

Laptop Fiyat Tahmin Modeli

1. Giriş

1.1 Projenin Amacı

Bu proje kapsamında, laptopların marka, model, komponent ve fiyatlarının bulunduğu bir veri seti üzerinden TensorFlow kullanılarak bir derin öğrenme modeli oluşturulmuştur. Bu modelin amacı, girilen laptop bilgilerinden yola çıkarak laptopun fiyatını tahmin edebilmesidir.

1.2 Proje Kapsamı

Model, eğitime sokulmadan önce veriler üzerinde optimizasyonlar yapılmış ve sonrasında modelin eğitim parametreleri değiştirilerek optimum doğrulukta model elde edilmeye çalışılmıştır. Çalışmada kullanılan veri seti, laptop modellerinin teknik özelliklerini (RAM, işlemci tipi, ekran boyutu vb.) ve fiyat bilgilerini içermektedir. Veri setinin analizi, ön işleme adımları ve özelliklerin ölçeklendirilmesi önemli bir yer tutmaktadır.

2. Veri Seti Analizi ve Öznitelik Mühendisliği

2.1 Veri Seti Özellikleri

Çalışmada kullanılan veri seti, çeşitli laptop modellerinin teknik özelliklerini ve fiyat bilgilerini içermektedir. Veri setinde bulunan özellikler şunlardır:

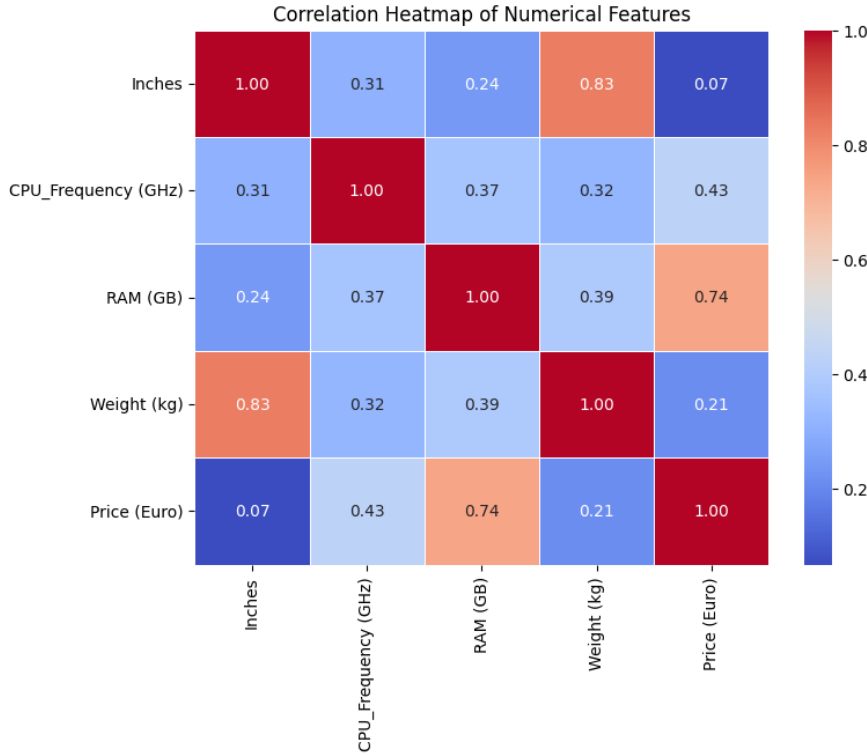
- Üretici firma (Company)
- Ürün adı (Product)
- Laptop türü (TypeName)
- Ekran çözünürlüğü (ScreenResolution)
- İşlemci üreticisi (CPU_Company)
- İşlemci tipi (CPU_Type)
- RAM kapasitesi (Memory)
- Ekran kartı üreticisi (GPU_Company)
- Ekran kartı tipi (GPU_Type)
- İşletim sistemi (OpSys)
- Fiyat bilgisi (Euro cinsinden)

Veri setinde yer alan numerik değişkenlerin istatistiksel özetine ait şema, şekil 2.1’de gösterilmiştir. Veri setindeki sayısal özellikler arasındaki ilişkilerin daha iyi anlaşılabilmesi için bir korelasyon heatmap’i oluşturulmuştur (Şekil 2.2). Bu heatmap, her bir sayısal özelliğin diğerleriyle olan doğrusal ilişkisini görselleştirmektedir. Korelasyon katsayısı değerleri -1 ile 1 arasında değişmekte olup, 1’e yakın değerler güçlü pozitif ilişkiyi, -1’e yakın değerler ise güçlü negatif ilişkiyi ifade

etmektedir. Özellikle fiyat ile diğer özellikler arasındaki korelasyon değerleri, modelin performansını etkileyebilecek belirleyici değişkenlerin tespiti açısından önemli ipuçları sunmaktadır.

	Inches	CPU_Frequency (GHz)	RAM (GB)	Weight (kg)	Price (Euro)
count	1275.000000	1275.000000	1275.000000	1275.000000	1275.000000
mean	15.022902	2.302980	8.440784	2.040525	1134.969059
std	1.429470	0.503846	5.097809	0.669196	700.752504
min	10.100000	0.900000	2.000000	0.690000	174.000000
25%	14.000000	2.000000	4.000000	1.500000	609.000000
50%	15.600000	2.500000	8.000000	2.040000	989.000000
75%	15.600000	2.700000	8.000000	2.310000	1496.500000
max	18.400000	3.600000	64.000000	4.700000	6099.000000

Şekil 2.1 Veri setinin numerik değişken istatistikleri



Şekil 2.2 Sayısal Verilerin arasındaki korelasyon ısı haritası

2.1 Kategorik Veri Dönüşümü

Veri setindeki kategorik değişkenler, makine öğrenmesi modelinin işleyebileceği sayısal formata dönüştürülmüştür. Bu dönüşüm için scikit-learn kütüphanesinin LabelEncoder sınıfı kullanılmıştır.

LabelEncoder, her bir kategorik değişkeni 0'dan başlayan tam sayılara dönüştürür. Kategorik veri dönüşümü öncesi ve sonrası veri tablosunun görünümü şekil 2.3'de gösterilmiştir.

Company	Product	TypeName	Company	Product	TypeName
Apple	MacBook Pro	Ultrabook	1	300	4
Apple	Macbook Air	Ultrabook	1	301	4
HP	250 G6	Notebook	7	50	3
Apple	MacBook Pro	Ultrabook	1	300	4
Apple	MacBook Pro	Ultrabook	1	300	4

Şekil 2.3 LabelEncoding öncesi ve sonrası veri tablosu

2.2 Veri Normalizasyonu ve Veri Bölümleme

Özellikler arasındaki ölçek farklılıklarını gidermek ve model performansını artırmak amacıyla MinMaxScaler kullanılmıştır. Bu normalizasyon işlemi:

- Giriş özelliklerini (X) [0,1] aralığına ölçeklendirir
- Hedef değişkeni (y) de aynı aralığa normalize eder
- Model tahminlerinin gerçek değerlere dönüştürülmesini kolaylaştırır

Model eğitimi esnasında K-Fold Cross Validation yöntemi uygulanmış ama aşırı öğrenme (overfitting) durumu gözlemlendiği için Hold Out yöntemi tercih edilmiştir. Veri seti şu şekilde

bölünmüştür:



Hold-out Yöntemi

Model eğitimi için veri setinin %80'i eğitim, %20'si test verisi olarak ayrılmıştır.

Şekil 2.4 Eğitim ve Test verisi bölümlendirmesi

3. Model Mimarisi Geliştirme

3.1 Temel Konfigürasyon

Geliştirilen derin öğrenme modelinde, optimizasyon algoritması olarak **Adam optimizer** tercih edilmiştir. Adam optimizer, adaptif öğrenme oranı ve momentum özelliklerini birleştirerek etkili bir gradyan iniş optimizasyonu sağlamaktadır. Model performansının değerlendirilmesi için temel kayıp fonksiyonu olarak **Mean Squared Error (MSE)** kullanılmıştır. MSE, tahmin edilen değerler ile gerçek değerler arasındaki farkın karesini alarak modelin performansını ölçmektedir.

Model eğitimi sırasında performans değerlendirmesi için çeşitli metrikler kullanılmıştır. **Mean Absolute Error (MAE)**, **Root Mean Squared Error (RMSE)** ve **Cosine Similarity** metrikleri, modelin tahmin performansını farklı açılardan değerlendirmemize olanak sağlamıştır. Özellikle RMSE, MSE'nin karekökünü alarak hata oranını daha anlaşılır bir ölçekte sunmaktadır.

Modelin aşırı öğrenmesini (overfitting) engellemek için **Early Stopping** mekanizması implemente edilmiştir. Bu mekanizma, validation loss değerini takip ederek, 10 epoch boyunca iyileşme görülmediği takdirde eğitimi sonlandırmaktadır. Ayrıca `restore_best_weights` parametresi `true` olarak ayarlanarak, eğitim sürecinde elde edilen en iyi model ağırlıklarının korunması sağlanmıştır.

Eğitim verisi, %80 eğitim ve %20 validasyon seti olarak bölünmüş, maksimum epoch sayısı 100 olarak belirlenmiştir. Bu yapılandırma, modelin yeterli öğrenme süresine sahip olmasını sağlarken, Early Stopping mekanizması ile gereksiz eğitim süresini önlemektedir.

Modelin eğitim sürecini optimize etmek için çeşitli normalizasyon ve regularizasyon teknikleri uygulanmıştır. Her Dense katmandan sonra **Batch Normalization** uygulanarak, katmanlar arası

değer dağılımlarının normalize edilmesi sağlanmıştır. Bu teknik, gradyan akışını iyileştirerek modelin daha hızlı ve kararlı bir şekilde öğrenmesine katkıda bulunmaktadır.

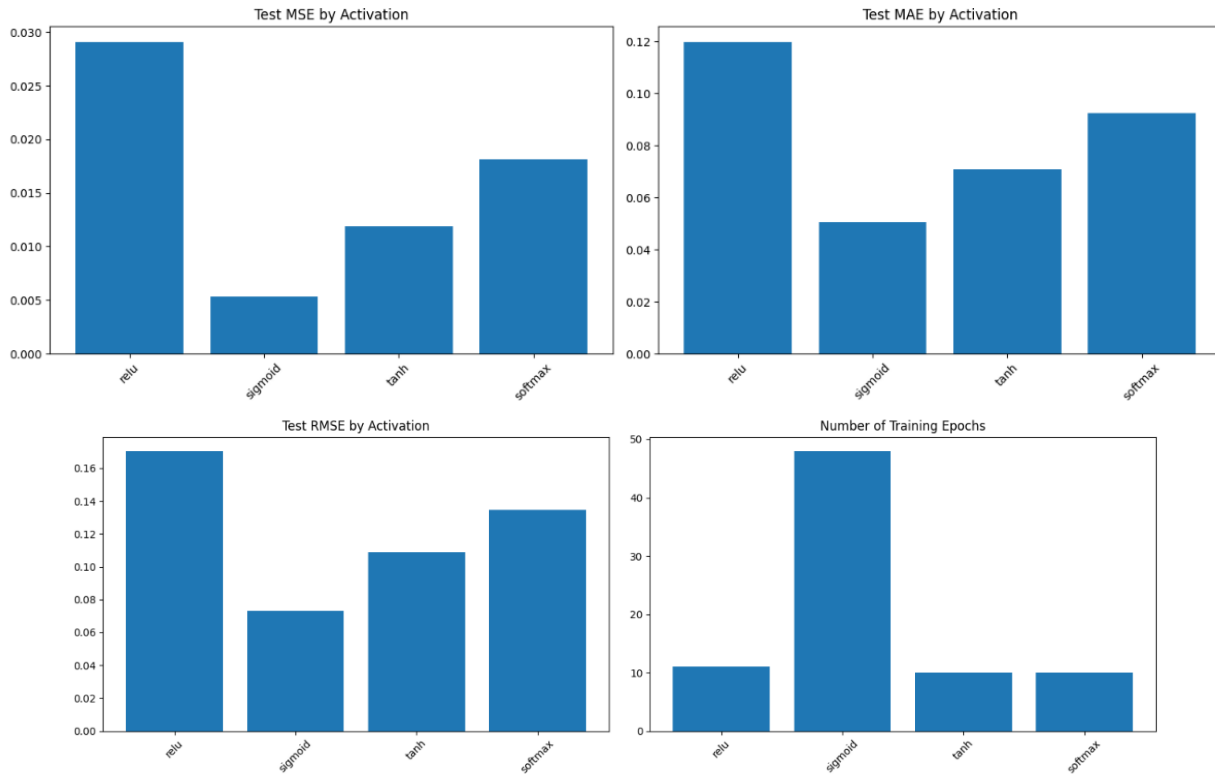
3.2 Parametre Optimizasyonu

3.2.1 Aktivasyon Fonksiyonu Analizi

Farklı aktivasyon fonksiyonlarının model performansına etkisini incelemek amacıyla bir dizi deney gerçekleştirilmiştir. Bu deneylerde, modelin mimarisi, eğitim parametreleri (optimizer, öğrenme hızı, epoch sayısı, vb.) ve veri ön işleme adımları sabit tutulmuş; yalnızca aktivasyon fonksiyonları değiştirilerek performans karşılaştırması yapılmıştır. Her aktivasyon fonksiyonu için modelin hata metrikleri (MSE, MAE ve RMSE) analiz edilmiş ve yakınsama hızları

değerlendirilmiştir. Bu yöntem, aktivasyon fonksiyonlarının fiyat tahmin doğruluğuna olan etkisini izole ederek belirginleştirmeyi amaçlamıştır.

Yapılan analizler sonucunda, **Sigmoid aktivasyon fonksiyonu** tüm hata metriklerinde en iyi performansı göstermiştir (MSE: 0.005, MAE: 0.05, RMSE: 0.07). Karşılaştırılan aktivasyon fonksiyonlarının aynı koşullardaki tahmin başarıları şekil 3.1’de gösterilmiştir.



Şekil 3.1 Aktivasyon Fonksiyonlarının Karşılaştırması

3.2.2 RandomizedSearchCV ile Hiperparametre Optimizasyonu

Modelin optimal mimarisini belirlemek amacıyla, farklı gizli katman sayıları ve hiperparametre kombinasyonları üzerinde sistematik bir optimizasyon çalışması gerçekleştirilmiştir. Bu çalışmada, 2, 3 ve 4 gizli katmanlı model mimarileri test edilmiş, her bir mimari için RandomizedSearchCV yöntemi kullanılarak kapsamlı bir hiperparametre optimizasyonu yapılmıştır.

Her bir model mimarisi (2, 3 ve 4 gizli katmanlı) için aşağıdaki hiperparametreler optimize edilmiştir:

- Öğrenme oranı (learning rate): [0.0001, 0.001, 0.01]
- Batch size: [16, 32, 64]
- Epoch sayısı: [50, 75, 100]
- Dropout size: [0.1, 0.2, 0.3]
- Activation: [relu, sigmoid, tanh, softmax]

Model mimarilerinin yapısı şu şekilde tasarlanmıştır:

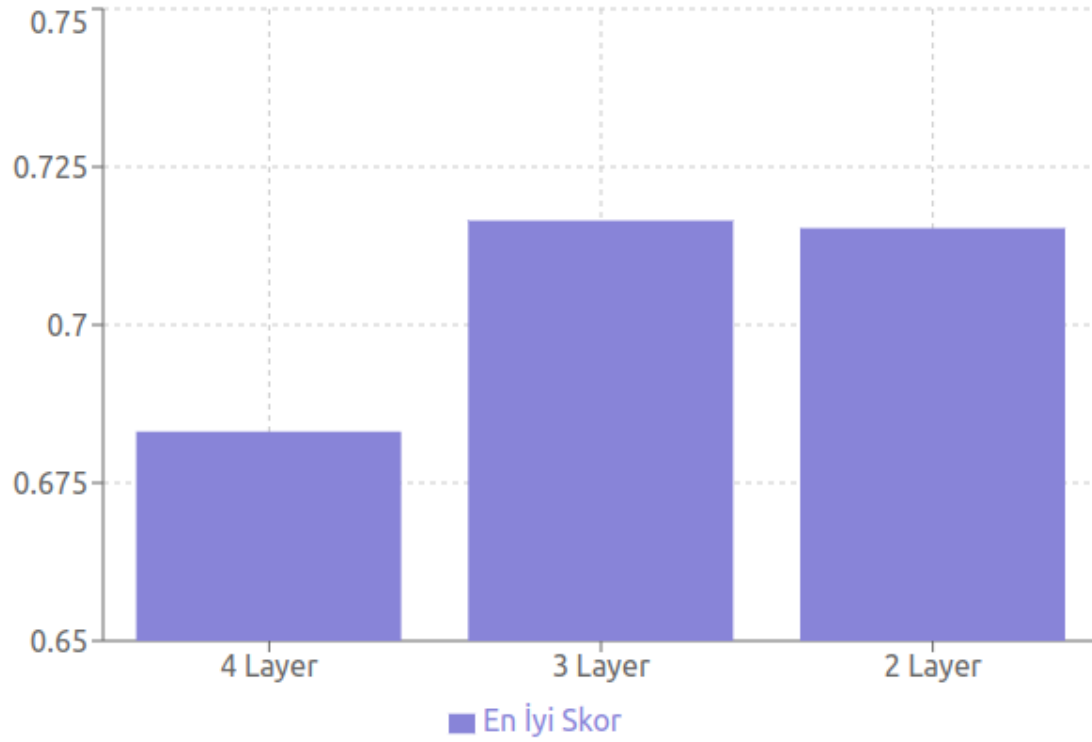
- 2 Gizli Katmanlı Model: [64, 32] nöron
- 3 Gizli Katmanlı Model: [64, 32, 16] nöron
- 4 Gizli Katmanlı Model: [64, 32, 16, 8] nöron

Her mimari için Dropout oranları katman sayısına göre optimize edilmiş ve BatchNormalization katmanları eklenmiştir. Önceki analizlerde en iyi performansı gösteren Sigmoid aktivasyon fonksiyonu tüm modellerde kullanılmıştır.

Hiperparametre optimizasyonu için RandomizedSearchCV yöntemi tercih edilmiştir. Bu yöntem:

- Her mimari için 20 farklı hiperparametre kombinasyonu test edilmiştir
- 3-katlı çapraz doğrulama (cross-validation) kullanılmıştır
- Paralel işlem yapılarak hesaplama verimliliği artırılmıştır
- MSE (Mean Squared Error) değeri optimize edilecek metrik olarak belirlenmiştir

Optimizasyon sonucunda tüm modeller için optimal hiperparametreler benzerlik göstermiş olup, **0.01** learning rate, **sigmoid** aktivasyon fonksiyonu, **100** epoch ve **32** batch size değerleri en iyi performansı sağlamıştır. Katman sayısına göre test skorları şekil 3.2’de gösterilmiştir.



Şekil 3.2 Katman sayısına göre test skorları

Performans karşılaştırmasında, **3 gizli katmanlı** model **0.716'lık** başarı skoruyla en iyi sonucu vermiştir. Bunu **0.715** skorla **2 gizli katmanlı** model takip etmiştir. **4 gizli katmanlı** model **0.683** skorla diğer mimarilerin gerisinde kalmıştır.

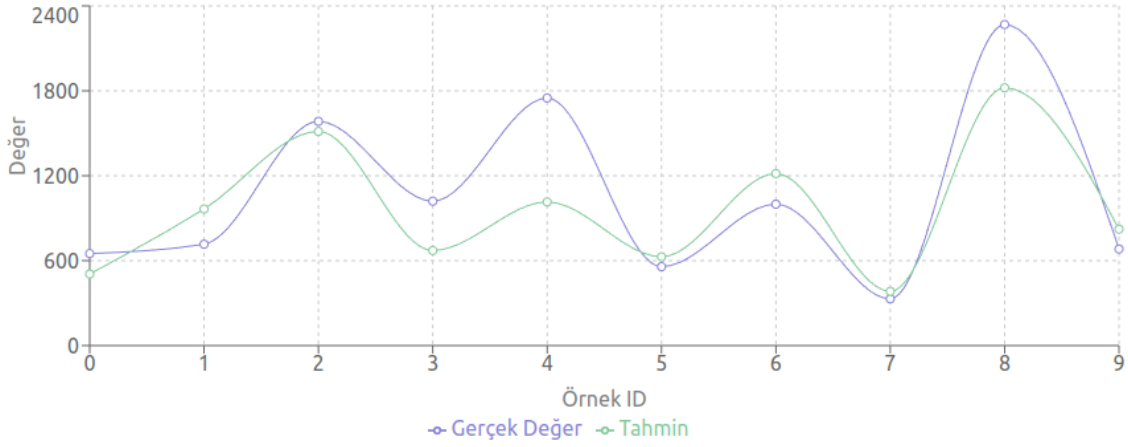
4. Sonuçlar ve Değerlendirme

Test seti üzerinde yapılan değerlendirmede, model 0.7825 R^2 skoru elde etmiştir. Bu değer, modelin fiyat varyasyonlarının yaklaşık %78'ini açıklayabildiğini göstermektedir. Model performansının detaylı metrikleri şu şekildedir:

- Ortalama Mutlak Hata (MAE): 247.38
- Kök Ortalama Kare Hata (RMSE): 332.15
- Ortalama Mutlak Yüzde Hata (MAPE): %22.82

Nihai modelin tahmin değerlerinin gerçek değerlere yakınlığı, şekil 4.1'de örnek test verisi kullanılarak grafik üzerinden görselleştirilmiştir. Detaylı model tahmini sonuçları, şekil 4.2'de gösterilmiştir.

Gerçek vs Tahmin Karşılaştırması



Şekil 4.1 Model Tahminlerinin Gerçek Veriler ile Karşılaştırması

ID	Gerçek Değer	Tahmin	Mutlak Fark	Yüzdelik Hata
0	650.00	505.30	144.70	22.26%
1	716.00	964.87	248.87	34.76%
2	1584.00	1512.67	71.33	4.50%
3	1020.00	671.63	348.37	34.15%
4	1749.00	1014.68	734.32	41.99%
5	557.37	627.00	69.63	12.49%
6	999.00	1215.43	216.43	21.66%
7	330.00	382.99	52.99	16.06%
8	2267.86	1821.20	446.66	19.70%
9	682.00	822.52	140.52	20.60%

Şekil 4.2 Tahmin Detayları

Yapılan hiperparametre optimizasyonu ve mimari karşılaştırmaları sonucunda, 3 gizli katmanlı model mimarisi (64, 32, 16 nöron) en iyi performansı göstermiştir. Optimal hiperparametreler şu şekilde belirlenmiştir:

- Öğrenme oranı: 0.01
- Batch size: 32
- Aktivasyon fonksiyonu: Sigmoid
- Epoch sayısı: 100