# Zaman Serisi Tabanlı Transformatör Yaklaşımları ile İkinci El Araç Fiyat Tahmini

Eren Şahyılmaz

Kocaeli Üniversitesi

Teknoloji Fakültesi

Bilişim Sistemleri Mühendisliği

201307003

erensahyilmaz@gmail.com

ÖZET

**Bu proje, Türkiye'nin önde gelen ikinci el araç platformu arabam.com'dan elde edilen veriler üzerinden, ileri düzey bir fiyat tahmin sistemi geliştirmeyi amaçlamaktadır. Bu sistem, geleneksel yöntemlerin ötesine geçerek, doğal dil işleme alanında devrim yaratan transformatör modellerini kullanacak. Proje kapsamında, araçların kilometresi, yaşı, markası, modeli, motor özellikleri, yakıt tüketimi gibi birçok faktörü analiz edilecek. Bu kapsamlı analiz, ikinci el araç fiyatlarını etkileyen tüm unsurları değerlendirerek daha gerçekçi ve güvenilir tahminler üretmeyi sağlayacak. Böylece, hem alıcılar hem de satıcılar için daha şeffaf bir pazar ortamı oluşturulmasına katkı sağlanacak.. Ayrıca, proje sonuçları akademik dünyaya da katkı sağlama ve zaman serisi analizi ile transformatör modelleri alanında yeni araştırmalara ilham verme potansiyeline sahiptir.**

***Anahtar kelimeler: makine öğrenmesi, regresyon, veri bilimi***

## 1.KONU

Otomotiv piyasası dünyanın en yaygın sektörlerinden biridir. Piyasaya her yıl yüzlerce yeni otomobil çıkmaktadır ve bu otomobillerin milyonlarca alıcısı olmaktadır. İkinci el piyasası ise otomotiv sektörünün en önde gelen pazarlarından biridir. Ülkemizde 2023 yılında 4 milyon 151 bin 644 ilan yayınlanırken bunların %45'i yani 1 milyon 881 bin 97 tanesi satılmıştır [1]. Dolayısıyla, ilanların yarısından fazlasının satılamadığı gözlenmektedir. Bunun nedenlerinden birinin, yanlış fiyatlandırma olma ihtimali yüksektir. Yanlış fiyatlandırmanın nedenleri ise, araç marka modeli, kilometresi, yaşı gibi durumların yanlış yorumlanarak piyasadaki talep ve şartların dışında olması muhtemeldir. Dolayısıyla, ikinci el araçların fiyatlandırılmasında hangi faktörlerin daha belirleyici olup bunların fiyat tahmini üzerine etkisi önem arz etmektedir.

Bu projede hedeflenen, makine öğrenmesi, derin öğrenme ile çeşitli transformatör modellerini kullanarak ikinci el araç piyasasını daha isabetli şekilde analiz edebilen bir model geliştirmektir. Böylece bu model yardımı ile hem satış istatistiğini artırabilir hem de satıcıların zamandan tasarruf etmesini sağlayabiliriz. Aynı zamanda ikinci el araç fiyat tahmininin araç piyasasında fiyatlardaki dalgalanmaları önleyebilecek potansiyeli de mevcuttur. Çünkü, bazı satıcılar yanlış fiyatlandırma sonucu satışa sunduğu aracını alıcı olmaması durumunda fiyatlandırmada aşağı çekebilmektedir. Önerilen projemiz ile satıcılar araçları hakkında bilgileri, oluşturulacak siteye yazabilir ve ikinci el aracın ortalama satış fiyatı hakkında bilgi sahibi olabilir ve böylece satıcıların ikinci el araç satış fiyatını belirlemek için harcadıkları zamandan tasarruf yapılması sağlanabilecektir.

Daha önce Sydney Üniversitesi'nde araç fiyat tahmini üzerine yapılan bir çalışmada [2], 13,120 adet Mercedes marka aracın üzerinde yapılan çalışmada Yıl,Vergi oranı, Mil kullanımı, MPG ( aracın bir galon yakıtla kaç mil gideceği ) ve Motor Hacmi gibi değişkenlerin kullanıldığı çalışmada Doğrusal, Polinomal, Destek Vektör, Rastgele Orman gibi çeşitli regresyon modelleri kullanılmış ve en yüksek $R^2$ skoru Rastgele Orman Regresyonunda 0.90416 olmuştur.

Rose-Hulman Teknoloji Ensitütü'sünde ikinci el araç fiyat tahmini üzerine yapılan bir çalışmada [3], 19,237 örnek ve 18 adet özellikten oluşan bir veri seti ele alınmıştır. Veriler marka, model gibi kategorik veya mil kullanımı, silindir sayısı ve üretim yılı gibi numerik değişkenlerden de oluşmaktadır. Doğrusal regresyon, Lasso Regresyonu, Ridge Regresyonu ve Rastgele Orman Regresyonunun yanı sıra Hiperparametre ayarlama fonksiyonlarından olan GridSearchCV arama ağacı da kullanılmıştır. En yüksek $R^2$ skoruna sahip model ise 0.799 puan ile Rastgele Orman Regresyonu olmuştur. Yine Chouaib Doukkali Üniversitesi'nde ikinci el araç fiyat tahmini üzerine yapılan bir çalışmada ise [4] Mil kullanımı, Üretim Yılı,Mali Güç,Yakıt Tipi, Marka ve Model gibi değişkenlerle oluşan veri seti kullanılmıştır. Veri seti 8,000 adet örnek içermektedir. Gradyan Artırmalı Regresyon, KNN Regresyonu, Rastgele Orman Regresyonu gibi modeller kullanılmıştır. En yüksek $R^2$ skoru ise 0.80 ile Gradyan Artırmalı Regresyon olmuştur. RMSE ise 44516,20 değerindedir.

Ülkemizde yapılan çalışmalara değinecek olursak ise Gazi Üniversitesi'nde ikinci el araç fiyat tahmini üzerine yapılan bir çalışmada [5] 120,000 veriden ve 11 özellik değerinden oluşan veri seti üzerinde yapılan bir çalışma mevcuttur. Doğrusal, Karar Ağacı, GBT, Rastgele Orman gibi regresyon modelleri kullanılmıştır. En yüksek performans ise 0.887 $R^2$ skoru ile Rastgele Orman Regresyonundan elde edilmiştir. RMSE değeri ise 21435,09 olmuştur. Son olarak Eskişehir Osmangazi Üniversitesi'nde ikinci el araç fiyatları üzerine 2019 yılında yapılan bir çalışmada [6] 5041 arabanın fiyatları 78 adet değişken ile listelenmiştir. Çalışmada yapılan algoritma sonucu $R^2$ skoru 0.891 olarak bulunmuştur. Oluşturulan tahmin modeli ile oluşturulan tahminler ile doğru sonuçlar arasında %10 hata payı ile başarı oranı %81,15 olmuştur.

Literatür taramasından da görüldüğü üzere genellikle makine öğrenmesi yöntemler, ikinci el araç tahmininde kullanılmıştır.

Önerdiğimiz projede, bu yöntemlere ek olarak derin öğrenme ve transformatör yaklaşımlarını kullanarak literatürdeki bu boşluğu doldurmayı hedefliyoruz. Model performanslarının değerlendirilmesi aşamasında, her modelin performansı hata metrikleri ile karşılaştırılacak ve en düşük hata payına sahip model değerlendirilecektir.

## 2.AMAÇ

Bu projede amacımız, ikinci el araç piyasasındaki dinamikleri öğrenip fiyat belirlemek için adil ve güvenilir bir fiyat tahmin modeli ortaya çıkarmaktır. Bu amaçla, ikinci el araç fiyatlarını belirlemek için verilerin yaş, kilometre, marka, model, seri, yakıt tipi, kasa tipi, renk, motor gücü, motor hacmi, ortalama yakıt tüketimi, vites tipi, yakıt deposu, il/ilçe bilgisi vb. özellikleri değerlendirilerek modelleme aşamasında makine öğrenmesi ve derin öğrenme algoritmaları ile transformatör yaklaşımlarına beslenecektir. Hedefimiz, literatür çalışmalarında tespit edilen performansların özellikle transformatör yaklaşımlarının performansları değerlendirilerek üzerine çıkmaktır.
Proje, $R^2$ skoru ve MAPE gibi hata metrikleri ile denetlenecektir. $R^2$, bir regresyon modelinin bağımsız değişkenlerin bağımlı değişken üzerindeki varyansı ne kadar iyi açıkladığını gösteren bir istatistiksel ölçüttür. 0 ile 1 arasında değerler alır. $R^2$ skoru 1 değerine yaklaştıkça model, verideki varyasyonu daha net açıklar. MAPE, bir regresyon modelinin tahminlerinin gerçek değerlerden ortalama olarak ne kadar uzakta olduğunu gösteren bir hata ölçüsüdür. Yüzde olarak ifade edildiğinden, farklı birimlerdeki veriler için karşılaştırılabilir bir ölçüt sunar. Proje çıktısının, $R^2$ skoru olarak 0.90-0.95 aralığına ulaşmasını, MAPE skorunun ise %10'un altında kalmasını hedefliyoruz.

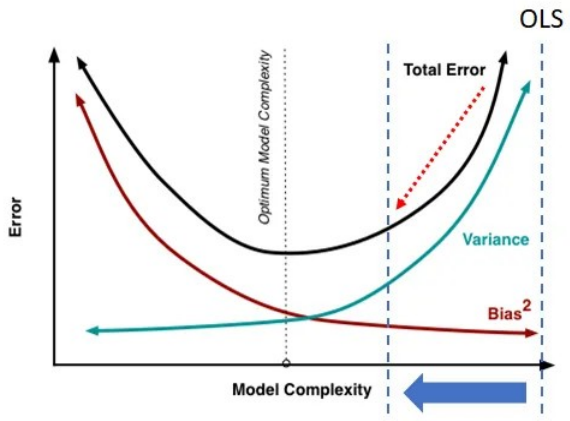
## 3.KULLANILACAK TEKNOLOJİLER

### *3.1 Verilerin Toplanması*

Python programlama dilinin BeautifulSoup kütüphanesi ile Arabam.com üzerinden araç ile ilgili özellikler toplanacaktır.

### *3.2 Modeller*

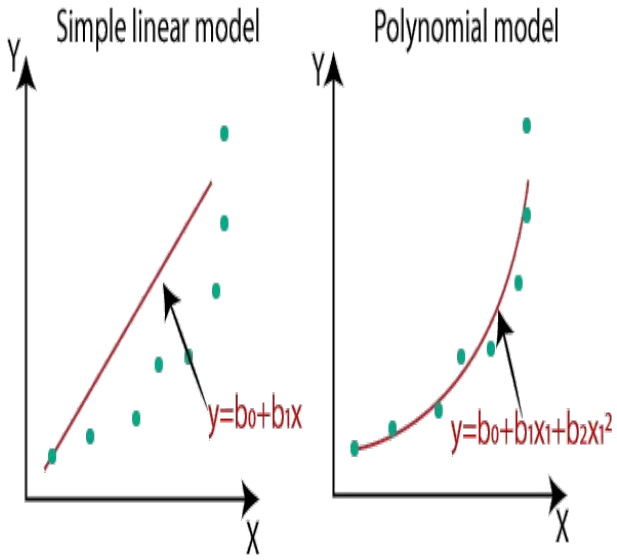
Doğrusal Regresyon: $y=\beta0+\beta1x+\epsilon$ basitçe bu formülden yola çıkılır. y, elde edilmek istenen değerdir ve x'e bağımlıdır. β0: Y-intercept (kesişim noktası), β1: Bağımsız değişkenin katsayısı, x: Bağımsız değişken (tahmin yapmaya çalıştığınız değişken) . x birden fazla olacaktır dolayısıyla her x için bir β katsayı değeri olacaktır. Python'da sklearn kütüphanesinde yer alan LinearRegression() fonksiyonu, doğrusal regresyonu kullanarak bağımsız ve bağımlı değişken arasında bir değer kurar ve bu ilişkiyi model haline getirir.




**Şekil 2:** Ridge Regresyonu [7]

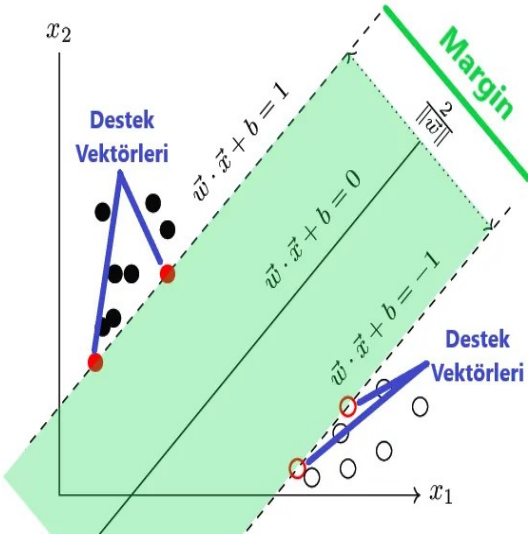
Ridge Regresyonu: Modelin aşırı öğrenmesini engellemek amacıyla L2 ceza terimi ekler. Ceza terimi, modelin ağırlıklarının büyüklüğünü cezalandırır ve çok büyük ağırlıkların oluşmasını engeller.

Lasso Regresyonu: L1 ceza terimi ekleyerek regresyon modelini düzenler. Özellikle, bazı özellikleri sıfırlayarak modelin basitleşmesini sağlar. Özellik sayısının fazla olduğu ve bazı özelliklerin modelde yer alması gerekmediği durumlarda kullanılır. Ayrıca, aşırı öğrenmeyi engellemek için de yaygın olarak tercih edilir.




**Şekil 3:** Polinomal Regresyon [8]

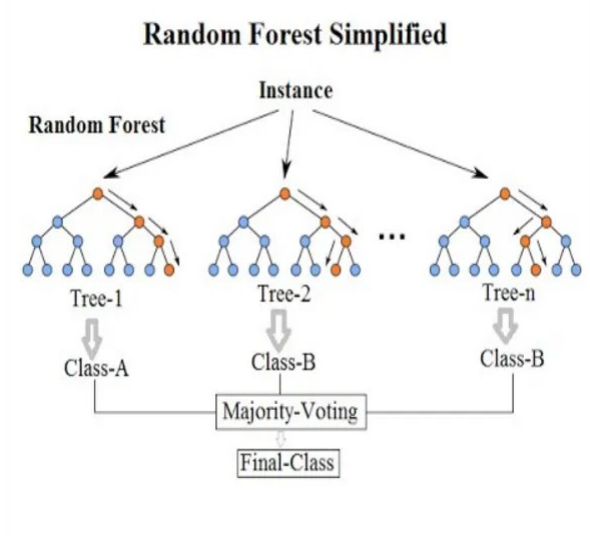
Polinomiyal Regresyon: doğrusal regresyonun bir uzantısıdır ve bağımsız değişkenler ile bağımlı değişken arasında doğrusal olmayan ilişkileri modellemek için kullanılır. Bu model, bağımsız değişkenlerin daha yüksek dereceden polinomlarını kullanarak, doğrusal regresyondan daha esnek bir model oluşturur.




**Şekil 4:** Destek Vektörleri [9]

Destek Vektörleri Regresyonu: Özellikle doğrusal olmayan ilişkileri modellemek için güçlü bir yöntemdir ve veri setindeki gürültüye (outlier'lara) karşı dayanıklı olma özelliğine sahiptir. DVG, genellikle doğrusal olmayan ve karmaşık verilerde, özellikle küçük ve orta büyüklükteki veri setlerinde tercih edilir

Karar Ağaçları Regresyonu: Veriyi bir ağaç yapısına böler ve her dalda bir tahmin yapar.



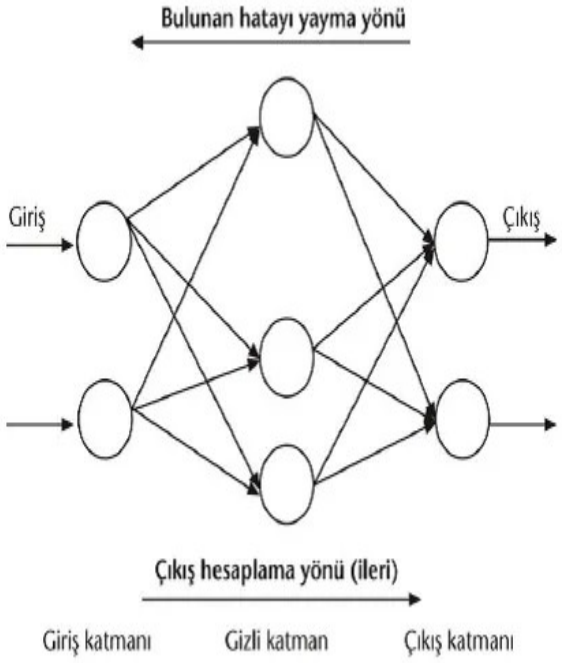

**Şekil 5:** Rastgele Orman Regresyonu [10]

Rastgele Orman Regresyonu: Rastgele Orman, birden çok karar ağacından (decision tree) oluşan bir modeldir ve her ağaç, verinin farklı bir alt kümesi üzerinde eğitilir. Sonuçlar, bu ağaçlardan alınan tahminlerin ortalaması (regresyon için) alınarak elde edilir.

Gradyan Artırmalı Regresyon: bir ensemble öğrenme yöntemi olup, doğrusal olmayan ve karmaşık regresyon problemleri için güçlü bir tekniktir. Bu yöntem, boosting adlı bir yaklaşımı kullanarak bir dizi zayıf öğreniciyi (genellikle karar ağaçları) birleştirir ve sonuç olarak güçlü bir model oluşturur. Gradient Boosting, her yeni modelin (ağaç) önceki modelin hatalarını düzeltmeye çalıştığı iteratif bir süreçtir.

XGBoost: özellikle büyük veri ve karmaşık yapılarla çalışırken yüksek doğruluk ve hız sağlayan bir makine öğrenmesi algoritmasıdır. XGBoost, Gradient Boosting algoritmasının geliştirilmiş bir versiyonudur ve genellikle sınıflandırma ve regresyon problemlerinde kullanılır. XGBoost, büyük veri setlerinde hızlı eğitim süresi ve yüksek doğruluk elde etmek için çeşitli optimizasyon tekniklerini ve özellikleri birleştirir.

Derin öğrenme, çok katmanlı yapay sinir ağları kullanarak verilerden öğrenmeyi sağlayan bir makine öğrenmesi yöntemidir. Bu yöntem, verileri temsil eden yüksek düzey özellikleri otomatik olarak öğrenebilir ve genellikle büyük veri setleriyle iyi performans gösterir. Derin öğrenme modelleri, her katmanda farklı özellikleri öğrenir ve bu sayede karmaşık problemleri çözebilir. Derin öğrenme, insan beynindeki sinir hücrelerini modelleyen yapay sinir ağları ile çalışır. Bu ağlar, çok sayıda gizli katmana sahip olabilir ve her katman, veriyi bir önceki katmandan aldığı bilgiyi işleyerek ilerler. Derin öğrenme, özellikle büyük miktarda veriyle eğitim yapıldığında başarılı sonuçlar verir ve çok katmanlı yapısı sayesinde karmaşık özellikleri ve ilişkileri öğrenebilir



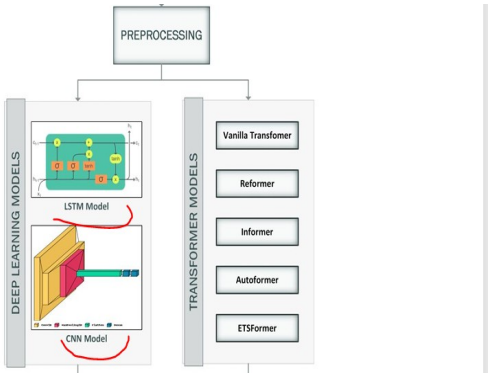

.

**Şekil 6 :** Sinir Ağları [11]

Sinir Ağları: Sürekli (sayısal) bir hedef değişkeni tahmin etmek için kullanılan bir yapay sinir ağı uygulamasıdır. Bu yöntem, doğrusal veya doğrusal olmayan ilişkileri öğrenebilmesi için gizli katmanlar ve aktivasyon fonksiyonları içeren bir yapay sinir ağı modeli kullanır. Regresyon sinir ağlarında, çıktı nöronunda herhangi bir sınıflandırma yapılmaz; bunun yerine, ağ, sürekli bir değeri tahmin etmeye çalışır.

RNN: RNN, insan beyninin geçmiş deneyimlerini hatırlayarak geleceği tahmin etmesine benzer şekilde, bilgisayarların da geçmiş verileri kullanarak gelecekteki olayları öngörmesini sağlayan bir yapay zekâ modelidir. Özellikle metinlerdeki kelimelerin birbiriyle olan ilişkilerini anlamak veya ses kayıtlarındaki konuşmaları metne dönüştürmek gibi görevlerde kullanılır.

Transformatörler, doğal dil işleme ve makine öğrenimi alanında büyük bir devrim yaratan bir derin öğrenme modelidir. Attention mekanizması sayesinde, bir dizideki her bir öğenin diğer tüm öğelerle olan ilişkisini hesaplayarak, uzun menzilli bağımlılıkları yakalar. Bu sayede, metin çevirisi, metin özetleme, soru-cevap sistemleri gibi birçok doğal dil işleme görevinde üstün performans gösterir. Transformer modelleri, geleneksel tekrarlayan sinir ağlarına göre daha paralel işlemlemeye uygun olduğu için, daha büyük ve karmaşık modelleri eğitmeyi mümkün kılar.




**Şekil 7 :** Ön İşleme ve Model Şeması

Vanilla Transformer: Transformatör mimarisinin en temel hali olan Vanilla Transformer, self-attention mekanizması sayesinde uzun dizilerdeki bağımlılıkları yakalama konusunda oldukça başarılıdır. Bu model, herhangi bir dizi üzerindeki öğeler arasındaki ilişkileri anlamak için her bir öğenin diğer tüm öğelerle olan ilişkisini hesaplar. Ancak, özellikle uzun dizilerde hesaplama maliyeti oldukça yüksek olabilir.

Reformer: Vanilla Transformer'ın hesaplama maliyeti sorununu çözmek için tasarlanmış bir modeldir. Lokalize self-attention ve Locality-Sensitive Hashing (LSH) gibi teknikleri kullanarak, uzun diziler üzerindeki hesaplamaları daha verimli hale getirir. Bu sayede, uzun metinler veya zaman serileri gibi büyük veri kümeleri üzerinde daha hızlı ve etkili bir şekilde çalışabilir.

Informer**:** Uzun zaman serileri tahmininde uzmanlaşmış bir Transformer çeşididir. Geleneksel Transformer modellerinin uzun zaman serilerindeki uzun vadeli bağımlılıkları yakalama konusunda zorlanması sorunu, Informer'da geliştirilmiş dikkat mekanizmaları sayesinde çözülmüştür. Ayrıca, mevsimsellik ve trend gibi zaman serilerindeki karakteristik özellikleri daha iyi modelleyebilme yeteneğine sahiptir.

ETSFormer: Klasik zaman serisi tahmin yöntemlerinden ilham alarak geliştirilmiş bir Transformer modelidir. Exponential smoothing (üstel düzeltme) gibi yöntemleri kullanarak zaman serilerindeki trend ve mevsimselliği daha iyi yakalar. Bu sayede, daha doğru ve güvenilir zaman serisi tahminleri yapabilir. Özellikle kısa vadeli tahminler için oldukça etkili bir yöntemdir.

AutoFormer: Transformatör mimarilerinin esnekliğini bir adım ileri taşıyarak, verilerin kendisine göre otomatik olarak uyarlanabilen bir modeldir. Geleneksel Transformatör modellerinde, modelin hiperparametreleri ve mimari yapısı genellikle önceden belirlenir ve tüm veriler için aynı kalır. Ancak, AutoFormer, veri setinin özelliklerine göre bu parametreleri ve yapıyı otomatik olarak öğrenerek, daha iyi performans elde etmeyi hedefler.

## 4. VERİ TOPLAMA

Veri toplama işlemleri için BeautifulSoup[12] kütüphanesi kullanılmıştır. Öncelikle arabam.com sitesi veri toplama işlemleri için seçilmiştir. Arabam.com sitesinin araç arama kısmına girilmiştir. Buradan BeautifulSoup kütüphanesi kullanılarak her bir ilanın verileri incelenmiştir. Öncelikle tüm bu işlemlerden önce websitenin açıldığı web tarayıcıda, öğeyi denetle seçeneği seçilerek websitenin kodları üzerinde inceleme yapılmıştır bunun sonucunda arama yapılacak etiket belirlenmiş ve ‘should-hover’ etiketi altında arama yapılmaya başlanmıştır.

Daha sonra ‘a’ etiketi içinde yer alan href ile kaydedilmiş bir link BeautifulSoup yardımı ile alınmıştır. Bu link, website içerisinde arama yapıldığında sergilenen her bir arabanın satışı için özelliklerinin bulunduğu o araca özel bir web sayfası linki içerir. Yani listelenenen her bir aracın detaylı özelliklerinin bulunduğu o araca özel bir sayfanın linki alınacaktır.

Bu sayfa elde edildikten sonra o sayfa içerisinde araç ile ilgili fiyat,marka,model,çekiş gücü gibi özellikler lxml kütüphanesinin etree fonksiyonu ile elde edilecektir. Yine öğeyi denetle yardımı ile özelliklerin yer aldığı sütunların xpath’i alınır ve kod içerisinde etree kütüphanesine gönderilen bu xpath ile websitede yer alan özellikler elde edilmiştir.

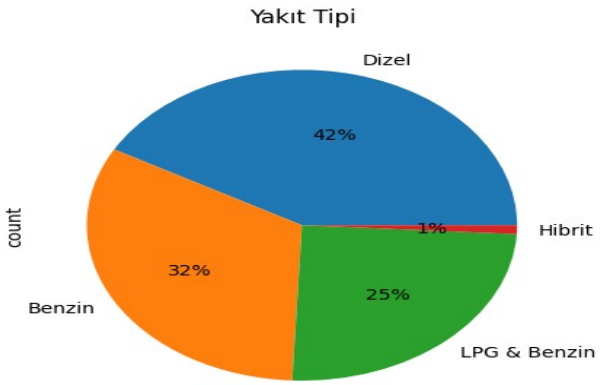
Elde edilen bu değerler Data isimli bir sözlük yapısında her bir özelliğin değeri karşısına gelecek şekilde kaydedilmiştir. For döngüsüne alınarak listelenen tüm araçlar için aynısı yapılmıştır. Daha sonra bu sözlük yapısı bir DataFrame’e dönüştürülmüştür ve csv dosyasına dönüştürülerek dışarı aktarılmıştır.

## 5. ÖZELLİK MÜHENDİSLİĞİ ve VERİ ANALİZİ

Bir önceki bölümde elde edilen veriler, makine öğrenmesi algoritmalarına girmeye hazır hale getirilmelidir. Bunun için de özellik mühendisliği denilen bir ön işleme süreci uygulanmalıdır. Elde edilen bu verilerin içerisinde boş değerler olabilir, websitesinden alınan değerler eksik olabilir yani bir araca ait özelliklerin hepsi siteye girilmediği için bazı özellikler alınmamış olabilir ve bunun sonucunda özelliklerde kayma meydana gelmiş olabilir, tekrar eden veriler de olabilir. Veri toplama süreci 1 günde yapılmadığı , haftalara uzayan bir süreç olduğu için aynı aracın bilgileri birden fazla kez kazınmış olabilir.

Bu veri setlerinde yapılan tüm işlemler için Pandas[13] kütüphanesine ait fonksiyonlar kullanılmıştır. Öncelikle boş değerlerin kontrolü yapılmıştır sonrasında da boş veriler silinmiştir.. Daha sonra da kopya verilerin temizliği yapılmıştır. Sayısal değerler dahi web kazıma aşamasında string değerler olarak alınmış olabilir. Mesela Fiyat ve Marka gibi değişkenlere ait değerler ‘140.000 TL’ ve ‘125.000 km’ olarak alınmıştır. Öncelikle bunların numerik değer olmaya hazır hale getirilmesi gerekmektedir. TL veya km gibi yazıların ve noktanın kaldırılması gibi işlemler yapılmalıdır. Bunun için ‘split’ ,’ replace’ gibi str fonksiyonları kullanılmıştır. Bu gibi sorunlara sahip tüm değişkenler için bu tip dönüşümler uygulanmış ve sonucunda numerik değerlere dönüşümleri tamamlanmıştır.

Daha sonra marka, renk gibi kategorik değişkenler incelenmiştir. Veri setinde veri kazıma aşamasında kaymaların meydana gelebileceğinden daha önce bahsetmiştik. Marka değişkenine ait ‘Kırmızı’, ‘Siyah’ gibi verilerin olduğu gözlemlenmiştir. Bunun sonucunda Pandas kütüphanesi ile bu aykırı değerler tek tek silinmiştir. Bu noktada kategorik değerleri incelemek için veri görselleştirme teknikleri uygulanmıştır.



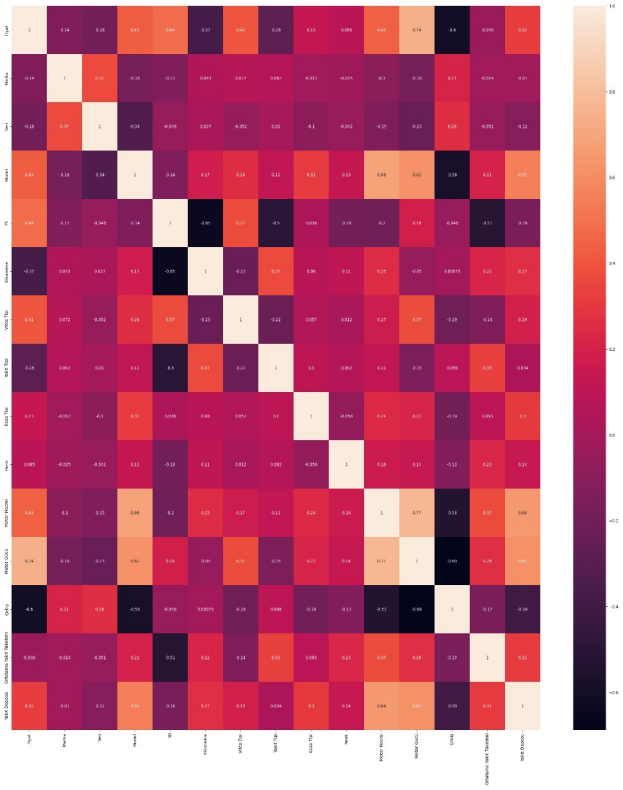

**Şekil 8 :** Yakıt Tipi Grafiği

Şekil-8’de de görüldüğü üzere Yakıt tipleri pasta grafiği ile görselleştirilmiştir.

Aykırı veriler örneğin aşırı yüksek fiyatlı veriler modelin doğruluğu ve güvenilirliğini zedeleyebilmektedir. Bundan dolayı aşırı yüksek fiyatlı verileri tespit edebilmek için ‘describe.T’ fonksiyonu kullanılmıştır. Aykırı verileri silmek için değerler sıralanmıştır ve belli bir fiyatın üstündeki değerler silinmiştir.

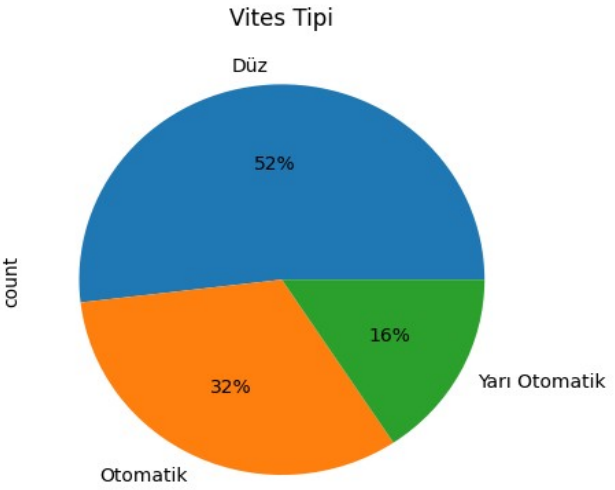
Boş verileri silme, kopyaları ve aykırı değerleri eleme işlemlerinden sonra geriye bir tek kategorik verilerin numerik hale getirilmesi kalmıştır. Bunun için sklearn [14] kütüphanesine ait ‘LabelEncoder’ fonksiyonu kullanılmıştır. Bu fonksiyon kategorik değişkenlere ait tüm verileri 1,2,3 olarak numaralandırır. Bütün kategorik değişkenler için ayrı ayrı sütunlarda bu işlem tekrarlanmış ve işlem sonucunda hangi numaranın hangi kategorik veriye ait olduğu listelenmiştir.

Tüm bu işlemlerden sonra verilerimiz makine öğrenmesi algoritmalarında kullanmak üzere hazır hale getirilmiştir. Daha sonrasında marka, fiyat gibi değişkenlerin birbiri arasındaki ilişkilerini görmek için bir dizi veri görselleştirme tekniği uygulanmıştır. Korelasyon matrisi ve ısı haritası gibi yöntemlere de başvurulmuştur.



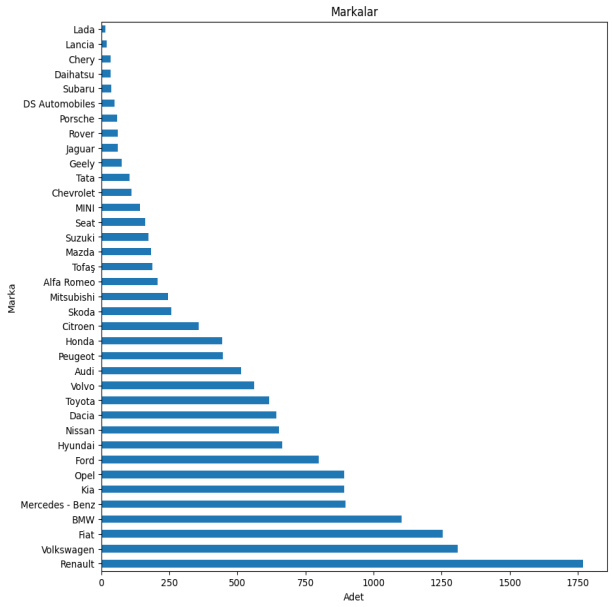
**Şekil 9 :** Isı Haritası

Şekil-9'da gözüktüğü gibi de ısı haritası oluşturulmuştur.



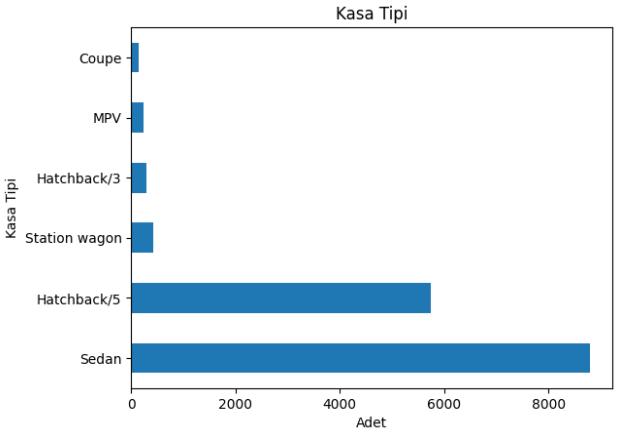

**Şekil 10 :** Vites Tipi

Şekil-10'da Vites Tiplerini gösteren bir pasta grafiği gösterilmiştir. %52 ile veri setinde en çok temsil edilen vites tipi düzdür.



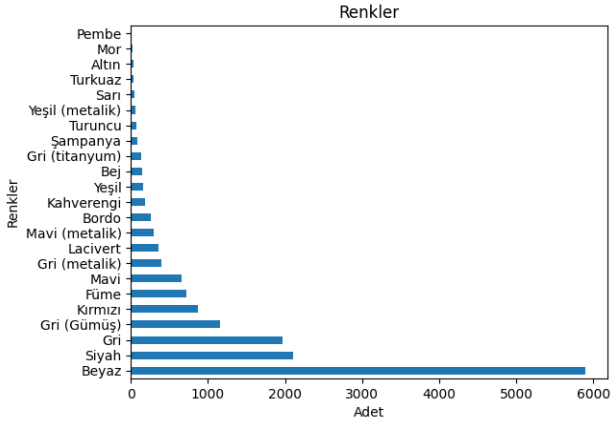

**Şekil 11:** Araba Markaları

Şekil-11'da araba markalarını gösteren grafik bulunmaktadır. Veriseti içerisinde en yaygın araba markası Renault olurken en az bulunan marka ise Lada olmuştur.



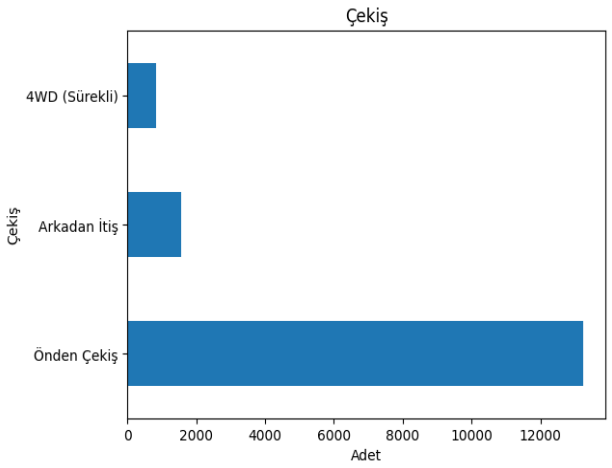

**Şekil 12 :** Kasa Tipleri

Şekil-12'de araçlara ait kasa tipleri görülmektedir. En fazla bulunan kasa tipi Sedan olurken en az rastlanan kasa tipi ise Coupe olmuştur.



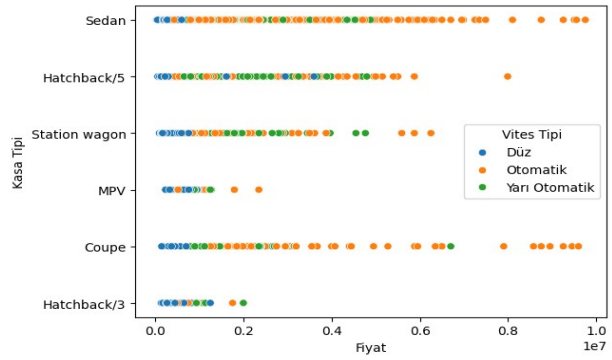

**Şekil 13:** Renkler

Şekil-13'te arabaların renkleri listelenmiştir. En çok bulunan araç rengi Beyaz olurken en az ise Pembe olmuştur.



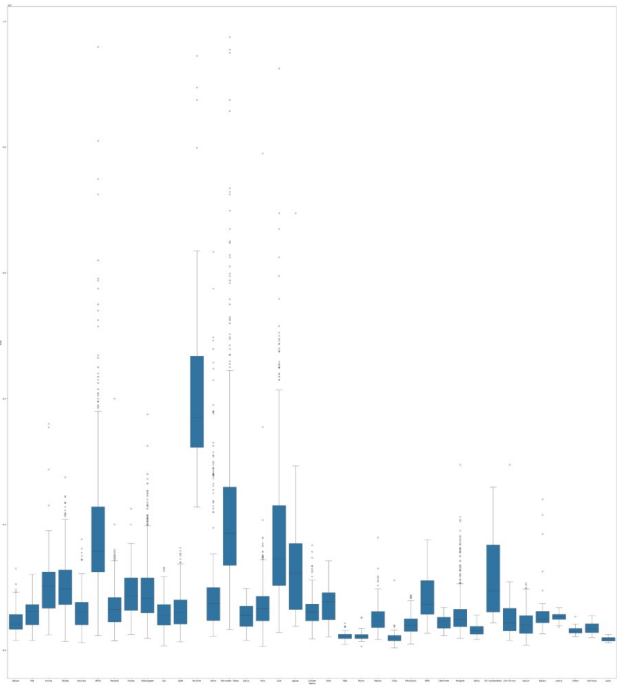

**Şekil 14:** Çekiş-İtiş

Şekil-14'de araçlara ait çekiş-itiş türleri listelenmektedir.



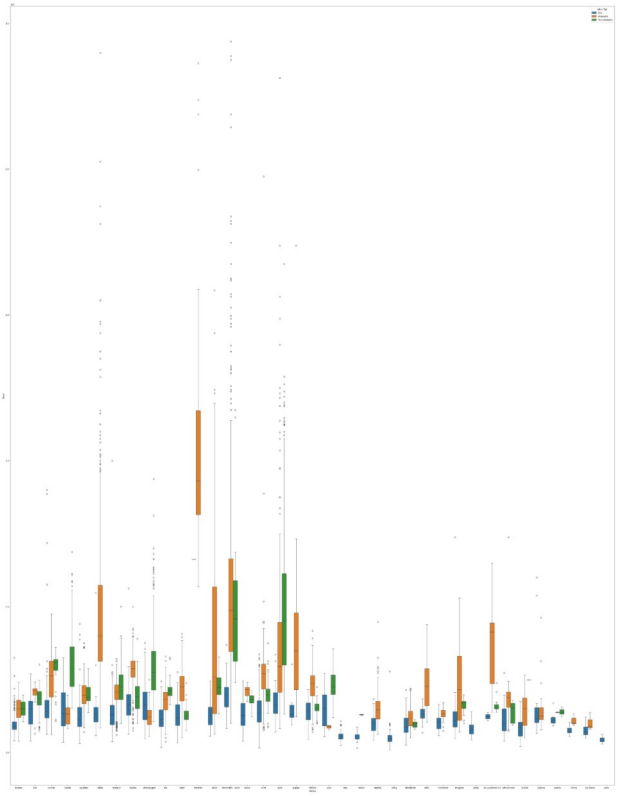

**Şekil 15:** Kasa Tipi-Vites Fiyat Tablosu

Şekil-15'teki tabloda. x-ekseninde fiyat yer alırken y-ekseninde kasa tipleri yer almaktadır. Turuncu renkli noktalar otomatik vitesi, maviler düz vitesi, yeşiller ise yarı otomatik vitesi temsil etmektedir. Her bir kasa tipinde genellikle düz vitesli araçların fiyatları en düşük olarak gözlemlenirken, genellikle en yüksek fiyatlı araçlar otomatik vites araçlarda görülmüştür.



**Şekil 16:** Marka Fiyat Tablosu

Şekil-16'daki tabloda ise her bir markanın fiyatları incelenmiştir. X-ekseninde araba markaları y-ekseninde ise fiyatları yer almaktadır. Bu bir boxplot grafiğidir. Kutu içerisindeki değerler 1. ve 3. çeyrek arası dağılımı ifade eder. Kutunun içindeki çizgi ise medyanı ifade eder. Kutu dışında kalan kuyruklar ise 1. ve 3. çeyrek dışlarında kalan maksimum ve minimum değerleri verir. Tüm bunların dışında kalan noktalar ise aykırı değerleri ifade eder. Grafikte en göze çarpan bilgi ise, en yüksek fiyatların Porsche markasında olduğudur.



**Şekil 17:** Marka-Fiyat-Vites Tipi Dağılımı

Şekil-17'deki tabloda marka ve vites tiplerine göre fiyatların boxplot grafiği alınmıştır. X-ekseninde markalar, y-ekseninde ise fiyatları yer almaktadır, buna ek olarak turuncu renkler otomatik vitesi, maviler düz vitesi, yeşiller ise yarı otomatik vitesi temsil etmektedir. Burada karşımıza çıkacak bilgi, neredeyse her markanın otomatik araç fiyatlarının daha yüksek olduğudur. Bazı markaların minimum fiyatları çoğu markadan daha pahalıdır. Bu da bize bu araçların lüks araçlar olduğunu göstermektedir.

## 6. MODELLER ve SONUÇ

| Parametreler | Model |
|---|---|
| | Linear Regresyon |
| n_estimators=1500,<br>min_samples_split=4,<br>min_samples_leaf=2,<br>max_features=0.48,<br>max_depth= 18 | Random Forest |
| n_estimators=300,<br>learning_rate=0.2,<br>max_depth=6,<br>subsample=0.8,<br>colsample_bytree=0.8,<br>reg_alpha=0.01,<br>reg_lambda=1,<br>gamma=0,<br>min_child_weight=1 | XGBoost |

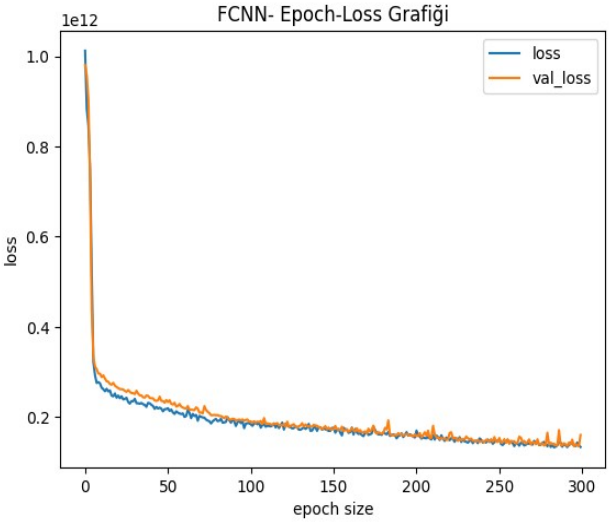
| | |
|---|---|
| hidden_units1 = 128<br>hidden_units2 = 64<br>learning_rate = 0.0001 | FCNN |
| dim = 32,<br>dim_out = 1,<br>depth = 6,<br>heads = 8,<br>attn_dropout = 0.3,<br>ff_dropout = 0.3,<br>mlp_hidden_mults = (4, 2),<br>mlp_act = nn.ReLU(), | TabTransformer |
| dim=32,<br>dim_out=1,<br>depth=8,<br>heads=8,<br>attn_dropout=0.2,<br>ff_dropout=0.2, | FTTransformer |
| transformer_dim=32,<br>dim_out =1,<br>transformer_depth=6,<br>transformer_heads=8,<br>attn_dropout=0.1,<br>ff_dropout=0.1,<br>mlp_act=nn.LeakyReLU(0),<br>mlp_depth=4,<br>mlp_dimension=32,<br>gmlp_enabled=True | Gated Tab Transformer |

Projede 3 adet transformer modeli kullanılmıştır. Tablo verileri için özelleştirilmiş TabTransformer[15] , FTTransformer[16] ve Gated Tab Transformer[17] gibi Transformer modelleri kullanılmıştır. Bu modeller github'da yer alan kodlardan ilham alınarak geliştirilmiştir.[18] Ayrıca bir adet FCNN sinir ağı modeline ek olarak Random Forest XGBoost ve Linear Regreasyon gibi klasik modeller de kullanılmıştır. Modeller MAPE,MAE,MSE,RMSE ve $R^2$ gibi performans metrikleri ile değerlendirilmiştir. En iyi $R^2$ skorunu gösteren model 0.937289 ile FTTransformer olmuştur. Aşağıdaki tabloda her bir modele ait performans metrikleri sonuçları verilmiştir.

| Model | MAE | MSE | RMSE | MAPE | $R^2$ |
|---|---|---|---|---|---|
| Linear Regresyon | 228603.375881 | 1.595249e+11 | 399405.707576 | 34.642957 | 0.620508 |
| Random Forest | 94320.635893 | 5.231751e+10 | 228730.214205 | 11.214376 | 0.894179 |
| XGBoost | 95849.039062 | 5.362738e+10 | 231575.870522 | 11.621774 | 0.904842 |
| FCNN | 242786.265625 | 1.599375e+11 | 399921.922620 | 33.091139 | 0.627151 |
| TabTransformer | 115476.226562 | 7.221381e+10 | 268726.267983 | 0.135986 | 0.882044 |
| FTTransformer | 83878.875000 | 3.899606e+10 | 197474.206437 | 0.103461 | 0.937289 |
| GatedTab Transformer | 99368.156250 | 6.684354e+10 | 258541.179513 | 0.117135 | 0.892506 |

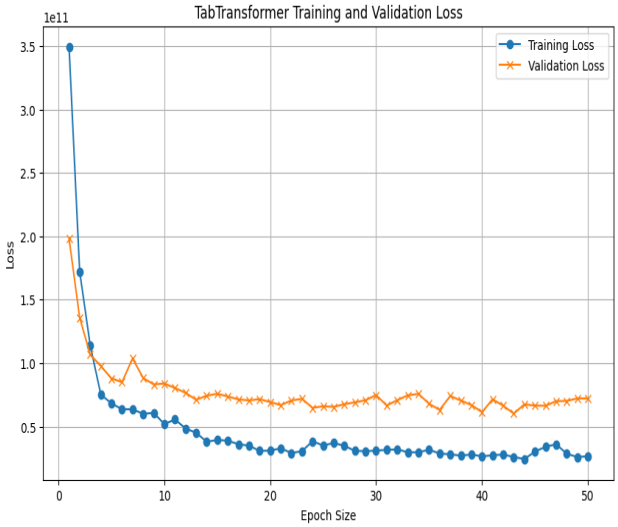
Tabloda da görüldüğü üzere özellikle FTTransformer yaklaşımı çok iyi bir sonuç ortaya çıkarmıştır. %93 $R^2$ puanı ile modelin genelinin varyansı ne kadar iyi açıkladığı ortaya çıkmıştır. Ayrıca %10.3 MAPE ile yine iyi bir sonuç ortaya koymuştur.

| Model | Eğitim Zamanı(s) | Çıkarım Zamanı(s) |
|---|---|---|
| Linear Regresyon | 0.032510 | 0.005085 |
| Random Forest | 19.663338 | 2.673624 |
| XGBoost | 1.719451 | 0.092776 |
| FCNN | 312.037215 | 0.684576 |
| Tab Transformer | 1594.786427 | 1.892621 |
| FTTransformer | 1710.360000 | 7.759836 |
| GatedTab Transformer | 3180.640170 | 15.541411 |



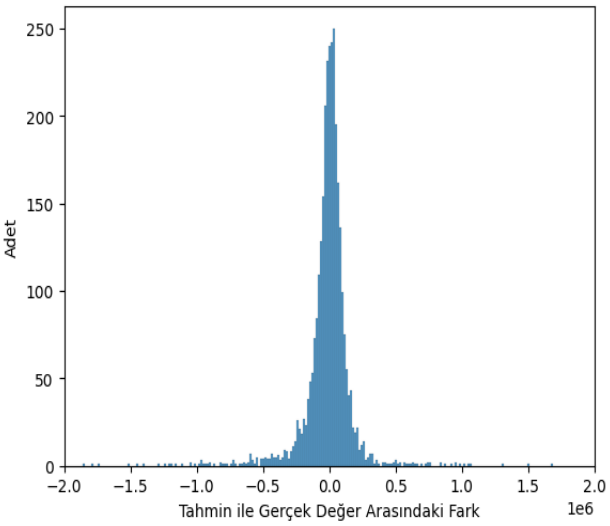

**Şekil 18:** FCNN Epoch Size Loss

Şekil-18'de bir sinir ağları modeli olan FCNN için epoch size loss grafiğini yer almaktadır. Bu model aşırı öğrenmeye yatkın değildir.



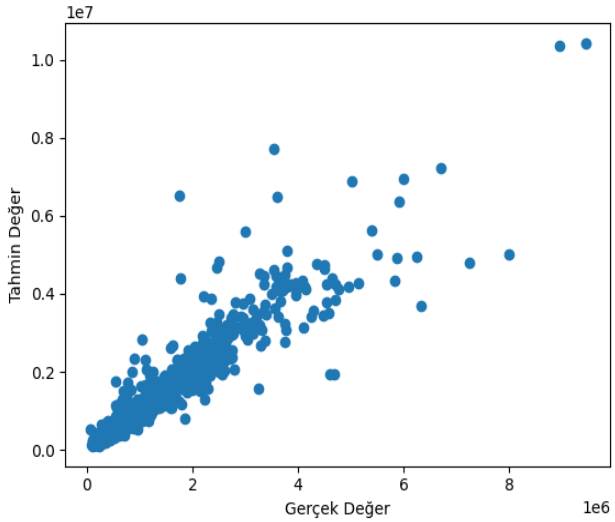

**Şekil 19:** TabTransformer Epoch Size Loss

Şekil-19'da TabTransformer modeli için epoch size -loss grafiği gösterilmiştir. Train ve val loss'ları benzer bir şekilde hareket etse dahi, val-loss verilerinde çok fazla yukarı aşağı iniş gözlemlenmektedir. Bu durum aşırı öğrenmeyi işaret edebilecek olsa dahi, fiyat verilerinde düşük fiyatlar ve yüksek fiyatlar aynı anda yer aldığı için bu dalgalanmalara modelin farklı fiyat durumlarında farklı tepkiler vermesi de sebep olmuş olabilir.



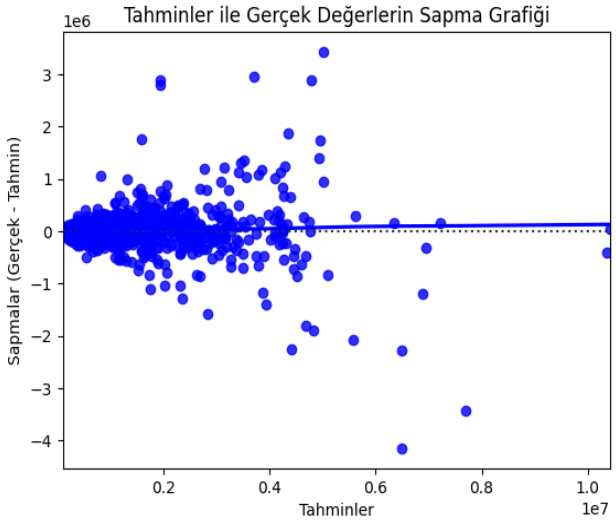
**Şekil 20:** TabTransfer Histogram

Şekil-20'de TabTransformer modeline ait bir histogram grafiği yer almaktadır. Bu grafikte x-ekseninde yer alan değerler gerçek değer ile tahminlerin farkına yer verirken, y-ekseninde ise bu farkın adet sayısı yer almaktadır. Dağılımın büyük bir çoğunluğu 250.000 ile -250.000 civarı arasında yer almaktadır. 2.000.000'a yakın farklar çok az dahi olsa bulunmaktadır.



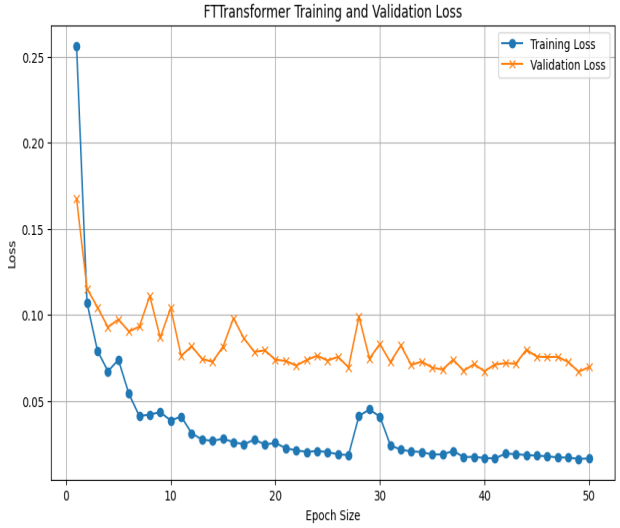

**Şekil 21:** TabTransformer Scatter

Şekil-21'de TabTransformer için scatterplot grafiği yer almaktadır. X-ekseninde yer alan noktalar gerçek değeri, y-ekseninde yer alan noktalar ise tahmin edilen değeri simgelemektedir.



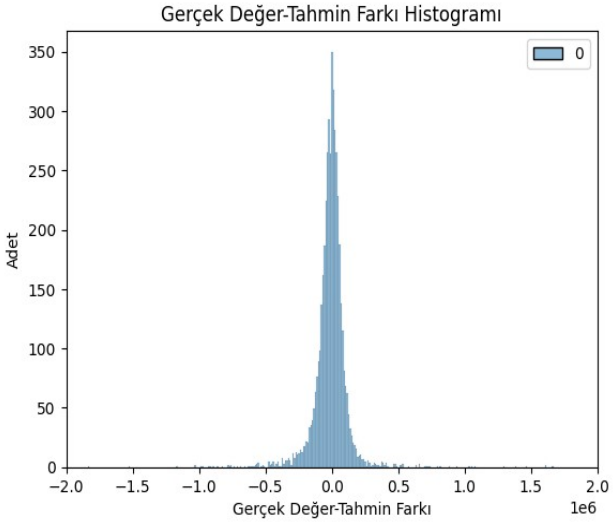

**Şekil 22:** TabTransformer Sapma Eğrisi

Şekil-22'de TabTransformer modeli için sapma eğrisi yer almaktadır. X-ekseninde tahminler, y-ekseninde ise gerçek değer ile tahminlerin farkı yer almaktadır. Mavi çizgi ortalama hatayı verir. Mavi çizginin üzerindeki noktalar, modelin gerçek değeri abarttığı tahminleri gösterir. Mavi çizginin altındaki noktalar ise modelin gerçek değeri küçümsediği tahminleri gösterir.



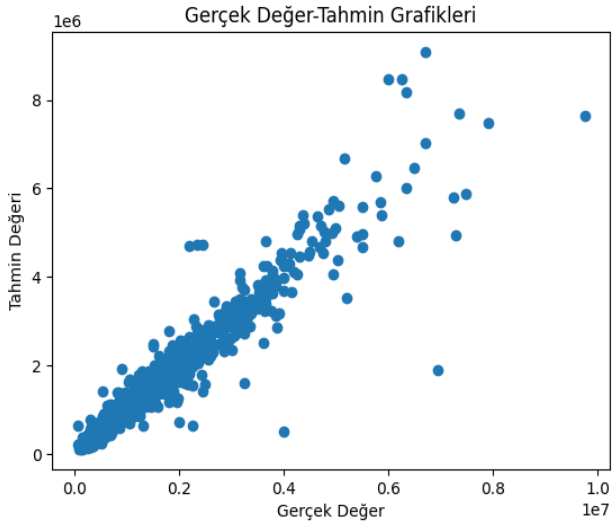

**Şekil 23:** FTTransformer Epoch-Loss Size

Şekil-23'te FTTransformer modeline ait epoch-size loss grafiğine yer verilmiştir. Val ve train verileri yine benzer bir şekilde hareket etmiştir.



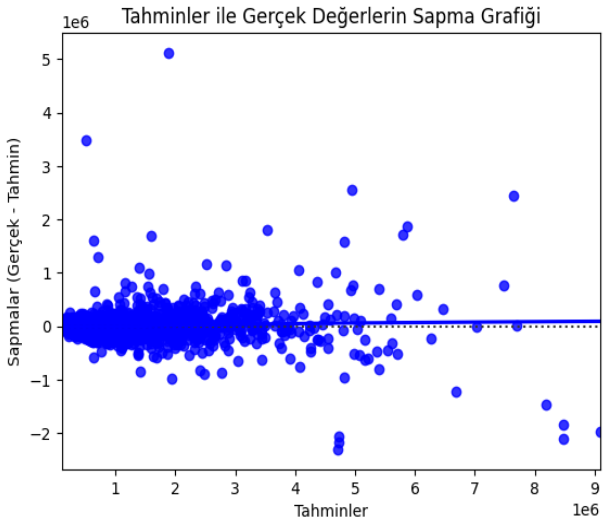

**Şekil 24:** FTTransformer Histogram

Şekil-24'te FTTransformer modeline ait histogram grafiğine yer verilmiştir.



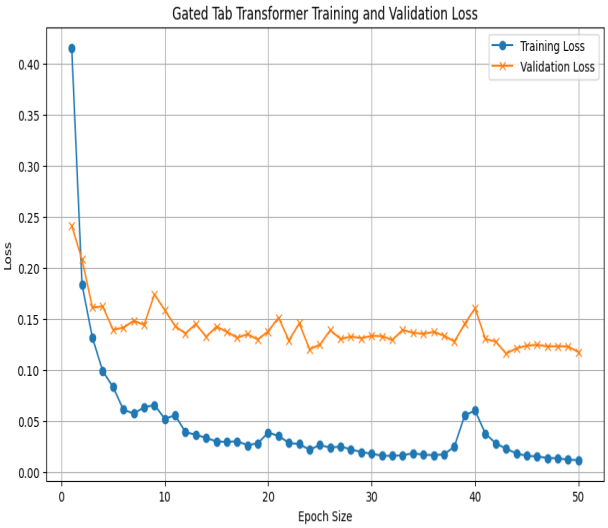
**Şekil 25:** FTTransformer Scatter

Şekil-25'te FTTransformer modeline ait scatterplot grafiğine yer verilmiştir. Bu grafik TabTransformer modeline göre daha düz bir çizgi halinde ilerleyerek daha iyi bir performans gösterdiğini belli etmiştir.



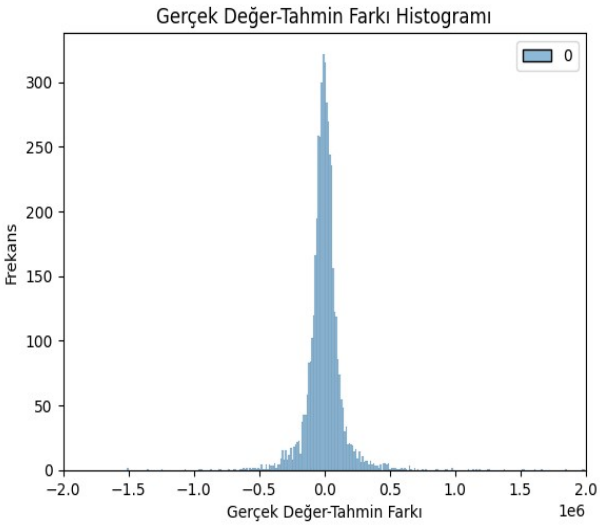

**Şekil 26:** FTTransformer Sapma Grafiği

Şekil-26'da FTTransformer modeline ait gerçek değerlerden sapma grafiğine yer verilmiştir.



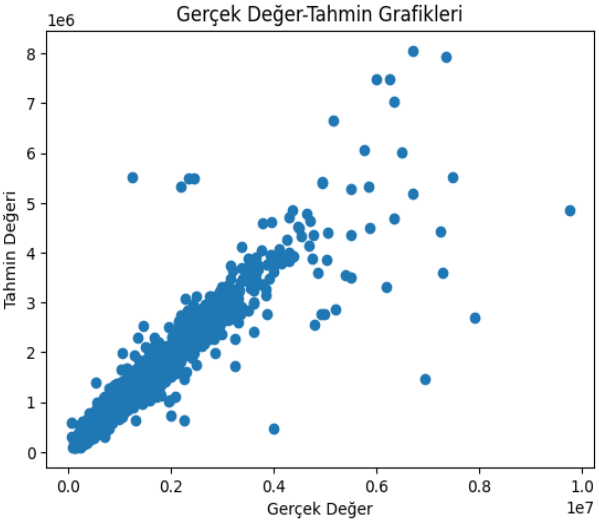

**Şekil 27:** GatedTab Transformer Epoch Size Loss Grafiği

Şekil-27'de GatedTab Transformer modeli için epoch size loss grafiğine yer verilmiştir. Model 43.epoch sonrasından itibaren overfitting'e maruz kalmıştır.



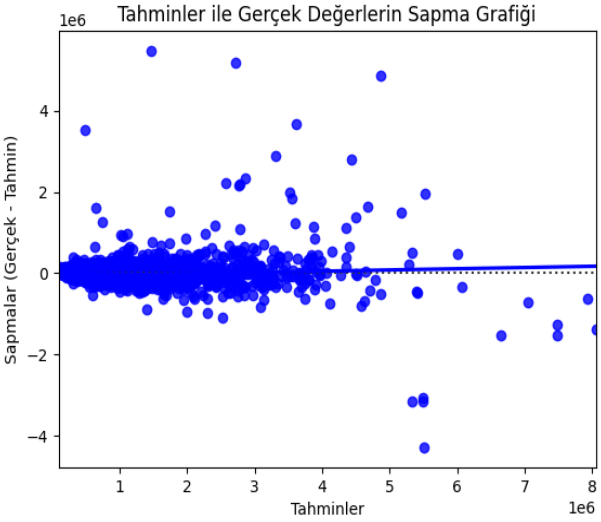

**Şekil 28:** GatedTab Transformer Histogram

Şekil-28'de GatedTab Transformer modeline ait histogram grafiğine yer verilmiştir.




**Şekil 29:** GatedTab Transformer Scatter

Şekil-29'da GatedTab Transformer modeli için scatterplot grafiğine yer verilmiştir.




**Şekil 30:** GatedTab Transformer Sapma Grafiği

Şekil-30'da GatedTab Transformer modeli için sapma grafiğine yer verilmiştir.

## 7. SONUÇ

Test edilen modeller arasında en iyi sonuçları ve grafik olarak da en doğru sonuçları FTTransformer modeli vermiştir. %93.7 $R^2$ başarısı göstermiştir. MSE skoru biraz yüksek çıkmıştır bu bize modelin yüksek fiyatlı verilerde olan sorununu ifade eder. İlerleyen çalışmalarda daha doğru sonuçlara ulaşmak için ortaya çıkan sonuçları iyi bir şekilde analiz edip, modellerin eksikliği kapatılmaya çalışılacaktır.

## 8. KAYNAKÇA


[1] en, "2023'te en çok satılan ikinci el oto markaları belli oldu," *Aa.com.tr*, 2023. https://www.aa.com.tr/tr/ekonomi/2023te-en-cok-satilan-ikinci-el-oto-markalari-belli-oldu/3107817 (accessed Oct. 23, 2024).

[2] C. Jin, "Price prediction of used cars using machine learning," *2021 IEEE International Conference on Emergency Science and Information Technology (ICESIT)*, pp. 223–230, Nov. 2021. doi:10.1109/icesit53460.2021.9696839

[3] X. Jiang, "Research for Car Price Prediction Base on Machine Learning," Transactions on Computer Science and Intelligent Systems Research, vol. 5, pp. 1608–1617, Aug. 2024, doi: https://doi.org/10.62051/k55feh59.

[4] M. Hankar, M. Birjali, and A. Beni-Hssane, "Used car price prediction using Machine Learning: A case study," *2022 11th International Symposium on Signal, Image, Video and Communications (ISIVC)*, pp. 1–4, May 2022. doi:10.1109/isivc54825.2022.9800719

[5] B. GÜLMEZ and S. KULLUK, "Türkiye'de Ikinci El Araçların Büyük Veri Ve Makine öğrenme Teknikleriyle Analizi ve Fiyat Tahmini," *Gazi Üniversitesi Mühendislik Mimarlık Fakültesi Dergisi*, vol. 38, no. 4, pp. 2279–2290, Apr. 2023. doi:10.17341/gazimmfd.980840

[6] Ö. Çelik and U. Ö. Osmanoğlu, "İkinci El Araba fiyatlarının tahmini," *European Journal of Science and Technology*, pp. 77–83, Aug. 2019. doi:10.31590/ejosat.542884

[7] K. Kargın, Ed., Accessed: Nov. 08, 2024. [Online]. Available: https://medium.com/kaveai/ridge-ve-lasso-regresyonu-temel-matemati%C4%9Fi-ve-python-uygulamas%C4%B1yla-363916e32d8d

[8]ruveydasn, Ed., 2023. Accessed: Nov. 08, 2024. [Online]. Available: https://medium.com/@nursenruveyda/polinomal-regresyon-nedir-pythonda-nas%C4%B1l-kodlan%C4%B1r-1255c4c500b9

[9] M. F. Akca, Ed., *Nedir Bu Destek Vektör Makineleri? (Makine Öğrenmesi Serisi-2)*. 2020. Accessed: Nov. 08, 2024. [Online]. Available: https://medium.com/deep-learning-turkiye/nedir-bu-destek-vekt%C3%B6r-makineleri-makine-%C3%B6%C4%9Frenmesi-serisi-2-94e576e4223e

[10] W. Koehrsen, Ed., *Random Forest Simple Explanation*. 2017. Accessed: Nov. 08, 2024. [Online]. Available: https://williamkoehrsen.medium.com/random-forest-simple-explanation-377895a60d2d

[11] F. Gulsen, Ed., *Sinir Ağları Regresyonu — TensorFlow*. 2021. Accessed: Nov. 08, 2024. [Online]. Available: https://furkangulsen.medium.com/sinir-a%C4%9Flar%C4%B1-regresyonu-tensorflow-bcb74c8d7861

[12]L. Richardson, "Beautiful Soup Documentation — Beautiful Soup 4.4.0 documentation," *Crummy.com*, 2019. https://www.crummy.com/software/BeautifulSoup/bs4/doc/

[13]Pandas, "pandas documentation — pandas 1.0.1 documentation," *pandas.pydata.org*, 2024. https://pandas.pydata.org/docs/

[14]Scikit-learn, "scikit-learn: Machine Learning in Python," *Scikit-learn.org*. https://scikit-learn.org/stable/

[15] X. Huang, Ashish Khetan, M. Cvitkovic, and Zohar Karnin, "TabTransformer: Tabular Data Modeling Using Contextual Embeddings," Dec. 2020, doi: https://doi.org/10.48550/arxiv.2012.06678.

[16] Yury Gorishniy, I. Rubachev, V. Khrulkov, and A. Babenko, "Revisiting Deep Learning Models for Tabular Data," *arXiv (Cornell University)*, Jun. 2021, doi: https://doi.org/10.48550/arxiv.2106.11959.

[17] radi-cho, "GitHub - radi-cho/GatedTabTransformer: A deep learning tabular classification architecture inspired by TabTransformer with integrated gated multilayer perceptron.," *GitHub*, 2021. https://github.com/radi-cho/gatedtabtransformer (accessed Jan. 19, 2025).

[18] lucidrains, "GitHub - lucidrains/tab-transformer-pytorch: Implementation of TabTransformer, attention network for tabular data, in Pytorch," *GitHub*, Dec. 28, 2024. https://github.com/lucidrains/tab-transformer-pytorch (accessed Jan. 19, 2025).