Ders 12 Model Seçimi, Optimizasyonu ve Değerlendirilmesi

1- Error Metriğinin Seçilmesi

Amacımız error fonksiyonunu minimize etmek. bu noktada, uygun error metriğini seçmek önemli.

RMSE

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$$

- simetrik. (Gerçek Değer "10" ise, "8" ve "10" tahminlerine aynı davranır)
- underestimation , overestimation durumları için cezalandırma sistemi yok.
- 1000 değerinin 1005 olarak tahmin edilmesiyle 5'in 10 olarak tahmin edilmesini eşit hata payı olarak değerlendirir. (Scale Dependency
- RMSE (Root Mean Squared Error) ve MAE (Mean Absolute Error) gibi metrikler **Mutlak Hata (Absolute Error)** ile ilgilenir, **Bağıl Hata (Relative Error)** ile değil.

RMSLE (Root Mean Squared Logarithmic Error)

$$RMSLE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\log(\hat{y}_i + 1) - \log(y_i + 1))^2}$$

- **Asimetrik:** *underestimation* durumunu (tahminin gerçekten düşük kalması), *overestimation* durumuna göre **daha ağır cezalandırır.**
- **Bağıl Hata (Relative Error):** Mutlak farka değil, orana odaklanır.
- **Scale Invariant:** 100 ile 120 arasındaki hata payı neyse, 1000 ile 1200 arasındaki hata payını da aynı görür (İkisinde de %20 fark vardır).
- **Robust to Outliers:** Büyük sayıların logaritması alındığı için (örn: fiyat tahminlerinde), çok büyük aykırı değerlerin (outliers) model üzerindeki yıkıcı etkisi törpülenir.
- *Not:* Logaritma 0 için tanımsız olduğundan, işlem sırasında değerlere $+1$ eklenir

RMSE vs RMSLE

Özellik	RMSE	RMSLE
Odak Noktası	Mutlak Hata (Absolute Magnitude)	Oransal Hata (Relative/Ratio)
Simetri	**Simetrik.** (Eksik ve fazla tahmini eşit görür)	**Asimetrik.** (eksik tahmini, yani underestimation'ı daha çok cezalandırır)
Hassasiyet	Büyük hatalara (Outliers) çok aşırı tepki verir.	Outlier etkisini sönümler (Robust).
Kullanım Alanı	Genel regresyon problemleri.	Stok tahmini, Emlak fiyatları, Satış adetleri.

Note

- **Gerçek Değer:** 1000
- **Tahmin A:** 600 (Eksik Tahmin / Under Estimation) $\rightarrow$ RMSLE cezası **YÜKSEK** (underestimation)
- **Tahmin B:** 1400 (Fazla Tahmin / Over Estimation) $\rightarrow$ RMSLE cezası DÜŞÜK (RMSE her iki durumda da 400 birimlik fark görüp eşit ceza verirdi.)

2- Model Seçiminde Validation ve Hata Metrikleri

Model seçimi yapılırken, modelin parametrelerini öğrenmesi için ayrılan **Eğitim Kümesi'nin (Train Set)** hepsini doğrudan eğitime tahsis etmeye gerek yoktur; aksine, daha güvenilir bir değerlendirme için bu küme içerisinden **validasyon örneklemi** (bağımsız bir alt küme) ayrılır. Modelin öğrenme sürecinde görmediği bu validasyon örneklemi, farklı model adaylarının veya hiperparametre setlerinin genelleme yeteneğini test etmek amacıyla kullanılır ve her bir model için hesaplanan **hata metrikleri** (MSE, MAE vb.) bu örneklem üzerinde karşılaştırılır. Bu karşılaştırma sonucunda **en düşük hata metriğine sahip olan model**, en uygun aday olarak belirlenir, böylece modelin yalnızca eğitim verisini ezberlemesi (aşırı öğrenme) engellenir ve bağımsız bir veri parçası üzerinde ne kadar iyi performans gösterdiği güvenilir bir şekilde ölçülür.

3- Modeller

Model seçim sürecinde karşılaştırılacak aday modeller, temelde iki farklı strateji kullanılarak oluşturulur:

1. **Farklı Temel Mimari Yapılar:** Modelin temel algoritmasının değiştirilmesi (Örn: **NN**, **DT**, **SVM**, **LR**). Bu, probleme en uygun öğrenme mekanizmasını bulmayı hedefler.
2. **Hiperparametre Optimizasyonu:** Aynı model mimarisi sabit tutularak, performansını yöneten **hiperparametrelerin** (örn. karar ağacının derinliği, L1 L2'de karşılaştığımız λ cezalandırma katsayısı vs.) değiştirilmesi.

Optimal hiperparametre ayarlarını bulmak için **Grid Search**, **Random Search** veya **Bayesian Optimizasyonu**(Optima) gibi sistematik optimizasyon yöntemleri kullanılır.

4- Overfitting - Regularization

- ***Parametrik yapılarda*** L1(Lasso) ve L2(Ridge) regularizasyonlarından yararlanılabilir.

> **Not**
>
> Modeller, veri kümesi hakkında yaptıkları varsayımlara ve yapılarının sabit olup olmamasına göre ikiye ayrılır:
>
> | Kategori | Tanım | Örnekler |
> | --- | --- | --- |
> | **Parametrik** | Modelin yapısı ve öğrenilen **parametre sayısı sabittir** (β katsayıları, ağırlıklar). Veri dağılımı hakkında katı varsayımlar yapar. | **Doğrusal Regresyon, Lojistik Regresyon, Yapay Sinir Ağları (NN)**. |
> | **Parametrik Olmayan** | Modelin yapısı ve potansiyel parametre sayısı **sabit değildir**, veri kümesinin karmaşıklığına bağlı olarak değişir. Veri hakkında esnek varsayımlar yapar. | **Karar Ağaçları (DT), Rastgele Orman, K-En Yakın Komşu (KNN)**. |
>
> Karar Ağacı tabanlı algoritmalar (**Random Forest** dahil) **parametrik değildir**; çünkü modelin yapısı (ağacın dallanma şekli) doğrudan veriden öğrenilir ve sabit bir parametre kümesiyle tanımlanamaz.

- Her iki model de, En Küçük Kareler (OLS) hatasını ve katsayıların büyüklüğünü kontrol eden bir ceza terimini minimize ederek aşırı öğrenmeyi (overfitting) engellemeyi amaçlar.

Lasso-Ridge Regularizasyonda Fayda-Maliyet Dengesi

L1 Regularization

$$\text{Modified loss function} = \text{Loss function} + \lambda \sum_{i=1}^{n} |W_i|$$

L2 Regularization

$$\text{Modified loss function} = \text{Loss function} + \lambda \sum_{i=1}^{n} W_i^2$$

> *Görseldeki $\mathbf{w_1}$ ve $\mathbf{w_2}$ notasyonları ile metinlerde bahsedeceğimiz $\hat{\beta}$ katsayıları aynı parametreleri ifade eder. Bu, yalnızca Makine Öğrenimi (Weight, $\mathbf{w}$) ve Geleneksel Regresyon/İstatistik ($\hat{\beta}$) terminolojilerinin farklılığından kaynaklanan bir gösterim farkıdır.*

$$\text{Linear Regression}: \sum_{i=1}^{N}\left(y_i - \sum_{j=1}^{p} x_{ij}\beta_j\right)^2$$

$$\text{Lasso Regression}: \sum_{i=1}^{N}\left(y_i - \sum_{j=1}^{p} x_{ij}\beta_j\right)^2 + \lambda \sum_{j=1}^{p} |\beta_j|$$

$$\text{Ridge Regression}: \sum_{i=1}^{N}\left(y_i - \sum_{j=1}^{p} x_{ij}\beta_j\right)^2 + \lambda \sum_{j=1}^{p} \beta_j^2$$

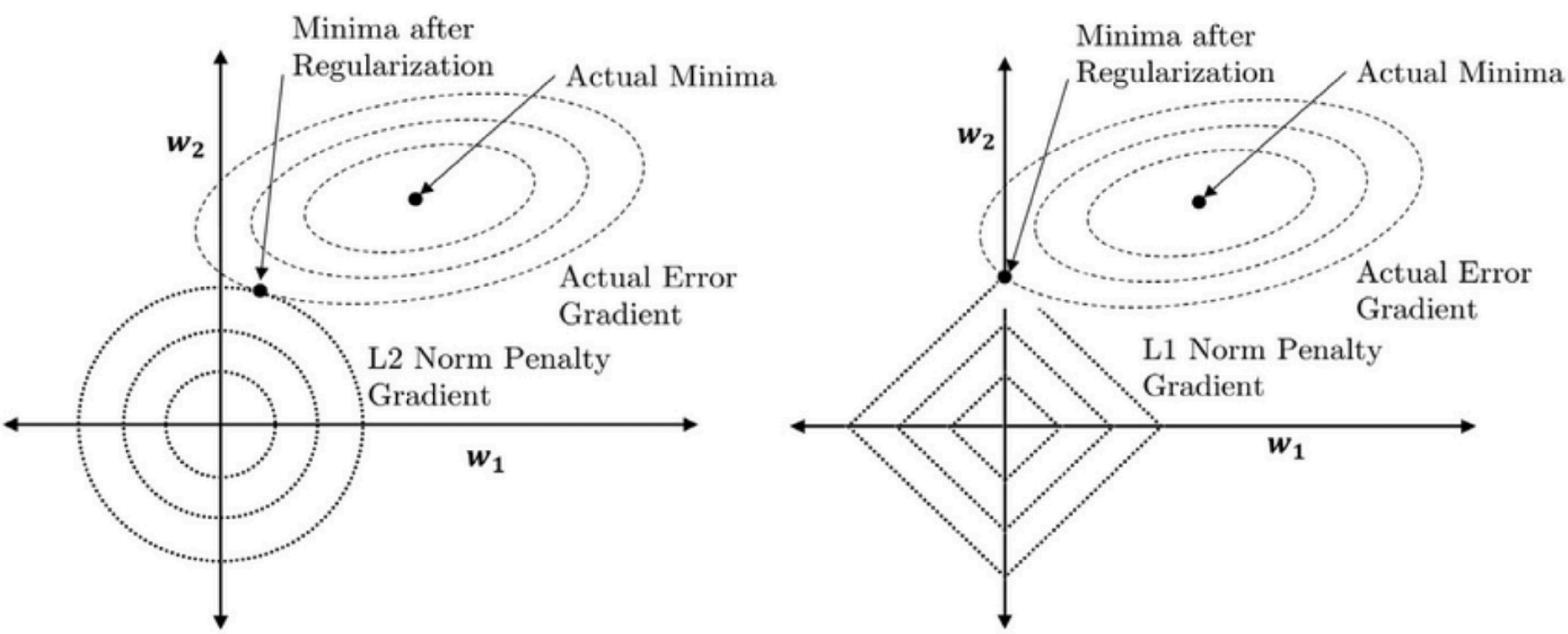
Bir katsayının ($\hat{\beta}$) artırılması (weight artışı) ancak aşağıdaki koşul sağlandığında mantıklıdır ve model iyiye gider:

$$\text{Loss Functiondaki Düşüş (Ödül)} > \text{Ceza Artışı (Maliyet)}$$

Eğer ağırlığı artırmanın getireceği hata azalışı (ödül), o ağırlık artışının ceza teriminde yarattığı artıştan(maliyet) daha büyükse, toplam maliyet $L_{lasso}(\hat{\beta})$ azalır. Bu, modelin doğru yönde ilerlediği anlamına gelir ve o güncelleme kabul edilir.

Aksi takdirde (eğer ağırlığı artırmanın getireceği hata azalışı, ceza terimindeki artışı karşılamıyorsa), toplam maliyet $L_{lasso}(\hat{\beta})$ artar; bu durumda optimizasyon algoritması ağırlığı artırmayı reddeder ve maliyeti azaltmak için o ağırlığı sıfıra doğru küçültür veya tam sıfır yapar (özellikle L1/Lasso'da).

Geometrik Bakış



> Görsel, **Lasso (L1)** ve **Ridge (L2)** regresyonunun katsayılar üzerindeki etkisini, ceza terimlerinin geometrik şekilleri üzerinden açıklar.

- Her iki grafikte de **actual minima çevresindeki elipsler** (Actual Error Gradient), modelin tahmin hatasını gösterir. Bir elips üzerindeki tüm noktalar (fotoğrafta üç elip var), aynı hata(loss) oranını verir. (topografik bir harita gibi, her bir elips bir yüksekliği temsil ediyor.) Merkezden(Actual Minima) uzaklaştıkça hata artar, yaklaştıkça hata azalır.

- **Actual Minima** ise regülarizasyon uygulanmamış çözümün minimum hatasını gösterir.

- Regülarizasyonlu çözüm (**Minima after Regularization**), hata elipslerinin ceza kısıtlama bölgesi ile **ilk kesiştiği** noktadır.

Ridge Regresyonu (Sol: L2 Norm)

- **Ceza Şekli:** L2 norm cezası ($\sum \hat{\beta}_j^2$) **dairesel** bir kısıtlama bölgesi (L2 Norm Penalty Gradient) oluşturur.
- **Özellik:** Dairesel şekil nedeniyle, kesişim noktasının (Minima after Regularization) eksenler üzerinde (x,y ; yani katsayıların sıfır olduğu yerde) olma olasılığı düşüktür.
- **Sonuç:** Ridge, katsayıları **sıfıra yaklaştırır** (küçültür), ancak onları **tam olarak sıfır yapmaz**. Tüm özellikler modelde kalır.

Lasso Regresyonu (Sağ: L1 Norm)

- **Ceza Şekli:** L1 norm cezası ($\sum |\hat{\beta}_j|$) **eşkenar dörtgen (elmas)** şeklinde, **köşeli** bir kısıtlama bölgesi (L1 Norm Penalty Gradient) oluşturur.
- **Özellik:** Bu köşeli yapı, hata elipslerinin elmasa **eksenler üzerindeki köşelerde** dokunma olasılığını artırır. Bu köşelerde katsayılardan en az biri ($\mathbf{w_1}$ veya $\mathbf{w_2}$) **tam olarak sıfırdır**.
- **Sonuç:** Lasso, önemsiz özelliklerin katsayılarını **tamamen sıfıra indirerek** modelden atar; bu da **özellik seçimi (feature selection)** yapar ve **seyrek (sparse)** bir model üretir.

L1 ve L2 Regresyon Özelliklerinin Karşılaştırması

Özellik	Ridge Regresyon (L2 Norm)	Lasso Regresyon (L1 Norm)
Ceza Terimi	Katsayıların karelerinin toplamı: $\sum \hat{\beta}_j^2$	Katsayıların mutlak değerlerinin toplamı: $\sum \|\|\hat{\beta}_j\|\|$
Geometrik Şekil	Dairesel / Küresel	Köşeli / Elmas
Katsayı Etkisi	Sıfıra yaklaştırır (Küçültme)	Tamamen sıfırlar (Elimine etme)
Özellik Seçimi	Yapmaz (Tüm özellikler kalır)	**Yapar** (Seyrek model üretir)
Amaç	Aşırı öğrenmeyi engeller, çoklu doğrusal bağıntıyı (multicollinearity) yönetir.	Aşırı öğrenmeyi engeller, **gereksiz özellikleri eler**.

5- Model Destekli Veri Analizi

Kurmuş olduğumuz model, verimize özel patternleri yakalıyor; ona özel çıkarımlar yapıyor. Modelin verdiği bu çıktıları veri analizinde tekrar kullanarak verimiz hakkında yeni içgörüler elde edebiliriz.

Feature Importance

Modelin hangi değişkenlere daha fazla ağırlık verdiğini inceleyerek, hedef değişkeni (target) etkileyen en kritik faktörleri belirleyebiliriz.

- **Örnek:** "Müşteri terkini (churn) en çok etkileyen faktör fiyat mı yoksa hizmet kalitesi mi?" sorusunun cevabını modelin katsayılarından veya önem skorlarından (örn: Random Forest `feature_importances_`) öğrenebiliriz.

6- Partial Dependence

Elde ettiğimiz tahmin sonucu üzerinden, verimizdeki parametreleri değiştirerek durumu nasıl istediğimiz hale getirebileceğimizi simüle ettiğimiz kavram.

Özellikle, random forrest veya gradient boosting gibi karmaşık "kara kutu" modellerin yorumlanabilirliğini artıran istatistiksel bir yöntemdir. Seçilen bir özelliğin model tahmini üzerindeki marjinal etkisini izole eder. Diğer değişkenlerin etkileri ortalamaya dahil edilerek, ilgilenilen değişkenin hedef üzerindeki etkisi (yönü ve yapısı) netleştirilir.

Hesaplama sürecinde, ilgilenilen özellik belirli bir değere sabitlenirken diğer tüm özelliklerin orijinal değerleri korunur. Modelin bu manipüle edilmiş veri seti üzerindeki tahminlerinin ortalaması alınır. Bu işlem tüm değer aralığı boyunca tekrarlanarak, diğer değişkenlerin gürültüsünden arındırılmış ortalama etki elde edilir.

Sınırlılıklar açısından grafikler, modelin alan bilgisiyle uyumunu test etmek (sanity check) için kritiktir. Ancak özellikler arasında yüksek korelasyon varsa, gerçek dışı veri kombinasyonları oluşabilir ve bu da yanıltıcı sonuçlara yol açabilir. Ayrıca PDP, sadece genel ortalama etkiyi gösterdiğinden, alt gruplardaki farklılaşan etkileri gizleyebilir.