

---

## 1. Aşama: Problemi Tanımlama ve Strateji 🎯

Her şeyden önce "Neyi çözüyoruz?" sorusuyla başladım. Elimizde 18.000 farklı ürün ve 500.000 siparişlik bir veri seti vardı. İki büyük riskimiz mevcuttu:

1. **Yok Satmak (Stockout):** Müşteri ürünü istiyor ama bizde yok (Ciro kaybı).
2. **Aşırı Stok (Overstock):** Ürün satılmıyor, depoda yer kaplıyor (Maliyet).

Veriye ilk baktığımda, satışların **seyrek (intermittent)** olduğunu gördüm. Yani bazı günler hiç satış yok, bazı günler aniden artıyor. Bu yüzden klasik yöntemler yerine **Derin Öğrenme (Deep Learning)** kullanmaya karar verdim.

---

## 2. Aşama: Veri Mühendisliği ve "Agregasyon" Sanatı 🏗️

Veriyi modele sokmadan önce onu "anlaşılır" hale getirmem gerekiyordu. main.py dosyasında şu işlemleri yaptım:

- **Temizlik ve Formatlama:** Tarihleri standartlaştırdım ve kategorileri (1: Gıda, 3: Elektronik gibi) isimlendirdim.
- **Seyrek Veri Çözümü (Aggregation):** İşte burası kritikti. Veri çok seyrek olduğu için modelin "0" görüp yanılmasını engellemek adına veriyi üç farklı zaman diliminde toplulaştırdım:
  1. **Aylık (Monthly):** Genel trendi görmek için.
  2. **Haftalık (Weekly):** Daha hassas takip için.
  3. **3 Aylık (Quarterly - Çeyreklik):** Bunu sonradan ekledim çünkü haftalık verideki "sıfırları" yutup ana dalgayı yakalamak için en iyi yöntem buydu.

---

## 3. Aşama: Yapay Zeka Motorunu Kurma (Forecasting) 🧠

İşin beyni forecasting.py dosyasıdır. Burada GRU ve LSTM olmak üzere iki farklı sinir ağı mimarisi kurdum.

- **Neden GRU ve LSTM?** Bu modeller "unutma kapıları" sayesinde geçmişteki uzun vadeli trendleri hafızasında tutabilir.
- **Model Mimarisi:**
  - Veriyi 0-1 arasına sıkıştırmak için MinMaxScaler kullandım (Yoksa model sapıttırdı).
  - Ezberlemeyi (Overfitting) önlemek için %20 Dropout (bazı nöronları rastgele kapatma) ve EarlyStopping (öğrenme durunca eğitimi kesme) tekniklerini ekledim.
- **İnovasyon (Pinball Loss):** Standart model "ortalama" satış tahmin eder (MSE). Ancak ben, stoksuz kalmamak için "**Güvenli Model**" de geliştirdim. Bunun için pinball\_loss fonksiyonunu yazdım. Bu fonksiyon modele şunu der: *"Bana ortalamayı verme, bana %90 ihtimalle stokun yeteceği üst sınırı ver".*

---

## 4. Aşama: Şampiyonlar Ligi (Model Yarıştırma) 🏆

Tek bir modele güvenemezdik. Bu yüzden run\_analysis.py içinde bir "Turnuva" sistemi kurdum.

- Sistem her ürün için **6 farklı senaryoyu** (Aylık-GRU, Aylık-LSTM, Haftalık-GRU... Çeyreklik-LSTM) tek tek dener.
- Geçmiş veriyi (Validation seti) en az hatayla (RMSE) tahmin eden kombinasyonu o ürünün **"Best Model"**i (En İyi Modeli) olarak seçer ve kaydeder.
- Böylece "Gıda" ürünleri için haftalık analiz çalışırken, yavaş satan "Mobilya" ürünleri için 3 aylık analiz otomatik olarak devreye girer.

---

## 5. Aşama: Müşteriyi Tanıma (RFM Analizi) 👤

Sadece "ne kadar satacağımızı" bilmek yetmez, "kime satacağımızı" da bilmeliyiz. Özellikle elimizde **Aşırı Stok** varsa, bunu kime satacağız?

- rfm.py dosyasında müşterileri 3 metriğe göre puanladım:
  - **Recency**: En son ne zaman geldi?
  - **Frequency**: Ne sıklıkla alıyor?
  - **Monetary**: Ne kadar para bırakıyor?.
- Bu puanlara göre müşterileri "Sadıklar", "Uykuda Olanlar", "Şampiyonlar" gibi segmentlere ayırdım. Dashboard'da aşırı stoklu bir ürüne tıkladığında, o ürünü en çok alan müşteri kitlesini görebilmeni sağladım.

---

## 6. Aşama: Ürünü Paketleme (Dashboard) 💻

Son olarak, tüm bu karmaşık matematiği yöneticinin kullanabileceği basit bir arayüze (app.py) dönüştürdüm. Streamlit kullanarak 3 sekmeli bir yapı kurdum:

1. **Genel Tablo**: Burada "Acil Durum" çanları çalar. Stokun bitmek üzere olduğu ürünleri en üste taşıdım.
2. **Overstock & RFM**: Depoda şişen ürünleri ve bunları hangi müşterilere (RFM segmentlerine) kampanya yaparak satabileceğimizi gösterdim.
3. **Simülasyon**: Burası yöneticinin oyun alanı. Bir ürün seçip modelin (MSE vs Pinball) nasıl tahmin yaptığını grafik üzerinde görebiliyor. veya gibi görsellerde gördüğün o karşılaştırmalı çizgileri (Standart vs Güvenli Tahmin) burada çizdirdim.

## Sonuç: Ne Başardım?

Sadece kod yazmadım; **veri odaklı bir karar destek sistemi** kurdum.

- Haftalık verideki gürültüyü **3 aylık analizle** aştım.
- Risk almak istemeyen yöneticiler için **Pinball Loss** ile "Güvenli Stok" tahminini geliştirdim.

- Elimdeki stoęu kime satacađımı söyleyen **RFM entegrasyonunu** yaptım.

Bu proje, bir řirketin milyonlarca liralık stok maliyetini kurtarabilecek canlı bir organizmadır.