil edilen bir dataset olduğu varsayılırsa,bu matrisin n adet D boyutluluğa sahip data vetörü $\mathbf{x_i}$ vardır.Devam etmek için bu datasetin d içsel boyutluluğunun olduğunu varsayılır. ($d < D$ hatta çoğunlukla $d << D$).Burada, matematik terimleriyle içsel boyutluluk, $\mathbf{X}$ datasetindeki noktalar D-boyutlu uzayın içine gömülmüş d boyutluluğuna sahip manifoldun yakınında ya da üstünde bulunmaktadır.Bu manifoldun yapısıyla ilgili hiçbir tahminde bulunulmamaktadır.Boyutluluk azaltma teknikleri,D boyutluluğuna sahip $\mathbf{X}$ datasetini d

boyutluluğa sahip yeni bir dataset olan **Y** datasetine dönüştürür.Bu işlem datanın geometrisini koruyarak yapılmaya çalışılır.Genel olarak **X** datasetinin,manifoldunun geometrisi veya içsel boyutluluğu *d* ile ilgili bir bilgi bilinmemektedir.Boyutluluk azaltma problemleri anca belli özelliklerin varsayımı ile çözülebilir(içsel boyutluluk gibi).

Boyutluluk azaltmak için çok sayıda değişken kullanılabilir örnek olarak noktalar arası uzaklıklara bakılabilir fakat 'Geodesic' uzaklığa bakılan teknik ile Öklid uzaklığının kullanıldığı teknik farklıdır.Aşağıdaki şekilde (Şekil-1) bu sınıflandırma verilmiştir.Boyutluluk azaltma,dış-bükey ve dış-bükey olmayan teknikler olarak ikiye bölünmüştür bu ikisi arasındaki ana fark dış-bükey yöntemler datayı bütün olarak incelerken dış-bükey olmayanlar ise lokal olarak inceleyip lokal olarak bir amaç fonksiyonu kullanır.



Şekil-1: Boyutluluk Azaltma Tekniklerinin Taksonomisi

Full Spectral tekniklerde tam matrise 'eigendecomposition' uygulanır ve boyutlar arası kovaryanslar ile birlikte data noktaları arası benzerlikler bulunur.Sparse Spectral tekniklerde ise datanın lokal yapısı korunmaya çalışır ve genelleşmiş bir 'eigenproblem' çözülmeye uğraşılır.

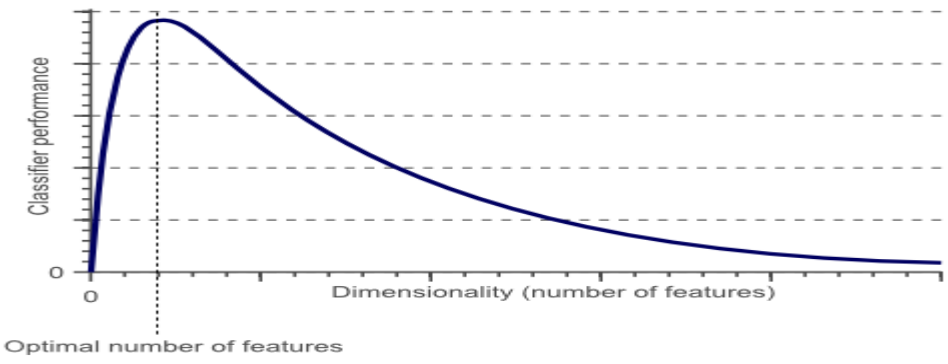
2.1 CURSE OF DIMENSIONALITY

Elde birer kedi veya köpeğe ait olan fotoğraflar bulunsun.Köpekleri kedilerden otomatik olarak ayıracak bir sınıflandırıcı yaratmak istenebilir.Fakat bunun uygulanması için ayırıcı bir faktör gerekir ve bu faktörün matematik algoritmaları ile gösterilebilmesi gerekir.Çok basit bir örnek olarak kedilerle köpeklerin renkleri farklıdır denebilir.Kedi ve köpek sınıflarını ayıracak bir tanımlayıcı olarak üç tane değişken kullanılabilir; kırmızı (kırm) renginin ortalaması, yeşil (yes) renginin ortalaması ve mavi (mv) renginin ortalaması olarak.Örnek olarak basit bir lineer sınıflandırıcı bu üç değişkeni lineer olarak birleştirip sınıfa karar verebilir:

*If 0.4*red + 0.5*green + 0.3*blue > 0.6:return cat; else return dog;*

Yukarıdaki kısım basit bir C# veya Java kodu olarak yazılabilir.

Bu üç değişken çok iyi bir sınıflandırma sağlamaz bundan dolayı fotoğrafların yapısını belirtecek birkaç değişken daha tanımlansın.Keskin kenarların ortalaması ve yönelme yoğunlukları(gradient density) X ve Y yönlerinde olmak üzere tanımlansın. Elde 5 tane değişken bulunmaktadır.Bu 5 değişkenle de sınıflandırma yapılabilir.Ve hatta daha da mükemmel olması için daha da değişken eklenebilir.Ama sınıflandırıcının performansı belli bir tanımlayıcı değişken sayısından sonra düşmeye başlayacaktır.Aşağıdaki şekilde (Şekil-1) bu olay gösterilmektedir ve bu olaya genellikle Boyutluluğun Laneti denir.



Şekil 1 : Boyutluluk arttıkca sınıflandırıcının performansı optimal bir tanımlayıcıya kadar artıyor.Boyutluluğu arttırdıkça eğitme örneklerinin sayısının yeterince artmaması sınıflandırıcının performansını azaltır.

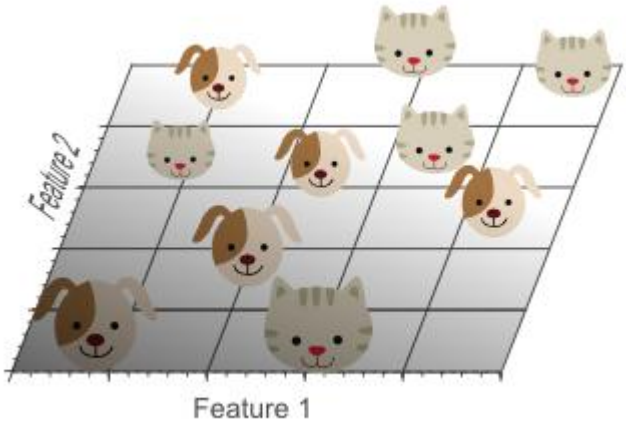
2.2 CURSE OF DIMENSIONALITY AND OVERFITTING

Önceki sayfada kedi ve köpek örneği kullanılmıştı, dünyada çok sayıda kedi ve köpek vardır.Fakat kısıtlı zaman ve işlem gücünden dolayı sadece elde bulunan 10 fotoğraf ile işlem yapılması gerektiğini varsayılsın.Amaç bu on fotoğrafı öğrenip sonra istenen kadar fazla sayıda köpek ve kediyi ayırt edebilecek ve üstüne kedi ve köpekle hiçbir bilgi gerektirmeyen bir sınıflandırıcı yapılmasıdır.

İlk olarak basit bir lineer sınıflandırıcı kullanılsın ve kusursuz sınıflandırma elde edilmeye çalışılsın.Başlangıçta sadece tek özellik kullanılırsa en başta denilen fotoğraftaki yeşil renginin ortalaması olursa:

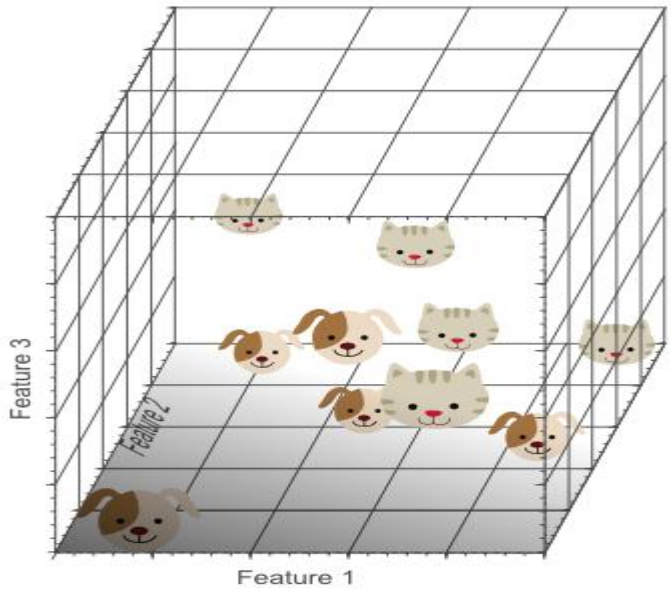
Şekil 2: Tek özellik iyi sonuç vermedi.

Daha iyi bir sonuç için ikinci bir özellik eklenirse ek olarak kırmızı renginin ortalaması da eklenirse:

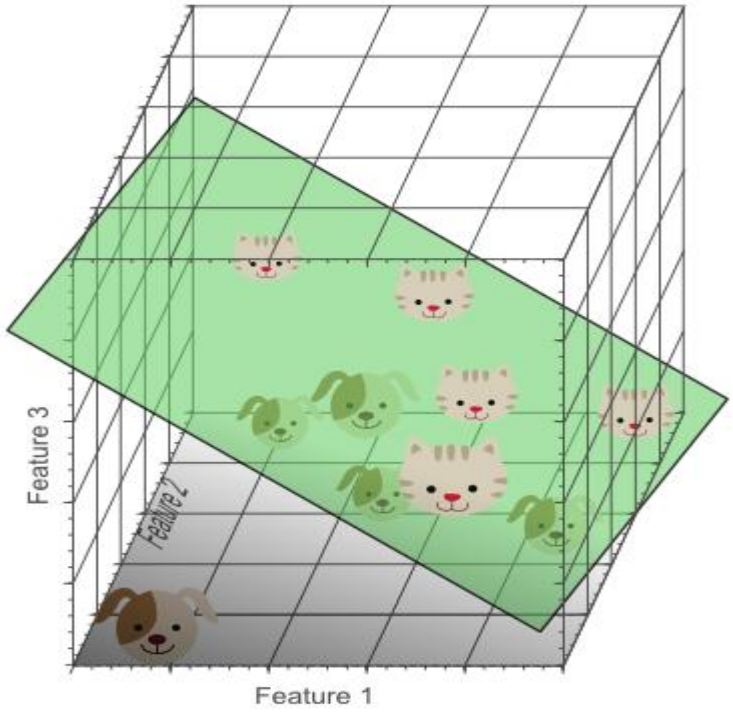


Şekil 3: İkinci özellik eklense de kedilerle köpekleri ayıran tek bir çizgi yok hala.

Son olarak mavi renginin ortalaması üçüncü özellik olarak eklenirse:



Şekil 4: Üçüncü bir özellik eklenince görüldüğü üzere kedilerle köpekleri ayıran bir düzlem bulunmaktadır.

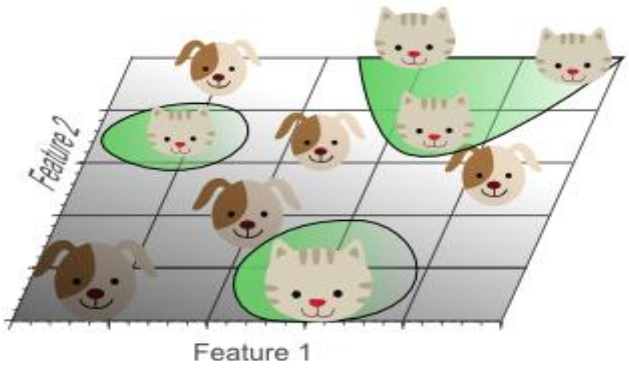


Şekil 5: Daha çok özellik kullandıkça sınıfları ayırabilme şansı artmaktadır.

Yukarıdaki şekiller özellik sayısının arttırılmasının kusursuz sınıflandırıcı bulunana kadar özellik sayısının arttırılmasını gösteriyor fakat Şekil-1'de böyle olmadığı gösterilmişti.Ve boyutluluk arttıkça örnek yoğunluğunun üstel olarak azaldığı şekillerde görülmektedir.

Şekil-2'de 10 eğitim örneği,genişliği 5 birim aralıklı olan özellik uzayının tamamını kaplamıştı.Örnek yoğunluğu bundan dolayı 2 örnek/aralık olarak gösterilebilir. Şekil-3'de yine 10 adet örnek vardı fakat bu sefer 2 boyutlu bir özellik uzayı olduğundan dolayı alan 25 birim karedir ve örnek yoğunluğu ise (10/25) 0.4 örnek/aralık olarak bulunur. Son olarak 3 boyutlu durumda 125 birim küplük bir alanda 10 adet örnek vardı.Yani 3 boyutlu durumda örnek yoğunluğu (10/125) 0.08 örnek/aralık olarak bulunur.

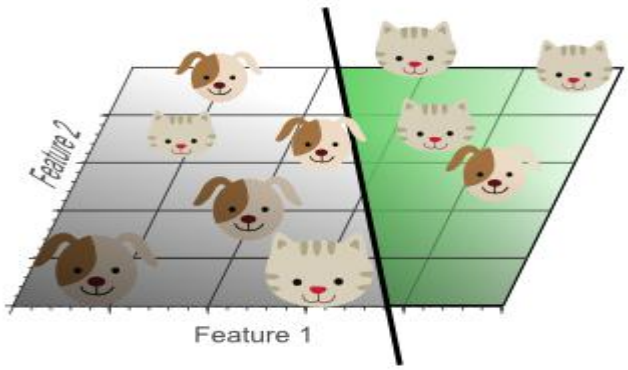
Özellik eklenmeye devam ederse, özellik uzayının boyutluluğu artmaya devam edecektir ve gittikçe aralık artacaktır. Bu aralıklılığın artışından, ayrılabilir bir hyperplane bulunması çok daha kolaylaşır bunun nedeni ise bir eğitim örneğinin en iyi hyperplane'in yanlış tarafında kalması ihtimali özellik uzayı büyüdükçe çok küçük olur. Yüksek boyutlu bir sınıflandırma düşük boyutta yeniden çizilirse buna bağlı bir hata ortaya çıkar:



Şekil 6: Çok fazla özellik kullanmak taşmaya sebep olur.Sınıflandırıcı eğitim örneğine özgü istisnalar öğrenmeye başlar ve yeni data gelince düzgün genelleştirilemez.

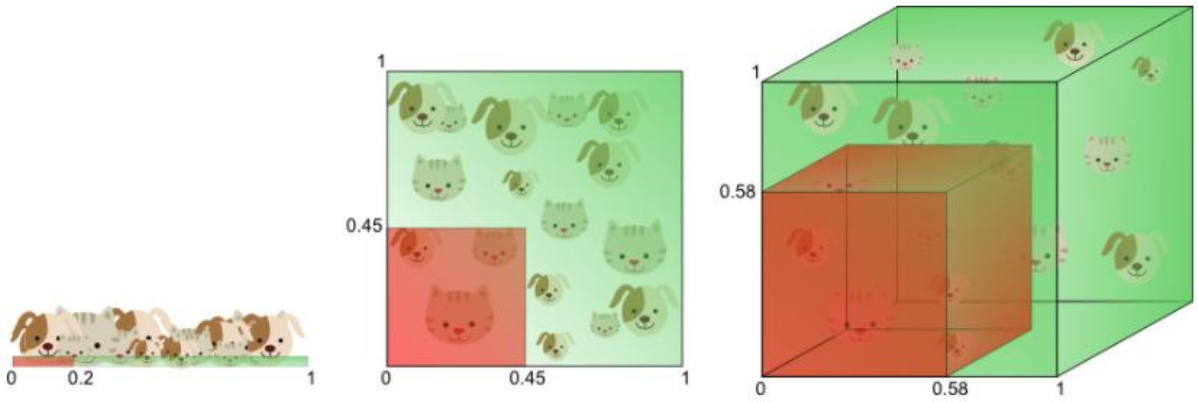
Şekil-6'da 3 boyutlu sınıflandırma sonucunun,2 boyuta yansıtılmasıdır.3 boyutta lineer olarak ayrılabilen data daha düşük boyutlu sahip bir uzayda lineer olarak ayrılamayabilir.Başka bir deyişle üçüncü bir boyut eklemek,daha düşük boyutlu bir özellik uzayında non-lineer sınıflandırıcı kullanmaya karşılık gelmektedir.Sonuç olarak sınıflandırıcı eğitim datasetine özgü belli özellikler ve istisnaları öğrenir. Bundan dolayı ortaya çıkan sınıflandırıcı gerçek dünya datasında sonsuz sayıda görülmemiş ve kedi olmasıyla birlikte aynı zamanda bu

istisnaları bilmediğinden dolayı hata verebilir. Bu kavrama 'overfitting' denir ve boyutluluğun lanetinin sonucudur.Şekil 7'de iki özellik kullanılarak eğitilen bir lineer sınıflandırıcının sonucunu gösterilmektedir.



Şekil-7: Data kusursuz olarak sınıflandırılmasa da bu sınıflandırıcı görülmemiş datada Şekil-5'deki sınıflandırıcıdan daha iyi sonuç vermektedir.

Şekil 8 farklı bir konuyu ele almaktadır. Değerleri 0 ile 1 arasında değişen tek özellik kullanarak bir sınıflandırıcı eğitilmek istendiği varsayılsın.Bu özelliğin kedi ve köpeklerde ayrı olduğu ele alınsın.Eğitim datasının bu aralığın %20'sini kapsaması istenirse gereken eğitim datası tüm köpek ve kedi nüfusunun %20'si olacaktır.Farklı bir eğitim özellik daha eklenirse sonuç olarak iki boyutlu bir özellik uzayı ortaya çıkar; 2 boyutlu bir özellik uzayının %20'si alınması istenirse kedi ve köpek nüfusunun %45'ine ihtiyaç duyulur.3 boyutlu olursa da %58 nüfusun bilinmesi gerekir.

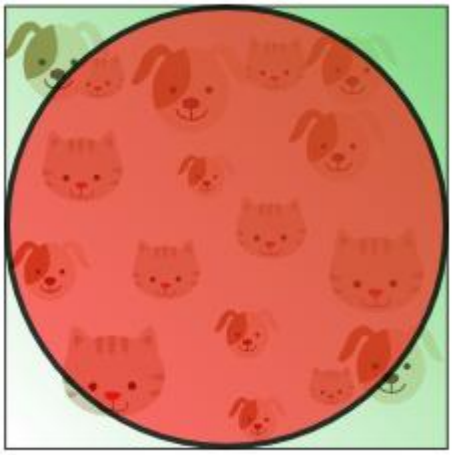


Şekil-8: Özellik aralığının %20'sinin alınması için gereken eğitim datası.

Başka bir deyişle, mevcut eğitim datası değeri sabit tutulup boyut eklenmeye devam edilirse overfit oluşur.Öteki yandan boyut eklemeye devam edersek, overfit oluşmaması için gereken eğitim datası değeri üstel olarak artış gösterir.

Önceki sayfadaki örnekte boyutluluğun lanetinin eğitim datasının seyrekliğini de getirdiği gösterilmiştir. Daha çok özellik kullanıldıkça data gittikçe ayrıklaşır ve sınıflandırıcının parametrelerinin hesaplanması zorlaşır. Boyutluluğun lanetinin başka bir efekti ise bu ayrılıklığın uzay boyunca düzgün olarak dağılmamış olmasıdır. Aslında orijin etrafındaki data köşelerde olan datadan çok daha seyrektir.Açıklaması olarak:

2 boyutlu bir özellik uzayı hayal ediniz. Bu uzayın ortalaması birim dairenin merkezidir ve tüm noktalar merkeze eşit uzaklıktadır.Eğitim dataları bu birim dairenin içine düşmez, merkezden çok köşelere yakındırlar. Bu örneklerin sınıflandırılması zordur çünkü özellik değerleri oldukça değişiktir.



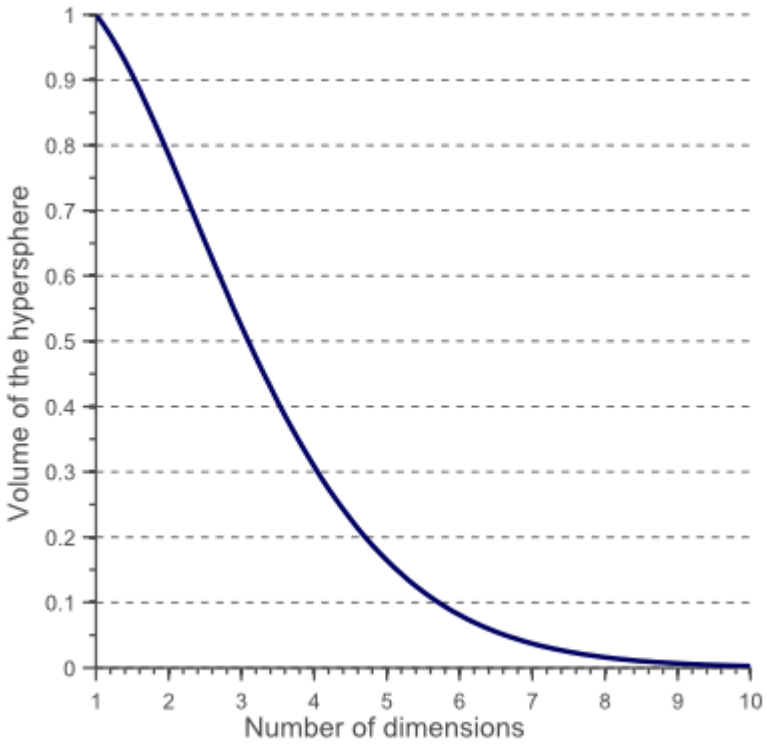
Şekil 9: Eğitim datası irim çemberin dışında kalmaktadır ve sınıflandırılması merkezdekilere nazaran daha zordur.

Şimdi ele alınacak soru ise özellik uzayının boyutluluğunu arttırdığımızda bu dairenin(hypersphere) hacminin karenin(hypercube) hacmine göre nasıl değiştiğidir. '*d*' boyutluluğun hypercube hacmi her zaman $1^d = 1$ olarak çıkar. 0.5 radius'a sahip ve d boyutlu bir hypersphere'in hacmi ise aşağıdaki eşitlikten hesaplanır.

$$V(d) = \frac{\pi^{d/2}}{r^{(\left(\frac{d}{2}\right)+1)}} . 0.5^d$$

Eşitlik 1

Şekil 10'da hypersphere'in hacminin boyutluluğa göre nasıl değiştiği gösterilmiştir.



Şekil 10: Boyutluluk arttıkça hacim 0'a yaklaşır.

3. PRINCIPAL COMPONENT ANALYSIS

Boyutluluk azaltmak için en çok kullanılan tekniklerden biri PCA'dir. Sınıflandırmaya da uygun bir tekniktir çünkü özellikleri sonucu en çok etkileyen faktöre göre sıralar.Fakat dezavantajlarından biri her ne kadar fazla etkilemeyen faktörler çıkarılsa da bir bilgi kaybı olmaktadır.Çalışmanın bu bölümünde bu tekniğin python programlama dili ile baştan yapılışı gösterilecektir.Yapılan örnek geçen dönem Python içinde bulunan kütüphaneler sayesinde kısaca gösterilmiştir. Bu dönem baştan sona yazılmış ve yapılışı gösterilmiştir.

Bu uygulamada PCA 3 boyutlu bir data kullanılacaktır.Uygulamanın çoğu ekran görüntüleriyle gösterilecektir.

İlk olarak farklı ortalamaları aşağıda verilen iki sınıf tanımlanacaktır:

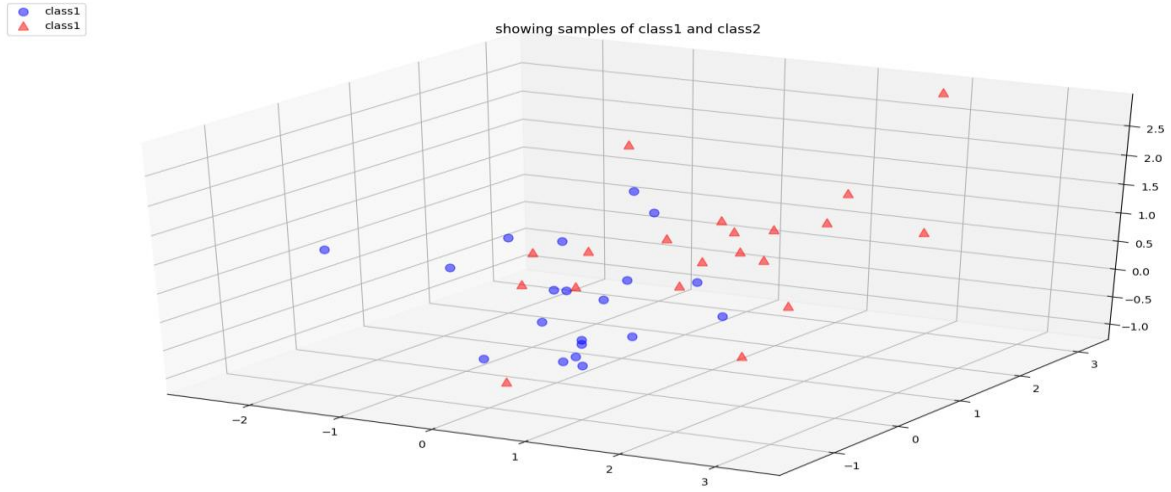
$$M_1 = \begin{matrix} 0 \\ 0 \\ 0 \end{matrix} \text{ ve } M_2 = \begin{matrix} 1 \\ 1 \\ 1 \end{matrix}$$

Ve kovaryans matrisleri de aşağıdaki şekilde olacaktır.

$$\sum_1 = \begin{matrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{matrix} \text{ and } \sum_2 = \begin{matrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{matrix}$$

Python programlama dili NumPy kütüphanesi "numpy.random.multivariate_normal(mean,cov,count)" fonksiyonu sayesinde kullanıcıya istediği ortalama ve kovaryansa sahip istenen sayıda rastgele matris oluşturma seçeneğini sunar.

Sınıfları belirlemek için mavi 'o' ve kırmızı '^' sembolleri kullanılıp 3 boyutlu olarak çizdirilip arka sayfada gösterilmiştir.



Şekil 11: Sınıfların gösterimi

Sınıflar 3 x 40 boyutlu bir matriste toparlanmıştır.Bu işlem için "numpy.concatenate' fonksiyonu kullanılmıştır.

```
[[ 1.4077577   1.45039771 -0.75362439  1.08822088 -0.32142669 -0.32203262
   1.5381673   0.7274116   0.09110041 -0.35578433 -0.19364509 -0.04994766
  -0.05317665  0.57930556 -2.49883807  0.3499062  -0.50637802  0.03350641
   0.87070202 -0.52900756  1.47907637  1.91502741  1.19866149 -0.02958172
  -0.06852473  0.74251098  1.44950192 -0.55951487  1.77166079  1.01250966
   3.24650464  0.94248644  1.49677109  0.27177084  0.70103383  1.3430552
   1.96620735  1.04089779 -0.02246304 -0.65240081]
 [ 0.1553301   0.4861947   2.25571643 -1.49433105 -0.63643219 -0.22391806
  -1.00299436 -0.70134747  0.95792371  0.28297427  0.43400379  0.17038731
   0.18959284  0.6867468  -0.03389171 -0.10174467  1.42309451  0.34179953
  -0.81642395 -0.79305317  1.22426327  3.1862446   2.49321017  2.29777307
  -0.64789334  0.08295353 -0.17369279  1.26897514  1.93707236  0.2514973
   0.90115723  1.39106774  0.38588782 -0.06469838  1.83948854  1.6852487
   0.3541545   1.40133625 -0.28103759  0.39707323]
 [ 0.71863625 -0.00817569  1.07722108  1.22169351 -0.63181921  1.31958715
   0.30701717 -0.35169456  0.17312142 -0.38230694  0.14356385 -0.97730273
   1.14345739  1.5453148   0.6211154  -0.39291786 -0.49167561 -1.11001963
  -0.04791633  0.9883487   1.22104489  2.89331714  0.77833123 -0.06587257
  -0.98360637  2.95325161  0.80601761  0.44182336  1.61552302  1.35782717
   1.62929509  1.00875394  1.7132726   0.51829768  0.42979759 -0.34515647
   1.13273093 -1.19522842  1.13744185  0.18605188]]
```

Şekil 12: Matrislerin toparlanmış hali.

Satırların ortalamaları alınıp bir ortalama vektörü oluşturulmuştur.

```
Mean Vector is :
 [[ 0.54494513]
 [ 0.53774272]
 [ 0.55245405]]
```

Şekil 13: Ortalama vektörü

Eigendecomposition'ı yapılacak Scatter matrisi aşağıdaki eşitlik ile hesaplanmıştır.

$$S = \sum_{k=1}^{n} (x_k - m)(x_k - m)^T$$

Eşitlik 2: Scatter matris formülü.Burada '*m*' ortalama vektörüdür.

Bu matrisin yerine kovaryans matrisi de kullanılabilirdi.Aradaki fark Eigendecomposition yapıldığında görülecektir.Kovaryans matrisinin bulunması için "np.cov()" fonksiyonu kullanılmıştır.Eigendecomposition için ise "np.linalg.eig()" fonksiyonu kullanılmıtşır.

```
Scatter matrix:
 [[ 41.07636706   7.34582884  13.75398161]
 [  7.34582884  43.61912951   7.76592859]
 [ 13.75398161   7.76592859  37.84027289]]
```

```
covariance matrix is:
 [[ 1.05324018  0.18835459  0.3526662 ]
 [ 0.18835459  1.11843922  0.19912637]
 [ 0.3526662   0.19912637  0.97026341]]
```

Şekil 14: Scatter ve kovaryans matrisleri

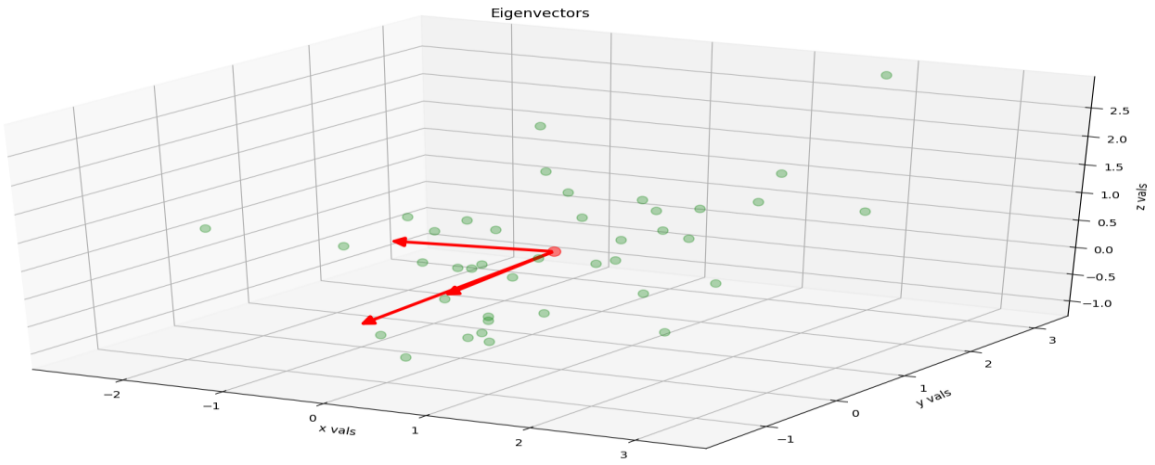
Yukarıda bahsedilen matrisler arasındaki ilişki aşağıda Eigendecomposition sonuçları ile birlikte gösterilmiştir.

```
Eigenvector 1:
 [[-0.61847844]
 [-0.54162901]
 [-0.56931751]]
Eigenvalue 1 from scatter matrix: 60.170155658075345
Eigenvalue 1 from covariance matrix: 1.5428245040532125
Scaling factor:  39.0
----------------------------------------
Eigenvector 2:
 [[-0.64401353]
 [-0.06575637]
 [ 0.76218284]]
Eigenvalue 2 from scatter matrix: 25.548721593614538
Eigenvalue 2 from covariance matrix: 0.6550954254772963
Scaling factor:  39.0
----------------------------------------
Eigenvector 3:
 [[ 0.45025659]
 [-0.83804183]
 [ 0.30814752]]
Eigenvalue 3 from scatter matrix: 36.81689220015941
Eigenvalue 3 from covariance matrix: 0.944022876927164
Scaling factor:  39.0
```

Şekil 15: Scatter matrisi ve kovaryans matrislerinden elde edilen eigenvalue ve eigenvectorler.

Görüldüğü üzere eigenvector'ler aynı çıkmaktadır.Ve eigenvalue'lar arasında 1/ (N-1) kadar bir ölçekleme faktörü oranı vardır. N bizim örneğimizde matrisimiz 3 x 40 olduğundan dolayı 40 olarak alınmıştır.

Aşağıda ortalanmış eigenvectorler gösterilmiştir.



Şekil 16: Ortalamaya göre ortalanmış eigenvectorler

PCA'de en çok bilgi içeren kısımların alındığı belirtilmişti.Elde olan 3 eigenvalue aşağıda büyükten küçüğe sıralanmıştır.

```
----------------------------------------
60.1701556581
36.8168922002
25.5487215936
```

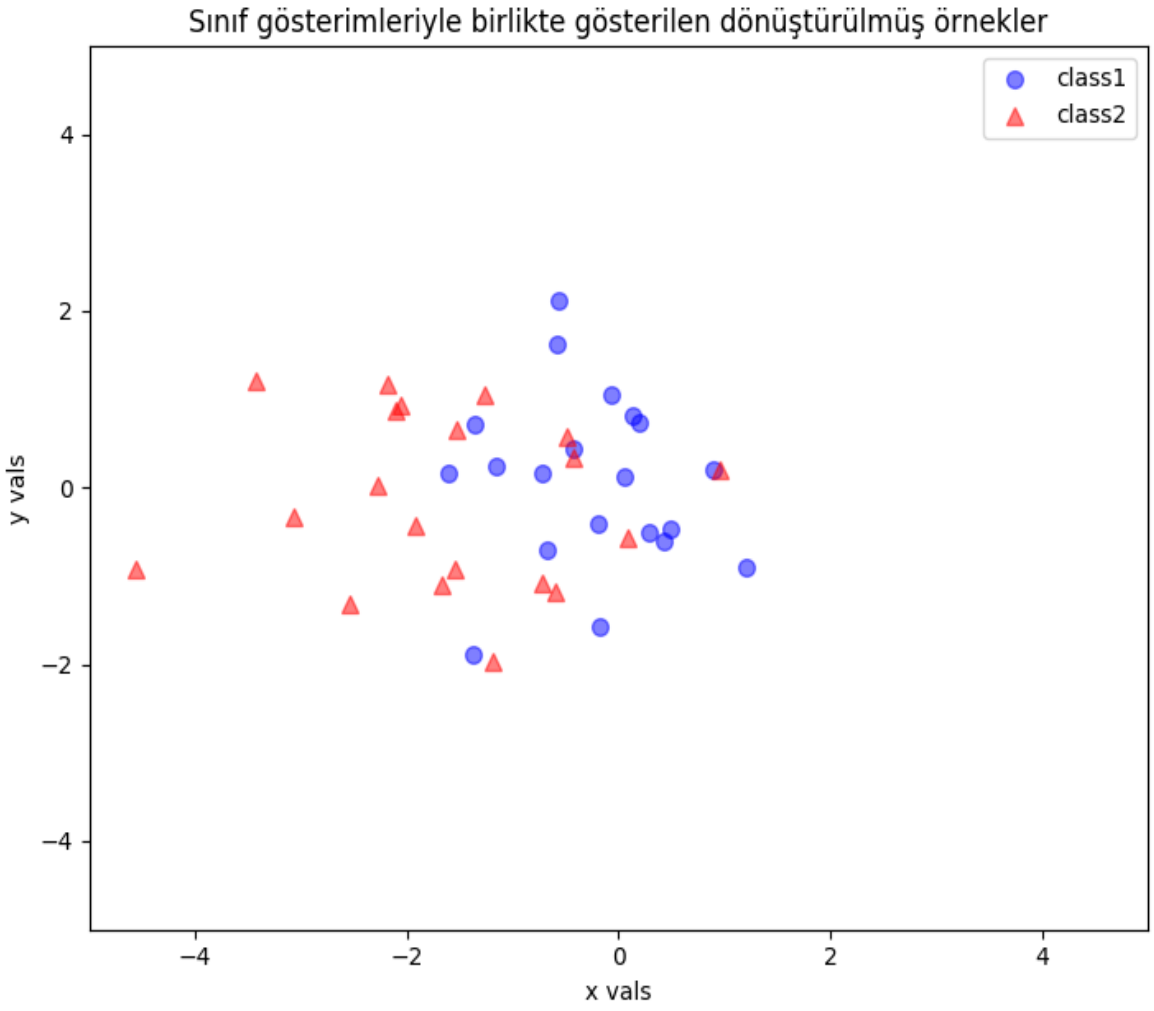
Şekil 17: Sıralanmış eigenvalue değerleri.

En çok bilgi en büyük eigenvalue değerlerine karşılık gelen eigenvectorler tarafından taşınır.Burada en çok bilgi içeren k adet eigenvalue seçilip W matrisi oluşturulur. $k = 2$ alınmıştır.

```
Matrix W is :
 [[-0.61847844  0.45025659]
 [-0.54162901 -0.83804183]
 [-0.56931751  0.30814752]]
```

Şekil 18: Yan yana yazılmış eigenvectorler.

Son aşamada 2 x 3 boyutlu *W* matrisi "y = W^T * x" eşitliği kullanılarak yeni bir alt uzaya indirgenmiştir.



Şekil 19 : Dönüştürülmüş dataset.

4. LINEAR DISCRIMINANT ANALYSIS

LDA ve PCA teknikleri genelde boyutluluk azaltma amacıyla kullanılan lineer dönüşüm teknikleridir. PCA, sınıf etiketlerini görmezden geldiğinden dolayı "eğiticisiz (unsupervised)" bir teknik olarak tanımlanabilir.PCA'in amacı varyansı en çok tutan yönleri(Principal Components) bulmaktır. PCA ile karşılaştırılırsa, LDA eğiticili(supervised) bir tekniktir ve sınıflar arası ayrılığı maksimumda tutan yönleri (Linear Discriminants) bulmayı amaçlar.

Yukarıdaki şekilde tanımlandığında çoklu sınıflandırma problemlerinde LDA'in PCA'den iyi olduğu düşünülebilir fakat bu her zaman doğru değildir.Örnek olarak görüntü tanımlamakta PCA tekniği, sınıflardaki örnek sayısı fazla olmadığı şartlarda LDA tekniğinden daha iyi sonuç vermektedir. Pratikte LDA ve PCA yöntemleri bir arada kullanılabilir örnek olarak boyutluluğu azaltmak için PCA tekniğinden sonra sınıflandırmayı belirtmek için LDA tekniği kullanılabilir.

Kod kısmında LDA tekniğinde sınıf etiketleri de kullanılacağından dolayı onlar da numerikleştirilmiştir.Bu aşama "LabelEncoder" ile yapılmıştır.

$$\begin{bmatrix} setosa \\ versicolor \\ virginica \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}$$

İlk aşama olarak her sınıf için ortalama vektörleri bulunmuştur.

$$m_i = \begin{matrix} \mu_{wi}\,(sepal\ length) \\ \mu_{wi}\,(sepal\ width) \\ \mu_{wi}\,(petal\ length) \\ \mu_{wi}\,(petal\ width) \end{matrix} \quad i = 1,2,3$$

Arka sayfada kullanılan datasetin bir kısmı ile ortalama vektörleri verilmiştir.

```
     sepal length in cm  sepal width in cm  petal length in cm  \
145                 6.7                3.0                 5.2
146                 6.3                2.5                 5.0
147                 6.5                3.0                 5.2
148                 6.2                3.4                 5.4
149                 5.9                3.0                 5.1

     petal width in cm     class label
145                2.3  Iris-virginica
146                1.9  Iris-virginica
147                2.0  Iris-virginica
148                2.3  Iris-virginica
149                1.8  Iris-virginica
Ortalama Vektörü Sınıf 1 : [ 5.006  3.418  1.464  0.244]

Ortalama Vektörü Sınıf 2 : [ 5.936  2.77   4.26   1.326]

Ortalama Vektörü Sınıf 3 : [ 6.588  2.974  5.552  2.026]
```

Şekil 20: Kullanılan datasetin bir kısmı ve ortalama vektörleri

Şimdi sınıf içi (Sw) ve sınıflar arası (S_B) olmak üzere iki tane 4 x 4 scatter matrisleri bulunacaktr.Sınıf içi olan scatter matrisi PCA'de gösterildiği gibi bulunmaktadır ve aynı şekilde ölçek faktörü ile bölünerek kovaryans matrisi de kullanılabilirdi.

$$S_w = \sum_{i=1}^{c} S_i \ ,$$

$$\sum_{x \in D_i}^{n} (x - m_i)(x - m_i)^T$$

Eşitlik 3: Sınıf içi matrisinin bulunması.m_i burada kullanılan sınıfın ortalama vektörüdür.

$$S_B = \sum_{i=1}^{c} N_i (m_i - m)(m_i - m)^T$$

Eşitlik 4 : Sınıflar arası scatter matrisinin bulunması. N_i burada sınıfın boyutu ve m_i sınıfın ortalama vektörüdür.

Sonuç olarak bulunan sınıflar arası ve sınıf içi scatter matrisleri arka sayfada gösterilmiştir.

```
within-class Scatter Matrix:
 [[ 38.9562  13.683   24.614    5.6556]
 [ 13.683   17.035    8.12     4.9132]
 [ 24.614    8.12    27.22     6.2536]
 [  5.6556   4.9132   6.2536   6.1756]]
between-classs Scatter Matrix:
 [[  63.2121  -19.534   165.1647   71.3631]
 [ -19.534    10.9776  -56.0552  -22.4924]
 [ 165.1647  -56.0552  436.6437  186.9081]
 [  71.3631  -22.4924  186.9081   80.6041]]
```

Şekil 21 : Sırasıyla sınıf içi ve sınıflar arası scatter matrisleri.

Bu aşamadan sonra LDA'in amacı olan sınıflar arası ayrılığı maksimumda tutan yönleri (Linear Discriminants) bulmak için "$S_W^{-1}S_B$" matrisinin eigenvalue ve eigenvectorleri bulunur.Aşağıda bulunan sonuçlar gösterilmiştir.

```
Eigenvector 1:
[[-0.2049]
 [-0.3871]
 [ 0.5465]
 [ 0.7138]]
Eigenvalue 1: 3.23e+01

Eigenvector 2:
[[-0.009 ]
 [-0.589 ]
 [ 0.2543]
 [-0.767 ]]
Eigenvalue 2: 2.78e-01

Eigenvector 3:
[[-0.8844]
 [ 0.2854]
 [ 0.258 ]
 [ 0.2643]]
Eigenvalue 3: 3.42e-15

Eigenvector 4:
[[-0.2234]
 [-0.2523]
 [-0.326 ]
 [ 0.8833]]
Eigenvalue 4: 1.15e-14
**********
```

Şekil 22 : Sonuç olarak bulunan eigenvalue ve eigenvectorler.

LDA'in giriş kısmında belirtildiği gibi tek amaç datayı sınıf ayrılığını arttıran bir alt uzaya yerleştirmek değildir, özellik uzayının boyutluluğunu azaltmak da önemlidir.(Eigenvectorler yeni alt uzayın aksislerini oluşturacağı bir özellik alt uzayına)

Fakat eigenvectorler hepsinin uzunluğunun 1 birim olmasından dolayı sadece yeni aksislerin yönünü belirtecektir.Hangi eigenvectorler'in kullanılacağını ise eigenvalue değerleri belirleyecektir.En düşük eigenvalue değerine karşılık gelen eigenvectorler datanın dağılımıyla ilgili en az bilgiyi taşıdığından dolayı onların çıkarılması istenir.Aşağıda eigenvalue değerleri büyükten küçüğe sıralanmıştır ve ek olarak eigenvalue değerlerinin varyansı da yüzde olarak verilmiştir.

```
Eigenvalues in decreasing order :

32.2719577997
0.27756686384
1.14833622793e-14
3.42245892085e-15
Variance explained:

eigenvalue 1: 99.15%
eigenvalue 2: 0.85%
eigenvalue 3: 0.00%
eigenvalue 4: 0.00%
```

Şekil 23: Sıralanmış eigenvalue değerleri ve varyanslar.Görüldüğü üzere en çok bilgi ilk eigenvalue ve eigenvector çifti üzerindedir.Data bu çift üzerinden 1 boyutlu bir alt uzaya indirgense bile çok bilgi kaybı olmayacaktır.

Artık sıra *k x d* boyutlu eigenvector matrisini oluşturmaya geldi (Bu problemde 4 x 2). Sonuç olarak 4 boyutlu özellik uzayı, 2 boyutlu bir alt uzaya indirgenecektir.

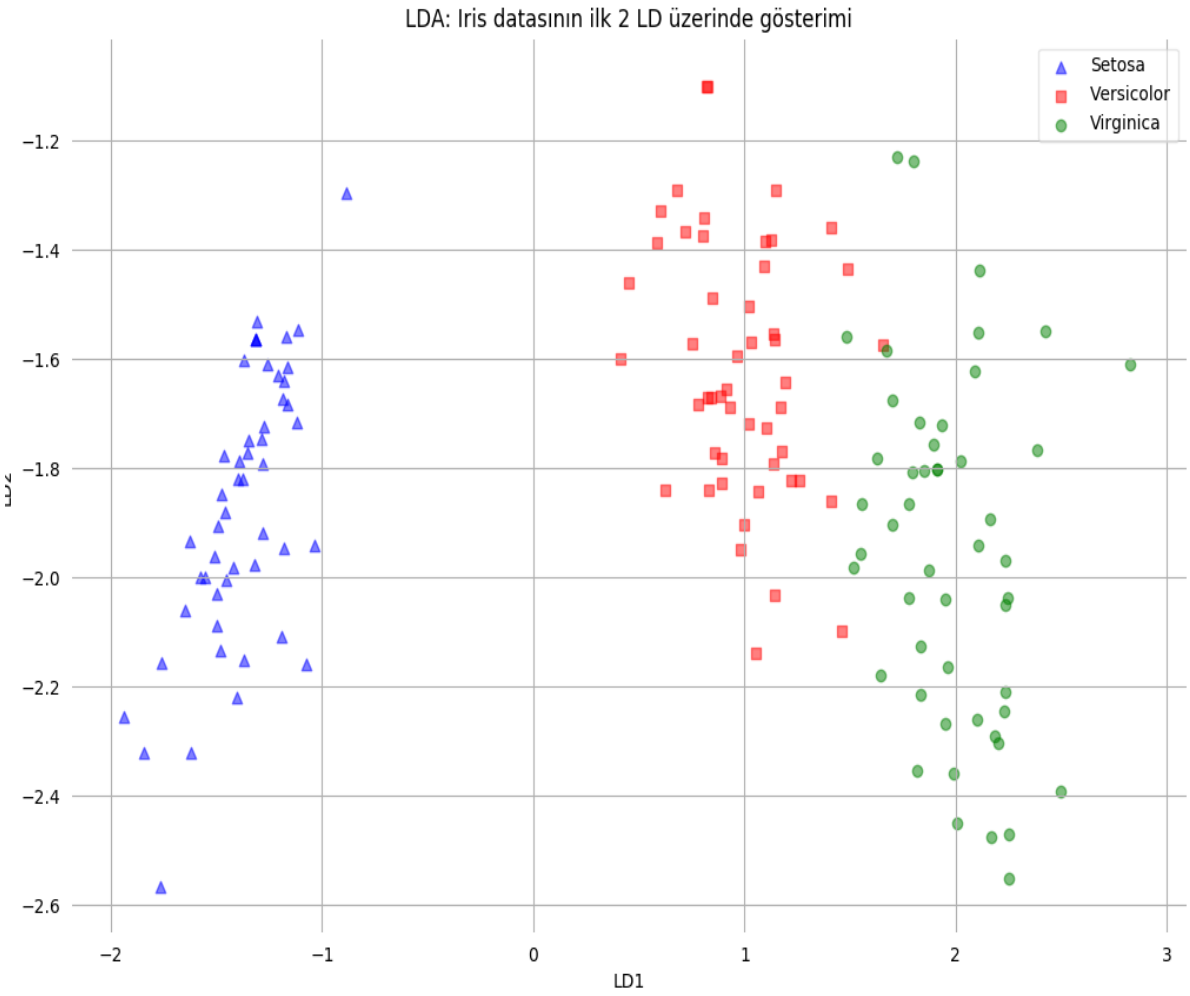
```
Matrix W:
 [[-0.2049 -0.009 ]
 [-0.3871 -0.589 ]
 [ 0.5465  0.2543]
 [ 0.7138 -0.767 ]]
```

Şekil 24: En çok bilgi taşıyan 1. ve 2. eigenvectorlerin yan yana yazılmasıyla oluşturulan W matrisi.

Son aşamada bu 4 x 2 boyutlu W matrisi kullanılarak örnekler yeni alt uzay üzerine yerleştirilip çizdirilecektir.

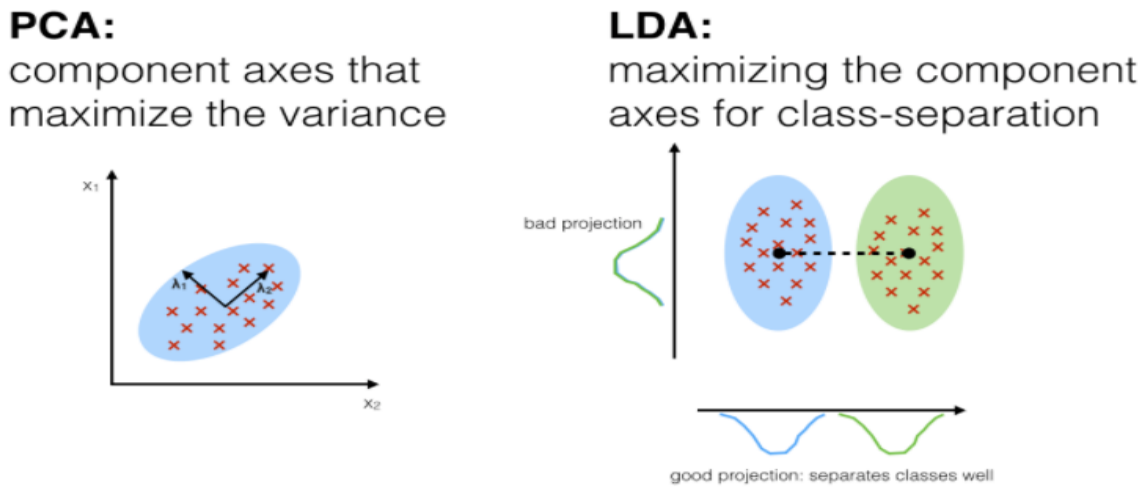
$$Y = X \; x \; W$$

Eşitlik 5 : X *n x d* boyutlu orjinal matris ve *Y* ise dönüştürülmüş *n x k* boyutlu matristir.



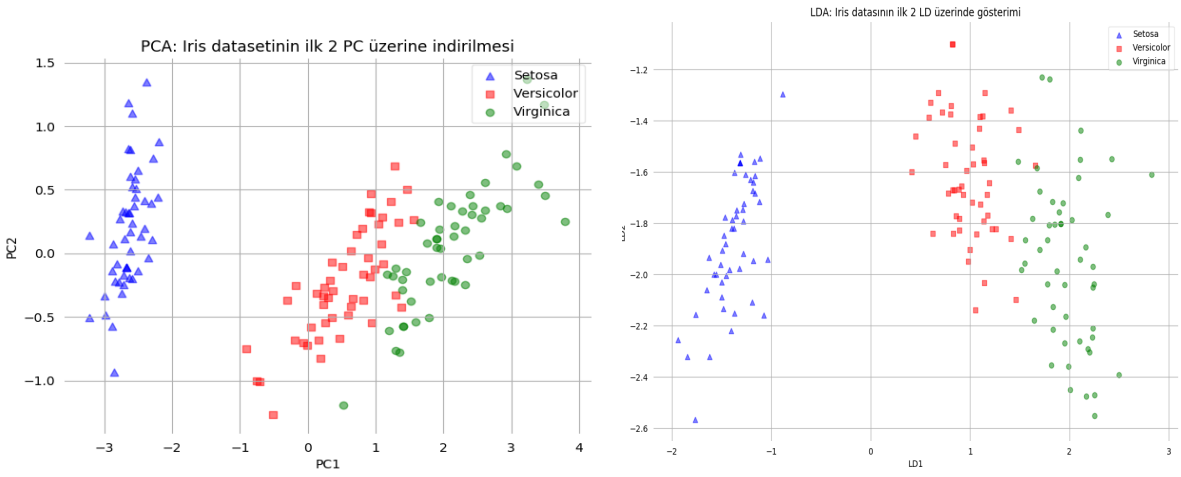
Şekil 25: Dönüştürülmüş dataset. Görüldüğü üzere LD1 sınıfları oldukça iyi ayırabilirken LD2 fazla bilgi eklemiyor.(Bkz. Şekil 23)

4. PCA VE LDA KARŞILAŞTIRMA



Şekil 26: Giriş kısmında bahsedilen PCA ve LDA arası fark.PCA varyansı maksimize etmeye çalışırken LDA sınıf ayrılığını maksimize etmeye çalışır.

Aşağıdaki şekilde PCA ve LDA tekniklerinin Iris datasetine uygulandıktan sonra bulunan sonucu verilmiştir.



Şekil 27: PCA(sol) ve LDA(sağ) sonuçları.

Yukarıdaki iki grafik önceki bahsedilenleri doğrulamaktadır. PCA tüm datasetteki varyansı maksimize etmeye çalışırken, LDA sınıflar arası varyansı maksimize etmeye çalışmaktadır.

LDA tekniği yalnızca bir boyutluluk azaltma tekniği olarak geçmemektedir.Sınıflandırma için de kullanılır.

5. BOYUTLULUK AZALTMA TEKNİKLERİYLE SINIFLANDIRMA

Önceki bölümlerde anlatıldığı gibi boyutluluk azaltma teknikleri sadece görüntüleme veya büyük boyutluluğa sahip datalar için bir pre-process amacıyla kullanılmamaktadır.Sınıflandırmanın performansını arttırmak için numerik datada bir pre-process tekniği olarak da kullanılabilir.

Boyutluluk azaldıktan sonra oluşan yeni datasete uygulanacak sınıflandırma algoritmaları orjinal datasete uygulanacağa kıyasla daha hızlı olur ve overfitting olayının önüne geçilir.

PCA tekniği[2] özellik çıkarmakta avantajı olsa da data içindeki sınıfların ayrılığına bakmadığından dolayı görüntü yeniden oluşturulması uygulamalarına daha uygundur.[3]LDA tekniği optimal bir ayrılık aradığından dolayı bu eksikliği kapatmaktadır. ICA (Independent Component Analysis) ise resimlerdeki pikseller arası ilişkiyi tanımlayarak daha iyi bir temel oluşturmayı amaçlar.[4]Bu kısımda bu üç teknik bir yüz tanıma sisteminde uygulanacaktır ve sınıflandırıcı olarak SVM (Support Vector Machines) kullanılacaktır.

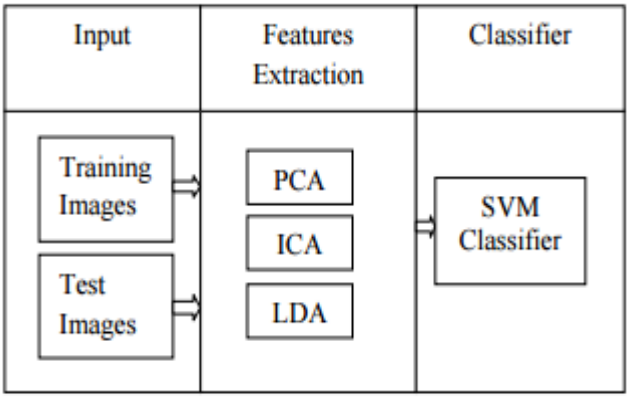
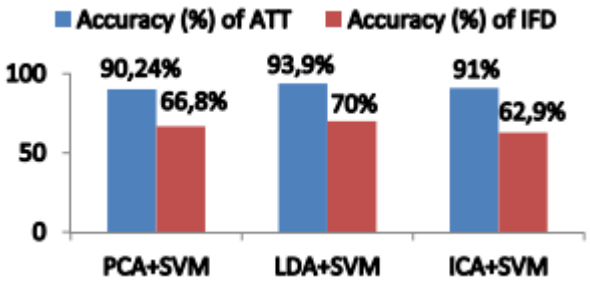


Fig 1: The face recognition system

Şekil 28: Sistemin şematiği

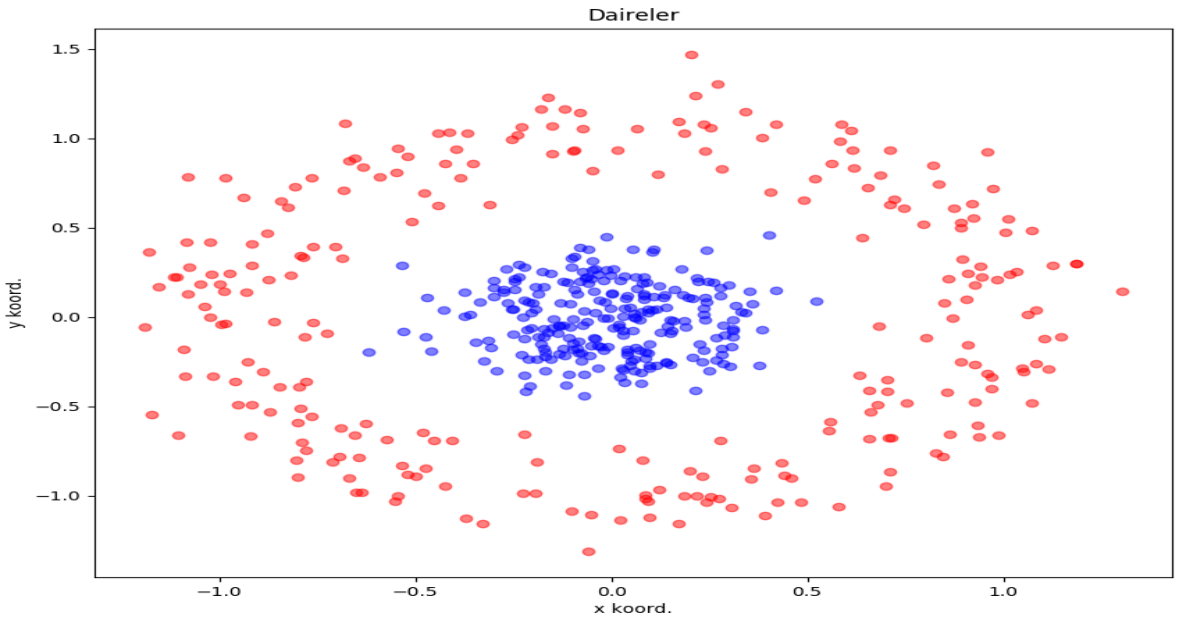
Bu uygulama "The ATT Face Database [6]" ve " The Indian Face Database (IFD) [7]" olmak üzere iki yüz datasetine uygulanacaktır. ATT datası her denek için yönleri çok az değişen görüntülerden oluşurken IFD ise her denek için 10 adet fotoğraf bulundurmaktadır ve IFD datasındaki resimler birbirinden farklı açılardadır.



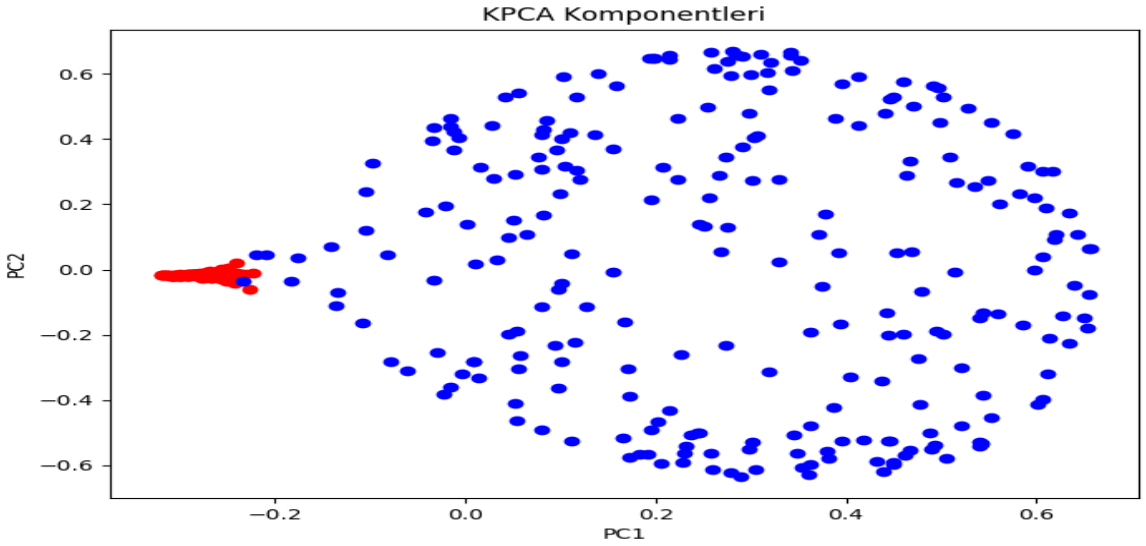
Şekil 29: Sonuçlar

Bu deneyin[5] sonucuna göre yüz tanıma problemlerinde LDA+SVM daha iyi sonuç vermektedir.

Sınıflandırmayla ilgili son örnek olarak lineer olmayan bir boyutluluk azaltma yöntemi kullanıldıktan sonra sınıflandırma yapılacaktır.Boyutluluk azaltmak için Kernel PCA kullanılıp sonrasında Support Vector Machines algoritması kullanılarak sonuçlar gösterilmiştir.Dataset olarak python dili sayesinde rastgele noktalardan oluşturulmuş iç içe geçmeyen daireler kullanılmıştır.



Şekil 30: Noktaların ayrık olmasının sebebi '0.14' kadar bir gürültü eklenmesidir.



Şekil 31: KPCA sonucu

SVM algoritmasının amacı sınıfları birbirinden ayıracak bir eğri bulmaktır,gürültü eklenmesinin sebebi bu eğrinin kolayca bulunmasını engellemektir.Aşağıda KPCA uygulanmadan önce ve sonraki sınıflandırma sonuçlarıyla beraber tahmin sonuçları da verilmiştir.

```
score:  0.96
data_k [1 1 0 0 1 1 1 1 0 0 0 1 0 1 0 1 0 1 0 1 1 0 1 0 1 0 1 1 0 1 0 0 0 0 0 0 1
 1 0 0 1 0 1 1 1 1 0 0 0 0 0 0 0 0 0 1 1 1 1 0 0 1 0 1 0 0 1 1 1 0 0 1 0 1
 0 1 0 1 0 1 1 1 1 1 0 1 1 1 1 0 0 1 0 1 1 1 1 1 1 1 0 0 0 1 0 1 1 0 0 0 0
 1 0 1 0 0 0 0 0 1 0 1 0 0 1 1 0 0 1 1 1 1 0 1 1 0 1 1 0 1 0 1 0 1 1 0 1 1
 0 1]
pred label:  [1 1 0 0 1 1 1 1 0 0 0 1 0 1 0 1 0 1 0 1 1 0 1 0 1 0 1 1 0 1 0 0 0 0 0 0 1
 0 0 0 1 0 0 1 1 1 0 0 0 0 0 0 0 0 0 1 1 1 1 0 0 1 0 1 0 0 1 1 1 0 0 1 0 1
 0 1 0 0 0 1 1 1 1 1 0 0 1 1 1 0 0 1 0 1 1 1 1 1 1 0 1 0 0 0 1 0 1 1 0 0 0 0
 0 0 1 0 0 0 0 0 1 0 1 0 0 1 1 0 0 1 1 1 1 0 1 1 0 1 1 0 1 0 1 0 1 1 0 1 1
 0 1]
score:  1.0
data [0 1 0 1 1 0 1 1 0 0 0 0 1 0 1 0 0 0 1 0 1 1 0 0 1 1 1 0 1 1 0 0 0 1 0 0 0
 1 0 1 0 1 0 1 0 0 0 0 1 0 1 0 1 0 1 1 1 1 0 1 0 1 1 0 1 0 1 0 0 1 1 0 1 1
 1 0 0 0 0 1 1 1 1 1 1 1 1 1 0 0 1 0 0 1 0 1 1 1 1 0 0 0 0 0 1 0 0 1 1 1 1
 1 0 1 1 0 1 0 1 1 1 0 1 1 0 0 1 0 1 0 0 0 0 0 0 0 0 0 0 0 1 0 1 0 0 1 1 0
 0 1]
pred label:  [0 1 0 1 1 0 1 1 0 0 0 0 1 0 1 0 0 0 1 0 1 1 0 0 1 1 1 0 1 1 0 0 0 1 0 0 0
 1 0 1 0 1 0 1 0 0 0 0 1 0 1 0 1 0 1 1 1 1 0 1 0 1 1 0 1 0 1 0 0 1 1 0 1 1
 1 0 0 0 0 1 1 1 1 1 1 1 1 1 0 0 1 0 0 1 0 1 1 1 1 0 0 0 0 0 1 0 0 1 1 1 1
 1 0 1 1 0 1 0 1 1 1 0 1 1 0 0 1 0 1 0 0 0 0 0 0 0 0 0 0 0 1 0 1 0 0 1 1 0
 0 1]
```

Process finished with exit code 0

Şekil 32: KPCA uygulanmadan önce(Üst) ve uygulandıktan sonra ki (Alt) sonuçlar

Görüldüğü üzere KPCA uygulandıktan sonra sınıflandırma sonucu daha iyi çıkmıştır.SVM sonucu her uygulamada değişecek olsa da bu örnek için her zaman daha iyi sonuç vermektedir.

6. BOYUTLULUK AZALTMA VE KÜMELEME

Boyutluluk azaltma yöntemleri kümelendirme ile de kullanılabilir bu sayede yüksek boyutlu uzayda tanımlanan data, daha küçük boyutlu bir uzayda gösterilebilir ve bu sayede datanın içinde gizli olan ilişkiler bulunabilir.

PCA önceden de anlatıldığı gibi datanın boyutluluğunu Principal Component'ları bularak düşürür. Bu komponentler datadaki varyansı maksimize eden yönleri belirtmektedir.

Eğiticisiz data analiz tekniği olan kümeleme, datayı değişkenlerinin yakınlığına göre düzenler..Bu sayede data noktalarının birbiriyle olan ilişkisi veya benzer olan gruplar keşfedilebilir.

Yapılan örnekte 1990 ve 2007 yılları arasında her 100.000 kişinin kaçının tüberküloz hastası olduğunu gösteren bir dataset kullanılmıştır.Anlaşılması için datasetin bir kısmı aşağıda gösterilmiştir.

year	1990	1991	1992	1993	1994	1995	1996	1997	1998	1999	\
country											
Afghanistan	436	429	422	415	407	397	397	387	374	373	
Albania	42	40	41	42	42	43	42	44	43	42	
Algeria	45	44	44	43	43	42	43	44	45	46	
American Samoa	42	14	4	18	17	22	0	25	12	8	
Andorra	39	37	35	33	32	30	28	23	24	22	

year	2000	2001	2002	2003	2004	2005	2006	2007
country								
Afghanistan	346	326	304	308	283	267	251	238
Albania	40	34	32	32	29	29	26	22
Algeria	48	49	50	51	52	53	55	56
American Samoa	8	6	5	6	9	11	9	5
Andorra	20	20	21	18	19	18	17	19

Şekil 33 : 1990-2007 yılları arası tüberküloz dataseti

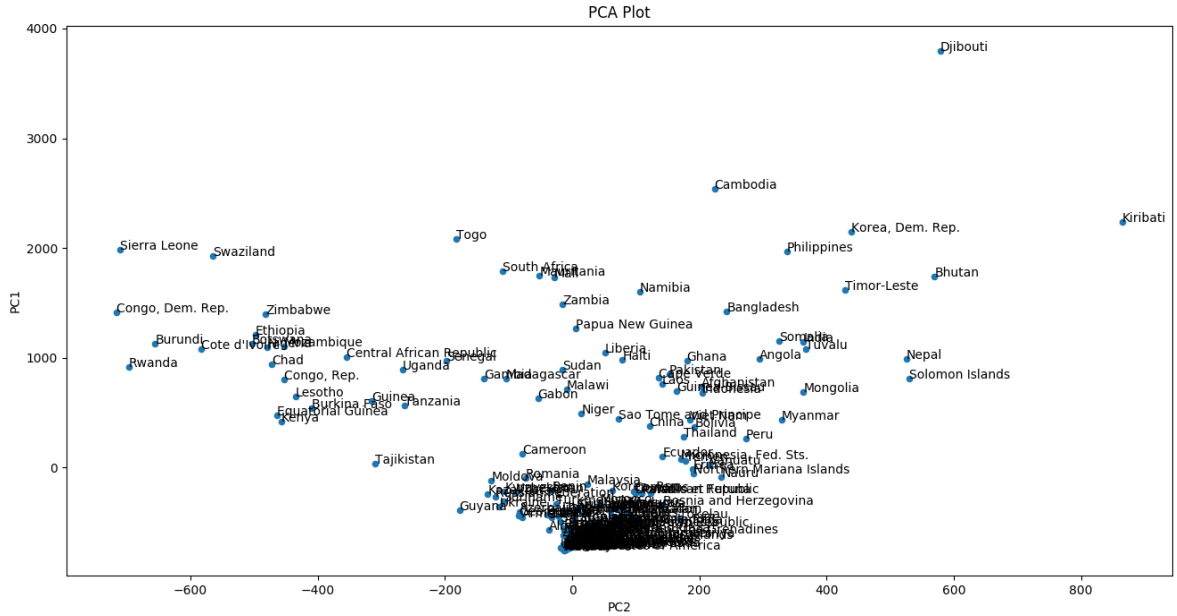
PCA uygulaması için "sklearn" kütüphanesinde bulunan PCA fonksiyonu kullanılacaktır.Ve PCA ile dönüştürülen dataset yeniden adlandırılacaktır.Aşağıda PCA sonucu ve PC'lerin varyansın yüzde kaçını temsil ettiği verilmiştir.

	PC1	PC2
country		
Afghanistan	732.215864	203.381494
Albania	-613.296510	4.715978
Algeria	-569.303713	-36.837051
American Samoa	-717.082766	5.464696
Andorra	-661.802241	11.037736

[0.91808789 0.060556]

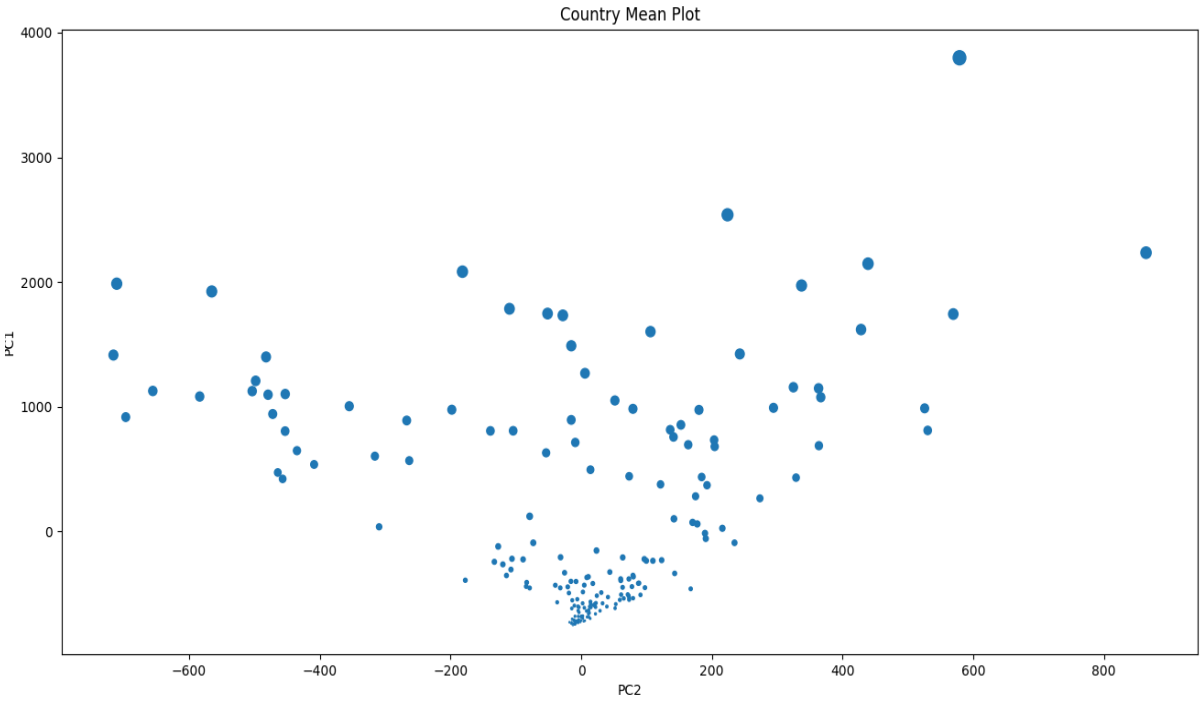
Şekil 34: Görüldüğü üzere PC1 varyansın %92'sini açıklamaktadır,ikinci komponent ise %6'lık varyansı açıklamaktadır.

Boyutluluk azaltıldığına göre dataset artık bir grafikte gösterilebilir.



,**Şekil 35 :** PCA sonucunda bulunan grafik, noktalarla temsil edilen ülkelerde gösterilmiştir.

Sonrasında noktaların boyutları yıllar boyu ortalamaya göre düzeltilip yeniden çizilmiştir.Ülke isimlerine gerek olmadığı için görmezden gelinmiştir.

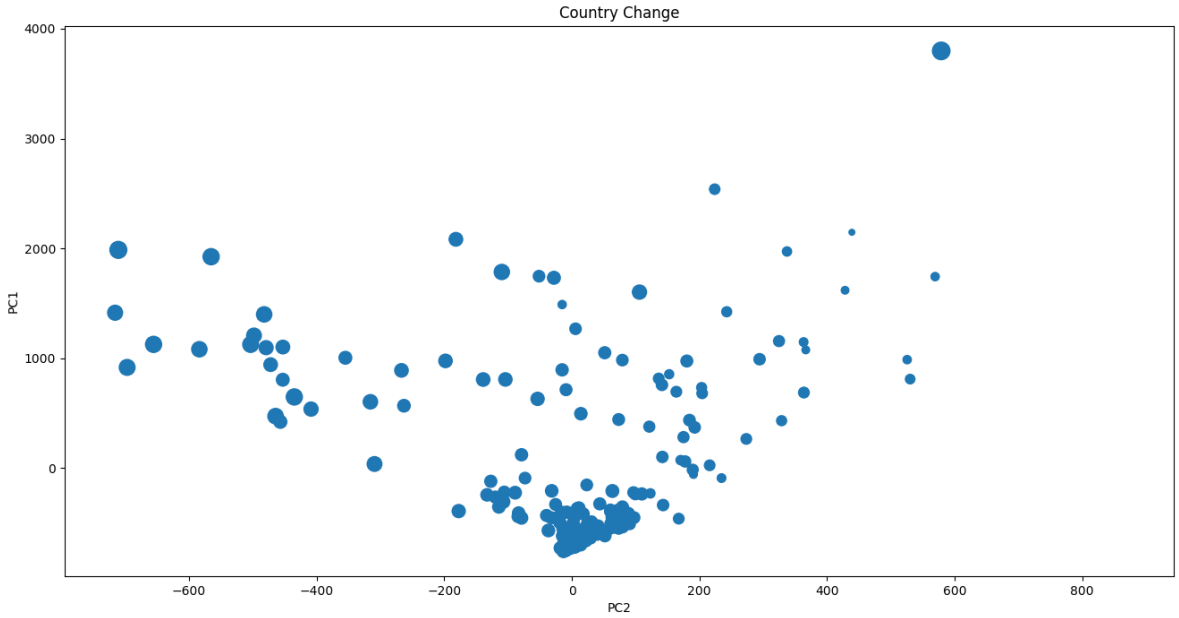


Şekil 36: Ortalamaya göre boyutları düzeltilen noktalar.

Son olarak noktaların boyutu 1990 ve 2007 yılları arasındaki değişime göre düzeltilerek grafik yeniden oluşturulmuştur.Sıfıra yakın değerler, ölçeklenme [0,1] aralığında yapıldığından dolayı ölçeklendirilmemiş değerlerdeki negatif değerlere karşılık gelecektir.

country	country_change	country_change_scaled
Afghanistan	-149	0.645977
Albania	-22	0.937931
Algeria	12	1.016092
American Samoa	-20	0.942529
Andorra	-4	0.979310

Şekil 37: Yıllar arası değişim.



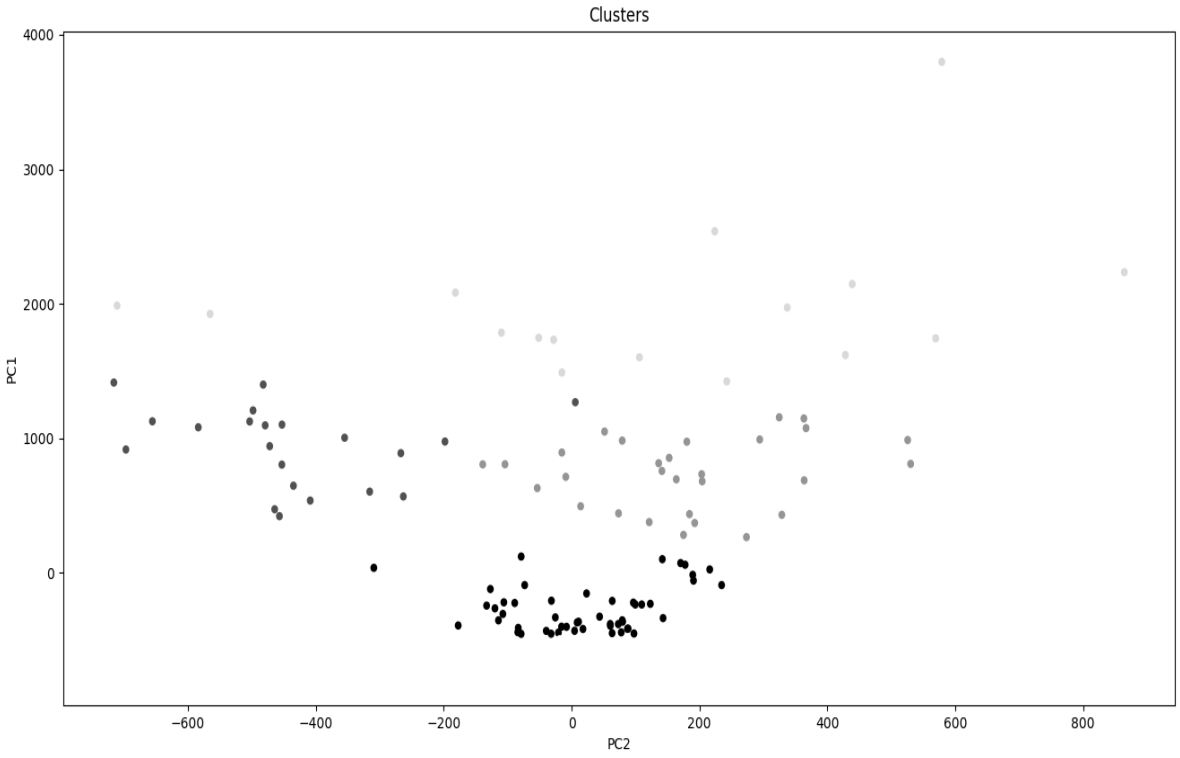
Şekil 38: Boyutları yıllar arası değişime göre düzenlenmiş grafik.

İlk sonuçta da söylendiği gibi çeşitlilik daha çok PC1 olarak atadığımız y ekseni üzerinde olmaktadır ki birinci komponentin çeşitliliğin %92'sini açıkladığı gösterilmişti.Noktaların grafiğin alt kısmında bir yoğunluk oluşturduğu görülmektedir.Bu ülkeler çoğunlukla gelişmiş ülkelerdir.Yukarı çıktıkça ülkelerin seyrekleştiği görülmektedir.Bu ülkeler daha az gelişmiş bölgelerde bulunan ülkelerdir.

Bir de yıllar arası hastalık farkını alınarak boyutlandırma yapılan bölümde, boyut daha çok pozitif değerle PC2 yönünde değişmiştir.PC1 çeşitliliğin çoğunu kapsadığından dolayı PC2 daha çok etkilenmiştir.

Bu bölümde K-Means Clustering algoritması kullanılarak ülkeler, yıl geçtikçe durumun değişimine göre gruplandırılacaktır.Bundan sonra 'Cluster Assignment' kullanılarak 2 boyutlu grafikte renklendirme yapılarak data içindeki gizli ilişkililer ve tüberküloz hastalığında dünyanın durumu gözlemlenecektir.

K-Means kullanılırken doğru grup sayısının bulunması önemlidir. Bu işlem farklı '*k*' değerlerinde işlem yapıldıktan sonra küme içi karesel uzaklığa bakılarak yapılabilir.Bu uzaklık küme merkezinin örneklerle olan uzaklıklarının toplamıdır.Bu örnek için 'k' sayısı 5 olarak alınacaktır.



Şekil 39: K-Means Clustering sonucu

4.BULGULAR

Boyutluluk azaltma tekniklerinin yüksek boyutlu dataların işlenmesinden önceki basamak olarak kullanımı dışında sınıflandırma veya kümelemede kullanıldığı söylenebilir. Fakat bu kullanım yapılırken datanın boyutluluğu çok küçültülürse bilgi kaybı olacağından dolayı buna dikkat edilmelidir veya tam tersi datanın boyutluluğu fazla olursa overfitting olayı olacağından dolayı sınıflandırıcıyı eğitecek data boyutu da oldukça artacaktır.Boyutluluğu doğru derecede azaltmak overfitting'i engellemenin yanı sıra performans olarak da önemlidir.

5.TARTIŞMA VE SONUÇ

Ödevde boyutluluk azaltma teknikleri ve kullanım alanları gösterilmiştir. Boyutluluk azaltma tekniklerinin yüksek boyutluluğa sahip dataların işlenmesinden önceki aşama olarak kullanım dışında sınıflandırma ve kümeleme için kullanılabileceği ve kullanılma sebeplerine yer verilmiştir.

KAYNAKLAR

[1] Hotelling, H. (1933) Analysis of a complex of statistical variables into principal components. Journal of Educational Psychology 24: 417–441

[2] M. Turk and A. P. Pentland, "Eigenfaces for recognition", Journal of Cognitive Neuroscience, vol. 3, no. 1,pp. 71–86, 1991.

[3] P. N. Belhumeur, J. P. Hespanha, and D. J. Kriegman, "Eigenfaces vs. Fisherfaces: recognition using class specific linear projection", IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 19, no. 7,pp. 711– 720, 1997.

[4] M. Bartlett, J. Movellan, and T. Sejnowski. Face recognition by independent component analysis. IEEE Trans. on Neural Networks,13(6):1450–1464, November 2002.

[5] http://www.jetwi.us/uploadfile/2014/1215/20141215110544125.pdf

[6] AT&T Laboratories Cambridge,http://www.cl.cam.ac.uk/Research/DTG/attarchive:pub/data/att_faces.tar.Z.

[7] Vidit Jain and Amitabha Mukherjee, The Indian Face Database.http://viswww.cs.umass.edu/~vidit/IndianFaceDatabase/, 2002

[8] Fisher, R.A (1936) The Use of Multiple Measurements in Taxonomic Problems. Annals of Eugenics 7: 179-188

[9] Peres-Neto P., Jackson D., Somers K. (2005) How many principal components? stopping rules for determining the number of non-trivial axes revisited. Computational Statistics & Data Analysis 49: 974–997

[10] Frank Plastria, Steven De Bruyne and Emilio Carrizosa Dimensionality Reduction For Classification Comparison of Techniques and Dimension Choice, 2008