

Makine Öğrenmesi Algoritmaları ile İkinci El Araç Fiyat Tahmini

Used Car Price Prediction with Machine Learning Algorithms

Samed ZIRHLIOĞLU
Bilgisayar Mühendisliği
18110131037

Kahramanmaraş Sütçü İmam Üniversitesi
Kahramanmaraş, Türkiye
zirhlioglusamed@gmail.com

Mehtap ÖKLÜ
Bilgisayar Mühendisliği
17110131052

Kahramanmaraş Sütçü İmam Üniversitesi
Kahramanmaraş, Türkiye
mehtap_oklu_06@hotmail.com

Özetçe —Teknolojinin çok hızlı ilerlemesine bağlı olarak otomotiv sektörü de hızlı bir şekilde gelişmektedir. Bundan kaynaklı olarak artan üretim çeşitliliği, ikinci el piyasasında da büyük etki bırakmaktadır. İkinci el piyasası aktörlerinin araç alım ve satışlarında piyasanın durumunu iyi anlayarak buna göre araç ve fiyat tespitlerini yapmaları çok önem arz etmektedir. Otomotiv sektörü yolcu ve yük taşıma amaçlarına bağlı olarak birçok ana ve alt başlıkta üretim yapmaktadır. Yapılacak çalışmamızda kişisel kullanıma yönelik BMW markası gibi özgün bir otomobil markası üzerinden modelleme yapılacaktır. BMW firması her yıl farklı teknik, donanım ve tasarım özelliklerde yeni model araçları üretmektedir. Bu markaya ait araçların özellikleri (model, yıl, motor, şanzıman, Km./Mil, yakıt tipi, vergi, yakıt tüketimi, fiyat vb.) incelenerek aracın değeri tahmin edilebilir. Yapay zekâ algoritmaları (Karar Ağacı, Gradient Boost, XGB, Rastgele Orman Ağacı, LightGBM, CatBoost) kullanılarak bu değerler incelenmiştir. Projedeki sorunun tipinden dolayı regresyon algoritmalarına yer verilmiştir. Bunun nedeni ise elimizdeki bağımsız değişkenlerden hareketle, yeni bir bağımlı değişken elde etmek istememizdir.

Anahtar Kelimeler—Araç, Araç Özellikleri, Karar Ağacı, Gradient Boost, XGB, Rastgele Orman Ağacı, LightGBM, CatBoost

Abstract—Depending on the rapid progress of technology, the automotive industry is also developing rapidly. Due to this, the increasing variety of production also has a great impact on the second-hand market. It is very important for the second-hand market actors to understand the situation of the market in vehicle purchases and sales and to determine the vehicle and price accordingly. The automotive sector produces in many main and sub-titles depending on the purposes of passenger and cargo transportation. In our study, modeling will be done on a unique automobile brand such as BMW brand for personal use. Every year, BMW company produces new model vehicles with different technical, equipment and design features. The value of the vehicle can be estimated by examining the characteristics of the vehicles of this brand (model, year, engine, transmission, Km./Mile, fuel type, tax, fuel consumption, price, etc.). These

values were examined using artificial intelligence algorithms (Decision Tree, Gradient Boost, XGB, Random Forest Tree, LightGBM, CatBoost). Regression algorithms are included due to the type of problem in the project. The reason for this is that we want to obtain a new dependent variable based on the independent variables we have.

Keywords—Car, Car Properties, Value, Decision Tree, Gradient Boost, XGB, Random Forest Tree, LightGBM, CatBoost

I. GİRİŞ

Ulaşım, insanlık tarihinin başlangıcından günümüze kadar, insan hayatının en temel ihtiyaçlarından birisi olmuştur. Arabalar icat edilmeden önce ulaşım; at, eşek ve deve gibi hayvanlar kullanılarak sağlanmıştır. Bu durum hem zaman hem de emek kaybına neden olmuştur.

Artan dünya nüfusu ve ihtiyaçlar doğrultusunda ulaşım çok büyük bir önem kazanmıştır. Bu ihtiyaçları gidermek ve ulaşımı daha kolay hale getirmek için yapılan çalışmalar hız kazanmıştır. Bu çalışmalar sonucunda insanlar ihtiyaçları doğrultusunda tarım ve ulaşım araçlarını geliştirmiştir. Günümüzde de insanların daha konforlu ve keyifli bir ulaşım deneyimi yaşayabilmesi için bu çalışmalar devam etmektedir.

Günümüzde otomobil sektörü, insanlık tarihinin en yaygın sektörlerinden ve yatırım yapılan iş kollarından birisi olmuştur. Başlangıçta sadece ulaşım ve ticaret amaçlı kullanılan otomobiller, günümüzde lüks ve ihtişamlı yaşamın sembolü haline gelmiştir [1].

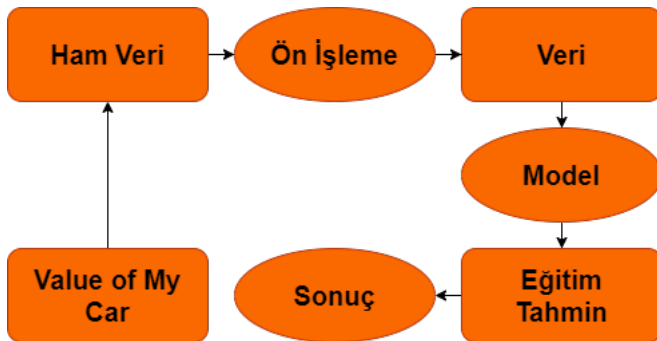
Arabalar uzun ömürlü araçlar oldukları için sürekli alınıp satılabilir. İkinci el olarak satılan araçların makul bir şekilde fiyatlandırılması gerekmektedir. İkinci el bir aracın makul fiyat tahmininin yapılması zor olduğu için doğru fiyatı belirleyecek bir tahmin mekanizmasına ihtiyaç duyulmaktadır [2]. Bu makalede yapay zekâ kullanarak ikinci el araçların fiyatlarını tahmin etmeye çalışılacaktır.

Bu proje için, "100.000 UK Used Car Dataset" isimli veri seti [3] Kaggle'dan temin edilmiştir. Elde edilen veri seti içerisindeki BMW marka araçların verileri alınarak gerekli ön işleme adımları yapılmıştır. Veri setinde 8 adet öznitelik (model, yıl, şanzıman, satış fiyatı, mil, yıllık vergisi, mesafe başına yakıt miktarı ve motor hacmi) bulunmaktadır. Düzenlediğimiz veriler %30 test, %70 eğitim verisi olarak ayrılmıştır.

Kullandığımız özniteliklerin özellikleri;

- **Model:** Markanın sınıflara ayırdığı farklı araç tiplerine verdiği isim
- **Yıl:** Aracın üretilip satışa sunulduğu yıl
- **Şanzıman:** Motordan istenilen hareketi şaft veya diferansiyele aktarma organıdır.
- **Satış fiyatı:** Aracın son durumundaki satış fiyatı
- **Mil:** Aracın gittiği toplam yol
- **Yıllık vergi:** Devletin araç kullananlardan düzenli olarak aldığı ücret
- **Mesafe başına yakıt:** Mil başına tüketilen galon miktarı
- **Motor hacmi:** Aracın motor büyüklüğü

Bu veri seti kullanılarak ve araçların öznitelikleri göz önünde bulundurularak aracın fiyat tahmini yapılmıştır. Bu fiyat tahmininin daha doğru olması için; Karar Ağacı, Gradient Boost, XGB, Rastgele Orman Ağacı, LightGBM, CatBoost algoritmaları kullanılmıştır. Proje; Python dilinde, Visual Studio Code editöründe yazılmıştır.



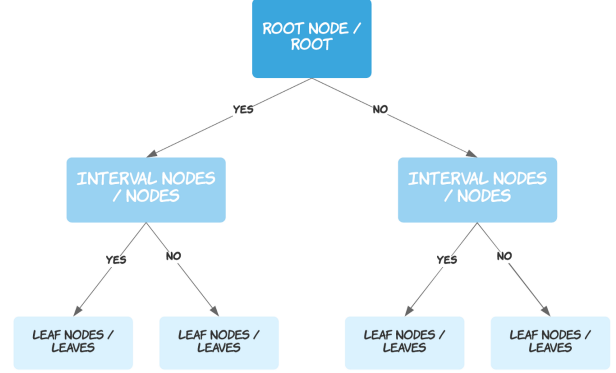
Şekil 1: Value of My Car Akış Diyagramı

Projenin akış diyagramı Şekil 1'de belirtilmiştir. Makalenin devamında kullanılan algoritmalar ile ilgili bilgiler ve sonuç kısmı bulunmaktadır.

II. METODLAR

A. Decision Tree

Karar Ağaçları, genellikle regresyon ve sınıflandırma problemlerinde kullanılmaktadır. Ağaç tabanlı algoritmalarından birisi olup karmaşık veri setlerinde kullanılabilir [4].



Şekil 2: Decision Tree

Kök (root), Karar Ağaçlarının ilk basamağına(hücreesine) verilen isimdir. Problemimizi gözlemlerken, kök kısmındaki koşullar doğrultusunda sınıflandırma yapılır.

Kök hücrelerin altında düğümler (nodes) yer almaktadır. Ele aldığımız her bir gözlem kökler aracılığı ile sınıflandırılır. Düğüm sayısının artma oranıyla doğru orantılı bir şekilde, elimizde bulunan modelin karmaşıklığı da artmaktadır. Bize sonucu veren yapraklardır ve yapraklar ağacın en altında bulunmaktadır [4].

Kök hücre seçilirken dikkat edilmesi gereken en önemli etken, veri setini iyi açıklayabilmesidir. Köke karar verirken bizim için önemli bazı değerler vardır [4];

Alt kümenin saflık değerine *Gini* denir ve T, p_j, j sınıfının gerçekleşme olasılığıdır. Elimizdeki her sınıf için ayrı ayrı hesaplanarak çıkan sonuçların karesi alınır. Alınan kareler 1'den çıkartılır. Bu işlemler sonucunda $(0, 1)$ aralığında bir değer elde edilir. Elde edilen değer 0'a ne kadar yakınsa o kadar iyi ayırım yapılmıştır [4].

Gini indexinin formülü Eşitlik 1'de verilmiştir [15]. Formül değişkenlerinin temsil ettiği veriler de aşağıdaki gibidir;

- T : Tüm veri seti
- p_j : Veri setindeki her bir verinin kendinden küçük ve kendinden büyük eleman sayılarına bölümü
- n : Seçilen verimiz

$$Gini(T) = 1 - \sum_{j=1}^n (p_j)^2 \quad (1)$$

Entropy rastgeleliği, belirsizliği ve beklenmeyen durumun ortaya çıkma olasılığını gösterir [5]. Bu olasılığı \log_2 tabanında yazar [4].

$$I_H = - \sum_{j=1}^c p_j \log_2(p_j) \quad (2)$$

Entropi daha dengeli bir ağaç çıkarmaya meyilli iken *Gini*, frekansı fazla olan sınıfı ayırtırmaya meyillidir.

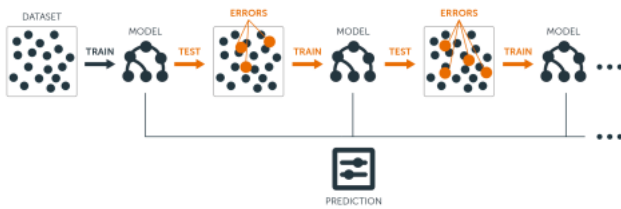
B. Gradient Boosting Regressor

Makine öğrenmesi modellerinde tahminlerin doğruluğunu güçlü bir hale getirmek için Gradient Boosting Regressor kullanılır. Yinelemeli olarak çalışan algoritma, karar ağacı tabanlıdır. Gradient Boosting Regressor algoritmasını güçlü kılan faktör; ağaç yapısının, hatayı en az seviyeye düşürecek şekilde olmasıdır [6].

Gradient Boosting Regressor algoritmasındaki temel amaç, elde edilecek olan maliyet fonksiyonunu en aza düşürebilmek için bulunan parametreleri tekrarlamaktır. Ağaç ekleme işlemi yapılırken kaybı en aza düşürebilmek için gradyan iniş prosedürü kullanılmaktadır [6].

Gradyan yükseltme üç unsur içerir:

- Optimize edilmesi beklenen kayıp fonksiyonun(loss) belirlenmesi.
- Optimize edilmiş veride belirlenen zayıf öğrenici ile tahmin yapmak.
- Loss'u minimum seviyeye indirmek için zayıf öğrenicilere bir katkı modeli eklenir [7].



Şekil 3: Gradyan yükseltme algoritması işleyiş süreci [16]

$$Loss^i = \sum_{j=1}^n (Y_j - F^i(X_j))^2 \quad (3)$$

C. XGB (Extreme Gradient Boosting) Regressor

XGBoost, problemleri diğer boosting algoritmalarına göre daha doğru bir yöntemle çözen bir ağaç güçlendirme algoritması olup paralel olarak çalışmaktadır [9]. Tahmin gücünü yükseltmek ve aşırı öğrenmeyi engellemek, algoritmanın temel özelliklerindendir [10].

XGB'yi tercih etmemizin sebebi, kütüphanesinin; model performansının yüksek ve çalışma süresinin düşük olmasıdır [9].

XGBoost algoritması kullanılırken öncelikle ilk tahmin (base score) yapılmalıdır. Sonraki adımlarda sonuç herhangi bir sayı olabilir. Çünkü yapılacak işlemler yakınsayarak doğru sonuca ulaşacaktır. Sonraki adımda hataları tahminleyen karar ağacı kurulur. Bu adımdaki amaç hataları öğrenip doğru tahmine yaklaşmaktır. Benzerlik skoru (similarity score) her ağacın her bir dalı için ayrı ayrı hesaplanır. Benzerlik skoru, elimizde bulunan verilerin dallarda ne kadar iyi kategorize olduğunu gösterir [10].

$$SimilarityScore = \frac{Sum of Residuals, Squared}{Number of Residuals + \lambda} \quad (4)$$

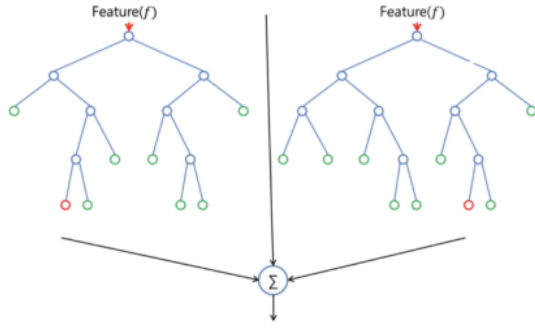
Benzerlik skorlarını hesapladıktan sonra elde ettiğimiz sonuçtan daha iyi bir tahmin yapıp yapılmayacağını öğrenmek için tüm olasılıklardaki ağaçlar kurulur. Hepsi için ayrı ayrı benzerlik skorları hesaplanır ve hangi ağacın daha etkili yani daha iyi olduğunu tespit etmek için Eşitlik 5'deki kazanç hesabı yapılır [10]. Yukarıda belirttiğimiz benzerlik skorunun sonucu ile dallar değerlendirilirken, elde ettiğimiz Eşitlik 5'deki kazanç değeri ile ağacın tamamı değerlendirilmektedir [10].

$$Kazanc = SolBenzerlikSkoru + SagBenzerlikSkoru - OncekiAgacinBenzerlikSkoru \quad (5)$$

D. Random Forest

Rastgele Orman Algoritması, denetimli öğrenme algoritması olup hem sınıflandırmada hem de regresyon analizlerinde kullanılır. Aşırı uyumu önler. Random Forest, karar ağaçlarını rastgele bir şekilde oluşturur. İstikrarlı ve doğru bir sonuç tahmini yapabilmek için bunları birleştirir. Rastgele Orman Algoritması'nda; ağaç sayısı ve çıktıları arasında doğrusal bir ilişki vardır. Ağaç sayısı ne kadar arttırılırsa o kadar kesin bir sonuç elde edilir [11] [12].

Rastgele Orman Algoritması, ağaç büyütme aşamasında, normal durumun üzerine rastgelelik durumunu da kullanmaktadır [12].

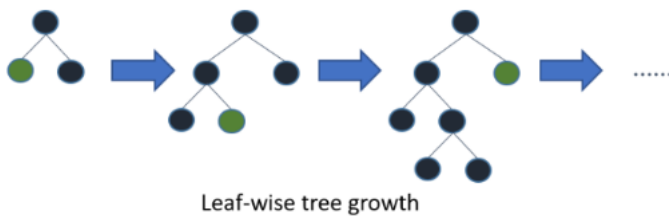


Şekil 4: Random Forest [12]

E. LightGBM Regressor

LightGBM, histogram tabanı kullanan bir algoritmadır. Sürekli verileri giriş olarak alır, bu verileri kesikli hale getirir. Bunu yapmasındaki amaç maliyeti azaltmaktır. Karar Ağacı Algoritması'nın bölünmesi ile doğru orantılıdır. Veriyi optimize etmesi, eğitim süresini düşürür ve daha az kaynak kullanılmasını sağlar [13].

Karar Ağacı Algoritması'nda yaprak ve seviye olarak iki farklı strateji kullanılabilir. Yaprak stratejisinde kayıp oranını azaltan yapraklar üzerinden bölünme işlemi devam eder. Seviye stratejisinde hedeflenen durum ise ağaç büyütülürken dengesini korumaktır. Yaprak stratejisinde hatayı minimize etmek amaçlandığı için, seviye stratejisine göre daha düşük hata oranına sahiptir. Bunun etkilediği durumlardan biri de öğrenme aşamasının hızlanmasıdır. Bu yüzden, boosting algoritmalarının geri kalanından ayrılır. Elimizde bulunan veri sayısı ne kadar çoksa, yaprak stratejisini kullanmak o kadar mantıklı olur. Aşırı öğrenmeye sebebiyet vermemek için veri sayısının az olduğu durumlarda kullanıma uygun değildir. Bu durumda da seviye stratejisi kullanılır [13].



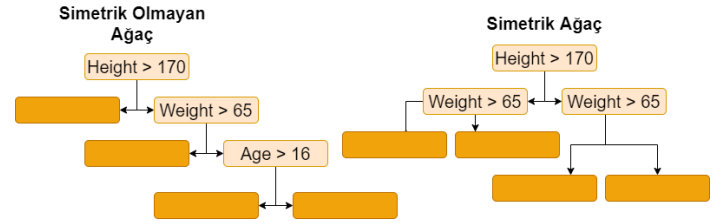
Şekil 5: Yaprak odaklı(leaf wise) strateji [13]



Şekil 6: Seviye odaklı(level wise) strateji [13]

F. CatBoost (Category Boosting) Regressor

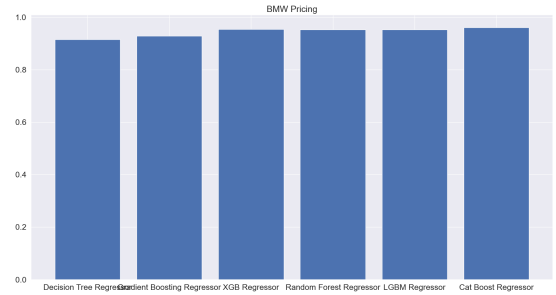
Gradient Boosting tabanında çalışan, açık kaynaklı bir algoritmadır. Veri hazırlığı süresini düşürdüğü için, boosting algoritmaları arasında önemli bir yere sahiptir. Elinde bulunan boş veriler ile baş edebilir ancak kategorik veriler için kodlamaya ihtiyaç duyar. Yüksek öğrenme hızı, sayısal, kategorik ve metin verileri ile çalışabilmesi en belirgin özelliklerindendir. Ayrıca aşırı öğrenme sorununu ortadan kaldırmak için simetrik ağaç kurar. Aşırı öğrenme durumunun oluşması durumunda, belirlenen özelliklere varmadan önce öğrenme işlemi durdurulur [14].



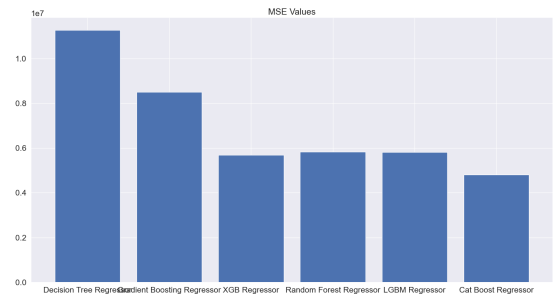
Şekil 7: Simetrik/Simetrik olmayan ağaç

III. SONUÇLAR

Veri setindeki veriler %70 eğitim, %30 ise test verisi olarak ayrılmıştır. Daha sonra ayrılan veriler, algoritmaların eğitim ve tahmin aşamalarında kullanılmıştır. Bu aşamalar sonucunda elde edilen doğruluk değerlerine Şekil 8'de, MSE (ortalama kare hatası) değerlerine de Şekil 9'da yer verilmiştir.



Şekil 8: Tüm Algoritmaların Doğruluk (r^2) Değerleri



Şekil 9: Tüm Algoritmaların MSE Değerleri

İkinci el araç fiyatlarının tahmin edilebilmesi için "100.000 UK Used Car" [3] veri setindeki "BMW.csv" verisi alındı. DT, GB, XGB, RF, LGBM ve CB regressor algoritmaları kullanılmıştır. Bu algoritmaların r^2 (doğruluk değeri) ve MSE (hata değeri) değerleri Tablo I'de yer almaktadır.

	r^2	MSE
DT	0.9209	0.0825
GB	0.9296	0.0657
XGB	0.9538	0.0447
RF	0.9533	0.0453
LGBM	0.9533	0.0449
CB	0.9614	0.0372

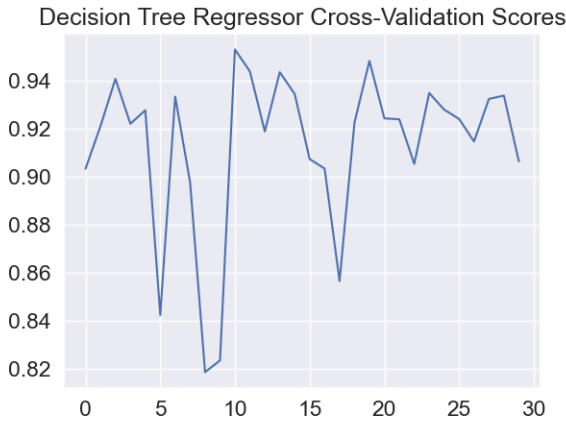
Tablo I: DT, GB, XGB, RF, LGBM ve CB regressor algoritmalarının r^2 ve MSE degerleri

Tablo I'i incelersek, en yüksek doğruluk değeri(r^2)nin %96.14 oranıyla CatBoost algoritmasına ait olduğunu görebiliriz. Aynı zamanda en düşük MSE (hata) değeri de 0.0372 oranıyla yine CatBoost algoritmasına aittir. Bu verilerden en düşük doğruluk değerine sahip olan algoritma ise Decision Tree(%92.09) algoritmasıdır. r^2 ve MSE değerlerinin arasındaki ters orantıdan dolayı, en yüksek MSE değeri yine Decision Tree(0.0825) algoritmasına aittir. Diğer algoritmaların r^2 ve MSE değerleri de Tablo I üzerinden incelenebilir.

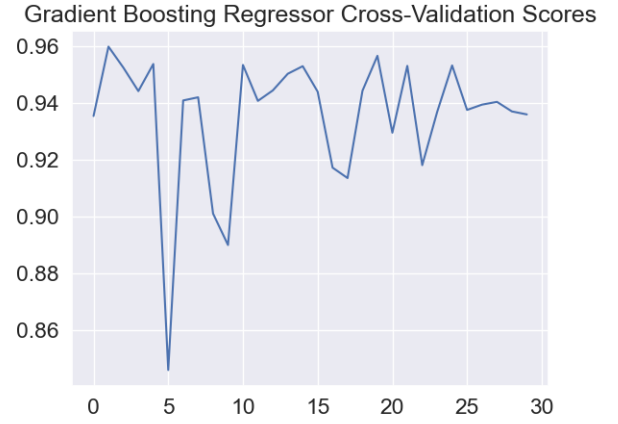
A. Cross-Validation (Çapraz Doğrulama)

Cross-Validation, makine öğrenmesi algoritmalarının tâbi tutulabileceği bir fonksiyondur. Modelin eğitiminde kullanılacak olan veriyi, her seferinde farklı kısımlardan bölerek (*train – test*) optimum sonucu bulmayı amaçlar.

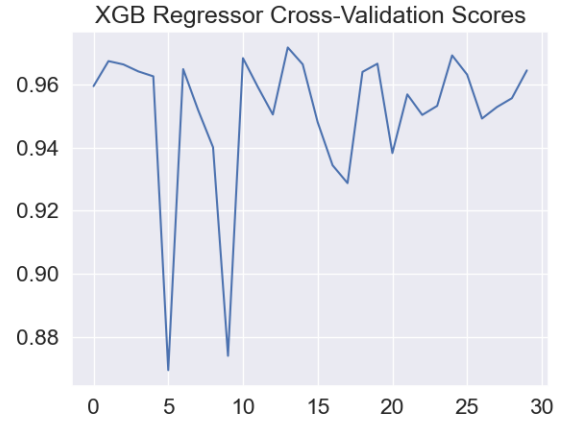
Regresyon algoritmaları birer kez eğitilmiştir. Bu eğitimlerin sonuçları da Tablo I'de belirtilmiştir. DT, GB, XGB, RF, LGBM ve CB algoritmaları **Cross-Validation** metoduyla 30 kez rastgele olacak şekilde çalıştırılmış ve sonuçları Şekil 10, 11, 12, 13, 14 ve 15'de verilmiştir.



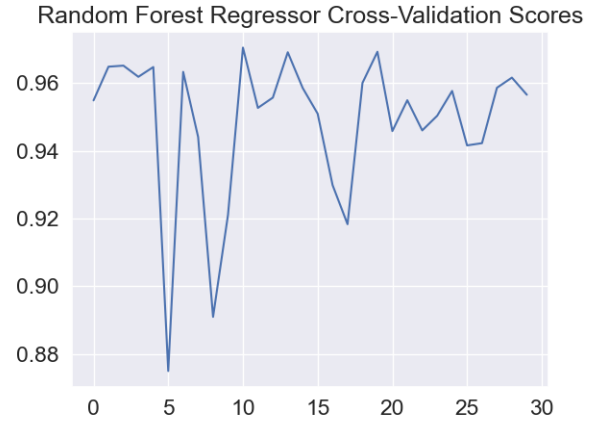
Şekil 10: Decision Tree Regressor Cross-Validation



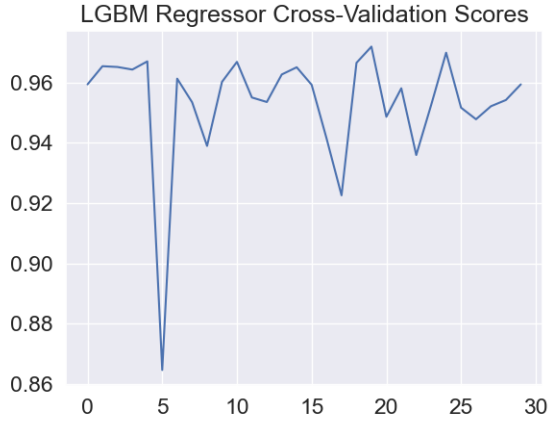
Şekil 11: Gradient Boost Regressor Cross-Validation



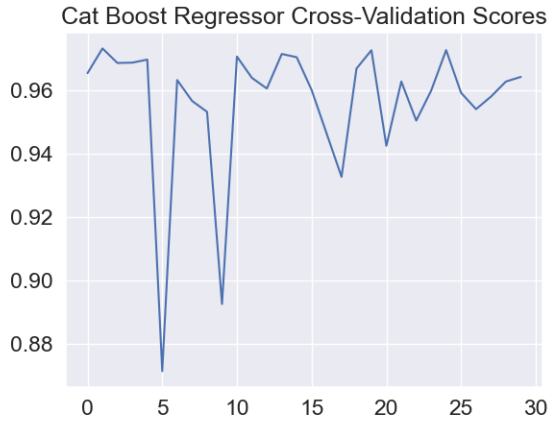
Şekil 12: XGB Regressor Cross-Validation



Şekil 13: Random Forest Regressor Cross-Validation

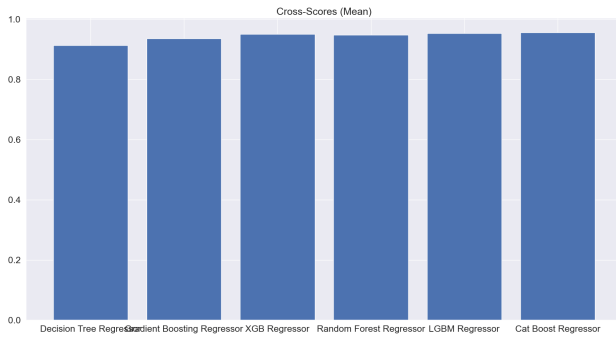


Şekil 14: LGBM Regressor Cross-Validation



Şekil 15: CatBoost Regressor Cross-Validation

Bu algoritmaların Cross-Validation sonuçlarının ortalamaları Şekil 16'daki gibidir. Bu değerler aynı zamanda Tablo II'de de sayısal olarak yer almaktadır.



Şekil 16: Cross-Validation Ortalama(Mean) Değerleri

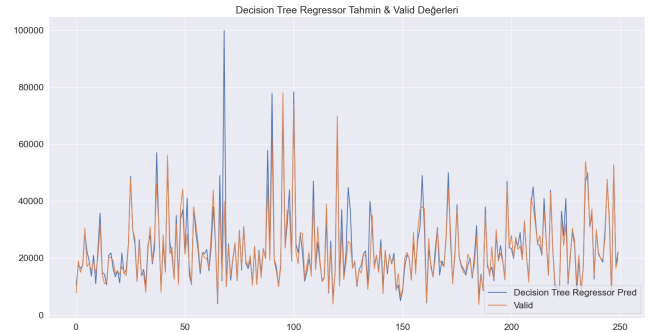
	Cross-Validation Score(mean)
DT	0.9160
GB	0.9356
XGB	0.9512
RF	0.9485
LGBM	0.9531
CB	0.9562

Tablo II: DT, GB, XGB, RF, LGBM ve CB algoritmalarının Cross-Validation sonuçlarının ortalama(mean) değerleri

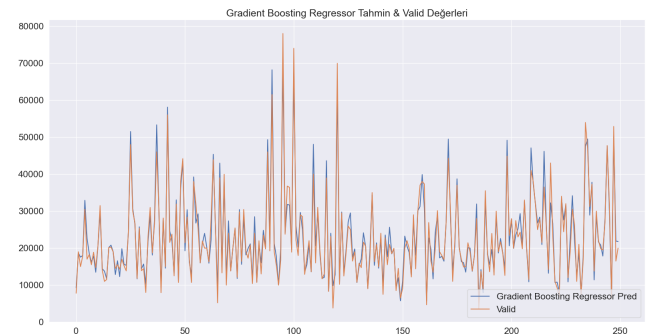
Tablo II'yi incelersek, en yüksek doğruluk değerinin %95.62 oranıyla CatBoost algoritmasına ait olduğu görülebilir. Bu verilerden en düşük doğruluk değerine sahip olan algoritma ise Decision Tree(%91.60) algoritmasıdır. Diğer algoritmaların Cross-Validation sonuçlarının ortalama değerlerini de Tablo II üzerinde incelenebilir.

B. Valid Data-Pred Data(Diff)

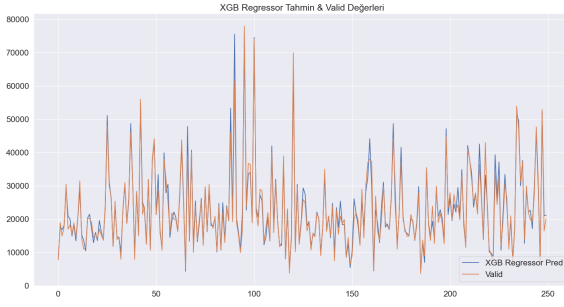
Şekil 17, 18, 19, 20, 21 ve 22'de de gösterildiği gibi; gerçek (*valid*) verilerimiz turuncu, algoritmanın tahmin (*predict*) değerleri ise mavi hatlarla belirtilmiştir. Bu grafikleri oluşturmaktaki amaç; tahmin verisinin gerçek veriden ne kadar saptığını(uzaklaştığını/saptığını) görmektir.



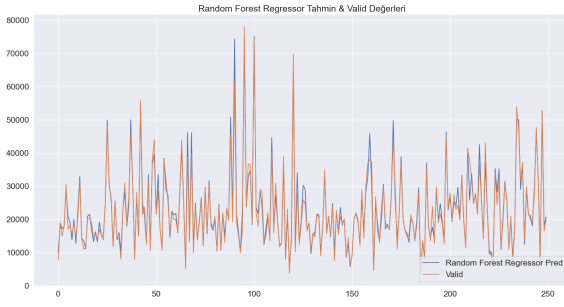
Şekil 17: Decision Tree Regressor Tahmin & Valid Değerleri



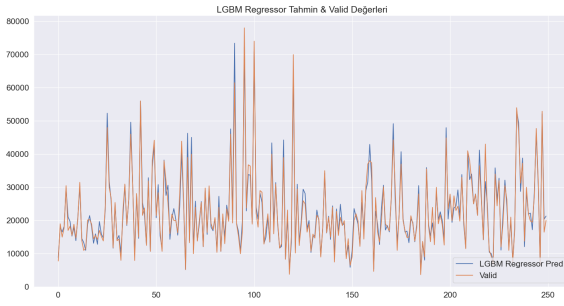
Şekil 18: Gradient Boosting Regressor Tahmin & Valid Değerleri



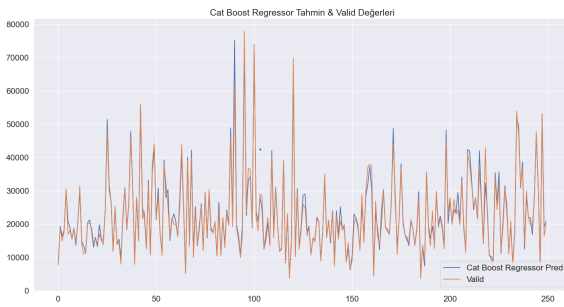
Şekil 19: XGB Regressor Tahmin & Valid Değerleri



Şekil 20: Random Forest Regressor Tahmin & Valid Değerleri



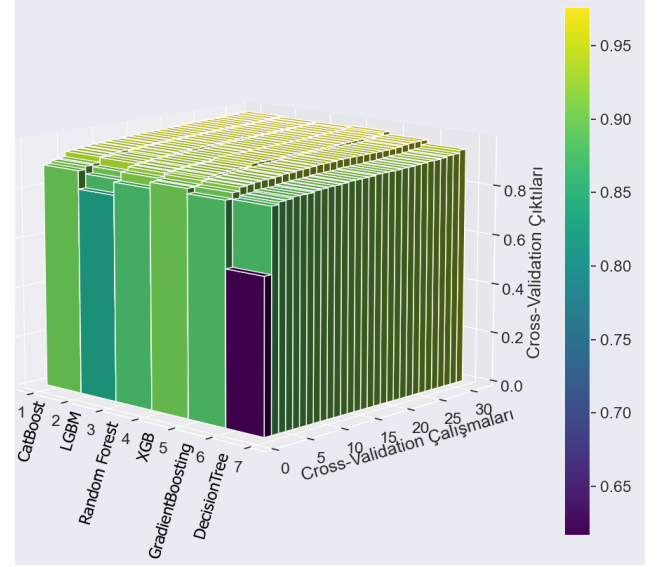
Şekil 21: LGBM Regressor Tahmin & Valid Değerleri



Şekil 22: CatBoost Regressor Tahmin & Valid Değerleri

IV. SONUÇ

Projede kullanılan veri seti sayesinde ikinci el araçların, değerinin altında veya üzerinde bir fiyata satılması yerine, olması gereken bir fiyatta satılmasını sağlamak için; yapay zekâ algoritmaları kullanılarak fiyat tahmini yapılmaya çalışıldı. Algoritmaları, makalenin önceki başlıklarında da anlatıldığı gibi çeşitli işlemlere tâbi tutularak, bazı sonuçlar elde edildi. Çıkan sonuçlar karşılaştırıldı ve en iyi algoritma bulunmaya çalışıldı. Eldeki veri setiyle en uyumlu çalışan algoritma bulundu ve raporlandı. Eğitim sonuçlarını daha yüksek oranlara taşıyabilmek için veri seti optimize edilebilir, gerekli öznetelikler eklenebilir veya gereksiz olanlar çıkartılabilir. Bu sayede daha doğru bir tahminleme işlemi yapılabilir.



Şekil 23: Algoritmaların Cross-Validation Sonuçları (3D Grafik)

X eksenine bakıldığı zaman algoritmaları, Y eksenine bakıldığı zaman algoritmaların çalışma sayıları görülebilir. Daha sonradan 3. boyuta ise bu çalışmaların skorları Z eksenini olarak eklendi ve Şekil 23'deki 3D grafik elde edildi. Algoritmalar ve çalışma skorları, kendi arasında sıralı olduğu için, sütunlara çapraz açıdan bakıldığında en verimli algoritmanın en yüksekte kaldığı görülebilir.

Aynı veri setini kullanan başka bir kullanıcının verilerinin [17] çıktıları da Tablo III'te yer almaktadır. Bu veriler incelendiği zaman bazı noktalarda daha yüksek, bazı noktalarda daha düşük doğruluk (r^2) değeri elde edildiği görülebilir. Ancak MSE değeri incelendiği zaman, projemizin daha düşük hata değerine sahip olduğu görülebilir. Bunun sebebi, veri setinin $zscore$ işlemine tâbi tutulmuş olmasıdır.

	Mevcut Proje		rajan1002	
	R2	MSE	R2	MSE
Random Forest	0.9533	0.0453	0.9508	0.1180
Gradient Boosting	0.9296	0.0657	0.9479	0.1214
XGB Regressor	0.9538	0.0447	0.9562	0.1113

Tablo III: **rajan1002**'nin eğitim sonuçları [17] ile karşılaştırma

KAYNAKÇA

- [1] DergiPark - İkinci El Otomobil Talep Fiyatının Regresyon Analizi
dergipark.org.tr/article-file/431769
- [2] IjcaOnline - Vehicle Price Prediction
ijcaonline.org/archives/noor-2017-ijca-914373
- [3] Kaggle - 100.000 UK Used Car Data Set
kaggle.com/adityadesai13/used-car-dataset-bmw
- [4] Medium - Karar Ağaçları
medium.com/deep-learning/karar-agac-lari
- [5] Başkent - Karar Ağacı
baskent.edu.tr/20410964/DM_8.pdf
- [6] Data Science - Boosting Algoritmaları
datascienceearth.com/boosting-algoritmaları
- [7] Tevfik Bulut - Gradyan Yükseltme Algoritması
tevfikbulut.com/prediction-of-breast-cancer
- [8] OpenGenus - Boosting Algorithms
opengenus.org/types-of-boosting-algorithms
- [9] OpenGenus - XGBoost
opengenus.org/xgboost
- [10] Veri Bilimi Okulu - XGBoost Nasıl Çalışır
veribilimiokulu.com/xgboost-nasil-calisir
- [11] Medium - Rastgele Orman Algoritması
medium.com/@cemthecebi/rastgele-orman
- [12] DevHunter - Rastgele Orman Algoritması
devhunteryz.wordpress.com/random-forest
- [13] Veri Bilimi Okulu - LightGBM
veribilimiokulu.com/lightgbm
- [14] Veri Bilimi Okulu - CatBoost Nedir
veribilimiokulu.com/catboost
- [15] SlideShare - Random Forest
slideshare.net/SezerFidanc/random-forest-algoritmasi
- [16] BradleyBoehmke - GBM
bradleyboehmke.github.io/gbm.html
- [17] Kaggle - Used Car Price Prediction
kaggle.com/rajan1002/used-car-price-prediction