

[Enes Fehmi Manan]: Ben Enes Fehmi Manan. Bu yayın boyunca da moderatör olarak burada yanımda. Eğitim konusunda oldukça deneyimli bir isim var, Engin Deniz Alpman. Hoş geldiniz Engin Hocam.

[Engin Deniz Alpman]: Merhaba.

[Enes Fehmi Manan]: İsterseniz Hocam, biz sizi tanımakla başlayalım. Ondan sonra da ders içeriği nasıl olacak, ona kısaca konuşup, soru varsa ufaktan yayının başında soruyu alıp, ardından da ben arkaya geçebilirim.

[Engin Deniz Alpman]: Tamam, olur bu formatta. Canlı sorabilirler, yani bilmiyorum formatı nasıl ama canlı olarak ben anlatırken de devam edebilir veya direkt chat'ten sorularını iletebilirler. Sorular geldikçe, siz chat'ten okuyunca cevap veririm.

[Enes Fehmi Manan]: Hızlıca aksiyon olması gereken bir şey varsa ben de araya girip iletebilirim.

[Engin Deniz Alpman]: Olur, tamam. Ben kendimden bahsedeyim. Yaklaşık 10 senedir Yapay Zeka alanında kurumsal firmalara ve bireylere eğitim, danışmanlık ve baştan sona proje desteği veriyorum. Bu baştan sona proje ve danışmanlık kısmı bireysel değil, daha çok yani kurumlarda oluyor ama eğitim kısmı birey ve kurumlarda. Son 1,5 senedir de kendi firmam RuneLab'de bunu yapıyorum. İsteyenler bizim firmaya da bakabilirler.

Bugün birlikte makine öğrenmesi ve veri bilimi nezdinde istatistik konularına bakacağız. Yani çok uzun bir eğitim değil tabii, o yüzden yani uzmanlaşmayı beklemek burada pek doğru olmaz ama böyle bir kıvılcım yakmak, "Aa bak, nerelerinde kullanılıyor bu makine öğrenmesi, işte veri bilimi, onlara birazcık bakmak, merak uyandırmak" üzerine bir eğitim.

[Enes Fehmi Manan]: Teşekkür ederiz diyeyim. Chat'ten de ben herhangi bir soru görmüyorum akışla alakalı. O zaman ben yavaştan arkaya geçeyim. Sizin ekranınız varsa ekrana vereyim onu.

(Teknik ayarlamalar hakkında kısa diyalog)

[Engin Deniz Alpman]: O zaman başlayalım. Ekran geliyor mu?

[Enes Fehmi Manan]: Geldi, ben ekrana veriyorum. Ekranı verdim şu anda.

[Engin Deniz Alpman]: Tamamdır.

[Enes Fehmi Manan]: Ben arkaya geçiyorum o zaman.

[Engin Deniz Alpman]: O zaman, "Makine Öğrenmesi İçin İstatistik" diye bir başlığımız var. İstatistik neden gerekli makine öğrenmesi, veri bilimi alanında? İstatistik bilmek nasıl bir fayda sağlayabilir bize? Hangi alanında, hangi aşamasında?

Bir en başta bir spoiler vermiş olalım, zaten ana durum bu olacak: **Gerekli değil!** En azından en başında diyerek böyle bağlayabiliriz. Makine öğrenmesiyle alakalı bir şeyler öğrenmek istediğiniz, veriyle alakalı bir şeyler yapıyorsunuz, modelleme yapmak istiyorsunuz, bir makine öğrenmesi modeli eğitip bunu işte "deneyeceğim, acaba iyi tahmin edebiliyor mu?" dediğinizde en başta gidip Lineer Cebir, matematiksel işte formüller, istatistik öğrenmenize gerek yok.

Buna da şöyle bir analogiyle yaklaşabiliriz: Diyelim ki futbol öğreniyorsunuz, basketbol öğreniyorsunuz daha çocukken. Girip size birileri şeyi anlatmıyor daha oynarken, yani "Topa vuracaksın, arkadaşla paslaşacaksın, kaleye attığın zaman işte bir sayı

yapıyorsun, fazla sayı yapan kazanır, eşit sayıda iki grup olur" gibi küçük bir anlatım oluyor. Avantaj, ofsayt var, işte dışarıya çıkması var, şu var, bilmem nesi var olaylarını zaman içinde öğreniyorsun. O yüzden en başta sana bir insan bütün ayrıntıları verse... Oynarken ben size söylerim demez de, en başından bir sürü bilgi verse, çoğu insan hiç en başta oynamadan basketbolu, futbolu veya herhangi başka bir oyunu bırakıp gider. Çoğu insanın ben bunu fark ettim, bu makine öğrenmesiyle başlayınca matematiksel, istatistiksel durumlarla "En başta hepsini öğrenmem lazım, bu formüller falan" diye bir şeye girip ondan sonra hevesi kaçıp oyuna küsmüş oluyor.

O yüzden istatistik tabii ki de gerekli, ama en başta başlarken ben önermiyorum.

Ben kendimin şu şeyinden de bahsetmedim; en sevdiğim şeylerden biri matematiksel formülleri sıfırdan çıkarmak, üzerine düşünmek falandır. Yani matematiği sevmiyorum ve size bunu söylüyorum değil. Güzel, tabii ki de öğrenin ve yararlı ama en başta başlarken değil. Çünkü bunun doğru sıralanması o değil.

Yaptık, makine öğrenmesi modelimizi kurduk, bundan bir sonuç aldık. İlk modellerimiz oldu, baktık "Evet, classification böyle yapılmış, sınıflandırma böyle, regresyon böyle yapılmış." Tamam, yoluma devam ediyorum. Sonra, "Ben bunu nasıl daha iyi hale getirebilirim?" ve "Ne kötü gitti?" sorularını sormaya başladıkça istatistiğin önemi gelmeye başlıyor. Oralarda gerçekten "Neden böyle oldu?", "Nelere dikkat etmeliyim?", "İstatistiksel olarak neler var buralarda?" olayları ön plana çıkmış oluyor.

Üretimdeki çoğu makine öğrenmesi hatası, birinin Random Forest yerine XGBoost seçmesinden dolayı değil tabii ki de. Seçim önemli, ama bunlardan daha çok karşılaştığımız neler vardır? "Ben işte şeyi seçmiştim, XGBoost'u, Random Forest'i seçmiştim ama model bir anda kötülemeye başladı." Büyük ihtimalle model drift oluyordur. Ne demek bu? Senin üzerinde eğittiğin verin - bunları ayrıntılı konuşacağız zaten, şu an böyle bir özet - üzerinde modelini eğittiğin veriden, gerçek hayatta görmeye başladığın şeyler farklılaşıyor. Yani bu belki makroekonomik bir durum oldu, COVID oldu, şu oldu, bu oldu, değişti, ekonomi değişti... bir şekilde iki distribution (dağılım) birbirinden farklılaşabilir. Sen hangi modeli seçersen seç, bundan kaçamazsın. Bu istatistiksel bir şey.

Eğitim verin yanlı (biased) mı? Değerlendirme metriklerini doğru seçtin mi, yanlış mı yorumluyorsun? Veriyi doğru böldün mü? Veri bölmek zaten direkt senin verinin istatistiğiyle alakalı bir şey. Bu durumlar geldiği zaman ve daha çok birçok daha istatistiksel durum geldiği zaman, benim modelimin optimal olmadığını, olabileceğinden daha kötü olduğunu görebiliyorum. Ama bunu ilk modelinizi kurarken düşünmeye gerek yok.

En başta yapmanız gereken, benim önerdiğim şey, elinizi kirletmek.

(Köpek analojisi bölümü)

Sol tarafta köpek var. Bu köpeği tanımlamak için bir sürü şey söyleyebilirsiniz. Bir canlı diyebilirsiniz, bir hayvan diyebilirsiniz, bir köpek diyebilirsiniz... Gitgide bu arkadaşı özelleştirebilirsiniz. Ne olduğunu daha böyle ayrıntılı ayrıntılı tanımlayarak daha açıklayıcı bir şey söyleyebilirsiniz.

Şimdi bunun ne alakası var? Bizim elimizde bir veri olduğu zaman, biz bununla alakalı analizler yapacaksak, en başta benim bu veriyi tanımlamam için de... ben bu arkadaşla ilk baktığım zaman, sağ taraftaki arkadaşla baktığım zaman, ilk başta

diyeceğim şey "Bu bir veri." E tamam, nasıl bir veri? Ortalaması sallıyorum 2.5 olan bir veri. Ama dağılımı dağınık olabilirdi, bitişik olabilirdi...

Benim burada gözlemlediğim veriyi bir daha ayrıntılı hale getirebilmek için burada ne kadar dağıldığını da söylemem lazım. Bu da standart sapmanın gelmesiyle oluyor. Veriyle alakalı belli istatistiksel özellikleri telaffuz ederken, "ortalaması buymuş, standart sapması buymuş, normal dağılımdan gelmiş" falan gibi şeyler derken, aslında yaptığımız şey bu olayı ayrıntılandırmak. Tıpkı "Bu bir canlı, bu bir hayvan, bu bir köpek, bu bir Golden Retriever" gibi.

Ayrıntılandırmak... Başkası senin verine baktığı zaman veya sen kendi verinle alakalı analizler yaptığın zaman, nasıl bir şeyle muhatap olduğunu bilmen lazım. Yani bir uğur böceğinin seni ısırması gariptir ama bir köpeğin ısırması garip değildir. Neden bunu söylüyorum? Sen böyle bir veri dağılımından böyle bir şey bekleyebilirsin ama öbüründen böyle bir şey gelse "ilginç" veya "garip" diye adlandırırsın. Ama bunları "garip" diye adlandırmak veya "normal" diye adlandırmak için sizin olayı isimlendirmeniz, ayrıntılandırmanız lazım. "Bunun ortalaması bu, bunun standart sapması budur" gibi.

(Model ve Fonksiyon bölümü)

Dağılımlar, gitgide modellerimizi iyileştirme çabası içerisinde girdiğimiz zaman önemli oluyor diye söylemiştim. Neden? Bu model ve veri birbirine tamamen bağlı. Makine öğrenmesinde... hayatta her şey bir fonksiyon. Siz bana chat'ten bir soru soruyorsunuz, Enes bana aktarıyor, ben size cevap veriyorum. Bana bir input geldi, ben size output veriyorum. Bizim makine öğrenmesinde yaptığımız şey de bu fonksiyonumuzu veriler üzerinden eğitmek ve biz buna "model" diyoruz. Ama bu aslında bir fonksiyon ve bu fonksiyonun oluşturulması verileri görerek oluyor. Normal klasik programlama paradigmasında sen buradaki input'u output'a dönüştürmek için satırları tek tek kodla yazarken, makine öğrenmesinde veriler üzerinden öğrenmesini sağlıyorsun.

Ama bu modelin günün sonunda yaptığı şey şu: Senin verilerinin dağılımını öğrenmesi. "Nasıl veriler geliyor, bu dağılımdan gelen şeylerin neresindesin ki bu duruma ben nasıl bir cevap veriyorum?" Direkt pattern matching yapıp gerçek hayatta gösterdiğin durumlara cevap vermesi üzerinden gidiyor. O yüzden benim modelim aslında gidip veri dağılımı üzerinde yaşayan bir fonksiyondur. **"Model, veri dağılımının üzerinde yaşayan bir fonksiyondur"** diye düşünebilirsiniz. Arkadaşın yaptığı başka bir şey yok. ChatGPT'ye de gitsen, Gemini'ye de gitsen, prediktif modellere de baksan, günümüzdeki modeller tamamen veride gördüğü paternleri, daha sonra sorduğumuz şeylere o paternleri match edip ona göre cevap veren yapılar. Başka bir şey değil. O yüzden sizin bu modeli nasıl bir veri üzerinde eğittiğiniz, nasıl şeyler üzerine test ettiğiniz, gerçek hayatta nasıl şeyler olduğunun hepsi ama hepsi bu "dağılım" şeyine geliyor.

(Enes ile diyalog ve EDA yaklaşımı üzerine)

[Enes Fehmi Manan]: ...dağılımına bakarım... aykırı değer... betimsel istatistiklere bakabilirim... Varyansına bakabilirim... Ben sabaha kadar bakarım orada onu söyleyeyim.

[Engin Deniz Alpman]: Enes'le burada biraz farklı bir bakış açımız var. Bunun doğru veya yanlış yok. Benim yaklaşımım biraz daha "deli dana gibi dan dun dan dun" batıp, direkt eğitip, ondan sonra "Neler yanlış gitmiş?" diye geri dönüp bakmak. Enes daha en başından bakıp, ondan sonra öyle daha sağlam ilerliyor. Ben genelde ilk yaptığım şey, çok böyle EDA'ya bakmadan model kurup, ondan sonra "Neler yanlış gidiyor, bunu nasıl daha iyi hale getiririm, hangi şeyler bunu bozuyor?" gibi yerlere bakmak.

(Standart Sapma ve Varyans'ın sıfırdan çıkarımı)

...standart sapma, ortalama falan gibi deęerlerin üzerinden bir geeeđim hızlıca. Bu standart sapma tam olarak neyi veriyor, normal dađılıma nasıl bađlanıyor gibi onlardan bir bahsedeeđim.

(Overfitting ve Veri Bölme)

Siz daha önceki oturumlarda veriyi bölme, overfit falan gibi konulardan bahsetmiş miydiniz? O da çünkü verdiđin verinin işte bias'a gitmesi falan gibi durumları etkilediđi için konuşmak isteyeceđim bir yer ama daha önce konuşulduysa...

[Enes Fehmi Manan]: Şöyle, çok detaylı bahsedilmedi. Siz gene bahsedebilirsiniz. Hani veri split noktalarında, hatta probleme göre de split olayı deđiştii için...

[Engin Deniz Alpman]: Tamam, onlardan bir kısa bahsedeeđim.

(Sezgisel Öğrenme Felsefesi)

Benim genel bir yaklaşımım... sezgisel eğitim diye bir isim kullanıyorum. Bir olayı sezgisel olarak öğrenme. Bu sadece şu an benim size söylediđim istatistiksel durum deđil. Bir şey öğreniyorken, içerisinde formüller varsa, öğrenmeniz gereken mantıklar varsa, bunu sezgisel olarak öğrenin. Formülü ezberleyerek öğrenmeyin. Gerçekten, içine sinecek şekilde, yani kemiklerinize işleyecek kadar öğreneceksiniz temeline kadar. Üniversitelerde genelde övülen bir şey vardır, hoca geldi tahtaya bir sürü şey yazdı, bir sürü formül... "Bu işte kadın veya adam çok şey biliyor." falan. Yok. Yani büyük ihtimalle bilmediđi için sadece... Yani olayın temelini anlamakla formül ezberlemek birbirinden farklı şeyler.

(Varyans formülünün mantıksal çıkarımı)

Dađınıklığı nasıl bulurum? Dađınıklık ne demek? Yani bir referans noktamız var, deđil mi? Ortalamaya göre bu arkadaşlar nasıl dađılıyor? O zaman ortalamaya göre bu arkadaşların uzaklığına bakmam lazım. Uzaklığa bakacaksam ne yapabilirim? Bu arkadaşları toplayıp ortalamasını alınca olmuyor. Neden? Pozitif, negatiflikten birbirini götürüyorlar. Peki bundan kaçmak için ne yapabiliriz? Ya mutlak deđerini alabiliriz ya da karesini alabiliriz.

Neden ben mutlak deđer almıyorum da bu arkadaşın karesini alıyorum? Burada birden çok durumdan bahsedilebilir. Bir tanesi, matematiksel olarak analiz etmek istediđiniz zaman... bir fonksiyonun türevini alabilmeniz için o fonksiyonun her yerde türevlenebilir olması lazım. Mutlak deđerin fonksiyonuna bakarsanız, sıfırda türevlenemiyor... O yüzden bir şeyleri öğrenirken nedenini öğrenmek önemli.

(Büyük farkların ağırlığı - Usain Bolt analojisi)

Büyük farklar daha çok ağırlık kazanır karede demiş Tolga. Bu gerçekten önemli... Ortalamadan uzaklaştıkça benim burada attığım adımlarda daha bir önemli oluyor. Yani benim buradaki birim, aslında ortalamanın yanındaki durumlardan daha önemli bir hale geliyor. Ben bu olayın önemini formüle işlemem lazım. Karesini aldıđın zaman lineer olmayan bir şekilde gitmiş oluyor.

(n-1 Düzeltmesi - Örneklem vs Popülasyon)

Benim elimde aldığım örneklem ve popülasyon için tahmin ettiğim durum arasında farklar olacaktır. Ben popülasyonun ortalama ve standart sapmasını hesaplamak isterdim. Ama benim elimde örneklemin standart sapması var. Şeyi görmüşsünüzdür, estimation yaparken $1/n$ yerine $1/(n-1)$ gibi alındığını görebilirsiniz. Şu formülü minimize eden şeyin, buradaki X'lerin ortalamasının buraya yazılması... Popülasyon ortalaması bundan daha farklı olacağı için, standart sapmanın aslında bir underestimation'ı oluyor. Bu underestimation'ı gidermek için $n-1$ 'e bölüyoruz. Daha küçük bir sayıya bölersen daha büyük bir şey elde edersin.

(Overfitting ve Model Karşılaştırması - Düz vs Kıvrımlı Çizgi)

Elimde şöyle veriler var. Model dediğimiz şey fonksiyon. Bir fonksiyon, bir model fit ettiğin zaman... aradaki farkı minimize etmek istiyorum. Şimdi size sorum: Model 1 mi daha iyi, Model 2 mi?

[Chat'ten Gelen Yorum]: Model 2'de overfitting gibi duruyor.

[Engin Deniz Alpman]: Çok büyük ihtimalle overfit var, evet. Neden? Çünkü benim elimdeki bütün verinin tamamen üstünden geçmiş. Ama %100 emin olamayız. Ne zaman emin oluruz? Test ettiğimizde.

(Ezberci vs Öğrenen Öğrenci Analojisi)

Senin modelini eğitim yerinde test edemezsin. Bu neye benziyor biliyor musun? Sen öğrenciye cevap anahtarlarıyla birlikte bir test veriyorsun... Öğrenci 1 dedi ki "Ben bunların cevaplarını ezberleyeceğim." Öğrenci 2 de diyor ki "Ben cevapları ezberlemeyeceğim, kendim öğrenip yapmak istiyorum." Sadece şuna baksanız, öğrenci 1 daha iyi dersiniz. Ama gerçekten sınava gittiğimiz zaman... öğrenci 2 daha iyi. Neden? Çünkü ezberlemeden öğrenmişti.

(Train/Validation/Test Ayrımı)

Benim bunu bir şekilde bölmem lazım. Verinin görüldüğü yerle, eğitildiği yerle test edildiği yerin birbirinden farklı olması lazım. Train, eğitildiği yer; Test, test edildiği yer gibi düşünebilirsiniz. Veriyi bölerken %80-%20 bölmek zorunda değilsiniz... verin çok büyüdüğü zaman... %99 Train, %0.5 Validation, %0.5 Test gibi şeyler görebilirsiniz.

Yaptığımız olay şu: Deneme yanılma yapmak. Bir sürü model deniyorsun. Model 1, 2, 3, 4 denedim... Ama ben bu test setindeki error'a veya metriğe göre seçtiğim zaman, ben burada dolaylı bir overfitting yapıyorum. Neden? Çünkü "Ağaç sayısını arttırmak error sayısını azaltıyor ya, ben o zaman bu ağaç sayısını 20 yapayım." Sen test setinde model sayılarının, ağaç sayısının artmasının iyi geldiğini görmüş oluyorsun. Buradan bir bilgi sızdırmış oluyorsun.

O zaman benim izole edebilmem için bunu ne yapmam lazım? Benim elimdeki veriyi üçe bölmem lazım: **Train, Validation, Test.**

Modeli sen Train setinde eğitirsin, Validation setinde deneyip seçersin... Test setindeki metriklerden memnun kalmayıp "Ben başka bir modeli de Test'te deneyeyim" olayı tamamen ama tamamen yanlış bir yaklaşım.

Bu üç arkadaş bize ne yapıyor?

- **Train Seti:** Modelin eğitildiği yer.
- **Validation Seti:** Modelin seçildiği yer.
- **Test Seti:** Gerçek hayat performansının saptandığı yer.

Bazen kötü bir model accuracy'si raporlamak da iyidir... Senin burada amacın pembe gözlük takıp iyi gibi görünmesi değil. Senin gerçek hayatta bu arkadaşın nasıl bir performans yaptığını saptayabiliyor olman lazım. Bitti. Yapacak başka hiçbir şey yok. Ama sen aynı Train, Validation ve Test veri setiyle böyle bir iterasyon daha yapamazsın. Yaparsan sadece kendini, müşterini, çalıştığın yeri kandırmış olursun. Hiçbir işe yaramaz.

Senin burada yapabileceğin şey şu: Test setini yeniden oluşturmak.

Benim test setim gerçek hayatı yansıtsın, değil mi? Gerçek hayatta bir dağılımım varsa, benim de test setimde buna yakın bir dağılımdan veriler almam lazım. Senin gerçek hayat verisinin dağılımına en çok benzetmeye çalıştığın yerin Validation ve Test seti olması lazım. Ne zaman ki olay ciddileşti, ben gerçekten bu olayı iyi hale getireceğim, bak yine istatistik kapısını çalman gerekiyor.

Aynı test setiyle birden çok deneme kesinlikle ama kesinlikle yapmayın.

[Engin Deniz Alpman]: Şimdi bu dağılım seçme olayında bir iki tane örnek vereceğim. Cross-validation falan gibi durumlar da var, o da her zaman uygulanabilecek şeyler değil ama her zaman uygulanmaya çalışılan şeyler maalesef.

Nedir bu? Bir Random Split. Bölme, veri bölme. İlk başta bir rastgele bölmek... Bir senaryodan gidelim.

(Kedi-Köpek Uygulaması Senaryosu)

Telefon uygulaması geliştiriyoruz. Kedi ve köpek fotolarını sınıflandıracacağız. Elimde telefondan çekilmiş 1000 kedi, 1000 köpek fotoğrafı var. Elimde 2000 tane veri var.

Şimdi, aklıma şöyle bir şey geldi: "Ne kadar fazla veri o kadar iyi." Soru var mı bu arada?

[Chat'ten gelen sorular ve cevaplar üzerine kısa diyalog]

Telefon uygulaması geliştiriyoruz. Kedi ve köpek fotoğraflarını sınıflandıracacağız. Telefonda çekilmiş... Evet, ne kadar fazla veri varsa o kadar iyi. Yani ne kadar fazla veri beslerseniz modeli, o kadar paterni daha iyi öğrenir.

Aklımıza şöyle bir şey geliyor belki: "Ya benim elimde 1000 tane kedi, 1000 tane köpek fotoğrafı var. Güzel, tamam, hoş. Ya şöyle bir şey yapsam?" Gidip Google'a "cat photo" yazdım, zilyon tane şey... "dog photo" yazdım, zilyon tane de köpek... Bunlar da kedi-köpek fotoğrafı. Yani bunları da ben alayım, kullanayım. Burada artık benim elimdeki veriler çok zenginleşmiş oldu. Ekstradan ben buraya 10.000 kedi, 10.000 köpek... Ne yaparım? Elimde şu an 22.000 tane veri var. "Ne yaparım mesela? Gerçekten bunu %80, %10, %10 bölerim. İşte %80'ini Train'e ayırıyorum, %10'unu Validation'a ayırdım, öbür %10'unu Test'e ayırdım. Rastgele böldüm mesela. Bunda bir sorun var mı sizce?" diye sormuş olayım. Elimde bu arkadaşları birleştirdim, sonra rastgele böldüm. Tamam mıdır bu olay?

(Kısa bir sessizlik ve cevap bekleme)

[Enes Fehmi Manan]: ...cevaplar hızlı gelmeye başlıyor.

[Engin Deniz Alpman]: Kedi-köpek oranları önemli. Oran aynı şu an, balanslı bir veri. Zaten veri dengeli olduğu için rastgele böldüğünüz zaman aşırı dengesiz bir şey beklemezsiniz.

Tespit etmeye çalıştığımız şey şu: Kedi ve köpek fotoğraflarını sınıflandıracamız. Telefonunuzdan kedi-köpek fotoğrafı çekiyorsunuz, bizim modelimiz diyor ki "Bu bir kedidir, abi bu köpektir."

(Ekran paylaşımı üzerine kısa diyalog)

Şimdi, asıl benim ortaya çıkarmak, size göstermek istediğim olay bu değildi. Oransal olarak bölmede bir sorun yok. Sıkıntı suydur ki... Gerçek hayatta böyle şeylerle çok karşılaşsınız. O yüzden "Gerçekte ben neyi göreceğim?" olayı... Google'dan indirdiğiniz fotoğrafların, telefonda çekilmiş olduğunu garantileyebilir misiniz? Asıl soru bu.

Çünkü ben günün sonunda telefonda çekilmiş kedi-köpek fotoğrafı tahmin edeceğim. Siz burada kedi-köpek olarak, şu profesyonel bir makineden çekilmiş gibi duruyor. Ayrıntıları daha fazla, derinliği daha fazla, renkleri daha iyi. Modelin işini kolaylaştırıyor veya... farklı bir dağılımı öğretiyor olabilir. Neden? Yine istatistiğe geldik. Telefonda çekilmiş kedi-köpek fotoğrafının istatistiği, dağılımıyla; profesyonel makineden çekilmiş... pixel özellikleri, resolution'ı... dağılımı bunlardan dolayı farklı.

Ben bu iki dağılımı birbirine karıştırıp bölersem... Train biraz farklı olabilir demiştim, çünkü o da kedi, o da köpek fotoğrafı. Ama gerçek hayatı simüle ediyorsan... "Gerçek hayatta nasıl bir cevap alacağım?" ve modeli seçerken de onlarda iyi olmasını istiyorsan, **Validation ve Test, senin gerçek hayatını yansıtmaları lazım**. O yüzden Validation ve Test'e sen bunları, hiçbir şekilde bu 10.000 taneyi sokmaman lazım. Burası önemli. Bu 10.000 tane kedi-köpek fotoğrafını Train'de kullanabilirsin ama Validation ve Test'te kullanamazsın. O yüzden Validation ve Test'in kesinlikle sadece bu [telefonda çekilen 2000 fotoğraftan] oluşması lazım.

Sadece telefonda olmayabilir çünkü. Sadece bunlardan Validation ve Test bölebilirsin. O yüzden rastgele bölme, hayır. Rastgele bölme ne zaman? Tamam, 2000 tanesini topladım buraya sadece Validation ve Test'e dağıtacağım diyelim. Bu senaryoda bu 2000 tanesi içerisinde rastgele 1000 tanesini Validation'a, rastgele 1000 tanesini Test'e koyabilirsin. Ama kesinlikle ama kesinlikle bu 10.000 tane kedi-köpeğin Validation ve Test'e girmemesi lazım.

(Veri Artırma ve Dağılım Benzetme Sorusu Üzerine)

[Enes Fehmi Manan]: Trende kullandım, bu sefer hani test ve validasyona bir şekilde bunu sokma ihtimalim hiç mi yok mesela?

[Engin Deniz Alpman]: Çok güzel bir soru oldu. Orada "augment etmek" olayları... Bunu yapabilirsin ama bunu yapması biraz... orada da yanlışlıkla bir overfit'e sebebiyet verebilirsin. Neden? Çünkü senin elindeki sample'ın distribution'ı ile... gerçek hayatta senin beklediğin distribution öyle olmayabilir. Ekstradan onu ezberlemiş olabilirsin. Yani baktın, oradan gelen verilerde de bir sorun yoksa... Bu Validation ve Test süreci, "Abi modeli bir kere yaptık, böyle kalacak, kimse bir şey demesin" gibi bir durum yok zaten. Kademeli olarak yapıp bir daha üzerinden geçilebilen bir şey.

(Emlak Tahmini Senaryosu ve Zamana Dayalı Bölme)

Bir örnek daha vereceğim. Diyelim ki emlak tahmini yapıyorum. Benim elimde işte 2015 - 2025... 110 bin tane veri var. Burada siz nasıl bölme yaparsınız? Train, Validation, Test'i nasıl ayırırsınız? Burada normal bildiğimiz default cross-validation yapmamız doğru olmaz. Nedenine geleceğim.

Neye dikkat edilebilir? Cevaplar geldi... Cevap doğru bu arada da, neden öyle olması gerekiyor? Bunların hepsi içerisinde olabilecek şeyler ama ana bir tane nedeni var ki o en önemlisi. O da şu: **Model, eğitildiği yerden sonrasını tahmin eder gerçek hayatta.**

Benim ekrana getirebilirsin... Ben daha ileride bu ev satılsa ne kadar olur diye tahmin ediyorum. Ben burada rastgele yaparsam, Train'e kırmızıyla çizeceğim... şu Train verisi oldu, validasyon burada, test verileri burada. Burada garip bir şey oldu. Neden? Çünkü normalde benim beklediğim şey şu: Ben sonrasını tahmin ediyorum. O yüzden ne olması lazım? Gerçek hayat simülasyonu yaparsın Validation ve Test'te. O zaman benim Validation ve Test'im, modelin eğitildiği Train verisinden daha güncel olması lazım.

Ne zararı olabilir? Matematikte interpolasyon ve ekstrapolasyon diye iki tane kavram vardır. Senin bu iki Train noktasını biliyor olarak, arasındaki bir noktayı tahmin etme ihtimalin... çok daha yüksek. Bu interpolasyon oldu. Ama elindeki veriler bu kırmızılar, dışında bir şey tahmin ediyorsan, bu ekstrapolasyon olayı. Benim yapmak istediğim bu. Ama benim modelim şu an interpolasyon da yapıyor ve işi daha kolay. Ben modelimi şu an böyle değerlendiriyorum ve hiç sağlıklı değil. Bir gün sonra nasıl satıldığını bilip bir gün önce nasıl satıldığını bilersen, o gün nasıl satıldığını sandviç yapıp bulman çok daha kolay.

İnsanlar şunu yanlış düşünebiliyor: "Ya benim sonuçlarım çok iyi çıktı ama..." Pembe gözlük takıyorsun. Burası çok önemli. Validation ve Test'te iyi sonuç almak önemli değil. Biz burada "Ne kadar iyi sonuç almışsın, kutlayalım" demiyoruz. Gerçek hayatta çıktığın zaman sen nasıl bir şeyle karşılaşacaksın, onu sana veriyor. Kötüyse onu çözmen lazım. Ve bu durumu gerçek hayatı yansıtacak şekilde yapmanız lazım. "Validation, Train, Validation, Test olarak yapınca daha iyi sonuçlar alıyorum." Tabii ki de daha iyi alırsın. Test'e çünkü dolaylı olarak overfit etmişsin.

Olay burada daha iyi sonuç alman değil. Test'in, daha önce görmediği bir yer olarak alınıp simülasyon yapması.

(Market Kasası Veri Kaybı Örneği ve Problem Tanımı)

Bir market var, kasiyerler bazı günlerde kasayı kapatmayı unutuyor. Ayın 10'u var, 12'si kayıp... 13'ü kayıp. Sen burada amacın bir sonrakini, daha sonrakini tahmin etmek değil. Burada sana diyorlar ki "Bu ay bittikten sonra biz sana vereceğiz bu veriyi, sen aradıkları tahmin et." Şunu göstermek istiyorum: Her zaman en güncelleri oraya alman lazım falan gibi, probleme göre değişen bir şey. Sen problemde ne ile karşılaşacaksın? Gerçek hayatı yansıtıyor mu tam olarak?

(Beş Önemli Kavram: Aykırı Gözlem, Eksik Veri, Bootstrap, Feature Importance, Partial Dependence)

Şimdi birkaç tane şeyden... Bütün hepsine zaman kalacak mı bilmiyorum ama... Birkaç tane bahsetmek istediğim şey şu: Aykırı gözlem, eksik veri, bootstrap, feature importance, bir de partial dependence.

Aykırı Gözlem (Outlier): Outlier, verinin normal trendinin dışında olan... Bu sizin verilerinizi bazen saptırır. Model bu aykırı gözlemleri de öğrenelim derken bir yerlerde saçmalayabilir. Atabilirsin, ama bu aykırı gözlemler senin için önemli de olabilir. Önemli bir bilgi olabilir. O yüzden sen bu aykırı gözlemi normalize edip, belli bir değere çekip... aynı zamanda şöyle yapabilirsin. Diyelim ki X1 feature'ı... 500 diye bir outlier var. Burada şöyle bir şey yapabilirsiniz... Bu 500 yerine gittin mesela 9 yazdın. Gidip yeni bir feature yaratıp X1_outlier, False/True diyerek yoluna devam edebilirsin. Bu 9'la bu 9 aynı olmasın. Bu 9 outlier olmayan bir 9'du, bu 9 outlier'dı. Bununla alakalı bir etiket koymak modeli daha iyi yerlere getirebiliyor. Aklınızda tutabilirsiniz bir tip olarak.

Eksik Veri: Eksik veriler için de aynısını söyleyeceğim. Çok yapılan yanlışlardan bir tanesi... 1-2 değer boş diye bütün satırı atmak, olan verileri de elden kaybetmenize sebebiyet veriyor. Bunun yerine bir şey yazma, imputation durumlarına gidebiliyorsunuz. Yine durum aynı, yukarıda anlattığım aykırı gözlem durumuyla... X1 diye veriler var... burada bir boşluk var. Ben bunun yerine gittim ortalama 7 imiş diyelim, 7 yazdım. Ama bununla bu aynı değil. Ben burada bir imputation yaptım. Belki bunun boş bırakılması da bir anlam ifade ediyor olabilir.

Şöyle bir soru sormuş olayım: Sokakta çıkıp anket yapıyoruz. İnsanların aldığı maaşları sorduk. Bazı insanlar cevap vermiyor. Sizce kimler cevap vermiyor?

[Chat'ten Gelen Cevaplar Üzerine]: Çok güzel, fazla para kazananlar, yüksek maaş alanlar... bir grup daha var aklımda benim... Aykırı değerler, çok güzel. Çok yüksek ve düşükse diye gelmiş zaten. Aynen öyle. Peki çok yüksek ve düşük olanlar aslında nedir? Outlier gibi düşünebilirsiniz. Şimdi ben burada ortalamasını yazınca olay biraz garip değil mi? Boş bırakan kişi outlier'dı, ben onun yerine normal bir değer yazdım. O yüzden bazen boş değerlerin yerine trendin orta değerini yazmak tamamen yanlış bir durum olabiliyor. Ne yapıyorsun o zaman? Modeline eksik veri olduğunu söylemen lazım. Çünkü bir

[Engin Deniz Alpman]: Eksik veriler için de aynısını söyleyeceğim. Aslında bu ikisinde de aynı olan bir durum. Eksik verilerle alakalı, çok yapılan yanlışlardan bir tanesi... çok yapılan yanlış da demeyeyim de, bir row'un (satırın) %99'u boşsa atmak fena olmayabilir belki ama işte 1-2 değer boş diye bütün satırı atma durumu, "Hani boş veri var, hepsini atayım" durumu, olan verileri de elden kaybetmenize sebebiyet veriyor. O yüzden veri kaybı yaşamamak için bunun yerine bir şey yazma, imputation (değer atama) durumlarına gidebiliyorsunuz.

Durumuna gittiğiniz zaman da yine durum aynı, hani bu yukarıda anlattığım aykırı gözlem durumuyla... Diyelim ki X1 diye veriler var, 10 var... Ben bunun yerine gittim, ortalama işte 7 imiş diyelim, tamam mı? Ben bunun yerine 7 yazdım ya da medyan... Ben o yerine 7 yazdıktan sonra bununla bu aynı değil. Yani ben burada medyan yazdım tamam ama bu değer gerçek değeri belki 25'ti, bilmiyorum ki. Bir de belki bunun boş bırakılması da bir anlam ifade ediyor olabilir. Ben oraya sadece ortalamayı, medyanyı yazarak bu olayı batırıyor olabilirim.

Mesela size şöyle bir şey sorayım, bunu yine bir soru-cevapla şey yapalım. Yine beni aşağıya alabilirsin... Şöyle bir soru sormuş olayım: Sokakta çıkıp anket yapıyoruz, insanların aldığı maaşları sorduk tamam mı? Bazı insanlar cevap vermiyor. Sizce kimler cevap vermiyor genellikle?

(Chat'ten gelen cevaplar üzerine)

Evet, çok güzel, fazla para kazananlar, yüksek maaş alanlar... bir grup daha var aklımda benim... Aykırı yerler, çok güzel. "Çok yüksek ve düşükse" diye gelmiş zaten. Aynen öyle. Peki çok yüksek ve düşük olanlar aslında nedir? Outlier gibi düşünebilirsiniz, değil mi? Genel trendin dışında...

Şimdi yine ekranı alabiliriz. Şimdi ben burada ortalamasını yazınca olay biraz garip değil mi? Yani insanların... diyelim ki bu maaşlar... Sallıyorum... Ya zaten boş bırakan kişi outlier'dı, yani trendin dışındaydı. Ben onun yerine normal bir değer yazdım. O yüzden bazen boş değerlerin yerine trendin orta değerini yazmak tamamen yanlış bir durum olabiliyor. Ne yapıyorsun o zaman? Yapacak bir şey de yok, o değer boş çünkü. Şöyle bir şey yapabilirsin: Öbür değerlerden bunu tahmin eden bir model oluşturabilirsin... Ama belki öyle bir şey yok. Senin modeline, eksik veri olduğunu söylemen lazım. Çünkü bir verinin eksikliği de bana bir şey anlatıyor olabilir. X1_missing diye bir şey oluşturup, burada False, True falan diye devam edebilirim. Ben

bunun True/False kısmını yapıp, ondan sonra bunun yerine doldurması şeklinde yapabilirim. Model böylece buraya bir değer yazmış ama aslında bu "missing" olayını da biliyor olabilir.

(Enes Fehmi Manan ile Eksik Veri Flag'lerinin Önemi Üzerine Diyalog)

[Enes Fehmi Manan]: ...Flag atama metodunun gerçekten önemli geldiği noktalar oluyor mu? Flagleme temalı bir modeli çıkardınız mı mesela?

[Engin Deniz Alpman]: Oluyor tabii ki de oluyor... Eksik verilerin önemli olduğu çok durum vardır. Yani insanların belli bir paterne göre... mesela demin gördüğünüz durumda... hayatta görebileceğiniz bir sürü durum var. Aslında o eksikliğin bir patern ifade ettiği bir durum. Veya işte kendi kişisel özelliğiyle alakalı bir şey söylemeyen insanlar daha böyle anonim takılmayı seven... ve anonim takılmayı sevenlerin de belki aralarında bir korelasyon vardır ve belli bir ürüne daha fazla harcama yapıyor olabilirler... gibi bir yere de, saklı bir paterne de doğru gidebilir.

Ama oralarda şöyle yerlerde sıkıntı yaşıyoruz... Flag eklediğim durum mu daha fazladır, feature importance'ı az çıktığı durum mu daha fazladır dersin, importance'ı az çıktığı durum daha fazla. Orada onun nedeni de şu... kayıp olduğu için o değer, yani tamamen %100 bir şey diyemiyorsun. Tek bir yerden gelmeyebilir... Kişiler gelebileceği için, günün sonunda sen onları sadece bir yere indirgediğin için belirleyici olmayabiliyor.

[Enes Fehmi Manan]: ...datanın boyutu küçükse, oradaki boş verileri atmaktansa "Ben bir flag atayım, gene o paterni model öğrenmeye çalışsın" bir bilgi anlamında...

[Engin Deniz Alpman]: Zaten makine öğrenmesi, veri bilimi bunun üzerine. Yani biz bir fotoğrafını aldık bu olayın. Ama onun tam olarak bireysel bazda nasıl yaşandığı, o verinin bana nasıl ulaştığı, neden boş olduğu... bunları bilemediğimiz için... deneme yanılma durumu.

(Bootstrap Kavramı)

Şeyden devam edelim, Bootstrap'ten devam edelim. Bootstrap'i belki duymuşsunuzdur... Bu arkadaşın yaptığı şey aslında veriyi çoklama ile alakalı bir durum... Bu da böyle gökten inmiş falan gibi gelmesin arkadaşlara. 10.000 tane verim var, ben bunun üzerinde 10 tane ağaç eğitmek istiyorum... Ben burada 20 tane ağaç eğitsem 500'er tane veri vereceğim, 40 tane eğitsem 250'şer tane veri vereceğim diye gitgide bu ağaçların gördüğü veri sayısı azalmaya başlıyor. Bu iyi bir şey değil. Ben bu ağaçların gördüğü şeyleri aynı tutup, sayıyı aynı tutup ama aynı zamanda farklı şeyler de görüyor gibi... farklı uzmanlıklar oluşturup bunların consensus'unu yaptırabilir miyim?

Veri, bir dağılım gibi düşünebilirsiniz. Bootstrap'in mantığı o zaten. Benim elimde diyelim ki böyle bir veri var... [7, 5, 2, 2, 5, 2, 4]. Bu dağılım... baktığınız zaman size söylediği şey aşağı yukarı nedir? Bunun frekansı, görünme frekansı. 7'nin görülme oranı nedir? 1/7. 2'nin görülme oranı nedir? 3/7...

Bu neye bağlayacağım? Sen elindeki veriyi... veriyi "çoklamak" ne demek? Verinin özünü koruyarak çoklamak, aynı distribution'a uyması demek ama farklı olması demek. Değerlerin biraz farklı gelmesini istiyorum, değil mi? Senin bu veriyi alıp, bunların içerisinde rastgele değerler seçebilirim... İlk ben sample'ımı oluşturuyorum, Sample 1. Rastgele çektim, 5 geldi. Bir daha rastgele çektim, yine 5 geldi... Bu benim birinci sample'ım. Burada ben ne yapmış oldum? Benim elimdeki veriden ben bu arkadaşları seçmiş oldum...

Ya Bootstrap'te şöyle bir varsayım var: Benim elimdeki veriyi ben distribution olarak sayıyorum. Çünkü elimdeki tek veri o. Bir popülasyon distribution'ından gelen bu. Ben popülasyon distribution'ı olarak davranacağım verime ve verimin içerisinde sample seçer gibi ben verimin içinden seçeceğim. Bootstrap'in yaptığı başka hiçbir şey değil.

(Normalizasyon ve Standardizasyon Üzerine)

[Chat'ten gelen soru]: Normalizasyon mu, standardizasyon mu, hangisi hangi algorithmada kullanılmalı?

[Engin Deniz Alpman]: Standardizasyonu anlatayım... Şöyle düşünebilirsiniz: "5 dakikada geldi. Geç mi erken mi?" Neye göre, kime göre deriz? Aynen, ortalamaya göre karar veririz, referans noktasına göre. 5 dakikada bir çay geldiyse "geç" diyebilirsiniz. Pizza geldiyse "hızlı"... Ben bu iki uzaydaki gecikme ve erkenlik durumunu nasıl standardize edebiliriz?

Diyelim ki ortalama çay gelme süresi 1 dakika, standart sapması 2 dakika. Pizza için ortalama 25 dakika, standart sapması 5 dakika. Şöyle bir şey diyebilirim... Ne kadar normalden uzak bir şey... Benim ortalamam 1 dakika. Ben 3 dakikada geldiğim zaman şaşırma faktörü ne kadar? Bir standart sapma... Yani bir standart sapma uzaktayım. Aynı şaşırma faktörünü nasıl aktarabilirim pizzaya? Yani sen bir çayı 7 dakika beklemen ile bir pizzayı 75 dakika beklemen aynı gariplik seviyesinde gibi bir standardizasyona gidebilirsin.

Ne yapmış olduk? O an gözlemlediğim değerden ortalamayı çıkartıp standart sapmaya bölersen, sen bu uzayları aynı uzaya çekersin. Standardizasyon, aynı uzaya çekmektir. Derin öğrenmenin ilk katmanında falan yapılmasının nedeni de o. Oda sayısı, parka olan uzaklık, metrekaresi... bunlar kendi uzayları. Şu an gözlemlediğim durum, normalde o metrekaleden kaç şaşkınlık daha fazla? Bunların aynı uzayda olması lazım çünkü çarpılma, bölme işlemlerine giriyor parametrik modellerde.

Bu standardizasyon gibi durumlar aynı uzaya çekmek için. Normalizasyon da yapılabilir bazı yerlerde ama normalizasyon aynı uzayı çekmek değil, range'ini (aralığını) ayarlamak için yapılan bir durum.

(Curse of Dimensionality - Boyutluluk Laneti Üzerine)

[Chat'ten gelen soru]: Çokça farklı feature (özellik) durumu olursa ne olacak?

[Engin Deniz Alpman]: ...Curse of Dimensionality diye bir durum var. Senin 3 tane verin tek boyutta bayağı dolgun duruyor. Sen bunu ama 2 boyuta alırsan... 5 metrede 3 verim var, 25 metrekarede 3 verim var. Daha seyrek görünmeye başlıyor. Ve daha az olduğu zaman da verin, işte model bunları ezberleme eğilimine gidebiliyor...

(Büyük Veri ile Çalışma ve Örneklem Alma Üzerine)

[Chat'ten gelen soru]: Veri çok büyük... model eğitmek saatler sürüyor... nasıl bir sürü model deneyeceğim?

[Engin Deniz Alpman]: Senin bütün verinle modeli eğitmek zorunda değilsin. Bu da çok nedense konuşulan bir şey değil... Benim elimde 10 milyon veri var. Türkiye'de kadın ve erkeklerin boy ortalaması var. Kimse bana boyumu sormadı... Eminim ki buradaki çoğu kişiye sorulmamıştır. Peki bu nasıl oluyor? Örneklem alıyorsun. Senin popülasyonunu yansıttığını düşündüğün bir örneklem alıyorsun.

Bizim bu durumu, model eğitirken de yapabiliriz. Senin elinde 10 milyon veri var. Sen şöyle bir şey yapabilirsin: Bu 10 milyon içerisinde belki 1 milyon veri aldın. Bu 1 milyon veriyle ben bunu eğittim ama validasyonda bakıyorum... Model seçerken relatif olarak bakıyoruz ya, şöyle bir şey yapabilirsin. Model 1, 2, 3, 4'ün accuracy'si kaç çıktı? Ben 1 milyonda tamam az çıktı

da, model seçerken "hangisi daha fazla çıktı?"ya göre seçiyorum. Eğer elimdeki verinin bir sub-sample'ını aldın ve o sub-sample senin sample'ını yansıtıyorsa... en iyi çıkanı alıp... Model 3 en iyi çıktı, Model 3'ü ondan sonra 10 milyonluk veride eğitirsin.

(Enes Fehmi Manan ile Optimal Özellik Sayısı ve Modeli Production'a Çıkarma Üzerine Diyalog)

[Enes Fehmi Manan]: ...kaç feature production'a çıkarıyorsun? ...model büyüdükçe oradaki inference (çıkartım) süresinden de bir maliyeti var. Buradaki optimal sayıyı nasıl buluyoruz?

[Engin Deniz Alpman]: Yapılabilecek bir sürü hareket var. Orada bir feature importance'ına bakılabilir... az olanları atıp bir bakılabilir... Belki o 1000 taneyi bir anda 700'e indirdin ve çok da bir şey değişmedi... Feature'lar arasında korelasyon var mı... Kolinerite denilen şeyler oluyor mu... aynı şeyi ifade edenlerden tek tek atıp bakabilirsin... Ne kadar az feature ile buna ulaşırsan, sen oradaki cost'unu, inference cost'unu, computation harcamanı... daha az olacak.

Bir de sadece hesaplama cost'u değil. Ne kadar fazla feature olursa, senin gerçek hayattaki değişkenlerin değişiminden etkilenme ihtimalin artar. Yani modelin drift olma ihtimali daha fazla. O yüzden ne kadar sadeleştirirsen o kadar iyi.

Benim yaklaşımım, genellikle eklenmesinin mantıklı olduğu şeyleri ekleyip sonra eksilterek gitmek. Çünkü gıdım gıdım gidersem, bir araya geldiğinde önemli olabilecek durumları da kaçırabilirim. A'yı ekledim daha iyi olmadı, B'yi ekleyeyim... ama belki A ve C bir aradayken bir önem kazanıyordu, kaçırdım ben o durumu. O yüzden varken eksiltmek... benim daha çok sevdiğim bir yaklaşımım.

(Normal Dağılım Varsayımı ve Gerçek Hayat Verisi Üzerine)

[Chat'ten gelen soru]: Farklı özneliliklerin farklı farklı dağılımları olması modeli olumsuz etkiliyor mu? Her zaman çarpık dağılımları normal dağılıma uydurmak gerekiyor mu?

[Enes Fehmi Manan]: Ben de istatistik mezunuyum. Okulda bu normal dağılım çokça bahsedilen bir şey... Ama kişi gerçek hayata geldiğinde, tırnak içerisinde o kadar da önemli olmadığını görüyorum gibi diyeyim.

[Engin Deniz Alpman]: Normal dağılım bize şunu söyler: Bir tane ortalama değer var, en çok görülen değer. Sen bundan uzaklaştıkça daha az gözüktür. Doğadaki birçok şey buna aşağı yukarı uyuyor. Ama normalden farklı olan şeyleri de siz ekstra bir takla atarak normale geçirebilirsiniz. Gidip ortalamalarının dağılımını çizerseniz...

Feature'ların dağılımları zaten gerçek hayatta çoğunlukla farklı oluyor. Bunları standardize edip aynı uzaya çekip, ondan sonra "şaşıрма faktörü" olayıyla devam ediyorsun... Standardizasyon, modelin en baştaki işini kolaylaştırmış oluyor. Bir modeli, bir bias katmadan işini ne kadar kolaylaştırırsan, o genellikle o kadar iyi öğrenir.

(Kapanış Konuşmaları)

[Enes Fehmi Manan]: Bayağı uzun bir ders oldu, şu an 2.5 saati devirdik.

[Engin Deniz Alpman]: Süper, umarım hoşunuza gitmiştir. Benim diyeceğim başka bir şey yok.

[Enes Fehmi Manan]: ...chat'te çok güzel yorumlar var. Gerçekten ağızınıza sağlık. Benim de dinlerken çok keyif aldığım bir yayın oldu. Özellikle böyle başında bir istatistiğe gittim, dersine gittim. Oradan bir machine learning'e gittim, oradan bir staj eğitim zamanlarına gittim. Böyle bir gittim geldim, sektöre gittim falan. Gerçekten inanılmaz güzel bir yayındı.

[Engin Deniz Alpman]: Süper, ben de çok keyif aldım. O zaman herkese iyi akşamlar. Ben teşekkür edeyim, size de dinlediğiniz için teşekkür ederim. Herkese iyi akşamlar dilerim.

[Enes Fehmi Manan]: Görüşmek üzere arkadaşlar, iyi akşamlar.