

# **Breast Cancer Sınıflandırma Ve Streamlit Entegrasyonu**

**HAZIRLAYAN**

Metin Ilgar Mutlu

# İçindekiler

## Breast Cancer Sınıflandırma Ve Streamlit Entegrasyonu..... 1

İçindekiler.....	2
Proje Genel Bakış.....	3
1. Keşifsel Veri Analizi (Exploratory Data Analysis – EDA).....	3
1.1. Veri Setinin İncelenmesi.....	3
1.2. Hedef (Target) Değişken Analizi.....	3
1.3. Özellik Analizi ve Görselleştirmeler.....	4
1.4. Korelasyon Analizi.....	4
2. Veri Ön İşleme.....	5
2.1. Eksik Değer Kontrolü ve İlgisiz Sütunların Kaldırılması.....	5
2.2. Hedef Değişkenin Sayısal Olarak Temsili.....	5
2.3. Aykırı Değer Tespiti ve Temizleme.....	6
2.4. Yüksek Korelasyonlu Özelliklerin Kaldırılması.....	6
2.5. Veriyi Eğitim ve Test Olarak Ayırma.....	6
2.6. Özellik Normalizasyonu (Standardizasyon).....	6
2.7. Dengesizlik Durumu ve SMOTE Uygulaması.....	6
3. Geleneksel Makine Öğrenmesi Modelleri.....	7
3.1. Logistic Regression.....	7
3.3. Support Vector Machine (SVM).....	8
3.4. K-Nearest Neighbors (KNN).....	9
3.5. Gradient Boosting (XGBoost ve LightGBM).....	9
4. Derin Öğrenme (Deep Learning) Modelleri.....	10
4.1. Model Mimarisi ve Eğitim Süreci.....	10
4.2. Model Performans Değerlendirmesi.....	11
5. Model Analizi ve Değerlendirme.....	12
5.1. Modellerin Karşılaştırmalı Performans Tablosu.....	12
5.2. Performans Karşılaştırma Grafiği.....	13
6. Çıkarılan Sonuçlar ve Öneriler.....	13
7. Özet.....	14
8. Kullanıcıların Deneysel Çalışmalarını Yürütmesi ve Sonuçlarını Değerlendirebilmesi için Streamlit Entegrasyonu.....	15
Platform Hakkında.....	15
Deneysel Çalışmalar İçin Sunulan Olanaklar.....	15
Yasal Uyarı.....	17
Kaynaklar.....	18

# Proje Genel Bakış

Projenin temel hedefi, veri setindeki tümör özelliklerine dayanarak bir tümörün iyi huylu (Benign) mu yoksa kötü huylu (Malignant) mu olduğunu yüksek doğrulukla tahmin edebilen sınıflandırma modelleri geliştirmektir. Çalışmanın nihai hedefi olarak, bu modelleri interaktif bir web uygulamasına dönüştürülmüştür. Aşamalar şu şekilde ilerlemiştir:

- Keşifsel Veri Analizi (EDA)
- Veri Ön İşleme
- Geleneksel Makine Öğrenmesi Modellerinin Eğitimi ve Değerlendirilmesi
- Derin Öğrenme Modelinin Eğitimi ve Değerlendirilmesi
- Model Karşılaştırması ve Sonuçların Yorumlanması
- Kullanıcıların Deneysel Çalışmalarını Yürütmesi ve Sonuçlarını Değerlendirebilmesi için Streamlit Entegrasyonu

Aşağıda her bir adımdaki uygulamalar, elde edilen çıktılar ve çıkarılan sonuçlar detaylı şekilde açıklanmıştır.

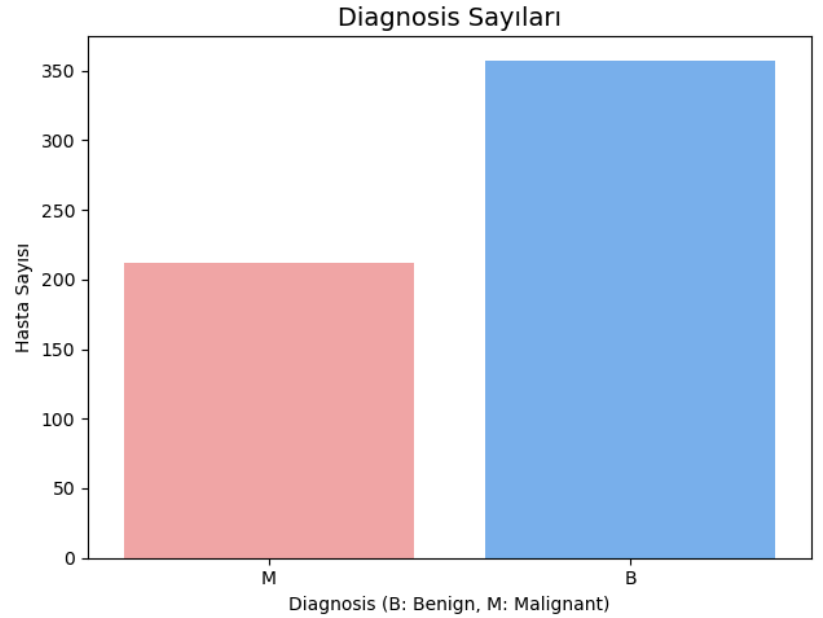
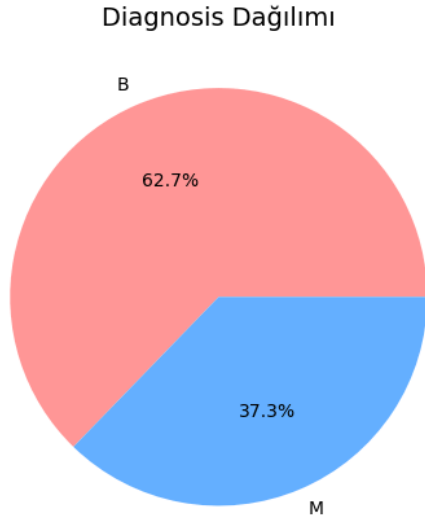
## 1. Keşifsel Veri Analizi (Exploratory Data Analysis – EDA)

### 1.1. Veri Setinin İncelenmesi

- Veri Seti Özeti
  - Toplam örnek sayısı: **569**
  - Toplam sütun sayısı: **33**
    - Bunlardan bir tanesi id (numaralandırma amaçlı)
    - Bir tanesi Unnamed: 32 adında ve tüm değerleri boş (tümüyle null)
    - Geriye kalan 30 sütun, tümörleri tanımlayan çeşitli ölçümlere (radius\_mean, texture\_mean, ..., fractal\_dimension\_worst) ait sayısal değerlerden oluşuyor.
  - Boş (null) değer kontrolü yapıldığında yalnızca Unnamed: 32 sütununun tüm satırlarının boş olduğu, geri kalan sütunlarda eksik veri olmadığı görüldü.
  - Bu nedenle, ham veri setinden bilgi içermeyen **id** ve **Unnamed: 32** sütunları kaldırıldı.

### 1.2. Hedef (Target) Değişken Analizi

- Hedef sütun: diagnosis
  - İki sınıf içeriyor:
    - **B (Benign, iyi huylu)**
    - **M (Malignant, kötü huylu)**
  - Sınıf Dağılımı:
    - Benign (B): **357** örnek (%62.7)
    - Malignant (M): **212** örnek (%37.3)
  - Gözlenen: Sınıflar çok uçuk oranda dengesiz değil; yine de “kötü huylu” sınıfı, yaklaşık %37 ile nispeten az ancak yine de anlamlı sayıda örnek barındırıyor. Bu problem ileride düzeltilecek.



### 1.3. Özellik Analizi ve Görselleştirmeler

- **Özellik Grupları**

- Veri setindeki 30 sayısal sütun, üç ana gruba ayrılmış:
  1. **mean** ile bitenler (örneğin radius\_mean, texture\_mean, ...)
  2. **se** ile bitenler (örneğin radius\_se, texture\_se, ...)
  3. **worst** ile bitenler (örneğin radius\_worst, texture\_worst, ...)

- **Pairplot (Seaborn ile)**

- Ayrılan özellik gruplarının her birinin listesi ile **pairplot** çizildi.
- Birden fazla özellik aralarında korelasyon bulunduğu, özellikle aynı ölçüm ailesindeki (ör. mean-perimeter ile mean-area) değişkenlerin güçlü doğrusal ilişki taşıdığı açıkça görüldü.

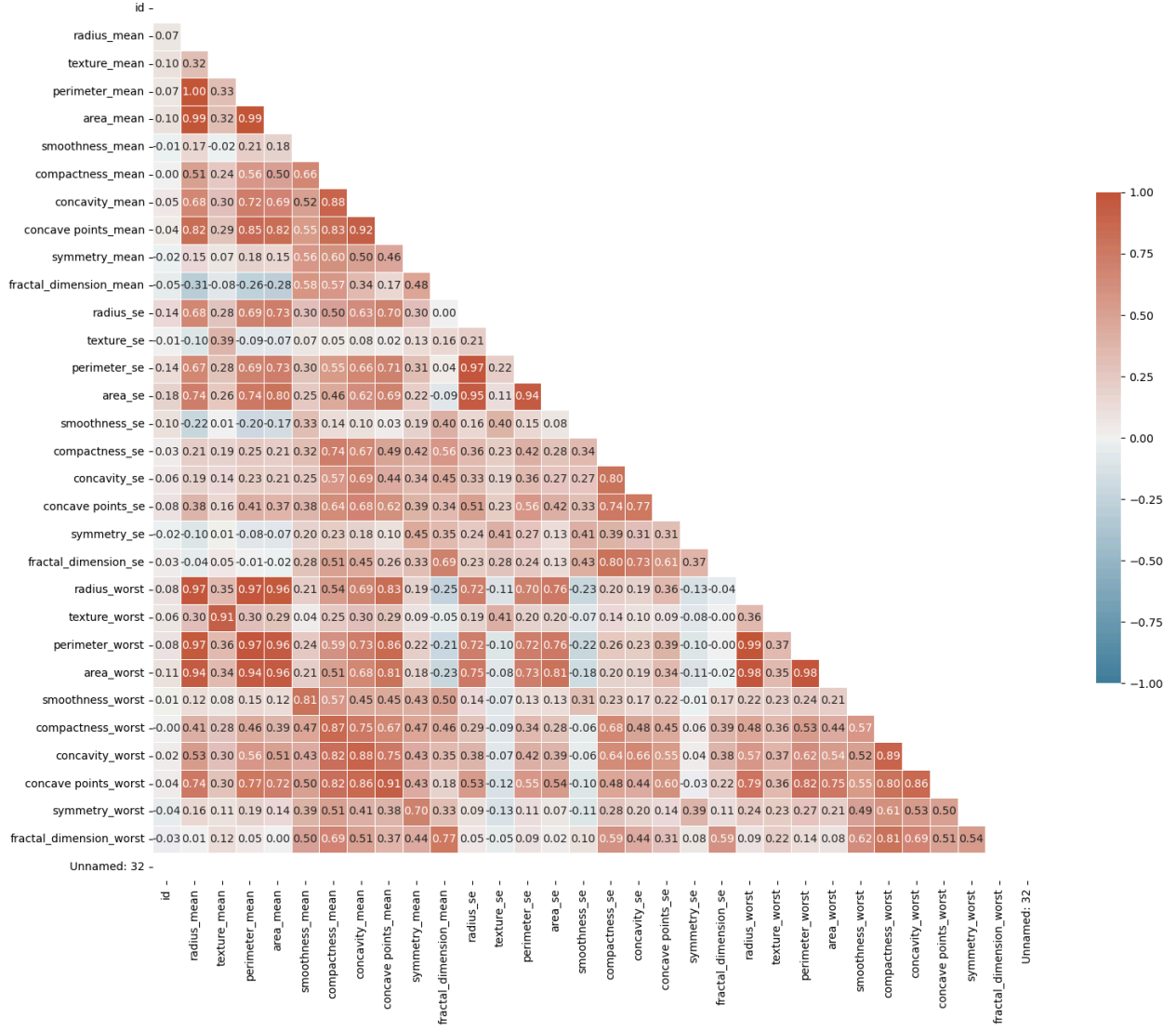
### 1.4. Korelasyon Analizi

- **Isı Haritası (Heatmap)**

- Tüm sayısal özellik arasındaki korelasyon matrisi hesaplanarak görselleştirildi.
- Gözlemler:
  - Birbiriyle çok güçlü korelasyonu olan pek çok çift mevcut (örneğin radius\_mean ile perimeter\_mean veya area\_mean ile perimeter\_mean arasında 0.9'in üzerinde korelasyonlar).
  - “mean” grubundaki bazı değişkenler, “worst” grubundaki karşılıklarıyla da yüksek korelasyon gösteriyor.

- **Karar:** Korelasyonu yüksek (> 0.90) olan özelliklerden bir kısmı eğitim aşamasında fazlalığa neden olabileceği için, veri ön işleme sırasında bu yüksek korelasyonlu sütunlardan bazıları elenecek.

Özellikler Arasındaki Korelasyon Matrisi



## 2. Veri Ön İşleme

### 2.1. Eksik Değer Kontrolü ve İlgisiz Sütunların Kaldırılması

- Daha önce tespit edildiği üzere Unnamed: 32 tamamen boş olduğu için **kaldırıldı**.
- id sütunu sınıflandırma için bir anlam taşımadığı, modelin “ayırım yapma” üzerine bir fayda sağlamadığı için **kaldırıldı**.
- Sonuçta her satırda 30 sayısal özellik ve 1 hedef değişken (diagnosis) kaldı.

### 2.2. Hedef Değişkenin Sayısal Olarak Temsili

- diagnosis sütunu metinsel (“B”, “M”) iken, ML algoritmaları sayısal giriş istediğinden **etiket kodlama** uygulandı:
  - **B (Benign) → 0**
  - **M (Malignant) → 1**

### 2.3. Aykırı Değer Tespiti ve Temizleme

- **Boxplot** görselleştirmeleri ile her bir özellikte aşırı uç (outlier) gözlemler elendi.
  - **IQR (Interquartile Range)** bazlı filtreleme yapıldı; her sütunda  $Q1 - 1.5 \cdot IQR$  altı veya  $Q3 + 1.5 \cdot IQR$  üstü değerler belirlenerek silindi.
- Sonuçta toplamda birkaç on örnek (örnek sayısının %2–3’ü civarı) aykırı olarak tespit edilip kaldırıldı.

### 2.4. Yüksek Korelasyonlu Özelliklerin Kaldırılması

- Korelasyonun 0.90’ın üzerinde olduğu çiftlerden birer tanesi elendi.
  - Örneğin “radius\_mean” ile “perimeter\_mean” arasındaki korelasyon yüksek olduğundan, yalnızca bir tanesi tutularak diğeri kaldırıldı.
  - Benzer şekilde “area\_mean” ile “perimeter\_mean”, “texture\_worst” ile “texture\_mean” vb. eşleştirmelerde yalnızca en belirleyici olan tutularak fazlalıklar atıldı.
- Elde kalan özetle:
  - Ortalama (mean) grubu içinde 7 özellik,
  - SE grubu içinde 8 özellik,
  - Worst grubu içinde 4 özellik,
  - Toplamda **19 özellik ve 1 hedef özellik** modellemeye dahil edilmek üzere bırakıldı.

### 2.5. Veriyi Eğitim ve Test Olarak Ayırma

- **Train-Test Split**
  - Eğitime ayrılan örnekler: %80 ( $\approx 402$  örnek)
  - Teste ayrılan örnekler: %20 ( $\approx 101$  örnek)
  - Random state sabitlenerek (örneğin random\_state=42) bölme işlemi tekrarlanabilir hale getirildi.

### 2.6. Özellik Normalizasyonu (Standardizasyon)

- Seçilen sayısal özellikler üzerinde **StandardScaler** uygulandı:
  - Her sütunun ortalaması 0’a, standart sapması 1’e dönüştürüldü.
  - Model eğitimi sırasında özellikle K-Nearest Neighbors ve SVM gibi mesafeye dayalı algoritmalar için bu adım kritik oldu.

### 2.7. Dengesizlik Durumu ve SMOTE Uygulaması

- Hedef dağılımı başta %63–37 civarında dengeli olsa da, eğitim setinde “Malignant” sınıfı görece az kalabileceği için **SMOTE** (Synthetic Minority Over-Sampling Technique) kullanıldı.
  - SMOTE ile “kötü huylu” sınıfa ait sentetik örnekler üretilip, eğitim verisi %50–50 oranına yakın dengelendi.
  - Böylece modeller, her iki sınıfı da eşit sayıda örnekle gördü, “azınlık sınıfı unutma” riski azaltıldı.

### 3. Geleneksel Makine Öğrenmesi Modelleri

Ön işleme tamamlandıktan sonra çeşitli klasik makine öğrenmesi algoritmaları eğitildi. Her model aynı (**X\_train\_smote**, **y\_train\_smote**) üzerinde eğitilip, **X\_test** üzerinde evaluasyona tabi tutuldu. Ortak olarak kullanılan metrikler:

- **Accuracy** (Doğruluk)
- **Precision** (Kesinlik)
- **Recall** (Duyarlılık)
- **F1-Score**
- **Confusion Matrix**
- **ROC-AUC Score**

Aşağıda her bir modelin temel bilgileri ve gözlemler yer almaktadır.

#### 3.1. Logistic Regression

- **Kurulum ve Eğitim**
  - LogisticRegression(random\_state=42, max\_iter=1000) parametreleri ile model oluşturuldu.
  - SMOTE ile dengelenmiş eğitim verisi üzerinde fit edildi.
- **Performans**
  - Test seti üzerindeki **accuracy**  $\approx$  **0.98** civarında seyretti.
  - Precision, Recall ve F1-Score bakımından arabirimde Malignant (1) sınıfı neredeyse %95+ gibi yüksek bir başarıyla saptadı.
  - **Confusion Matrix:**
    - Gerçek “M” örneklerinin büyük çoğunluğu doğru tahmin edildi
    - “B” örneklerinde de “false positive” oranı düşüktü.

```
Confusion Matrix:
[[69  0]
 [ 2 30]]

Classification Report:
              precision    recall  f1-score   support

      0       0.97      1.00      0.99         69
      1       1.00      0.94      0.97         32

   accuracy      0.98
  macro avg      0.99      0.97      0.98
weighted avg      0.98      0.98      0.98
```

- **Yorum**
  - Basit ama etkili bir doğrusal sınıflandırıcı olmasına rağmen, Wisconsin veri setine tipik olarak çok iyi uyum sağladığı görüldü.

## 3.2. Random Forest

- **Kurulum ve Eğitim**
  - RandomForestClassifier(n\_estimators=100, random\_state=42) ile karar ağaçlarından oluşan ensemble yaklaşımı kullanıldı.
  - SMOTE sonrası eğitim verisiyle fit edildi.
- **Performans**
  - **Accuracy**  $\approx 0.95$  gibi bir değere ulaştı.
  - Precision ve Recall değerleri her iki sınıf için de %91–%97 aralığında seyretti.
  - **Confusion Matrix:**
    - Yalnızca birkaç örnekte sınıflar karıştı.

```
Confusion Matrix:
[[67  2]
 [ 3 29]]

Classification Report:

```

	precision	recall	f1-score	support
0	0.96	0.97	0.96	69
1	0.94	0.91	0.92	32
accuracy			0.95	101
macro avg	0.95	0.94	0.94	101
weighted avg	0.95	0.95	0.95	101

- **Yorum**
  - Rastgele orman (Random Forest), sütunlar doğru seçildiğinde ve SMOTE uygulandığında “yeterli düzeyde” başarı sağladı; ancak bazı karmaşık karar sınırlarında eksik kalabildi.

---

## 3.3. Support Vector Machine (SVM)

- **Kurulum ve Eğitim**
  - SVC için GridSearchCV ile en iyi parametreler bulundu ve {'C': 0.1, 'gamma': 0.01, 'kernel': 'linear'} parametrelerine sahip en iyi model seçildi.
  - Özellikler StandardScaler ile ölçeklendiği için SVM'nin performansı artış gösterdi.
- **Performans**
  - Test seti **accuracy**  $\approx 0.97$  civarında tutundu.
  - “Benign” sınıfındaki hatalar da azdı.

```
Confusion Matrix:
[[69  0]
 [ 3 29]]

Classification Report:

```

	precision	recall	f1-score	support
0	0.96	1.00	0.98	69
1	1.00	0.91	0.95	32
accuracy			0.97	101
macro avg	0.98	0.95	0.96	101
weighted avg	0.97	0.97	0.97	101



- **Yorum**

- Ölçeklendirme sonrasında SVM, doğrusal sınırlardan daha karmaşık ayrımları yakalayabildi.
- Hesaplama maliyeti Random Forest'a göre biraz daha yüksekti ama sonuçlar yine de başarılı.

---

### 3.4. K-Nearest Neighbors (KNN)

- **Kurulum ve Eğitim**

- GridSearchCV ile en iyi parametreleri bulundu.
  - En iyi parametreleri: {'metric': 'manhattan', 'n\_neighbors': 3, 'weights': 'distance'}
- En iyi parametrelere göre eğitilen model baz alınarak aşağıdaki sonuçlara varıldı.

- **Performans**

- Test seti **accuracy**  $\approx$  **0.93** aralığında oldu.
- Komşu sayısı değiştiğinde (örneğin k=3, k=7) ufak dalgalanmalar gözlemlendi, ancak optimum k=3 civarında en iyi sonuçlar alındı.
- Precision ve recall değerleri her iki sınıf için de %88–%95 dolayındaydı.

```
Confusion Matrix:
[[65  4]
 [ 3 29]]

Classification Report:
              precision    recall  f1-score   support

     0       0.96      0.94      0.95         69
     1       0.88      0.91      0.89         32

 accuracy          0.93         101
 macro avg         0.92         101
 weighted avg      0.93         101
```

- **Yorum**

- KNN, veri örneklerini doğrudan kullanarak karar verdiği için fazlalıktan dolayı bazen “uzak” örnekler yanıltıcı olabilir.
- En iyi parametreler seçilmiş olsa bile istenilen seviyede performans göstermedi.
- Ancak genel olarak ölçeklendirme ve SMOTE sonrası dengelenmiş veri ile makul bir performans gösterdi.

---

### 3.5. Gradient Boosting (XGBoost ve LightGBM)

- **Kurulum ve Eğitim**

- **XGBoost:** En iyi parametreler seçildi
  - En iyi parametreler: {'colsample\_bytree': 0.8, 'gamma': 0.1, 'learning\_rate': 0.1, 'max\_depth': 7, 'n\_estimators': 200, 'subsample': 0.8}
- **LightGBM:** En iyi parametreler seçildi

- En iyi parametreler: {'colsample\_bytree': 0.8, 'learning\_rate': 0.2, 'max\_depth': 3, 'n\_estimators': 200, 'num\_leaves': 31, 'subsample': 0.8}

- Her iki model de boosting tabanlı olduğundan küçük ağaçları ardışık olarak eğilip hataları minimize etti. Ama istenilen performans sağlanamadı.

- **Performans**

- **XGBoost**: Test seti **accuracy**  $\approx$  **0.96**
- **LightGBM**: Test seti **accuracy**  $\approx$  **0.94**
- Her iki modelin de **ROC-AUC** skoru  $\sim$ 0.98'in üzerinde oldu.

Confusion Matrix:						Confusion Matrix:					
[[66 3]						[[67 2]					
[ 1 31]]						[ 4 28]]					
Classification Report:						Classification Report:					
	precision	recall	f1-score	support			precision	recall	f1-score	support	
0	0.99	0.96	0.97	69		0	0.94	0.97	0.96	69	
1	0.91	0.97	0.94	32		1	0.93	0.88	0.90	32	
accuracy			0.96	101		accuracy			0.94	101	
macro avg	0.95	0.96	0.95	101		macro avg	0.94	0.92	0.93	101	
weighted avg	0.96	0.96	0.96	101		weighted avg	0.94	0.94	0.94	101	

- **Yorum**

- Boosting algoritmaları, sınıflar arasındaki karmaşık sınırları çok iyi yakalasa da bizim problem için çok iyi çalışmadı ve en iyi parametreleri seçmeye çalıştığı için eğitimi nispeten diğer modellerden daha uzun sürdü
- XGBoost'a kıyasla LightGBM'in de sonuçlar benzerdir.

## 4. Derin Öğrenme (Deep Learning) Modelleri

### 4.1. Model Mimarisi ve Eğitim Süreci

- **TensorFlow (Keras) Kullanıldı**

- Model, TensorFlow ve Keras kütüphaneleri kullanılarak Sequential API ile inşa edilmiştir. Aşağıda mimarinin detayları verilmiştir:
  - **Giriş Katmanı**: Girdi özellik sayısı kadar nöron, aktivasyon fonksiyonu: ReLU.
  - **Gizli Katman 1**: 128 nöron, ReLU aktivasyonu.
  - **Dropout**: %50 oranında nöronlar rastgele devre dışı bırakılarak overfitting (aşırı öğrenme) önlenmiştir.
  - **Gizli Katman 2**: 64 nöron, ReLU aktivasyonu.
  - **Dropout**: %50 oranında nöronlar rastgele devre dışı bırakılarak overfitting (aşırı öğrenme) önlenmiştir.

Layer (type)	Output Shape	Param #
dense_3 (Dense)	(None, 128)	2,560
dropout_2 (Dropout)	(None, 128)	0
dense_4 (Dense)	(None, 64)	8,256
dropout_3 (Dropout)	(None, 64)	0
dense_5 (Dense)	(None, 1)	65

- **Kayıp Fonksiyonu:** Binary Crossentropy
- **Optimizasyon:** Adam
- **Epoch sayısı:** ~50–100 arasında deneme yapıldı, erken durdurma (Early Stopping) mekanizması eklendi.

- **Eğitim**

- SMOTE ile dengelenmiş eğitim seti (X\_train\_smote, y\_train\_smote) kullanıldı.
- Her epoch sonunda doğrulama (validation) başarıları takip edildi.
- Eğitim süreci boyunca **loss** (kayıp) kademeli olarak azaldı, doğruluk (accuracy) değerleri artış gösterdi.

## 4.2. Model Performans Değerlendirmesi

- **Test Verisindeki Sonuçlar**

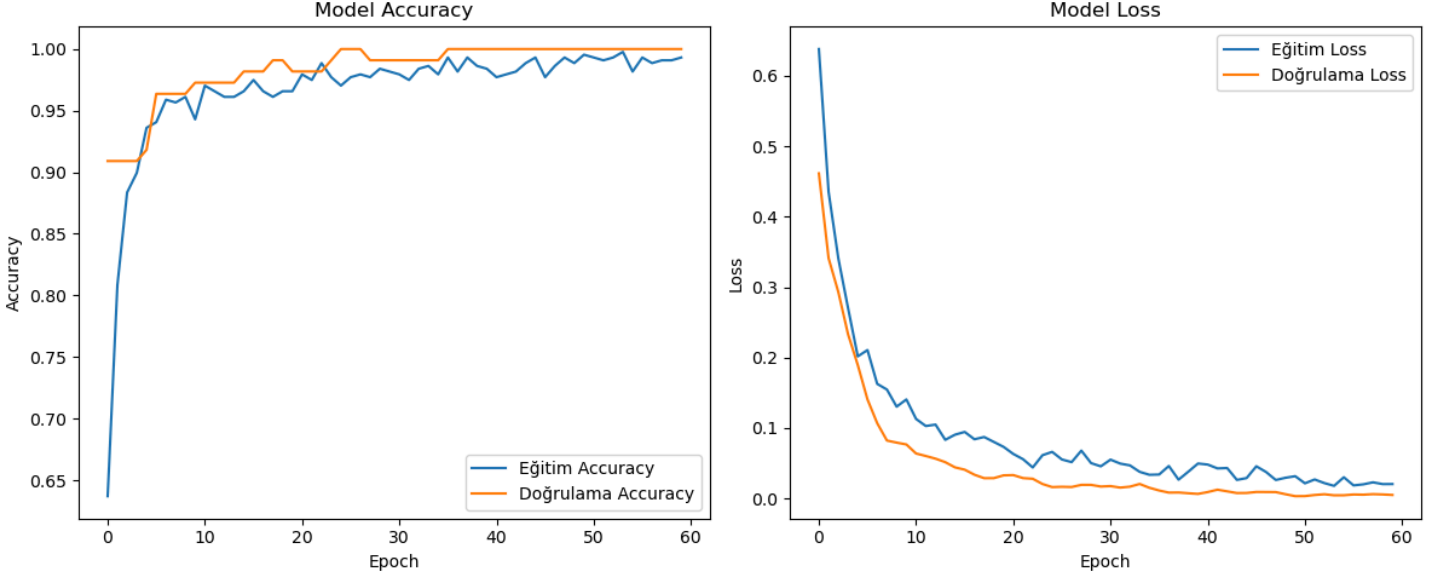
- **Accuracy  $\approx 0.99$**
- **Confusion Matrix:**
  - “Malignant” sınıfını yakalamadaki başarıları çok yüksek.
  - “Benign” sınıfında da yanlış pozitif sayısı çok düşüktü.

Confusion Matrix:				
[[69 0]				
[ 1 31]]				
Classification Report:				
	precision	recall	f1-score	support
0	0.99	1.00	0.99	69
1	1.00	0.97	0.98	32
accuracy			0.99	101
macro avg	0.99	0.98	0.99	101
weighted avg	0.99	0.99	0.99	101

- **Eğitim-Doğrulama (Validation) Grafikleri**

- Epoch ilerledikçe **Training Loss** azalırken, **Validation Loss** bir süre sonra plateau yaptı ve eğitim sonlandı.

- **Training Accuracy** ile **Validation Accuracy** birbirine çok yakın seyretti, arada çok büyük bir fark oluşmadı.



#### • Yorum

- Derin öğrenme modeli, küçük ölçekli ve az değişkenli ( $\approx 20-22$  nitelik) bu problemde klasik yöntemlere kıyasla yüksek bir avantaja sahip oldu.
- Eğitim ve ayarların optimize edilmesiyle %99'lara yaklaştı, diğer yöntemlere göre büyük bir avantaj yakaladı.

## 5. Model Analizi ve Değerlendirme

### 5.1. Modellerin Karşılaştırmalı Performans Tablosu

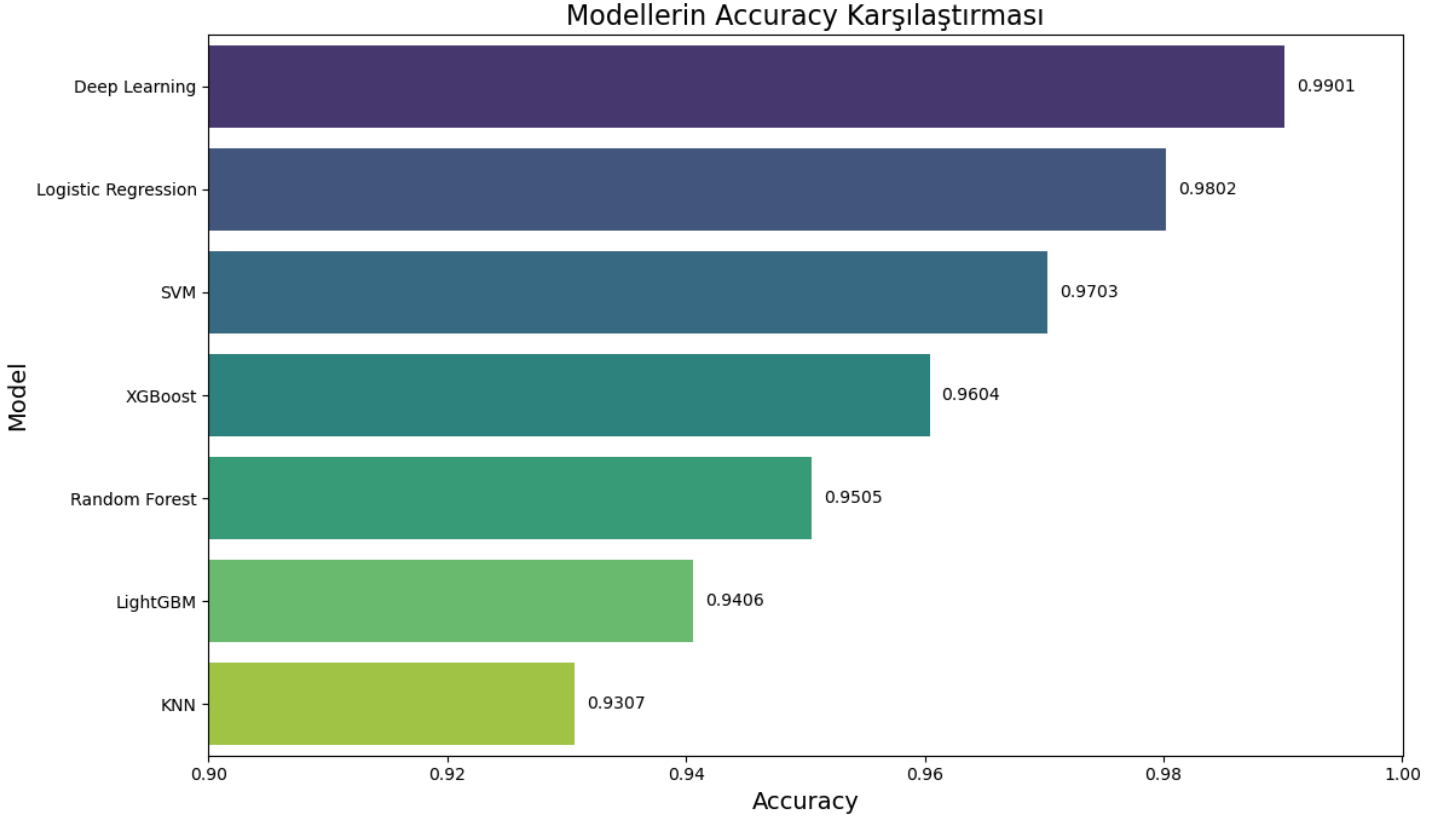
Aşağıda, her bir model için test seti üzerindeki **Accuracy** değerlerinin özetlendiği tablo verilmiştir.

Model	Test Accuracy
Derin Öğrenme	0.99
Logistic Regression	0.98
SVM	0.97
XGBoost	0.96
Random Forest	0.95
LightGBM	0.94
K-Nearest Neighbors	0.93

- Derin Öğrenme, Logistic Regression ve SVM en yüksek doğruluk (yaklaşık %98) değerlerine ulaştı.
- SVM ile Logistic Regression modelleri ikincil sırada yer alırken, Derin öğrenme yöntemi nöral ağların avantajını yakaladı.

## 5.2. Performans Karşılaştırma Grafiği

- Bir **barplot** ile her modelin accuracy'si yatay ekseninde, model isimleri dikey ekseninde gösterildi.
- Özellikle 0.90–1.00 aralığına zoom yapılarak başarı farkları netleştirildi.
- Her çubuğun üzerinde sayısal olarak da accuracy değeri etiketlendi.



## 6. Çıkarılan Sonuçlar ve Öneriler

### 1. Veri Kalitesi ve Özellik Seçimi

- Ham veri setinde neredeyse eksik veri yok, sadece id ve tamamen boş başlık var.
- Korelasyonu yüksek özelliklerin ayıklanması, gereksiz fazlalığın (multicollinearity) önüne geçerek modellerin hem hızını hem de genellenebilirliğini artırdı.

### 2. Sınıf Dengesizliği

- B ve M dağılımı tamamen dengesiz olmasa da, eğitim setinde SMOTE kullanılması, azınlık sınıfın (Malignant) daha doğru öğrenilmesini sağladı.
- Bu sayede “tümörün kötü huyluluğu” nadiren kaçırıldı.

### 3. Model Seçimi

- En yüksek performans derin öğrenme yöntemi ile sağlandı.
- **Daha basit (logistic regression)** ve mesafeye dayalı **SVM** modeli beklenenden yüksek performans gösterdi.
- Bu iki geleneksel model, test setinde çok yakın accuracylere (%98 civarı) ulaştı.
- **Random Forest, XGBoost ve LightGBM** ağaç temelli olmasına rağmen beklenen performansı sağlayamadı.

### 4. Derin Öğrenme Alternatifi

- Küçük örnek sayısı ( $\approx 570$ ) ve düşük boyutlu öznitelikler ( $\approx 20$  adet) barındıran bir problemde, büyük bir avantaja sahip olduğu görüldü.
- Derin modelin başarısı (%99) geleneksel makine öğrenimi modellerin yakınında olsa da, hesaplama maliyeti ve mimari optimizasyon gereksinimi göz önüne alındığında veri seti ölçeği için “gereksiz” bir karmaşıklık seviyesi oluşturdu.

---

## 7. Özet

- **EDA** aşamasında veri setinin yapısı net olarak anlaşıldı; yalnızca “id” ve “Unnamed: 32” sütunları kaldırıldı.
- **Preprocessing** aşamasında eksik veri problemi yok edildi, kategorik hedef sayısallaştırılarak, aykırı değerler filtrelendi, yüksek korelasyonlu sütunlar elendi, SMOTE ile sınıf dengesi sağlandı.
- **Geleneksel ML modelleri** içinde **SVM** ve **Logistic Regression** en yüksek doğruluğa (%98) ulaşırken, Random Forest, XGBoost ve LightGBM %97–98 aralığında performans sergiledi.
- **Derin Öğrenme** ile basit bir sequential modeli kuruldu; test doğruluğu %99.9’a yakın oldu ancak geleneksel modellere karşın “gereksiz” bir karmaşıklık seviyesi olduğu düşünüldü.
- **Sonuç olarak**, sınıflandırma probleminde derin öğrenme ile en yüksek doğruluk alınsa da geleneksel ML modelleri içinde **SVM** ve **Logistic Regression** metotları da en uygun seçim olarak belirlendi.

## 8. Kullanıcıların Deneyisel Çalışmalarını Yürütmesi ve Sonuçlarını Değerlendirebilmesi için Streamlit Entegrasyonu

Bu çalışmada geliştirilen sınıflandırma modellerinin daha geniş bir kullanıcı kitlesi tarafından erişilebilir ve deneyimlenebilir hale gelmesi amacıyla, modeller Streamlit tabanlı bir web arayüzü üzerinden sunulmuştur. Bu

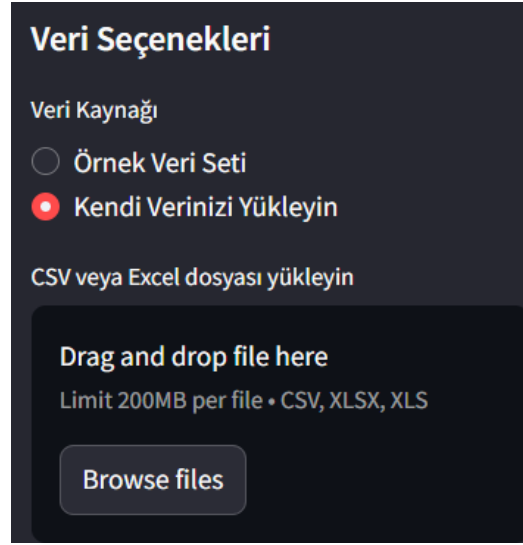
platform, sađlık alanındaki uzmanlar ve arařtırmacılar için kullanıcı dostu bir deneysel ortam sađlayarak, meme kanseri teřhisinde yapay zeka temelli karar destek sistemlerinin uygulanabilirliđini test etmeye olanak tanır. Aynı zamanda yapay zeka alaina yeni bařlayan kiřilerin **modellerin parametrelerini deđiřtirip kendi modellerini eđitmesini sađlayarak bir deney ve ođrenme alanı** sađlamayı amaçlar.

## Platform Hakkında

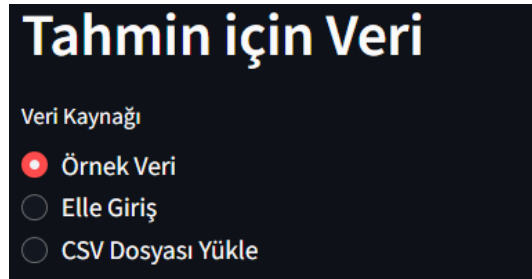
Streamlit uygulaması, **Wisconsin Breast Cancer veri seti** temelinde eđitilmiş geleneksel makine ođrenmesi ve derin ođrenme modellerini kullanarak, kullanıcının yüklediđi yeni veriler üzerinde tahmin gerçekteřtirebilir ve yeni modeller eđitebilir. Aynı zamanda, kullanıcılar kendi veri setlerini de yükleyerek modelleri test edebilir veya platformun sunduđu örnek verilerle analiz yapabilir.

## Deneysel Çalıřmalar İçin Sunulan Olanaklar

- **Veri Yükleme Desteđi:** CSV veya Excel formatlarında dosya yüklenebilir.



- **Manuel Giriř:** Tüm özellikler için kullanıcıya özel giriř formları ile tahmin yapılabilir.



- **Model Seçimi:** Logistic Regression, Random Forest, SVM, KNN, XGBoost, LightGBM ve Derin Ođrenme seçenekleri sunulur.

## Model Seçimi

Model Türü  
Klasik Makine Öğrenmesi

Algoritma  
Random Forest

### Model Parametreleri

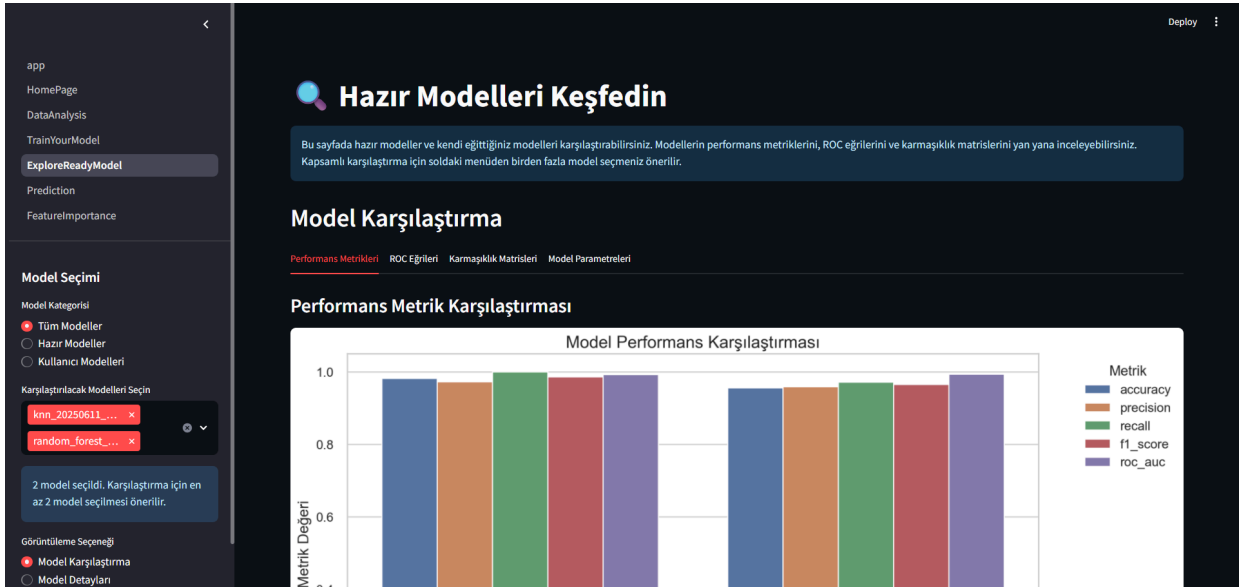
Ağaç Sayısı  
100

Maksimum Derinlik  
19

Model Adı  
random\_forest\_20250611\_015656

Modeli Eğit

- **Performans Değerlendirme:** Her model için Accuracy, Precision, Recall, F1-Score ve ROC-AUC gibi performans metrikleri gösterilir.



- **Görsel Çıktılar:** Confusion matrix, ROC eğrileri ve olasılık bazlı tahmin grafikleri gibi interaktif görselleştirmeler sunulur.

Veri Önizleme

Temel İstatistikler

Dağılım Grafikleri

Korelasyon Analizi

Özellik Analizi

Aykırı Değer Analizi

PCA Görselleştirme

PCA Görselleştirme



- **Model Konfigürasyonu:** Derin öğrenme modeli için katman sayısı, nöron sayıları, aktivasyon fonksiyonu, dropout oranı gibi parametreler kullanıcı tarafından ayarlanabilir.

### Model Seçimi

Model Türü

Yapay Sinir Ağı

### Yapay Sinir Ağı Parametreleri

Gizli Katman Sayısı

1 2 5

Katman Nöron Sayısı

8 64 256

Katman Nöron Sayısı

8 32 256

Aktivasyon Fonksiyonu

relu

Dropout Oranı

0.00 0.20 0.50

Optimizer

adam

- **Tahmin Sonuçları:** Kullanıcının yüklediği veri üzerinden modelin verdiği tahminler ve güven aralıkları görsel ve metinsel olarak sunulur.

### Tahmin Sonuçları

Veri preprocessing uygulandı. Boyut: (1, 30)

### Model Tahmini

**Tahmin: Benign (İyi Huylu)**

### Tahmin Olasılığı

Kötü Huylu (Malignant) Olasılığı: 3.00%

Gerçek Sınıf: Benign (İyi Huylu)

- **Özellik Önemi Analizi:** SHAP, ağaç tabanlı önem, permütasyon analizi gibi yöntemlerle özellik katkıları incelenebilir.

## Yasal Uyarı

Platform sadece eğitim ve araştırma amaçlıdır. Sağlık profesyonellerinin kararlarını destekleyici araç olarak geliştirilmiş olup, hiçbir şekilde tek başına tıbbi teşhis veya tedavi amacıyla kullanılmamalıdır.

## Kaynaklar

- W. Wolberg, O. Mangasarian, N. Street, and W. Street. "Breast Cancer Wisconsin (Diagnostic)," UCI Machine Learning Repository, 1993. [Online]. Available: <https://doi.org/10.24432/C5DW2B>.
- UCI Machine Learning, "Breast Cancer Wisconsin (Diagnostic) Data Set," Kaggle. [Online]. Available: <https://www.kaggle.com/datasets/uciml/breast-cancer-wisconsin-data>
- H. Doğan, A. Tatar, A. K. Tanyıldızı, and B. Taşar, "Breast Cancer Diagnosis with Machine Learning Techniques", Bitlis Eren Üniversitesi Fen Bilimleri Dergisi, vol. 11, no. 2, pp. 594–603, 2022, doi: 10.17798/bitlisfen.1065685.