T.C.
BEYKENT ÜNİVERSİTESİ
LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ
BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI
BİLGİSAYAR MÜHENDİSLİĞİ BİLİM DALI

# MAKİNE ÖĞRENMESİ METODLARI İLE MÜŞTERİ PROFİLLEME VE YAPILAN PROFİLLEMENİN ETKİNLİĞİNİ DEĞERLENDİRME

Yüksek Lisans Tezi

Tezi Hazırlayan:

**Fikri Murat SİMAV**

İstanbul, 2020

T.C.
BEYKENT ÜNİVERSİTESİ
LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ
BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI
BİLGİSAYAR MÜHENDİSLİĞİ BİLİM DALI

# MAKİNE ÖĞRENMESİ METODLARI İLE MÜŞTERİ PROFİLLEME VE YAPILAN PROFİLLEMENİN ETKİNLİĞİNİ DEĞERLENDİRME

Yüksek Lisans Tezi

Tezi Hazırlayan:

**Fikri Murat SİMAV**

Öğrenci No:

17080208005

Danışman:

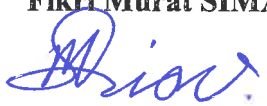
Dr. Öğr. Üyesi Atınç YILMAZ

İstanbul, 2020

## YEMİN METNİ

Yüksek Lisans Tezi olarak sunduğum "Makine öğrenmesi metotları ile müşteri profilleme ve yapılan profillemenin etkinliğini değerlendirme" başlıklı bu çalışmanın, bilimsel ahlak ve geleneklere uygun şekilde tarafımdan yazıldığını, yararlandığım eserlerin tamamının kaynaklarda gösterildiğini ve çalışmamın içinde kullanıldıkları her yerde bunlara atıf yapıldığını belirtir ve bunu onurumla doğrularım. 03/02/2020

**Fikri Murat SİMAV**



**T.C.**
**BEYKENT ÜNİVERSİTESİ**
**LİSANÜSTÜ EĞİTİM ENSTİTÜSÜ MÜDÜRLÜĞÜ**
**TEZLİ YÜKSEK LİSANS SINAV TUTANAĞI**

03.02.2020

Enstitümüz ***Bilgisayar Mühendisliği*** Anabilim Dalı ***Bilgisayar Mühendisliği*** Programı yüksek lisans öğrencilerinden ***17080208005*** numaralı ***Fikri Murat SİMAV***'ın *"Beykent Üniversitesi Lisansüstü Eğitim – Öğretim Yönetmeliği"*nin ilgili maddesine göre hazırlayarak, Enstitümüze teslim ettiği ***"Makina Öğrenmesi Metodları İle Müşteri Profilleme ve Yapılan Profillemenin Etkinliğini Değerlendirme"*** konulu tezini, Yönetim Kurulumuzun 28/01/2020 tarih ve 2020/04 sayılı toplantısında seçilen ve Taksim Yerleşkesinde toplanan biz jüri üyeleri huzurunda, Beykent Üniversitesi Lisansüstü Eğitim ve Öğretim Yönetmeliğinin 29. maddesinin 3. fıkrası gereğince (47) dakika süre ile aday tarafından savunulmuş ve sonuçta adayın tezi hakkında ~~***oyçokluğu/oybirliği***~~ ile ***Kabul/~~Red veya Düzeltme~~*** kararı verilmiştir.

İşbu tutanak, 4 nüsha olarak hazırlanmış ve Enstitü Müdürlüğü'ne sunulmak üzere tarafımızdan düzenlenmiştir.

**DANIŞMAN**
Dr. Öğr. Üyesi Atınç YILMAZ
**(Beykent Üniversitesi)**

**ÜYE**
Dr. Öğr. Üyesi Ediz ŞAYKOL
**(Beykent Üniversitesi)**

**ÜYE**
Dr. Öğr. Üyesi Ömer ÇETİN
**(Milli Savunma Üniversitesi)**

Adı ve Soyadı : Fikri Murat SİMAV
Danışmanı : Dr. Öğr.Üyesi Atınç YILMAZ
Türü ve Tarihi : Yüksek Lisans, 2020
Alanı : Bilgisayar Mühendisliği
Anahtar Kelimeler : Müşteri Profili Belirleme, Makine Öğrenmesi, Veri Madenciliği, Veri, Analitiği, rassal ormanlar, tahminleme

# ÖZ

## MAKİNE ÖĞRENMESİ METODLARI İLE MÜŞTERİ PROFİLLEME VE YAPILAN PROFİLLEMENİN ETKİNLİĞİNİ DEĞERLENDİRME

Veri Madenciliği ve makine öğrenmesi, insan gözüyle ya da manuel hesaplamasıyla ulaşılamayacak çıkarımları toplu ve otomatize olarak yapıp kurumların ellerindeki veriden maksimum fayda sağlamalarına yarayacak güçlü yöntemler içermektedir. Yapılan analizlerin ve çıkarımların başarı oranlarını değerlendirmek ve kullanılan yöntem parametre ve hiper parametrelerin iteratif olarak iyileştirilmesi ve gözlemlenmesi başarılı bir makine öğrenmesi projesi için olmazsa olmazlardandır.

Bu çalışmada bir turizm firmasının son altı yıla ait müşteri ve rezervasyon verileri, kişisel verilerin korunması kanununa uygun şekilde anonimleştirilmiş ve karar ağaçları, rassal ormanlar, destekçi vektör makineleri ve yapay sinir ağları yöntemleri kullanılarak segmentlere ayrılarak, geçmiş satın alma alışkanlıklarından takil köyü pazarlama faaliyetlerinin verimliliği arttırılması hedeflenmiştir.

Çalışmada yukarıda belirtilen makine öğrenmesi algoritmaları, farklı parametre ve hiper parametrelerle kullanılarak, başarı oranları karşılaştırılmıştır. Algoritmalar uygulanıp sonuçları denenirken test ve eğitim veri kümeleri çapraz doğrulama yöntemi uygulanarak seçilmiştir.

Eğitim için algoritmalara verilen verinin özniteliklerinin seçimi ve mevcut özniteliklerden yeni öznitelikler oluşturmanın, algoritmaların başarısında büyük önem taşıdığı gözlemlenmiştir.

Name and Surname : Fikri Murat SİMAV
Supervisor : Assist. Prof. Dr. Atınç YILMAZ
Degree and Date : Master, 2020
Major : Computer Engineering
Keywords : Machine learning, Customer Profiling, Data mining, Random forest, prediction

## ABSTRACT

## CUSTOMER PROFILING WITH VARIOUS MACHINE LEARNING ALGORITHMS AND VALIDATING THE SUCCESS RATE OF THIS PROFILING

With the help of data mining and machine learning, we get see the patterns and have insights from data, which were otherwise not possible to get by manuel caculations. In this study, a tourism company's anonymized customer and reservation data covering the last six years of transactions were examined and analysed, with the help of algorithms such as decision trees, random forests, support vector machines and artificial neural networks. Aforementioned algorithms were fine tuned improve the success rate to predict high potential customers who will buy a resort type accomodation.

Data and features fed to the algorithms were observed to significantly change the success rates of these machine learning algorithms.

# İÇİNDEKİLER

# TABLOLAR LİSTESİ

**Sayfa No.**

# ŞEKİLLER LİSTESİ

**Sayfa No.**

# GİRİŞ

Günümüzde veri saklama ve işleme maliyetleri düşmeye ve internet kullanım penetrasyonu artmaya devam etmektedir. Dünya genelindeki mevcut tüm verilerin toplam hacminin %90'ı son 2 yılda üretilmiş verilerden oluşmaktadır.

Üretilen ve saklanan verinin boyutu ve çeşitliliği arttıkça, veriyi işlemenin, yorumlamanın ve bunu optimize ve otomatize şekilde yapıp hızlı iş çıktıları elde etmenin önemi artmaktadır.

Bu çalışmada Türkiye'de turizm sektöründe lider konumda olan bir firmanın 2003 yılından itibaren satış ve rezervasyon verisi anonimleştirilip, çeşitli makine öğrenmesi yöntemleri ile temizlenerek ve işlenerek müşteriler gruplara ayrılacak, ve müşterilerin Tatil Köyü satın alma ihtimali yüksek olanlarına bu yönde doğrudan satış teknikleri ile pazarlama faaliyetlerinin yürütülmesi hedeflenecektir.

Hem kişisel verileri koruma kanunu gereği, hem de tatilcilerin sektörden beklentisi sebebiyle, turizm sektöründe satın alma aşamasında seyahat eden kişilerle ilgili çok detaylı bilgiler tutulmamaktadır. Gerekli detaylar oteller ile müşteriler arasında yüz yüze kayıt altına alınmaktadır.

Bu sebeple, bu çalışmada eldeki veri imkan verdiği sürece kişilerin nitelikleri, ve bunu desteklemek için kişilerin bugüne kadar satın aldıkları tatil hizmetlerinde bahsi geçen otellerin nitelikleri, ve bu otelleri tercih eden diğer müşterilerin ortak yanları bulunmaya çalışılıp sonuca ulaşılmaya çalışılacaktır.

Çalışma çıktısında, kurumlara satış ve pazarlama faaliyetlerinde hangi müşterilere odaklanmalarının daha çok fayda sağlayacağı ve hangi müşteriye hangi tatili önermenin müşteri talep ve beklentileriyle uyumlu olacağı ve dolayısıyla satışla sonuçlanmasının daha olası olacağı tahminlenmeye çalışışaktır. Kayıp müşterilerin (churn) ortak yanları incelenip, kaybedilme ihtimali yüksek olan müşteriler tahminlenecektir.

Türkiye'de turizm sektöründe lider konumda olan bir firmanın geçmiş satış ve rezervasyon verileri analiz edilerek, tatil köyü satış ve pazarlama faaliyetlerinde hedef kitle olarak seçilecek kişilerin tespitine odaklanılacaktır.

Firmanın müşteri veritabanındaki, tatil köyü rezervasyonu yapma ihtimali yüksek müşterilerin tespiti ve bu potansiyel müşterilerden, yüksek fiyatlı hizmet satın alması öngörülen kişiler tespit edilerek, kurumun pazarlama kaynaklarını verimli kullanarak satış ve karlılığın artırılmasına katkı sağlamak hedeflenecektir.

# BİRİNCİ BÖLÜM

## 1. VERİ, YÖNTEM VE TEKNOLOJİLER

Bu çalışmada, Türkiye'de lider konumda olan bir turizm firmasının 2003 yılından itibaren satış ve rezervasyon verisi anonimleştirilip aşağıda ilgili başlıklarda deteylandırılan yöntem ve teknolojilerle işlenmiştir.

Kullanılan veri, farklı veri tabanı yönetim sistemlerinden (Oracle, Postresql ve MS SQL) derlenerek bir Hadoop cluster'ında barındırılmakta ve belli aralıklarla çalışan programlar veriyi Hadoop cluster'ına, anonimize ederek kopyalamaktadır. Hadoop'un dağıtık dosyalama sistemi HDFS sayesinde, bir sunucuya sığamayacak, ya da bir sunucuda işleneyemeyecek kadar büyük veriler kolaylıkla işlenebilmektedir. Bu hadoop veritabanını sorgulamak, farklı tablolardaki veriyi birleştirmek ve dönüştürmek için Datameer isimli web tabanlı bir yazılım kullanılmıştır.

Datameer bu süreçte, bu bilgiyi lokal makinelere alıp işlemekten çok daha hızlı ve verimli şekilde yapılmasına olanak sağlamıştır. Uzun süren işlemleri, programatik olarak sıraya alıp, paralelde başka sorgular yapılabilmesine olanak sağlamıştır. Çalışma için ihtiyaç duyulan veri üç ayrı tabloda derlenip, metin tabanlı, virgüllerle ayrılmış CSV formatında yerel makinelere alınmıştır.

### 1.1. Veri Kaynağı

Bu çalışmaya konu olan veri, Türkiye'de faaliyet gösteren bir turizm firmasının Hadoop veri ambarında derlenerek temin edilmiştir. Firmanın çalışma şekli ve müşterilerine zorunlu tuttuğu alanlar kısıtlı olduğu için müşteriler hakkında çok detaylı nitelikler bulunmadığı görülmüştür. Bazı durumlarda ise, veri dağılımları incelendiğinde müşterilerin, veri girişini kolaylaştıracak şekilde veri girişi yaptıkları yine bu çalışma ile tespit edilmiştir. Bazı kategorik bilgilerde ise, sistemlerin mükerrerlik içerebilecek şekilde kodladığı tespit edilip bu durumu kompanse edecek şekilde tekilleştirme çalışması yapılmıştır.

### 1.2. Veri sözlüğü

Derlenen ve analiz edilmek üzere alınan verilerle ilgili oluşturulan veri üç ana tabloda toparlanmıştır.

- Rezervasyonlar
- Kişiler
- Katılımcılar

### 1.1.1. Kişiler

Bir kişi ilk defa sisteme geldiği ve kayıt olduğu ilk aşamada kendisi için bir tekil anahtar (unique identifier) oluşturulup kişiler tablosunda kalıcı olarak saklanıyor. Bu tablodaki özniteliklerde boş kayıt adetleri ve bunların tüm kayıtlara oranı Tablo 1'de verilmiştir.

**TABLO 1: Kişiler Tablosundaki Boş Kayıt Oranları**

| Alan Adı | Boş Kayıt Adedi | Boş Kayıt Yüzdesi |
|---|---|---|
| **country** | **13.406.200** | **100** |
| **city** | **13.406.200** | **100** |
| **email_ipo** | **13.406.200** | **100** |
| **sms_ipo** | **13.406.200** | **100** |
| **phone** | **13.406.200** | **100** |
| **fax_ipo** | **13.406.200** | **100** |
| **passport_type** | **13.406.200** | **100** |
| **birth_place** | **13.380.998** | **99,81** |
| **passport_valid_date** | **13.361.957** | **99,66** |
| **marital_status** | **13.042.507** | **97,28** |
| birth_date | 5.347.755 | 39,89 |
| gender_code | 3.000.892 | 22,38 |
| contact_id | 0 | 0 |
| create_date | 0 | 0 |
| update_date | 0 | 0 |
| state_code | 0 | 0 |
| customer_type | 0 | 0 |

Koyu ile işaretlenmiş alanlarda boş oranı çok büyük olduğundan analizler sırasında veri setinden çıkartılmıştır.

### 1.1.2. Katılımcılar

Katılımcılar tablosu, Kişiler tablosu ve Rezervasyonlar tablosun arasında çoka çok (many to many) ilişki kurulmak için kullanılmıştır. Bir kişinin ister satın almış olduğu, ister misafir olarak konaklama yaptığı her kayıt için bu tabloda reservation_id ve contact_id eşleşmesi ile saklanmaktadır.

Boş kayıt adet ve oranları en fazla 2013 yılı verilerinde bulunmaktadır. Ancak bu toplam %0,44'e ancak ulaştığı için bu satırlar veri setinden çıkarılmıştır. Çünkü bu 10.000 de 44 kayıtlık oran göz ardı edilebilir bir değer ve contact_id si bulunmayan satırların analiz için işimize yaramayacaktır.

### 1.1.3. Rezervasyonlar

Her bir konaklama aksiyonunun detayları Rezervasyonlar tablosunda tutulmaktadır. Rezervasyonlar tablosundaki özniteliklerin boş kayıt adetleri ve bunların tüm kayıtlara oranı Tablo 2'de verilmiştir.

**TABLO 2: Rezervasyonlar Tablosu Öznitelikleri**

| Alan Adı | Boş Kayıt Yüzdesi | Boş Kayıt Adedi |
|---|---|---|
| foreign_total_amount | 100 | 6.624.934 |
| invoice_tax_number | 87,03 | 5.765.713 |
| invoice_tax_office | 85,92 | 5.692.000 |
| invoice_company_name | 85,58 | 5.669.474 |
| invoice_address | 85,58 | 5.669.435 |
| comments | 51,84 | 3.434.411 |
| acc_discount_total | 37,01 | 2.451.674 |
| pansion_type | 36,84 | 2.440.512 |
| foreign_acc_discount_total | 36,84 | 2.440.561 |
| gof | 33,74 | 2.234.992 |
| operator | 18,98 | 1.257.393 |

| | | |
|---|---|---|
| destination_name | 15,96 | 1.057.323 |
| agent_personal_id | 6,96 | 527.167 |
| vendor_name | 5,1 | 337.702 |
| product_description | 4,62 | 305.851 |
| product_code | 4,61 | 305.549 |
| vendor_code | 4,61 | 305.224 |
| destination_code | 1,57 | 103.762 |
| unit_price | 1,32 | 87.351 |
| reservation_id | 0 | 0 |
| corporte_invoice | 0 | 0 |
| agent_id | 0 | 190 |
| matrix | 0 | 189 |
| total_amount | 0 | 189 |
| departure_date | 0 | 189 |
| dont_see_agency | 0 | 0 |
| sales_date | 0 | 189 |
| res_date | 0 | 189 |
| quote_date | 0 | 189 |
| arrival_date | 0 | 189 |
| child_count | 0 | 189 |
| adult_count | 0 | 189 |
| pax_count | 0 | 189 |
| status | 0 | 189 |
| state_code | 0 | 0 |
| update_date | 0 | 0 |
| created_date | 0 | 0 |
| res_number | 0 | 0 |
| acc_night_count | 0 | 189 |

### 1.3. Kullanılan Ürün ve Teknolojiler

Veri Datameer vasıtasıyla CSV formatın alındıktan sonra veri Python dili ile, ve aşağıdaki açık kaynaklı ve ücretsiz bir kütüphane olan Pandas kütüphanesinden faydalınılmıştır.

#### 1.3.1. Numpy

Numpy, numerik programlama için hazırlanmış bir Python kütüphanesidir. Çok boyutlu diziler ve matrisler üzerinde işlemler yapabilme, ve bu diziler üzerinde matemetiksel işlemleri hızlı ve etkince yapılabilmesine olanak sağlamaktadır.

Numpy'ın diziler üzerinde yapılan işlemleri vektörize sekilde yapabilmesi, modern işlemcilerde bulunan, bir komut çoklu veri (SIMD: Single Instruction Multiple Data) özelliğinden faydalanarak çok etkin ve hızlı şekilde yapılabilmesine olanak sağlayacak şekilde çalışmaktadır. Bunu yaparken, yazım kuralları (syntax) itibarıyla da, insanlar tarafından da hem yazılması hem okunması daha kolay bir yazım tarzı vardır.

Örnek vermek gerekirse, 10 milyon elemanlı bir dizimiz (array) olduğu durumda, bunu hem numpy'ın vektörize özelliklerini kullanarak ve kullanmayarak iki yazım tarzını, ve programın çalışma süresini aşağıdaki gibidir.

```
# Numpy kütüphesini np kısaltması (aliası) ile import et
import numpy as np

# Numpy ile, 0 ile 1 arasında 10 milyon adet rastgele sayı üret
x = np.random.rand(10000000)
For döngüsüyle yazım şekli
%%timeit
y = np.empty(x.shape)
for i in range(len(x)):
    y[i] = 2 * x[i] + 1
```

**Numpy ile vektörize şekilde yazım**

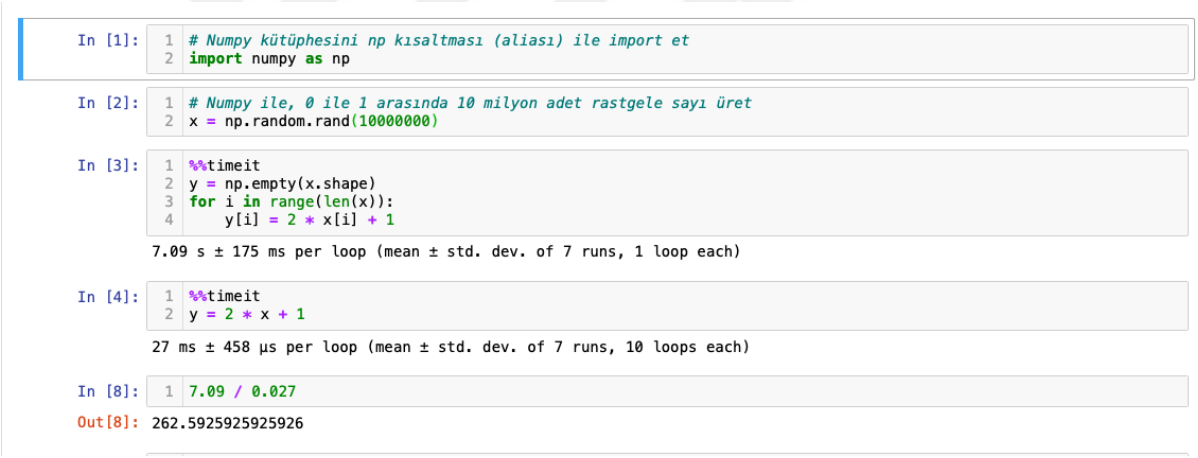
%%timeit

y = 2 * x + 1

Burada görüldüğü gibi numpy ile, y = 2 * x + 1 şeklindeki yazım daha okunabilir ve matematiksel notasyon ile tutarlı görünümdedir.

Çalışma süresi olarak ise, numpy ile yazılmış hali 4 çekirdekli bir Intel i7 işlemcili bilgisayarda for döngüsüyle yazılmış halinden ortalama olarak 262 kat hızlı çalıştığı gözlemlenmiştir. Bu operasyonların Jupyter'de çalıştılırılıp, süre kıyaslaması Şekil 1'de sunulmuştur.

**ŞEKİL 1: Numpy Vektörize İşlem Hızı ve Kodlama Kolaylığı**




Veri işleme sarıdasında sıklıkla yapılması gereken, bir tabloyu satır satır, ya da sütun sütun gezerek yapılması gereken işlemlerde, tüm programlama dillerinde olan "döngüler" kullanarak değil, Numpy'ın bize sağladığı, vektörel operasyonları kullanmak hem performans artışı, hem de yazım kolaylığı sağlamaktadır.

### 1.3.2. Pandas

Pandas ismini istatistikte ve R programlama dilindeki **Pan**el **Da**ta'dan esinlenerek almış, veri işleme ve ön işleme için yaygın olarak kullanılan bir

kütüphanedir. CSV'lerdeki veriler, Pandas'ın DataFrame isimli,tablosal verileri tutan veri yapısına aktarılmıştır.

Pandas, veri yapıları olarak numpy'ı temel aldığı için vektörize işlem yapmada numpy'ın sağladığı avantajdan yararlanır. Ek olarak sadece bir dizide değil, tablosal veriler üzerinde çalışma imkanları sunar.

Pandas projesini başlatan ve hala projenin başında olan Wes McKinney, Pandas'ı kurarken hedeflerinin temel olarak;

- Eksenleri isimlendirilebilen veri yapıları
- İçinde zaman serisi fonksiyonları barındıran
- Aritmetik işlemler ve veri indirgeme yapabilen
- Eksik veriyi esnek şekilde yönetebilen
- Veri birleştirme ve diğer ilişkisel operasyonları yerine getirebilen

Bir kütüphane geliştirmeyi hedeflediğini ve Pandas kütüphanesinde bunları hayata geçirdiklerini belirtmiştir (McKinney 2012).

Kolonlardaki verileri normalize etme, yeniden adlandırma, sayı, metin, ya da tarih formatına dönüştürme işlemleri kolaylıkla ve büyük bir verimlilikle yapabilmektedir.

### 1.3.3. Matplotlib ve Seaborn

Bu çalışmada Python ekosisteminde bulunan iki farklı raporlama ve grafik kütüphanesi kullanılmıştır.

Matplotlib, Pandas ve Seaborn da dahil, pek çok kütüphanenin de temellerinin dayandığı raporlama ve çizim kütüphanesidir.

Yazım tarzı ve genel mantık olarak Matlab'a benzerlik göstermektedir (VanderPlas 2019).

Veriyi Pandas ile raporlanabilecek formata getirdikten sonra, Matplotlib ile çeşitli grafikler hazırlamak çoğu zaman tek satırlık kodlama ile mümkün olmuştur.

Seaborn da, aslında temelleri Matplotlib'e dayansa da, raporlama kısmında Matplotlib'e göre şöyle bir artısı vardır; Seaborn, raporlamak istenen veriyi önceden formatlamanıza ve ya birleştirmenize gerek kalmadan, bu işlemleri kendisi yapabilmektedir.

Örneğin, her bir satışın bir satır olacağı şekilde bulunan veri, Aylık satışlar gruplanarak raporlanmak istendiğinde, satışlar önceden, pivot yapılma ya da gruplanmadan Seaborn'un ilgili metoduna verildiğinde, Seaborn hem bu gruplama ve hesaplama işlemini yapıyor, hem de bu hesaplamanın sonucunu grafik üzerinde gösteriyor.

### 1.3.4. Scikit Learn

Scikit Learn içerisinde pek çok gözetimli ve gözetimsiz öğrenme algoritmasını içeren ve sürekli gelişen bir kütüphanedir. Python ekosisteminin yaygın olarak kullanılan temel bir makine öğrenmesi kütüphanesidir. Aynı zamanda, API tutarlılığı ve üst seviye yaklaşımı sebebiyle algoritma geliştirme ve araştırmacı veri analizi (exploratory data analysis) için yaygın olarak kullanılmasına imkan sağlamıştır.

Açık kaynaklı ve ücretsiz olması, ve büyük bir katılımcı kitlesini arkasında bulundurması sebebiyle, yaygın olarak kullanılan veri analitiği ve veri doğrulama yöntemlerini içinde barındıran kütüphenedir. Farklı algoritmalarla ve farklı hiper parametrelerle veri analitiği yapmanıza ve yapılan analitik sonucunu istediğiniz metrikler ile, değerlendirilmesine olanak sağlamaktadır.

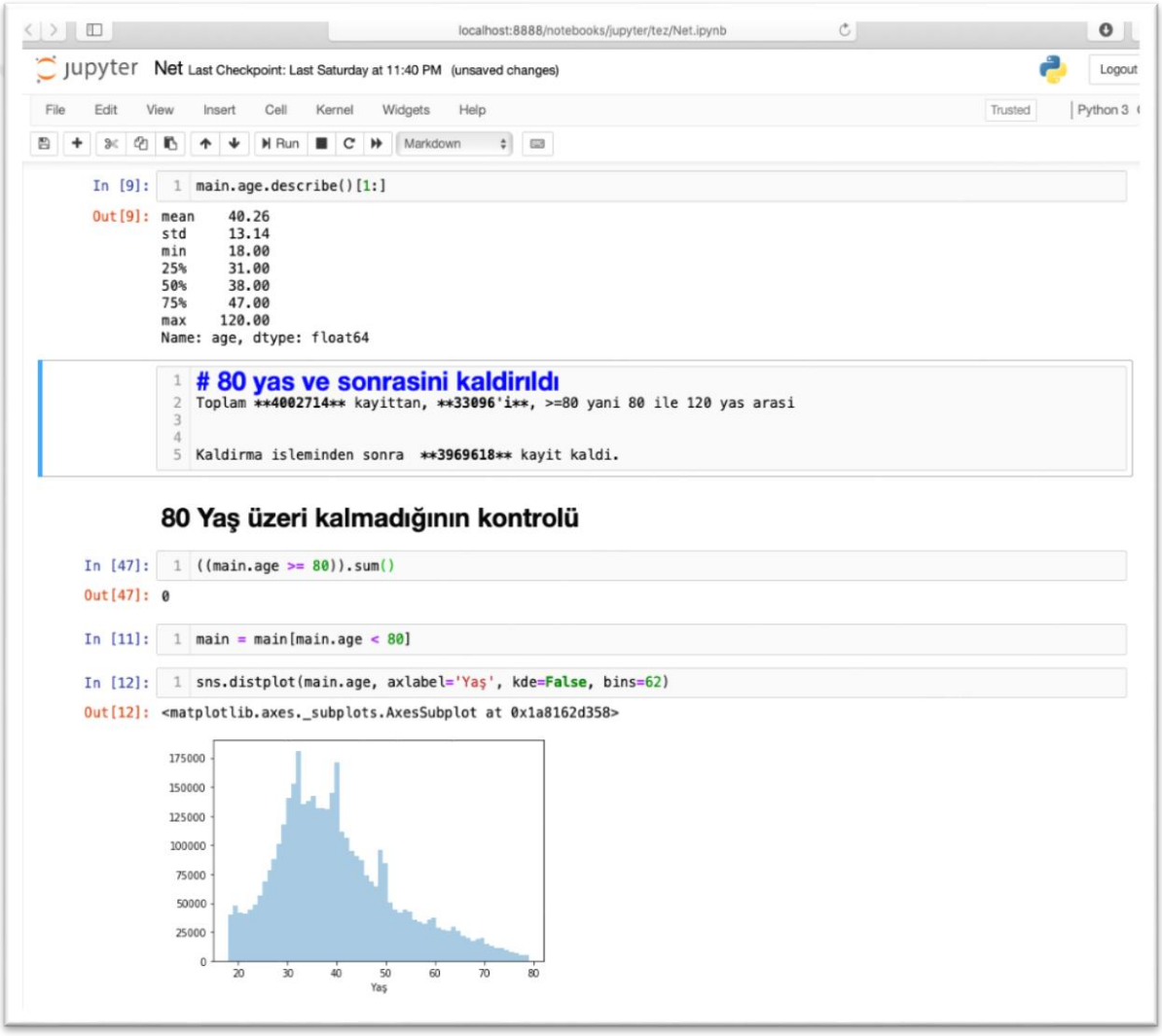
### 1.3.5. Jupyter Notebook

Bu çalışmalarda kod editörü olarak web tabanlı Jupyter Notebook kullanılmıştır. Öncelikle Ipython Notebook adıyla ortaya çıkmış, sonradan ismi Jupyter Notebook'a dönümüş açık kaynaklı geliştirme ortamının diğer kod editörlerine göre artısı, kod ve sunum elementlerini bir arada barındırmasıdır. IPython yani Interactive Python temelli olup, anlık olarak kodu çalıştırıp sonuçları göstermesi ve

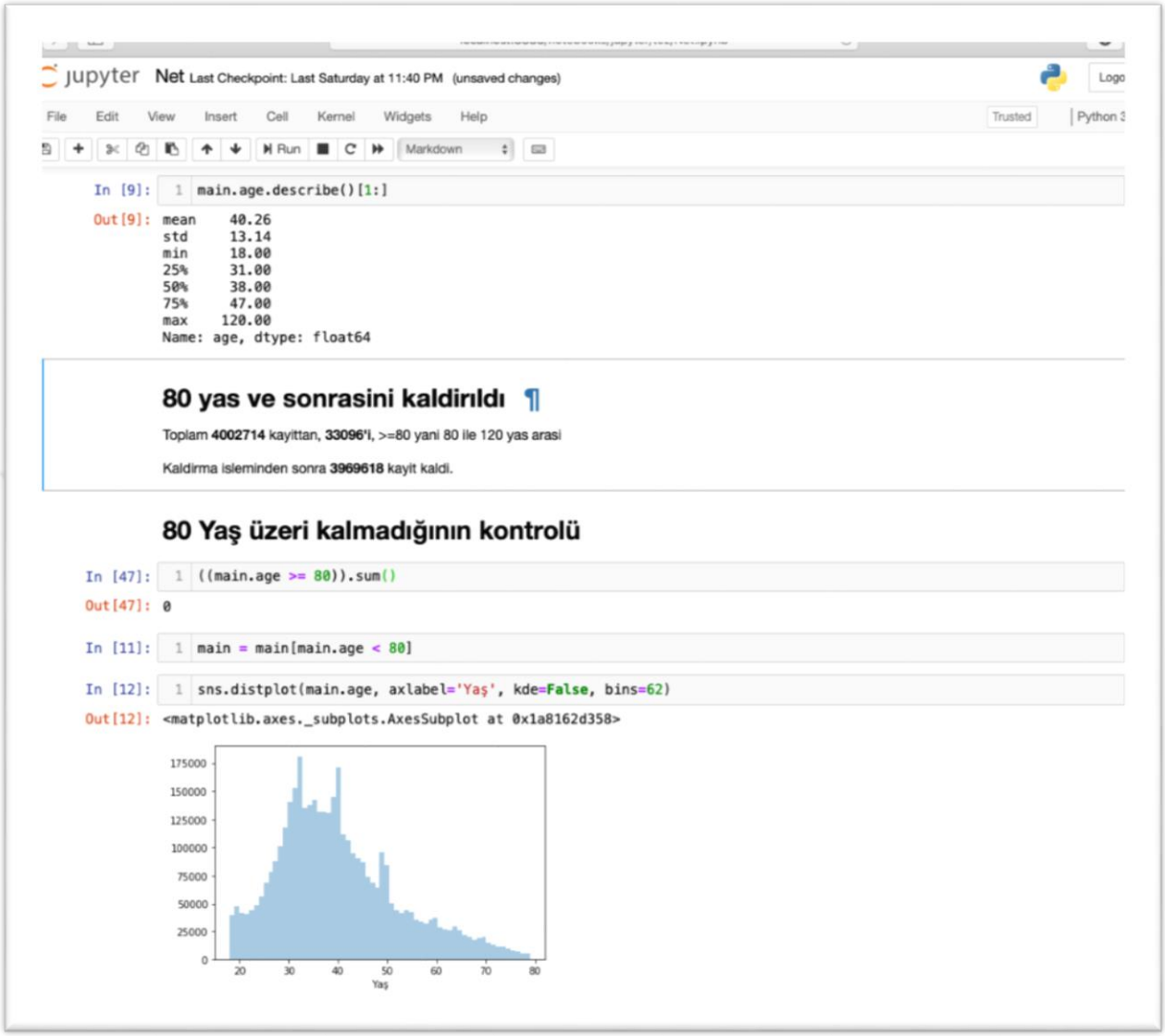
kod blokları arasına markdown isimli, basit bir HTML kodlama formatına imkan vermesi çalışmaların hem diğer iş arkadaşlarıma sunması, hem de makine öğrenmesi iş akışında çalışmaların bütünselliğini kaybetmeden çalışılabilmesine olanak sağlıyor.

Örnek olarak markdown kullanılan bir hücrenin yorumlanmadan önceki hali Şekilde 2'de, Jupyter tarafından yorumlandıktan sonraki görünümü Şekil 3'te sunulmuştur.

**ŞEKİL 2: Jupyter ve Markdown ile yazılmış kod yorumlanmadan önce**



**ŞEKİL 3: Jupyter ve Markdown ile yazılmış kod yorumlandıktan sonra**




Ayrıca Şekil 3'de göründüğü gibi, sns.distplot(main.age, axlabel='Yaş', bins=62) komutuyla Seaborn'dan kullanılarak, Jupyter üzerinde grafik gösterimleri de yapmak mümkündür. Jupyter ile sadece Python değil, R ve Julia kodları da aynı şekilde (ancak farklı Notebook'lar içerisinde) çalıştırılabilmektedir.

### 1.3.6. Anaconda

Bilimsel Python ekosisteminde Python, Matplotlib, Numpy, Scipy, Bokeh, Jupyter, Scikit Learn, ve daha pekçok faydalı kütüphane bulunmaktadır. Zaman zaman da, kütüphaneler, birbirinden faydalanmakta ve bu da kütüphaneler arasında bir versiyon yönetimi (dependency management) ihtiyacı oluşturmaktadır. Aslında Python'un pip isimli kendine özgü bir paket yükleme sistemi vardır. Ancak içerisinde pek çok bilimsel kütüphaneyi barındırması sebebiyle, ücretsiz olan Anaconda, ve bunun daha kompakt hali miniconda paketleri kurulum ve yönetim için büyük kolaylık sağlamaktadır.

# İKİNCİ BÖLÜM

## 2. UYGULAMA

### 2.1. Veri ve Hafıza Optimizasyonu

CSV'lerden okunan veriyi lokal bilgisayardaki Pandas DataFrame'lerde yönetirken, verinin boyutunun yaklaşık 4 gigabyte boyutuna ulaşması, makine öğrenme algoritmalarını uygularken de, daha da fazla hafıza gerektirecek olması sebebiyle veri aşağıdaki şekilde optimize edilmiştir.

Öncelikle tarih içeren verileri, tarih veri tipine, metin olarak saklanan sayısal değerleri de sayısal veri türüne dönüştürdükten sonra bile, ve hafızada (RAM) 4 GB alan kaplamaktaydı.

Veri incelendiğinde, aslında negatif sayısal değerler içermeyen alanlar ve 8 bitlik alana sığabilecek kadar küçük olan (yani 256 dan küçük olan) alan bile pandas'ın varsayılan sayısal alanı olan 32 bitlik alanlarda tutuluyordu.

Bu alanları ve en büyük en küçük değerlerini tek tek taramak yerine, hafıza kullanımını optimize eden bir metod kullanarak bu tüm dataframe için sağlanmıştır (Guillaume 2019).

Bu optimizasyon ile hafızada 4,1 GB yer kaplayan veri, %71 iyileştirme ile, 1,1 GB'a inmiştir. Burada yapılan optimizasyonun sonucu Şekil 4'te gösterilmiştir.

**ŞEKİL 4: Pandas ile Hafıza Optimizasyonu**

In [12]:
```python
# https://www.kaggle.com/gemartin/load-data-reduce-memory-usage
def reduce_mem_usage(df):
    """ iterate through all the columns of a dataframe and modify the data type
        to reduce memory usage.
    """
    start_mem = df.memory_usage().sum() / 1024**2
    print('Optimizasyon öncesi hafıza kullanımı {:.2f} MB'.format(start_mem))

    for col in df.columns:
        col_type = df[col].dtype

        if col_type != object:
            c_min = df[col].min()
            c_max = df[col].max()
            if str(col_type)[:3] == 'int':
                if c_min > np.iinfo(np.int8).min and c_max < np.iinfo(np.int8).max:
                    df[col] = df[col].astype(np.int8)
                elif c_min > np.iinfo(np.int16).min and c_max < np.iinfo(np.int16).max:
                    df[col] = df[col].astype(np.int16)
                elif c_min > np.iinfo(np.int32).min and c_max < np.iinfo(np.int32).max:
                    df[col] = df[col].astype(np.int32)
                elif c_min > np.iinfo(np.int64).min and c_max < np.iinfo(np.int64).max:
                    df[col] = df[col].astype(np.int64)
            else:
                if c_min > np.finfo(np.float16).min and c_max < np.finfo(np.float16).max:
                    df[col] = df[col].astype(np.float16)
                elif c_min > np.finfo(np.float32).min and c_max < np.finfo(np.float32).max:
                    df[col] = df[col].astype(np.float32)
                else:
                    df[col] = df[col].astype(np.float64)
        else:
            df[col] = df[col].astype('category')

    end_mem = df.memory_usage().sum() / 1024**2
    print('Optimizasyon sonrası hafıza kullanımı: {:.2f} MB'.format(end_mem))
    print('Düşüş oranı %{:.1f}'.format(100 * (start_mem - end_mem) / start_mem))

    return df
```

In [13]:
```python
optimized_main_data = reduce_mem_usage(main_data)
```

```
Optimizasyon öncesi hafıza kullanımı 4130.30 MB
Optimizasyon sonrası hafıza kullanımı: 1164.59 MB
Düşüş oranı %71.8
```

Veride yapılan bu optimizasyonu ve uygun veri tiplerine göre ayarlanmış DataFrame'i, Python'ın makine kodu serializasyon (binary serialization) kütüphanesi pickle ile saklayarak sonraki kullanımlarda da bu optimizasyondan yararlanmaya devam edilmiştir.

Her ne kadar veritabanı normalizasyon metodlarının ve verinin üretilme ve saklanma standartları olgunlaşmış olsa da, canlı ve komplex sistemlerde eksik, gürültülü ya da tutarsız veriler oluşabilmektedir. Sistem mimarisi, programcı ya da veri giriş hatası gibi pek çok farklı sebeple oluşabilen bu veri kirliliğini veri önişleme metodlarıyla en aza indirmek, veriden doğru çıkarımlar yapmak için zorunlu bir adımdır.

Verinin yapısına ve kullanılacak modele göre doğru veri işleme teknikleri uygulanarak mümkün olan en doğru sonuca gitmeyi hedeflenmelidir.

Temel veri ön işleme teknikleri aşağıdaki gibidir;

1. Veri Temizleme
2. Veri Birleştirme
3. Veri Dönüştürme
4. Veri İndirgeme

### 2.2. Hafızadan daha büyük veri ile çalışma

Verimizin işlememiş hali, lokal makinemizdeki hafızadan büyük olduğu durumlarda, Python ekosistemindeki dağıtık sistemler ve parelelleştirme kütüphanesi Dask kullanılarak hafızadaki bu kısıtlamayı aşılabilmektedir.

Dask, Numpy ve Pandas nesnelerini, hafıza optimize şekilde, ve hem lokal makinelerde hem de dağıtık sistemlerde yönetebilmemize olanak sağlıyor. Sabit disk, hafıza ya da ekran kartı işlemcilerinde paralel olarak çalışabilme imkanı sağlıyor.

Veriyi optimize şekilde işleyebilir konuma geldikten sonra, eldeki verinin dağılımını, boş sayı ve oranlarını, ayrkırı değerleri aşağıdaki yöntemlerle incelenmiştir.

### 2.3. Boş Değerler

Veri setindeki boş değerler verideki oranına ve kritikliğine göre değerlendirilip;

- Boş değer içeren satırlar veri setinden çıkarılabilir
- Boş değerler, ilgili öznitelikteki ortamala değer olarak atanabilir
- Regresyon veya karar ağacı gibi teknikler kullanılarak) eksik olan değer için en uygun değer bulunup kullanılabilir.

### 2.4. Min Max Normalizasyonu

Veri setimizdeki sayısal değerlerin birimleri ve minimum maksimum değerleri birbirinden çok fazla farklılık gösterebilir. Makine öğrenme algoritmalarımızın elde edeceği sonuçları, bu birim ve sayısal farklılıktan etkilenmemeleri için min-max normalizasyonu isimli metodu kullanılabilmektedir.

Bu metotla, tüm sayısal değerler 0 ile 1 arasında bir değer alacak şekilde normalize edilmiş olur.

Min – Max normalizasyonunun formülü aşağıdaki gibidir;

$$\Upsilon_i = \frac{\mathcal{X}_i - \min(\mathcal{X})}{\max(\mathcal{X}) - \min(\mathcal{X})}$$

Bu formül sonucunda sayısal bir öznitelikteki minimum değer 0, maksimum değer ise 1 olacak şekilde bir transformasyon yapılmış olunur.

### 2.5. Çarpıklık (Skewness)

Normal dağılım yani çan eğsisinden sapma derecesini gösteren değerdir. Normal dağılımın çarpıklık değeri dır.

Pozitif çarpıklık, eğrinin kuyruğunun artı yöne olduğu durumlarda, negatif çarpıklık ise kuyruğun negatif yönde olmasıyla olur.

Çarpıklık değeri -0,5 ile 0,5 olduğu durumlarda veri oldukça simetrik denilebilir. Çarpıklık değeri -1 ile -0,5 arasıysa negatif çarpıklık vardır. 0,5 ile 1 arasındaysa pozitif çarpıklık vardır. Eğer verilerin çarpıklığı 1 den büyük ya da -1 den küçük ise, yüksek oranda çarpıklık var demektir.

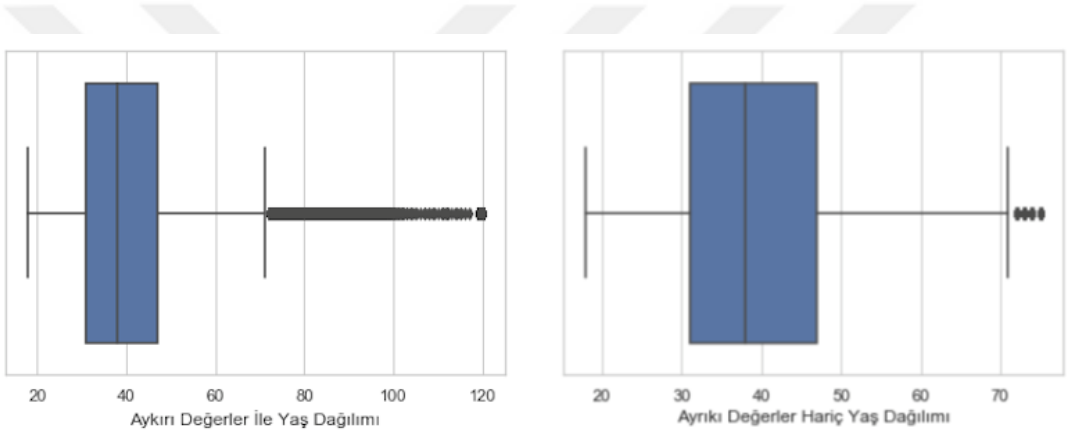
### 2.6. Aykırı Değerler

Veride yaş özniteliğinin dağılımı ve çeyrekler açıklığını (interquartile range) kontrol ettiğimizde, yaş özniteliğininin 18 ile 120 arasında değiştiği görülmüştür. 75 yaş üzeri analizimiz için temsil edici olduğu düşünülen kitlenin dışında kaldığı için, 75 yaş ve üzeri kayıtları veri setinden çıkarılmıştır. 75 yaş ve üzeri kayıtlar çıkarılmadan önce ve sonraki çeyrekler açıklık değerleri Tablo 3'de, bu tablosunun kutu grafiği Şekil 5'de verilmiştir.

**TABLO 3: Yaş Özniteliği Çeyrekler Açıklığı**

| | Aykırı Değerlerle Birlikte | Aykırı Değerler Hariç |
|---|---|---|
| Kayıt Adedi | 4.002.714 | 3.944.682 |
| Ortalama | 40,26 | 39,64 |
| Standart Sapma | 13,14 | 12,18 |
| Minimum | 18 | 18 |
| % 25'lik çeyrek | 31 | 31 |
| % 50'lik çeyrek | 38 | 38 |
| % 75'lik çeyrek | 47 | 47 |
| Maksimum | 120 | 75 |

**ŞEKİL 5: Yaş Özniteliği Çeyrekler Açıklığı Grafiği**




Bu aykırı değerler çıkarıldığında veri seti 4.002.714 kayıttan, 3.944.682 kayda inmiştir.

### 2.7. Nitelik Oluşturma

Elimizdeki veri setini gözetimli ve gözetimsiz makine öğrenmesi modellerini uygulamadan önce aşağıdaki gibi, tamamen sayısal değerler içeren yeni öznitelikler oluşturdum. Böylece örneğin doğum tarihi gibi, çok fazla farklı değer içerebilecek, ve ordinal bir veriyi, ratio veri tipine, yani anlamlı bir sıfır değeri olan bir veri şekline dönüştürmüş oluruz. Rezervasyonlar, kişiler ve katılımcılar tablolarındaki veriler derlenerek oluşturulmuş öznitelikler Tablo 4'te verilmiştir.

**TABLO 4: Feature Extraction sonucu öznitelikler**

| Alan Adı | Açıklama |
|---|---|
| Age | Müşterinin yaşı |
| total_res_count | Müşterinin toplam yaptığı rezervasyon sayısı |
| months_since_last_res | Müşterinin yaptığın son rezervasyondan bugüne kadar geçen ay ortalaması |
| avg_month_between_vocation_arrival_dates | Tatiller arasındaki ay ortalaması |
| total_amount | Tüm tatillere harcanan toplam para |
| total_accomodation | Tüm tatillerde kalınan gece sayısı |
| unit_price | Bir kişi için bir gece ödenen para miktarı |
| total_days_between_res_and_arrival | Rezervasyon yapılmasıyla tatile gidilmesi arasında geçen toplam gün sayısı |
| average_days_between_res_and_arrival | Rezervasyon yapılmasıyla tatile gidilmesi arasında geçen ay ortalaması |
| min_days_between_res_and_arrival | Rezervasyon ile tatile gidiş arasında geçen en küçük gün sayısı |
| max_days_between_res_and_arrival | Rezervasyon ile tatile gidiş arasında geçen en büyük gün sayısı |
| average_res_amount | Rezervasyon başına harcanan ortalama para |
| average_res_accomodation | Rezervasyonlarda kalınan ortalama gece sayısı |
| total_pax | Rezervasyonlarda toplam para ödenen kişi sayısı |
| total_adult | Rezervasyonlarda toplam yetişkin sayısı |

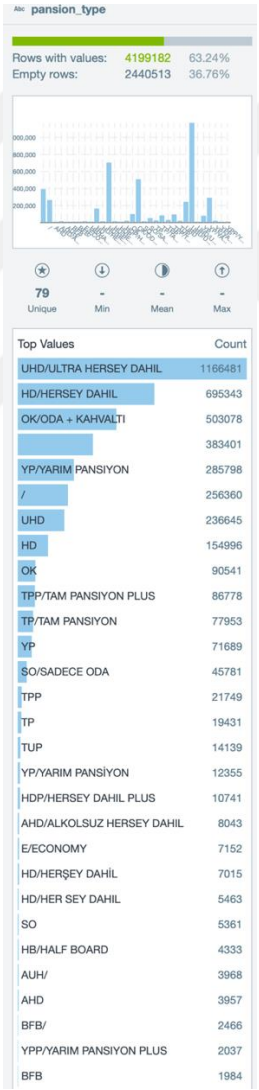
| | |
|---|---|
| total_child | Rezervasyonlarda toplam çocuk sayısı |
| _2014_res_count | Müşterinin 2014 yılında yaptığı toplam rezervasyon sayısı |
| _2015_res_count | Müşterinin 2015 yılında yaptığı toplam rezervasyon sayısı |
| _2016_res_count | Müşterinin 2016 yılında yaptığı toplam rezervasyon sayısı |
| _2017_res_count | Müşterinin 2017 yılında yaptığı toplam rezervasyon sayısı |
| _2018_res_count | Müşterinin 2018 yılında yaptığı toplam rezervasyon sayısı |
| _2019_res_count | Müşterinin 2019 yılında yaptığı toplam rezervasyon sayısı |
| q1 | Müşterinin yılın ilk çeyreğinde yaptığı toplam rezervasyon sayısı |
| q2 | Müşterinin yılın ikinci çeyreğinde yaptığı toplam rezervasyon sayısı |
| q3 | Müşterinin yılın üçüncü çeyreğinde yaptığı toplam rezervasyon sayısı |
| q4 | Müşterinin yılın dördüncü çeyreğinde yaptığı toplam rezervasyon sayısı |
| active_year_count | Müşterinin rezervasyon yaptığı toplam yıl sayısı |
| tkoy_res_count | Tatil köyü rezervasyon sayısı |
| kult_res_count | Kültür turu rezervasyon sayısı |
| ydis_res_count | Yurtdışı rezervasyon sayısı |
| | |

### 2.8. Tekilleştirme

Bu alanda, rezervasyon sistemlerinin veriyi tutuş şekli veri girişi hatası sebebiyle aynı türdeki konaklama tipleri farklı isimlerle tutulduğu tespit edilmiştir.

Farklı değerler ve dağılımları aşağıdaki Şekil 6'te verilmiştir.

**Şekil 6: Konaklama Türü Alanında Veri Dağılımı**



Makine öğrenmesi yöntemlerimize veriyi bu haliyle girdi olarak verirsek, olması gerekenden daha fazla kırılım yapmış ve hatalı sonuçlar elde etmiş olurduk. Çünkü çalıştırdığımız modelin örneğin OK yani oda kahvaltı ile, BFB yani "bed and breakfast" anlamına geldiklerini ve bu ikisinin aslında aynı anlama geldiklerini bilmesinin imkanı yoktur. Bu sebeple eldeki veri, veriyi üreten sistemleri kullanan iş birimi uzmanlarıyla ortak çalışarak tekilleştirilmiştir.

Aşağıdaki tabloda örnek olarak oda kahvaltı ve yarım pansiyon değerleri için veride bulunan değerler ve adetler aşağıdaki Tablo 5'te verilmiştir.

**TABLO 5: Tekilleştirme Öncesi Dağılım**

| Verideki değer | Adet | Olması Gereken değer |
|---|---|---|
| OK/ODA + KAHVALTI | 503.079 | ODA KAHVALTI |
| OK | 89.050 | ODA KAHVALTI |
| BFB/ | 2.466 | ODA KAHVALTI |
| BFB | 1.920 | ODA KAHVALTI |
| OK/ODA KAHVALTI | 645 | ODA KAHVALTI |
| BB | 290 | ODA KAHVALTI |
| BB/BAD & BREAKFAST | 233 | ODA KAHVALTI |
| YP/YARIM PANSIYON | 285.798 | YARIM PANSIYON |
| YP | 70.625 | YARIM PANSIYON |
| YP/YARIM PANSİYON | 12.355 | YARIM PANSIYON |
| HB/HALF BOARD | 4.333 | YARIM PANSIYON |

Yaptığımız veri temizleme çalışmasından sonraki dağılım Tablo 6'te gösterilmişir.

**TABLO 6: Tekilleştirme Sonrası Dağılım**

| Yeni Değer | Adet |
|---|---|
| ODA KAHVALTI | 597.683 |
| YARIM PANSIYON | 373.111 |

Geri kalan eşleşmeler de yapılarak konaklama tipi alanı, olması gereken değerlerlerle standartlaştırılmıştır.

### 2.8. Makine Öğrenmesi Uygulaması

Gözetimli öğrenmede öğrenme algoritmalarına girdi değişkenleriniz (x) ve bir adet çıktı değişkeniniz (y) ayrı ayrı verilip öğrenme sağlanır.

$Y = f(X)$

Öğrenme sonrasında yeni X değişkenlerinizi girdi olarak verdiğinizde, size olması gereken Y değerini tahminleme yeteneğini verir.

Gözetimli öğrenme yöntemleri de temelde sınıflandırma (classification) ve regresyon (regression) olarak ikiye ayrılmaktadır.

#### 2.8.1. Sınıflandırma

Tahminlemek istediğimiz Y değeri kategorik bir bilgi ise (mavi/kırmızı/yeşil, üçgen/ kare, hasta/hasta değil vb.) bu tip problemler sınıflandırma problemleri olarak adlandırılmaktadır.

#### 2.8.2. Regregresyon ve Sınıflandırma Problemlerinin Farkı

Değişkenler nitel (kalitatif ya da kategorik) ve nicel (kantatif) olarak ikiye ayrılırlar.

Kantatif değişkenler yaş, boy, kilo, satış fiyatı gibi değişkenlerdir ve sayısal değerler alırlar. Kalitatif değişkenler ise, (cinsiyet, renk, hasta / hasta değil vb.) kategorik değerler alabilirler.

Sayısal yani kantatif değerleri tahminlenmeye çalıştığımız problemleri regresyon, kategorik verileri tahminlenmeye çalışan problemlere sınıflandırma problemleri olarak bakılabilir.

### 2.8.3. Lojistik Regresyon

Logistik regresyon ise, temelde regresyon yöntemi olsa da, tahminlenmeye çalıştığı değer iki kategorik bilgiden birine ait olup (hasta, hasta değil gibi) zaman zaman sınıflandırma metodu olarak da kullanılabilir. Bu durumda, örneğin 'hasta mı' değerini tahminlediğimizde, burada elde edeceğimiz değerler 0 ile 1 arasında olup, 0,9 gibi bir değer elde ediyorsak, %90 ihtimalle hasta, %10 ihtimalle hasta değil sonucuna ulaşmış olunur.

Bu çalışmada Lojistik Regresyon, Tatil Köyü alır ya da almaz olarak iki ayrı sınıfa ayrılıp, çıkan sonuçlar diğer algoritmaların sonuçlarıyla karşılaştırılmıştır.

### 2.8.4. Karar Ağaçları

Karar ağaçları, veri yapılarındaki ağaç (tree) mantığını temel alarak çalışmaktadır. Her uç nokta 0 ila 2 arasında dallanma yapabilir (node). Makine öğrenmesi algoritmalarında CART (Classification and Regression Trees) olarak adlandırılır. Her karar noktasında bir evet / hayır sorusunun sorulduğu ve cevaplandığı ve bir akış vardır.

Her sorulan soru ile, hedef kümeye giden yolda en çok veri kazanımı hedeflenmektedir. Karar ağaçlarının en büyük avantajı, veriyi biri sıcak (one-not kodlama, ya da sayısal hale gibi işlemlere gerek duymadan işleyebilmesi ve kategorik veriyle çalışabilmesi ve bu sebeple yorumlaması da sonuçlarınının yorumlanması diger algoritmalara göre daha anlaşılırdır.

Karar ağaçlarında nodeların seçimi yapılırken, algoritmada sorulan sorular, mümkün olan en çok bilgi kazanımı (information gain) sağlacak şekilde optimize edilmektedir.

Her node'da, yapılabilecek kırlım sonrasında edinilecek bilgi kazanımı aşağıdaki formül ile hesaplanmaktadır.

$$IG_{x,a}(X,a) = D_{kL}(P(x|a)||P_x(x|I))$$

Veri analiz edilmeden önce verinin %70'ini eğitim, %30'unu da test verisi olarak ayrılmıştır.

Bu ksım Scikit Learn ile aşağıdaki kod bloğu ile yapılmıştır.

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(
   X, y, test_size=0.3, random_state=17)
```

Burada, scikit learn kütüphanesinden hem algoritmaya girdi olarak ydis_res_count, krms_res_count, kult_res_count, age, average_child_per_res, yaz_haziran_eylul_adet kolonlarından oluşan özniteliklerimizi, hem de bunların karşılığı olan Y değerleri, Tatil Köyü alır mı verilerek elde edilen sonuçların kesinlik ve hassasiyet değerleri Tablo 7'de verilmiştir.

**TABLO 7: Kesinlik ve Hassasiyet Değerleri**

| | **Tatil Köyü Almaz** | **Tatil Köyü Alır** | **Doğruluk** |
|---|---|---|---|
| **Kesinlik** | 0.94 | 0.93 | 0.94 |
| **Hassasiyet** | 0.98 | 0.82 | 0.94 |
| **$F_1$-score** | 0.96 | 0.87 | 0.94 |

$F_1$ Skoru aşağıdaki formüle göre hesaplanmıştır (Şeker 2019)

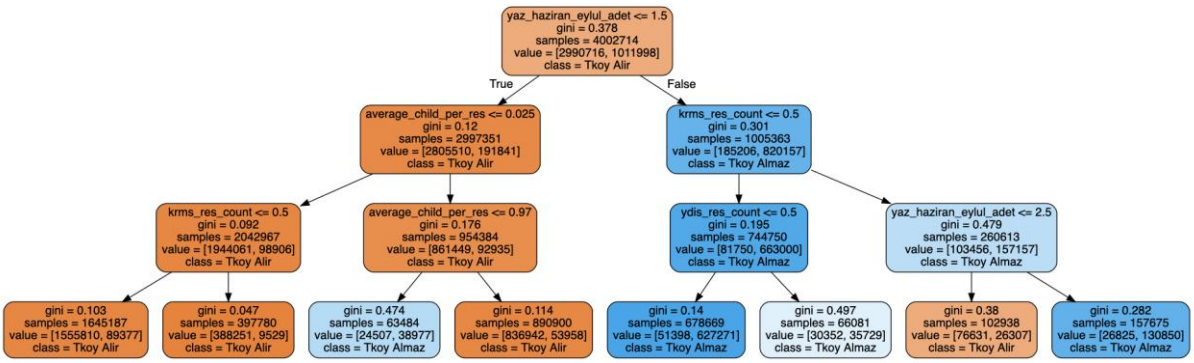
$$\boldsymbol{F_1 = 2 \times \frac{Kesinlik \times Hassasiyet}{Kesinlik + Hassasiyet}}$$

$F_1$ Skor Formülü

### 2.8.5. Rassal Ormanlar

Rassal ormanlar karar ağacı yöntemindeki veriyi ezberleme (overfitting) durumunu aşacak şekilde çalışmaktadır (Friedman, Jerome, Hastie ve Tibshirani 2001). Çalıştırılan 3 seviyeki rassal ormanlar model çıktı örneği Şekil 7'da verilmiştir.

**Şekil 7:Rassal Ormanlar Model Çıktısı**




### 2.8.6. Yapay Sinir Ağları ile Derin Öğrenme

Yapay sinir ağları (artificial neural networks) temellerini insan beyninin karar alma yönteminden esinlenerek kurgulanmış bir yöntemdir. Nöronların çalışma şeklinin basitleştirilmiş halinden ilham alan bir makine öğrenmesi yöntemidir.

Perceptron ya da algılayıcılar, 1957 yılında Frank Rosenblatt tarafından geliştirilen, temel bir yapay sinir ağı yapı taşıdır (Géron 2019, 132-138).

Hata/Kayıp formülü aşağıdaki gibi hesaplanmaktadır;

$$J(w) = \frac{1}{2}\sum_{i=0}^{n}(hedef^{(i)} - çıktı^{(i)})^2.$$

Çok katmanlı algılayıcı her gizli katmanda bu hata değerini minimize edecek şekilde çalışmaktadır (Şeker 2019).

Hatayı minimize ederken ise, perceptronların çıktıları aşağıdaki fonksiyon ile hesaplanmaktadır (Nicholson 2020)

$$y = \varphi\left(\sum_{i=1}^{n} w_i x_i + b\right) = \varphi(w^t x + b)$$

### 2.8.7. Yapay Sinir Ağları ile Tatil Köyü Satın Alacak Müşteri Tahminleme

Multilayer Perceptron yapay sinir ağı ile, 5 adet gizli katman kullanılarak yapılan analitik sonucunda tatil köyü alma tahminlemesi 0.92 kesinlik ve 0.77 hassasiyeti ile tahminlenmiştir.

Analizde kullanılan multilayer perceptron sınıflandırıcısının parametreleri aşağıdaki gibidir;

MLPClassifier(activation='relu', alpha=1e-05, batch_size='auto', beta_1=0.9, beta_2=0.999, early_stopping=False, epsilon=1e-08, hidden_layer_sizes=(5, 5, 5, 5, 5, 5), learning_rate='constant', learning_rate_init=0.001, max_fun=15000, max_iter=500, momentum=0.9, n_iter_no_change=10, nesterovs_momentum=True, power_t=0.5, random_state=1, shuffle=True, solver='lbfgs', tol=0.0001, validation_fraction=0.1, verbose=False, warm_start=False)

Bu model, test veri setine uygulanıp başarı oranları değerlendirilmiş ve kesinlik, hassasiyet ve $F_1$ Skorları aşağıdaki TABLO 8'te verilmiştir.

**TABLO 8: Yapay Sinir Ağları, Kesinlik ve Hassasiyet Değerleri**

| | Tatil Köyü Almaz | Tatil Köyü Alır | Doğruluk |
|---|---|---|---|
| Kesinlik | 0.93 | 0.92 | 0.92 |
| Hassasiyet | 0.98 | 0.77 | 0.92 |
| $F_1$-Skoru | 0.95 | 0.84 | 0.92 |

### 2.8.8. Kayıp Müşteri Analizi

Kaybedilmiş müşteri, toplamda en az iki alış verişi bulunan, ancak son iki yıldır herhangi bir alış veriş yapmamış kişiler olarak tanımlanmış, ve bu tanıma uyan kişiler kayıp müşteri olarak değerlendirilmiş, ek bir öznitelik olarak veriye eklenmiştir.

Rassal ormanlar yöntemiyle yapılan analiz sonucu kesinlik ve hassasiyet değerleri Tablo 9'da, yapay sinir ağları yöntemi ile yapılan analiz sonucu ise Tablo 10'da verilmiştir.

**TABLO 9: Kayıp Müşteri Analizi Rassal Ormanlar Kesinlik ve Hassasiyet**

| | Kayıp Müşteri | Kayıp Müşteri Değil | accuracy |
|---|---|---|---|
| Kesinlik | 0.97 | 0.93 | 0.96 |
| Hassasiyet | 0.99 | 0.84 | 0.96 |
| $F_1$-Skoru | 0.98 | 0.88 | 0.96 |

**TABLO 10: Kayıp Müşteri Yapay Sinir Ağları Kesinlik ve Hassasiyet**

| | Kayıp Müşteri | Kayıp Müşteri Değil | Doğruluk |
|---|---|---|---|
| precision | 0.82 | 0.17 | 0.79 |
| Hassasiyet | 0.94 | 0.06 | 0.79 |
| $F_1$-Skoru | 0.88 | 0.08 | 0.79 |

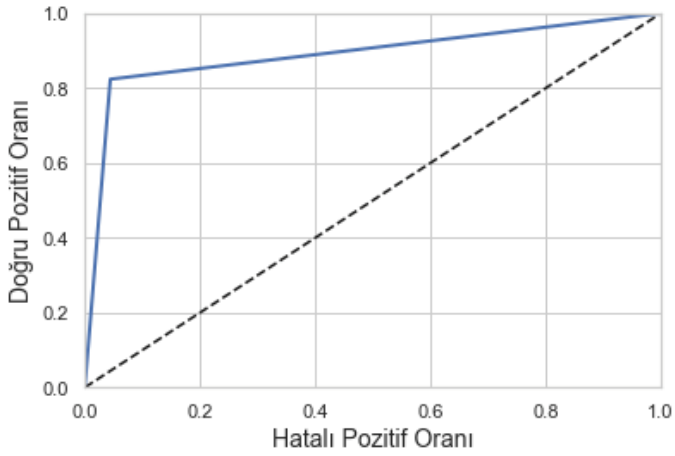
Yapay sinir ağları yönteminde kullanılan hiper parametreler aşağıdaki gibidir;

MLPClassifier(activation='relu', alpha=1e-05, batch_size='auto', beta_1=0.9,
beta_2=0.999, early_stopping=False, epsilon=1e-08,
hidden_layer_sizes=(8, 8, 8, 4), learning_rate='constant',
learning_rate_init=0.001, max_fun=15000, max_iter=1000,
momentum=0.9, n_iter_no_change=10, nesterovs_momentum=True,
power_t=0.5, random_state=17, shuffle=True, solver='lbfgs',
tol=0.0001, validation_fraction=0.1, verbose=False,
warm_start=False)

### 2.8.9. ROC Eğrisi

İkili sınıflandırma (binary classification) sistemlerinde, seçilen ayrım eşik değerinin hassasiyet ve kesinliğe etkisi ROC (Receiver Operating Characteristic) değeri kullanılmaktadır (Géron 2019). Çeşitli eşik değerlendirme seviyelerine göre alınan ROC eğrisi ŞEKİL 8'da gösterilmiştir.

**ŞEKİL 8: Seçilen Eşik Değerlere Göre ROC Eğrisi**




### 2.8.10. Grid Search ve Çapraz Doğrulama

Modellerimizden başarılı sonuçlar alabileceğimi düşündüğümüz modelleri, daha da iyileştirmek için farklı hiper parametrelerle denemeler yapıp, en iyi sonuç vereni tespit etmek için ızgara araması çapraz doğrulama (grid search) metodu uygulanmıştır.

Grid search yöntemiyle denediğimiz hiper parametre konfigürasyonları

{'n_estimators': [3, 10, 30], 'max_features': [2, 4, 6, 8]},

{'bootstrap': [False], 'n_estimators': [3, 10], 'max_features': [2, 3, 4]},

Şeklinde ve öznitelikler olarak aşağıdaki öznitelikler kullanılmıştır.

- age
- sira_no_first_count
- yaz_haziran_eylul_adet
- kbrs_res_count
- kult_res_count
- ydis_res_count
- etcr_res_count
- krms_res_count
- crus_res_count
- dsny_res_count

- tuls_res_count
- ydmf_res_count
- ydot_res_count
- haci_res_count
- call_center_count
- total_adult
- total_child
- active_year_count
- unitprice_mapped_for_2019

Verimiz üzerinde rassal ormanlar algoritmasını çalıştırdığımızda elde ettiğimiz sonuçları farklı parametrelerle denemek için grid search yöntemini kullanarak yaptığımız çapraz doğrulama sonuçları aşağıdaki gibidir.

1. 569.7595571096091 {'max_features': 2, 'n_estimators': 3}
2. 531.6905412382102 {'max_features': 2, 'n_estimators': 10}
3. 518.9701285673287 {'max_features': 2, 'n_estimators': 30}
4. 562.8805172731247 {'max_features': 4, 'n_estimators': 3}
5. 527.2983708924961 {'max_features': 4, 'n_estimators': 10}
6. 517.7489405804271 {'max_features': 4, 'n_estimators': 30}
7. 572.510011434212 {'max_features': 6, 'n_estimators': 3}
8. 528.727388448286 {'max_features': 6, 'n_estimators': 10}
9. 517.5471958647355 {'max_features': 6, 'n_estimators': 30}
10. 563.1981497286762 {'max_features': 8, 'n_estimators': 3}
11. 527.7567198903652 {'max_features': 8, 'n_estimators': 10}
12. 514.5251888088804 {'max_features': 8, 'n_estimators': 30}
13. 564.7923425578042 {'max_features': 10, 'n_estimators': 3}
14. 527.4708749364369 {'max_features': 10, 'n_estimators': 10}
15. 515.5980468157312 {'max_features': 10, 'n_estimators': 30}
16. 564.300421823666 {'max_features': 12, 'n_estimators': 3}
17. 527.2435706435394 {'max_features': 12, 'n_estimators': 10}
18. 517.0855277545315 {'max_features': 12, 'n_estimators': 30}

19. 569.6152167936464 {'bootstrap': False, 'max_features': 2, 'n_estimators': 3}
20. 537.0411571576315 {'bootstrap': False, 'max_features': 2, 'n_estimators': 10}
21. 572.4338960374732 {'bootstrap': False, 'max_features': 3, 'n_estimators': 3}
22. 533.1536611210014 {'bootstrap': False, 'max_features': 3, 'n_estimators': 10}
23. 567.6323326602893 {'bootstrap': False, 'max_features': 4, 'n_estimators': 3}
24. 538.5522864318272 {'bootstrap': False, 'max_features': 4, 'n_estimators': 10}
25. 572.166333908437 {'bootstrap': False, 'max_features': 6, 'n_estimators': 3}
26. 538.3721092998153 {'bootstrap': False, 'max_features': 6, 'n_estimators': 10}
27. 573.9781949068911 {'bootstrap': False, 'max_features': 8, 'n_estimators': 3}
28. 538.7460299628991 {'bootstrap': False, 'max_features': 8, 'n_estimators': 10}
29. 575.8041142205362 {'bootstrap': False, 'max_features': 10, 'n_estimators': 3}
30. 538.5352612660749 {'bootstrap': False, 'max_features': 10, 'n_estimators': 10}

Burada elde ettiğimiz minimum hata oranlarından biri olan, ve model kompleksitesi düşük olan "517.7489405804271 {'max_features': 4, 'n_estimators': 30}" degeri en uygun hiper parametre değerlerinden biri olarak gözükmektedir.

Grid search üzerinden tüm veriyi çapraz doğrulama ve çesitli hiper parametrelerle denemeler yaptığımız için bu sonuclari elde etmek 4 cekirdekli bir Intel i7 işlemcili bir bilgisayarda 4 saat 25 dakika sürmüştür.

### 2.8.11. Öznitelik önem ağırlıkları

Her özniteliğin önem ağırlıkları her ağaç düğümünde azalttığı belirsizliklerin ortalaması alınarak bulunur. Izgara arama yönteminden faydalanarak hangi hiper parametrelerle hangi niteliklerin doğru sonuçlar verdiğini çapraz doğrulama yöntemiyle tespit edip, doğru sonuca ulaşmak için hangi öznitelikleri kullanmamız gerektiğini tespit edebiliriz.

Çalıştırılan çapraz doğrulama sonucunda özniteliklerin önem ağırlıkları aşağıdaki olmuştur.

[(0.41302156574226956, 'total_adult'),
(0.1938532586192395, 'yaz_haziran_eylul_adet'),
(0.11410477354068124, 'sira_no_first_count'),
(0.0867560707241225, 'active_year_count'),
(0.05576998915735845, 'age'),
(0.04049215557987964, 'total_child'),
(0.027806963427197236, 'krms_res_count'),
(0.027001873953564504, 'kbrs_res_count'),
(0.01611274059155251, 'ydis_res_count'),
(0.006496005244404733, 'dsny_res_count'),
(0.0057655314405680004, 'etcr_res_count'),
(0.004114369332862147, 'kult_res_count'),
(0.003537738337291377, 'ydmf_res_count'),
(0.002346465675831998, 'ydot_res_count'),
(0.002341534992954713, 'crus_res_count'),
(0.0004746086464746311, 'haci_res_count'),
(4.354993747271865e-06, 'tuls_res_count'),
(0.0, 'call_center_count')]

### 2.8.12. Aşırı Uyma ve Aşırı Genelleme

Aşırı uyma (overfitting) ve aşırı genelleme (underfitting) makine öğrenmesi uygulamalarında, çıkrığının başarısını etkileyen iki kavramdır.

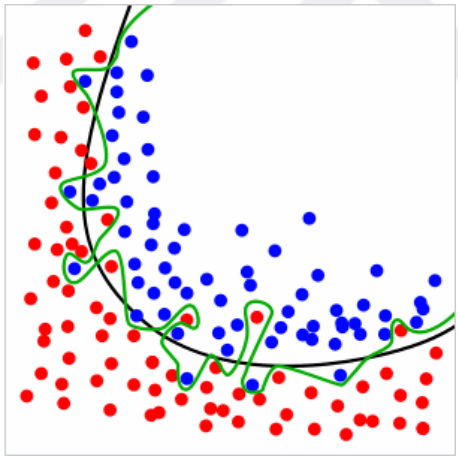
Makine öğrenme algoritmalarına gereğinden fazla öz nitelik besler ve hiper parametre konfigürasyonlarını da fazla hassaslaştırırsak, algoritmamız eğitim kümesine aşırı uygun bir öğrenim yapabilir. Bu, aslında öğrenme ve yeni örneklerde doğru çalışmadan çok, eğitim kümesinin ezberlenmesi olarak yorumlanabilir.

Bunun sonucunda algoritmamız test kümesi için çok doğru sonuçlar verse de, test kümesinde ya da algoritmanın görmediği yeni bir örnek geldiğinde iyi sonuçlar vermeyecektir.

Makine öğrenmesi algoritmalarında amacımız, eğitim ve tahminlime surecinin uygun bir genelleme oluşturmasını sağlamaktır. Şekil 8'de görülen mavi ve kırmızı iki ayrı sınıfın ayrımı, yeşil çizgi ile gösterilmiş bir sınıflandırıcıyla ayrıştırıldığında acık yeşil çizgi ile gösterilen şekilde ayrıldığında, bu ayrım test kümemiz için fazlasıyla iyi sonuclar verecektir (yüksek varyans).

Şekilde 9'deki siyah çizgi ile gösterilen ayrımı yaparsak, algoritma henüz görememiş olduğu veriye daha uygun genelleme yapabilecektir. Aşırı genelleme yapmak (bias) ise, algoritmamızın özellikle sınır değerlerde yanlış sınıflandırmalar yapmasına sebep olacaktır.

**ŞEKİL 9: Aşırı Uyma ve Aşırı Genelleme Grafiği**



## 2.9 Literatür Araştırması

Literatürde makine öğrenmesi metotlarının doğrudan satış ve yüksek miktarda verinin bulunduğu pek çok farklı alanda kullanıldığı görülmektedir.

Özellikle müşteri profilleme, satış ve doğrudan satış kampanyaları için yaygın olarak kullanılmaktadır ve artan rekabet ortamında kurumların ayakta kalabilmesi için belli oranda zorunlu da olmuştur (Küçükbatır 2018).

Teknoloji artık turizm sektörü için sadece basit bir ek satış kanalı olmaktan çok, büyümeyi sağlamak için ve sürdürmek için zorunlu ana akım satış kanalı haline gelmiştir. Turizm sektöründe metasearch uygulamaları üreten Trivago firmasının 2016 yılında yaptığı araştırmaya göre Amerika ve Avrupa'da turizm sektöründeki satışlardaki payı sırasıyla %79 ve %69 a çıkmıştır. Yine aynı çalışmanın 2015 için verdiği değerler Amerika ve Avrupa için sırasıyla %59 ve %79 dur. 2015 yılı rakamlarında sağlanan bu artış göz önüne alındığında online satış kanallarının önemi açıktır (Domo 2019).

Diğer yandan, teknolojinin ve verinin önemi, diğer tüm sektörler için kritik önem arz etmesinden dolayı pek çok sektörde geçmişteki satın alma hareketlerinin incelendiği görülmektedir. T. K Das bildirisinde naive Bayes, k en yakın komşu, destek vektör makinesi algoritmalarını kullanarak hangi müşteri profilinin hangi ürünleri satın almaya daha yatkın olduğunu tahminleme üzerinde çalışmıştır (Das 2015).

Özer, veri analitik yöntemlerini müzik sektöründe uygulamış ve bulanık küme analizi ile online müzik servislerini kullanan kişiler arasında homojen grupların bulunduğu tespit ederek bu grupların farklı stratejiler ile hedeflenmesini önermiştir (Özer 2001).

Turizm sektöründe gerçekleştirilen Hedef kitle analizi çalışmasında, Kegeci ve arkadaşları, uygun ürünü uygun kitle ile buluşturmak için 2 sınıflı bir sınıflandırma tekniği kullanmışlardır (Kegeci, Özbek ve Türkel, 2018).

Yılmaz, kişilerin genel sağlık davranışları, genetik ve çevre faktörlerini girdi olarak kullanan ve akciğer, kolon ve meme kanser türleri için ön tanı ve risk değerlemelerini makine öğrenmesi yöntemlerinden sinirsel bulanık mantık algoritması ile hesaplayıp, bunun etkin bir ön-tanı yöntemi olarak kullanılabileceğini önermiştir (Yılmaz 2015).

Kira, Kenji, ve Larry A. Rendell makine öğrenmesi algoritmalarında öznitelik seçiminin önemini ve öznitelik seçiminde kullanılan yöntemleri detaylandırmışlardır (Kira, Kenji ve Rendell 1992).

Dolmicar, makalesinde turizm sektöründe yapılan kümeleme ve müşteri segmentasyon çalışmalarını ve bu çalışmaların başarısında verilerin miktar ve niteliklerinin önemini vurgulamıştır (Dolnicar 2002, 22).

Caruana, Rich ve Mizil, makalelerinde çeşitli gözetimli öğrenme algoritmalarını karşılaştırma yöntemleriyle ilgili alternatifleri değerlendirmişlerdir (Caruana, Rich ve Mizil, 2006. 161-168.).

Osbourne ve Waters, Çoklu regresyon testinde gözönünde bulundurulması gereken, verilerin normal dağılım göstermesi, bağımsız değişenlerin linear ilişki içermeleri ve verilerin sağlıklı toplanmış ve gerçek veriyi iyi özetleyen kümelerden oluşması gerektiğiyle ilgili bildiri yayınlamışlardır (Osborne ve Waters 2002, 1-9).

Williams, Grajales ve Kurkiewics, çoklu regresyon analizlerinde çoklu doğrudaşlık (multicollinearity) ve ayrıkı değerlerin analiz sonuçlarına etkilerini değerlendirmiştir (Williams, Matt, Grajales ve Kurkiewicz 2013, 11).

Bayrak ve arkadaşları, veri tekilleştirme çalışmalarıyla veri setinde bulunan ve birbirinden farklı görünen kayıtlarda yazım ve veri giriş hataları sebebiyle oluşabilecek çoklamayı gidermek için yöntemler geliştirmiştir (Bayrak, Yılmaz, Düzağaç ve Yıldız 2018, 4).

Krstajic ve arkadaşları regresyon ve sınıflandırma modellerinde çapraz doğrulama yöntemlerinin uygulanmasının önemini ortaya koyan bir makaleyle, eğitim ve test küme seçiminin önemini vurgulamışlardır (Krstajic, Damjan, Buturovic, Leahy ve Thomas 2014, 1-15).

# SONUÇ

Yapılan çalışmada Türkiye'de online olarak faaliyet gösteren bir firmanın geçmiş satış verileri kullanılmıştır. Öncelikle özniteliklerin dağılımı ve boş veri oranları kontrol edilmiştir. Rassal ormanlar yöntemiyle öznitelik önem dereceleri tespit edilmiş ve veriyi yeteri kadar nitelemediği tespit edilen öznitelikler analiz kapsamı dışına alınmıştır. Kategorik öznitelikler one-hot encode edilerek sayısal niteliklere dönüştürülmüştür. Sayısal değerler ise feature scaling yöntemlerinden min max normalizasyonu yöntemi ile aynı 0 ile 1 arasında olan ortak skalaya alınmıştır. Aykırı değerlere sahip kayıtlar veri setinden hariç tutulmuş ve hedef kitle kitleyi daha iyi temsil ettiği ve yapılan filtreleme sonucunda sınıflandırma başarısının arttığı görülmüştür.

Kullanılan algoritmanın hiper parametreleri, grid search metoduyla optimize edilmiştir. Tatil köyü alma ihtimali sınıflandırma probleminde ROC eğrisi kullanılarak eşik, doğru pozitiflerin yanlış pozitif değerlere oranı 0,8 oranıyla optimize edilmiştir.

Çalışma sonucunda, modelimizin tanımadığı test kümesi üzerinde yapılan tahminleme, bir sonraki alışverişlerinde tatil köyü hizmeti satın alma ihtimali yüksek olarak tespit edilen müşterilerin %93 isabet (precision) ile ve %82 hassasiyet (recall) ile tahminlendiği görülmüştür.

Hem kayıp müşteri analizinde, hem de tatil köyü satın alma ihtimalini tahminlemede,

Test kümesi ile yapılan doğrulama testi sonucunda, rassal ormanlar yöntemi kesinlik (precision) ve hassasiyet (recall) açısından daha doğru sonuçlar verdiği görülmüştür. Rassal ormanlar ve Yapay Sinir Ağları algoritmalarının çıktılarının, test kümesindeki başarı oranları Tablo 11 ve Tablo 12'de verilmiştir.

**TABLO 11: Tatil Köyü Satın Alma, Tahmin Başarı Oranları**

| | Rassal Ormanlar | | | Yapay Sinir Ağları | | |
|---|---|---|---|---|---|---|
| | Tatil Köyü Alır | Tatil Köyü Almaz | Doğruluk | Tatil Köyü Alır | Tatil Köyü Almaz | Doğruluk |
| Kesinlik | 0.94 | 0.93 | 0.94 | 0.93 | 0.93 | 0.96 |
| Hassasiyet | 0.98 | 0.82 | 0.94 | 0.98 | 0.84 | 0.96 |
| F1-Scoru | 0.96 | 0.87 | 0.94 | 0.95 | 0.88 | 0.96 |

**TABLO 12: Kayıp Müşteri Analizi, Tahmin Başarı Oranları**

| | Rassal Ormanlar | | | Yapay Sinir Ağları | | |
|---|---|---|---|---|---|---|
| | Kayıp Müşteri | Kayıp Müşteri Değil | Doğruluk | Kayıp Müşteri | Kayıp Müşteri Değil | Doğruluk |
| Kesinlik | 0.97 | 0.93 | 0.96 | 0.82 | 0.17 | 0.79 |
| Hassasiyet | 0.99 | 0.84 | 0.96 | 0.94 | 0.06 | 0.79 |
| F1-Scoru | 0.98 | 0.88 | 0.96 | 0.88 | 0.08 | 0.79 |

Firmanın tatil köyü hizmeti pazarlama faaliyetlerinde rassal ormanlar yöntemi inle yapılan tahminleme yöntemini kullanılarak yapılan analiz sonuçlarına odaklanmasının, pazarlama faaliyetlerinin maddi getirisini artırmada faydası olacağı düşünülmektedir.

## KAYNAKÇA


Bayrak, Yılmaz, Düzağaç, ve Yıldız. "Near duplicate detection in relational databases." In 2018 26th Signal Processing and Communications Applications Conference (SIU), pp. 1-4. IEEE, 2018.

Caruana, Rich, ve Mizil. "An empirical comparison of supervised learning algorithms." In Proceedings of the 23rd international conference on Machine learning, pp. 161-168. 2006.

Das, T. K. "A customer classification prediction model based on machine learning techniques." In 2015 International Conference on Applied and Theoretical Computing and Communication Technology (iCATccT), pp. 321-326. IEEE, 2015.

Dolnicar, Sara. "A review of data-driven market segmentation in tourism." *Journal of Travel & Tourism Marketing* 12, no. 1 (2002): 1-22.

Domo 2017 Data Never Sleeps raporu: https://web-assets.domo.com/blog/wp-content/uploads/2017/07/17_domo_data-never-sleeps-5-01.png , Erişim tarihi: 01.10.2019

Friedman, Jerome, Hastie, and Robert Tibshirani. The elements of statistical learning. Vol. 1, no. 10. New York: Springer series in statistics, 2001.

Géron, Aurélien. Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems. O'Reilly Media, 2019, pp. 132-138

Guillaume, Martin, https://www.kaggle.com/gemartin/load-data-reduce-memory-usage . Erişim tarihi: 18.11.2019

https://en.wikipedia.org/wiki/Overfitting, Aşırı Uyma, Erişim Tarihi: 16.01.2019

Kegeci, Özbek, Türkel, Düzağaç ve Yıldız. "Doğrudan pazarlama amaçlı hedef kitle analizi." (2018).

Kira, Kenji, ve Larry A. Rendell. "A practical approach to feature selection." In Machine Learning Proceedings 1992, pp. 249-256. Morgan Kaufmann, 1992.

Küçükbatır, Merve. "The adoption and influences of big data in tourism industry in Turkey." Kadir Has Üniversitesi, Sosyal Bilimler Enstitüsü, Yüksek Lisans Tezi, İstanbul (2018).

Krstajic, Damjan, Ljubomir J. Buturovic, David E. Leahy, and Simon Thomas. "Cross-validation pitfalls when selecting and assessing regression and classification models." Journal of cheminformatics 6, no. 1 (2014): 1-15.

Lee, Sangjae, and Joon Yeon Choeh. "Predicting the helpfulness of online reviews using multilayer perceptron neural networks." Expert Systems with Applications 41.6 (2014): 3041-3046.

McKinney, Wes. Python for data analysis: Data wrangling with Pandas, NumPy, and IPython. " O'Reilly Media, Inc.", 2012.

Nicholson, Chris, A Beginner's Guide to Multilayer Perceptrons (MLP) , https://pathmind.com/wiki/multilayer-perceptron, Erişim tarihi: 09.01.2020

Osborne, J., & Waters, E. (2002). Four assumptions of multiple regression that researchers should always test. *Practical assessment, research & evaluation*, *8*(2), 1-9.

Özer, Muammer. "User segmentation of online music services using fuzzy clustering." Omega 29, no. 2 (2001): 193-206.

Şeker, Şadi Evren, Information Gain (Bilgi Kazanımı), http://bilgisayarkavramlari.sadievrenseker.com/2012/11/13/information-gain-bilgi-kazanimi/, Erişim tarihi: 12.12.2019

Şeker, Şadi Evren, F1 Değerlendirme (F1-Scoring), http://bilgisayarkavramlari.sadievrenseker.com/2010/09/30/f1-degerlendirme-f1-scoring/, Erişim tarihi: 16.01.2019

Trivago, *"A Beginners Guide To Digital Hotel Distribution and Marketing",* https://businessblog.trivago.com/wp-content/uploads/2019/11/eBook_A-Beginner's-Guide-to-Digital-Hotel-Distribution-and-Marketing.pdf , Erişim tarihi 28.12.2019

Yılmaz, A. (2015). Sinirsel Bulanık Mantık Modeliyle Kanser Risk Analizi (Doctoral dissertation, Doktora Tezi, Sakarya Üniv., Fen Bilimleri Enstitüsü).

VanderPlas, Jake, Pydata : https://speakerdeck.com/jakevdp/intro-to-pydata?slide=58 , Erişim tarihi: 01.11.2019

Williams, Matt N., Carlos Alberto Gómez Grajales, and Dason Kurkiewicz. "Assumptions of multiple regression: Correcting two misconceptions." Practical Assessment, Research, and Evaluation 18, no. 1 (2013): 11.

# ÖZGEÇMİŞ

1984 İstanbul doğumlu Fikri Murat Simav Işık Üniversitesi Enformasyon Teknolojileri bölümünden 2010 yılında mezun oluştur. Kariyer hayatına Junior Yazılım Geliştirme uzmanı olarak İ-Lab Holding iştiraki Treda Bilişim Teknolojileri firmasında başlamış ve Holding iştiraki Sigortam.Net firmasının yöntetim panelleri üzerinde çalışmıştır.

Sonrasında analist programcı olarak kurumsal bankacılık yazılımları geliştiren Proje Sinerji firmasında görev alarak Türk Hava Yolları, Abdi İbrahim, Adidas, Peagaut gibi firmaların ödeme sistemleri entegrasyonları üzerinde çalışmıştır.

İnsan Kaynakları yazılımları geliştiren ve danışmanlığını veren Bilin Yazılım firmasında öncelikle Odea Bank, Kazancı Holding, Arçelik, Ford Otosan gibi firmalarla uygulama danışmanı olarak çalışmış ve son olarak da kurumda Eczacıbaşı Holding'in tüm İnsan Kaynakları süreçlerinin yazılım, uyarlama ve entegrasyon süreçlerinde teknik analist ve proje yöneticisi olarak çalışmıştır.

2018 Aralık döneminden itibaren de, Türkiye'nin lider bir Turizm firmasında Proje Yönetim Ofisi, Kurumsal Uygulamalar Bölümünde Kıdemli Proje Yöneticisi olarak çalışmaktadır.

**Fikri Murat SİMAV**