SLAYT NOTLARI TÜRKÇE

1.HAFTA

Makine Öğrenimi Nedir?

Makine Öğrenimi, bilgisayarları verilerden öğrenebilmeleri için programlama bilimidir (ve sanatıdır).

Daha genel bir tanım:

[Makine Öğrenimi] bilgisayarlara açıkça programlanmadan öğrenme yeteneği veren çalışma alanıdır.
—Arthur Samuel, 1959

Mühendislik odaklı bir tanım:

Bir bilgisayar programının, P ile ölçülen T üzerindeki performansı, E deneyimi ile gelişiyorsa, bazı T görevlerine ve bazı performans ölçütlerine P ilişkin olarak E deneyiminden öğrendiği söylenir.

—Tom Mitchell, 1997

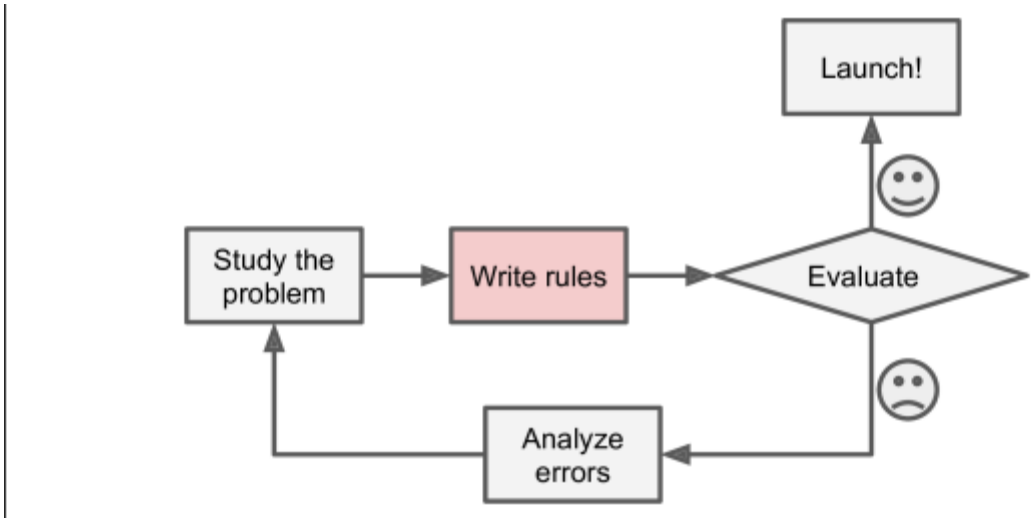
- Örneğin, spam filtreniz, spam e-posta örnekleri (ör. kullanıcılar tarafından işaretlenmiş) ve normal (spam olmayan, "ham" olarak da adlandırılan) e-posta örnekleri verilen spam'i işaretlemeyi öğrenebilen bir Machine Learning programıdır.
- Sistemin öğrenmek için kullandığı örneklere eğitim seti denir. Her eğitim örneğine eğitim örneği (veya örneği) adı verilir.
- Bu durumda, T görevi yeni e-postalar için istenmeyen e-postaları işaretlemektir, E deneyimi eğitim verileridir ve performans ölçüsü P'nin tanımlanması gerekir; örneğin, doğru sınıflandırılmış e-postaların oranını kullanabilirsiniz.
- Bu özel performans ölçüsüne doğruluk denir ve genellikle sınıflandırma görevlerinde kullanılır.

Neden Makine Öğrenimi Kullanılmalı?-Spam Filter

1.İlk önce spam'in tipik olarak nasıl göründüğüne bakarsınız. Bazı kelimelerin veya ifadelerin (“4U”, “kredi kartı”, “ücretsiz” ve “inanılmaz” gibi) konuyla ilgili çok fazla ortaya çıkma eğiliminde olduğunu fark edebilirsiniz. Belki gönderenin adında, e-postanın gövdesinde vb. birkaç başka kalıp daha fark edeceksiniz.

2. Fark ettiğiniz kalıpların her biri için bir algılama algoritması yazarsınız ve bu kalıplardan birkaçı algılanırsa programınız e-postaları spam olarak işaretler.

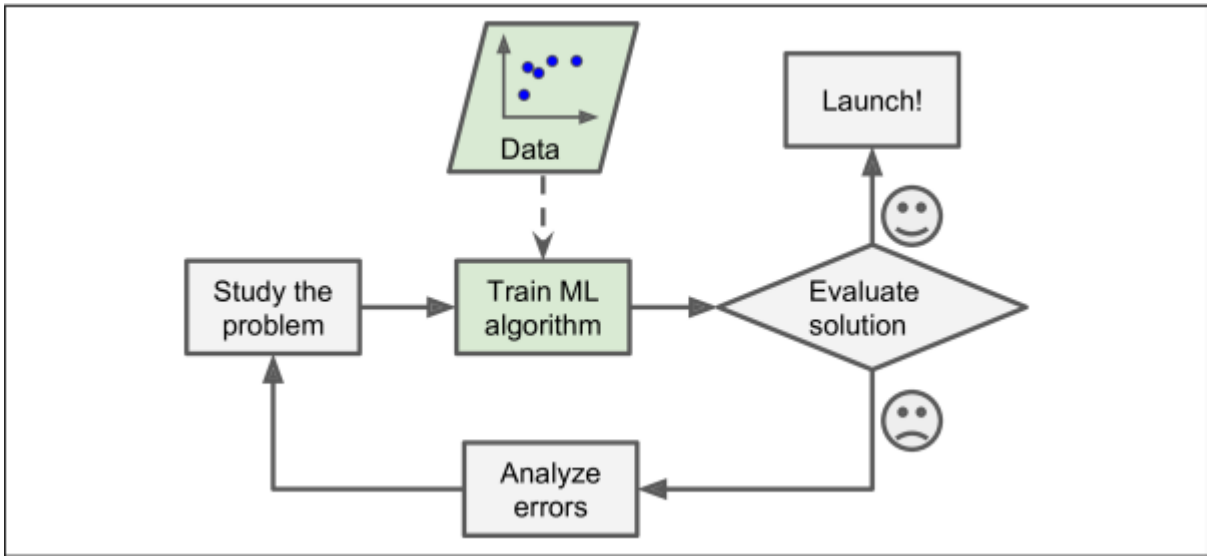
3. Programınızı test eder ve yeterince iyi olana kadar 1. ve 2. adımları tekrarlarsınız.



Şekil 1-1. geleneksel yaklaşım

Sorun önemsiz olmadığından, programınız büyük olasılıkla uzun bir karmaşık kurallar listesi haline gelecektir - bakımı oldukça zordur.

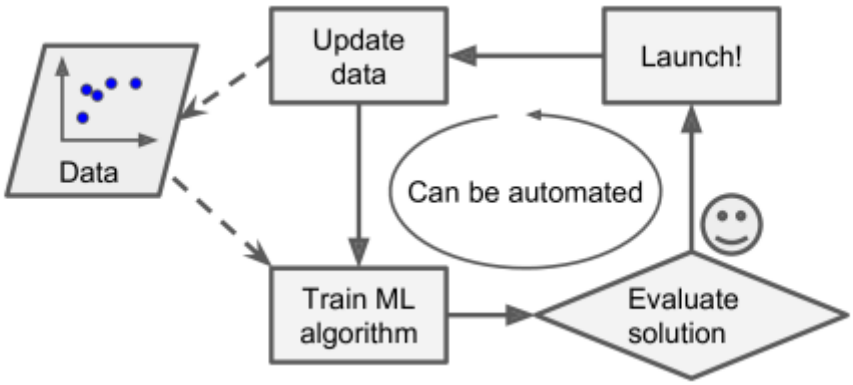
Buna karşılık, Makine Öğrenimi tekniklerine dayalı bir istenmeyen e-posta filtresi, istenmeyen posta örneklerinde jambon örneklerine kıyasla alışılmadık sıklıkta sözcük kalıplarını algılayarak hangi sözcük ve ifadelerin istenmeyen postanın iyi tahmin edicileri olduğunu otomatik olarak öğrenir (Şekil 1-2). Program çok daha kısa, bakımı daha kolay ve büyük olasılıkla daha doğru.



Şekil 1-2. Makine Öğrenimi yaklaşımı

Spam gönderenler, "4U" içeren tüm e-postalarının engellendiğini fark ederse, bunun yerine "U İçin" yazmaya başlayabilirler. Geleneksel programlama tekniklerini kullanan bir spam filtresinin "U İçin" e-postalarını işaretlemek için güncellenmesi gerekir. İstenmeyen e-posta gönderenler, istenmeyen e-posta filtrenizin etrafından dolaşmaya devam ederse, sonsuza kadar yeni kurallar yazmaya devam etmeniz gerekecektir.

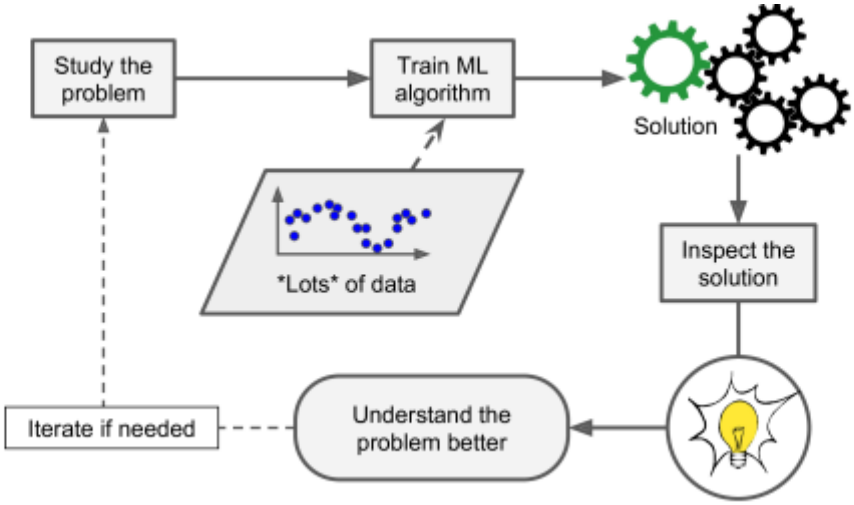
Buna karşılık, Makine Öğrenimi tekniklerine dayalı bir spam filtresi, kullanıcılar tarafından işaretlenen istenmeyen e-postalarda “U için” ifadesinin alışılmadık derecede sıklaştığını otomatik olarak fark eder ve sizin müdahaleniz olmadan onları işaretlemeye başlar **(Şekil 1-3).**



Şekil 1-3. Değişime otomatik olarak uyum sağlama

Son olarak, Makine Öğrenimi insanların öğrenmesine yardımcı olabilir (Şekil 1-4): Makine öğrenimi algoritmaları ne öğrendiklerini görmek için incelenebilir (ancak bazı algoritmalar için bu zor olabilir). Örneğin, spam filtresi yeterince spam konusunda eğitildikten sonra, spam'in en iyi tahmin edicileri olduğuna inandığı kelimelerin ve kelime kombinasyonlarının listesini ortaya çıkarmak için kolayca incelenebilir. Bazen bu, beklenmedik ilişkileri veya yeni eğilimleri ortaya çıkaracak ve böylece sorunun daha iyi anlaşılmasına yol açacaktır.

Büyük miktarda veriyi kazmak için makine öğrenimi tekniklerini uygulamak, hemen belirgin olmayan kalıpları keşfetmeye yardımcı olabilir. Buna veri madenciliği denir.



Şekil 1-4. Makine Öğrenimi, insanların öğrenmesine yardımcı olabilir

Özetlemek gerekirse, Makine Öğrenimi aşağıdakiler için harikadır:

• Mevcut çözümleri çok fazla elle ayarlama veya uzun kural listeleri gerektiren problemler: Tek bir Makine Öğrenimi algoritması genellikle kodu basitleştirebilir ve daha iyi performans gösterebilir.

• Geleneksel bir yaklaşım kullanarak hiçbir şekilde iyi bir çözümü olmayan karmaşık problemler: en iyi Makine Öğrenimi teknikleri bir çözüm bulabilir.

• Değişken ortamlar: Bir Makine Öğrenimi sistemi yeni verilere uyum sağlayabilir.

• Karmaşık sorunlar ve büyük miktarda veri hakkında bilgi edinme.

Makine Öğrenimi Sistemlerinin Türleri

- İnsan denetimiyle eğitilip eğitilmediklerine göre **(denetimli, denetimsiz, yarı denetimli ve Takviyeli Öğrenme**)
- Anında aşamalı olarak öğrenip öğrenemeyecekleri (çevrimiçi ve toplu öğrenme)
- İster yeni veri noktalarını bilinen veri noktalarıyla karşılaştırarak çalışsınlar, ister bunun yerine eğitim verilerindeki kalıpları tespit edip tahmine dayalı bir model oluştursunlar, tıpkı bilim adamlarının yaptığı gibi **(örnek tabanlı ve model tabanlı öğrenme**)

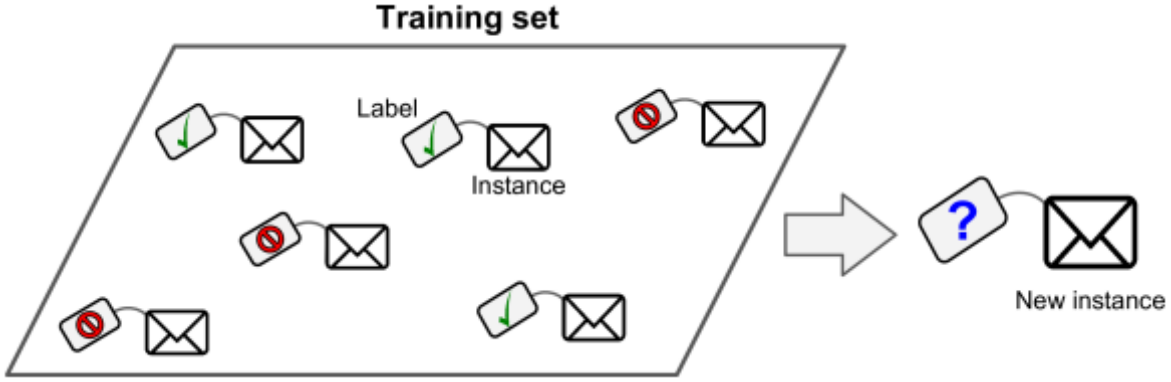
Supervised/Unsupervised Learning(Denetimli/Denetimsiz Öğrenme)

Dört ana kategori vardır:

1. Denetimli öğrenme
2. Denetimsiz öğrenme
3. Yarı denetimli öğrenme
4. Takviye Öğrenimi.

Denetimli öğrenme

Denetimli öğrenmede, algoritmaya beslediğiniz eğitim verileri, etiket adı verilen istenen çözümleri içerir **(Şekil 1-5).**



Şekil 1-5. Denetimli öğrenme için etiketlenmiş bir eğitim seti (ör. spam sınıflandırması)

Tipik bir denetimli öğrenme görevi **sınıflandırmadır**. İstenmeyen e-posta filtresi buna iyi bir örnektir: sınıflarıyla birlikte (spam veya jambon) birçok örnek e-posta ile eğitilmiştir ve yeni e-postaları nasıl sınıflandıracağını öğrenmelidir.

Diğer bir tipik görev, tahmin edici olarak adlandırılan bir dizi özellik (kilometre, yaş, marka vb.) veriliyken, bir arabanın fiyatı gibi bir hedef sayısal değeri tahmin etmektir. Bu tür bir göreve regresyon denir **(Şekil 1-6).**

Figure 1-6. Regression

en önemli denetimli öğrenme algoritmalarından bazıları

- **k-En Yakın Komşular**
- **Doğrusal Regresyon**
- **Lojistik regresyon**
- **Destek Vektör Makineleri (SVM'ler)**
- **Karar Ağaçları ve Rastgele Ormanlar**
- **Nöral ağlar**

Denetimsiz öğrenme

Denetimsiz öğrenmede tahmin edebileceğiniz gibi eğitim verileri etiketsizdir**.** Sistem öğretmensiz öğrenmeye çalışır.

en önemli denetimsiz öğrenme algoritmalarından bazıları:

- **Kümeleme**
  - K-Means
  - DBSCAN
  - Hierarchical Cluster Analysis (HCA)
- **Anormallik tespiti ve yenilik tespiti**
  - One-class SVM
  - Isolation Forest
- **Görselleştirme ve boyut azaltma**
  - **Temel Bileşen Analizi (PCA)**
  - **Çekirdek PCA**
  - **Yerel Doğrusal Gömme (LLE)**
  - **t-dağıtılmış Stokastik Komşu Gömme (t-SNE)**
- **Birliktelik kuralı öğrenme**
  - **Apriori**
  - **Eclat**

Yarı Denetimli öğrenme

Bazı algoritmalar, kısmen etiketlenmiş eğitim verileriyle, genellikle çok sayıda etiketlenmemiş veri ve biraz etiketlenmiş veri ile ilgilenebilir. Buna yarı denetimli öğrenme denir (Şekil 1-11).

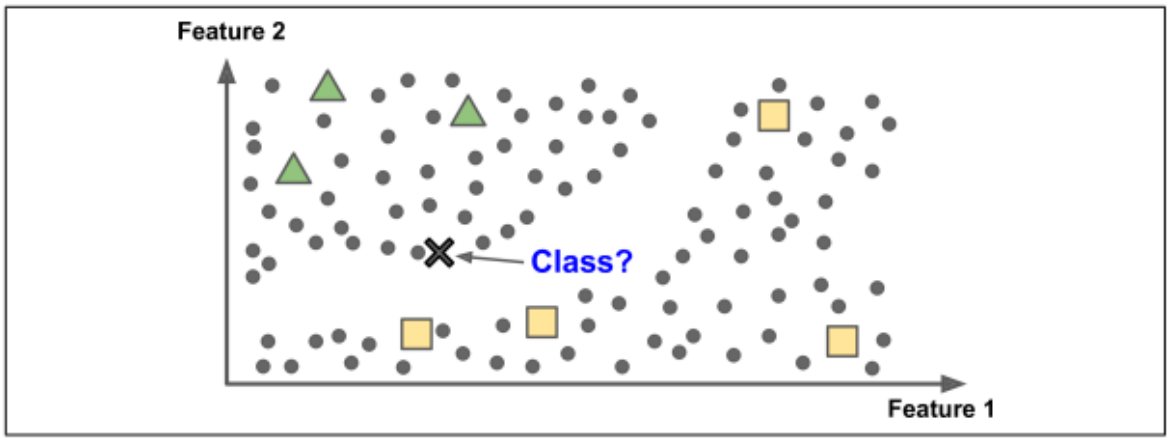


Figure 1-11. Semisupervised learning

Yarı denetimli öğrenme algoritmaları, denetimsiz ve denetimli algoritmaların kombinasyonlarıdır.

Pekiştirmeli Öğrenme

Pekiştirmeli Öğrenme çok farklı bir canavardır.

Bu bağlamda aracı olarak adlandırılan öğrenme sistemi, çevreyi gözlemleyebilir, eylemleri seçip gerçekleştirebilir ve karşılığında ödüller (veya Şekil 1-12'deki gibi olumsuz ödüller şeklinde cezalar) alabilir.

Daha sonra, zaman içinde en fazla ödülü almak için politika adı verilen en iyi stratejinin ne olduğunu kendi kendine öğrenmelidir.

Toplu ve Çevrimiçi Öğrenme

Toplu öğrenme Toplu öğrenmede, sistem aşamalı olarak öğrenemez: mevcut tüm veriler kullanılarak eğitilmelidir.

Bu genellikle çok fazla zaman ve bilgi işlem kaynağı alacaktır, bu nedenle genellikle çevrimdışı yapılır. Önce sistem eğitilir, ardından üretime alınır ve artık öğrenmeden çalışır; sadece öğrendiklerini uygular. **Buna çevrimdışı öğrenme denir**

Çevrimiçi Öğrenme

Çevrimiçi öğrenmede, tek tek veya mini gruplar adı verilen küçük gruplar halinde, sırayla veri örneklerini besleyerek sistemi aşamalı olarak eğitirsiniz. Her öğrenme adımı hızlı ve ucuzdur, böylece sistem yeni veriler hakkında anında bilgi edinebilir.

Çevrimiçi öğrenme, verileri sürekli bir akış (örneğin hisse senedi fiyatları) olarak alan ve hızlı veya bağımsız bir şekilde değişime uyum sağlaması gereken sistemler için harikadır.

Çevrimiçi öğrenme sistemlerinin önemli bir parametresi, değişen verilere ne kadar hızlı adapte olmaları gerektiğidir: buna **öğrenme oranı** denir. Yüksek bir öğrenme oranı belirlerseniz, sisteminiz yeni verilere hızla adapte olur, ancak aynı zamanda eski verileri hızla unutma eğiliminde olur.

Düşük bir öğrenme oranı ayarlarsanız, sistem daha fazla atalete sahip olacaktır; yani, daha yavaş öğrenecek, ancak yeni verilerdeki gürültüye veya temsili olmayan veri noktaları (aykırı değerler) dizilerine daha az duyarlı olacaktır.

Örnek Tabanlı ve Model Tabanlı Öğrenme

Örneğe dayalı öğrenme: sistem örnekleri ezbere öğrenir, ardından bir benzerlik ölçüsü kullanarak öğrenilen örneklerle (veya bunların bir alt kümesiyle) karşılaştırarak yeni durumlara genelleme yapar.

Model tabanlı öğrenme: Bir dizi örnekten genelleme yapmak, bu örneklerin bir modelini oluşturmak ve ardından bu modeli tahminlerde bulunmak için kullanılır.

2.HAFTA

Makine Öğreniminin Zorlukları

Zorluklar

Ana görev, bir öğrenme algoritması seçmek ve onu bazı veriler üzerinde eğitmektir**.**

- Kötü algoritma
- Kötü veri

Kötü Veri

Yetersiz Eğitim Verisi Miktarı

Yeni yürümeye başlayan bir çocuğun elmanın ne olduğunu öğrenmesi için, bir elmayı işaret etmeniz ve “elma” demeniz yeterlidir (muhtemelen bu işlemi birkaç kez tekrarlamak). Artık çocuk her çeşit renk ve şekildeki elmaları tanıyabilir.

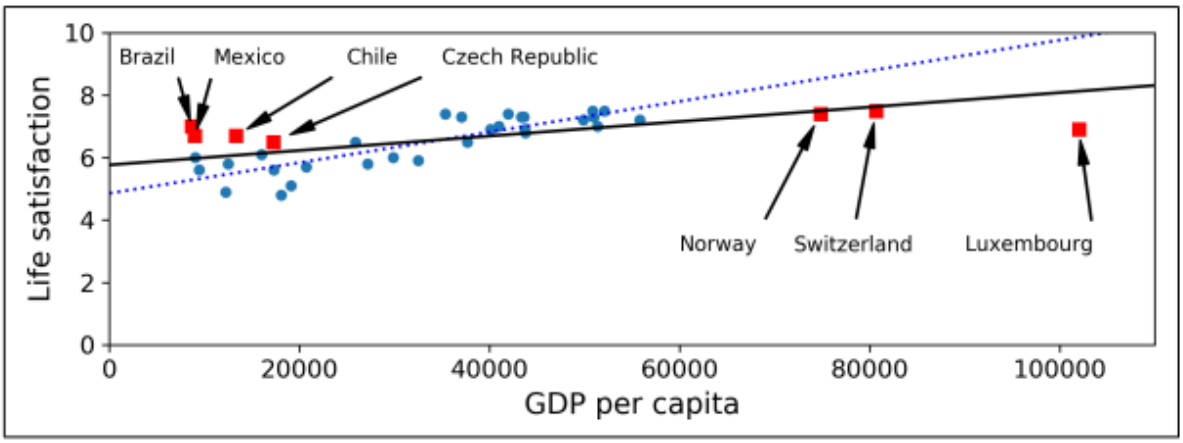
Makine Öğrenimi henüz tam olarak orada değil; çoğu Makine Öğrenimi algoritmasının düzgün çalışması için çok fazla veri gerekir. Çok basit problemler için bile tipik olarak binlerce örneğe ihtiyacınız vardır ve görüntü veya konuşma tanıma gibi karmaşık problemler için milyonlarca örneğe ihtiyacınız olabilir (mevcut bir modelin parçalarını yeniden kullanamıyorsanız).

Verilerin Mantıksız Etkinliği

Oldukça basit olanlar da dahil olmak üzere farklı Makine Öğrenimi algoritmaları, kendilerine yeterli veri verildiğinde, karmaşık bir doğal dil belirsizliği giderme probleminde neredeyse aynı derecede iyi performans gösterdi.

Temsili Olmayan Eğitim Verileri

İyi bir genelleme yapabilmek için, eğitim verilerinizin genellemek istediğiniz yeni vakaları temsil etmesi çok önemlidir. Örnek tabanlı öğrenme veya model tabanlı öğrenme kullansanız da bu doğrudur.



Şekil 1-21. Daha temsili bir eğitim örneği

Bu veriler üzerinde doğrusal bir model eğitirseniz, düz çizgiyi elde edersiniz, eski model ise noktalı çizgi ile temsil edilir. Gördüğünüz gibi, sadece birkaç eksik ülke eklemek, modeli önemli ölçüde değiştirmekle kalmıyor.

Düşük Kaliteli Veriler

Eğitim verileriniz hatalarla, aykırı değerlerle ve gürültüyle doluysa **(örneğin, düşük kaliteli ölçümler nedeniyle),** sistemin temeldeki kalıpları algılamasını zorlaştıracağından, sisteminizin iyi performans gösterme olasılığı daha düşüktür. Eğitim verilerinizi temizlemek için zaman harcamak genellikle çabaya değer. Gerçek şu ki, çoğu veri bilimcisi zamanlarının önemli bir bölümünü tam da bunu yaparak geçiriyor.

Örneğin:

• Bazı örnekler açıkça aykırı değerlerse, bunları atmak veya hataları manuel olarak düzeltmeye çalışmak yardımcı olabilir.

Bazı durumlarda birkaç özellik eksikse (örneğin, müşterilerinizin %5'i yaşlarını belirtmediyse), bunu isteyip istemediğinize karar vermelisiniz.

- **bu özelliği tamamen görmezden gelin,**
- **bu durumları görmezden gel,**
- **eksik değerleri doldurun**(örneğin, ortanca ile),

veya bir modeli özellik ile ve bir modeli onsuz eğitin

Alakasız Özellikler

Sisteminiz yalnızca, eğitim verilerinin yeterli sayıda ilgili özelliği içermesi ve çok fazla alakasız özelliği içermemesi durumunda öğrenme yeteneğine sahip olacaktır. Bir Makine Öğrenimi projesinin başarısının kritik bir parçası, üzerinde eğitim almak için iyi bir dizi özellik bulmaktır.

Özellik mühendisliği adı verilen bu süreç şunları içerir:

• **Özellik seçimi:** mevcut özellikler arasından üzerinde çalışılacak en kullanışlı özelliklerin seçilmesi.

• **Özellik çıkarma:** daha kullanışlı bir tane üretmek için mevcut özellikleri birleştirmek (daha önce gördüğümüz gibi, boyut azaltma algoritmaları yardımcı olabilir).

• Yeni veriler toplayarak yeni özellikler oluşturma.

Overfitting(Aşırı Öğrenme)

Eğer modelimiz, eğitim için kullandığımız veri setimiz üzerinde gereğinden fazla çalışıp ezber yapmaya başlamışsa ya da eğitim setimiz tek düze ise **overfitting** olma riski büyük demektir. Eğitim setinde yüksek bir skor aldığımız bu modele, test verimizi gösterdiğimizde muhtemelen çok düşük bir skor elde edeceğiz. Çünkü model eğitim setindeki durumları ezberlemiştir ve test veri setinde bu durumları aramaktadır. En ufak bir değişiklikte ezberlenen durumlar bulunamayacağı için test veri setinde çok kötü tahmin skorları elde edebilirsiniz. Overfitting problemi olan modellerde yüksek varyans, düşük bias durumu görülmektedir.

Bu genellikle model çok karmaşık olduğunda (yani gözlem sayısına kıyasla çok fazla özellik / değişken varsa) gerçekleşir.

Underfitting (Eksik Öğrenme)

Aşırı öğrenmenin aksine, bir model yetersiz öğrenmeye sahipse, modelin eğitim verilerine uymadığı ve bu nedenle verilerdeki trendleri kaçırdığı anlamına gelir. Ayrıca modelin yeni veriler için genelleştirilemediği anlamına da gelir. Tahmin ettiğiniz gibi bu problem genellikle çok basit bir modelin sonucudur (yetersiz tahminleyici bağımsız değişken eksikliği).

Underfitting sorunu olan modellerde hem eğitim hem de test veri setinde hata oranı yüksektir. Düşük varyans ve yüksek bias'a sahiptir. Bu modeller eğitim verilerini çok yakından takip etmek yerine, eğitim verilerinden alınan dersleri yok sayar ve girdiler ile çıktılar arasındaki temel ilişkiyi öğrenemez.

Bu sorunu çözmek için ana seçenekler şunlardır:

• Daha fazla parametreli daha güçlü bir model seçmek

• Öğrenme algoritmasına daha iyi özellikler beslemek (özellik mühendisliği)

• Model üzerindeki kısıtlamaları azaltmak (örneğin, düzenlileştirme hiper parametresini azaltmak)