

## 1. Giriş (Projenin Amacı Nedir?)

Bu projenin amacı, CIFAR-10 veri setini kullanarak çeşitli makine öğrenimi ve derin öğrenme modellerinin performansını değerlendirmektir. CIFAR-10, 10 farklı sınıfa ait 60.000 renkli görüntü içeren popüler bir veri setidir. Bu çalışmada, çeşitli modellerin doğruluk, precision, recall, f1-score ve ROC AUC skorları gibi metrikler kullanılarak performansları karşılaştırılacaktır.

## 2. Materyal-Metod (Kullanılan Yöntemler Hakkında Bilgi)

Bu projede kullanılan yöntemler şunlardır:

- Veri Ön İşleme: Görüntü verilerinin normalize edilmesi ve etiketlerin binarize edilmesi.
- Modeller: Evrişimli Sinir Ağı(CNN), K-En Yakın Komşular(KNN), Lojistik Regresyon(Logistic Regression), Karar ağaçları(Decision Tree), Rastgele Orman(Random Forest), LightGBM, XGBoost, CatBoost, Gradient Boosting Machine(GBM) ve Destek Vektör Makineleri (SVM).
- Değerlendirme Metrikleri: Doğruluk (accuracy), precision, recall, f1-score, ROC AUC skoru, sınıflandırma raporu, karışıklık matrisi ve ROC eğrisi.

## 3. Veri Seti Hakkında Bilgi

CIFAR-10 veri seti, 32x32 piksel boyutunda ve 10 farklı sınıfa ait 60.000 renkli görüntüden oluşmaktadır. Veri seti, 50.000 eğitim ve 10.000 test görüntüsünden oluşur. Sınıflar şunlardır: uçak, araba, kuş, kedi, geyik, köpek, kurbağa, at, gemi ve kamyon.

## 4. Deneyde Kullanılan Modeller/Mimariler

Projede kullanılan modeller ve mimariler şunlardır:

- **Evrişimli Sinir Ağları (CNN):** Görüntü verileri üzerinde etkili olan derin öğrenme modeli.
- **K-En Yakın Komşular (KNN):** Veri noktalarının sınıflandırılması için en yakın komşularını kullanan basit ve etkili bir algoritma.
- **Lojistik Regresyon (Logistic Regression):** İkili sınıflandırma problemleri için kullanılan istatistiksel bir model.
- **Karar Ağacı (Decision Tree):** Temel bir sınıflandırma algoritması.
- **Rastgele Orman (Random Forest):** Birden fazla karar ağacının birlikte kullanıldığı bir topluluk (ensemble) yöntemi.

- **LightGBM:** Büyük veri kümelerinde hızlı ve etkili olan bir gradient boosting framework'ü.
- **XGBoost:** Performans ve hız için optimize edilmiş, gradient boosting algoritmasına dayalı bir kütüphane.
- **CatBoost:** Kategorik verilerle daha iyi performans gösteren gradient boosting algoritması.
- **Gradient Boosting Machine (GBM):** Hata oranını minimize etmek için tahmin modelleri oluşturan güçlü bir topluluk yöntemi.
- **Destek Vektör Makineleri (SVM):** Veri noktalarını sınıflandırmak için hiper düzlemler kullanan güçlü bir sınıflandırıcı.

## 5. Deney Sonuçları

Deney sonuçları aşağıda özetlenmiştir:

### - Evrişimli Sinir Ağları (CNN):

- Accuracy(Doğruluk): %71
- Precision(Kesinlik): %72
- Recall(Geri Çağırma): %71
- F1-Score: %71
- ROC AUC: %96
- Karmaşıklık Matrisi ve ROC Eğrisi grafiklerle gösterilmiştir.

### - K-En Yakın Komşular(KNN):

- Accuracy(Doğruluk): %34
- Precision(Kesinlik): %46
- Recall(Geri Çağırma): %32
- F1-Score: %33
- Karmaşıklık Matrisi grafikte gösterilmiştir.

### - Lojistik Regresyon (Logistic Regression):

- Accuracy(Doğruluk): %37
- Precision(Kesinlik): %37
- Recall(Geri Çağırma): %37

- F1-Score: %37
- ROC AUC: %79
- Karmaşıklık Matrisi grafikte gösterilmiştir.

**- Decision Tree:**

- Accuracy(Doğruluk): %27
- Precision(Kesinlik): %27
- Recall(Geri Çağırma): %27
- F1-Score: %27
- ROC AUC: %60
- Karmaşıklık Matrisi grafikte gösterilmiştir.

**- Random Forest:**

- Accuracy(Doğruluk): %48
- Precision(Kesinlik): %48
- Recall(Geri Çağırma): %48
- F1-Score: %48
- ROC AUC: %86
- Karmaşıklık Matrisi grafikte gösterilmiştir.

**- LightGBM:**

- Accuracy(Doğruluk): %53
- Precision(Kesinlik): %53
- Recall(Geri Çağırma): %53
- F1-Score: %53
- ROC AUC: %90
- Karmaşıklık Matrisi grafikte gösterilmiştir.

**- XGBoost:**

- Accuracy(Doğruluk): %54

- Precision(Kesinlik): %54
- Recall(Geri Çağırma): %54
- F1-Score: %54
- ROC AUC: %90
- Karmaşıklık Matrisi grafikte gösterilmiştir.

**- CatBoost:**

- Accuracy(Doğruluk): %45
- Precision(Kesinlik): %44
- Recall(Geri Çağırma): %45
- F1-Score: %45
- ROC AUC: %86
- Karmaşıklık Matrisi grafikte gösterilmiştir.

**- Gradient Boosting Machine (GBM):**

- Accuracy(Doğruluk): %34
- Precision(Kesinlik): %34
- Recall(Geri Çağırma): %34
- F1-Score: %34
- ROC AUC: %75
- Karmaşıklık Matrisi grafikte gösterilmiştir.

**- Support Vector Machine (SVM):**

- Accuracy(Doğruluk): %54
- Precision(Kesinlik): %54
- Recall(Geri Çağırma): %54
- F1-Score: %54
- ROC AUC: %90
- Karmaşıklık Matrisi ve ROC Eğrisi grafiklerle gösterilmiştir.

## 6. Tartışma

Deney sonuçlarına göre, kullanılan modellerin performansları karşılaştırılmıştır. CNN, özellikle görüntü verileri üzerinde en yüksek performansı göstermiştir. SVM, yüksek doğruluk oranına sahip olmasına rağmen hesaplama maliyetleri daha yüksektir. Random Forest, karar ağacına göre daha iyi performans göstermiştir çünkü topluluk yöntemleri genellikle tek başına kullanılan modellerden daha iyidir. Ancak, Random Forest ve Decision Tree modellerinin ROC AUC skorları, CNN ve SVM modellerinden daha düşüktür.

### Eksiklikler:

Model performansları yalnızca doğruluk ve benzeri metriklerle sınırlıdır. Eğitim süresi ve hesaplama maliyetleri de dikkate alınmalıdır.

Veri seti küçültülerek kullanılmıştır, bu da sonuçların genelleştirilebilirliğini azaltabilir.

### Evrişimli Sinir Ağları (CNN):

Doğruluk (Accuracy) ve F1-Score diğer modellere göre yüksek (%71).

ROC AUC oldukça yüksek (%96), bu da modelin sınıflandırma performansının genel olarak iyiliğini gösterir.

Grafiklerle desteklenen karmaşıklık matrisi ve ROC eğrisi sunumu, modelin sınıflandırma hatalarını ve başarılarını görsel olarak anlamamıza yardımcı olur.

### K-En Yakın Komşular (KNN):

Doğruluk ve F1-Score düşük (%34, %33), bu da modelin genel olarak zayıf performans gösterdiğini işaret eder.

Kesinlik (Precision) ve Geri Çağırma (Recall) da düşüktür (%46, %32).

### Lojistik Regresyon:

Diğer modellere kıyasla ortalama performans gösteriyor (%37 doğruluk, precision, recall ve F1-Score).

ROC AUC %79, yani sınıflandırma performansı iyi ancak CNN gibi değil.

### Decision Tree:

Genel performans düşük (%27). Karmaşıklık matrisi ve ROC AUC değeri de düşük (%60), modelin sınıflandırma hatalarının belirgin olduğunu gösterir.

**Random Forest:**

Doğruluk ve F1-Score düşük (%48) ROC AUC yüksek değerde (%86)

Bu deneyde %48 doğruluk ve F1-Score ile orta seviyede bir performans sergilemiştir. Ancak, ROC AUC değeri %86 ile oldukça yüksektir, yani modelin sınıflandırma yeteneği iyi seviyededir.

**LightGBM:**

Doğruluk ve F1-Score düşük (%53) ROC AUC yüksek değerde (%90)

Deneyde %53 doğruluk ve F1-Score ile daha iyi bir performans sergilemiş ve ROC AUC değeri %90 ile oldukça yüksektir.

**XGBoost:**

Doğruluk ve F1-Score düşük (%54) ROC AUC gene yüksek değerde (%90)

Deneyde %54 doğruluk ve F1-Score ile en iyi performans gösteren model olarak öne çıkmıştır. ROC AUC değeri de %90 ile LightGBM ile benzerdir, yani sınıflandırma performansı oldukça yüksektir.

**CatBoost:**

Diğer ağaç tabanlı modellere benzer bir performans sergiliyor (%45 doğruluk ve F1-Score).

ROC AUC %86, bu da modelin sınıflandırma yeteneğinin orta seviyede olduğunu gösterir.

**Gradient Boosting Machine (GBM):**

Genel olarak düşük performans gösteriyor (%34). Bu, modelin veri kümesindeki karmaşıklıkları işlemeye zorlandığını gösterebilir.

**Support Vector Machine (SVM):**

Diğer en iyi performans gösteren modellerden biri (%54 doğruluk, precision, recall ve F1-Score).

ROC AUC %90, bu da SVM' nin yüksek doğrulukla sınıflandırma yapabilen bir model olduğunu gösterir.

## Genel Değerlendirme

Deney, farklı algoritmaların aynı veri kümesi üzerindeki performansını karşılaştırarak en iyi ve en kötü performans gösterenleri belirlememizi sağlar. Algoritmalar arasında doğruluk, kesinlik, geri çağırma ve F1-Score değerlerinin yanı sıra ROC AUC gibi daha genel ölçütler üzerinden değerlendirme yapılabilir. Grafiklerle desteklenen karmaşıklık matrisleri ve ROC eğrileri, modellerin sınıflandırma hatalarını ve başarılarını daha iyi anlamamıza yardımcı olur.

Sonuç olarak, deneyin eksik yanı veri kümesi boyutu, özellik seçimi gibi faktörlere daha detaylı bir analiz ve optimizasyon gerektirebilir. Ayrıca, her algoritmanın spesifik veri setlerine uyum sağlama yeteneği ve eğitim süresi gibi pratik uygulama faktörleri de göz önünde bulundurulmalıdır.

## 7. Referanslar

Bu çalışmada kullanılan yöntemler ve tekniklerle ilgili referanslar:

- **CIFAR-10 Veri Seti:** [CIFAR-10 dataset](<https://www.cs.toronto.edu/~kriz/cifar.html>)
- **Scikit-learn:** [Scikit-learn Documentation](<https://scikit-learn.org/stable/>)
- **TensorFlow:** [TensorFlow Documentation](<https://www.tensorflow.org/>)
- **Kaggle:** [Kaggle Documentation] (<https://www.kaggle.com/>)
- **GitHub:** [Github Documentation] (<https://github.com/>)
- **Keras:** [Keras Documentation] (<https://keras.io/>)
- **Matplotlib:** [Matplotlib Documentation] (<https://matplotlib.org/stable/contents.html>)
- **Seaborn:** [Seaborn Documentation] (<https://seaborn.pydata.org/>)