VERİ BİLİMİ DERSİ

Sınıflandırma ve Kaggle

Hafta 6 · Modül 6
Lojistik Regresyon, KNN, Karar Ağacı ve İlk Kaggle Yarışması

Dr. Murat Altun

Veri Bilimi ve Yapay Zekâ Eğitimi · 2026

6
Saat
Uygulama Ağırlıklı

3
Notebook
(Temel + Titanic + Diyabet)

3
Algoritma
(LR · KNN · DT)

İçindekiler

Sınıflandırma Nedir?

Binary Sınıflandırma

Sadece 2 sınıf: Evet/Hayır, Spam/Normal, Hasta/Sağlıklı.
Çıktı: 0 veya 1 (olasılık ile)

Multi-class Sınıflandırma

3+ sınıf: Hayvan türü, rakam tanıma (0-9), hastalık tipi.
Çıktı: N sınıftan biri

Gerçek Dünya Uygulamaları

Spam Tespiti

E-posta spam mı değil mi?

Tıbbi Teşhis

Tümör iyi huylu mu kötü mü?

Kredi Riski

Müşteri ödeyecek mi?

Rakam Tanıma

El yazısı → 0-9 sınıfı

Lojistik Regresyon

Sigmoid Fonksiyonu

Doğrusal kombinasyonu 0–1 arasına sıkıştırır.

σ(z) = 1 / (1 + e^(-z))

- z > 0 → olasılık > 0.5 → Sınıf 1
- z < 0 → olasılık < 0.5 → Sınıf 0
- Karar eşiği varsayılan: 0.5

Avantajlar

Hızlı · Yorumlanabilir · Olasılık çıktısı

Dezavantajlar

Doğrusal sınır · Karmaşık ilişkilerde zayıf

```
from sklearn.linear_model import LogisticRegression

model = LogisticRegression(max_iter=200)
model.fit(X_train, y_train)
y_pred = model.predict(X_test)
y_proba = model.predict_proba(X_test)  # Olasılık çıktısı
```

K-Nearest Neighbors (KNN)

Nasıl Çalışır?

1. Yeni veri noktasına en yakın K komşuyu bul
2. Komşuların çoğunluk sınıfını ata
3. Uzaklık ölçütü: Öklid, Manhattan

K Seçimi Kritik:
- K küçük → gürültüye hassas (overfitting)
- K büyük → fazla genelleme (underfitting)
- Genelde K = √n veya tek sayı seçilir

K=5
Varsayılan Komşu Sayısı

O(n)
Tahmin Maliyeti

Avantajlar

- Basit ve sezgisel
- Eğitim süresi yok (lazy learner)
- Non-linear sınırlara uyum sağlar

Dezavantajlar ve Dikkat Edilecekler

- Büyük veri setlerinde yavaş (tüm mesafeler hesaplanır)
- Özellik ölçekleme zorunlu (StandardScaler / MinMaxScaler)
- Yüksek boyutlu veride performans düşer (boyut laneti)
- Kategorik özelliklerle doğrudan çalışamaz

Karar Ağacı (Decision Tree)

Ağaç Yapısı ve Bölme Kriterleri

Kök Düğüm → Dallar → Yapraklar

Her düğüm bir özelliğe göre veriyi böler:
- Gini Impurity: $1 - \Sigma(p_i^2) \rightarrow 0 = \text{saf}$
- Entropy: $-\Sigma(p_i \times \log_2(p_i)) \rightarrow 0 = \text{saf}$

Ağaç en saf yapraklara ulaşana kadar büyür.
Bilgi kazancı en yüksek özellik seçilir.

Gini Impurity

Hızlı hesaplama · sklearn varsayılanı

Entropy (Bilgi Kazancı)

Daha dengeli bölme · Daha yavaş hesaplama

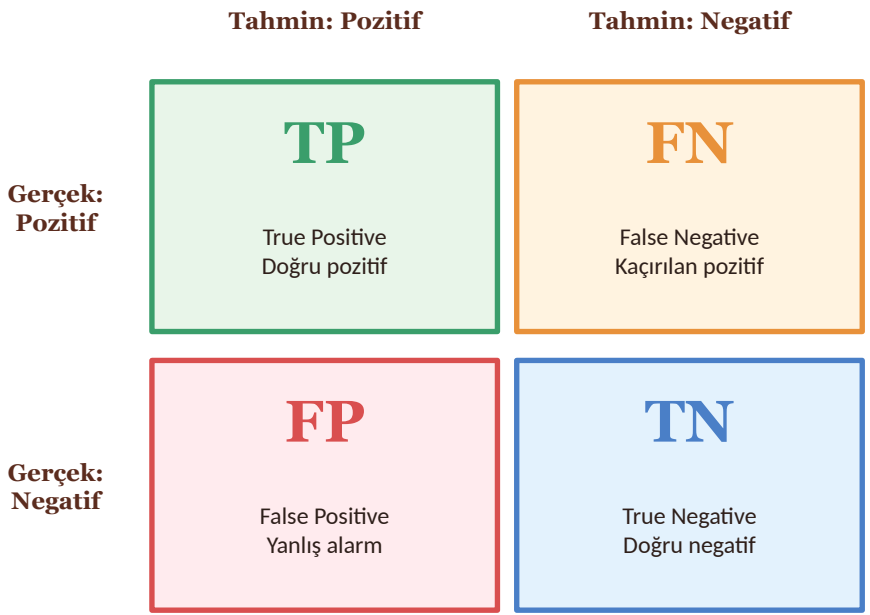
Overfitting Riski ve Çözümler

`max_depth`	Ağaç derinliğini sınırla (ör: 5)
`min_samples_split`	Bölme için minimum örnek (ör: 10)
`min_samples_leaf`	Yaprakta minimum örnek (ör: 5)
`Pruning`	ccp_alpha ile budama (cost-complexity)

3 Algoritma Karşılaştırma Tablosu

Özellik	Lojistik Regresyon	KNN	Karar Ağacı
Tip	Parametrik	Non-parametrik	Non-parametrik
Eğitim Hızı	Çok hızlı	Yok (lazy)	Orta
Tahmin Hızı	Çok hızlı	Yavaş (O(n))	Çok hızlı
Yorumlanabilirlik	Yüksek (katsayılar)	Düşük	Çok yüksek (görsel)
Non-linear Sınır	Hayır	Evet	Evet
Ölçekleme Gerekli	Evet	Evet (zorunlu)	Hayır
Overfitting Riski	Düşük	Orta	Yüksek
Hiperparametre	C, penalty	K, distance metric	max_depth, criterion

Confusion Matrix (Karışıklık Matrisi)



Neden Önemli?

Accuracy tek başına yanıltıcı olabilir!

Örnek: 1000 hasta, 950 sağlıklı
Model hep "sağlıklı" derse:
- Accuracy = %95 (harika görünür!)
- Ama hiç hasta tespit edilmedi

Confusion Matrix bize 4 farklı hata türünü ayrı ayrı gösterir.

Özellikle dengesiz veri setlerinde kritik bir araçtır.

Metrikler: Accuracy, Precision, Recall

Accuracy

`(TP + TN) / Toplam`

Genel doğruluk oranı.

Ne zaman kullan?
Dengeli sınıf dağılımı varsa.

Precision

`TP / (TP + FP)`

Pozitif dediğimin kaçı gerçekten pozitif?

Ne zaman kullan?
FP maliyetli: Spam filtresi

Recall

`TP / (TP + FN)`

Gerçek pozitiflerin kaçını yakaladım?

Ne zaman kullan?
FN maliyetli: Kanser teşhisi

Senaryo Karşılaştırması

Senaryo	Öncelik	Neden?
Spam filtresi	Precision	Normal mail spam klasörüne düşmesin (FP azalt)
Kanser tespiti	Recall	Hasta kaçırılmasın, yanlış alarm kabul edilebilir (FN azalt)
Kredi dolandırıcılığı	Precision + Recall	İkisi de kritik: hem kaçırma hem yanlış alarm pahalı

F1-Score: Denge Metriği

Harmonik Ortalama

$$F1 = 2 \times (Precision \times Recall) / (Precision + Recall)$$

Aritmetik ortalama yerine harmonik ortalama: düşük olan değer F1'i daha çok etkiler

0.67
P=1.0, R=0.5
Yüksek P, Düşük R

0.80
P=0.8, R=0.8
Dengeli

0.91
P=0.9, R=0.92
İyi Denge

1.00
P=1.0, R=1.0
Mükemmel

Dengesiz Veri Setlerinde F1-Score Kritik

- Accuracy %95 olsa bile F1-Score %30 olabilir (nadir sınıf yakalanmıyorsa)
- Macro F1: Her sınıfın F1'ini eşit ağırlıkla ortalar → dengesiz setlerde adil
- Weighted F1: Sınıf büyüklüğüne göre ağırlıklı → genel performans göstergesi

ROC Eğrisi ve AUC

ROC Eğrisi Ne Gösterir?

X ekseni: False Positive Rate (FPR)
Y ekseni: True Positive Rate (TPR = Recall)

Her eşik değeri (threshold) için bir nokta:
- Sol üst köşeye yakınlık = iyi model
- Köşegen çizgi = rastgele tahmin
- Eğri altında kalan alan = AUC


```
from sklearn.metrics import roc_curve, roc_auc_score

y_proba = model.predict_proba(X_test)[:, 1]
fpr, tpr, thresholds = roc_curve(y_test, y_proba)
auc_score = roc_auc_score(y_test, y_proba)
print(f"AUC: {auc_score:.3f}")  # AUC: 0.876
```

Cross Validation (Çapraz Doğrulama)

BÖLÜM 3

Neden Tek Split Yetmez?

- Train/test bölümü rastgeledir
- Şanslı/şanssız bir bölme olabilir
- Model performansı değişkenlik gösterir
- Küçük veri setlerinde özellikle sorunlu

K-Fold Cross Validation

1. Veriyi K eşit parçaya böl
2. Her seferinde 1 parça test, K-1 eğitim
3. K kez tekrarla, sonuçları ortala

K=5 veya K=10 yaygın tercih


```
from sklearn.model_selection import cross_val_score

scores = cross_val_score(model, X, y, cv=5, scoring='accuracy')
print(f"Ortalama: {scores.mean():.3f} ± {scores.std():.3f}")
```

Titanic: İlk Kaggle Yarışması

BÖLÜM 4

Veri Seti Tanıtımı

891 yolcu · 12 özellik · Binary sınıflandırma

Hedef: Survived (0 = Hayır, 1 = Evet)
Kaggle'ın en popüler başlangıç yarışması!
"Machine Learning from Disaster"
Submission: PassengerId + Survived (0/1) → CSV dosyası yükle

891
Eğitim Örnekleri

418
Test Örnekleri

Temel Özellikler

Özellik	Tip	Açıklama
Pclass	Kategorik (1-3)	Yolcu sınıfı (1st, 2nd, 3rd)
Sex	Kategorik	Cinsiyet (male, female)
Age	Sayısal	Yaş (eksik veri var!)
SibSp / Parch	Sayısal	Kardeş + Ebeveyn/Çocuk sayısı
Fare	Sayısal	Bilet ücreti
Embarked	Kategorik	Biniş limanı (C, Q, S)

Titanic: Veri Hazırlama

1

Eksik Veri

Age → median ile doldur
Embarked → mod ile doldur
Cabin → düşür (çok eksik)

2

Encoding

Sex → 0/1 (LabelEncoder)
Embarked → One-Hot Encoding
Pclass zaten sayısal

3

Feature Engineering

FamilySize = SibSp + Parch + 1
IsAlone = FamilySize == 1
Title: Name'den çıkar (Mr, Mrs...)

4

Özellik Seçimi

Name, Ticket, Cabin → düşür
PassengerId → düşür
Kalan: Pclass, Sex, Age, Fare, +yeni

Titanic: Model Eğitimi ve Karşılaştırma


```
from sklearn.linear_model import LogisticRegression
from sklearn.neighbors import KNeighborsClassifier
from sklearn.tree import DecisionTreeClassifier

models = {
    "Lojistik Regresyon": LogisticRegression(max_iter=200),
    "KNN (K=5)":          KNeighborsClassifier(n_neighbors=5),
    "Karar Ağacı":        DecisionTreeClassifier(max_depth=5),
}

for name, model in models.items():
    scores = cross_val_score(model, X, y, cv=5)
    print(f"{name}: {scores.mean():.3f} ± {scores.std():.3f}")
```

%80.2

Lojistik Regresyon

%78.5

KNN (K=5)

%79.8

Karar Ağacı (depth=5)

Diyabet Teşhisi: Pima Indians

Pima Indians Diabetes Dataset

768 kadın hasta · 8 klinik özellik
Hedef: Outcome (0 = Diyabet yok, 1 = Diyabet)

Özellikler:
Pregnancies · Glucose · BloodPressure
SkinThickness · Insulin · BMI
DiabetesPedigreeFunction · Age

768
Toplam Örnek

%34.9
Diyabet Oranı

Dikkat!

Dengesiz veri seti!
Accuracy yerine F1-Score kullan.

Uygulama Pipeline

1. **Veri Yükle**
   pd.read_csv('diabetes.csv')
2. **Temizle**
   0 değerler → NaN → median
3. **Ölçekle**
   StandardScaler fit_transform
4. **Böl**
   train_test_split (80/20)
5. **Eğit+Değerlendir**
   3 model + classification_report

Kaggle Kültürü ve İlk Adımlar

BÖLÜM 4

1 Hesap Açma

kaggle.com'da ücretsiz kayıt.
Google hesabı ile giriş.
Profil bilgilerini doldur.
Portföy görünürlüğü!

2 Notebook Kullanımı

Kaggle Notebooks: ücretsiz GPU!
30 saat/hafta GPU kotası.
Kod + markdown + çıktı bir arada.
Paylaş ve oy topla.

3 Veri Setleri

50.000+ açık veri seti.
CSV, JSON, resim, metin...
Kendi veri setini yükle.
Dataset yarışmaları.

4 Yarışma & Submission

Submit → Leaderboard sıralaması.
Public / Private LB farkı!
Kernel-only yarışmalar.
Medal sistemi (Bronz→Altın).

Haftalık Notebook'lar

1 siniflandirma_temel.ipynb

Sınıflandırma algoritmalarının temel uygulaması.

- Iris veri seti ile 3 algoritma
- Confusion matrix ve metrikler
- Karar sınırı görselleştirme
- Cross validation karşılaştırma

~60 dk

2 titanic_kaggle.ipynb

Kaggle Titanic yarışması tam çözüm.

- EDA ve veri temizleme
- Feature engineering (Title, FamilySize)
- 3 model eğitim + karşılaştırma
- Kaggle submission dosyası oluşturma

~90 dk

3 diyabet_teshisi.ipynb

Pima Indians diyabet tahmini projesi.

- Klinik veri analizi ve ön işleme
- Dengesiz veri ile başa çıkma
- F1-Score ve ROC/AUC değerlendirme
- En iyi modeli seçme stratejisi

~60 dk

Ödev ve Kaynaklar

Haftalık Ödev

1 Kaggle Titanic Submission

Titanic notebook'unu tamamla ve Kaggle'a submit et. Skor ekran görüntüsünü paylaş.

2 3 Algoritma Raporu

Her algoritma için: Accuracy, Precision, Recall, F1-Score ve Confusion Matrix tablosu.

3 En İyi Model Analizi

Neden o model en iyi? Hangi metriğe göre seçtin? 1 sayfa yazılı açıklama.

Kaynaklar

- **Kaggle Learn: Intro to ML**
  kaggle.com/learn/intro-to-machine-learning
- **sklearn Classification Docs**
  scikit-learn.org/stable/supervised_learning.html
- **Titanic Tutorial (Kaggle)**
  kaggle.com/competitions/titanic
- **Confusion Matrix Explained**
  YouTube: StatQuest
- **ROC ve AUC Açıklaması**
  YouTube: StatQuest
- **Pima Indians Dataset**
  kaggle.com/datasets/uciml/pima-indians-diabetes

Teslim

Kaggle submission ekran görüntüsü + rapor → bir sonraki hafta dersten önce. Kaggle hesap linkinizi paylaşmayı unutmayın!

Hafta 6 — Önemli Çıkarımlar

1. Sınıflandırma, veriyi önceden tanımlı kategorilere atama işlemidir — binary veya multi-class
2. Lojistik Regresyon, KNN ve Karar Ağacı farklı güçlere sahip: probleme göre seçin
3. Confusion Matrix ve metrikler (Precision, Recall, F1) doğru değerlendirmenin temelidir
4. Cross Validation ile model performansını güvenilir şekilde ölçün
5. Kaggle ile gerçek dünya problemlerinde pratik yapın — öğrenmenin en iyi yolu uygulamadır!

"Veri bilimcinin en güçlü silahı doğru soruyu sormak ve doğru metrikle değerlendirmektir."

Dr. Murat Altun · Veri Bilimi ve Yapay Zekâ Eğitimi · 2026