BATCH : **B 150** Data Science
LESSON : **Machine Learning**
DATE : 17.10.2022
SUBJECT : **UNSupervised Learning**

techproeducation
techproeducation
techproeducation
techproeducation
techproedu

techproeducation.com info@techproeducation.com +1 (917) 768-7466

MACHINE LEARNING - 6

Makine Öğrenmesi – 6
Unsupervised Learning

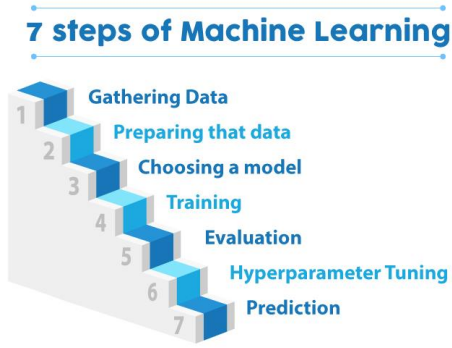
Overall Table of Contents

General Content

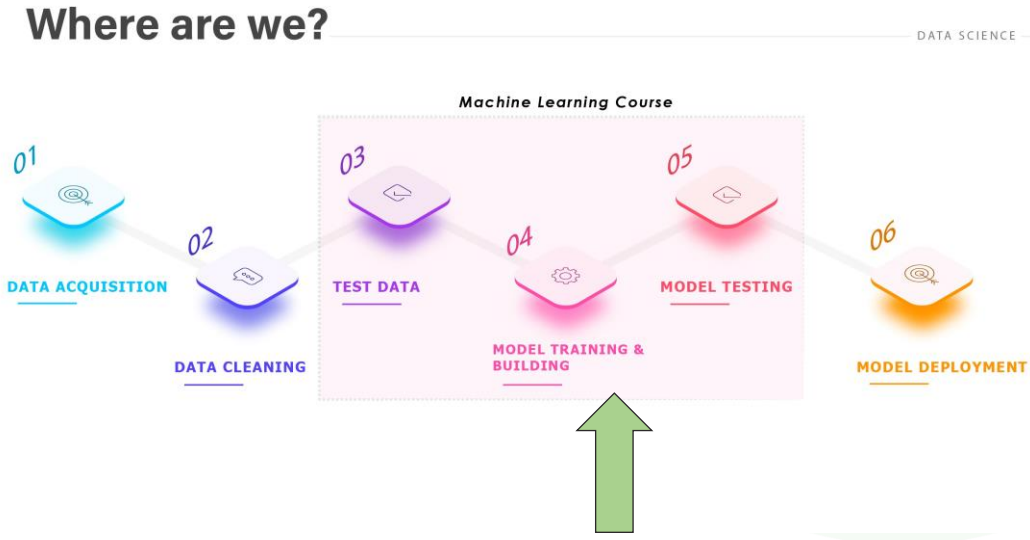
- UNsupervised Learning–

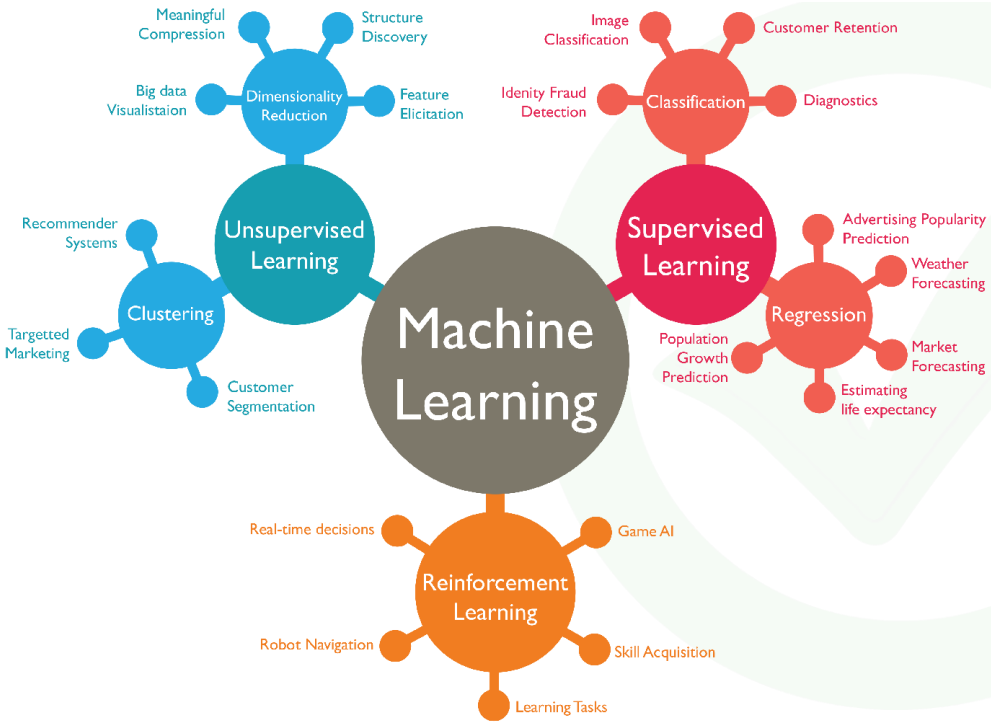
K-Means - PCA

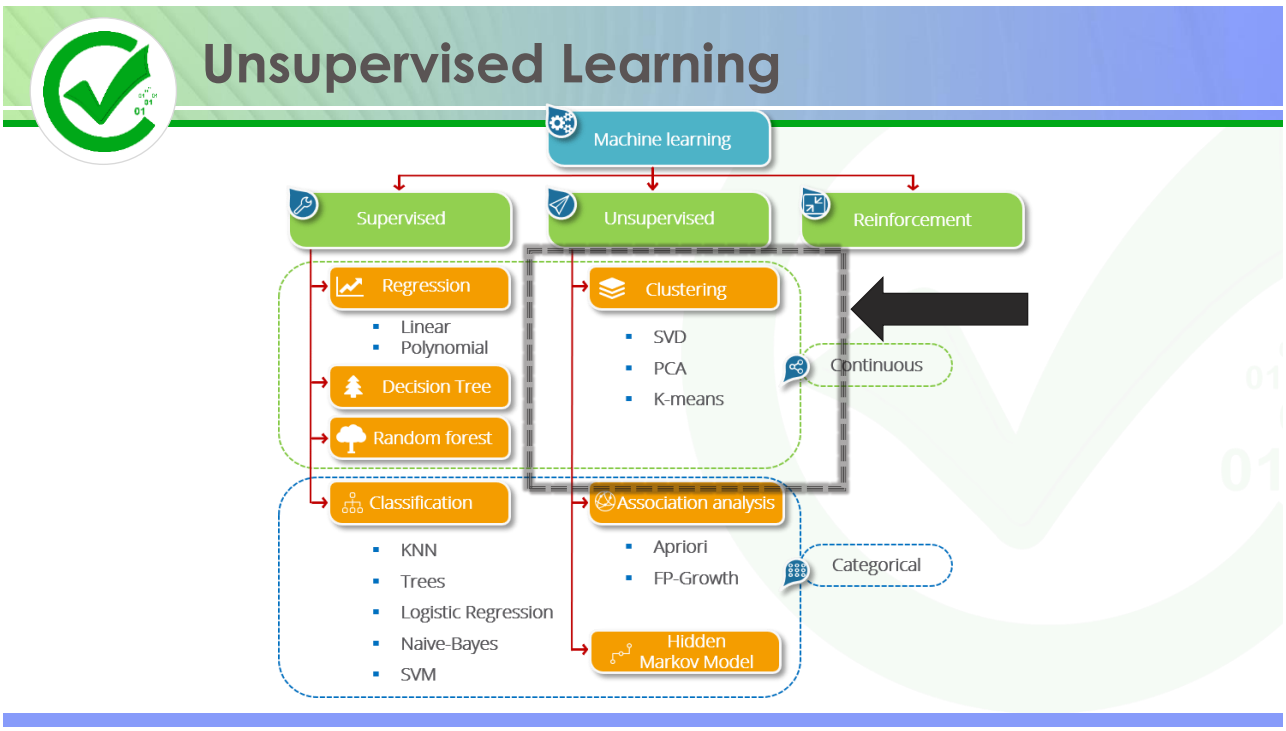
- UNsupervised Algorithm practices Python application
- Projects Solutions



Run Navigation…




Machine Learning
Unsupervised Learning
Meaningful Compression
Structure Discovery
Dimensionality Reduction
Big data Visualistaion
Feature Elicitation
Clustering
Recommender Systems
Targetted Marketing
Customer Segmentation
Supervised Learning
Image Classification
Customer Retention
Classification
Idenity Fraud Detection
Diagnostics
Regression
Advertising Popularity Prediction
Weather Forecasting
Market Forecasting
Estimating life expectancy
Population Growth Prediction
Reinforcement Learning
Real-time decisions
Game AI
Robot Navigation
Skill Acquisition
Learning Tasks

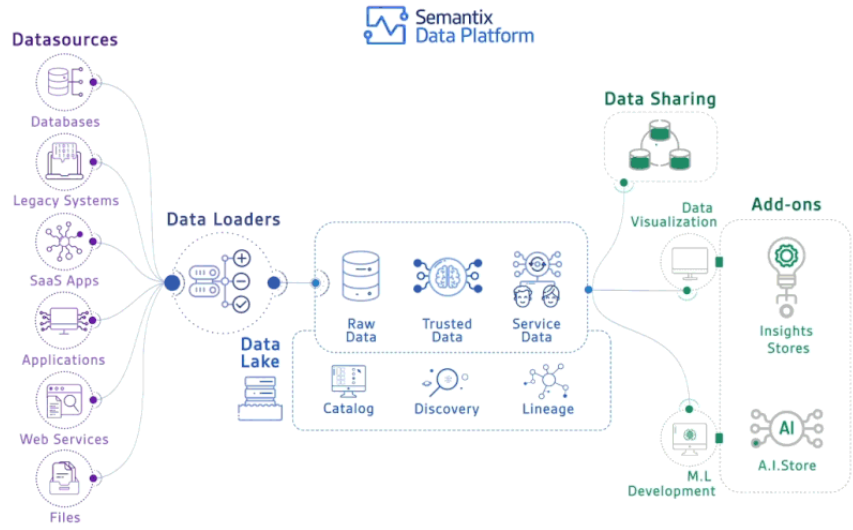

Unsupervised Learning
Machine learning
Supervised
Unsupervised
Reinforcement
Regression
Linear
Polynomial
Decision Tree
Random forest
Clustering
SVD
PCA
K-means
Continuous
Classification
KNN
Trees
Logistic Regression
Naive-Bayes
SVM
Association analysis
Apriori
FP-Growth
Categorical
Hidden Markov Model

Unsupervised Learning

Overall Table of Contents

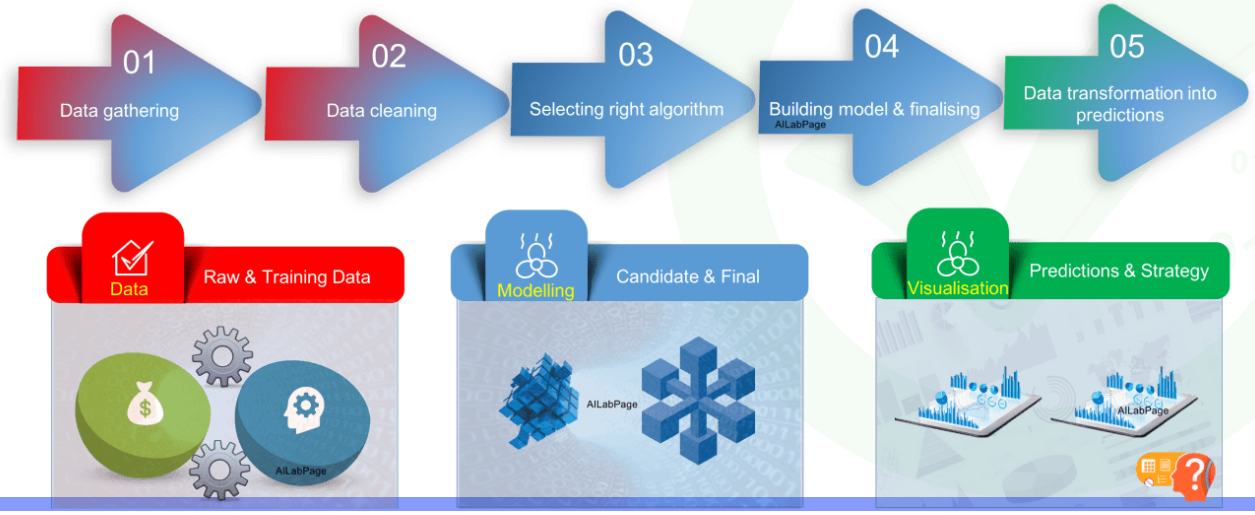
- Unsupervised Learning Algorithm (Denetimsiz Öğrenme)
- Unsupervised Algorithm practices Python application
- Projects Solutions

Unsupervised Learning

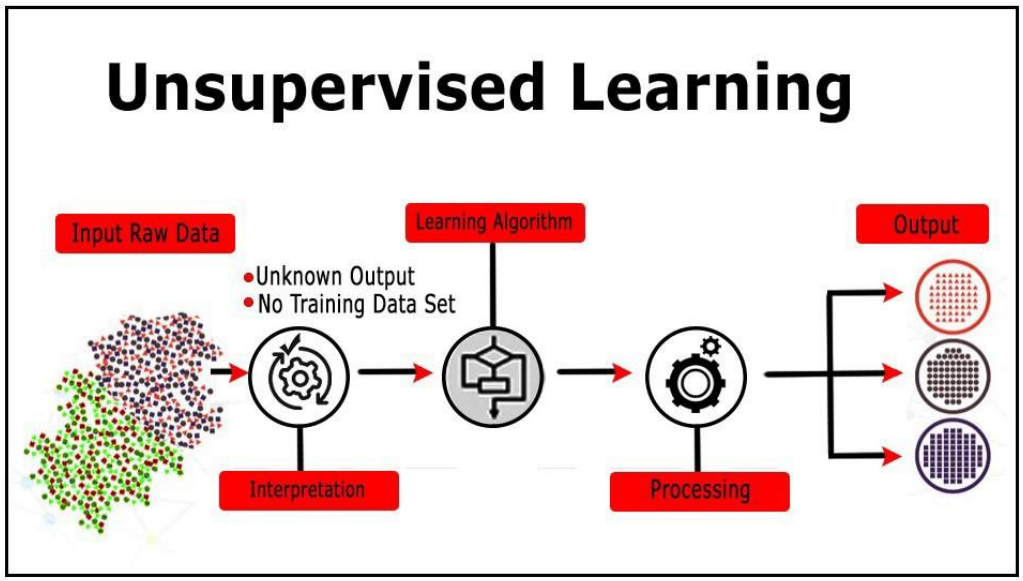


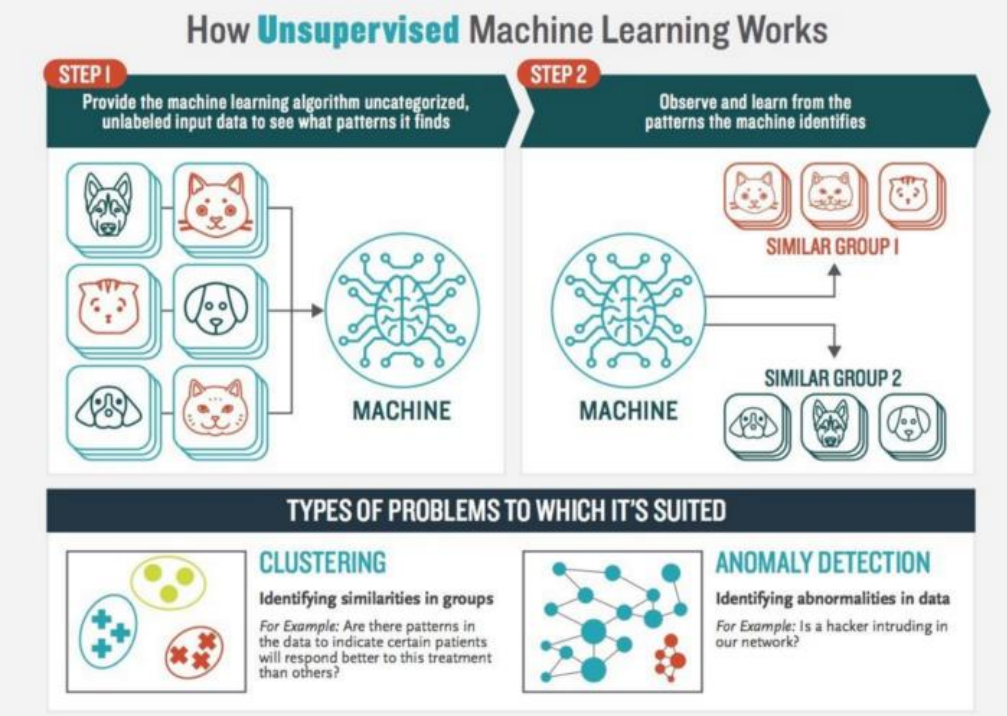
Unsupervised Learning

Machine Learning Process



Unsupervised Learning (Denetimsiz Öğrenme)



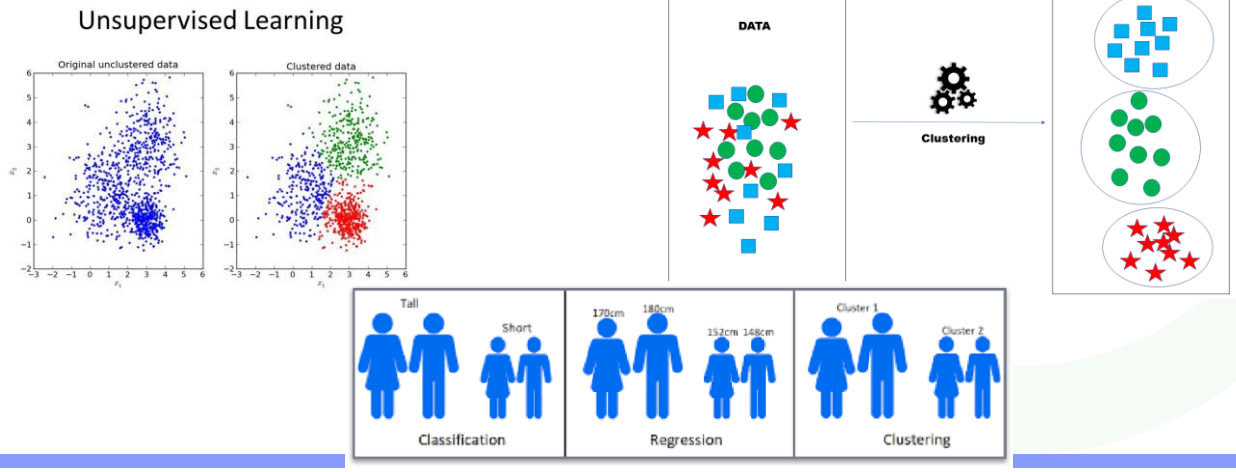


- **Exploratory Data Analysis and Visualization**
- **Machine Learning**
  - **Train | Test Split**
    - X_train, X_test, y_train, y_test = train_test_split()
  - **Scalling (if needed)**
    - scaler = scaler_name()
    - scaler.fit_transform(X_train)
    - scaler.transform(X_test)
  - **Modelling**
    - model = model_name().fit(X_train, y_train)
    - y_pred = model.predict(X_test)
    - y_pred_proba = model.predict_proba(X_test)
  - **Model Performance**
    - Regression => r2_score, MAE, MSE, RMSE
    - Classification => accuracy, recall, precision, f1_score (confusion_matrix, classification_report)
    - Cross Validate => cross_val_score, cross_validate
  - **Tunning (if needed)**
    - grid_param = {}
    - GridsearchCV(grid_param)
  - **Final Model**
    - model = model_name().fit(X, y)
- **Model Deployment**

Unsupervised Learning

CLUSTERING' E GİRİŞ (Kümeleme)

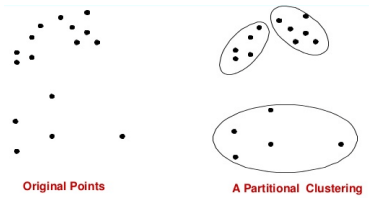
- Meaningfulness - Usefulness

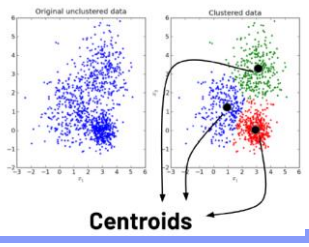


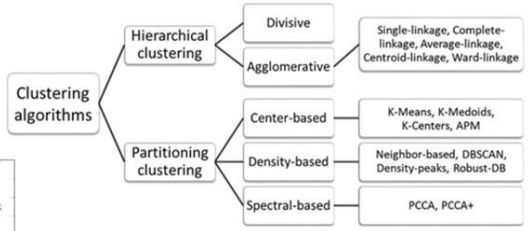
Unsupervised Learning

CLUSTERING' E GİRİŞ (Kümeleme)

- Clustering tipleri
  1. Partitional clustering (K-means)



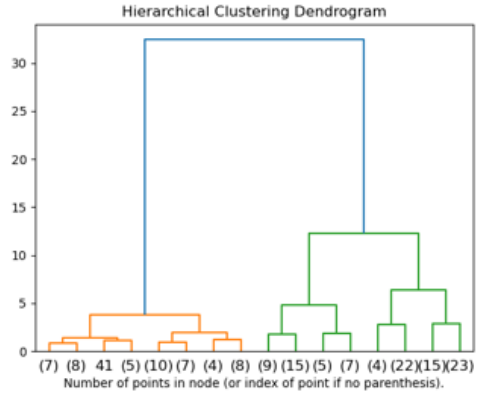


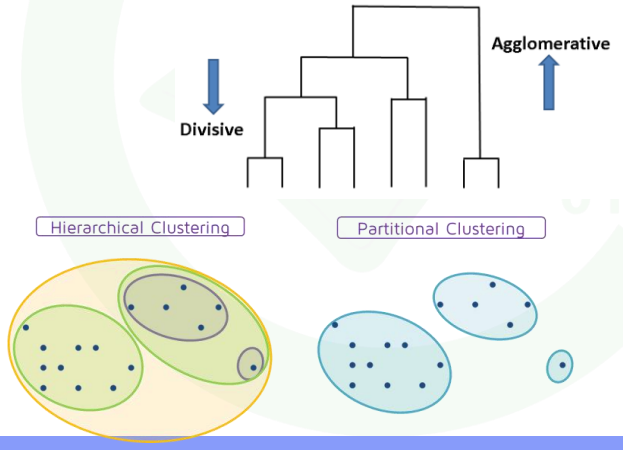


Unsupervised Learning

CLUSTERING' E GİRİŞ (Kümeleme)

- Clustering tipleri
  - 2. Hierarchical clustering (PCA)

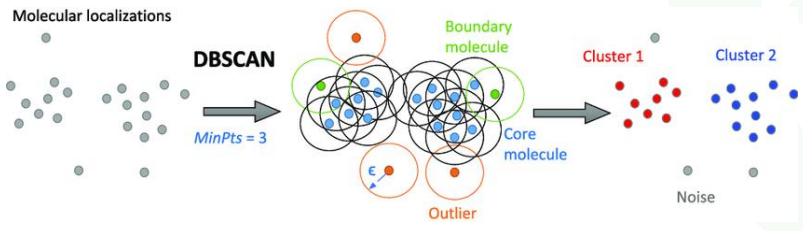


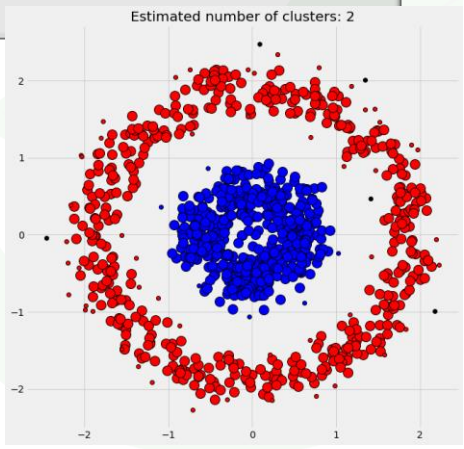


Unsupervised Learning

CLUSTERING' E GİRİŞ (Kümeleme)

- Clustering tipleri
  - 3. Density-based clustering

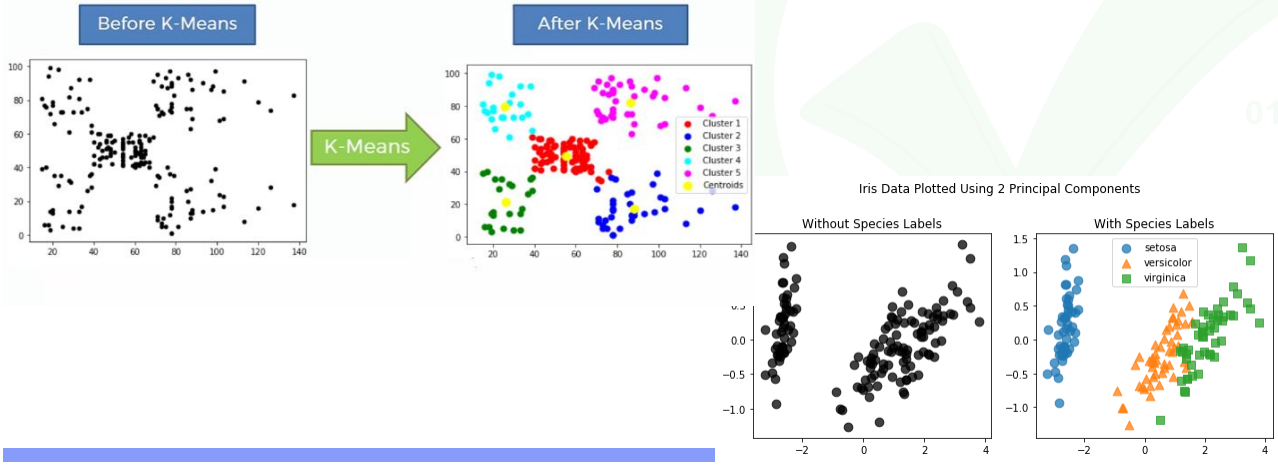




Unsupervised Learning

K-MEANS (K-Ortalamalar) CLUSTERING

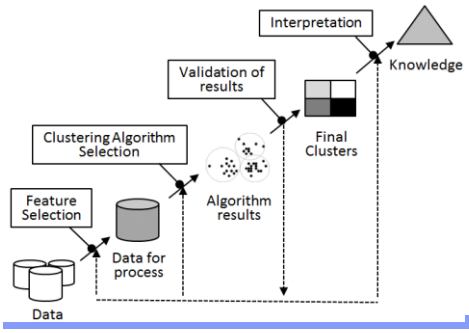
- Hyperparameters

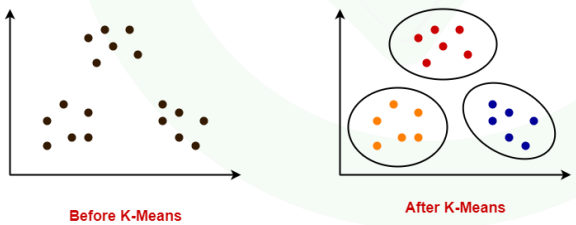


Unsupervised Learning

K-MEANS (K-Ortalamalar) CLUSTERING

- **Cluster Modelleri için PERFORMANS DEGERLENDİRME METRIKLERI**
  - Cluster'lar ne kadar anlamlı
  - Optimum sayıda cluster var mı
  - Cluster ları doğrulamaya ihtiyaç var mı





Unsupervised Learning

K-MEANS (K-Ortalamalar) CLUSTERING

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

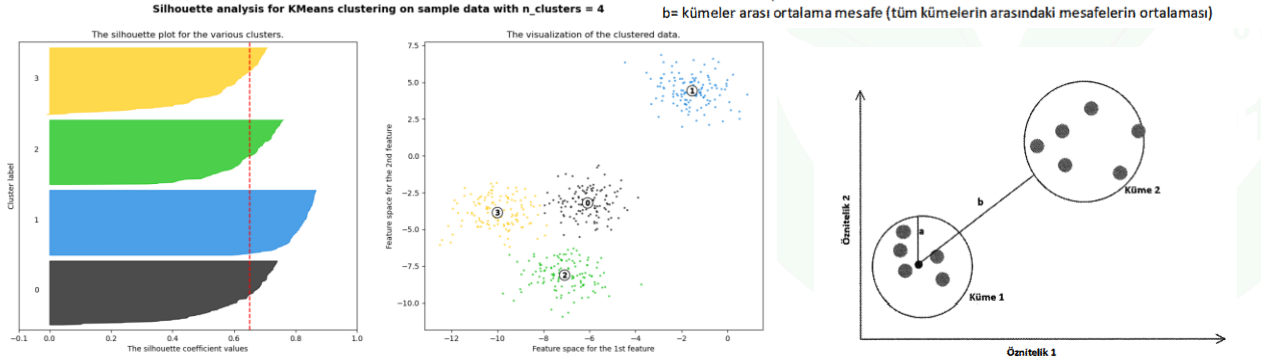
Unsupervised Learning

K-MEANS (K-Ortalamalar) CLUSTERING

- Performans Ölçütleri
  - Silhouette Metodu

$$Silhouette\ Katsayısı = (b - a)/\max(a, b)$$

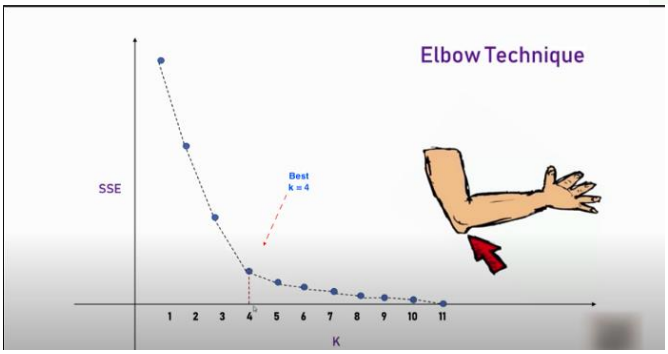
a= küme içi ortalama mesafe (bir kümenin içindeki tüm veri noktalarının birbirleri arasındaki ortalama mesafe)
b= kümeler arası ortalama mesafe (tüm kümelerin arasındaki mesafelerin ortalaması)

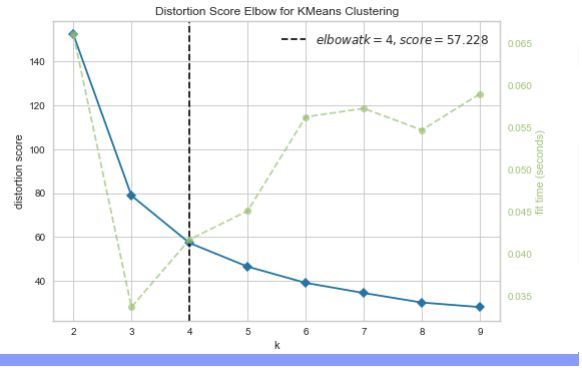


Unsupervised Learning

K-MEANS (K-Ortalamalar) CLUSTERING

- Performans Ölçütleri
- Elbow Metodu

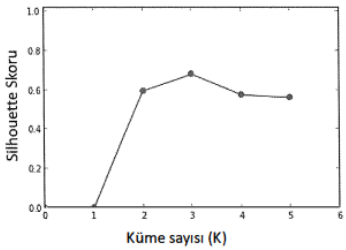




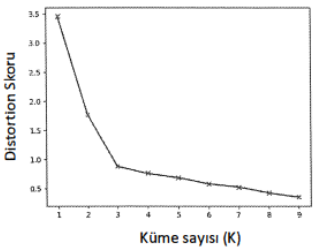
Unsupervised Learning

K-MEANS (K-Ortalamalar) CLUSTERING

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.

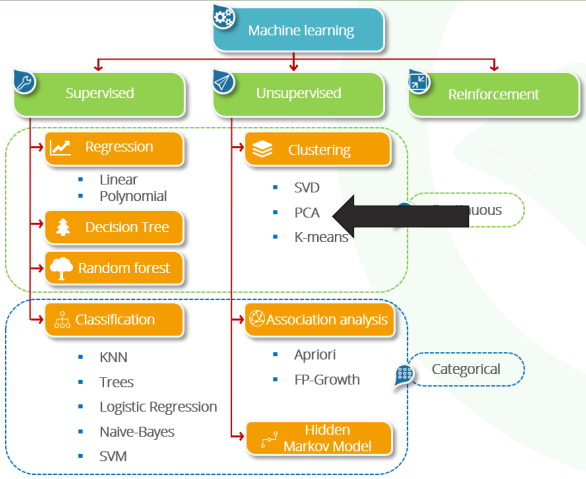


Dirsek Metodunda, optimum küme sayısı (K) grafiğin dirsek şeklinde karıldığı/büküldüğü noktaya karşılık gelen değerdir. Bu örnekte K=3 optimum olarak görülmektedir.

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

Unsupervised Learning

PRINCIPAL COMPONENT ANALYSIS - PCA (Temel Bileşenler Analizi)

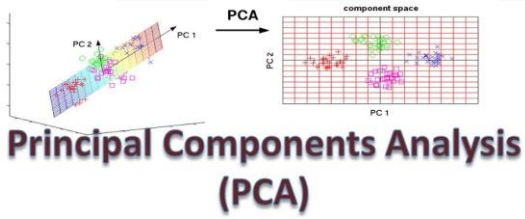


Unsupervised Learning

PRINCIPAL COMPONENT ANALYSIS –PCA

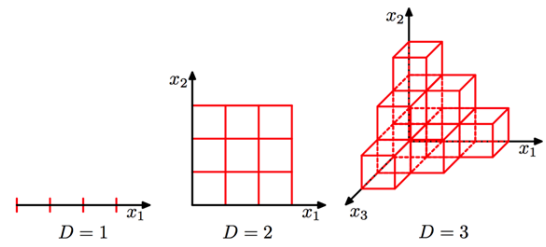
- Feature Selection
- Dimensionality Reduction

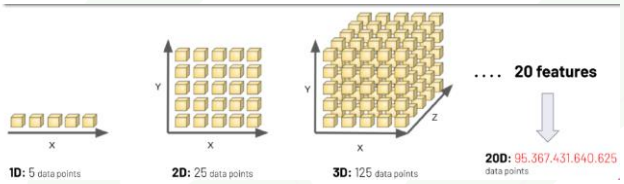
Set A

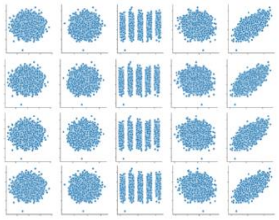


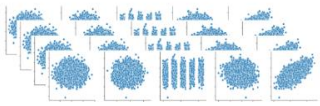
PCA, değişkenlerdeki bilgilerin çoğunu temsil eden daha küçük bir bileşen kümesine indirmeye çalışan bir karmaşıklık azaltma tekniğidir.

PRINCIPAL COMPONENT ANALYSIS –PCA







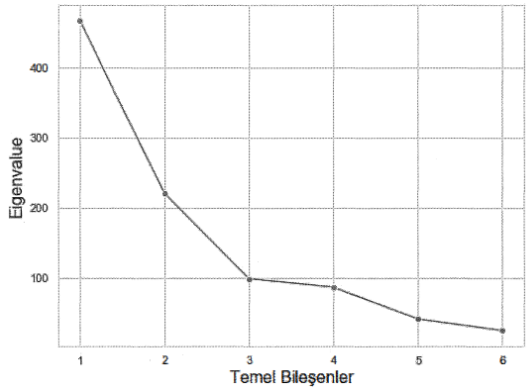


Unsupervised Learning

PRINCIPAL COMPONENT ANALYSIS –PCA

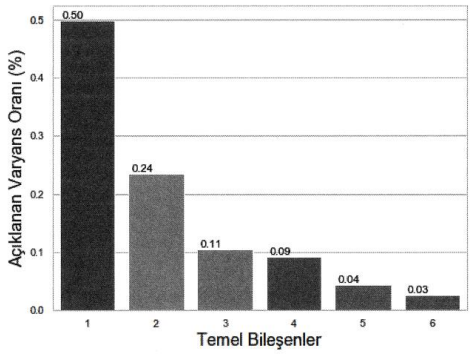
```
87 # PCA modeli için hiperparametrelere değerleri girelim:
88 import prince
89 pca = prince.PCA(n_components=6, n_iter=10, rescale_with_mean=False,
90       rescale_with_std=False, copy=True, check_input=True,
91       engine='sklearn', random_state=42)
92 # PCA modelini pca adı altında oluşturalım:
93 pca = pca.fit(df2[features])
94
```

```
99 # Grafik: Elbow metodyla optimum temel bileşen sayısı
100 ev = pd.DataFrame()
101 ev['pca'] = range(1,7)
102 ev['eigenvalue'] = pd.DataFrame(pca.eigenvalues_)
103 plt.figure(figsize=(8,6))
104 sns.lineplot(x='pca', y='eigenvalue', marker="o", data=ev)
105 plt.ylabel('Eigenvalue', fontsize=16)
106 plt.xlabel('Temel Bileşenler', fontsize=16)
107 plt.show()
108
```



Unsupervised Learning

PRINCIPAL COMPONENT ANALYSIS –PCA

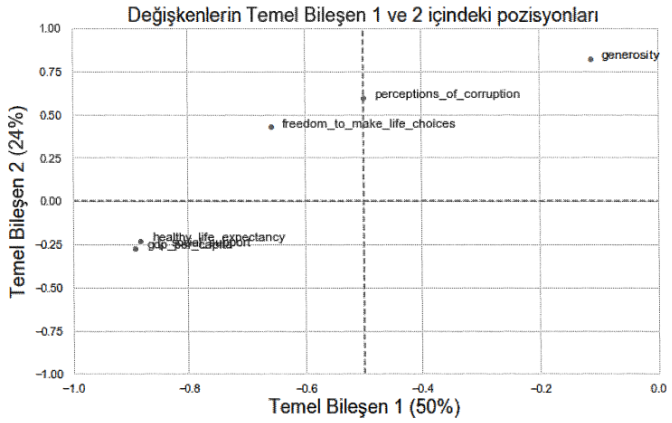



```
# Açıklanan Varyans Oranları:
# Hangi temel bileşen incelene olayı ne kadar açıklıyor?
explained_variance=pca.explained_inertia_
print(explained_variance)

```

```
[0.4984116865967306, 0.2355096143382977,
0.10512215087762719, 0.09197939016712874,
0.0434179988625223, 0.025559159157693058]
```

Unsupervised Learning - PCA

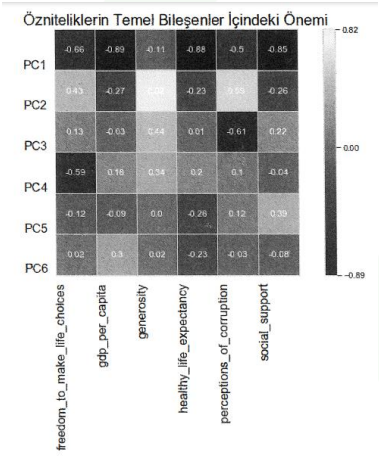
```
# Grafik: Sütun Korelasyonu
pca.column_correlations(df2[features])

scatter = pd.DataFrame(pca.column_correlations(df2[features])).reset_index()
plt.figure(figsize=(10,6))
ax = sns.scatterplot(x=0, y=1, data=scatter)
ax.set(ylim=(-1, 1), xlim=(-1,0))
def label_point(x, y, val, ax):
    a = pd.concat({'x': x, 'y': y, 'val': val}, axis=1)
    for i, point in a.iterrows():
        ax.text(point['x']+.02, point['y'], str(point['val']))
label_point(scatter[0], scatter[1], scatter['index'], plt.gca())
plt.axvline(-0.5, ls='--')
plt.axhline(0, ls='--')
plt.title('Değişkenlerin Temel Bileşen 1 ve 2 içindeki pozisyonları',
          fontsize=18)
plt.xlabel('Temel Bileşen 1 (50%)', fontsize=18)
plt.ylabel('Temel Bileşen 2 (24%)', fontsize=18)
plt.show()

```



Unsupervised Learning - PCA

```
# Öznitliklerin herbir temel bileşen içinde açıklanan varyansa yaptığı
# katkıyı (Eigenvectors) tabLolaştırarak inceleyelim:
t=pca.column_correlations(df2[features]).transpose()
features_names=pd.Series(t.columns)
s= pd.Series(["PC1", "PC2", "PC3","PC4", "PC5", "PC6"])
t=t.set_index(s)

# Grafiğe hazır hale getirmek için t dataframeini numpay array fromatına
# dönüştürelim ve 2 basamaklı ondalık değerine yuvarlayalım:
t=t.to_numpy()
t=t.round(2)
```

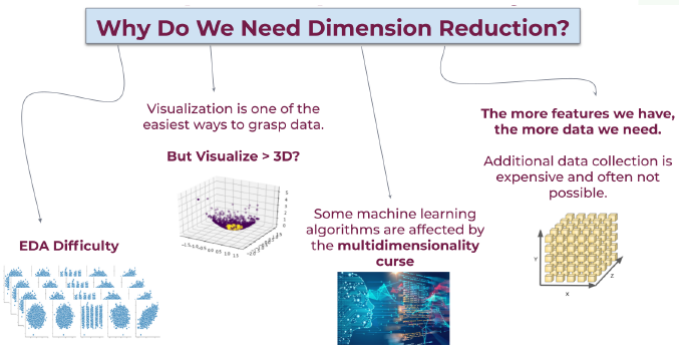
```
# Graf: Özniteliklerin Temel Bileşenlerdeki Önemi:
fig, ax = plt.subplots(figsize=(12, 6))
plt.imshow(t, interpolation = 'none', cmap =  'plasma')
feature_names = features_names
# Temel bileşenler ve özniteliklerin grafikteki konumları...
ax.set_xticks(np.arange(-.5, len(feature_names)));
ax.set_yticks(np.arange(0.5, 6));
ax.set_xticklabels(feature_names, rotation=90, ha='left', fontsize=16);
ax.set_yticklabels(s, va='bottom', fontsize=16);
# Kutucuklarda egeanvalues değerlerini göstermek için...
for i in range(len(s)):
    for j in range(len(feature_names)):
        text = ax.text(j, i, t[i, j],
                       ha="center", va="center", color="w")

plt.title('Özniteliklerin Temel Bileşenler İçindeki Önemi', fontsize=20)
plt.colorbar(orientation='vertical', ticks=[t.min(), 0,t.max()]);
```

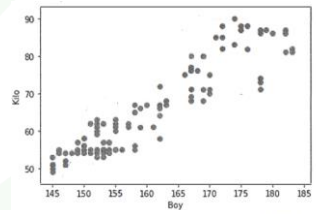


Unsupervised Learning

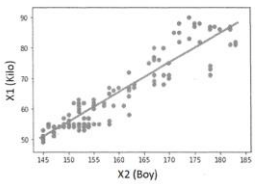
DIMENSION REDUCTION (Boyut Azaltma)

Giriş





Boyut Azaltma



Burada boy (X1) ve kilo (X2) bir ilk okul sınıfındaki öğrencilerin cinsiyetini tanımlayan iki öznitelik olsun. Her iki değişken de bir ölçüde tek başlarına öğrencilerin cinsiyetlerini tanımladığını varsayalım (ilkokul çağında erkekler kızlardan daha ağır; daha önce boy atıyor). Bu değişkenlerin her ikisini de kullanırsak, benzer bilgileri aktarırlar. Bu nedenle, yalnızca bir değişken kullanmak mantıklı olacaktır. Verileri 2D'den (X1 ve X2) 1D (Y1) 'e aşağıda gösterildiği gibi dönüştürebiliriz. Benzer şekilde, verilerin d sayıdaki boyutunu aynı bilgiyi içermek kaydıyla bir alt kümesine indirgeyebiliriz. Buna **boyut azaltma** denir.

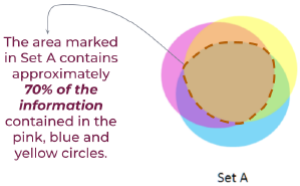
Unsupervised Learning

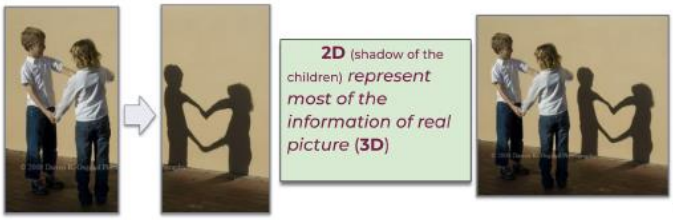
- DIMENSION REDUCTION (Boyut Azaltma)
  - Feature Selection
  - Feature Extraction

Unsupervised Learning

- DIMENSION REDUCTION (Boyut Azaltma)
  - PCA ile Dimension Reduction

- 2 human
- children
- long hair
- put their hands together
- height
- etc.



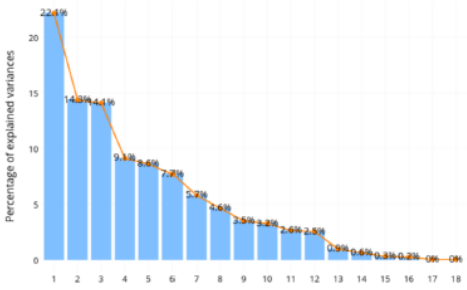


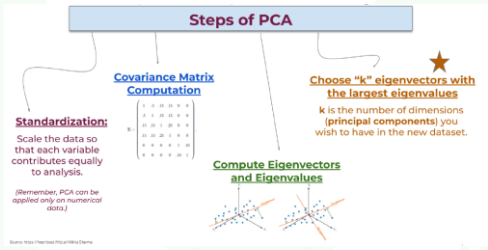
2D (shadow of the children) *represent most of the information of real picture* (**3D**)

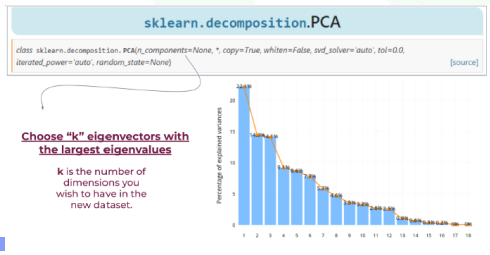
Unsupervised Learning

DIMENSION REDUCTION (Boyut Azaltma)

- PCA ile Dimension Reduction







Unsupervised Learning

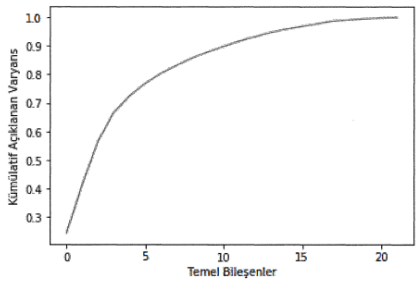
DIMENSION REDUCTION (Boyut Azaltma)

- PCA ile Dimension Reduction

```
# KNN Sınıflandırıcısı:
n_neighbors=3 # en yakın komşu sayısını 3 seçelim.
KNN = neighbors.KNeighborsClassifier(n_neighbors, weights='distance')
KNN.fit(X_train, y_train)

# Eğitim Seti performansı
pred_invest_grade_train =KNN.predict(X_train)
print('Doğruluk (Accuracy):%0.2f'% accuracy_score(y_train, pred_invest_grade_train))
# Test Seti Performansı
pred_invest_grade_test =KNN.predict(X_test)
print('Doğruluk (Accuracy):%0.2f'% accuracy_score(y_test, pred_invest_grade_test))

Doğruluk (Accuracy):1.00
Doğruluk (Accuracy):0.92
```

```
# top. varyansın %80'ini açıklayan temel bileşenlere
# göre modeli tekrar oluşturalım:
pca = PCA(n_components = 0.80)
X_pca = pca.fit_transform(X_train) # eğitim setini dönüştürüp boyutunu azaltır.
print(pca.n_components_) #  top. varyansın %80'i kaç temel bileşen ile açıklanıyor?
```

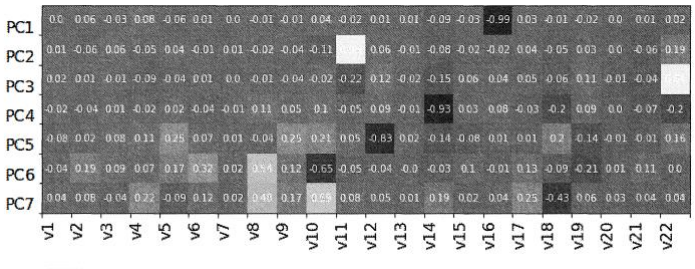
```
# sklearn kütüphanesi:PCA ile Boyut Azaltma
df.iloc[:, 1:23].columns # öznitelikler

pca = PCA()
X_pca = pca.fit(X_train)

# Grafik: Kümülatif Açıklanan Varyans (KAV)
plt.plot(np.cumsum(pca.explained_variance_ratio_))
plt.xlabel('Temel Bileşenler')
plt.ylabel('Kümülatif Açıklanan Varyans');
```



Unsupervised Learning

DIMENSION REDUCTION (Boyut Azaltma)

- PCA ile Dimension Reduction
  - Avantajları
  - Dezavantajları

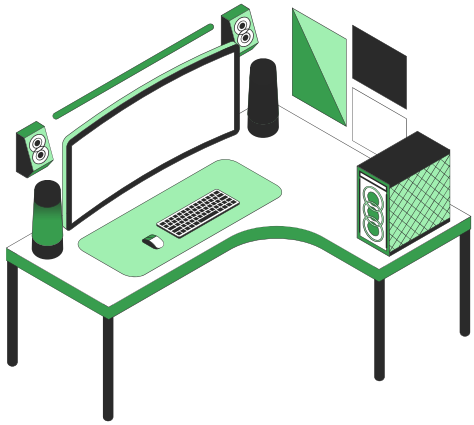



```
104 # Eğitim Seti Performansı
105 pred_invest_grade_pca_train =clf.predict(PC_Df)
106 print('Doğruluk (Accuracy):%0.2f'%
107       accuracy_score(y_train,pred_invest_grade_pca_train))
108 # Test Seti Performansı
109 pred_invest_grade_pca_train =clf.predict(PC_Df)
110 print('Doğruluk (Accuracy):%0.2f'%
111       accuracy_score(y_train,pred_invest_grade_pca_train))
112
```

```
Doğruluk (Accuracy):1.00
Doğruluk (Accuracy):1.00
```

Bu dersi anladım..

Everything is clear ?



TECHPROED

Do you have any questions?

Send it to us! We hope you learned something new.