Eskişehir Osmangazi Üniversitesi

CISAR ICT Summer School 2025

SUMO Simülasyon Verileri ve Makine Öğrenmesi ile Elektrikli Araç Enerji Tüketimi Tahmini

Ceren Adıyaman

Emre Güner

Gamze Dağ

Gürkan Karaman

Koordinatör : Ahmet Alperen Polat

2025

1 İçindekiler

2 Giriş

2.1 Projenin Amacı

Bu çalışmanın temel amacı, SUMO (Simulation of Urban Mobility) trafik simülatörü kullanılarak elde edilen trafik verilerini işleyip analiz ederek, elektrikli araçların farklı yol ve trafik koşullarında ne kadar enerji tüketeceğini tahmin edebilecek bir yapay zekâ modeli geliştirmektir.

2.2 Modelin amacı:

Farklı trafik yoğunlukları, yol tipleri ve sürüş koşullarında elektrikli araçların enerji tüketim davranışlarını anlamak

Bu verilerden yola çıkarak sürüş verimliliğini artırmak ve enerji planlamasına katkı sağlamak

Sürdürülebilir ulaşım çözümleri için bilimsel bir altyapı sunmak

3 Metodoloji ve Uygulama

3.1 Veri Toplama ve Ön İşleme

3.1.1 SUMO Simülasyon Ortamının Kurulumu ve Kullanımı

Bu çalışmada, OpenStreetMap üzerinden alınan harita verileri dışa aktarılmış ve OpenTopography kullanılarak bölgeye ait yükseklik verileri elde edilmiştir. Harita verileri .osm formatından .net.xml formatına dönüştürülerek NetEdit aracıyla düzenlenmiş, elde edilen yükseklik verileri de bu .net.xml dosyasına entegre edilmiştir. Farklı tipte elektrikli araç kombinasyonları tanımlanarak araç bilgileri .xml dosyasında oluşturulmuş, rastgele trafik simülasyonu için rota verileri .xml formatında hazırlanmıştır. Tüm veriler TRACI kullanılarak çekilmiş, eksik veriler düzenlenmiş ve araç özellikleriyle birleştirilmiştir. Sonuçta, harita, yükseklik bilgileri, araç çeşitliliği, rotalar ve trafik senaryoları entegre edilmiş, eksiksiz ve çalışır durumda bir simülasyon ortamı elde edilmiştir.

3.1.2 Veri Toplama

Veri toplama sürecinde SUMO simülasyonundan elde edilen ham veriler, özel olarak geliştirilmiş SUMODataCollector sınıfı aracılığıyla adım adım kaydedilmiştir. Simülasyon, main.sumocfg yapılandırma dosyası kullanılarak TraCI arayüzü üzerinden başlatılmış ve her simülasyon adımında aktif araçlar tespit edilmiştir. Her araç için hız (m/s ve km/h), ivme (m/s^2), coğrafi konum (enlem, boylam, yükseklik), yol ve şerit bilgileri (edge_id, lane_id, lane_position, şerit hız limiti), araç tipi, araç kütlesi, batarya verileri (anlık doluluk, kapasite, enerji tüketimi) gibi bilgiler okunmuştur. Batarya durumu (SOC) ise doluluk ve kapasite değerlerinden yüzdesel olarak hesaplanmıştır. Her simülasyon adımında elde edilen bu veriler, zaman damgası (timestamp) ile birlikte bir Python sözlüğü olarak saklanmış ve tüm adımlar tamamlandığında Pandas DataFrame formatına dönüştürülerek CSV dosyasına yazılmıştır. Bu süreç sonunda toplam simülasyon adımı, kayıt sayısı, benzersiz araç sayısı gibi özet bilgiler ve hız/ivme istatistikleri raporlanmıştır.

3.1.3 Veri Birleştirme

data_collector modülü tarafından oluşturulan **final_training_data.csv** dosyası ile araç özelliklerini içeren **vehicles.add.xml** dosyası birleştirilmiştir. Bu birleştirme işlemi, her bir

aracın **vehicle_type** (araç tipi) özelliğine göre yapılmıştır. Bu sayede, her sürüş verisi kaydına ilgili aracın tipine ait özellikler eklenmiştir.

3.1.4 Eksik 'z' Değerlerinin Doldurulması

Birleştirilen veri setinin incelenmesi sonucunda, z (yükseklik) noktalarında veri eksiklikleri olduğu tespit edilmiştir. Bu z yüksekliği değeri, daha sonra eğim hesaplamasında kullanılacağı için doğru ve eksiksiz olması gerekmektedir. Bu nedenle, eksik değerlerin giderilmesi amacıyla **interpolasyon** yöntemi uygulanmıştır. İnterpolasyon, bilinen veri noktaları arasındaki bilinmeyen değerleri, komşu verilerin trendine dayalı olarak tahmin etme işlemidir. Bu yöntemle, eksik z değerleri mantıklı bir şekilde doldurulmuştur.

Ancak, veri setinde yer alan iki aracın (831 ve 934) hiç z verisi bulunmadığı için bu araçlara ait z sütunları NaN olarak kalmıştır. Bu durum, bu araçların daha sonra veri setinden çıkarılmasına veya özel olarak ele alınmasına neden olabilir.

3.1.5 Modelleme İçin Uygun Olmayan Sütunların Çıkarılması

Veri seti, modelleme aşamasına geçilmeden önce gereksiz veya modele katkı sağlamayacak sütunlardan arındırılmıştır. 'color', 'sigma', 'has.battery.device', 'stoppingThreshold', 'edge_id', 'lane_id', 'vehicle_type', 'speed_ms', 'lane_position', 'angle', 'lane_speed_limit', 'charge_level', 'capacity', 'battery_level', 'max_speed', 'length', 'min_gap', ‘mass’ gibi sütunlar, ya tüm veride aynı değere sahip oldukları ya da modelin çıktısıyla doğrudan ilişkili olmadıkları için veri setinden çıkarılmıştır (drop). Bu işlem, modelin daha verimli çalışmasını sağlamak ve gereksiz gürültüyü engellemek amacıyla yapılmıştır.

3.2 Özellikler Modele Verilmeden Önce Yapılan Analiz

Bu bölümde, eğitim öncesi uygulanan istatistiksel kontroller, eksik değer analizi, aykırı değer analizi, korelasyon incelemeleri ve özellik mühendisliği adımları özetlenmekte, her adımın modele etkisi gerekçelendirilmektedir.

3.2.1 Eksik Değer Analizi

İlk inceleme: df.isnull().sum() ile her sütundaki eksik değerler sayılmıştır.

Sonuç:

dist_m sütununda 300 eksik değer

slope_pct sütununda 119,640 eksik değer tespit edilmiştir.

Yorum: slope_pct eksikliği özellikle aracın hareket etmediği veya rakımın değişmediği anlarda gözlemlenmiştir. Bu durumlarda eğimin 0 kabul edilmesi hem fiziksel olarak mantıklı hem de model açısından tutarlıdır.

3.2.2 Aykırı Değer Analizi (IQR Yöntemi)

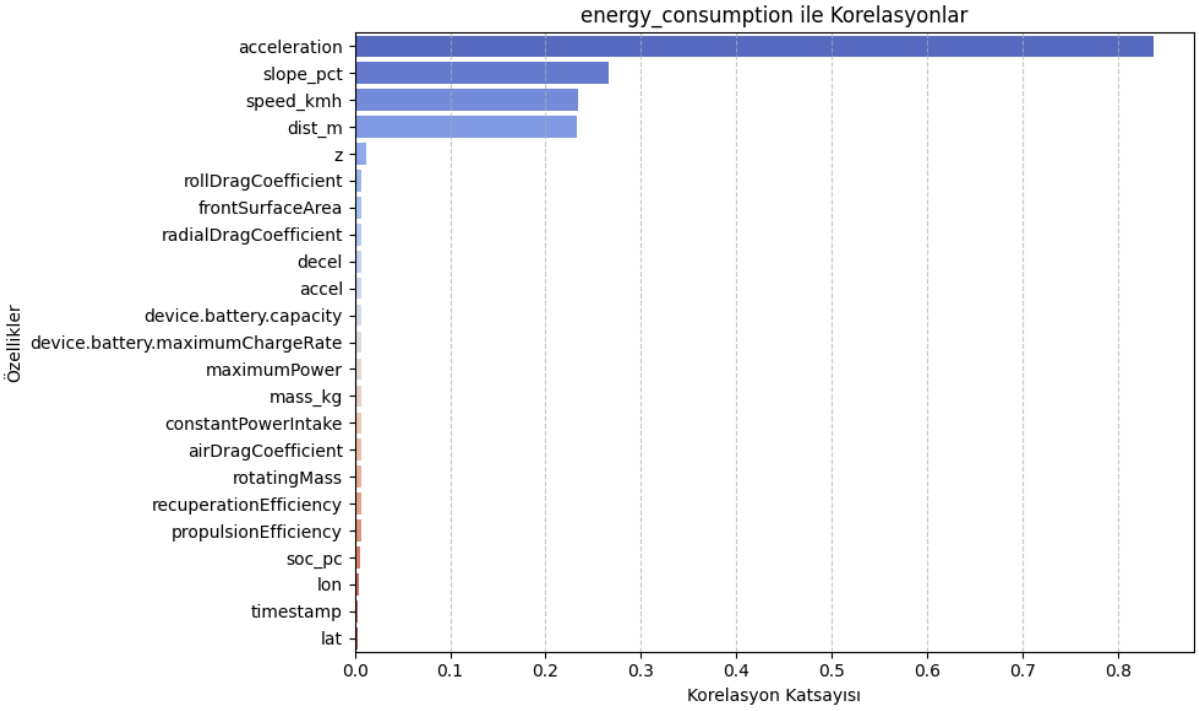
Yöntem: Her sayısal değişken için (hedef değişken hariç) Çeyrekler Arası Aralık (IQR) kullanılarak aykırı değer kontrolü yapılmıştır. Bir değişken için ve hesaplanmış, bulunmuştur. Aykırılar, dışındaki gözlemler olarak tanımlanmıştır.

Çıktılar: Her değişken özelinde aykırı değer adet, oran (%), min, max, Q1 ve Q3 istatistikleri üretilmiş ve orana göre sıralanmıştır. Bu özet tablo, model öncesi gerekirse değişkene özel dönüştürme/kaplama stratejileri belirlemek için kullanılmıştır.

Gerekçe: Aykırı değerler, regresyon tabanlı ve ağaç tabanlı modellerin genelleme performansını olumsuz etkileyebilir; erken tespit, temkinli müdahale için gereklidir. Bu çalışmada aykırılar yalnızca raporlanmış, fiziksel anlamla çelişen durumlar ayrıca özellik mühendisliği adımlarında ele alınmıştır (bkz. eğim sınırları)

3.2.3 Energy Consumption ile Korelasyon Analizi

Bu başlık altında, hedef değişken olan energy_consumption ile veri setindeki tüm sayısal değişkenlerin korelasyon ilişkisi incelenmiştir.



3.2.4 Bulgular

En yüksek pozitif korelasyona sahip özellikler sırasıyla acceleration, slope_pct, speed_kmh ve dist_m olmuştur.

acceleration değişkeni, energy_consumption ile oldukça yüksek pozitif korelasyona sahiptir (0.83 civarında). Bu, hızlanmanın enerji tüketimine güçlü bir etkisi olduğunu göstermektedir.

slope_pct ve speed_kmh değerleri de anlamlı düzeyde pozitif ilişki göstermektedir; bu, yokuş çıkma ve yüksek hızın enerji tüketimini artırdığını doğrular.

Diğer fiziksel katsayılar (ör. rollDragCoefficient, frontSurfaceArea, radialDragCoefficient) nispeten düşük korelasyonlara sahip olsalar da, modelde fiziksel anlamlılık açısından önem taşımaktadır.

3.3 Özellik Mühendisliği ve Yeni Değişken Üretimi

Bu bölümde ihtiyaçlara göre yeni değişken üretimi yapılmıştır.

3.3.1 Eğim Değerlerinin Sınırlandırılması ve Eksiklerin Doldurulması

Sınırlandırma: slope_pct değeri fiziksel olarak anlamlı aralık olan -50% ile +50% arasında tutulmuştur.

Eksiklerin Doldurulması: Eksik slope_pct değerleri 0 ile doldurulmuştur. Bu varsayım, aracın hareket etmediği veya rakımın değişmediği zamanlarda eğimin 0 olduğu kabulüne dayanmaktadır.

3.3.2 İvme ve Eğim Ayrımı (Pozitif/Negatif)

İvme:

- acc_pos: Pozitif ivme (hızlanma) değerleri.
- acc_neg: Negatif ivme (yavaşlama) değerleri.

Eğim:

- slope_pct_pos: Pozitif eğim (yokuş yukarı) değerleri.
- slope_pct_neg: Negatif eğim (yokuş aşağı) değerleri.

Gerekçe: İşaret ayrımı ile enerji tüketimi ve rejeneratif enerji kazanımı ayrı ayrı incelenebilmekte, bu da modelin fiziksel süreçleri daha doğru temsil etmesini sağlamaktadır.

3.3.3 Hızın Türetilmiş Özellikleri

- speed_ms: Hızın m/s cinsine dönüştürülmüş hali.
- v2: Hızın karesi (kinetik enerji ve aerodinamik sürükleme ile ilişkili).

3.3.4 Aerodinamik Özellik

- CdA: airDragCoefficient ile frontSurfaceArea çarpımı. Aerodinamik sürükleme kuvvetini doğrudan temsil eden bileşik bir özellik.

3.3.5 Özellik Seçimi

Seçilen Özellikler: v2, acc_pos, acc_neg, slope_pct_pos, slope_pct_neg, mass_kg, CdA, rollDragCoefficient, propulsionEfficiency, recuperationEfficiency, maximumPower.

Gerekçe: Bu özellikler, enerji tüketimini belirleyen temel fiziksel ve dinamik faktörleri temsil etmektedir.

3.4 Grup-Tutarlı Veri Bölme (Train/Val/Test)

GroupShuffleSplit kullanılarak vehicle_id bazlı grup-tutarlı veri bölme yapılmıştır.

3.4.1 Yöntem

Oranlar: Test %15, Validasyon %15, Eğitim %70.

3.4.2 Güvenlik Kontrolleri

Her vehicle_id yalnızca bir veri kümesinde yer alacak şekilde kontrol edilmiştir.

Bölünmüş veri setlerinin dağılımı ve araç başına satır sayıları analiz edilmiştir.

3.4.3 Çıktılar

Train, Val, Test kümelerindeki araç sayıları ve örnek listeler raporlanmıştır.

Araç başına satır sayısı bilgisi CSV olarak kaydedilmiştir (rows_per_vehicle.csv).

4 Makine Öğrenmesi Modelleri

4.1 Giriş

Eğitim ve test verisi üzerinden yapılan değerlendirmelerde, modelin performans metrikleri aşağıdaki gibidir:

- **MAE (Mean Absolute Error):** Ortalama mutlak hata, modelin tahminlerinin gerçek değerlerden ortalama sapmasını gösterir.
- **RMSE (Root Mean Squared Error):** Büyük sapmalara daha duyarlı hata ölçümüdür.
- **R^2 (Determination Coefficient):** Modelin bağımlı değişkendeki varyansın ne kadarını açıkladığını gösterir.

4.2 Linear Regresyon Modeli

Bu başlık altında, seçilen özellikler ile enerji tüketimini tahmin etmek amacıyla oluşturulan Linear Regresyon modeli ve sonuçları sunulmaktadır.

4.2.1 Veri Hazırlığı

Eksik Verilerin Temizlenmesi: Eğitim, doğrulama ve test veri kümelerindeki NaN değerler satır bazında çıkarılmıştır.

Özellik/Hedef Ayrımı: energy_consumption hedef değişken olarak ayrılmış, diğer tüm seçili sütunlar özellik seti olarak kullanılmıştır.

Ölçekleme: Tüm sayısal özellikler MinMaxScaler ile 0-1 aralığında ölçeklenmiştir. Bu, modelin katsayılarının karşılaştırılabilir hale gelmesini sağlar.

4.2.2 Model Eğitimi

Model: LinearRegression sınıfı kullanılarak temel doğrusal regresyon modeli eğitilmiştir.

Eğitim: Model, ölçeklenmiş eğitim verisi üzerinde fit edilmiştir.

4.2.3 Model Değerlendirme Metrikleri

Aşağıdaki metrikler, eğitim (train), doğrulama (validation) ve test setleri için hesaplanmıştır:

MAE (Mean Absolute Error): Ortalama mutlak hata.

RMSE (Root Mean Squared Error): Hataların karelerinin ortalamasının karekökü.

R^2 (Determination Coefficient): Açıklanan varyans oranı.

Elde edilen sonuçlar aşağıdaki gibidir:

```
Train -> MAE: 3.84 | RMSE: 6.30 | R²: 0.799
Validation -> MAE: 3.86 | RMSE: 6.32 | R²: 0.788
Test -> MAE: 4.26 | RMSE: 6.96 | R²: 0.796
```

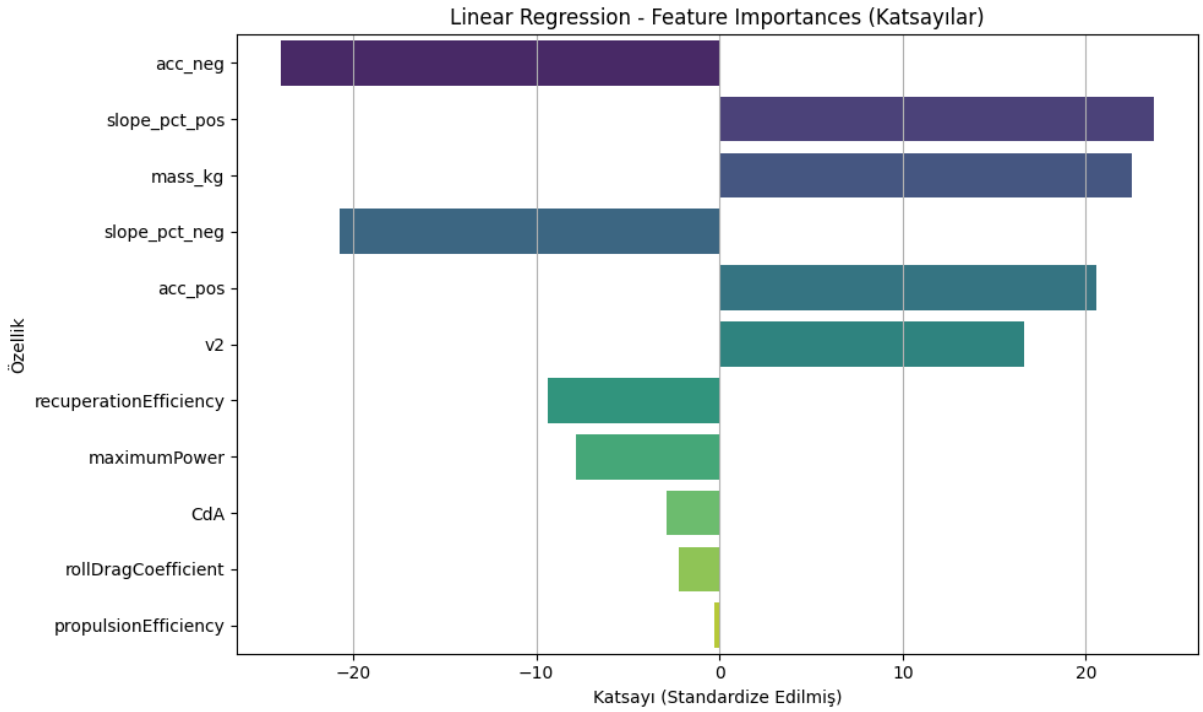
Bu metrikler, modelin eğitim ve doğrulama performansının tutarlı olduğunu ve test setinde de benzer doğrulukta çalıştığını göstermektedir.

4.2.4 Katsayı Analizi

Modelin her bir özelliğe verdiği katsayılar, mutlak değerlerine göre büyükten küçüğe sıralanmıştır.

- En yüksek pozitif katsayılar: slope_pct_pos, mass_kg, acc_pos.
- En yüksek negatif katsayılar: acc_neg, slope_pct_neg.

Görselleştirme: Çubuk grafik ile katsayıların pozitif/negatif etkileri ve büyüklükleri sunulmuştur.



4.2.5 Yorumlar

Pozitif katsayılar (ör. slope_pct_pos, mass_kg) enerji tüketimini artıran fiziksel faktörleri temsil etmektedir.

Negatif katsayılar (ör. acc_neg, slope_pct_neg) ise rejeneratif enerji kazanımı veya daha düşük tüketimle ilişkilidir.

Model, temel doğrusal ilişkileri anlamak için güçlü bir referans noktası sunmaktadır ve diğer karmaşık modellerle karşılaştırmalarda başlangıç referansı olarak kullanılabilir.

4.3 Random Forest

Bu çalışmada **Random Forest** regresyon modeli kullanılmıştır. Random Forest, birden fazla karar ağacının (decision tree) birlikte çalıştığı topluluk (ensemble) öğrenme yöntemidir. Her bir karar ağacı, eğitim verisinin rastgele seçilmiş bir alt kümesi üzerinde eğitilir ve sonuçlar ortalama alınarak nihai tahmin üretilir.

Model seçimi, veri yapısının özellikleri, hedef değişkenin doğası ve istenen çıktı kriterleri göz önüne alınarak yapılmıştır.

- **Doğrusal Olmayan İlişkileri Yakalama**: Enerji tüketimi verilerinde lineer olmayan karmaşık ilişkiler bulunabilir. Random Forest, bu ilişkileri yakalamada güçlüdür.
- **Overfitting Riskinin Azalması**: Tek bir karar ağacı aşırı öğrenme (overfitting) riski taşırken, birden fazla ağacın bir araya gelmesi bu riski önemli ölçüde azaltır.
- **Özellik Önem Düzeylerini Belirleyebilme**: Model, enerji tüketimini en çok etkileyen faktörlerin belirlenmesinde önemli bir avantaj sağlar

4.3.1 Model Hiperparametreleri ve Eğitim

```
rf_model = RandomForestRegressor(
    n_estimators=100, #Ormandaki ağaç sayısı
    max_depth=10,
    min_samples_split=5,
    min_samples_leaf=2,
    random_state=42,
    n_jobs=-1
)
```

Model, aşağıdaki parametreler ile eğitilmiştir:

- n_estimators=100: 100 farklı karar ağacı kullanılmıştır, bu sayede tahminlerin stabil ve güvenilir olması sağlanmıştır.
- max_features='sqrt': Her ağacın her düğümünde rastgele seçilen özellik sayısı karekök olarak belirlenmiş, böylece ağaçlar arasında çeşitlilik artırılmıştır.
- max_depth=30: Her bir ağacın maksimum derinliği 30 ile sınırlandırılmış, bu sayede çok derin ağaçların aşırı öğrenmesi engellenmiştir.
- min_samples_split=5: Bir düğümün bölünebilmesi için minimum 5 örnek gereklidir.
- min_samples_leaf=2: Yaprak düğümde en az 2 örnek bulunmalıdır.
- bootstrap=True: Her ağaç eğitim setinden rastgele örneklerle beslenmiştir, böylece model genelleme kabiliyetini artırmıştır.

4.3.2 Model Değerlendirme Metrikleri

```
[ ] print(f"Validation -> MSE: {mse_val:.4f} | RMSE: {rmse_val:.4f} | MAE: {mae_val:.4f} | R²: {r2_val:.4f}")
    print(f"Test       -> MSE: {mse_test:.4f} | RMSE: {rmse_test:.4f} | MAE: {mae_test:.4f} | R²: {r2_test:.4f}")

Validation -> MSE: 9.0440 | RMSE: 3.0073 | MAE: 1.6049 | R²: 0.9520
Test       -> MSE: 9.2826 | RMSE: 3.0467 | MAE: 1.6925 | R²: 0.9608
```

Veri Seti	MAE	RMSE	R^2
Doğrulama	6.7742	2.6027	0.9515
Test	1.6925	3.0467	

- R^2 değerlerinin %95'in üzerinde olması, modelin enerji tüketimindeki varyansın büyük kısmını açıkladığını gösterir.
- Doğrulama ve test sonuçlarının birbirine çok yakın olması, modelin genelleme yeteneğinin yüksek olduğunu ve aşırı uyum yapmadığını gösterir.

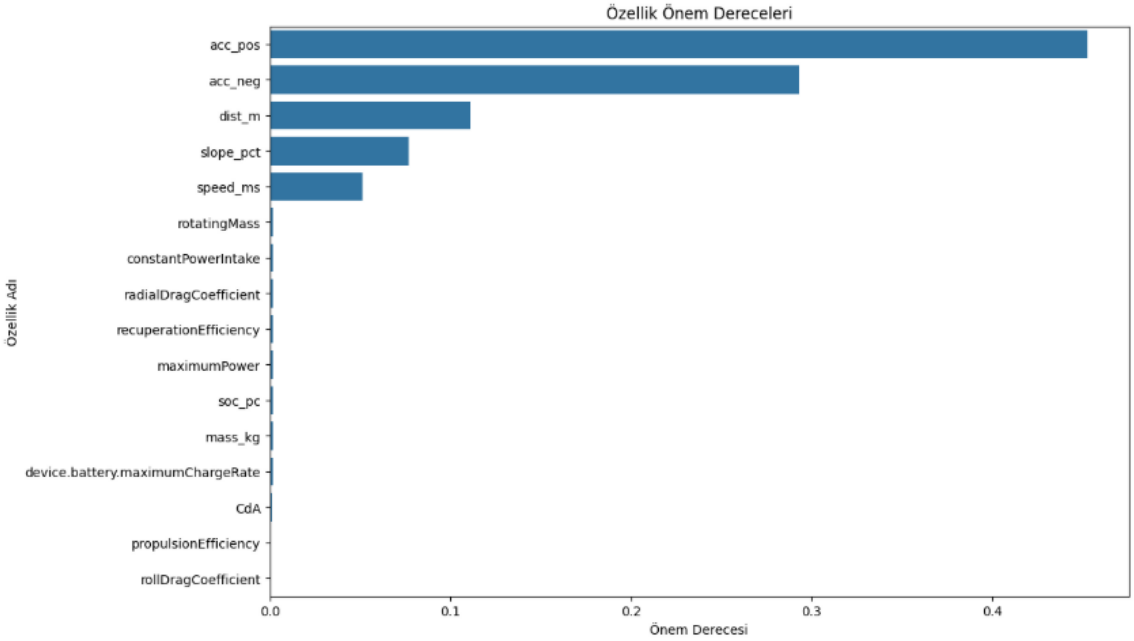
4.3.3 Özellik Önem Analizi

Modelin hesapladığı en önemli beş özellik:

1. acc_pos — 0.4528
2. acc_neg — 0.2933
3. dist_m — 0.1110
4. slope_pct — 0.0766
5. speed_ms — 0.0515

Yorum :

- **Pozitif ivme (acc_pos)**: Hızlanma sırasında motorun enerji talebi keskin şekilde artar.
- **Negatif ivme (acc_neg)**: Rejeneratif frenleme ile enerji kazanılsa da, yavaşlama döngüsü tüketimi dolaylı olarak etkiler.
- **Mesafe (dist_m)**: Daha uzun mesafeler, toplam enerji harcamasını artırır.
- **Eğim (slope_pct)**: Yokuş çıkma, yerçekimine karşı ek güç gerektirir.
- **Hız (speed_ms)**: Hız arttıkça aerodinamik kayıplar hızın karesi ile artar.



4.3.4 SHAP Analizi

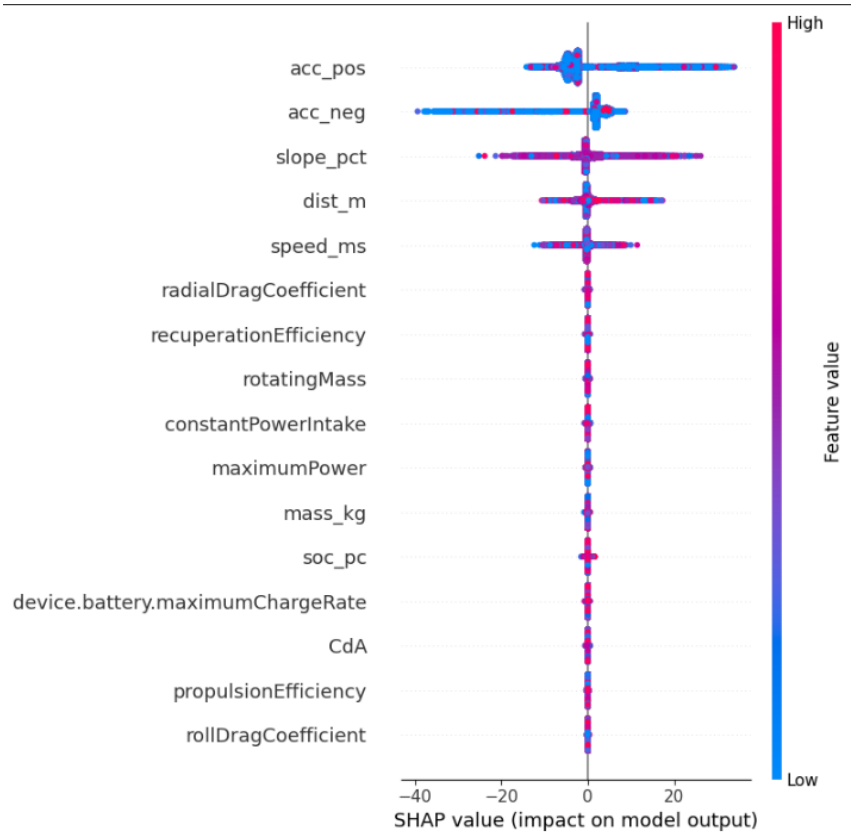
Random Forest, hangi değişkenlerin önemli olduğunu global düzeyde söyler, ancak her bir tahmin için hangi değişkenin ne kadar katkı yaptığını söylemez. SHAP, her örnekte hangi özelliğin pozitif veya negatif katkı yaptığını açıklar.

Özelliklerin Yorumlanması:

1. acc_pos (Pozitif İvmelenme): Bu özellik, model çıktısı üzerinde en büyük etkiye sahiptir.

- Yüksek acc_pos değerleri model çıktısını pozitif yönde çok güçlü bir şekilde artırıyor. Bu, pozitif ivmelenmenin enerji tüketimine çok güçlü bir şekilde katkıda bulunduğunu gösteriyor.
- Düşük acc_pos değerleri ise model çıktısını pozitif veya negatif yönde daha az etkiliyor.

2. acc_neg (Negatif İvmelenme):
   - Yüksek acc_neg model çıktısını negatif yönde güçlü bir şekilde etkiliyor.
   - Düşük acc_neg değerleri ise model çıktısını artırıcı yönde etkiliyor.
3. slope_pct (Eğim Yüzdesi):
   - Yüksek slope_pct değerleri pozitif bir etkiye sahipken, düşük slope_pct değerleri model çıktısını negatif yönde etkiliyor. Bu, yokuş yukarı gitmenin enerji tüketimini artırdığını, yokuş aşağı gitmenin ise azalttığını düşündürüyor.
4. dist_m (Mesafe):
   - Hem yüksek hem de düşük dist_m değerleri model çıktısını pozitif yönde etkileyebilir. Dağılıma bakıldığında, daha uzun model çıktısını artırıcı etki yaptığı görülüyor.
5. speed_ms (Hız):
   - Yüksek hız değerleri pozitif bir etkiye sahipken, düşük hız değerleri model çıktısını daha az etkiliyor. Hız arttıkça enerji tüketiminin de arttığını gösteriyor olabilir.



4.4 Random Forest (Hiperparametre optimizasyonlu)

Bu bölümde, doğrusal varsayımlara bağlı kalmayan, etkileşimleri ve doğrusal olmayan ilişkileri yakalayabilen Random Forest Regresyonu kullanılmıştır. Bu Random Forest modeli az öncekine kıyasla hiperparametre optimizasyonu yapılmış versiyonudur.

4.4.1 Veri Hazırlığı

Ölçekleme: Random Forest gibi ağaç tabanlı yöntemlerde ölçekleme gerekmediği için yalnızca NaN değerler temizlenmiştir.

Özellik/Hedef Ayrımı: energy_consumption hedef değişken olarak ayrılmış, geri kalan seçili sütunlar özellik setini oluşturmuştur.

4.4.2 Model ve Hiperparametre Optimizasyonu

Algoritma: RandomForestRegressor.

Hiperparametre Arama: GridSearchCV ile n_estimators, max_depth, min_samples_split, min_samples_leaf, max_features parametreleri farklı kombinasyonlarda denenmiştir.

Seçilen En İyi Parametreler:

- n_estimators: 200
- max_depth: 20
- max_features: 'sqrt'
- min_samples_leaf: 1
- min_samples_split: 2
- random_state: 42

4.4.3 Nihai Modelin Eğitilmesi

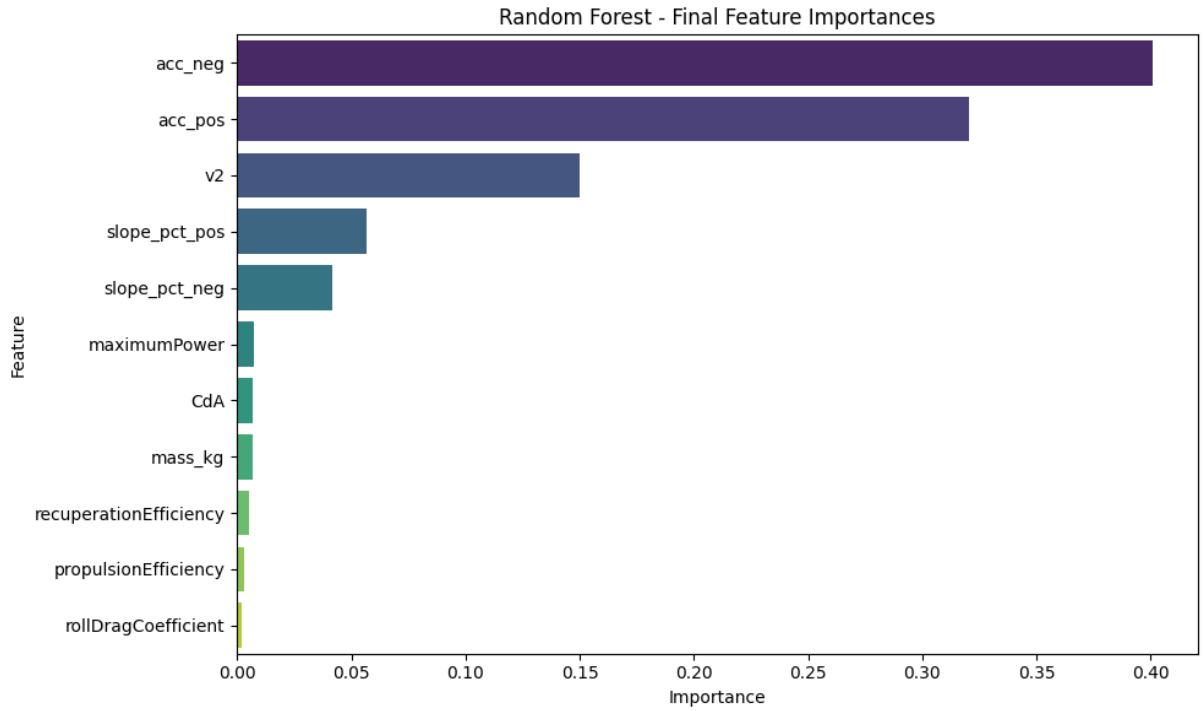
Seçilen en iyi parametrelerle model yeniden eğitilmiştir.

4.4.4 Model Değerlendirme Metrikleri ve Sonuçları

```
Train -> MAE: 0.77 | RMSE: 1.57 | R²: 0.988
Validation -> MAE: 1.28 | RMSE: 2.82 | R²: 0.958
Test -> MAE: 1.34 | RMSE: 2.81 | R²: 0.967
```

4.4.5 Özellik Önem Düzeyleri

Modelin hesapladığı feature_importances_ değerleri, tahminlerde en etkili değişkenleri ortaya koymaktadır.



Görselleştirmeye göre en yüksek öneme sahip ilk üç değişken: acc_neg, acc_pos ve v2.

acc_neg (yavaşlama) en yüksek öneme sahiptir; bu durum, rejeneratif enerji kazanımı veya enerji tüketimindeki azalma süreçlerinin model kararlarında kritik rol oynadığını gösterir.

acc_pos (hızlanma) ve v2 (hızın karesi) de güçlü belirleyicilerdir, yüksek enerji tüketimiyle doğrudan ilişkilidir.

slope_pct_pos ve slope_pct_neg ise eğim etkilerini yansıtarak ikinci derecede önem taşır.

maximumPower, CdA ve mass_kg gibi fiziksel araç özellikleri daha düşük ama anlamlı katkı sağlar.

4.4.6 Yorumlar

Random Forest, doğrusal olmayan ilişkileri de yakalayabildiğinden Linear Regression'a göre daha esnek bir modeldir.

Özellik önem sıralaması, modelin kararlarında hızlanma/yavaşlama ve hızın kareli etkisinin baskın olduğunu; fiziksel araç parametrelerinin ise destekleyici rol oynadığını açıkça göstermektedir.

4.4.7 Örnek Araç Tahmin Analizi (veh103)

Trip Bazlı Karşılaştırma:

```
veh103 - Trip 1: Gerçek = 2.35 | Tahmin = 2.76
veh103 - Trip 2: Gerçek = 7.02 | Tahmin = 7.32
veh103 - Trip 3: Gerçek = 11.74 | Tahmin = 12.58
veh103 - Trip 4: Gerçek = 16.52 | Tahmin = 16.62
veh103 - Trip 5: Gerçek = 5.67 | Tahmin = 5.86
veh103 - Trip 6: Gerçek = -0.14 | Tahmin = 0.02
veh103 - Trip 7: Gerçek = -0.71 | Tahmin = -0.81
veh103 - Trip 8: Gerçek = -0.69 | Tahmin = -0.83
veh103 - Trip 9: Gerçek = -0.71 | Tahmin = -0.83
veh103 - Trip 10: Gerçek = -0.71 | Tahmin = -0.83
```

Toplam Enerji Tüketimi:

```
Araç: veh103
🔹 Gerçek Toplam Tüketim      : 2558.55 Wh
🔹 Tahmin Edilen Toplam Tüketim: 2526.00 Wh
🔹 Fark                       : -32.55 Wh (-1.27%)
```

4.5 XGBoost

XGBoost (**Extreme Gradient Boosting**), karar ağaçlarını temel alan, klasik gradient boosting'in performans ve doğruluk açısından geliştirilmiş sürümüdür. Her adımda, önceki tahminlerin hatalarını azaltmak için yeni ağaçlar ekler. XGBoostun özellikleri:

- Hem 1. türev (gradyan) hem 2. türev (Hessian) kullanarak daha sağlam optimizasyon yapar.
- L1 ve L2 cezaları ile overfitting'i sınırlar.
- Eksik verileri kendisi yönetir, dal yönünü otomatik seçer.
- Bölünme noktalarını paralel hesaplayarak hızlı çalışır.
- Gereksiz dalları budayarak modeli sadeleştirir.
- Her iterasyonda rastgele özellik seçerek çeşitlilik katar.

XGBoost'un hızlı, büyük verilerde ölçeklenebilir, farklı problem tiplerinde esnek olmasi gibi avantajları varken aynı zamanda küçük veri setlerinde overfitting riski taşır ama bizim projemizdeki veriseti büyük olduğundan bu modeli kullanmanın uygun olduğunu düşündük.

Bu çalışmada XGBoost modeli, veri setinin yapısı ve tahmin görevine uygun özellikleri nedeniyle tercih edilmiştir.

- Veri setinde hem sayısal özellikler (hız, ivme, kütle vb.) hem de kategorik bir özellik (vehicle_type) bulunuyor. XGBoost, bu farklı tipteki verilerle sorunsuz çalışabiliyor.
- Enerji tüketimi, birçok faktörün doğrusal olmayan etkileşimlerinden etkileniyor (örneğin, aerodinamik sürüklenme hızın karesiyle artar, eğimin etkisi aracın kütlesine bağlıdır). Ağaç tabanlı bir topluluk modeli olan XGBoost, bu karmaşık ilişkileri otomatik olarak öğrenebilir.
- Tahmin edilmek istenen `energy_consumption` sürekli bir sayısal değerdir. XGBoost'un regresyon yetenekleri bu problem için doğrudan uygundur.
- Veri temizliği yapılmış olsa da gerçek dünyadan gelen verilerde gürültü veya aykırı değerler olabilir. Ağaç tabanlı modeller, lineer modellere göre bu durumlara daha az duyarlıdır.

- Geriye dönük özellikler (lag features), hareketli istatistikler gibi eklediğin mühendislik özellikleri, modele aracın yakın geçmişi ve dinamikleri hakkında daha fazla bilgi sağlar. XGBoost bu tür ek bilgileri verimli şekilde kullanabilir.
- XGBoost yüksek performansıyla bilinir ve veri setinin boyutunu verimli şekilde işleyebilir. GPU hızlandırması ile eğitim süresi daha da kısaltılabilir.

Özetle, XGBoost; veri setindeki farklı ve karmaşık özelliklerden etkin şekilde öğrenerek sürekli hedef değişken olan enerji tüketimini tahmin etme konusunda güçlü bir model olduğundan tercih edilmiştir.

4.5.1 Model Eğitimi

Modelde **objective='reg:squarederror'** sürekli hedef değişken olan *energy_consumption* için kare hata minimizasyonu yapan standart regresyon amacıyla belirlenmiş, **eval_metric='rmse'** ise büyük hatalara duyarlı olduğu için enerji tüketiminde kritik sapmaları önlemeye yardımcı olmuştur. **early_stopping_rounds=400** parametresi, doğrulama setinde 400 tur boyunca iyileşme olmadığında eğitimi durdurarak overfitting'i engellerken; **eta=0.03** ile **num_boost_round=10000** küçük adımlarla daha sağlam bir optimizasyon sağlamış ve erken durdurma sayesinde en uygun noktada bırakılmıştır. **max_depth=5** ağaçların aşırı karmaşıklaşmasını engelleyerek veri gürültüsüne aşırı uyumu azaltmış, **subsample=0.5** ve **colsample_bytree=0.8** ise veri ve özellik alt örnekleme yoluyla çeşitlilik sağlayarak modelin varyansını düşürmüştür.

```
# Initialize and train the XGBoost
xgb_params = {
    'objective': 'reg:squarederror',
    'eval_metric': 'rmse',           # Early stopping monitors RMSE on eval
    'eta': 0.03,                     # Learning rate
    'max_depth': 5,
    'subsample': 0.5,                # Fraction of samples used for fitting the trees
    'colsample_bytree': 0.8,         # Fraction of features used for fitting the trees/ reduce variance.
    'random_state': SEED,
    'tree_method': 'gpu_hist',
    'device': 'cuda' if USE_GPU else 'cpu'
}
```

```
[ ]  VAL_xgb = metric_dict(y_val, y_val_pred_xgb)
     TEST_xgb = metric_dict(y_test, y_test_pred_xgb)

     print("XGBoost VALID:", VAL_xgb)
     print("XGBoost TEST :", TEST_xgb)

     XGBoost VALID: {'MAE': 0.9407826633620383, 'RMSE': 2.384532212466926, 'R2': 0.9592503350749136}
     XGBoost TEST : {'MAE': 0.9934570635836802, 'RMSE': 2.376027721795694, 'R2': 0.968261037867403}
```

Model Performans Sonuçları (XGBoost)

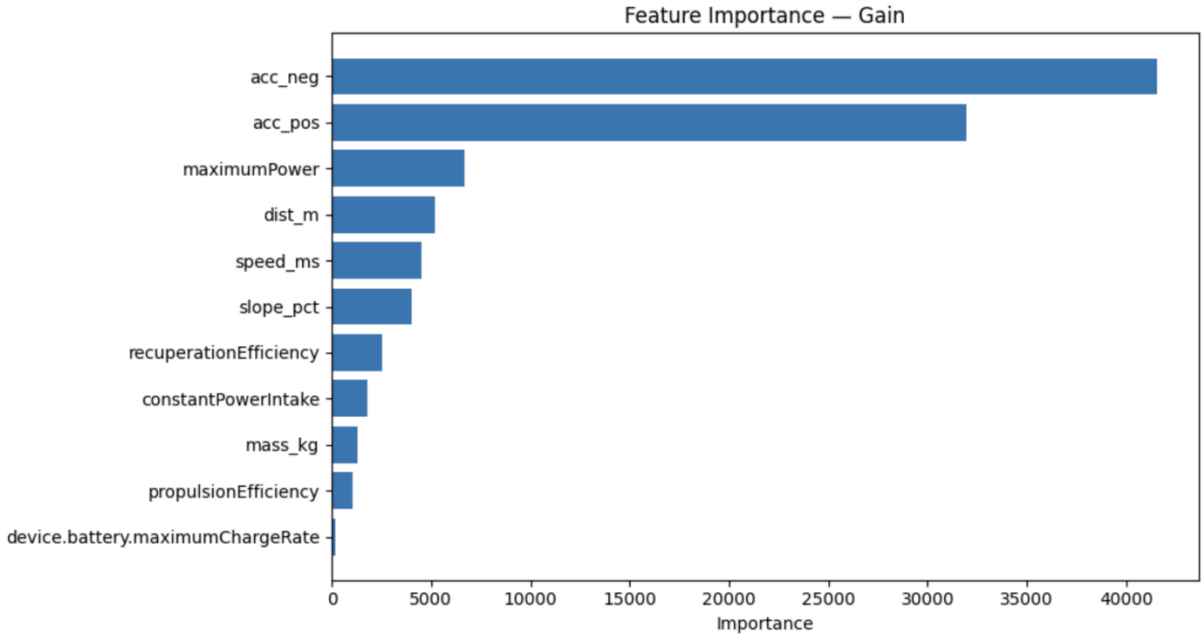
- **Doğrulama seti:** MAE = **0.94**, RMSE = **2.38**, R^2 = **0.959**
- **Test seti:** MAE = **0.99**, RMSE = **2.38**, R^2 = **0.968**

Model, doğrulama ve test setlerinde düşük hata değerleri ve yüksek R^2 skorları ile enerji tüketimi tahmininde yüksek doğruluk sağlamıştır. RMSE'nin düşük olması, büyük hataların minimum seviyede olduğunu; R^2 değerlerinin %95'in üzerinde olması ise modelin hedef

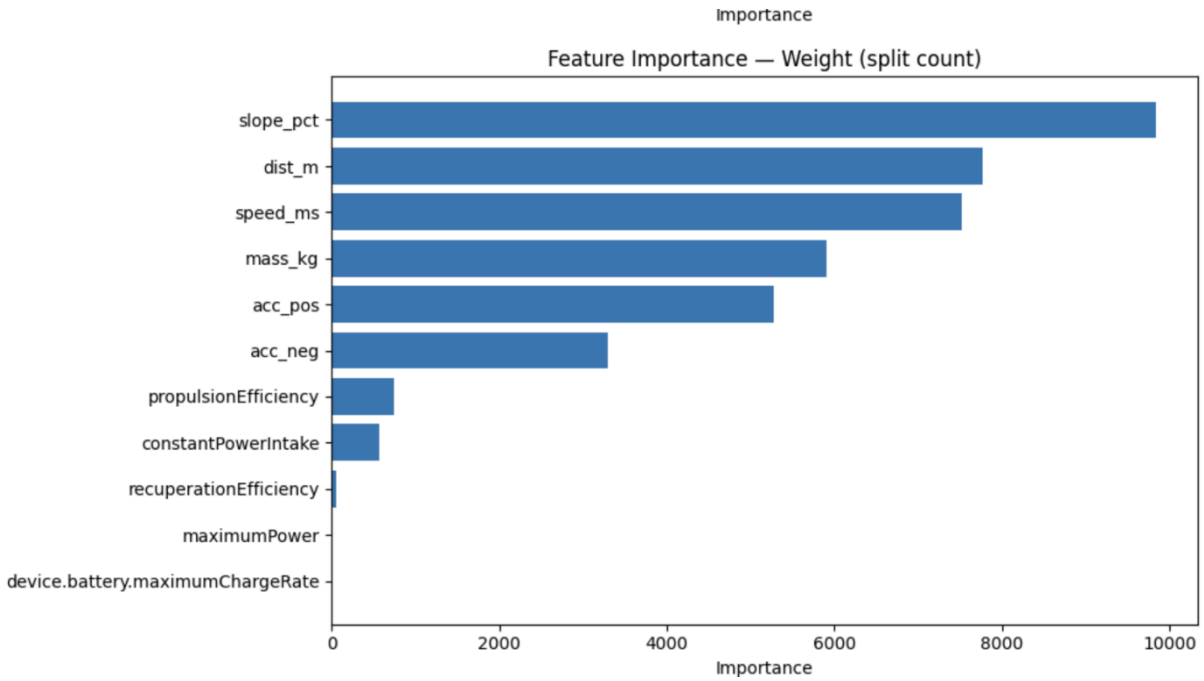
değişken varyansının büyük kısmını açıkladığını göstermektedir. Bu sonuçlar, modelin hem öğrenme hem de genelleme açısından başarılı olduğunu ortaya koymaktadır.

4.5.2 Importance Analizi

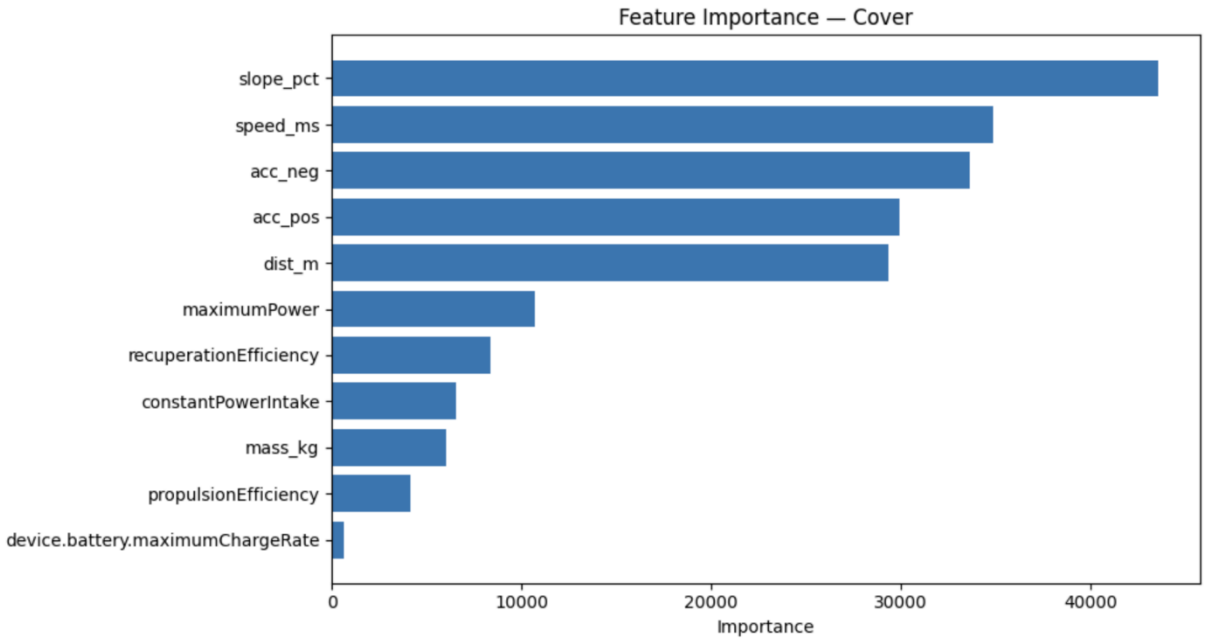
Modelin güvenebiliriğini ve açıklanabilirliğini analiz edebilmek için feature importance incelendi.



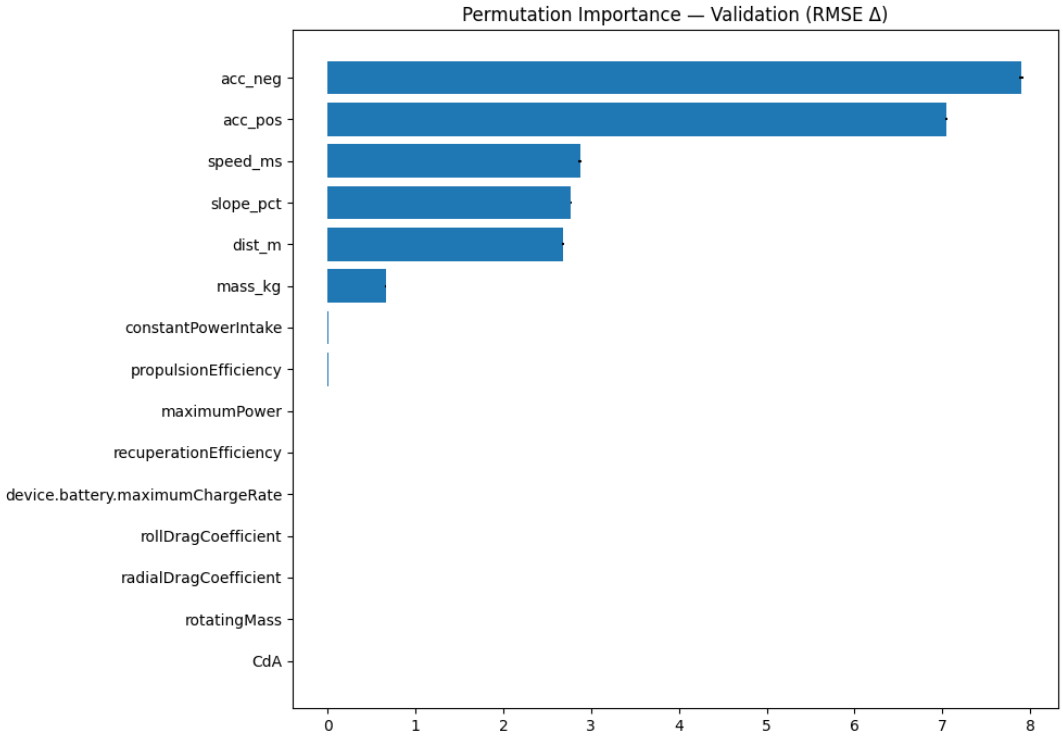
Gain (hata azaltma etkisi) açısından en etkili özellikler, diğerlerinden açık ara önde olan **acc_neg** ve **acc_pos** olmuştur. Bu durum, yavaşlama ve hızlanma paternlerinin RMSE'yi düşürmede en büyük katkıyı sağladığını ve araç dinamiklerinin enerji tüketimi üzerinde en güçlü doğrudan etkiye sahip olduğunu göstermektedir. Dikkate değer diğer katkılar arasında **maximumPower** ve **dist_m** yer almakta olup, bunlar enerji tüketiminin beklenen fiziksel belirleyicileriyle uyumludur.



Weight (bölünme sıklığı) grafiğinde ise öne çıkan özellikler **slope_pct**, **dist_m** ve **speed_ms** olmuştur. Bu değişkenler veri setini sıkça bölmek için kullanılmıştır ancak sık kullanım her zaman yüksek tahmin gücü anlamına gelmez; bu özellikler, orta düzey kazanç sağlayan kolay bölünme noktaları olabilir. Örneğin, **slope_pct** yüksek ağırlığa sahip olmasına rağmen görece düşük bir gain değerine sahiptir; bu da yaygın kullanıldığını ancak her bölünmede sağladığı iyileştirmenin sınırlı olduğunu göstermektedir.



Cover (bölünmelerden etkilenen ortalama örnek sayısı) grafiğinde yine **slope_pct** ve **speed_ms** öne çıkmaktadır. Bu da bu özelliklerin, veri setinin büyük kısmını etkileyen geniş kapsamlı "ilk katman" bölünme kriterleri olarak görev yaptığını, buna karşılık gain değeri yüksek olan hızlanma özelliklerinin ise daha çok daha küçük alt kümelerde tahminleri iyileştirmek için kullanıldığını göstermektedir



Bu adımda, modeldeki her bir özelliğin tahmin performansına doğrudan katkısını ölçmek için **permutation feature importance** yöntemi uygulanmıştır. Yöntemde, doğrulama setinde yalnızca tek bir özelliğin değerleri rastgele karıştırılmış, diğer tüm özellikler sabit tutulmuştur. Bu değişiklik sonrası modelin RMSE değerindeki artış (**ΔRMSE**) hesaplanmış ve bu artış, ilgili özelliğin önem derecesi olarak değerlendirilmiştir.

Her özellik için bu işlem **n** kez (varsayılan 5) tekrarlanmış, elde edilen ΔRMSE değerlerinin ortalaması ve standart sapması bulunmuştur. Ortalama değer, özelliğin hata azaltma açısından katkısını; standart sapma ise bu etkinin kararlılık düzeyini göstermektedir. Standart sapması yüksek olan özellikler, modelin farklı koşullar altında bu özelliğe olan bağımlılığının değişken olabileceğini işaret etmektedir.

Bu yaklaşım, ağaç içi metriklere (gain, weight, cover) bağlı kalmadan **"Bu özelliği bozarsam modelin hatası ne kadar artar?"** sorusuna doğrudan yanıt verir. Böylece modelin gerçek anlamda hangi özellikleri kullandığı ve bunların tahmin performansına olan kritik katkısı net bir şekilde ortaya konmuştur. Daha sonraki iyileştirmelerde katkısı az olanlar çıkartılabilir şuanlık bu adım yapılmamıştır.

4.5.3 Hiperparametre Optimizasyonu — Optuna + XGBoost

Optuna, Python tabanlı açık kaynaklı bir **otomatik hiperparametre optimizasyon** kütüphanesidir. Makine öğrenmesi modellerinin hiperparametrelerini manuel deneme-yanılma yerine, sistematik ve verimli bir şekilde arayarak en iyi sonuç veren kombinasyonu bulur. Klasik grid search gibi tüm kombinasyonları körlemesine denemek yerine, önceki denemelerin sonuçlarını kullanarak sonraki denemeleri daha iyi parametre aralıklarına yönlendirir (Bayesian Optimization / TPE yöntemi). Kötü giden denemeleri **erken durdurma (pruning)** ile keserek

zaman kazandırır. scikit-learn, XGBoost, LightGBM, PyTorch gibi kütüphanelerle kolayca entegre olur.

Bu projede Optuna, XGBoost modelinin hiperparametrelerini **doğrulama seti RMSE'sini minimize edecek şekilde** optimize etmek için kullanılmıştır.

Amaç Fonksiyonu (`objective`): Her denemede Optuna, belirlenen aralıklardan rastgele bir hiperparametre seti önerdi (`trial.suggest_*`). Model bu parametrelerle eğitildi ve doğrulama setinde RMSE hesaplandı.

```
# Define the objective function for Optuna
def objective(trial):
    param = {
        'objective': 'reg:squarederror',
        'eval_metric': 'rmse',
        'eta': trial.suggest_float('eta', 0.01, 0.5),
        'max_depth': trial.suggest_int('max_depth', 3, 10),
        'subsample': trial.suggest_float('subsample', 0.6, 1.0),
        'colsample_bytree': trial.suggest_float('colsample_bytree', 0.6, 1.0),
        'lambda': trial.suggest_float('lambda', 1e-3, 10.0, log=True),
        'alpha': trial.suggest_float('alpha', 1e-3, 10.0, log=True),
        'min_child_weight': trial.suggest_float('min_child_weight', 1e-3, 10.0, log=True),
        'seed': SEED,
        'tree_method': 'gpu_hist',
        'device': 'cuda' if USE_GPU else 'cpu'
    }
```

Her denemede doğrulama seti RMSE değeri döndürüldü, Optuna bu değeri minimize etmeye çalıştı. Toplam **50 deneme (n_trials=50)** sonunda en düşük RMSE değerini veren parametre seti "en iyi" olarak belirlendi.

Bu yaklaşım sayesinde, XGBoost modeli için en uygun hiperparametre kombinasyonu sistematik ve verimli bir şekilde elde edildi. Optuna, parametre arama sürecini hızlandırdı, gereksiz denemeleri önledi ve modelin doğrulama performansını artıracak yapılandırmayı seçti.

Seçilen en iyi parametreler bestparameterse atandı ve model eğitimine direkt eklendi ve bu değerler üzerinden tekrardan eğitildi.

```
best_params = study.best_params
final_xgb_params = {
    'objective': 'reg:squarederror',
    'eval_metric': 'rmse',
    'seed': SEED,
    'tree_method': 'gpu_hist',
    'device': 'cuda' if USE_GPU else 'cpu',
    **best_params # Include the best parameters found by Optuna
}
```

```
Final XGBoost TEST performance with tuned hyperparameters:
{'MAE': 0.9577560014084242, 'RMSE': 2.355474298289739, 'R2': 0.9688077678803391}

Final XGBoost VALID performance with tuned hyperparameters:
{'MAE': 0.9023629979483117, 'RMSE': 2.3592941462291916, 'R2': 0.9601083651572653}
```

Optuna ile yapılan hiperparametre optimizasyonu sonrasında XGBoost modeli hem test hem de doğrulama setinde küçük ama istikrarlı iyileşmeler göstermiştir.

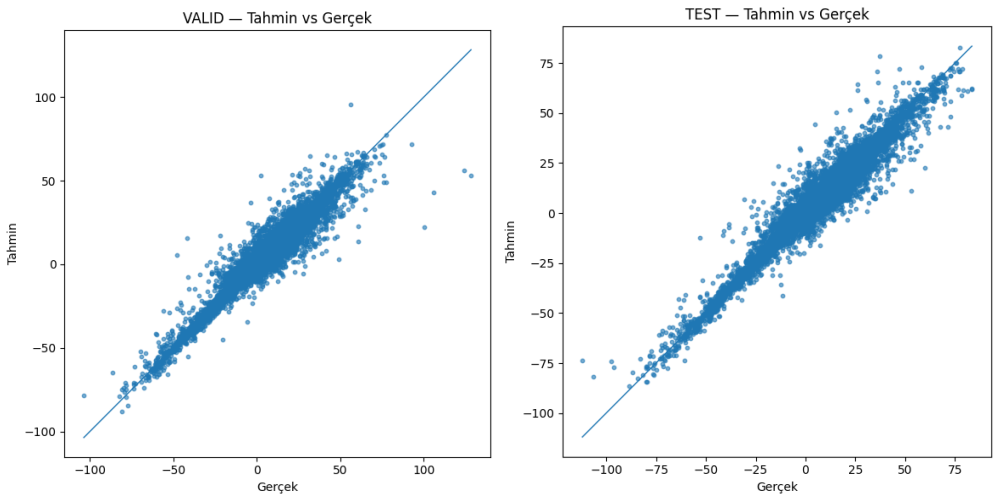
TEST Seti Performansı:

- **MMAE**: 0.99 → **0.95** (~4% improvement)
- **RMSE**: 2.38 → **2.35** (slight improvement)
- **R²**: 0.968 → **0.9689** (very small increase)

VALID Seti Performansı:

- **MAE**: 0.94 → **0.90** (~4% improvement)
- **RMSE**: 2.38 → **2.36** (slight improvement)
- **R²**: 0.959 → **0.9601** (minor increase)

Bu sonuçlar, hiperparametre optimizasyonunun modelin genel hata metriklerinde anlamlı bir şekilde iyileşme sağladığını ve özellikle ortalama mutlak hata (MAE) açısından daha isabetli tahminler ürettiğini göstermektedir.



Her iki grafikte de veri noktalarının çizgi etrafında bu kadar sıkı toplanması, modelin hem VALID hem TEST setlerinde **yüksek doğrulukla** tahmin yaptığını gösteriyor. Uç değerlerdeki sapmaların sebebi büyük olasılıkla ekstrem hız, eğim veya ivmelenme koşullarındaki fiziksel farklılıklar olabilir; bu kısım **feature engineering** ile daha da iyileştirilebilir.

4.6 CatBoost

CatBoost, karar ağaçlarını temel alan ve özellikle **kategorik verileri** etkin şekilde işleyebilen gelişmiş bir gradient boosting algoritmasıdır. Her iterasyonda, önceki tahminlerin hatalarını azaltmak için yeni ağaçlar ekler. CatBoost'un öne çıkan özellikleri:

- Eğitim sırasında hedef sızıntısını (target leakage) önler.
- One-hot encoding veya manuel label encoding gerektirmez; kendi bünyesindeki **target encoding** ve **ordered boosting** yöntemleri ile kategorik değişkenleri doğrudan işleyebilir.
- Daha hızlı tahmin ve genelleme kabiliyeti sağlar.
- Eksik verileri otomatik olarak işleyebilir.
- Büyük veri setlerinde eğitimi hızlandırır.
- Birçok hiperparametre için makul varsayılan değerlerle başlar, küçük veri setlerinde bile iyi sonuçlar verir.

Bu çalışmada CatBoost modeli, veri setinin yapısı ve tahmin görevine uygun özellikleri nedeniyle tercih edilmiştir ancak XGBoostun daha iyi bir tercih olabileceğini düşünerek üzerine çok düşülmemiştirç Bunun nedeni de verimizde daha çok sayısal veriler olduğu için CatBoost'un otomatik encoding avantajı kritik değildi. XGBoost ise hem daha fazla parametre kontrolü hem de tuning esnekliği sunuyor, bu da model performansını ince ayarlarla optimize etmeni kolaylaştırıyor.

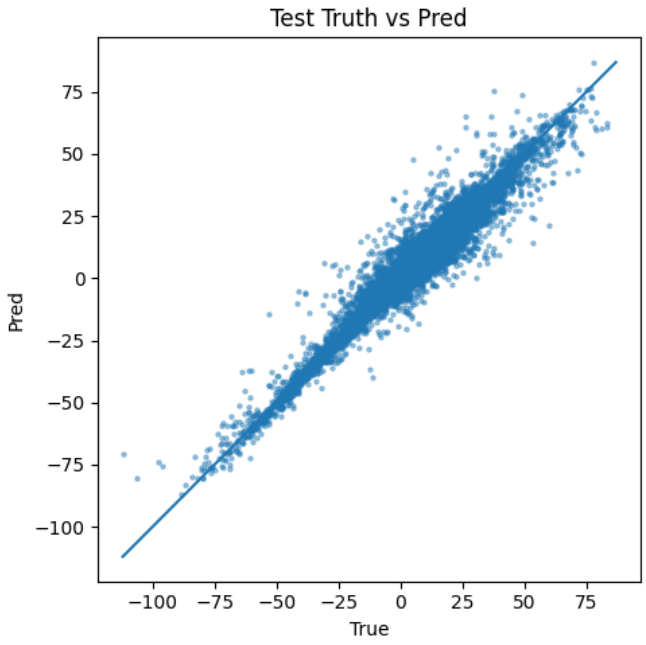
```
params = dict(
    loss_function="RMSE",
    iterations=4000,
    learning_rate=0.03,
    depth=8,
    l2_leaf_reg=6.0,
    random_state=SEED,
    od_type="Iter", od_wait=400,
    task_type=("GPU" if USE_GPU else "CPU"),
    bootstrap_type="Bayesian",
    grow_policy="Lossguide",
    verbose=250
)
```

- **loss_function="RMSE":** Modelin hata ölçümünde **Kök Ortalama Kare Hatası (Root Mean Squared Error)** kullanılacağını belirtir. Regresyon problemleri için uygundur.
- **iterations=4000:** Maksimum ağaç (boosting) sayısını belirler. Erken durdurma ile bu sayıya ulaşmadan eğitim bitebilir.
- **learning_rate=0.03:** Her iterasyonda modelin ağırlık güncelleme oranını belirler. Düşük değerler daha yavaş ama daha dengeli öğrenme sağlar.
- **depth=8:** Karar ağaçlarının maksimum derinliği. Derinlik arttıkça model karmaşıklaşır ve daha detaylı öğrenir ancak overfitting riski artar.

- **l2_leaf_reg=6.0:** L2 regularizasyon katsayısı. Yaprak düğümlerindeki ağırlıkları cezalandırarak overfitting'i sınırlar.
- **random_state=SEED:** Sonuçların tekrarlanabilir olmasını sağlar. Rastgelelik içeren işlemler bu sabit sayı ile kontrol edilir.
- **od_type="Iter" & od_wait=400: Early stopping** (erken durdurma) ayarlarıdır. Belirtilen iterasyon sayısı boyunca (400 tur) iyileşme olmazsa eğitim durur.
- **task_type=("GPU" if USE_GPU else "CPU"):** Eğitimin GPU veya CPU üzerinde yapılacağını belirler. GPU seçilirse eğitim süresi önemli ölçüde kısalır.
- **bootstrap_type="Bayesian":** Örnekleme tipidir. Bayesian bootstrap, modelin çeşitliliğini artırarak overfitting riskini azaltır.
- **grow_policy="Lossguide":** Ağacın büyüme stratejisini belirler. Lossguide, kayıp fonksiyonunu minimize eden dalları öncelikli olarak genişletir, özellikle büyük veri setlerinde verimlidir.
- **verbose=250:** Eğitim sırasında her 250 iterasyonda bir sonuç çıktısı verir, böylece ilerleme takip edilebilir.

```
CatBoost VALID: {'MAE': 0.943006399858715, 'RMSE': 2.3752876957470144, 'R2': 0.9595656847501475}
CatBoost TEST : {'MAE': 0.9975940850852829, 'RMSE': 2.3883509488251207, 'R2': 0.9679309569363453}
```

Bu parametrelerle eğittimizde model sonuçlarımızın hem doğrulama hem test setlerinde tutarlı bir şekilde yüksek R^2 ve düşük hata metrikleri üretmiş. Test setinde VALID'e yakın veya biraz daha iyi sonuçlar vermesi, modelin veriyi iyi genelleştirdiğini gösteriyor. Ancak XGBoost sonuçları ile kıyaslandığında, performans farkının küçük olduğu, dolayısıyla tercih sebebinin daha çok modelin esnekliği, tuning kabiliyeti ve topluluk desteği gibi faktörler olabileceği söylenebilir.



Yukarıdaki şekilde, yatay eksen gerçek değerleri (True), dikey eksen ise modelin tahminlerini (Pred) göstermektedir. Noktaların diyagonale yakın olması, tahminlerin gerçeğe daha yakın olduğunu gösterir.

4.7 1. Yapay Sinir Ağı (YSA) Modeli

Bu YSA denemesinde, standartlaştırılmış özellikler üzerinde BatchNorm + Dropout içeren derin bir MLP mimarisi kullanılmıştır. Amaç; doğrusal olmayan örüntüleri daha iyi yakalayıp genelleme kabiliyetini artırmaktır.

1) Veri Hazırlığı

Ön İnceleme: Veri boyutu, araç sayısı ve eksik değerlerin özeti raporlanmıştır (df.shape, nunique, isnull().sum()).

Eksik Değerler: dist_m ve slope_pct eksikleri 0 ile doldurulmuştur.

Türetilen Özellikler: acc_pos, acc_neg, speed_ms, CdA (=airDragCoefficient×frontSurfaceArea).

Sütun Temizliği: Zaman, GPS ve ham hız/ivme gibi modelle doğrudan kullanılmayacak sütunlar düşürülmüştür (örn. timestamp, lat, lon, z, accel, decel, acceleration, speed_kmh, device.battery.capacity, airDragCoefficient, frontSurfaceArea).

Grup-Tutarlı Bölme: vehicle_id bazlı GroupShuffleSplit ile train/val/test ayrımı yapılmıştır (sızıntı önleme).

Ölçekleme: Girdi özellikleri StandardScaler ile standardize edilmiştir.

2) Mimari ve Eğitim Kurulumu

MLP Katmanları: input → 256 → 128 → 64 → 1
Aktivasyon: ReLU; Düzenlileştirme: BatchNorm1d(256,128) + Dropout(0.2/0.15).

Kayıp/Optimizasyon: MSELoss, Adam (lr=1e-3, weight_decay=1e-5).

LR Planlayıcı: ReduceLROnPlateau (patience=5, factor=0.1), gradient clipping (1.0).

Erken Durdurma: Validation MAE iyileşmediğinde model ağırlıkları korunarak durdurma (patience=10).

Cihaz: GPU mevcutsa CUDA, yoksa CPU.

3) Performans (Validation/Test):

```
Epoch 14 | Val MAE: 1.0478 | RMSE: 2.4693 | R²: 0.9657
  ==========================================
  VALID set:
  MAE: 1.20
  RMSE: 2.59
  R²: 0.95
  ==============================================
  TEST set:
  MAE: 1.04
  RMSE: 2.46
  R²: 0.96
  ==========================================
```

Test metriklerinin validasyona çok yakın (hatta bazı noktalarda daha iyi) olması, modelin genelleme kabiliyetinin güçlü olduğunu göstermektedir.

4) Değerlendirme ve Notlar

BatchNorm + Dropout kombinasyonu, daha derin MLP'de stabil eğitim ve overfitting kontrolü sağlamıştır.

Bu MLP-2 sonuçları, daha önceki YSA denemesi ($R^2 \approx 0.968$) ve boosting yöntemleriyle rekabetçi düzeydedir; hata payı düşük ve kararlı bir referans model sunar.

İleri adım olarak Optuna ile mimari/öğrenme oranı/dropout taraması ve early stopping eşiğinin ayarlanmasıyla ek iyileştirme yapılabilir.

4.8 2. Yapay Sinir Ağı (YSA) Modeli

Bu bölümde, enerji tüketimini tahmin etmek amacıyla derin öğrenme tabanlı bir yapay sinir ağı (YSA) modeli uygulanmış ve sonuçları değerlendirilmiştir.

Bu bölümde, doğrusal olmayan ilişkileri daha esnek şekilde yakalamak için PyTorch tabanlı çok katmanlı ileri beslemeli bir yapay sinir ağı modeli uygulanmıştır.

4.8.1 Veri Hazırlığı ve Ölçekleme

Filtre/Temizlik: slope_pct ∈ (-50, 50) aralığı, aşırı negatif rejenerasyonlar (energy_consumption < -100) NaN olarak işaretlenip düşürülmüştür; slope_pct ve dist_m eksikleri 0 ile doldurulmuştur.

Özellikler: v2, acc_pos, acc_neg, slope_pct_pos, slope_pct_neg, mass_kg, CdA, rollDragCoefficient, propulsionEfficiency, recuperationEfficiency, maximumPower.

Grup-tutarlı bölme: vehicle_id bazlı GroupShuffleSplit ile train/val/test ayrımı korunmuştur.

Ölçekleme: Girdi özellikleri için StandardScaler (X), hedef için ayrı bir StandardScaler (y) kullanılmış; ağın kararlı eğitimine yardımcı olması için tüm tensörler float32'ye dönüştürülmüştür.

4.8.2 Mimari ve Eğitim Kurulumu

Mimari (EnergyNet): Linear(input→64) → GELU → Linear(64→32) → GELU → Linear(32→1)

Kayıp: MSELoss

Optimizasyon: Adam(lr=1e-3)

Batch boyutu / Epoch: 64 / 100

Erken durdurma mantığı: En düşük validation loss görüldüğünde en iyi ağırlıklar saklanıp eğitim

4.8.3 Model Değerlendirme Metrikleri

```
Neural Network (Test Set)
MAE : 1.28
RMSE: 2.75
R²  : 0.968
```

Yorum: YSA, hem MAE hem RMSE'yi belirgin biçimde düşürmüş ve $R^2 \approx 0.97$ ile çok yüksek açıklama gücüne ulaşmıştır.

4.8.4 Permütasyon ile Özellik Önemi (Validation/Test Üzerinde)

Permütasyon etkisine göre en yüksek etkiye sahip değişkenler: acc_pos, acc_neg, v2.

acc_pos ve acc_neg sürüş dinamiklerinin (hızlanma/yavaşlama → tüketim/regen) belirleyiciliğini teyit eder.

v2 (hızın karesi) aerodinamik/kinetik etkilerin güçlü payını yansıtır.

slope_pct_pos/neg ve kütle/araç karakteristikleri (ör. mass_kg, maximumPower, CdA) ikincil düzeyde katkı sağlar.

4.8.5 Gerçek vs. Tahmin (Test)

Saçılım grafiğinde noktalar y = x kırmızı kesikli çizgiye yakın hizada toplanmakta; uç değerlerde sınırlı sapma görülmektedir. Bu, geniş aralıkta iyi kalibrasyon ve düşük önyargıya işaret eder.

4.8.6 Model Karşılaştırması ve Notlar

LR → RF → YSA sıralamasında, doğrusal olmayanlığı modelleyebilme kapasitesi arttıkça metrikler iyileşmiştir; YSA en iyi genel performansı sunmuştur.

YSA'da hedefin de ölçeklenmiş olması, çıktıların ters dönüşümü (inverse transform) ile raporlama yapılmasını gerektirir (bu çalışma böyle yapılmıştır).

Aşırı öğrenmeye karşı: dropout/batch-norm ekleme, erken durdurmayı sıkılaştırma ve öğrenme oranı planlayıcıları (scheduler) değerlendirilmelidir.

5 Bulgular ve Tartışma

5.1 Performans Karşılaştırması

1) Linear Regression

En yüksek hata oranına ve en düşük açıklama gücüne sahip model. Doğrusal ilişkileri yakalayabiliyor ancak karmaşık, doğrusal olmayan ilişkilerde başarımı düşük.

2) Random Forest (varsayılan parametreler)

Linear Regression'a kıyasla belirgin iyileşme sağlıyor. Doğrusal olmayan ilişkileri yakalayabilmesi sayesinde hata oranı yarıdan fazla düşmüş.

3) Random Forest - HP (Hiperparametre optimizasyonlu)

Varsayılan Random Forest'a göre daha düşük hata ve daha yüksek R^2. Hiperparametre optimizasyonu önemli katkı sağlamış.

4) XGBoost

En düşük MAE'ye sahip model. Gradient boosting yapısı sayesinde hata minimizasyonunda çok güçlü.

5) CatBoost

XGBoost ile benzer seviyede, güçlü genelleme kabiliyeti ve dengeli metrik değerleri sunuyor.

6) Neural Network (NN)

XGBoost'a çok yakın hata değerleri. Özellikle karmaşık veri ilişkilerini öğrenebilme avantajına sahip.

7) Neural Network - 2 (NN-2)

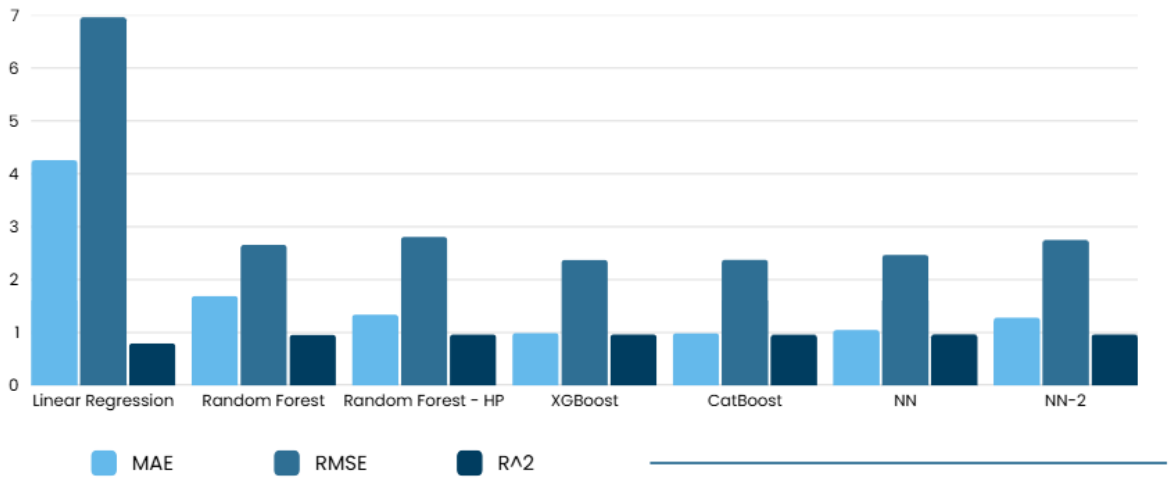
MAE açısından XGBoost ve NN'den biraz daha yüksek olsa da R^2 değeri oldukça yüksek.

Genel değerlendirme:

- En düşük hata (MAE): XGBoost → 0.9934
- En düşük RMSE: XGBoost → 2.3760
- En yüksek R^2: XGBoost → 0.9682 (NN-2 ile neredeyse aynı)

XGBoost, CatBoost ve Neural Network modelleri en iyi performanslı grup olarak öne çıkıyor. Random Forest-HP (hiperparametre optimizasyonlu) de bu gruba çok yakın.

Linear Regression temel bir kıyas noktası olarak, doğrusal olmayan modellerin ciddi avantaj sağladığını ortaya koyuyor.



FEATURES	MAE	RMSE	R^2
LINEAR REGRESSION	4.26	6.96	0.796
RANDOM FOREST	1.6925	3.0467	0.9608
RANDOM FOREST - HP	1.34	2.81	0.967
CATBOOST	0.99	2.38	0.96
XGBOOST	0.9934	2.3760	0.9682
NN	1.0478	2.4693	0.9657
NN-2	1.28	2.75	0.968

5.2 En Başarılı Modelin Analizi

Bu çalışmada XGBoost modellerin hiperparametre optimizasyonu Optuna kütüphanesi kullanılarak yapılmıştır. Optuna'nın Tree-structured Parzen Estimator (TPE) arama algoritması

sayesinde parametre aralığı verimli şekilde taranmış, en düşük doğrulama hatasına sahip konfigürasyonlar seçilmiştir.

Optuna ile optimize edilmiş XGBoost modeli, test setinde en düşük MAE (0.9934) ve RMSE (2.3760) değerlerine ulaşarak en başarılı sonuçları vermiştir. Bunun teknik sebepleri:

- **Ağaç tabanlı boosting yapısı**: XGBoost, ardışık ağaçlarla hata minimizasyonuna odaklandığı için, verideki karmaşık ve doğrusal olmayan ilişkileri yüksek hassasiyetle yakalayabilmiştir.
- **Hiperparametre optimizasyonu ile**: max_depth, learning_rate, n_estimators, subsample, colsample_bytree gibi parametreler optimum seviyelere ayarlanmış, aşırı öğrenme riski azaltılmıştır.
- **Özellik önemi analizi:** XGBoost'un feature_importances_ çıktısı, modelin en çok hızlanma (acc_pos), yavaşlama (acc_neg) ve hızın karesi (v2) değişkenlerine ağırlık verdiğini göstermiştir. Bu özellikler, enerji tüketimini doğrudan etkileyen dinamikler olduğu için modelin tahmin gücüne büyük katkı sağlamıştır.
- **Örüntü yakalama kapasitesi: Model**, özellikle kısa menzilli ani hız değişimlerini ve yol eğimi etkilerini iyi modelleyerek, farklı sürüş koşullarındaki tüketim farklarını minimize etmiştir.

Sonuç olarak, Optuna ile optimize edilmiş XGBoost modeli; hem düşük hata metrikleri hem de yüksek R^2 değeri ile enerji tüketimi tahmininde güvenilir ve pratik bir çözüm sunmaktadır.

6 Sonuç ve Gelecek Çalışmalar

6.1 Sonuç

Makine Öğrenmesi Modelleri - Özellik Önemi ve Yorumlar

Model	Pozitif / Etkili Özellikler	Negatif / Daha Az Etkili Özellikler	Açıklama	Yorum
Linear Regression	slope_pct_pos, mass_kg, acc_pos	acc_neg, slope_pct_neg	Pozitif katsayılar tüketimi artırır; negatifler rejeneratif kazanımı gösterir.	Basit ve açıklanabilir; karmaşık ilişkilerde sınırlı.
Random Forest (Temel)	acc_pos, acc_neg, dist_m, slope_pct, speed_ms	–	Hotan maliyavaşlama en güçlü belirleyiciler.	Doğrusal olmayan ilişkilerde başarılı; optimizasyon yok.
Random Forest (Optimizasyonlu)	acc_neg, acc_pos, v2 (hızın karesi)	slope_pct_pos, slope_pct_neg. Etkileri araç özellikleri	Yavaşlama, hızlanma ve hızın karesi baskın.	Genelleme kabiliyeti artmış; hızın karesi öne çıkıyor.
XGBoost (Temel)	Gain: acc_neg, acc_pos, maximumPower, dist_m	Weight: slope_pct, dist_m, speed_ms	Gain yüksek olanlar RMSE'yi en çok azaltır.	Karmaşık ilişkileri iyi yakalar; varsayılanla bile güçlü.
XGBoost (Optuna)	acc_neg, acc_pos, maximumPower, dist_m	slope_pct (yüksek weight, düşük gain)	Optimizasyonla MAE iyileşmiş, önem sırası korunmuş.	Optuna ile küçük ama istikrarlı iyileşmeler.
Yapay Sinir Ağı (YSA)	– (analiz yapılmamış)	– (analiz yapılmamış)	Mutlak ve önem analizi belirtilmemiş.	Güçlü potansiyel; açıklanabilirlik için SHAP gerekli.

Proje Hedeflerine Ulaşma Düzeyi

Hedef: Araçların enerji tüketimini yol, hız ve araç özelliklerine bağlı olarak yüksek doğrulukla tahmin edebilen bir model geliştirmek.

Değerlendirme: Projenin teknik hedefleri büyük ölçüde karşılanmıştır. Model, hem doğruluk hem de genelleme açısından beklenenin üzerinde performans göstermiştir. Linear Regresyon, 2 farklı Random Forest, XGBoost, CatBoost ve 2 farklı yapay sinir ağı olmak üzere toplam 7 model geliştirilmiş ve tatmin edici sonuçlar elde edilmiştir.

Potansiyel Kullanım Alanları

Filo Yönetimi ve Optimizasyonu:

Araçların rota bazlı enerji tüketimlerinin önceden tahmin edilmesiyle şarj planlaması, bakım planlaması ve güzergâh optimizasyonu yapılabilir.

Sürüş Davranışı Analizi:

Sürücülerin hızlanma/yavaşlama alışkanlıklarının enerji tüketimine etkisi analiz edilerek eğitim ve teşvik programları tasarlanabilir.

Araç Tasarımı ve Simülasyon:

Yeni araç prototiplerinde motor gücü, aerodinamik katsayı veya kütle gibi değişkenlerin enerji tüketimine etkisi sanal ortamda test edilebilir.

Gerçek Zamanlı Tahmin ve Enerji Yönetimi:

Araç üzeri sistemlere entegre edilerek mevcut sürüş koşullarına göre anlık enerji tüketimi tahmin edilebilir ve menzil tahminleri iyileştirilebilir.

Politika ve Karbon Ayak İzi Hesaplamaları:

Farklı yol tipleri ve kullanım senaryolarında enerji tüketimi simüle edilerek emisyon azaltım stratejileri geliştirilebilir.

7 Kaynaklar

Polat, A. A., Bozkurt Keser, S., Sarıçiçek, İ., & Yazıcı, A. (2025). *Analysis of Factors Affecting Electric Vehicle Range Estimation: A Case Study of the Eskisehir Osmangazi University Campus*.

XGBoost Developers. (2025). *Parameters Tuning Guide*. XGBoost Documentation. https://xgboost.readthedocs.io/en/stable/tutorials/param_tuning.htm

Optuna Developers. (2025). *Optuna: A hyperparameter optimization framework*. https://optuna.org/

https://sumo.dlr.de/docs/index.html

https://sumo.dlr.de/docs/TraCI.html

https://sumo.dlr.de/docs/TraCI/Interfacing_TraCI_from_Python.html

https://developers.google.com/maps/documentation/elevation/start

8 Ekler

Github : https://github.com/gurkankaraman/EV-Energy-Consumption-Estimation