

Açıklama Üretim sektöründe aktif faaliyet gösteren bir şirkette çalıştığınızı farzediniz. Üst yönetim üretim hattının verimini arttırmak ve olası robot arızalarını önceden tahminlemek istiyor. Üretim hattında 3 robotun olduğunu düşünelim. A robotu → B robotu → C robotu Robotlar sırasıyla ürünü alır, işlemini yapar ve ürünü bırakır.

Sorular

1 Üretim hattının verimini arttırmak ve olası robot arızalarını önceden tahminleyebilmek için nasıl bir verisetine ihtiyacınız olurdu?

- İstedığınız verisetinde bulunan verilere/kolonlara neden ihtiyaç duyduğunuzu açıklanmanız ve birkaç satırlık veriseti örneği eklemeniz beklenmektedir.

2 Üretim hattının verimini arttırmak ve olası robot arızalarını önceden tahminleyebilmek için nasıl bir algoritma kurardınız?

- Algoritmanın adımlarını ve adımların açıklamalarını yazmanız beklenmektedir.

3 Kurduğunuz algorithmadan çıkan sonuçların üst yönetime real-time görselleştirilmesi için nasıl bir teknolojik mimari tasarım oluştururdunuz?

- Herhangi bir diyagram çizme uygulaması (ÖRN:LucidChart) kullanarak tasarımınızı çizmeniz ve adımları açıklamanız beklenmektedir

Problemi Denetimli öğrenme ile çözmeye karar verdik. Buna ek olarak da data setini ve gelen datalar olası hataları yakalayamama ihtimaline karşı Anomaly detection (Anormallik Tespiti) adında makine öğrenmesi algoritması ile de algorithmaya fayda sağlamak istedik. Bu kapsamda her soru ikiye bölünüp anlatılacaktır.

1.Soru

Denetimli öğrenme

- Literatür araştırılması yapıldığında birçok farklı türde robot olduğu görülmüştür. Burda verimliliği artırmak için 7 farklı robot türü için data seti oluşturulmuştur.
- Burda her bir robot türü için ayrı ayrı motor arızalanma ve sonrasında verimliliğe etkisi hesaplanıp en ideal robot seçimi yapılması öngörülmüştür.
- Literatür araştırmaları sonucunda: Robotların bozulmasını yağ miktarı, robot çalışma süresi, çalışma sıcaklığı, bir üründe harcanan süre, çalışma frekansı, çalışma gücü etkileyeceği görülmüştür.
- Bu sebeple bu etkenleri ölçen sensörlerden anlık olarak gelen veriler kullanılmıştır. Motor bozulmadan 1 saat önceki tahmini verilerdir.
- Robotun 1 saat önceden hangi sensör/sensörlerden dolayı bozulacağı ihtimalini tespit edip önceden önlem alınması için bu şekilde düşünüldü.
- Data setinde oluşturulan veriler gerçek değerleri göstermemektedir. Olası değerleri göstermektedir. Burda amaç ne türde bir data setine ihtiyaç duyulduğunu göstermektir.
- Output olarak her sensörden kayıplı bozulma olup olmadığını belirleyecektir. Aynı anda birden fazla sensörden olumsuz veri gelebilir.
- **1:**Sensörün bozulmaya sebep olacak bilgisi, **0:** Sensörün bozulmaya sebep olamayacağı bilgisi
- Eğer tahmini 1 saat öncesinden hangi sensörün bozulacağı tespit edilebilirse verimliliği artırmak için bakımları yapılır ve motorun çalışması sorunsuz devam edebilir.
- Hattının verimi: Minimum sürede maksimum ürün üretme değildir. Daha uzun vadede robotun bozulmaması ve üretim hızı etkenlerinin birleşiminden oluşan bir terimdir.
- Bu sebeple uzun vadede verimi arttırmak için 6 farklı sensör verilerine göre robotun bozulmadan çalışmasını sağlayacak bir mekanizma kurulması ön görülmüştür.
- Bu çalışma yapısını 7 farklı robot için yapıp uzun vadede en verimli olabilecek robot seçilebilir.
- Bu sistem sorudaki A robotu , B robotu , C robotu için ayrı ayrı denemelidir. Çünkü sadece bir robotun bile çalışmaması üretimi durduracaktır.

Veri ön işleme için gerekli adımlar:

- Her kolon 0-1 arasına normalize edilmelidir. Çünkü her kolonun çıktıya etkisi eşit olması istenmektedir.
- Veri setindeki boş(None) değerler var mı kontrol edilmelidir.
- Bu boş değerler aynı özellik içindeki diğer verileri ortalaması veya medyanı ile doldurulabilir ya da silinebilir, sebebi aykırı (Outlier) değerler olmasından dolayı başarı oranını indirmesidir.
- Boş olmadığı halde aykırı (Outlier) değerler olabilir bunlar ise quartile ile ayıklanır.

Destekliyi Makine Öğrenmesi (Anormallik Tespiti)

- Olası robot arızalarını önceden tahminleyebilmek için veriseti denetimli öğrenmedeki(supervised learning) gibi etiketli olamaz çünkü bilinmeyen hatalar bütünü bu yüzden denetimsiz öğrenme(unsupervised learning) yolu kullanılmalı, gelecekte olası hataları kestirmek ve engellemek için anolmaly detection en etkili yoldur. Anormallik algılama algoritmaları, normal olayların etiketlenmemiş bir veri kümesine bakar ve böylece olağandışı

veya anormal bir olay olup olmadığını algılamayı tespit etmeyi öğrenir. Robot örneğinde de mesela x_1 özelliği sıcaklığı ve x_2 özelliği ise titreşim yoğunlu olsun.

- Öğrenme algoritması, robotların ne kadar ısı üretildiği ve ne kadar titreştikleri açısından tipik olarak nasıl davrandığına dair geçmişten m örneklerinin elimizde olsun.
- Anomali tespitini gerçekleştirmenin en yaygın yolu, yoğunluk tahmini adı verilen bir tekniktir. Bunun anlamı, m örnekten oluşan eğitim kümeleri olduğundan, yapılan ilk şey, x 'in olasılığı için bir model oluşturmaktır. Başka bir deyişle, öğrenme algoritması, x_1 ve x_2 özelliklerinin yüksek olasılığa sahip değerlerinin neler olduğunu ve veri setinde görülme olasılığı daha büyük veya daha düşük olan değerlerin neler olduğunu bulmaya çalışacaktır. Bu x_1 ve x_2 özelliği örnek olarak verilmiştir çok daha fazla örnek ekleyip olasılık yoğunluk fonksiyonları oluşturulabilir.
- x_1 'den x_m 'ye kadar bir eğitim seti var, burada her x örneğinin bitiş özellikleri var. Yani, her x örneği sayılar içinde bir vektördür. Robot örneğinde, ısı ve titreşimlere karşılık gelen iki özelliğimiz vardır.

2.Soru

Denetimli öğrenme

- Bir makine öğrenimi algoritması oluşturmada etkili olmanın anahtarı, zamanın nereye harcanacağı konusunda iyi seçimler yapmaktır.
- Makine öğrenimi sistemi oluşturma sürecinin anahtarı, performansını artırmak için bir sonraki adımda ne yapılacağına nasıl karar verileceğidir.
- Bir öğrenme algoritmasının bias ve variance (overfitting) bakmanın, bir sonraki denemeniz konusunda size çok iyi rehberlik sağlar.
- Bunu anlamamanın yolu algoritmanızın eğitim setindeki ve çapraz doğrulama setindeki performansına bakmak olacaktır.

Veri ön işleme adımından sonra,

1)

ilk önce veriyi eğitim (train), geliştirme(development) ve test datası olarak 3'e bölmek gerekir.

Verinin miktarına göre: Eğer çok az veri varsa %60 train, %20 development, %20 test seti olarak ayırırdık. Eğer verinin miktarı çok fazla ise %95 train, %2.5 development, %2.5 test seti olarak ayırırdık. (Bu yüzdeler verinin boyutuna göre değişebilir.)

2)

Mimariyi deeplearning ile kurmak daha iyi sonuçlar verecektir. Bu yüzden tensorflow veya pytorch kullanarak yapay sinir ağı modeli oluştururduk.

Mimamiride;

- **Dense:** Nöroların oluşturulduğu katmandır,
- **BatchNormalization:** Her batch de Normalize yaparak her nöronun ağırlığını dengelemektedir,
- **Activation:** relu, sigmoid, softmax gibi aktivasyon katmanları (genelde ara katmalarda relu, output katmanında sigmoid veya softmax tercih edilir),
- **Droupout, Regularization** gibi katmalar eğitim datası üzerinde aşırı öğrenmeyi (overfit etmeyi) önlemek için eklenebilir.
- **Çıktı katmanında (Dense Layer)** birden fazla output beklendiği için activation olarak softmax kullanmalıdır.
- **keras.Callback fonksiyonu** ile istenilen accuracy elde edilince eğitimi durdurabiliriz.
- **keras.callback.LearningRateScheduler** ile yaklaşık 100 epoch eğitip en ideal learning rate seçimi yapardık.
- **Optimizasyon algoritması** olarak başlangıçta "adam" algoritması kullandık. Optimizasyon algoritması hiperparametre olduğu için gerekirse diğer optimizasyon algoritmaları da denenebilir.
- **loss parametresi** olarak 'categorical_crossentropy' kullandık. Çünkü categorical_crossentropy her kategori için olası eşleşmeyi içeren bir dizi (One-Hot-Encoding) üretir.
- **metric olarak 'accuracy'** kullandık. Çünkü accuracy "doğru tespit edilen tahmin" / "tüm tahminler" olduğu için probleme en uygun metric budur.

Tüm parametreler belirlendikten sonra modeli eğitirdik

3)

- MLOps adı verilen büyüyen bir alan var. Bu, Makine Öğrenimi İşlemleri anlamına gelir. Bu, makine öğrenimi sistemlerinin sistematik olarak nasıl oluşturulacağı ve dağıtılacağı ve sürdürüleceği pratiğini ifade eder.
- Makine öğrenimi modelinizin güvenilir olduğundan, iyi ölçeklendiğinden, iyi yasalara sahip olduğundan, izlendiğinden emin olmak için tüm bunları yapmak için ve ardından, modelin iyi çalışmasını sağlamak için uygun şekilde güncellemeler yapma fırsatınız olur.

MLOps kapsamında sistemi daha iyileştirmek için aşağıdaki adımları gerçekleştirdik:

- **Train datası** için elde edilen accuracy ve loss sonuçlarına bakarak yorum yapardık.

Eğer train datası istenildiği kadar öğrenilmemişse bias yani yeterince öğrenmeme problemi vardır ve bunu çözmek için dense layer sayısı, katman sayısı, batchsize, epoch sayısı öğrenme katsayısı vb hiperparametreler üzerinden değişiklik yaparak istenilen seviyede accuracy elde etmeye çalışırdık.

- Daha sonra **geliştirme datası** üzerinde elde edilen accuracy ve loss sonuçlarına bakarak yorum yapardık.

Eğer geliştirme datası istenildiği kadar öğrenilmemişse yani train datası üzerinde aşırı öğrenme (overfit) yapmış ise bu problemi çözmek için Droupout, Regularization gibi katmalar ekleyip veya data seti artırılıp istenilen seviyede accuracy elde etmeye çalışırdık.

- Daha sonra **test seti** üzerinde elde edilen accuracy ve loss sonuçlarına bakarak yorum yapardık.

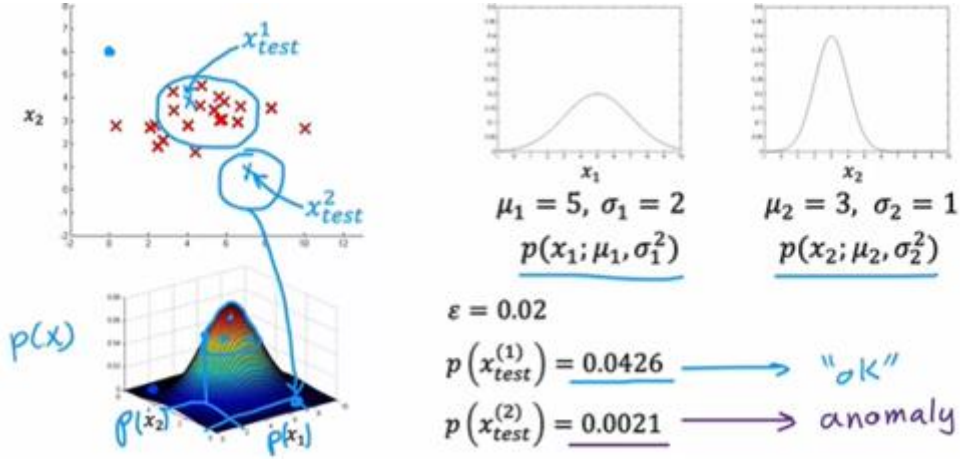
Eğer test datası istenildiği kadar öğrenilmemişse yani geliştirme datası üzerinde aşırı öğrenme (overfit) yapmış ise bu problemi çözmek için geliştirme data setini arttırıp bu sorunuda çözerdik.

- Paralel olarak Denetimli mimari başarıya ulaştıktan sonra real time olarak sistemi çalıştırıp test ederdik. ML Destekleyici sistemde paralel olarak çalışıp sistemde bir sorun olup olmadığını tespit edip veri bilimciye raporlar buna göre data setinde ve veride hata olup olmadığını tespit eder.
- Daha sonra farklı bir sistemide paralel olarak çalıştırdık. İlk iki sistem real time olarak çalıştırdık. Üçüncü sistemi ise real time çalışmanın daha iyi sonuçlar vermesi için sistemi Transfer learning mantığı gibi elde edilen ağırlıklar (weights) ve mimariyi kullanarak; yeni gelen datalar ile mimariyi baştan eğitip elde edilen sonuçları değerlendirip eğer daha iyi sonuçlar elde edilirse eski ağırlıkları yeni ağırlıklar ile gücelleyebiliriz.

Böylece dinamik bir yapıya sahip MLOps bir sistem ile sürekli daha iyi sonuçlar elde edebiliriz.

Destekliyi Makine Öğrenmesi (Anormallik Tespiti)

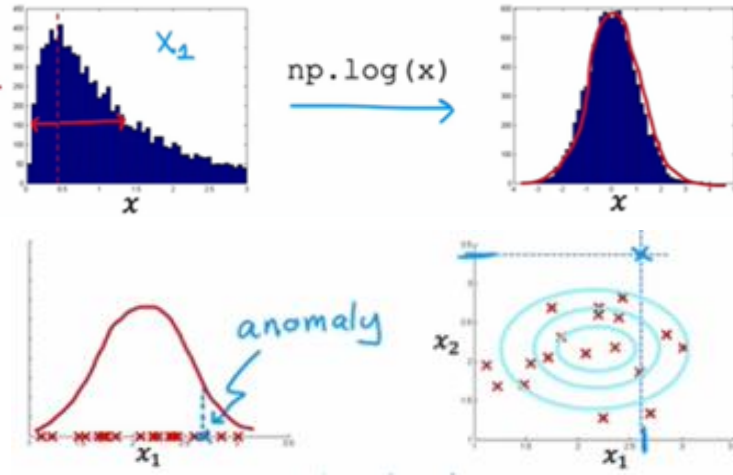
- Anormallik tespitini uygulamak için normal dağılım olarak da adlandırılan Gauss dağılımının kullanılması gerekir. Yapılmak istenen yoğunluk tahminidir ve bunun tek anlamı, bir model kurulacak ve $p(x)$ için olasılığı tahmin edilecek. $p(x)$ için modelimiz şu şekilde olacaktır, x, x_1, x_2 ve x_n 'ye kadar devam eden değerlere sahip bir özellik vektörüdür. Her değer için olasılığı hesaplanır, her bir değer için olma olasılığı istatistiksel olarak birbirinden bağımsızdır. Aslında algoritma mantığı olasılık yoğunluk fonksiyonlarının dağılımına göre olası hataları tespit etme üzerine kuruludur. Mesela x_1 ve x_2 için olasılığını bilip anormaly olup olmadığını şöyle anlayabiliriz;



- Olası robot arızasını tespit sistemini nasıl oluşturabileceğinizi görmek için hepsini bir araya getirilmesi gerekir. İlk adım, anormal örneklerin göstergesi olabileceğini düşündüğünüz özellikleri (x_i) seçmek olacaktır; yukarıdaki $p(x_2)$ değeri gibi değerler. Bu sayede etiketlenmemiş eğitim setindeki bu parametreleri tahmin ederek, artık modelimizin tüm parametrelerini hesaplamış oluyoruz. Bir olasılık sonucunun anomaly olup olmadığını threshold ile seçebiliriz.
- Anomali tespitinin iyi çalışıp çalışmadığını **Threshold** değeri belirler, bu değer seçimi önemlidir. Algoritmayı bir özelliği değiştirmek veya bir parametreyi değiştirmek gibi bir şekilde hızlı bir şekilde değiştirebilirsek ve algoritmanın daha iyi veya daha kötü olup olmadığını bize söyleyen bir sayı hesaplama yöntemine sahipsek (cross validation test), o zaman algoritmanın daha iyi veya daha kötü olup olmadığına karar vermeyi çok daha kolaylaştırır. Bu değişikliğe almaya bağlı kalmak için anomaly tespitinde genellikle bu şekilde yapılır. Anomaly tespit sisteminin geliştirilmesi için birkaç adım da uygulanabilir mesela belki birkaç robot çalışmasını yaptıktan sonra, anormal olan birkaç robot görüldü ve anormal olduğunu bildiğiniz örnekler için, bu anormalliği belirtmek için y eşittir 1 etiketini ilişkilendirelim ve normal olduğunu düşündüğümüz örnekler için ise y etiketi 0'a eşit olur.
- Anomali tespit algoritmasının öğreneceği eğitim seti hala x_1 'den x_m 'e kadar olan bu etiketsiz eğitim setidir ve tüm bu örnekleri sadece normal ve anormal olmadığını varsayacağımız örnekler olarak düşünebiliriz, bu yüzden y eşittir 0. Geçmişteki yapılan robotlardan elde edilen anormal örneği varsa, çapraz doğrulama kümesi oluşturabilmek için çok yararlı olacaktır, bu sayede hem çapraz doğrulama hem de test setlerinin birkaç anormal örnek içerdiği bazı örneklerden oluşan bir test setine sahip olmuş oluyoruz. Özetle anormal olan bazı örnekler varsa, **ancak yanlışlıkla y eşittir 0 ile etiketlenmişse, anormallik saptama algoritması iyi çalışacaktır.**
- Örnekleme gerekirse Threshold değerini seçmek için denetimli öğrenme (supervised learning) kullanılır bu threshold değerini ise denetimsiz öğrenme (unsupervised learning) için kullanılır. Threshold değerinin önceki elde edilen etiketlemiş anomaly ve normal verilerle çapraz doğrulama testi ile deneme yanılma yoluyla seçebiliriz, anomaly bildiğimiz etiketleri bulunduğu zaman threshold parametresini bulmuş oluruz daha sonra ise Gauss dağılımları ile etiketsiz veriler üzerinde robotun iyi çalışıp çalışmadığını ısı ve titreşim yoğunluk fonksiyonlarına bakarak olası hataları tespit ederiz, biraz karışık gözüküyor ama basitçe tekrar etmek gerekirse; ısı ve titreşim verilerinden oluşan bir etiketsiz veri setimiz var ancak bunlar arasında korelasyonda neye göre anomaly veya normal olacağını kestirmek için threshold değerini bilemiyoruz, bu değeri bilmek için ise geçmişteki anomaly ve normal etiketli veriler ile çapraz doğrulama testi ile threshold değerini bilebiliyoruz bu sayede gelecekte farklı hiç

görülmemiş değerler ile olasılık yoğunluk fonksiyonları oluşturup threshold ile anormaly tespiti yapıp **olası robot arızalarını önceden tahminleyebiliriz.**

- Burada dikkat çekmek istediğim nokta robot arızasının tespitini denetimli öğrenme ile yapabiliriz ancak **olası hatalar** için geçmişteki hataların bilinmesi gelecekteki hataları kesin olarak vermeyecektir bu yüzden anomaly detection kullanılacaktır.
- Bir anormallik tespit algoritması oluştururken, iyi bir özellik seçimi seçmenin(bizim varsayımız da sıcaklık ve titreşimdi) önemlidir. Denetimli öğrenmede, özelliklere tam olarak sahip değilsek veya sorunla alakalı olmayan birkaç ekstra özellik varsa, bu genellikle iyi olur. Algoritmanın denetimli sinyal vermesi gerektiğinden, algoritmanın hangi özelliklerin yok sayıldığını veya özelliğin nasıl yeniden ölçeklendirileceğini ve ona verilen özelliklerden en iyi şekilde yararlanabileceğini anlaması için yeterli etiket vardır. Ancak, yalnızca etiketlenmemiş verilerden çalışan veya öğrenen anomaly tespiti için, anomalinin hangi özelliklerin göz ardı edileceğini anlaması daha zordur. Bu yüzden, özellikleri dikkatlice seçmenin, anormallik tespiti için denetimli öğrenme kadar önemlidir.
- Seçilen özellik etiketsiz verisi normal dağılım veya sağ\sol yanlı gauss olmalıdır, yanlı gauss verileri matematiksel denklemlerle normal dağılıma çekilebilir.



3.Soru

