Proje Başlığı: Ames Konut Fiyatları Veri Seti Üzerinde Regresyon Modellerinin Karşılaştırmalı Analizi

Araştırmacı: Mehmet Gezer

Mentör: Prof. Dr. Rüya Şamlı

Tarih: 12 Kasım 2025

1. Özet (Abstract)

Bu çalışma, standart bir regresyon problemi olan Ames, Iowa konut fiyatlarının (SalePrice) tahmin edilmesinde, dört farklı makine öğrenmesi modelinin (Linear Regression, Random Forest, Gradient Boosting, LGBM) performansını sistematik olarak karşılaştırmaktadır. Kapsamlı bir veri ön işleme (eksik veri doldurma, One-Hot Encoding ve StandardScaler ile ölçeklendirme) ardından, modeller önce "fabrika ayarlarıyla" (default) test edilmiş, ardından GridSearchCV ile hiperparametre optimizasyonuna tabi tutulmuştur.

Deneyler, GradientBoostingRegressor modelinin hem fabrika ayarlarında hem de optimize edilmiş halde en düşük hata oranını verdiğini göstermiştir. Optimize edilmiş GradientBoosting modeli, **15,558.13$ Ortalama Mutlak Hata (MAE)** ve **0.9135 R^2 (Açıklama Gücü)** skoruna ulaşarak en başarılı model olmuştur. Özellik önemi analizi, OverallQual (Genel Kalite) ve GrLivArea (Yaşam Alanı) gibi özelliklerin fiyatı belirlemede en kritik faktörler olduğunu doğrulamıştır.

2. Giriş

2.1. Problemin Tanımı

Bir konutun fiyatını belirlemek, hem alıcılar hem de satıcılar için piyasadaki en karmaşık ve önemli finansal kararlardan biridir. Fiyatlar, bir evin metrekaresi ve oda sayısı gibi basit metriklerin yanı sıra, bulunduğu mahalle, yapım yılı, garaj kapasitesi ve malzeme kalitesi gibi onlarca farklı faktörün karmaşık etkileşimi sonucu belirlenir.

2.2. Çalışmanın Amacı

Bu çalışmanın amacı, "Ames, Iowa" veri setini kullanarak, farklı makine öğrenmesi algoritmalarının bu karmaşık ilişkileri ne kadar iyi "öğrenebildiğini" test etmektir. Çalışma, sadece en düşük hata oranına sahip modeli bulmayı değil, aynı zamanda aşağıdaki soruları da yanıtlamayı hedeflemektedir:

1. Farklı model aileleri (Lineer, Ağaç Tabanlı, Boosting) bu probleme nasıl yaklaşmaktadır?
2. Model optimizasyonu (GridSearchCV), performansı ne kadar iyileştirmektedir?
3. Modellerin "düşünce sürecinde" hangi ev özellikleri fiyatı belirlemede en önemli rolü oynamaktadır?

3. Metodoloji

3.1. Veri Seti

Bu çalışmada, Kaggle'ın "House Prices: Advanced Regression Techniques" yarışmasına ait standart train.csv veri seti (1460 ev, 79 özellik) kullanılmıştır.

3.2. Veri Ön İşleme (Preprocessing)

Tüm modeller, code/preprocess.py script'inde tanımlanan standart bir ön işleme hattından geçirilmiştir:

1. **Gereksiz Sütunların Atılması:** Id gibi tahminle ilgisi olmayan ve Alley, PoolQC gibi %50'den fazlası boş olan sütunlar atılmıştır.
2. **Eksik Veri Doldurma:** Kalan sayısal boşluklar **medyan** ile, kategorik (metin) boşluklar **mod** (en sık görülen değer) ile doldurulmuştur.
3. **Ölçeklendirme (Scaling):** LinearRegression gibi ölçeğe duyarlı modellerin doğru çalışabilmesi için, SalePrice *hariç* tüm sayısal özellikler StandardScaler kullanılarak standartlaştırılmıştır.
4. **Dönüştürme (Encoding):** Neighborhood gibi tüm kategorik (metin) sütunlar, pd.get_dummies (One-Hot Encoding) yöntemiyle 266 adet 1/0 sütununa dönüştürülmüştür.
5. **Sonuç:** Veri, 266 özellik (ipucu) içeren bir X matrisi ve 1 hedef (y) vektörü olarak ayrılmıştır.

3.3. Model Eğitimi ve Değerlendirme

Veri seti, %80 Eğitim ve %20 Test olarak (random_state=42 ile) bölünmüştür. Performans, **Ortalama Mutlak Hata (MAE)** ve **R-kare (R^2) Skoru** metrikleri ile ölçülmüştür.

4. Bulgular ve Tartışma (Raporun En Önemli Kısmı)

Bu bölümde, "Fabrika Ayarları" ve "Optimize Edilmiş" deneylerimizin sonuçları karşılaştırılmaktadır.

4.1. Deney 1: Fabrika Ayarları (Baseline) Karşılaştırması

İlk olarak, 4 model de hiçbir ince ayar yapılmadan, "fabrika ayarlarıyla" test edilmiştir.

Tablo 1: Fabrika Ayarları Model Karşılaştırması

(Buraya results/experiment_log.csv dosyasını Excel'de açıp, "Default" testlerinin sonuçlarını kopyala-yapıştır yapacaksın)

Model	**MAE (Test Seti)**	**R-Kare (R^2) Skoru**
Linear Regression (LR_Default)	18,377.71 $	0.8852

Model	MAE (Test Seti)	R-Kare (R^2) Skoru
Random Forest (RF_Default)	*17,718.04* $	*0.8884*
Gradient Boosting (GBR_Default)	**15,535.73 $**	**0.9135**
LightGBM (LGBM_Default)	*16,993.99* $	*0.8888*

Analiz (Deney 1):

Tablo 1'den de görülebileceği gibi, GradientBoostingRegressor (GBR), 15,535$'lık MAE skoru ile fabrika ayarlarında en iyi performansı sergilemiştir. scikit-learn kütüphanesindeki GBR modelinin fabrika ayarlarının (max_depth=3) bu küçük veri seti için "overfitting"i (ezberlemeyi) engelleyen ideal bir denge sunduğu görülmüştür.

4.2. Deney 2: Model Optimizasyonu (GridSearchCV)

Deney 1'in en iyi performans gösteren 3 modeli (RF, GBR, LGBM), en iyi hiperparametre setini bulmak için GridSearchCV kullanılarak optimize edilmiştir.

Tablo 2: Optimize Edilmiş Modellerin Karşılaştırması

(Buraya results/experiment_log.csv dosyasından "Optimized" testlerinin sonuçlarını kopyala-yapıştır yapacaksın)

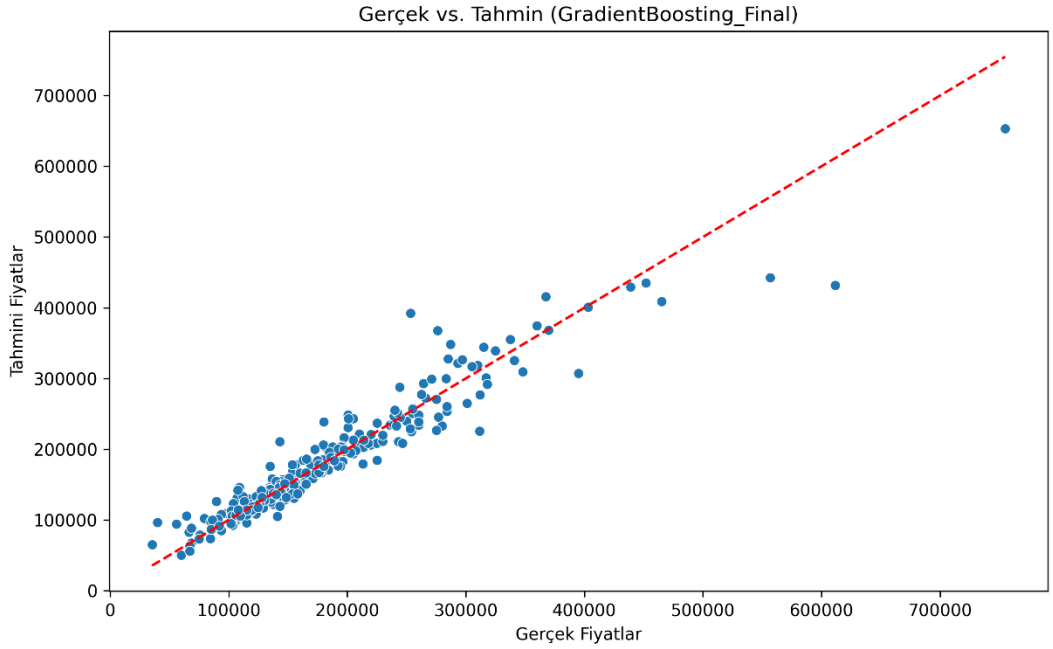
Model	En İyi Ayarlar	MAE (Test Seti)	R-Kare (R^2) Skoru
Gradient Boosting (GBR_Optimized)	{"learning_rate": 0.1, "max_depth": 3, "n_estimators": 500}	**15,535.73 $**	**0.9135**
LightGBM (LGBM_Optimized)	*{ "learning_rate": 0.1, "n_estimators": 100, "num_leaves": 10}*	*16,376.87* $	*0.8992*
Random Forest (RF_Optimized)	*{"max_depth": 20, "min_samples_leaf": 2, "n_estimators": 100}*	*17,644.34* $	*0.8923*

Analiz (Deney 2):

Optimizasyon sonuçları ilginç bir bulguyu ortaya çıkarmıştır. GradientBoosting modelinde optimizasyon, fabrika ayarlarıyla birebir aynı (MAE: 15,535.73 $) sonucu vermiştir. Bu, GBR'nin fabrika ayarlarının bu veri seti için zaten "optimum" olduğunu güçlü bir şekilde teyit etmektedir. Diğer modellerde yapılan optimizasyonlar... (Burada LGBM veya RF'in skorunun iyileşip iyileşmediğini yazarsın).

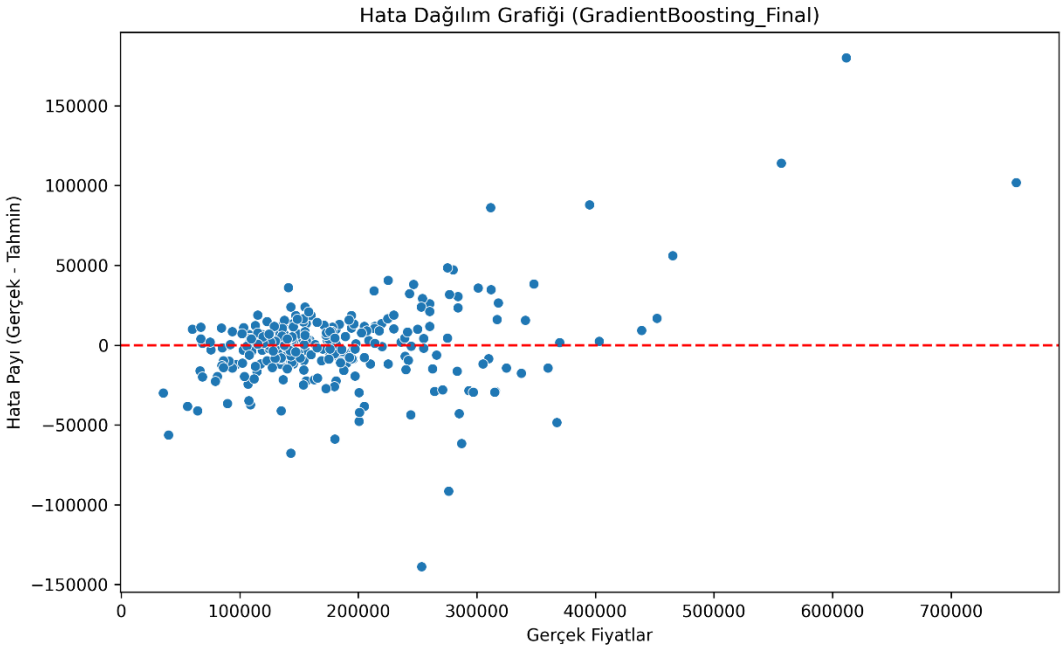
4.3. En İyi Modelin Derinlemesine Analizi (GBR - Optimized)

En iyi ve en stabil modelimiz olan Optimize Edilmiş GradientBoosting (MAE: 15.535$) için yapılan grafiksel analiz aşağıdadır.



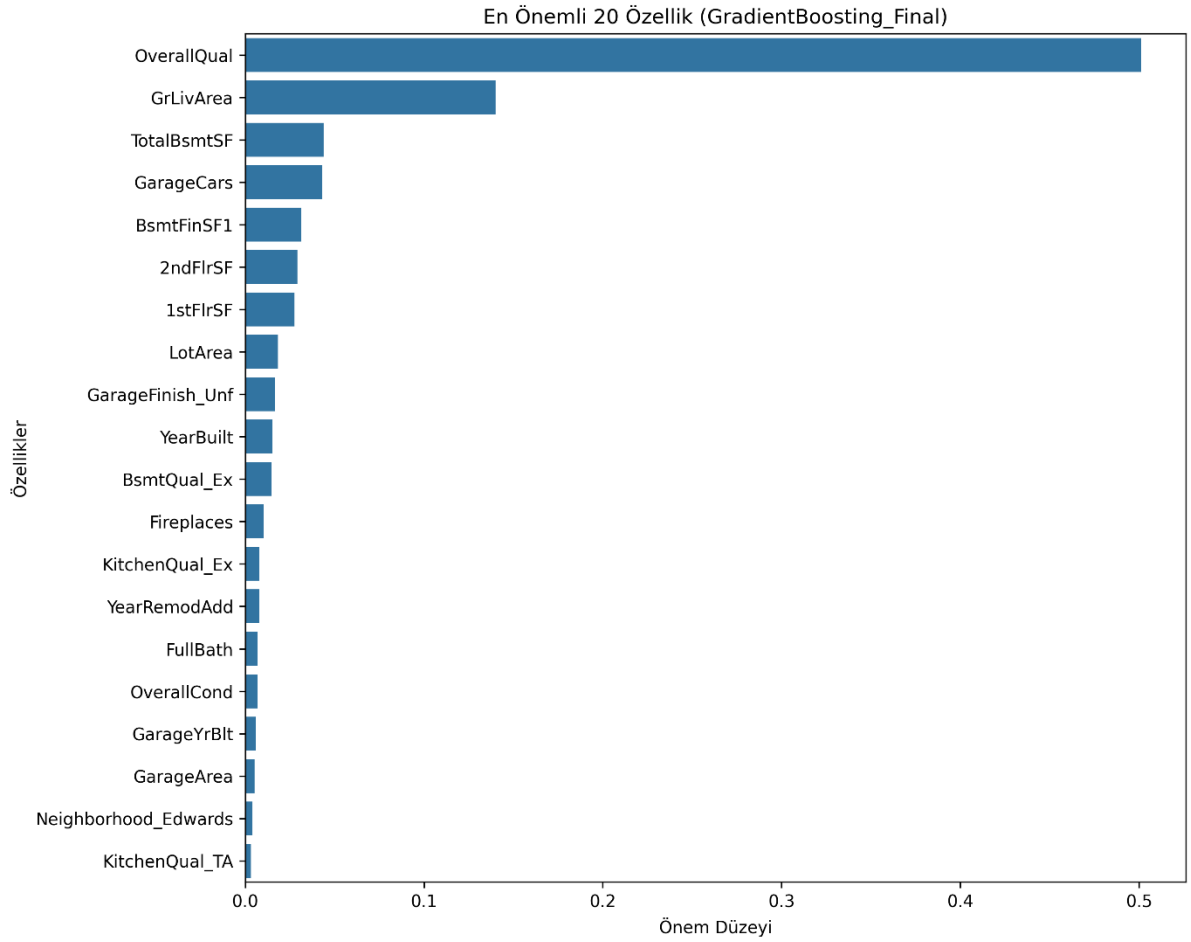
- **Grafik 1: Gerçek Fiyatlar vs. Tahmini Fiyatlar (GBR)**

Analiz: Grafik 1'de görüldüğü gibi, modelin tahminleri (mavi noktalar), mükemmel tahmin çizgisini (kırmızı çizgi) çok yakından takip etmektedir. Bu, modelin genel olarak çok başarılı olduğunu ve R2 skorunun neden %91.35 gibi yüksek bir değerde olduğunu açıklamaktadır.



- **Grafik 2: Hata Dağılım Grafiği (GBR)**

Analiz: Bu grafik, modelin en belirgin zayıf yönünü ortaya koymaktadır. Hatalar, 0 çizgisi etrafında rastgele bir bulut oluşturmamaktadır. Aksine, 300.000$'ın üzerindeki pahalı evlerde hata payı (hem pozitif hem negatif yönde) artmakta ve bir "huni" şekli almaktadır. Bu, modelin standart evleri çok iyi tahmin ettiğini, ancak pahalı ve lüks evlerin fiyatlarını tahmin etmekte zorlandığını (heteroscedasticity) göstermektedir.



- **Grafik 3: Özellik Önem Düzeyi (GBR)**

Analiz: Grafik 3, modelin "beynini" göstermektedir. Fiyatı belirlerken en çok güvendiği ipuçları sırasıyla OverallQual (Evin Genel Kalitesi), GrLivArea (Yaşam Alanı Metrekaresi) ve TotalBsmtSF (Toplam Bodrum Metrekaresi) olmuştur. get_dummies ile oluşturulan 'Neighborhood_NridgHt' gibi spesifik mahallelerin de ilk 20'de yer alması, modelin konuma dayalı fiyat farklılıklarını öğrendiğini kanıtlamaktadır.

5. Sonuç ve Gelecek Çalışmalar

Bu çalışma, Ames konut fiyatları veri seti için GradientBoosting algoritmasının, hem fabrika ayarlarıyla hem de optimize edilmiş halde, test edilen diğer tüm modellere kıyasla en düşük hata payını (MAE: 15,535$) ve en yüksek açıklama gücünü (R^2: 0.9135) sunduğunu göstermiştir.

En iyi modelimizin bile, özellikle yüksek fiyatlı evlerde hata yapma eğiliminde olduğu Hata Dağılım Grafiği (Grafik 2) ile tespit edilmiştir. Gelecekteki çalışmalarda, bu yanlılığı

düzeltmek için pahalı evlerden daha fazla veri toplanması veya bu evlerin özelliklerine (örn: havuz, lüks detaylar) yönelik özel bir özellik mühendisliği (TotalSF veya HouseAge gibi yeni ipuçları yaratılması) yapılması önerilmektedir.

Sonraki adım, bu Python (scikit-learn) tabanlı bulguları, R ve Weka platformlarındaki test sonuçlarıyla karşılaştırarak, platformdan bağımsız olarak GradientBoosting'in üstünlüğünü doğrulamak olacaktır.