



BURDUR MEHMET AKİF ERSOY ÜNİVERSİTESİ

Gölhisar Uygulamalı Bilimler Yüksekokulu

PROJE DERSİ

PROJE KONUSU: Destek Vektör Makinaları (Support Vector Machines)

Öğrenci Ad-Soyad/Numara

1- Erencan YILMAZ / 2112903024

MAYIS 2024
BURDUR

- **GİRİŞ**

Projede kullanılan veri kümesi, Wisconsin Meme Kanseri (Breast Cancer Wisconsin) veri setidir.

Meme kanseri teşhisi için gerekli olan 30 farklı özellik ve 569 örnek içermektedir. Her örnek, bir tümör hücresine ait olup, özellikler tümör hücrelerinin çeşitli ölçümlerini temsil etmektedir.

Veri seti, iki sınıfa ayrılmıştır: malign (kötü huylu) ve benign (iyi huylu) olmak üzere ayrılmıştır. benign (iyi huylu) yani 1 , malign (kötü huylu) yani 0'ı temsil eder.

Projede kullanılan sınıflandırma yöntemi, Support Vector Machines (SVM) olarak adlandırılan bir makine öğrenmesi algoritmasıdır.

SVM, veri noktalarını sınıflandırmak için en iyi ayırım çizgisini veya hiper düzlemi bulmaya çalışır. Bu yöntem, özellikle yüksek boyutlu veri setlerinde etkili bir şekilde çalışır ve genellikle sınıflandırma problemlerinde yüksek doğruluk sağlar.

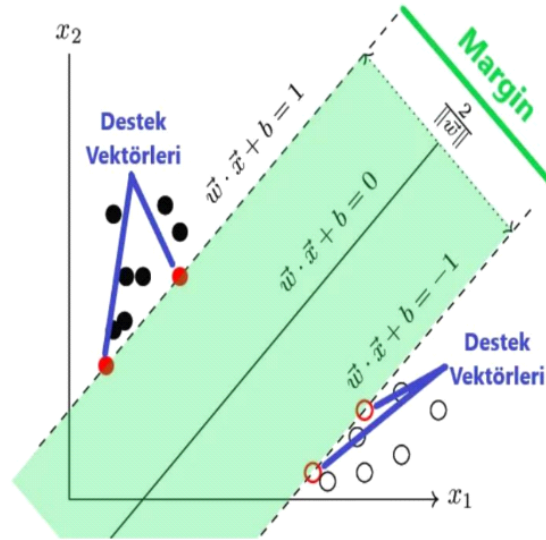
- Avantajlar
 - SVM, lineer ve non-lineer ayrımlar için kullanılabilir.
 - SVM, yüksek boyutlu özellik uzaylarında iyi performans gösterir.
 - SVM, çok sayıda özellikte çalışabilir ve veri setindeki aykırı değerlere ve gürültülere dayanıklıdır.
 - SVM, sınıflandırma problemlerinde diğer modellere kıyasla daha az hiperparametreye sahiptir.
- Dezavantajlar
 - SVM, büyük veri setlerinde yüksek hesaplama maliyetine sahip olabilir.
 - Hiperparametre ayarlaması (örneğin, C ve gamma) genellikle deneme yanılma yoluyla belirlenir ve doğru ayarlamak zaman alabilir.
 - SVM'nin sonuçlarının yorumlanması ve anlaşılması diğer modellere kıyasla daha zordur.

• GEREÇ VE YÖNTEM

Support Vector Machines (SVM), Vladimir Vapnik ve Alexey Chervonenkis tarafından 1960'ların sonlarında ve 1990'ların başlarında geliştirilen bir denetimli öğrenme algoritmasıdır. SVM ilk olarak 1992 de tanıtılmıştır. SVM, iki sınıf arasındaki en iyi ayırım çizgisini bulmak için optimize edilmiş bir yöntemdir. Bu yöntem, özellikle yüksek boyutlu veri setlerinde ve küçük örneklem büyüklüklerinde oldukça etkilidir. SVM, biyoinformatik, görüntü işleme ve doğal dil işleme gibi birçok alanda yaygın olarak kullanılmaktadır.

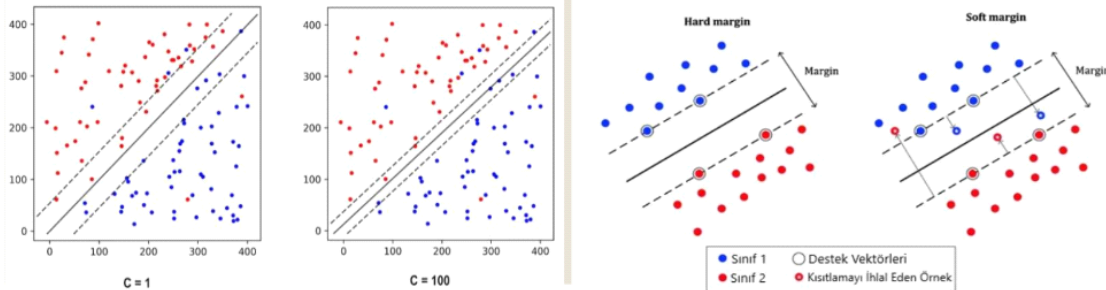
SVM

- Siyahlar ve beyazlar olmak üzere iki farklı sınıf var. Sınıflandırma problemlerindeki asıl amacımız gelecek verinin hangi sınıfta yer alacağını karar vermektir.
- Bu sınıflandırmayı yapabilmek için iki sınıfı ayıran bir doğru çizilir ve bu doğrunun ± 1 'i arasında kalan yeşil bölgeye **Margin** adı verilir. Margin ne kadar geniş ise iki veya daha fazla sınıf o kadar iyi ayrıştırılır.



SVM->Hard Margin vs Soft Margin

- Marginimiz her zaman bu şekilde olmayabilir. Bazen örneklerimiz Margin bölgesine girebilir. Buna **Soft Margin** denir.
- **Hard Margin**, verimiz doğrusal olarak ayrılabilirse çalışır ve aykırı değerlere karşı çok duyarlıdır. Bu yüzden bazı durumlarda Soft Margin'i tercih etmemiz gerekebilir.
- İkisi arasındaki dengeyi SVM içerisindeki C hiperparametresi ile kontrol edebiliriz. C ne kadar büyükse Margin o kadar dardır. Ayrıca model overfit olursa C'yi azaltmamız gerekir.

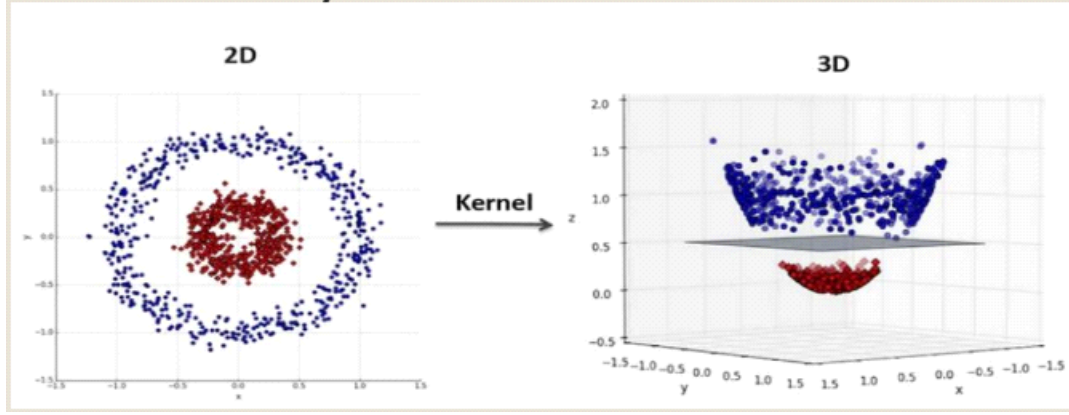


Polynomial Kernel

Bu yöntemde problemimizi çözmek için 2 boyuttan çıkıp 3 veya daha fazlası boyutta işlem yapıyormuş gibi hareket ediyoruz.

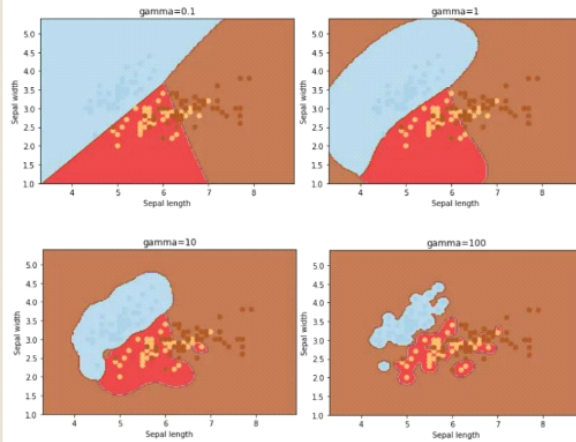
Soldaki (2 boyut) dağılımı bir doğru ile sınıflandıramayız. Bunun için bu gibi problemlerde Polynomial Kernel'i kullanabiliriz. 3. boyutta işlem yaparken sınıflara ayırmak için doğru yerine bir düzlem kullanılırız ve çok daha düzgün bir şekilde sınıflandırabiliriz.

Modelimiz overfit olmuşsa derecesini düşürmeniz, underfit olmuşsa derecesini yükseltmeniz gerekir. Ayrıca coef0 hiperparametresiyle modelinizin yüksek dereceli denklemlerden ne kadar etkileneceğini ayarlayabilirsiniz (sadece 'poly' ve 'sigmoid' kernelda etkili olur).



SVM-> Gaussian RBF (Radial Basis Function) Kernel

- Sonsuz boyuttaki Destek Vektör Makinelerini bulur ve her bir noktanın belirli bir noktaya ne kadar benzediğini normal dağılım ile hesaplar, ona göre sınıflandırır.
- Dağılımın genişliğini gamma hiperparametresi ile kontrol ederiz. Gamma ne kadar küçükse dağılım o kadar geniş olur.
- Model overfit olmuşsa gamma değerini düşürmemiz, model underfit olmuşsa gamma değerini yükseltmemiz gerekir.
- Veri setiniz aşırı büyük değilse genellikle RBF Kernel tercih edilir.



• BULGULAR

`pip install ucimlrepo`

- ucimlrepo paketini yüklemek için kullanılan kod bloğudur.

`from ucimlrepo import fetch_ucirepo`

```
breast_cancer_wisconsin_original = fetch_ucirepo(id=15)
X = breast_cancer_wisconsin_original.data.features
y = breast_cancer_wisconsin_original.data.targets
print(breast_cancer_wisconsin_original.metadata)
print(breast_cancer_wisconsin_original.variables)
```

-ucimlrepo paketini kullanarak Wisconsin Meme Kanseri veri setini indiriyor ve özellikleri bir DataFrame'e yüklüyor.

```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline
```

-Gerekli kütüphaneleri yüklüyor ve matplotlib ile grafiklerin not defteri içinde gösterilmesi için %matplotlib inline komutunu kullanıyor.

```
from sklearn.datasets import load_breast_cancer
```

-Scikit-learn kütüphanesinden meme kanseri veri setini yüklemek için gerekli fonksiyonu import ediyor.

```
cancer = load_breast_cancer()
```

-Meme kanseri veri setini yüklüyor ve cancer değişkenine atıyor.

```
cancer.keys()
```

-Veri setinin anahtarlarını listeliyor.

-dict_keys(['data', 'target', 'frame', 'target_names', 'DESCR', 'feature_names', 'filename', 'data_module'])

```
print(cancer['DESCR'])
```

-Veri setinin açıklamasını yazdırıyor.

.. _breast_cancer_dataset:

Meme kanseri wisconsin (tanısal) veri kümesi

****Veri Seti Özellikleri:****

: Örnek Sayısı: 569

:Öznitelik Sayısı: 30 sayısal, tahmine dayalı öznitelik ve sınıf

:Özellik Bilgileri:

- yarıçap (merkezden çevredeki noktalara olan mesafelerin ortalaması)
- doku (gri tonlamalı değerlerin standart sapması)
- çevre
- alan
- pürüzsüzlük (yarıçap uzunluklarında yerel değişiklik)
- kompaktlık ($\text{çevre}^2 / \text{alan} - 1,0$)
- içbükeylik (konturun içbükey kısımlarının ciddiyeti)
- içbükey noktalar (konturun içbükey kısımlarının sayısı)
- simetri
- fraktal boyut ("kıyı şeridi yaklaşımı" - 1)

Ortalama, standart hata ve "en kötü" veya en büyük (üçünün ortalaması) bu özelliklerin en kötü/en büyük değerleri her görüntü için hesaplandı, 30 özellik ortaya çıkıyor. Örneğin, 0 alanı Ortalama Yarıçaptır, alan 10 Yarıçap SE'dir, alan 20 En Kötü Yarıçaptır.

- sınıf:

- WDBC- Kötü huylu
- WDBC- İyi huylu

:Özet İstatistikler:

=====

En az en çok

=====

yarıçap (ortalama):	6,981 28,11
doku (ortalama):	9,71 39,28
çevre (ortalama):	43,79 188,5
alan (ortalama):	143,5 2501,0
pürüzsüzlük (ortalama):	0,053 0,163
kompaktlık (ortalama):	0,019 0,345
içbükeylik (ortalama):	0,0 0,427
içbükey noktalar (ortalama):	0,0 0,201
simetri (ortalama):	0,106 0,304
fraktal boyut (ortalama):	0,05 0,097
yarıçap (standart hata):	0,112 2,873
doku (standart hata):	0,36 4,885
çevre (standart hata):	0,757 21,98
alan (standart hata):	6,802 542,2
düzgünlük (standart hata):	0,002 0,031
kompaktlık (standart hata):	0,002 0,135
içbükeylik (standart hata):	0,0 0,396
içbükey noktalar (standart hata):	0,0 0,053
simetri (standart hata):	0,008 0,079

fraktal boyut (standart hata):	0,001 0,03
yarıçap (en kötü):	7,93 36,04
doku (en kötü):	12,02 49,54
çevre (en kötü):	50,41 251,2
alan (en kötü):	185,2 4254,0
pürüzsüzlük (en kötü):	0,071 0,223
kompaktlık (en kötü):	0,027 1,058
içbükeylik (en kötü):	0,0 1,252
içbükey noktalar (en kötü):	0,0 0,291
simetri (en kötü):	0,156 0,664
fraktal boyut (en kötü):	0,055 0,208

=====

:Eksik Öznitelik Değerleri: Yok

:Sınıf Dağılımı: 212 - Malign(kötü huylu) 357 - Benign(iyi huylu)

:Yaratıcı: Dr. William H. Wolberg, W. Nick Street, Olvi L. Mangasarian

:Bağışçı: Nick Street

:Tarih: Kasım 1995

Bu, UCI ML Meme Kanseri Wisconsin (Tanısal) veri kümelerinin bir kopyasıdır.
<https://goo.gl/U2Uwz2>

Özellikler ince bir iğnenin sayısallaştırılmış görüntüsünden hesaplanır
Bir meme kitlesinin aspirasyonu (FNA). Onlar tanımlar
görüntüde bulunan hücre çekirdeklerinin özellikleri.

Yukarıda açıklanan ayırma düzlemi kullanılarak elde edildi
Çok Yüzeyle Yöntem Ağacı (MSM-T) [K. P. Bennett, "Karar Ağacı
Doğrusal Programlama Yoluyla İnşaat." 4. Bildiriler Kitabı
Midwest Yapay Zeka ve Bilişsel Bilim Topluluğu,
s. 97-101, 1992], doğrusal kullanan bir sınıflandırma yöntemi
Bir karar ağacı oluşturmak için programlama. İlgili özellikler
1-4 aralığında kapsamlı bir arama kullanılarak seçildi
özellikler ve 1-3 ayırma düzlemi.

Ayıran düzlemi elde etmek için kullanılan gerçek doğrusal program
3 boyutlu uzayda şu şekilde tanımlanır:
[K. P. Bennett ve O. L. Mangasarian: "Sağlam Doğrusal
Doğrusal Olarak Ayrılamayan İki Kümenin Programlanması Ayrımcılığı",
Optimizasyon Yöntemleri ve Yazılım 1, 1992, 23-34].

Bu veritabanına UW CS ftp sunucusu aracılığıyla da erişilebilir:

ftp ftp.cs.wisc.edu
cd matematik-prog/cpo-veri kümesi/makine-öğrenme/WDBC/

.. konu:: Referanslar

- W.N. Caddesi, W.H. Wolberg ve O.L. Mangasaryalı. Nükleer özellik çıkarma Meme tümörü tanısı için. IS&T/SPIE 1993 Uluslararası Sempozyumu Elektronik Görüntüleme: Bilim ve Teknoloji, cilt 1905, sayfalar 861-870, San Jose, Kaliforniya, 1993.
- O.L. Mangasarian, W.N. Street ve W.H. Wolberg. Meme kanseri tanısı ve Doğrusal programlama yoluyla prognoz. Yöneylem Araştırması, 43(4), sayfa 570-577, Temmuz-Ağustos 1995.
- W.H. Wolberg, W.N. Street ve O.L. Mangasaryalı. Makine öğrenimi teknikleri ince iğne aspirasyonu ile meme kanserini teşhis etmek

```
df_feat = pd.DataFrame(cancer['data'],columns=cancer['feature_names'])
```

-Veri setinin özelliklerini bir DataFrame'e dönüştürüyor.

```
df_feat.head()
```

-Veri setinin ilk beş satırını görüntülüyor.

	mean radius	mean texture	mean perimeter	mean area	mean smoothness	mean compactness	mean concavity	mean concave points	mean symmetry	mean fractal dimension	...	worst radius	worst texture	worst perimeter	worst area	worst smoothness	worst compactness	worst concavity	worst concave points	worst symmetry	worst fractal dimension
0	17.99	10.38	122.80	1001.0	0.11840	0.27760	0.3001	0.14710	0.2419	0.07871	...	25.38	17.33	184.60	2019.0	0.1622	0.6656	0.7119	0.2654	0.4601	0.11890
1	20.57	17.77	132.90	1326.0	0.08474	0.07864	0.0869	0.07017	0.1812	0.05667	...	24.99	23.41	158.80	1956.0	0.1238	0.1866	0.2416	0.1860	0.2750	0.08902
2	19.69	21.25	130.00	1203.0	0.10960	0.15990	0.1974	0.12790	0.2069	0.05999	...	23.57	25.53	152.50	1709.0	0.1444	0.4245	0.4504	0.2430	0.3613	0.08758
3	11.42	20.38	77.58	386.1	0.14250	0.28390	0.2414	0.10520	0.2597	0.09744	...	14.91	26.50	98.87	567.7	0.2098	0.8663	0.6869	0.2575	0.6638	0.17300
4	20.29	14.34	135.10	1297.0	0.10030	0.13280	0.1980	0.10430	0.1809	0.05883	...	22.54	16.67	152.20	1575.0	0.1374	0.2050	0.4000	0.1625	0.2364	0.07678

5 row x 21 columns

```
df_feat.info()
```

-DataFrame'in yapısı hakkında bilgi veriyor.

```
<class 'pandas.core.frame.DataFrame'>
```

RangeIndex: 569 girişi, 0 ila 568

Veri sütunları (toplam 30 sütun):

# Sütun	Boş Olmayan	Sayım Türü
0 ortalama yarıçap	569 boş olmayan	float64
1 ortalama doku	569 boş olmayan	float64
2 ortalama çevre	569 boş olmayan	float64
3 ortalama alan	569 boş olmayan	float64
4 ortalama pürüzsüzlük	569 sıfır olmayan	float64

5 ortalama kompaktlık	569 boş olmayan	float64
6 ortalama içbükeylik	569 sıfır olmayan	float64
7 ortalama içbükey nokta	569 boş olmayan	float64
8 ortalama simetri	569 sıfır olmayan	float64
9 ortalama fraktal boyut	569 sıfır olmayan	float64
10 yarıçap hatası	569 boş olmayan	float64
11 doku hatası	569 boş olmayan	float64
12 çevre hatası	569 boş olmayan	float64
13 alan hatası	569 boş olmayan	float64
14 düzgünlük hatası	569 boş olmayan	float64
15 kompaktlık hatası	569 boş olmayan	float64
16 içbükeylik hatası	569 boş olmayan	float64
17 içbükey nokta hatası	569 boş olmayan	float64
18 simetri hatası	569 boş olmayan	float64
19 fraktal boyut hatası	569 boş olmayan	float64
20 en kötü yarıçap	569 boş olmayan	float64
21 en kötü doku	569 boş olmayan	float64
22 en kötü çevre	569 boş olmayan	float64
23 en kötü alan	569 boş olmayan	float64
24 en kötü düzgünlük	569 boş olmayan	float64
25 en kötü kompaktlık	569 boş olmayan	float64
26 en kötü içbükeylik	569 sıfır olmayan	float64
27 en kötü içbükey nokta	569 boş olmayan	float64
28 en kötü simetri	569 boş olmayan	float64
29 en kötü fraktal boyut	569 boş olmayan	float64

dtypes: float64(30)
bellek kullanımı: 133,5 KB

df_feat.nunique()

-Her sütundaki benzersiz değerlerin sayısını listeliyor.

-ortalama yarıçap	456
ortalama doku	479
ortalama çevre	522
ortalama alan	539
ortalama pürüzsüzlük	474
ortalama kompaktlık	537
ortalama içbükeylik	537
ortalama içbükey noktalar	542
ortalama simetri	432
ortalama fraktal boyut	499
yarıçap hatası	540
doku hatası	519
çevre hatası	533
alan hatası	528
düzgünlük hatası	547

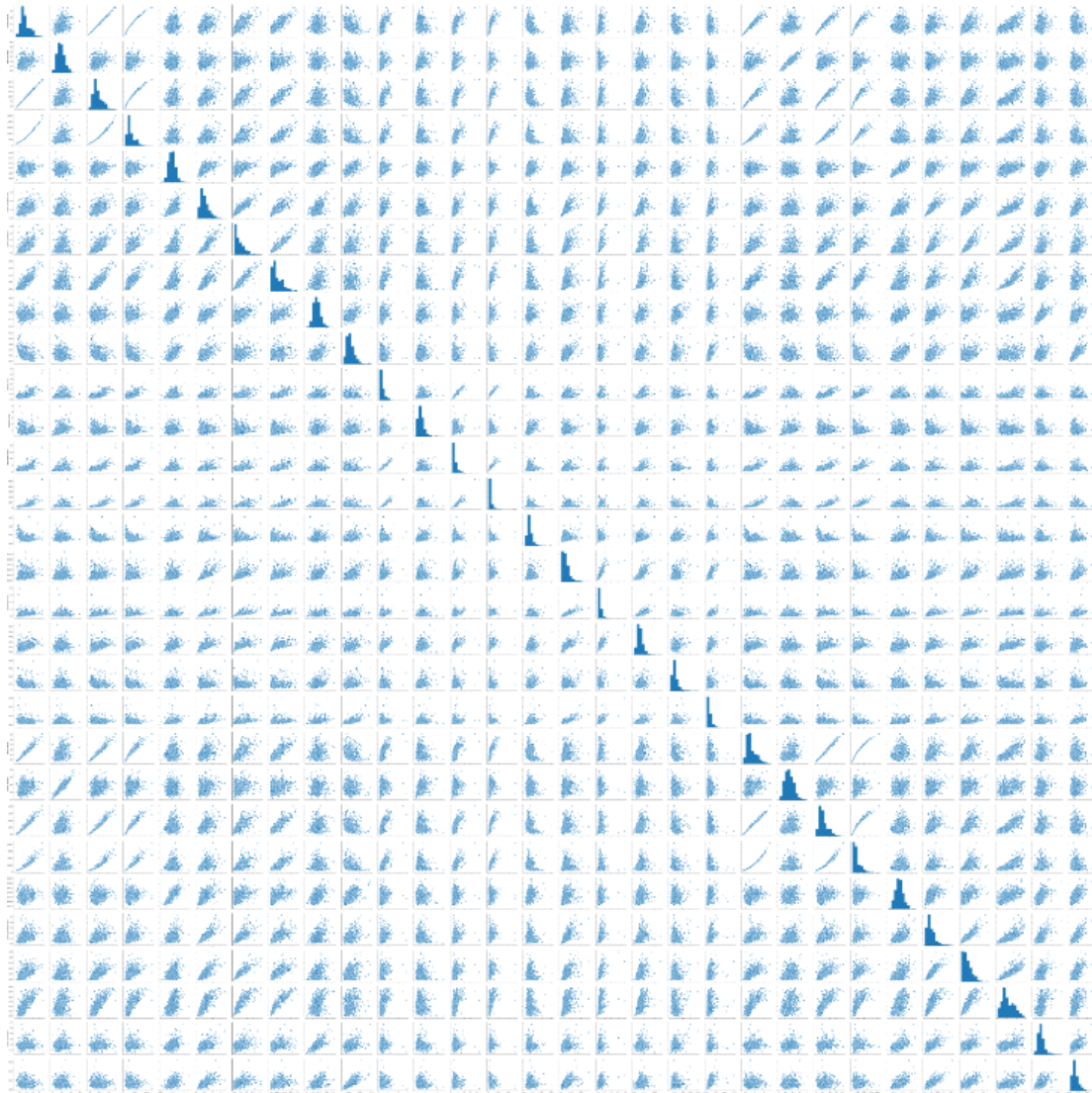
kompaktlık hatası	541
içbükeylik hatası	533
içbükey noktalar hatası	507
simetri hatası	498
fraktal boyut hatası	545
en kötü yarıçap	457
en kötü doku	511
en kötü çevre	514
en kötü bölge	544
en kötü pürüzsüzlük	411
en kötü kompaktlık	529
en kötü içbükeylik	539
en kötü içbükey noktalar	492
en kötü simetri	500
en kötü fraktal boyut	535
veri türü: int64	

`cancer['target_names']`

- Hedef değişkenin sınıf isimlerini listeliyor.
- `array(['malignant', 'benign'], dtype='<U9')`

`plt.figure(figsize=(10,6), dpi=600)`
`sns.pairplot(df_feat)`

- Özellikler arasındaki çiftli grafiklerin (pairplot) görselleştirilmesini sağlıyor.



```
from sklearn.model_selection import train_test_split
```

-Scikit-learn'den eğitim ve test verilerini ayırmak için kullanılan fonksiyonu import ediyor.

```
X = df_feat
```

```
y = cancer['target']
```

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.4, random_state=42)
```

-Özellikler ve hedef değişkeni belirleyip, veri setini eğitim ve test setlerine ayırıyor.

```
from sklearn.svm import SVC
```

-Destek vektör makineleri (SVM) modeli için gerekli sınıfı import ediyor.

```
model = SVC()
```

-SVM modelini oluşturuyor.

```
model.fit(X_train,y_train)
```

-SVM modelini eğitim verileriyle eğitiyor.

```
from sklearn.metrics import confusion_matrix,classification_report
```

-Modelin performansını değerlendirmek için gerekli metrikleri import ediyor.

```
print(confusion_matrix(y_test,prediction))
```

```
print('\n')
```

```
print(classification_report(y_test,prediction))
```

-Karışıklık matrisi ve sınıflandırma raporunu yazdırıyor.

```
[[ 68 12]
 [ 1 147]]
```

	precision	recall	f1-score	support
0	0.99	0.85	0.91	80
1	0.92	0.99	0.96	148
accuracy			0.94	228
macro avg	0.96	0.92	0.94	228
weighted avg	0.95	0.94	0.94	228

```
from sklearn.model_selection import GridSearchCV
```

-Hiperparametre optimizasyonu için GridSearchCV sınıfını import ediyor.

```
param_grid = {'C':[0.1,1,10,100,1000], 'gamma':[1,0.1,0.01,0.001,0.0001]}
```

-Hiperparametreler için bir grid tanımlıyor.

```
grid = GridSearchCV(SVC(), param_grid, verbose=3)
```

-GridSearchCV nesnesini oluşturuyor.

```
grid.fit(X_train,y_train)
```

-GridSearchCV ile modelin en iyi hiperparametrelerini bulmak için eğitim verilerini kullanıyor.

Fitting 5 folds for each of 25 candidates, totalling 125 fits(25 adayın her biri için 5 katlama, toplam 125 uyum)

```
[CV 1/5] END .....C=0.1, gamma=1,; score=0.609 total time= 0.0s
```

```
[CV 2/5] END .....C=0.1, gamma=1,; score=0.618 total time= 0.0s
```

```
[CV 3/5] END .....C=0.1, gamma=1,; score=0.618 total time= 0.0s
```

[CV 4/5] ENDC=0.1, gamma=1;, score=0.618 total time= 0.0s
 [CV 5/5] ENDC=0.1, gamma=1;, score=0.603 total time= 0.0s
 [CV 1/5] ENDC=0.1, gamma=0.1;, score=0.609 total time= 0.0s
 [CV 2/5] ENDC=0.1, gamma=0.1;, score=0.618 total time= 0.0s
 [CV 3/5] ENDC=0.1, gamma=0.1;, score=0.618 total time= 0.0s
 [CV 4/5] ENDC=0.1, gamma=0.1;, score=0.618 total time= 0.0s
 [CV 5/5] ENDC=0.1, gamma=0.1;, score=0.603 total time= 0.0s
 [CV 1/5] ENDC=0.1, gamma=0.01;, score=0.609 total time= 0.0s
 [CV 2/5] ENDC=0.1, gamma=0.01;, score=0.618 total time= 0.0s
 [CV 3/5] ENDC=0.1, gamma=0.01;, score=0.618 total time= 0.0s
 [CV 4/5] ENDC=0.1, gamma=0.01;, score=0.618 total time= 0.0s
 [CV 5/5] ENDC=0.1, gamma=0.01;, score=0.603 total time= 0.0s
 [CV 1/5] ENDC=0.1, gamma=0.001;, score=0.609 total time= 0.0s
 [CV 2/5] ENDC=0.1, gamma=0.001;, score=0.618 total time= 0.0s
 [CV 3/5] ENDC=0.1, gamma=0.001;, score=0.618 total time= 0.0s
 [CV 4/5] ENDC=0.1, gamma=0.001;, score=0.618 total time= 0.0s
 [CV 5/5] ENDC=0.1, gamma=0.001;, score=0.603 total time= 0.0s
 [CV 1/5] ENDC=0.1, gamma=0.0001;, score=0.855 total time= 0.0s
 [CV 2/5] ENDC=0.1, gamma=0.0001;, score=0.897 total time= 0.0s
 [CV 3/5] ENDC=0.1, gamma=0.0001;, score=0.882 total time= 0.0s
 [CV 4/5] ENDC=0.1, gamma=0.0001;, score=0.882 total time= 0.0s
 [CV 5/5] ENDC=0.1, gamma=0.0001;, score=0.868 total time= 0.0s
 [CV 1/5] ENDC=1, gamma=1;, score=0.609 total time= 0.0s
 [CV 2/5] ENDC=1, gamma=1;, score=0.618 total time= 0.0s
 [CV 3/5] ENDC=1, gamma=1;, score=0.618 total time= 0.0s
 [CV 4/5] ENDC=1, gamma=1;, score=0.618 total time= 0.0s
 [CV 5/5] ENDC=1, gamma=1;, score=0.603 total time= 0.0s

[CV 1/5] ENDC=1, gamma=0.1,, score=0.609 total time= 0.0s
 [CV 2/5] ENDC=1, gamma=0.1,, score=0.618 total time= 0.0s
 [CV 3/5] ENDC=1, gamma=0.1,, score=0.618 total time= 0.0s
 [CV 4/5] ENDC=1, gamma=0.1,, score=0.618 total time= 0.0s
 [CV 5/5] ENDC=1, gamma=0.1,, score=0.603 total time= 0.0s
 [CV 1/5] ENDC=1, gamma=0.01,, score=0.609 total time= 0.0s
 [CV 2/5] ENDC=1, gamma=0.01,, score=0.618 total time= 0.0s
 [CV 3/5] ENDC=1, gamma=0.01,, score=0.618 total time= 0.0s
 [CV 4/5] ENDC=1, gamma=0.01,, score=0.618 total time= 0.0s
 [CV 5/5] ENDC=1, gamma=0.01,, score=0.603 total time= 0.0s
 [CV 1/5] ENDC=1, gamma=0.001,, score=0.928 total time= 0.0s
 [CV 2/5] ENDC=1, gamma=0.001,, score=0.897 total time= 0.0s
 [CV 3/5] ENDC=1, gamma=0.001,, score=0.912 total time= 0.0s
 [CV 4/5] ENDC=1, gamma=0.001,, score=0.897 total time= 0.0s
 [CV 5/5] ENDC=1, gamma=0.001,, score=0.868 total time= 0.0s
 [CV 1/5] ENDC=1, gamma=0.0001,, score=0.899 total time= 0.0s
 [CV 2/5] ENDC=1, gamma=0.0001,, score=0.912 total time= 0.0s
 [CV 3/5] ENDC=1, gamma=0.0001,, score=0.926 total time= 0.0s
 [CV 4/5] ENDC=1, gamma=0.0001,, score=0.912 total time= 0.0s
 [CV 5/5] ENDC=1, gamma=0.0001,, score=0.868 total time= 0.0s
 [CV 1/5] ENDC=10, gamma=1,, score=0.609 total time= 0.0s
 [CV 2/5] ENDC=10, gamma=1,, score=0.618 total time= 0.0s
 [CV 3/5] ENDC=10, gamma=1,, score=0.618 total time= 0.0s
 [CV 4/5] ENDC=10, gamma=1,, score=0.618 total time= 0.0s
 [CV 5/5] ENDC=10, gamma=1,, score=0.603 total time= 0.0s
 [CV 1/5] ENDC=10, gamma=0.1,, score=0.609 total time= 0.0s
 [CV 2/5] ENDC=10, gamma=0.1,, score=0.618 total time= 0.0s

[CV 3/5] END	C=10, gamma=0.1,;	score=0.618 total time=	0.0s
[CV 4/5] END	C=10, gamma=0.1,;	score=0.618 total time=	0.0s
[CV 5/5] END	C=10, gamma=0.1,;	score=0.603 total time=	0.0s
[CV 1/5] END	C=10, gamma=0.01,;	score=0.609 total time=	0.0s
[CV 2/5] END	C=10, gamma=0.01,;	score=0.618 total time=	0.0s
[CV 3/5] END	C=10, gamma=0.01,;	score=0.632 total time=	0.0s
[CV 4/5] END	C=10, gamma=0.01,;	score=0.632 total time=	0.0s
[CV 5/5] END	C=10, gamma=0.01,;	score=0.603 total time=	0.0s
[CV 1/5] END	C=10, gamma=0.001,;	score=0.913 total time=	0.0s
[CV 2/5] END	C=10, gamma=0.001,;	score=0.912 total time=	0.0s
[CV 3/5] END	C=10, gamma=0.001,;	score=0.912 total time=	0.0s
[CV 4/5] END	C=10, gamma=0.001,;	score=0.897 total time=	0.0s
[CV 5/5] END	C=10, gamma=0.001,;	score=0.838 total time=	0.0s
[CV 1/5] END	C=10, gamma=0.0001,;	score=0.957 total time=	0.0s
[CV 2/5] END	C=10, gamma=0.0001,;	score=0.926 total time=	0.0s
[CV 3/5] END	C=10, gamma=0.0001,;	score=0.926 total time=	0.0s
[CV 4/5] END	C=10, gamma=0.0001,;	score=0.926 total time=	0.0s
[CV 5/5] END	C=10, gamma=0.0001,;	score=0.912 total time=	0.0s
[CV 1/5] END	C=100, gamma=1,;	score=0.609 total time=	0.0s
[CV 2/5] END	C=100, gamma=1,;	score=0.618 total time=	0.0s
[CV 3/5] END	C=100, gamma=1,;	score=0.618 total time=	0.0s
[CV 4/5] END	C=100, gamma=1,;	score=0.618 total time=	0.0s
[CV 5/5] END	C=100, gamma=1,;	score=0.603 total time=	0.0s
[CV 1/5] END	C=100, gamma=0.1,;	score=0.609 total time=	0.0s
[CV 2/5] END	C=100, gamma=0.1,;	score=0.618 total time=	0.0s
[CV 3/5] END	C=100, gamma=0.1,;	score=0.618 total time=	0.0s
[CV 4/5] END	C=100, gamma=0.1,;	score=0.618 total time=	0.0s

[CV 5/5] ENDC=100, gamma=0.1,, score=0.603 total time= 0.0s
 [CV 1/5] ENDC=100, gamma=0.01,, score=0.609 total time= 0.0s
 [CV 2/5] ENDC=100, gamma=0.01,, score=0.618 total time= 0.0s
 [CV 3/5] ENDC=100, gamma=0.01,, score=0.632 total time= 0.0s
 [CV 4/5] ENDC=100, gamma=0.01,, score=0.632 total time= 0.0s
 [CV 5/5] ENDC=100, gamma=0.01,, score=0.603 total time= 0.0s
 [CV 1/5] ENDC=100, gamma=0.001,, score=0.913 total time= 0.0s
 [CV 2/5] ENDC=100, gamma=0.001,, score=0.912 total time= 0.0s
 [CV 3/5] ENDC=100, gamma=0.001,, score=0.912 total time= 0.0s
 [CV 4/5] ENDC=100, gamma=0.001,, score=0.897 total time= 0.0s
 [CV 5/5] ENDC=100, gamma=0.001,, score=0.838 total time= 0.0s
 [CV 1/5] ENDC=100, gamma=0.0001,, score=1.000 total time= 0.0s
 [CV 2/5] ENDC=100, gamma=0.0001,, score=0.941 total time= 0.0s
 [CV 3/5] ENDC=100, gamma=0.0001,, score=0.941 total time= 0.0s
 [CV 4/5] ENDC=100, gamma=0.0001,, score=0.912 total time= 0.0s
 [CV 5/5] ENDC=100, gamma=0.0001,, score=0.912 total time= 0.0s
 [CV 1/5] ENDC=1000, gamma=1,, score=0.609 total time= 0.0s
 [CV 2/5] ENDC=1000, gamma=1,, score=0.618 total time= 0.0s
 [CV 3/5] ENDC=1000, gamma=1,, score=0.618 total time= 0.0s
 [CV 4/5] ENDC=1000, gamma=1,, score=0.618 total time= 0.0s
 [CV 5/5] ENDC=1000, gamma=1,, score=0.603 total time= 0.0s
 [CV 1/5] ENDC=1000, gamma=0.1,, score=0.609 total time= 0.0s
 [CV 2/5] ENDC=1000, gamma=0.1,, score=0.618 total time= 0.0s
 [CV 3/5] ENDC=1000, gamma=0.1,, score=0.618 total time= 0.0s
 [CV 4/5] ENDC=1000, gamma=0.1,, score=0.618 total time= 0.0s
 [CV 5/5] ENDC=1000, gamma=0.1,, score=0.603 total time= 0.0s
 [CV 1/5] ENDC=1000, gamma=0.01,, score=0.609 total time= 0.0s

```
[CV 2/5] END .....C=1000, gamma=0.01;;, score=0.618 total time= 0.0s
[CV 3/5] END .....C=1000, gamma=0.01;;, score=0.632 total time= 0.0s
[CV 4/5] END .....C=1000, gamma=0.01;;, score=0.632 total time= 0.0s
[CV 5/5] END .....C=1000, gamma=0.01;;, score=0.603 total time= 0.0s
[CV 1/5] END .....C=1000, gamma=0.001;;, score=0.913 total time= 0.0s
[CV 2/5] END .....C=1000, gamma=0.001;;, score=0.912 total time= 0.0s
[CV 3/5] END .....C=1000, gamma=0.001;;, score=0.912 total time= 0.0s
[CV 4/5] END .....C=1000, gamma=0.001;;, score=0.897 total time= 0.0s
[CV 5/5] END .....C=1000, gamma=0.001;;, score=0.838 total time= 0.0s
[CV 1/5] END .....C=1000, gamma=0.0001;;, score=1.000 total time= 0.0s
[CV 2/5] END .....C=1000, gamma=0.0001;;, score=0.912 total time= 0.0s
[CV 3/5] END .....C=1000, gamma=0.0001;;, score=0.868 total time= 0.0s
[CV 4/5] END .....C=1000, gamma=0.0001;;, score=0.912 total time= 0.0s
[CV 5/5] END .....C=1000, gamma=0.0001;;, score=0.926 total time= 0.0s
```

`grid.best_params_`

-En iyi hiperparametreleri gösteriyor.

-{'C': 100, 'gamma': 0.0001}

`grid.best_estimator_`

-En iyi modeli gösteriyor.

- SVC(C=100, gamma=0.0001)

`grid_predictions = grid.predict(X_test)`

-Test verileriyle en iyi modelin tahminlerini yapıyor.

```
print(confusion_matrix(y_test,grid_predictions))  
  
print('\n')  
  
print(classification_report(y_test,grid_predictions))
```

-En iyi modelin performansını karışıklık matrisi ve sınıflandırma raporu ile değerlendiriyor.

```
[[ 72  8]  
 [ 5 143]]
```

	precision	recall	f1-score	support
0	0.94	0.90	0.92	80
1	0.95	0.97	0.96	148
accuracy			0.94	228
macro avg	0.94	0.93	0.94	228
weighted avg	0.94	0.94	0.94	228

KAYNAKLAR

<https://medium.com/@poornaprudhvi/support-vector-machines-bf0242ade522>

https://en.wikipedia.org/wiki/Support_vector_machine#:~:text=History,trick%20to%20maximum%2Dmargin%20hyperplanes.

<https://github.com/ryanschub/Breast-Cancer-Classification-using-Support-Vector-Machine-Models/blob/master/Homework%203.ipynb>

[https://github.com/babak01/Breast-cancer-wisconsin-dataset/blob/master/SVM%20-%20Breast%20cancer%20wisconsin%20\(diagnostic\)%20dataset.ipynb](https://github.com/babak01/Breast-cancer-wisconsin-dataset/blob/master/SVM%20-%20Breast%20cancer%20wisconsin%20(diagnostic)%20dataset.ipynb)

https://github.com/kts2001/Breast-Cancer-Dataset/blob/main/kareem_zahra_svm.ipynb

https://github.com/Paritoshyadav/breast-cancer-classifier/blob/master/Breast_Cancer.ipynb

nilgün_ŞENGÖZ_pdf_sunumu

<https://archive.ics.uci.edu/dataset/17/breast+cancer+wisconsin+diagnostic>

KAYNAK KOD ADRESİ

<https://github.com/Erencanylmz/Erencan-SVM>