

**T.C.
SAKARYA ÜNİVERSİTESİ
BİLGİSAYAR VE BİLİŞİM BİLİMLERİ FAKÜLTESİ**

**BSM 498 BİLGİSAYAR MÜHENDİSLİĞİ BİTİRME
ÇALIŞMASI**

Müşteri Davranış Analizi ve IoB Entegrasyonu

B181210052 – Arslançan SARIKAYA

**Bölüm : BİLGİSAYAR MÜHENDİSLİĞİ
Danışman : Prof. Dr. Cüneyt BAYILMIŞ**

2022-2023 Bahar Dönemi

ÖNSÖZ

Lisans eğitimim boyunca öğrendiğim teorik bilgiler, edindiğim tecrübeler sonucu böyle bir bitirme projesini tamamlamanın haklı gururu içerisindeyim. Eğitim ve öğretim hayatım boyunca bana bu yolculukta her zaman destek olan ve yolumu aydınlatan tüm öğretmenlerime sonsuz teşekkürlerimi sunarım. Öğretmenlerimle karşılaşmamı ve onlarla çalışmamı sağlayan vizyonu ve misyonu bana kattıkları için aileme ve arkadaşlarıma minnettarım.

Üniversiteye başlamadan dört ay önce babamı kaybetmeme rağmen bana babamın yokluğunu hissettirmeyen, bu maddi ve manevi savaşta her zaman yanımda hissettiğim anneme ‘Gülnur Sarıkaya’ya’ ne kadar borçlu olduğumu da vurgulamak isterim.

Bu projede, müşteriler çeşitli kriterlere göre sınıflandırılarak anlaşılmaya ve şirketlerin, ürünleri veya kampanyalarında kişisel teklifler sunarak müşteri ihtiyaçlarına IoB teknolojilerinden faydalanarak cevap verebilmesi amaçlanmıştır.

İÇİNDEKİLER

ÖNSÖZ	ii
İÇİNDEKİLER	iii
SİMGELER VE KISALTMALAR LİSTESİ	v
ŞEKİLLER LİSTESİ	vi

BÖLÜM 1. GİRİŞ.....	1
1.1. Problemin Tanımı	1
1.2. Customer Relations Management (CRM)	2
1.2.1. CRM analitiği	2
1.2.2. Müşteri segmentasyonu	3
1.2.3. Müşteri segmentasyonu için RFM tekniği	4
1.3. Davranışların İnterneti (IoB)	4
1.3.1. IoB'nin yararlandığı teknolojiler	5
1.3.1. IoB'deki sorunlar	6
1.3.2. IoB'nin farklı alanlardaki uygulamaları	7
1.4. Makine Öğrenmesi	7
1.4.1. Makine öğrenmesinin tarihi	8
1.4.2. Makine öğrenmesi teknikleri	8
1.4.2.1. <i>Supervised (Gözetimli) öğrenme</i>	8
1.4.2.2. <i>Unsupervised (Gözetimsiz) öğrenme</i>	8
1.4.2.3. <i>Reinforcement (Takviyeli) öğrenme</i>	9
1.5. Mesaj Sıralama Sistemleri	9
1.6. Başarı Ölçütleri	10

BÖLÜM 2. PROJEDE KULLANILAN TEKNOLOJİLER VE YÖNTEMLER	11
2.1. Makine Öğrenmesinde Kullanılan Kümeleme (Clustering) Algoritmaları	11
2.1.1. K-means kümeleme algoritması	11
2.1.1.1. <i>Elbow methodu ile k-means küme sayısının belirlenmesi</i>	12
2.1.2. Hiyerarşik kümeleme algoritması	13
2.1.2.1. <i>Dendrogram ile hiyerarşik küme sayısının bulunması</i>	14
2.1.3. Apriori algoritması	15
2.1.4. K-means, hiyerarşik Kümeleme, Apriori algoritmasının karşılaştırılması	16
2.2. Veri Seti Seçimi	17
2.2.1. Veri seti	17
2.2.1.1. <i>UCI machine learning repository</i>	18
2.3. Mesaj Sıralama Sistemleri	18
2.3.1. Kafka	20

2.3.2. RabbitMQ	21
2.3.3. RocketMQ	22
2.3.4. ActiveMQ	22
2.3.5. Pulsar	24
2.3.6. Karşılaştırma sonucu	26
2.4. Mesaj Tetiğinin Seçilmesi	26
 BÖLÜM 3. Geliştirilen Sistem	 27
3.1. Geliştirilen Sistemin İşleyişi.....	28
3.2. Veritabanı Kavramsal Modeli.....	29
3.3. Yazılım Gereksinim Listesi	29
 BÖLÜM 4. Endüstriyel Boyut ve Kısıtlar	 30
4.1.1. İş zaman ve Maliyet Analizi	30
4.1.2. Veritabanı Maliyet Analizi.....	31
 BÖLÜM 5. UYGULAMA.....	 32
5.1. Segmentasyon Aşamaları.....	32
5.1.1. Müşteriler için RFM analizinin yapılması.....	32
5.1.2. Market sepet analizi.....	34
5.2. Apache Kafka Kurulumu.....	37
5.2.1. Producer ve consumer oluşturulması.....	38
5.2.1.1. <i>Kafka desteği olmayan platformdan Kafkaya sub nasıl olunur?</i>	39
5.2.2. Cloud firestore bağlantıları.....	39
5.2.2.1. <i>Python firestore bağlantısı</i>	39
5.2.2.2. <i>Flutter firestore bağlantısı</i>	39
5.3. Flutter Üzerinden GPS Sensörünün Kullanılması.....	40
5.4. Geliştirilen Flutter Android Uygulaması	41
 BÖLÜM 6. SONUÇLAR VE ÖNERİLER.....	 44
 KAYNAKLAR.....	 46
ÖZGEÇMİŞ	48

SİMGELER VE KISALTMALAR LİSTESİ

IoB	Internet of Behaviors (Davranışların İnterneti)
IoT	Nesnelerin İnterneti
CRM	Customer Relations Management (Müşteri İlişkileri Yönetimi)
RFM	Recency, Frequency, Monetary

ŞEKİLLER LİSTESİ

Şekil 1.1	RFM Üç Boyut'un Açıklama Tablosu.....	4
Şekil 2.1	K-Means Algoritması.....	12
Şekil 2.2	Elbow Grafiği ve Şekil Sayısının Bulunması.....	13
Şekil 2.3	Agglomerative ve Divisive Hiyerarşik Kümeleme.....	14
Şekil 2.4	Örnek Dendrogram Diyagramı.....	14
Şekil 2.5	Apriori Algoritması Akış Diyagramı.....	15
Şekil 2.6	Apriori Örnek Veri Tablosu.....	16
Şekil 2.7	Apriori Örnek Tarama Sonuçları.....	16
Şekil 2.8	Veri Seti.....	17
Şekil 2.9	Kafka Mimarisi.....	20
Şekil 2.10	RabbitMQ Mimarisi.....	21
Şekil 2.11	RocketMQ Mimarisi.....	22
Şekil 2.12	ActiveMQ Mimarisi.....	23
Şekil 2.13	Pulsar Mimarisi.....	24
Şekil 2.14	Mesaj Kuyruk Sistemlerinin Karşılaştırması.....	25
Şekil 2.15	Mesaj Kuyruk Sistemlerinin Test Sonuçları.....	25
Şekil 3.1	Sistemin İş Akış Diyagramı.....	27
Şekil 3.2	Sistemin Akış Şeması.....	28
Şekil 3.3	RFM Veritabanı Kavramsal Modeli.....	29
Şekil 4.1	İş Zaman Çizelgesi Maliyet Analizi.....	30
Şekil 5.1	RFM Analizi'ne Hazırlanan Tablo.....	32
Şekil 5.2	RFM Değerlerinin Grafik Üzerinde Dağılışı.....	33
Şekil 5.3	RFM Puan Aralıkları.....	33
Şekil 5.4	Müşteri Grupları.....	34
Şekil 5.5	One-Hot Encoding (OHE) Sonucu.....	35
Şekil 5.6	Ürün Kodlarının Satın Alma Sayısına Dağılışı.....	35
Şekil 5.7	Birliktelik Analizi Sonuçları.....	36
Şekil 5.8	Confidence Box Plot.....	36
Şekil 5.9	Sepet Analizi Filtreleme Sonucu.....	37

Şekil 5.10	Kafdrop Arayüzü.....	38
Şekil 5.11	Geolocator Fonksiyonu.....	40
Şekil 5.12	Uygulamaya İlk Giriş İşlemi	41
Şekil 5.13	Flutter Kampanyaların Gösterilmesi.....	42
Şekil 5.14	Uygulamaya Gelen Veri Formatı.....	42
Şekil 5.15	Flutter Kampanyalarının Gösterilememesi.....	43

BÖLÜM 1. GİRİŞ

İşletmelerin başarısı veya başarısızlığı çeşitli faktörlere bağlıdır. İşletmeler için birinci gelir kaynağı olan müşteriler maddi olmayan varlıklar olarak tanımlanabilir. İş dünyasındaki artan rekabete karşı işletmelerin müşteri kaybetmemesi ve yeni müşteriler kazanmaya çalışması gerekmektedir. Pfeifer, [1] yeni müşteriler kazanma maliyetinin mevcut müşterilerle ilişkileri sürdürme maliyetinden beş kat daha fazla olduğunu vurguladı.

“Herkes uyan tek ürün” anlayışı yerine müşteri odaklı bir strateji için pazarlamacılar, müşterileri ortak özelliği paylaşan gruplara ayırmaya ve müşteri segmentlerini coğrafi, demografik, psikografik ve davranışsal değişkenlere göre belirlemeye çalışır [2]. Bunu yaparken müşteri segmentasyonunun veri madenciliğinde kümeleme modellerinden faydalandığını söyleyebiliriz.

Recency Frequency Monetary (RFM) model satın alma davranışlarının analizine dayalı müşteri segmentasyonu için popüler bir tekniktir. Model satın alma verilerini analiz eder ve her işlemi üç boyuta göre temsil eder. RFM’de her müşteri üç boyuta göre ayrı ayrı puanlanır ve her bir boyuta göre sıralanır. Sonuç olarak benzer satın alma davranışına sahip müşteriler tespit edilip sınıflandırılır. Her sınıf, üyelerin işlem geçmişinde tipik bir satın alma modelini özetleyen 3 haneli bir RFM kombinasyonu ile temsil edilir [3].

1.1. Problemin Tanımı

Müşteri profillerinin giderek çeşitlenmesi ve tek bir kalıba sığamaması nedeniyle, müşteriler kişiselleştirilmiş çözümlere ihtiyaç duymaktadır. Bu projede, müşteriler çeşitli kriterlere göre sınıflandırılarak anlaşılmalı ve şirketlerin, ürünleri veya kampanyalarında kişisel teklifler sunarak müşteri ihtiyaçlarına IoB teknolojilerinden faydalanarak cevap verebilmesi amaçlanmıştır.

1.2. Customer Relations Management (CRM)

CRM, bir şirketin müşterileriyle olan ilişkisini yönetmek, bunları daha iyi anlamak ve müşteri memnuniyetini arttırmak amaçlı kullanılan bir stratejidir. Şirketler, müşteri verilerini toplamak, analiz etmek, onlara özel kampanyalar düzenlemek, müşteri taleplerini hızlı bir şekilde karşılamak ve daha pek çok farklı yöntem kullanarak müşterileriyle olan ilişkilerini güçlendirmeye çalışırlar. [3] CRM'in şirketlere faydaları şu şekilde özetlenebilir.

1.2.1. CRM analitiği

İşletmelerin en değerli müşterilerine ulaşabilmesi ve bu müşterilerle ilişkilerini geliştirebilmesi için analitik araçlar ve teknikler gerekir. Müşterilere uygun ürünlerin önerilmesi, farklı müşteri gruplarına kampanyalar başlatılması gibi çeşitli kararlarda veri analizi sıklıkla kullanılmaktadır. [3]

Pazarlama kararlarını gerçekleştirmek için müşteri verileri üzerinde algoritmaların ve istatistiksel yöntemlerin kullanılması çoğunlukla CRM Analitiği ile ilgilidir. Müşteri verilerini veri madenciliği teknikleriyle analiz ederek müşteriye anlayıp müşterinin hizmet kalitesi arttırılabilir. CRM Analitiği sık satın alınan ürünleri keşfetmek ya da birlikte alınan ürünleri gruplandırarak öneriler sunmak için oldukça uygun bir yöntemdir. [3]

Bu yöntem aynı zamanda satış tahmini için de kullanılabilir. Veri madenciliği kapsamında, müşteri segmentasyonu ve sınıflandırması gibi Analitik CRM görevleri için kümeleme yöntemleri yararlıdır. [3] CRM Analitiğinin sağlayabileceği avantajlar şu şekilde sıralanabilir.

Müşteri Satın Alma Davranışlarını Anlamak: CRM analitiği, müşteri satın alma davranışlarını analiz ederek, müşterilerin satın alma alışkanlıkları, tercihleri, demografik özellikleri gibi birçok faktörü belirleyerek işletmelere buna göre stratejiler oluşturma konusunda yardımcı olur.

Müşteri Sadakati Oluşturma: CRM analitiği, müşterilerin satın alma alışkanlıklarını ve tüketici davranışlarını derinlemesine analiz eder. Bu sayede, işletmeler müşterilerine daha özel bir hizmet sunarak müşteri sadakatini oluşturabilirler. [3]

Yeniden Pazarlama: CRM analizi, müşteri davranışlarına ve satın alma geçmişlerine bağlı olarak işletmelerin müşterilerle yeniden pazarlama fırsatlarını belirlemelerine yardımcı olur. [3]

Satış Arttırma: CRM analitiği, tüketici davranışlarını analiz ederek, işletmelerin ürünleri ve hizmetleri hakkında daha fazla bilgi sahibi olmalarını sağlar. Böylece işletmeler, müşterilerine daha iyi bir deneyim sunarak satışlarını artırabilirler. [3]

Müşteri Memnuniyeti: CRM analitiği, müşterilerin geri bildirimlerini toplama ve analiz etme konusunda işletmelere yardımcı olur. Böylece işletmeler müşteri memnuniyetini artırarak müşterilerinin sadakatini kazanabilirler. [3]

Pazarlama Stratejisi: CRM analitiği, işletmelerin müşterilerin ihtiyaçlarını ve tercihlerini daha iyi anlamalarına olanak tanıyarak daha doğru pazarlama stratejileri oluşturmalarına yardımcı olur. [3]

1.2.2. Müşteri segmentasyonu

Müşteri Segmentasyonu; şirketlerin kendisinden hizmet alan benzersiz kullanıcı gruplarını ihtiyaçları, tercihleri ve davranışlarına göre sınıflandırarak pazarlama stratejileri üretmesine olanak sağlayan bir pazarlama stratejisidir. Şirketler müşterilerini tatmin etmek ve rekabet avantajı elde etmek için, bu segmentler üzerinden özelleştirilmiş kampanyalar ve ürünler sunabilirler. Müşteri Segmentasyonu, şirketlerin pazarlama faaliyetlerini daha etkin ve verimli bir şekilde yönetmeye olanak sağlar. Sonuç olarak, Müşteri Segmentasyonu şirketlerin müşteriler hakkında daha fazla bilgi edinip pazarlama faaliyetlerinin başarısını arttırmak, rekabet avantajı kazanmak ve müşteri bağlılığını sağlamak için önemlidir. [3]

Birden fazla müşteri segmentasyon tekniği bulunmaktadır ve içlerinden “recency (yenilik), frequency (sıklık) ve monetary (parasal)” RFM, benzer satın alma alışkanlıklarına sahip üyelerden oluşan müşteri gruplarını tanımlayan, kullanımı kolay bir tekniktir. [3]

1.2.3. Müşteri segmentasyonu için RFM tekniği

Recency (yenilik), frequency (sıklık) ve monetary (parasal) tekniği müşterilerin satın alma modelleriyle ilgili özellikleri belirlemek için kullanılan yaygın bir tekniktir. Teknik, müşterilerini satın alma verilerini işleyerek benzer satın alma kararlarına sahip müşterileri gruplandırır. RFM davranışsal bir bölümlendirme modeli olarak sınıflandırılabilir. RFM tekniğinde, satın alma davranışları temel olarak işlemlerin zamanlamasına ve toplam tutarına bağlı olan üç boyutun bir kombinasyonu olarak temsil edilir. Özellikle satın alma geçmişi bu boyutlar üzerinde bir değerlendirme ile analiz edilir ve her müşteri bu doğrultuda derecelendirilir. Bu sayede, şirketler müşterilerini daha iyi anlayarak, pazarlama stratejilerini optimize etmek ve müşteri deneyimini iyileştirmek için daha doğru karar verebilir. RFM analizi için bahsedilen üç boyut Şekil 1.1’de tanıtılmaktadır. [3]

Ölçüt	Açıklama
Recency	Son satın alma tarihinden itibaren geçen süre
Frequency	Satın alma sıklığı
Monetary	Satın alınan ortalama parasal miktar

Şekil 1.1: RFM Üç Boyut’un Açıklama Tablosu

1.3. Davranışların İnterneti (IoB)

IoB’yi anlatırken IoT ile arasındaki düşünsel yaklaşım farkına değinirsek konunun biraz daha kolay anlaşılacağını düşünüyorum.

IoT, doğrusal bir şekilde çalışır. Evinize gitmeden önce dijital asistanınızdan evinizin sıcaklığını ayarlamasını istediğinizde IoT bunu herhangi bir veri analizi olmadan

doğrudan gerçekleştirecektir. IoB ise kullanıcının sıcaklık tercihlerini ve hangi saat aralıklarında evde olduğunu anlayarak güç tüketimini azaltmak ile ilgilenecektir. [4]

Internet of behaviors, insan davranışlarının internet aracılığıyla toplandığı, analiz edildiği ve kullanıldığı bir teknolojidir. Bu teknoloji, insanların internet kullanımı, arama terimleri, web sitelerini ziyaret ettiği süreler, sosyal medya gönderileri gibi birçok farklı veriyi toplar ve bu verileri kullanarak insanların davranışlarını anlamaya çalışır. [4]

Bu teknolojinin amacı, insanların davranışlarını anlamak ve bu davranışlarının nedenlerini araştırmaktır. Örneğin, bir kişinin internette arama yaptığı kelime ve terimler, o kişinin ne hakkında bilgi aradığını gösterebilir. Bu bilgi, o kişinin ilgi alanlarını, müşteri potansiyelini ve diğer özelliklerini belirleyebilir. [4]

Internet of behaviors teknolojisi, birçok farklı alanda kullanılabilir. Örneğin, bir şirket, müşterilerinin internette ne hakkında arama yaptıklarını öğrenerek, müşteri taleplerine daha iyi cevap verebilir. Ayrıca, bir sosyal medya şirketi, kullanıcılarının ne tür içerikleri paylaştıklarını ve hangi tür içerikleri daha fazla ilgi gördüğünü öğrenerek, kullanıcılarına daha iyi hizmet sunabilir.

1.3.1. IoB'nin yararlandığı teknolojiler

Veri Toplama: IoB teknolojisi insanların davranışlarını anlamak için insanların verilerine ihtiyaç duyar ve insanların arama terimleri, web sitelerini ziyaret ettiği süreler, sosyal medya gönderileri gibi birçok farklı veriyi toplar. [6]

Veri Analizi: Toplanan verilerin anlamlandırılması için veri analizine başvurulur. Bu işlem, yapay zekâ ve makine öğrenimi gibi teknolojilerle gerçekleştirilebilir. [6]

Veri Güvenliği: Internet of behaviors teknolojisi kullanırken, toplanan verilerin güvenliği önemlidir. Bu nedenle, verilerin korunması için güvenlik duvarı, şifreleme gibi önlemler alınabilir. [6]

Veri Depolama: Toplanan veriler, internet of behaviors teknolojisi çalışırken, veri depolama sistemlerine kaydedilir. Bu sistemler, verilerin daha sonra analiz edilmesine ve kullanılmasına olanak sağlar.[6]

1.3.1. IoB'deki sorunlar

Güvenlik: IoT cihazları, Bilgisayar korsanlarının en büyük hedeflerinden biridir. Ağımızdaki IoT cihazların varlığı, ağa sızmak için potansiyel alanların sayısını artırır. Bu tür güvenlik açıkları, IoT cihazlarının kullanımını ciddi şekilde kısıtlamaktadır. Bu, ağdaki cihazlar tarafından toplanan verilerin bütünlüğünü azaltır. Verilerin bütünlüğünden ödün verilmesi nedeniyle verilerin IoB analizi tehlikeye girebilir ve üzerinde oynanmış bir sonuca yol açabilir.[7][6]

Bant Genişliği: IoT cihazlar her zaman internete bağlı olmak zorundadır ve bağlı oldukları sürece gerçek zamanlı upload ve download işlemi yaparlar. IoT cihazlarında veri depolayıp toplu olarak yükleyebilecekleri bir bölüm yoktur. Bu yüzden düşük bant genişliği IoT cihazının gerçek zamanlı veri yüklemesini engelleyebilir. [7][6]

Uyumsuzluk: IoB cihazları arasında herhangi bir standart yoktur. Bu yüzden her cihaz aynı şekilde veri toplamayabilir. Cihazların farklı veri organizasyon türü olacaktır ve bu IoB ağının analizini yaparken oluşan veri tutarsızlıkları can sıkıcı olabilir.[6]

Donanım Sorunları: IoB cihazları [8]'de yazdığı gibi her türlü veriyi toplamak için kullanılabilir; ama bu cihazlar çevresel faktörlerin etkisine karşı savunmasızdır. Oda sıcaklığını ölçen bir sistemde sıcaklık sensörünün yanına koyulan bir kibrit parçası verilerinizin doğruluğunu ortadan kaldıracaktır. [6]

Gizlilik: IoB insan davranışlarını anlamak için veri madenciliği kullanır ve kullanıcı deneyimini iyileştirmek için daha fazla veriye ihtiyaç duyar ve bu da mahremiyet sorununu gündeme getiriyor. [6]

1.3.2. IoB'nin farklı alanlardaki uygulamaları

Pazarlama: Bir şirket, müşterilerinin internette ne hakkında arama yaptıklarını ve hangi web sitelerini ziyaret ettiklerini öğrenerek, müşteri taleplerine daha iyi cevap verebilir. Ayrıca, şirket, müşterilerinin hangi tür içerikleri daha fazla ilgi gördüğünü öğrenerek, bu tür içerikleri üretebilir ve müşterilere daha iyi hizmet sunabilir. [6]

Sağlık: Internet of behaviors teknolojisi, insanların sağlıkla ilgili davranışlarını inceleyerek, sağlık risklerini azaltmaya yardımcı olabilir. Örneğin, bir kişinin internette ne hakkında arama yaptığı, hangi web sitelerini ziyaret ettiği ve hangi sağlık uygulamalarını indirdiği öğrenilerek, o kişinin sağlık durumunun takip edilebilir ve olası sağlık sorunları önceden tespit edilebilir. [6]

Eğitim: Öğrencilerin internette ne hakkında arama yaptıkları, hangi web sitelerini ziyaret ettikleri ve hangi öğrenme kaynaklarını kullandıkları öğrenilerek, eğitim sistemlerine daha iyi hizmet verilebilir. Örneğin, bir öğrencinin hangi konuları daha fazla ilgi gösterdiği öğrenilerek, o öğrenciye daha fazla ilgi göstereceği konular hakkında dersler verilebilir. [6]

Sosyal Medya: IoB, sosyal medya şirketlerinin kullanıcılarının ne tür içerikleri paylaştıklarını ve hangi tür içeriklerin daha fazla ilgi gördüğünü öğrenmeye yardımcı olur. Bu bilgiler, sosyal medya şirketlerinin kullanıcılarına daha iyi hizmet sunabilmelerine yardımcı olur. [6]

Müşteri Profili ve Hedefleme: Bir şirket, müşterilerinin internette ne hakkında arama yaptıklarını ve hangi web sitelerini ziyaret ettiklerini öğrenerek, müşteri profillerini oluşturabilir ve müşteri hedefleme çalışmalarında kullanabilir. [6]

1.4. Makine Öğrenmesi

İnsan davranışları analiz edilip davranışların tahmin edilmesinde makine öğrenmesinden faydalanılır.

1.4.1. Makine öğrenmesinin tarihi

Makine öğrenimi terimi 1959'da bilgisayar oyunları ve yapay zeka alanında öncü ve IBM çalışanı olan Amerikalı Arthur Samuel tarafından icat edildi. 1960'larda makine öğrenimi araştırmasının temsili bir kitabı, Nilsson'un Öğrenme Makineleri hakkındaki kitabıydı ve çoğunlukla örüntü sınıflandırması için makine öğrenimi ile ilgiliydi. Model tanıma ile ilgili ilgi, 1973'te Duda ve Hart tarafından tanımlandığı gibi 1970'lerde de devam etti. 1981'de, bir sinir ağı'nın bir bilgisayar terminalinden 40 karakteri (26 harf, 10 rakam ve 4 özel sembol) tanımayı öğrenmesi için öğretim stratejilerinin kullanımına ilişkin bir rapor verildi. [5]

Tom M. Mitchell, makine öğrenimi alanında incelenen algoritmaların geniş ölçüde alıntılanan daha resmi bir tanımını yaptı: "Bir bilgisayar programının performans ölçüsü "P" ve bazı "T" görev sınıflarıyla ilgili olarak "T" görevlerindeki performansı "E" deneyimiyle iyileşiyorsa "P" ile ölçüldüğü gibi E deneyiminden öğrendiği söylenir. Makine öğreniminin söz konusu olduğu görevlerin bu tanımı, alanı bilişsel terimlerle tanımlamak yerine temelde operasyonel tanım sunar.[5]

1.4.2. Makine öğrenmesi teknikleri

Makine öğrenmesi, öğrenme yöntemine göre üç gruba ayrılır; Supervised (Gözetimli), Unsupervised (Gözetimsiz) ve Reinforcement (Takviyeli). [5]

1.4.2.1. Supervised (Gözetimli) öğrenme.

Bu öğrenme tekniğinde verilen X girdi kümesinden istenen Y çıktı kümesinin elde edilmesi için bir fonksiyon öğrenilir.[5]

1.4.2.2. Unsupervised (Gözetimsiz) öğrenme

Bu yöntem gözetimli öğrenmeden farklı olarak, verileri sebep-sonuç ya da giriş-çıkış şeklinde etiketlemeden, veri içerisinde var olan ilişkilerin ve yapılarının öğrenilmesidir. Veri örneklerinin birbirine olan uzaklıklarını, komşuluk ilişkilerini ve

yoğunluklarını kullanarak veriyle ilgili çıkarımlar yapılmasını sağlar. Bu tekniğin en önemli yaklaşımları boyut indirgeme ve kümelemedir. [5]

1.4.2.3. Reinforcement (Takviyeli) öğrenme

Bu Teknik, öznelerin bir ortamda en yüksek ödül miktarına ulaşabilmesi için hangi eylemleri yapması gerektiğiyle ilgilenen bir makine öğrenmesi yaklaşımıdır. Takviyeli öğrenme, doğru girdi/çıkı eşleşmelerinin verilmemesi ve optimal olmayan eylemlerin dışarıdan düzeltilmemesi yönleriyle gözetimli öğrenmeden ayrışır. Dahası, pekiştirmeli öğrenmede bilinmeyen uzayda keşif ile mevcut bilgiden istifade arasında bir denge kurma söz konusudur. [5]

1.5. Mesaj Sıralama Sistemleri

Bir sistemde, veri üretimi ve alıcıya gönderim işlemleriyle uğraşmak, kaynak ve alıcı sayısının artmasıyla birlikte giderek karmaşık hale gelebilir. Bu da iletişimi zorlaştırabilir. Veri üreticileri veriyi doğru kaynağa üretmek için çaba sarf ederken, alıcılar da doğru verileri almak için zaman ve kaynak kullanabilir. Bu durum sistemin verimliliğini düşürebilir. Bu nedenle veri üreticileri ve alıcıları arasında bir ara katmana ihtiyaç duyulmaktadır. Message Broker olarak adlandırılan sistemler de tam olarak bu ihtiyaca çözüm olarak ortaya çıkmıştır. Message Broker kısaca mesaj sıralama sistemi olarak tanımlanabilir. Veri üreticilerinden gelen mesajları sıraya koyar ve sırası gelen veri ileticisine ileten sistemlerdir. Eğer yaşantımızdan örnek vermek gerekirse postaneye ya da kargo şubesine benzetilebilir.

Bu projede müşteriler veri alıcı, yapay zekâ ise veri gönderici rolündedir. Veri alıcılar sistemi anlık olarak tetikleyecek ve müşteri analizi sonuçlarına göre her veri alıcısına özel öneriler message broker vasıtasıyla iletilecektir.

1.6. Başarı Ölçütleri

Bu bitirme çalışmasında müşterilerden elde edilen alışveriş alışkanlıklarını içeren veri setinden faydalanarak müşterilerin alışveriş eğilimlerinin tespiti, Müşterilerin RFM analizlerinin yapılması ve bu verilerden elde edilen sonuçlara göre mağazamızın yüksek karlılığını sağlamaya yönelik kampanyaların genel geçer olmasından ziyade kişileştirilmiş kampanyalara dönüştürülmesi amaçlanmıştır. Hazırladığımız kampanyalar ne kadar kişisel ise bu kampanyaların ulaştığı zaman aralıkları da farklılık göstermektedir. Kampanyalarımızın ne zaman ortaya çıkacağına müşterilerimizin IoT cihazlar ve sensörler ile etkileşimlerine göre karar vererek bu kampanyaları onlara iletebilirsek projemizi başarılı olarak nitelendirebiliriz.

BÖLÜM 2. PROJEDE KULLANILAN TEKNOLOJİLER VE YÖNTEMLER

Bu bölümde tasarım projesinin uygulama aşamasında hangi yöntemlerin kullanıldığı anlatılıyor olacaktır.

Bu proje kapsamında, kullanıcıların alışveriş davranışlarına göre kümeleme (clustering) analizi yapıp, müşteri davranışları analiz edilerek müşteri skorları belirlenecektir. Elde edilen sonuçlara ve işletmenin pazarlama tercihlerine uygun kampanyalar müşterilere gönderilecektir. Böylece, müşterilerin daha etkili bir şekilde kazanımı hedeflenmektedir. Bu proje, veri madenciliği ve pazarlama odaklı bir yaklaşımla müşterilerin memnuniyetini arttırmayı hedeflemektedir.

2.1. Makine Öğrenmesinde Kullanılan Kümeleme (Clustering) Algoritmaları

Kümeleme (Clustering), Denetimsiz Öğrenme (Unsupervised Learning) konseptlerinden biridir ve basitçe bir veri kümesindeki verileri benzer özelliklerine göre kümeler ayırmak olarak tanımlayabiliriz.

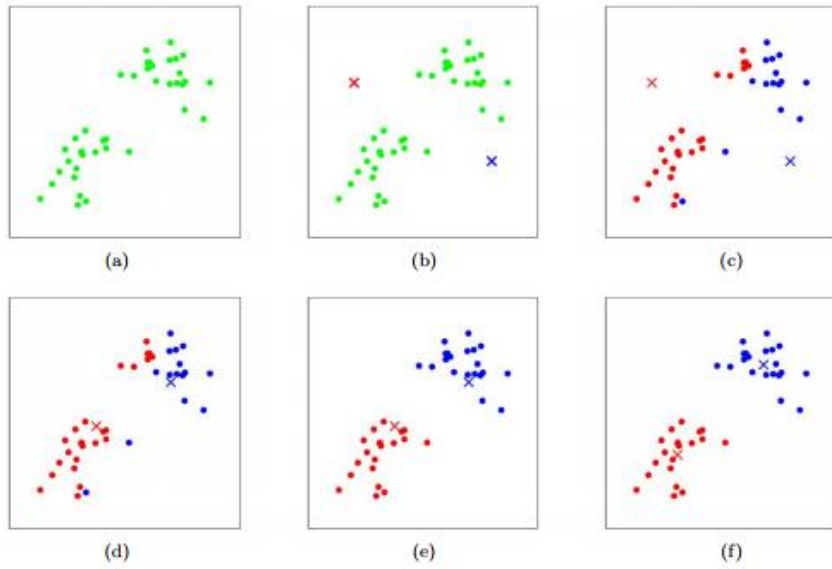
Kümeleme algoritmaları, Denetimli Öğrenme (Supervised Learning) içerisindeki sınıflandırma algoritmalarıyla karıştırılabilir. Kümeleme algoritmalarında veri setimizdeki veriler benzerliklerine göre kümelenir ve veri içerisinde herhangi bir bilgi bulunmamaktadır. Sınıflandırma algoritmalarında ise verimizin hangi tür veri içerdiği ve hangi sınıflara ayrılacağı bilinmektedir. Kısaca sınıflandırma algoritmalarında etiketler (label) bulunur fakat kümeleme algoritmalarında bu durum söz konusu değildir.

2.1.1. K-means kümeleme algoritması

K-means algoritması, doğrusal olmayan verileri kümeler ayırmak için kullanılan bir kümeleme algoritmasıdır. Bu kümeleme işlemi, noktaların birbirlerine olan

benzerliklerine göre yapılır. Benzerlikleri hesaplamak için K-means, Öklid uzaklık ölçüm yöntemini kullanır. Algoritmanın çalışma adımları aşağıdaki gibidir.[13]

- 1.Adım: İlk adımda, küme sayısı belirlenir ve rastgele yerleştirilen merkez noktaları oluşturulur.
2. Adım: Veri setimizdeki her bir veri noktası, en yakın merkez noktasına atanır.
3. Adım: Her merkez noktası kendi kümesinin ağırlık merkezine taşınır.
4. Adım: İkinci ve üçüncü adımlar ağırlık merkezleri değişmeyene kadar devam eder.[13][15]



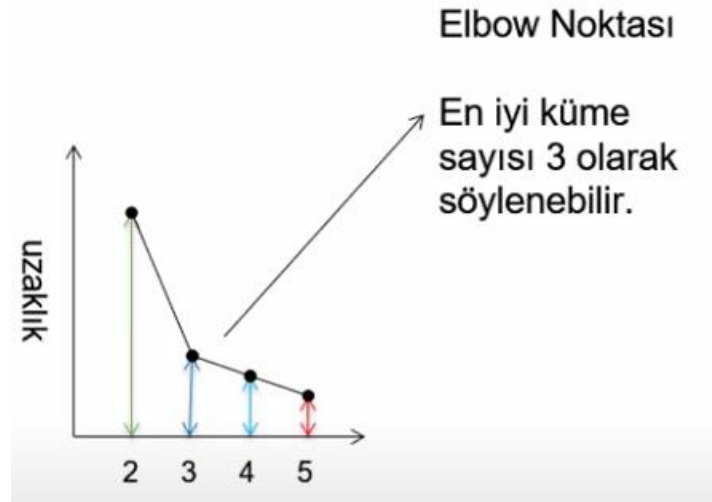
Şekil 2.1: K-means Algoritması [15]

K-means algoritması, geniş veri kümelerini küçük ve yönetilebilir parçalara ayırabilmesi ile ön plana çıkar. Ancak, K-means algoritmasında K değerinin seçilmesi önemli bir problemdir. En uygun K değerini seçmek için çeşitli yöntemler bulunmaktadır. Elbow methodu popüler yöntemlerden biridir. [9][15]

2.1.1.1. Elbow methodu ile k-means küme sayısının belirlenmesi

Elbow methodu, veri setindeki optimum küme sayısını belirlemek için kullanılan sezgisel bir methottur. Bu yöntemde kümeleme işlemi sırasında her bir kümenin açıkladığı değişkenliğin oranı hesaplanır ve küme sayısına karşı grafik haline getirilir.

Oluşturulan grafikte küme sayısı arttıkça bir kırılma meydana gelir ve bu kırılma olan kısım K değeri olarak seçilir. [9][10]



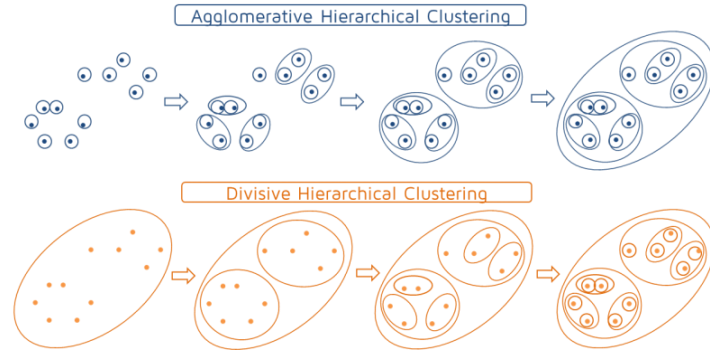
Şekil 2.2: Elbow Grafiği ve Küme Sayısının Bulunması [10]

2.1.2. Hiyerarşik kümeleme algoritması

Hiyerarşik kümeleme algoritması, adından da anlaşılacağı gibi bir kümeleme algoritmasıdır. Agglomerative (Parçadan bütüne) ve Divisive (Bütünden parçaya) olarak ikiye ayrılır.[16]

Agglomerative: Veri setimizdeki her bir veri tek başına bir küme oluşturur. Birbirine en yakın kümeler birleşerek yeni bir küme oluşturur ve bu işlem tek bir küme oluşuncaya kadar devam eder.

Divisive: Bu yaklaşım Agglomerative'in tam tersidir. Tüm veriler tek bir küme içinde ele alındıktan sonra her bir veri noktası birer küme oluncaya kadar dek bölünmeye devam eder.

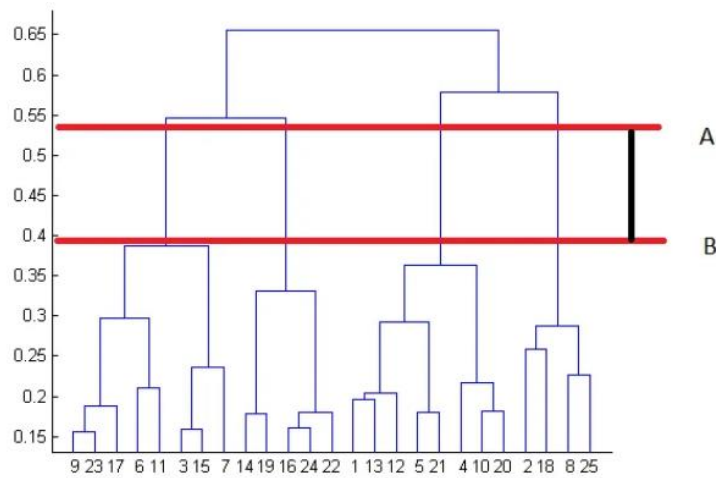


Şekil 2.3: Agglomerative ve Divisive Hiyerarşik Kümeleme [17]

Hiyerarşik kümeleme analizlerinde küme sayısını belirlemek veya elde edilen sonuçların kolaylıkla yorumlanması için dendrogramlar kullanılmaktadır.[16]

2.1.2.1. Dendrogram ile hiyerarşik küme sayısının bulunması

Dendrogram bir ağaç benzeri diyagramdır ve dalların uzunluğu iki grup arasındaki benzerliği gösterir. Üst düzeydeki dallar daha büyük ve geniş kapsamlı grupları gösterirken alt düzey dallar daha küçük ve homojen grupları gösterir. Grafikte birbirine en yakın gruplar birleştirilir ve tek grup olana kadar işlem devam eder. Optimum küme sayısı ise bir küme ile kesişmeden yatay olarak çizilebilen iki çizgi arası mesafenin maksimum olduğu noktadaki dikey küme birleşim çizgilerinin sayısıdır.[16][17]

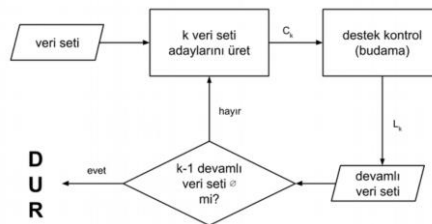


Şekil 2.4: Örnek Dendrogram Diyagramı [17]

2.1.3. Apriori algoritması

Apriori algoritması, birliktelik analizlerinin yapılması ve birliktelik kurallarının ortaya çıkarılması için en çok bilinen ve kullanılan bir algoritmadır. Bu algoritma, geniş nesne kümelerini belirlemek için verileri tarar. Geniş nesne kümelerini belirlemek için, algoritma verileri birden çok kez tarar. İlk taramada, her bir nesnenin destek seviyesi hesaplanır ve kullanıcı tarafından belirlenen minimum destek seviyesiyle karşılaştırılır. Bu şekilde her bir nesnenin geniş olup olmadığına bakılır. Ardışık taramalar, önceki taramada geniş olarak tespit edilen nesnelerden başlar ve geniş nesne kümeleri oluşturur. Bu oluşturulan geniş nesne kümelerine aday nesne kümeleri denir. Tarama işlemi tamamlandığında, hangi aday nesne kümelerinin gerçekten geniş olduğu kontrol edilir. Bir nesne kümesinin geniş olarak adlandırılabilmesi için, kullanıcı tarafından belirlenen minimum destek seviyesinin üzerinde bir destek seviyesine sahip olması gerekmektedir. Bir sonraki taramada, yine önceki taramada geniş olarak seçilen nesne kümelerinden başlanır ve veritabanının sonuna kadar bu nesne kümelerinin destekleri hesaplanır. Bu işlem, başka yeni geniş nesne kümeleri bulunamayana kadar devam eder.

Apriori algoritması, aday nesnelerin üretilmesi sırasında veritabanındaki işlemleri dikkate almadan, yalnızca bir önceki taramada geniş olduğu tespit edilen nesne kümelerini kullanarak oluşturur. Bu algoritma, geniş bir nesne kümesinin herhangi bir alt kümesinin de geniş olacağı kabulüne dayanır. Bu kabul sayesinde, k adet nesneden oluşan bir nesne kümesi, $k-1$ adet nesneye sahip geniş nesne kümelerinin birleştirilmesi ve geniş olmayan kümelerin silinmesiyle elde edilebilir. Bu birleştirme ve silme işlemi sonucunda daha az sayıda aday nesne kümesi oluşur. Bu şekilde, aday nesne kümesi üretme işlemi basitleştirilir ve hesaplama süresi azaltılır.



Şekil 2.5: Apriori Algoritması Akış Diyagramı

Örnek: Aşağıda bulunan bir alışveriş tablosu örneğindeki verilere minimum destek %25 ve güven %55 olacak şekilde Apriori algoritmasını uygulayalım.

ID	SEPET
1	Ayran, Tuz, Şeker, Simit
2	Ayran, Tuz, Simit
3	Makarna, Simit
4	Makarna, Sucuk
5	Ayran, Tuz
6	Ayran, Tuz, Makarna

Şekil 2.6: Apriori Örnek Veri Tablosu

Birinci Tarama

Ürün	Miktar	Destek
Ayran	4	%67
Tuz	4	%67
Şeker	1	%17
Simit	3	%50
Makarna	3	%50
Sucuk	1	%17

İkinci Tarama

Ürün	Miktar	Destek
Ayran, Tuz	4	%67
Ayran, Simit	2	%33
Ayran, Makarna	1	%17
Tuz, Simit	2	%33
Tuz, Makarna	1	%17
Simit, Makarna	1	%17

Üçüncü Tarama

Ürün	Miktar	Destek
Ayran, Tuz, Simit	2	%33

Şekil 2.7: Apriori Örnek Tarama Sonuçları

2.1.4. K-means, hiyerarşik Kümeleme, Apriori algoritmasının karşılaştırılması

K-means yöntemi, başlangıç noktası ve küme sayısı verildiğinde daha yüksek doğruluğa sahiptir ve büyük hacimli veri setleri üzerinde çalışabilmek için daha uygundur. Bu avantajların yanı sıra küme sayısını belirleyememesi ve başlangıç noktasının ve küme sayısının rastsal olarak seçilmesi bu yöntemin dezavantajlarıdır [13].

Hiyerarşik kümeleme'nin avantajlarına gelirse, küme sayısını belirleyebilmesi diyebiliriz. Fakat bu yöntem büyük hacimli veri setleri üzerinde çalışmaya pek uygun değildir bunun yanı sıra aykırı değerlerden (outlier) çok kolay etkilenmesi de dezavantajları arasındadır [13].

Apriori algoritması, birliktelik kurallarını keşfetmek için kullanılan bir veri madenciliği algoritmasıdır. Veri tabanında bulunan öğelerin birlikte görülme sıklığını analiz eder ve birliktelik kurallarını tespit eder. "Destek" ve "güven" gibi metrikler kullanılarak birliktelik kurallarının önemli olup olmadığı belirlenir.[24]

Bu üç yöntemin de özelliklerine baktığımızda Apriori algoritmasının market sepet analizi için en uygun yöntem olduğuna karar kıldım.

2.2. Veri Seti Seçimi

Projede kullanılacak veri setini belirlerken veri seti üzerinde yer alan müşterilerin RFM analizi yapılabilmesi hem de müşterilerin satın aldığı ürünler kümelendirilebilmelidir. Bu sayede kampanyalarımızı belirlerken ürün kümelerine ek olarak müşterilerimizin puanları da belirleyici bir unsur olabilecektir. Veri setindeki veri boyutunun fazla olması da bize tutarlılık konusunda yardımcı olacaktır.

2.2.1. Veri seti

Seçilen veri setinde 541910 satırlık veri bulunmakta ve 8 sütundan oluşmaktadır. kullanılacak veri setinin örnek 4 satırını aşağıdaki şekilde görebilirsiniz.

InvoiceNo	StockCode	Description	Quantity	InvoiceDate	UnitPrice	CustomerID	Country
536365	85123A	ITE HANGING HEART T-LIGHT HOL	6	01-12-10 8:26	2.55	17850	United Kingdom
536367	84879	5SORTED COLOUR BIRD ORNAMEN	32	01-12-10 8:34	1.69	13047	United Kingdom
536370	21791	1TAGE HEADS AND TAILS CARD GA	24	01-12-10 8:45	1.25	12583	France
536370	22631	CIRCUS PARADE LUNCH BOX	24	01-12-10 8:45	1.95	12583	France

Şekil 2.8: Veri Seti

Veri setimizi UCI Machine Learning Repository web sitesinden elde ettik [11]. Elde edilen veri setinde 2010 ve 2011 yıllarında bir markette yapılan satışların bilgileri tutulmaktadır. Veri setimizde bulunan fatura numalarının bir alışverişi temsil ettiğini düşünerek fatura numaraları ve stok numaralarını eşleştirip o müşterinin bir alışverişinde ne aldığını tespit edebiliriz. Bu eşleştirme sonrasında sepet analizi yapılarak alınan ürünler kümelenebilir ve kampanyalarımızda kullanacağımız ürün paletimizi seçmemize yardımcı olabilir. Yapılan alışverişlerin tarihlerinin, alınan

miktarını, birim fiyatlarının ve alışverişi yapan müşterilerin kayıtlarının da veri setimizin içerisinde olması müşterilerimizin hangi sıklıkla alışveriş yaptıkları ve şirketimize ne kadar nakit akışı sağladıkları hakkında bize bilgi verebilir. Bu verileri kullanarak müşterilerimizi sınıflandırabilir ve bize bağlı müşterilerimize kampanyalar sunabilir. İleriki dönemlerde bağlılıkları azalan müşterilerimize özel fırsatlar sunarak onların sadakatlerini geri kazanabiliriz.

2.2.1.1. UCI machine learning repository

UCI Machine Learning Repository, makine öğrenimi topluluğu tarafından makine öğrenimi algoritmalarının ampirik analizi için kullanılan veritabanları, etki alanı teorileri ve veri oluşturuculardan oluşan bir koleksiyondur. Arşiv, 1987 yılında David Aha ve UC Irvine'deki yüksek lisans öğrencileri tarafından bir ftp arşivi olarak oluşturuldu. O zamandan beri, tüm dünyada öğrenciler, eğitmenler ve araştırmacılar tarafından makine öğrenimi veri kümelerinin birincil kaynağı olarak yaygın bir şekilde kullanılmaktadır. Arşivin etkisinin bir göstergesi olarak, 1000'den fazla alıntı yapıldı ve bu da onu tüm bilgisayar bilimlerinde en çok alıntı yapılan 100 "makaleden" biri haline getirdi. Web sitesinin güncel hali 2007 yılında Arthur Asuncion ve David Newman tarafından tasarlanmış olup, bu proje Massachusetts Amherst Üniversitesi'nden Rexa.info ile ortaklaşa yürütülmektedir.[12]

2.3. Mesaj Sıralama Sistemleri

Mesaj sıralama sistemleri mesajları depolar ve farklı terminallerin mesajları tüketebilmesi için kuyruğa bağlanmasına izin verir. Dağıtılmış mesaj sistemi mesaj hizmetini sağlamak için çalışan çok sayıda makineden (Broker) meydana gelir. Mesajı yayınlayan elemana üretici (producer), sistemden mesaj alan elemana ise alıcı (consumer) denir. Mesajlar, farklı producer/ consumer'lara ait mesajları ayırmak için kuruklar (queue) veya konular (topic) halinde gruplandırılır. Producer tarafından gönderilen mesajların hangi topic'e gönderildiği belirtilmelidir. Mesaj sıralama sistemleri aşağıdaki dört senaryoda akış işlemeyi iyileştirebilir.

Asenkron İşleme (Asynchronous Processing): Senkron işlemede, veri üretici bir sonraki mesajı üretmeden önce alıcıdan bir geri bildirim bekler. Mesaj sıralama sistemiyle birlikte alıcıdan geri bildirim beklemeden bir sonraki mesajı üretmeye başlar. Bu durum kaynak israfının önüne geçer.

Üreticilerin ve Tüketicilerin Ayrılması: Pratikte sistem farklı arayüzler sayesinde birden fazla sistemle iletişim kurar. Üretici-tüketici sistemlerinin bağımlılığı, programcıların farklı sistemlere bağlantıları yönetmekle uğraşmalarını gerektirir. Mesaj sıralama sisteminde, üretici, farklı tüketicilerle çeşitli iletişim mekanizmalarını dikkate almaksızın yalnızca mesajı kuyruğa yayınlamaktan sorumludur.

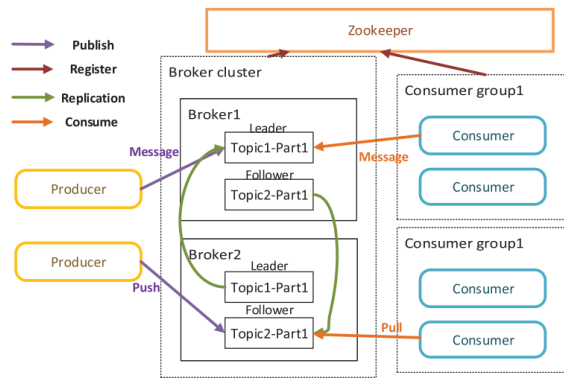
Sunucu Yükünü Azaltmak: Yüksek senkron senaryolarda, sunucular kısa sürede çok sayıda istek aldıklarında çökme yaşayabilirler. Mesaj sıralama sistemi, verileri arabelleğe almak için kullanılabilir ve tüketiciler kendi işleme verilerine göre mesaj verilerini çekebilirler. Kısaca bir mesaj sistemi, yoğun trafik durumlarında sunucuların üzerindeki yükü hafifletmek ve sunucuların çökmesini önlemek için kullanılabilir.

Günlük (Log) Hizmeti: Bazı üretim sistemlerinin günlükleri potansiyel olarak son derece büyük olabilir. Mesaj sıralama sistemi, günlük hizmeti sağlamak için kullanılabilir, Günlük verilerini almak, depolama ve iletmekle görevlidir. Ayrıca, günlük oluşturma zamanına dayalı sıkı bir sıralama garanti edebilir. Bu, günlük verilerinin doğru bir şekilde işlenmesi ve kaydedilmesini sağlar. Böylece, büyük günlük verilerinin yönetimi ve işlenmesi kolaylaştırılır ve günlük hizmetleri daha güvenilir hale getirilir.[5]

Yaygın olarak kullanılan mesaj sıralama sistemleri bulunmaktadır. Bunlara Kafka [19], RabbitMQ [20], RocketMQ [21], ActiveMQ [22] ve Pulsar [23] örnek gösterilebilir. Bu sistemler farklı tasarım mimarilerine ve karakterlerine sahiptir. Bu proje için doğru mesaj sıralama sistemini tercih edebilmek için hepsinin karakteristik özelliklerini karşılaştırmamız ve en uygun olana karar vermemiz gerekmektedir.

2.3.1. Kafka

Kafka [19], TCP protokolüne dayanan Scala tarafından geliştirilmiş, dağıtık, ayırılabilir, yedekli ve kalıcı bir günlük hizmetidir. Kafka aktif bir topluluğua sahiptir ve bu durum onun popülerliğini giderek arttırmaktadır. Kafka'da her mesaj konularına (topic) göre ayrılabilir ve her konu da kendi içinde bölümlere (partion) bölünebilir. Bu bölümler farklı brokerlara dağıtılarak sistemdeki senkronizasyon arttırılmaktadır. Buna ek olarak birden fazla alıcı bir veya daha fazla bölümü okuyarak paralel işlem kapasitesini arttırabilir. [26]



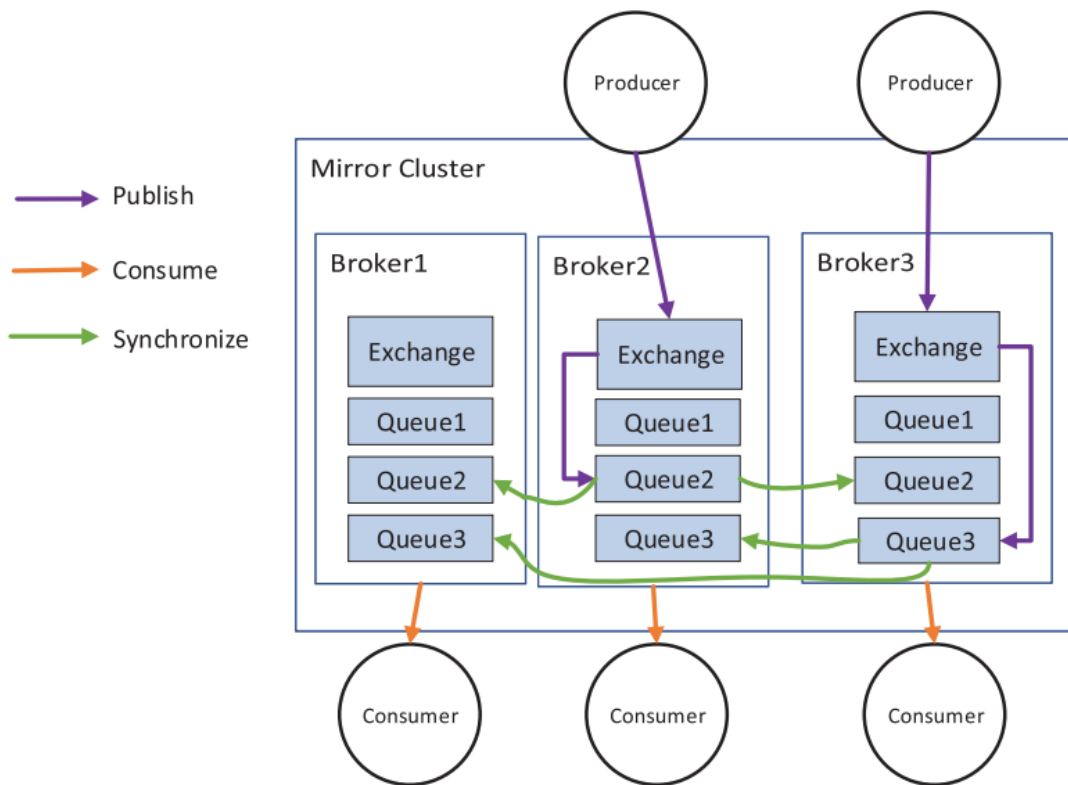
Şekil 2.9: Kafka Mimarisi

Kafka, yüksek verimliliğiyle ön plana çıkmaktadır ve bu özelliğinin üç temel nedeni bulunmaktadır.

- 1) Kafka, tekrarlanan kopyalama işlemlerini önlemek için sıfır kopya teknolojisini kullanır.
- 2) Kafka, verileri gruplandırma yöntemini kullanarak verimliliği artırır ve RPC maliyetini azaltır.
- 3) Kafka, veri sıkıştırma tekniklerini kullanarak veri iletimi verimliliğini artırır. GZIP, Snappy ve LZ4 olmak üzere üç sıkıştırma tekniğini destekler.[28]

2.3.2. RabbitMQ

RabbitMQ [20], Erlang ile geliştirilen açık kaynaklı bir AMQP protokol uygulamasıdır ve oldukça aktif bir topluluğa sahiptir. RabbitMQ, Erlang'ın sağladığı dağıtık hesaplama desteği sayesinde Zookeeper gibi üçüncü parti küme yöneticilerine ihtiyaç duymamaktadır.

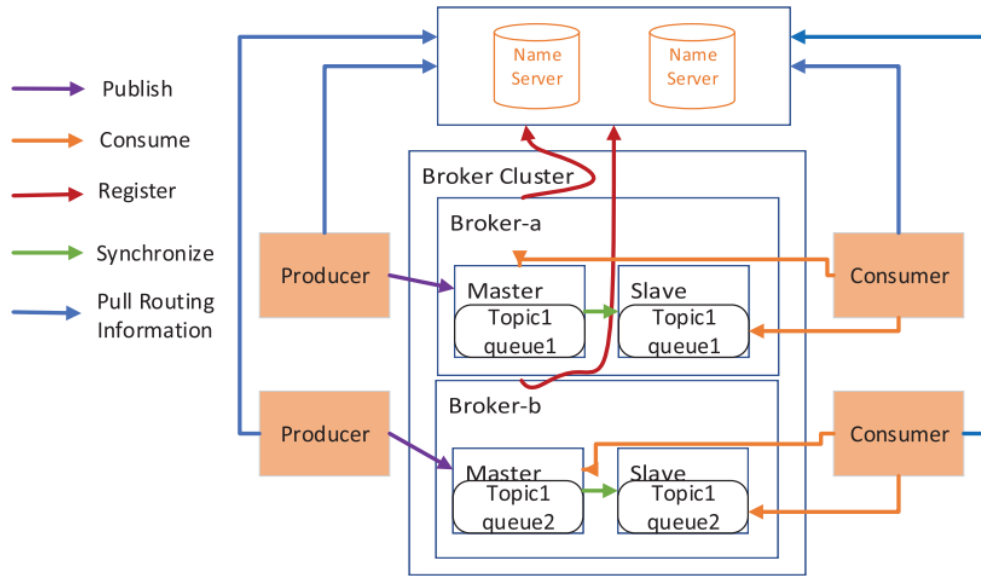


Şekil 2.10: RabbitMQ Mimarisi

RabbitMQ, öncelik, gecikme sıralaması gibi çeşitli işlevleri ve protokolleri destekleyen genel amaçlı bir mesaj brokerdir. Yönetimi kolaylaştıran bir çok yararlı eklenti sunar. Ancak, işlem mekanizmasından dolayı Kafka'dan hızlı değildir. Yayıncı Onay mekanizması, işlem maliyeti sorununu çözmek ve aynı zamanda mesaj teslimatını garanti etmek için önerilir. RabbitMQ, mesajları diske kaydedebilir ama sıralamasını garanti etmez.[18]

2.3.3. RocketMQ

RocketMQ [21], Java ile geliştirilen bir mesaj broker'ıdır ve ilk olarak Alibaba tarafından kullanılıp daha sonra açık kaynaklı genel kullanıma açılmıştır.

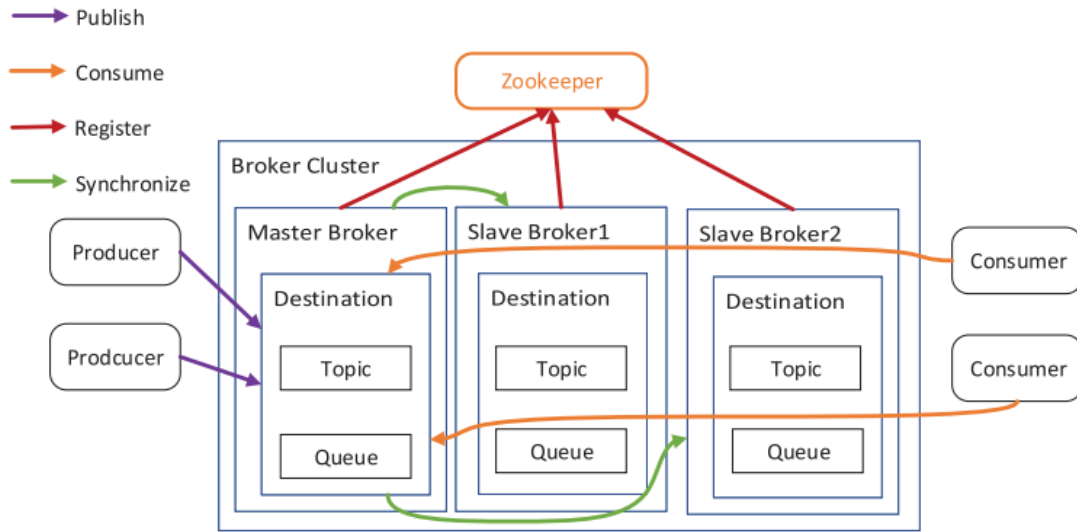


Şekil 2.11: RocketMQ Mimarisi

RocketMQ, tüm mesajları aynı fiziksel dosyada saklar ve bu merkezi depolama tasarımı, konu sayısı arttıkça verimliliği artırır. Konu/bölüm sayısının artması Kafka'nın verimliliğinde sert düşüslere neden olurken RocketMQ istikrarlı bir performans sergiler. Bu nedenle, Kafka yalnızca birkaç konu ve alıcı içeren senaryolar için daha uygunken RocketMQ daha kalabalık konu ve alıcı sayısı gerektiren senaryolarda iyi bir seçimdir. Ancak, RocketMQ topluluğu Kafka ve RabbitMQ'ya göre çok aktif değildir.[18]

2.3.4. ActiveMQ

ActiveMQ [22], Java ile yazılmış ve çeşitli iletişim protokollerini destekleyen Java Mesaj Servisi (JMS)'e sahiptir. OpenWire, STOMP, REST, XMPP ve AMQP bu protokollere örnek gösterilebilir.



Şekil 2.12: ActiveMQ Mimarisi

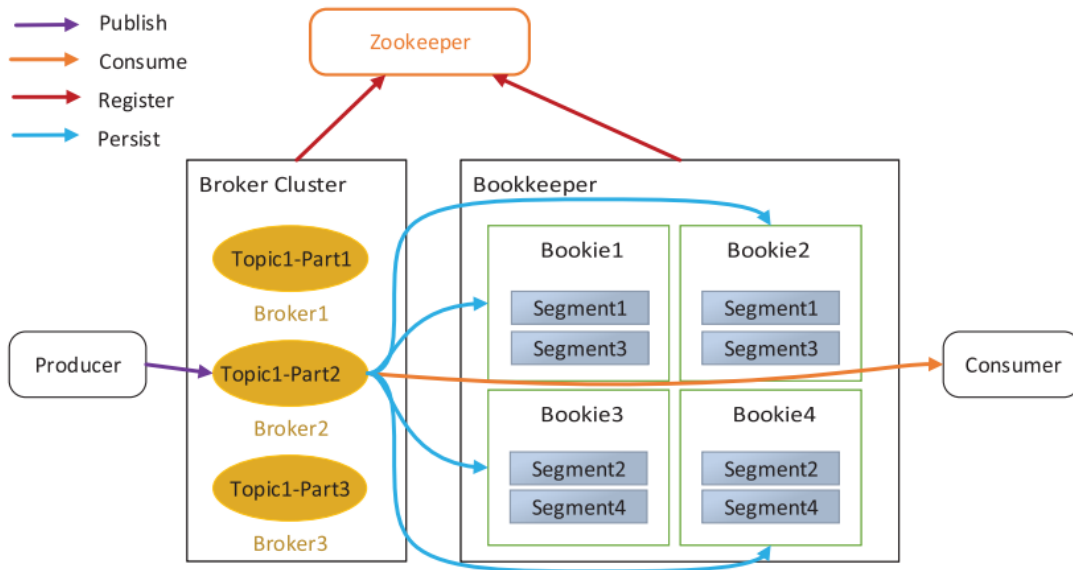
ActiveMQ, kuyruğu(queue) ve konuyu(topic) farklı bir şekilde yönetir. Bir mesaj sıralaması Exclusive Queue ayarlanarak garanti edilebilir fakat konuda(topic) herhangi bir garanti yoktur. Kuyruktaki mesajlar diskte ya da veritabanında saklanırken, topic'teki mesajlar varsayılan olarak kalıcı değildir. [18]

ActiveMQ'yu kurmak kolaydır fakat dökümantasyonu yeteri kadar detaylı değildir. Kullanıcıları çalışan örnekler vererek nasıl kullanıldığı hakkında bilgi vermez. ActiveMQ, sistemi yönetebilmek için basit bir arayüz sunar. Flink ile uyumludur fakat Spark için veri kaynağı olarak kullanılamaz. [18]

ActiveMQ, iyi ölçeklenebilir ve öncelikli sıralama, toplu iletim vb. destekler. Ancak, topluluk aktivitesi giderek azalmaktadır. Parçalama mekanizmasını desteklemez ve bu önemli bir eksiktir. Kullanıcılar gerektiğinde bunu kendi başlarına uygulamalıdır.[18]

2.3.5. Pulsar

Pulsar [23], Yahoo'da ortaya çıkan ve şu an Apache Software Foundation'ın parçası olan açık kaynaklı, büyük ölçekli, dağıtılmış bir pub-sub mesajlaşma sistemidir. Ve TCP protokolüne dayalı olarak java ile geliştirilmiştir.



Şekil 2.13: Pulsar Mimarisi

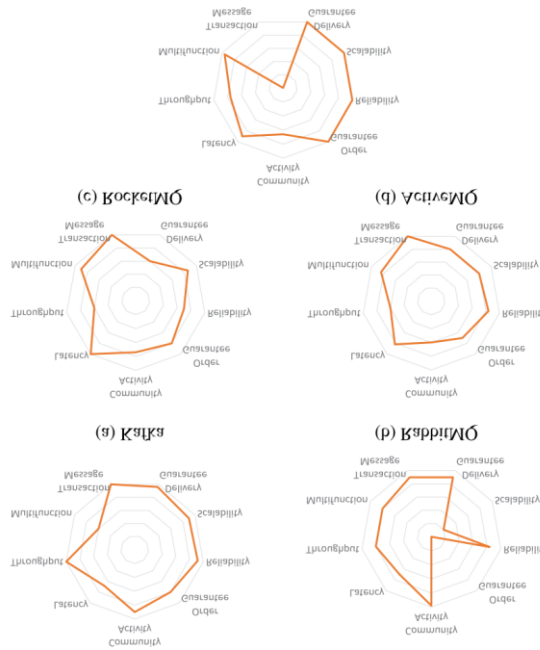
Broker-Bookie ayrılmış mimarisinin en belirgin faydası ölçeklenebilirliktir. Ayrıca gecikme kuyruğu, toplu gönderim ve öncelikli kuyruklama gibi birçok yararlı işlev sağlar. Pulsar, modern sistemlerin çok çekirdekli avantajlarından yararlanarak aynı görev isteğini aynı iş parçacığına atayarak mümkün olduğunca iş parçacıkları arasında geçiş yapmanın getirdiği maliyeti önler. Ancak, Broker yerel olarak mesaj aramak yerine başka bir Bookkeeper kümesine bağlanmak zorunda kaldığı için ağ üzerindeki trafiği artırır ve performansı etkiler.[18]

Şekil 2.10'da ilk üç satır github'ta projelerin ne kadar yıldıza, fork'a ve Issue'ya sahip olduğunu gösteriyor ve bu bilgiler bize projelerin popülerliği hakkında bilgi veriyor. Kalite garantisi özellikleri (yani 13-17 satırlar), ve bu sistemlerin desteklediği ek işlevsellikler (yani 18-20 satırlar) tabloda yer almaktadır.

System	Kafka	RabbitMQ	RocketMQ	ActiveMQ	Pulsar
Stars	17.3K	7.8K	12.8K	1.8K	6.8K
Forks	9.3K	2.9K	7K	1.2K	1.7K
Issues	764	193	169	55	793
Develop language	Scala	Erlang	Java	Java	Java
Comm. protocol	TCP	AMQP	Customized	Multiple	TCP
Cluster manager	Zookeeper	Erlang	NameServer	Zookeeper	Zookeeper
Architecture	P2P	master-slave	master-slave	master-slave	P2P
Queuing model	Pub-Sub	P2P	Pub-Sub	Pub-Sub&P2P	Pub-Sub
Consume Mode	Pull	Push/Pull	Push/Pull	Pull	Push
Persistence	Disk	Mem/Disk	Mem/Disk/DB	Mem/Disk/DB	Mem/Disk
Usability	Medium	High	High	Medium	Medium
Compatibility	Excellent	Good	Excellent	Poor	Good
Deliver guarantee	ALL	at-least/most-once	at-least-once	at-least-once	ALL
Order guarantee	Partition-order	No-order	Queue-order	Queue-order	Global-order
Reliability	High	High	High	High	High
Scalability	Good	Poor	Good	Good	Excellent
Transactions	Yes	Yes	Yes	Yes	No
Batching	Yes	Yes	Yes	Yes	Yes
Delay queuing	No	Yes	Yes	No	Yes.
Priority queuing	No	Yes	No	Yes	Yes

Şekil 2.14: Mesaj Kuyruk Sistemlerinin Karşılaştırması [18]

[18]'da yapılan çalışmada üzerinde durduğumuz beş sistem aynı ortam ve koşullarda Şekil 2.11'de gösterilen kriterlerce test edilmiştir ve yapılan testlerden çıkan sonuçları görselleştirilmiştir.



Şekil 2.15: Mesaj Kuyruk Sistemlerinin Test Sonuçları

Yapılan testlerin detaylarına [18]'den ulaşabilirsiniz. Bir sonraki sayfada ise yapılan testlerden çıkarılabilecek sonuçları inceleyebilirsiniz.

2.3.6. Karşılaştırma sonucu

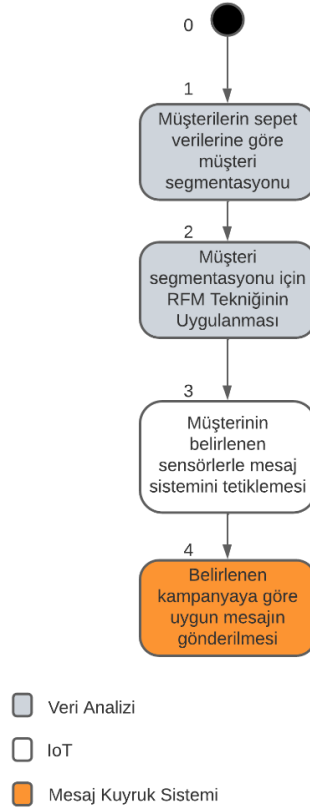
Referans aldığımız çalışmada [18], Kafka, RabbitMQ, RocketMQ, ActiveMQ ve Pulsar'ı, üretim özellikleri, kalite garantisi özellikleri, performans özellikleri ve işlevsellik özellikleri gibi özellikler karşılaştırılmıştır ve çıkan sonuçlar Şekil 2.10'da özetlenmiştir. Test sonuçlarına göre Kafka daha yüksek bir verimliliğe sahipken, RocketMQ'nun daha düşük gecikme süresine sahip olduğu ortaya çıkmıştır. Bu durum da bize tek bir iyi olmadığını kullanılacak iş senaryolarına göre farklı doğruların olduğunu göstermektedir. Düşük gecikme ile mesaj iletimine ihtiyaç duyuluyorsa RocketMQ, günlük işleme, büyük veri analizi ve akış işlemesi gibi büyük miktarda mesaj işlemek için Kafka, bazı özel işlevlere ihtiyaç duyuluyorsa Pulsar uygun bir seçenektir.[18]

2.4. Mesaj Tetiğinin Seçilmesi

Şu ana kadar müşteri verilerimizi elde edildi, işlendi ve müşteriye nasıl gönderileceği kararlaştırıldı. Geriye kalan tek ve en önemli kısım mesaj kuyruk sistemini nasıl tetiklemeyeliz? sorusuna vereceğimiz cevaptır. Tetikleme işlemi birçok farklı şekilde yapılabilir. Aklımdan geçen birkaç fikir bulunmaktadır. Örneğin büyük bir mağazamız olduğunu ve her reyon bölümünün alanını gps konumu olarak tanımladığımızı varsayalım. Müşteriler reyonlar arasında gezerken hangi tip ürünlerin arasında olduğunu tespit edebilir ve kolundaki akıllı saatten ürünlere duyduğu ilgiyi tespit edebiliriz. Elde edilen bu veri seti müşterilerin alışveriş alışkanlıklarının ve yönelimlerinin tespit edilmesinde rol gösterici olacaktır. Aklıma gelen bir diğer iş senaryosu ise alışveriş merkezinizin önünden çocuğunu okula bırakmak için arabasıyla geçen bir insanı düşünelim. Bu insanın konumunu GPS ile tespit ederek mağazamızın önünden geçmeden hemen önce kişileştirilmiş bir reklam gönderirsek bu kişinin mağazamıza girme olasılığı artacaktır. Bunun gibi çok fazla örnek verilebilir. Benim tercih edeceğim yöntem ise her müşteride halihazırda bulunan telefonlarının sensörlerini kullanmaktır. Seçtiğim bu yol hem projenin maliyetini düşürecek hem de halka mal olmasını hızlandıracaktır.

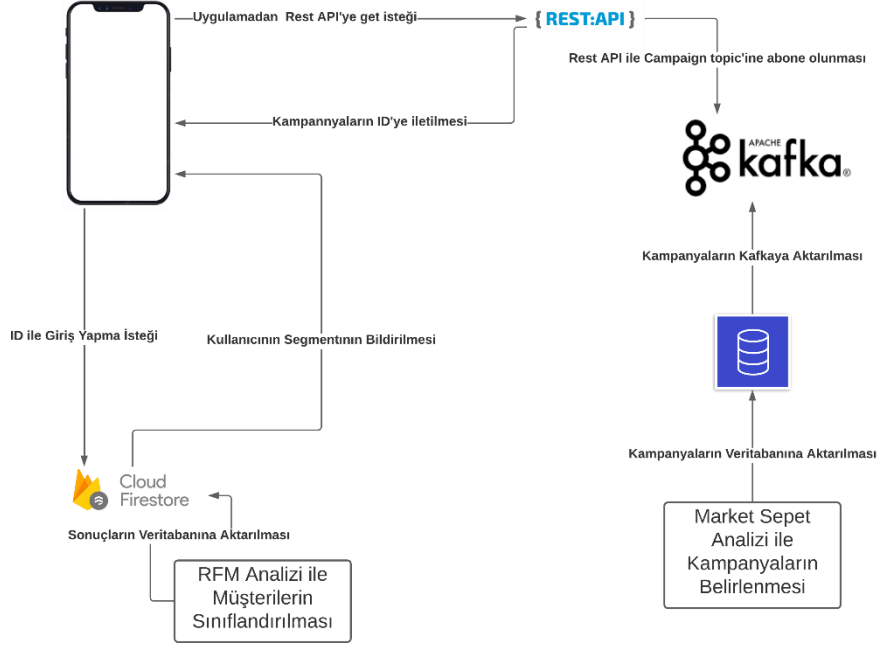
BÖLÜM 3. Geliştirilen Sistem

Projemizde tasarlanan iş senaryosunda, daha önceden toplanan müşteri verileri sayesinde müşterilerimizin alışveriş sepetlerini nasıl doldurdukları ve hangi sıklıkla ne kadar alışveriş yaptıkları verisini elimizde bulundurmaktayız. Bu verileri işleyerek müşterilerimizi daha yakından tanıyabiliriz. Örneğin bira ve fıstığı birlikte satın almaya eğilimli bir müşteriye bu konuda kampanyalar hazırlayarak ihtiyacı olmadığı halde harcama yapmasına ya da müşterimizin puanı bir önceki aya göre azalmışsa bu iki ürünle ilgili kampanyalar hazırlayak onun tekrar mağazamızda alışveriş yapmasına neden olabiliriz. Bu sistemin çalışma mantığının biraz daha anlaşılır olması açısından Şekil 3.1’de yer alan şemayla modelleyebiliriz.



Şekil 3.1: Sistemin İş Akış Diyagramı

3.1. Geliştirilen Sistemin İşleyişi

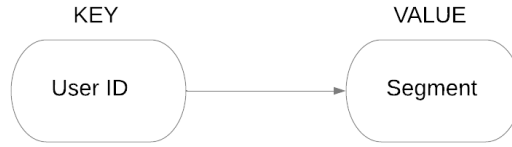


Şekil 3.2: Sistemin Akış Şeması

Geliştirilen sistemin işleyişi kullanıcıdan başlıyor. Kullanıcı veritabanına giriş yaptıktan sonra geri dönüş olarak RFM sonucunu kısaca hangi müşteri grubunda olduğunu öğreniyor. Flutter'da kafka desteği olmadığından Flask ile yazılan bir API aracılığıyla Kafka'da oluşturduğumuz Campaign topicini dinliyor. Market sepet analizi ile belirlenen kampanyalar veritabanından kafkaya aktarılıyor ve buradan da Rest API aracılığıyla kullanıcıya ulaşıyor. Bütün kampanyalar aynı kanaldan geçiyor fakat mobil uygulama üzerinde yapılan basit bir filtreleme sayesinde kullanıcılar sadece kendi segmentine gönderilen uygulamaları görebiliyor.

3.2. Veritabanı Kavramsal Modeli

Veritabanı olarak Firebase Firestore kullandık ve bu veritabanında kullanıcıların RFM sonuçlarından elde edilen segmentleriyle eşleştirilen ID'lerini sakladık. Saklanan veriler key-value ilişkisi içinde saklanmaktadır. Veri analizinde kullanılan tabloya Şekil 2.8'de ulaşabilirsiniz.



Şekil 3.3: RFM Veritabanı Kavramsal Modeli

3.3. Yazılım Gereksinim Listesi

Yazılımı geliştirirken sahip olunması gereken yazılım ve kütüphaneler şunlardır;

Python v3.9.0

Pandas kütüphanesi v1.2.4

Numpy kütüphanesi v1.23.4

Matplotlib kütüphanesi v3.6.2

Flask kütüphanesi v2.3.2

Flutter v3.10.2

Geolocator kütüphanesi v9.0.2

Cloud_firestore kütüphanesi v4.8.0

Firebase_core kütüphanesi v2.13.1

Http kütüphanesi v1.0.0

BÖLÜM 4. Endüstriyel Boyut ve Kısıtlar

TMMOB Yönetim Kurulu tarafından alınan karar uyarınca çalışan Mühendis, Mimar ve Şehir Plancıları Asgari Ücreti 2023 yılı için brüt 17.500 TL olarak tespit edilmiştir. Bu bölümde maliyetlendirmeyi brüt ücret üzerinden yapacak olup bu maliyetlendirme 2023 ilk ve ikinci çeyrek için geçerlidir.[27]

4.1.1. İş zaman ve Maliyet Analizi

Etkinlikler	Süre (Hafta)	Maliyet
Proje Öncesi Çalışma	4	17.500
Uygun Veri Setinin Bulunması	3	13.125
Ön Tasarım	1	4.375
Detaylı Tasarım	3	13.125
Prototip Oluşturma	2	8.750
Testler	1	4.375
TOPLAM	14	62.250

Şekil 4.1: İş Zaman Çizelgesi Maliyet Analizi

Tabloda verilen sürelerin toplamı proje için tanınan süreden fazladır. Bunun nedeni aynı anda birden fazla sürecin yürütülmesi ve projede tek çalıştığım için görev dağılımı yapamamamdır.

4.1.2. Veritabanı Maliyet Analizi

Sistemimizin veritabanı yaklaşık 20-megabyte boyutlarındadır ve bu verinin giderek büyüme potansiyelinden dolayı maliyetlerin önüne geçilebilmesi için herhangi bir bulut platformunda depolanmamıştır. Bu yüzden veritabanı maliyetimiz kullandığımız bilgisayarın donanımına ve kullanılan veritabanı türüne göre değişiklik göstermektedir. Bulutta depoladığımız tek unsur RFM score sonuçları ve karşılarındaki kullanıcı ID'leridir. Firebase 10 Gigabyte'ı geçen işlemlerden 1\$ almaktadır. Fakat yaptığımız işlemlerin boyutu kilobyte'lar ile zor tarif edilebilmektedir. Bu aşamada herhangi bir veritabanı maliyeti söz konusu değildir.[26]

BÖLÜM 5. UYGULAMA

5.1. Segmentasyon Aşamaları

Segmentasyon işlemlerini müşterileri daha iyi anlamak ve alışveriş alışkanlıklarını tanımak, ayrıca alışveriş sepetlerini popülerliklerine göre gruplandırmak amacıyla kullanacağız. Segmentasyon işlemine başlamadan önce veri setimizi segmentasyon için hazırlamamız gerekmektedir. Bu hazırlık aşamasında, geçersiz ve gerçekçilikten uzak değerleri ayıklıyoruz. Ayrıca, RFM analizi ve market sepeti analizi için farklı sütunlar kullanacağız. Her iki analiz için de ana veri setinden farklı iki veri seti oluşturulması gerekmektedir. Sonuçların doğruluğu ve gerçekçiliği, elde edilecek sonuçların güvenilirliği için son derece önemlidir. Analiz sürecinde kullanılan tabloları Şekil 4.1 ve Şekil 4.5'te görebilirsiniz.

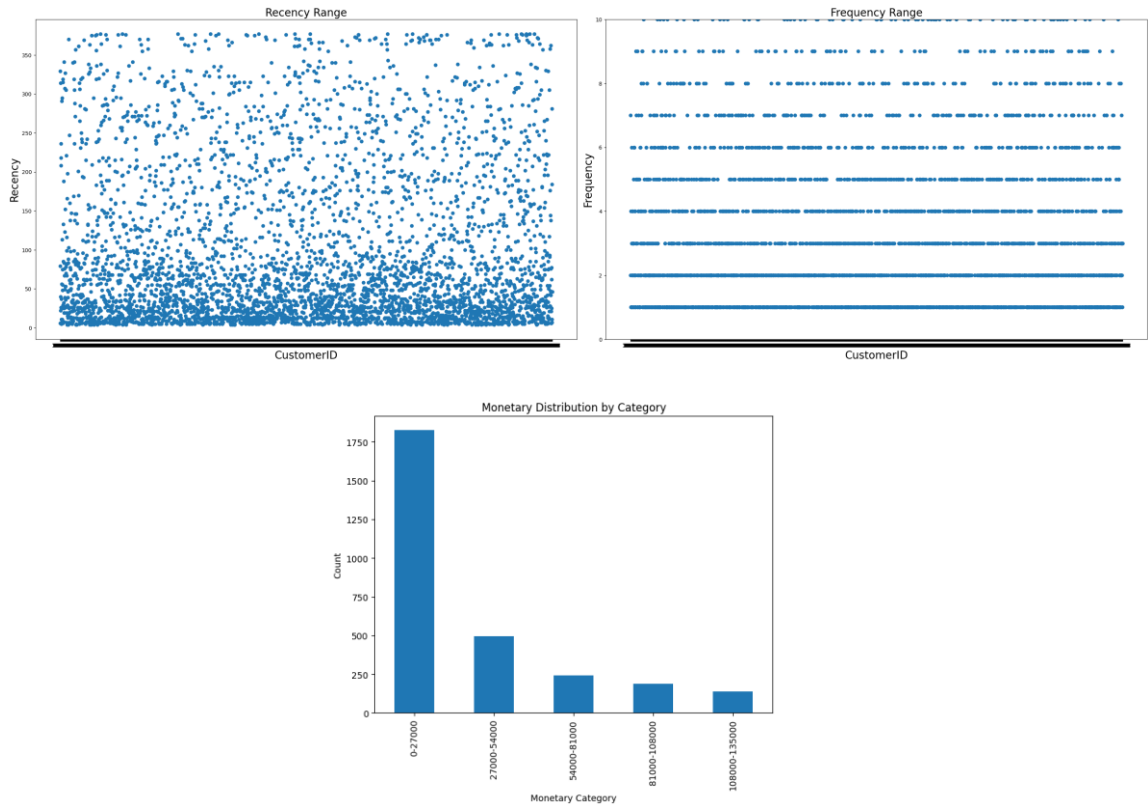
5.1.1. Müşteriler için RFM analizinin yapılması

Müşterilerimizin yenilik (recency), frekans (frequency) ve parasal(monetary) değerlerini kullanarak müşterilerimizin alışveriş alışkanlıklarını belli gruplara ayırmalı ve bu sayede müşteri yönetimimizi kolaylaştırmalıyız. Frekans ve yenilik değerlerimizi hesaplarken veri setimizde bulunan InvoiceDate sütunundan faydalanacağız. RFM analizi için veri setimizi şekillendirirken veri setimizin 2011 yılında bitmesinden dolayı bugünün tarihini 2011-12-14 olarak aldım ve bu tarihe göre Recency ve Frequency değerlerini hesapladım. Monetary değerini hesaplayabilmek için ana veri setimizde bulunan Quantity ve UnitPrice sütunlarındaki verileri kullandım. Son olarak elde edilen bütün verileri aşağıdaki gibi bir tabloya kaydettim.

CustomerID	Recency	Frequency	Monetary
12346	329	2	308734.4
12347	6	7	1182814.18
12348	79	4	418360.11
12349	22	1	381818.1
12350	314	1	12864.1
12352	40	11	1331082.2

Şekil 5.1: RFM Analizi'ne Hazırlanan Tablo

Yeni veri setimizi kullanarak veri setimizdeki verilerimizin dağılışını inceliyor ve Recency, Frequency ve Monetary değerlerimiz için puan aralıkları belirlemeye çalışıyoruz.



Şekil 5.2: RFM Değerlerinin Grafik Üzerinde Dağılışı

Grafiklerimizi inceledikten sonra puanlama aralıklarını aşağıdaki gibi seçtim. Bu aralıkların belirlenmesi bu yöntemi kullanan kurumun hangi müşteri grubuna hitap etmek istediğine ve kampanya politikasına göre değişmektedir.

Recency	Point	Frequency	Point	Monetary	Point
0 - 50	5	0 - 2	5	0 - 27000	5
50 - 100	4	2 - 4	4	27000 - 54000	4
100 - 200	3	4 - 6	3	54000 - 81000	3
200 - 300	2	6 - 8	2	81000 - 108000	2
300 - ∞	1	8 - ∞	1	108000 - ∞	1

Şekil 5.3: RFM Puan Aralıkları

Bu puan aralıkları belirlendikten sonra hedef müşteri grubuma göre bir gruplandırma yaptım ve RFM analizine göre müşteri segmentasyonum tamamlanmış oldu.

```
555: 'Champions',
554: 'Loyal_Customers',
151: 'Potential_Loyalists',
251: 'Fresh_Visitors',
515: 'Recent_Rich_Customers',
451: 'Promising',
444: 'Need_Attention',
344: 'About_toSleep',
333: 'At_Risk',
222: 'Fukara',
111: 'Lost',
```

Şekil 5.4: Müşteri Grupları

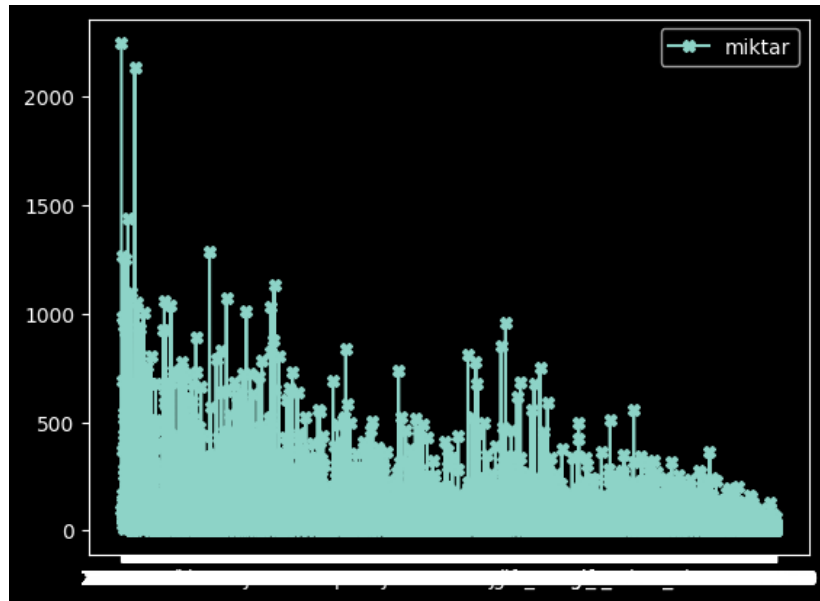
5.1.2. Market sepet analizi

Market sepet analizinde bir müşteri süt satın alıyorsa, aynı alışverişte sütun yanında ekmek olasılığı nedir? Bu tip bilgiler ışığında kampanyalarını düzenleyen bir işyeri satış oranını arttırabilir. Bu verilere ulaşabilmek için market sepet analizi gerekir. Market sepeti analizi yapabilmek için gerekli olan sütunlarımız CustomerID ve her CustomerID'nin aynı InvoiceDate'de aldığı ürünlerin listesi. Bunu yapabilmek için veri setimizi CustomerID ve InvoiceDate'e göre grpluyoruz. Gruplama işlemi bittikten hemen sonra yapay zeka'nın veri setimizi daha iyi analiz edebilmesi için verimizi Tek Değişkenli Kodlama (One-Hot Encoding (OHE))'ya göre yeniden düzenliyoruz. Tek değişkenli kodlama işlemine bir tabloda true değerini 1 ile false değerini 0 ile ifade etmek örnek verilebilir. Kodlama işlemimiz bittikten sonra tablomuz Şekil 5.5'deki gibi gözükecektir. [25]

	23287.0	37500	23167.0	22433	21945	22905	21981	21719.0	23529	22761.0	...	21755	84920	35924
0	0	0	0	0	0	0	0	0	0	0	...	0	0	0
1	0	0	0	0	0	0	0	0	0	0	...	0	0	0
2	0	0	0	0	0	0	0	0	0	0	...	0	0	0
3	0	0	0	0	0	0	0	0	0	0	...	0	0	0
4	0	0	0	0	0	0	0	0	0	0	...	0	0	0
...
25889	0	0	0	0	0	0	0	0	0	0	...	0	0	0
25890	0	0	0	0	0	0	0	0	0	0	...	0	0	0
25891	0	0	0	0	0	0	0	0	0	0	...	0	0	0
25892	0	0	0	0	0	0	0	0	0	0	...	0	0	0
25893	0	0	0	0	0	0	0	0	0	0	...	0	0	0

Şekil 5.5: One-Hot Encoding (OHE) Sonucu

Şu anki veri setini kullanarak matplotlib kütüphanesi kullanarak verimizin grafiksel ifadesini aşağıdaki gibi görebilirsiniz. [25]



Şekil 5.6: Ürün Kodlarının Satın Alma Sayısına Dağılışı

Veri setimizin analizine başlayabiliriz. Analizde Apriori algoritmasından faydalanacağız. Apriori algoritmasının ilk adımı, ürünlerin tekrar sayılarını kullanarak ürünlerin destek (support) değerlerini hesaplamaktır. [25]

Birliktelik kurallarında, öğeler arasındaki birliktelik destek ve güven kriterleriyle hesaplanır. Destek (Support) kriteri, verideki öğeler arasındaki ilişkinin ne kadar sık olduğunu belirtir. X ve Y farklı ürünler olsun. X ürünü için destek, tüm alışverişlerde

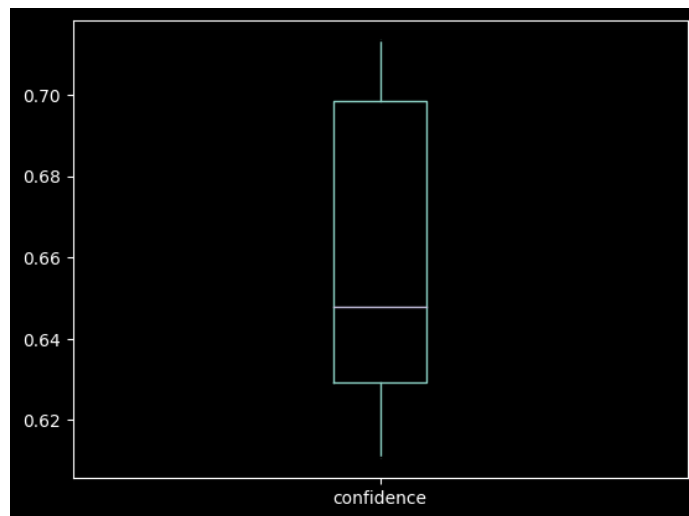
X ürününün oranını gösterir: $DESTEK(X) = X \text{ Sayısı} / \text{Toplam Alışveriş Sayısı}$. X ve Y ürünleri için destek, X ve Y'nin birlikte tüm alışverişlerde bulunma olasılığını ifade eder: $DESTEK(X, Y) = (X, Y) \text{ Sayısı} / \text{Toplam Alışveriş Sayısı}$. Güven (confidence) kriteri ise Y ürününün hangi olasılıkla X ürünü ile beraber olacağını söyler.

$GÜVEN(X, Y) = (X, Y) \text{ Sayısı} / X\text{'i İçeren Alışveriş Sayısı}$ $GÜVEN(X \Rightarrow Y) = DESTEK(X, Y) / DESTEK(X)$ Elde edilen kuralların güvenilirliği, destek ve güven değerleri ile doğru orantılıdır. Confidence değeri 0 ile 1 arasındadır. Yüksek değer kuralımızın güvenli (doğru olabilecek) olduğunu belirtir. [25]

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction	zhangs_metric
3	(22386)	(85099B)	0.043639	0.082452	0.031127	0.713274	8.650832	0.027529	3.200092	0.924760
5	(21928)	(85099B)	0.028192	0.082452	0.020082	0.712329	8.639364	0.017757	3.189573	0.909903
0	(85099F)	(85099B)	0.032054	0.082452	0.021047	0.656627	7.963788	0.018404	2.672159	0.903389
4	(21931)	(85099B)	0.042558	0.082452	0.027188	0.638838	7.748048	0.023679	2.540549	0.909648
1	(85099C)	(85099B)	0.036572	0.082452	0.022901	0.626188	7.594619	0.019886	2.454572	0.901290
2	(22411)	(85099B)	0.040589	0.082452	0.024793	0.610847	7.408556	0.021447	2.357808	0.901616

Şekil 5.7: Birliktelik Analizi Sonuçları

Tablodaki antecedents ve consequents, birlikte alınan ürünleri gösteriyor. Antecedents support toplam alışverişlerde antecedents ürününün görülme oranını, consequents support ise toplam alışverişlerde consequents ürününün toplam alışverişlerde bulunma oranını, confidence bu iki ürünün birlikte görülme olasılığını ve birliktelik kuralının ne kadar doğru olduğunu gösterir. [25]



Şekil 5.8: Confidence Box Plot

Şekil 5.8’de confidence değerlerimizi Boxplot kutu grafiği üzerinde gösterimini görmekteyiz. Bu grafiğe bakarak confidence değerlerimizin çoğunun 0.65’ten yüksek olduğunu, verilerimizin büyük bir kısmını 0.65 ve 0.70 arasındaki verilerin oluşturduğunu söyleyebiliriz.

Bu verilerer bakarak verilerimizi kendi kampanya şartlarımıza göre sınıflandırabilir ve en doğru ürünleri bulabiliriz. Ben confidence değeri 0.5’in üzerinde ve support değeri 0.025 üzerinde olan ürün setlerini kampanyama koymak istedim ve sonuç Şekil 4.9 gibi belirlendi. [25]

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction	zhangs_metric
3	(22386)	(85099B)	0.043639	0.082452	0.031127	0.713274	8.650832	0.027529	3.200092	0.924760
4	(21931)	(85099B)	0.042558	0.082452	0.027188	0.638838	7.748048	0.023679	2.540549	0.909648

Şekil 5.9: Sepet Analizi Filtreleme Sonucu

Bu sonuçlardan antecedents ve consequents’de bulunan ürün kodlarını kaydederek kampanyamda yer alacak ürün gruplarımı belirlemiş oldum. [25]

5.2. Apache Kafka Kurulumu

Apache Kafkayı resmî sitesinden indirdikten hemen sonra Kafkayı ayağa kaldıracaklarımız için ilk önce zookeeper daha sonra kafka server’ı ayağa kaldırmamız gerekiyor. Zookeeper’i çalıştırmak için indirdiğimiz dosyanın bin\Windows dizininde bir cmd dosyayı çalıştırıyoruz ve aşağıdaki komutu giriyoruz.

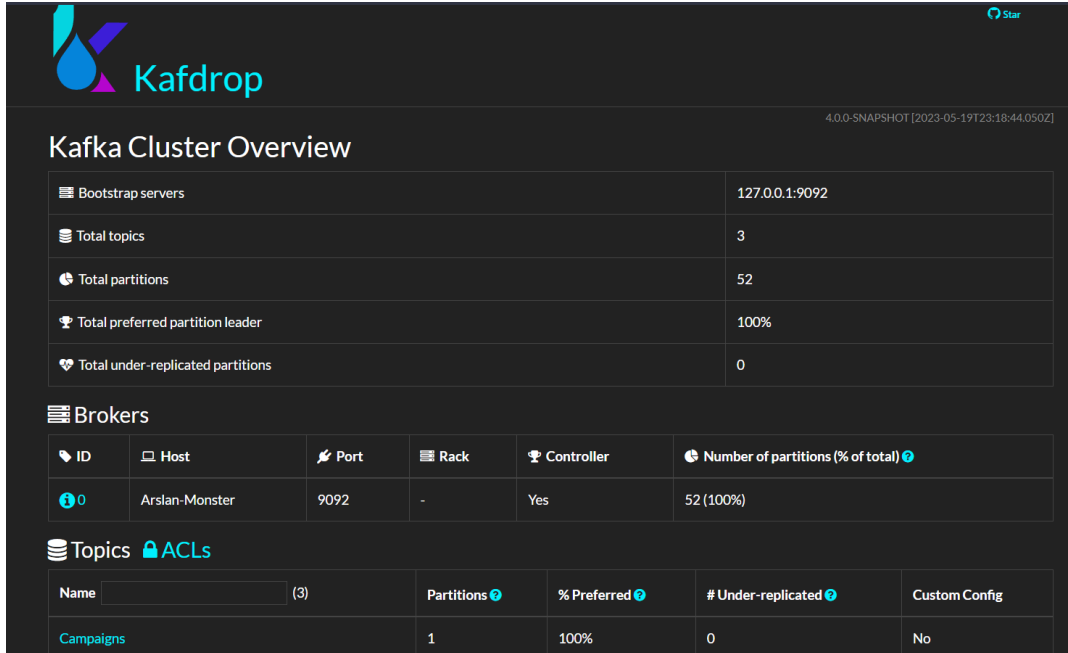
```
zookeeper-server-start.bat "D:\Kafka\kafka_2.13-3.4.0\config\zookeeper.properties"
```

Bu komutta zookeeper-server.bat dosyasına zookeeper config dosyasını parametre olarak veriyor ve başlamasını sağlıyoruz. Zookeeper’i çalıştırdıktan sonra aynı işlemleri kafka server için de yapmamız gerekiyor. Aşağıdaki komutu kullanarak bunu yapabiliriz.

```
kafka-server-start.bat "D:\Kafka\kafka_2.13-3.4.0\config\server.properties"
```

Kafka serverımız başladı fakat herhangi bir UI’a sahip değil. Kafka yönetimini kolaylaştırmak için kafdrop [24] adlı kafka server’a UI desteği sağlayan bir paket

kullandım. Bu paketin detaylarına 24 numaralı kaynaktan ulaşabilirsiniz. Üç aşamayı da bitirdiğimizde localhost:9092'den ulaşabileceğimiz bir arayüze ulaşabiliriz. [28]



The image shows the Kafdrop web interface for a Kafka cluster. The top header includes the Kafdrop logo and a version snapshot: 4.0.0-SNAPSHOT [2023-05-19T23:18:44.050Z]. The main section is titled 'Kafka Cluster Overview' and contains a table with cluster statistics:

Metric	Value
Bootstrap servers	127.0.0.1:9092
Total topics	3
Total partitions	52
Total preferred partition leader	100%
Total under-replicated partitions	0

Below the overview, there is a 'Brokers' section with a table listing the cluster's brokers:

ID	Host	Port	Rack	Controller	Number of partitions (% of total)
0	Arslan-Monster	9092	-	Yes	52 (100%)

The 'Topics' section is also visible, showing a table with columns: Name, Partitions, % Preferred, # Under-replicated, and Custom Config. The 'Campaigns' topic is listed with 1 partition, 100% preferred, 0 under-replicated, and no custom config.

Şekil 5.10: Kafdrop Arayüzü

5.2.1. Producer ve consumer oluşturulması

Serverımız hazırlandıktan sonra Apache Kafkaya kampanya detaylarımızı göndereceğimiz producer ve gönderdiğimiz verileri tüketiceğimiz consumer yapısı. Producer ve consumer yapısı için kullandığım kod örneklerini dökümanın son kısmından erişebilirsiniz. Bu kısımda kodun tamamen yazılmasından ziyade nelere dikkat etmemiz gerektiğinden bahsedeceğim. Producer kısmında standart şablona uymam yeterli oldu fakat consumer kısmı için bazı önerilerim olacak. Producer'dan yollanan verileri topic'ten birden fazla cihaz dinleyecekse bu topic'teki verileri kopyalamak için her bir cihazı farklı bir gruptan topic'e bağlamanız gerekmektedir. Kullandığımız client platformunun kafka desteği var ise introduction'ı takip ederek kolay bir şekilde bağlanabiliyoruz fakat eğer yoksa ne yapmalıyız? Bu sorunla Flutter ile uygulama geliştirip Kafka'ya bağlanmaya çalışırken karşılaştım. [28]

5.2.1.1. Kafka desteği olmayan platformdan Kafkaya sub nasıl olunur?

Döküman sonunda paylaştığım uygulamada python'da yazılmış consumera entegre ettiğim Flask kütüphanesini nasıl kullandığımı görebilirsiniz. Consumer ile gelen veriyi Flask ile yazdığım API ile birleştirerek Kafka'nın topic'ten gelen veriyi bir endpoint aracılığıyla paylaşmasını sağladım. Bu yöntemi endpoint'e request atabildiğimiz her sisteme uygulayabiliriz.

5.2.2. Cloud firestore bağlantıları

Kullanıcı segmentlerini kullanıcılarımıza ulaştırabilmek için hem python üzerinden hem de flutter üzerinden firestore'a bağlanmamız gerekiyor.[26]

5.2.2.1. Python firestore bağlantısı

Firebase Admin SDK'nın firebase-admin paketi aracılığıyla Python ile bağlantı kurmak için aşağıdaki adımları izleyebilirsiniz:

1. Firebase-admin paketini yükleyin.
2. Firebase Console'da yeni bir proje oluşturun veya mevcut bir projeye gidin. Sol menüde Firestore'a tıklayarak yeni bir veritabanı oluşturun. Ardından, projenizin Ayarlar sayfasına gidin ve Sol menüde "Service Accounts" (Hizmet Hesapları) bölümüne tıklayın. Burada, Python'a bağlanmak için kullanacağınız bir hizmet hesabı anahtarını indirebilirsiniz. Anahtar bir JSON dosyası olarak kaydedilecektir.
3. Aynı sayfada gösterilen kod parçasını kullanarak veritabanına rahatça bağlanabilirsiniz. .[26]

5.2.2.2. Flutter firestore bağlantısı

Aşağıdaki adımları uygulayarak siz de kolayca flutter firestore bağlantısını gerçekleştirebilirsiniz.

1. Cloud-firestore paketini flutter'a yükleyin.

2. Firebase Console'a gidin ve Firebase projenizi oluşturun veya mevcut bir projeye gidin. Firestore'u etkinleştirdikten sonra, projenize Flutter uygulamanızı ekleyin. Ekleme sayfasında ilerlerken size söylenen satırları uygulamanıza eklemeyi unutmayın. Eğer talimatları doğru gerçekleştirdiyseniz uygulamanız artık firestore'a bağlandı. .[26]

5.3. Flutter Üzerinden GPS Sensörünün Kullanılması

Projemizde müşterilerimiz bizim belirlediğimiz konumlara belirlenen mesafe kadar yaklaştığında onlara uygun kampanyalara erişebilecekler. Müşterilerimizin konumlarını belirleyebilmek için GPS sensörünü kullanacağız. Uygulamamıza geolocator paketini indirerek işlemlerimize başlayalım.

Bu kurulumda dikkat edilmesi gereken en önemli şey izinlerin android/app/src/main içindeki manifest dosyasına doğru eklenmiş olmasıdır. ACCESS_COARSE_LOCATION, ACCESS_FINE_LOCATION ve ACCESS_BACKGROUND_LOCATION izinlerini ekleyerek getCurrentPosition fonksiyonunu kullanabilirsiniz.[29]

```
Future getCurrentPosition() async {
  LocationPermission permission = await Geolocator.checkPermission();

  if (permission == LocationPermission.denied ||
      permission == LocationPermission.deniedForever) {
    LocationPermission asked = await Geolocator.requestPermission();
  } else {
    Position position = await Geolocator.getCurrentPosition(
      desiredAccuracy: LocationAccuracy.high);
    startLatitude = position.latitude;
    startLongitude = position.longitude;
  }
}
```

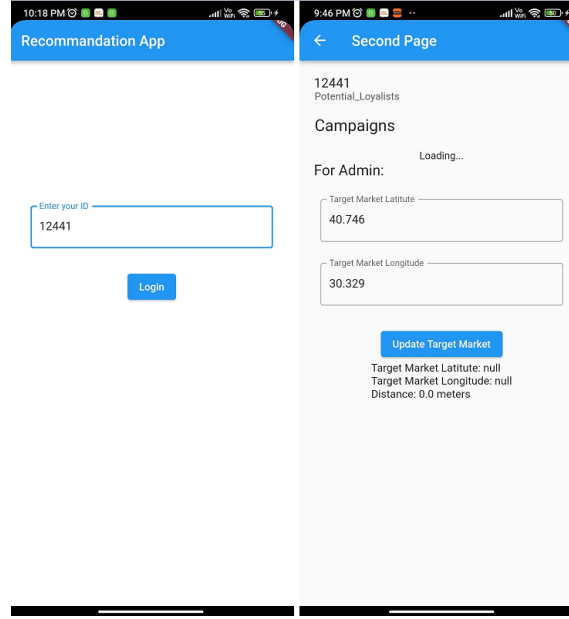
Şekil 5.11: Geolocator Fonksiyonu

Yukarıda sizinle paylaştığım fonksiyon sayesinde telefon anlık konumumuza erişebiliyor ve eğer telefonumuzda gps kullanımı için izin alınmamışsa izin isteği de yollayabiliyor. [29]

Uzaklık hesaplanırken aynı kütüphanenin içinde bulunan `distanceBetween()` fonksiyonu sayesinde şu anda bulunduğunuz konumun enlem ve boylam bilgilerini kullanarak hedeflenen mesafeye olan uzaklığınızı hesaplayabilir ve bu sonuca göre işlemlerinizi bir koşula bağlayabilirsiniz.[29]

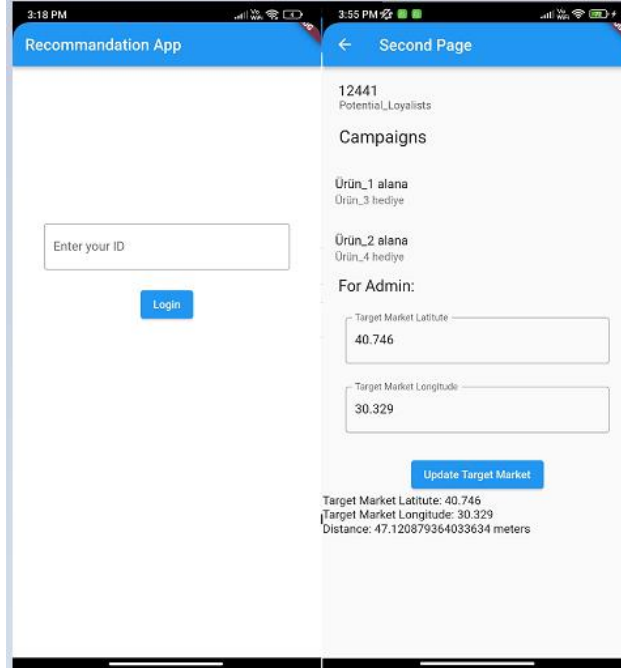
5.4. Geliştirilen Flutter Android Uygulaması

Flutter android uygulamamızın ilk sayfasından kullanıcı ID'mizi girerek login olalım. Login butonuna bastığımız an ID'mizi Cloud Firestore aracılığıyla tarayıp hangi müşteri grubunda olduğumuz bilgisi aktarıyor. Şekil 2.14'te sırasıyla 12441 ID'li müşterinin girişini ve uygulamaya girildiğinde ne ile karşılaştığını görebiliriz.



Şekil 5.12: Uygulamaya İlk Giriş İşlemi

Evet ikinci sayfamızda'da gördüğümüz gibi kullanıcımız Potansial_Loyalist müşteri grubunda fakat kampanyaları göremiyor. Bunun nedeni kampanyaların hangi hedef konuma göre gösterileceğini belirlemememizdir. Hedef konumu kendi konumuma yakın bir yer yaptım ve Şekil 2.14'teki gibi hedef konuma olan uzaklığımızı 47 metre olarak hesaplandı ve yazılımda 3000 metre olarak belirlediğim değerden daha yakın olduğu için kampanyaları görebildik.



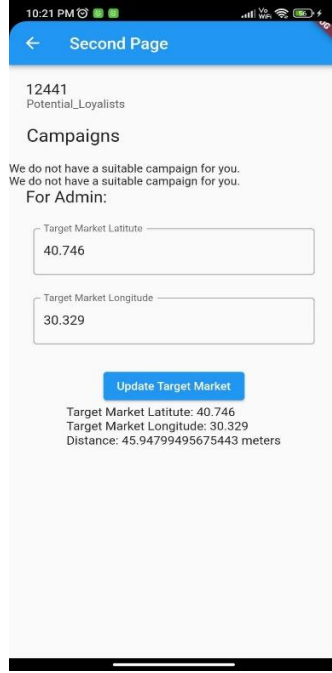
Şekil 5.13: Flutter Kampanyaların Gösterilmesi

Eğer kafka bize farklı bir grubun kampanyasını yollasaydı, biz bu grubun kampanyalarını konuma yakın olsak bile görmeyecektik; çünkü kafka bize API aracılığıyla veriyi yollarken Şekil 5.14'teki gibi müşteri grubunu da kampanyayla birlikte yolluyor ve uygulamamızın içerisinde o anki kullanıcının grubuna göre bir filtreleme işlemi uygulanıyor. Bu sayede her kullanıcı kendi grubuna özgü kampanyaları görebiliyor.

```
producer.send({
  'Campaigns',
  value=
  {
    "segment": str(segment),
    "ansequents": ansequents.values.tolist(),
    "consequents": consequents.values.tolist()
  }
})
```

Şekil 5.14: Uygulamaya Gelen Veri Formatı

Eğer yukarıda bahsettiğim senaryo gerçekleşirse uygulamaya kampanyaların geldiğini ve sadece bize gösterilmediğini anlamamız için Şekil 5.15'teki gibi bir uyarı yerleştirdim.



Şekil 5.15: Flutter Kampanyaların Gösterilememesi

BÖLÜM 6. SONUÇLAR VE ÖNERİLER

Bu projede RFM ve market sepeti analizi teknikleri kullanarak 2010 ve 2011 yılları arasında bir mağazadan yapılan alışverişlerin kayıtlarından faydalanarak müşterilerimizi tanımaya ve telefonumuzun GPS sensörünü kullanarak onlar mağazamıza yakın konumlarda bulunduklarında, onları alışverişe teşvik etmeye çalıştık. Uygulamamızı pilot hedef noktalar kullanarak test ettik, market sepeti ve RFM analizlerimizi başarılı bir şekilde gerçekleştirdik.

Bu projede GPS sensörü kullanmak oldukça önemliydi çünkü projeyi bu konuda yapılan çalışmalardan ayıran en büyük özellik yapılan veri analizi ve madencilik sonuçlarının gerçek yaşamda kullanılan bir donanım ile birlikte çalışarak hayatımıza dahil olabilmesidir.

Bu projede mobil uygulama geliştirmek için platforma seçerken öznel davrandım ve daha önce tecrübem olan bir platform olan Flutter'ı tercih ettim. Bu kararımın hem olumlu hem de olumsuz sonuçları oldu. El yatkınlığımdan dolayı hızlı sonuç alabiliyordum fakat Flutter'ın Kafka desteği olmamasından ve 3. Parti kütüphaneler bulunmamasından dolayı Kafka'ya bağlanabilmek için Python'da bulunan Flask kütüphanesi kullanarak bir Rest API yazmak zorunda kaldım ve bu durum bazı durumlarda kararlılığı olumsuz etkiledi. Kafka ile uygulama arasında API olması bazen Kafka'nın yayınladığı mesajların uygulamaya ulaşmamasına ya da geç gelmesine neden olabiliyor. Bunun yerine basit bir web arayüzü yazarak projenin daha kararlı bir versiyonunu yazabilirdim diye düşünüyorum fakat bu sefer de gps sensörünü okumak daha da zorlaşacaktı. Bir süre düşündükten sonra en mantıklı çözümü native platformlar kullanarak mobil uygulama geliştirmek olarak buldum fakat proje için ayrılan sürenin sonuna gelmiştim.

Geliştirdiğim mobil uygulamada ikinci sayfada hedeflenen konumun enlem ve boylam bilgisini girebilme imkanı verdim. Bu özellik normal şartlarda admin panelinde

bulunabilecek bir özellik ve sunumda olumsuz bir etki yapabilir fakat projenin hızlı bir şekilde test edilebilmesi açısından mantıklı bir karardır.

Projede telefon kullanmamın sebeplerinden birisi hem maliyet yükünden kendimi kurtarabilmek hem de halihazırda kullanılan bir cihaz olduğu için projemin gerçek hayata ne kadar uygun olabileceğini de görmek istedim. Hem sensör çeşitliliği hem de çok farklı çözümler sunabiliyor oluşundan dolayı bu tarz projelerde telefon kullanımını tavsiye ederim.

Projeyi başarı kriterlerine göre değerlendirirsek, Bir GPS sensörü kullanarak hedef kitlemizin konumu hakkında bilgi sahibi olarak veri işleme teknolojilerini kullanarak elde ettiğimiz kampanyaları müşterilerimize ulaştırabildik. Bu sayede bu projeyi hedefine ulaşmış ve başarılı olarak nitelendirebiliriz.

KAYNAKLAR

- [1] P. E. Pfeifer, “The optimal ratio of acquisition and retention costs”, Journal of Targeting, Measurement and Analysis for Marketing, 13(2), 179–188, 2005, Erişim Tarihi: 28.03.2023
- [2] P. Kotler, G. Armstrong, Principles of Marketing, 14th Edition; Pearson Education, USA, 2012, Erişim Tarihi 30.03.2023
- [3] <https://dergipark.org.tr/en/download/article-file/951937>, Erişim Tarihi: 27.03.2023
- [4] <https://www.techfunnel.com/information-technology/internet-of-behaviors/>, Erişim Tarihi: 12.04.2023
- [5] https://tr.wikipedia.org/wiki/Makine_%C3%B6%C4%9Frenimi, Erişim Tarihi: 10.03.2023
- [6] <https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9702450> Erişim Tarihi 12.03.2023
- [7] <https://softtek.eu/en/tech-magazine-en/user-experience-en/what-is-the-internet-of-behaviour-iob-and-why-is-it-the-future/>, Erişim Tarihi 20.03.2023
- [8] “What is IoT Data Collection?” <https://www.enertiv.com/resources/faq/what-is-iot-data-collection>, Erişim Tarihi: 18.03.2023
- [9] ”EBK-Means: A Clustering Technique based on Elbow Method and K-Means in WSN” <https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=5771aa21b2e151f3d93ba0a5f12d023a0bfcf28b>, Erişim Tarihi: 12.04.2023
- [10] Sakarya Üniversitesi Endüstri Mühendisliği 2020-Veri Madenciliği Ders Dökümanları (Dr. Öğr. Üyesi Caner ERDEN), Erişim Tarihi: 12.03.2023
- [11] <https://archive.ics.uci.edu/ml/datasets/Online+Retail#>, Erişim Tarihi: 12.04.2023
- [12] <https://archive.ics.uci.edu/ml/about.html>, Erişim Tarihi: 12.04.2023
- [13] R.J. KUO, L.M. HO ve C.M. HU, “Integration of Self-Organizing Feature Map and Kmeans Algorithm for Market Segmentation”, Computers&Operations Research, Erişim Tarihi: 12.04.2023

- [14] https://www.researchgate.net/publication/366558741_BEHAVIOURAL_CUSTOMER_SEGMENTATION_BASED_ON_MACHINE_LEARNING_ALGORITHM, Erişim Tarihi: 12.03.2023
- [15] <https://stanford.edu/~cpiech/cs221/handouts/kmeans.html>, Erişim Tarihi: 13.03.2023
- [16] <https://dergipark.org.tr/en/download/article-file/460618>, Erişim Tarihi: 13.03.2023
- [17] <https://towardsdatascience.com/machine-learning-algorithms-part-12-hierarchical-agglomerative-clustering-example-in-python-1e18e0075019>, Erişim Tarihi: 14.03.2023
- [18] “Integration of Self-Organizing Feature Map and Kmeans Algorithm for Market Segmentation”, <https://ieeexplore.ieee.org/document/9303425>, Erişim Tarihi: 03.05.2023
- [19] J. Kreps, N. Narkhede, and J. Rao, “Kafka: A distributed messaging system for log processing, Erişim Tarihi: 02.05.2023.
- [20] S. Dixit and M. Madhu, “Distributing messages using Rabbitmq with advanced message exchanges,” Erişim Tarihi: 02.05.2023.
- [21] J. Yongguo, L. Qiang, Q. Changshuai, S. Jian, and L. Qianqian, “Messageoriented middleware: A review,”.
- [22] B. Christudas, “Activemq,” in Practical Microservices Architectural Patterns. Berlin, Germany: Springer, 2019, Erişim Tarihi 04.05.2023
- [23] K. Ramasamy, “Unifying messaging, queuing, streaming and light weight compute for online event processing,” in Proc. 13th ACM Int. Conf. Distrib. Event-Based Syst., Jun. 2019.
- [24] <https://github.com/obsidiandynamics/kafdrop>, Erişim Tarihi; 15.05.2023
- [25] M. Emin Eker, Recai Oktaş, Gökhan Kayhan Apriori Algoritması ve Türkiye'deki Örnek Uygulamaları, Erişim Tarihi: 30.05.2023
- [26] firebase.google.com, Erişim Tarihi : 30.05.2023
- [27] tmmob.org.tr, Erişim Tarihi: 30.05.2023
- [28] <https://kafka.apache.org/intro>, Erisim Tarihi: 20.05.2023
- [29] <https://pub.dev/packages/geolocator>, Erisim Tarihi: 01.05.2023

ÖZGEÇMİŞ

Arslanca Sarıkaya, 16 Ağustos 1999 yılında Ordu Merkez’de doğmuştur. İlk, Orta ve lise eğitimini Ordu’da tamamlamıştır. Üniversite eğitimine, Sakarya Üniversitesi Bilgisayar Mühendisliği Bölümü’nde devam ettirmektedir.

BSM 498 BİLGİSAYAR MÜHENDİSLİĞİ TASARIMI DEĞERLENDİRME VE SÖZLÜ SINAV TUTANAĞI

KONU: Müşteri Davranış Analizi ve IoB Entegrasyonu

ÖĞRENCİLER (Öğrenci No/AD/SOYAD): B181210052 ARSLANCAN SARIKAYA

Değerlendirme Konusu	İstenenler	Not Aralığı	Not
Yazılı Çalışma			
Çalışma klavuza uygun olarak hazırlanmış mı?	x	0-5	
Teknik Yönden			
Problemin tanımı yapılmış mı?	x	0-5	
Geliştirilecek yazılımın/donanımın mimarisini içeren blok şeması (yazılımlar için veri akış şeması (dfd) da olabilir) çizilerek açıklanmış mı?			
Blok şemadaki birimler arasındaki bilgi akışına ait model/gösterim var mı?			
Yazılımın gereksinim listesi oluşturulmuş mu?			
Kullanılan/kullanılması düşünülen araçlar/teknolojiler anlatılmış mı?			
Donanımların programlanması/konfigürasyonu için yazılım gereksinimleri belirtilmiş mi?			
UML ile modelleme yapılmış mı?			
Veritabanları kullanılmış ise kavramsal model çıkarılmış mı? (Varlık ilişki modeli, noSQL kavramsal modelleri v.b.)			
Projeye yönelik iş-zaman çizelgesi çıkarılarak maliyet analizi yapılmış mı?			
Donanım bileşenlerinin maliyet analizi (prototip-adetli seri üretim vb.) çıkarılmış mı?			
Donanım için gerekli enerji analizi (minimum-uyku-aktif-maksimum) yapılmış mı?			
Grup çalışmalarında grup üyelerinin görev tanımları verilmiş mi (iş-zaman çizelgesinde belirtilebilir)?			
Sürüm denetim sistemi (Version Control System; Git, Subversion v.s.) kullanılmış mı?			
Sistemin genel testi için uygulanan metotlar ve iyileştirme süreçlerinin dökümü verilmiş mi?			
Yazılımın sızma testi yapılmış mı?			
Performans testi yapılmış mı?			
Tasarımın uygulamasında ortaya çıkan uyumsuzluklar ve aksaklıklar belirtilerek çözüm yöntemleri tartışılmış mı?			
Yapılan işlerin zorluk derecesi?	x	0-25	
Sözlü Sınav			
Yapılan sunum başarılı mı?	x	0-5	
Soruları yanıtlama yetkinliği?	x	0-20	
Devam Durumu			
Öğrenci dönem içerisindeki raporlarını düzenli olarak hazırladı mı?	x	0-5	
Diğer Maddeler			
Toplam			

DANIŞMAN: PROF. DR. CÜNEYT BAYILMIŞ

DANIŞMAN İMZASI: