Zero2End ML Stage 2: “ETA Prediction”

Konuşmacı: Özge Usta

Moderatör: Enes Fehmi Manan

(Özge Usta):

Bu problem aslında bir girişim problemi; burada biz bu süreyi tahmin etmeye çalışıyoruz. Çok farklı kullanım alanları var ama ben temel olarak kendi çalıştığım alanlardan biraz size bahsedeceğim.

İşte birisi **Ride-sharing** dediğimiz, yani bu yolculuk paylaşımı hizmeti veren şirketler. Üzerine çalıştıkları en büyük problemlerden biri bu. En büyük oyuncuları dünyada **Uber**, **Lyft** ve Çin pazarından **Didi**. Türkiye’de de şu an **Şirket 1** aslında bu işi yapıyor.

Tabii burada bu 3 büyük oyuncu... Uber zaten dünyanın birçok yerinde var, Didi daha çok Çin pazarında, Lyft de aynı şekilde büyük firmalardan biri. Tabii bu firmalar bu **ETA Prediction** (Tahmini Varış Süresi) işini çok detaylı **Deep Learning** (Derin Öğrenme) modelleri üzerine kurguluyorlar. Mesela ben buraya birkaç tane örnek koydum, sunumla paylaşırım. Buna böyle çok detaylı girmeyeceğim ama en azından bir görmeniz açısından göstermek istiyorum.

Benim bildiğim kadarıyla sadece bir paylaşımcı araç işinde değiller; aynı zamanda **Uber Eats** var, şu anda da zaten **Şirket 2** satın aldı ve Türkiye’de de bu aktivitelerini (Şirket 2) yapıyorlar. Bu tarz farklı işler için aslında bir ETA Prediction yapıyorlar. Bu **Paper**'dan (Makaleden) aslında, "Deep ETA" diye bir modelleri var. Bu modelin makalesi de var, blog yazısı da var; detaylı merak ederseniz inceleyebilirsiniz.

Çok kısa göstermem gerekirse; burada aslında bayağı detaylı bir **Deep Learning** modeli kullanıyorlar. Şuradan çok kısa bahsetmem gerekirse: Bir harita düşünün ve bu haritada aslında bir sürü ufak ufak lokasyonlar var. Siz bir yerden bir yere giderken mesela Google size bir rota çiziyor. Bu her bir rotanın içinde ufak ufak parçalar var. Genelde bu rotanın parçalarını böyle hepsini bölerek bir Deep Learning modeline veriyorlar ve her bir parçanın aslında süresini tahmin etmeye çalışıyorlar gibi çok detaylı modeller çalışılıyor.

Burada tabii Uber gibi büyük firmaların avantajı bu işleri çok uzun yıllardır yapıyor olmaları. O yüzden de biz bugün bu kadar böyle Deep Learning gibi detaylara girmeyeceğiz. Ben daha çok size **Tree-based** (Ağaç tabanlı) modellerden bahsedeceğim.

İkinci olarak şundan bahsedebilirim: Neden böyle bir şeye ihtiyaç duyuyorlar? Neden bu süreyi tahmin etmek istiyorlar? Tabii ki de bunu Google’ın API’ları da yapıyor, Apple’da da var; bu süreleri çekebiliyorsunuz, trafikli sonuçları çekebiliyorsunuz. Ama bu çok **maliyetli** bir iş. Öyle olduğu için de, sizin çok büyük bir veriniz olduğunda; mesela diyelim ki saniyelik yüz binlerce çağrı geliyor ve o gelen sürücüleri yolcularla eşlemeye çalışıyorsunuz. Böyle bir durumda bu ETA için sürekli Google’dan istek atmak çok maliyetli bir şey. O yüzden de bir şekilde kendi modelleriyle yapıp bunu daha az maliyetli hale getirmek istiyorlar.

Burada da tabii bu ETA Prediction'lar üzerine çalışıyorlar ve bunu çok kısa sürede de yapmaya çalışıyorlar. Hatta Didi’nin bununla ilgili makalesinde modellerini nasıl, atıyorum bir saniye içinde tahmin alabildikleri, hatta milisaniyelerden

bahsediyoruz... Yani çoğu zaman mesela bir saniyede tahmin almak uzun bile olabiliyor. Dediğim gibi saniyeler içinde belki yüz binlerce onlara istek geliyor ve bir anda sonuçları çıkartmaları gerekiyor.

Bu yayında bu detaylara o kadar girmeyeceğiz ama bunlar da gerçekten çok ilginç **challenge**'lar (zorluklar) oluyor. Aslında data scientist (veri bilimci) model yapmanın da ötesine çıkıp bu tarz süre challenge'larını da bir şekilde halletmeye çalışıyorsunuz. Burada farklı altyapıları araştırıyorsunuz; işte **Teacher-Student** modelleri nasıl daha hızlı çalışır, nasıl daha hızlı prediction alabilirim, nasıl bunları yansıtabilirim? Eğer bir uygulaman varsa vesaire, farklı sorulara da cevap vermeniz gerekiyor. Tabii bu Data Scientist olarak tek başınıza olmuyorsunuz bu süreçlerde ama sizin de illaki bu altyapıları öğrenmeniz, uygulamanız gerekiyor. Özellikle canlıya alma (deployment) tarafında; mesela model için **feature**'a (özelliğe) ihtiyacınız var. Bu feature'ı da bir şekilde bir veri ambarından çekmeniz gerekiyor. Bunun olabildiğince hızlı olması çok önemli. Yani siz buradan 5 saniyede veriyi çekerseniz, isterseniz saniyeler içinde tahmin edin; o veriyi yavaş çektiğiniz için bunun hiçbir anlamı kalmıyor.

Kısa şehir, uzun şehir farklarından bahsettim. Bu ikisinin aslında biraz da böyle ufak farklarından da bahsetmek istiyorum. Ben işte dediğim gibi Şirket 2'dayken de ilk zamanlarda bu etap tahminleri üzerine çalışmıştım. Burada tabii Şirket 2'nun en büyük avantajı kısa mesafeler olduğu için; yani mesela bir kurye bir restorana atanacak, oradaki yemeği alıp siparişin verildiği yere götürecek. Böyle bir durumda tabii böyle 3 kilometrelik ufak bir çapın içinde gittiği için bazen işte trafik verisi vesaire bu kadar önemli olmayabiliyor. Ama siz çok daha **long distance** (uzun mesafe) hesapladığınız durumda; mesela işte akşam 20.00'de Beşiktaş'tan Kartal'a gidecek bir aracın ETA'sını hesaplamak istediğinizde, orada mesela trafik verisi çok önemli olabiliyor, hava durumu verisi çok önemli olabiliyor.

(Burada tahtaya çizim yaparak anlatıyor)
Uzun mesafe ve kısa mesafe hesabı oluyor. Onu şu şekilde anlatabilirim: İşte atıyorum burada bir sürücümüz var... Daha sonrasında bir yolcumuz var burada. Şimdi yolcunun gitmek istediği bir **Destination** (varış) noktası, bir final noktası var. Şimdi burada aslında hesaplamanız gereken iki nokta olabiliyor: Bir, sürücünün yolcuya gideceği, **Pick-up** dediğimiz, yani onu alacağı yer; burada bir ETA süresi var. Daha sonrasında aldıktan sonra da ulaşacağı Destination noktası var. Aslında bu ikisi de benzer gibi görünen ama gerektiğinde bazen farklı dinamikleri olabilen problemler diyebilirim.

Buraya dediğim gibi farklı farklı makaleler de koydum. Mesela **DoorDash**, yine dünyaca ünlü yemek firması. Burada da mesela DoorDash'in kendi sayfasında çok farklı böyle blogları var. Sırf bu ETA Prediction'a odaklanarak farklı farklı çalışmalar yapmışlar, "Nasıl bunu daha accurate (doğru) tahmin edebiliriz?" diye.

Buradan aslında en önemli problemlerden birine geliyoruz. Şimdi, buraya gene döneyim... (Çizim yapıyor). Hepiniz duymuşsunuzdur aslında **Normal Dağılım**'ı (Gaussian Distribution). Şöyle düşünün; simetrik, gayet güzel dağılmış bir veri. Tam şu ortası **Mean** (ortalama) oluyor. Şurası işte bizim medyan oluyor. Şuradan birlik standart sapması vesaire... Onunla ilgili bir şey yapmayacağım ama biraz ETA Prediction'ın doğasını anlatmak için bundan bahsediyorum.

Şimdi böyle bir Target'ı (hedefi) tahmin etmek gerçekten çok tatlı oluyor. Çünkü veriniz çok güzel dağılmış, çoğu zaten hesaplamanız gerekenler burada, mean'e yakın. Şurada ufak outlier'lar (aykırı değerler) var. Tabii ki de buraya gittikçe zorlaşan tahminleriniz var ama büyük oranda verinizin büyük bir kısmı mean'e yakın olduğu için gene tahmin etmesi çok daha kolay.

Ama ETA Prediction'da, mesela DoorDash'in şu **"Long Tail"** (Uzun Kuyruk) olayını hatırlıyorum. Bu makalede başarılarını artırmak için bazı yöntemler üzerine konuşmuşlar. Burada "Long Tail Event" demişler. Şimdi Long Tail dediğimiz ne oluyor? Az önceki görselde tam Mean ile Medyan üst üste geliyordu. Ama burada farkındaysanız Mean biraz daha bu tarafa doğru kaymış, yani şurada "Long" bir, gerçekten kuyruk gibi bir **Long Tail** oluşuyor. Bunun sebebi de aslında bu tarz zaman tahminlerinde illaki sıfırdan başlıyor veri; buraya doğru yani böyle uzun mesafeli bir şekilde outlier'lar olmuş olabilir ya da bir şekilde uzak mesafeye gidenler... Farkındaysanız veri genelde burada birikiyor ama bir şekilde uzayıp gidiyor. Ve şey de diyemiyorsunuz: "Ben şuraya kadarını tahmin edeyim, burayı tahmin etmeyeyim" deme şansınız yok.

Bunlar üzerine zaten konuşacağız. Bazen modelinizde şu kısmı (yoğun kısmı) tahmin etmek kadar, buradaki olan sayının (uç değerlerin) tahmini de çok önemli. Zaten DoorDash'in birçok yazısında da "Burayı (uç kısmı) nasıl daha iyi tahmin edebilirim?" sorusuna odaklanıyor. "Ben burayı bir şekilde ediyorum tahmin, çünkü verinin çoğu toplanmış burada, yeterince verim var ama bu kuyruk kısmını nasıl daha iyi tahmin ederim?" buna odaklanıyor.

Burada farklı farklı yaklaşımlar olabiliyor. Mesela DoorDash bu blog yazısını tamamen bu Long Tail Event'in nasıl daha iyi tahmin edileceği üzerine yazmış. Kendilerine histogram çizdirmişler, bir yerden sonra uzamaya başlamış burası. Burada yanlış veri de olabilir ama yanlış veri olmadığına karar vermişler.
Daha sonrası ise farklı farklı metrikler; **MSE (Mean Squared Error)** mesela kullanmışlar. Regresyon dersinden hatırlarsınız, aslında neden farkın karesini (square) aldığımızda şu oluyor: Gerçek değerimiz eksi tahminimiz, bunun karesini alırsak bu **Mean Squared Error** (MSE) oluyor. Burada karesini almamızın sebebi aslında ortalamadan daha uzak olan kısımlardaki hatayı biraz daha büyütmek, cezalandırmak.

Burada "Absolute Error (MAE) mu kullanmalıyım, Squared Error (MSE) mu kullanmalıyım?" dediğinizde bu tamamen sizin **Business** (iş) kararınıza bağlı. ETA'da mesela böyle bir problem çözeceksiniz ve o an şunu düşünüyorsanız; "Burada bunun doğrusu yanlışı yok ama hani bir şekilde bunu yönetmem lazım." Mesela bir case çözüyorsunuz, bir şirketten problem aldınız; yüksek hataların daha çok cezalandırılmasını istiyorsanız **Squared Error** kullanmak daha mantıklı. Çünkü burada siz karesini aldığınızda o hatayı daha çok büyütmüş oluyorsunuz.

Aslında en önemli dediğim gibi ETA'nın challenge'larından biri bu ileri uzayan kuyruğu olduğunu söyleyebilirim. Burada yeterince iyi feature'larınız varsa bazen bu kuyruğu iyi tahmin edebilirsiniz. Elinizde iyi feature'lar olmayabilir, bu arada. Öyle bir durumda maalesef burayı iyi tahmin ederken, buradaki değerleri de (kuyruğu) buraya (ortalamaya) yakın tahmin ediyorsunuz, çok iyi tahmin edemiyor. O yüzden de ben bu tarz böyle özellikle bir tarafa kaymış olanlarda sadece genel "Error"a bakmıyorum. Burada bir sürü **Segmentation** yapıyorum. Sonucunda da mesela gruplayarak hatalara bakabiliyorum. Yani mesela atıyorum 3'e bölüyorum diyelim burayı: Total bir Mean Squared Error baktım; sonra bu parçalar için de ayrı ayrı bakıyorum. "Ya ben buraları (kısa mesafeleri) iyi tahmin etmişim ama burayı (uzun mesafeleri) çok kötü tahmin etmişim" de olabilirim. Bu da hani Business'a vesaire anlatırken veya kendiniz modelinizin hangi kısımlarda iyi sonuç verdiğini anlamanız için güzel bir yöntem oluyor.

Zorluklar üzerine aslında biraz değindik ama böyle çok kısa bir üstünden geçmem gerekirse: Aslında bahsettiğim gibi mesela trafik olayı... Trafiği tamamen görmekte zorlandığımız bir şey. Tabii ki de geçmiş verilerimiz bunları bize gösterebiliyor. Zaten sizin elinizde atıyorum İstanbul'da akşam trafiğinin yoğun olduğu verisi varsa model bir şekilde bunu **learn** edebiliyor (öğrenebiliyor). Ama tabii ekstrem bir durum oldu; mesela bugün maç var, o yüzden çok trafik var. Hani böyle bir durumu

bazen modele koymak çok zor olabiliyor. Yani burada belki o outlier'ları elemek mantıklı olabilir ya da belki gerçekten "maç günleri" diye ayrı bir **feature** bile yaratmak mantıklı olabilir.
Hava koşulları olabiliyor; yağmur yağdığında mesela İstanbul'da çok fazla trafik oluyor, sis olduğunda veya kar yağdığında çok fazla trafik olabiliyor gibi durumlar. Saatlik hafta sonu yoğunluğu oluyor.
Mesela böyle bir proje yaparken sizin böyle ters feature'lar türetmeniz, alabiliyorsanız hava durumunu belki tahminlerini almanız gibi feature'lar modelinizi iyileştirebilir. Tabii ki hiçbir zaman garantisi yok.
Rota belirsizliği dediğimiz kısım da şey oluyor; yani bazen saatlere göre rotalar değişebiliyor. Öyle bir durumda tabii siz çok doğru tahmin edemeyebiliyorsunuz. Bir rotaya göre, mesela Uber'de gösterdiğim gibi bir rotaya göre tahmin yapıyorsunuz, bu rota üzerinde trafik var, o bölgede başka bir rotadan gidiyor, süre uzuyor gibi.
İşte bahsettiğim gibi büyük ölçekler ve uzun mesafeler de işi bazen zorlaştırabilir.

Bugün nasıl projeye bakacağız derseniz; bir **New York City Taxi** verisinde yolculukların süresini hesaplamaya çalışacağız birlikte.
Hızlıca bakalım, elimizde nasıl bir veri var? Train ve Test olarak var ama ben sadece Train verisini kullanacağım size göstermek için.
Burada işte ne zaman alınmış (pickup datetime), ne zaman bu trip bitiyor (dropoff datetime), kaç tane **Passenger** (yolcu) olduğu, **Longitude/Latitude** (enlem/boylam) verileri... Hem aldığı hem bırakacağı lokasyonlar. Trip duration zaten bizim hesaplamamız gereken. Baktım zaten çok az **feature** var. İlk böyle çok bu alanı bilmediğimizde "Hani bu feature'lar ne kadar üretebilirim?" gibi insanın aklına gelebiliyor ama en basitinden böyle bir durum varsa işte zaman verilerini burada türetmek mantıklı.

Benim ilk yaptığım en basit hareketlerden biri hızlıca bir bölge **Describe** edip nasıl bir dağılım var ona bakmak. Şimdi burada benim ilk gözüme çarpan mesela şu sayı. (Max değere bakıyor). Yani bu kaç dakika oluyor? Yani bu 979 saat gibi bir şey... Yani böyle bir şey mesela mümkün değil, burada büyük bir sıkıntı var. Yani bunu şu şekilde açıklayabilirim: Belki taksicilerle falan konuşursunuz, duyarsınız; böyle mesela uygulamaya girerler ve uygulamada trip'in bittiği ile ilgili eğer bir şeye tıklamazlarsa bazen o trip devam eder. Size verilecek bazı mesela **Case**'lerde size bilerek kirli data da verebilirler. Yani sizin mesela bu tarz durumları görüp "Yani bunu, bu outlier durumunu benim çıkarmam lazım" demeniz gerekiyor.
Mesela tam tersine de bakalım; mesela 1 saniye sürmüş bir trip var. 1 saniye de çok gerçekçi bir veri değil. Büyük ihtimal burada ya bir **Fraud** durumu oldu, bazen işte kişi belki kendine puan vermek için çok kısa bir trip açtı kapadı gibi bir durum olmuş olabilir. İşin doğasını bilemiyoruz ama ben böyle biraz daha case şeklinde anlatmaya çalışıyorum size.
Ya ben işte Şirket 1, Uber, bu tarz genelde çalışıyorum, böyle bir haberim var. Ya bu veriye baktığımda illaki outlier'lar var. Bu trip gerçekçi değildir, bunu temizlemem gerekiyor.

Sonra enlem boylamda da böyle biraz yüksek değerler görüyorum ama bunlar şey de olabilirler; gerçekten o bölgeye ait olabilir. Sonra mesela onları kontrol ettim, gerçekten de New York'un biraz uzak köşeleri. Passenger count'larda (yolcu sayısı) bir baktım, öyle çok yüksek mesela 9 gibi bir şey var; aynı şeyi de bilmiyorum. Onu da daha sonrasında incelemek için kendime notumu çıkarttım.
Daha sonrasında böyle bir hızlıca bir histogramını da çıkarayım dedim. Sayılardan evet biraz az çok dağılımı tahmin ettim; dehşet bir **Long Tail** var gibi görünüyor.
Sadece 2 tane değer vermiş zaten 1 ve 2 değeri Passenger Count'ta... Bu enlem boylamları da aslında dediğim gibi

outlier'lardan dolayı mesela histogram çok düzgün çıkmamış. Trip duration da mesela burada çok korkunç görünüyor. Trip duration ile ilgili detaylı bir analiz yapmam lazım, gerekiyorsa buradan işte bazı outlier'ları elemem lazım.

Ben bu arada elerken hani şu an böyle çok basit bir yöntem anlatacağım. Biraz da böyle case'lerde bu basit yaklaşımlardan bence çekinmeyin. Bir şekilde bunu kendiniz, dediğim gibi açıklayabildiğiniz sürece aslında bu tarz dokunuşlar hoşlarına da gidiyor.
Yüzde birlik noktasında outlier'lar varken mesela yüzde beşlik bir kısımda 180 saniye... 180 saniye de 3 dakika ediyor. Mesela atıyorum 180 saniye benim için makul. Yani en az, şöyle bir hesap işi yapıyorum; trip'in gerçekten, yani bir taksi sürüşünün gerçekten mantıklı olması için en azından bir 3 dakika sürmüş olması gerekiyor gibi kendimce bir sınır belirledim ve bu da verimin yaklaşık yüzde beşini atıyorum ön taraftan.
Arka taraftan (üst sınırdan) elemek biraz daha zor oldu. Çünkü burada dediğim gibi net bir şekilde keşke biri bana "İşte ya 3 saatin üstü şöyledir" dese... New York'ta yaşamadığım için burada tabii çok net bir şey söyleyemem, çok kolay olmuyor ama en azından böyle bir şekilde bakıp bir **insight** çıkartmaya çalışıyorum.
Önce böyle bir detaylı, işte binde bire kadar bir persentil'leri (yüzdelikleri) çıkarttım. Şurada bir **Jump** (sıçrama) fark ettim. Bu jump benim ilgimi çektiği için zaten orada biraz daha böyle parçalayarak ilerlemeye karar verdim.
Burada mesela en son 8 binlerde kesmeye karar verdim. Ki zaten 8 bine de baktığınızda mesela ne saate denk geliyor...

Daha sonrasında dediğim gibi bir Passenger Count'u incelemek istedim. Burada da Passenger Count bazında böyle bir şey... Şurada 0 gibi bir değer var ki hani çok makul değil zaten ve çok az gelmiş, burada büyük ihtimal bir sıkıntı var. Daha sonrasında yukarıda hatırlarsanız zaten demiştim 6'dan sonra böyle çok görünmeyen bir veri seti vardı. Oradaki gerçekten şurada 3, 1, 1 gibi sayılar görüyoruz. Burada birkaç farklı şey yapılabilir. Yani bir tanesi bunların doğru olmadığını bilirsiniz, ama buraya farklı bir şekilde, buradaki sayıları mesela işte **Unknown** diye bir şekilde, işte bilinmeyen bir şey diye etiketleyebilirsiniz. Ama ben burada çok az veri olduğu için, yani aklımda totalde 65 tane veri olduğu için hiç bunları verimi kirletmesini istemiyorum. Passenger Count'un önemli olabileceğini düşünerek bunları verimden çıkarıyorum.
İşte finalde yaptığım şey aslında ilk belirlediğim üst sınır, ondan sonrası en alttan bir 3 dakikanın altını eliyorum, yukarıdan da işte yaklaşık bir eleme yaptım.
Sonra da trip duration'a baktığımda aslında ilk haline göre şu anda çok daha belirgin bir **Distribution** (dağılım) görüyorum. Tabii burada hala fark ettiğiniz üzere bir şey var ama bunlar olabilecek sayılar. Yani mesela 3000 saniye dediğimiz değere baktığımızda, yani 1 saat bile değil aslında. Yani bu çok olabilecek bir trafik sürüşü ama veri az. Yani çoğu insan büyük ihtimal daha kısa sürüşler, daha kısa mesafelere gitmeyi tercih ettiği için burada tabii veri birikmesi var. Ama burası da benim için önemli; ben gidip de bu 3000'i işte 1000 diye tahmin edersem aslında burada büyük bir hata yapmış olurum.

(Feature Engineering Kısmı)
O zaman devam ediyorum. Şimdi burada dediğim gibi işte haftanın günü, saati...
Daha sonrasında şimdi bizim elimizde dediğim gibi lokasyon verisi var ve bu lokasyon verisini kullanmak pek de kolay değil. Burada bazen şöyle bir yöntem kullanıyorlar, ondan bir bahsetmek istedim: **PCA (Principal Component Analysis)** duymuşsunuzdur. Daha çok **Dimensionality Reduction** (Boyut İndirgeme) için kullanılan bir yöntem ama burada biraz veriyi daha somutlaştırmak için kullanılıyor. Bu yüzde yüz işe yarar diye bir şey yok hani. Bu yöntemde hani şöyle diyeyim; ben de biraz araştırırken gördüm, deneyenler işe yaradığını ve modellerde iyi sonuç aldıklarını söylüyorlar. O yüzden de hani bu tarz farklı yöntemler deneyip kendi datanızda modelinizde iyi çalışıyor mu diye kontrol edebilirsiniz. Bu her zaman işe yaramıyor;

bazen mesela bir noktada bu şekilde PCA'ya çevirmek işe yarayabilir ama başka bir noktada yaramayabilir. Ama ben de böyle bir yaklaşım gördüğüm için mesela merak edip denedim. Bence böyle şeylerde özellikle araştırıp denemeniz önemli.
Onun dışında böyle çok güzel yakın bir değere getiriyor.

Burada mesela **Distance** hesabı... Dünya düz değil, yuvarlak. Öyle olduğu için de latitude'lar arasında direkt olarak sayıları çıkarıp işte hesap yaparsanız olmaz. Şöyle bir hesaba giderseniz, şuradan anlatayım: İşte şöyle mesela iki noktamız var. Bu noktayla bu nokta arasında dünyanın ovali biraz esnetiyor. **Haversine Distance** dediğimiz, sektörde de aslında kullanılan, yani **Kuş Uçuşu** mesafeyi daha doğru ölçmek için kullanılan bir yöntem bu. Ve bunun formülünü ezberlemenize kesinlikle gerek yok, zaten herhangi bir yere sorarsanız bunu size veriyor.
Bu da aslında şunu gösteriyor: Şimdi mesela şöyle diyeyim; Beşiktaş'tan Kartal'a gitmek ve Kartal'dan Beşiktaş'a gitmek diyelim. Aslında ikisinin yönü farklı. Aslında bu değer de onun yönünü veriyor, yani ne tarafa doğru gidiş var şu an. Bu **Bearing** (Yön açısı) değeri de mesela bu tarz böyle ETA tahminlerinde kullanılıyor. Çünkü sizin ne tarafa doğru gidiyor olduğunuz da önemli. Bu Bearing değeri de size onu söylüyor, tarafa doğru gittiğini söylüyor.
Daha sonrasında da ben PCA'den, **Manhattan Distance** dediğimiz aslında bizim klasik hesaplama, şuradaki 1. değerle bunun 1.'sini, 2. ile 2.'sini çıkarıp işte mutlak değerini alıp hesaplama kısmı, en basitinden mesafe hesabı aslında.

Daha sonrasında nasıl bir hesap yaptım? Bir de işte genelde işte son 60 dakika içinde, yani son 1 saat içinde totalde ne kadar talep gelmiş, ona bir hesapladım. Burada ekstra olarak bunun sebebi de şu: O bölgedeki yoğunluğu anlarsam ETA tahmininde bana yardımcı olabilir diye düşündüm.
O bahsettiğim mesai saati durumunda mesela işte sabah saatlerinde insanlar daha çok seyahat etmeye ihtiyacı duyuyorlar ve bu aynı zamanda neyi gösteriyor? Daha çok trafikte oluyor. Tabii daha çok insan yola çıktığı için aslında sizin orada işte son 60 dakikada, son yarım saatte ne kadar talebinizin olduğu bir şekilde trafiği de size belki gösterebilir.
Uber gibi çok büyük firmalarda bu dalgalanmalar çok daha net görülebiliyor. Böyle bir durumda da model için önemli bir veri olabilir.

Şimdi **H3**. H3 nedir? H3 bu arada ben de daha öncesinde aslında çok bildiğim bir kavram değildi. Şöyle bir yapı... Bahsettiğim dünyanın yuvarlak oluşundan dolayı bölgeleri ayırırken kare kare ayırdığımızda karelerde kırılmalar oluyor. O yüzden de böyle yaklaşık bir altıgen (**Hexagon**) şeklinde aslında ayırıyor. Ve bu H3 (Haş-Üç) dediğimiz kütüphaneyi **Uber** yapmış. Bu kütüphane sektörde çoğu firma tarafından kullanılıyor. Bunların hepsini dereceleri var; işte 9, 10, 11 böyle gidiyor. Bu sayı büyüdükçe, yani mesela 11 olduğunda artık çok daha küçük bir bölge, 9 daha büyük vesaire gibi, böyle böyle büyüyor.
Bunu neden anlatıyorum derseniz; bazen bazı feature'ları böyle **H3** seviyesinde çıkarabiliyoruz.
Kullanımı da bayağı basit yani çok büyük bir şey yapmama gerek yok. Ben sadece ne yaptım? H3'ü bir import ettim şurada yukarıda. Eğer yoksa önce pip install edebilirsiniz.
Burada ben 7'yi kullandım (Resolution olarak). Neden derseniz şu; aslında ben biraz deneme yanılma yaptım, öyle söyleyeyim. Şimdi benim amacım şuydu: H3 seviyesinde feature'lar çıkartmak istiyordum. Yani atıyorum ben işte Beşiktaş bölgesindeki H3'ler için geçmiş verileri hesaplamak istiyorum. Mesela işte geçmişte buradan çıkan insanlar, atıyorum işte geçmişte Beşiktaş'tan Kartal'a giden insanlar aynı saatte yaklaşık 30 dakikada gidiyorlardı... Mesela böyle bir veri bulmak istiyorum, bunu modelime eklemek istiyorum.
Şimdi bunu ne kadar yüksek yaparsam (resolution'ı artırırsam) bölgeyi o kadar küçültmüş oluyorum. Şimdi bölgeyi küçültmenin dezavantajı orada bir **Sparsity** (seyreklik) yaratıyorum. Yani şöyle; atıyorum şu bölgeyi aldım. Şimdi ben ne kadar

küçültürsem veri o kadar azalıyor. Yani bu bölgeden belki sadece 3 tane taksi çağırdı. Ama bu bölgeden totalde 1000 tane çağrıldıysa o verdiğiniz feature çok daha anlamlı, çok daha güvenilir oluyor.
Ama bunu çok büyütürseniz, o zaman da çok geniş bir bölge aldığınız için feature çok anlamlı olmuyor. Şimdi ben bütün Beşiktaş bölgesini tek bir **Hexagon**'a alırsam; Beşiktaş'ın atıyorum Akaretler belki daha yoğun bir bölge, o an çok daha çıkması zor trafik var; Beşiktaş'ın başka bir uç bir bölgesi aslında çok da trafik olmayan bir bölge. Bunların hepsini birlikte almış olacağım. O yüzden böyle durumlarda dengeli davranmak çok önemli. Tabii bunlar da biraz deneme yanılma oluyor açıkçası.
Ben de burada özellikle şeye dikkat ettim, yani öyle bir böleyim ki hani sayısına baktım, yaklaşık benim işte 700-273 tane olmuş. Çok aşırı da ufak ufak bölmeyeyim ama çok geniş de olmasın. Böyle birkaç denemede 7 değerinin iyi olabileceğine karar verdim.
Bu H3 de böyle aslında güzel, sektörde kullanılan farklı bir kütüphane. Burada **Kaggle**'da da proje var. Bu projeyi genelde ben baktığımda burada H3 yerine hep şeyi kullanmışlar, **K-Means** kullanmışlar. K-Means kullanmak biraz sakıncalı olabilir. Çünkü çok boşluklu yerler de var ve o boşluk yerleri bir anda böyle büyük bir şekilde mesela şuraya gruplarsa çok anlamlı bir şey olmaz. Yani oradaki bölge kısmı benim için çok daha önemli. O yüzden ben böyle H3 gibi kütüphanelerin K-Means'ten daha iyi çalışabileceğini düşünüyorum. Ben o yüzden H3 kullanmayı tercih ettim.

Tamam feature'ları yarattım, artık verimde benim kolonum var. Burada sonra ne yaptım? İşte aynı şeyi, işte son 60 dakikada o H3'lerde ne kadar yoğunluk olduğuna baktım. Yani aslında artık bütün New York'un içine bakmıştım ama şimdi bölge bölge baktım. Yani işte Şişli'deki ayrı baktım, Beşiktaş'taki ayrı baktım. Aslında bu sayede o bölge üzerindeki yoğunluğu da anlayabildim.
Burada daha detaylı ne yapabilirsiniz? Mesela işte ben sadece **Pickup** (alış) H3 çıkarttım. Siz burada işte **Drop-off** (bırakış) H3 de çıkarabilirsiniz. Sonra o 2 nokta arasındaki, Beşiktaş ile Şişli arasındaki daha önce yapılmış triplerin sürelerini buradan hesaplayabilirsiniz mesela. Gene gruplayarak hani böyle farklı farklı feature'lar çıkartmanızı sağlıyor H3.
Aynı zamanda burada onu da bir göstereyim isterseniz; çelik kodları da insanlar paylaşıyorlar, bunları da inceleyebilirsiniz. Burada da güzel böyle farklı feature önerileri de görebilirsiniz.

Ben bir **Model Base** oluşturdum. Yani bu model base benim için feature'ların yaratılmadan önceki hali. Bir de normal model kolonları, yani benim ekstra yarattığım feature'lar...
Modelleme aşamasında...
Şimdi o az önce bahsettiğim noktaya geleceğim: **Metrikler**i hesaplarken.
Burada benim bir kuyruğum (Long Tail) var. O yüzden de bütün verinin hani, evet tabii ki işin sonunda optimize ederken hani genel **Mean Squared Error** ve **Absolute Error**'ı optimize edeceğim ama ben parça parça da görmek istiyorum. Yani ben bu bölgede (kısa mesafe) ne kadar başarılı olmuşum? Şu (uzun mesafe) noktada ne kadar başarılı olmuşum? Bunları da görmek istiyorum. O yüzden ben kendimce bir tane fonksiyon yarattım işte "Evaluation Calculation" diye.
Kendime **Low**, **Middle**, **High** threshold (eşik değeri) belirledim. Onlara da biraz veriye bakarak geldim. İşte 10 dakikanın altındakileri seçeceğim, 10 dakika ile 40 dakika arası, ve üstü. Şurada kuyruğu da görmüş olacağım. Burada tek yaptığım şey işte Low, Middle ve High olarak bölüp her parça için **Mean Squared Error** hesaplamak. Dediğim gibi hani bu aslında benim direkt modeldeki optimizasyonum değil; benim modelimin ne kadar iyi olduğunu, bu bölgede ne kadar başarılı olduğunu

görmemi sağlıyor. Özellikle böyle hani eğer iş mülakatlarına giriyorsanız, farklı metrikler çıkararak sunumlarınıza eklemeniz de bence gayet güzel olur.

Daha sonrasında bir **Cross Validation** yapacağım. Ben Cross Validation ama **Time Series Split** yaptım.
Burada aslında gene ben ne yaptım? 4'e bölmeyi tercih ettim kendim. Daha sonrasında ben burada dedim ki, ben biraz **Mean Squared Error** şey yapmak istedim, optimize etmek istedim. Bunun sebebi de özellikle yüksek hataları daha iyi görmek istiyorum, yüksek hataları daha çok cezalandırmak istiyorum ki modelim onları da daha iyi tahmin edebilsin. Ama tabii arada mutlaka **MAE** (Mean Absolute Error)'ye de bakıyorum. O da çünkü bazen de yukarıya çok odaklanırsam bu sefer de böyle şey oluyor; ortadaki asıl yani tahmin etmem gereken kısmı da bazen kaçırabilirim. Yani kuyruğu iyi tahmin ediyorum, bu sefer o daha yığılmış tarafı tahmin etmekte zorlanıyorum. O yüzden her zaman burada dengeli davranmaya çalışın.

Burada **Default** parametreler belirledim, ilk bir modelimin performansını görmek için.
Önce **Base** kolonlar ile tahmin aldım, bir de normal model kolonları yani benim ekstra yarattığım feature'lar ile.
Sonuçlarımı bir inceledim. Ama gerçekten hani hem aramasıyla hem de sonuçlar iyi görünüyor; yani gerçekten benim yeni feature'larım **Improvement** (iyileştirme) sağlamış.
Kütüphanesi ile bunların detaylarını inceleyebilirsiniz. Hani belki burada ekstra 10 feature ekledim, bunları arasan belki sadece üçü bu iyi etkiyi yarattı. Ki büyük ihtimal genelde öyle oluyor, 10'u birden iyi çalışmıyor.
Split'lerin ayrı ayrı performanslarına bakmanız bu tarz durumları görmenizi sağlıyor.
En son dediğim gibi şeye bakacağım şimdi; benim zaten biliyorsunuz trip duration'ımın dağılımı şu şekildeydi. Benim tahminlerim nasıl oldu? Yani çok mu böyle ortaya kaydılar yoksa gerçekten o dağılımı biraz yakalayabildi mi diye bir bakmak istedim. Baktığımda da aslında evet fena yakalamamış gibi duruyor. Yani tabii ki mesela burada 8 bine kadar uzamış, o kadar tahmin etmemiş ama zaten o bahsettiğimiz şuradaki iyice görünmeyen artık kısımda çok çok az veri olduğu için, ya benim modelim de zaten oraya aşırı **Overfit** (aşırı öğrenme) olsun istemiyorum.
Hani baktığımda en azından şu **Distribution**'ı görebilmesi ve hani şuradaki daha uç yüksek değerleri görebilmesi benim için önemli. En azından buradaki Long Tail'ı görebilecek feature'lar çıkartabilmek... Çünkü bazı durumlarda maalesef şöyle benzer bir şekilde kalıyor ve buradaki daha yüksek değerleri model çok görmeyebiliyor.
Ortalamada model başarılı olmuş diyebilirim.

Bir de şurayı göstermeyi unuttum, şu şekilde bir **Error Analysis** çıkarttım: **Low**, **Middle**, **High** segment olarak. Burada tabii beklediğimiz gibi Low'dan High'a giderken hatalarımız artıyor doğal olarak. Çünkü dediğim gibi yani zaten verinin çoğu burada (Low/Middle) olduğundan dolayı model burayı tahminlemeye daha çok odaklanıyor. Hocam tiyatro düşünelim; modeliniz buraya daha çok odaklandı, burayı iyi tahmin edebilir ama verinin çok büyük bir kısmını kaçırırsa sizin ortalamadaki hatanız çok çok daha yüksek olacak. O yüzden amacımız burayı iyi tahmin ederken, burayı da bir şekilde modelin görebilmesini sağlamak. Bunu sağladığınızda zaten hani ortalama başarılı bir model elde etmiş oluyorsunuz.

Aslında benim anlatım kısmım bu şekilde.
Varsa böyle sorularınızı alabilirim.

(Enes F. Manan):

Enes ne soracaksın sor hahaha. Sorularınızı bekliyoruz. Ben bir ara, tam önce işlemenin galiba outlier kısmını anlatıyordun, orada bir gittim, bir 5 dakika AFK kaldım orayı kaçırdı tamam mı? Bazı şeyleri kaçırmış olabilirim o yüzden sıkıntı yok. Çok hızlı geçtim, modelin başarısını gösterdin mi?

(Özge Usta):

Gösterdim evet.

(Enes F. Manan):

Nasıl bir başarısı var?

(Özge Usta):

Ya bu arada ben tabii **Tune** (ince ayar) etmedim modeli çok fazla, hani o kısma o kadar odaklanmadım. Sadece şu an şeyi gösterdim; Base kolonlar ile olan sonuçlara baktım. Bir de normal hani bizim ekstra eklediğimiz işte Feature'lar ile. O da baktığınızda aslında hani kaç feature fazla... Ekstra eklediğimiz işte şu daha lokasyon verileri bar ekstradan.

(Enes F. Manan):

İlk mesela aklıma gelen şey burada, daha önce ben de şeye katılmıştım, yapıya benzer bir şey diyebilirim. Hani **GDZ Elektrik**'in (Gediz) yarışması vardı 2024'te yaptığı **Kaggle** yarışması. Daha önceden kesinti olacağının bilgisi olmadan elektrik kesintisinin olma ihtimalini tahmin etmeye çalışıyordu. Orada skoru artıran şeyler; birincisi, en birincisi diyelim, dışarıdan veri eklerken bu **Weather API**'dan alınan mesela hava durumu verileriydi. Burada böyle bir şey ben görmedim, hava durumu verilerini eklemeyi düşünsek burada?

(Özge Usta):

Tabii ben onu başta söyledim bu arada ekleyebiliriz diye. Ben sadece dahil etmedim ama tabii ki de dahil edilebilir yani. Biraz mesela sektörde çalışırken biraz bunun testleri yapılıyor. Önce çünkü hava verisinin sürekli tahminlerini almak... Yani şöyle; iyi çalıştı, bunu canlıya alacaksın. Canlıya aldığında sürekli bu tahminleri çekmen gerekiyor ya. O yüzden burada önce bir "hava koşulları iyi çalışıyor mu gerçekten?" diye test ediliyor. Önce bir çekiyorlar, belki işte bir aylık iki aylık verinin iyi çalıştığını görürlerse genelde bu API'ları dahil ediyorlar.
Ama evet yani weather verisi bence bazen iyi çalışmayabiliyor ama ben bu tarz, yani sonuçta trafiği çok etkileyen bir şey. En azından bir yağmur yağdı, yağmadı; böyle basit bir veri bile bence bayağı işe yarayabilir diye düşünüyorum.

(Enes F. Manan):

Ya orada şeyi çok doğru söyledin; yani bu model son kullanıcıya açılan bir model aslında. Baktığın zaman dakikada binlerce prediction, belki yüz binlerce prediction alınması gereken, canlı çalışması gereken bir şey. İçeride bilgi gelecek, onu tahmin ettikten sonra işte bacaklarda burada kesinti olacak, ekipler gidecek falan... Yani direkt kaç tane şey var. Senin de düşünmen lazım; şimdi bir gerçekleşen verisi var, bir de tahmin verisi var ya. Hava durumunda gerçekleşen veriye göre eğitimi çalıştırdın, sonra tahmin verisini kullanıyorsun. Yarın yağmur yağacak mı? Ben bugün şey yaşadım mesela; sabah 8 miydi evden çıkıyordum ve hava durumuna baktığımda o an yağmur yağmıyor, çıktığımda yağıyordu. Yani o anı bile tahmin edemiyorlar. Mesela o da, yani baktığında işin sonunda sen gerçek veriyle eğitiyorsun ama senin production'da, belki yarın işte kesinti olacak mı olmayacak mı atıyorum, yarının verisi bir tahmin verisi olacağı için bunları da tabii göz önünde

bulundurmak belki gerekecek. Senin aldığın hava durumu mesela **Provider** (sağlayıcı) ne kadar çalışıyor? Belki de bunu da analiz etmen gerekecek ayrıyeten yani. Orada farklı farklı şeyler var. O yüzden evet **Kaggle**'da böyle iyi çalışıyor ama böyle iş hayatına girince biraz sıkıntılı olabiliyor onlar.

(Özge Usta):

Bir yorum daha gelmiş, hava durumunun bir şeyleri daha **interpret** edeceği hakkında... Evet bence zaten yapar yani yapmaması için bir sebep yok dediğim gibi. Orada böyle girdiğinizde yağış durumu da oluyor, nemlilik oluyor, işte ne bileyim sıcaklık oluyor, böyle farklı farklı rüzgar açısı falan gibi şeyler falan da oluyor. Dediğim gibi orada sonuç **intuitive** düşündüğünüzde yani yağmur veya güneşli hava durumları kesinlikle çok etkiliyor.

Bahsettiği mesela bu **Enerjisa**'nın da vardı galiba böyle yarışması elektrik üretimi, mesela orada da rüzgar, güneş, işte sıcaklık, o saatin verileri falan da modelde gayet iyi çalışan feature'lar.

Uğraşıp ekstra ekleyebilirsiniz. Yani ben biraz daha burada H3 vesaire girdiğim için çok hava durumu kısmına odaklanmadım açıkçası. Ama tabii ki de hani bence deneyip eklediğinizde mesela **Portfolio**'nuza (portföyünüze) böyle proje koyduğunuzda gayet tatlı olur. Yani çünkü bu sizin biraz da yaratıcılığınızı; "Ben uğraştım, bu veriyi aldım kullandım"ı göstermeniz gerçekten önemli diye ben düşünüyorum. Yani ben en azından bir işi alan biri olsam bunları gördüğümde hoşuma gider. Bu konuda çok daha deneme sebepleri...

(Enes F. Manan):

Ben teşekkür ediyorum, bu arada H3'ü ilk defa senden duydum zaten. Müthiş bir yaklaşım gerçekten.

(Özge Usta):

Evet ilginç ya, ben de işte Şirket 1'ya girdiğimde bu arada öğrendim. Ya direkt böyle bir modelde kullanılmıyordu aslında şu an. Ben hani kendinizi görünce hani mesela orada da çok mantıklı gelmedi bana **K-Means** sadece şey için, bir lokasyon datası için, oradan a3'e (H3'e) gidilmesi...

(Enes F. Manan):

Facebook'un kütüphanesi değil mi?

(Özge Usta):

Yok, **Uber**'in kütüphanesi. Yarışma 6 sene önce açılmış, geç mi? Ha, şeye baktım da 8 sene önce şey var, commit var. Bayağı eski bir repo ama çok bilinmiyor olabilir. Şimdi ben de mesela böyle bir yarışmaya girsem benim de aklıma K-Means gelirdi ama işte biraz daha orada araştırıp edip bir tık orada mesela H3 kullandığında karşıdaki zaten şaşırır mesela; "Böyle bunu biliyor, bunu almış kullanmış" gibi.

(Enes F. Manan):

Ben açıkçası bilmiyorum o dönemler belki açık kaynak değildir, belki bu kadar popüler değildir veya iyi çalışmıyordur. Yani bilmiyorum, şu an tabii bayağı etkiledi beni gösterdiklerin.

(Özge Usta):

Bence de güzel bir kütüphane ve çok hani kullanımı daha kolay. Yani en azından benim verimde çalıştı şu an yani. Böyle tabii saniyede kaç tane düşürmüş olabilir ama...

Mehmet mesajın için teşekkür ediyorum, umarım faydalı oluyordur bu oturumlarınızın hepsi.

(Enes F. Manan):

Ya bence de güzel oluyor açıkçası. Hani burada sadece böyle teknik şeyleri konuşmuyoruz, güzel **insight**'lar da oluyor tüm hocalarda. Püf noktaları vermeye çalışıyorlar. Çünkü gerçekten bence modeli daha iyi yapan her zaman böyle minik dokunuşlar oluyor diye düşünüyorum. Yani bir feature bir anda modeli böyle havalara uçuruyor.

(Özge Usta):

Ben, Enes sen de yaşamışsındır öyle şeyler, ben çok yaşadım. Bir yorum var; "Şehir merkezi gibi aslında değişmeyen kurumsal veri de kullanılabilir, belki de havalimanları, AVM yakınları gibi..."
Tabii yani kullanılabilir. Ya burada tabii biraz şu da önemli oluyor; mesela dediğim New York çok daha bildiğiniz bir bölgede bunu çalıştığınızda tabii oradaki farklar çok daha, orada çok daha fark yaratabiliyor. Sonuçta oraya özgü şeyler, işte o bahsettiğim bayram olayı bile; yani siz Türkiye'de bayramları bildiğiniz için veriye bunu ekliyorsunuz. Ya da o bölgenin yapısını bildiğiniz için... Yani mesela çok basit şey anlatayım, neredeyiz? İstanbul'dayız ya mesela. Aslında buranın çok farklı bir coğrafi özelliği var. İstanbul'da mesela atıyorum işte şu bölgeden şu bölge dediğinizde, yani şu bölgeyle şu bölgenin belki kuş uçuşu aynı mesafedir ama mesela boğaz etkisini bile koymanız gerekiyor. Veya işte farklı bölgeye gidiyorsunuz, körfez bölgesi... Yani bu bölgeyi mesela şu bölgeyle şu bölge birbirine çok yakındır ama aslında şöyle bir kısım (dolaşması gereken). Ya siz oranın bir şekilde coğrafyasını gerçekten işin içine katmanız gerekiyor ve bu da tabii çok kolay olmuyor aklınızda.

Mesela bunları bir yere, şöyle bir şey bu arada yaşadım: Biz bir işte iş yaparken yanlışlıkla ilk böyle gitmişiz, nerede İzmir... Şöyle bir... Bunu bu arada testlerimizde görüyoruz. Şurayla şurayı başlamışız yanlışlıkla, yani şu şekilde bir rota çizmişiz. Hemen bununla ilgili aksiyon aldık. Yani o yüzden de hani o bahsettiğiniz işte havaalanı bölgeleri olsun, işte o benim verimde de gösterdiğim New York'un mesela farklı uçları görünüyordu lokasyonlarda. Burada bu bir outlier değil, yanlışlıkla... Mesela ben İstanbul için model eğitirken araya bir İzmir karışmış da olabilir, yanlış veri gelmiş, kullanmış İstanbul diye. Bunları tabii ki elemem lazım ama bazen de gerçekten outlier olmayabilir. Yani ben verileri analiz ederken en çok aslında böyle dikkat ettiğim şeylerden biri; "Hani bir şey gerçekten outlier mı, elenmeli mi çıkarılmalı mı yoksa o aslında öğrenilmesi gereken bir nokta mı?" kısmını da bence analiz etmek gerekiyor diyeyim.

Bir yorum daha gelmiş; "Önce seni kötü hatta devlet hastanelerinden biraz yumuşadı, koşulların henüz olgunlaşmamış olması..." Yok bu arada farklı bir nokta olabilir bence. Ya sonuçta mesela 8 yıl önce bulundu dedi, bu proje 6 yıl önce... Ben bilmiyorum bütün **Kaggle**'da çalışan herkes o zaman mı yapmıştı? Onun tarihine çok dikkat etmedim açıkçası ama... Ama şu var; mesela diyelim ki siz bu projesine baktınız, işte buradaki notebook'ları inceliyorsunuz. Ben orada herkes **K-Means** yapmış diye siz de K-Means yapmak zorunda değilsiniz. Yani çünkü teknoloji gelişiyor, farklı şeyler, kütüphaneler de gelişiyor. Burada farklı yaklaşımları araştırmak ve onları denemek de önemli. O yüzden evet yani 6 yıl önce K-Means yapıldı diye K-Means daha iyi çalışmak zorunda değil. Ya burada dediğim gibi mesela ben de **PCA** kullanılıyor merak ettim bir denedim. İyi çalışmaya da bilir, çalışabilir. Belki o da eskide kalmış yöntem de olabilir. Ya sonuçta yeni bir alana girdiğimizde hepimiz, yani şu an ben de mesela hiç bilmediğim bir alanda model yapmak istesem ben de nasıl bakılıyor, nasıl bakış açısı var diye farklı şeyleri, bakış açılarını araştırıyorum. Bunları Enes de yapıyordur, diğer burada arkadaşlarımız da yapıyorlardır. Bazı yaklaşımlar eskide kalabiliyor ama bazıları gerçekten denediğinizde işe yarayabilir. Veri setinize göre de bu çok değişiyor tabii.

(Enes F. Manan):

Bence **Key** (anahtar) noktasında olan nokta bu diye ben düşünüyorum açıkçası; denemek. Bir de **sağduyu** yani iyi bir

sağduyunuz olması gerekiyor. Yani işte business'tan önce senin zaten bir işte orada Özge'nin gösterdiği gibi, veride verilen işte 1 saniyede veya ne bileyim orada gösterdiği 900 küsur saat miydi öyle bir şeydi, bu kadar olamayacağı belli zaten. Yani bu sadece bu konu için değil, yani bütün konular için sağduyu çok kritik. Bazı şeyleri senin kendine düşünüp birtakım aksiyonlar alabiliyor olman lazım.

Şimdi ben şey soracağım, merak ettiğim konu; bu sistemlerin altyapısı tam nasıl çalışıyor gerçekten hiçbir fikrim yok. Böyle bir firmaya mesela atıyorum bir taksi veya ne bileyim Şirket 1 ile bir şey seçtin sen. Senin ne kadar zamanda bunu söylerken Google'dan bir ETA yapıyor. Şimdi haritayı Google'dan alıyordu zaten, nereden alacak? Kullanılıyor mu bu?

(Özge Usta):
Bir yerde **Pipeline** da içeriye bir şey olarak, input olarak birisi giriyor...
Şunu net söyleyeyim; birçoğu tabii hala Google'ı kullanıyor. Google haritasını... Çünkü özellikle İstanbul gibi bir bölgede şeyi tahminlemek çok zor, doğru zamanı tahminlemek çok zor. Ama daha büyük firmaların bununla ilgili çok detaylı çalışmaları var dediğim gibi. Ve orada hani mesela dediğim gibi o Didi'nin makalesini okursanız; "Yani bizim bunu milisaniyeler içinde çıkartmamız lazım, çünkü bu insanların Google API'dan sürekli veri çekmesi aşırı maliyetli. O yüzden de bir şekilde kendi altyapılarını korumak zorundalar" diyorlar.
İşte oradan mesela ne yapıyorlar? Aklıma hatırladıklarım; yani büyük bir **Deep Learning** modelleri var, işte bir kısmını daha sabit bırakıyorlar mesela modelin geri modelleme tekniği var. İşte **Teacher-Student** dediğimiz, büyük asıl model onu işte daha ufak bir modele indiriyorlar gibi böyle farklı farklı hızlandırma metotları kullanmaya çalışıyorlar. Çünkü dediğim gibi yani bir eşleme yaptığınızda bunu milisaniyeler, bir saniye önce şey diyemiyorsunuz; "Ya sen bir 2-3 dakika bekle biz sana ayarlıyoruz" diyemiyorsunuz. Yani basıyorsunuz, birkaç saniyede taksinin ya da artık paylaşımcı aracınız geliyor. Öyle olduğu için de gerçekten burada hız çok önemli. Aslında Google API de o konuda çok aşırı hızlı değil.

(Enes F. Manan):
Çağrı bıraktığında burada yavaş dönüyorlar Enes.

(Özge Usta):
O yüzden orada çok daha hızlı bir sistem yapmak istediğinde biraz kendi yöntemlerine de dönmen gerekiyor.

(Enes F. Manan):
Şirket 1da ne kullanıyorsun söyleyebiliyor musun?

(Özge Usta):
Şirket 1da henüz... Ya orada daha çok Google kullanılıyor, henüz o tarafa çok girilmedi mesela. Ama bir ileride bir girilme hedefi var o tarafa. O yüzden dedim yani çok buralarda kullanılmıyor diye.

(Enes F. Manan):
Ya hazır şeyin var senin, hani API alabileceğin bir platform var Google gibi. Kaç tane ülkede, WhatsApp'ta milyonlarca anlık müşterisi olan bir şey. E tabii o onun elinde acayip bir kaynak var yani.

(Özge Usta):
Tabii zaten olay kaynak tamamen. Ya şu anlamda; şimdi bunu yapmak için de bir ekip lazım. Yani şu an hani ölçekler biraz daha farklı olduğu için aslında trafik odaklanıyor ama uzun vadede hani girilecek ve önemli hedeflerden biri. Mesela biz

Şirket 2'da da ETA'ya mesela API'a gidilmiyor, oradan orada biz tamamen geçmiş verileri kullanarak bir ETA hesaplaması yapıyorduk. Oranın avantajı ama şuydu; restoranların zaten yerleri sabittir. Öyle olduğu için de orada ortalamadan hesaplamak bizim işimizi görüyordu. Yani şey kadar, Uber gibi bir firma sana "Araç 5 dakikada gelecek" dediğinde hani en azından 6-7 dakikada gelmeli. 15 dakikada gelirse sıkıntı ama aslında Şirket 2 sana yaklaşık bir süre veriyor, onunla 5-10 dakika gecikiyor, sen yemeği bekle. Normal 5 dakika dediyse 5 dakikada gelmiyor yani.

(Enes F. Manan):
Çok kötü çalışıyor ya.

(Özge Usta):
Yani burada Şirket 2'un ETA diyoruz, ikisinde de ETA Prediction diyoruz ama asıl doğaları da çok farklı yani. Şirket 2'da Google'a gidilmeden orada bir, hatta şöyleydi ben 2020-22 o dönem sadece böyle geçmişteki ortalamadan hesaplıyor, sonra biz bunu modele çevirelim dedik. İşte hız önemliydi orada da yani. Enes siparişini verecek, anından onun ne kadar sürede hazırlandığını, ne kadar sürede ise kurye gelecek, kurye alıp Enes'e götürecek... Bunların hepsinin böyle milisaniye hesaplanması lazım. O yüzden de tabii altyapı deseniz, yok öyle bir şey, hani **Tree-based** bir şey çalışamıyor. Biz orada binlerce **Linear Regression** (Doğrusal Regresyon) eğittik ve katsayı verdik. Sonra hesaplamaları işte arkada yaptılar. Hatta ben hesaplamaları böyle kontrol ettim doğru mu yanlış mı, sonra verdiğimiz serinler, lineer regresyondaki katsayıları kullanarak anında bir sonuç dönüyordu mesela. Orada hız olsun diye böyle bir şey yapılmıştı tamamen. Ya benim herhalde kullandığım, yani 4 senede kullandığım tek Linear Regression modeli buydu. Genelde Tree-based kullanılır.
Bazen de mesela böyle hızlı olsun, hızlı sonuç dönsün ve kolay adapte edebilmek için bu basit çözümlere de gidilebiliyor. Çünkü **Linear Regression**'da x1 + x2 gibi böyle basit bir formülü olduğu için tabii implementasyonu da çok kolay oluyor, hesaplamaya vakit ayırıyorlar.
Uzun sürede bu kullanıldı, sonra biraz sanırım değiştirdiler ama o altyapıyı o kadar da geliştirmemiş, şirketin belli kaynağı bazı taraflarda biraz da öncelikli oluyor. O yüzden de böyle geri kalan kısımları bazen hani çalışan bir şey varsa "çok müthiş olmasa da okey" deyip geçiyorlar.
Ya tabii ki de Şirket 2 ETA'yı daha iyi yapamaz mı, tabii yapar. **Search** tarafında daha çok ağırlık veriyorlar, daha çok insanlar çalışıyor. Farklı bir taraftan daha çok çalışıyor. Öyle olunca da orada zaten çalışan, okey olan, can sıkmayan bir şey varsa dokunmuyorlar. O benim bahsettiğim modele de mesela yani bir buçuk iki sene hiç dokunmadım. Ders (model) çalışıyordu, ortalamada niye çalışıyordu? Bunlardan bu arada prediction avantajı; yani bu tarz böyle ETA gibi modellerin test etmesi çok kolay, **simülasyon** yapması çok kolay. Mesela **Recommendation** içinde böyle problem var. Sen Recommendation'da biraz Furkan da bahsetmişti sanırım Enes değil mi ondan; yani şöyle bir **Bias** (yanlılık) oluyor, siz hiç göstermediğiniz bir ürünü kullanıcı satın alır mı tahmin etmeye çalışıyorsunuz. Zaten bir süreyi tahmin ettiğiniz için sürene belli ne olacak belli. Sizin modellerde bence en güzel avantaj bu tarz simülasyonları çok rahat yapabiliyor olmanız.
Biz mesela teste çıkmadan önce simülasyon yaptık, baktık başarılı, zaten direkt çıktık. Sonra canlı beklediğimiz gibi yani bize **Leakage** (veri sızıntısı) yapmıyorsanız, modelinizi doğru bir şekilde eğittiyseniz çok büyük oranda model özel bir gün değilse, özel bir döneme denk gelmediyseniz çalışıyor. Öyle bir avantajı da var diyebilirim.

(Enes F. Manan):
Ben şeyi merak ediyorum; buradaki doğru **label** (etiket) ne madem? Yani bu sürenin doğru olduğunu nasıl belli ediyorsunuz? Zamanla ve Getir'de çalışan bir arkadaşım vardı onunla üzerine konuşmuştuk o dönem. Getir şöyle bir şey yapıyormuş:

Uygulama içerisinde direkt böyle teslim eder etmez kurye teslim etti mi işaretler anlık. Bonus puan veriyormuş buna. Alan kişi hızlıca teslim ettiğini söylerse ona da bonus puan verirse...

(Özge Usta):

Anladım anladım. Şimdi şöyle; hem Şirket 2'da hem de Şirket 1'da bununla ilgili bazı durumlar oluyor, onlardan bahsettim. Şirket 2'daki sıkıntı Enes'in bahsettiği şu; doğru anlamış mıyım ya da bir teyit edeyim: Kurye bazen geldim işaretliyor, gelmiyor. Yani gidip siparişi "Ben geldim, almaya geldim" diyor restorandan. Restoran bir bakıyor kurye yok. Kuryenin geldiği zamana bakıyoruz, gelmemiş. Ya da kurye siparişi teslim etmeden "Teslim ettim" diyor, zile basmıyor. Böyle zaman kaymaları oluyor.

Şirket 1 gibi uygulamalarda ne oluyor? İşte sürücü sürüşü bitiriyor ama bitirdim diye basmıyor. Yani o süre oluyor sana bir saatlik yolculuk, 1,5 saat, 2 saat; böyle şeyler evet olabiliyor. Şirket 2'da biz şöyle yapıyorduk: Mesela örnek veriyorum kurye saat işte 12.15'te ben geldim diye basmış. Ama restoranda 12.15'te bitirmiş ama işte 12.18'e kadar teslim edememiş. Şimdi oradaki zaman kaymaları olduğunda biz... Bizim tarafta orada şeyi anlamaya çalışıyorduk biraz kurye **Fraud** (sahtecilik) ediyor mu diye anlamaya çalışıyorduk. Çünkü kurye de böyle hemen olabilmek için daha gelmeden geldim diye işaretliyor ama aslında gelmemiş gibi durumlar oluyor. Mecburen bazen de göz yummak zorunda kalıyorduk. Ama bazı durumlarda bu dediğim gibi o zaman kaymasından anlayabiliyor; yani kurye "geldim" demiş, restoran "teslim ettim" demiş... Onları verimizden çıkarıyoruz açıkçası. Ama evet bu önemli problemdi. Yani bence güzel bir noktaya değindim; zaten label'ıng, neyin üzerine modeli koyacaksın? Başta datayı doğru toplaman lazım, sonra ben bunu düzeltemem.

(Enes F. Manan):

Uygulaması açık kalmış, büyük ihtimal kapamayı unutmuş.

(Özge Usta):

Öyle olunca yani, yoksa 900 saat mümkün değil yani.

(Enes F. Manan):

Orada ben şeyden bahsediyorum; sen Getir bunu nasıl **Handle** ediyormuş? Hala çalışıyor galiba proje. Bonus verme olayı okey. Hani kurye gidiyor, işte verince siparişi basıyor "Ben siparişi verdim" okey. Yani zamanında... Sonra adam veya kadın, siparişi alan kişi, erken dönemlerde hala Getir böyle mi çalışıyor emin değilim ama şimdi bunlar onaylandıktan sonra **Geofence** dediğimiz mevzu... Seni haritada bir dinliyor ve sana böyle bir daire çiziyor. Kurye ve "ben aldım" diyen kişi yani alıcı o dairenin içerisindeyse alışveriş doğru zamanda yapılmıştır diye seni **True** olarak bekliyor diyelim. Bak bu çok büyük bir şekilde yazmış yani sağlam bir proje. Getir'in projelerden biri, bunun development'ını da anlatmıştı bana, gerçekten çok enteresan. Yaklaşmaya saygı duydum, hesaplamalar falan filan böyle değişik yani şeylerle birlikte çalışmışlar aslında. **GIS** var biliyorsun, Coğrafi Information System (Geographic Information System), o getiren arkadaşımdı.

(Özge Usta):

Belki araştırmak isteyenler olursa diye **OSRM (Open Source Routing Machine)** diye bir kütüphane var. Bu açık kaynak bir kütüphane. Yani Google'ın böyle ucuz vermiş bedava versiyonu gibi düşünün. Tabii Google kadar iyi değil ama sayfaya girdiğinizde, yani siz iki lokasyonu verdiğinizde, işte lat-lon verdiğinizde oradaki rotayı veriyor. Burada **OpenStreetMap**'i yaptı değil mi? OpenStreetMap'i kullanıyor ve oradaki haritalar sürekli yenileniyor.

Hatta ben bu şeyde kullanmayı düşünüyordum ama tabii bu 8 yıl önceki bir veri olduğu için OSRM'den hani şu an 2025

olduğu için çok mantıklı olmayacak, sonradan vazgeçtim. Ya kurulum için biraz işte **Docker** ayağa kaldırıp çalıştırman falan gerekiyor bu arada.
Bir de şöyle, çok büyük bir taraf olursa; mesela ben New York'u denediğimde benim bilgisayarım kaldırmadı. Ama daha ufak böyle bölgelerde denemek isterseniz böyle farklı bir kütüphane de var. Bunu da mesela Getir'in ben bunu kullandığını duydum. Benim de bir arkadaşım, Mete, çalışmıştı; aynı kişi mi diye bir düşündüm arkadaşlar... Ya ben şimdi Getir'i övmüş gibi olmayayım. Kütüphanede hani hem rotayı veriyor hem böyle kavşak sayısını falan veriyor, böyle farklı farklı feature'lar veriyor. O açıdan hani güzel. Hani sadece onunla biraz işte canlı çalıştırmak için sanırım hani makine mi ayakta tutmak gerekiyordu, ama için bilmiyorum ama o tarz maliyetleri mutlaka vardır. Yani full full olarak bedava değildir ama Google'dan daha ucuz olduğunu düşünüyorum.

(Enes F. Manan):
Ben bir de mesela merak ettiğim şeylerden bir diğeri, diyelim senin kullandığın model neydi? Buradaki mi?

(Özge Usta):
Genel olarak gösterdiğim mimarlar hep **Deep Learning** kullanılmış.

(Enes F. Manan):
Mesela bunun sebebi ne olabilir?

(Özge Usta):
Deep Learning kullanıyorlar çünkü orada biraz daha bu bahsettiğim işte rota nasıl çiziyoruz? Rotanın parçaları var ya, mesela işte...
Hadi gelin Beşiktaş'la performans... Beşiktaş diyelim meydandan Kartal'a gidelim... Beyazıt kartalı... Kartal... Şimdi birini seçelim, şunları seçelim. Ne renktir böyle ya neyse ilginç geldim. Çin seçtin, kötü yolu seçtin ya. Kısa özet, neyse çok karışık olduğu için...
Şimdi şöyle giderken burada farklı farklı parçalar var bir sürü ya; minik işte sokaklar, kavşaklar, dönüşler vesaire falan. Siz bütün bu rotayı tek bir anda modellemeniz çok zor. Burada da biraz **Embedding** yöntemlerini kullanıyorlar. Ama burada şey challenge'lar da var; mesela uzun yollar var, kısa yollar var. Mesela kısayolun belki hani parçalara böldüğünde burada parçaları da... Bir kelime vardı, bir parçaya bir isim söylüyorlar ama unuttum ya... Unuttum ya neyse aklıma gelirse söyleyeceğim.
Burada tabii hepsini aynı boyuta getirmeye çalışıyorlar. Burada bazen işte bunları ufak ufak işte birleştirip... Yok Çankaya değildi ya farklı bir kelimeydi. Bu şeye özel biraz daha rotaya özel bir kelimeydi bu arada. Deep Learning'e özel bir kelime değil, birazcık böyle şey yapınca kelimesinin akıllara... Normal.
Burada tabii şey verilerini de kullanıyorlar; mesela **Attention** falan kullanıyorlar. Enes sen seversin Allah Allah. Burada çok değişik, zaten okurken şey oluyor; "Evet evet yani şey yapıyor, hatta lokasyonları Attention yapıyor."

(Enes F. Manan):
Yani doğru, yani bir dikkat mekanizması burada olması aslında bazı şeylerin öne atılması için. Yani burada bir yol seçimi... Bunu da sanırım var, olabilir ya. Bunlar o seçimde tam hatırlayamadım.

(Özge Usta):

Bir süre çünkü daha böyle ilk geldiğim zamanlarda okumuştum. Bu çok detaylı bu arada, yani bu hakikaten bunu yayınlamış ama bunda bile "Nasıl bir şey ya?" falan oluyorsun. İşte ondan sonra böyle Didi'nin, Lyft'in falan okuyunca da mesela... Asıl şey mi? Bu mesela şöyle bir yaklaşım yapıyordu; mesela bütün rotayı total hesaplamak yerine hepsini parça parça hesaplıyor. Ya da bazen şöyle bir yaklaşım var; mesela burada **Residual** diyor ya, şu residual'ı hesaplıyor, çıkarıyor. Buradaki hani ne kadar sapacak, olayı bile hesaplıyorlar. Yani böyle iyice abartıyorlar. Artık böyle Uber, Lyft ve Didi'nin böyle okursanız şeyi görüyorsunuz; artık böyle birbirlerine şov yapmaya dönmüşler. O kadar böyle deli dehşet mimariler çalışmışlar ki... Yani ben böyle parça parça okumam gerekmişti böyle net anlayabilmek için. O yüzden böyle ilk okuduğunuzda çok anlamazsanız kesinlikle kötü hissetmeyin.

Başka şeyin için nerede baksana, üzerine inanılmaz bir Ar-Ge var yani. **Bearing** falan çok büyük bir... [Anlaşılamadı] değildi, farklı bir şeydi ama gerçi hatırlayamıyorum. Lyft'in daha basit değil. Burada Lyft'in Medium sayfası var. Böyle ufak ufak paylaşımlar yapıyor. Üzerine bir yazıları... İşte burada zaten görüyorsunuz mesela 7 dakikada olacak, burada pickup süresi üzerine mesela odaklanmış yazarlar. Burada biraz **Long Tail** şuna değiniyor: Aslında diyor uzak mesafeler bu yakın mesafeleri daha zorlanıyoruz tahmin etmekte gibi bir yaklaşıma odaklanıp bunu biraz daha mesela çözmeye çalışmışlar. Burada metot olarak mesela ne yapmışlar? Tek sıkıntısı böyle çok genel yazıp geçiyorlar. Biraz Didi'nin mesela, bak; "Learning to Estimate Travel Time"... **Access Mobile Paper** falan mı ya? Ya diyorum ya Didi falan gerçekten çok çok dehşet işler yapıyorlar. Yani öyle bu davranışları şey yapmıyoruz bu arada, tasvip etmiyoruz.

(Enes F. Manan):

Neyi tasvip etmiyorsun?

(Özge Usta):

WhatsApp'tan açmalı falan demeyeceğim Allah Allah. Kullanım harika bir şey, ben kullanmıyorum. Burada iki ayrı şey kullanmıştı, bak eğer bunu şimdi hatırladım, aynı taraftan geliyor... Anlatıyordu ama gerçekten şu an detayları hatırlayamadığım için size çok anlatamıyorum. Ama böyle özellikle Deep Learning alanına ilginiz varsa falan ama hemen bununla başlamayın, bununla başlanmaz. Yani özellikle makale tarafında bence üçlü bayağı güzel yayınlar yapıyorlar. **Apple** ve **Grocery Delivery** tarafında pek iyi makale yok açıkçası. İşte bu DoorDash arada böyle kendi paylaş blog sayfasında paylaşıyor daha böyle ufak analizlerini. Tabii böyle çok detaylı paper ama son zamanlarda biraz paylaşmışlar, sonradan gördüm. Ben mesela Şirket 2'dayken okuduğum paylaşmamış... Hatta Enes bak, senin az önce bahsettiğin olaya DoorDash da değiniyor, onu bak göstermeyi unuttum.

Anadolu **Unobserved** dedikleri, yani aslında senin o gözlemleyemediğimiz oradan mesela bahsediyor. Bahsediyor bundan. Ya şöyle diyor; mesela orada **submit** ediliyor, almaya geliyor, aslında diyor şu kadar sürede hazırlanıyor diye ama bu burada almaya geldiği için sen bunu daha uzun görüyorsun. Selamdan bahsediyorsun. Buradan asılmış mesela. Ordu satılık, tamam bu tahmin edilen, burada bekliyor mesela. Bu zaten tamam, şurada mesela bir etmeye çalıştıklarından falan mesela bahsediyor. Hani aslında bu genel bir problem yani sektörde de. Orada bilinmeyen süreler oluyor, işte kuryeler, restoranlar kaynaklanan durumlar olabiliyor. Burada restoranda bazen "ben tamamladım siparişi" diyor, kurye gidiyor bekliyor, tam tersi durumlar da oluyor.

(Enes F. Manan):

Ya o kısmın da herhalde bir bonus tarafta şeyi var, pozitif etkisi var yani o puan artırmak için falan ama işte orada da yalan

yapabilir. Bu yalanı nasıl yakalayacaksın? İşte karşı tarafın da o zaman bunu onaylaması gerekir, **Double Check** yapmam lazım. Double Check'i de... Evet Gizem Hocam sizi buradan yazdığım şey mi, ETA mı yoksa daha **Ride-sharing** sevgili...
Şimdi bir mevzu da şu: Sen şimdi burada New York için yaptın bu modeli, alıp Konya'da kullanamazsın. İstanbul için yaptın, gidip de Çorum'da kullanamazsın. Şimdi her şehir için ayrı model mi eğiteceğiz?

(Özge Usta):

Güzel soru. Yani burada eğitim edebilir, bazen eğitiyor. Hatta ben şey demiştim; **Data Boat**'tan bir arkadaşla işte konuşmuştuk, bir şey demişti mesela, ülkelere böyle focus onlar çalışıyormuş. Yani aslında her ülkenin dinamiği de farklı olduğu için hani bu sadece ETA'ya değil de genel konuşuyorduk. O yüzden de baktığında yani bölgesel değildi ama bazen şey de yapılabiliyor, yani o da mesela bir... Ya sen belki de sadece şehirlere bunu vereceksin, o modeli eğiteceksin. Belki o bile işe yarayabilir. Yani senin ayrı ayrı eğer benzer bir sonuç alıyorsan ayrı ayrı modelleri hem uzatmaya çalışmazsın. Ama baktın iyi sonuç alamıyorsun, ayrı ayrı eğitmen gerekiyor. Ve İstanbul versus adres falan uyduruyorum; yani orada bir tane ağaç yapıyorum, geri kalan için **Regresyon** bas geç. Bu arada bazen oluyor yani böyle ufak bölgeler için düz ortalama alıp geçilebiliyor. Çünkü şu da var; atıyorum yani Çorum'u şey yapmış olmayayım ama Çorum'da çok trafik yoktur diye düşünüyorum. Öyle olunca hani o kadar da belki büyük bir modele ihtiyaç yok. Ama İstanbul gerçekten çok karmaşık. İstanbul modellemesi de zor. Belki de işte Çorum'da yapacağın model çok iyi çalışacak çünkü daha pratik bu ama İstanbul daha bol olduğu için İstanbul'da yaptığın model daha kötü çalışabilir.
İşte orada o hatayı ne kadar **Tolerate** edeceğine bağlı. Şirket 2'da sana 5 dakika geç getirmesi problem değil ama Uber'de sana 5 dakika deyip 10 dakikada gelmesi problem bence. Çünkü belki dışarıda soğukta bekliyorsun o aracı sen. O yüzden de hata payı daha kritik oluyor. Hani belki de o yüzden firmalar yeterince altyapısı yokken hemen Google'dan... İşte hemen biz ETA modelimizi yapalım demiyorlar. Daniel (zor) bir problem bu arada, öyle kolay bir şey değil yani.

(Enes F. Manan):

Hocam dediğin gibi bence Uber Master. Çünkü mesela şeydeki, Şirket 2'daki gayet okey. Yani Şirket 2 gibi bir firmaysa orada bir şey göstermek yani kullanıcı... He evet yani restoranı aradı, işte restoranın numarası var, arıyorsun "nerede bu kardeşim?" diye. O zaman "Abi yeni çıktı, sıcak çıktı hemen geliyor" hem de aynı yalan ya. Bir de şunu söyleyebilirim yani Uber üç kuruş para için bu kadar yeterli uğraşmaz gerçekten. Ya çok iyi kar ediyordur yani çünkü Uber çok zengin bir firma. Yani görüyorsunuz herhangi bir ülkeye girdiğinde işte mesela çok iyi **Marketing** yapar, çok iyi işte kampanyalar yapar çünkü gerçekten iyi bir bütçesi var. Proje yapmak için gerçekten hani bir şekilde Google ne bileyim yavaş kalıyor olabilir onun için. Burada bilmiyorum, **insight** olarak tamamen fikirlerimi söylüyorum. O kadar işte belki yüz milyonlarca, belki de baktığınızda dünyada birçok yerde var. Ya da gerçekten hani bu modeli yaptıklarında aradaki kar marjı çok iyi. Yani bir şekilde bundan ciddi bir kazanç sağlıyorlar ki zamansal ya da parasal bu işe giriyorlar ve bununla ilgili makaleler yayınlıyorlar. O kadar işte belki 5-10 kişi, belki 20-30 kişilik bir ekip bunun üzerine çalışıyor. O zaman önemli bir proje bence. Tabii ben böyle daha en temel sıfırdan nasıl yaparsınız gibi anlattım ama ileri seviyeleri vesaire böyle farklı farklı işte **Attention** falan farklı farklı...

(Enes F. Manan):

Parasal olarak bunun şeyi bir öncesi ne olacak? Yani bu modeli yaptımda bu model burada çıktı, şirkete ne getiriyor evde tarafından düşünüyorum. Hani gerçekten şey getirebilir, hani müşteri memnuniyeti. Onun haricinde belki **Customer** hareket, kampanya, bir şeyler düşünebilir. Ya bir de şey var, dedim ya riski yolu tahmin ettim olayı. Yani adamlar Google'ın da üstüne geçmeye çalışıyor. Düşüneceğim ama acaba birbiriyle ufak bir kapışma halindedir.

(Özge Usta):

Bence şey yapıyorlar çünkü hepsi en iyi mühendisleri almak istiyor. Bunları da tabii kendilerine çekmek için "Bakın biz bunları yapıyoruz, biz bunları geliştiriyoruz" işte blog yazıları yazıyorlar, makalelerini paylaşıyorlar vesaire. Bence hiçbir firma o an yaptığı şeyi paylaşmaz yani her zaman zaten onun üstüne geçmişti, öncekini paylaşır diye ben düşünüyorum. Çünkü öbür türlü sırrını vermiş olursun. O yüzden yani bunları görünce diyorum ki o arkada çok daha büyük şeyler var ki hani bunlar paylaşılmaya başlandı diye düşünüyorum.

(Enes F. Manan):

O sırada ortalama alan bir s****

Yapacak bir şey yok. Google ya, Google'da iyidir. Azıcık da geciktin ne olacak ya, birkaç saniye bekleyiverelim.

Şimdi o zaman güzel yayın oldu, eğlenceliydi arkadaşlar. Sizin sorularınız varsa son soruları alalım. Bir tane fotoğraf çektirelim, yavaştan da kapatalım yani.

(Katılımcı):

Tedarik zinciri konusunda acaba neler yapılıyor diye düşündüm ben Enes hocam, Özge hocam. **Route Optimization** yapılıyor, **Gezgin Satıcı** var. **Transportation Management** deniyor. Türkiye'de pek çok yerlere gidiyor ya da dülke dışında ışında pek çok yerlere gidiliyor. Tedarik süresi hesaplanıyor, Route optimizasyonu hesaplanıyor. Burada Transportation Management ile ilgili ML yani makine öğrenmesi kullanılıyor mu kullanılmıyor mu bilmiyorum.

(Özge Usta):

Orada şey diyebilirim; bu **Traveling Salesman** bayağı aslında popüler bir konu. Endüstri mühendisliğinde bizim de derslerimizde vardı bu. Orada tabii biraz daha aslında sayıyı direkt tahmin etme değil de ETA'yı kullanıp bir optimizasyon. Orada farklı noktalar var. Dağlar adından da anlaşılacağı gibi bir işte seviyorsun ben dolaşıyor gibi düşünebiliriz de en optimize nasıl dolaşmalı? Yani oradaki rotayı en kısa nasıl tur atmalı gibi oluyor. Orada işte noktalar, uzaklıkları düşünürsek bizim **Operations Research** dediğimiz alana kapsamında. Hatta bu aralar şunu da çok görüyorum yeni ilanlarda da; **Operations Research** falan yani optimizasyon bilen Data Scientist aramaya... Belki yok gerçekten yok ya ben aranan ilanları görüyorum, birini bulamıyorlar.

Orada endüstri mühendisleri, matematik, endüstri orada kalsın abi karışmasın diye ama şöyle aslında; mesela ilk aslında mezun olmadan önce çok Operations Research tarafına devam etmek istiyordum ama maalesef Türkiye'de "kaçtı yok" demeyeyim de, yani böyle "çok hızlıca çözelim çok bu işlere girmeyelim" gibi bakıyorlar. Aslında dünyada çok gerçekten optimizasyon projeleri çok önemli, yeni yeni biraz değeri biliniyor ama maalesef bütün yurt dışına gitti işte ben dedim "yok olmayacak" dedim, "ben hiç bulamam burada" değil. Mesela daha önerdim gibi oldu ama gerçekten önemli bir konu orası da. Bence tabii şu an AI ile birlikte birleşip nerelere gider bilemiyorum ama... Çok teşekkürler bu arada yaptığınız için yani eğer merak edenler varsa bence **Traveling Salesman**...

(Enes F. Manan):

Gezgin satıcı problemi güzel ama bu Operations Research dediğimiz şey çok bela bir şey. Özellikle ben derslerinden nefret ediyordum. Yani bu **Simplex Algoritması**, işte ne bileyim 2 aşamalı yöntem falan, bir de elle hesaplıyorsun. Ben o yüzden bu şeyi aşırı uzağım.

(Özge Usta):

Ya öyle yapıyorlar, derslerde öyle yaptırıyorlar. Ben sevmiyorum, arzu etmiyorum. Ya farklı bir konu, ilgisi olan varsa tabii ki de direkt buraya yönlendirin diyemem. Olan varsa bence güzel bir konu. Dediğim gibi biraz maalesef Türkiye'de iş alanı az bir alan. Bir tık daha niş bir alan yani her alanda kullanılmıyor. Mesela Şirket 2'da da bu arada optimizasyon ile ilgili çalışanlar var. Orada da mesela işte bu kurye dağıtımları gibi farklı farklı problemler var. Burada da **Recommendation**'da da bazen aldığı ile ilgili track'ler duydum ama biraz daha böyle işin artık "Tamam biz bu alanda ilerledik, optimizasyon da kullanalım."
Hem mesela **Matching** dediğimiz bir problem var. Burada amacımız işte sürücülerle yolcuları eşlemek. Burada da mesela optimizasyon algoritmaları kullanılıyor. Yani aslında sektörde yavaş yavaş artıyor ama tabii geleceği nereye gider, daha mı çok artar, böyle patlar mı, böyle kendi halinde 3-5 tane ilanla devam eder orayı ben de bilemiyorum.

(Enes F. Manan):

Ya ben zamanında şeyle görüşmüştüm, Flo ile görüşmüş. Flo'da daha çok ve kargolama optimizasyonu üzerine çalışmalar vardı. İşte kutuların içerisine ne bileyim ayakkabı kutusu veya bir şeyleri kaç tane sığdırır? Ben bunun üzerine birazcık daha böyle matematiksel ve görüntü işleme projeler vardı. Ben de dedim "Yok ya ben almayayım." İlginç sözler ama genelde çoğu zaman asıl Cemil Hoca'nın bahsettiği hani **Supply Chain** tarafında çok kullanılan optimizasyon. Daha böyle e-ticarette falan o kadar da girmiyorlar. Bankada falan bilmiyorum Enes sen daha iyi açık olabilir aslında **Heuristic** kullanamazsınız belki de numarasını.
Akşam bir saçınızı başınızı düzeltin şöyle bir bazılarıyla yani kaç kişiyle görüşmüyorum ama alanda da görüştük ya bir off-site etkinliği vardı Data Science konferansı, oradan da böyle gördüklerim var. Bayağı bir kişi açtı teşekkür ediyorum. Hazırsanız o zaman alacağım. Alıyorum... Bir daha açtı ama gülün biraz. Tamam alıyorum şimdi. Alıyorum... Bir tane aldım vallahi güzel oldu ama tam Mehmet Onur hocam yeni kamerasını açmıştı. Bir tane daha alıyorum o zaman. Bu güzel oldu ya herkes bunda güzel çıktı. Piksel piksel çıkıyor ben de biraz kameram kirlenmiş olabilir değiştireceğim ya bu bilgisayarı. O zaman çok teşekkür ederiz, güzel oldu gerçekten samimi de bir yayın oldu. Farklı alan konusunda bilgisayarımı ben de hiç duymadım ya, enteresan da vallahi. En çok **H3** şey oldu ya böyle, değer gördü değil hahaha.
Bir sonraki oturum çarşamba günü olacak **Zaman Serileri** üzerine. Enerjisa tarafında çalışan bir arkadaşımız gelip anlatacak. Bir sonraki oturumda görüşmek üzere diyelim, herkesin iyi akşamlar, çok teşekkür ederiz.

(Özge Usta):

Teşekkür ederiz.

(Enes F. Manan):

Evet merhabalar şu an...

Önemli Noktalar ve Anahtar Kelimeler (Belirginleştirilmiş)

- **Problemin Tanımı: ETA (Estimated Time of Arrival) Prediction** (Tahmini Varış Süresi) problemi; **Uber**, **Lyft**, **Didi**, **Şirket 1**, **Şirket 2** gibi **Ride-sharing** ve teslimat firmalarının temel sorunlarından biridir.
- **Kullanılan Yöntemler:**

    - **Deep Learning (Derin Öğrenme):** Uber ve Didi gibi büyük firmalar, rotayı parçalara (segments) bölüp **Embedding** ve **Attention** mekanizmaları kullanarak çok detaylı modeller eğitmektedir.
    - **Tree-based (Ağaç Tabanlı) Modeller:** Sunumda daha çok **XGBoost/LightGBM** gibi ağaç tabanlı modeller tercih edilmiştir.
    - **Google Maps API Maliyeti:** Firmaların kendi modellerini geliştirmesinin ana sebebi, sürekli API çağrısı yapmanın **maliyetli** olması ve **latency** (gecikme) sorunlarıdır.
- **Veri Zorlukları:**
    - **Long Tail (Uzun Kuyruk) Dağılımı:** ETA verileri normal dağılım göstermez; kısa sürüşler çoğunluktadır ancak uzun sürüşler (kuyruk kısmı) tahmin edilmesi zor outlier'lar içerir.
    - **Outlier Detection:** 1 saniyelik veya 900 saatlik sürüşler gibi hatalı verilerin temizlenmesi gerekir.
    - **Label Noise:** Sürücülerin/kuryelerin uygulamayı geç/erken kapatması veriyi kirletir.
- **Feature Engineering (Özellik Mühendisliği):**
    - **Zaman Verileri:** Gün, saat, hafta sonu bilgisi.
    - **Coğrafi Veriler: Haversine Distance** (Kuş uçuşu mesafe), **Bearing** (Yön açısı), **Manhattan Distance**.
    - **H3 (Hexagonal Hierarchical Spatial Index): Uber**'in geliştirdiği, haritayı altıgenlere bölerek lokasyon bazlı feature (yoğunluk, ortalama hız vb.) üretmeyi sağlayan kütüphane. Model başarısını en çok artıran özelliklerden biridir.
    - **PCA (Principal Component Analysis):** Koordinatları dönüştürerek modelin daha iyi öğrenmesini sağlamak için kullanılabilir.
- **Metrikler:**
    - **Mean Squared Error (MSE):** Büyük hataları (uzun mesafedeki sapmaları) daha çok cezalandırmak için tercih edilir.
- **Optimizasyon ve Diğer Alanlar:**
    - **Operations Research (Yöneylem Araştırması):** Rota optimizasyonu ve **Traveling Salesman Problem** (Gezgin Satıcı Problemi) ile ilişkilidir.
    - **Teacher-Student Networks:** Büyük modelleri hızlandırmak (latency düşürmek) için kullanılan bir tekniktir.