# DATASCIENCE-1

Emirhan Köksal,Tuna Başkurt,Alper Özgür Şahin,İrem Nur Şener

June 2024

## Veri Setinin Özeti

Bu veri seti, Rohlik Group'un Avrupa genelindeki 11 depoda toplam sipariş sayısını tahmin etmek için kullanılmaktadır. Depolardaki günlük sipariş sayısı, tarih, resmi tatiller, depo kapanmaları gibi çeşitli faktörlere bağlı olarak değişiklik gösterebilmektedir. Tahminlerin doğruluğu, iş gücü tahsisi, teslimat lojistiği, envanter yönetimi ve tedarik zinciri verimliliği gibi planlama süreçlerini etkilemektedir. Bu nedenle doğru tahminlerle operasyonel verimliliği artırmak ve sürdürülebilirlik hedeflerine ulaşmak hedeflenmektedir.

- **warehouse (depo adı)**: Depo ismi.
- **date (tarih)**: Tarih.
- **orders (siparişler)**: Depoya atfedilen müşteri siparişlerinin sayısı.
- **holiday_name (tatil adı)**: Resmi tatilin adı (varsa).
- **holiday (tatil)**: Resmi tatilin varlığını gösteren bir gösterge (0/1).
- **shutdown (kapanma)**: Operasyonel nedenlerle depo kapanması veya sınırlaması (testte verilmemiştir).
- **mini_shutdown (kısmi kapanma)**: Operasyonel nedenlerle depo kapanması veya sınırlaması (testte verilmemiştir).
- **shops_closed (dükkanların kapalı olması)**: Çoğu dükkanın veya büyük bir kısmının kapalı olduğu resmi tatil.
- **winter_school_holidays (kış okulu tatilleri)**: Kış okulu tatilleri.
- **school_holidays (okul tatilleri)**: Okul tatilleri.
- **blackout (elektrik kesintisi)**: Operasyonel nedenlerle depo kapanması veya sınırlaması (testte verilmemiştir).
- **mov_change (minimum sipariş değeri değişikliği)**: Minimum sipariş değerinde değişiklik, müşteri davranışında potansiyel değişiklik gösterir (testte verilmemiştir).

- **frankfurt_shutdown (Frankfurt depo kapanması)**: Operasyonel nedenlerle Frankfurt deposunun kapanması veya sınırlaması (testte verilmemiştir).
- **precipitation (yağış)**: Depo lokasyonu çevresindeki yağış miktarı (mm), müşteri lokasyonuyla korelasyon gösterir (testte verilmemiştir).
- **snow (kar yağışı)**: Depo lokasyonu çevresindeki kar yağışı miktarı (mm), müşteri lokasyonuyla korelasyon gösterir (testte verilmemiştir).
- **user_activity_1 (kullanıcı aktivitesi 1)**: Web sitesindeki kullanıcı aktivitesi (testte verilmemiştir).
- **user_activity_2 (kullanıcı aktivitesi 2)**: Web sitesindeki kullanıcı aktivitesi (testte verilmemiştir).
- **id (kimlik)**: Depo adı ve tarihten oluşan satır kimliği.

# Bazı Özelliklerin Test Veri Setinde Olmamasının Nedenleri

| Eğitim Veri Seti (Train) | Test Veri Seti (Test) |
|---|---|
| warehouse | warehouse |
| date | date |
| orders | holiday_name |
| holiday_name | holiday |
| holiday | shops_closed |
| shutdown | winter_school_holidays |
| mini_shutdown | school_holidays |
| shops_closed | id |
| winter_school_holidays | |
| school_holidays | |
| blackout | |
| mov_change | |
| frankfurt_shutdown | |
| precipitation | |
| snow | |
| user_activity_1 | |
| user_activity_2 | |
| id | |

Table 1: Eğitim ve test veri seti özellikleri

### Test Verisinde Kullanılmayan Özellikler

Test veri setinde olmayan özellikler genellikle eğitim sırasında modelin performansını artırmak için dahil edilir. Eğitim verisi, modelin çeşitli koşullara göre eğitilmesini sağlar ve daha iyi genelleme yeteneği kazandırır. Ancak, bu özellikler test sırasında bulunmayabilir çünkü her zaman ölçülebilir veya toplanabilir durumda olmayabilirler.

### Gerçek Zamanlı Olarak Ölçülmesi Zor Olan Özellikler

- **Operasyonel Veriler**: *shutdown*, *mini_shutdown*, *blackout*, *frankfurt_shutdown* gibi özellikler operasyonel olayları temsil eder ve bu olaylar anlık olarak ölçülemeyebilir veya kaydedilemeyebilir. Bu tür veriler, eğitim aşamasında geçmiş olaylara dayanarak modele katkı sağlar, ancak test aşamasında her zaman mevcut olmayabilir.
- **Kullanıcı Aktivitesi Verileri**: *user_activity_1* ve *user_activity_2* gibi kullanıcı aktiviteleri, gerçek zamanlı veri toplama zorlukları ve gizlilik endişeleri nedeniyle test verisinde yer almayabilir. Bu tür veriler anlık değişiklik gösterebilir ve sürekli izlenmesi zor olabilir.

## Bu Durumda Ne Yapılabilir?

### Eksik Özellikler İçin Alternatifler Kullanma

Test verisinde bulunmayan ancak eğitim verisinde kullanılan özellikler için alternatifler oluşturulabilir. Örneğin, *shutdown* veya *blackout* gibi özellikler için benzer tarihsel olaylara veya operasyonel verilere dayalı tahminler yapılabilir. Bu sayede model, bu özellikler eksik olduğunda bile tahmin yapabilir.

### Tahmin ve İmputation Teknikleri

Eksik özellikler için tahmin (imputation) teknikleri kullanılabilir. Örneğin, *precipitation* ve *snow* verileri için geçmiş hava durumu verilerine dayanarak tahminler yapılabilir veya bu veriler ortalama değerlerle doldurulabilir. Bu, modelin eksik verilerle başa çıkmasına yardımcı olabilir.

### Özellik Mühendisliği (Feature Engineering)

Test verisinde bulunmayan özelliklerin yerine geçebilecek yeni özellikler oluşturulabilir. Örneğin, *user_activity_1* ve *user_activity_2* gibi kullanıcı aktiviteleri yerine, kullanıcıların site ziyaretleri, tıklama verileri veya arama trendleri gibi diğer ölçümler kullanılabilir. Bu, modelin kullanıcı davranışlarını daha iyi anlamasını sağlayabilir.

Bu yaklaşımlar, modelin eksik verilerle başa çıkabilmesini sağlayarak daha doğru tahminler yapmasına yardımcı olabilir. Eğitim aşamasında kullanılan geniş veri seti ve özellikler, modelin genel performansını artırır ve test aşamasında eksik verilerle karşılaşıldığında modelin daha esnek ve dayanıklı olmasını sağlar.

## Literatür Taraması

Satış tahmini, iş planlaması ve karar verme süreçleri için son derece önemlidir. Gelecekteki satışları doğru bir şekilde tahmin ederek, işletmeler envanter yönetimini, kaynak tahsisini ve pazarlama stratejilerini optimize edebilirler. Python, çok yönlü bir programlama dilidir ve geniş kütüphaneleri, veri manipülasyon yetenekleri ve makine öğrenimi algoritmaları sayesinde satış tahmini için güçlü bir araç haline gelmiştir.

Bu literatür taraması, veri bilimi projeleri içindeki Python tabanlı satış tahmini yaklaşımlarının mevcut peyzajına derinlemesine inmektedir.

## Geleneksel İstatistiksel Yöntemler

Geleneksel istatistiksel yöntemler uzun süredir satış tahmini için kullanılmaktadır. Otoregresif Entegre Hareketli Ortalama (ARIMA) ve Mevsimsel ARIMA (SARIMA) modelleri, zaman serisi desenlerini ve mevsimsellikleri yakalamak için geniş bir şekilde kullanılmaktadır. Bu yöntemler, uygulamaları ve yorumlaması kolay olduğundan başlangıç tahmin görevleri için uygundur. Ancak, karmaşık veri ilişkilerini ve doğrusal olmayan trendleri ele alamama gibi sınırlamaları, tahmin doğruluğunu olumsuz etkileyebilir.

## Makine Öğrenimi Yaklaşımları

Makine öğrenimi algoritmaları, karmaşık veri desenlerini ve ilişkilerini modellemek için sofistike teknikler sunarak satış tahminini devrimleştirmiştir. Doğrusal Regresyon, Destek Vektör Makineleri (SVM) ve Rastgele Orman gibi denetimli öğrenme algoritmaları, tarihî veri ve ilgili özelliklere dayalı olarak satış tahmini yapmada başarılıdır. Bu algoritmalar, geleneksel istatistiksel yöntemlerin gözden kaçırabileceği gizli desenleri ve doğrusal olmayan ilişkileri belirleyebilir.

## Derin Öğrenme ve Satış Tahmini

Makine öğreniminin bir alt kümesi olan derin öğrenme, büyük veri kümelerini işlemekte ve karmaşık desenleri çıkarmada yeteneğiyle satış tahmininde önemli bir yer edinmiştir. Özellikle Uzun Kısa Süreli Hafıza (LSTM) ağları gibi Tekrarlayan Sinir Ağları (RNN'ler), zaman serisi tahmin görevleri için uygun bir biçimde tasarlanmıştır. LSTM ağları, uzun vadeli bağımlılıkları yakalayabilir ve zamansal dinamikleri etkili bir şekilde ele alabilir, bu da uzun dönem satış eğilimlerini tahmin etmek için idealdir.

## Python Kütüphaneleri ile Satış Tahmini

Python, satış tahmini görevlerini kolaylaştıran zengin bir kütüphane ekosistemi sunar. Pandas, NumPy ve Matplotlib gibi kütüphaneler temel veri manipülasyonu ve görselleştirme araçları sağlar. Scikit-learn ve TensorFlow ise sırasıyla kapsamlı makine öğrenimi ve derin öğrenme çerçeveleri sunarak çeşitli tahmin modellerinin uygulanmasını sağlar.

## Örnekler ve Uygulamalar: Teoriden Uygulamaya

Çeşitli araştırma çalışmaları ve pratik uygulamalar, Python tabanlı satış tahmini yaklaşımlarının etkinliğini göstermektedir:

- **Perakende Satış Tahmini:** "Python ile Satış Tahmini: Zaman Serisi Analizi ve Tahmin" başlıklı bir çalışma, perakende mağaza zinciri için ARIMA ve SARIMA modellerini kullanarak satış tahmini yapmıştır. Modeller, mevsimsel varyasyonları ve eğilimleri yakalama konusunda önemli iyileştirmeler sağlamıştır [1].
- **E-ticaret Satış Tahmini:** "Makine Öğrenimi Modelleri Kullanarak Satış Tahmini" başlıklı başka bir araştırma projesi, e-ticaret platformu için Rastgele Orman ve LSTM ağlarını kullanarak satış tahmini yapmıştır. Çalışma, bu makine öğrenimi modellerinin geleneksel istatistiksel yöntemlere kıyasla daha yüksek doğruluk ve genelleme yeteneği sunduğunu bulmuştur [2].
- **Reklam Etkisi Analizi:** "Etkili Reklam Medya Satış Verilerine Dayalı Satış Tahmini: Python Uygulama Yaklaşımı" başlıklı bir araştırma makalesi, çeşitli Python tabanlı makine öğrenimi algoritmalarını kullanarak reklam medyasının satışlar üzerindeki etkisini analiz etmiştir. Çalışma, Lineer Regresyon ve SVM gibi modellerin reklam çabalarını satış artışlarıyla etkili bir şekilde ilişkilendirebildiğini göstermiştir [3].
- **Tedarik Zinciri Optimizasyonu:** NAAC akreditasyon raporunda sunulan bir vaka çalışması, Python tabanlı satış tahmin modellerinin tedarik zinciri operasyonlarını optimize etmek için nasıl kullanıldığını göstermiştir. Daha doğru satış tahminleri yaparak işletme envanter seviyelerini daha verimli yönetmeyi, stok eksikliklerini azaltmayı ve fazla envanter maliyetlerini minimize etmeyi mümkün kılmıştır [4].

## Sonuç

Python, veri bilimi projelerinde güçlü bir satış tahmini aracı olarak öne çıkmaktadır. Geleneksel istatistiksel yöntemler, makine öğrenimi algoritmaları ve derin öğrenme teknikleri; hepsi Python kütüphaneleri kullanılarak satış tahmini için çeşitli yaklaşımlar sunar. Hangi yöntemin seçileceği, belirli veri özelliklerine, tahmin süresine ve istenen karmaşıklık düzeyine bağlıdır.

Veri hacimleri ve hesaplama gücü büyüdükçe, Python tabanlı satış tahmini yöntemlerinin iş kararları almak için çok daha önemli bir rol oynaması beklenmektedir. Gelecekteki araştırmalar, aşağıdaki konuları keşfedebilir:

- **Karma Modelleme:** Geleneksel istatistiksel yöntemlerin ve modern makine öğrenimi algoritmalarının güçlerini birleştiren hybrid modeller.
- **Harici Veri Entegrasyonu:** Sosyal medya trendleri ve ekonomik göstergeler gibi harici veri kaynaklarının tahmin doğruluğunu artırmak için nasıl entegre edilebileceği.
- **Kullanıcı Dostu Araçlar:** Çeşitli endüstrilerde bu ileri düzey tahmin tekniklerinin daha geniş bir şekilde benimsenmesini kolaylaştırmak için daha kullanıcı dostu Python kütüphaneleri ve araçların geliştirilmesi.

Bu yaklaşımlar sürekli olarak geliştirilerek, işletmeler Python tabanlı satış tahmini gücünden yararlanarak rekabet avantajı elde edebilir ve en iyi performans için veri odaklı kararlar alabilirler.

## Kullanılan Kütüphaneler ve Fonksiyonlar

Koddaki temel kütüphaneler ve işlevleri şu şekildedir:

1. **Scikit-learn (sklearn):**
   - `mean_absolute_percentage_error`: Tahmin hatalarını değerlendirmek için kullanılır.
   - `TimeSeriesSplit`: Zaman serisi verilerini katlamalı olarak ayırır.
   - `clone`: Bir modelin kopyasını oluşturur.
2. **Optuna:**
   - `TPESampler`: Hiperparametre optimizasyonunda kullanılan bir örnekleme yöntemidir.
   - `create_study ve optimize`: Optuna ile hiperparametre optimizasyonu için bir çalışma oluşturur ve optimize eder.
3. **XGBoost:**
   - `XGBRegressor`: XGBoost kütüphanesinin regresyon modeli.

4. **CatBoost:**
    - `CatBoostRegressor:` CatBoost kütüphanesinin regresyon modeli.
5. **LightGBM:**
    - `LGBMRegressor:` LightGBM kütüphanesinin regresyon modeli.
6. **Matplotlib:**
    - `plt:` Grafiklerin oluşturulması ve gösterilmesi için kullanılır.
7. **Seaborn:**
    - `sns:` İstatistiksel veri görselleştirmesi için kullanılır.
8. **Pandas:**
    - `pd:` Veri manipülasyonu ve analiz için kullanılır.
9. **NumPy:**
    - `np:` Sayısal hesaplamalar ve dizi işlemleri için kullanılır.
10. **Warnings:**
    - `warnings.filterwarnings('ignore'):` Olası uyarı mesajlarını engeller.

# 1 Makine Öğrenmesi Algoritmaları ve Dataset Uygulanabilirliği

Bu bölümde, zaman serisi tahminlemesi ve geleneksel makine öğrenmesi modelleri olmak üzere datasetimizde kullanılabilecek ana algoritmaları tartışacağız.

## 1.1 Zaman Serisi Tahminleme

### 1.1.1 ARIMA (AutoRegressive Integrated Moving Average)

ARIMA, zaman serisi verilerinde trend ve mevsimsel değişimleri modellemek için kullanılan klasik bir yöntemdir. ARIMA modeli, verinin durağan (stationary) hale getirilmesi, otoregresif (AR) ve hareketli ortalama (MA) terimlerinin belirlenmesiyle oluşturulur. Bu model, özellikle trend ve mevsimsel değişimlerin belirgin olduğu veri setlerinde etkilidir. Datasetimizdeki tarih ve mevsimsel özellikler (örneğin tatiller ve okul tatilleri), ARIMA modeli için uygun bir temel oluşturabilir.

### 1.1.2 SARIMA (Seasonal ARIMA)

SARIMA, mevsimsel bileşenlerin de dahil olduğu ARIMA modelinin genişletilmiş bir versiyonudur. Mevsimsel değişimlerin yanı sıra ARIMA'nın temel bileşenlerini de içerir. SARIMA modeli, zaman serisi verilerinde mevsimsel desenlerin güçlü bir şekilde modellemesi gerektiği durumlarda tercih edilir. Datasetimizde mevsimsel etkilerin belirgin olduğu özellikler varsa SARIMA modeli kullanılabilir.

### 1.1.3 Prophet

Prophet, Facebook tarafından geliştirilmiş açık kaynaklı bir zaman serisi tahminleme aracıdır. Güçlü bir sezgisel model olup, tatil etkileri ve trend değişimleri gibi dışsal faktörleri otomatik olarak ele alabilir. Datasetimizdeki tatil adı ve tatil özellikleri, Prophet'in veri setimizde doğru modellemeler yapmasına yardımcı olabilir.

### 1.1.4 LSTM (Long Short-Term Memory)

LSTM, özellikle uzun süreli bağımlılıkların ve zaman serilerindeki karmaşık yapıların modellemesi için kullanılan bir derin öğrenme modelidir. LSTM, önceki zaman adımlarından gelen bilgileri hatırlayabilen ve bunları gelecekteki tahminlerde kullanabilen özel bir yapay sinir ağı türüdür. Datasetimizde zaman serisi verilerinin uzun dönem bağımlılıkları varsa, LSTM modeli bu yapıları etkili bir şekilde modelleyebilir.

### 1.1.5 Transformer-based Models

Transformers, ¨ozellikle dil i¸sleme ve zaman serisi tahmininde etkili olan modern derin ¨o˘grenme modelleridir. Transformers, sekans verilerindeki uzun s¨ureli ba˘gımlılıkları yakalamada ¸cok ba¸sarılıdır. Avantajları: • Zaman serisi verilerindeki uzun s¨ureli ba˘gımlılıkları yakalayabilir. • Karma¸sık yapıları modelleyebilir. • Hem e˘gitimi hem de tahmini paralel ¸sekilde ger¸cekle¸stirebilir

### 1.1.6 Linear Regression

Doğrusal regresyon, bağımlı değişken ile bağımsız değişken(ler) arasındaki ilişkiyi modelleyen basit bir modeldir. Datasetimizdeki tatil, dükkan kapanışı ve okul tatili gibi doğrusal ilişkileri modellemek için uygun olabilir.

### 1.1.7 Random Forest Regressor

Random Forest, birden fazla karar ağacının oluşturduğu bir topluluk yöntemidir. Her bir ağaç bağımsız olarak tahmin yapar ve sonuçların ortalaması alınarak nihai tahmin elde edilir. Model, özellikle karışık etkileşimlerin ve doğrusal olmayan ilişkilerin olduğu veri setlerinde etkilidir. Datasetimizdeki tatil, dükkan kapanışı ve okul tatili gibi kategorik özellikler, Random Forest Regressor modeli için anlamlı değişkenler olabilir.

### 1.1.8 Gradient Boosting Machines (GBM)

GBM, ardışık karar ağaçları oluşturarak her adımda hatayı azaltmaya çalışan bir yöntemdir. GBM, büyük veri setlerinde yüksek doğruluk sağlar ve genellikle karmaşık yapıları modelleme konusunda başarılıdır. Datasetimizdeki çok sayıda özellik ve bunların karmaşıklığı, GBM'in doğruluk ve performansını artırabilir.

## 1.2 Uygulanabilirlik Açıklaması

Veri setimiz, zaman serisi bileşenleri ve çok sayıda özellik içermesi nedeniyle hem zaman serisi tahminleme yöntemleri hem de geleneksel makine öğrenmesi modelleri için uygun bir zemin sunmaktadır.LSTM ve Transformer tabanlı modeller, zaman serisi verilerindeki uzun donemli bagımlılıkları yakalamada basarılı olabilir. Zaman serisi tahminleme yöntemleri, tarihsel trendleri ve mevsimsel etkileri modellemek için kullanılabilirken, geleneksel makine öğrenmesi modelleri, çoklu değişkenler arasındaki karmaşık ilişkileri çözebilir. Bu çeşitlilik, veri setimizdeki farklı yapıları ve ilişkileri anlamak ve tahmin etmek için farklı model seçenekleri sunar.

# 2 Zaman Serisi Özelliği ve Önemi

Veri setinde zaman serisi özelliği bulunması, zaman içindeki trendleri, mevsimsel değişimleri ve diğer zaman bağımlı etkileri modellemek için kritik bir rol oynar. Örneğin, günlük satış verilerinde haftalık, aylık veya yıllık periyodik desenler olabilir. Bu desenleri tanımlamak ve modellemek, zaman serisi tahminleme modelleri için gereklidir. ARIMA, SARIMA gibi modeller bu tür yapıları anlamak ve tahmin etmek için uygun yöntemlerdir. Ayrıca, Prophet gibi zaman serisi spesifik araçlar da tatil etkileri gibi dışsal faktörleri otomatik olarak ele alabilir.

# 3 Diğer Feature Özellikleri ve Önemi

Zaman serisi tahminleme modelleri, sadece tarih bazlı verileri değil, aynı zamanda diğer özellikleri de dikkate alabilir. Örneğin, tatil günleri (`holiday`), dükkan kapanışları (`shops_closed`), okul tatilleri (`school_holidays`) gibi kategorik değişkenler, satışları doğrudan etkileyebilir. Bu değişkenler, modelin doğruluğunu artırmak için önemli ölçüde katkıda bulunabilir. Dummy değişkenler oluşturarak (örneğin, tatil adı gibi) bu kategorik verileri modellemek de önemlidir.

# 4 Uygun İndikatörler

370 gün gibi uzun bir süre boyunca tahmin yaparken, mevcut veriler üzerindeki trendleri anlamak ve gelecekteki değişiklikleri tahmin etmek için uygun indikatörler kullanmak önemlidir. Örneğin, mevsimsel değişkenler (`winter_school_holidays`

gibi), önceki dönemlerdeki satışlar (*lags*), hava durumu (`precipitation`, `snow` gibi), kullanıcı aktiviteleri (`user_activity_1`, `user_activity_2` gibi) gibi faktörler tahminin doğruluğunu artırabilir. Bu indikatörler, modelin geçmiş performansı ve gelecekteki eğilimler hakkında daha fazla bilgi edinmesine yardımcı olabilir.

# 5 Model Seçimi ve Uygulama

Veri setindeki zaman serisi bileşenleri ve diğer değişkenlerin önemi göz önüne alındığında, uygun model seçimi oldukça kritiktir. ARIMA, SARIMA gibi zaman serisi tahminleme modelleri, zaman içindeki desenleri ve mevsimsel etkileri modellemek için idealdir. Prophet gibi otomatik zaman serisi tahmin araçları, tatil etkileri gibi dışsal faktörleri de göz önünde bulundurabilir. Ayrıca, Random Forest, Gradient Boosting Machines gibi geleneksel makine öğrenmesi modelleri de zaman serisi dışı etkileri ve karmaşık ilişkileri modellemek için uygun seçeneklerdir.

# 6 Sonuç

370 gün boyunca tahmin yaparken, veri setimizdeki zaman serisi bileşenlerinin yanı sıra diğer değişkenlerin de önemli olduğunu anlıyoruz. Bu nedenle, zaman serisi tahminleme modelleri ile geleneksel makine öğrenmesi modellerini dikkatli bir şekilde seçmeli ve uygulamalıyız. İndikatörlerin doğru kullanımı ve uygun model seçimi, tahminin doğruluğunu artırabilir ve karar verme süreçlerinde değerli bilgiler sağlayabilir.

# Referanslar


1. "Python ile Satış Tahmini: Zaman Serisi Analizi ve Tahmin" (Medium)
   `https://medium.com/@melodyyip_/time-series-analysis-and-forecasting-with-python-67488cf`
2. "Makine Öğrenimi Modelleri Kullanarak Satış Tahmini" (RCB'nin Annaleri)
   `https://www.researchgate.net/publication/309885039_Explaining_machine_learning_models_in`


# Ek Referanslar


- "NAAC Akreditasyon Raporunda Satış Tahmini Analizi" (NAAC)
  `https://kct.ac.in/naac-accreditation-2022/`
- "Etkili Reklam Medya Satış Verilerine Dayalı Satış Tahmini: Python Uygulama Yaklaşımı" (Allied Business Academies)
  `https://www.abacademies.org/articles/prediction-of-sales-based-on-an-effective-advertis`