## Finding unusual events

İkinci denetimsiz öğrenme algoritmamıza bakalım. Anormallik algılama algoritmaları, normal olayların etiketlenmemiş bir veri kümesine bakar ve böylece olağandışı veya anormal bir olay olup olmadığını algılamayı veya kırmızı bayrak yükseltmeyi öğrenir. Bir örneğe bakalım. Bazı arkadaşlarım, imal edilmekte olan uçak motorlarındaki olası sorunları tespit etmek için anormallik tespitini kullanma üzerinde çalışıyorlardı. Bir şirket bir uçak motoru yaptığında, o uçak motorunun gerçekten güvenilir olmasını ve iyi çalışmasını istersiniz çünkü bir uçak motoru arızasının çok olumsuz sonuçları vardır. Bu yüzden bazı arkadaşlarım, bir uçak motorunun üretildikten sonra anormal görünüp görünmediğini veya onunla ilgili bir sorun olup olmadığını kontrol etmek için anormallik tespitini kullanıyorlardı.

Buradaki fikir, bir uçak motoru montaj hattından çıktıktan sonra, uçak motorunun bir dizi farklı özelliğini hesaplayabilirsiniz. Diyelim ki x1 özelliği motor tarafından üretilen ısıyı ölçüyor. Özellik x2, ek özellikler için titreşim yoğunluğunu vb. ölçer. Ama slaytı biraz basitleştirmek için, sadece motorun ısısına ve titreşimlerine karşılık gelen iki x1 ve x2 özelliğini kullanacağım.

Şimdi, uçak motoru üreticilerinin o kadar çok kötü motor yapmadığı ortaya çıktı. Bu nedenle, toplanması daha kolay olan veri türü, bu m motorlarının nasıl davrandığına dair x1 ve x2 özelliklerini toplamak için m uçak motorları üretmiş olsaydınız ve muhtemelen çoğu, kusurlu veya akışlı olanlardan ziyade normal motorlardan iyidir. içlerinde.

Ve anormallik saptama sorunu, öğrenme algoritması, uçak motorlarının ne kadar ısı üretildiği ve ne kadar titreştikleri açısından tipik olarak nasıl davrandığına dair bu m örnekleri gördükten sonradır. Yepyeni bir uçak motoru montaj hattından çıkarsa ve Xtest tarafından verilen yeni bir özellik vektörüne sahipse, bu motorun daha önce üretilmiş motorlara benzer olup olmadığını bilmek isteriz. Yani bu muhtemelen iyi mi? Veya bu motorda, bu performansın şüpheli olmasına neden olabilecek gerçekten tuhaf bir şey mi var, yani belki de onu gönderip bir uçağa takılmasına izin vermeden önce daha da dikkatli bir şekilde incelemeliyiz ve umarız hiçbir şey ters gitmez. . Bir anormallik algılama algoritmasının nasıl çalıştığı aşağıda açıklanmıştır. Burada x1'den xm'ye kadar olan örnekleri çizeyim. Eğer bu yeni uçak motoru Xtest montaj hattından çıkarsa ve bu x1 ve x2 değerlerini çizerseniz ve eğer buradaysa, tamam, bu muhtemelen iyi görünüyor dersiniz. Diğer uçak motorlarına çok benziyor. Belki de bunun için endişelenmeme gerek yok.

Ama eğer bu yeni uçak motorunun bir ısı ve titreşim imzası varsa, yani aşağısı kadar, o zaman buradaki veri noktası yukarıda gördüğümüzden çok farklı görünüyor. Ve muhtemelen diyeceğiz oğlum, bu bir anormallik gibi görünüyor. Bu daha önce gördüğüm örneklere benzemiyor, bu motorun uçağa takılmasına izin vermeden önce bunu daha dikkatli incelesek iyi olur. Bu sorunu ele alan bir algoritmaya nasıl sahip olabilirsiniz? Anomali tespitini gerçekleştirmenin en yaygın yolu, yoğunluk tahmini adı verilen bir tekniktir. Bunun anlamı, size bu m örnekten oluşan eğitim kümeleri verildiğinde, yaptığınız ilk şey, x'in olasılığı için bir model oluşturmaktır. Başka bir deyişle, öğrenme algoritması, x1 ve x2 özelliklerinin yüksek olasılığa sahip değerlerinin neler olduğunu ve veri setinde görülme olasılığı daha düşük veya daha düşük veya daha düşük olan değerlerin neler olduğunu bulmaya çalışacaktır. . Elimizdeki bu örnekte, ortadaki o küçük elipsteki örnekleri görmenin oldukça muhtemel olduğunu düşünüyorum, yani ortadaki bölgenin olasılığı yüksek olabilir, belki bu elipsteki şeylerin biraz daha düşük olasılığı vardır. Bu ovalin bu elipsindeki şeylerin olasılığı daha da düşük ve dışarıdaki şeylerin olasılığı daha da düşük. Hangi bölgelerin daha yüksek ve daha düşük olasılığa sahip olduğuna eğitim setinden nasıl karar vereceğinizin ayrıntılarını sonraki birkaç videoda göreceğiz. Ve p(x) için modellemiş veya modellemeyi öğrenmiş olmak size yeni test örneği Xtest verildiğinde. Yapacağınız şey, Xtest olasılığını hesaplamak olacaktır. Ve eğer küçükse veya daha doğrusu epsilon diyeceğim küçük bir sayıdan küçükse, bu bir Yunan alfabesi epsilon'dur. Küçük bir sayı olarak düşünmeniz gereken şey, yani p(x)'in çok küçük olması veya başka bir deyişle, x'in belirli bir kullanıcı için gördüğünüz özgül değerinin, gördüğünüz diğer kullanıcılara göre çok düşük olması. Ancak Xtest'in p'si küçük bir eşikten veya küçük bir epsilon sayısından daha az, bunun bir anormallik olabileceğini söylemek için bir işaret kaldıracağız. Örneğin, Xtest buraya kadar inmişse, bir örneğin buraya kadar inme olasılığı aslında oldukça düşüktür. Ve umarım Xtest'in bu değeri için Xtest'in p'si epsilon'dan küçük olur ve bu yüzden bunu bir anormallik olarak işaretleriz. Oysa p of Xtest epsilon'dan küçük değilse, p of Xtest epsilon'dan büyükse, o zaman tamam görünüyor, bu bir anormallik gibi görünmüyor diyeceğiz. Ve bu yanıt, eğer burada bir örneğiniz varsa, p x modelimizin nerede olduğunu söyleyin, burada ortaya yakın örneklerin aslında oldukça yüksek olasılıklar olduğunu söyleyecektir. Yeni uçak motorunun bu iç elipslere yakın özelliklere sahip olma ihtimali çok yüksek. Ve böylece bu örnekler için p of Xtest büyük olacak ve bunun bir anormallik olmadığını söyleyeceğiz. Anomali tespiti günümüzde birçok uygulamada kullanılmaktadır. Örneğin, birçok farklı özelliğe sahip bir web sitesi çalıştırıyorsanız, dolandırıcılık tespitinde sıklıkla kullanılır. Xi'yi i kullanıcısının etkinliklerinin özellikleri olarak hesaplarsanız. Özellik örnekleri, bu kullanıcının ne sıklıkta oturum açtığını ve kaç web sayfasını ziyaret ettiğini içerebilir. Yazma hızları ne kadara kaç işlem yapıyorlar veya tartışma forumunda kaç gönderi yapıyorlar? Saniyede kaç karakter yazabiliyorlar? Bunun gibi verilerle, belirli bir kullanıcının tipik davranışını modellemek için verilerden p x'i modelleyebilirsiniz. Dolandırıcılık tespitinin yaygın iş akışında, bir hesabı anormal göründüğü için otomatik olarak kapatmazsınız. Ancak bunun yerine, güvenlik ekibinden daha yakından incelemesini veya kullanıcıdan kimliğini bir cep telefonu numarasıyla doğrulamasını istemek veya insan olduklarını kanıtlamak için bir yakalama geçirmesini istemek gibi bazı ek güvenlik kontrolleri yapmasını isteyebilirsiniz. üzerinde. Ancak bunun gibi algoritmalar, olağandışı veya belki biraz şüpheli etkinlikleri bulmaya çalışmak için bugün rutin olarak kullanılmaktadır. Dolandırıcılık olmadığından emin olmak için bu hesapları daha dikkatli bir şekilde tarayabilirsiniz. Ve bu tür bir dolandırıcılık tespiti, hem sahte hesapları bulmak için kullanılır hem de bu tür bir algoritma, çok alışılmadık bir satın alma modeli varmış gibi finansal dolandırıcılığı belirlemeye çalışmak için de sıklıkla kullanılır. O zaman bu, bir güvenlik ekibinin daha dikkatli incelemesine değecek bir şey olabilir. Anormallik tespiti, imalatta da sıklıkla kullanılır. Bir önceki slaytta uçak motoru imalatı ile ilgili bir örnek gördünüz. Ancak birçok kıtadaki birçok üretici, ürettikleri her şeyi görmek için rutin olarak anormallik tespitini kullanıyordu. Bir uçak motorundan baskılı devre kartına, akıllı telefondan motora, bir şekilde garip davranan üniteyi henüz üretip üretmediğinizi görmek için pek çok şeye kadar her şey.

Çünkü bu, uçak motorunuzda veya baskılı devre kartlarınızda bir sorun olduğunu gösterebilir veya sizde o nesneyi müşteriye göndermeden önce daha dikkatli bir şekilde bakmak istemenize neden olabilecek bir şey olabilir. Ayrıca kümelerdeki ve veri merkezlerindeki bilgisayarları izlemek için kullanılır, burada X I, belirli bir makinenin özellikleriyse, bellek kullanıcılarının yakaladığı özellikler, saniyedeki disk erişim sayısı gibi. CPU yükü özellikleri, CPU yükünün ağ trafiğine oranı gibi ırksal da olabilir. O zaman, belirli bir bilgisayar diğer bilgisayarlardan çok farklı davranırsa, bir sorun olup olmadığını görmek için o bilgisayara bir göz atmaya değer olabilir. Örneğin, bir sıcak test hatası veya ağ hackleme hatası olmuş veya onda bir sorun var veya belki de hacklenmiş gibi. Anomali tespiti, insanların bu konuda çok fazla konuştuğunu duymuyor gibi görünseniz de çok yaygın olarak kullanılan algoritmalardan biridir. Anormallik tespitinin ticari uygulaması üzerinde ilk kez çalıştığım zamanı hatırlıyorum, bir telekomünikasyon şirketine, baz istasyonlarından herhangi birinin alışılmadık bir şekilde davrandığını görmek için anormallik tespitini yerleştirmesine yardım ederken. Çünkü bu muhtemelen baz istasyonunda bir sorun olduğu anlamına geliyordu ve bu yüzden daha fazla insanın iyi bir cep telefonu kapsama alanı elde etmesini ummak için bir teknisyene götürmek istiyorlar. Anormallik tespitini hileli finansal işlemleri bulmak için de kullandım ve bu günlerde bunu genellikle imalat şirketlerinin ürettikleri ancak daha sık denetlemeleri gereken anormal parçaları bulmalarına yardımcı olmak için kullanıyorum. Bu yüzden alet sandığınızda olması çok faydalı bir alettir. Ve sonraki birkaç videoda, bu algoritmaları kendi başınıza nasıl oluşturabileceğiniz ve çalıştırabileceğiniz hakkında konuşacağız. Anonim algılama albümlerinin çalışmasını sağlamak için, x'in p verilerini modellemek üzere bir Gauss dağılımı kullanmamız gerekecek. Öyleyse, Gauss dağılımları hakkında konuşmak için bir sonraki videoya geçelim.

Anormallik tespitini gerçekleştirmenin en yaygın yolu, yoğunluk tahmini adı verilen bir tekniktir. Bunun anlamı, size bu m örnekten oluşan eğitim setleriniz verildiğinde, yaptığınız ilk şey, x olasılığı için bir model oluşturmaktır. Başka bir deyişle, öğrenme algoritması, x1 ve x2 özelliklerinin yüksek olasılığa sahip değerlerinin neler olduğunu ve veri kümesinde görülme olasılığı daha düşük veya daha düşük bir olasılık veya daha düşük olasılıklı değerlerin neler olduğunu bulmaya çalışacaktır.

Anonim algılama albümlerinin çalışması için, x verisini modellemek için bir Gauss dağılımı kullanmamız gerekecek. O halde Gauss dağılımları hakkında konuşmak için bir sonraki videoya geçelim.

Ama eğer bu yeni uçak motorunun bir ısı ve titreşim imzası varsa, yani burada ta aşağısında, bu veri noktası bir zamanlar yukarıda gördüğümüzden çok farklı görünüyor. Ve muhtemelen diyeceğiz ki, oğlum, bu bir anormallik gibi görünüyor. bu görünmüyor

##Gaussian (normal) distribution

Anormallik tespitini uygulamak için normal dağılım olarak da adlandırılan Gauss dağılımını kullanmamız gerekecek. Gauss dağılımı veya normal dağılım dediğimi duyduğunuzda, tamamen aynı şeyi kastediyorlar. Çan şeklindeki dağılımı duyduysanız, bu da aynı şeyi ifade eder. Ancak çan şeklindeki dağılımı duymadıysanız, bu da iyi. Ama Gauss veya normal dağılımın ne olduğuna bir göz atalım. X'in bir sayı olduğunu ve x'in bazen rasgele değişken olarak adlandırılan rasgele bir sayı olduğunu varsayalım, x rasgele değerler alabilir. x'in olasılığı, bir Gaussian veya normal dağılım tarafından, ortalama Mu parametresi ve varyans Sigma kare ile verilirse. Bunun anlamı, x olasılığının şöyle giden bir eğri gibi görünmesidir. Eğrinin merkezi veya ortası, ortalama Mu tarafından verilir ve bu eğrinin standart sapması veya genişliği, bu varyans parametresi Sigma tarafından verilir. Teknik olarak, Sigma'ya standart sapma denir ve Sigma'nın karesi veya Sigma'nın karesine dağılımın varyansı denir. Buradaki eğri, p x'in ne olduğunu veya x olasılığını gösterir. Çan şeklindeki eğriyi duyduysanız, bu çan şeklindeki eğridir çünkü kulelerdeki birçok klasik çanın, çan tokmağı burada asılıyken bu şekilde şekillendirildiği söylenir ve bu nedenle bu eğrinin şekli belli belirsiz anımsatır. Bugün hala bazı eski binalarda bulacağınız büyük çanların şekli. Benim elle çizilmiş olandan daha iyi görünüyor. Özgürlük Çanı'nın bir resmi var. Gerçekten de, Özgürlük Çanı'nın tepedeki şekli belli belirsiz çan şeklindeki bir eğridir. p(x)'in gerçekte ne anlama geldiğini merak ediyorsanız? İşte onu yorumlamanın bir yolu. Bunun anlamı, eğer bu olasılık dağılımından çekilen 100 sayıyı elde ederseniz ve bu dağılımdan alınan bu 100 sayının bir histogramını çizerseniz, buna benzeyen bir histogram elde edebilirsiniz. Belli belirsiz çan şeklinde görünüyor. Soldaki bu eğrinin gösterdiği şey, yalnızca 100 veya 1.000 veya bir milyon veya bir milyar örneğiniz olup olmadığı değildir. Ama pratik olarak sonsuz sayıda örneğiniz olsaydı ve bu pratik olarak sonsuz sayıda örneğin çok ince bir histogram kutusuyla bir histogramı çizseydiniz. Sonra, esas olarak burada, solda bu çan şeklindeki eğriyi elde edersiniz. p(x) formülü bu ifade ile verilir; p(x) eşittir 1 bölü karekök 2 Pi. Buradaki Pi, 3.14159 veya yaklaşık 22 bölü 7. Dairenin çapının çevre çarpı Sigma çarpı e üzeri eksi x eksi Mu, ortalama parametrenin karesi bölü 2 Sigma karesinin oranı. Mu ve Sigma'nın herhangi bir değeri için, bu fonksiyonu x'in bir fonksiyonu olarak çizerseniz, merkezi Mu'da olan ve bu çan şeklindeki eğrinin genişliği tarafından belirlenen bu tür çan şekilli bir eğri elde edersiniz. Sigma parametresi. Şimdi Mu ve Sigma'yı değiştirmenin Gauss dağılımını nasıl etkileyeceğine dair birkaç örneğe bakalım. İlk olarak, Mu'yu 0'a ve Sigma'yı 1'e ayarlamama izin verin. İşte ortalama 0, Mu'nun 0 ve standart sapmanın Sigma'nın 1 olduğu Gauss dağılımı grafiğim. Bu dağılımın sıfır merkezli olduğunu ve bunun standart sapma olduğunu fark ettiniz. Sigma 1'e eşittir. Şimdi Sigma'nın standart sapmasını 0,5'e indirelim. Gauss dağılımını Mu eşittir 0 ve Sigma eşittir 0,5 olacak şekilde çizerseniz, şimdi böyle görünüyor. Mu sıfır olduğu için hala sıfır merkezli olduğuna dikkat edin. Ama çok daha ince bir eğri oldu çünkü Sigma artık 1.5. Sigma'nın standart sapmasının 0,5 olduğunu, Sigma karesinin de varyans olarak adlandırıldığını hatırlayabilirsiniz. Bu, 0,5'in karesine veya 0,25'e eşittir. Olasılıkların toplamının her zaman bir olması gerektiğini duymuş olabilirsiniz, bu nedenle eğrinin altındaki alan her zaman bire eşittir, bu nedenle Gauss dağılımı zayıfladığında, aynı zamanda uzaması gerekir. Mu ve Sigma'nın başka bir değerine bakalım. Şimdi, Sigma'yı 2'ye çıkaracağım, yani standart sapma 2 ve varyans 4. Bu şimdi çok daha geniş bir dağılım yaratıyor çünkü buradaki Sigma artık çok daha büyük ve artık daha geniş bir dağılım olduğu için şu anda daha kısa oluyor. çünkü eğrinin altındaki alan hala 1'e eşittir. Son olarak Mu ortalama parametresini değiştirmeyi deneyelim ve Sigma'yı 0,5'e eşit bırakacağım. Bu durumda Mu dağılımının merkezi burada sağa doğru hareket eder. Ancak dağılımın genişliği üstteki ile aynıdır çünkü sağdaki bu iki durumda da standart sapma 0,5'tir. Bu, Mu ve Sigma'nın farklı seçimlerinin Gauss dağılımını nasıl etkilediğidir. Bunu anomali tespitine uygularken, yapmanız gerekenler şunlardır. Size m örnekten oluşan bir veri kümesi veriliyor ve burada x yalnızca bir sayı. İşte, 11 örnekli eğitim setlerinin grafikleri. Yapmamız gereken, ortalama Mu parametresi ve varyans parametresi Sigma kare için iyi bir seçimin ne olduğunu tahmin etmeye çalışmaktır.

Bunun gibi bir veri kümesi verildiğinde, burada bir merkez ve bunun gibi bir standart sapma ile böyle görünen bir Gauss dağılımı olabilir. Bu, verilere oldukça uygun olabilir. Mu ve Sigma karesini matematiksel olarak hesaplama şekliniz, Mu için tahminimizin tüm eğitim örneklerinin ortalaması olacağıdır. 1 bölü m çarpı i eşittir 1'den m'ye kadar eğitim örneklerinizdeki değerlerin toplamıdır. Sigma kareyi tahmin etmek için kullanacağımız değer, iki örnek arasındaki kare farkının ve burada solda az önce tahmin ettiğiniz Mu'nun ortalaması olacaktır. Görünüşe göre bu iki formülü Mu için bu değerle ve Sigma kare için bu değerle kodlarsanız, o zaman yukarıda elle çizdiğim Gauss dağılımını hemen hemen elde edersiniz. Bu size bir Gauss dağılımı için bir Mu ve Sigma seçeneği sunacaktır, böylece 11 eğitim örneği bu Gauss dağılımından çekilmiş gibi görünecektir. İleri düzey bir istatistik dersi aldıysanız, Mu ve Sigma kare için bu formüllerin teknik olarak Mu ve Sigma için maksimum olasılık tahminleri olarak adlandırıldığını duymuş olabilirsiniz. Bazı istatistik dersleri, 1 bölü m yerine 1 bölü n eksi 1 formülünü kullanmanızı söyleyecektir. Uygulamada, 1 bölü m veya 1 bölü n eksi 1 kullanmak çok az fark yaratır. Her zaman 1 bölü m kullanırım, ancak bazı istatistikçilerin tercih ettiği m eksi 1'e bölmenin diğer bazı özellikleri. Ama az önce söylediklerini anlamadıysanız, bunun için endişelenmeyin. Bilmeniz gereken tek şey, Mu'yu bu formüle göre ve Sigma'nın karesini bu formüle göre ayarlarsanız, Mu ve Sigma'nın oldukça iyi bir tahminini elde edersiniz ve özellikle olası bir olasılık olacak bir Gauss dağılımı elde edersiniz. eğitim örneklerinin geldiği olasılık dağılımı açısından dağılım. Sırada ne olduğunu muhtemelen tahmin edebilirsiniz. Buradan bir örnek alacak olursanız, o zaman p(x) oldukça yüksektir. Oysa bir örnek verecek olursanız, buradayız, o zaman p(x) oldukça düşüktür, bu yüzden bu örneği tamam, gerçekten anormal değil, diğerlerine pek benzemez. Oysa bir örnek, gördüğümüz örneklere göre oldukça sıra dışı olmak için buradayız ve bu nedenle daha anormal çünkü bu eğrinin yüksekliği olan p(x) burada soldaki bu noktadan çok daha düşük. , ortaya daha yakın. Şimdi, bunu sadece x bir sayı olduğunda yaptık, sanki anormallik tespit probleminiz için sadece tek bir özelliğiniz varmış gibi. Pratik anomali tespiti uygulamaları için genellikle birçok farklı özelliğe sahipsiniz. Artık Gauss dağılımının nasıl çalıştığını gördünüz. x tek bir sayıysa, bu if'e karşılık gelir, diyelim ki anormallik algılama probleminiz için yalnızca bir özelliğiniz var. Ancak pratik anormallik algılama uygulamaları için, birçok özelliğe sahip olacaksınız, iki veya üç veya hatta daha fazla sayıda n özellik. Tek bir Gauss için gördüklerinizi alalım ve daha gelişmiş bir anomali algılama algoritması oluşturmak için kullanalım. Birden çok özelliği işleyebilirler. Bir sonraki videoda bunu yapalım.

## Anomaly detection algorithm

Artık tek bir sayı için Gauss veya normal dağılımın nasıl çalıştığını gördüğünüze göre, anormallik algılama algoritmamızı oluşturmaya hazırız. Haydi konuya girelim. x1'den xm'ye kadar bir eğitim setiniz var, burada her x örneğinin bitiş özellikleri var. Yani, her x örneği sayılar içinde bir vektördür. Uçak motoru örneğinde, ısı ve titreşimlere karşılık gelen iki özelliğimiz vardı. Ve böylece, bu tüketim değerlerinin her biri iki boyutlu bir vektör olacak ve n, 2'ye eşit olacaktır. Ancak birçok pratik uygulamada n çok daha büyük olabilir ve bunu düzinelerce hatta yüzlerce özellikle yapabilirsiniz. Bu eğitim seti göz önüne alındığında, yapmak istediğimiz yoğunluk tahmini yapmaktır ve bunun tek anlamı, bir model kuracağız veya p(x) için olasılığı tahmin edeceğiz. Herhangi bir özellik vektörünün olasılığı nedir? Ve p(x) için modelimiz şu şekilde olacaktır, x, x1, x2 ve xn'ye kadar devam eden değerlere sahip bir özellik vektörüdür. Ve özellik vektörlerindeki n'inci özellik için p(x)'i x1'in olasılığı çarpı x2'nin olasılığı çarpı x3'ün olasılığı çarpı xn'nin olasılığı olarak modelleyeceğim. Daha önce muhtemelen istatistik alanında ileri düzeyde bir ders aldıysanız, bu denklemin x1, x2 ve benzeri xm'ye kadar olan özelliklerin istatistiksel olarak bağımsız olduğunu varsaymaya karşılık geldiğini fark edebilirsiniz. Ancak, bu algoritmanın, özellikler aslında istatistiksel olarak bağımsız olmasa bile genellikle iyi çalıştığı ortaya çıktı. Ama ne dediğimi anlamıyorsan, merak etme. Tartışmayı tam olarak tamamlamak ve ayrıca anormallik tespit algoritmasını çok etkili bir şekilde kullanabilmek için fiziksel bağımsızlığı anlamak gerekli değildir. Şimdi, bu denklemi biraz daha doldurmak için, x vektör özelliklerinin tüm özelliklerinin olasılığının p(x) 1 ve p(x2)'nin çarpımı olduğunu ve p(xn)'ye kadar böyle devam ettiğini söylüyoruz. ). Ve x1'in olasılığını modellemek için, diyelim ki bu örnekteki ısı özelliği iki parametremiz olacak, yeni 1 ve sigma 1 veya sigma kare 1. Bunun anlamı, x1 özelliğinin ortalaması ve ayrıca x1 özelliğinin varyansı ve bu yeni 1 ve sigma 1 olacaktır. p(x2) x2'yi modellemek, uçak motorunun titreşimlerini ölçen tamamen farklı bir özelliktir. Mu 2, sigma 2 kare olarak yazacağım iki farklı parametremiz olacak. Ve bunun, titreşim özelliğinin ortalamasına veya ortalamasına ve titreşim özelliğinin varyansına vb. karşılık geleceği ortaya çıktı. Ek özellikleriniz varsa mu 3 sigma 3'ün karesi yukarı mu n ve sigma n karesi. Olasılıkları neden çarptığımızı merak ediyorsanız, belki burada sezgi oluşturabilecek 1 örnek var. Bir uçak motorunun gerçekten sıcak, alışılmadık derecede sıcak olma ihtimalinin 1/10 olduğunu ve gerçekten sert titreme ihtimalinin 20'de 1 olduğunu varsayalım. O halde, gerçekten sıcak çalışma ve gerçekten sert titreme şansı nedir? Bunun olasılığının 1/10 çarpı 1/20 yani 1/200 olduğunu söylüyoruz. Bu nedenle, her iki cephenin de gerçekten sıcak ve gerçekten sert titreşen bir motora sahip olması pek olası değildir. Bu denklemi buraya yazmanın biraz daha derli toplu bir yolu, bunun j = 1'den n'ye kadar p(xj)'nin çarpımına eşit olduğunu söylemektir. Mu j ve sigma kare j olur mu parametreleri. Ve buradaki sembol, toplama sembolüne çok benzer, ancak toplama sembolü toplamaya karşılık gelirken buradaki sembol, j = 1'den n'ye kadar buradaki terimleri çarpmaya karşılık gelir.

Öyleyse, aday tespit sistemini nasıl oluşturabileceğinizi görmek için hepsini bir araya getirelim. İlk adım, anormal örneklerin göstergesi olabileceğini düşündüğünüz özellikleri (xi) seçmektir. Kullanmak istediğiniz özellikleri bulduktan sonra, veri kümenizdeki n özellik için mu 1'den mu n'ye ve sigma kare 1'den sigma kare n'ye kadar olan parametreleri uydurursunuz. Tahmin edebileceğiniz gibi, mu j parametresi, eğitim setinizdeki tüm örneklerin j özelliğinin xj'sinin ortalaması olacaktır. Ve sigma kare j, özellik ile az önce hesapladığınız mu j değeri arasındaki kare farkının ortalaması olacaktır. Ve bu arada, vektörize bir uygulamanız varsa, mu'yu aşağıdaki gibi eğitim örneklerinin ortalaması olarak da hesaplayabilirsiniz, buradayız, x ve mu'nun her ikisi de vektördür. Ve böylece bu, mu 1'den mu'ya ve hepsini aynı anda hesaplamanın vektörleştirilmiş yolu olacaktır. Ve etiketlenmemiş eğitim setinizde bu parametreleri tahmin ederek, artık modelinizin tüm parametrelerini hesaplamış oldunuz. Son olarak, size yeni bir örnek verildiğinde, x testi veya buraya x olarak yeni bir örnek yazacağım, yapacağınız şey p(x)'i hesaplamak ve onun büyük mü küçük mü olduğuna bakmak. Son slaytta gördüğünüz gibi p(x), bireysel özelliklerin olasılığının j = 1 ile n arasındaki çarpımıdır. Yani p(x) j, mu j ve tek kare j parametreleriyle. Ve bu olasılığı formülde yerine yazarsanız, bu ifadeyi 1 bölü kök 2 pi sigma j e'nin buradaki ifadeye göre elde edersiniz. Ve böylece xj özelliklerdir, bu yeni örneğinizin bir j özelliğidir, muj ve sigma j önceki adımda hesapladığınız sayılar veya parametrelerdir. Ve bu formülü hesaplarsanız, p(x) için bir sayı elde edersiniz. Ve son adım, p(x)'in epsilondan küçük olduğunu görmektir. Ve eğer öyleyse, bunun bir anormallik olduğunu işaretlersiniz.

Bu albümün arkasındaki sezgi, 1 veya daha fazla özelliğin eğitim setinde gördüklerine göre çok büyük veya çok küçük olması durumunda bir örneği anormal olarak işaretleme eğiliminde olmasıdır. Böylece, x j özelliklerinin her biri için, bunun gibi bir Gauss dağılımına uyuyorsunuz. Ve böylece, yeni örneğin özelliklerinden 1 tanesi bile buraya çok uzak olsaydı, o zaman P f xJ çok küçük olurdu. Ve eğer bu çarpımdaki terimlerden sadece 1 tanesi çok küçükse, o zaman bu toplam çarpım, birlikte çarptığınızda çok küçük olma eğiliminde olacak ve p(x) küçük olacaktır. Ve anormallik tespitinin algoritmaya yaptığı şey, bu yeni örnek x'in alışılmadık derecede büyük veya alışılmadık derecede küçük herhangi bir özelliğe sahip olup olmadığını ölçmenin sistematik bir yoludur. Şimdi, 1 örnek üzerinde tüm bunların aslında ne anlama geldiğine bir göz atalım, İşte x 1 ve x 2 özelliklerine sahip veri seti.

Ve x 1 özelliklerinin x 2 özelliklerinden çok daha geniş bir değer aralığı aldığını fark edersiniz.

Özellikler x 1'in memesini hesaplarsanız, sonunda beş elde edersiniz, bu yüzden 1'e eşit olmasını istiyorsunuz. yaklaşık 2. Ve mu'yu özelliklerin ortalamasına göre hesaplarsanız, ortalamanın yanındaki ortalama üçtür ve benzer şekilde varyans veya standart sapma çok daha küçüktür, bu nedenle Sigma 2 1'e eşittir. Yani bu Gauss'a karşılık gelir x 1 için dağılım ve x 2 için bu Gauss dağılımı. Eğer gerçekten p(x) 1 ve p(x) 2'yi çarpacak olsaydınız, p(x) için bu üç D yüzey grafiğini elde edersiniz, burada herhangi bir nokta, bunun yüksekliği p(x) 1 çarpı p(x) 2'nin çarpımıdır. Karşılık gelen x 1 ve x 2 değerleri için. Bu da p(x)'in daha yüksek veya daha olası olduğu değerleri ifade eder. Yani, buradaki orta türe yakın değerler daha olasıdır. Buradaki değerler uzaktayken, buradaki değerler çok daha az olasıdır. şansım çok daha düşük

Şimdi, test örneklerini seçeyim, ilk 1'i buraya x test 1'i ve ikinci 1'i buraya x test 2 olarak yazacağım. Ve bu 2 örnekten hangisini albümün anormal olarak işaretleyeceğini görelim.

ε Parametresini 0,02'ye eşit olarak seçeceğim. Ve p(x) test 1'i hesaplarsanız, yaklaşık 0.4 olduğu ortaya çıkıyor ve bu epsilondan çok daha büyük. Ve böylece albüm bunun iyi göründüğünü, bir anormallik gibi görünmediğini söyleyecek. Oysa tersine, buradaki bu nokta için p(x)'i hesaplarsanız, x 1 yaklaşık sekize ve x 2 yaklaşık 0,5'e eşittir.

Bir nevi aşağıda, o zaman p(x) test 2 0,0021'dir. Yani bu epsilondan çok daha küçük. Ve böylece albüm bunu olası bir anormallik olarak işaretleyecek. Yani, umduğunuz gibi, x testi 1'in oldukça normal göründüğüne karar verir. Oysa eğitim setinde gördüğünüz her şeyden çok daha uzakta olan aşırılık, bir anormallik olabilir gibi görünüyor. Bir anormallik tespit sisteminin nasıl kurulacağını gördünüz. Ancak epsilon parametresini nasıl seçersiniz? Bir sonraki videoda anormallik tespit sisteminizin iyi çalışıp çalışmadığını nasıl anlarsınız, bir anormallik tespit sisteminin performansını geliştirme ve değerlendirme sürecini biraz daha derinlemesine inceleyelim. Bir sonraki videoya geçelim

##Developing and evaluating an anomaly detection system

Anomali tespit sistemi geliştirmek için bazı pratik ipuçlarını sizinle paylaşmak istiyorum. En önemli fikirlerden biri, bir sistemi geliştirilirken bile değerlendirmenin bir yolunu bulabilirseniz, kararlar alıp sistemi değiştirebilecek ve çok daha hızlı bir şekilde geliştirebileceksiniz. Bunun ne anlama geldiğine bir bakalım. Bir öğrenme algoritması geliştirirken, örneğin farklı özellikler seçmek veya epsilon gibi parametrelerin farklı değerlerini denemek, bir özelliği belirli bir şekilde değiştirip değiştirmeyeceğinize veya epsilon veya diğer parametreleri artırıp azaltmayacağınıza karar vermek, bu kararları vermek çok daha kolaydır. öğrenme algoritması. Buna bazen gerçek sayı değerlendirmesi denir, yani algoritmayı bir özelliği değiştirmek veya bir parametreyi değiştirmek gibi bir şekilde hızlı bir şekilde değiştirebilirseniz ve algoritmanın daha iyi veya daha kötü olup olmadığını size söyleyen bir sayı hesaplama yöntemine sahipseniz, o zaman algoritmanın daha iyi veya daha kötü olup olmadığına karar vermeyi çok daha kolaylaştırır. bu değişikliğe algoritmaya bağlı kalmak için. Anomali tespitinde genellikle bu şekilde yapılır. Yani, esas olarak etiketlenmemiş verilerden bahsetmiş olsak da, bu varsayımı biraz değiştireceğim ve genellikle daha önce gözlemlenen anomalilerin az bir kısmı da dahil olmak üzere bazı etiketli verilerimiz olduğunu varsayacağım. Belki birkaç yıldır uçak motorları yaptıktan sonra, anormal olan birkaç uçak motoru gördünüz ve anormal olduğunu bildiğiniz örnekler için, bu anormalliği belirtmek için y eşittir 1 etiketini ilişkilendireceğim ve normal olduğunu düşündüğümüz örnekler için, bu anormalliği belirtmek için y eşittir 1 etiketini ilişkilendireceğim. y etiketi 0'a eşittir. Anomali tespit algoritmasının öğreneceği eğitim seti hala x1'den xm'e kadar olan bu etiketsiz eğitim setidir ve tüm bu örnekleri sadece normal ve anormal olmadığını varsayacağımız örnekler olarak düşüneceğim, bu yüzden y eşittir 0. Pratikte, bu eğitim setine nereye gireceğinize dair birkaç anormal örneğiniz var, algoritmanız yine de genellikle iyi sonuç verecektir. Algoritmanızı değerlendirmek için, gerçek sayı değerlendirmesine sahip olmanızın bir yolunu bulun, az sayıda anormal örneğiniz varsa, çapraz doğrulama kümesi oluşturabilmeniz için çok yararlı olduğu ortaya çıkıyor. x\_cv ^ 1, y\_cv ^ 1 ile x\_cv'yi göstereceğim^mcv, y\_cv^ mcv. Bu, bu uzmanlığın ikinci dersinde gördüğünüz gibi benzer bir gösterimdir. Benzer şekilde, hem çapraz doğrulama hem de test setlerinin umarım birkaç anormal örnek içerdiği bazı örneklerden oluşan bir test setine sahip olun. Başka bir deyişle, çapraz doğrulama ve test kümeleri, y'nin 1'e eşit olduğu birkaç örneğe, aynı zamanda y'nin 0'a eşit olduğu birçok örneğe sahip olacaktır. Yine, pratikte, aslında anormal olan bazı örnekler varsa, ancak yanlışlıkla y eşittir 0 ile etiketlenmişse, anormallik saptama algoritması iyi çalışacaktır. Bunu uçak motoru örneğiyle gösterelim. Diyelim ki yıllardır uçak motorları üretiyorsunuz ve bu nedenle 10.000 iyi veya normal motordan veri topladınız, ancak yıllar içinde 20 hatalı veya anormal motordan da veri topladınız. Genellikle anormal motorların sayısı, yani y eşittir 1, çok daha küçük olacaktır. Bu tür bir algoritmayı, örneğin bilinen 2-50 anormallik arasında herhangi bir yerde uygulamak tipik olmayacaktır. Bu veri kümesini alıp bir eğitim kümesine, çapraz doğrulama kümesine ve test kümesine ayıracağız. İşte bir örnek. Eğitim setine 6.000 iyi motor koyacağım. Yine, bu sete kaymış birkaç anormal motor varsa, aslında sorun değil, bunun için fazla endişelenmem. O zaman çapraz doğrulama setine 2.000 iyi motor ve bilinen anormalliklerden 10'unu ve test setine ayrı bir 2.000 iyi ve 10 anormal motor koyalım. O zaman yapabileceğiniz şey, algoritmayı eğitim setinde eğitmek, Gauss dağılımlarını bu 6.000 örneğe sığdırmak ve ardından çapraz doğrulama setinde, anormal motorların kaçının doğru şekilde işaretlediğini görebilirsiniz. Örneğin, epsilon parametresini ayarlamak için çapraz doğrulama kümesini kullanabilir ve algoritmanın bu 2.000 iyi motordan çok fazla almadan ve bunları anormallik olarak işaretlemeden bu 10 anormalliği güvenilir bir şekilde tespit edip etmediğine bağlı olarak daha yüksek veya daha düşük ayarlayabilirsiniz. Epsilon parametresini ayarladıktan ve belki de X\_J özelliklerine ekledikten veya çıkardıktan veya ayarladıktan sonra algoritmayı alabilir ve bu 10 anormal motordan kaç tanesini bulduğunu ve kaç tane hata yaptığını görmek için test setinizde değerlendirebilirsiniz. iyi motorları anormal motorlar olarak işaretleyerek yapar. Bunun hala öncelikli olarak denetimsiz bir öğrenme algoritması olduğuna dikkat edin, çünkü eğitim setlerinde gerçekten etiket yok veya hepsinin y'nin 0'a eşit olduğunu varsaydığımız etiketleri var ve bu nedenle önceki videoda gördüğünüz gibi Gauss dağılımlarını uydurarak eğitim setinden öğrendik. Ancak, algoritmayı değerlendirmek için kullanılacak az sayıda anomaliye sahip pratik bir anomali tespit sistemi oluşturuyorsanız, çapraz doğrulama ve test setlerinizin algoritmayı ayarlamak için çok yardımcı olduğu ortaya çıkıyor. Kusurlu motorların sayısı o kadar küçük olduğu için, insanların anomali tespiti için sıklıkla kullandıkları bir alternatif daha var, bu da bir test seti kullanmamak, sadece bir eğitim seti ve çapraz doğrulama seti kullanmak gibi. Bu örnekte, train'i 6.000 iyi motora ayarlayacaksınız, ancak verilerin geri kalanını, kalan 4.000 iyi motoru ve tüm anormallikleri alacak ve bunları çapraz doğrulama kümesine koyacaksınız. Daha sonra Epsilon parametrelerini ayarlar ve çapraz doğrulama kümesinde değerlendirildiği kadar iyi yapabilmek için hayali x\_j'yi ekler veya çıkarırsınız. Çok az kusurlu motorunuz varsa, bu nedenle yalnızca iki kusurlu motorunuz varsa, bunların hepsini çapraz doğrulama kümesine koymak gerçekten mantıklıdır. Çapraz doğrulama kümenizden farklı, tamamen ayrı bir test kümesi oluşturmak için yeterli veriye sahip değilsiniz. Buradaki bu alternatifin dezavantajı, algoritmanızı ayarladıktan sonra, test setine sahip olmadığınız için bunun gelecekteki örneklerde gerçekte ne kadar iyi olacağını söylemenin adil bir yolunun olmamasıdır. Ancak veri kümeniz küçük olduğunda, özellikle sahip olduğunuz anormalliklerin sayısı, veri kümeniz küçük olduğunda, sahip olduğunuz en iyi alternatif bu olabilir. Ayrı bir test seti oluşturmak için yeterli veriye sahip olmadığınızda bunun da oldukça sık yapıldığını görüyorum. Eğer durum buysa, Epsilon ve özellik seçimi ile ilgili bazı kararlarınızı çapraz doğrulama kümesine fazla sığdırmanız riskinin daha yüksek olduğunu ve bu nedenle gelecekte gerçek veriler üzerindeki performansının beklediğiniz kadar iyi olmayabileceğini unutmayın. Şimdi, algoritmanın çapraz doğrulama kümelerinizde veya test kümenizde gerçekte nasıl değerlendirileceğine daha yakından bakalım. Şöyle yapardın. İlk önce eğitim setine x'in p modelini sığdıracaksınız. Bu, 6.000 mal motoru örneğiydi. Daha sonra herhangi bir çapraz doğrulama veya test örneği x'de, x'in p'sini hesaplar ve y'nin 1'e eşit olduğunu tahmin edersiniz. P x'in Epsilondan küçük olması ve p x'in Epsilondan büyük veya ona eşit olması durumunda y'nin 0 olduğunu tahmin etmeniz bu anormaldir. Buna dayanarak, artık bu algoritmanın çapraz doğrulama veya test kümesindeki tahminlerinin etiketlerle ne kadar doğru eşleştiğine bakabilirsiniz. , y çapraz doğrulamada veya test kümelerinde var. İkinci dersin üçüncü haftasında, y'nin 1'e eşit olduğu pozitif örneklerin sayısının, y'nin 0'a eşit olduğu negatif örneklerin sayısından çok daha küçük olabileceği, oldukça çarpık veri dağılımlarının nasıl ele alınacağına dair birkaç isteğe bağlı video çektik. Çapraz doğrulama kümenizdeki anormalliklerin sayısının çok daha az olduğu uygulamalarda birçok anormallik tespiti için de durum böyledir. Bir önceki örneğimizde, 10 anomalimiz ve 2.000 normal örneğimiz olduğu için belki 10 olumlu ve 2.000 olumsuz örneğimiz vardı. Bu isteğe bağlı videoları gördüyseniz, gerçek pozitif, yanlış pozitif, yanlış negatif ve gerçek negatif [duyulmuyor] gibi şeyleri hesaplamanın yararlı olabileceğini gördüğümüzü hatırlayabilirsiniz. Ayrıca, hassas geri çağırma veya F\_1 puanını hesaplayın ve bunların, veri dağıtımınız çok çarpık olduğunda daha iyi çalışabilecek alternatif ölçümler ve sınıflandırma doğruluğu olduğunu hesaplayın. Bu videoyu gördüyseniz, öğrenme algoritmanızın normal düzlem motorlarının bu çok daha büyük negatif örnek kümesinin ortasında o küçük avuç anormalliği veya olumlu örnekleri bulmada ne kadar iyi olduğunu söylemek için bu tür değerlendirme metriklerini uygulamayı düşünebilirsiniz. O videoyu izlemediysen, endişelenme. Sorun değil. Umarım elde edeceğiniz sezgi, kaç anomalinin bulunduğuna ve ayrıca kaç normal motorun bir anomali olarak yanlış işaretlendiğine bakmak için çapraz doğrulama setini kullanmaktır. O zaman bunu sadece Epsilon parametresi için iyi bir seçim yapmaya çalışmak için kullanın. Bilinen anomalilerin az sayıda etiketli örneğine sahipseniz, bir anomali tespit sistemi oluşturmanın pratik sürecinin çok daha kolay olduğunu görürsünüz. Şimdi, bu soruyu gündeme getiriyor, eğer birkaç etiketli örneğiniz varsa, hala denetimsiz bir öğrenme algoritması kullanacağınıza göre, neden bu etiketli örnekleri alıp bunun yerine denetimli bir öğrenme algoritması kullanmıyorsunuz? Bir sonraki videoda, anomali tespiti ile denetimli öğrenme arasındaki karşılaştırmaya ve birini diğerine ne zaman tercih edebileceğinize bir göz atalım. Bir sonraki videoya geçelim.

## Anomaly detection vs. supervised learning

Michael'ın 1'i ile birkaç olumlu örneğiniz olduğunda ve çok sayıda olumsuz örnek y = 0 dediğinde? Anomali tespitini ne zaman kullanmalı ve denetimli öğrenmeyi ne zaman kullanmalısınız? Karar aslında bazı uygulamalarda oldukça ince. Bu nedenle, bu iki algoritma türü arasında nasıl seçim yapılacağına dair bazı düşünceleri ve bazı önerileri sizinle paylaşmama izin verin. Ve anomali tespit algoritması, çok az sayıda pozitif örneğiniz olduğunda tipik olarak daha uygun seçim olacaktır, 0-20 pozitif örnek nadir değildir. Ve x'in p'si için bir model oluşturmaya çalışılacak nispeten çok sayıda olumsuz örnek. X'in p'si için parametrelerin yalnızca olumsuz örneklerden öğrenildiğini ve bunun çok daha küçük olduğunu hatırladığınızda. Bu nedenle, olumlu örnekler yalnızca parametre ayarlama ve değerlendirme için çapraz doğrulama kümenizde ve test kümenizde kullanılır. Buna karşılık, daha fazla sayıda olumlu ve olumsuz örneğiniz varsa, denetimli öğrenme daha uygulanabilir olabilir. Şimdi, yalnızca 20 olumlu eğitim örneğiniz olsa bile, denetimli bir öğrenme algoritması uygulamak sorun olmayabilir. Ancak, anormallik tespitinin veri kümesine bakma biçimine karşı denetimli öğrenmenin veri kümesine bakma biçiminin oldukça farklı olduğu ortaya çıktı. O ana farktır, yani birçok farklı türde bariz veya birçok farklı türde olumlu örnek olduğunu düşünüyorsanız. O zaman bir uçak motorunun ters gitmesinin birçok farklı yolu olduğunda anormallik tespiti daha uygun olabilir. Ve eğer yarın bir uçak motorunda bir sorun olması için yepyeni bir yol olabilirse. O zaman 20 olumlu örneğiniz, bir uçak motorunun yanlış gidebileceği tüm yolları kapsamayabilir. Bu, herhangi bir algoritmanın küçük pozitif örnekler kümesinden anomalilerin, pozitif örneklerin neye benzediğini öğrenmesini zorlaştırır. Ve gelecekteki anomaliler, şimdiye kadar gördüğümüz anormal örneklerden hiçbirine benzemeyebilir. Bunun probleminiz için doğru olduğuna inanıyorsanız, o zaman bir anormallik tespit algoritması kullanmaya yönelirim. Çünkü anormallik tespitinin yaptığı, y = 0 negatif örnekleri olan normal örneklere bakması ve sadece neye benzediklerini modellemeye çalışmasıdır. Ve normalden çok sapan her şey Bir anormallik olarak işaretlenir. Veri kümenizde daha önce hiç görülmemiş bir uçak motorunun arızalanması için yepyeni bir yol olup olmadığı dahil. Buna karşılık, denetimli öğrenmenin soruna farklı bir bakış açısı vardır. Denetimli öğrenmeyi ideal olarak uygularken, ortalamanın olumlu örneklerin neye benzediğine dair bir fikir edinmesi için yeterli olumlu örneğe sahip olmayı umarsınız. Ve denetimli öğrenme ile, gelecekteki olumlu örneklerin eğitim setindekilere benzer olma ihtimalinin yüksek olduğunu varsayma eğilimindeyiz. Öyleyse bunu bir örnekle açıklayayım, bulmak için bir sistem kullanıyorsanız, finansal dolandırıcılık deyin. Ne yazık ki, bazı kişilerin finansal dolandırıcılık yapmaya çalışmasının birçok farklı yolu vardır. Ve ne yazık ki, birkaç ayda bir veya her yıl yeni tür finansal dolandırıcılık girişimleri var. Ve bunun anlamı, tamamen yeni ortaya çıkmaya devam ettikleri için. Ve benzersiz finansal dolandırıcılık anormallik tespiti biçimleri genellikle farklı olan herhangi bir şeyi aramak için kullanılır, o zaman geçmişte gördüğümüz işlemler. Buna karşılık, e-posta spam algılama sorununa bakarsanız, birçok farklı spam e-posta türü vardır, ancak uzun yıllar boyunca bile. Spam e-postalar benzer şeyleri satmaya veya benzer web sitelerine gitmenizi sağlamaya çalışmaya devam eder. Önümüzdeki birkaç gün içinde alacağınız spam e-postaların geçmişte gördüğünüz spam e-postalara benzemesi çok daha olasıdır. Bu nedenle, denetimli öğrenmenin spam için iyi çalışmasının nedeni, eğitim setinizde muhtemelen geçmişte gördüğünüz spam e-posta türlerini daha fazla tespit etmeye çalışmasıdır. Oysa aday gösterme bölümünde daha önce hiç görülmemiş yepyeni dolandırıcılık türlerini tespit etmeye çalışıyorsanız, belki daha uygulanabilir. Birkaç örnek daha gözden geçirelim. Sahtekarlık tespitinin anomali tespitinin bir kullanım durumu olduğunu zaten gördük. Denetimli öğrenme, daha önce gözlemlenen dolandırıcılık biçimlerini bulmak için kullanılmasına rağmen. Ve e-posta spam sınıflandırmasının genellikle denetimli öğrenmeyi kullanarak adres olduğunu gördük. Daha önce görülmemiş yeni kusurlar bulmak isteyebileceğiniz üretim örneğini de gördünüz. Gelecekte hala tespit etmek istediğiniz bir uçak motorunun arızalanması için yepyeni yollar olup olmadığı gibi. Eğitim setinizde böyle olumlu bir örneğiniz olmasa bile. Üretim denetimli öğrenmenin kusurları bulmak için de kullanıldığı ortaya çıktı. Bilinen ve daha önce görülen kusurları bulmak için daha fazla. Örneğin, bir akıllı telefon üreticisiyseniz, cep telefonları yapıyorsunuz demektir. Ve bazen akıllı telefonun kasasını yapmak için makinenizin yanlışlıkla kapağı gereceğini biliyorsunuz. Bu nedenle, esnemeler akıllı telefonlarda yaygın bir kusurdur ve bu nedenle y = 1 etiketine yanıt veren gerilmiş akıllı telefonların yeterli eğitim örneklerini alabilirsiniz. Ve yeni ürettiğiniz yeni bir akıllı telefonun içinde herhangi bir uzantı olup olmadığına karar vermek için sistemi eğitin. Aradaki fark, akıllı telefonların tekrar tekrar gerildiğini görürseniz ve telefonlarınızın gerilip gerilmediğini kontrol etmek istiyorsanız, denetimli öğrenme iyi çalışır. Gelecekte bir şeylerin ters gitmesi için yepyeni yollar olacağından şüpheleniyorsanız, anormallik tespiti iyi sonuç verecektir. Veri merkezindeki izleme makineleri hakkında konuştuğumu duyduğunuz diğer bazı örnekler, özellikle de makine saldırıya uğradı. Davranışındaki önceki herhangi bir yoldan farklı olarak yepyeni bir şekilde farklı davranabilir. Yani bu daha çok bir anormallik tespit uygulaması gibi hissettirirdi. Aslında, bir tema, güvenlikle ilgili birçok uygulamanın, bilgisayar korsanlarının genellikle sistemlere girmenin yepyeni yollarını bulmasıdır. Güvenlikle ilgili birçok uygulama anomali tespitini kullanacaktır. Denetimli öğrenmeye geri dönerken, hava durumunu iyi tahmin etmeyi öğrenmek istiyorsanız, tipik olarak gördüğünüz yalnızca birkaç hava türü vardır. Güneşli mi, yağmurlu mu, kar yağacak mı? Ve böylece aynı çıktı etiketlerini tekrar tekrar gördüğünüz için, hava durumu tahmini denetimli bir öğrenme görevi olma eğiliminde olacaktır. Ya da hastanın daha önce gördüğünüz belirli bir hastalığı olup olmadığını görmek için hastanın semptomlarını kullanmak istiyorsanız. O zaman bu aynı zamanda denetimli bir öğrenme uygulaması olma eğiliminde olacaktır. Umarım bu, küçük bir dizi olumlu örneğe ve belki de büyük bir dizi olumsuz örneğe ne zaman sahip olduğunuza karar vermeniz için size bir çerçeve sunar. Anomali tespiti veya denetimli öğrenme kullanılıp kullanılmayacağı. Anomali tespiti, daha önce gördüğünüz hiçbir şeye benzemeyen yepyeni olumlu örnekler bulmaya çalışır. Denetimli öğrenmenin olumlu örneklerinize baktığı ve gelecekteki bir örneğin daha önce gördüğünüz olumlu örneklere benzer olup olmadığına karar vermeye çalıştığı yer. Şimdi, bir anomali tespit algoritması oluştururken, özelliklerin seçiminin çok önemli olduğu ve anomali tespit sistemleri oluştururken olduğu ortaya çıktı. Bir sonraki videoda sistem için kullandığım özellikleri ayarlamak için sık sık biraz zaman harcıyorum. Beslediğiniz özelliklerin anormallik algılama algoritmasına nasıl ayarlanacağına dair bazı pratik ipuçlarını paylaşmama izin verin.

##Choosing what features to use

Bir anormallik tespit algoritması oluştururken, iyi bir özellik seçimi seçmenin gerçekten önemli olduğunu buldum. Denetimli öğrenmede, özelliklere tam olarak sahip değilseniz veya sorunla alakalı olmayan birkaç ekstra özelliğiniz varsa, bu genellikle iyi olur. Algoritmanın denetimli sinyal vermesi gerektiğinden, algoritmanın hangi özelliklerin yoksayıldığını veya özelliğin nasıl yeniden ölçeklendirileceğini ve ona verdiğiniz özelliklerden en iyi şekilde yararlanabileceğini anlaması için neden yeterli etiket vardır. Ancak, yalnızca etiketlenmemiş verilerden çalışan veya öğrenen anomali tespiti için, anomalinin hangi özelliklerin göz ardı edileceğini anlaması daha zordur. Bu yüzden, özellikleri dikkatlice seçmenin, anormallik tespiti için denetimli öğrenme yaklaşımlarından daha önemli olduğunu buldum. Size mümkün olan en iyi performansı elde etmeye çalışmak için anormallik tespiti için özelliklerin nasıl ayarlanacağına dair bazı pratik ipuçları olarak bu videoya bir göz atalım. Anomali tespit algoritmanıza yardımcı olabilecek bir adım, ona verdiğiniz özelliklerin az çok Gauss olduğundan emin olmaya çalışmaktır. Ve eğer özellikleriniz Gauss değilse, bazen onu biraz daha Gauss yapmak için değiştirebilirsiniz. Sana ne demek istediğimi göstereyim. Bir X özelliğiniz varsa, genellikle python komutu plt'yi kullanarak yapabileceğiniz özelliğin bir gramını tıslamak için çizerim. Bunu uygulama laboratuvarında da görseniz de, verilerin geçmişine bakmak için graham. Buradaki dağılım oldukça Gauss görünüyor. Yani bu iyi bir aday özellik olacaktır. Bunun anomaliler ve normal örnekler arasında ayrım yapmayı ümit eden bir özellik olduğunu düşünüyorsanız. Ancak çoğu zaman, özelliklerinizin bir gramını tısladığınızda, özelliğin böyle bir dağılıma sahip olduğunu görebilirsiniz. Bu hiç de simetrik çan şeklindeki eğriye benzemiyor. Bu durumda, bu X özelliğini alıp daha Gauss yapmak için dönüştürüp dönüştüremeyeceğinizi düşünürdüm. Örneğin, belki de X'in günlüğünü hesaplayacak ve bir gram X günlüğünü tıslayacak olsaydınız, şuna bakın ve bu çok daha Gaussça görünüyor. Ve eğer bu özellik özellik X bir ise, o zaman soldaki gibi görünen orijinal özellik X one'ı kullanmak yerine, bu dağıtımı buraya getirmek için bu özelliği log of X one ile değiştirebilirsiniz. Çünkü X bir daha Gauss yapıldığında. Böyle bir Gauss dağılımı kullanan anomali tespit modelleri P X one olduğunda, verilere iyi uyma olasılığı daha yüksektir. Günlük işlevi dışında, yapabileceğiniz diğer şeyler, farklı bir özellik X iki verildiğinde, onu X iki ile değiştirebilirsiniz, X iki artı bir günlüğü. Bu, X ikiyi dönüştürmenin farklı bir yolu olacaktır. Ve daha genel olarak, X iki artı C'nin günlüğü, X'i daha Gauss yapmak için değiştirmek için kullanabileceğiniz bir formülün bir örneği olacaktır. Ya da farklı bir özellik için, karekökü almayı deneyebilir ya da gerçekten kare bu X ipucunu bir yarının gücüne götürürdü ve bu terimi katlanarak değiştirebilirsiniz. Yani farklı bir özellik için X dört, örneğin üçte birinin gücüne X dört kullanabilirsiniz. Bu yüzden bir anormallik tespit sistemi kurarken, bazen özelliklerime bir göz atacağım ve tıslayan bir gram çizerek Gauss olmayan herhangi bir şey görürsem, denemek için bunlar veya diğerleri gibi dönüşümleri seçebilirim. daha fazla Gauss yapmak için. Daha büyük bir C değeri ortaya çıkıyor, bu dağılımı daha az dönüştürecek. Ama pratikte sadece bir sürü farklı C değeri deniyorum ve sonra dağılımı daha Gaussian yapmak açısından daha iyi görünen birini seçmeye çalışıyorum. Şimdi, bunu nasıl yaptığımı ve bir defter koyduğunuzu göstermeme izin verin. Yani özelliklerdeki farklı dönüşümleri keşfetme süreci böyle görünebilir.

Bir X özelliğiniz olduğunda, tıslayan bir gramını aşağıdaki gibi çizebilirsiniz. Aslında bir gram tıslamanın güzel bir nedeni var gibi görünüyor. Geçmiş gramımdaki kutu sayısını 50'ye çıkarmama izin verin. Yani bidonlar orada 50'ye eşittir. Bir gram bidonu tıslayan da buydu. Ve bu arada, rengi değiştirmek istiyorsanız, bunu aşağıdaki gibi de yapabilirsiniz. Ve eğer farklı bir dönüşüm denemek istiyorsanız, örneğin X'in karekökünü çizmeyi deneyebilirsiniz. Yani X, yine 50 tıslama gramlık kutularla 0,5'in gücüne eşittir, bu durumda şöyle görünebilir. Ve bu aslında biraz daha Gauss görünüyor. Ama mükemmel değil ve farklı bir parametre deneyelim. 4.25'in gücünü deneyeyim. Belki biraz fazla uzaktayım. Oldukça Gauss'a benzeyen eski 0.4. Yani yapabileceğiniz bir şey, X'i 0.4'lük mükemmel güçle değiştirmek. Ve böylece X'i X'e eşit olacak şekilde 0.4'ün gücüne ayarlarsınız. Ve bunun yerine eğitim sürecinizde sadece X değerini kullanın. Ya da size başka bir dönüşüm göstereyim. Burada, X'in günlüğünü almayı deneyeceğim. Bu nedenle, 50 kutu ile tespit edilen X'in günlüğü, numpy log işlevini aşağıdaki gibi kullanacağım. Ve bir hata aldığınız ortaya çıktı, çünkü bu mükemmel çıktı. Bu örnekte sıfıra eşit bazı değerler vardır ve sıfırın günlüğüne negatif sonsuzluk tanımlanmamıştır. Çok yaygın bir numara, oraya çok küçük bir sayı eklemektir.

Böylece ihracat 0.001, negatif olmaz. Ve böylece böyle görünen tıslayan gram elde edersiniz. Ancak dağılımın daha Gauss görünmesini istiyorsanız, bunun bir değeri olup olmadığını görmek için bu parametreyle de oynayabilirsiniz. Kullanıcı verilerinin daha simetrik görünmesine ve belki de aşağıdaki gibi daha Gauss görünmesine neden olun. Ve şu anda gerçek zamanlı olarak yaptığım gibi, bunu görebilirsiniz, bu parametreleri çok hızlı bir şekilde değiştirebilir ve tıslayan gramı çizebilirsiniz. Bir göz atmaya çalışmak ve yukarıdaki tıslayan gramda gördüğünüz orijinal verilerden biraz daha Gaussian bir şey elde etmeye çalışmak için. Makine öğrenimi literatürünü okursanız, bu dağılımların Gaussçe'ye ne kadar yakın olduğunu otomatik olarak ölçmenin bazı yolları vardır. Ama pratikte buldum, büyük bir fark yaratmıyor, sadece birkaç değer denerseniz ve size uygun görünen, tüm pratik amaçlar için işe yarayacak bir şey seçerseniz. Jüpiter defterinde bir şeyler deneyerek, verilerinizi daha Gauss yapan bir dönüşüm seçmeye çalışabilirsiniz. Bir hatırlatma olarak, eğitim setine hangi dönüşümü uygularsanız uygulayın, lütfen aynı dönüşümü çapraz doğrulama ve test seti verilerinize de uygulamayı unutmayın. Verilerinizin yaklaşık olarak Gauss olduğundan emin olmanın yanı sıra, anormallik algılama algoritmanızı eğittikten sonra, güven doğrulama kümenizde bu kadar iyi çalışmıyorsa, anormallik tespiti için bir hata analizi işlemi de gerçekleştirebilirsiniz. Başka bir deyişle, hata yaparken algoritmanın henüz iyi gitmediği yere bakmayı deneyebilir ve ardından iyileştirmeler yapmak için bunu kullanabilirsiniz. Bir hatırlatma olarak, istediğimiz şey X'in P'sinin büyük olması. Normal örnekler için X, epsilona eşitten çok büyük ve p f X, anormal örnekler için epsilondan küçük veya daha küçük olmalıdır X. Model öğrendiğinizde P nın-nin X Etiketlenmemiş verilerinizden karşılaşabileceğiniz en yaygın sorun, P nın-nin X karşılaştırılabilir değer olarak, hem normal hem de anormal örnekler için büyüktür. Somut bir örnek olarak, eğer bu sizin veri setinizse, o galaksiyi ona sığdırabilirsiniz. Ve eğer çapraz doğrulama kümenizde veya test kümenizde bir örneğiniz varsa, bu burada, bu anormal, o zaman bu oldukça yüksek bir olasılığa sahiptir. Ve aslında, eğitim setinizdeki diğer örneklere oldukça benziyor. Ve böylece, bu bir anormallik olmasına rağmen, X'in P'si aslında oldukça büyüktür. Ve böylece algoritma bu özel örneği bir anormallik olarak işaretleyemeyecektir. Bu durumda, normalde yapacağım şey, bu örneğe bakmaya çalışmak ve bu özellik X one diğer eğitim örneklerine benzer değerler alsa bile, beni bir anormallik olarak düşündüren şeyin ne olduğunu anlamaya çalışmaktır. Ve eğer yeni bir özelliği tanımlayabilirsem, X iki diyelim, bu örneği normal örneklerden ayırmaya yardımcı olur. Ardından bu özelliği eklemek, algoritmanın performansını artırmaya yardımcı olabilir. İşte ne demek istediğimi gösteren bir resim. Yeni bir özellik bulabilirsem X iki, diyelim ki hileli davranışları tespit etmeye çalışıyorum ve yaptıkları işlem sayısı X bir ise, belki bu kullanıcı bazı işlemleri herkes gibi yapıyor gibi görünüyor. Ancak, bu kullanıcının delicesine hızlı bir yazma hızına sahip olduğunu keşfedersem ve yeni bir özellik X iki ekleseydim, bu kullanıcının yazma hızıdır. Ve eğer bu verileri eski X bir özelliğini ve bu yeni X iki özelliğini kullanarak çizdiğimde, X iki'nin burada öne çıkmasına neden olduğu ortaya çıkarsa. Daha sonra anomali tespit algoritmasının bir X ikinin anormal bir kullanıcı olduğunu tanıması çok daha kolay hale gelir. Çünkü bu yeni X iki özelliğine sahip olduğunuzda, öğrenme anomalisi, bu bölgedeki noktalara yüksek olasılık atayan, bu bölgede biraz daha düşük ve bu bölgede biraz daha düşük bir Gauss dağılımına uyabilir. Ve böylece bu örnek, X ikinin çok anormal değeri nedeniyle, bir anormallik olarak tespit edilmesi daha kolay hale gelir. Bu nedenle, geliştirme sürecinin sık sık geçeceğini özetlemek gerekirse, modeli eğitmek ve ardından algoritmanın tespit edemediği çapraz doğrulama kümesindeki hangi anormallikleri görmek gerekir. Ve sonra, algoritmanın tespit etmesine izin verecek yeni özelliklerin yaratılmasına ilham verip veremeyeceğini görmek için bu örneklere bakmak. Bu örnek, yeni özelliklerde alışılmadık derecede büyük veya alışılmadık derecede küçük değerler alır, böylece artık bu örnekleri anormallikler olarak başarıyla işaretleyebilirsiniz. Bir örnek olarak, veri merkezindeki bilgisayarları izlemek için bir anormallik algılama sistemi oluşturduğunuzu varsayalım. Bir bilgisayarın garip davranıp davranmadığını ve belki bir donanım arızası nedeniyle mi yoksa saldırıya uğradığı için mi daha yakından bakmayı hak ettiğini anlamaya çalışmak. Yani yapmak istediğiniz şey, bir anormallik durumunda alışılmadık derecede büyük veya küçük değerler alabilecek özellikleri seçmek. X one bellek kullanımı, X two saniyede disk erişimi sayısı, ardından CPU yükü ve ağ trafiğinin hacmi gibi özelliklerle başlayabilirsiniz. Algoritmayı eğitirseniz, bazı anormallikleri tespit ettiğini ancak diğer bazı anormallikleri tespit edemediğini görebilirsiniz. Bu durumda, eski özellikleri birleştirerek yeni özellikler oluşturmak alışılmadık bir durum değildir. Örneğin, çok garip davranan bir bilgisayar olduğunu fark ederseniz, ancak ne CPU yükü ne de ağ trafiği o kadar sıra dışıdır. Ancak olağandışı olan, çok düşük bir ağ trafiği hacmine sahipken gerçekten yüksek bir CPU yüküne sahip olmasıdır. Video akışı sağlayan veri merkezini çalıştırıyorsanız, bilgisayarlarda yüksek CPU yükü ve yüksek ağ trafiği olabilir veya düşük CPU yükü olabilir ve ağ trafiği olmayabilir. Ancak bu makinede olağandışı olan şey, çok düşük trafik hacmine rağmen çok yüksek bir CPU yüküdür. Bu durumda, CPU yükünün ağ trafiğine oranı olan yeni bir özellik X beş oluşturabilirsiniz. Ve umutla bu yeni özellik, anormallik tespit algoritması, anormal olarak gördüğünüz belirli makine gibi gelecekteki örnekleri işaretledi. Veya CPU yükünün karesi gibi ağ trafiği hacmine bölünmüş diğer özellikleri de düşünebilirsiniz. Ve bu özelliklerin farklı seçenekleriyle oynayabilirsiniz. Bunu elde etmeye çalışmak için, X'in P'si normal örnekler için hala büyüktür, ancak çapraz doğrulama kümenizdeki anormalliklerde küçük olur. Demek bu kadar. Bu haftanın sonuna kadar benimle kaldığın için teşekkürler. Umarım hem kümeleme algoritmalarını hem de anormallik tespit algoritmalarını duymaktan keyif alırsınız. Ayrıca uygulama laboratuvarlarında bu fikirlerle oynamaktan da keyif alıyorsunuz.

Gelecek hafta, tavsiye sistemleri hakkında konuşmaya devam edeceğiz. Bir web sitesine gittiğinizde ve size ürün, film veya başka şeyler önerdiğinizde. Bu algoritma aslında nasıl çalışır? Bu, makine öğreniminde şaşırtıcı derecede az konuşulan ticari açıdan en önemli algoritmalardan biridir, ancak gelecek hafta bu algoritmaların nasıl çalıştığına bir göz atacağız, böylece bir dahaki sefere web sitesine gittiğinizde anlayacaksınız ve sonra size bir şeyler önereceksiniz. Belki de böyle olmuştur. Olduğu gibi, kendiniz için de bunun gibi başka algoritmalar oluşturabileceksiniz. Bu yüzden laboratuvarlarla eğlenin ve gelecek hafta sizi görmeyi dört gözle bekliyorlar.