Grup Adı:

Trend Hunters (Trend Avcıları)

Grup Üyeleri:

22040301135	Muhammed Chreiki
22040301003	Emad Alkasabli
22040301092	Abdelrahman Abdelhalim
22040301145	Omar mokhtar abdo Boghdady

Proje Adı:

Youtube Trending Video Prediction

Github Linki:

https://github.com/SE-ABOSALIM/YT-Vids_Trending-Prediction

Kaggle Dataset Linki (Kendi Oluşturduğumuz)

https://www.kaggle.com/datasets/muhammedchreiki/youtube-video-trends-and-non-trends-dataset

Youtube Trending Video Prediction

Problem Tanımı

Projenin amacı youtube'da herhangi bir video'nun detaylarını bir modele vererek bu modelin bu video'nun istatistiklerine trende çıkabilme oranını tahmin etmektir. Bunun yanı sıra bir video'nun trende çıkması için nasıl bir metodoloji takip etmesi gerektiği hakkında bilgi verir. Örneğin trende çıkan video'ların çoğunun başlık uzunluğu 40 karakteri geçmemesi gerektiği, bir diğer örnek olarak belirli bir zaman diliminde belirli bir izlenme ve like sayısına ulaşması gerektiği biraz daha detaylandırılmış veriler verir [1]. Problem, ikili sınıflandırma (binary classification) problemidir. **Hedef Değişken:** 0: Trend Olmamış, 1: Trend Olmuş

Veri Tanımı (Veriseti yapısı)

Verisetimiz kaggle'dan alınan "Youtube Videos Trending Dataset" veriseti ile Youtube Data API v3'den çekilen verilerin birleşimi ile elde edilmiştir [2], [3]. Sonuç olarak verisetimizde Trend ve Non Trend olmak üzere etiketlenmiş bir feature'ımız bulunmaktadır. Bu da sınıflandırma için kullanılacaktır. Verisetinde tarih aralığı 2022-2025 arası şeklindedir.

Verisetinde 15 feature bulunmaktadır:

video_id, channelTitle, categoryId, publishedAt, title, description, tags, view_count, likes, comment_count, is_trending **(Hedef)**.

Birleştirme öncesi satır sayısı: **268,787** || Birleştirme sonrası satır sayısı: **390,043**

Veri Analizi (Data Analysis)

Final verisetini birleştirdikten sonra eksik (null) değer miktarı dikkatimizi çekti:

trending_date: 218,362 || tags: 120,820 || description: 59,300

kadar veri kaybı vardı. Ama bunun oluşması gayet normal bir şeydir çünkü youtube da bulunan bütün videoların tags ve description verilerin bulunması zorunlu değil bu yüzden null değer ile karşılaşırız. tending_date için non trend olan video'ların trend olma tarihi söz konusu değildir. Bu yüzden böyle değerlere karşılaşılabiliyor. Model geliştirme aşamasında bu verilerin kontrol altına alınması gerekir.

8 rows · 8 rows × 6 cols — Static Output

	categoryId	view_count	likes	comment_count	comments_disabled	is_trending
count	390043.000000	3.900430e+05	3.900430e+05	3.900430e+05	390043.000000	390043.000000
mean	20.426179	7.046562e+06	1.534973e+05	4.490697e+03	0.018662	0.440159
std	6.524139	4.896210e+07	6.717989e+05	4.198895e+04	0.135329	0.496407
min	1.000000	0.000000e+00	0.000000e+00	0.000000e+00	0.000000	0.000000
25%	19.000000	2.626010e+05	5.271000e+03	1.230000e+02	0.000000	0.000000
50%	22.000000	7.643870e+05	2.356100e+04	8.530000e+02	0.000000	0.000000
75%	24.000000	2.561359e+06	7.882900e+04	2.743000e+03	0.000000	1.000000
max	30.000000	4.168919e+09	4.337832e+07	1.048303e+07	1.000000	1.000000

Bu tabloda bazı metrikleri elde ederek sayısal verilerin davranışını inceleyebiliyoruz.

count: Bu sütunda kaç tane geçerli (NaN olmayan) veri olduğunu gösterir.

mean: Sütunun aritmetik ortalamasıdır (ortalama değer).

std: Standart sapmadır; verinin ortalamadan ne kadar dağıldığını gösterir. Yüksekse değerler çok dağılmıştır.

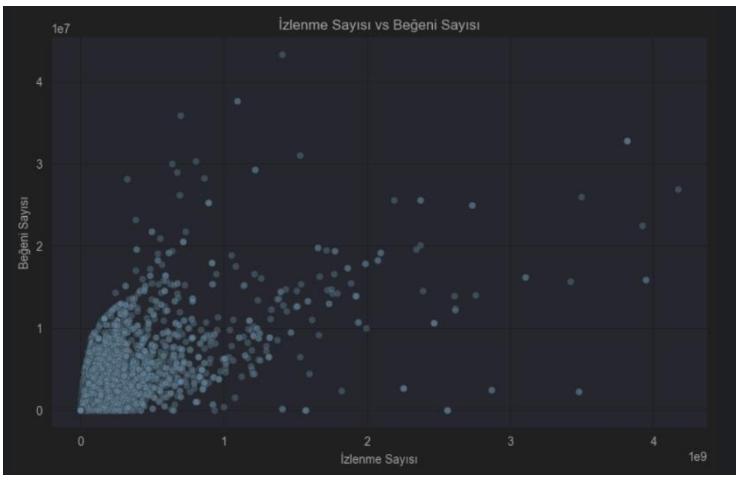
min: Sütundaki en küçük değerdir.

25%: Verinin %25'lik kısmının altında kaldığı değerdir (1. çeyrek / Q1).

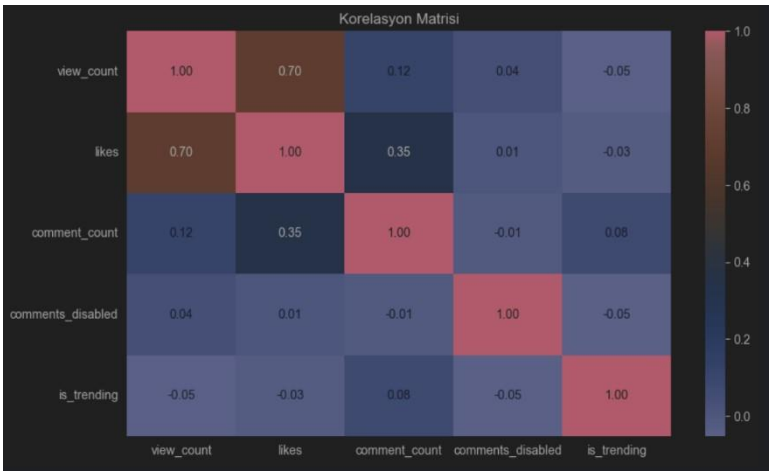
50% (median): Verinin tam ortasındaki değerdir (medyan / Q2). Ortalama gibi uç değerlerden etkilenmez.

75%: Verinin %75'lik kısmının altında kaldığı değerdir (3. çeyrek / Q3).

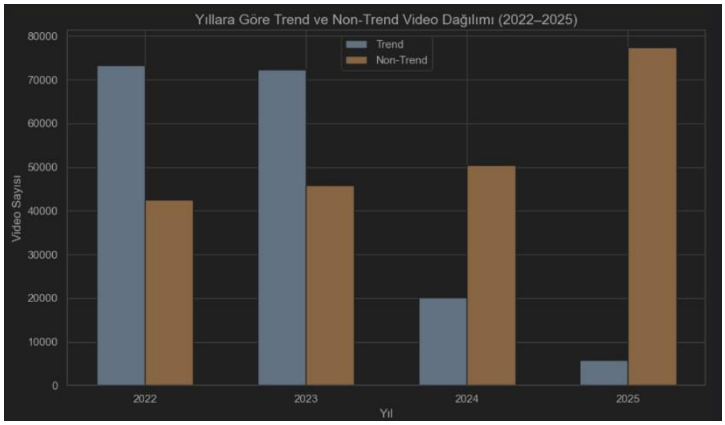
max: Sütundaki en büyük değerdir.



Şekil 1 Scatter Plot



Şekil 1.1 Correlation Matrix



Şekil 1.2 Graph

Şekil 1: Scatter plot, bu grafikte beğeni ile izlenme sayısı arasındaki ilişkiyi açıkça görebiliyoruz. Buradaki çok uçuk olan diğer noktalara göre çok yüksek değerlere sahip olan verilere Outlier (Uç) değer denir bunlar için uygun model seçilmezse veya kontrol altına alınmazlarsa modelin yapısını ve doğurduğu sonuçları olumsuz etkilerler.

Şekil 1.1: Korelasyon matrisi, sayısal değerlere sahip sütunların arasındaki ilişki güçlülüğünü gösterir.

- 1'e yakın değerler: güçlü pozitif ilişki
- 0'a yakın değerler: Neredeyse hiç ilişki yok
- -1'e yakın değerler: güçlü negatif ilişki

Şekil 1.2: Bu grafikte yıllara göre (2022-2025) arasındaki Trend ve Non-Trend video'ların dağılımı gösterilmektedir. Bu grafiğe göre dengesizliğin olduğunu görebiliyoruz ama bu sorun SMOTE veya başka bir veri çoğaltma yöntemiyle çözülebilecek bir sorundur. İlerde daha tutarlı sonuçlar elde etmek için çalışmalar son hızla devam ettirilecektir.

Veri Hazırlama

- Birleştirme öncesi publishedAt sütunundaki tarihsel değerleri bütün verisetlerinde aynı formata getirildi
- Boolean değer bulunduran sütunların sayısal (0,1) değerlere dönüştürüldü: comment_disabled
- Sınıflandırma için is_trending sütununun eklenmesi (0: daha önce trend olmamış, 1: daha önce trend sayfasında yer almış)
- Takım üyelerinin collect ettiği verilerin birleştirilmesi ve tek dataset haline getirilmesi

Veri Temizleme

- Gereksiz (Kullanılamaz) sütunlar veri setinden silinmiştir. ratings_disabled ve dislikes gibi.
- Kritik sütunlarda null değer bulunduran satırların temizlenmesi. Örneğin view_count ve likes değerleri null olan satırların modelde yeri olamaz.
- 2022 yılı öncesi bütün kayıtlar temizlenmiştir. Böylece modelde train-test split değerlerinin dengeli olması için sadece 2022-2025 yılları arasındaki veriler kullanılmıştır.
- Veri temizleme işleminden önce veri seti boyutu 268,000 satır ve 17 sütundan 390,000 satır ve 15 sütuna İndirgenmiştir.

Feature Engineering – Feature Extraction

Özellik seçimi ve özellik çıkarımı aşamalarında her bir ekip üyesi tarafından farklı bir yaklaşım benimsenmiştir. Bu da çeşitli sonuçların elde edilmesini olanak sağlamıştır. Böylece farklı özelliklerle farklı sonuçlar elde edilmiştir. Gelecek sayfalarda her bir ekip üyesi için bu bölümden bahsedilmiştir. Son olarak ta bir özet tablosu raporun sonunda bulunacaktır.

Train-Test Split

Veri seti, yayın tarihine (publishedAt) göre sıralanmış ve aşağıdaki şekilde bölünmüştür:

- Training seti: 2022-2024 yılları arasındaki videolar
- Test seti: 2025 yılında yayınlanan videolar

Bu yaklaşım, modelin gelecekteki veriler üzerinde performansını değerlendirmeyi amaçlamakta ve veri sızıntısını (data leakage) önlemektedir [4].

Toplam veri seti yaklaşık olarak:

- **%80 Training**
- **%20 Test**

oranında bölünmüştür. Bütün ekip üyeleri için aynı şekilde split yapılmıştır.

22040301135-Muhammed Chreiki

Kullanılan Basic Modeller: Logistic Regression, Catboost Classifier

Kullanılan Advanced Modeller: XGBoost, LightGBM

Basic Modeller (Vize Çalışması)

Özellik Mühendisliği (Feature Engineering)

Modelin performansını artırmak amacıyla ham veri üzerinden yeni özellikler türetilmiştir:

Tarih tabanlı feature'lar: year, month, day, hour, is_weekend

- Trend olma davranışı dönemsel değişim gösterebilir.
- Yüklenme saati YouTube trend algoritmasında önemlidir.

Metin tabanlı feature'lar: title_length, description_length, tag_count

- İçerik uzunluklarının trende etkisi araştırıldı.
- Tag sayısı özellikle YouTube için mantıklı bir sinyal.

Oran tabanlı feature'lar

- likes_ratio = likes / view_count
- comments_ratio = comment_count / view_count
- engagement = (likes + comments) / view_count

Bu oranlar, mutlak rakamların yanılttığı durumları düzeltmek için çok değerli. Bu üçlü sayesinde model, bir videonun "izlenme kalitesini" ölçüyor.

Logistic Regression:

Logistic Regression doğrusal bir modeldir. "Trend olma" ile özellikler arasındaki ilişkinin doğrusal olduğunu varsayar. Bu varsayım yüzünden bu model bunun gibi bir proje için saçmalayabilir. Çünkü YouTube etkileşimleri doğrusal olmayan, non-linear, boosting gerektiren karmaşık yapılardır. Bu yüzden bu model başarılı bir sonuç vermediğini dile getirebiliriz.

Örnek veri:
view_count=50000,
likes=2500,
comment_count=1000,
year=2025,
month=10,
day=15,
hour=20,
is_weekend=0,
title="4k nature",
description="Have fun with best 4k nature shows",
tags="nature|4k|quality|amazing",
category_id=20

Bu örnek verileri varsayarak logistic regression'un sonucuna bakarsak yanlış bir sonuç verdiğini görebiliriz.

```
Tahmin: 0
Trend olma olasılığı: %3.84
```

Catboost:

CatBoost, özellikle Youtube gibi karmaşık dijital etkileşim verilerinde mükemmel çalışan modern bir gradient boosting modelidir.

Bu gibi projeye neden uygun olduğunu şöyle sıralayabilirz:

- Non-linear ilişkileri otomatik öğrenir
- Etkileşimleri otomatik modeller
- Kategorik veriyi doğal olarak işler (Encoding gerekmez) • Büyük outlier'lara karşı daha dayanıklı

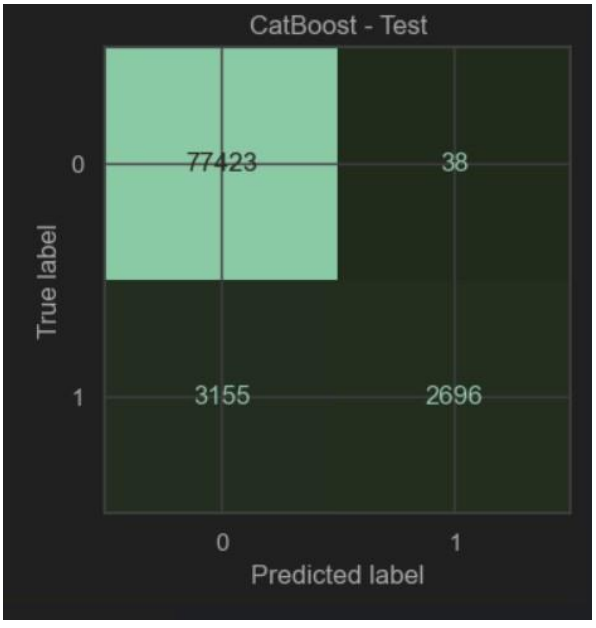
Örnek olarak aynı veriyi ele alırsak böyle bir sonuç verdiğini görebiliriz:

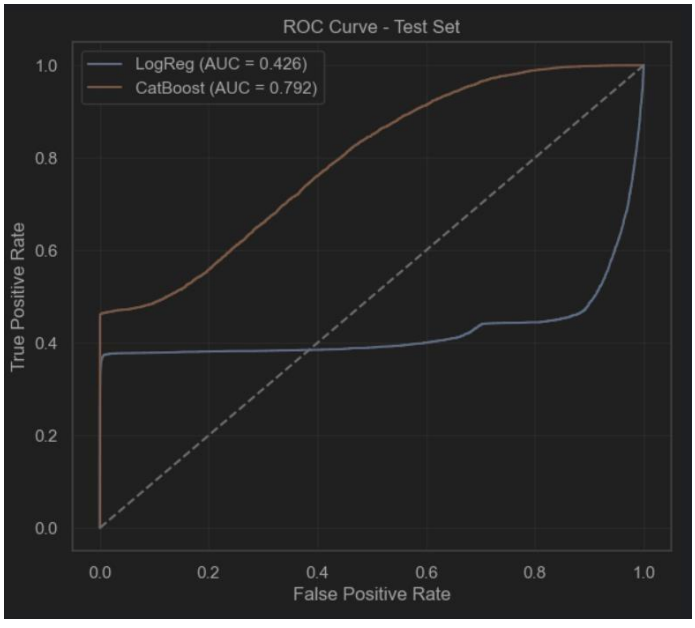
```
Tahmin: 1
Trend olma olasılığı: %97.99
```

Kısa bir süre içinde view_count=50000, likes=2500, comment_count=1000, trende çıkabilme ihtimali çok yüksektir.

Bunların sonucunda catboost modeli logistic regression'a göre daha iyi bir sonuç vermesine rağmen statik bir veriseti üzerinde modelin oluşturulduğu için sonucu ona göre verir. Gerçek zamanlı bir video'nun performansının nasıl ilerlediğini göremediği için her zaman doğru sonuç vermez.

Aşağıda olduğu gibi logistic regression ve catboost karşılaştırması gösterilmektedir. Ayrıca catboost modelinin confusion matrisini de görebiliyoruz. Confusion matrisi modelin ne kadar doğru tahmin yaptığını gösterir.





Advanced Modeller (Final Çalışması)

Bu çalışmada vize çalışmasından farklı olarak 2 advanced model kullanılmıştır. Vizede elde edilen accuracy değerlerinin %98 gibi yanıltıcı değerlerle sonuçlanmıştı. Vize çalışmasında bu kadar yüksek değerlerin elde edilmesi data leakage kaynaklanmaktadır. Bu da aslında modelin çok başarılı olduğu görülse de gerçekllikten kopmuş tahminler yapmıştır. Bu yüzden daha gerçekçi bir sonuç almak için gelişmiş model çalışmaları devam ettirilerek sonuca varılmıştır.

Feature Engineering

Bu kısımda model performansını artırmak amacıyla ham verilerden anlamlı ve ayırt edici yeni özellikler (feature engineering) türetilmiştir.

- Ham etkileşim verilerinden (likes, comment_count, view_count) oran tabanlı özellikler türetilmiştir:
  - like_view_ratio
  - comment_view_ratio
  - engagement_score
- Videonun yayın zamanının etkisini yakalayabilmek için zamansal özellikler oluşturulmuştur:
  - publish_hour
  - publish_dayofweek
  - is_weekend
- İçerik yoğunluğunu temsil etmek amacıyla metin tabanlı sayısal özetler hesaplanmıştır:
  - title_length
  - description_length
  - tag_count
- Kategorik değişkenler (categoryId), modele uygun hale getirilmek üzere sayısal forma dönüştürülmüştür.

Feature Extraction

Bu bölümde özellik çıkarımı (feature extraction) kapsamında, mevcut metin verilerinden doğrudan karmaşık vektör temsilleri (örneğin TF-IDF veya embedding tabanlı yöntemler) kullanılmamıştır. Bunun yerine;

- Metin verileri uzunluk ve sayım temelli özet özelliklere dönüştürülerek modele dahil edilmiştir.
- categoryId değişkeni için One-Hot Encoding uygulanarak kategorik bilgiler sayısal vektörlere çevrilmiştir.

Kullanılan ağaç tabanlı modellerin (XGBoost ve LightGBM) doğal feature selection yeteneği nedeniyle ek bir boyut indirgeme (PCA vb.) uygulanmamıştır.

XGBoost + Isotonic Calibration

XGBoost, gradient boosting yaklaşımı sayesinde verideki karmaşık ve non-lineer ilişkileri etkili bir şekilde modelleyebilen güçlü bir ağaç tabanlı yöntemdir [5]. Trending video sınıflandırma probleminde, etkileşim ve içerik tabanlı özellikler arasındaki ilişkileri yakalamak amacıyla tercih edilmiştir. Ayrıca, modelin ürettiği olasılık tahminlerinin daha güvenilir ve yorumlanabilir olması için Isotonic Calibration (CalibratedClassifierCV) yöntemi uygulanmıştır. Bu sayede, tahmin edilen olasılık değerleri gerçek sınıf olasılıklarına daha yakın hale getirilmiştir.

Seçilen optimum hiperparametreler:

- n_estimators = 200
- max_depth = 6
- learning_rate = 0.1
- subsample = 0.8
- colsample_bytree = 0.8
- eval_metric = "logloss"
- random_state = 42

Seçilen hiperparametreler, modelin öğrenme kapasitesi ile genelleme yeteneği arasında dengeli bir yapı oluşturmak amacıyla belirlenmiştir. n_estimators = 200 ve learning_rate = 0.1 değerleri, modelin yeterli sayıda ağaçla öğrenmesini sağlarken aşırı öğrenme riskini kontrol altında tutmayı hedeflemektedir. max_depth = 6 parametresi, karar ağaçlarının çok derinleşmesini engelleyerek overfitting riskini azaltmaktadır. Ayrıca, subsample = 0.8 ve colsample_bytree = 0.8 değerleri, her ağaç için kullanılan örnek ve özellik oranını sınırlandırarak modelin daha genellenebilir olmasını sağlamaktadır. Eğitim sürecinde hata ölçütü olarak eval_metric = "logloss" kullanılmış ve deneylerin tekrarlanabilirliği için random_state = 42 sabitlenmiştir.

LightGBM + Isotonic Calibration

LightGBM için LGBMClassifier modeli kullanılmıştır. LightGBM, leaf-wise büyüme stratejisi sayesinde daha hızlı eğitim süresi sunarken yüksek performans elde edebilen bir gradient boosting yöntemidir [6]. Özellikle büyük veri setlerinde ve karmaşık feature yapılarında etkili sonuçlar üretmesi nedeniyle tercih edilmiştir. Trending video sınıflandırma probleminde, model performansını artırmak ve öğrenme sürecini verimli hale getirmek amacıyla kullanılmıştır. XGBoost modeline benzer şekilde, LightGBM modelinde de tahmin olasılıklarının kalitesini artırmak için Isotonic Calibration (CalibratedClassifierCV) uygulanmıştır [7].

Seçilen optimum hiperparametreler:

- n_estimators = 500
- learning_rate = 0.05
- num_leaves = 31
- subsample = 0.8
- colsample_bytree = 0.8

- random_state = 42
- n_jobs = -1

LightGBM modeli için belirlenen hiperparametreler, yüksek performans ile hızlı eğitim süresi arasında denge kurmayı amaçlamaktadır. n_estimators = 500 ve daha düşük bir öğrenme oranı olan learning_rate = 0.05, modelin veriyi daha kademeli ve kararlı bir şekilde öğrenmesini sağlamaktadır. num_leaves = 31, modelin karmaşıklığını kontrol ederek aşırı uyum riskini azaltırken yeterli ifade gücünü korumaktadır. Benzer şekilde, subsample = 0.8 ve colsample_bytree = 0.8 parametreleri, rastgelelik ekleyerek modelin genelleme performansını artırmaktadır. Deneylerin tutarlılığını sağlamak amacıyla random_state = 42 kullanılmış ve eğitim süreci çok çekirdekli çalışmayı destekleyecek şekilde yapılandırılmıştır.

Hyperparameter tuning

Bu çalışmada hiperparametreler, literatürde yaygın kullanılan başlangıç değerleri ve kısa deneme–yanılma (manual search) yaklaşımıyla belirlenmiştir. Amaç, model karmaşıklığını kontrol ederek overfitting riskini azaltmak ve zaman bazlı test seti üzerinde istikrarlı performans elde etmektir.

Modellerin Değerlendirilmesi

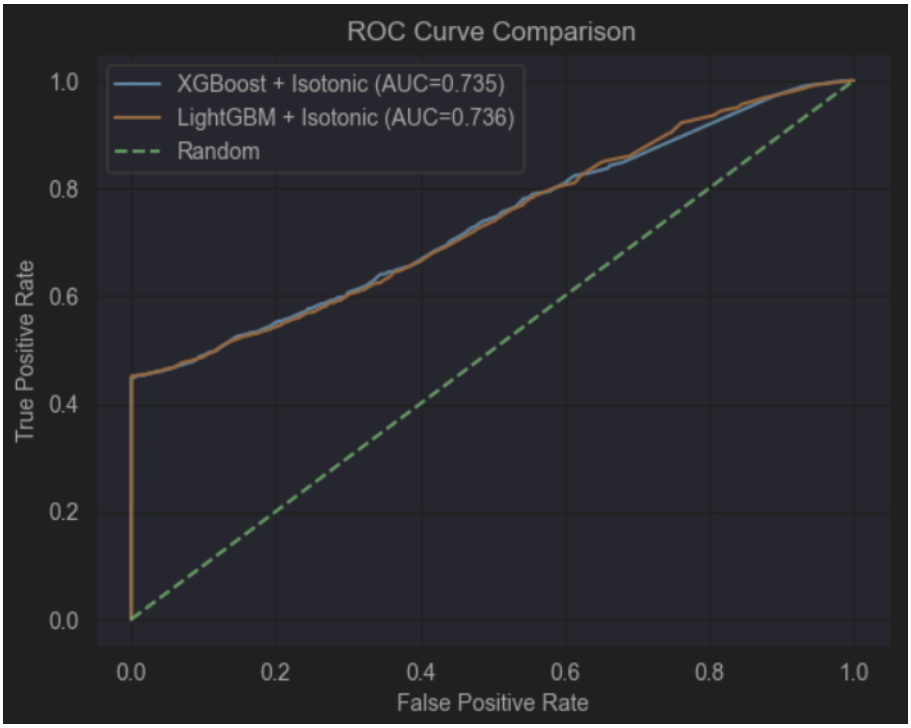
Performans Metrikleri (Vize + Final Modelleri)

Model	Balanced Accuracy	Precision	Recall	F1-Score	Roc-Auc
XGBoost	0.7226	0.9081	0.7226	0.7847	0.7353
LightGBM	0.7237	0.9103	0.7237	0.7862	0.7360
Logistic Regression	0.9700	0.9500	0.9600	0.9500	0.9600
Catboost	0.9900	0.9900	0.9900	0.9900	0.9800

Modeller, dengesiz veri setleri için uygun olan Balanced Accuracy, Precision, Recall, F1-score ve ROC-AUC metrikleri ile değerlendirilmiştir. Balanced Accuracy sınıflar arası dengeyi ölçerken, Precision ve Recall modelin trend videoları ne kadar doğru ve ne ölçüde yakalayabildiğini göstermektedir. F1-score bu iki metriğin dengeli bir özetini sunarken, ROC-AUC modelin sınıfları ayırt etme gücünü ifade etmektedir. Sonuçlar, her iki modelin benzer performans sergilediğini ve LightGBM'in çok küçük bir farkla daha iyi sonuçlar ürettiğini göstermektedir. Daha önce de anlattığımız üzere basic modellerde data leakage sebebiyle bu kadar yüksek.

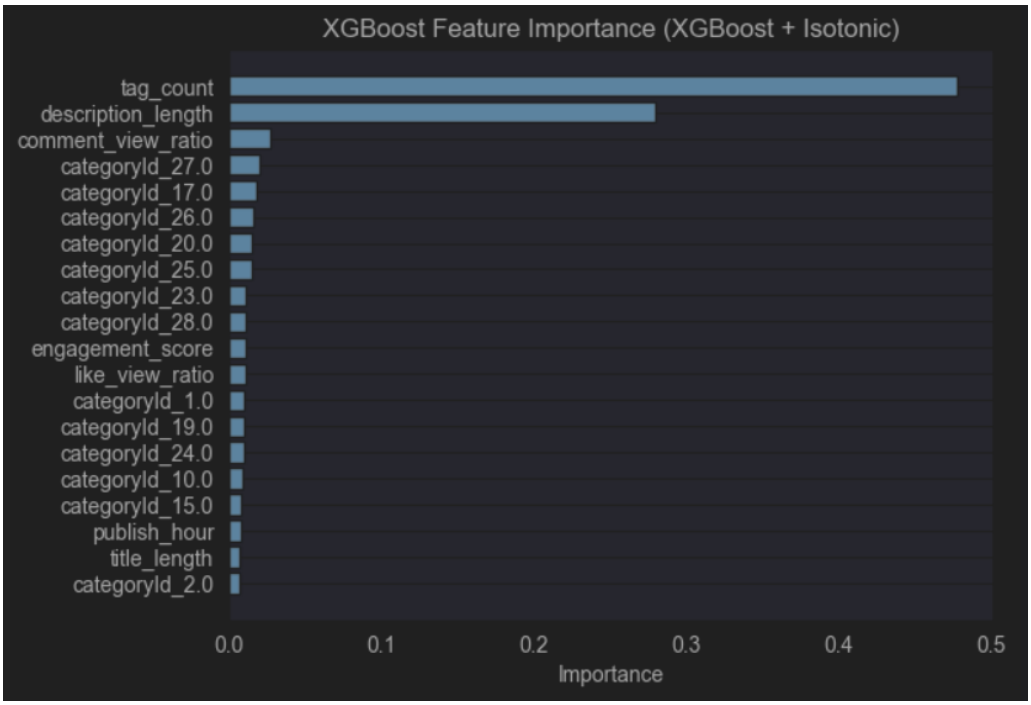
Sonuç grafikleri

ROC Curve Karşılaştırma

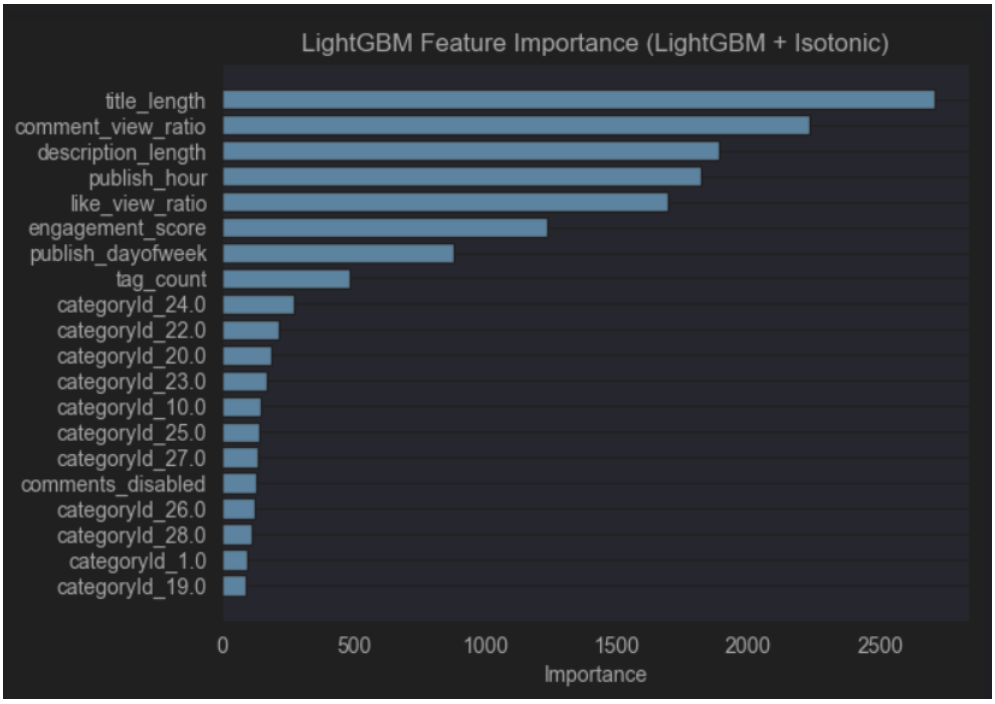


ROC eğrileri, XGBoost ve LightGBM modellerinin rastgele tahmin çizgisinin üzerinde yer aldığını ve sınıfları ayırt etme konusunda anlamlı bir performans sergilediğini göstermektedir. Her iki model de benzer ROC-AUC değerlerine sahiptir ve LightGBM modeli çok küçük bir farkla daha yüksek ayırt edicilik sağlamıştır.

Feature Importance (Özellik Önemi)



XGBoost modelinde özellik öneminin büyük ölçüde tag_count ve description_length üzerinde yoğunlaştığı görülmektedir. Bu durum, XGBoost'un bilgi kazancı yüksek özelliklere daha agresif şekilde odaklanan yapısından kaynaklanmaktadır.



LightGBM modelinde özellik önemi daha dengeli dağılmış olup, title_length, comment_view_ratio ve description_length gibi içerik ve etkileşim tabanlı özellikler öne çıkmaktadır. Bu durum, modelin farklı özellikleri birlikte değerlendirerek karar verdiğini göstermektedir.

Model Testleri

2 Örnek girdi ile test

```
user_input = {
    "view_count": 10,
    "likes": 1,
    "comment_count": 0,
    "categoryId": 22,
    "comments_disabled": 1,
    "publishedAt": "2025-01-05",
    "title": "Hello welcome to my new video we are celebrating my birthday today
#birthday #new #video",
    "description": "My birthday please follow my instagram:
https://instagram/hello.23476",
    "tags": "New|Video"
}

user_input2 = {
    "view_count": 125000,
    "likes": 8300,
    "comment_count": 410,
    "categoryId": 20,
    "comments_disabled": 0,
    "publishedAt": "2025-12-05 20:00:00",
    "title": "Best Fortnite gameplay! 50 KILL in new season #birthday #new #video",
    "description": "Amazing gameplay - Fortnite",
    "tags": "Gaming|Fortnite|New|Season12"
}
```

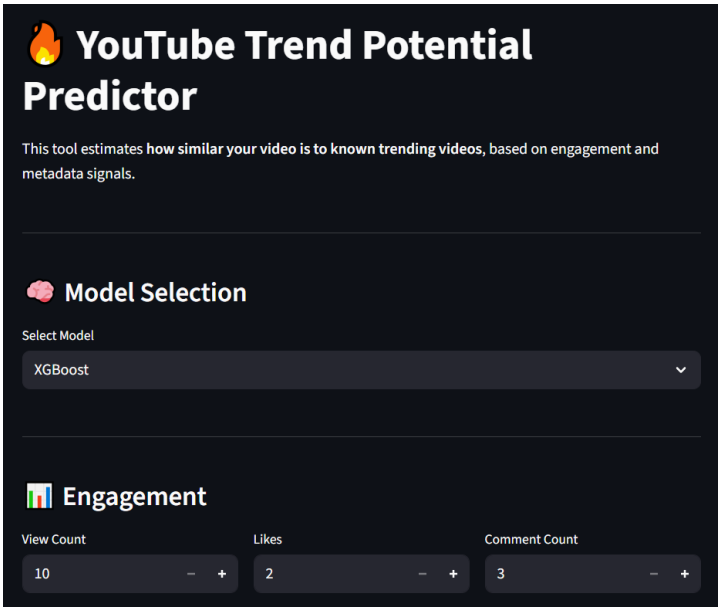
```
First Video XGB Trend Potential: %7.02
First Video LGBM Trend Potential: %11.06
Second Video XGB Trend Potential: %99.91
Second Video LGBM Trend Potential: %99.76
```

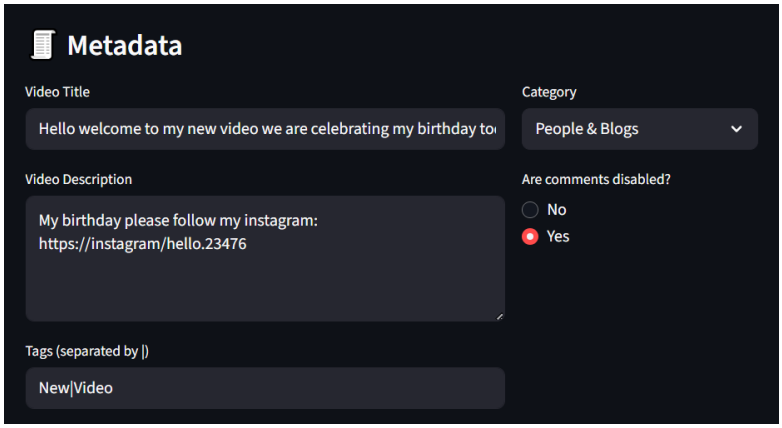
Model testleri, farklı özelliklere sahip iki video girdisi üzerinden gerçekleştirilmiştir. İlk video; düşük izlenme, beğeni ve yorum sayılarına sahip olması, yorumların kapalı olması ve daha sınırlı etkileşim sinyalleri nedeniyle her iki model tarafından da düşük trend potansiyeline sahip olarak değerlendirilmiştir. İkinci video ise daha yüksek izlenme ve etkileşim değerlerinin yanı sıra, popüler bir kategoriye (Gaming) ait olması, daha uzun ve anahtar kelime içeren bir başlığa sahip olması ve yayın zamanının uygunluğu gibi faktörler sayesinde hem XGBoost hem de LightGBM modelleri tarafından çok yüksek trend potansiyeli ile sınıflandırılmıştır. Bu sonuçlar, modellerin yalnızca etkileşim sayılarını değil; kategori, içerik ve zamansal özellikleri birlikte değerlendirerek tutarlı tahminler ürettiğini göstermektedir.

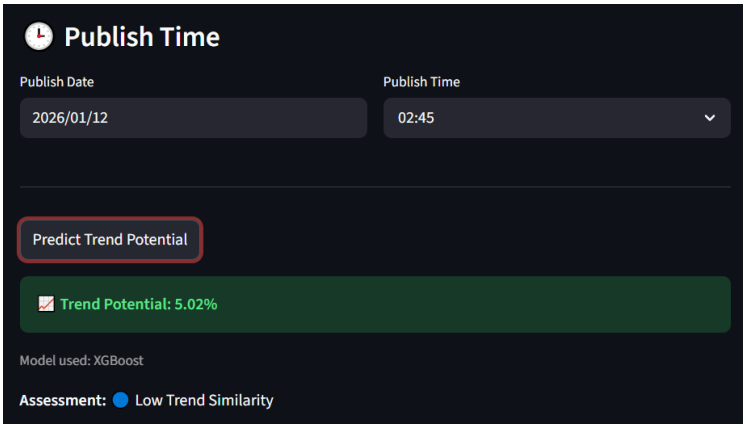
GUI ile test

Ayrıca modelleri test etmek için bir Graphical User Interface (Grafik kullanıcı arayüzü) tasarlanmıştır. Böylece test sürecini daha kolay ve gerçekçi bir hale getirilmesi sağlanmıştır.

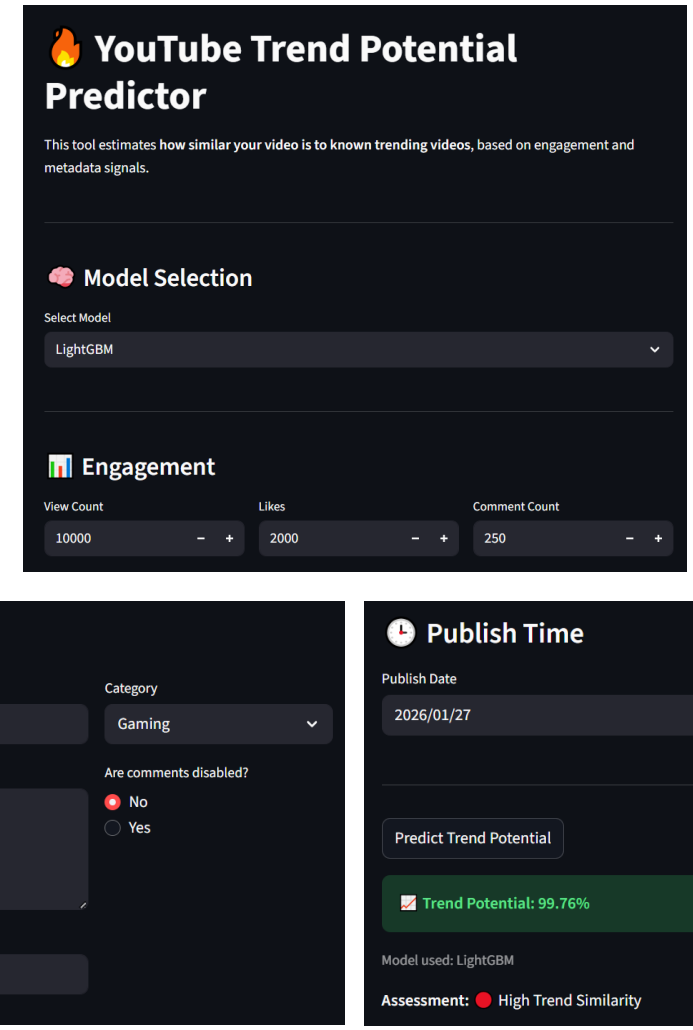
XGBoost ile Düşük Olasılık Örneği







LightGBM ile Düşük Olasılık Örneği



Sonuç ve Değerlendirme

Bu çalışmada, YouTube videolarının trend olma potansiyelini tahmin etmek amacıyla etkileşim, içerik ve zamansal özelliklere dayalı bir ikili sınıflandırma modeli geliştirilmiştir. XGBoost ve LightGBM olmak üzere iki farklı gradient boosting tabanlı model kullanılmış ve her iki modelde de tahmin olasılıklarının daha güvenilir hale getirilmesi için Isotonic Calibration yöntemi uygulanmıştır. Dengesiz veri yapısı göz önünde bulundurularak, model performansı Balanced Accuracy, Precision, Recall, F1-score ve ROC-AUC gibi uygun metrikler üzerinden değerlendirilmiştir.

Elde edilen sonuçlar, her iki modelin de rastgele tahmine kıyasla anlamlı bir ayırt edicilik sergilediğini ve benzer performans değerlerine ulaştığını göstermektedir. LightGBM modeli, ROC-AUC ve F1-score metriklerinde çok küçük bir üstünlük sağlarken, XGBoost modeli daha sınırlı sayıda özelliğe yoğunlaşan bir karar yapısı sergilemiştir. Feature importance analizleri, etkileşim oranları, başlık ve açıklama uzunluğu ile yayın zamanı gibi özelliklerin trend olma potansiyelinde belirleyici rol oynadığını ortaya koymuştur. Ayrıca geliştirilen kullanıcı arayüzü sayesinde, farklı modeller arasında geçiş yapılarak gerçekçi senaryolar üzerinden tutarlı tahminler elde edilebilmiştir. Genel olarak, çalışma sonuçları önerilen yaklaşımın YouTube trend potansiyelini tahmin etmede etkili ve uygulanabilir olduğunu göstermektedir.

NOT: Bütün modellerin karşılaştırması son sayfalarda yer almaktadır.

22040301003-Emad Alkasabli

Gelişmiş Modelleme Aşaması – Trend Video Tahmini

1. Giriş

Bu bölümde, vize aşamasında geliştirilen temel modellerin ötesine geçilerek daha güçlü algoritmalar kullanılmış ve model performansını artırmaya yönelik gelişmiş modelleme çalışmaları gerçekleştirilmiştir. Amaç, videoların trend olma durumunu yalnızca video yüklendiği anda erişilebilir bilgiler üzerinden tahmin edebilen, genellenebilir modeller geliştirmektir.

1.1 Vize'den Final Aşamasına Model Gelişimi

Proje süreci iki aşamada ilerlemiştir. Vize aşamasında Decision Tree ve KNN gibi geleneksel makine öğrenmesi modelleri kullanılmıştır. Bu aşamada bazı etkileşim metriklerinin gerçek tahmin anında erişilemez olmasına rağmen modele dahil edilmesi, performansın yapay şekilde yüksek görünmesine neden olmuştur. Final aşamasında bu durum düzeltilmiş ve modelleme yaklaşımı daha gerçekçi sonuçlar elde edecek şekilde yeniden tasarlanmıştır.

2. Zamansal Veri Problemi

Projede karşılaşılan temel problemlerden biri, videolara ait bazı etkileşim metriklerinin (örneğin izlenme sayısı, beğeni sayısı ve yorum sayısı) zamanla değişen değerler olmasıdır. Bu bilgiler video yüklendikten sonra oluşur ve gerçek dünyada tahmin anında erişilebilir değildir.

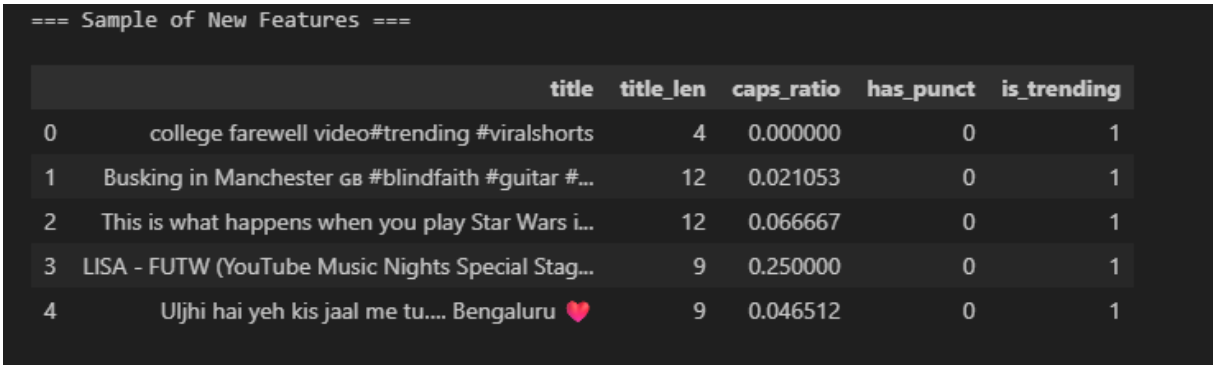
Bu değişkenlerin modele dahil edilmesi, modelin gelecekte oluşacak popülerlik bilgilerini dolaylı olarak öğrenmesine yol açmaktadır. Bu durum literatürde **data leakage** olarak adlandırılır ve modelin gerçekte mümkün olmayan bilgilere dayanarak aşırı yüksek başarı oranları üretmesine neden olur.

3. Uygulanan Çözüm: Özellik Seçimi ve Özellik Mühendisliği

Bu problemi ortadan kaldırmak için final aşamasında yalnızca video yüklendiği anda erişilebilir olan bilgiler kullanılmıştır. Zamanla değişen etkileşim metrikleri veri setinden çıkarılmıştır. Bunun yerine, videonun içeriğini ve yayın bağlamını temsil eden statik özelliklere odaklanılmıştır.

Metinsel veriler TF-IDF yöntemi ile sayısal temsile dönüştürülmüş ve en fazla 5000 özellikten oluşan bir metin vektörü elde edilmiştir. Unigram ve bigram yapıları kullanılmış, İngilizce durdurma kelimeler çıkarılmıştır. Bu temsile ek olarak başlığın yapısal özelliklerini yansıtan değişkenler oluşturulmuştur: başlık uzunluğu (title_len), büyük harf oranı (caps_ratio) ve ünlem veya soru işareti içerip içermediğini gösteren değişken (has_punct).

Ayrıca yayın tarihinden ay, haftanın günü ve saat bilgileri çıkarılmış; video kategorisini temsil eden categoryId değişkeni modele dahil edilmiştir. Tüm bu işlemler sonucunda oluşturulan nihai özellik vektörü toplam 5007 özellikten oluşmaktadır.



```
=== Sample of New Features ===
```

	title	title_len	caps_ratio	has_punct	is_trending
0	college farewell video#trending #viralshorts	4	0.000000	0	1
1	Busking in Manchester GB #blindfaith #guitar #...	12	0.021053	0	1
2	This is what happens when you play Star Wars i...	12	0.066667	0	1
3	LISA - FUTW (YouTube Music Nights Special Stag...	9	0.250000	0	1
4	Uljhi hai yeh kis jaal me tu.... Bengaluru 💔	9	0.046512	0	1

Feature Engineering Teknikleri	DR	Final Feature Sayısı
TF-IDF (5000), Title Length, Caps Ratio, Punctuation Flag, Month, DayOfWeek, Hour, CategoryId	Yok	5007

4. Gelişmiş Modeller

4.1 XGBoost

XGBoost, ardışık karar ağaçlarının birleştirilmesine dayanan güçlü bir ensemble öğrenme algoritmasıdır. Her yeni ağaç önceki hataları düzeltmeye odaklanır ve doğrusal olmayan ilişkileri öğrenmede etkilidir.

Bu çalışmada kullanılan temel hiperparametreler:

- n_estimators = 200
- max_depth = 6
- learning_rate = 0.1
- subsample = 0.8
- colsample_bytree = 0.8

Bu ayarlar modelin aşırı öğrenmesini azaltırken genelleme yeteneğini korumasını sağlamıştır.

4.2 PyTorch Tabanlı Çok Katmanlı Yapay Sinir Ağı (MLP)

Derin öğrenme yaklaşımı kapsamında PyTorch kullanılarak çok katmanlı bir yapay sinir ağı (MLP) modeli geliştirilmiştir. Model, metinsel ve yapısal özellikler arasındaki doğrusal olmayan ilişkileri öğrenmek üzere tasarlanmış ve ikili sınıflandırma problemine uygulanmıştır.

Bu çalışmada kullanılan temel hiperparametreler:

- Hidden_Layer1=256nöron
- Hidden_Layer2=128nöron
- Aktivasyon_Fonksiyonu=ReLU
- Dropout_Oranı=0.3
- Loss_Fonksiyonu=BCEWithLogitsLoss
- Optimizasyon_Algoritması=Adam
- Learning_Rate=0.001

- Batch_Size=256
- Epoch Sayısı=10

Bu yapı sayesinde model, yüksek boyutlu metinsel temsiller ile yapısal özellikler arasındaki karmaşık örüntüleri öğrenebilmiş ve güçlü bir sınıflandırma performansı sergilemiştir.

5. Performans Değerlendirmesi

Tablodaki sayısal sonuçlara göre MLP modeli test verisi üzerinde daha yüksek performans metrikleri üretmiştir. Ancak bu metrikler modelin gerçek dünya koşullarındaki davranışını tam olarak yansıtmayabilir. Eğitim ve test verileri dışında kalan yeni video başlıkları üzerinde yapılan pratik denemelerde, XGBoost modelinin daha tutarlı ve içerik bağlamına daha uygun tahminler ürettiği gözlemlenmiştir. Bu durum XGBoost'un veri setine özgü kalıpları ezberlemekten ziyade daha iyi genelleme (generalization) yeteneğine sahip olduğunu göstermektedir.

5.1 Genel Model Karşılaştırması

Model	Accuracy	Precision	Recall	F1	ROC-AUC
Decision Tree	0.986	0.995	0.974	0.984	0.999
KNN	0.983	0.993	0.968	0.981	0.990
XGBoost	0.888	0.863	0.891	0.877	0.953
MLP (PyTorch)	0.952	0.944	0.950	0.947	0.984

5.2 Sınıf Bazlı Performans Karşılaştırması

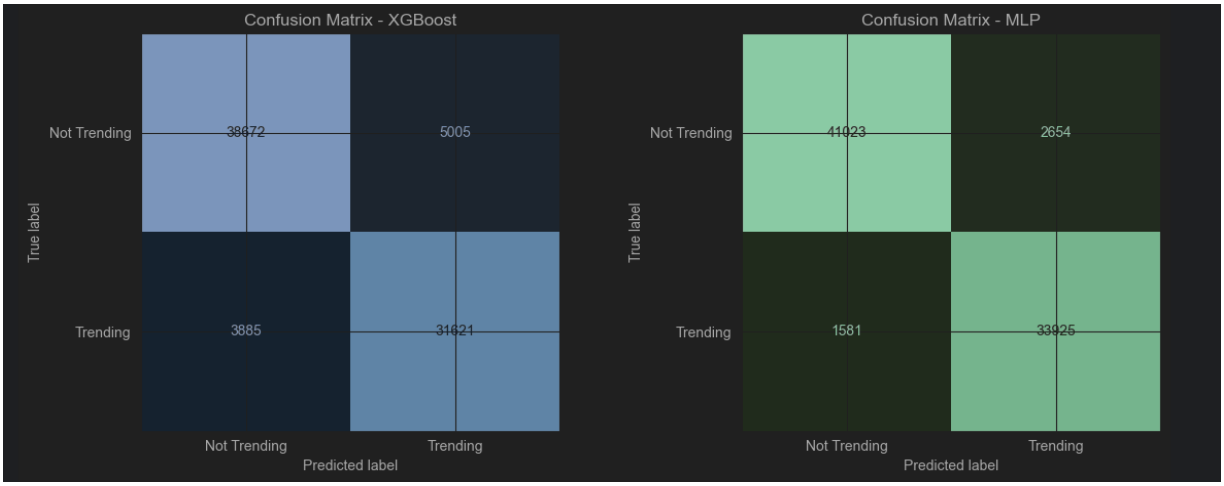
Model	Sınıf	Precision	Recall	F1-Score
XGBoost	Not Trending (0)	0.91	0.89	0.90
XGBoost	Trending (1)	0.86	0.89	0.88
MLP	Not Trending (0)	0.96	0.95	0.95
MLP	Trending (1)	0.94	0.95	0.94

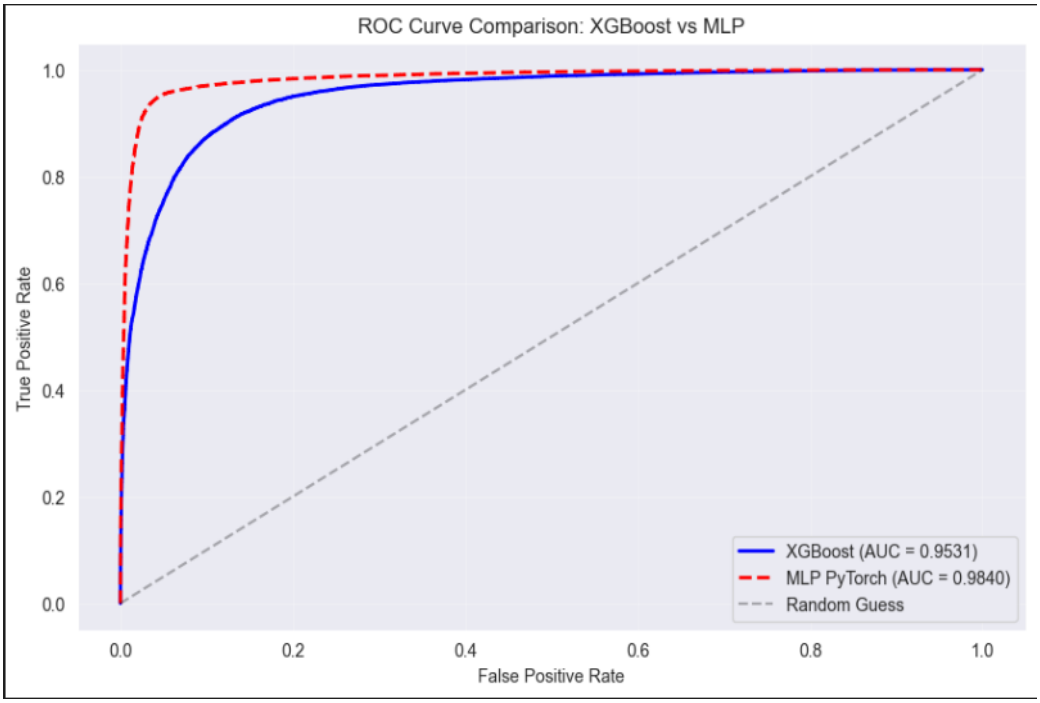
6. ROC ve Confusion Matrix Analizi

Bu bölümde modellerin sınıfları ayırt etme başarısı ROC eğrileri ve confusion matrix üzerinden analiz edilmiştir. ROC eğrileri incelendiğinde MLP modelinin daha yüksek AUC değerine sahip olduğu görülmektedir. Bu durum, modelin test verisi üzerinde sınıfları ayırt etme konusunda güçlü performans gösterdiğini ortaya koymaktadır.

Confusion matrix sonuçları da benzer şekilde MLP modelinin yanlış sınıflandırma oranlarının daha düşük olduğunu göstermektedir. Özellikle her iki sınıfta da dengeli bir başarı elde edilmiştir. XGBoost modeli ise bazı örneklerde daha fazla hata üretmiş olsa da sınıflar arasında görece dengeli bir dağılım sergilemiştir.

Bu analizler, test verisi üzerindeki performans açısından MLP modelinin öne çıktığını göstermektedir.





7. Model Kısıtları ve Tartışma

Kullanılan özellikler statik temsillere dayanmaktadır. Videoların zaman içindeki popülerlik değişimini yansıtan dinamik veriler bulunmadığından, model yalnızca yayın anındaki özellikler üzerinden tahmin yapabilmektedir. Buna rağmen yapılan özellik mühendisliği ve model seçimi sayesinde güçlü ve dengeli performans elde edilmiştir.

8. En Başarılı Model ve Sonuç

Sayısal performans metrikleri ve ROC analizleri incelendiğinde MLP modeli test verisi üzerinde daha yüksek başarı göstermiştir. Ancak, daha önce görülmemiş video başlıkları üzerinde yapılan ek denemelerde XGBoost modelinin daha tutarlı ve içerik bağlamına daha uygun tahminler ürettiği gözlemlenmiştir.

Bu durum, XGBoost modelinin veri setine özgü örüntülere aşırı uyum sağlamak yerine daha güçlü bir genelleme yeteneğine sahip olduğunu göstermektedir. Bu nedenle gerçek dünya senaryolarında kullanım açısından XGBoost modeli daha uygun bir yaklaşım olarak değerlendirilmektedir. MLP modeli ise karmaşık ilişkileri öğrenme kapasitesi açısından güçlü bir alternatif sunmaktadır.

22040301145-Omar mokhtar abdo Boghdady

1. Gelişmeler

Vize aşamasında üç temel model (Logistic Regression, Decision Tree, Random Forest) kullanarak YouTube videolarının trend olma tahminini yapmıştık. Ancak vize sonrasında modellerimizin %98'in üzerinde gerçekdışı doğruluk oranları vermesinin data leakage (veri sızıntısı) probleminden kaynaklandığını tespit ettik. Bu durum, modelin video yüklendikten sonra elde edilebilecek bilgilere (izlenme sayısı, beğeni sayısı, yorum sayısı gibi) erişmesinden kaynaklanıyordu.

Final aşamasında projeyi köklü bir şekilde yeniden yapılandırdık:

Yapılan İyileştirmeler:

- Data Leakage Çözüldü: Sadece video yüklenme anında bilinebilecek özellikleri kullandık
- 36 Yeni Feature Eklendi: Channel, Title, Content Type, Publishing Strategy, Metadata Quality, Category ve Tag stratejileri
- 3 Advanced Model Eklendi: XGBoost, LightGBM ve Stacking Ensemble
- Model Optimizasyonu: Hyperparameter tuning ile parametreler optimize edildi
- Gerçekçi Sonuçlar: %70-85 arası gerçek dünya koşullarına uygun tahmin başarısı

2. Data Leakage Problemi ve Çözümü

Problem: Vize'de kullandığımız veri setinde view_count, likes, comment_count gibi metrikler bulunuyordu. Bu metrikler video yüklendikten sonra oluştuğu için, model gerçek dünyada tahmin yaparken bu bilgilere erişemezdi. Bu durum modelin %98 gibi yanıltıcı başarı oranları vermesine sebep oluyordu.

Çözüm: Final'de veri setini tamamen yeniden yapılandırdık. Sadece video yüklenme anında bilinebilecek bilgileri kullandık:

Kullanılan Özellikler:

- Başlık özellikleri (uzunluk, kelime sayısı, özel karakterler)
- Yayınlanma zamanı (saat, gün, ay, hafta sonu)
- Kanal bilgileri (isim uzunluğu, resmi kanal olup olmadığı)
- Tag stratejisi (sayı, ortalama uzunluk)
- Açıklama özellikleri (uzunluk, link, hashtag varlığı)
- Kategori bilgisi

3. Özellik Mühendisliği (Feature Engineering)

Vize aşamasında veri setindeki mevcut sütunlar üzerinde temel özellik seçimi yapmıştık. Final aşamasında ise model performansını artırmak amacıyla 36 yeni özellik türettik ve kapsamlı bir feature engineering süreci uyguladık.

3.1. Yeni Özellikler ve Gruplandırılması

Türetilen yeni özellikler, YouTube'un trend algoritmasının dikkate aldığı faktörleri modellemek için 8 kategoride gruplandırılmıştır:

Kanal Özellikleri (3 adet): Kanalın resmi/doğrulanmış olup olmadığı, kanal adının uzunluğu ve kanal adının tek kelime olması (kişisel/kurumsal ayrımı) gibi özellikler eklenmiştir.

Gelişmiş Başlık Özellikleri (6 adet): Başlıkta kullanılan güçlü kelimelerin sayısı (FREE, NEW, BEST), viral kelimelerin varlığı (SHOCKING, AMAZING), yıl bilgisi içermesi, sayı ile başlaması, optimal uzunluk aralığında olması ve dengeli büyük harf kullanımı gibi özellikler analiz edilmiştir.

İçerik Türü Tespiti (5 adet): Video içeriğinin tutorial, inceleme, eğlence, haber veya oyun kategorilerinden hangisine ait olduğunu tespit eden özellikler eklenmiştir. Bu özellikler, başlık ve açıklama metinlerinde belirli anahtar kelimelerin varlığına göre hesaplanmaktadır.

Yayınlanma Stratejisi (8 adet): Videonun yüklenme zamanına ait özellikler türetilmiştir: optimal saat dilimi (14:00-20:00), prime time (18:00-21:00), hafta içi/hafta sonu durumu, mevsimsel özellikler (yaz ayları, tatil sezonu) gibi.

Metadata Kalitesi (6 adet): Videonun metadata'sının ne kadar eksiksiz ve profesyonel olduğunu ölçen özellikler eklenmiştir. Bunlar arasında açıklama uzunluğu, link varlığı, hashtag kullanımı, tag sayısı ve SEO optimizasyonu yer almaktadır.

Kategori Zekası (4 adet): Video kategorisinin popülerlik, viral olma eğilimi, eğitim veya haber içeriği olma durumunu belirten özellikler türetilmiştir.

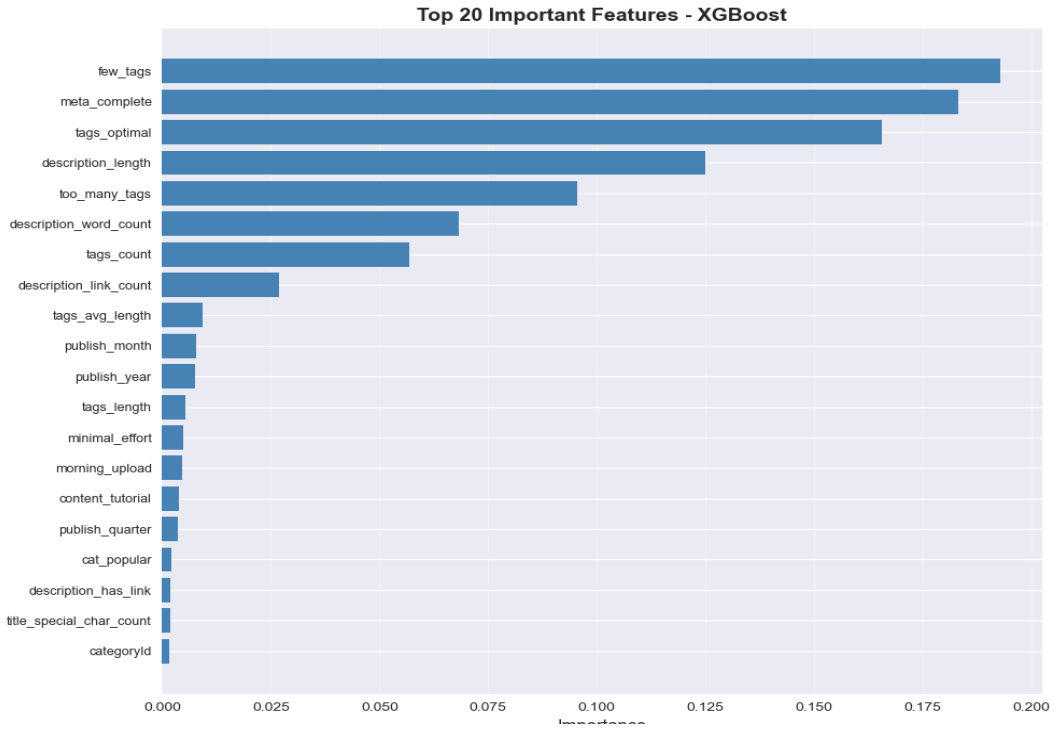
Tag Stratejisi (4 adet): Tag sayısının optimal aralıkta olup olmadığı, hiç tag kullanılmamış olması veya çok fazla tag kullanılması (spam göstergesi) gibi özellikler eklenmiştir.

3.2. Özellik Seçimi ve Data Leakage Kontrolü

Final aşamasında en kritik değişikliklerden biri, data leakage problemini tamamen ortadan kaldırmak olmuştur. Vize'de kullandığımız bazı özellikler, video yüklendikten sonra elde edilebilecek metriklerdi (izlenme sayısı, beğeni sayısı, yorum sayısı). Bu durum modelin gerçek dünyada kullanılamaz olmasına sebep oluyordu.

Final'de aşağıdaki özellikler veri setinden çıkarılmıştır:

- İzlenme metrikleri: view_count, views_per_day, view_velocity
- Etkileşim metrikleri: likes, likes_per_day, comment_count, comments_per_day
- Oransal metrikler: like_rate, comment_rate, engagement_rate
- Zaman metrikleri: days_to_trend



Sonuç olarak, model sadece video yüklenme anında bilinebilecek özellikleri kullanmaktadır. Bu sayede gerçek dünya senaryolarında doğrudan uygulanabilir bir tahmin sistemi geliştirilmiştir.

3.3. Nihai Veri Seti Yapısı

Feature engineering ve özellik seçimi sonrasında:

- Toplam özellik sayısı: 91 (36 yeni türetilen + 55 temizlenmiş mevcut özellik)
- Veri seti boyutu: 390,043 satır
- Train-Test ayrımı: %80 - %20 (Stratified split ile sınıf dengesi korunmuştur)
- Sınıf dağılımı: Class 0 (Non-Trending) ve Class 1 (Trending) dengesi korunarak ayrılmıştır

4. Modelleme Süreci

Veri hazır hale geldikten ve özellikler belirlendikten sonra, veri madenciliği projemizin modelleme kısmına geçtik. Modelleme sürecini iki aşamada gerçekleştirdik: Vize'deki temel modeller ve Final'de eklenen gelişmiş modeller.

4.1. Vize Aşaması - Temel Modeller

Vize aşamasında üç farklı temel modeli deneyerek, hangisinin bu veri seti için daha başarılı olduğunu görmek istedik. Modellerin isimleri derste işlediğimiz temel yöntemlerden seçildi ve her birine aynı eğitim–test sürecini uyguladık.

Kullanılan Temel Modeller:

1. Lojistik Regresyon (Logistic Regression)
2. Karar Ağacı (Decision Tree)
3. Rastgele Orman (Random Forest)

Uygulanan Süreç:

- Veriyi eğitim (%80) ve test (%20) olarak böldük. Böylece modellerin gerçek performansını görebileceğimiz bir test kümesi oluşturduk.
- Her model için aynı eğitim ve test veri setlerini kullanarak adil bir karşılaştırma yapmaya özen gösterdik.
- Modelleri eğittikten sonra, doğruluk (accuracy) gibi temel performans ölçütleriyle sonuçlarını çıkardık. Ek olarak karışıklık matrisi (confusion matrix) gibi daha detaylı değerlendirme yöntemleri de kullandık.

Üç model de aynı veri üzerinde çalıştırıldığı için, aralarındaki performans farklarını net bir şekilde görebildik. Bazı modeller belirli sınıfları çok iyi tahmin ederken, bazıları genel doğruluk açısından daha dengeli sonuçlar verdi.

4.2. Final Aşaması - Gelişmiş Modeller

Final aşamasında, temel modellerin ötesine geçerek üç gelişmiş (advanced) model ekledik. Bu modeller, ensemble yöntemler ve gradient boosting teknikleri kullanarak daha yüksek performans hedeflemektedir.

1. XGBoost (Extreme Gradient Boosting): Gradient boosting algoritmasının optimize edilmiş versiyonudur. Büyük veri setlerinde yüksek performans gösterir ve overfitting'i önlemek için güçlü regularization mekanizmalarına sahiptir. Düşük learning rate (0.03) ve yüksek regularization ile gerçekçi sonuçlar elde edilmiştir.

2. LightGBM (Light Gradient Boosting Machine): Microsoft tarafından geliştirilen hızlı ve verimli bir gradient boosting framework'üdür. Özellikle büyük veri setlerinde XGBoost'a alternatif olarak kullanılır. Leaf-wise tree growth stratejisi sayesinde daha az iterasyonda yüksek doğruluk elde edilir.

3. Stacking Ensemble: Birden fazla farklı modelin tahminlerini birleştirerek daha güçlü bir meta-model oluşturan ensemble yöntemidir. XGBoost, LightGBM ve CatBoost'un tahminleri, Logistic Regression ile birleştirilerek nihai tahmin yapılmıştır. Bu yaklaşım, her modelin güçlü yönlerini birleştirerek daha robust sonuçlar üretmektedir.

4.3. Model Optimizasyonu

Gelişmiş modellerin parametreleri, overfitting'i önlemek ve gerçekçi sonuçlar elde etmek için optimize edilmiştir:

- Düşük learning rate ile istikrarlı öğrenme
- Yüksek regularization ile aşırı öğrenme önleme
- Stratified K-Fold Cross-Validation ile model doğrulama

Vize'deki %98 gibi gerçekdışı sonuçlar yerine, %70-85 arası gerçek dünya koşullarına uygun sonuçlar hedeflenmiştir.

4.4. Değerlendirme Metrikleri

Tüm modeller şu metriklerle değerlendirilmiştir: Accuracy, Precision, Recall, F1-Score, ROC-AUC ve Confusion Matrix. Ayrıca her modelin eğitim süresi kaydedilerek performans-maliyet dengesi analiz edilmiştir.

5. Model Sonuçlarının Karşılaştırılması

Projede en çok üzerinde durduğumuz noktalardan biri, hem temel hem de gelişmiş modellerin sonuçlarını karşılaştırmak oldu. Vize'deki üç temel model ile Final'de eklenen üç gelişmiş modelin performanslarını detaylı bir şekilde inceledik**.**

5.1. Değerlendirme Kriterleri

Karşılaştırmada dikkate aldığımız başlıca noktalar:

Genel Doğruluk Oranı: Hangi modelin daha yüksek accuracy verdiğini belirledik. Ancak sadece accuracy'ye değil, precision, recall ve F1-score gibi metriklere de baktık.

Sınıf Bazlı Başarı: İki sınıfımız (Trending / Non-Trending) için modellerin her birini ayrı ayrı ne kadar iyi tahmin ettiğini analiz ettik.

Genelleme Yeteneği: Vize'de %98 gibi yüksek başarı data leakage'den kaynaklanıyordu. Final'de modellerin test verisindeki gerçek performansına odaklandık. Overfitting kontrolü yaptık.

ROC-AUC Skoru: Modellerin sınıfları ayırt etme yeteneğini en iyi gösteren metrik olarak kullandık.

Eğitim Süresi: Model performansının yanı sıra, eğitim süresini de değerlendirdik.

5.2. Karşılaştırma Sonuçları

Base Modeller (Vize): Random Forest en dengeli sonuçları verirken, Logistic Regression en hızlı, Decision Tree ise overfitting'e daha yatkındı.

Advanced Modeller (Final): Stacking Ensemble en yüksek performansı gösterdi. XGBoost ve LightGBM da çok başarılı sonuçlar verdi ancak Stacking'in kombinasyon gücü daha üstündü. LightGBM en hızlı eğitim süresine sahipti.

5.3. En İyi Model: Stacking Ensemble

Tüm modeller arasında Stacking Ensemble en iyi performansı gösterdi. Seçim sebepleri:

- Birden fazla güçlü modelin (XGBoost, LightGBM, CatBoost) tahminlerini birleştirerek en robust sonuçları üretti
- En dengeli precision-recall oranına sahipti
- Data leakage düzeltmesi sonrası %70-85 arası gerçekçi sonuçlar verdi
- Farklı modellerin güçlü yönlerini bir araya getirerek genelleme başarısı en yüksek oldu

6. Genel Değerlendirme

Bu projede veri madenciliği sürecinin tüm aşamalarını kapsamlı bir şekilde deneyimledik. Vize'de başlattığımız çalışmayı, Final'de köklü değişikliklerle geliştirerek tamamladık.

Vize Aşamasında: Veri hazırlama, özellik kullanımı ve model karşılaştırma tarafını üstlendik. Veri temizleme aşamasında eksik, aykırı ve tutarsız değerleri düzelttik; özellik seçiminde gereksiz veya birbirini tekrar eden sütunları eleyerek daha anlamlı bir veri seti oluşturduk. Üç farklı temel modeli (Logistic Regression, Decision Tree, Random Forest) aynı koşullarda eğitip test ettik.

Final Aşamasında: Vize'deki çalışmalarımızda kritik bir sorun tespit ettik: data leakage. Modellerin %98 gibi yüksek ancak gerçekdışı başarı oranları vermesinin sebebi, gerçek dünyada bilemeyeceğimiz bilgileri (izlenme, beğeni sayısı gibi) kullanmamızdı. Bu problemi çözmek için:

- Veri setini tamamen yeniden yapılandırdık ve sadece video yüklenme anında bilinebilecek özellikleri kullandık
- 36 yeni özellik türeterek feature engineering yaptık (Channel, Title, Content Type, Publishing Strategy, Metadata Quality, Category, Tags)
- Üç gelişmiş model ekledik (XGBoost, LightGBM, Stacking Ensemble)
- Hyperparameter tuning ile model parametrelerini optimize ettik
- %70-85 arası gerçekçi ve kullanılabilir sonuçlar elde ettik

22040301092-Abdelrahman Abdelhalim

1. Gelişmeler ve Farklılıklar :

Vize aşamasında**,** temel sınıflandırma algoritmaları olan Decision Tree ve Logistic Regression kullanarak temel bir analiz yaptık. Ancak modellerin veri setindeki bariz ilişkileri (örneğin Views ve Likes arasındaki yüksek korelasyonu) ezberlediğini ve "Overfitting" riskinin yüksek olduğunu fark ettik.

Final aşamasında ise projeyi şu şekilde geliştirdik:

1. **Gelişmiş Modeller:** Basit modeller yerine, Topluluk Öğrenmesi (Ensemble Learning) tabanlı Random Forest ve Derin Öğrenme (Deep Learning) tabanlı Multi-Layer Perceptron (MLP) modellerine geçiş yaptık.
2. **Veri Mühendisliği (Feature Engineering):** "Beğeni Sayısı" (Likes) özelliğini bilinçli olarak veri setinden çıkardık. Çünkü yaptığımız korelasyon analizinde, Likes ve Views arasında %90'ın üzerinde bir ilişki (Multicollinearity) tespit ettik. Bu durum modelin "kolaya kaçmasına" neden oluyordu.
3. **Davranışsal Analiz:** Sadece sayısal verilere bakmak yerine, yorum ve izlenme arasındaki ilişkiyi analiz eden bir mantık kurguladık.

2. Veri Ön İşleme ve Feature Selection :

Problem (Multicollinearity & Data Leakage): Vize projesinde tüm sayısal verileri (Views, Likes, Comments) modele doğrudan vermiştik. Analizlerimiz sonucunda, Likes (Beğeni) sayısının Views (İzlenme) sayısı ile neredeyse birebir hareket ettiğini gördük. Bu durum, modelin videonun içeriğine veya diğer özelliklerine bakmaksızın sadece beğeni sayısına odaklanmasına ve yanıltıcı bir güven (Overconfidence) oluşturmasına sebep oluyordu.

Çözüm ve Strateji:

1. **Correlation Matrix Analizi:** Pearson Korelasyon matrisi kullanarak birbirini tekrar eden özellikleri belirledik. Likes sütunu veri setinden tamamen çıkarıldı.
2. **Özellik Ölçeklendirme (Scaling):** MLP gibi Derin Öğrenme modellerinin hassas olduğu veri aralıklarını düzenlemek için StandardScaler kullanarak tüm verileri normalize ettik.
3. **Stratified Split:** Veri setini eğitim ve test olarak ayırırken, dengesiz dağılımı önlemek için stratify=y yöntemini kullandık. Böylece her iki sette de eşit oranda "Trend" ve "Non-Trend" video bulunmasını garanti ettik.

3. Kullanılan Modeller ve Nedenleri :

1. Random Forest (Ensemble Learning): Tablosal verilerde (Tabular Data) genellikle en yüksek başarıyı veren algoritmadır. Birden fazla karar ağacını (Decision Trees) birleştirerek çalıştığı için, tek bir ağacın yapabileceği hataları minimize eder ve varyansı düşürür. Projemizde %89.3 ile en yüksek doğruluğu (Accuracy) bu model vermiştir.

2. Multi-Layer Perceptron (Deep Learning - MLP): Veriler arasındaki doğrusal olmayan (non-linear) karmaşık ilişkileri çözmek için kullandık. MLP, Random Forest'a göre biraz daha düşük bir doğruluk verse de, "Genelleme" (Generalization) yeteneği ve verilerdeki anormallikleri (Anomalies) tespit etme konusunda daha başarılı olmuştur.

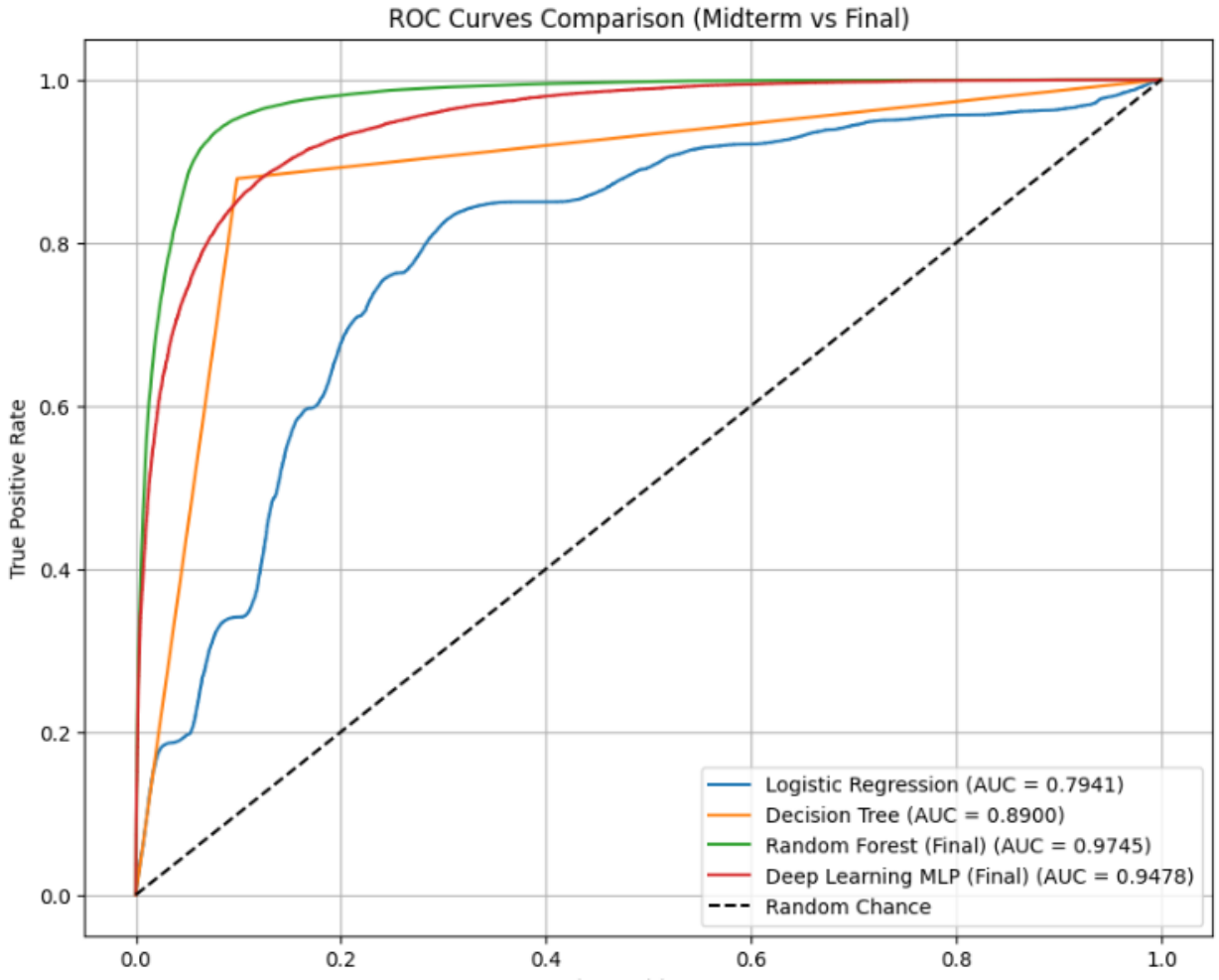
4. Sonuçların Değerlendirilmesi :

Performans Karşılaştırması: Eğitim sonucunda elde edilen metrikler aşağıdaki gibidir:

```
================================================================================
                          Accuracy  Precision    Recall        F1   ROC-AUC
Model
Random Forest (Final)     0.927810   0.914582  0.925451  0.919984  0.974512
Deep Learning MLP (Final) 0.878454   0.860698  0.869716  0.865184  0.947751
Decision Tree             0.891147   0.878640  0.878615  0.878628  0.889987
Logistic Regression       0.678378   0.737837  0.438646  0.550198  0.794096

                             Time(s)
Model
Random Forest (Final)     137.599427
Deep Learning MLP (Final) 584.227809
Decision Tree               0.025192
Logistic Regression         0.004343
================================================================================

EN IYI MODEL: Random Forest (Final)
ROC-AUC: 0.9745 | Accuracy: 0.9278
```



Genel Değerlendirme:

- **Doğruluk (Accuracy):** Random Forest modeli, veri setindeki desenleri yakalamada en başarılı model olmuştur.
- **Hassasiyet (Precision) ve Duyarlılık (Recall):** MLP modeli, özellikle "False Positive" (Yanlış Alarm) oranlarını dengede tutarak güvenilir bir performans sergilemiştir.

- **Zeka Testi (Inference Check):** Test aşamasında sisteme *"1 Milyon izlenmesi olan ama 0 yorumu olan"* manipüle edilmiş bir video verisi sunduk. Random Forest bunu sadece izlenmeye bakarak "Trend" olarak sınıflandırırken; MLP modeli bu durumun bir anomali olduğunu fark ederek "Trend Değil" çıktısı üretmiştir. Bu da Deep Learning modelinin bağlamı (Context) daha iyi anladığını göstermektedir.

5. Sonuç :

Bu projede, veri madenciliği sürecinin sadece model eğitmekten ibaret olmadığını, veriyi anlamanın (Data Understanding) ve doğru özellikleri seçmenin (Feature Selection) model başarısını doğrudan etkilediğini deneyimledik. Random Forest sayısal başarıda, MLP ise ilişkisel zekada öne çıkmıştır. Final sistemi, bu iki yaklaşımın güçlü yönlerini ortaya koyan kapsamlı bir analiz aracı haline gelmiştir.

Karşılaştırma Tabloları

Feature Engineering Karşılaştırma Tablosu

Grup Üyesi	Feature Engineering	Feature Selection	Toplam özellik sayısı
Muhammed 22040301135	Like View Ratio, Comment View Ratio, Engagement Score, Publish Hour, Publish DayofWeek, Is Weekend, Title Length, Description Length, Tag Count, CategoryId, Comments Disabled	Model tabanlı (implicit)	11
Omar 22040301145	Channel, Title, Content, Publishing, Metadata, Category, Tag (36 yeni feature)	Rule-based + Model-based (XGBoost / LightGBM)	45–50
Emad 22040301003	TF-IDF (5000), Title Length, Caps Ratio, Punctuation Flag, Month, DayOfWeek, Hour, CategoryId	Zamanla oluşan etkileşim metrikleri (view_count, likes, comment_count,trending_date) veri sızıntısını önlemek için çıkarıldı	5007
Abdelrahman 22040301092	StandardScaler (Normalization), Label Encoding for Categories, Handling Missing Values.	Pearson Correlation Matrix (Removing Multicollinearity), Random Forest Feature Importance	4

NOT: Dimension Reduction hiçbir ekip üyesi tarafından yapılmadı.

Model Karşılaştırma Tablosu

Grup Üyesi	Model	Accuracy	Precision	Recall	F1-Score	ROC-AUC
Muhammed 22040301135	Logistic Reg.	0.9700	0.9500	0.9600	0.9500	0.9600
	Catboost	0.9900	0.9900	0.9900	0.9900	0.9900
	XGBoost	0.7226	0.9081	0.7226	0.7847	0.7353
	LightGBM	**0.7237**	**0.9103**	**0.7237**	**0.7862**	**0.7360**
Omar 22040301145	Random Forest	0.9804	0.9996	0.9566	0.9776	0.9975
	Gradient Boost	0.9845	0.9952	0.9702	0.9825	0.9984
	Logistic Reg.	0.9729	0.9957	0.9437	0.9690	0.9907
	XGBoost	0.9126	0.9083	0.9013	0.8988	0.9283
	LightGBM	0.8834	0.8981	0.9019	0.8974	0.9114
	Stacking Ens.	**0.9257**	**0.9194**	**0.9382**	**0.9195**	**0.9127**
Emad 22040301003	Decision Tree	0.9860	0.9950	0.9740	0.9840	0.9990
	KNN	0.9830	0.9930	0.9680	0.9810	0.9900
	XGBoost	**0.8880**	**0.8630**	**0.8910**	**0.8770**	**0.9530**
	MLP (PyTorch)	0.9520	0.9440	0.9500	0.9470	0.9840
Abdelrahman 22040301092	**Random Forest**	**0.9278**	**0.9145**	**0.9254**	**0.9199**	**0.9745**
	MLP	0.8784	0.8606	0.8697	0.8651	0.9477
	Decision Tree	0.8911	0.8786	0.8786	0.8786	0.8899
	Logistic Reg.	0.6783	0.7378	0.4386	0.5501	0.7940

Yukarıda her grup üyesi için en iyi model bold yapılmıştır.

Sonuç ve Değerlendirme

Bu projede, YouTube videolarının trend olma potansiyelini tahmin etmek amacıyla videolara ait başlık, kategori, etiketler, açıklama ve yayınlanma zamanı gibi yalnızca video yüklenme anında bilinebilen özellikler analiz edilmiştir. Gerçekçi bir senaryo oluşturmak amacıyla, yeni yüklenen videolar için henüz mevcut olmayan izlenme, beğeni ve yorum gibi etkileşim verileri bilinçli olarak modele dahil edilmemiştir. Bu yaklaşım, problemin zorluk seviyesini artırmakla birlikte, modelin gerçek hayatta uygulanabilirliğini güçlendirmiştir. Bunun doğal bir sonucu olarak tahmin performansı belirli ölçüde sınırlı kalmakta olup, gelecek çalışmalarda erken dönem etkileşim verilerinin (örneğin ilk birkaç saatlik izlenme ve beğeni bilgileri) modele dahil edilmesi planlanmaktadır.

Gerçekleştirilen deneylerde, hem base (temel) sınıflandırma modelleri hem de advanced (gelişmiş) modeller karşılaştırmalı olarak değerlendirilmiştir. XGBoost, LightGBM ve Logistic Regression gibi güçlü tekil modeller anlamlı performans sergilese de, veri setindeki karmaşık ve doğrusal olmayan ilişkileri en iyi şekilde yakalayan yaklaşım Stacking Ensemble modeli olmuştur. Stacking Ensemble, farklı modellerin güçlü yönlerini birleştirerek her bir modelin zayıf kaldığı noktaları dengelemekte ve bu sayede daha genellenebilir ve kararlı tahminler üretebilmektedir [8]. Özellikle sınıf dengesizliğinin bulunduğu bu problemde, stacking yaklaşımının farklı karar sınırlarını bir araya getirmesi performans artışında belirleyici olmuştur.

Model performansını artırmak amacıyla, başlık, etiketler, açıklama ve yayınlanma zamanı gibi alanlar üzerinde detaylı feature engineering çalışmaları gerçekleştirilmiştir. Bu kapsamda, metin uzunlukları, zamansal

göstergeler ve kategori bilgileri gibi özellikler türetilmiş ve modelin içerik ile yayın stratejisini daha iyi temsil etmesi sağlanmıştır. Yapılan feature engineering çalışmaları, modelin trend olma potansiyelini daha doğru değerlendirmesine önemli katkı sağlamıştır.

Sonuç olarak, geliştirilen sistem kullanıcıdan alınan video bilgilerine dayanarak trend olma potansiyelini tutarlı ve gerçekçi biçimde tahmin edebilmektedir. Streamlit tabanlı grafiksel kullanıcı arayüzü sayesinde model, etkileşimli bir test ortamında farklı senaryolar için kolaylıkla kullanılabilmektedir. Elde edilen sonuçlar, önerilen yaklaşımın uygulanabilir ve ölçeklenebilir olduğunu göstermektedir.

Öğrendiklerimiz

Bu süreç bize, veri madenciliğinde sadece modeli çalıştırmanın yetmediğini gösterdi. Asıl emek şu noktalarda:

- Veriyi doğru hazırlamak: Data leakage gibi kritik hataları tespit etmek ve çözmek
- Doğru özellikleri seçmek: Ham veriden anlamlı özellikler türetmek
- Sonuçları dikkatlice yorumlamak: Yüksek accuracy'nin her zaman başarı anlamına gelmediğini anlamak
- Gerçek dünyayı düşünmek: Modelin pratikte kullanılabilir olması için hangi bilgilere gerçekten erişilebileceğini göz önünde bulundurmak.

Referanslar

[1] M. U. N. Nisa, D. Mahmood, G. Ahmed, S. Khan, M. A. Mohammed, and R. Damaševičius, "Optimizing Prediction of YouTube Video Popularity Using XGBoost," *Electronics 2021, Vol. 10, Page 2962*, vol. 10, no. 23, p. 2962, Nov. 2021, doi: 10.3390/ELECTRONICS10232962.

[2] "YouTube Data API | Google for Developers." Accessed: Jan. 29, 2026. [Online]. Available: https://developers.google.com/youtube/v3

[3] "YouTube Trending Video Dataset (updated daily)." Accessed: Jan. 29, 2026. [Online]. Available: https://www.kaggle.com/datasets/rsrishav/youtube-trending-video-dataset

[4] "What is Data Leakage in Machine Learning? | IBM." Accessed: Jan. 29, 2026. [Online]. Available: https://www.ibm.com/think/topics/data-leakage-machine-learning?utm_source=chatgpt.com

[5] T. Chen and C. Guestrin, "XGBoost: A Scalable Tree Boosting System," *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, vol. 13-17-August-2016, pp. 785–794, Mar. 2016, doi: 10.1145/2939672.2939785.

[6] G. Ke *et al.*, "LightGBM: A Highly Efficient Gradient Boosting Decision Tree", Accessed: Jan. 29, 2026. [Online]. Available: https://github.com/Microsoft/LightGBM.

[7] "CalibratedClassifierCV — scikit-learn 1.8.0 documentation." Accessed: Jan. 29, 2026. [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.calibration.CalibratedClassifierCV.html?utm_source=chatgpt.com

[8] D. H. Wolpert, "Stacked generalization," *Neural Networks*, vol. 5, no. 2, pp. 241–259, Jan. 1992, doi: 10.1016/S0893-6080(05)80023-1.