# Makine Öğrenmesi Performanslarının Arttırılması için İpuçları

## Makine öğrenmesi nedir?

- Makine öğrenmesinin en önemli kullanım amacı tahminlemedir. Tahminlemede hız, doğruluk oranı, bilgisayım maliyetleri önem arz eder.

Performansı arttırma yöntemleri temelde 4 ana gruba indirgenebilir.

1) Data ile performansı arttırma
2) Kullanılan modeller (algoritmalar) ile performansı arttırma
3) Algoritma parametreleri ile performans arttırma
4) Ensemble'lar ile performans arttırma

Genelde listeden aşağı inildikçe performans artışındaki oranlar azalma eğiliminde olacaktır. (Listeleme ordinal olarak yapılmıştır)

## 1) Data ile Performansın arttırılması

Eğitim setinde ve problemin tanımlanmasında yapılacak değişiklikler diğer performans arttırma yöntemlerine kıyasla en yüksek faydayı elde edebileceğimiz yöntemdir.

Temel stratejimiz, altta yatan problemin yapısını daha iyi anlamamızı sağlayacak bir perspektif geliştirmektir.

Taktik seviyede yapılabilecekler,

- Daha fazla veri toplamak : Daha yüksek miktarda ya da daha kaliteli veri toplayabilir miyiz?
  (Bazı algoritmalar veri canavarıdır, veri ile besledikçe performansları artar, özellikle Derin Öğrenmede. Ne kadar çok veri daha iyi bir genelleme ve doğruluk)

- Yeni veri eklemek : Daha fazla verinin bulunamadığı durumlarda, üretebiliyor musun? Varolan veriden devşirme ya da arttırma yapabilir misin ya da daha iyisi olasılıksal model kurabilir misin?

- Veriyi temizlemek : Verinin asıl mesajını netleştirebilmek mümkün mü? Gürültüyü (kayıp, değiştirilmiş, hatalı ya da outlier) iyileştirebilir ya da ortadan kaldırabilir misin kaliteyi yükseltmek için.

- Veride örnekleme : Veri büyüklüğünü ya da dağılımını uygun bir örnekleme ile gerçekleştirebilmek (Temsiliyet) Hızlandırmak için daha küçük bir veri seti mesela?

- Tekrardan çerçeveleme : Çözeceğin problemi uygun forma dönüştürme. (Regresyon, binary sınıflandırma, zaman serisi, anormallik tespiti, rating, öneri sistemi)

- Tekrardan ölçekleme : Girdilerin normalizasyon ve standartizasyonu ağırlıklandırılmış ve mesafeye dayalı hesaplamalarda performans artışına yardımcı olur. (Bu algoritmaları yaz)

- Veriyi dönüştürme :

- Boyut indirgeme : Sıkıştırılmış bir hale getirebilir misin temsiliyetini etkilemeden.

- Değişken seçimi : Bütün değişkenler eşit ölçüde mi önemli? Değişken seçimi ve önem methodlarını kullanmaya çalış.

- Değişken mühendisliği : Yeni değişken ekleyebiliyor musun?
  Yeni değişkenlerin içinde ayrıştırılabilecek ve toplanabilecek kategoriler? Tarihler, metinler,

## 2) Algoritmalarla Performansın İyileştirilmesi

Temel stratejimiz baz performansın üzerinde olabilecek ve ortalamanın üzerinde yer alan ilgilendiğimiz probleme uygun algoritmalar bulmaktır.

## Kullanılabilecek Taktikler;

- Resampling Metodu : Hangi metod veya düzenlemenin eldeki veride en iyi sonucu vereceğini bilmek.
  Cross- validation ve hold out validation en iyisi oluyor geneldeç

- Değerlendirme Metrikleri : Probleme ve çalışılan alanın gereklerine en uygun metriklerin kararlaştırılması.

- Baz performans : Algoritmaların performansının karşılaştırılması için baz alınacak performans seviyesi nedir?

*Random algoritma belirle ya da zero rule algorithm kullanarak rank belirle aralarında.

- Lineer Algoritmaları kontrol et : Hangi lineer algoritma iyi çalışacak?
  Lineer metodlar daha kolay ve hızlı eğitilebilirler ve anlaşılabilirler ama sonuçları tartışmalıdır.
  Lineer metodları kullandığında iyi sonuç almak istiyorsan karşılaştırma yap mutlaka içlerinde.

- Non-lineer algoritmaları kontrol et : Hangi non-lineer algoritmalar iyi çalışacak.
  Daha fazla veriye ihtiyaç duyarlar, daha komplekstir ve daha iyi sonuç verirler.

- Literatür Taraması : Uğraştığın problemde hangi algoritmalar daha önce kullanılmış ve
  literatürde bahsediliyor? Edindiğin bilgileri kullanabilirsin.

- Standart Düzenlemeler : Parametre seçiminden önce standart algoritmaları denemek ve araştırmak.

## 3) Algoritma Tuning ile Performans Arttırma

Genel stratejimiz en iyi performans verecek algoritmayı bulmak ve istifade etmektir.

Kullanılabilecek Taktikler;

- Keşifsel analiz :

Kullandığımız metod problemi ezberliyor mu ya da yeteri seviyede öğrenebiliyor mu? (overfitting or underfitting) Farklı algoritmalar farklı sonuçlar ve keşifsel veri analizi verebilir. Algoritmanın neyi doğru neyi yanlış verdiğini gözden geçir.

- Sezgiler :

Sezgilerin algoritma hakkında ne söylüyor?

Parametreler üzerinde yeterince zaman harcadıysan algoritma üzerinde düzenleme yapabileceğin bir sezgi oluşturabilirsin ve daha büyük veri setlerinde de işe yarayabilir bu durum.

- Literatürden faydalanmak :

Literatürde hangi parametreler ya da parametre aralıkları kullanılıyor?

Standart parametrelerin performanslarının değerlendirilmesi önemli bir destek olabilir çoğunlukla.

- Rastgele seçim :

Algoritmada yapılabilecek yeni düzenlemeler için parametrelerin rastgele seçimi de performans arttırmada etkili olabilir.

- Grid search :

İyi düzenlemeler yapabileceğin standart parametre grid’leri mümkün mü?

Sonrasında süreci daha iyi grid’lerle tekrar etmek.

- Optimizasyon :

Optimize edebileceğin parametreler nelerdir?

Belki de öğrenme oranı gibi doğrudan arama prosedürü ya da stokastik optimizasyon ile iyileştirilebilecek parametreler vardır.

- Uygulamaların Çeşitlendirilmesi :

Algoritmaların kullanımı uygun başka alternatifi var mı?

Aynı veri üzerinde alternatif uygulamalar daha iyi sonuç verebilir ve kontrol edilmesi önemli performans açısından.

Her algoritmada uygulayıcının alması gereken sayısız miktarda küçük karar vardır ve bu küçük kararlar problem çözümünü ve performansı etkileme gücüne sahiptir.

- Algoritma geliştirmeleri :

Kullandığın algoritmaların bilinen geliştirmeleri nelerdir?

Belki de performansı arttırmak için algoritmaya özgü standart geliştirme metodlarını uygulayabilirsin.

Uygulama (deneme-yanılma yapman gerekebilir)

- Algoritma Uyarlaması

Üzerinde çalıştığın problem özelinde algoritmada herhangi bir özelleştirme yapabilir misin?

Belki de verisetin, kayıp veriye bakış açın ve ne yapacağın, problemi hedef ve odaklayan algoritma içi bir özelleştirme yapabilir misin?

- Uzmanlarla iletişime geçmek (bir bilene danışmak) :

Uzmanlar tarafından önerilen çözümler nelerdir? Akademisyenler veya belli alanda çalışan birisinden destek alabilir misin?

Ufacık bir bilgi bile vizyonunu arttırabilir.

**Çıktı** :

Bu sürecin sonunda elinde probleme uygulayabileceğin yüksek seviyede tune edilmiş modeller olacak.

Bundan sonra daha da ileri gitmek istersen modellerin bir kombinasyonunu deneyebilirsin.

4) Performansı modellerin bir kombinasyonu ile arttırmak

“Yeteri seviyede iyi” performans gösteren algoritmaların kombinasyonlarını kullanmak yüksek seviyede tune edilmiş algoritmalardan daha iyi sonuçlar vermektedir genellikle.

Yüksek seviyede tune edilmiş algoritmalar daha kırılgan olabilmektedir.

(Gerçek hayatta da problemler her zaman aynı homojenlikte değildir parametreler ve içerdikleri challenge'lar açısından bu yüzden daha ortalama çözümler daha yüksek performanslar verebilirler)

Strateji : İyi performans veren çok sayıda performansı kombine etmektir.

Kombinasyon taktikleri;

- Model tahminlemelerinin karışımı :

Tahminlemeleri bir çok modelden alıp kombinleyebiliyor musun?

Kullandığın aynı ya da farklı algoritmalardan oluşturduğun modellerin sonuçlarının ortalaması ya da medyanını alabilir misin?

- Farklı verisetlerinde eğitilen modellerin karışımı :

Veri setinin farklı bölümlerinde eğitilmiş ve iyi sonuçlar vermiş modellerin sonuçlarının karışımını içeren bir yaklaşım sergilemek.

Bootstrap aggregation , torbalama,

- Veri temsillerinin karışımı
- Farklı veri örneklemlerinde eğitilmiş modellerin sonucunda elde edilmiş tahminlerden kombin yapabiliyor musun?

- Tahminlerin düzeltilmesi :

Tahminleri iyileştirilmesi sürecini açık bir şekilde yapmak veya boosting metodunu kullanarak tahmin hatalarını nasıl iyileştireceğini denemek.

- Kombinleme yapmayı öğrenmek

İyi çalışan birkaç modeli en optimal sonucu verecek şekilde kombinlemeyi öğrenebilir misin?

Stacked generalization :

Alt modellerin farklı parametreler bazında birbirinden farklı ve iyi olabildiği aggregator modelin tahminlerin basit lineer ağırlıklandırıldığı durumlar için geçerlidir.

Bu süreç birkaç katman özelinde de tekrar edilebilir.

Bagging'den bahset

Çıktı :

Bu sürecin sonunda herhangi bir modelden daha yüksek performans verebilecek modeller kombinasyonu elde edebiliriz.

Sonuç :

**Performans İyileştirme Algoritması** :

Performans iyileştirmek için ilk 4 gruptan biri ile başla.

Gruplardaki metodlardan birini seç.

Seçtiğin metoda alternatif başka bir metodu uygula.

Sonuçları karşılaştır, iyileşme varsa o metodu kullan.

Tekrarla.