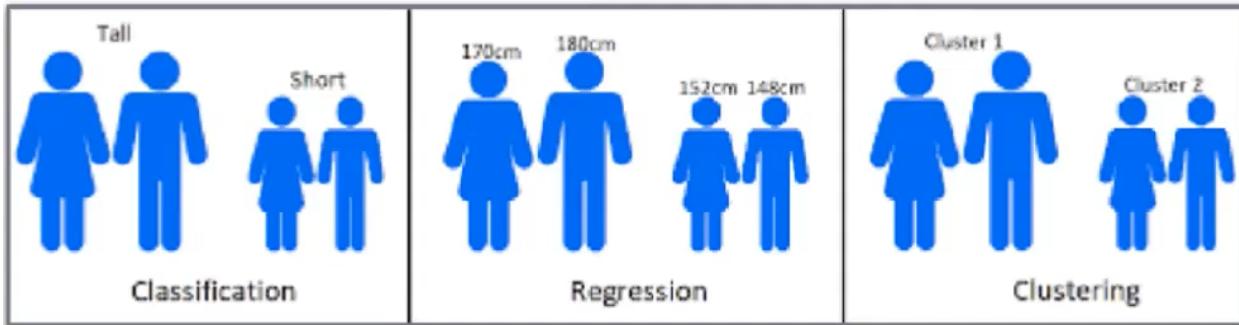
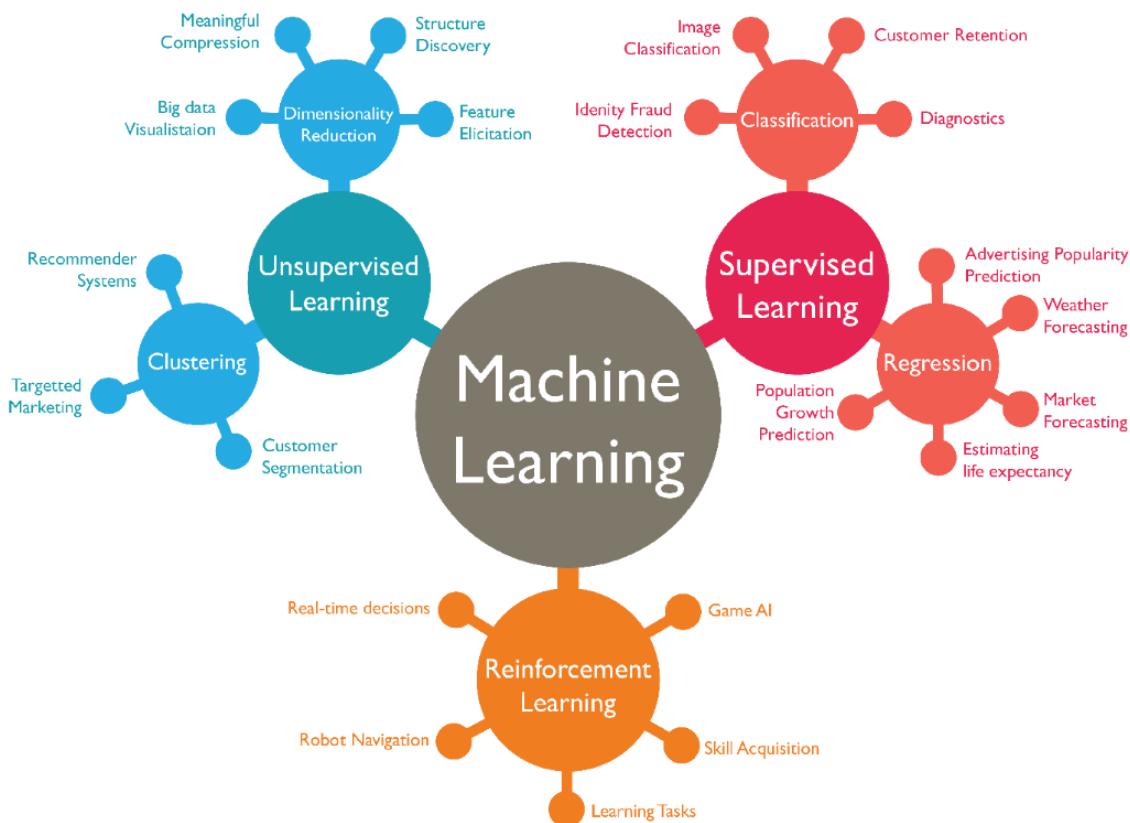


# UNSUPERVISED LEARNING



Unsupervised Learning yöntemleri Supervised Learning yöntemlerden ayıran en önemli özellik **target variable olmamasıdır**.

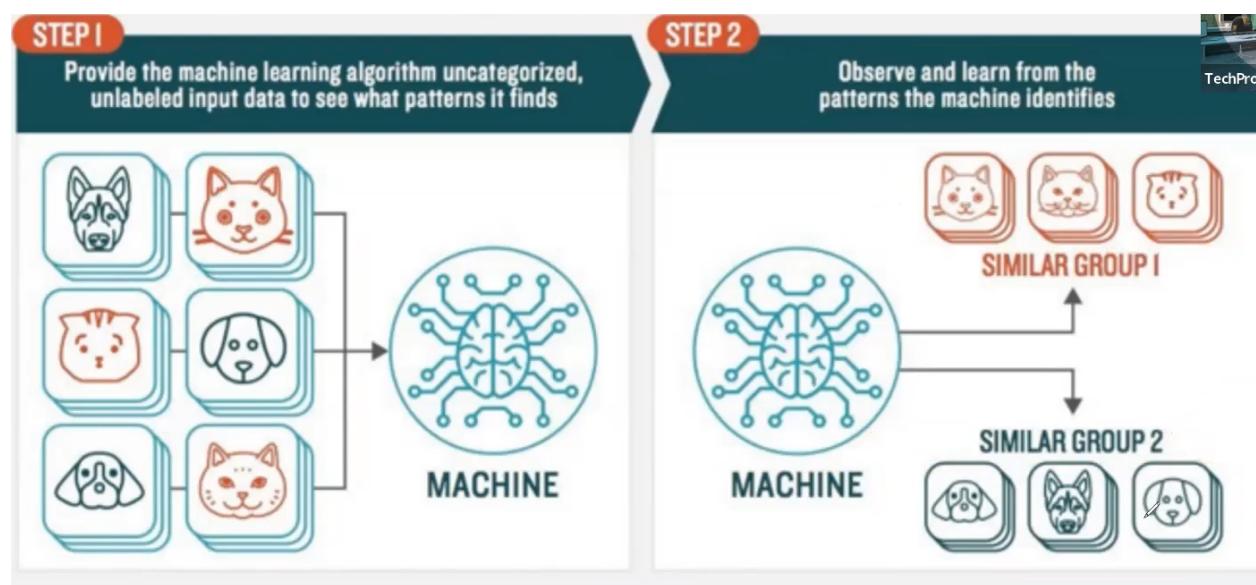


Unsupervised Learning de en öne çıkan yöntem clustering (kümeleme) dir. Clustering yöntemde en öne çıkan başlıklar ise customer segmentation, recommender system ve targetted marketing dir. müşteri segmentasyonu ticari firmalar çok önemli olduğu gibi, tavsiye sistemleride youtube ve netflix gibi platformlar için çok önemlidir. En bilinen kullanım alanları bunlardır.

Unsupervised Learning de öne çıkan diğer yöntem Dimensionality Reduction (Boyut Azaltma) dir.

PCA kavramı üzerinden açıklanan bu yöntem de boyut azaltılmış olur

Target variable in olmadığı sadece features lardan oluşan dataset içerisinde bir patern, desen yakalanmaya çalışarak benzerlikleri olan observationların farklı kümeler halinde ayırtırılmasıyla yapılan öğrenme şeklidir.

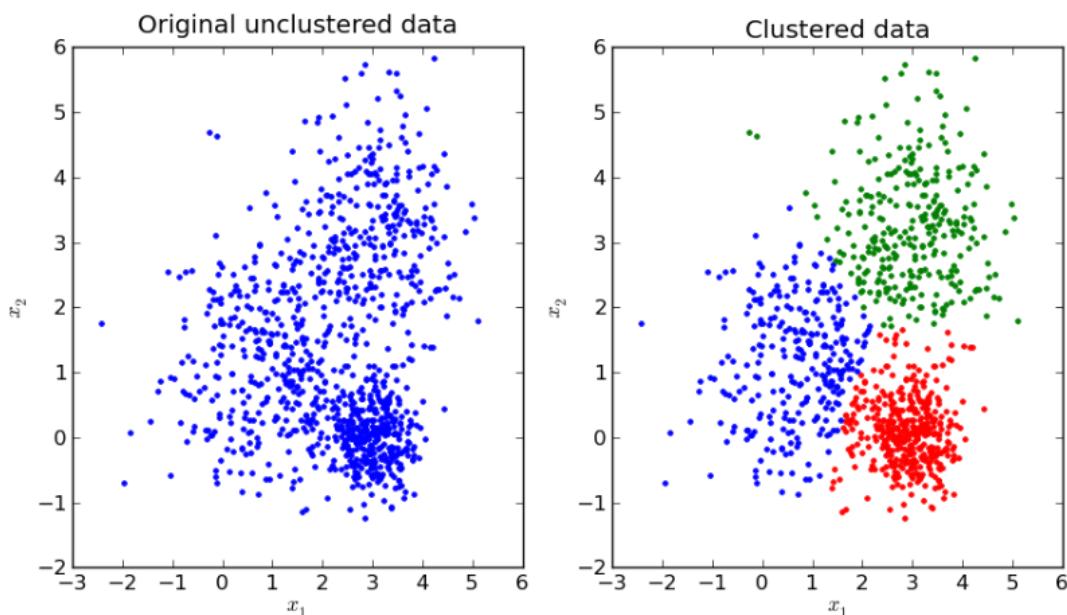


yukarıdaki örnekte sadece kedi ve köpek resimlerinden oluşan ve bunların kedi yada köpek olduğu belirtilmeyen yani sadece resimlerden oluşan datasette, bu resimler benzerlikler göz önünde bulundurularak sağ tarafta olduğu gibi ayırtırılır. Bu kümeleme yapılırken kedi yada köpek olarak yapılmaz. Bu resimler arasında bir patern yakalanmaya çalışılarak benzer gruptalar olarak kümelenir, isimlendirme yapılmaz. Sonrasında biz isimlendirme yapabiliriz.

## CLUSTERING

Clustering verileri grumlara ve kümelere ayırmak üzere kullanılan algoritmik modelleme tekniğidir. Burada amaç birbirine benzer yada yakın olan veri noktalarını belirleyerek oluşacak tablodan bilgi elde etmektir. Örneğin bir sepette çeşitli mevyeler olsun gözleriniz kapalı bir şekilde bunları benzerliklerine göre ayırdığınızı düşünün.

Kümelemede veri noktalarının gruplandırılması dataın iki özelliğine göre yapılır. Ortaya çıkan kümelerin anlamlılığı (Meaningfulness) ve faydalılığı (Usefulness) dır. belirli sayıda oluşturulan kümeler anlamlı mı yada faydalımı. Burdan çıkan sonuçlar görecelidir. Model sonrası ortaya çıkan kümelerin domain knowledge ile değerlendirilmesi gereklidir. Customer segmentation yapılırken domain knowledge önemlidir.

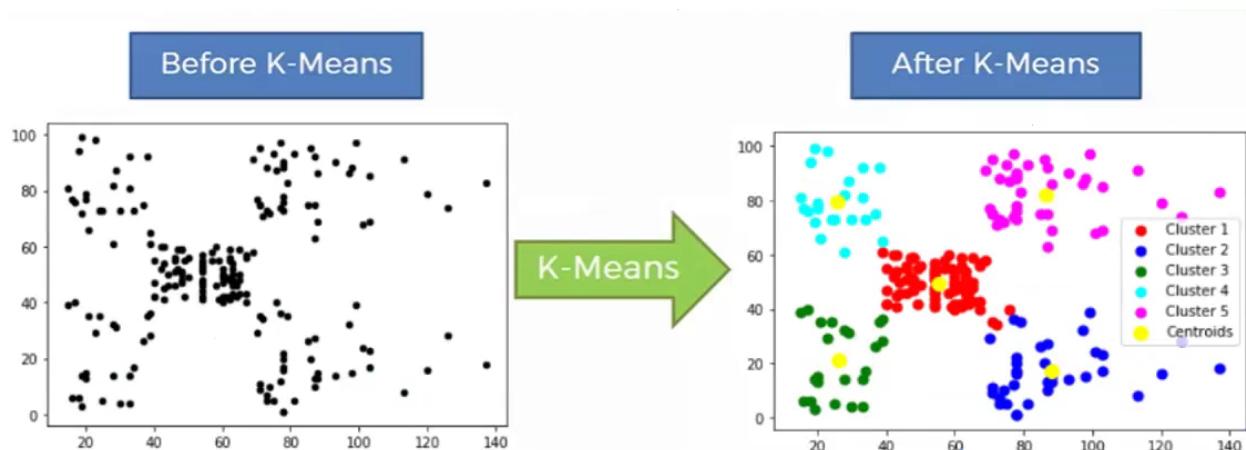


Müşterilerin kaç gruba ayrılacağı, anlamlılığı ve faydalılığı değerlendirilerek karar verilebilecek bir durumdur. Yada işin sahibinden gelecek talebe göre kaç gruba ayıracagımıza karar verebiliriz. Bizden 4 gruba ayırmamız istenirken model sonrası 3 yada 5 grubun olduğu da ortaya çıkabilir. Buna göre değerlendirilir. Sektörel olarak grup sayısı değişebilir. Bankacılık sektöründe ortaya çıkacak grup sayısı ile e-ticaret sektöründe ortaya çıkacak grup sayısı farklı olabilir.

Clustering yapılırken biri çok az kullanılmakla birlikte 3 yöntem kullanılır. Partitioning Clustering (Bölümlü kümeyeleme), Hierarchical Clustering ve Density-based Clustering.

**PARTITIONING CLUSTERING** (Bölümlü kümeyeleme) de veri noktaları birbirleriyle örtüşmeyecek şekilde gruptara ayrılmışdır. En bilinen algoritması K-Means dır. Bu algoritmada siuet ve elbow metodu kullanılarak sonuca ulaşılır. Center-based bir yöntemdir. Kümeler için centroids ler oluşturularak, her bir veri noktasının bu merkezlere uzaklığuna bakarak bir kümeye dahil eder. Burada başlangıçta kaç kümeye olacağının modelimizde bir parametre ile belirlenir. Kaç kümeye olacağının domain knowledge, iş sahibi talebiyle karar verilebileceği gibi matematiksel yöntemlerle de (elbow ve siluet yöntemleri) karar verilebilir.

Algoritmamız; kümeye sayısının belirlenmesi sonrasında kümeye sayısına göre centroids ler yani kümeye merkezleri belirler ve veri noktalarının bu centroids lere olan uzaklıklarına göre kümeyeleme yapar, ilk kümeyeleme sonrasında her kümeye için yeniden centroids leri belirler ve bu centroids lere göre baştan tüm veri noktaları üzerinden tekrar kümeyeleme yapar. Bu işlemler bizim parametre olarak girdiğimiz sayı kadar tekrarlar ve en son aşamada bulmuş olduğu optimum kümeleri oluşturmuş olur. Bu yöntemde aynı dataset üzerinde algoritmamızın her çalıştırılmasında ilk başta seçilecek centroids lerin farklı olması durumu göz önüne alındığında farklı sonuçlar almak mümkündür.



K-means clustering algoritmasında

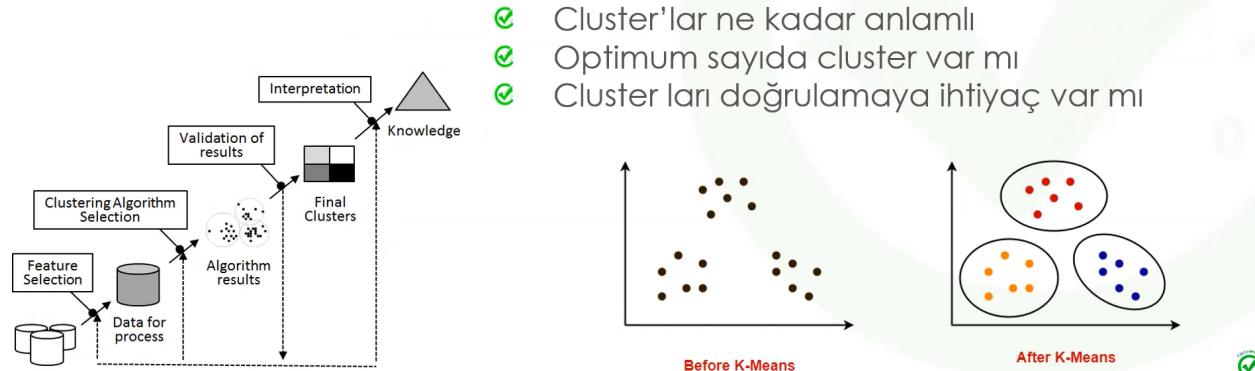
**n\_cluster** : küme sayısı belirlenir ve dataset bu sayıda kümeye ayırtırılır

**init** : k-means++ yada random seçeneği ile kümelerin merkezini yani centroids leri belirler

**n\_init** : centroids lere göre kümeleri oluşturduktan sonra tekrar kümelerin merkezini belirler ve bu merkezlere göre tekrar kümeleri oluşturma işlemini kaç defa tekrarlayacağı belirlenir

**max\_iter** : 10000 olarak denemek faydalı

## Cluster Modelleri için PERFORMANS DEĞERLENDİRME METRIKLERİ



Clustering de performance değerlendirme görecelidir. Supervised modellerde target variable olduğu için alınan sonuçların farklı score larla değerlendirilmesi mümkün. Unsupervised modellerde target varaible olmadığı için score almak mümkün olmuyor ve **sonuçlar domain knowledge ile değerlendirilir**.

**cluster lar ne kadar anlamlı ve amaca ne kadar uygun, optimum sayıda cluster var mı ve cluster sayısını doğrulanmaya ihtiyacı var mı, cluster lar karar alma süreci için sağlam bir bilgi veriyor mu, bu soruların cevabına göre değerlendirme yapılır.**

# ✓ Performans Ölçütleri

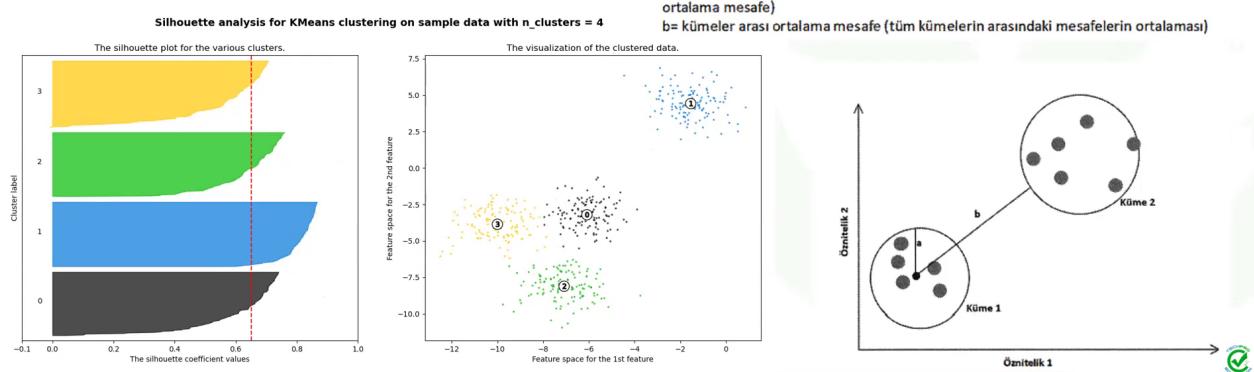
- ✓ Dışsal Kümeleme Doğrulaması
- ✓ True-Maching (TM)
- ✓ Adjusted Rand Index (ARI)
- ✓ Mutual Information Score (MIS)
- ✓ V-measure:
- ✓ Eksiksizlik (Completeness)
- ✓ İçsel Kümeleme Doğrulaması

clustering de yukarıdaki performans ölçütleri olsa da pek kullanılmıyor.  
aşağıdaki iki performance ölçütünü kullanlıyoruz.

## ✓ Performans Ölçütleri

### ✓ Silhouette Metodu

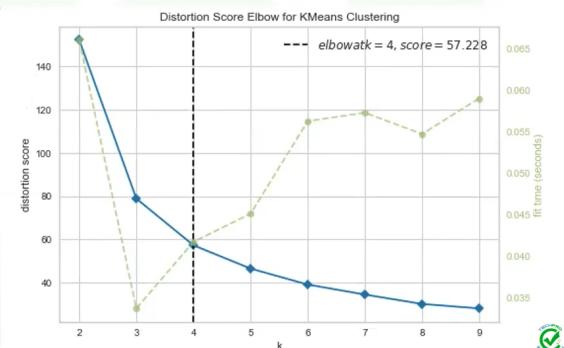
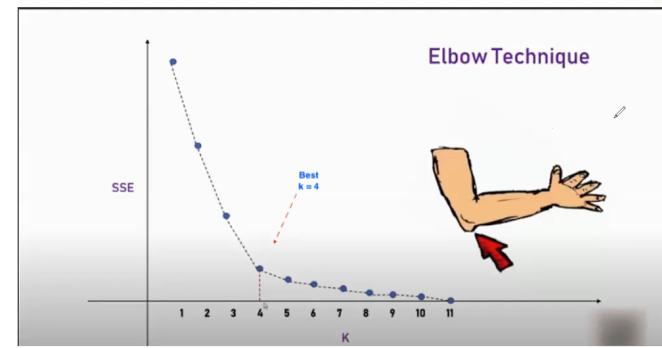
$$\text{Silhouette Katsayı} = (b - a) / \max(a, b)$$



Silhouette Medtodunda Silhouette katsayısı kümelemenin ne kadar başarılı yapıldığını gösterir. -1 ile 1 arasında değer alır. -1 kümelerin veri noktalarını iyi ifade edemediği grupları iyi oluşturamadığı grupların yanlış oluşturulduğunu ifade eder. 0 ile kümelerin birbirleriyle alakasız bir şekilde oluşturulduğunu,

aralarındaki mesafelerin anlamsız olduğunu ifade eder. 1 ise kümelerin veri noktalarını tamamen ayırtılacak şekilde oluşturduğunu yani kümelerin doğru bir şekilde oluşturulduğunu ifade eder. **1 e ne kadar yakınsa o kadar başarılıdır.**

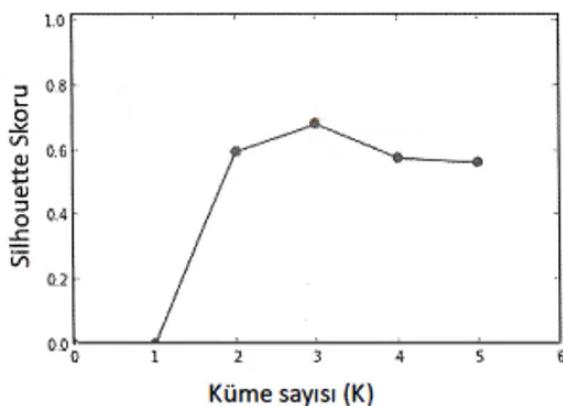
## ✓ Performans Ölçütleri ✓ Elbow Metodu



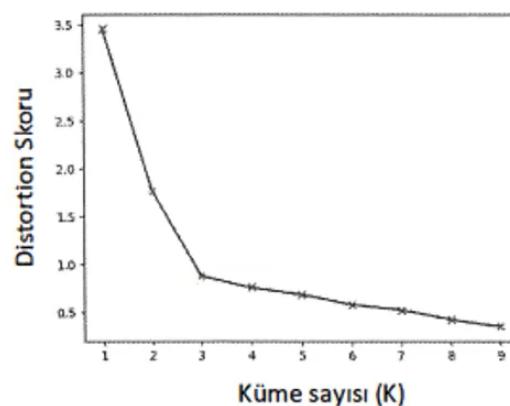
Elbow Metodunda 1 den başlayarak istenilen sayıya kadar küme oluşturarak farklı iterasyonlarla kümeler oluşturulur ve veri noktaları en yakın olduğu merkeze göre kümeye atanır. Bu mesafeler distortion ve inertia olarak iki şekilde hesaplanır, bunlardan biri tercih edilir. Distortion kullanılır genelde. Her K sayısına göre veri noktalarının merkeze olan mesafeleri hesaplanarak en iyi K değeri bulunur.

**Silouette ve elbow sonuçları karşılaştırılarak küme sayısı belirlenir.**

### Silhouette ve Elbow Metotlarına Göre Optimum Küme Seçimi



Silhouette Metodunda, optimum küme sayısı (K) farklı küme sayıları için hesaplanan skorlardan en büyüğüne karşılık gelen değerdir. Bu örnekte K=3 optimum olarak görülmektedir.



Dirsek Metodunda, optimum küme sayısı (K) grafiğin dirsek şeklinde karıldığı/büküldüğü noktaya karşılık gelen değerdir. Bu örnekte K=3 optimum olarak görülmektedir.

Son tahlilde, bir değerlendirme ölçüyü olmakla birlikte denetimli makine öğrenmesinde kullanılan değerlendirme ölçütleri kadar kesinlik atfetmeden bir miktar ihtiyatla yaklaşmak gereklidir.

**HIERARCHICAL CLUSTERING** de ise iki yöntem vardır, Divisive (yukardan aşağı) yada Agglomerative (aşağıdan yukarı) olarak hierarşik kümeleme yapılır. Agglomerative yöntemde, tüm noktalar tek bir kümede birleşinceye kadar en çok birbirine benzeyen iki noktası birleştirerek ilerliyor. Divisive de ise tüm noktalara tek bir küme olarak bakarak başlar ve yalnızca bir veri noktası kalana kadar her aşamada benzer olanları bölgerek ayırtılmaya devam eder. Burada Partitioning Clustering den önemli bir ayrıntı olarak aynı datasette algoritma farklı bir şekilde tekrar çalıştırılmış olsa da sonuç değişmez ayışma yani kümeler aynı olur.



Veri noktaları arasındaki bu örtülü ilişkileri göstermek için yorumlanması daha kolay olan görselleştirilmiş **dendogram** oluşturulur.

Hierarchical clustering de kullanılan en önemli algoritmaların biri Principal Component Analysis (Temel Bileşenler Analizi) PCA dır. PCA i fetaure extraction ile benzetebiliriz.

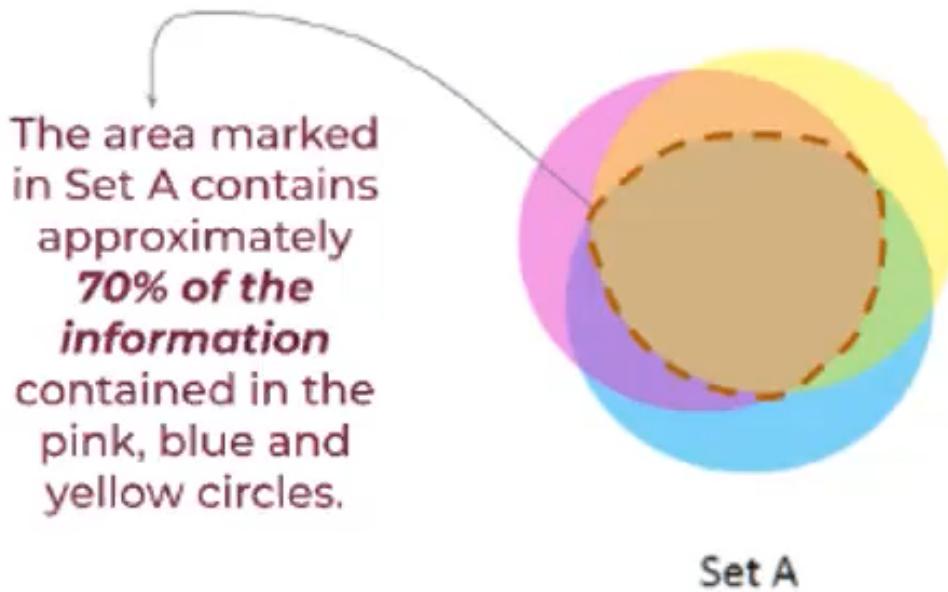
**PCA, değişkenlerdeki bilgilerin çoğunu temsil eden daha küçük bir bileşen kümese indirmeye çalışan bir karmaşıklık azaltma tekniğidir.**

Datasetinde yer alan her bir değişken fetaure boyut ifade eder. 3 boyuttan sonrası insan algılaması mümkün değildir. Fetaure sayısının artması boyut sayısının artmasına complexity nin artmasına neden olur. Fetaure sayısının çok fazla olması durumunda aralarında ilişki olan fetauresların bir araya gelerek yeni bir bileşen oluşturmasını temel alan bir yöntemdir. Telekomunikasyon şirketine ait verilerde aylık yada yıllık ödemelerin yer aldığı fetauresların aralarında güçlü bir ilişki olması yada vücut ölçülerinin yer aldığı verilerde bel genişliğini belirten fetaure ile vücut ya  oranını ifade eden feature arasında ilişki olması gibi. Multicollinearity olarak ifade etti imiz durum. Bu özelliklerden yeni

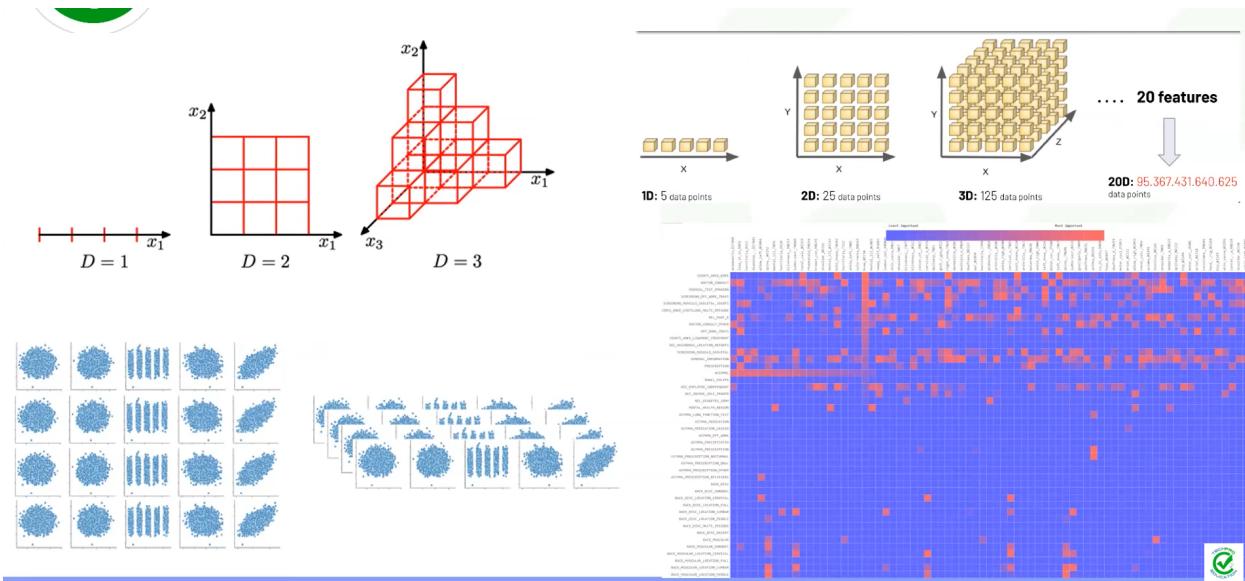
bir fetaure oluşturabiliriz yada boyut azaltabiliriz (dimensioanality reduction). Bir olayı açıklarken 100 fetaures kullanmak yerine 10 fetaures kullanmak daha user friendly, anlaşılabilir bir değerlendirmektedir. Complexity azaltılmış olur.

Kavramsal olarak PCA; varyansı paylaşan değişken kümelerini tanımlar ve bu varyansı temsil eden yeni bir bileşen oluşturur.

Büyük datasetlerdeki çok boyutluluk PCA ile azaltılmış olur



yukarıdaki görselde görüldüğü gibi 3 farklı rengin bir araya gelmesiyle 3 ünün kesişim noktasında olan alan %70 oranında 3 rengi ifade ediyorsa onları açıklayabiliyorsa 3 features yerin tek bileşen kullanarak boyut azaltmış oluruz. Bu durumun 100 lerce featuresın yer aldığı bir datasette 10 larla ifade edilen bileşenle yapıldığını düşünürsek ortaya çıkan kazanç anlaşılabilir. Ayrıca analiz için görselleştirme kolaylaştırılmış olur.



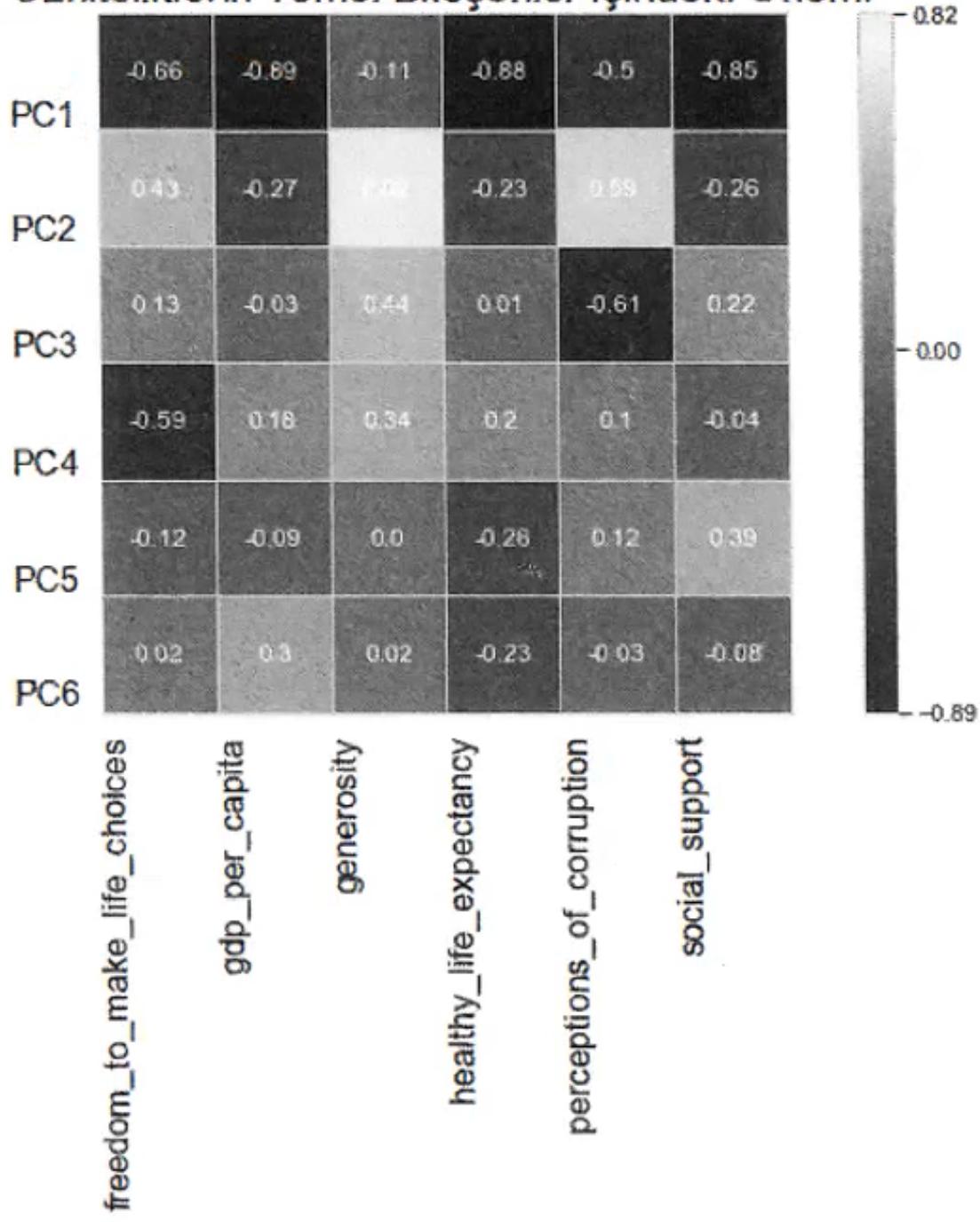
yukarıda sağ altta yer alan features lar arasındaki ilişkiyi gösteren heatmap göz önüne alındığında features lar arasında değerlendirme yapmanın ne kadar zorlaştığı gözlenebilir.

sol alttaki görselde ise 25 feature ve 300 observation dan oluşan dataseti için scatter plot yapmak istediğimizde 25 fetaures arasındaki ilişki için  $25 \times 25$  scatter plot tablosu oluşturulmalı. Bu durumda olayın anlaşılması ve okunması zorlaşır.

**PCA, verilerin boyutsallığını azaltmak için ve daha iyi görselleştirme yapabilmek için mevcut bileşenlerden daha az sayıda yeni bileşenler elde etme teknigidir.**

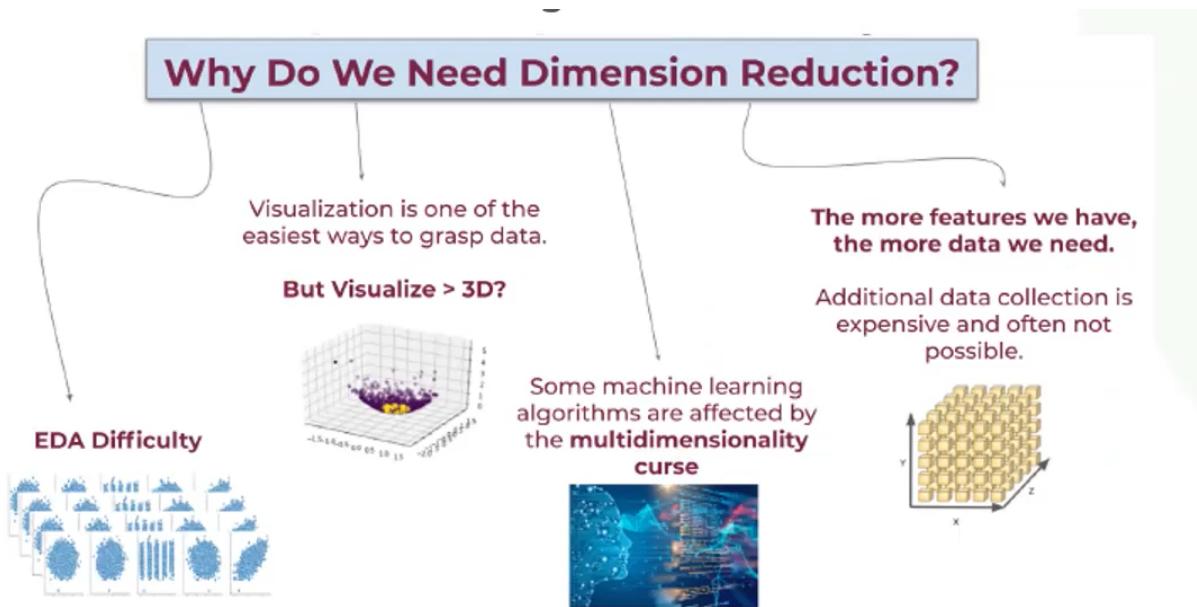
PCA metodunda daha az sayıda bileşen oluştururken **hiçbir feature dışlanmaz**. Ridge gibi davranışır.

## Özniteliklerin Temel Bileşenler İçindeki Önemi



PCA bizim vereceğimiz sayıda yeni bileşen oluşturur ve biz her bileşende her bir fetaure ne kadar temsil ediliyor bunu görebiliriz ama açıkçası önemli değil.

PCA ile multicollinearity ile overfite gitme durumu ortadan kalkar. Aralarında ilişki olan featureslar bir bileşenle ifade edildiğinde bu sorun ortadan kalkmış olur.



PCA için güzel bir örnek :

40kg nardan elde edilen 2kg nar suyunda aslında 40kg nardan elde edilecek tüm vitaminlerin tamamı vardır. Ağırlıklardan yani posadan kurtulmuş oluruz. Ama tüm olayın %90 i açıklanmış olur, biz 2kg lık nar suyundan aynı faydayı sağlamış oluruz. 2kg lık nar suyunun görüntü ve şekil itibariyle 40kg lık nar ile hiçbir alakası olmamakla birlikte onun sağladığı faydayı sağlayabilmektedir.

PCA de elde edilen bileşenlerin nar suyuörneğinde olduğu gibi features larla benzerliği alakası yoktur ama olayın tamamına yakını temsil eder. %80 üzeri temsil durumu varsa başarılıdır.

PCA de scaling olmazsa olmazdır. Features lar aynı birimden değerlerle oluşuyorsa belki yapılmayabilir ama birimler farklıysa olmazsa olmazdır.

Veriyi sıkıştırarak depo alanını azaltmış olur, hesaplama süresi azalır, overfitting riski azalmış olur, daha iyi görselleştirme imkanı sağlar.

Veri kaybı olabilir, features lar arası ilişki yeterince ifade edilemezse sorun olabilir, features lar dan yeni bileşenler olduğu için features ların yorumlanması zorlaşır.

Supervised yada unsupervised modellerde kullanılabilir.

**DENSITY-BASED CLUSTERING** olarak ifade edilen üçüncü yöntem pek kullanılmıyor.