

PaySim Verisiyle Fraud Detection: Random Forest ve Isolation Forest Karşılaştırması



Deryakalay 6 min read · Just now



Finansal işlemlerde dolandırıcılık tespiti, makine öğrenmesinin en ilgi çekici kullanım alanlarından biri. Ancak bu problemde yalnızca yüksek doğruluk oranına ulaşmak yeterli değil.

Çünkü fraud veri setlerinde normal işlemler çok fazla, dolandırıcılık işlemleri ise oldukça azdır. Bu nedenle bir model hiçbir fraud işlemini yakalamadan bile yüksek bir accuracy değerine ulaşabilir.

Bu projede **PaySim veri setini kullanarak finansal işlemlerde fraud tespiti** üzerine çalıştım. Amacım yalnızca bir sınıflandırma modeli kurmak değil; aynı zamanda gözetimli öğrenme ile anomali tespit yaklaşımının aynı problem üzerindeki davranışını karşılaştırmaktı.

Bu nedenle iki farklı yöntem kullandım:

- **Random Forest**
- **Isolation Forest**

1. Projenin Amacı

Projede yanıt aradığım temel soru şuydu:

Bir finansal işlem gerçekleşmeden önce elimizde bulunan bilgiler kullanılarak fraud işlemler normal işlemlerden ayırt edilebilir mi?

Burada özellikle “işlem gerçekleşmeden önce” kısmı önemli.

Çünkü veri setinde işlem gerçekleştikten sonra oluşan bazı bakiye bilgileri bulunuyor. Bu değişkenleri modele dahil etmek model performansını artırabilir; ancak gerçek hayatta fraud kararı verildiği anda bu bilgiler henüz mevcut olmayabilir.

Bu nedenle final modelde yalnızca işlem öncesinde veya işlem anında bilinebilecek değişkenleri kullandım.

2. Veri Seti: PaySim

PaySim, mobil para transferlerini simüle eden sentetik bir finansal işlem veri setidir.

Çalıştığım veri setinde:

- **6.362.620 işlem**
- **8.213 fraud işlem**
- **Yaklaşık %0,129 fraud oranı**

bulunuyordu.

Bu oran problemin ne kadar dengesiz olduğunu açıkça gösteriyor.

Başka bir ifadeyle yaklaşık her 1.000 işlemin yalnızca 1–2 tanesi fraud.

Bu nedenle model değerlendirmesinde yalnızca accuracy kullanmak yanıltıcı olacaktı.

3. İlk Adım: Fraud Hangi İşlemlerde Gerçekleşiyor?

Keşifsel veri analizi sırasında ilk olarak işlem türlerinin dağılımını ve fraud işlemlerin hangi kategorilerde görüldüğünü inceledim.

PaySim içerisinde farklı işlem tipleri bulunuyor:

- CASH_IN
- CASH_OUT
- DEBIT
- PAYMENT
- TRANSFER

Analiz sonucunda fraud işlemlerin yalnızca:

TRANSFER

ve

CASH_OUT

işlemlerinde bulunduğunu gördüm.

Bu nedenle keşifsel veri analizini tüm PaySim verisi üzerinde gerçekleştirirken, modelleme aşamasında TRANSFER ve CASH_OUT işlemlerine odaklandım.

Bunun önemli bir nedeni vardı.

Fraud bulunmayan PAYMENT veya CASH_IN gibi işlem türlerini modele dahil ettiğimizde model çok kolay bir kural öğrenebilirdi:

“Bu işlem PAYMENT ise fraud değildir.”

Bu durumda model fraud davranışını öğrenmek yerine doğrudan işlem türünü kullanarak kolay bir ayırım yapabiliirdi.

Modeli TRANSFER ve CASH_OUT işlemleriyle sınırlandırmak problemi daha anlamlı hale getirdi.

4. Hipotezim

Projeye başlamadan önce temel hipotezimi şu şekilde oluşturdum:

H0 — Null Hipotezi

İşlem öncesinde bilinebilen PaySim değişkenleri fraud ve normal işlemleri anlamlı biçimde ayırt etmek için yeterli değildir.

H1 — Alternatif Hipotez

İşlem öncesinde bilinebilen PaySim değişkenleri kullanılarak fraud işlemler normal işlemlerden anlamlı biçimde ayrılabilir.

Projede elde ettiğim model sonuçlarını bu hipotezi değerlendirmek için kullandım.

5. Data Leakage Konusuna Özellikle Dikkat Ettim

Fraud projelerinde model performansını yapay biçimde artırabilecek en önemli problemlerden biri **data leakage**, yani veri sızıntısıdır.

PaySim veri setinde şu değişkenler bulunuyor:

- oldbalanceOrg
- newbalanceOrig
- oldbalanceDest
- newbalanceDest

newbalanceOrig ve newbalanceDest , işlem gerçekleştikten sonraki bakiyeyi gösteriyor.

Fraud sisteminin amacı ise ideal olarak işlem gerçekleşmeden veya gerçekleştiği anda riskli işlemi belirlemek.

Bu nedenle final modelimde işlem sonrası bilgileri kullanmadım.

Kullandığım temel değişkenler:

- step
- type
- amount

- oldbalanceOrg
- oldbalanceDest

oldu.

Ayrıca işlem anında hesaplanabilecek bazı yeni özellikler oluşturdum.

Örneğin:

- işlemin gün içerisindeki yaklaşık saati
- işlem tutarının gönderici bakiyesine oranı
- gönderici bakiyesinin işlem tutarı için yeterli olup olmadığı
- alıcının işlem öncesi bakiyesinin sıfır olup olmadığı

Bu sayede yalnızca ham değişkenlere bağlı kalmadan bazı davranışsal sinyaller de modele dahil edilmiş oldu.

6. Neden Accuracy Tek Başına Yeterli Değil?

Veri setindeki işlemlerin yaklaşık %99,87'si normal.

Bu durumda tüm işlemlere:

“Fraud değil.”

diyen bir model bile yaklaşık %99,87 accuracy elde edebilir.

Ancak böyle bir model tek bir fraud işlemini bile yakalayamaz.

Bu nedenle çalışmada şu metriklere odaklandım:

Precision

Model fraud dediğinde ne kadar doğru?

Recall

Gerçek fraud işlemlerinin ne kadarını yakalayabiliyoruz?

F1 Score

Precision ve Recall arasındaki dengeyi gösteriyor.

PR-AUC

Özellikle dengesiz sınıflandırma problemlerinde model performansını değerlendirmek için önemli bir metrik.

ROC-AUC değerini de hesapladım ancak fraud gibi azınlık sınıfının çok küçük olduğu problemlerde PR-AUC sonucunu özellikle dikkate aldım.

7. Model 1: Random Forest

İlk model olarak Random Forest Classifier kullandım.

Random Forest gözetimli bir algoritma olduğu için eğitim sırasında işlemlerin gerçekten fraud olup olmadığını biliyor.

Başka bir ifadeyle model geçmiş fraud örneklerini inceleyerek fraud davranışını öğrenmeye çalışıyor.

Sınıf dengesizliği nedeniyle modelde:

```
class_weight="balanced_subsample"
```

yaklaşımını kullandım.

Bu sayede fraud sınıfının eğitim sırasında tamamen normal işlemlerin altında ezilmesini azaltmaya çalıştım.

8. Model 2: Isolation Forest

İkinci yaklaşım olarak Isolation Forest kullandım.

Burada temel fikir tamamen farklı.

Isolation Forest eğitim sırasında:

```
isFraud
```

etiketini kullanmıyor.

Model normal işlemlerin genel davranışını öğreniyor ve bu davranıştan ciddi şekilde uzaklaşan işlemleri **anomali** olarak değerlendiriyor.

Bu nedenle Isolation Forest'ın amacı Random Forest'ı geçmek değildi.

Asıl soru şuydu:

Fraud etiketini hiç görmeden yalnızca sıra dışı işlem davranışlarını inceleyen bir algoritma fraud işlemlerini ne kadar yakalayabilir?

Isolation Forest'ı mümkün olduğunca normal davranışı öğrenmeye yönlendirmek için yalnızca normal eğitim işlemleri üzerinde eğittim.

9. Train, Validation ve Test Ayrımı

Veriyi üç parçaya ayırdım:

- %60 Train
- %20 Validation
- %20 Test

Buradaki validation setinin önemli bir görevi vardı:

Threshold seçimi.

Modelin varsayılan 0.50 threshold değerini doğrudan kabul etmek yerine, validation setinde Precision–Recall davranışını inceleyerek F1 skorunu optimize eden threshold değerini belirledim.

Test setini ise threshold seçiminde kullanmadım.

Bu sayede test verisi yalnızca nihai değerlendirme için kullanıldı.

10. Sonuçlar

Final test sonuçlarında aşağıdaki değerleri elde ettim:

Model	Precision	Recall	F1	ROC-AUC	PR-AUC	Random
Forest	0.948	0.810	0.874	0.997	0.910	Isolation
Forest	0.630	0.611	0.620	0.993	0.619	2

Sonuçlar Random Forest'in fraud tespitinde belirgin biçimde daha güçlü olduğunu gösterdi.

Random Forest:

- Fraud olarak işaretlediği işlemlerde yaklaşık **%94,9 precision**
- Gerçek fraud işlemlerinde yaklaşık **%81,1 recall**
- **0,874 F1**
- **0,910 PR-AUC**

değerine ulaştı.

Bu sonuçlar başlangıçtaki H1 hipotezimi destekledi.

Yani yalnızca işlem öncesinde kullanılabilecek bilgilerle fraud ve normal işlemler arasında güçlü bir ayrım yapılabildi.

11. Confusion Matrix Bize Ne Söylüyor?

Test verisinde Random Forest sonuçlarını yalnızca oranlarla değil, işlem sayılarıyla da değerlendirdim.

Model:

- **240 fraud işlemi doğru yakaladı**
- **56 fraud işlemi kaçırdı**
- **Yalnızca 13 normal işlemi yanlışlıkla fraud olarak işaretledi**

Bu noktada fraud probleminin iş tarafı ortaya çıkıyor.

False Negative

Fraud olduğu halde normal kabul edilen işlemdir.

Bu durum doğrudan finansal kayıp anlamına gelebilir.

False Positive

Normal olduğu halde fraud olarak işaretlenen işlemdir.

Bu durumda ise:

- işlem gereksiz yere durdurulabilir,
- müşteri ek doğrulamaya yönlendirilebilir,
- manuel inceleme ihtiyacı oluşabilir,
- müşteri deneyimi olumsuz etkilenebilir.

Dolayısıyla fraud modelinde yalnızca:

“Kaç fraud yakaladık?”

sorusuna değil,

“Fraud yakalamak için kaç normal müşteriyi yanlış alarma düşürdük?”

sorusuna da bakmak gerekiyor.

12. Threshold Aslında Bir İş Kararı

Projede benim için en öğretici bölümlerden biri threshold analizi oldu.

Random Forest’ın validation setinde belirlenen threshold değeri yaklaşık:

0.4225

olarak bulundu.

Ancak farklı threshold değerleri test edildiğinde önemli bir trade-off ortaya çıktı.

ThresholdPrecisionRecallFalse PositiveFalse

Negative0.300.85570.861543410.400.94900.817613540.42250.94860.810813560.

500.97000.76357700.600.98130.7095486

Burada çok net bir ilişki görüyoruz.

Threshold düşürüldüğünde:

Recall artıyor.

Yani daha fazla fraud yakalıyoruz.

Ancak aynı zamanda daha fazla normal işlem fraud olarak işaretleniyor.

Threshold yükseltildiğinde ise:

Precision artıyor ve yanlış alarm azalıyor.

Ancak daha fazla gerçek fraud kaçırılıyor.

Dolayısıyla aslında tek bir “en iyi threshold” yok.

Threshold değeri şirketin risk yaklaşımına göre değişebilir.

Örneğin çok yüksek finansal kayıp riski olan işlemlerde daha düşük threshold tercih edilebilir.

Müşteri deneyiminin kritik olduğu düşük riskli işlemlerde ise daha yüksek threshold kullanılabilir.

Bu nedenle threshold seçiminin yalnızca bir makine öğrenmesi parametresi değil, aynı zamanda iş ve risk yönetimi kararı olduğunu düşünüyorum.

13. Random Forest mı Isolation Forest mı?

Sonuçlara baktığımızda Random Forest açık biçimde daha başarılı.

Ancak bu iki algoritmanın kullanım amacı aslında tamamen aynı değil.

Random Forest

Geçmiş fraud örneklerinden öğreniyor.

Bu nedenle:

“Daha önce gördüğüm fraud davranışına benzeyen bir işlem var mı?”

sorusuna güçlü cevap veriyor.

Isolation Forest

Fraud etiketini görmüyor.

Daha çok:

“Bu işlem normal işlem davranışından ne kadar farklı?”

sorusunu cevaplıyor.

Dolayısıyla gerçek bir fraud sisteminde bu iki yaklaşımı rakip modeller olarak değil, birbirini tamamlayan iki farklı sinyal olarak değerlendirmek mümkün.

Örneğin:

Random Forest → Fraud Probability

Isolation Forest → Anomaly Score

üretebilir.

Bu iki skor daha sonra bir risk motorunda birlikte değerlendirilebilir.

14. Projeden Çıkardığım En Önemli Dersler

Bu proje sırasında model kurmanın ötesinde birkaç önemli nokta öğrendim.

1. Accuracy fraud problemlerinde yanıltıcı olabilir

Özellikle ciddi sınıf dengesizliği varsa accuracy tek başına neredeyse hiçbir şey söylemeyebilir.

2. Data leakage modeli olduğundan çok başarılı gösterebilir

Gerçek kullanım anında bulunmayan değişkenleri modele vermek etkileyici sonuçlar üretebilir; ancak böyle bir model gerçek hayatta kullanılamaz.

3. Fraud detection yalnızca classification değildir

Anomaly detection algoritmaları da özellikle bilinmeyen ve yeni davranışları araştırmak için değerli olabilir.

4. Threshold modelin kendisi kadar önemli olabilir

Model aynı kalmasına rağmen threshold değiştiğinde fraud yakalama ve yanlış alarm davranışı ciddi biçimde değişiyor.

5. Teknik başarı ile iş başarısı aynı şey değildir

En yüksek F1 skoruna sahip model her durumda en uygun fraud sistemi olmayabilir.

Fraud kaynaklı kaybın maliyeti, manuel inceleme maliyeti ve müşteri deneyimi birlikte değerlendirilmelidir.

Sonuç

Bu projede PaySim veri seti üzerinde hem gözetimli sınıflandırma hem de anomali tespit yaklaşımını uyguladım.

Random Forest final modelinde:

Precision: %94,86

Recall: %81,08

F1: %87,43

PR-AUC: %91,01

sonuçlarına ulaştım.

Isolation Forest ise fraud etiketlerini kullanmadan anlamlı ölçüde anomali yakalayabildi; ancak Random Forest'ın gerisinde kaldı.

Bu sonuçlarla başlangıçta oluşturduğum:

“İşlem öncesinde bilinebilen PaySim değişkenleri kullanılarak fraud işlemler normal işlemlerden anlamlı biçimde ayrılabilir.”

hipotezi desteklendi.

Ancak burada önemli bir sınırlılığın da belirtmek gerekiyor:

PaySim sentetik bir veri seti.

Dolayısıyla elde edilen sonuçlar gerçek bir banka veya ödeme kuruluşunun fraud sisteminde aynı performansın elde edileceği anlamına gelmiyor.

Buna rağmen proje; sınıf dengesizliği, data leakage, feature engineering, supervised learning, anomaly detection, threshold optimizasyonu ve fraud modellerinin iş açısından yorumlanması gibi birçok konuyu tek bir çalışma içerisinde değerlendirmeme imkan sağladı.

Benim için bu projenin en önemli çıktısı yalnızca yüksek bir model skoru elde etmek değil, bir fraud modeline şu soruyla bakmaya başlamak oldu:

“Model ne kadar doğru?” yerine, “Modelin verdiği kararın operasyonel ve finansal karşılığı nedir?”



Written by Deryakalay

0 followers · 1 following

Edit profile

[Help](#) [Status](#) [About](#) [Careers](#) [Press](#) [Blog](#) [Store](#) [Privacy](#) [Rules](#) [Terms](#) [Text to speech](#)