
GAZİ ÜNİVERSİTESİ
Gazi
1926

MAKİNE ÖĞRENMESİ METHODLARI ile MÜŞTERİ MEMNUNİYETİNİ ANLAMA

İsmail GÜRLER

DÖNEM PROJESİ
YÖNETİM BİLİŞİM SİSTEMLERİ ANABİLİM DALI

GAZİ ÜNİVERSİTESİ
BİLİŞİM ENSTİTÜSÜ

ARALIK 2022

ETİK BEYAN

Gazi Üniversitesi Fen Bilimleri Enstitüsü Tez Yazım Kurallarına uygun olarak hazırladığım bu tez çalışmasında;

- Tez içinde sunduğum verileri, bilgileri ve dokümanları akademik ve etik kurallar çerçevesinde elde ettiğimi,
- Tüm bilgi, belge, değerlendirme ve sonuçları bilimsel etik ve ahlak kurallarına uygun olarak sunduğumu,
- Tez çalışmasında yararlandığım eserlerin tümüne uygun atıfta bulunarak kaynak gösterdiğimi,
- Kullanılan verilerde herhangi bir değişiklik yapmadığımı,
- Bu tezde sunduğum çalışmanın özgün olduğunu,

bildirir, aksi bir durumda aleyhime doğabilecek tüm hak kayıplarını kabullendiğimi beyan ederim.

....……

İsmail Gürler

22.12.2022

DÖNEM PROJESİ ONAY SAYFASI

İsmail Gürler tarafından hazırlanan "MAKİNE ÖĞRENMESİ METHODLARI ile MÜŞTERİ MEMNUNİYETİNİ ANLAMA" başlıklı Dönem Projesi tarafımdan Yönetim Bilişim Sistemleri Anabilim Dalında Dönem Projesi olarak kabul edilmiştir.

Danışman: Prof. Dr. Cevriye GENCER
Yönetim bilişim sistemleri anabilim dalı, Gazi Üniversitesi
Bu çalışmanın, kapsam ve kalite olarak Dönem Projesi olduğunu onaylıyorum.

.....……

Dönem Projesi Teslim Tarihi: ...../……/…… Danışmanı tarafından kabul edilen bu çalışmanın Dönem Projesi olması için gerekli şartları yerine getirdiğini onaylıyorum.

…….……

.....

Bilişim Enstitüsü Müdürü

MAKİNE ÖĞRENMESİ METHODLARI ile MÜŞTERİ MEMNUNİYETİNİ ANLAMA
(Dönem Projesi)

İsmail GÜRLER

GAZİ ÜNİVERSİTESİ

BİLİŞİM ENSTİTÜSÜ

Aralık 2022

ÖZET

Halihazırda kullanılan çok fazla makine öğrenimi algoritmaları mevcuttur. Bu makine öğrenimi algoritmalarının hangi veri setlerinde ve veri tiplerinde en doğru şekilde kullanılacağı ya da model oluşturacağı uzmanlık gerektiren zor bir işlemdir. Bu algoritmaları anlayabilmek ve çalışma mantıklarını kavrayabilmek için önce makine öğrenimi alt sınıflarına ayrılarak aktarılmıştır. Bu çalışmada, gerçek bir veri seti olan havayolu şirketinin anket verileri kullanılarak müşteri memnuniyetini en iyi oranda tahmin edebilmesini sağlamak amacıyla 5 farklı makine öğrenimi algoritması modellenmiştir. Bir kod yapısı ile en iyi tahmin modelini oluşturan algoritma bulunmuş ve sonuçları görselleştirilerek aktarılmıştır. Daha sonra en iyi sonucu veren tahmin modelinin sonuçları ile gerçek veri sonuçları kıyaslanmıştır. Ayrıca model f1 skoru, sınıflandırma raporu ve hata matrisi ile incelenmiştir. Modeli eğitmeden önce veri setinin ön işleme ile hazırlığı yapılmış; eksik verilerin tamamlanması ya da özniteliklerin korelasyonu incelenerek düzeltilmiştir. Özniteliklerin anlaşılması için de görselleştirmeler yapılmıştır. Aynı veri tipiyle çalışsa bile farklı sonuçlar veren makine öğrenimi algoritmalarının olduğunu ve bu algoritmalar arasında da yazılım yoluyla en iyi sonucu verene ulaşabileceği sonucuna ulaşılmıştır.

Bilim Kodu : 114611
Anahtar Kelimeler : Makine öğrenimi algoritmaları, veri bilimi kod yapısı, havayolu müşteri memnuniyet anketi
Sayfa Adedi : 31
Danışman : Prof. Dr. Cevriye GENCER

UNDERSTANDING CUSTOMER SATISFACTION
with MACHINE LEARNING METHODS
(Term Project)

İsmail GÜRLER

GAZİ UNIVERSITY

INSTITUTE OF INFORMATICS

December 2022

ABSTRACT

There are many machine learning algorithms currently in use. The most accurate use or modeling of these machine learning algorithms on which data sets and data types is a difficult process that requires expertise. In order to understand these algorithms and their working logic, machine learning is first subclassified. In this study, a real data set, the survey results of an airline company, is used. Using this dataset, 5 different machine learning algorithms are modeled and the aim is to ensure that the model can predict customer satisfaction at the best rate. With a code structure, the algorithm that creates the best prediction model is found and the results are visualized and presented. Then, the results of the best prediction model are compared with real data. The model is also analyzed with f1 score, classification report and error matrix. Before training the model, the dataset was preprocessed. Missing data was corrected by completing or examining the correlation of the attributes. Visualizations were also made to understand the attributes. It was concluded that there are machine learning algorithms that give different results even if they work with the same data type, and that among these algorithms, the one that gives the best result can be reached through software.

Science Code : 114611
Keywords : Machine learning algorithms, data science code structure, airline customer satisfaction survey
Number of Pages : 31
Advisor : Prof. Dr. Cevriye GENCER

İÇİNDEKİLER

Sayfa

ETİK BEYAN i
DÖNEM PROJESİ ONAY SAYFASI ii
ÖZET iii
ABSTRACT iv
İÇİNDEKİLER v
ŞEKİLLERİN LİSTESİ vi
KISALTMALAR VE TANIMLARI vii
1. GİRİŞ 1
2. MAKİNE ÖĞRENİMİ VE ALGORİTMALAR 2
2.1 Denetimli Öğrenme (Supervised Learning) 3
2.1.1 Karar ağacı (Decision tree (DT)) 3
2.1.2 Lojistik regresyon 4
2.1.3 LDA (Linear discriminant analysis): Doğrusal ayrımcılık analizi 5
2.1.4 NB (Gaussian NB): Naive baise 5
2.1.5 KNN (K nearest neighbour): K en yakın komşu algoritması 6
2.2 Denetimsiz Öğrenme (Unsupervised Learning) 6
2.3 Pekiştirmeli Öğrenme (Reinforcement Learning) 7
3. MÜŞTERİ MEMNUNİYETİ UYGULAMASI 8
3.1 Problemin Tanımlanması 8
3.2 Verinin Hazırlanması 8
3.2.1 Kütüphanelerin Çağrılması 8
3.2.2 Veri setinin yüklenmesi ve özetlenmesi 10
3.2.3 Veri setinin görselleştirilmesi 16
3.3 Modellerin Oluşturulması 24
3.4 En İyi Modelin Değerlendirilmesi 26
4. SONUÇ 28
KAYNAKLAR 29

ŞEKİLLERİN LİSTESİ

Şekil **Sayfa**

Şekil 2.1 -Makine Öğrenmesi Algoritmalarının Sınıflandırılması....2
Şekil 2.2 - Karar Ağacı Modelinin Temsili Diyagramı....3
Şekil 2.3 - Binary Lojistik Regresyon Modelinin Örneği....4
Şekil 2.4 - Çoklu Lojistik Regresyon Modelinin Örneği....4
Şekil 2.5 - K En Yakın Komşu Algoritması Örnek Grafiği....6
Şekil 2.6 - Pekiştirmeli Öğrenme Algoritmasının Basit Gösterimi....7
Şekil 3.1 - Veri Seti için Kütüphanelerin Çağrılması....8
Şekil 3.2 - Veri Görselleştirme için Kütüphanelerin Çağrılması....9
Şekil 3.3 - Modellerin Kurulması için Kütüphanelerin Çağrılması....9
Şekil 3.4 - Modelleri Değerlendirmek için Kütüphanelerin Çağrılması....9
Şekil 3.5 - Veri Setinin Yüklenmesi ve Çağrılması....10
Şekil 3.6 - Sütunların Çağrılması....10
Şekil 3.7 – Özniteliklerin Detaylı Bilgisi....12
Şekil 3.8 - Eksik Verilerin Kontrol Edilmesi....12
Şekil 3.9 - Eksik Verinin Görselleştirilmesi....13
Şekil 3.10 - Korelasyon Isı Haritası....14
Şekil 3.11 - Sayısal Faktörlerin Korelasyon Tablosu....15
Şekil 3.12 - Memnuniyet Sonuç Sayımı....16
Şekil 3.13 - Memnuniyet Oranının Pasta Grafiği....16
Şekil 3.14 - Memnuniyet Oranının Dağılımı....17
Şekil 3.15 - Kategorik Veriyi Görselleştirme İşlemleri....17
Şekil 3.16- Cinsiyet ve Müşteri Türüne Göre Memnuniyet Dağılımı....18
Şekil 3.17 - Sınıf ve Seyahat Türüne Göre Memnuniyet Dağılımı....18
Şekil 3.18 - Online Rezervasyon ve Kapı Lokasyonuna Göre Memnuniyet Dağılımı....19
Şekil 3.19 - Wifi Hizmeti ve Zamanlama Doğruluğuna Göre Memnuniyet Dağılımı....19
Şekil 3.20 - Yiyecek Servisi ve Online Check in Türüne Göre Memnuniyet Dağılımı....20
Şekil 3.21 - Koltuk Komforu ve Uçakiçi Eğlenceye Göre Memnuniyet Dağılımı....20
Şekil 3.22 - Biniş Servisi ve Koltuk Mesafesine Göre Memnuniyet Dağılımı....21
Şekil 3.23 - Bagaj Servisi ve Checkin Servisine Göre Memnuniyet Dağılımı....21
Şekil 3.24 - Temizliğe Göre ve Genel Memnuniyet Dağılımı....22
Şekil 3.25 - Sayısal Sütunların Görselleştirilmesi için Yazılan Kod Dizisi....22
Şekil 3.26 - Yaşa Göre Memnuniyet Dağılımı....23
Şekil 3.27 - Geç Kalkış Durumuna Göre Memnuniyet Dağılımı....23
Şekil 3.28 - Uçuş Mesafesine Göre Memnuniyet Dağılımı....23
Şekil 3.29 - Veri Setinin Eğitim ve Validasyon Verisi Olarak Ayrılması....24
Şekil 3.30 - Modellerin Listelenmesi....24
Şekil 3.31 - Modellerin Fonksiyon İçinde Çalıştırılması ve Sonuçları....25
Şekil 3.32 - Algoritma Doğruluk Oranlarının Kutu Diyagramında Kıyaslanması....25
Şekil 3.33 - Tahminleme Kod Yapısı....26
Şekil 3.34 - Karar Ağacı Modelinin Çeşitli Fonksiyonlarla Değerlendirilmesi....26
Şekil 3.35 - Hata Matrisi Diyagramı....27

KISALTMALAR VE TANIMLARI

Kısaltmalar	**Açıklama**
a. ***ML*** *(Machine Learning)*	Makine öğrenmesi
b. ***KNN*** *(K nearest Neighbour)*	K En Yakın Komşu Algoritması
c. ***LDA*** *(Linear Discriminant Analysis)*	Doğrusal Ayrımcılık Analizi Algoritması
d. ***SVM*** *(Support Vector Machine)*	Karar Destek Makinesi Algoritması
e. ***PCA*** *(Principal Component Analysis)*	Temel Bileşen Analizi
f. ***DT*** *(Decision Trees)*	Karar Ağaçları Algoritması
g. ***CART*** *(DecisionTreeClassifier)*	Karar Ağacı Sınıflandırıcısı
h. ***LR*** *(Logistic Regression)*	Lojistik Regresyon
i. ***NB*** *(Gaussian NB)*	Naive Baise Algoritması
j. **CSV**	Virgülle Ayrılmış Değerler Dosyası

1. GİRİŞ

Makine öğrenimi algoritmaları, geçmiş deneyimlerden öğrenmek ve büyük, yapılandırılmamış ve karmaşık veri kümelerinden faydalı çıkarımlar yapmak ve veri desenlerini tespit etmek için çeşitli istatistiksel, matematiksel ve optimizasyon yöntemleri kullanır. Bu algoritmalar, otomatik metin kategorizasyonu ve çevirisi, ağ saldırı tespiti, spam e-posta tespiti, kredi kartı dolandırıcılığının tespiti, müşteri davranışlarının tespiti ve duygu analizi, üretim sürecinin optimizasyonu ve hastalık teşhisi gibi geniş bir uygulama alanına sahiptir. Bu uygulamaların çoğu, denetimsiz (unsupervised) öğrenme kullananlar hariç makine öğrenimi algoritmalarının denetimli (supervised) varyantları kullanılarak gerçekleştirilmiştir. Denetimli öğrenme varyantlarında, etiketin belli olduğu bir veri kümesi öğrenilerek bir tahmin modeli geliştirilir ve buna göre etiketsiz örneklerin sonucu tahmin edilebilir. [1]

Bu çalışmada, makine öğrenimi algoritmaları kullanılarak veri bilimi tahmin çalışması yapılması hedeflenmiştir. Online bir veri kaynağından python arayüzü olan jupyter lab programı ile havayolları anket verileri çekilip veri kümesi oluşturulmuş; bu ham verinin analizinin yapılıp kullanılabilmesi için veri önişleme yapılmış ve sonrasında denetimli (supervised) makine öğrenmesi algoritmaları ile model oluşturup veriler eğitilmiştir. Veriler eğitip model oluşturulurken Lojistik regresyon (logistic regression), karar ağaçları (decision tree), K en yakın komşu (K nearest Neighbour), gaussian naive bayes, ve doğrusal ayrımcılık analizi (Linear Discriminant Analysis) algoritmaları kullanılmıştır.

Tüm kod çalışmaları python ara yüzü olarak kullanılabilen Jupyter Lab üzerinde çalışılmıştır.

Çalışmanın 2. Bölümünde makine öğrenimi algoritmaları anlatılmış; 3. Bölümde uygulama çalışması yapılmış ve son bölümde sonuçlar yer almaktadır.

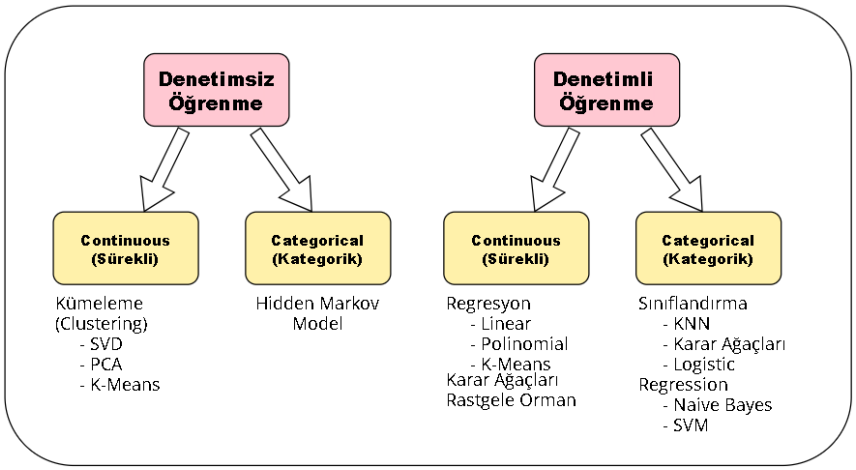
2. MAKİNE ÖĞRENİMİ VE ALGORİTMALAR

Geçmişten bu yana insanlar, çeşitli işleri daha basit bir şekilde yerine getirmek için birçok araç icat edip kullanmaktadırlar. İnsanın yaratıcılığı farklı makinelerin icat edilmesine olanak sağlamıştır. Bu makineler, insanların eğlence, eğitim, ulaşım, endüstri ve bilgi işlem gibi çeşitli yaşam ihtiyaçlarını karşılayarak insan hayatını kolaylaştırmıştır. Makine öğrenimi de bun icatların arasında yer almaktadır.

Makine öğrenimi (ML), bilgisayar sistemlerinin belirli bir görevi yerine getirmek için kullandıkları algoritmaların ve istatistiksel modellerin programlandığı bilimsel çalışmasıdır. Batta Mahesh'in belirttiği gibi makine öğrenimi, bilgisayarlara açıkça programlanmadan öğrenme yeteneği kazandıran çalışma alanı olarak tanımlanmaktadır. [2] Makine öğrenimi algoritmaları günlük hayatta kullandığımız birçok uygulamada kendine yer edinmiştir. Google gibi bir web arama motoru internette arama yapmak için her kullanıldığında, bu kadar iyi çalışmasının nedenlerinden biri, web sayfalarını nasıl sıralayacağını öğrenen bir öğrenme algoritması olmasıdır. Bu algoritmalar veri madenciliği, duygu analizi, görüntü işleme, tahmine dayalı analitik vb. gibi çeşitli amaçlar için kullanılmaktadır. Makine öğrenimini kullanmanın temel avantajı, bir algoritmanın verilerle eğitildikten sonra işini kendi yapabilmesidir.

Makine öğrenimi, makinelere verileri nasıl daha verimli kullanacaklarını öğretmek için kullanılır. Halihazırda çok fazla veri üretmemiz ile makine öğreniminin kullanım alanı artmaktadır. Birçok endüstri, ilgili verileri çıkarmak, analiz etmek ve kullanmak için makine öğrenimini uygulamaktadır. Makinelerin açıkça programlanmadan kendi kendilerine öğrenmelerini sağlamak için birçok çalışma yapılmıştır ve daha da fazla yapılacağı aşikardır. Birçok matematikçi ve programcı, büyük veri setlerine sahip olan problemlerin en doğru ve sürekli çözümünü bulmak için çeşitli makine öğrenimi yaklaşımları uygulamaktadır.

Makine Öğrenimi, veri problemlerini çözmek ve çözerken en iyi sonucu almak için farklı algoritmaları kullanmayı ya da en iyi algoritmayı bulmayı gerektirir. Kullanılan algoritma türü, çözmek istenilen sorunun türüne, değişken sayısına, veri tipine ve buna en uygun model türüne bağlıdır. Şekil 2.1'de değişken türlerine göre kullanılabilecek modeller aktarılmıştır. Genel olarak, 3 ana Makine Öğrenimi Algoritması türü vardır.



Şekil 2.1 -Makine Öğrenmesi Algoritmalarının Sınıflandırılması

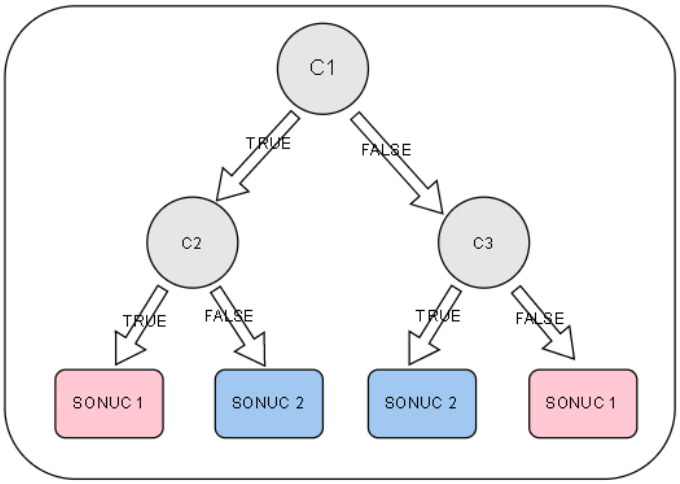
2.1 Denetimli Öğrenme (Supervised Learning)

Denetimli öğrenme, örnek girdi-çıktı değerlerine dayalı olarak girdiyi çıktıya eşleyen bir işlevi öğrenip model oluşturmaya yönelik makine öğrenimi görevidir. Etiketli bir dizi eğitim verisinden bir metot oluşturur. Denetimli makine öğrenimi algoritmaları, harici yardıma ihtiyaç duyan algoritmalardır. Sahip olunan veri kümesi, eğitim ve test veri kümelerine ayrılır. Eğitim veri kümesi, tahmin edilmesi veya sınıflandırılması gereken girdi ve çıktı değişkenine sahiptir. Tüm algoritmalar eğitim veri kümesinden öğrenerek bir model oluşturur ve modelden ürettiği çıktılarını test veya doğrulama için test veri kümesine uygular. [2] Denetimli öğrenme algoritmaları iki tür problem için kullanılmaktadır: sınıflandırma veya regresyon problemleri.

Eğitim süreci, model eğitim verileri üzerinde istenen doğruluk seviyesine ulaşana kadar devam eder. Denetimli Öğrenme Algoritmaları: Lojistik Regresyon, Karar Ağacı, Rastgele Orman, KNN, Lojistik Regresyon vb.

2.1.1 Karar ağacı (Decision tree (DT))

Karar ağacı, ilk kullanılan makine öğrenimi algoritmalarından biridir. Bir karar ağacı, veri kümelerini ağaç benzeri bir yapıda sınıflandırmak için karar adımlarını, yani testleri ve karşılık gelen sonuçları modeller. Bir karar ağacının düğümleri, ilk veya en üst düğüm olan kök düğümle bu kök düğümünün bir veya daha fazla seviyede dallanarak ayrıldığı yavru düğümlerden oluşmaktadır. Tüm dahili düğümler (yani en az bir çocuğu olan düğümler) girdi değişkenleri veya nitelikler üzerindeki karar testlerini temsil eder. Test sonucuna bağlı olarak, sınıflandırma algoritması uygun alt düğüme doğru dallanır ve burada test ve dallanma süreci yaprak düğüme ulaşana kadar tekrar eder. Yaprak düğümler karar sonuçlarına karşılık gelir. DT'lerin yorumlanması kolay ve öğrenilmesi hızlı bulunduğu için birçok sınıflandırma alanlarında kullanılmıştır ve tıbbi tanı protokolünde kullanımı da buna bir örnektir. Bir problemin sınıflandırılması için ağaç boyunca her düğümdeki tüm karar testlerinin sonuçları, sınıf hakkında varsayımda bulunmak için yeterli bilgi sağlayacaktır. Bir DT'nin öğeleri ile gösterimi Şekil 2.2'de gösterilmektedir [1].

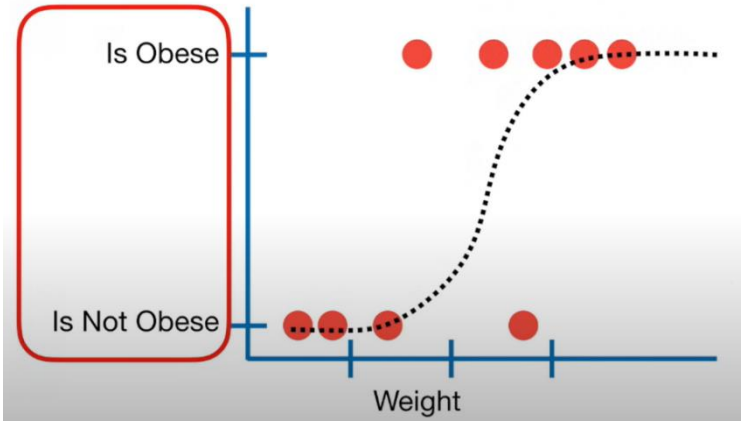


Şekil 2.2 - Karar Ağacı Modelinin Temsili Diyagramı

2.1.2 Lojistik regresyon

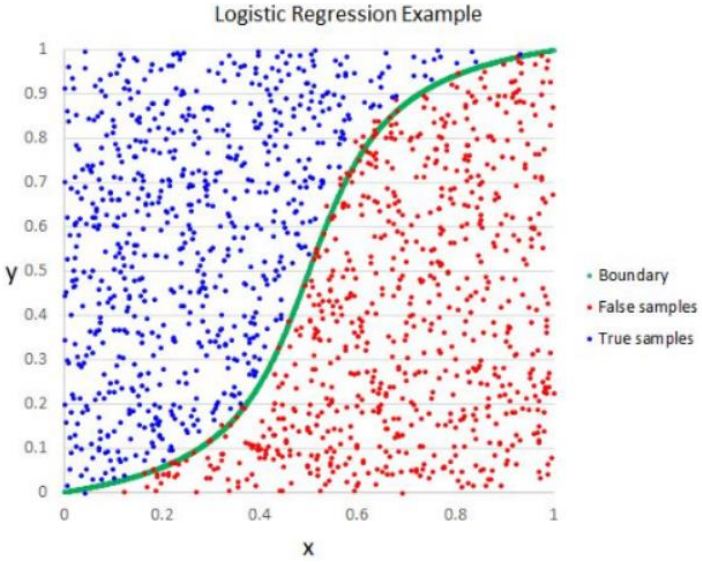
Lojistik regresyon algoritması bir regresyon değil bir sınıflandırmadır. Belirli bir bağımsız değişken(ler) kümesine dayalı olarak ayrık değerleri (0/1, evet/hayır, doğru/yanlış gibi ikili değerler) tahmin etmek için kullanılır. Olasılığı tahmin ettiğinden, çıktı değeri genellikle 0 ile 1 arasında yer alır.

Lojistik regresyon, sınıflar arasındaki sınırların belirlenmesini sağlar. Veri seti daha büyük olduğunda uç noktalara yani doğru karara daha hızlı ulaşır. Lojistik regresyonu bir sınıflandırıcıdan daha fazlası yapan olasılık değerleriyle çalışmasıdır. Lojistik regresyon, uygulamalı istatistik ve kesikli veri analizi için en yaygın kullanılan araçlardan biridir. Lojistik regresyon doğrusal enterpolasyondur. Şekil 2.3'te bir binary (ikili) lojistik regresyon sonuç grafiği örneği görülmektedir ve Şekil 2.4'te çoklu bir lojistik regresyon örneği bulunmaktadır [4].



Şekil 2.3 - Binary Regresyon Lojistik Modelinin Örneği

Şekil 2.3 incelendiğinde kişinin kilosuna göre obez olup olmadığını kilo bağıntısı ile bulunabilmekte olduğu görülmektedir.



Şekil 2.4 - Çoklu Lojistik Regresyon Modelinin Örneği

2.1.3 LDA (Linear discriminant analysis): Doğrusal ayrımcılık analizi

"*Doğrusal diskriminant analizi, özniteliklerin bir doğrusal birleşimini bularak veriyi sınıflara ayırmaya yarayan yöntemdir. LDA, makine öğrenimi sınıflandırma uygulamaları için ön işlemede boyutluluğu azaltmaya yönelik yaygın bir tekniktir. LDA, sınıflar arası varyansın sınıf içi varyansa oranını en aza indirerek özellikleri daha düşük boyutlu bir uzaya dönüştürmek ve böylece maksimum sınıf ayrılabilirliğini garanti etmek için geliştirilmiştir. LDA'nın temel amacı, bir uzayı (N boyutlu veri), sınıf ayrımını korurken daha küçük bir K alt uzayına (K ≤ n- 1) yansıtmaktır."* [5].

Birden fazla özelliğe sahip iki veya daha fazla sınıfı verimli bir şekilde kümeleme ihtiyacı olduğunda, bu algoritma türü sınıflandırma problemlerini çözmek için en yaygın tekniklerden biri kabul edilir. Örneğin, birden fazla özelliğe sahip iki sınıf varsa ve bunların verimli bir şekilde kümelenmesi gerekiyorsa LDA etkin bir şekilde kullanılabilir [6].

2.1.4 NB (Gaussian NB): Naive baise

Naïve Bayes, Bayes'in Olasılık Teoremini kullanarak sınıflandırma problemlerini çözmek için kullanılır. Bayes teoremi, P(A/B) ile temsil edilen B olayının önceki olasılıklarını göz önüne alarak bir olayın (A) olasılığını aşağıdaki gibi hesaplar:

P(A/B) = $P(B/A) * P(A) / P(B)$

- ✓ A ve B olaylardır.
- ✓ P(A) ve P(B), birbirinden bağımsız olarak A ve B'yi gözlemleme olasılıklarıdır.
- ✓ P(A/B) koşullu olasılıktır, yani B'nin doğru olması durumunda A'yı gözlemleme olasılığıdır.
- ✓ P(B/A), A'nın doğru olması durumunda B'yi gözlemleme olasılığıdır.

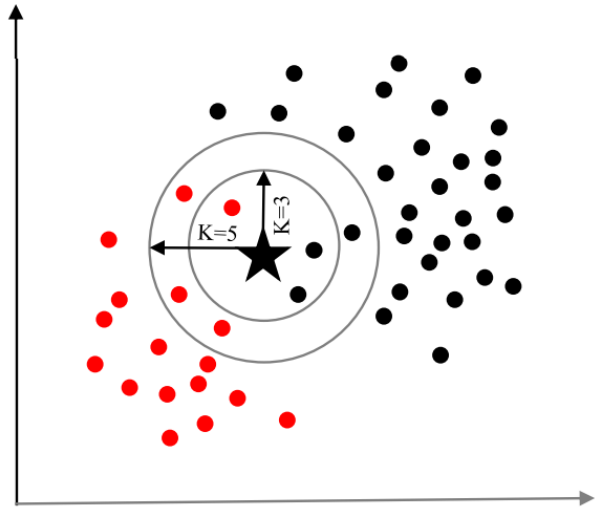
Naïve Bayes sınıflandırıcıları, özellikler arasında güçlü bağımsızlık varsayımlarına sahip Bayes Teoremi kavramına dayanan basit olasılıksal sınıflandırıcılar olarak kullanılırlar. Yani değişkenler arasında güçlü bir korelasyon olmaması öngörülmektedir [7].

Bu teorem, bir olayın olasılığını, o olayla ilgili koşulların önceki bilgisine dayanarak tanımlayabilir. Bu sınıflandırıcı, bir sınıftaki belirli bir özelliğin diğer herhangi bir özellikle doğrudan ilişkili olmadığını varsayar, ancak o sınıfa ait özellikler kendi aralarında karşılıklı bağımlılığa sahip olabilir. Bu yüzden bu algoritmayı kullanırken korelasyona dikkat edilmesi gerekmektedir [1].

2.1.5. KNN (K nearest neighbour): K en yakın komşu algoritması

"*K-en yakın komşu (KNN) algoritması, en basit ve en eski sınıflandırma algoritmalarından biridir. NB sınıflandırıcısının daha basit bir versiyonu olarak düşünülebilir. NB tekniğinin aksine, KNN algoritması olasılık değerlerini dikkate almayı gerektirmez. KNN algoritmasında 'K', 'oy' almak için dikkate alınan en yakın komşuların sayısıdır. Oy alması demek hangi küme elemanına yakınsa o gruba ait olmak için artı bir değer kazanması demektir. 'K' için farklı değerlerin seçilmesi, aynı örnek nesne için farklı sınıflandırma sonuçları üretebilir.*"[1].

Şekil 2.5, KNN'nin yeni bir nesneyi sınıflandırmak için nasıl çalıştığının bir örneğini göstermektedir. K=3 için yeni nesne (yıldız) 'siyah' olarak sınıflandırılmıştır; ancak K=5 olduğunda 'kırmızı' olarak sınıflandırılmıştır.



Şekil 2.5 - K En Yakın Komşu Algoritması Örnek

Sınıflandırma ve regresyon için kullanılabilen bir yöntemdir. N eğitim vektörü verildiğinde KNN algoritması, sınıfı belirlenecek olan bilinmeyen bir vektörün k-en yakın komşularını tanımlar [7].

2.2 Denetimsiz Öğrenme (Unsupervised Learning)

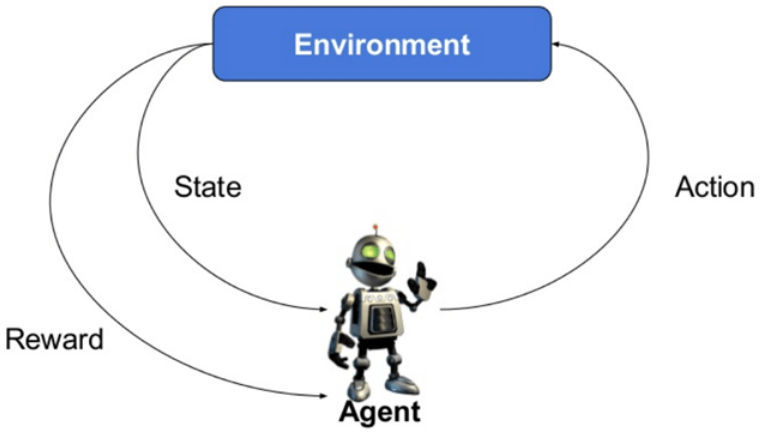
Denetimsiz öğrenme metodu, verilerden kurallar üretmek için tanımlanmamış mevcut örüntüleri tanımlamayı sağlamaktadır. Bu teknik, veri kategorilerinin bilinmediği ve verilerin etiketlerin olmadığı durumlarda kullanılmaktadır. Burada eğitim verileri etiketlenmemiştir. Denetimsiz öğrenme, öğrenme için istatistik tabanlı bir yaklaşım olarak kabul edilir ve bu nedenle etiketsiz verilerdeki gizli yapıyı bulma problemini ifade eder.

Bu algoritmada, tahmin edilecek / kestirilecek herhangi bir hedef veya sonuç değişkenimiz yoktur. Sınıfları farklı gruplarda kümelemek için kullanılır, bu da müşterileri belirli bir müdahale için farklı gruplarda segmentlere ayırmak için yaygın olarak kullanılır. Denetimsiz Öğrenme Örnekleri: Apriori algoritması, K-means.

2.3 Pekiştirmeli Öğrenme (Reinforcement Learning)

"Pekiştirmeli öğrenme, yazılım ajanlarının kümülatif ödül kavramını en üst düzeye çıkarmak için bir ortamda nasıl eylemde bulunmaları gerektiği ile ilgilenen bir makine öğrenimi alanıdır." [2].

Bu algoritma kullanılarak makine belirli kararlar vermek üzere eğitilir. Makine, ceza ödül sistemini kullanarak kendini sürekli eğittiği bir ortama maruz bırakılır. Bu makine geçmiş deneyimlerden öğrenir ve doğru iş kararları vermek için mümkün olan en iyi bilgiyi yakalamaya çalışır. Pekiştirmeli Öğrenme Örneği: Markov Karar Süreci [8] . Şekil 2.6' da pekiştirmeli öğrenme algoritmasının çalışma prensibini anlatan bir görsel mevcuttur.



Şekil 2.6 - Pekiştirmeli Öğrenme Algoritmasının Basit Gösterimi

3. MÜŞTERİ MEMNUNİYETİ UYGULAMASI

3.1 Problemin Tanımlanması

Çalışmada, beş farklı makine öğrenimi algoritmasıyla, havayolu firması yolcu memnuniyet anket verileri kullanılarak müşteri memnuniyetinin en iyi oranda tahmin edebilmesi hedeflenmiştir.

3.2 Verinin Hazırlanması

Veri kümesi Kaggle platformundan eğitim ve test verisi olarak çekilmiştir. Kaggle, veri bilimcileri ve makine öğrenimi çalışanları için çevrimiçi bir topluluk platformudur. Kaggle, kullanıcıların diğer kullanıcılarla iş birliği yapmasına, veri kümeleri bulmasına ve veri bilimi zorluklarını çözmek için diğer veri bilimcilerle rekabet etmesine olanak tanır.

3.2.1 Kütüphanelerin Çağrılması

Veri setini yüklemek ve ön işleme yapmak, veri setiyle istenildiği gibi oynamak, üzerinde analiz yapmak ve sonuçları görselleştirmek için çeşitli pyhton kütüphaneleri kullanılmaktadır.

NumPy, Python'da bilimsel hesaplama yapabilen temel paketlerden biridir. Çok boyutlu diziler, doğrusal cebir işlemleri ve Fourier dönüşümü gibi üst düzey matematiksel işlevler ve sözde rastgele sayı üreteçleri için işlevsellik içerir. Şekil 3.1'de kullanılan kod dizisi yardımı ile munpy kütüphanesi çağrılmıştır ve np kısaltması oluşturarak sonraki kod yazılımlarında kolaylık sağlanmıştır. Scikit-learn'de NumPy dizisi temel veri yapısıdır. scikit-learn verileri NumPy dizileri şeklinde alır. Kullanılam tüm verilerin bir NumPy dizisine dönüştürülmesi gerekecektir. NumPy'nin temel işlevi ndarray sınıfıdır, birçok boyutlu (n boyutlu) dizisi olmasıdır [9].

```
import numpy as np
import pandas as pd
from sklearn.metrics import accuracy_score
import warnings
warnings.simplefilter("ignore")
```

Şekil 3.1 - Veri Seti için Kütüphanelerin Çağrılması

"Pandas, veri işleme ve analiz için kullanılan bir Python kütüphanesidir. DataFrame adı verilen bir veri yapısı etrafında inşa edilmiştir. Bir pandas DataFrame, Excel elektronik tablosuna benzer bir tablodur. Pandas, bu tabloyu değiştirmek ve üzerinde çalışmak için çok çeşitli yöntemler sunar; özellikle SQL benzeri sorgulara ve tabloların birleştirilmesine izin verir. Bir dizideki tüm girdilerin aynı türde olmasını gerektiren NumPy'nin aksine, pandas her sütunun ayrı bir türe sahip olmasına izin verir (örneğin, tamsayılar, tarihler, kayan noktalı sayılar ve dizeler). Pandas tarafından sağlanan bir başka değerli araç da SQL, Excel dosyaları ve virgülle ayrılmış değerler (CSV) dosyaları gibi çok çeşitli dosya biçimlerinden ve veri tabanlarından alım yapabilmesidir." [9] .

Şekil 3.2'de kullanılan kod dizisi yardımı ile pandas kütüphanesi çağrılmıştır ve yine sonraki işlemlerde kullanılmak üzere pd kısaltmasına dönüştürülmüştür.

```
import matplotlib.pyplot as plt
plt.style.use("seaborn-whitegrid")
import seaborn as sns
# Let's set the theme of plots.
sns.set_theme()
sns.set(rc={"figure.dpi":300, "figure.figsize":(8,6)})
```

Şekil 3.2 - Veri Görselleştirme için Kütüphanelerin Çağrılması

Matplotlib, Python'daki en çok kullanılan bilimsel görsel ve çizim kütüphanelerinden biridir. Çizgi grafikleri, histogramlar, dağılım grafikleri gibi görselleştirmeler yapmak için işlevler sağlar. Verilerinizi ve analizinizin farklı yönlerini görselleştirmek için önemli içgörüler sağlayabilir [9].

Matplotlib son derecede kullanışlı ve popüler bir görselleştirme kütüphanesi olmasına rağmen eksiklikleri vardır. Seaborn, Matplotlib üzerinde çizim stili ve renk varyantları için uygun seçenekler sunan, yaygın istatistiksel çizim türleri için basit üst düzey işlevler tanımlayan ve Matplotlib'in işlevselliği ile entegre olan bir API sağlar [10] . Şekil 3.3'de görülen kod dizisi sayesinde matplotlib ve seaborn görselleştirme kütüphaneleri çağrılmış ve stilleri de belirlenmiştir.

```
from sklearn.model_selection import train_test_split
from sklearn.model_selection import cross_val_score
from sklearn.model_selection import StratifiedKFold
from sklearn.linear_model import LogisticRegression
from sklearn.tree import DecisionTreeClassifier
from sklearn.neighbors import KNeighborsClassifier
from sklearn.discriminant_analysis import LinearDiscriminantAnalysis
from sklearn.naive_bayes import GaussianNB
from sklearn.svm import SVC
```

Şekil 3.3 - Modellerin Kurulması için Kütüphanelerin Çağrılması

Scikit-learn (Sklearn) Python'da makine öğrenimi için en kullanışlı ve sağlam kütüphanedir. Python'da tutarlı bir arayüz aracılığıyla sınıflandırma, regresyon, kümeleme ve boyutsallık azaltma dahil olmak üzere makine öğrenimi ve istatistiksel modelleme için bir dizi verimli araç sağlar. Büyük ölçüde Python'da yazılmış olan bu kütüphane NumPy, SciPy ve Matplotlib üzerine inşa edilmiştir. Bu çalışmada da bolca Sklearn kullanılmaktadır.

```
from sklearn.metrics import accuracy_score
from sklearn.metrics import confusion_matrix
from sklearn.metrics import classification_report
```

Şekil 3.4 - Modelleri Değerlendirmek için Kütüphanelerin Çağrılması

3.2.2 Veri setinin yüklenmesi ve özetlenmesi

Şekil 3.5'de shape fonksiyonu ile veri setinde toplamda 103.904 satırın ve 25 tane öz niteliğin olduğu ve yüklenen verinin ilk 5 satırı; çağırılan özniteliklerin yani sütunların hepsi Şekil 3.6'da ve öznitelikler açıklamalarıyla detaylı olarak aktarıldığı Tablo 3.1'de görülmektedir.

```
[2]: data = pd.read_csv('train.csv')
     data.head()
```

[2]:

	Unnamed: 0	id	Gender	Customer Type	Age	Type of Travel	Class	Flight Distance	Inflight wifi service	Departure/Arrival time convenient	...
0	0	70172	Male	Loyal Customer	13	Personal Travel	Eco Plus	460	3	4	...
1	1	5047	Male	disloyal Customer	25	Business travel	Business	235	3	2	...
2	2	110028	Female	Loyal Customer	26	Business travel	Business	1142	2	2	...
3	3	24026	Female	Loyal Customer	25	Business travel	Business	562	2	5	...
4	4	119299	Male	Loyal Customer	61	Business travel	Business	214	3	3	...

5 rows × 25 columns

```
[4]: data.shape
```

```
[4]: (103904, 25)
```

Şekil 3.5 - Veri Setinin Yüklenmesi ve Çağrılması

```
[5]: data.columns
```

```
[5]: Index(['Unnamed: 0', 'id', 'Gender', 'Customer Type', 'Age', 'Type of Travel',
       'Class', 'Flight Distance', 'Inflight wifi service',
       'Departure/Arrival time convenient', 'Ease of Online booking',
       'Gate location', 'Food and drink', 'Online boarding', 'Seat comfort',
       'Inflight entertainment', 'On-board service', 'Leg room service',
       'Baggage handling', 'Checkin service', 'Inflight service',
       'Cleanliness', 'Departure Delay in Minutes', 'Arrival Delay in Minutes',
       'satisfaction'],
      dtype='object')
```

Şekil 3.6 - Sütunların Çağrılması

Tablo 3.1 Özniteliklerin Açıklaması

Sütun (Öznitelik)	**Özniteliklerin Açıklaması**
Gender:	Yolcuların cinsiyeti (Kadın, Erkek)
Customer Type:	Müşteri tipi (Sadık müşteri, sadakatsiz müşteri)
Age:	Yolcuların gerçek yaşı
Type of Travel:	Yolcuların uçuş amacı (Kişisel Seyahat, İş Seyahati)
Class:	Yolcuların uçaktaki seyahat sınıfı (Business, Eco, Eco Plus)
Flight distance:	Bu yolculuğun uçuş mesafesi
Inflight wifi service:	Uçak içi wifi hizmetinden memnuniyet düzeyi (0: Geçerli Değil;1-5)
Departure/Arrival time convenient:	Kalkış/Varış saatinin uygunluğuna ilişkin memnuniyet düzeyi
Ease of Online booking:	Online rezervasyonun kolaylığıyla ilgili memnuniyet düzeyi
Gate location:	Uçağa giriş kapısının konumunun memnuniyet düzeyi
Food and drink:	Yiyecek ve içecekten memnuniyet düzeyi
Online boarding:	Online check-in memnuniyet düzeyi
Seat comfort:	Koltuk konforu memnuniyet seviyesi
Inflight entertainment:	Uçak içi eğlenceden memnuniyet düzeyi
On-board service:	Uçağa biniş hizmetinden memnuniyet düzeyi
Leg room service:	Koltuklar arası ayak mesafesinden memnuniyet düzeyi
Baggage handling:	Bagaj işlemlerinden memnuniyet düzeyi
Check-in service:	Check-in hizmetinden memnuniyet düzeyi
Inflight service:	Uçak içi hizmetlerden memnuniyet düzeyi
Cleanliness:	Temizlikten memnuniyet düzeyi
Departure Delay in Minutes:	Kalkışta gecikilen dakika
Arrival Delay in Minutes:	Varışta gecikilen dakika
Satisfaction:	Havayolu memnuniyet düzeyi (Memnuniyet, nötr veya memnuniyetsizlik)

Şekil 3.7'de özniteliklerin detaylı bilgileri görülmektedir. Şekil 3.7'de Int64, sütuna ait verilerin türünün tam sayı olduğu; Float, sütuna ait verilerin türünün reel sayı olduğu ve Object, sütuna ait verilerin türünün sözel veri olduğu gösterilmektedir.

Şekil 3.7'de yer alan "Unnamed:0" sütunu sıralama için kullanıldığından ve "Id" sütunu kişiler için ayırt edici nitelik olarak kullanıldığı için, müşteri memnuniyetini anlamak için modeller de kullanmanın bir anlamı olmayacaktır. Bu yüzden aşağıdaki kod ile veri kümesinden çıkarılmıştır.

data = data.drop(data.iloc[:,[0, 1]], axis = 1)

```
[6]: data.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 103904 entries, 0 to 103903
Data columns (total 25 columns):
 #   Column                            Non-Null Count   Dtype
---  ------                            --------------   -----
 0   Unnamed: 0                        103904 non-null  int64
 1   id                                103904 non-null  int64
 2   Gender                            103904 non-null  object
 3   Customer Type                     103904 non-null  object
 4   Age                               103904 non-null  int64
 5   Type of Travel                    103904 non-null  object
 6   Class                             103904 non-null  object
 7   Flight Distance                   103904 non-null  int64
 8   Inflight wifi service             103904 non-null  int64
 9   Departure/Arrival time convenient 103904 non-null  int64
 10  Ease of Online booking            103904 non-null  int64
 11  Gate location                     103904 non-null  int64
 12  Food and drink                    103904 non-null  int64
 13  Online boarding                   103904 non-null  int64
 14  Seat comfort                      103904 non-null  int64
 15  Inflight entertainment            103904 non-null  int64
 16  On-board service                  103904 non-null  int64
 17  Leg room service                  103904 non-null  int64
 18  Baggage handling                  103904 non-null  int64
 19  Checkin service                   103904 non-null  int64
 20  Inflight service                  103904 non-null  int64
 21  Cleanliness                       103904 non-null  int64
 22  Departure Delay in Minutes        103904 non-null  int64
 23  Arrival Delay in Minutes          103594 non-null  float64
 24  satisfaction                      103904 non-null  object
dtypes: float64(1), int64(19), object(5)
memory usage: 19.8+ MB
```

Şekil 3.7 – Özniteliklerin Detaylı Bilgisi

Şekil 3.8'de Isnull() fonksiyonu ile eksik verilerin kontrolü görülmektedir.

Handling The Missing Data

```
[14]: data.columns[data.isnull().any()]

[14]: Index(['Arrival Delay in Minutes'], dtype='object')

[15]: data.isnull().sum()

[15]: Unnamed: 0                          0
      id                                  0
      Gender                              0
      Customer Type                       0
      Age                                 0
      Type of Travel                      0
      Class                               0
      Flight Distance                     0
      Inflight wifi service               0
      Departure/Arrival time convenient   0
      Ease of Online booking              0
      Gate location                       0
      Food and drink                      0
      Online boarding                     0
      Seat comfort                        0
      Inflight entertainment              0
      On-board service                    0
      Leg room service                    0
      Baggage handling                    0
      Checkin service                     0
      Inflight service                    0
      Cleanliness                         0
      Departure Delay in Minutes          0
      Arrival Delay in Minutes          310
      satisfaction                        0
      dtype: int64
```

Şekil 3.8 - Eksik Verilerin Kontrol Edilmesi

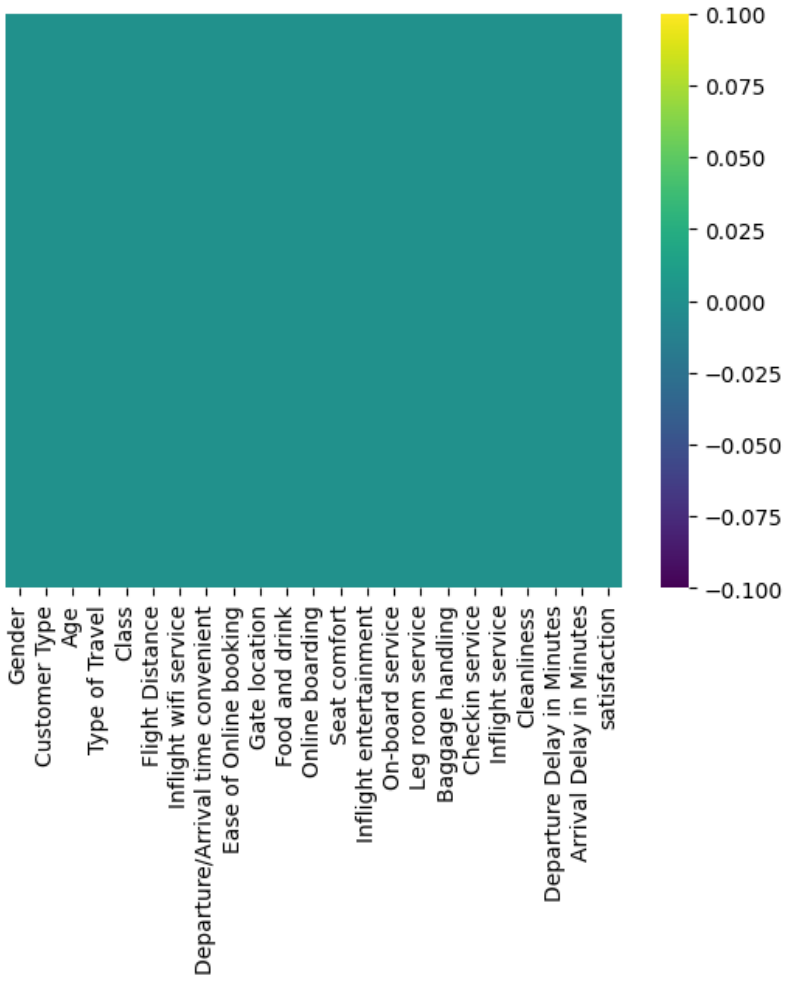
Şekil 3.8'de Isnull() fonksiyonu ile veri de eksik veri olup olmadığı detaylı olarak kontrol edildiğinde "Arrival Delay in Minutes" sütununda 310 satırın eksik olduğunu bilgisine ulaşılmıştır.

"Arrival Delay in Minutes" sütununun silinmesi düşünülebilir, ancak bu sütundaki eksik değerler çok az olduğu için, sadece eksik değerleri ortalama(mean) fonksiyonu ile aşağıdaki kod ile tahmin edilip veri setine eklenmesi modelin oluşmasını olumsuz olarak etkilemeyecektir.

data['Arrival Delay in Minutes'].fillna(data['Arrival Delay in Minutes'].mean(axis = 0), inplace = True)

Yukarıdaki işlemden sonra tekrar eksik veri olup olmadığını görselleştirerek Şekil 3.9'da görülen kod ile kontrol edilmiştir. Eğer eksik veri olsaydı yeşil zemin üzerinde kesikli sarı çizikler ile grafik üzerinde görülebilirdi. Ama şimdi verilerin tam olduğu anlaşılmaktadır.

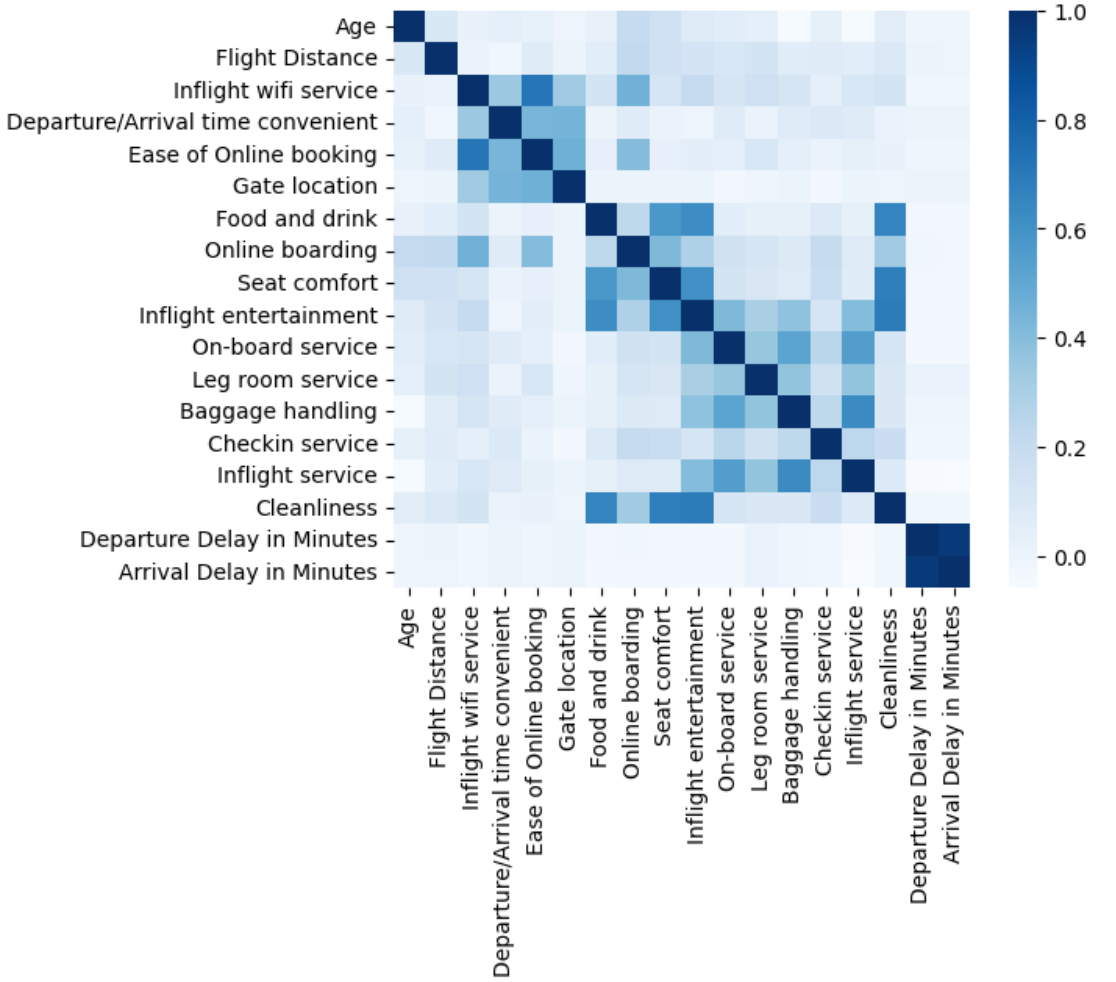
```
[23]: import seaborn as sns
      sns.heatmap(data.isnull(), yticklabels=False, cmap="viridis")

[23]: <AxesSubplot: >
```



Şekil 3.9 - Eksik Verinin Görselleştirilmesi

Model için sütunlar arasında da korelasyon olup olmadığını da model için kontrol edilmesi gerekmektedir. Yüksek yoğunluklu korelasyon modellerin hatalı oluşturulmasını ve yanlış sonuç üretmesini sağlayabilir. Şekil 3.10'da ısı haritası ile hangi özniteliklerin birbiriyle korelasyonunun olup olmadığı görülebilmektedir. Mavinin yoğun tonları korelasyonun yüksek olduğunu göstermektedir.

```
[24]: corr_mat = data.corr()
      sns.heatmap(corr_mat, square = True, cmap = 'Blues')
      pass
```



Şekil 3.10 - Korelasyon Isı Haritası

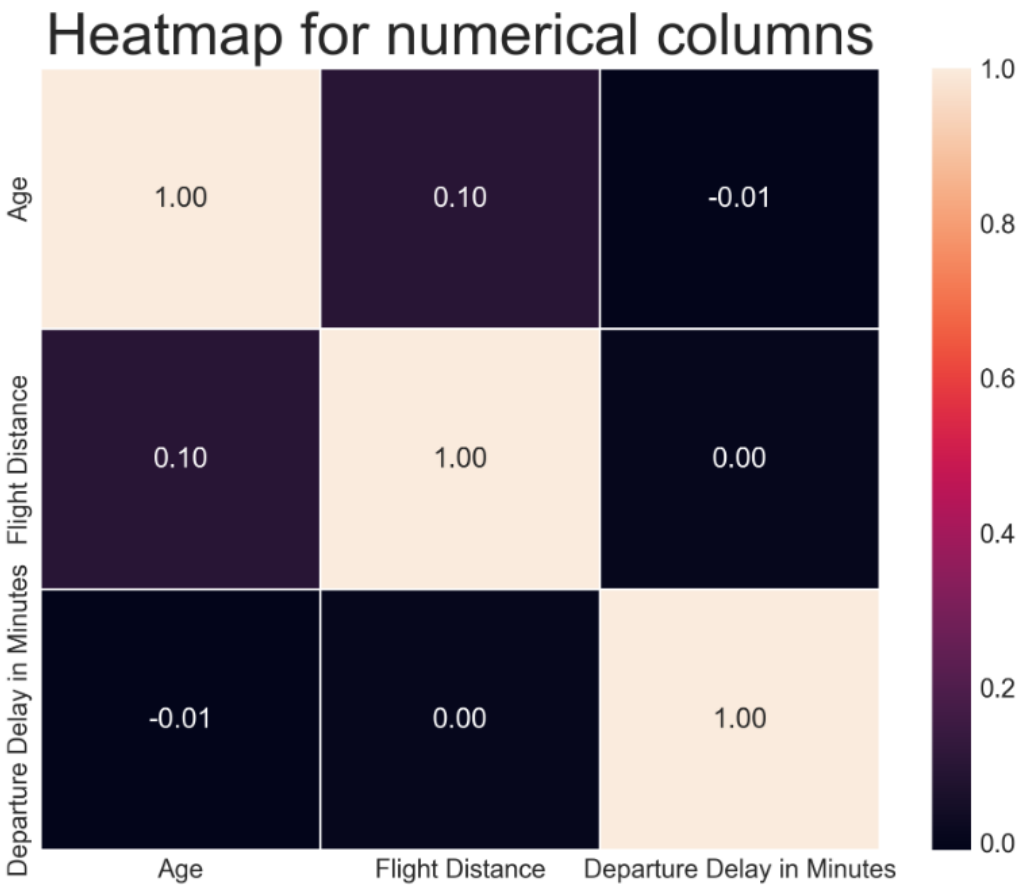
Şekil 3.10 incelendiğinde, mavinin yoğun tonlarının korelasyonun yüksek olduğunu göstermektedir. Ayrıca "Arrival Delay in Minutes" ve "Departure Delay in Minutes" sütunlarının birbirlerini çok etkilediği ve korelasyonun yüksek olduğu da görülmektedir. Beklenildiği gibi geç kalkış yapan uçak geç varış yapacaktır. Dolayısıyla, "Arrival Delay in Minutes" sütununu silerek, aynı verileri tekrarlamadan doğru sonuca ulaşılabilmektedir. Aşağıdaki kod ile "Arrival Delay in Minutes" sütunu veri setinden temizlendi ve korelasyon kaldırılmıştır.

data = data.drop(data.iloc[:,[21]], axis = 1)

Korelasyonun olup olmadığından emin olmak için sadece sayısal özniteliklerin korelasyonu Şekil 3.11'de incelenmiştir.

```
sns.heatmap(data.corr(), annot = True, linewidths=.5, fmt=".2f")
plt.title("Heatmap for numerical columns", size=25)
```

```
Text(0.5, 1.0, 'Heatmap for numerical columns')
```



Şekil 3.11 - Sayısal Faktörlerin Korelasyon Tablosu

Şekil 3.11 yani sayısal faktörlerin korelasyon tablosunun ısı haritası incelendiğinde yüksek bir değer görülmemekte ve güçlü bir korelasyon olmadığı görülmektedir. Bu şekilde modellerin eğitilmesinde hata oluşmayacaktır.

Varış zamanındaki gecikme, yaş ve uçuş mesafesi sütunları değişken sayısal değerlere sahip olduğu için bunların kontrolü de gerekmektedir. Bu bahsedilen öznitelikler haricindeki diğer sütunlar 1 ile 5 arasında bir değere ve sayısal değerlere sahip gözükmelerine rağmen aslında anket formunun yapısına göre kategorik değerlere sahiptirler. Şekil 3.16'dan ve Şekil 3.28'ye kadar olan görsel grafiklerde bu özniteliklerin kategorik olduğu net olarak görülmektedir.

3.2.3 Veri setinin görselleştirilmesi

Veri görselleştirme, bilgi ve verilerin görsel veya grafiksel temsilidir. Veri görselleştirme araçları; çizelgeler, grafikler ve haritalar gibi görsel unsurları kullanarak verilerdeki eğilimleri, aykırı değerleri ve kalıpları görmek ve anlamak için erişilebilir bir yol sağlar. Ayrıca, çalışanların veya işletme sahiplerinin verileri teknik olmayan kitlelere kafa karışıklığı olmadan sunmaları için mükemmel bir yol sağlar.

Veri görselleştirmenin önemi basittir: insanların verileri görmesine, etkileşime girmesine ve daha iyi anlamasına yardımcı olur. İster basit ister karmaşık olsun, doğru görselleştirme uzmanlık seviyeleri ne olursa olsun herkesi aynı noktada buluşturabilir [11].

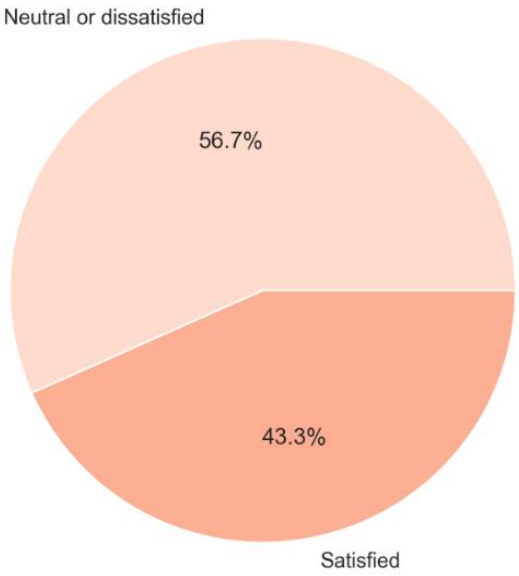
```
data['satisfaction'].value_counts()
```

```
neutral or dissatisfied    58879
satisfied                  45025
Name: satisfaction, dtype: int64
```

Şekil 3.12 - Memnuniyet Sonuç Sayımı

Şekil 3.12'de kod yapısı ve çıktısıyla birlikte veri setindeki memnuniyet sütunun sonuçları sayısal olarak görülebilir ve bu sonuçlar incelendiğinde 103904 kişiden 45025 kişinin memnun olduğu okunmaktadır. Bu sonucun görselleştirilerek incelenmesi sağlanabilir ve böylelikle karmaşık sayı dizilerinden kurtularak herkesin kolayca anlayabileceği bir hale getirilebilir. Şekil 3.13'de bulunan kod yapısı ve bu kod yapısının çıktısı olan pasta grafiği ile Şekil 3.12'de bulunan sonuçların görselleştirilmesi aktarılmıştır.

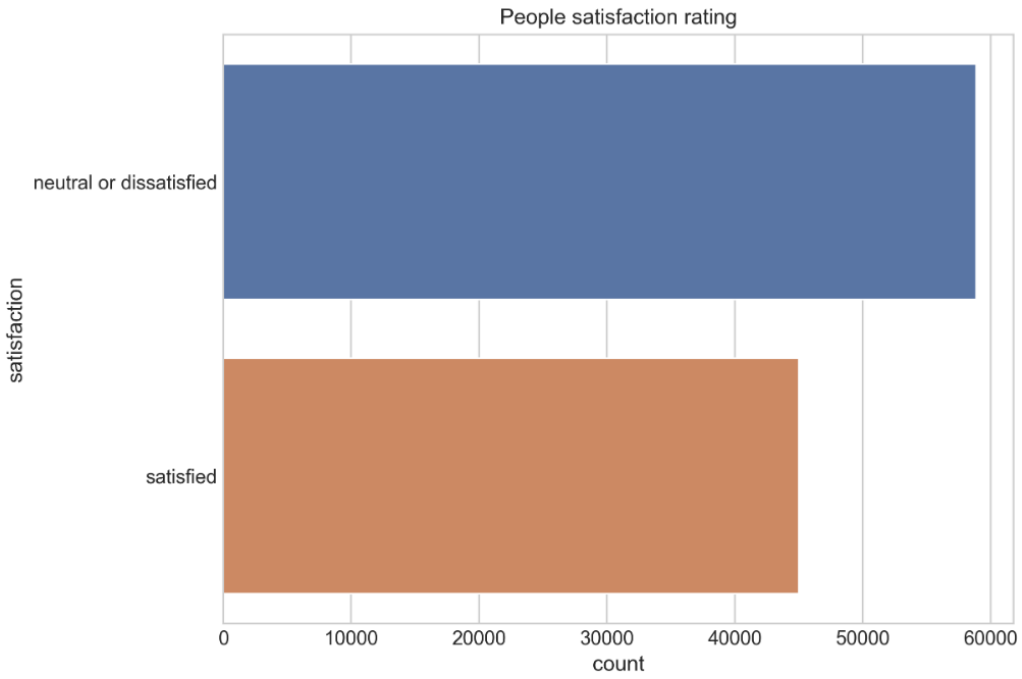
```
plt.pie(data.satisfaction.value_counts(), labels = ["Neutral or dissatisfied", "Satisfied"],
        colors = sns.color_palette("Reds"), autopct = '%1.1f%%')
pass
```



Şekil 3.13 - Memnuniyet Oranının Pasta Grafiği

Şekil 3.14'de memnuniyet sütununun farklı bir görsel çıktısı mevcuttur.

```
# Bar plot for the satisfaction column with the counting

sns.countplot(y = "satisfaction", data = data)
plt.title("People satisfaction rating")
```

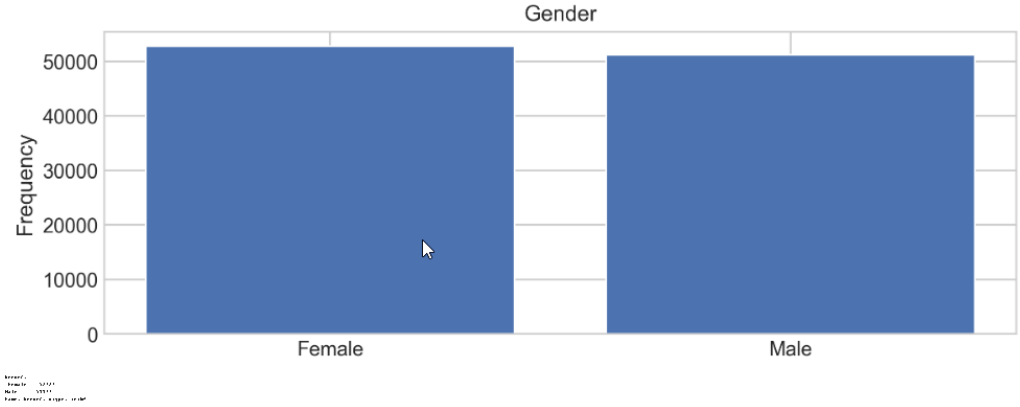


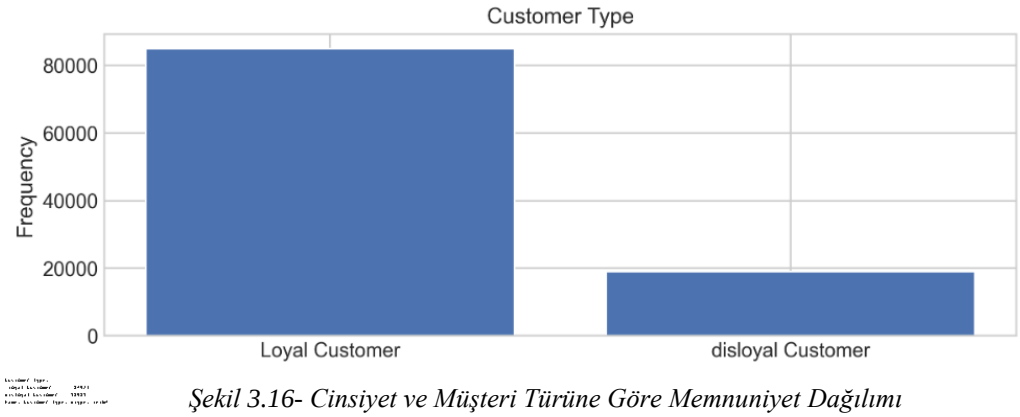
Şekil 3.14 - Memnuniyet Oranının Dağılımı

Memnuniyet sütunun yanında diğer sütunların yani öz niteliklerin daha iyi anlaşılması için veri görselleştirme uygulanacaktır. İlk olarak sayısal gözüken ama kategorik değerlere sahip olması gereken sütunları kategorik veri tipine çevrilmesi sağlanmıştır. Daha sonra kategorik sütunların tek tek çubuk grafik olarak gösterilmesi için Şekil 3.15'deki fonksiyon yazılmış ve Şekil 3.16-3.24'de verilen grafikler elde edilmiştir.

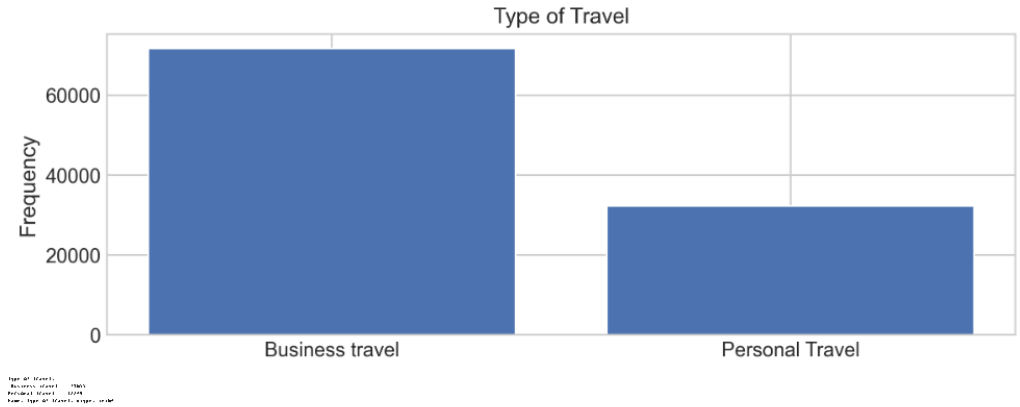
```
[22]: categorical_indexes = [0, 1, 3, 4] + list(range(6, 20))
      data.iloc[:,categorical_indexes] = data.iloc[:,categorical_indexes].astype('category')

[25]: numeric_df = data.select_dtypes(['float', 'int'])
      categorical_df = data.select_dtypes(['object', 'category'])

[26]: def bar_plot(variable):
          # we get the features
          var = data[variable]
          # count number of categorical variable
          varValue = var.value_counts()

          plt.figure(figsize = (9,3))
          plt.bar(varValue.index, varValue)
          plt.xticks(varValue.index, varValue.index.values)
          plt.ylabel("Frequency")
          plt.title(variable)
          plt.show()
          print("{}: \n {}".format(variable,varValue))

[27]: ## categorical_var = ["Gender", "Customer_Type", "Type_of_Travel", "Class"]
      for each in categorical_df:
          bar_plot(each)
```

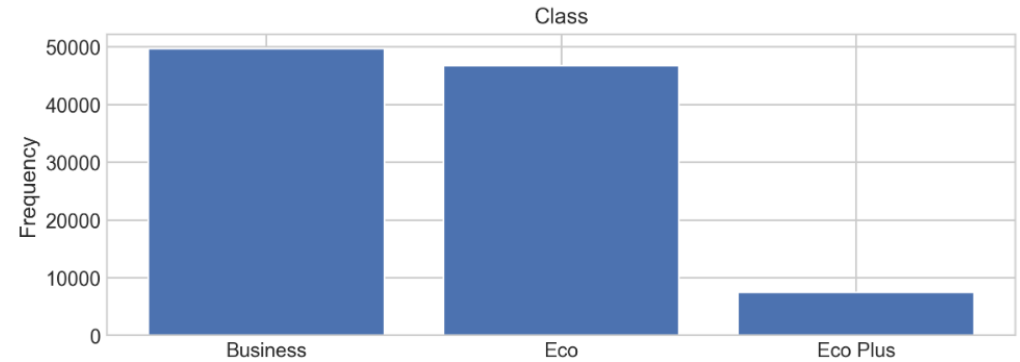
Şekil 3.15 - Kategorik Veriyi Görselleştirme İşlemleri



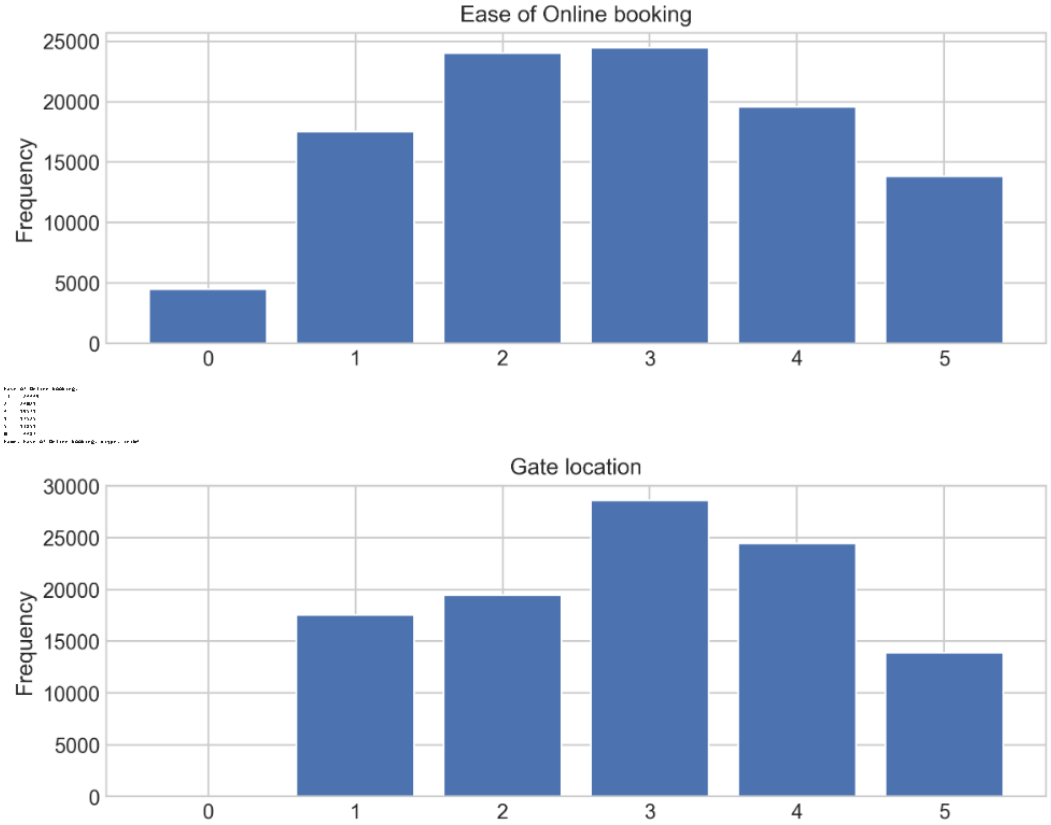


Şekil 3.16- Cinsiyet ve Müşteri Türüne Göre Memnuniyet Dağılımı

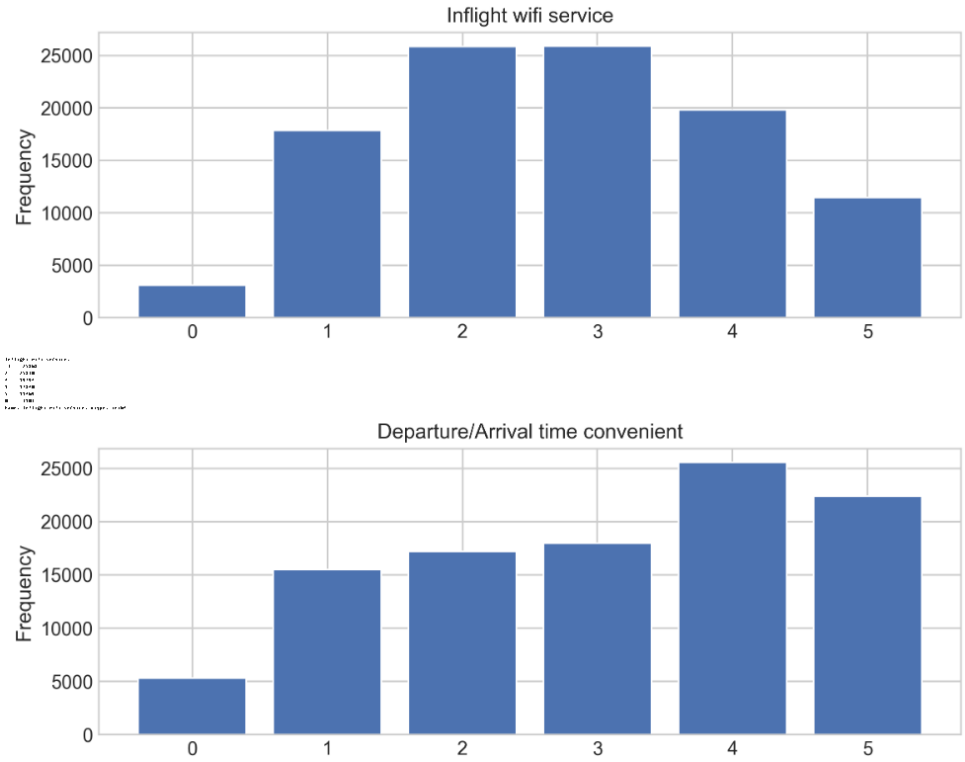




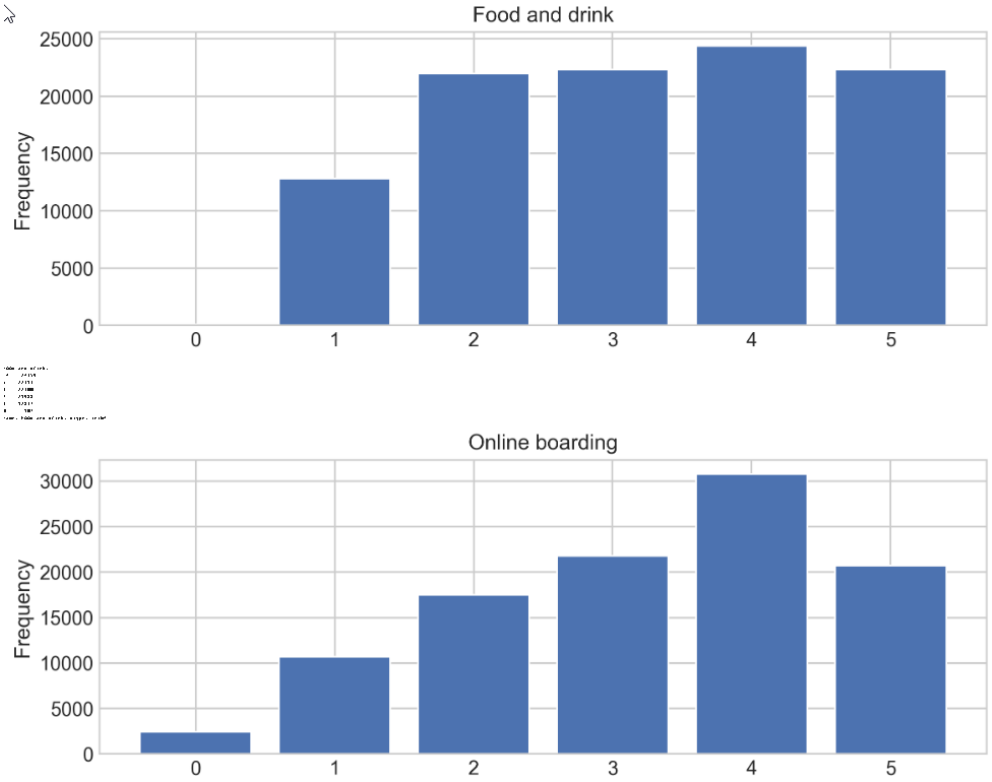
Şekil 3.17 - Sınıf ve Seyahat Türüne Göre Memnuniyet Dağılımı



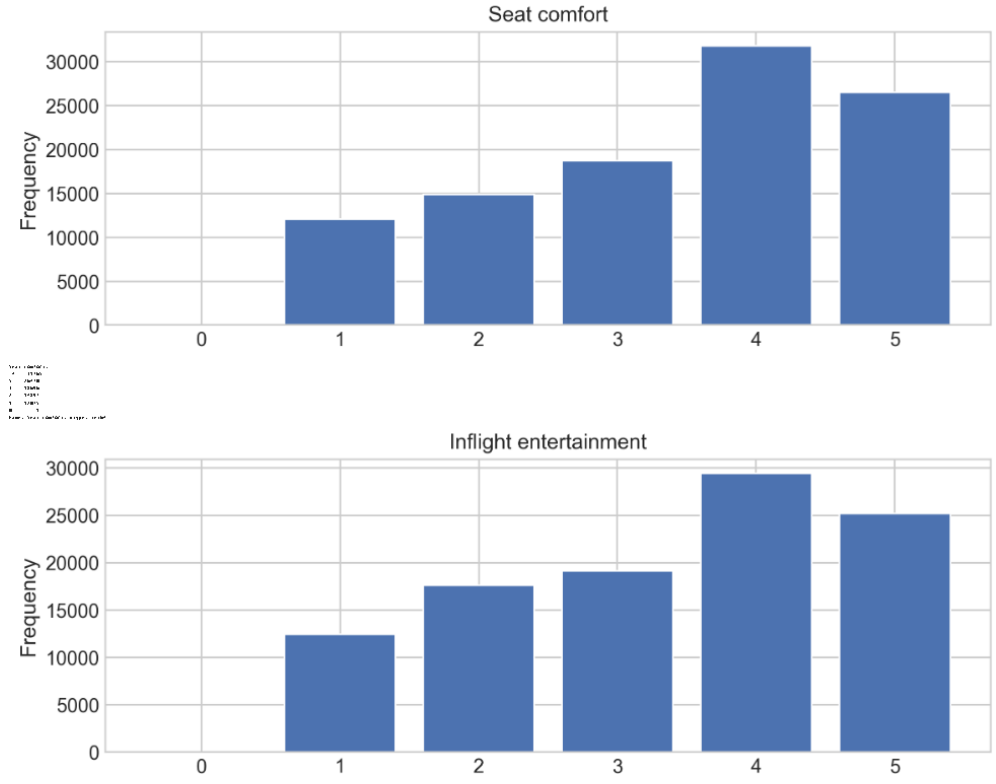
Şekil 3.18 - Online Rezervasyon ve Kapı Lokasyonuna Göre Memnuniyet Dağılımı



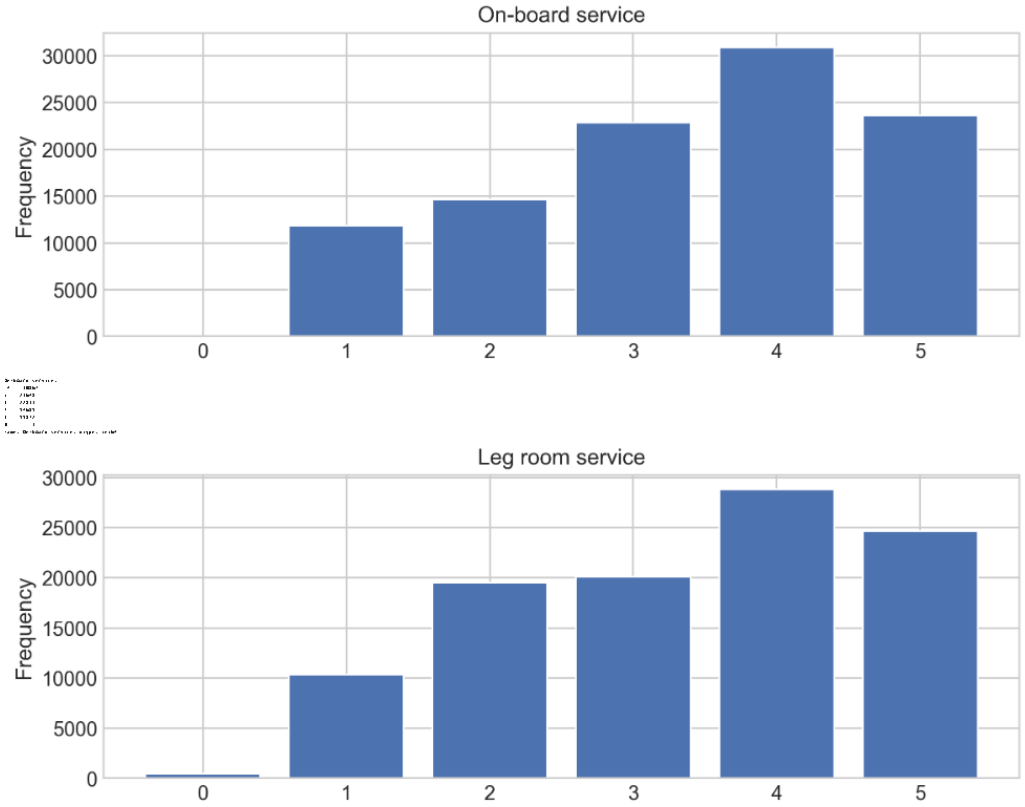
Şekil 3.19 - Wifi Hizmeti ve Zamanlama Doğruluğuna Göre Memnuniyet Dağılımı



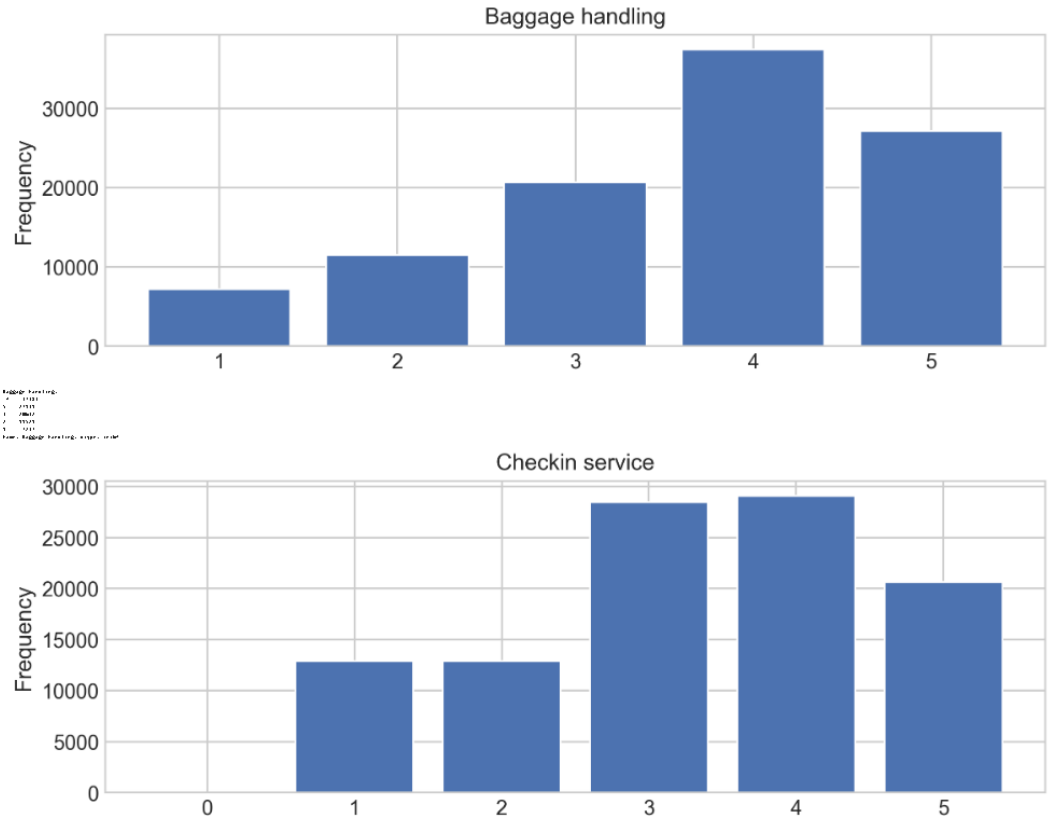
Şekil 3.20 - Yiyecek Servisi ve Online Check in Türüne Göre Memnuniyet Dağılımı



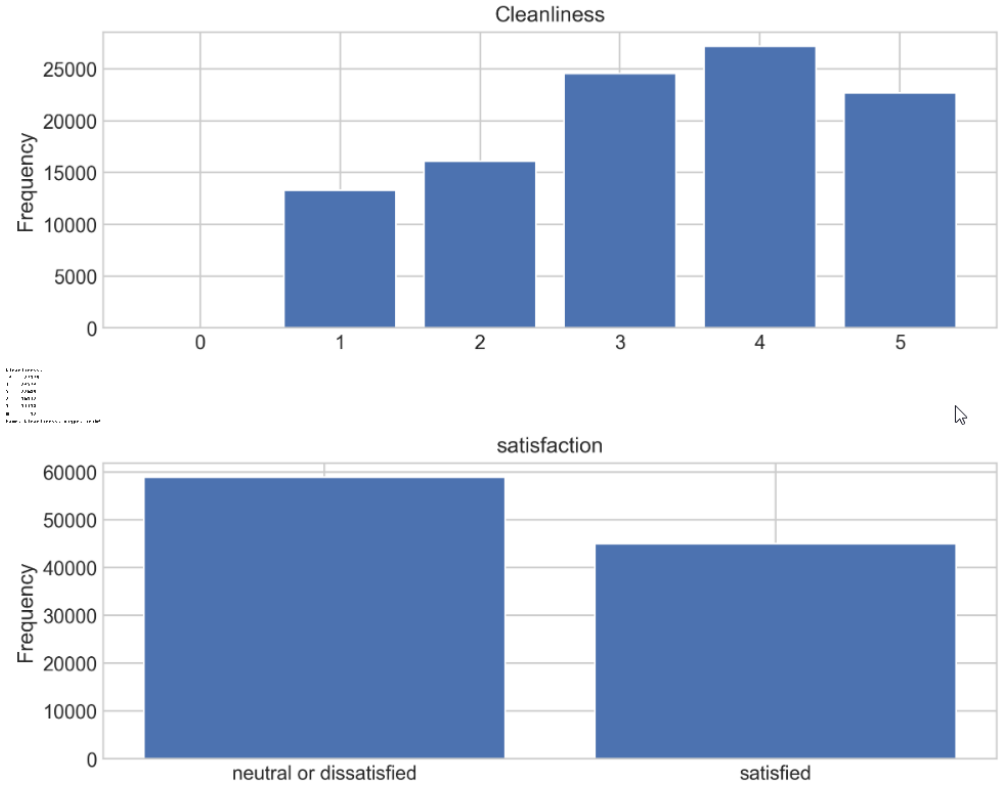
Şekil 3.21 - Koltuk Komforu ve Uçakiçi Eğlenceye Göre Memnuniyet Dağılımı



Şekil 3.22 - Biniş Servisi ve Koltuk Mesafesine Göre Memnuniyet Dağılımı



Şekil 3.23 - Bagaj Servisi ve Checkin Servisine Göre Memnuniyet Dağılımı

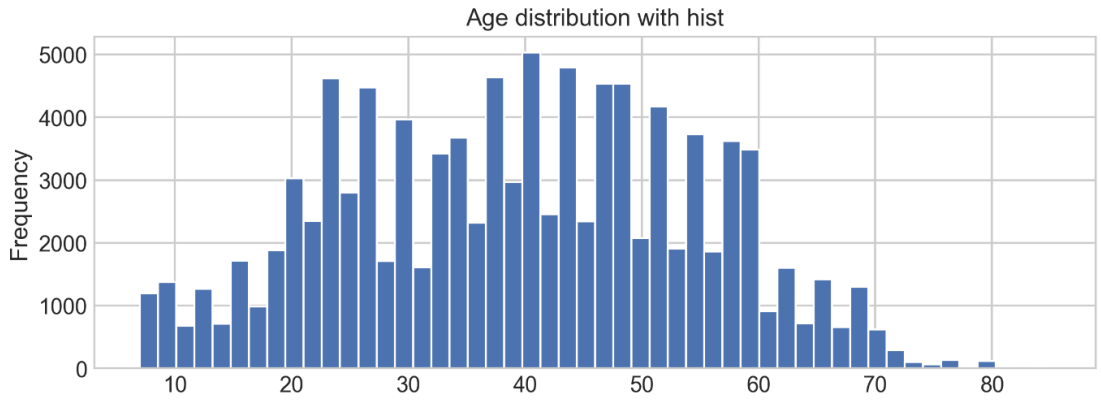


Şekil 3.24 - Temizliğe Göre ve Genel Memnuniyet Dağılımı

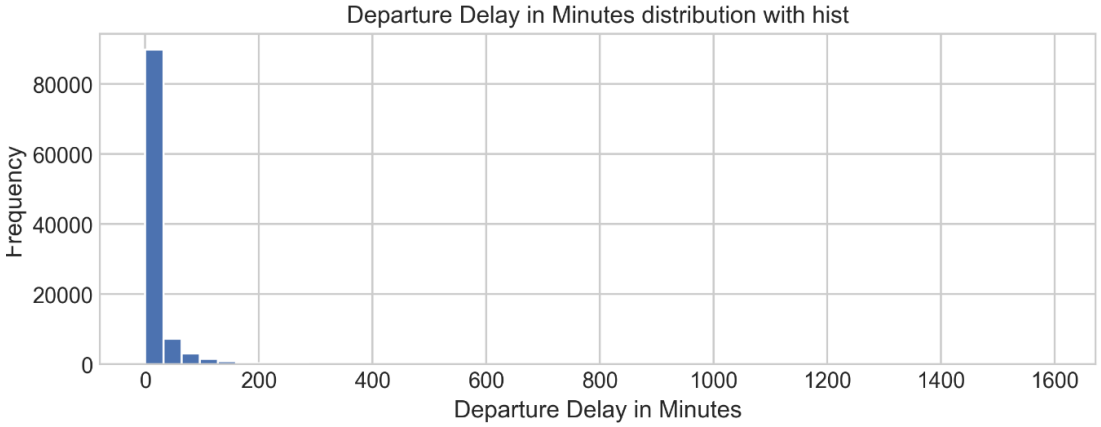
Kategorik sütunlardan sonra sayısal sütunların görselleştirilmesi ile dağılımları incelenmiştir. İnceleme için yazılan kod yapısı Şekil 3.25'de ve grafikleri Şekil 3.26-3.28'de verilmektedir.

```
[30]: def plot_hist(variable):
          plt.figure(figsize = (9,3))
          plt.hist(data[variable], bins = 50)
          plt.xlabel(variable)
          plt.ylabel("Frequency")
          plt.title("{} distribution with hist".format(variable))
          plt.show()

[31]: # numerical_var = ["Age", "Flight Distance", "Departure Delay in Minutes"]
      for n in numeric_df:
          plot_hist(n)
```

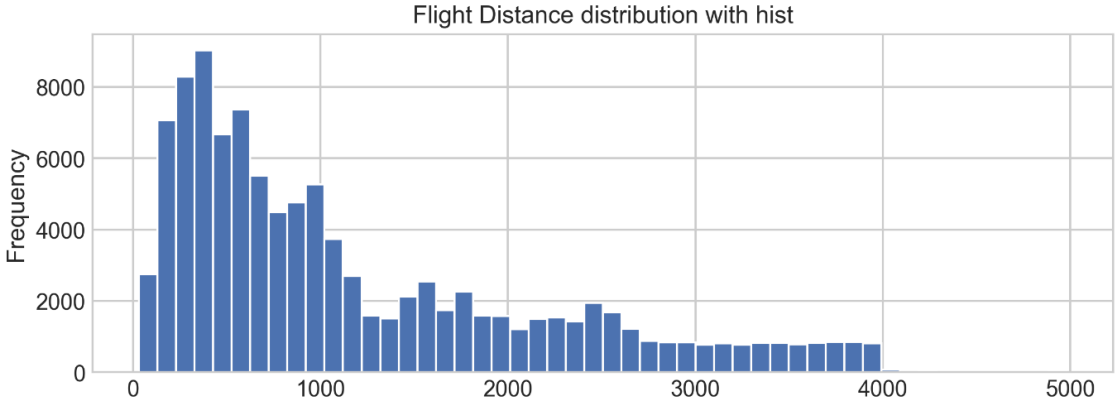
Şekil 3.25 - Sayısal Sütunların Görselleştirilmesi için Yazılan Kod Dizisi



Şekil 3.26 - Yaşa Göre Memnuniyet Dağılımı



Şekil 3.27 - Geç Kalkış Durumuna Göre Memnuniyet Dağılımı



Şekil 3.28 - Uçuş Mesafesine Göre Memnuniyet Dağılımı

3.3 Modellerin Oluşturulması

Modelin oluşturulması için veri setinin "x" ve "y" olmak üzere ayrılması gerekmektedir. Şekil 3.29'da bulunan X değişkeninin tuttuğu değerler "satisfaction(memnuniyet)" sütunu hariç diğer tüm sütunlardır. Y değişkeni ise hedef sütun yani memnuniyet değerlerini tutacak olan veri kümesidir. Aslında burada yapılan özniteliklerin hedef öznitelik ve diğerleri şeklinde ayrılması ve değişkenlere atanmasıdır. Şekil 3.29'da bu işlemin gerçekleştiği kod dizisi görülmektedir.

```
# Split-out validation dataset
X = data.drop("satisfaction", axis = 1)
y = data["satisfaction"]
X_train, X_validation, Y_train, Y_validation = train_test_split(X, y, train_size=0.80,
                                                                random_state=1, shuffle=True)
```

Şekil 3.29 - Veri Setinin Eğitim ve Validasyon Verisi Olarak Ayrılması

Daha sonrasında X ve y veri setleri de eğitim verisi ve validasyon verisi olarak ayrılması gerekmektedir. Şekil 3.29'da görüldüğü gibi bu ayırma işlemi %80 olarak tanımlanmıştır. Yani verilerin yüzde 80'i eğitim verileri ve %20'si ise validasyon yani doğrulama ya da test verisi olacaktır.

Grafikte elde bulunan veriler rastgele olarak iki alt örneğe (öğrenme örneği ve doğrulama veya test örneği) ayrılmaktadır. Model eğitim seti üzerinden tahmin edilmekte ve performansı değerlendirme örneği kullanılarak incelenmektedir. Değerlendirme örneğinin rastgele seçiminden kaynaklanan herhangi bir önyargıyı azaltmak için, kayıp verilerin birkaç rastgele bölünmesi üzerinden ortalaması alınmaktadır.

Şekil 3.30'da bulunan kod dizisi ile önceden Scikit-Learn kütüphanesinden çağrılan algoritmalar daha sonra kullanılmak üzere bir değişkene liste olarak atanmışlardır.

```
# Spot Check Algorithms
models = []
models.append(('LR', LogisticRegression(solver='liblinear', multi_class='ovr')))
models.append(('LDA', LinearDiscriminantAnalysis()))
models.append(('KNN', KNeighborsClassifier()))
models.append(('CART', DecisionTreeClassifier()))
models.append(('NB', GaussianNB()))
```

Şekil 3.30 - Modellerin Listelenmesi

Listeye alınan algoritmalar Şekil 3.31'deki fonksiyon ile çalıştırılmaktadır. Her biri tek tek daha öncesinde %80 olarak ayrılan eğitim verisiyle eğitilmektedir. Ve sonrasında farklı algoritmalar ile oluşan her bir modelin tahminleri, accuracy (doğruluk) adlı fonksiyon ve %20 olarak ayrılan test verisiyle kontrol edilmektedir. Şekil 3.31'deki bulunan sonuç skorları ile en iyi sonucu veren algoritmanın CART (DecisionTreeClassifier) yani karar ağacı olduğu anlaşılmaktadır. Makine öğrenmesi birçok sınıflandırma algoritmasına sahiptir. Her veri seti için hedeflendiği veri setine göre farklı algoritmalar kullanılarak tahmin modeli oluşturulabilir. Ama en iyi modeli bularak en iyi tahmin oluşturulması sağlanması için algoritmalar değerlendirilmelidirler.

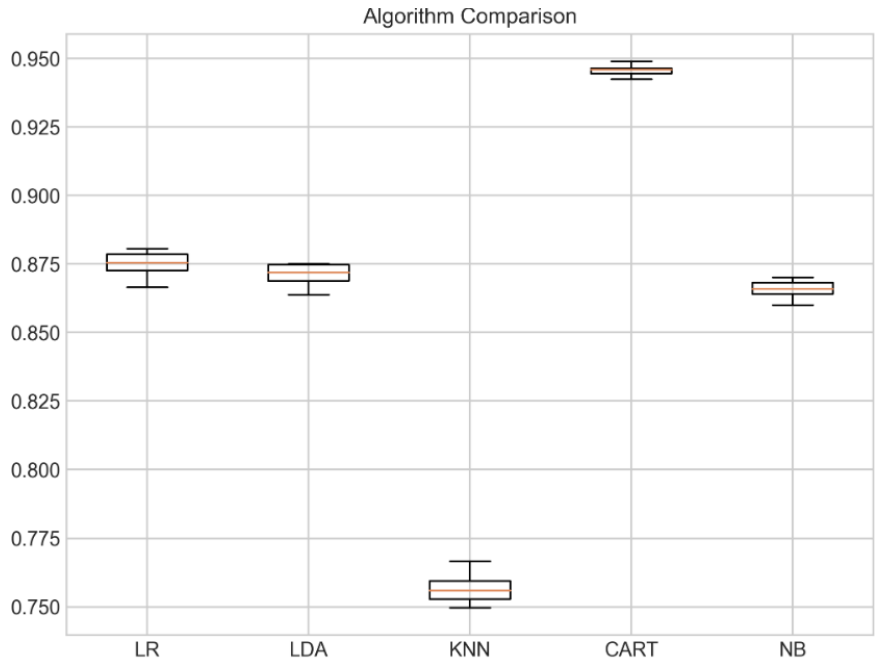
```
# evaluate each model in turn
results = []
names = []
for name, model in models:
    kfold = StratifiedKFold(n_splits=10, random_state=1, shuffle=True)
    cv_results = cross_val_score(model, X_train, Y_train, cv=kfold, scoring='accuracy')
    results.append(cv_results)
    names.append(name)
    print('%s: %f (%f)' % (name, cv_results.mean(), cv_results.std()))
```

```
LR: 0.874956 (0.004616)
LDA: 0.871010 (0.003905)
KNN: 0.756866 (0.005169)
CART: 0.945527 (0.001787)
NB: 0.865693 (0.003199)
```

Şekil 3.31 - Modellerin Fonksiyon İçinde Çalıştırılması ve Sonuçları

Şekil 3.31'da çıkan sonuçları daha iyi anlamak için Şekil 3.32'de kutu diyagramı ile görselleştirilmesi incelenmektedir.

```
# Compare Algorithms
from matplotlib import pyplot
pyplot.boxplot(results, labels=names)
pyplot.title('Algorithm Comparison')
pyplot.show()
```



Şekil 3.32 - Algoritma Doğruluk Oranlarının Kutu Diyagramında Kıyaslanması

Şekil 3.32'de anlaşıldığı gibi KNN algoritması en zayıf sonucu verirken karar ağacı algoritması neredeyse %95'lik bir tahmin skoru üretmiştir.

3.4 En İyi Modelin Değerlendirilmesi

En iyi algoritma seçildikten sonra Şekil 3.33'deki şekliyle tekrar karar ağacı algoritması ile model oluşturulup bu model bir değişkene atanmıştır. Daha sonra bu model kullanılarak X validasyon verilerinden tahminler üretilmiştir. X validasyon, veri setinden %20 olarak ayırdığımız test verisidir ve bu veri setinde "satisfaction" (memnuniyet) hedef sütunu bulunmamaktadır. Oluşturulan model ile X validasyon veri setinden bir memnuniyet sonucunun tahminlerini oluşturması sağlanmış ve çıkan tahminler y_tahmin değişkenine atanmıştır. Ve sonrasında X validasyon veri setinden üretilen y_tahmin verisi ile gerçek memnuniyet değerlerini saklayan y validasyon değerleri karşılaştırılmıştır. Modelin tahminleri ile gerçek değerlerin %94 oranında uyuştuğu sonucu ortaya çıkmıştır.

```
model = DecisionTreeClassifier()
model.fit(X_train, Y_train)
```

```
▾ DecisionTreeClassifier
DecisionTreeClassifier()
```

```
y_tahmin = model.predict(X_validation)
print(y_tahmin)
```

```
['satisfied' 'neutral or dissatisfied' 'neutral or dissatisfied' ...
 'satisfied' 'neutral or dissatisfied' 'satisfied']
```

```
print(np.mean(y_tahmin==Y_validation))
```

```
0.9437466916895241
```

Şekil 3.33 - Tahminleme Kod Yapısı

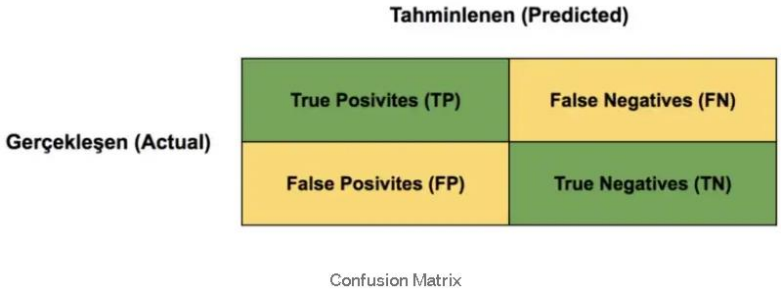
Şekil 3.34'de yukarıda anlatılan işlem tekrar edilerek yeniden farklı bir tahmin oluşturulmuş ve bu üretilen tahminler "predictions" adlı değişkene atanmıştır ve gerçek değerler ile üretilen tahminlerin yukarıdaki gibi ortalama fonksiyonu yerine Accuracy score (doğruluk skoru) adlı kütüphaneden çağrılan fonksiyon kullanılarak karşılaştırılması yapılmıştır. İşlem ve sonucu Şekil 3.34'de mevcuttur. Accuracy score (doğruluk skoru) değeri %94 olarak çıkmıştır. Accuracy score ile Confusion matrix ve Classification Report fonksiyonları da modelin durumu hakkında bilgi verilmek üzere kullanılmıştır.

```
model = DecisionTreeClassifier()
model.fit(X_train, Y_train)
predictions = model.predict(X_validation)
print('Accuracy Score(Doğruluk Skoru):')
print(accuracy_score(Y_validation, predictions))
print('**************************************')
print('Confusion Matrix(Hata Matrisi):')
print(confusion_matrix(Y_validation, predictions))
print('**************************************')
print('classification_report(Sınıflandırma Raporu):')
print(classification_report(Y_validation, predictions))
```

```
Accuracy Score(Doğruluk Skoru):
0.9443241422453202
**************************************
Confusion Matrix(Hata Matrisi):
[[11254   621]
 [  536  8370]]
**************************************
classification_report(Sınıflandırma Raporu):
                        precision    recall  f1-score   support

neutral or dissatisfied       0.95      0.95      0.95     11875
             satisfied        0.93      0.94      0.94      8906

              accuracy                           0.94     20781
             macro avg        0.94      0.94      0.94     20781
          weighted avg        0.94      0.94      0.94     20781
```

Şekil 3.34 - Karar Ağacı Modelinin Çeşitli Fonksiyonlarla Değerlendirilmesi

Hata matrisi, veri setindeki var olan durum ile sınıflandırma modelinin doğru ve yanlış tahminlerinin sayısını tablo olarak göstermektedir. Hata matrisi, kaç kişi memnunken model memnun yani doğru değeri verdiğini ya da kaç kişi memnun değil yani yanlış değeri verdiğini detaylı olarak göstermektedir. Aynı şekilde tam tersi durum da aktarılmaktadır. Kısacası TRUE POSİTİVE (Gerçek doğru), TRUE FALSE (Gerçek negatif), FALSE POSİTİVE (Yanlış doğru) ve FALSE NEGATIVE (Yanlış negatif) değerleri matrisde görülmektedir ve daha iyi anlaşılması için hata matrisinin görseli Şekil 3.35'de aktarılmaktadır. Şekil 3.34'deki hata matrisinin ilk satırı incelendiğinde toplamda (11254 +621) 11875 kişinin memnun olmadığı ya da nötr olduğu görülmektedir. Ayrıca model 11875 kişiden 11254'ünün memnun olmadığını ve kalan 621 kişinin memnun olduğunu göstermektedir. Buradan çıkarılacak sonuç, model memnun olmayan 621 kişi için doğru değeri bulamamış ve memnun olarak göstermiştir.



Şekil 3.35 - Hata Matrisi Diyagramı

"F1-skoru, bir sınıflandırıcının Kesinlik (Precision) ve Duyarlılık (Recall) değerlerini harmonik ortalamalarını alarak tek bir metrikte birleştirmektedir. İki sınıflandırıcının performansını karşılaştırmak için kullanılmaktadır. A sınıflandırıcısının daha yüksek duyarlılık oranına ve B sınıflandırıcısının daha yüksek kesinliğe sahip olduğunu varsayalım. Bu durumda, hangisinin daha iyi sonuçlar ürettiğini belirlemek için her iki sınıflandırıcı için F1 skorları kullanılabilir." [12].

Bir sınıflandırma modelinin F1-skoru aşağıdaki gibi hesaplanır:

$$F_1 = 2 * \frac{precision * recall}{precision + recall}$$

Şekil 3.34'de classification report (sınıflandırma raporu) fonksiyonun çıktısı olarak hem memnun değil ya da nötr hem de memnun olma ikili karar durumları için f1 skorları üretilmiştir. Model memnun değil ya da nötr kişiler için %95 oranında doğru karar üretebilirken memnun olan kişiler için %94 oranında doğru karar verebilmiştir. Model memnun olmayan ya da nötr kişiler için memnun olan kişilere göre daha doğru tahmin modeli oluşturmuştur.

4. SONUÇ

Bu çalışma, makine öğrenimi algoritmalarını ve bu algoritmalar ile verinin doğasını daha derinlemesine incelemeyi hedeflemiştir. Sınıflandırma modelleri için birçok algoritma kullanılmaktadır ama her veri seti için en iyi sonucu veren ve en iyi tahmin üretebilen algoritmanın da bulunması gerektiği vurgulanmıştır.

Bu çalışma da denetimli ve denetimsiz makine öğrenimi konularından bahsedilmiştir. Detaylı olarak denetimli makine öğrenmesi algoritmalarından sınıflandırma algoritmaları incelenmiştir. Bununla ilgili bir havayolu şirketi anketinin veri seti kullanılarak 5 farklı makine öğrenmesi algoritmaları ile model oluşturulmuş ve modellerin sonuçları analiz edilmiştir. En iyi sonucu veren algoritma ile eğitim veri seti kullanılarak tahminler üretilmiş; bu tahminleri test veri setiyle kıyaslanarak doğruluk skoru, hata matrisi ve f1 skorları bulunmuş ve bu skorlar ile modelin hangi hedef öznitelik için daha doğru çalışıp çalışmadığı belirlenmiştir.

Doğru bir model kurulumu için veri setinin ön işlemesi yapılarak verinin hazırlığının yapılması gerekliliği vurgulanmıştır. Veri setinin ve özniteliklerin daha iyi anlaşılması için veri setiyle çeşitli görselleştirmeler yapılması tavsiye edilmiştir.

Bu çalışma ile makine öğreniminin farklı bir alanda kullanımı ve kullanılmak istenen veri setiyle en iyi algoritma ve tahmin modelini bulmayı ve bu model ile en iyi oranda tahmin üretilebileceği gösterilmiştir.

KAYNAKLAR

[1] Uddin, S., Khan, A., Hossain, M., "Comparing different supervised machine learning algorithms for disease prediction," vol. 19, no. 281, 21 December 2019.

[2] B. Mahesh, "Machine Learning Algorithms -A Review," 2019.

[3] Sharma, Diksha & Kumar, Neeraj, "A Review on Machine Learning Algorithms, Tasks and Applications," vol. 6, October 2017.

[4] V. Granville, "Why Logistic Regression should be the last thing you learn when becoming a Data Scientist," TechTarget, 20 May 2018. [Online]. Available: https://www.datasciencecentral.com/why-logistic-regression-should-be-the-last-thing-you-learn-when-b/.

[5] Egwom, O.J., Hassan, M., Tanimu, J.J., Hamada, M., Ogar, O.M., "An LDA–SVM Machine Learning Model for Breast Cancer Classification," vol. 2, no. 345-358, 26 June 2022.

[6] "Linear Discriminant Analysis (LDA) in Machine Learning," Java Point, [Online]. Available: https://www.javatpoint.com/linear-discriminant-analysis-in-machine-learning.

[7] Alzubi, Jafar ; Nayyar, Anand; Akshi, Kumar;, "Machine Learning from Theory to Algorithms: An Overview," vol. 1142, no. 012012, 5 December 2018.

[8] D. Johnson, «Reinforcement Learning: What is, Algorithms, Types & Examples,» GURU99, 19 November 2022. [Çevrimiçi]. Available: https://www.guru99.com/reinforcement-learning-tutorial.html.

[9] Müller, Andreas C.; Guido, Sarah;, Introduction to Machine Learning, Sebastopol, United States of America: O'Reilly Media, Inc., 2017.

[10] J. VanderPlas, Python Data Science Handbook, Sebastopol, the United States of America: O'Reilly Media, Inc., 2017.

[11] «What Is Data Visualization? Definition, Examples, And Learning Resources,» tableau, [Çevrimiçi]. Available: https://www.tableau.com/learn/articles/data-visualization#definition.

[12] "What is the F1-score?," educative, 2022. [Online]. Available: https://www.educative.io/answers/what-is-the-f1-score.

GAZİ GELECEKTİR...