

## Parametre Tuning (C, gamma)

"Parametre tuning", yani parametre ayarlama, özellikle destek vektör makineleri (SVM - Support Vector Machine) gibi algoritmalar kullanılırken modelin performansını artırmak için çok önemli bir adımdır. SVM'de en çok ayarlanan iki parametre C ve gamma'dır.

### 1. C Parametresi (Regularization parameter)

- **Amaç:** Hataları ne kadar cezalandıracağımızı belirler.
- **C büyük** → Model **daha az hataya** izin verir, **karmaşık** bir sınırlayıcı çizgi çizebilir (overfitting riski artar).
- **C küçük** → Model daha **basit** bir sınırlayıcı çizgi çizer, **daha çok hataya** izin verir (underfitting riski artar).

### 2. Gamma (γ) Parametresi

- **Gamma**, bir noktanın etkisinin ne kadar uzaklığa yayılacağını belirler. Özellikle **RBF (Radial Basis Function)** gibi çekirdek (kernel) fonksiyonlarında kullanılır.
- **Gamma büyük** → Her veri noktası çok **dar** bir etki alanına sahiptir → model **çok karmaşık** olabilir (overfitting).
- **Gamma küçük** → Veri noktalarının etkisi daha **geniş alanlara** yayılır → model **daha genel** olur (underfitting olabilir).

## Parametre Ayarlaması (Tuning) Nasıl Yapılır?

Parametreleri doğru seçmek için:

- **Grid Search:** C ve gamma değerlerini bir aralıkta dener ve en iyi sonucu seçer.
- **Random Search:** Rastgele kombinasyonlar dener.
- **Cross-validation:** Hataları test etmek için veriyi parçalara ayırarak her bir kombinasyonu test eder.

### Örnek Python kodu (scikit-learn ile):

```
from sklearn.model_selection import GridSearchCV

from sklearn.svm import SVC

param_grid = {
    'C': [0.1, 1, 10, 100],
    'gamma': [1, 0.1, 0.01, 0.001],
    'kernel': ['rbf']
}

grid = GridSearchCV(SVC(), param_grid, refit=True, verbose=2)

grid.fit(X_train, y_train)

print("En iyi parametreler:", grid.best_params_)
```

## Faker Nedir?

**Faker**, isimler, adresler, telefon numaraları, e-postalar, tarihler, şirket adları, kredi kartı bilgileri, vs. gibi **gerçeğe yakın ama sahte** veriler oluşturur.

### Kurulum:

```
pip install faker
```

### Temel Kullanım:

```
from faker import Faker

fake = Faker()

print(fake.name())    # Rastgele bir isim

print(fake.address())  # Rastgele bir adres
```

```
print(fake.email())    # Rastgele bir e-posta
```

#### **Çıktı:**

John Doe

1234 Main Street

Springfield, IL 62704

[john.doe@example.com](mailto:john.doe@example.com)

#### **Dil Ayarı (Türkçe dahil!)**

```
fake = Faker('tr_TR') # Türkçe veriler üretir
```

```
print(fake.name())    # Türkçe bir isim
```

```
print(fake.address()) # Türkçe bir adres
```

#### **Çıktı:**

Ahmet Yılmaz

Çiçek Mah. 45. Sk. No:7

Kadıköy, İstanbul

#### **Çoklu Veri Üretme:**

```
for _ in range(10):
```

```
    user = {
```

```
        'ad': fake.first_name(),
```

```
        'soyad': fake.last_name(),
```

```
        'email': fake.email(),
```

```
        'adres': fake.address()
```

```
}  
print(user)
```

### Kullanılabilir Bazı Özellikler:

Özellik	Açıklama
<code>fake.name()</code>	İsim
<code>fake.address()</code>	Adres
<code>fake.email()</code>	E-posta
<code>fake.phone_number()</code>	Telefon
<code>fake.company()</code>	Şirket adı
<code>fake.date_of_birth()</code>	Doğum tarihi
<code>fake.text()</code>	Paragraf metni
<code>fake.job()</code>	Meslek
<code>fake.iban()</code>	IBAN numarası
<code>fake.credit_card_number()</code>	Kredi kartı no

## Ekstra Bilgiler

- `Faker.seed(123)` → Aynı sahte verileri her seferinde üretmek için.
- `fake.unique.name()` → Benzersiz isimler üretir (aynı isim bir daha gelmez).
- `fake.profile()` → Ad, doğum tarihi, e-posta gibi bilgileri tek seferde verir.