

**KOCAELİ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI

YÜKSEK LİSANS TEZİ

**AMAZON WEB SERVİSİ ALTYAPISI İLE MAKİNE
ÖĞRENMESİ UYGULAMALARI**

FATMA ZEHRA SAĞLAM

KOCAELİ 2021

**KOCAELİ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

**BİLGİSAYAR MÜHENDİSLİĞİ
ANABİLİM DALI**

YÜKSEK LİSANS TEZİ

**AMAZON WEB SERVİSİ ALTYAPISI İLE MAKİNE
ÖĞRENMESİ UYGULAMALARI**

FATMA ZEHRA SAĞLAM

Prof. Dr. Nevcihan DURU
Danışman, Kocaeli Üniversitesi

Prof. Dr. Banu DİRİ
Jüri Üyesi, Yıldız Teknik Üniversitesi

Dr. Öğr. Üy. Meltem KURT PEHLİVANOĞLU
Jüri Üyesi, Kocaeli Üniversitesi

Tezin Savunulduğu Tarih: 29.06.2021

ÖNSÖZ VE TEŞEKKÜR

Bu çalışmada, tüm dünyada büyük etkisi olan COVID-19 pandemisinin başlangıcından itibaren tüm ülkelerdeki vaka ve ölüm sayılarını içeren bir veri kümesi işlenerek grafikler çıkarılmış, işlenen veriler üzerinde makine öğrenmesi çalıştırılarak tahminler üretilmiştir. Makine öğrenmesi algoritması olarak zaman serileri tahmin algoritmaları kullanılmıştır. Çalışmanın sonucunda, otomatik olarak her gün güncel verileri internetten alıp mevcut verilere ekleyen ve bu veri kümesi üzerinde makine öğrenmesi gerçekleştirip tahminler üreten bir sistem oluşturulmuştur.

Bana bu alanda çalışma fırsatı veren değerli danışman hocam Prof. Dr. Nevcihan DURU'ya, çalışma boyunca desteğini esirgemeyen sevgili aileme ve arkadaşlarıma çok teşekkür ederim.

Haziran – 2021

Fatma Zehra SAĞLAM

İÇİNDEKİLER

| | |
|--|------|
| ÖNSÖZ VE TEŞEKKÜR | i |
| İÇİNDEKİLER | ii |
| ŞEKİLLER DİZİNİ..... | iv |
| TABLolar DİZİNİ | vii |
| SİMGELER VE KISALTMALAR DİZİNİ | viii |
| ÖZET..... | ix |
| ABSTRACT..... | x |
| GİRİŞ | 1 |
| 1. BÜYÜK VERİ VE BULUT BİLİŞİM..... | 4 |
| 1.1. Giriş | 4 |
| 1.2. Büyük Veri | 4 |
| 1.3. Bulut Bilişim | 6 |
| 1.4. Bulut Bilişim'in Tarihçesi | 6 |
| 1.5. Bulut Bilişim'in Özellikleri..... | 9 |
| 1.6. Bulut Bilişim Çeşitleri..... | 11 |
| 1.7. Bulut Bilişim Servis Modelleri..... | 13 |
| 1.8. Bulut Bilişim'in Avantajları | 16 |
| 1.9. Bulut Bilişim ile İlgili Endişeler | 16 |
| 1.10. Bulut Bilişim'e Geçiş | 18 |
| 1.11. Bulut Bilişim Sağlayıcısı Örnekleri ve Karşılaştırma | 19 |
| 2. MAKİNE ÖĞRENMESİ VE TAHMİNE YÖNELİK ALGORİTMALAR..... | 21 |
| 2.1. Giriş | 21 |
| 2.2. Makine Öğrenmesi | 21 |
| 2.3. Makine Öğrenmesinin Terminolojisi | 23 |
| 2.4. Makine Öğrenmesinin Kullanılabileceği Alanlar..... | 23 |
| 2.5. En Yaygın Makine Öğrenmesi Algoritma Modelleri..... | 24 |
| 2.5.1. Gözetimli öğrenme (Supervised learning) | 24 |
| 2.5.2. Gözetimsiz öğrenme (Unsupervised learning)..... | 25 |
| 2.5.3. Takviyeli öğrenme (Reinforcement Learning) | 25 |
| 2.6. Makine Öğrenmesi Adımları..... | 27 |
| 3. ARAÇ VE HİZMETLER..... | 29 |
| 3.1. Giriş | 29 |
| 3.2. AWS Lambda | 29 |
| 3.2.1. Lambda özellikleri | 30 |
| 3.3. Amazon S3 | 31 |
| 3.3.1. Amazon S3 kullanmanın avantajları | 32 |
| 3.3.2. Amazon S3 kavramları | 32 |
| 3.4. AWS Glue | 34 |
| 3.4.1. AWS Glue'nun çalışma mantığı | 36 |
| 3.4.2. AWS Glue ile ilgili temel kavramlar | 38 |
| 3.4.3. AWS Glue'nun bileşenleri | 40 |
| 3.5. Amazon SageMaker | 42 |
| 3.5.1. İş akışı | 43 |

| | |
|--|----|
| 3.6. Amazon Forecast | 45 |
| 3.7. Amazon QuickSight | 47 |
| 3.7.1. Gösterge tabloları | 48 |
| 4. UYGULAMA ADIMLARI | 49 |
| 4.1. Giriş | 49 |
| 4.2. Ortamın Oluşturulması | 50 |
| 4.3. Verinin Elde Edilmesi | 52 |
| 4.4. Verinin İşlenmesi | 55 |
| 4.5. İş Akışının Otomatik Hale Getirilmesi | 59 |
| 4.6. Veri Görüntüleme | 62 |
| 4.7. Makine Öğrenmesi Algoritmasının Uygulanması | 68 |
| 4.8. Makine Öğrenmesi ile Üretilen Tahminlerin Değerlendirilmesi | 77 |
| 5. SONUÇLAR VE ÖNERİLER | 78 |
| KAYNAKLAR | 80 |
| KİŞİSEL YAYIN VE ESERLER | 86 |
| ÖZGEÇMİŞ | 87 |

ŞEKİLLER DİZİNİ

| | |
|--|----|
| Şekil 1.1. Statisca firmasının 2015- 2025 yılları arası internete bağlantılı cihaz sayısı grafiği..... | 5 |
| Şekil 2.1. Basit bir bulut mimarisi | 12 |
| Şekil 2.2. Son kullanıcı tarafındaki servislerin tanımları..... | 13 |
| Şekil 2.3. Bulut Bilişim servis modelleri | 13 |
| Şekil 3.1. AWS Glue Mimarisi | 39 |
| Şekil 3.2. Yarı yapılandırılmış verilerin ilişkisel veri şemalarına dönüştürülmesi | 42 |
| Şekil 3.3. Makine öğrenimi modeli için iş akışı | 44 |
| Şekil 4.1. Çalışma diyagramı | 49 |
| Şekil 4.2. Amazon S3 konsolunda bucket yaratılması..... | 50 |
| Şekil 4.3. Yaratılan bucket'a eklenen veri dosyaları..... | 50 |
| Şekil 4.4. IAM role eklenen kod parçası..... | 51 |
| Şekil 4.5. Database oluşturma ekranı..... | 51 |
| Şekil 4.6. Crawler'a isim verilmesi..... | 52 |
| Şekil 4.7. Crawler için kaynak tipi belirlenmesi | 52 |
| Şekil 4.8. Veri kaynağının belirlenmesi..... | 52 |
| Şekil 4.9. Crawler'a erişim izni olan IAM rolü seçilmesi..... | 53 |
| Şekil 4.10. Crawler'ın çalışma zamanlamasının seçilmesi | 53 |
| Şekil 4.11. Crawler çıktısının kaydedileceği veri tabanının seçilmesi..... | 53 |
| Şekil 4.12. Veri kaynağında değişiklik olması durumunda AWS Glue'nun ne yapacağını belirlenmesi..... | 54 |
| Şekil 4.13. Crawler'ı oluşturma tamamlandığı | 54 |
| Şekil 4.14. Çalışması tamamlanan crawler | 54 |
| Şekil 4.15. Database listesi | 55 |
| Şekil 4.16. Crawler'ların oluşturduğu tablolar..... | 55 |
| Şekil 4.17. Daily tablosuna kaydedilen korona veri kümesinin özellikleri | 55 |
| Şekil 4.18. Daily tablosuna kaydedilen korona veri kümesinin kolonları ve tipleri | 55 |
| Şekil 4.19. Job oluştururken isim verilmesi, IAM rolü ve kod tipi seçilmesi, neyi çalıştıracağını belirlenmesi..... | 56 |
| Şekil 4.20. Job oluştururken script'e isim verilmesi ve script'in yerinin belirtilmesi..... | 56 |
| Şekil 4.21. Job oluştururken donanım tipi, aynı anda çalışan job sayısı ve zaman aşımı değerlerinin belirlenmesi..... | 57 |
| Şekil 4.22. Job script'inde kaynakların tanımlandığı, katalogdan veri kaynağının yaratıldığı, veri kaynağındaki bozuk satır ve sütunların düzeltildiği kod parçası | 58 |
| Şekil 4.23. Çıktıların belirlenen path'lere kaydedilmesi..... | 58 |
| Şekil 4.24. Amazon S3'teki bucket'a job'ın çalışmasıyla eklenen dosyalar | 59 |
| Şekil 4.25. Amazon S3'teki bucket'a job'ın çalışmasıyla eklenen dosyanın içeriği..... | 59 |

| | |
|---|----|
| Şekil 4.26. Amazon S3'teki bucket'a job'ın çalışmasıyla parquet tipinde kaydedilen veri dosyaları | 59 |
| Şekil 4.27. Workflow tasarımı | 60 |
| Şekil 4.28. Talep geldiğinde çalışan "crawler" adındaki trigger | 60 |
| Şekil 4.29. Crawler trigger'ı çalışınca tetiklenen "corona-crawler" | 61 |
| Şekil 4.30. Corona-crawler'ın işinin başarıyla tamamlanması koşuluyla çalışan "glue-job" adlı trigger | 61 |
| Şekil 4.31. Glue-job trigger'ı çalışınca tetiklenen "corona-glue-job" | 61 |
| Şekil 4.32. Corona-glue-job'ın işinin başarıyla tamamlanması koşuluyla çalışan "glue-output-crawler" adlı trigger..... | 61 |
| Şekil 4.33. Glue-output-crawler trigger'ı çalışınca tetiklenen "glue-output" adlı crawler..... | 61 |
| Şekil 4.34. Workflow elemanlarının çalışma sırasındaki görüntüsü..... | 62 |
| Şekil 4.35. QuickSight'ta veri kaynağı yaratılması | 62 |
| Şekil 4.36. QuickSight'ta yaratılan veri kaynağının ilgili tablosunun seçilmesi | 63 |
| Şekil 4.37. Veri kaynağının sorgulanma biçiminin seçilmesi..... | 63 |
| Şekil 4.38. Veri kaynağı yaratma tamamlandıktan sonraki QuickSight ekranı | 63 |
| Şekil 4.39. QuickSight'ta sürükle-bırak ile grafik oluşturma | 64 |
| Şekil 4.40. SQL Query sayfası..... | 64 |
| Şekil 4.41. QuickSight'ta oluşturulan bir grafik | 64 |
| Şekil 4.42. Birer dosyada eklenen 9 günlük covid19 veri kümesi | 65 |
| Şekil 4.43. 01.01.2021 – 19.01.2021 tarihinde korona virüs sebebiyle en çok ölüm, vaka ve iyileşme sayılarının ülkelere göre grafiği | 65 |
| Şekil 4.44. 2020 Ocak – 2021 Ocak tarihleri arasında korona virüs sebebiyle en çok ölüm, vaka ve iyileşme sayılarının ülkelere göre grafiği..... | 66 |
| Şekil 4.45. 2020 Ocak – 2021 Ocak tarihleri arasında korona virüs sebebiyle en çok ölüm ve iyileşme sayılarının ülkelere göre grafikleri..... | 66 |
| Şekil 4.46. 2020 Ocak – 2021 Mayıs tarihleri arasında sekiz ülkeye ait yeni vaka sayılarının grafiği..... | 67 |
| Şekil 4.47. 2020 Ocak – 2021 Mayıs tarihleri arasında sekiz ülkeye ait yeni vaka sayılarının ülkelerin popülasyonuna oranını gösteren grafik..... | 67 |
| Şekil 4.48. 2020 Ocak – 2021 Mayıs tarihleri arasında sekiz ülkeye ait vaka sayılarının kümülatif toplamını gösteren grafik | 68 |
| Şekil 4.49. 2020 Ocak – 2021 Mayıs tarihleri arasında sekiz ülkeye ait vaka sayılarının kümülatif toplamının ülkelerin popülasyonuna oranını gösteren grafik..... | 68 |
| Şekil 4.50. Veri kümesinin kolon isimleri ve ilk satırları | 69 |
| Şekil 4.51. Veri kümesinin son satırları | 69 |
| Şekil 4.52. Lambda fonksiyonunun birinci kısmı | 70 |
| Şekil 4.53. Lambda fonksiyonunun ikinci kısmı | 70 |
| Şekil 4.54. Lambda fonksiyonunu zamanlayıcı ile çalıştırma | 70 |
| Şekil 4.55. SageMaker'daki oluşturulan not defteri..... | 71 |
| Şekil 4.56. Kütüphanelerin tanımlanması | 71 |
| Şekil 4.57. Verinin veri çerçevesine eklenmesi | 72 |
| Şekil 4.58. Verinin görüntülenmesi | 72 |
| Şekil 4.59. Tarih özelliğinin formatının değiştirilmesi | 72 |
| Şekil 4.60. Özelliklerin sayısal değerlere çevrilmesi | 73 |
| Şekil 4.61. Verinin makine öğrenmesi için eğitim ve test verisine ayrılması..... | 73 |
| Şekil 4.62. IAM rol işlemi ve hedef klasör oluşturma | 73 |

| | |
|---|----|
| Şekil 4.63. Veri kümesinin RecordIO formatına dönüştürülmesi..... | 74 |
| Şekil 4.64. Zaman serisi algoritmasının sonucu..... | 74 |
| Şekil 4.65. Modelin eğitilmesi | 75 |
| Şekil 4.66. Forecast yöntemleri..... | 75 |
| Şekil 4.67. Belirtilen tarih aralığına ait tahmin..... | 76 |
| Şekil 4.68. Tahminler ve gerçek veri grafiği | 76 |
| Şekil 4.69. Tahminlerin yakınlık değeri..... | 77 |



TABLolar DİZİNİ

| | |
|--|----|
| Tablo 2.1. En çok kullanılan bulut platformları..... | 19 |
|--|----|



SİMGELER VE KISALTMALAR DİZİNİ

Kısaltmalar

| | |
|---------|---|
| AHHP | : APRANET Host-to-Host Protocol (APRANET Ana Bilgisayardan Ana Bilgisayara Protokolü) |
| AMQT | : Advanced Message Queuing Protocol |
| API | : Application Programming Interface (Uygulama Programlama Arayüzü) |
| APRANET | : Advanced Research Projects Agency Network (Gelişmiş Araştırma Projeleri Dairesi Ağı) |
| AWS | : Amazon Web Service |
| COVID | : Corona Virus Disease (Korona Virüs Hastalığı) |
| CRM | : Customer Relationship Management (Müşteri İlişkileri Yönetimi) |
| CSV | : Comma Separated Values (Virgülle Ayrılmış Değerler) |
| ETL | : Extract – Transform – Load (Ayıkla – Dönüştür - Yükle) |
| IaaS | : Infrastructure as a Service (Servis Olarak Altyapı) |
| ICP | : Internet Protocol Platform (İnternet Protokolü Platformu) |
| IMDB | : Internet Movie Database (İnternet Film Veri tabanı) |
| IoT | : Internet of Things (Nesnelerin İnterneti) |
| IP | : Internet Protocol (İnternet Protokolü) |
| JSON | : JavaScript Object Notation (JavaScript Nesne Gösterimi) |
| LAN | : Local Area Network (Yerel Alan Ağı) |
| MQTT | : Message Queuing Telemetry Transport |
| NIST | : National Institute of Standards and Technology (Ulusal Standartlar ve Teknoloji Enstitüsü) |
| PaaS | : Platform as a Service (Servis Olarak Platform) |
| RGB | : Red, Green, Blue (Kırmızı, Yeşil, Mavi) |
| SaaS | : Software as a Service (Servis Olarak Yazılım) |
| SDK | : Software Development Kit (Yazılım Geliştirme Kiti) |
| SLA | : Service Level Agreement (Hizmet Seviyesi Anlaşması) |
| SQL | : Structured Query Language (Yapılandırılmış Sorgu Dili) |
| S3 | : Simple Storage Service (Basit Depolama Servisi) |
| TCP | : Transmission Control Protocol (İletim Kontrol Protokolü) |
| UWP | : Universal Windows Platform (Evrensel Windows Platformu) |
| VM | : Virtual Machine (Sanal Makine) |
| VPC | : Virtual Private Cloud (Sanal Özel Bulut) |
| WHO | : World Health Organization (Dünya Sağlık Örgütü) |
| XML | : Extensible Markup Language (Genişletilebilir İşaretleme Dili) |
| .NET | : Microsoft tarafından geliştirilen, açık İnternet protokolleri ve standartları üzerine kurulmuş bir "uygulama" geliştirme platformu. |

AMAZON WEB SERVİSİ ALTYAPISI İLE MAKİNE ÖĞRENMESİ UYGULAMALARI

ÖZET

2019 yılında hayatımıza girmiş olan korona adlı virüs, tüm dünyaya yayılarak pandemiye sebep olmuştur. Salgınla ilgili verilerin işlenmesi; alınabilecek önlemlerin belirlenmesi, olası başka salgınların önlenmesi, doğru planlama yapılabilmesi ve hayat akışını en az etkileyecek şekilde kurallar getirilebilmesi için büyük önem taşır.

Pandemi verileri, “büyük veri” olması sebebiyle, bu verinin saklanması ve işlenmesi ancak büyük veri işlemeyi sağlayan teknolojik yöntemlerle mümkündür. Ayrıca verinin çeşitli kaynaklardan, farklı veri yapılarıyla alınması ve dinamik olması, veri üzerinde işlem yapabilmeyi zorlaştırmaktadır. Bilişim dünyasında verinin büyüklüğü, verilere sürekli erişim zorunluluğu, finansal zorluklar ve güvenlik sorunları gibi birçok neden bilgi teknolojilerini dijital dönüşüme yöneltmiştir. Bulut Bilişim Teknolojisi dijital dönüşüm sayesinde hayatımızda yerini almıştır.

Bu çalışmada pandemi verileri üzerinde Amazon’un Bulut Bilişim servisleri ve araçları ile çalışılarak anlamlı veriler elde edilip grafiklerle görselleştirilmiş, elde edilen anlamlı veriler üzerinde zaman serileri makine öğrenmesi algoritmaları çalıştırılarak tahminler üretilmiştir. Vaka sayısına dair tahmin üretilen zaman diliminde, üretilen tahminlere oranla daha düşük vaka gerçekleşmiştir. Aşılmanın artmasının etkileri çalışmada açıkça gözlemlenmiştir. Çalışmanın sonucunda, otomatik olarak internet üzerinden güncel verileri elde edip bunları bulut ortamında saklayan ve işleyen bir sistem oluşturulmuştur.

Anahtar Kelimeler: Bulut Bilişim, Büyük Veri, ETL, Makine Öğrenmesi, Zaman Serileri.

MACHINE LEARNING APPLICATIONS USING AMAZON WEB SERVICES INFRASTRUCTURE

ABSTRACT

The corona virus, which entered our lives in 2019, spread all over the world, causing a pandemic. The processing of data related to the epidemic is necessary for the measures that can be taken, to prevent other possible epidemics, to make correct planning and to decide in a way that will affect the life flow at least.

Since pandemic data is "big data", storage and processing of this data is only possible with technological methods that enable big data processing. In addition, the fact that the data is received from various sources with different data structures and is dynamic makes it difficult to operate on the data. In the world of informatics, many reasons such as the size of the data, the necessity of continuous access to data, financial difficulties and security problems have led the information technology to digital transformation. Cloud Computing Technology has taken its place in our lives because of digital transformation.

In this study, by working with Amazon's cloud computing services and tools on pandemic data, meaningful data were obtained and visualized with graphics, and predictions were produced by running time series machine learning algorithms on the obtained meaningful data. In the time frame for which the number of cases was predicted, fewer cases were realized than the predictions. The effects of increased vaccination were clearly observed in the study. As a result of the study, a system was established that automatically obtains up-to-date data over the internet and stores and processes them in the cloud.

Keywords: Cloud Computing, Big Data, ETL, Machine Learning, Time Series.

GİRİŞ

Pandeminin dünyadaki etkinlik verileri de tıpkı hava durumu, sosyal medya, online alış-veriş verileri gibi büyük ve sürekli yenilenen verilerdir. Dolayısıyla depolanması da işlenmesi de zordur. Bu büyük verilerle işlem yapılabilmesi için nitelikli sunucular gereklidir. Örneğin beş yıllık veriyle bir sonuç çıkarılması gerektiğinde bu büyüklükteki veriyle işlem yapmak daha da zorlaşmaktadır. Sayısı ve çeşitliliği çok olan dosyaların gerçek zamanlı olarak algılanması ve şemalarının oluşturulması gerekir. Bu şemalardan meta veriler elde edilir ve meta veriler aracılığıyla işlemler yapılabilmektedir. Meta veriler, bir kütüphanenin envanter tablosu gibi düşünülebilir. Kitapların içeriklerini değil özelliklerini bulundurulur. Örneğin, kitaplara ait başlık, yazar adı, yayınevi bilgisi, yayınlanma tarihi, konusu gibi özellikler meta veri kapsamındadır. Böylece daha az veri ile işlem yapılması sağlanmış olur.

Makine öğrenmesi gibi büyük verilerle yapılmak istenen işlemler de artık bulut bilişim üzerinde gerçekleştirilebilmektedir. Makine öğrenmesinden çıkarılan sonuçlar, bulut ortamında değerlendirilebilir ve yeni dosyalara kaydedilebilir.

Makine öğrenmesi ile filmlerin başarı oranlarının tahmin edilmesine yönelik yapılan bir çalışmada, filmlerin başarısı aldığı IMDB puanı olarak değerlendirilmiştir. 5000 adet filmin başarı tahminleri 27 değişkenin analizi ile yapılmıştır. Analiz sonuçları R dilinde GGPlot2 kullanılarak sunulmuştur [1].

6 ayrı makine öğrenmesi algoritması kullanılan çalışmada, algoritmalar başarı oranlarına göre sıralanmıştır [1].

1660 yönetmen ve 3621 aktörün bulunduğu IMDB veri seti üzerinde yapılan makine öğrenmesi çalışması sonucunda altı algoritmadan en doğru sonucu veren Random Forest Algoritması olmuştur. Ona en yakın sonucu veren algoritma ise Boosting algoritmasıdır [1]. Bir başka çalışmada ise, kolektif öğrenme yöntemleri uygulanarak oluşturulan erken uyarı sistemi ile gelişmekte olan bir çocuğun obezite riskine dair tahminler üretilmiştir [2].

Web istatistikleri kullanılarak yapılan bir çalışmada web istatistiklerinin ve web trafiği ile ilgili veriler makine öğrenmesi ile analiz edilerek yararlı bilgiler çıkarılması amaçlanmıştır. Web sitelerindeki ziyaret sayısının niteliğinin diğer sayılara kıyasla yüksek olduğu belirlenmiştir. Ayrıca ortalama ziyaret sayısı ve ortalama süre niteliklerinin de algoritmalarda belirleyici rol oynadığı gözlenmiştir [3].

Bir başka çalışmada, makine öğrenmesi meme kanseri riskinin saptanması amacıyla kullanılmıştır. Yapılan makine öğrenmesi ve tahmin üretme sonucunda sistemin doğruluğu %90.0, hassasiyeti %86.0 ve özgüllüğü %93.0 olarak ölçülmüştür. Çalışma yazarı tarafından sınıflandırmanın doğruluğunun artırılması için daha iyi doku özellikleri çıkarılması ve daha yüksek sayıda örnek ile öğrenmenin tekrarlanması önerilmektedir [4].

Araştırmanın Amacı

2019 yılının sonlarında ortaya çıkan korona virüsü ile hayatımıza giren salgın, dünyanın birçok yerine yayılarak insanları etkisi altına. Bu çalışmada, öncelikle korona virüsünden kaynaklanan salgına ilişkin bir veri kümesi, gerçek zamanlı ve sürekli olarak elde edilerek, bu veri kümesi üzerinde işlemler yapılmış ve okunabilir tablolar oluşturulmuştur. Daha sonra, işlenmiş olan veri kümesi üzerinde çeşitli makine öğrenmesi algoritmaları çalıştırılmış ve tahminler üretilmiştir. Salgına ilişkin tahminleri, mevcut güncel salgın verilerinden otomatik olarak üretebilen bir sistem geliştirilerek, salgınlara karşı mücadelede kullanılacak bir yapı oluşturulması amaçlanmıştır.

Bu çalışma Büyük Veri ve Bulut Bilişim, Makine Öğrenmesi ve Tahmine Yönelik Algoritmalar, Araç ve Hizmetler, Uygulama Adımları, Sonuçlar ve Öneriler olmak üzere beş bölümden oluşmaktadır.

“Büyük Veri ve Bulut Bilişim” adlı bölümde büyük veri ve Bulut Bilişim’in tanımları yapılmış, Bulut Bilişim’in çeşitleri aktarılmıştır. Bulut Bilişim sistemlerinin avantajları ve zorlayıcı yanları sıralanmış, bu sistemlere geçiş adımları açıklanmıştır. Ayrıca başta gelen Bulut Bilişim sağlayıcıları belirtilerek bunlar arasında karşılaştırma yapılmıştır.

“Makine Öğrenmesi ve Tahmine Yönelik Algoritmalar” adlı bölümde makine öğrenmesinin tanımı ve açıklamaları yer almaktadır. Aynı zamanda makine öğrenmesinin mevcut uygulama alanları ve gelecekte makine öğrenmesi ile yapılabilecek çalışmalar belirtilmiştir.

“Araç ve Hizmetler” adlı bölümde, çalışmada kullanılan araç ve hizmetler tanıtılmış, çalışma prensipleri açıklanmıştır.

“Uygulama Adımları” adlı bölümde, çalışma kapsamında geliştirilen projenin tüm adımları yer almaktadır. Proje için ortamın oluşturulmasından çıktıların görüntülenmesine kadar yapılan tüm işlemler ayrıntılarıyla aktarılmış ve şekillerle desteklenerek açıklanmıştır. Veri kümesinin üzerinde uygulanan algoritmalar ile makine öğrenmesi gerçekleştirilmesi ve tahminler üretilmesine dair ayrıntılar bu bölümde yer almaktadır. Bölümün sonunda, üretilen tahminlerin doğruluk değerleri hesaplanmış ve sonuçlar değerlendirilmiştir.

“Sonuçlar ve Öneriler” adlı bölümde, çalışmada kullanılan yöntemler, araçlar, uygulama çıktıları ve hesaplanan doğruluk değerleri analiz edilmiş ve bu alanda yapılacak çalışmalara yönelik önerilerde bulunulmuştur.

1. BÜYÜK VERİ VE BULUT BİLİŞİM

1.1. Giriş

Büyük veri ve Bulut Bilişim teknolojileri, birbiriyle yakından ilgili iki kavramdır. Bilginin büyüklüğünün hızla arttığı göz önünde bulundurulursa, günümüzde Bulut Bilişim teknolojisi olmadan dijital işlemlerin sürdürülebilmesi mümkün değildir. Çalışmanın bu bölümünde, büyük veri ve Bulut Bilişim tanımlanarak, özellikleri anlatılmıştır. Literatürde Bulut Bilişim'in tarihçesine ilişkin hazırlanan çalışmalar derlenmiştir. Ayrıca Bulut Bilişim'in kullanım alanlarına göre çeşitleri açıklanmış, avantaj ve dezavantajları sıralanmıştır. Önde gelen Bulut Bilişim sağlayıcıları belirtilmiş ve karşılaştırma tablosu hazırlanmıştır.

1.2. Büyük Veri

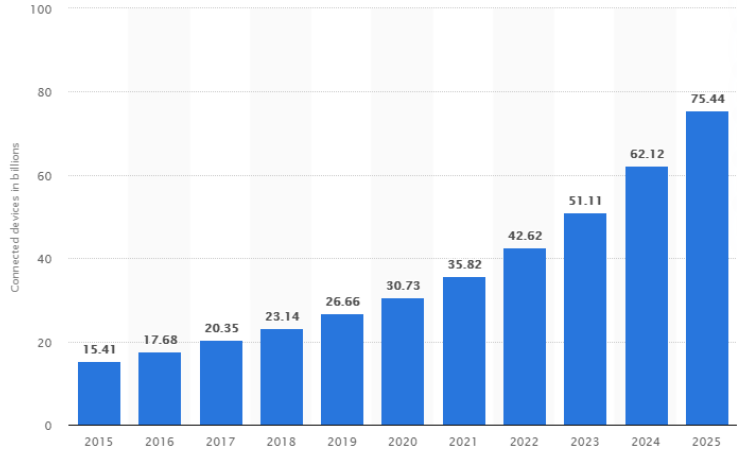
Günümüzde teknolojinin gelişme hızına paralel olarak bilginin büyüklüğü, çeşitliliği ve önemi de hızla artmaktadır. Günlük hayatta kullanılan sayısız küçük elektronik alette depolanan bilginin yanı sıra, kurumların ihtiyaç duyduğu ve işlemek üzere sakladığı oldukça büyük bilgiler mevcuttur.

Örneğin, internet alışveriş sitelerinin verileri işlenerek, kişilerin alışveriş alışkanlıkları ve yatkınlıkları, yaş grubuna, cinsiyetine, ekonomik duruma ya da mesleğine göre saptanabilir. Alışveriş mağazaları da bu bilgiye göre kampanya ve reklamlar yaratarak satışlarını arttırmayı hedefler.

Bir başka örnek olarak, insanların sosyal medya paylaşımları yoluyla sağlanan veri işlenerek psikolojik ve sosyokültürel raporlar çıkarılabilir. Bu kişilerin suç işleme eğilimleri, ekonomik durumları, eğitim seviyeleri ya da sağlık durumları ile ilgili tahminler yapılabilir. Bu tahminler, bir güvenlik soruşturmasına, kredi başvurusu için durum değerlendirmesine ya da sağlık sigortası araştırmasına katkı sağlayabilir.

2011 senesinde dünyadaki birbiriyle ilişkili teknolojik aletlerin sayısı, yaşayan insan nüfusunu geçmiştir [5]. 2025 yılına kadar çevrimiçi elektronik alet sayısının yetmiş

milyarı geçmesi beklenmektedir. Şekil 1.1’de 2015- 2025 yılları arasında öngörülen internete bağlantılı cihaz sayısı grafik olarak verilmiştir [6].



Şekil 1.1. Statista firmasının 2015- 2025 yılları arası internete bağlantılı cihaz sayısı grafiği

Bu veriler, kişilerin yararına kullanılabileceği gibi, zararına da kullanılabilir. Bu durum, verilerin depolanmasının yanı sıra güvenliğinin de önemli bir sorun teşkil etmesine sebep olur. Güvenliğin önemi özellikle devlet ve banka sistemlerinde açıkça görülürken, kişisel bilgilerin gizliliğinde de çok kritik olduğu zamanla anlaşılmıştır. Bilgiye kesintisiz erişim ise bir diğer zorunluluktur. Bu durum, klasik bilişim anlayışından Bulut Bilişim Teknolojisi’ne geçiş zemin hazırlamıştır.

Elde edilen bu büyük veri, kendi başına bir anlam ifade etmemekte, yapılandırılması ve işlenmesi gerekmektedir. Ancak verinin karmaşıklığı, yapısız-zamansız olması ve büyüklüğü, üzerinde işlem yapılmasını neredeyse imkansız hale getirmektedir. Zor koşullarda işlemler yapılabilse de zaman açısından verim sağlanamamaktadır. Bulut Bilişim Teknolojisi’nde büyük pay sahibi olan Amazon Web Service, büyük veri üzerinde yapılmak istenen karmaşık işlemlerin yönetimini kolaylaştırmak için AWS Glue’yu geliştirmiştir.

Sosyal medya paylaşımları, alışveriş sitelerinin satış bilgileri, bloglar, fotoğraf ve videolar, internet sitelerinin tıklanma sayıları, web service logları gibi birçok kaynaktan elde edilen, yapılandırılmış ve yapılandırılmamış verilerin bütünü “veri gölü” olarak adlandırılır. Veri gölünden işe yarar bilgi sağlanabilmesi için; ilgili veri, analiz yapılabilmesini sağlayan teknolojiler bütününe, yani “veri ambarı” adındaki

depoya aktarılır [7]. Böylece veri tabanındaki yük azaltılmış olur. Yapılan analiz sonucunda elde edilen yapılandırılmış ve anlamlı veri, büyük veridir.

1.3. Bulut Bilişim

Bulut Bilişim, bilgisayar kaynaklarının kullanıcılar tarafından paylaşımlı olarak kullanılabilirdiği, internet tabanlı bilişim hizmetlerine verilen isimdir. Bulut Bilişim Teknolojisi sayesinde hem yazılım hem donanım işleri daha verimli hale gelmiş, iş yükü azaltılmıştır [8].

Bulut Bilişimin gelişmesi ve yaygınlaşması sayesinde artık veri depolama, dosya paylaşma gibi halihazırda var olan bulut hizmetlerinin yanı sıra farklı yazılımsal ve donanımsal ihtiyaçlar da karşılanmaya başlanmıştır. Yapay zeka, makine öğrenmesi, görüntü işleme, büyük veri analizi gibi alanlarda da hizmet sağlamaya başlayan Bulut Bilişim, henüz gelişiminin çok başındadır [9].

1.4. Bulut Bilişim'in Tarihçesi

Bulut Bilişim şu anki bildiğimiz haliyle ortaya çıkmadan önce, Bulut Bilişim'i oluşturan bazı bileşenler farklı şekillerde kullanılmaya başlanmıştır. Bu durum, Bulut Bilişim'in tanımlanmasında kronolojik olarak pürüz yaratmaktadır. Ayrıca Bulut Bilişim'in çok sayıda farklı amaç için kullanılabiliyor olması da birden fazla tanımının bulunmasına sebep olmaktadır. Bulut Bilişim Teknolojisinin daha iyi tanımlanabilmesi için, Bulut Bilişim tabanlı sistemlerde kullanılan bazı bileşenlerin detaylı şekilde incelenmesi gerekir.

1950'li yıllarda bilgisayar yeni geliştirilmekteydi. Paket ağları ile ilgili ilk tasarımlar da bu yıllarda başladı. Tek bir fiziksel ortamda birden çok farklı ortamın kullanılabilmesi fikri geliştirildi [10].

1950'lerde ana bilgisayarlara sahip olmak oldukça maliyetliydi. Kaynakların paylaşımı söz konusu değildi. Farklı kullanıcıların farklı zamanlarda dahi kaynakları paylaşması mümkün değildi. Telekomünikasyon hatlarının kurulmasıyla merkez sunuculara uzaktan erişilebilmeye başlanmış ve zaman paylaşımlı bağlantı kurulabildi.

1960'lı yıllarda oda büyüklüğündeki mainframe bilgisayarlar ile büyük şirketlerin dijital işleri yapılmaya çalışılmaktaydı. Sadece kullanıcılar ile ana bilgisayarlar arasında ve terminallerle bağlantı kurabilmekteydi [11]. Küresel bir ağ üzerinden kaynakların başka kullanıcılara da sunulması fikri Bulut Bilişim kavramı olarak adlandırılmaya başladı. ARPANET, 1960'lı yılların başında IP (İnternet Protokolü) kullanan ilk ağ olarak tarihe geçmiştir. O yıllarda çeşitli haberleşme protokolleri ile paket anahtarlamalı ağların geliştirilmesi günümüzdeki internetin temel yapısını oluşturmuştur. Bu kavramın fikir yaratıcısı John McCarthy olarak görülmektedir [10].

1970'li yıllarda ilk kez VM (sanal makine) kavramı ortaya çıktı. Küresel bilgisayar merkezleri bu yıllarda kuruldu [10].

1980'de istemci-sunucu sistemi geliştirildi. Bilgisayarlar arasındaki ağların gelişmesiyle internet kavramı daha da yaygınlaşmış oldu. AHHP ve ICP protokollerinden TCP/IP protokollerine geçilmesi, internetin önceki döneme göre daha esnek bir yapıya kavuşmasını sağladı.

İşlemci kapasitesi ve performansın artırıldığı yeni bilgisayarlar, kişisel bilgisayar adı ile evlerde yerini almaya başlamıştır. Kişisel bilgisayarlar ile artık kontrol ana bilgisayarda değil, her bir bilgisayarın kendi bünyesinde bulunmaya başladı ve bu durum işlemlerin hızlanmasını sağladı [11].

İlk kez 1980'lerin sonunda İngiltere'de World Wide Web üzerine yapılan çalışmalar ile zenginleştirilmiş text dokümanlarının bir ağ sistemine entegre edilmesi başarılı.

İnternet kavramının günümüzde akla gelen haliyle ortaya çıkması ise 1990'lı yıllarda gerçekleşmiştir. Farklı lokasyonlardaki bilgilere ulaşmak ve paylaşım yapmak için oluşturulan LAN'lar ile internetin ilk hali meydana getirilmiştir. Bağlantı hızının yükselmesi, bant genişliğinin artması ve internetin yaygınlaşması ile hem kullanım alanı genişlemiş hem de güvenlik açıkları ortaya çıkmaya başlamıştır [11]. Anlık mesaj gönderimi, e-posta iletişimi gibi işlemler bu yıllarda başarılı ve hızlıca yaygınlaşmıştır. İnternet kavramı, sunucular ile kullanıcılar arasında ve kullanıcıların kendi aralarında ağ kurmasını sağlamıştır. Böylelikle yavaş yavaş "bilişim" kelimesi kullanılmaya başlanmıştır. İnternet ağlarının gelişmeye devam etmesiyle, kullanıcılar

kendilerine özel sanal ağına sahip olmaya başlamıştır ve bu ağına “telekomünikasyon” denir.

1990’lı yıllardan önceden yalnızca bir noktadan diğer noktaya veri bağlantısı sunan telekomünikasyon şirketleri, 1990’lı yıllarda sanallaştırılmış özel bağlantılar sunmaya başlayarak aynı hizmeti daha düşük ücrete mal etmeyi başarmıştır. Ayrıca internet üzerinden uygulamaların hizmet olarak sunulması da bu yıllarda başlamıştır. Fiziksel olarak altyapının artırılması ve geliştirilmesi yerine, mevcut altyapı daha fazla kullanıcının ortak erişimine sunulmuştur.

2000’li yıllarda yüksek kapasiteli ağlar yaygınlaştı. Donanımların yerini sanal kaynakların alması hızlandı ve donanım maliyetleri düşüşe geçti. Şirketler yavaş yavaş bilişim hizmetlerini kendi bünyelerinin dışındaki kaynaklardan almaya yöneldiler. Bunun başlıca sebepleri bakım-onarım işlerinin zorluğu ve masrafların yüksek olması olarak özetlenebilir. Dış kaynaklara olan bu yönelim, çeşitli hizmet modellerinin doğmasına sebep oldu ve hizmetin kullanıldığı kadar ödeme yapılması mantığı ilk kez gündeme geldi. Bu hizmet modelleri Bulut Bilişim hizmet modelinin geliştirilmesine sebep oldu [11]. 2004 yılında Web 2.0’den bahsedilmeye başlandı. Kullanıcıların yarattığı dinamik web arayüzleri, kişisel paylaşımlar ve iletişime duyulan ihtiyacın artması, Web 2.0’den geçişi hızlandırdı.

2006 yılında ilk gerçek bulut bilişim hizmeti Amazon şirketinin geliştirdiği S3 ürünü olmuştur. Başlarda ağ sistemlerinin sadece küçük bir bölümü kullanırken, ilerleyen zamanlarda kullanılmayan kaynakların değerlendirilebildiği sistemler geliştirilmiştir. Ayrıca Google şirketi de Google Documents hizmetini bu yıl sunmuştur.

2007 yılında Apple kablosuz ağlarla kullanılabilen ilk mobil cihazı (iPhone) geliştirdi.

2008 yılında artık kişisel buluttan ve bulut güvenliğinden söz edilmeye başlandı.

İlk açık kaynaklı bulut 2008 yılında RESERVOIR European Commission desteği ile yapılmış Nebula’dır. Bu projeden sonra artık gerçek zamanlı bulut sistemleri konuşulmaya başlanmıştır.

2000’li yıllarda Google, Microsoft, Apple, IBM ve Amazon şirketlerinin geliştirdiği hizmetler ile bilgisayardan bağımsız online uygulamalar, özel ve açık bulutlar, kablosuz ortamda veri depolama ve paylaşma gibi kolaylıklar hayatımıza girmiştir.

Bulut Bilişim’in yaygınlaşmasından önce üretilmiş uygulamalar dahi, altyapıları yenilenerek Web 2.0 ile hayatımıza giren Bulut Bilişim hizmetleri ile uyumlu hale getirilmeye çalışılmaktadır [12].

1.5. Bulut Bilişim’in Özellikleri

NIST’in yaptığı tanımlamada, Bulut Bilişim sistemlerinin özelliklerini beş ana maddede toplanmıştır: İsteğe bağlı servis, platform bağımsız yaygın ağ erişimi, kaynak havuzu oluşturma, esneklik ve ölçülebilirlik [13].

İsteğe Bağlı Servis: Hizmet sağlayıcıları ile iletişimsizdir. Sunucuların kullanım süreleri ve depolama hizmeti ihtiyaca göre sunulmaktadır. Ulaşmak için onaylama gerekmemektedir. Kaynak havuzu oldukça verimli şekilde hizmet vermektedir.

Platform Bağımsız Yaygın Ağ Erişimi: Standart mekanizmalar bulunmaktadır. Bulut Bilişim sistemi hizmetleri internet aracılığıyla kullanılabilen farklı istemciler üzerinden gerçekleşmektedir. Cep telefonları, bilgisayarlar, tablet bilgisayarlar bu erişim türüne örnek verilebilir. Hizmet kaynaklarına ulaşılması sınırlıdır. Özel protokoller mevcuttur. Lisanslar ve ücretler bulunmamaktadır. Ağ ve iletişim hizmetleri sağlayıcısı http protokolleri ile sağlanır. Bu protokolleri sağlayan her cihazdan sunucuya erişmek mümkün hale gelmiştir. Kullanılan ortak ağlar sebebiyle hem kullanıcıların hem de farklı programların ve API’lerin Bulut Bilişim sistemlerinin altyapısına erişmesi sağlanabilmektedir. Programların ve API’lerin yoğun veri alış-verişi yapması, Bulut Bilişim sistemlerinin hizmet sağlayıcılarını zorlamaktadır.

Kaynak Havuzu Oluşturma: Bulut Bilişim sistemlerinin temel mantığı olan, birden çok sayıdaki kullanıcıya aynı zamanda hizmet sağlayabilmeyi başarabilmesi için, kullanıcı talepleri baz alınmaktadır. Çoklu kiracı modeli sayesinde, sunucular her kullanıcıya ayrı ayrı atanabilir ve serbest bırakılabilir. Bu hizmet modelinde, farklı lokasyonlardaki kaynaklar, hizmet sağlayıcılar tarafından ortak bir havuza dahil edilmektedir. Bu yerler hakkında bilgi edinilmemesi ve kontrol edilememesi endişelendirici gibi görünmektedir ancak verilerin sanallaştırılması, verilerin nerelerde

depolandığına dair şehir bilgisi sunulmasını sağlar [14]. Bulut Bilişim'den önce, sunucuların atanması, bu işlerin sunucuya mal olacağı yük ile belirlenmekteydi. Bulut Bilişim sistemlerinde ise her bir uygulamanın tamamlamak istediği işe göre ihtiyaç duyduğu kaynağa ulaşması sağlanmaktadır. İşini tamamlayan uygulamaların bu kaynaklara erişimi kesilerek ihtiyaç duyacak başka uygulamalar için kaynaklar tekrar ulaşılabilir hale getirilmiş olur. Böylece kesintisiz olarak hizmet sağlanmaktadır. Kamu kurumları çok fazla sayıda ve çeşitte hizmeti sağlayabilmeyi amaçlamaktadır. Bazı hizmetler zamana ve olaylara bağlı olarak daha yoğun kullanılmaktadır.

Farklı birimlere atanmış depolama alanı ve işlemci gücü gibi kaynaklar, yoğunluk yaşanması durumunda yoğunluk yaşanan birime aktarılmaktadır. Böylece ihtiyaç duyulan uygulamaların sürekli erişilebilir olması sağlanmaktadır.

Esneklik: Bulut Bilişim sistemlerinde, kaynaklar ihtiyaca göre farklı birimlere atanabildiği gibi, kaynak kapasiteleri de değiştirilebilmektedir. Uygulamalara verilen işlerin ne ölçüde kaynak gerektirdiğinin kullanılmaya başlamadan önce doğru hesaplanamaması, Bulut Bilişim sistemleri sayesinde artık sorun teşkil etmemektedir. Çünkü kullanıldıkça ölçülebilir ve ihtiyaca göre arttırılıp azaltılabilir. Genel olarak kullanıcının müdahale etmesine dahi gerek kalmadan, sistemler bu düzenlemeyi otomatik olarak yapar.

Ölçülebilirlik: Hizmet sağlayıcılarının bant genişliği ve depolama kapasitesi göz önünde bulundurularak kaynakların en verimli şekilde kullanılması için kontroller yapılarak ölçülmesi sağlanır. Kullanıcılar, dilerlerse kaynakların kullanım durumunu gözlemleyebilir, yönetebilir ve log kayıtları tutabilir. Bulut Bilişim, hizmet seviye anlaşması (SLA) haricinde ek bir sorumluluk olmadan, kaynak havuzuna sürekli erişim imkanı vermektedir. Bu nedenle, ölçülebilirlik, kaynakların ücretlendirilebilmesi, değişken kaynak trafiğinin verimli kullanılmak üzere yönetilebilmesi, ihtiyaç duyulan kaynakların hesaplanabilmesi gibi işlemler için bir gerekliliktir. Böylelikle, Bulut Bilişim sistemlerini kullananlar, kullanım sürelerini takip edebilmekte ve ihtiyaca göre kaynak arttırma ve azaltma yapabilmektedir [12].

Tüm bu özellikleri ile Bulut Bilişim birçok kişi ve kurum için uygun teknoloji çözümü konumuna gelmektedir.

1.6. Bulut Bilişim Çeşitleri

Bulut Bilişim sistemleri, kullanıcıların kaynakları kendi özel ihtiyaçlarına göre en verimli olacak şekilde kullanabilmesi için farklı modeller sunmaktadır. Kullanıcılar, kendilerine en uygun modeli seçip kullanmakta özgür oldukları gibi, ihtiyaçların veya önceliklerin değişmesi durumunda farklı sistem modellerine de geçebilmektedir. Her model, kendine özgü fiyatlandırmaya sahiptir [15].

NIST'in yapmış olduğu tanıma göre, Bulut Bilişim Sistemleri dört farklı çeşittir ve şu şekilde sıralanabilir [13]:

Genel bulut

Tüm kullanıcıların erişim sağlayabildiği bir ağ üzerinden sunulan Bulut Bilişim Hizmeti modelidir. Genel Bulut Hizmetini sağlayan şirket, gerekli yazılımı ve sunucu kaynaklarını internet üzerinden tüm kullanıcılara sunmaktadır [16]. Bu sistem, kullanıcıların kendi veri merkezlerinde veya farklı merkezlerde konumlandırılabilir [13]. Bilgilerin bulut dışına taşınabilmesine olanak sağlar. Kullanıcıların Bulut Bilişim Sistemini kullandığı kadar ödeme yaptığı bir sistem modelidir [17]. Genelde orta ve küçük ölçekli şirketler için ideal sistem modelidir [18].

Özel bulut

Sistem ve uygulamalar, firmanın kendi veri merkezinde veya farklı merkezlerde bulunabilmektedir. Firmaların sistem kaynaklarının yönetimine erişme ve yönetimi kontrol etme seçeneği bulunmaktadır [13]. Belirlenen kişilerin ulaşımına izin verilmektedir. Güvenlik duvarının entegre edilmesine olanak sağladığı için gizlilik ve güvenlik konusunda öne çıkmaktadır [19]. Büyük şirketlerin kullanımı için ideal olan Bulut Bilişim Hizmeti modelidir [13].

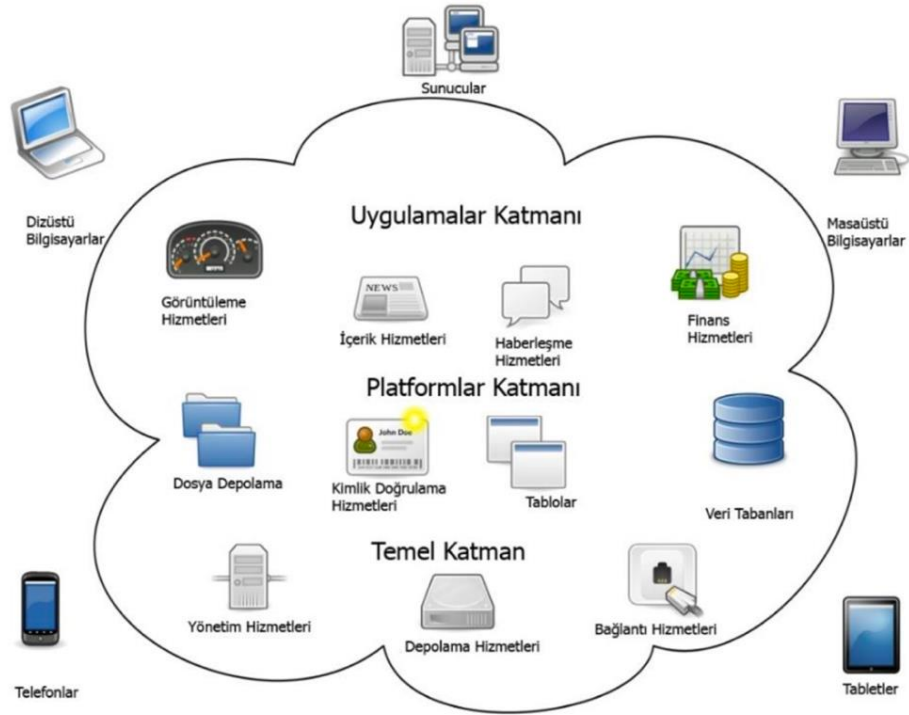
Topluluk bulutu

Güvenlik ve gizlilik gibi ihtiyaçları birbirine yakın olan kurumlar için tasarlanmış olan Bulut Bilişim hizmet modelidir. Bu Bulut Bilişim hizmetinin altyapısı, benzer ihtiyaçlara sahip kurumlara paylaştırılmış şekilde kullanılmaktadır [13].

Karma (Hibrit) bulut

Birden çok farklı Bulut Bilişim Hizmet modelinin bulunduğu bulut ortamıdır [13]. Genel ve Özel Bulut Bilişim hizmet modellerinin her ikisinin de kullanıldığı ve birbirleri arasında veri paylaşımı mantığıyla çalışan Bulut Bilişim hizmet modelidir. İhtiyaca göre, güvenliğin öncelikli olduğu işlemler için Özel Bulut Bilişim Sistemi kullanılırken, güvenlik sorunu teşkil etmeyen işlemler için Genel Bulut Bilişim Sistemi kullanılmaktadır [20].

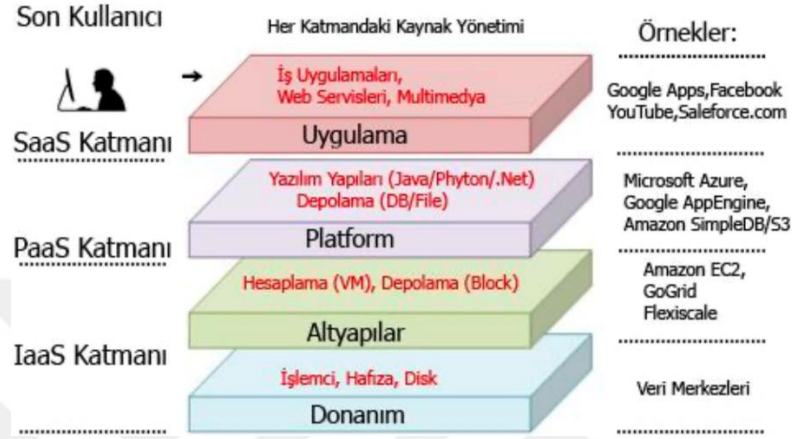
Bulut Bilişim'in genel yapısında önyüz ve arka yüz olmak üzere iki temel bölüm bulunur. Önyüz, kullanıcının görebildiği ve isteğine göre değişiklikler yapabildiği alandır. Arka yüzde ise kullanıcının isteklerine cevap verilmesini sağlayan sunucular, bilgisayarlar ve depolama alanları bulunur [21]. Şekil 2.1'de genel bir bulut mimarisi verilmiştir [22].



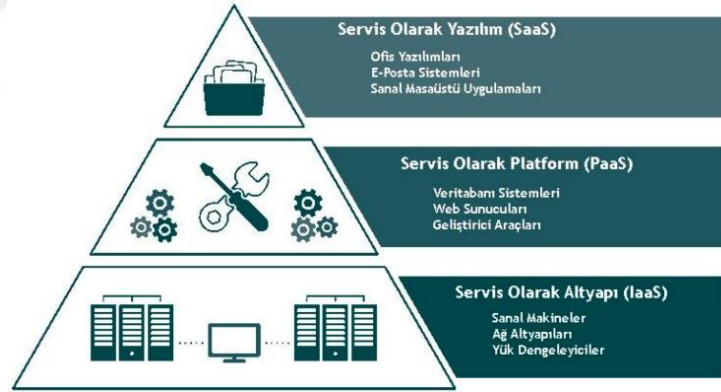
Şekil 2.1. Basit bir bulut mimarisi

1.7. Bulut Bilişim Servis Modelleri

Bulut Bilişim Sistemleri, kullanım şekillerine göre farklı modellere ayrıldığı gibi, hizmetlerin sunulma yöntemlerinin farklılığına göre de altyapı servis sınıflarına ayrılır. En sık kullanılan servisler Şekil 2.2 ve Şekil 2.3'te gösterilmiştir. [23, 24, 9].



Şekil 2.2. Son kullanıcı tarafından kullanılan servislerin tanımları



Şekil 2.3. Bulut Bilişim servis modelleri

Bulut bilişim tabanlı sistemlerde kullanılan servis modelleri şunlardır [25]:

Servis olarak altyapı (Infrastructure as a service, IaaS)

Kullanıcının Bulut Bilişim hizmetlerinde, diğer modellere kıyasla en fazla değişiklik yapabildiği modeldir [9]. Bulut Bilişim sisteminin en yaygın ve en basit hizmetidir [26].

Kullanıcıların ihtiyaç ve istekleri doğrultusunda işletim sistemi ve uygulamaları kullanmasına olanak sağlar. Veri depolama ve kaynak ihtiyacı ölçeklendirmeyi de

sağlamaktadır. Bu modelde, Bulut Bilişim sistemi kullanıcı tarafından kontrol edilememekte olmasına karşın, depolama birimleri, işletim sistemi ve birtakım uygulamalar sınırlı olarak kullanıcılar tarafından yönetilebilmektedir [14].

Kullanıcıların ihtiyaç duyduğu kadar sunucu alanı, ağ bağlantıları ve bant genişliğini IaaS modeli tarafından sağlanmaktadır. Kaynakların fiziksel konumlandırılması, Bulut Bilişim sağlayıcısındaki farklı veri kaynaklarına dağıtılmış halde bulunur. İstek yapıldığında kaynak havuzundan çekilerek, tamamen Bulut Bilişim sisteminin yönetiminde işlemler tamamlanır. IaaS modeli kullanıcılarının tek yapması gereken, kendilerine tahsis edilmiş sanal kaynaklara ulaşım sağlayarak işlemlerini yürütmektir. Kullanıldığı kadar ödeme yapılan ödeme türü, bu modelde en çok tercih edilendir [12].

IaaS Bulut Bilişim modelinin avantajları şu şekilde sıralanabilir:

- Ölçeklenebilir olması sayesinde kaynaklara duyulan ihtiyaç değiştiğinde hızlı ve kolay bir şekilde kaynak artırılabilir ya da azaltılabilir.
- Herhangi bir donanım satın alınması gerekmemektedir.
- Kullanıldığı kadar ödeme yapılması tasarrufludur.
- Her kullanıcının kendi veri merkezini kurmasına kıyasla, birçok kullanıcıya hizmet sağlayan büyük bir veri merkezi siber saldırılara ve fiziksel tehlikelere karşı daha güvenlidir.
- Veri merkezlerindeki donanımların fazla sayıda olması, olası bir arıza durumunda hizmetin kesintiye uğramamasını sağlar.
- Bulut Bilişim sistemi sağlayıcısı, gerekli tüm kaynakların sağlanması, çalıştırılması ve bakımını üstlenir. Bu işler için sermaye ayrılması ihtiyacı ortadan kalkar [12].
- Kullanıcı, ihtiyaca göre işletim sistemi kurabilir, web servislerini çalıştırabilir ve istediği güvenlik modelini bu yapıya entegre edebilir [9].

Servis olarak yazılım (Software as a service, SaaS)

Bu Bulut Bilişim modelini kullanan kullanıcılar, mevcut sistemlerine ek bir kurulum yapmaksızın Bulut Bilişim sistemlerine erişebilmektedir. Ancak kullanıcılar yalnızca uygulamalar katmanında değişiklikler yapmaya izinlidir [9]. E-posta sağlayıcıları bu modelin en yaygın örneğidir [26].

Bu modelin kullanıldığı başka alanlar; muhasebe, mesajlaşma, satış takibi olarak sıralanabilir. Bu modelin kullanıldığı en bilinen uygulamalar ise Microsoft Office paketleri, Google, Facebook, Twitter ve Instagram'dır.

SaaS Bulut Bilişim modelinin avantajları şu şekilde sıralanabilir:

- Donanım maliyet bulunmamaktadır. Çünkü gerekli donanımlar Bulut Bilişim sisteminde bulunmaktadır.
- Kurulumu oldukça kolaydır.
- Kullanıldıkça ödenebilen ödeme şekline sahiptir.
- Kullanıcı, ihtiyaca göre hızlı ve kolayca kaynak arttırabilir veya azaltabilir.
- Kullanımı için web tarayıcısı olması yeterlidir.

Servis olarak platform (Platform as a service, PaaS)

Bu Bulut Bilişim modeli, çoğunlukla uygulama geliştiricileri tarafından kullanılmaktadır. Çalışma prensibi olarak SaaS modeline benzemektedir [14]. Erişim web tarayıcısıyla sağlanır ve uygulama geliştirme için ihtiyaç duyulan programlama araçları gibi altyapı hizmetleri kullanıcıya sağlanmaktadır. Kullanıcı, kendi uygulaması dışındaki bileşenlere erişim iznine sahip değildir [9]. Dolayısıyla işletim sistemleri ve ağ yönetiminin de içinde bulunduğu temel katmandaki bileşenlerin kontrolü Bulut Bilişim sistemlerindedir. Yalnızca ortam yapılandırılırken müdahale edilebilmektedir. Uygulama servisleri, kütüphane ve kimlik yönetimi bu servisin kullanıldığı örnek uygulama alanlarıdır. Önemli bir fark olarak, bu modelde, ölçeklendirme işlemi Bulut Bilişim sistemi tarafından yapılmaktadır. Bu durum, geliştiricilerin omzundan kaynak yönetimi yükünü alarak, geliştirdikleri uygulamalara yoğunlaşmalarına yardımcı olmaktadır. Bu modelde kullanıcıların ihtiyaçları ve geri bildirimleri baz alınarak güncellemeler yapılmaktadır.

PaaS Bulut Bilişim modelinin avantajları şu şekilde sıralanabilir:

- Birçok yazılım aracı ve kütüphanesi sunduğu için özellikle uygulama geliştirmeye yeni başlayacak kullanıcılara kolaylık sağlamaktadır.
- Donanımsal maliyet bulunmamaktadır.
- Uygulama geliştiricilerin kullandıkları geliştirme araçlarını ihtiyaçları doğrultusunda yönetebilmesine imkan sunmaktadır.

- Platform bağımsız olması sayesinde, farklı lokasyonlardaki uygulama geliştiricilerinin, aynı uygulama üzerinde çalışmalarına olanak sağlamaktadır.
- Veri güvenliği, yedekleme ve bir arıza durumunda veri kurtarma işlemleri Bulut Bilişim sisteminin sorumluluğundadır.

1.8. Bulut Bilişim'in Avantajları

Bulut bilişim teknolojilerinin sağlamış olduğu avantajlar şu şekilde sıralanabilir:

- **Güvenlik:** Kişilerin ve kurumların, kullandıkları yazılımlarda güvenliği kendilerinin sağlaması gerekirken, Bulut Bilişim sistemlerinden hizmet alınması durumunda, bu gereklilik ortadan kalkmaktadır.
- **Yazılım Güncellemeleri:** Güvenlik konusunda olduğu gibi, Bulut Bilişim sistemlerinden hizmet alan kullanıcılar, kullandıkları yazılımların güncellemeleri hizmet sağlayıcıları tarafından yapılması sayesinde bu yükten kurtulmaktadır.
- **Çevrecilik:** Birçok kişi ve kurumun her birinin kendine ait kaynaklar edinmesine kıyasla, bu kaynakların ortak kullanımının sağlanması, ekosistemin korunması açısından oldukça önemli bir adımdır.
- **Düşük Maliyet:** Kaynakların ortak kullanımı maliyet açısından oldukça faydalıdır. Böylece hem yeni girişimciler kaynaklara daha kolay ve az maliyetle ulaşarak, girişimlerini daha güvenle kurmakta, hem de büyük firmalar kaynak teminatından tasarruf ettikleri için başka harcamalara öncelik verebilmektedir.
- **Zaman ve Mekan Bağımsız Çalışma:** Bulut Bilişim sistemlerinden hizmet almak için internet bağlantısına sahip olmak yeterlidir.
- **Esneklik ve Verimlilik:** Bulut Bilişim sistemleri, kullanıcılarının ihtiyaç durumuna göre, kullandıkları yazılım ve donanım kaynaklarını artırma veya azaltmasına olanak sağlamaktadır [27, 28]. Böylece kullanılan tüm kaynaklardan verim alınmış olmaktadır.

1.9. Bulut Bilişim ile İlgili Endişeler

Bulut Bilişim'in kullanıcılara sağladığı bütün kolaylıkların yanı sıra, Bulut Bilişim sistemlerine geçmek isteyen kişi ve kurumları endişelendiren bazı konular da mevcuttur. Bulut Bilişim ile ilgili endişelendirici konular şu şekilde sıralanabilir:

- **Sektör Dışında Olma:** Sektöre yabancı olan firmaların Bulut Bilişim sistemlerine geçmesi için bu alanda profesyonel olan kişi veya kurumlardan danışmanlık alması gerekmektedir. Dışarıdan alınan danışmanlık da tıpkı sistem geçişinde olacağı gibi, ekstra maliyet anlamına gelmekte ve firmalarda çekince yaratmaktadır.
- **Entegrasyon:** Benzersiz yapıda işlere yönelik oluşturulan Bulut Bilişim uygulamaları, o işlere özel yapıya sahiptir. Bu nedenle farklı işlere entegre edilememesi, kullanıcıları düşündürmektedir.
- **Güvenlik:** Hem şahıslar hem kurumlar için verilerin kendi kaynaklarının dışında bir ortamda tutulması güvenlik endişesi yaratmaktadır. Hem verinin yönetiminin başka kuruluşa verilmesi hem de bulut ortamındaki veriye uzaktan erişim sağlanıyor olması, erişim sırasında olabilecek saldırılara karşı bir güvenlik zafiyeti olarak görülmektedir. Ayrıca farklı kullanıcıların verileri ortak bir sunucuda tutulabildiği için, bir kullanıcının verisine yapılan saldırıda diğerlerinin de zarar görme riski öngörülmektedir.
- **Gereksinimlerin Belirlenmesi:** Bulut Bilişim sistemlerine geçiş yapacak firmalar, Bulut Bilişim hizmet sağlayıcısıyla iletişime geçtiğinde, firma gereksinimlerini detaylı ve kapsayıcı şekilde açıklayabilmelidir. Çünkü gereksinimlerin doğru belirlenememesi durumunda, kendilerine uygun olmayan Bulut Bilişim modeliyle vakit ve maliyet kaybı yaşanabilir.
- **Yetkinlik:** Bulut Bilişim hizmeti verecek olan firmanın bu alandaki yetkinliği ve başarısı çok önemli kriterlerden biridir. Yetkin olmayan bir firmayla ilgili; geçiş sırasında veya sonrasında yapılabilecek olası hatalar, firmanın ilerleyen zamanlarda sektör değiştirmesi, firmanın sektörde tutunamaması gibi ihtimaller mevcuttur. Alanında yetkin olmayan bir firmayla Bulut Bilişim sistemlerine geçiş deneyimi yaşamak ve hizmet devamlılığının sağlanamaması riski, firmalar açısından bir diğer çekince sebebi olabilir.
- **Denetim:** Bulut Bilişim sistemlerine geçildiğinde veriler dağıtık yapıda, farklı lokasyonlardaki sunucularda tutuluyor olacağı için, verilerin ve sunucuların denetimini sağlayamama endişesi yaşanabilir.
- **Fiziksel Güvenlik:** Sunucuların dağıtık yapıda ve farklı lokasyonlarda bulunması, aynı zamanda fiziksel güvenlik ile ilgili soru işaretleri yaratabilmektedir.
- **Belirsizlik:** Verilerin hangi sunuculara dağıtılmış olduğunun bilinmemesi, bazı kullanıcılar için bir endişe sebebi olabilmektedir.

- Faturalandırma: Bulut Bilişim sistemlerine geçiş sırasında ve sonrasında karşılaşılabilecek kullanım maliyetleri, ilk kez kullanacak olan firmalar için bir başka korkutucu madde olabilmektedir [29].
- Sürekli Bağlantı: Bulut Bilişim sistemlerinin kullanılması için internet bağlantısı gereklidir. Veriler üzerinde aktif olarak çalışılıyorsa, internet bağlantısının kesintisiz olması kritik önem taşır. Daha iyi internet daha yüksek bant genişliğini, dolayısıyla da daha yüksek internet ücretini beraberinde getirir.
- Gelişmekte Olması: Bulut Bilişim Teknolojileri, hızla gelişmekte olan ve sürekli yenilikler ortaya koyan bir alandır. Verimliliğin artması ve daha iyi hizmet sağlanabilmesi açısından yenilikler bir avantaj olsa da, büyük geliştirmeler yapıldığında mevcut sistemlerin güncellenmesi gerekliliği oluşmaktadır [30, 31].

1.10. Bulut Bilişim'e Geçiş

Bir Bulut Bilişim hizmet sağlayıcısından hizmet almak, yani Bulut Bilişim sistemlerine geçmek için yapılması gereken bir dizi işlem bulunmaktadır. Bu işlemler ihtiyaçlara göre ufak farklılıklar göstermektedir. Literatürde Bulut Bilişim sistemlerine geçiş için birçok farklı temel adım belirlenmiştir. Çalışmalara istinaden bu adımlar şu şekilde sıralanabilir [32, 33, 34, 35, 36]:

- Analiz ve Tasarım: İlk aşamada, Bulut Bilişim hizmetini almak isteyen kurumun ihtiyaçları ve bu ihtiyaçları sağlayacak çözümler belirlenmeye çalışılır. Kurumun aradığı çözümün Bulut Bilişim sistemleriyle karşılanıp karşılanamayacağı değerlendirilir. Çözüm olacak sistem tasarlanarak, hizmeti alacak ve sağlayacak olan tarafların karşılıklı olarak birbirlerine onay vermeleri beklenir.
- Pilot Uygulama ve Yapılandırma: İlk aşamada oluşturulan tasarım ışığında bir pilot uygulama hazırlanarak çözüm senaryosu simüle edilir. Başarılı versiyona ulaşılan kadar güncellemeler devam eder. Başarıya ulaşıldığında Bulut Bilişim sistemlerine geçiş için yapılandırma çalışmaları başlatılır.
- Değerlendirme: Bulut Bilişim sistemlerinin performansı ve ödeme şekillerinin uygunluğu ile ilgili değerlendirmeler yapılır. Gerekliyse düzeltmeler istenir.
- Sürekli Hale Getirme: Değerlendirme aşaması da tamamlandıktan sonra mevcut sistemlerin sürekli hale getirilmesi için ayarlamalar yapılır. Denetim ve bakım işlemleri planlanır. Bulut Bilişim sistemlerine geçiş işlemi tamamlanmış olur.

1.11. Bulut Bilişim Sağlayıcısı Örnekleri ve Karşılaştırma

Tablo 2.1’de en çok kullanılan bulut platformlarının temel özellikleri verilmiştir.

Tablo 2.1. En çok kullanılan bulut platformları

| Özellik | Amazon Web Services | IBM Bluemix | Google Cloud Platform | Microsoft Azure |
|------------------------|----------------------------|---|---|---|
| Protokol | HTTP, MQTT, Web Sockets | HTTP, MQTT, Web Sockets | HTTP, MQTT | HTTP, AMQP, MQTT |
| SDK / Programlama Dili | C, NodeJS, Java Script | Java, C#, php, C, Go, Swift, Ruby, Python, NodeJS | Java, Python, NodeJS, Ruby, Go, .NET ve PHP | .Net, UWP, Java |
| Fiyatlandırma | 1 Milyon mesaj başına 5 \$ | İlk 30 gün ücretsiz, sonraki kullanımlar ücretli | Hizmet tabanlı isteğe bağlı fiyatlandırma | İlk 30 gün ücretsiz, sonraki kullanım ücretli |

Ibm Bluemix

IBM firması, Bulut Bilişim sistemleri üzerinde Bluemix adında bir platform geliştirmiştir. Bluemix, geliştiricilerin en yaygın olarak kullandığı Java, php, Node.js, .Net, Python gibi birçok programlama dilini ve programlama araçlarını desteklemektedir. Ayrıca, uygulama geliştiricilerinin bulut merkezli, açık kaynak kodlu uygulamalar geliştirmesine olanak sağlamaktadır [37].

Microsoft Azure Bulut Hizmetleri

Microsoft, çeşitli ihtiyaçları karşılamaya yönelik hizmetler sunmak ve bu hizmetleri bulut tabanında geliştirdiği uygulamalarla kullanıcılarına servis etmek için Azure’u geliştirmiştir [38].

Google Cloud Platform

Diğer Bulut Bilişim hizmet sağlayıcıları gibi, depolama, IoT, makine öğrenmesi gibi hizmetleri sunmaktadır. Google platformu, arama motoru ve YouTube bu platformu kullanmaktadır [39].

Amazon Web Services

Amazon Web Service, sistemine bağlı olan tüm cihazların bulut sistemine ve başka cihazlarla olan bağlantısını sağlayan bir güvenlik altyapısı vardır. Milyonlarca cihaz ve bu cihazlar aracılığıyla iletilecek olan milyarlarca iletiyi desteklemektedir [40].

Bulut Bilişim hizmetlerinin haricinde hem bulut bilişim hem de veri tabanı sistemlerini aynı anda sunan hibrit sistemleri de mevcuttur. Bilişim ve Bilgi Teknolojileri sektöründe faaliyet gösteren yaklaşık 1 000 000 firma Amazon Web Service hizmetlerinden faydalanmaktadır [40].

2. MAKİNE ÖĞRENMESİ VE TAHMİNE YÖNELİK ALGORİTMALAR

2.1. Giriş

Çalışmanın bu bölümünde, makine öğrenmesi ve makine öğrenmesi için kullanılan tahmin algoritmaları anlatılmıştır. Makine öğrenmesinin gerçekleştirilebilmesi için izlenecek adımların da açıklandığı bu bölümde, makine öğrenmesinin performansını arttırmak için izlenebilecek adımlar da açıklanmıştır.

2.2. Makine Öğrenmesi

Makine öğrenmesi ilk olarak 1959 yılında terminolojiye girmiştir. Bilgisayar biliminin bir alt dalıdır. Yapay zeka ile öğrenme gerçekleştirilmesine ve tahmin üretmeye çalışan algoritmalarından oluşan bir sistemdir. Bu algoritmalar, bilinen klasik algoritmaların aksine, kuralları önceden belirlenmiş ve yapılacak işlemleri sabit olan algoritmalar değildir. Bu algoritmalar veri girişlerinden öğrenme gerçekleştirebilmek için tahminler ve kararlar üretmek üzere programlanmıştır [41].

Makine öğrenmesi, veriler üzerinde matematiksel işlemler yapılarak tahminler üretilen çalışma bütünüdür [42].

Makine öğrenmesi, makinelerin verileri öğrenmesi zorunluluğunun fark edilmesiyle bir ihtiyaç haline gelmiş ve geliştirilmeye başlanmıştır. Daha önce bilgisayar biliminin bir alt dalı olan makine öğrenmesi, hakkında yapılan çalışmaların ve duyulan ihtiyacın artmasıyla 1990'lı yıllarda yeniden adından söz ettirmiş ve bu sefer ayrı bir bilim alanı olarak gelişimine devam etmiştir. Makine öğrenmesi sıklıkla veri madenciliği ile karıştırılmaktadır. Temelde, makine öğrenmesi elde olan verilerle öğrenme gerçekleştirip tahmin üretmeyi hedefler. Veri madenciliği ise eldeki verilerden, o veriye ait daha önce bulunmamış özellikleri çıkarıp tanımlamayı hedefler. Bu iki sistemin ortak kullandığı metotlar bulunmaktadır [41].

Aslında makine öğrenmesi, insan beynini taklit ederek geliştirilmiştir. Tıpkı insanlar gibi, mevcut durumları görüp değerlendirdikçe, öğrenme becerisi de tahmin başarısı

da artmaktadır. Derin öğrenme, makine öğrenmesinden çıkarımlar sağlamak için kullanılan özel bir makine öğrenmesi çeşididir ve sinir ağları kullanır [43].

Günümüzde karşı karşıya kaldığımız büyük veriden sonuç çıkarmak, klasik yöntemlerle artık mümkün değildir. Mümkün olsa bile çok zaman ve kaynak gerektirmektedir. Makine öğrenmesinin her geçen gün popülerliğinin artmasında, büyük veriden sonuç çıkarmak için duyulan gerekliliğin etkisi vardır [41].

Algoritmalar ile verilere ait kalıpları öğrenen makine öğrenmesi, kalıpları kullanarak tahminler üretir. Makine öğrenmesi sabit komutlarla değil, verinin kendisiyle öğrenme gerçekleştirdiği için, değişken verilerle çalışırken oldukça kullanışlıdır [43].

En yaygın makine öğrenimi kullanım alanları şunlardır: Kişisel reklamlar gösterme, sahtekarlık tespiti, e-posta analizi, kullanıcı davranışlarına uygun cevaplar veren chatbotlar, sosyal medya ana sayfaları, CRM (Müşteri İlişkileri Yönetimi) [44].

Makine öğrenmesi sistemleri, her ne kadar öğrenme ve tahmin üretme işini kendileri yapsalar da öğrenilecek girdi verisini ve hangi çıktının isteneceğini belirleyen kişiler veri bilimi uzmanlarıdır. Modelin özelliklerini ve modelin daha iyi sonuçlar vermesi için gereklilikleri veri bilimi uzmanları belirler [44].

Çalışmanın giriş kısmında büyük veri olarak tanımlanan verinin, depolanmasının yanı sıra işlenmesinin de zorluğundan bahsedilmişti. Büyük verinin analiz edilebilmesi için geçmişteki verileri kullanarak gelecekteki veriye dair tahminde bulunulması, makine öğrenmesinin temel amacıdır [45].

Makine öğrenmesi uygulanacak olan veri kümesinde, elemanların özellikleri birbirine çok benzer olduğunda sınıflandırma işleminin doğru çalışması güçleşir. Bu gibi durumlarda, kullanılan verinin sayısının artırılması algoritmanın daha doğru sonuç vermesine olanak sağlamaktadır [46].

Verilerin Sayısallaştırılması

Resim: Her bir pikseli koda dönüştürülür. Eğer resim renkli ise bu kodlar RGB değerlerinden oluşturulur. Siyah-beyaz resimlerde ise 1-255 arası sayılara çevrilir [45].

Metin: Harfler, heceler ve kelimeler genelde frekanslarına göre kodlanarak sayılara çevrilir. Ayrıca sıralamasını belirten zaman bilgisi de bulunmaktadır [45].

Ses: Genlik ve frekansın zaman içinde değişimiyle kodlanır [45].

2.3. Makine Öğrenmesinin Terminolojisi

Veri: Yapılandırılmış veya yapılandırılmamış bir veri olabilir. Üzerinde işlem yapılacak verinin içeriğinde, bağlı değişkenler ve bu değişkenlerin sonucu olduğu düşünülen bir veya daha fazla sonuç değişkeni olur [47].

Eğitim Verisi: Verinin bir kısmı, makine öğrenmesi modelinin eğitiminde kullanılır. Genelde yaklaşık olarak verinin %70'i eğitim verisi olarak etiketlenir. Öğrenme modeli bu veriyi inceler ve mantıksal çıkarımlar yapar [47].

Test Verisi: Verinin eğitim için ayrılan bölümünden kalan kısımdır. Makine öğrenmesi modelinin çıkarım yaptığı mantığı test edebilmesi için kullanılır. Test verisindeki bağlı değişkenler ve tahmin edilmeye çalışılan sonuç değişken birbirinden ayrılır. Makine öğrenmesi modelinin bu değişkenlere bakarak sonuç değerlerini tahmin etmesine çalışılır [47].

Özellikler: Verideki değişkenlerin hepsi birer özelliktir ve sayısal olarak ifade edilir. Sayısal olmayan özellikler, makine öğrenmesi ile işlenmeden önce sayısal hale getirilir [47].

Doğruluk: Yapılan test sonucu doğru tahmin edilen veri sayısının toplam veriye oranıdır [47].

2.4. Makine Öğrenmesinin Kullanılabileceği Alanlar

Veri Kalıplarını Tanımlama: Yapılandırılmış veya yapılandırılmamış veriler üzerinde çalışılarak, bu verilerle ilgili ilişkiler ve kurallar bulmaya çalışılır. Verilerin kalıpları araştırılır [43].

Tahmin Üretme: Çıkarım yapılan veri yapılarıyla test verisi üzerinde tahminler üretilir. Örneğin, kredi talebinde bulunan müşterilerin kredi borçlarını düzenli ödeme ihtimali,

bankanın tüm müşterilerinin ödemelerini barındıran bir veri kümesi üzerinde makine öğrenmesi algoritması çalıştırarak tahmin edilebilir.

Veri Bütünlüğünü Geliştirme: Araştırılan verilere yeni özellikler katılarak veri geliştirilebilir [43].

Kullanıcı Deneyimini Geliştirme: Chatbot'lar gibi müşteri deneyimini iyileştirmek için üretilen çalışmalar, makine öğrenmesi ile geliştirilmektedir [43].

Risk Azaltma: Makine öğrenmesi, dolandırıcılık yapmak isteyen kişilerin attığı adımlardan da çıkarımlar yapmayı öğrenir ve buna karşı önlem almak üzere geliştirilebilir [43]. Dolandırıcılık taktikleri sürekli olarak değiştikçe, makine öğrenmesi de buna ayak uydurur. Makine öğrenmesi, dolandırıcılık denemeleri başarılı olmadan önce bunları yakalamak için yeni desenleri izler ve belirler [43].

Beyin Aktivitelerinin İncelenmesi: İnsanların farklı aktiviteler sırasında beyinlerinin hangi bölümlerinin çalıştığını belirlenerek, beyin haritaları çıkarılabilir [45].

Gen Dizilimlerinden Hastalık Tespiti: Gen dizilimlerinden hastalık tespiti ve tedavi önerisi yapılabilir. Olası hastalıklar önceden saptanabilir [45].

El Yazısı / Kitap Yazısı Tanıma: Yeterli veriyle eğitilmiş bir makine öğrenmesi algoritması, el ve kitap yazılarını tanıyabilir. Dijital ortamda bulunmayan eski yazılı kaynakların dijital ortama hızlıca aktarılması gerçekleştirilebilir.

Sıralanan örnekler dışında da tahmin üretilmesi gereken hemen hemen her alanda makine öğrenmesi kullanılabilir. Makine öğrenmesinin kullanılabilmesi için eldeki verilerin sayısal olması gerekmektedir. Çünkü makine öğrenmesi algoritmaları, temelde matematiksel işlemlerden oluşmaktadır [45].

2.5. En Yaygın Makine Öğrenmesi Algoritma Modelleri

2.5.1. Gözetimli öğrenme (Supervised learning)

Gözetimli öğrenme modelinde, giriş değerleri ile çıkış değerleri arasında anlamsal bağ kuran bir fonksiyon yaratılır. Bu giriş değerleri önceden belirlenmiş ve işaretlenmiş verilerdir. Yaratılan fonksiyon sınıflandırma veya regresyon algoritmaları ile

oluşturulabilir. Veri setindeki veriler eğitim verisi ve test verisi olarak ikiye ayrılır. Her iki grupta da hem giriş değerleri hem de çıkış değerleri mevcuttur [42].

Regresyon (Regression)

Regresyon öğrenme modelinde, modelin öğrendiği mantıksal ilişkiler ile değer tahminleri yapılır [47].

Karar ağaçları

Veriler üzerinde yaptığı gözlemlere dayanarak, istenen sonuca ulaşmak için en uygun yolu bulmaya çalışır [44].

Nöral ağlar

Büyük miktarda veriyi eğitim için kullanarak sonucunu bilmediği verilerle ilgili tahminler üretir [44].

2.5.2. Gözetimsiz öğrenme (Unsupervised learning)

Gözetimsiz öğrenme modelinde işlenen giriş verisi işaretlenmemiş veridir ve sınıfı belirsizdir. Veriler gruplandırılıp, kalıplar ve ilişkiler bulunur [42]. Bu model “sinir ağları” olarak da adlandırılır. Doğal dil işleme ve görüntü tanıma gibi karmaşık öğrenme işlemleri için kullanılır. Bu öğrenme modeli için çok büyük veriler gerekmektedir [44]

Kümeleme (Clustering)

Kümeleme algoritmaları bir gözetimsiz öğrenme metodu örneğidir. Üzerinde çalıştığı verinin ortak özelliklerini bularak kümelendirmeye çalışır [47].

2.5.3. Takviyeli öğrenme (Reinforcement Learning)

Öğrenmeyi gerçekleştirebilmek için yinelenen modeller içerir. Kullanışlı çıktılar üretildiğinde o çıktıyı üreten adımlara puan eklenirken diğer adımlardan ise puan azaltılarak, en uygun yöntem bulunmaya çalışılır [44]. Takviyeli öğrenme modelinde öğrenen elemanın bağımsız elemanlarla etkileşime girmesi sonucu bu elemanlardan edinilen tepkiler ile davranış biçimini öğrenmesi gerçekleştirilir. Öğretici, sistemin

ürettiği sonuçların doğruluğunu araştırır. Bu model davranışsal psikolojiden yola çıkılarak geliştirilmiştir [42].

Yoğun öğrenme

Yoğun öğrenme modelinde kullanılan veriler; doğrusal ve doğrusal olmayan eşitliklerden ve çoklu işlem kaynaklarından oluşan işlenmesi zorlu verilerdir. Karmaşık işlemler barındıran birden fazla algoritma ile geliştirilir [41].

Zaman Serileri

Bu çalışmada veri kümesi üzerinde zaman serileri öğrenme algoritmaları çalıştırılmıştır.

Bu çalışmada kullanılan veri kümesi zamana bağlı olarak değişen değerlere sahiptir. Zamana bağlı olarak değişen veri içeren veri kümeleri üzerinde zaman serisi analizi kullanılır [48]. Zaman serisi analizleri ile belli zaman aralıklarında ölçülmüş değerlere bakılarak ileriye yönelik tahminler üretmek amaçlanır. Analizin yapılabilmesi için, analiz yapılacak veri kümesinin denetimli makine öğrenmesi için işlenmesi gereklidir [49]. Bu işlem için birkaç farklı yöntem uygulanabilir. En yaygın olarak kullanılan yöntem Prophet'tir ve Facebook tarafından geliştirilmiştir [50].

Zaman serilerini etkileyen üç bileşen bulunur. Bunlar şu şekilde sıralanabilir [51]:

Trend (T): Bir zaman serisindeki uzun süreli ana eğilimdir. Trend sabit değildir, zaman ilerledikçe değişebilir. Doğrusal ya da eğrisel olabilir.

Mevsimsel Dalgalanmalar (M): Zaman serisi içinde aynı periyotta birden fazla kez gözlemlenen dalgalanmalardır.

Serinin, mevsimsel dalgalanmaların etkisinin arındırılması gerekir. Bu işlem için çeşitli yöntemler uygulanır: Hareketli ortalamalar, ortalama yüzde, mevsim indeksi, regresyon ve varyans analizi, gibi.

Konjonktürel Dalgalanmalar (K): Zaman serisindeki dalgalanmalar mevsimlerden daha uzun süreli bir dönemde gözlemleniyorsa bu dalgalanmalar konjonktüeldir.

Genelde ekonominin normal seyrini ve kriz dönemini içeren bölümler konjonktürel dalgalanmalardır. Bu dalgalanmalar düzensizdir ve periyodik değildir.

Düzensiz Hareketler (D): Tekrarlı olmayan değişimler düzensiz hareketlerdir. Önceden tahmin edilemezler ve hata payı olarak ifade edilirler [51].

Zaman serilerinin analiz edilmesi ve zaman serileri ile tahminler üretilmesi için serilerin bu faktörlerin etkilerinden arındırılması gerekir.

2.6. Makine Öğrenmesi Adımları

Veriyi hazırlama

Üzerinde işlem yapılacak veri yapılandırılmış veya yapılandırılmamış bir veri olabilir. İçeriğinde, bağlı değişkenler ve bu değişkenlerin sonucu olduğu düşünülen bir veya birden çok sonuç değişkeni olur. Sonuç değişkeninin tipine göre kullanılacak makine öğrenmesi metodu seçilir [43].

Özellik çıkarımı

Makine öğrenmesi modelinin en verimli şekilde çalışabilmesi için verinin hangi özelliklerinin sonuç değişkeni üzerinde etkisi olduğunun bulunması ve o özelliklerin işlem yapılmak üzere seçilmesidir [43].

Modelin eğitilmesi

Veri, eğitim ve test verisi olarak ikiye ayrılır. Makine öğrenmesi modeli eğitim verisi üzerinde çalışır [43]. Üretilen tahminler ise test verisi ile karşılaştırılır.

Yapı çıkarımı

Öğrenme modeli, seçilen özellikler ve sonuç değişkeni üzerinde çalışarak veriye dair yapı ve mantıksal ilişkiler çıkarır. Seçilen özellik sayısının fazla olması, öğrenme modelinin daha karmaşık çıkarımlar yapabilmesini sağlar. Ancak modelin karmaşıklığı çok fazla olursa, bu da öğrenme değil ezberleme yapmasına neden olabilir. Bu yüzden bu denge dikkatli şekilde sağlanmalıdır [47]. Özellik sayısı öğrenmeye yeterli olacak kadar fazla, ezbere neden olmayacak kadar az olmalıdır.

Tahmin üretme

Öğrenilen yapıya dayanarak test verisindeki özelliklerden test verisindeki sonuç değerlerine dair tahminler yapılır [43].

Öğrenme testi

Yapılan tahminlerle gerçek sonuç verileri karşılaştırılarak doğruluk değeri hesaplanır [43].



3. ARAÇ VE HİZMETLER

3.1. Giriş

Salgın ile ilgili verileri işleyebilmek için öncelikle verilerin depolanması gereklidir. Bu verilerin “büyük veri” olması nedeniyle, fiziksel bir sunucuda depolama yapmak çok maliyetli olacaktır. Bu sebeple veriyi depolamak için bir Bulut Bilişim hizmetine ihtiyaç vardır. Ayrıca, veriden çıkarılacak tabloların da veri gibi sürekli güncellenmesi gerekmektedir. Sürekli çalışan, güncelleme olup olmadığını denetleyen, güncelleme olduğunda güncel veriyi elde edip depolayan ve üzerinde işlem yapacak bir program yazılmalıdır. Bu programın yazılması ve sürekli çalışır halde tutulması da maliyetli olacaktır.

Bu çalışma, Bulut Bilişim hizmetlerinden Amazon Web Services ekosisteminde geliştirilmiştir. AWS'nin kullanılan hizmetleri ve kullanım amaçları şu şekilde sıralanabilir:

- AWS Lambda: Veriyi internet üzerinden elde edip depolama.
- Amazon S3: Elde edilen verinin depolandığı alan.
- AWS Glue: Veri üzerinde işlem yapma.
- Amazon SageMaker: Veri üzerinde makine öğrenmesi gerçekleştirme.
- Amazon Forecast: Zaman serileri ile çalışarak tahmin üretme.
- Amazon QuickSight: İşlenen veri ile grafik üretme.

Çalışmanın bu bölümünde, kullanılan AWS araçlarının nasıl çalıştığı ve bileşenleri ile ilgili bilgi verilmiştir.

3.2. AWS Lambda

Lambda, fiziksel olarak sunucuya sahip olma gereksinimi olmadan kod çalıştırmaya olanak tanıyan bir bilgi işlem hizmetidir. İşletim sistemi bakımı, kapasite yönetimi, ölçeklendirme gibi yönetim işlemlerini kullanıcının iş yükü olmaktan çıkarır. Her çeşit uygulama ve endpoint için kod çalıştırılabilen bir bilgi işlem altyapısına sahiptir [52].

Lambda işlevlerine göre düzenlenen kod, kullanıcının istediği olaylar sonucunda veya belirttiği zamanlarda otomatik olarak çalıştırılır. Ayrıca, ayarlanan çalışma zamanları arasında kalan süreler için faturalandırma yapılmamaktadır. Bu durum önemli ölçüde maliyet düşürmektedir [52].

Lambda işlevlerini Amazon S3 gibi AWS hizmetlerinde veri işleme tetikleyicisi olarak kullanılabilir veya AWS ekosisteminde çalışan endpoint'ler yaratılabilir.

Bellek, CPU ve ağ gibi kaynakları Lambda yönettiği için, çalışma sırasında sistem özelliklerine müdahale edilemez. Sistem özelliklerine müdahale edilebilmesi için Amazon Elastic Compute Cloud ve AWS Elastic Beanstalk uygulamaları bulunmaktadır.

3.2.1. Lambda özellikleri

Lambda'nın özellikleri sayesinde ölçeklenebilir, güvenli ve kolayca genişletilebilir Lambda uygulamaları geliştirilebilmektedir [52]. Lambda'ya ait temel özellikler şunlardır:

Dosya sistemleri erişimi

Bir işlev, bir Amazon Elastic File System (Amazon EFS) dosya sistemi yerel bir dizine bağlayacak şekilde yapılandırılabilir. Amazon EFS ile işlev kodu, paylaşılan kaynaklara güvenli bir şekilde ve yüksek eşzamanlılıkla erişebilir ve bunları değiştirebilir.

Eşzamanlılık ve ölçekleme kontrolleri

Bu kontroller ile uygulamaların ölçeklendirilmesi sağlanmaktadır. Ayrıca uygulamaların yanıt verme hızı üzerinde denetim imkanı verilir ve böylece verimlilik artışı sağlanır.

İşlev planları

Bir işlev planı, Lambda'nın diğer AWS hizmetleri veya üçüncü taraf uygulamalarla nasıl kullanılacağını gösteren örnek kod sağlar. Taslaklar, Node.js ve Python çalışma zamanları için örnek kod ve işlev yapılandırma ön ayarlarını içerir.

Veri tabanı erişimi

Bir veri tabanı proxy'si, bir veri tabanı bağlantıları havuzunu yönetir ve bir işlevden gelen sorguları aktarır. Bu, bir işlevin veri tabanı bağlantılarını tüketmeden yüksek eşzamanlılık düzeylerine ulaşmasını sağlar.

Kapsayıcı görüntüleri olarak tanımlanan fonksiyonlar

Lambda işlevlerini oluşturmak, test etmek ve dağıtmak için tercih edilen kapsayıcı görüntü araçlarını, iş akışlarını ve bağımlılıkları kullanılabilir.

Lambda uzantıları

Lambda işlevlerini artırmak için Lambda uzantıları kullanılabilir. Örneğin, Lambda'yı izleme, gözlemleme, güvenlik ve yönetişimde kullanılan araçlarla daha kolay entegre etmek için uzantılar kullanılabilir.

Kod imzalama

Lambda için kod imzalama, yalnızca onaylanan geliştiricilerin yayınladığı değiştirilmemiş kodun Lambda işlevlerine yerleştirildiğinin doğrulanmasına olanak tanıyan güven ve bütünlük denetimleri sağlar.

3.3. Amazon S3

Bu çalışmada, verilerin depolanması için Amazon'un S3 hizmetinden yararlanılmıştır. S3 ismi, "Simple Storage Service" tanımlamasının kısaltmasıdır. S3, ölçeklenebilir, erişilebilir ve performanslı bir veri depolama hizmetidir [53]. İnternetteki herhangi bir siteden, istenilen boyutta veri alınıp depolanabilmesini sağlar. İnternet depolama alanı kullanan bir uygulama yazıldığında, veri deposu olarak S3 kullanılabilir [54]. S3 nesne tabanlı bir veri deposudur. S3'teki nesnelerin içinde bulunduğu depo bölümleri "bucket" olarak adlandırılır. Bucket'lar S3'teki verilerin düzenlenmesinde kullanılır. Örneğin bir projeye başlanırken bir bucket yaratılır ve bu projeye ilgili tüm veriler burada tutulur. Projeye ilişkili kişilerin projeye erişebilmesi için, ilgili bucket'a erişim izinleri verilir [55].

Amazon Simple Storage Service (Amazon S3), İnternet için depolamadır. İnternet ölçeğindeki bilgi işlem işlerini kolaylaştırmak üzere tasarlanmıştır.

Amazon S3, istenilen miktarda veriyi istenilen zaman, internetin herhangi bir yerinden depolamak ve almak için kullanılabilir basit bir internet hizmetleri arayüzüne sahiptir. Tüm geliştiricilere, Amazon'un kendi küresel internet sitesi ağını çalıştırmak için kullandığı aynı yüksek düzeyde ölçeklenebilir, güvenilir, hızlı ve ucuz veri depolama altyapısına erişim sağlar. Hizmet, ölçek faydalarını en üst düzeye çıkarmayı ve bu avantajları geliştiricilere aktarmayı amaçlamaktadır [56].

3.3.1. Amazon S3 kullanmanın avantajları

Amazon S3, basitlik ve sağlamlığa odaklanan minimal bir özellik kümesiyle oluşturulmuştur. Aşağıda, Amazon S3 kullanmanın avantajlarından bazıları verilmiştir [56]:

Bucket oluşturma: Bucket'lar, Amazon S3'teki veri depolama için temel kapsayıcılardır.

Verileri saklama: Bir bucket'ta sonsuz miktarda veri depolanabilir. Bir Amazon S3 klasörüne yüklenen her nesne 5 TB'a kadar veri içerebilir. Her nesne, geliştiricinin atadığı benzersiz bir anahtar kullanılarak depolanır ve geri alınır.

Veri indirme: Veriler, veri sahibi veya izin verilen başka kişiler tarafından indirilebilir.

İzinler: Amazon S3 paketine veri yüklemek veya indirmek isteyen diğer kişilere erişim verilebilir. Kimlik doğrulama mekanizmaları, verilerin yetkisiz erişimden korunmasına yardımcı olabilir.

Standart arayüzler: Herhangi bir internet geliştirme araç setiyle çalışmak üzere tasarlanmış standartlara dayalı REST ve SOAP arayüzleri kullanılır.

3.3.2. Amazon S3 kavramları

Amazon S3'ü etkili bir şekilde kullanmak için gereken temel kavramlar ve terminoloji şu şekilde sıralanır [56]:

- Bucket'lar

- Nesnelere
- Anahtarlar
- Bölgeler
- Amazon S3 veri tutarlılık modeli

Bucket'lar

Bucket, Amazon S3'te depolanan nesnelere paketleri gibi düşünülebilir. Her nesne bir pakette bulunur ve bir pakette birçok nesne olabilir. Birbiriyle ilişkili nesnelere veya aynı projeye ait nesnelere aynı paketlere konulur. Bir bucket'a bir nesne eklendiğinde Amazon S3 benzersiz bir sürüm kimliği oluşturacak ve bunu nesneye atayacak şekilde yapılandırılabilir [56].

Bucket kullanımının faydaları şunlardır [56]:

Amazon S3'te depolanan nesnelere erişim yönetimini sağlar.

Depolama ve veri aktarımı ücretlendirmelerinde hangi bucket'ın ne kadar maliyet oluşturduğu gözlemlenebilir.

Nesnelere

Nesnelere, Amazon S3'te depolanan temel varlıklardır. Nesnelere, nesne verileri ve meta verilerden oluşur. Veri kısmı, Amazon S3 için görüntülenebilir değildir. Meta veriler, nesneyi tanımlayan bir dizi ad-değer çiftidir. Bunlar, son değiştirilme tarihi gibi bazı varsayılan meta verileri ve Content-Type gibi standart HTTP meta verilerini içerir. Ayrıca, nesnenin saklandığı anda özel meta veriler de belirtilebilir.

Bir nesne, bir bucket içinde bir anahtar (ad) ve bir sürüm kimliği ile benzersiz şekilde tanımlanır.

Anahtarlar

Anahtar, bucket içindeki bir nesnenin benzersiz tanımlayıcısıdır. Bir bucket'taki her nesnenin tam olarak bir anahtarı vardır. Bucket, anahtar ve sürüm kimliğinin kombinasyonu, her bir nesneyi benzersiz şekilde tanımlar. Dolayısıyla, Amazon S3, "bucket + anahtar + sürüm" ile nesnenin kendisi arasındaki temel bir veri haritası

olarak düşünülebilir. Amazon S3'teki her nesne, web hizmeti uç noktası, bucket adı, anahtar ve isteğe bağlı olarak bir sürümün birleşimi aracılığıyla benzersiz şekilde adreslenebilir. Örneğin, <https://mybucket.s3.amazonaws.com/2020-03-01/AmazonS3.wSDL> URL'sinde "mybucket" bucket'ın adıdır ve "2020-03-01 / AmazonS3.wSDL" anahtardır.

Bölgeler

Amazon S3'ün oluşturulan bucket'ları depolayacağı coğrafi AWS bölgesi kullanıcı tarafından seçilebilir. Gecikmeyi optimize etmek, maliyetleri en aza indirmek veya yasal gereklilikleri karşılamak için bölge seçimi yapma seçeneği sunulmuştur. Bir bölgede depolanan nesnelere, kullanıcılar onları açıkça başka bir bölgeye transfer etmediği sürece farklı bir bölgeye kesinlikle aktarılmaz.

Amazon S3 veri tutarlılık modeli

Amazon S3, tüm AWS bölgelerinde Amazon S3 klasöründeki nesnelere yazma ve silme komutları için güçlü bir tutarlılık sağlar [56].

Tek bir anahtara yapılan güncellemeler atomiktir. Örneğin, bir iş parçasından var olan bir anahtara PUT işlemi yapılırsa ve aynı anda ikinci bir iş parçasından aynı anahtar üzerinde bir GET işlemi gerçekleştirilirse, eski veriler veya yeni veriler ile dönüş sağlanır. Kullanıcının bozuk veya kısmi veri ile karşılaşması imkansızdır. Bucket yapılandırılmalarının nihai bir tutarlılık modeli vardır. Bir bucket silinip hemen tüm bucket'lar listelenirse, silinen bucket listede görünmeye devam edebilir [56].

3.4. AWS Glue

Güncellemelerin algılanıp harekete geçildiği bu sistemde, gelen her verinin kendine özel tipi olduğundan, her biri için özel olarak yazılmış meta veri çıkarma algoritmaları hazırlanmalıdır. Son olarak, elde edilen meta veriler ile anlamlı tablolara oluşturmak için işlem yapılması ve tabloların oluşturulması gerekir.

Amazon şirketinin analitik ve büyük veriyi işlemek için ürettiği hizmet, düşük maliyetli, Bulut Bilişim Teknolojileri üzerinde kolay entegre edilebilir ve kullan-öde ücretlendirme sistemine sahiptir. Analitik ve büyük veri üzerinde işlem yapmak için

gerekli olan ETL işlemlerini kolaylaştıran AWS Glue hizmeti, bu çalışmada verileri elde etmek, işlemek ve tablolar çıkarmak için kullanılmıştır.

Amazon'un ETL (extract, transform, load) hizmeti Glue, veriyi kategorize etmeyi, ayıklamayı, çeşitlendirmeyi ve farklı veri depoları ile veri akışları arasında taşımayı sağlamaktadır. Bu işlemleri güvenilir ve uygun maliyetli şekilde gerçekleştirmektedir. AWS Glue bileşenleri; AWS Glue Catalog (meta veri havuzu), Python veya Scala kodunu otomatik olarak yaratan bir ETL motoru ve esnek bir zamanlayıcıdır [57].

Glue sunucusuz bir hizmettir, altyapı gerektirmemektedir. Yarı yapılandırılmış verilerle çalışabilmektedir. Bileşenlerinden biri olan dinamik çerçevedir ve ETL komut dosyalarında kullanılabilir. Dinamik çerçeve, verileri satır ve sütunlar formunda getirebilmek için kullanılan bir veri soyutlaması olan Apache Spark'a benzer. Ancak dinamik çerçevelerde her kaydın kendi kendini tanımlaması ve şemasının olmasına ihtiyaç yoktur. Şema esnekliği ve özel olarak tasarlanmış ileri seviye bir dönüştürme sağlamaktadır. Ayrıca dinamik çerçeveler ile Apache Spark veri çerçeveleri arasında dönüştürme yapılabilir. İhtiyaca göre hangisi daha uygunsa, hızlıca o çerçeveye geçiş yapılabilir [57].

İstenen verilere ulaşma, onları dönüştürme, ihtiyaca uygun şekle getirme işleri AWS Glue konsolunda gerçekleştirilebilir. Veriler üzerinde yapılacak işlemlerle ilgili gerekli temel altyapı bu konsolda mevcuttur. AWS Glue hizmetini kullanmak için kendi AWS Glue API'si kullanılabilir gibi, Python veya Scala gibi geliştirme ortamları da kullanılabilir [57].

AWS Bulut verileri AWS Glue ile ihtiyaca uygun forma dönüştürülebilir, istenilen veri deposuna aktarılabilir. Verilere ayıklama, eşleştirme, yeniden oluşturma, biçimlendirme gibi işlemler yapılabilir. Veriler üzerinde bu değişiklikler yapılırken veya veriler sadece depolanırken, verilerle ilgili loglar tutulabilir ve bu loglardan analiz çıkarılmasına olanak sağlar. Verilerin ortak bir veri gölünde veya veri ambarında tutulmasının en önemli avantajı, farklı birimler tarafından ihtiyaç duyulan verilerin depolanmasında tekrara düşmemek ve kaynak tasarrufu sağlamak olabilir [57].

Bir veri gölü veya veri ambarı oluştururken karşılaşılan birçok zorluk, AWS Glue ile kolaylaştırılmıştır. Bu zorlukları kolaylaştırmak için AWS Glue'nun yaptığı işlemler şu şekilde sıralanabilir [57]:

- Veri depoları ile ilişkili büyük verileri bularak ortak bir katalogta tutar.
- Yarı yapılandırılmış verileri işlemeyi kolaylaştırır.
- Veri kataloğunu oluştururken verileri tarayıcı programlarından otomatik olarak alarak kataloğu doldurur. Verileri veri kataloğuna yerleştirmek için verilerin biçimini ve türünü saptamaya çalışır. Bu saptamayı da veriler üzerinde sınıflandırıcı algoritmalar çalıştırarak yapar. Kısmen yapılandırılmış veriler, veri kataloğunda tablolar ile tutulur.
- Veriler üzerinde uygulanacak ETL (extract, transform, load) işlemleri için gerekli olan komut dosyalarını kendisi oluşturur.
- Verilerin biçimlerinde değişiklik olduğunda, değişikliği algılayarak kullanıcının tercihine uygun şekilde yeniden yapılandırır.
- ETL işlemlerini, kullanıcının isteğine göre belli bir olaya veya zamana göre tetikleyebilir. Ayrıca, işlerin birbiriyle arasındaki başlangıç-bitiş ilişkisini tanımlamak için de tetikleyiciler kullanılabilir.
- Veri gölündeki veya veri ambarındaki işlemlerin takip edilebilmesini kolaylaştırmak için zaman ölçümleri yapar.
- Veri işlemleri yapılırken performans kaybı yaşanmaması için kaynakları uygun şekilde ölçeklendirmektedir.
- ETL job'ları tetikleyici ile çalıştırılabileceği gibi sorgu ile de başlatılabilir.
- Amazon S3 veri gölündeki verileri kataloglar ve ETL işlemlerine hazır hale getirir.
- Mevcut verilerin ölçülmesini hızlandırır. Farklı AWS hizmetlerinde tuttuğunuz verilerin bütün halde AWS Veri Kataloğunda tutulmasını sağlar. Veri kataloğu görüntülendiğinde veri kümelerinde hızlıca arama yapılabilir.

3.4.1. AWS Glue'nun çalışma mantığı

AWS Glue, veri gölü veya veri ambarı oluşturacak ETL işlemlerini düzenlemek için diğer AWS hizmetlerini kullanmaktadır. Veriler ile ilgili yapılacak birçok işlem için API işlemlerini çağırır. AWS Glue konsolu, bu işlemleri otomatik olarak yapacak şekilde tasarlanmıştır. Böylece kullanıcının yapacağı iş, ETL işlemlerini tasarlamak

ve logları takip etmekten ibaret olmaktadır. AWS Glue'ya kişi bilgileri tanımlanmış kullanıcılar, veri kaynaklarına erişebilir ve işlem yapabilir [58].

AWS Glue, işlemlerin yapılabilmesi için gerekli kaynakları sağlar ve yönetir. Kullanıcıların altyapı kurmasını gerektirmez, kendi altyapısı mevcuttur. İşlemler için kaynak gerektiğinde, hızlı şekilde kaynağa ulaşır ve hizmet sağlar.

Kullanıcılar AWS Glue ile veri kataloğundaki tabloları kullanarak job'ları oluşturur. Bu job'lar verilerin üzerinde yapılacak işlemlerin komut dosyalarıdır. İşlem yapmak istenilen verilerin depolanacağı yeri, işlem yapılacak verileri, işlemlerin hangi olay veya zamanla başlatılacağı kullanıcı tarafından belirlenir. AWS Glue ise bu belirlenen özelliklere uygun şekilde çalışacak kodu üretmektedir. Veriler üzerinde yapılacak işlemler için AWS Glue konsolunda komut dosyaları üretilebilir [58].

Veri kaynakları

AWS Glue'nun desteklediği veri kaynakları: Veri depoları, Amazon S3, Amazon RDS (Amazon Relational Database Service), third party JDBC erişimli veri tabanları, Amazon DynamoDB, veri akışları, Amazon Kinesis Veri Akışları, Apache Kafka.

Veri hedefleri

AWS Glue'nun desteklediği veri hedefleri: Veri depoları, Amazon S3, Amazon RDS (Amazon Relational Database Service), third party JDBC erişimli veri tabanları.

AWS Glue, ETL işlemlerini sunucusuz bir ortam sunan Apache Spark'ta çalıştırır. Kullanıcı verilerini ayrıştırır ve veriler aktarılırken veya kullanımda değilken korur. Kullanıcı verilerine erişilmesi gerektiğinde, AWS Glue bu verilerin kapsamını olabildiğince daraltarak veri güvenliğini en yüksekte tutmaya çalışır. Kullanıcı verilerine ancak kullanıcının isteğiyle veya kullanıcının hesabındaki IAM rollerinin onayı için erişilir [58].

Bir ETL işinin gerçekleştirilebilmesi için, kullanıcıların VPC (sanal özel bulut)'sinde veri kaynakları ve veri hedefleri bulunur. Kullanıcı, bu kaynaklara erişim için gerekli IAM rolünü, alt ağ ve VPC kimliğini sağlamalıdır. AWS Glue, her grup için diğer tüm ortamlardan yönetim bazında ayrıştırılmış yeni bir Spark ortamı oluşturur ve grup

bilgilerini burada tutar. Spark job'ları, veri kaynaklarına ve hedeflerine erişim için ağ arayüzüne ihtiyaç duyar. AWS Glue, kullanıcının özel IP adresi ile kullanıcının alt ağında esnek ağ birimleri oluşturur. Spark ortamındaki veri trafiğini VPC ve ağ ilkeleri yönetir. Bir istek geldiğinde AWS Glue, VPC ile AWS Glue API işlemlerine giden trafiği proxy olarak kullanabilir. AWS Glue API ile yapılan işlemlerin tamamı loglanır. İsteğe bağlı olarak, AWS CloudTrail kullanılarak veri erişimleri kolayca denetlenebilir. AWS, Spark ortamlarını da diğer hizmetlerinde uyguladığı güvenlik önlemlerini uygulayarak korumaktadır [58].

3.4.2. AWS Glue ile ilgili temel kavramlar

AWS Glue'da veriler üzerinde ETL işlemlerini gerçekleştirmek için job'lar tanımlanır.

Job, ETL işlemlerinin gerçekleştirilmesini sağlayan mantıktır. ETL komut dosyalarının içinde veri depoları (işlenecek verilerin alınacağı veri kaynakları ve işlenen verilerin depolanacağı veri hedefleri) bulunur. Job'lar, bir olaya ya da zamana bağlı olarak tetiklenerek çalıştırılabileceği gibi, istenilen zamanda doğrudan da çalıştırılabilir. Veri kaynaklarından verileri elde eden, ETL işlemlerini gerçekleştiren komut dosyalarından oluşur [59].

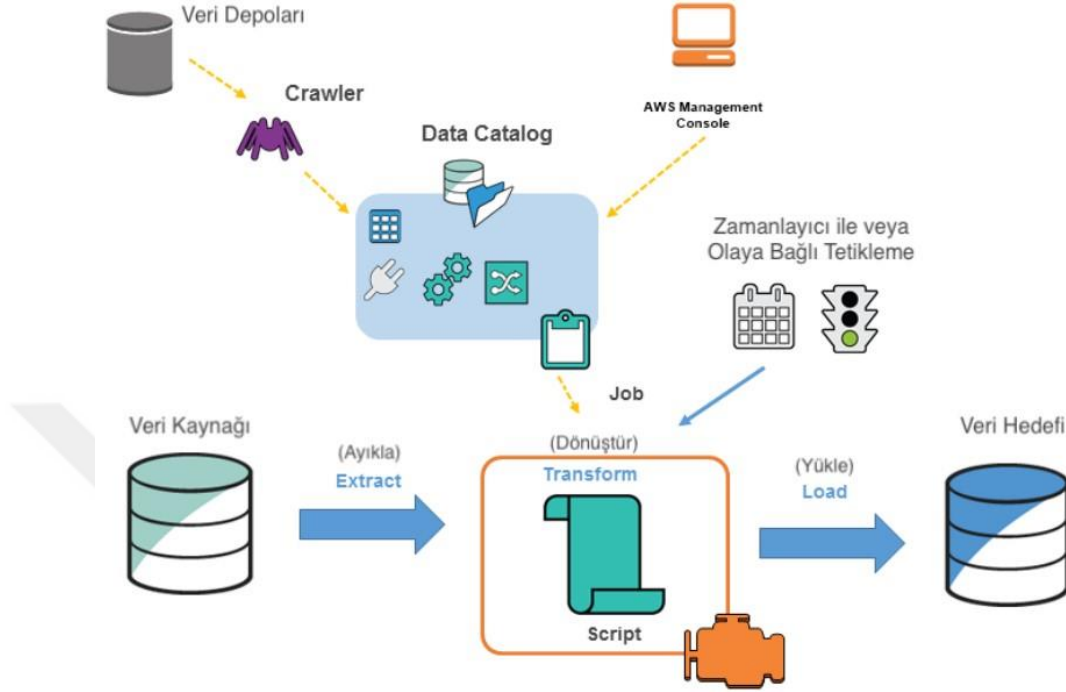
Anlamında kayıp yaşanmaması için bu çalışmada "ETL job" ismi değiştirilmeden kullanılmıştır.

ETL job'ları Python dilinde yazılmak istenirse, PySpark kütüphanesi kullanılabilir.

Veriler üzerinde ETL job'ları tanımlamak, onları bir zamanlayıcıya bağlamak ve çalıştırmak, AWS Glue Jobs sistemi ile yapılmaktadır.

ETL job'larını çalıştırmak için gerekli olan tüm meta veriler AWS Glue Veri Kataloğu'ndadır. Bir job tanımlandığında, buradaki meta veri kullanılır. Veri metin tabanlı ise UTF-8 formatında olmalıdır. AWS Glue'daki veri tabanları ve tablolar, AWS Glue Veri Kataloğu'nda nesnelere olarak bulunur. Veriler üzerinde işlem yapmak için gerekli komut dosyası kullanıcı tarafından AWS Glue konsolunda veya API'de hazırlanabileceği gibi AWS Glue tarafından da hazırlanabilir. Şekil 3.1'de mimari görsel olarak verilmiştir. Bu komut dosyası, AWS Glue'da bir Apache Spark

ortamında çalışmaktadır. Tanımlanan job, isteğe, zamana veya olaya bağlı olarak başlatılabilir [59].



Şekil 3.1. AWS Glue Mimarisi

AWS Glue Veri Kataloğu, AWS Glue'daki kalıcı meta veri deposu olarak tanımlanabilir. Tablo tanımlarını, job'ları ve yönetim bilgilerini içermektedir [59].

Sınıflandırıcılar, verilerin şemasını belirlemektedir. Verilerin XML, JSON, CSV gibi dosya türlerinde sınıflandırılmasının yanı sıra, kullanıcılar kendi sınıflandırıcılarını yazabilmektedir [59].

Bağ, bir AWS Glue Veri Kataloğu nesnesidir. Belirlenmiş veri deposuna bağlanmak için gerekli özellikleri içerir [59].

Paletli, verilerin şemasını belirlemek için sınıflandırıcılar listesinde öncelik sırasına göre ilerleyen ve AWS Glue Veri Kataloğu'nda meta veri tabloları oluşturan bir programdır [59].

Veri tabanı, ilişkisel Veri Kataloğu tablolarının bütünüdür [59].

Veri deposu ise, verilerin kalıcı olarak depolandığı bir havuzdur. İlişkisel veri tabanları ve Amazon S3 veri tabanları, veri depoları örnekleridir. İki çeşit veri deposu

bulunmaktadır. İlki, uygulanacak bir işlemde girdi olarak kullanılacak veri kaynağıdır. İkincisi ise işlemin sonucunun kaydedileceği veri deposudur [59].

Geliştirme uç noktası, ETL komut dosyalarının geliştirilmesi ve test edilmesi için kullanılan ortamdır [59].

Dinamik çerçeve, iç içe geçmiş verileri destekleyen dağıtık bir tablo olarak tanımlanır. ETL komut dosyaları dinamik çerçeveleri veya Apache Spark veri çerçevelerini kullanabilir. Her bir veri kaydı kendi özelliklerini belirtir. Yarı yapılandırılmış verilerin de tabloya eklenebilmesi için şema esnekliğine uygundur [59].

Veriler, meta veriler olarak tablolarda tanımlanır. Veri deposunun çeşidi fark etmeksizin, tüm verilerin şeması bir tabloda tanımlanır. AWS Glue Veri Kataloğu'ndaki bir tabloyu oluşturan bileşenler; sütun isimleri, bölüm bilgileri, veri çeşidi bilgileri, temel veri kümesi ve meta veriler olarak sıralanabilir. AWS Glue tablosunda verilerin yalnızca şeması tanımlanırken, verilerin kendisi, mevcut veri depolarında saklanmaya devam eder. Bu ilişkisel tablolar AWS Glue Veri Kataloğu'nda kataloglanır. Veriler üzerinde ETL job çalıştırılmak istendiğinde, AWS Glue Veri Kataloğu hem kaynak veri deposu hem de hedef veri deposu olarak kullanılarak işlem gerçekleştirilir [59].

ETL job'lar, bir olay veya zamana bağlı olarak başlatmayı sağlayan kod sistemi tetikleyicilerdir. Ayrıca sürekli devam eden bir işlem akışı üzerinde de bir başka ETL işlemi tanımlanabilir [59].

3.4.3. AWS Glue'nun bileşenleri

AWS Glue nesnelерinin (job'lar, tarayıcılar, tablolar vs.) tanımlanması, tetikleyicilerin planlanması, ETL dosyalarının düzenlenmesi, AWS Glue nesnelерinde arama ve filtreleme yapma gibi işler için AWS Glue Veri Kataloğu'nda ve AWS Glue Jobs sisteminde birçok API işlemleri çağrılmaktadır [60].

AWS Glue Veri Kataloğu, AWS Cloud'da meta verilerin depolanmasını, verilerle ilgili bilgiler eklemeyi, düzenleme yapmayı ve verileri paylaşımaya açmayı sağlayan hizmettir. Farklı sistemlerdeki veri depolarını tarayabilmek, meta verilere ulaşım işlemler yapabilmek için tek bir çeşit veri havuzu sağlamaktadır. AWS Glue Veri

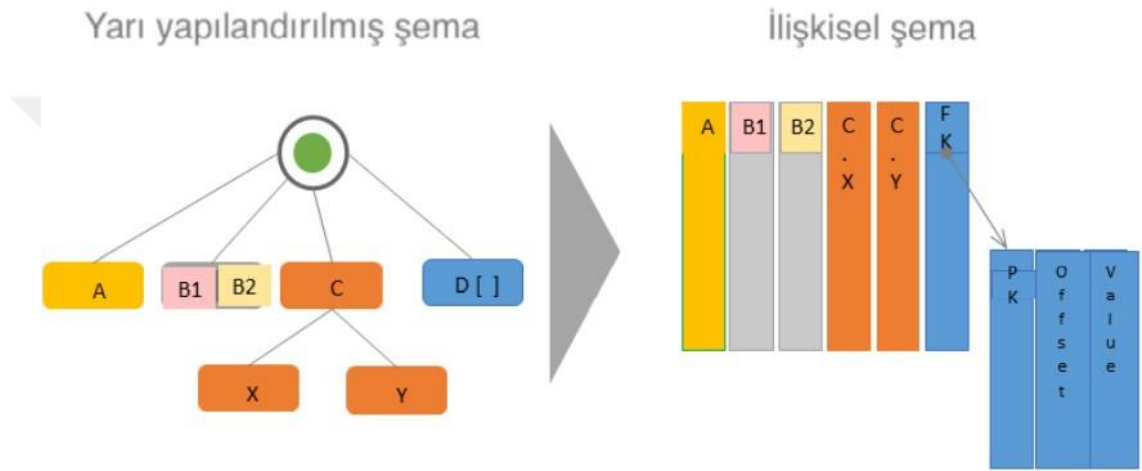
Katalogu, şema deęişiklerini izlemeye olanak saęlar. Veriler üzerinde istenmeyen bir deęişiklik yapılmasına veya verilerin istenmeyen kullanıcılarla paylaşılmasına karşı önlem oluşturmayı kolaylaştırır. Verilere erişimlerinin yönetimi, AWS Glue Veri Kataloęu'nun yaratıcısının isteęi doęrultusunda gerçekleşir. Bu veri kaynaklarına kimlerin erişeceğini yönetmek için AWS IAM (Identity and Access Management) politikaları kullanılmaktadır. AWS IAM politikaları sayesinde veriler hem daha güvenli şekilde paylaşımına açılır, hem de paylaşılmak istenmeyen veriler daha güvenli şekilde saklanır. Kullanıcıların coęrafi konumları fark etmeksizin doęru ve tutarlı bir yolla politika işlemeyi sürdürür [60].

Çeşitli veri havuzlarındaki verilerin işlenebilir hale gelmesi için öncelikle onlardan şema bilgileri ve meta veriler üretmek gerekmektedir. AWS Glue, bu işlemi yapmak ve meta verileri AWS Glue Veri Kataloęu'na otomatik olarak kaydetmek için crawler ve classifier adında iki sistem kullanmaktadır.

Crawler ve Classifier (Tarayıcı ve Sınıflandırıcılar), ETL job'larının çalışması için gerekli olan komut dosyaları, kullanıcı tarafından yazılabileceęi gibi, AWS Glue tarafından otomatik olarak da üretilebilir. AWS Glue Veri Kataloęu'ndaki şemalardan ham veri elde edilebilir, veri istenilen şekilde işlenebilir ve başka bir veri deposuna saklanmak üzere gönderilebilir. Bu işlemleri yapmak üzere hazırlanması gereken komut dosyası kullanıcı tarafından hazırlanmak istenirse Scala veya Spark kullanılabilir. ETL jobları, yalnızca depolanmış veriler üzerinde deęil, çalışmakta olan akış verileri üzerinde de çalıştırılabilir. ETL akışı, Apache Spark'ın Yapılandırılmış Akış Motoru üzerinde inşa edilmiştir. AWS Glue Streaming ETL, İot akışlarını, web sayfalarındaki tıklamaları, aę ve kullanıcı hareketleri logları ve benzeri olay kayıtlarını işlemek için kullanılır. Bu akış verilerini edindięi bazı uygulamalar; Amazon Kinesis Data Streams ve Apache Kafka'dır. Akışların veri kaynaęı şeması bilindięi sürece, bu akış AWS Glue Veri Kataloęu'na eklenebilir. Veri kaynaęı şemasının bilinmedięi durumda ise, Streaming ETL job'ında otomatik şema algılama aktif hale getirilerek, şema elde edilir [60].

Yarı yapılandırılmış veya yapılandırılmamış verilerin ilişkisel veri şemalarına çevrilmesi, daęınk katmanlara sahip bir yapıyı, ilişkisel ve daha az katmanlı bir yapıya çevirmek olarak düşünülebilir. Yarı yapılandırılmış veriler iç içe geçmiş veri

yapılarını, eksik bilgili verileri, sıralama veya filtreleme yapmayı imkansız hale getirebilecek yapı bozukluklarını içerebilir. İlişkisel veri şemalarında, anlamlı satır ve sütunlarla yaratılmış tablo verileri, tablolara ait primary key (birincil anahtar) ve foreign key (yabancı anahtar) ilişkileri mevcuttur. İlişkisel veri şemaları oluşturmak, büyü veriyle ilgili karşılaşılan en büyük sorunların başında gelir. AWS Glue, bu büyük sorunu çözmeyi hedeflemektedir. İlişkisel veri şemalarını yaratmak için crawler'lar kullanıldıktan sonra ETL job'ları ile karmaşık veriler bu şemalara dönüştürülür. Bu işlemi açıklayan bir diyagram aşağıda verilmiştir:



Şekil 3.2. Yarı yapılandırılmış verilerin ilişkisel veri şemalarına dönüştürülmesi

Şekil 3.2'de verilen şemadakiler şöyle açıklanır [61]:

- A kutucuğu tek bir değer olduğu için tek bir ilişkisel bir sütuna dönüşmüştür.
- B1 ve B2 kutucukları iki ilişkisel sütuna dönüşmüştür.
- C kutucuğunun X ve Y çocukları olduğu için iki ilişkisel sütuna dönüşmüştür.
- D bir dizidir ve başka bir ilişkisel tabloyu işaret eden bir foreign key ile ilişkisel bir sütuna dönüşmüştür. Primary key ile birlikte, ikinci ilişkisel tablo, dizideki öğelerin ofsetini ve değerini içeren sütunlara sahiptir.

3.5. Amazon SageMaker

Amazon SageMaker, makine öğrenmesi modellerinin hızlıca oluşturulup kullanılabilmesini sağlayan bir servistir [62].

Bu çalışmada zaman serileri makine öğrenmesi yöntemi kullanılmıştır.

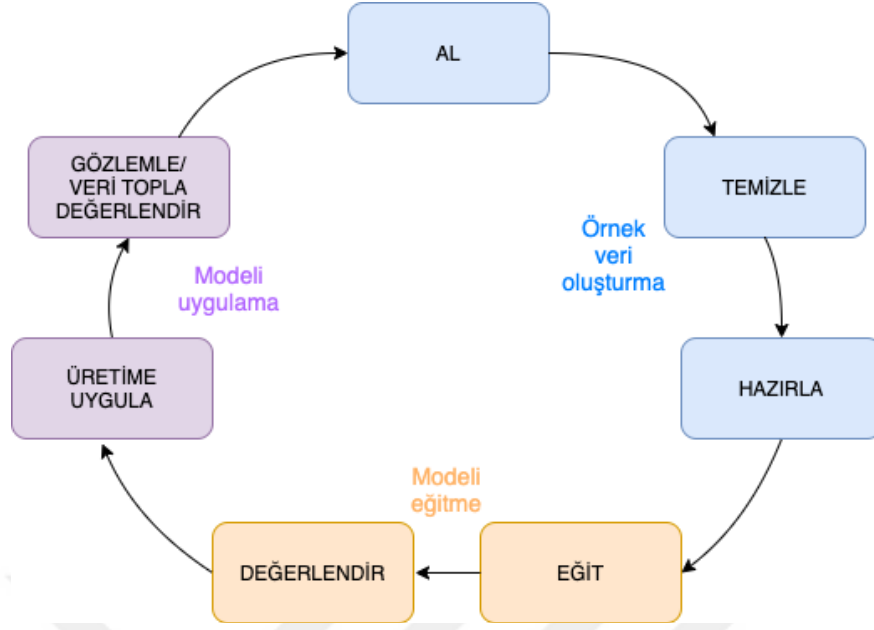
Makine öğrenmesi modellerinin eğitilmesi için öncelikle büyük ölçüde verinin yönetilmesi ve verinin eğitilmesi için en uygun algoritmanın seçilmesi gerekmektedir. Eğitim sırasında hesaplamaların yapılabilmesi ve makine öğrenmesi modelinin bir üretim ortamında uygulanması gerekir. Amazon SageMaker bu işlemleri daha sade hale getirmektedir. İçerdiği seçenekler arasından projeye uygun olan algoritmayı seçmek yeterlidir [62].

Amazon SageMaker, tümüyle yönetilen bir makine öğrenimi hizmetidir. SageMaker ile veri bilimcileri ve geliştiricileri, makine öğrenimi modellerini hızlı ve kolay bir şekilde oluşturup eğitebilir ve ardından bunları üretime hazır bir barındırılan ortama doğrudan dağıtabilir. Keşif ve analiz için veri kaynaklarına kolay erişim için entegre bir Jupyter yazma not defteri örneği sağlar, böylece sunucuların kullanıcılar tarafından yönetilmesine ihtiyaç kalmaz. Ayrıca, dağıtılmış bir ortamda son derece büyük verilere karşı verimli bir şekilde çalışacak şekilde optimize edilmiş yaygın makine öğrenimi algoritmaları sağlar. Kullanıcı tarafından seçilen makine öğrenmesi algoritmaları ve veri kaynaklarına kolayca erişim sağlayabilen bir altyapıya sahiptir. Bir model, SageMaker Studio veya SageMaker konsolundan birkaç basit işlemle başlatılarak güvenli ve ölçeklenebilir bir ortama yüklenebilir. Eğitim ve depolama, minimum ücret ve peşin taahhüt olmaksızın kullanım dakikalarına göre faturalandırılır [63].

3.5.1. İş akışı

Makine öğreniminde, bir bilgisayara tahminler veya çıkarımlar yapma öğretilir. İlk olarak, bir modeli eğitmek için bir algoritma ve örnek veriler kullanılır. Ardından, gerçek zamanlı ve uygun ölçekte çıkarımlar oluşturmak için model uygulamaya entegre edilir. Bir üretim ortamında, bir model tipik olarak milyonlarca örnek veri ögesinden öğrenir ve 20 milisaniyenin altında yüzlerce çıkarım üretir [64].

Amazon SageMaker'ın çalışma modeli makine öğrenmesi algoritmalarının çalışma modeline paraleldir. Şekil 3.3'teki şema, Amazon SageMaker kullanılarak bir makine öğrenimi modeli oluşturmak için tipik iş akışını göstermektedir:



Şekil 3.3. Makine öğrenimi modeli için iş akışı

Şemada gösterilen adımlar şöyle açıklanır [64]:

1. Örnek veri oluşturma: Bir modeli eğitmek için örnek verilere ihtiyaç vardır. İhtiyaç duyulan veri türü, modelin çözmesi istenilen iş sorununa bağlıdır. Örneğin, el yazısıyla yazılmış bir rakamın giriş görüntüsü verilen bir sayıyı tahmin etmek için bir model oluşturmak üzere geliştirme yapılsın. Böyle bir modeli eğitmek için el yazısıyla yazılmış örnek resimlere ihtiyaç var.

Veri bilimcileri, model eğitimi için kullanmadan önce, örnek verileri araştırmak ve ön işlem yapmak için genellikle çok zaman harcarlar. Verileri önceden işlemek için genellikle şu adımlar izlenmektedir:

- Verileri elde edilmesi: Kullanıcının kendine ait veri kaynakları veya herkese açık olan veri kümeleri kullanılabilir. Genel olarak, veri kümesi veya veri kümeleri tek bir depoya yüklenir.
- Verilerin temizlenmesi: Model eğitimini iyileştirmek için veriler incelenir ve gerektiği şekilde ayklanır.
- Verilerin dönüştürülmesi: Performansı artırmak için ek veri dönüşümleri yapılması gerekebilir. Örneğin, nitelikleri birleştirmek ya da gerekli görülmeyen nitelikleri tamamen silmek gibi işlemler yapılabilir.

SageMaker'da, örnek verileri işlemek için tarayıcıda bir Jupyter not defterinde önceden işlemek mümkündür. Veri kümesini almak, keşfetmek ve model eğitimi için hazırlamak da Jupyter not defterinde yapılır.

2. Bir model eğitme: Model eğitimi modelin hem eğitilmesini hem de değerlendirilmesini içerir. Bir modeli eğitmek için bir makine öğrenmesi algoritması kullanılır. Seçilen algoritma bir dizi değişkene bağlıdır. SageMaker'ın sağladığı algoritmalarından biri kullanılabileceği gibi, kullanıcı tarafından seçilen herhangi bir makine öğrenmesi algoritması da kullanılabilir.

Ayrıca eğitim için bilgi işlem kaynaklarına ihtiyaç var. Eğitim veri kümesinin boyutuna ve sonuçlara ne kadar hızlı ihtiyaç duyulduğuna bağlı olarak, tek bir genel amaçlı bulut sunucusu kullanılabileceği gibi, dağıtılmış bir GPU bulut sunucusu kümesi de kullanılabilir.

- Modelin değerlendirilmesi: Modelin eğitilmesinden sonra, çıkarımların doğruluğunun kabul edilebilir olup olmadığını belirlemek için değerlendirme yapılır. SageMaker'da, ya AWS SDK for Python (Boto) ya da SageMaker'ın sağladığı Python kitaplığı kullanılabilir. Modeli eğitmek ve değerlendirmek için SageMaker'da Jupyter not defteri kullanılır.

3. Modelin işletilmesi: SageMaker depolama hizmetleriyle, modeli uygulama kodundan ayırarak bağımsız olarak işletmek mümkündür. Makine öğrenimi sürekli bir döngüdür. Bir modeli uyguladıktan sonra, çıkarımlar ve test verilerini karşılaştırılır ve eğitim verileri yeni toplanan verileri içerecek şekilde güncellenerek çıkarımların doğruluğu artırılır. Bu işlem, model yeni veri kümesiyle yeniden eğitilerek tekrarlanır.

3.6. Amazon Forecast

Ürün talepleri, ekonomi, hava durumu gibi veri kümelerinin tarih verisine bakarak geleceğe yönelik tahminler üretmek birçok kurum için yüksek önem taşımaktadır. Amazon Forecast, Amazon'un zaman serileri ile çalışarak tahminler üretmeyi sağlamak için geliştirdiği hizmettir. Amazon Forecast, Amazon'un kendi alış-veriş sitesinde kullanılan teknolojiyle aynı kökten temel alır. Tam olarak yönetilen bir

hizmet olduđu için tedarik edilecek sunucular ve oluşturulacak, eğitilecek veya dağıtılacak makine öğrenimi modelleri yoktur [65].

Amazon Forecast için bazı yaygın kullanım örnekleri şunlardır [66]:

- Perakende talep planlaması: Ürün talebi tahmin edilerek, farklı mağaza konumlarında envanterin ve fiyatlandırmanın daha doğru bir şekilde değiştirilmesine olanak tanır.
- Operasyonel planlama: Web trafiği, AWS kullanımı ve IoT sensör kullanımı düzeyleri tahmin edilebilir.
- Kaynak planlaması: Personel, reklam, enerji tüketimi ve sunucu kapasitesi için gereksinimler tahmin edilebilir.

Özellikleri

Amazon Forecast, zaman serileri ile tahmin sürecinin çoğunu otomatikleştirir ve kullanıcının yalnızca veri kümelerini hazırlaması ve tahminleri yorumlaması yeterli olur [66].

Otomatikleştirilmiş makine öğrenimi: AutoML, algoritma seçimi, hiperparametre ayarlama, yinelemeli modelleme ve model değerlendirme gibi karmaşık makine öğrenimi görevlerinin otomatikleştirilmesine olanak tanır [66].

Son teknoloji algoritmalar: Kullanıcılarına, Amazon.com'da kullanılan makine öğrenmesi teknolojilerini temel alan algoritmaları kullanma fırsatı sunar. Forecast, yaygın olarak kullanılan istatistiksel yöntemlerden karmaşık sinir ağlarına kadar çok çeşitli eğitim algoritmaları içerir [66].

Eksik değer desteği: Forecast, veri kümelerindeki eksik değerleri otomatik olarak işlemek için çeşitli doldurma yöntemleri uygular [66].

Yerleşik özellikler: Forecast, modeli geliştirmek için yerleşik özellikli veri kümelerini otomatik olarak dahil edebilir. Bu veri kümeleri zaten özel olarak tasarlanmıştır ve ek yapılandırma gerektirmez [66].

Önceden oluşturulmuş bir algoritma seçilerek veya Amazon Forecast'in yüklenen veri kümesine göre en iyi algoritmayı seçmesini sağlamak için AutoML seçeneğini

belirleyerek bir makine eğitimi başlatılır. Daha sonra tahminler üretilir ve görselleştirilebilir [66].

3.7. Amazon QuickSight

Amazon QuickSight, anlaşılması kolay çıkarımlar sunmayı sağlayan bulut bilişim tabanlı bir iş zekası hizmetidir. Amazon QuickSight, buluttaki verilere bağlanır ve birçok farklı kaynaktan gelen verileri birleştirir. QuickSight, tek bir veri panosunda AWS verilerini, üçüncü taraf verilerini, büyük verileri, elektronik tablo verilerini, SaaS verilerini, B2B verilerini ve daha fazlasını içerebilir. Tümüyle yönetilen bulut tabanlı bir hizmet olan Amazon QuickSight, kurumsal düzeyde güvenlik, küresel kullanılabilirlik sağlar. Ayrıca, tümü dağıtılacak veya yönetilecek altyapı olmadan binlerce kullanıcıya ulaşmayı mümkün kılar [67]. QuickSight, karar vericilere bilgileri etkileşimli bir görsel ortamda keşfetme ve yorumlama fırsatı verir. Ağınızdaki herhangi bir cihazdan ve mobil cihazlardan gösterge tablolarına güvenli erişime sahiptirler.

Analiz, veri görselleştirme ve raporlama için Amazon QuickSight kullanmanın avantajlarından bazıları şunlardır [67]:

- Çok hızlı yanıt sağlar.
- Düşük maliyetlidir.
- Uygulama yüklemeye gerek kalmadan analiz sağlar.
- Çeşitli verileri tek bir analizde birleştirebilir.
- Analizi bir gösterge tablosu olarak yayınlamak mümkündür.
- Ayrıntılı veri tabanı izinlerini yönetmeye gerek yoktur.

Amazon QuickSight kullanımı ile yapılabilecek daha detaylı çalışmalar şu şekilde sıralanabilir [67]:

- Makine öğrenimi (ML) ile desteklenen otomatik ve özelleştirilebilir veri çıkarımlarıyla zamandan ve paradan tasarruf sağlar.
- Otomatik olarak güvenilir tahminleri yapılabilir.
- Aykırı değerleri otomatik olarak belirler.
- Önemli iş faktörlerine göre hareket eder.
- Verileri, gösterge tablosu için başlık alanı gibi okunması kolay grafiklere çevirir.

- Ekstra kurumsal güvenlik özellikleri sağlar.
- AWS Identity and Access Management Federation, SAML, OpenID Connect veya Microsoft Active Directory için AWS Directory Service ile kullanıcılar, gruplar ve çoklu oturum açma imkanı sağlar.
- AWS veri erişimi için ayrıntılı izinler düzenlenebilir.
- Satır düzeyinde güvenlik seçenekleri vardır.
- Yüksek güvenli veri şifreleme özelliğine sahiptir.
- Amazon Virtual Private Cloud'da AWS verilerine erişime olanak sağlar.
- Gömülü konsol analitiği ve gösterge tablosu oturumları dağıtılarak QuickSight web sitelerinin ve uygulamaların bir parçası haline getirilebilir.
- Diğer AWS hesaplarına aktarılabilen pano şablonları programlı bir şekilde komut dosyası oluşturulmasına olanak tanır.
- Analitik varlıklar için paylaşılan ve kişisel klasörlerle erişim yönetimini ve organizasyonu basitleştirir.

3.7.1. Gösterge tabloları

Amazon QuickSight'ta veri panosu; çizelgeler, grafikler ve çıkarımlardan oluşan bir koleksiyondur. Gösterge tabloları, yapılmak istenen analitiklere bağlı olarak çok çeşitli tasarımlara sahiptir. QuickSight kullanılarak, bir web sayfasında veya cep telefonundaki verilerle etkileşim sağlanabilir [67].

Verilerin anlattığı olaylar, gösterge tablolarını oluşturan analistlerin ve veri bilimcilerinin uzmanlığına bağlı olarak değişkenlik gösterebilmektedir. Analistler ve veri bilimciler verileri iyileştirir, hesaplamalar ekler, olayın açılarını bulur ve nasıl sunulacağına karar verir. Yayıncı, gösterge tablosunu tasarlar ve onu etkileşimli veri görselleştirmeleri ve kontrolleri ile doldurur. Yayıncılar, filtre dahil olmak üzere mevcut etkileşim düzeyini özelleştirebilir ve arama seçenekleri, filtreleme, sıralama, detaya inme gibi işlemlere olanak sağlar [67]. Bir kontrol panelini görüntülediğinde, en son alınan veriler gösterilir. Ekrandaki öğeler ile etkileşime girildikçe, yapılan herhangi bir değişiklik kontrol panelinin görünümünü değiştirir. Böylece yayıncı hangi öğelere bakıldığını raporlar ama cihaz gizliliğiniz sağlanmaya devam eder. Kontrol paneli kapandıktan sonra, keşifler ve veriler saklanmaz [67].

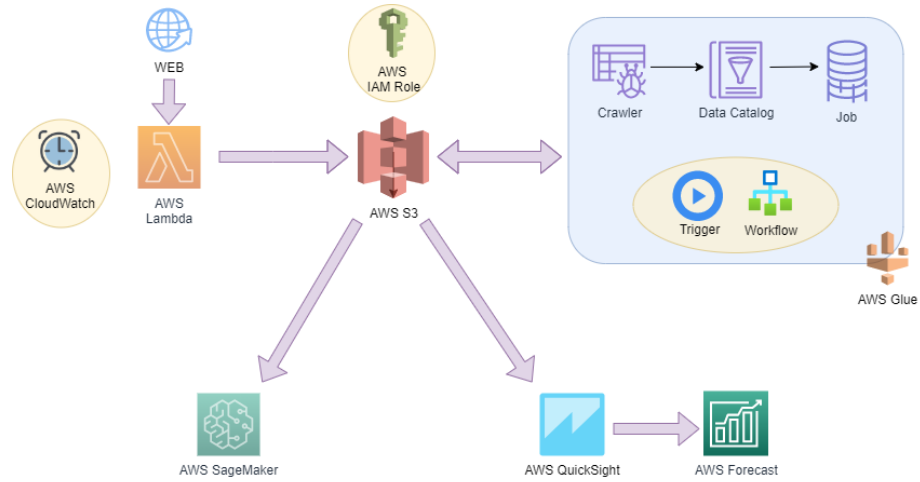
4. UYGULAMA ADIMLARI

4.1. Giriş

Çalışmada izlenen adımlar maddeler halinde şu şekilde sıralanabilir:

- Ortamı oluşturma
- IAM rollerini ayarlama
- İnternet sitelerinden veri elde etme
- Crawler ile Data Kataloğu'nu oluşturma
- ETL job'u oluşturma
- Workflow tasarlama
- Triger'larla job ve crawler'ları çalıştırma
- Veri görüntüleme
- Veriyi makine öğrenmesi için hazırlama
- Makine öğrenmesinin gerçekleştirilmesi
- Tahminler üretme
- Tahmin sonuçlarını iyileştirmek için makine öğrenmesini yeniden tasarlama
- Makine öğrenmesi sonuçlarından çıkarım yapma

Çalışmada izlenen yol ve kullanılan tüm araçlar Şekil 4.1'deki diyagramda gösterilmiştir.



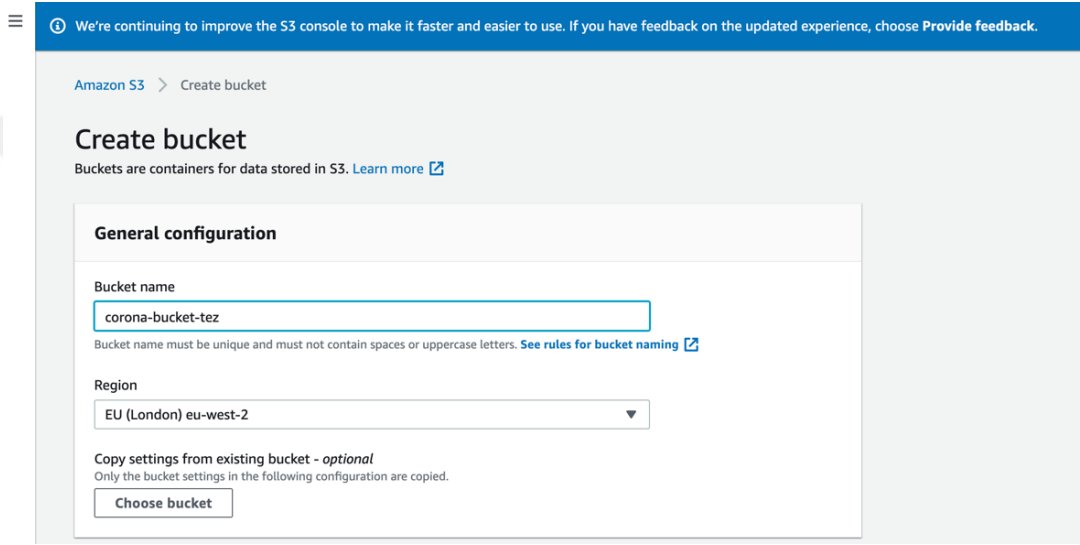
Şekil 4.1. Çalışma diyagramı

4.2. Ortamın Oluşturulması

İşlenecek korona veri kümesi Kaggle'dan temin edilmiştir.

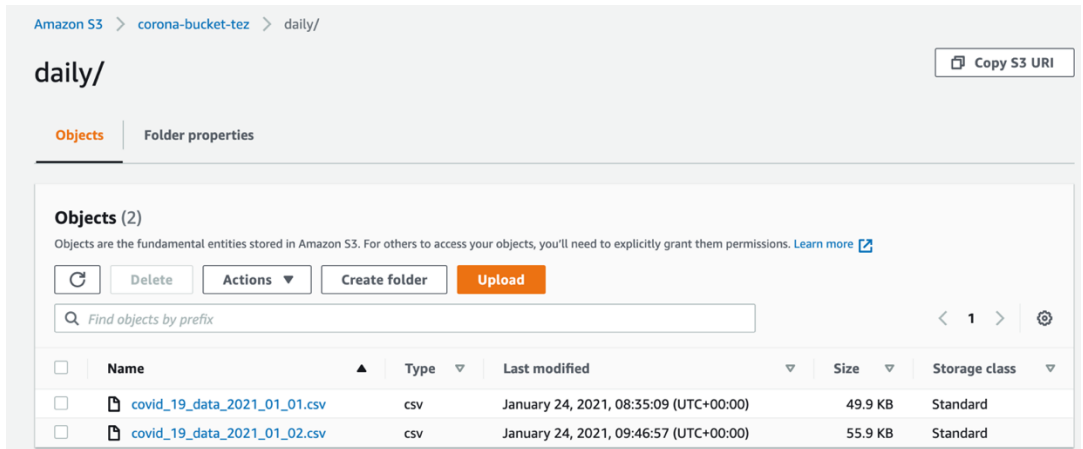
<https://www.kaggle.com/sudalairajkumar/novel-corona-virus-2019-dataset>

Veriler üzerinde işlem yapmak için öncelikle veri kaynaklarının ayarlanması gerekmektedir. Bunun için Amazon S3 konsolunda bir bucket yaratılmıştır. Şekil 4.1'de bucket yaratma sayfası gösterilmiştir.



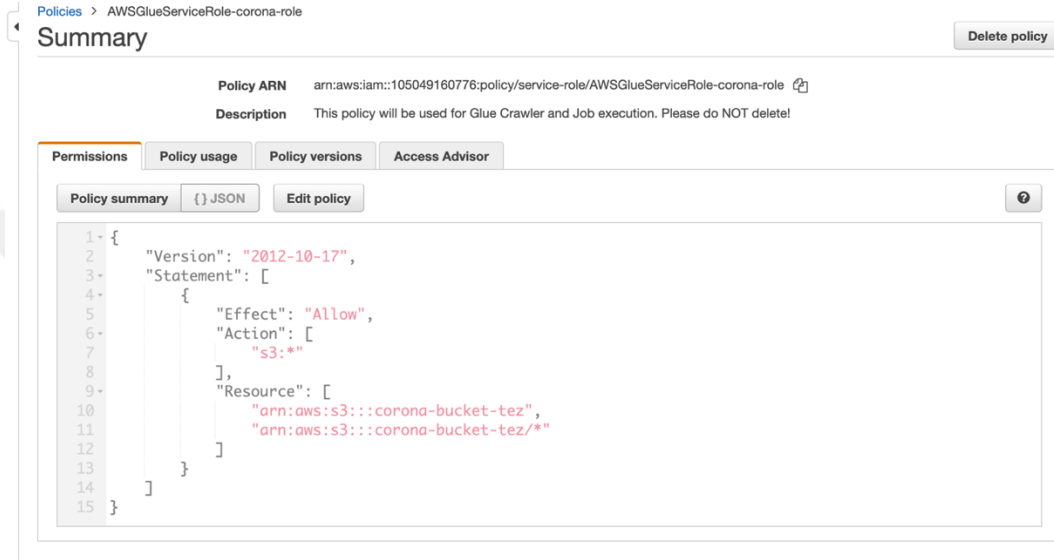
Şekil 4.2. Amazon S3 konsolunda bucket yaratılması

Şekil 4.2'de gösterildiği gibi Amazon S3'teki bucket'a iki günlük covid19 veri kümesi eklenmiştir.



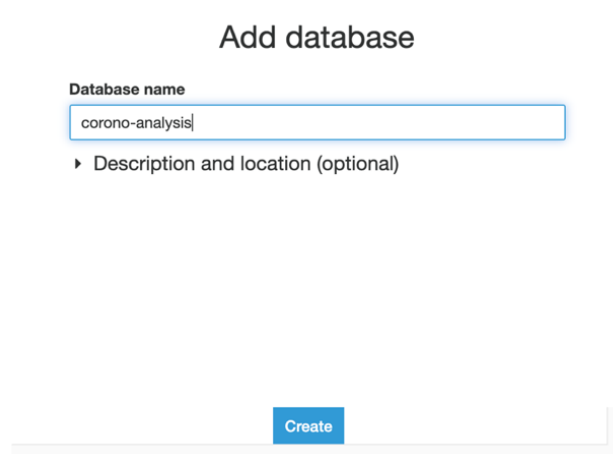
Şekil 4.3. Yaratılan bucket'a eklenen veri dosyaları

IAM rollerini ayarlama: AWS Glue'da rol yaratmak için IAM konsolu kullanılmaktadır. "AWSGlueServiceRole-corona-role" adında bir IAM rolü yaratılmış ve AWSGlueServiceRole politikası bu role eklenmiştir. Böylece bu roldeki kullanıcıların diğer AWS platformlarına erişimine izin verilmiştir. Korona veri kümesi AWS S3'te tutulduğu için, bu izin gereklidir. Şekil 4.4'te role eklenen izinlerle ilgili kod parçası gösterilmiştir.



Şekil 4.4. IAM role eklenen kod parçası

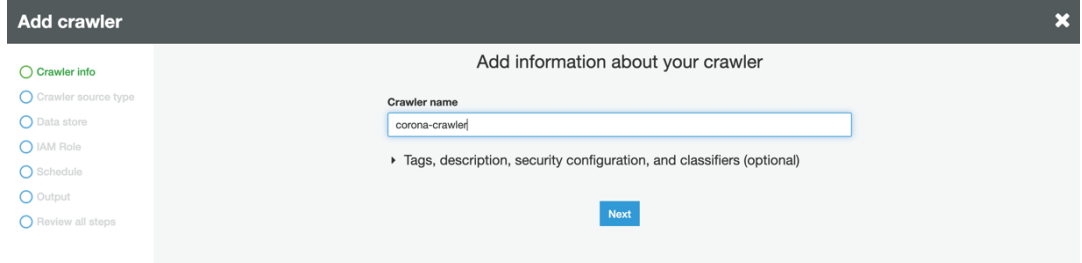
Veri kümesinden çıkarılacak katalogun tutulabilmesi için bir veri tabanı oluşturmak gereklidir. Şekil 4.5'te veri tabanı oluşturma ekranı gösterilmiştir.



Şekil 4.5. Database oluşturma ekranı

4.3. Verinin Elde Edilmesi

Crawler ile data kataloğunu oluşturma: Daha sonra veri kümesini kataloglayacak olan crawler yaratılmış ve özellikleri Şekil 4.6, Şekil 4.7, Şekil 4.8, Şekil 4.9, Şekil 4.10, Şekil 4.11, Şekil 4.12 ve Şekil 4.13'te gösterildiği gibi ayarlanmıştır.



Add crawler [X]

Add information about your crawler

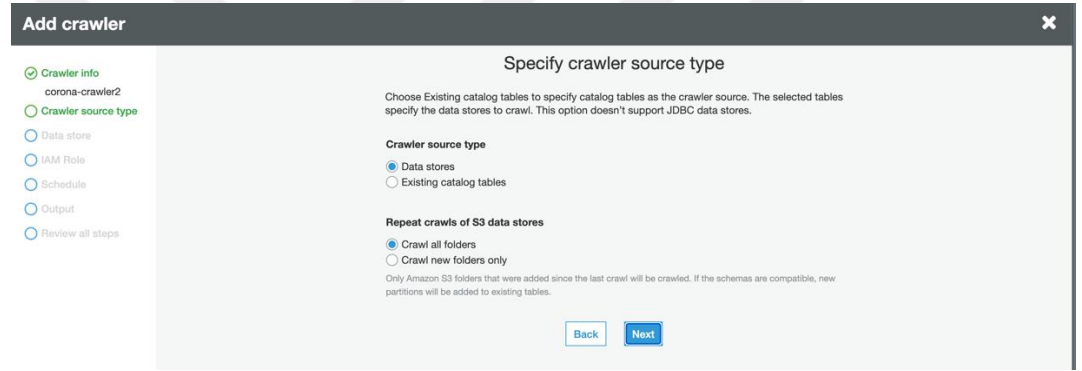
- Crawler info
- Crawler source type
- Data store
- IAM Role
- Schedule
- Output
- Review all steps

Crawler name
corona-crawler

Tags, description, security configuration, and classifiers (optional)

Next

Şekil 4.6. Crawler'a isim verilmesi



Add crawler [X]

Specify crawler source type

Choose Existing catalog tables to specify catalog tables as the crawler source. The selected tables specify the data stores to crawl. This option doesn't support JDBC data stores.

Crawler source type

- Data stores
- Existing catalog tables

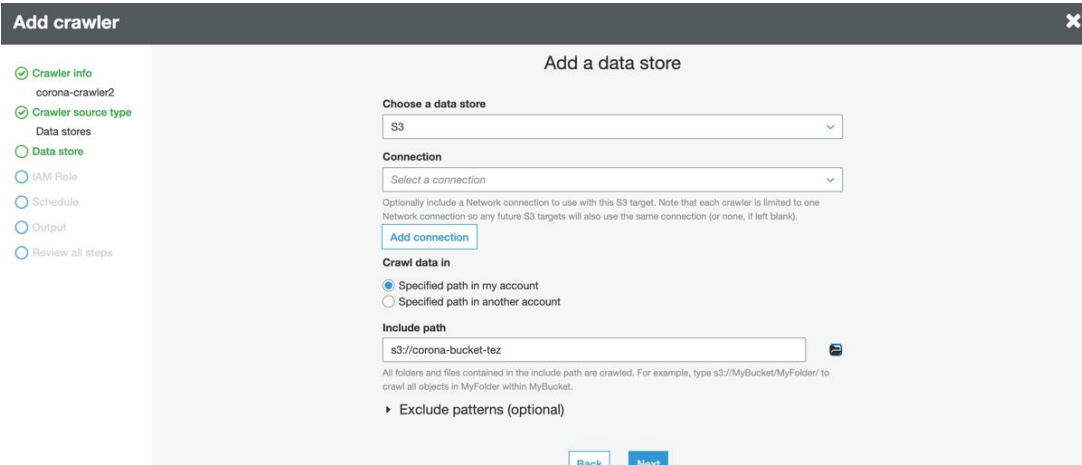
Repeat crawls of S3 data stores

- Crawl all folders
- Crawl new folders only

Only Amazon S3 folders that were added since the last crawl will be crawled. If the schemas are compatible, new partitions will be added to existing tables.

Back **Next**

Şekil 4.7. Crawler için kaynak tipi belirlenmesi



Add crawler [X]

Add a data store

Choose a data store
S3

Connection
Select a connection

Optionally include a Network connection to use with this S3 target. Note that each crawler is limited to one Network connection so any future S3 targets will also use the same connection (or none, if left blank).

Add connection

Crawl data in

- Specified path in my account
- Specified path in another account

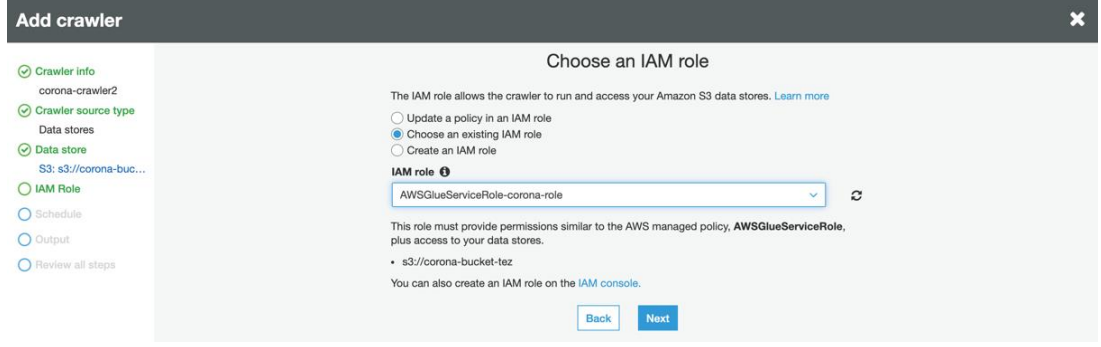
Include path
s3://corona-bucket-tez

All folders and files contained in the include path are crawled. For example, type s3://MyBucket/MyFolder/ to crawl all objects in MyFolder within MyBucket.

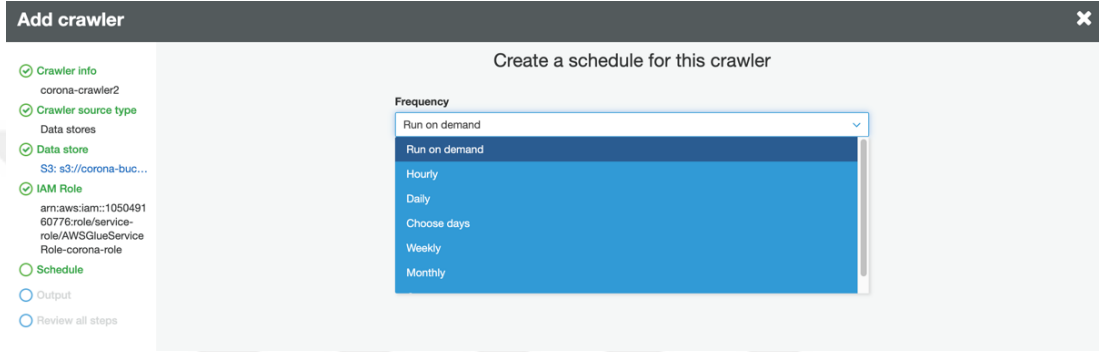
Exclude patterns (optional)

Back **Next**

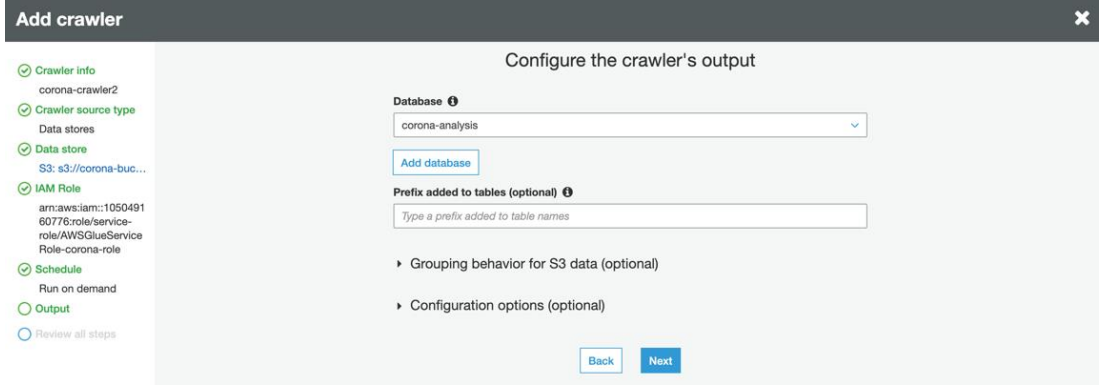
Şekil 4.8. Veri kaynağının belirlenmesi



Şekil 4.9. Crawler'a erişim izni olan IAM rolü seçilmesi

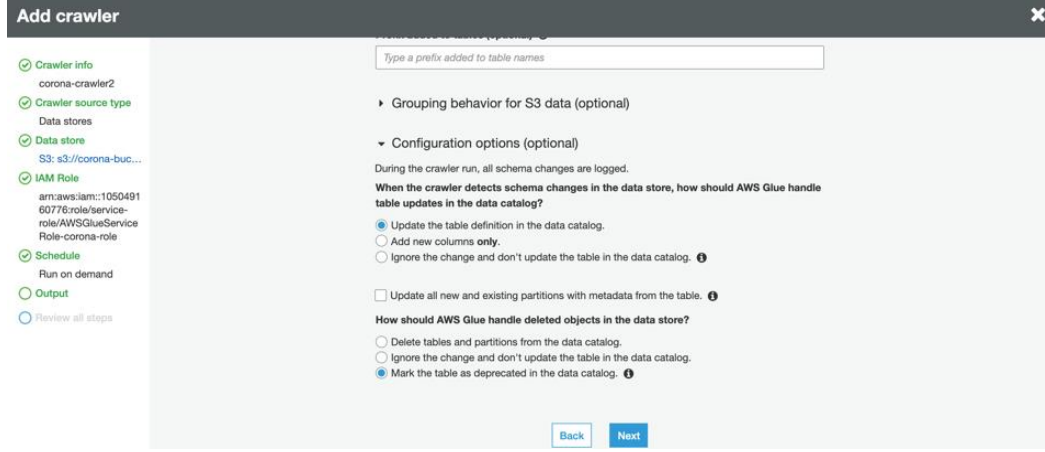


Şekil 4.10. Crawler'ın çalışma zamanlamasının seçilmesi

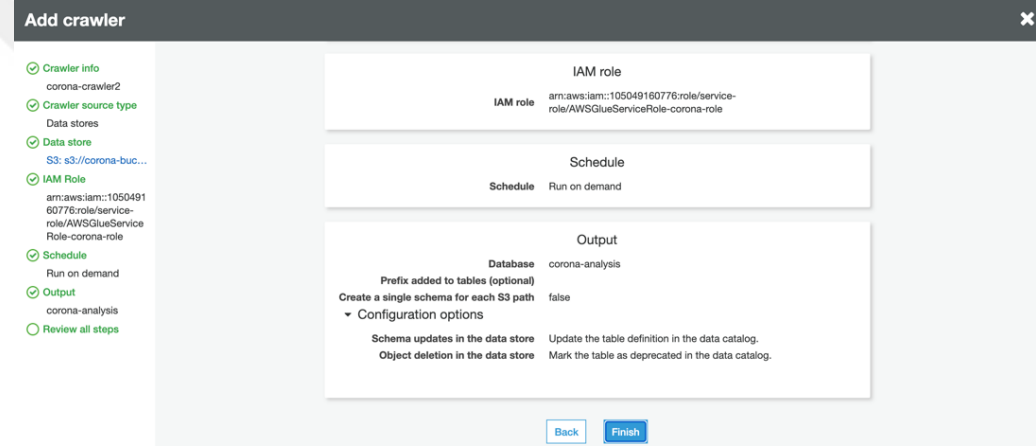


Şekil 4.11. Crawler çıktısının kaydedileceği veri tabanının seçilmesi

Şekil 4.12'de gösterilen düzenleme, iş akışı açısından oldukça önemlidir. Burada veri kaynağında değişiklik olması durumunda crawler'ın tetiklenip çalışması için gerekli ayarlama yapılmıştır. Böylece veri şeması, daima veri kümesinin en güncel haline uygun şekilde otomatik olarak oluşturulacaktır. Farklı senaryolar için, veri kaynağı değişikliği ile tetiklenmesi istenmeyebilir. Bu durumlarda istek gönderildiğinde çalışacak şekilde ayarlama yapılabilir

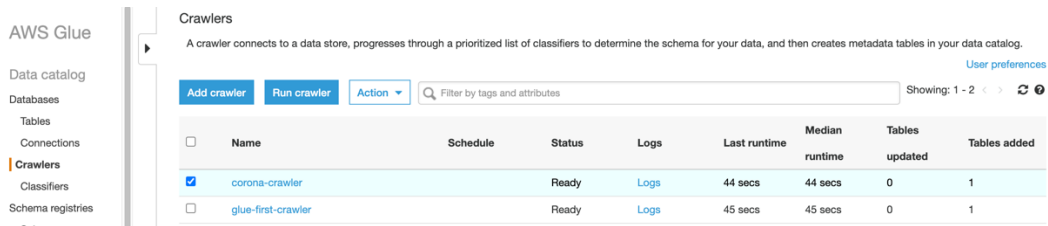


Şekil 4.12. Veri kaynağında değişiklik olması durumunda AWS Glue'nun ne yapacağını belirlenmesi



Şekil 4.13. Crawler'ı oluşturmanın tamamlanışı

Şekil 4.14'te görüldüğü gibi, crawler başarılı şekilde çalışmış ve eklenen tablolar kolonunda 1 rakamı görülmüştür. Şekil 4.15'teki database listesinden, bu crawler için hedef olarak gösterilen corona-analysis database'i açıldığında "daily" tablosu oluşturulmuştur. Bucket'a daha önce iki günlük veri yüklenmişti. Bu verinin üzerine günlük veriler birer birer ya da istenilen sayıda toplu dosyalar halinde eklendiğinde crawler yeni dosyaları da dahil ederek tabloyu oluşturur. Şekil 4.16'da crawler'lar tarafından oluşturulan tablolar listesi gösterilmiştir.



Şekil 4.14. Çalışması tamamlanan crawler

AWS Glue

Data catalog

Databases

Tables

Connections

Crawlers

Databases A database is a set of associated table definitions, organized into a logical group.

Add database View tables Action

Showing: 1 - 2

| Name | Description |
|-----------------|-------------|
| corona-analysis | |
| glue-first-db | |

Şekil 4.15. Database listesi

AWS Glue

Data catalog

Databases

Tables

Connections

Crawlers

Tables A table is the metadata definition that represents your data, including its schema. A table can be used as a source or target in a job definition.

Add tables Action Database: corona-analysis Filter or search for tables... Save view Showing: 1 - 2

| Name | Database | Location | Classification | Last updated | Deprecated |
|-------------|-----------------|-------------------------------------|----------------|------------------------------|------------|
| daily | corona-analysis | s3://corona-bucket-tez/daily/ | csv | 29 January 2021 11:04 AM UTC | |
| glue_output | corona-analysis | s3://corona-bucket-tez/glue-outp... | parquet | 29 January 2021 11:09 AM UTC | |

Şekil 4.16. Crawler'ların oluşturduğu tablolar

Şekil 4.17 ve Şekil 4.18'de crawler'ın tüm dosyalara bakarak oluşturduğu "daily" tablosunun içeriği görüntülenmektedir.

Tables > daily

Last updated 29 Jan 2021 11:04 AM Table Version (Current version)

Edit table Delete table View properties Compare versions Edit schema

Name daily

Description

Database corona-analysis

Classification csv

Location s3://corona-bucket-tez/daily/

Connection

Deprecated No

Last updated Fri Jan 29 11:04:37 GMT+000 2021

Input format org.apache.hadoop.mapred.TextInputFormat

Output format org.apache.hadoop.hive.q1.io.HiveIgnoreKeyTextOutputFormat

Serde serialization lib org.apache.hadoop.hive.serde2.lazy.LazySimpleSerDe

Serde parameters field.delim ,

skip.header.line.count 1 sizeKey 14931562 objectCount 1 UPDATED_BY_CRAWLER corona-crawler

Table properties CrawlerSchemaSerializerVersion 1.0 recordCount 210303 averageRecordSize 71 CrawlerSchemaDeserializerVersion 1.0 compressionType none columnsOrdered true areColumnsQuoted false delimiter , typeOfData file

Şekil 4.17. Daily tablosuna kaydedilen korona veri kümesinin özellikleri

Schema

Showing: 1 - 8 of 8

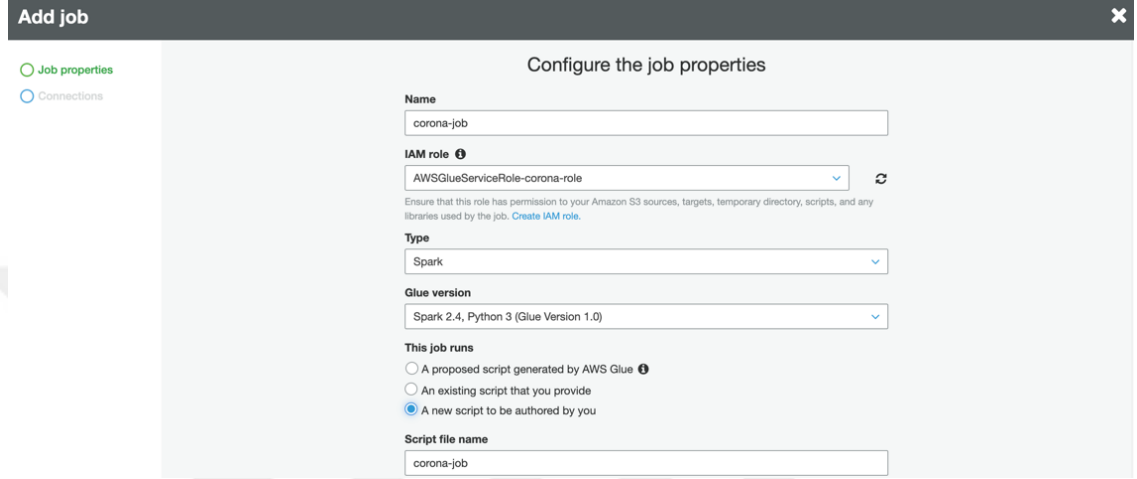
| Column name | Data type | Partition key | Comment |
|-------------------|-----------|---------------|---------|
| 1 sno | bigint | | |
| 2 observationdate | string | | |
| 3 province/state | string | | |
| 4 country/region | string | | |
| 5 last update | string | | |
| 6 confirmed | double | | |
| 7 deaths | double | | |
| 8 recovered | double | | |

Şekil 4.18. Daily tablosuna kaydedilen korona veri kümesinin kolonları ve tipleri

4.4. Verinin İşlenmesi

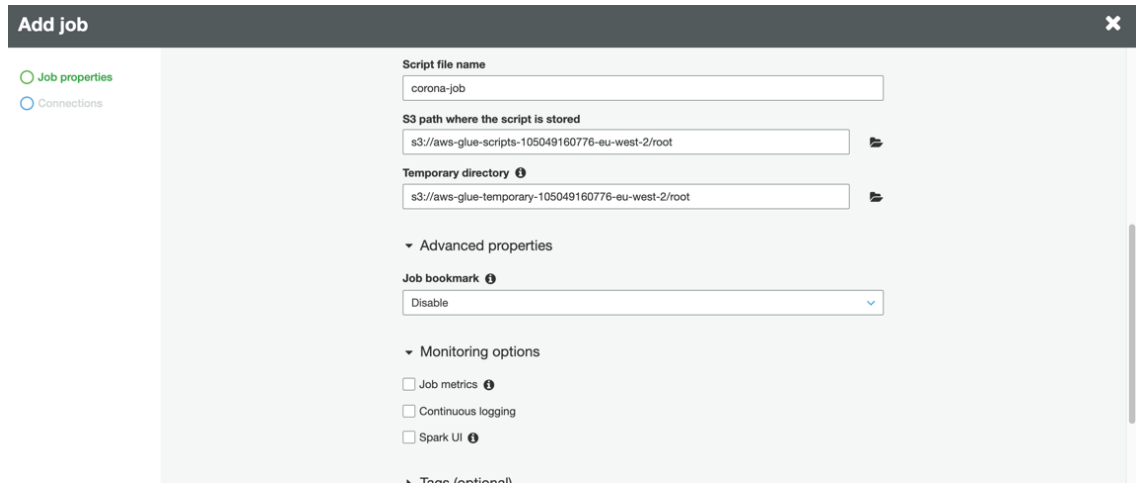
ETL Job'u oluşturma: Veri kümesi kataloglandıktan sonra, artık veri kümesi üzerinde

ETL işlemleri yapılabilir. Veri kümesi üzerinde ETL işlemleri yapmak için Glue job oluşturmak gereklidir. AWS Glue, job oluşturacak olan python veya pyspark kodunu kendisi üretebilmektedir. Job yapısının işleyişinin ve yapısının anlaşılması için, ilk job yeni bir script yazılarak oluşturulmuştur. Job oluşturma aşamasındaki seçenekler Şekil 4.19, Şekil 4.20 ve Şekil 4.21'deki gibi seçilmiştir.



Şekil 4.19. Job oluştururken isim verilmesi, IAM rolü ve kod tipi seçilmesi, neyi çalıştıracağını belirlenmesi

- IAM role: Bu analiz için yaratılan IAM role seçilmiştir.
- Type: Spark seçilmiştir.
- A new script to be authored by you: Job'ın, yönetici tarafından yazılan yeni bir script'i çalıştıracağını ifade etmektedir.



Şekil 4.20. Job oluştururken script'e isim verilmesi ve script'in yerinin belirtilmesi

The screenshot shows the 'Add job' configuration interface. On the left, there are two tabs: 'Job properties' (selected) and 'Connections'. The main area contains several configuration fields:

- Referenced files path:** A text input field containing 's3://bucket/prefix/object'.
- Worker type:** A dropdown menu showing 'G.1X (Recommended for memory-intensive jobs)'.
- Number of workers:** A text input field containing '2'. Below it, a note states: 'The maximum number of workers you can define are 299 for G.1X, and 149 for G.2X.'
- Max concurrency:** A text input field containing '1'.
- Job timeout (minutes):** A text input field containing '2880'. Below it, a note states: 'The default is 2,880 minutes (48 hours).'
- Delay notification threshold (minutes):** An empty text input field.
- Number of retries:** A text input field containing '0'.

Şekil 4.21. Job oluştururken donanım tipi, aynı anda çalışan job sayısı ve zaman aşımını değerlerinin belirlenmesi

- **Worker type:** Job'ın çalıştırılacağı donanım tipini ifade eder. Verinin özelliğine göre önerilen tip seçilmiştir.
- **Max concurrency:** Aynı anda en fazla kaç job'ın çalışacağı, bu özellik ile belirlenir.
- **Job timeout:** Bu özellik ile belirlenen süre aşıldığında, job çalışmayı durdurur.

Glue Job bu özellikleriyle kaydedildikten sonra script yazma işlemine geçilmiştir. Bu script, AWS S3'te tutulan csv formatındaki dosyaları okur, ETL işlemlerini gerçekleştirir, ülke ve bölgelerin en fazla vaka, en fazla ölüm ve en fazla iyileşme sayılarını listeler ve oluşturduğu listeleri hedef olarak verilen adrese parquet formatında kaydeder.

Parquet açık kaynak kodlu bir dosya formatıdır [68]. Veriyi sütun şeklinde depolar ve sıkıştırır [69]. Düz sütunlu dosya formatının satır tabanlı dosya formatlarına göre daha performanslı olduğu düşünülerek geliştirilmiştir [68].

Aynı veri için, csv formatına göre dosya büyüklüğünü yaklaşık %80 oranında azaltırken, veri sorgulama hızını 30 katına çıkarır [69].

Şekil 4.22'de verilen kod parçası; Glue job'ta veri kaynaklarının tanımlanmasını, veri kataloğundan veri kaynaklarının belirlenmesini, veri kümesindeki bozuk satır ve sütunların temizlenmesini sağlamaktadır. Şekil 4.23'teki kod parçasında ise bu işlemlerin çıktılarının daha önceden belirlenmiş olan path'lere kaydedilmesi işlemi yapılmıştır. Böylece job script tamamlanmıştır.

```

1 import sys
2 from aws glue.transforms import *
3 from aws glue.utils import getResolvedOptions
4 from pyspark.context import SparkContext
5 from aws glue.context import GlueContext
6 from aws glue.dynamicframe import DynamicFrame
7 from pyspark.sql.functions import col
8 from aws glue.job import Job
9
10 args = getResolvedOptions(sys.argv, ['JOB_NAME'])
11
12 # Glue and Spark kaynaklari yaratilir
13 sc = SparkContext()
14 glueContext = GlueContext(sc)
15 spark = glueContext.spark_session
16 job = Job(glueContext)
17 job.init(args['JOB_NAME'], args)
18
19 db_name = "glue-catalog-database"
20 tbl_name = "glue-catalog-table"
21 s3_write_bucket = "s3 bucket path"
22
23 # EXTRACT #
24
25 # Elde edilen katalogdan datasource yaratilir
26 datasource0 = glueContext.create_dynamic_frame.from_catalog(
27     database=db_name, table_name=tbl_name)
28
29 # Glue DynamicFrame'i Spark Dataframe'ine cevrilir
30 dataframe = datasource0.toDF()
31
32 # TRANSFORM #
33
34 # son guncelleme sutunu kaldirilir
35 datasource_df = dataframe.drop('last update')
36
37 # Bir satir 4'ten fazla bos deger iceriyorsa kaldirilir
38 corona_df = datasource_df.dropna(thresh=4)
39
40 # Bolge sutundaki eksik deger varsayilan degerle doldurulur
41 cleansed_data_df = corona_df.fillna(
42     value='na_province_state', subset='province/state')
43
44

```

Şekil 4.22. Job script'inde kaynakların tanımlandığı, katalogdan veri kaynağının yaratıldığı, veri kaynağındaki bozuk satır ve sütunların düzeltildiği kod parçası

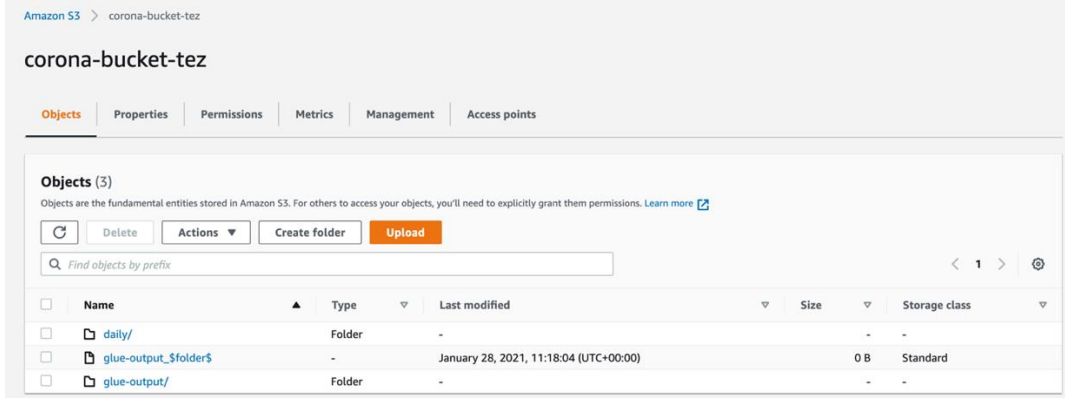
```

44
45 # Kayitlar maksimum vaka sayisi eklenerek bolge ve ulkeye gore gruplanir ve bunlar maksimum onaylanmis vakaya gore siralanir
46
47 most_cases_province_state_df = cleansed_data_df.groupBy('province/state', 'country/region').max('confirmed')\
48     .select('province/state', 'country/region', col("max(confirmed)").alias("Most_Cases"))\
49     .orderBy('max(confirmed)', ascending=False)
50
51 # Kayitlar maksimum vaka sayisi eklenerek bolge ve ulkeye gore gruplanir ve bunlar maksimum onaylanmis vakaya gore siralanir
52 most_deaths_province_state_df = cleansed_data_df.groupBy('province/state', 'country/region').max('deaths')\
53     .select('province/state', 'country/region', col("max(deaths)").alias("Most_Deaths"))\
54     .orderBy('max(deaths)', ascending=False)
55
56
57 # Kayitlar maksimum vaka sayisi eklenerek bolge ve ulkeye gore gruplanir ve bunlar maksimum onaylanmis vakaya gore siralanir
58 most_recoveries_province_state_df = cleansed_data_df.groupBy('province/state', 'country/region').max('recovered')\
59     .select('province/state', 'country/region', col("max(recovered)").alias("Most_Recovered"))\
60     .orderBy('max(recovered)', ascending=False)
61
62
63 # Spark Dataframe'leri Glue DynamicFrame'lere donusturme
64 transform1 = DynamicFrame.fromDF(
65     most_cases_province_state_df, glueContext, 'transform1')
66 transform2 = DynamicFrame.fromDF(
67     most_deaths_province_state_df, glueContext, 'transform2')
68 transform3 = DynamicFrame.fromDF(
69     most_recoveries_province_state_df, glueContext, 'transform3')
70
71
72 # LOAD #
73
74
75 # Veri S3'te belirlenen path'e kaydedilir
76
77 datasink1 = glueContext.write_dynamic_frame.from_options(frame=transform1, connection_type="s3", connection_options={
78     "path": s3_write_bucket+'/most-cases'}, format="parquet", transformation_ctx="datasink1")
79 datasink2 = glueContext.write_dynamic_frame.from_options(frame=transform2, connection_type="s3", connection_options={
80     "path": s3_write_bucket+'/most-deaths'}, format="parquet", transformation_ctx="datasink2")
81 datasink3 = glueContext.write_dynamic_frame.from_options(frame=transform3, connection_type="s3", connection_options={
82     "path": s3_write_bucket+'/most-recoveries'}, format="parquet", transformation_ctx="datasink3")
83
84 job.commit()
85
86

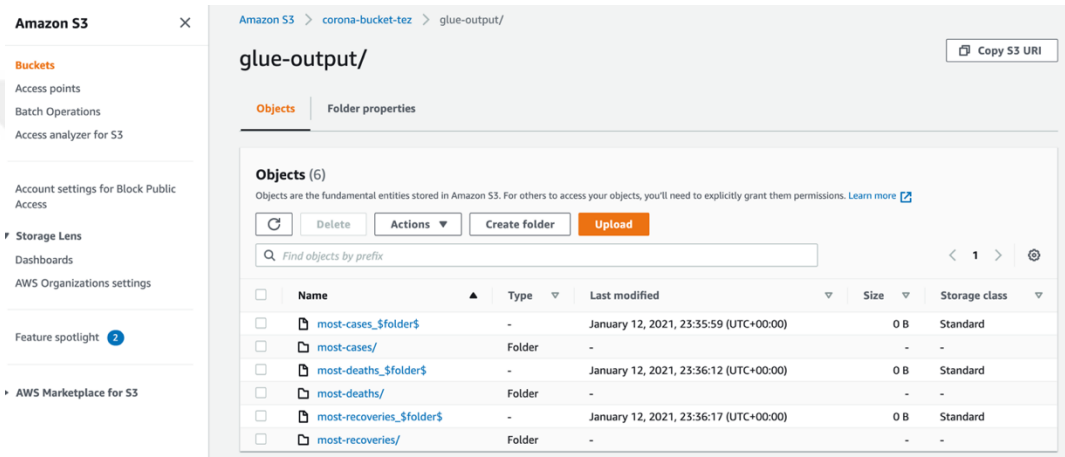
```

Şekil 4.23. Çıktıların belirlenen path'lere kaydedilmesi

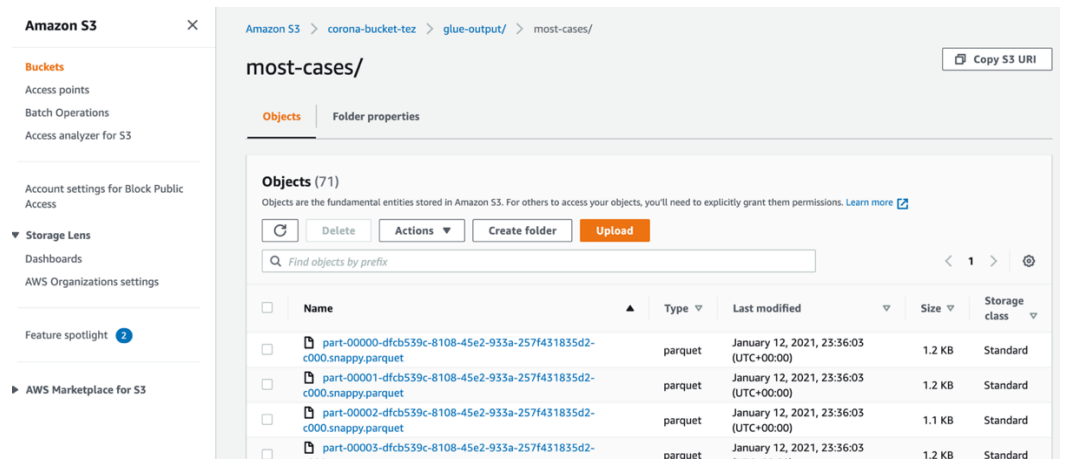
Yazılan script kaydedilip hatasız çalıştığında hedef olarak verilen bucket path'ine sonuçların yazılmış olduğu Şekil 4.24, Şekil 4.25 ve Şekil 4.26'da gösterilmiştir.



Şekil 4.24. Amazon S3'teki bucket'a job'ın çalışmasıyla eklenen dosyalar



Şekil 4.25. Amazon S3'teki bucket'a job'ın çalışmasıyla eklenen dosyanın içeriği

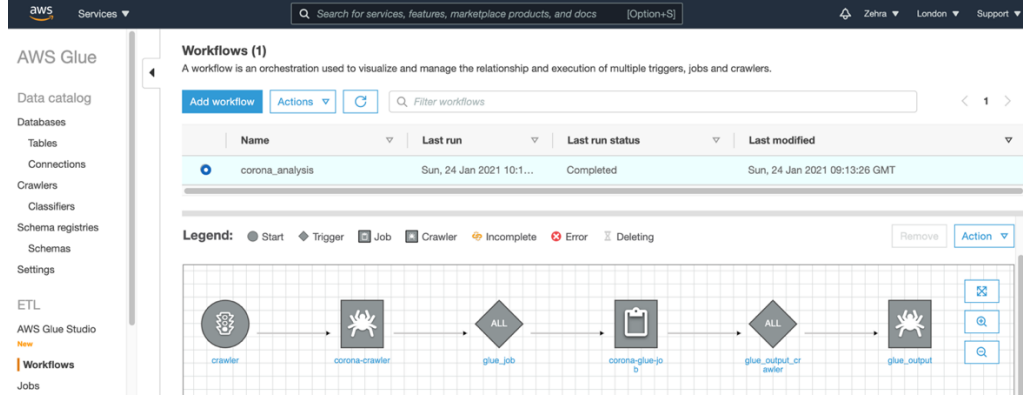


Şekil 4.26. Amazon S3'teki bucket'a job'ın çalışmasıyla parquet tipinde kaydedilen veri dosyaları

4.5. İş Akışının Otomatik Hale Getirilmesi

Workflow tasarlama: Workflow oluşturularak veri kümesinin AWS S3'ten alınıp kataloglanması, job çalıştırılması, job sonucu ile ikinci crawler'ın çalıştırılarak elde

edilen verinin parquet formatında kaydedilmesi otomatik hale getirilmiştir. Tasarlanan workflow Şekil 4.27’deki gibidir.

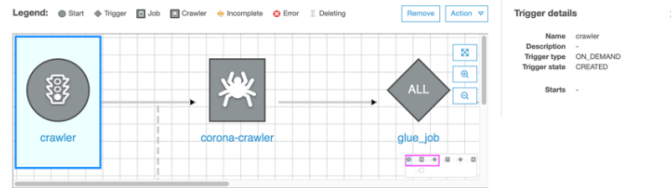


Şekil 4.27. Workflow tasarımı

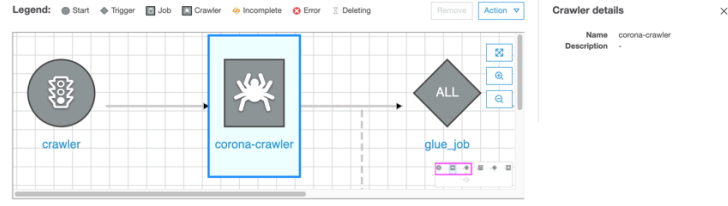
Workflow elemanları şu şekilde sıralanabilir:

- Crawler (trigger): Talep geldiğinde çalışır ve başarıyla çalıştığında corona-crawler’ı tetikler.
- Corona-crawler (crawler): Crawler trigger’ı çalışınca tetiklenir ve başarıyla çalıştığında glue-job’ı tetikler.
- Glue-job (trigger): Corona-crawler crawler’ı çalışınca tetiklenir ve başarıyla çalıştığında corona-glue-job’ı tetikler.
- Corona-glue-job (job): Glue-job trigger’ı çalışınca tetiklenir ve başarıyla çalıştığında glue-output-crawler’ı tetikler.
- Glue-output-crawler (trigger): Corona-glue-job job’ı çalışınca tetiklenir ve başarıyla çalıştığında glue-output’u tetikler.
- Glue-output (crawler): Glue-output-crawler trigger’ı çalışınca tetiklenir.

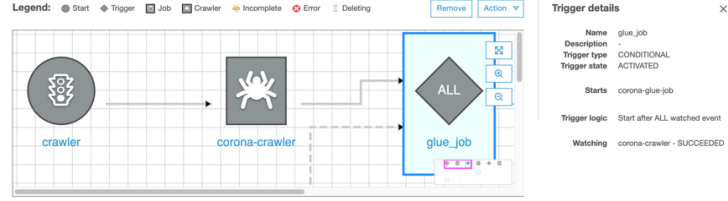
Bu workflow elemanlarının açıklama görselleri Şekil 4.28, Şekil 4.29, Şekil 4.30, Şekil 4.31, Şekil 4.32 ve Şekil 4.33’te verilmiştir.



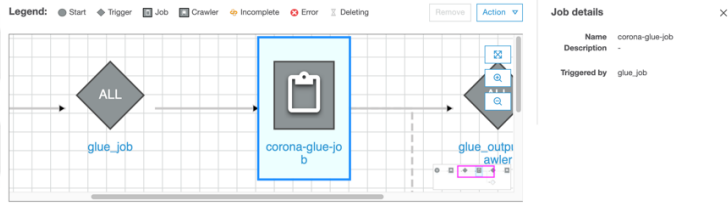
Şekil 4.28. Talep geldiğinde çalışan “crawler” adındaki trigger



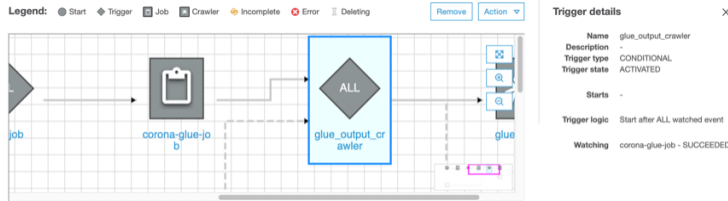
Şekil 4.29. Crawler trigger'ı çalışınca tetiklenen "corona-crawler"



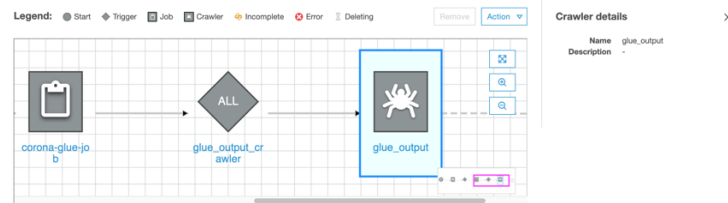
Şekil 4.30. Corona-crawler'ın işinin başarıyla tamamlanması koşuluyla çalışan "glue-job" adlı trigger



Şekil 4.31. Glue-job trigger'ı çalışınca tetiklenen "corona-glue-job"

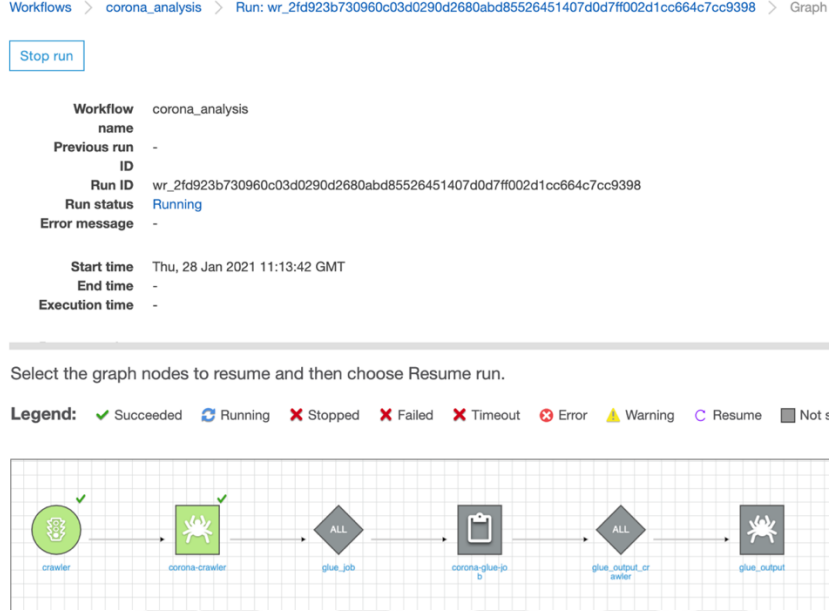


Şekil 4.32. Corona-glue-job'ın işinin başarıyla tamamlanması koşuluyla çalışan "glue-output-crawler" adlı trigger



Şekil 4.33. Glue-output-crawler trigger'ı çalışınca tetiklenen "glue-output" adlı crawler

Workflow çalıştırıldığında, hangi workflow elemanın çalışmasının tamamlandığı Şekil 4.34'te görüldüğü gibi anlık olarak izlenebilir.

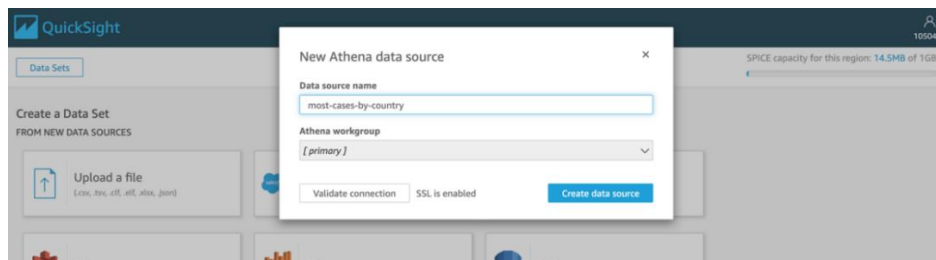


Şekil 4.34. Workflow elemanlarının çalışma sırasındaki görüntüsü

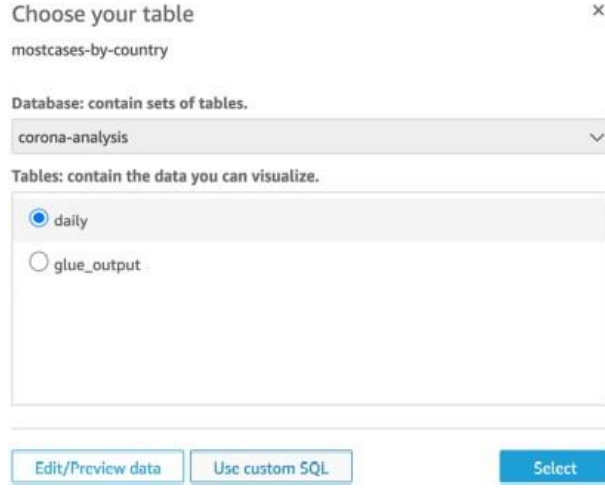
Trigger'larla job ve crawler'ları çalıştırma: Oluşturulan workflow çalıştırıldığında ilk trigger çalışmış ve corona-crawler'ı tetiklemiş, corona-crawler veri kümesi kataloglamıştır. Kataloglama işlemi başarılı şekilde gerçekleştiğinde, ikinci trigger çalışarak corona-glue-job'ı tetiklemiş, bu katalog üzerinde job çalıştırılmıştır. Job'ın başarıyla tamamlanması ile üçüncü ve son trigger çalışıp glue-output crawler'ını çalıştırmış, böylece veri, grafik olarak gösterilebilir hale getirilmiştir.

4.6. Veri Görüntüleme

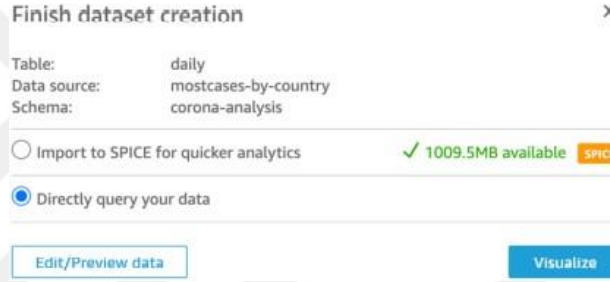
Veriden grafik üretmek için Amazon'un QuickSight hizmeti kullanılmıştır. Öncelikle QuickSight'ta bir veri kaynağı yaratılıp, hedef tablo seçilmiştir. Şekil 4.35, Şekil 4.36 ve Şekil 4.37'de ilk adımların görüntüleri verilmiştir.



Şekil 4.35. QuickSight'ta veri kaynağı yaratılması

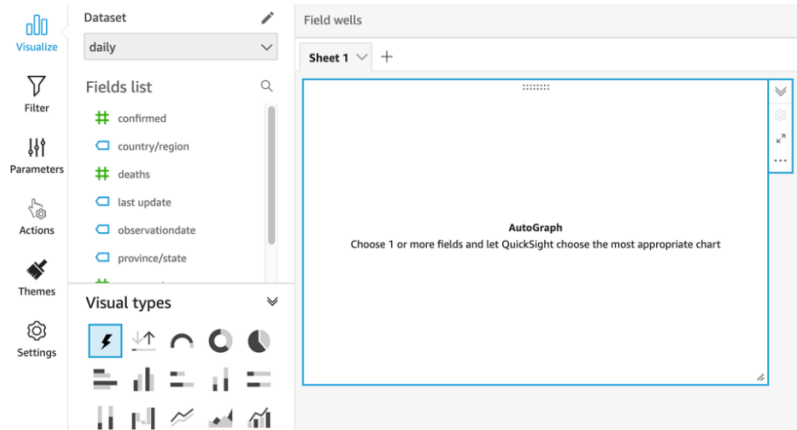


Şekil 4.36. QuickSight'ta yaratılan veri kaynağının ilgili tablosunun seçilmesi

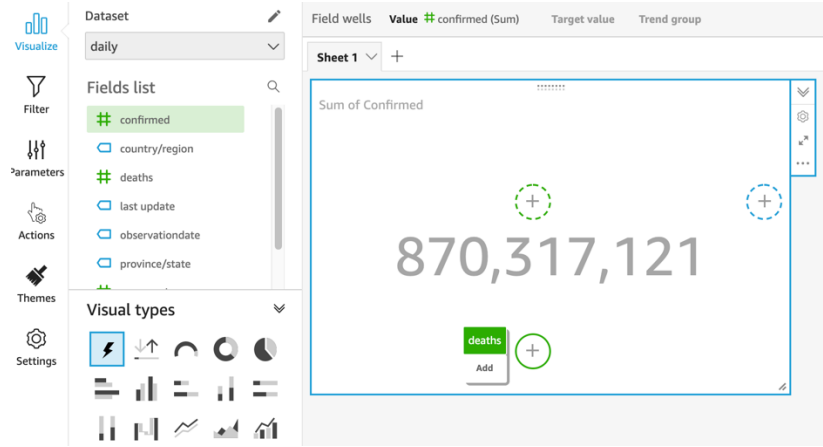


Şekil 4.37. Veri kaynağının sorgulanma biçiminin seçilmesi

Şekil 4.36'da tabloyu seçip devam edilirse, veri kümesi yaratma tamamlandığında Şekil 4.38'deki ekran açılır. Bu ekranın sol tarafında tablodan alınan kolonların bir listesi bulunur. Ortadaki sheet'e bu kolonlar Şekil 4.39'daki gibi sürükle-bırak ile grafikler oluşturulabilir.



Şekil 4.38. Veri kaynağı yaratma tamamlandıktan sonraki QuickSight ekranı



Şekil 4.39. QuickSight'ta sürükle-bırak ile grafik oluşturma

Bunun yerine, Şekil 4.36'da "Use custom SQL" seçilirse, Şekil 4.40'taki SQL Query sayfası açılır ve istenilen sorgulama yazılabilir. Her iki yöntemle de Şekil 4.41'deki grafik oluşturulabilir.

Enter custom SQL query ×

Type in a SQL query.

New custom SQL

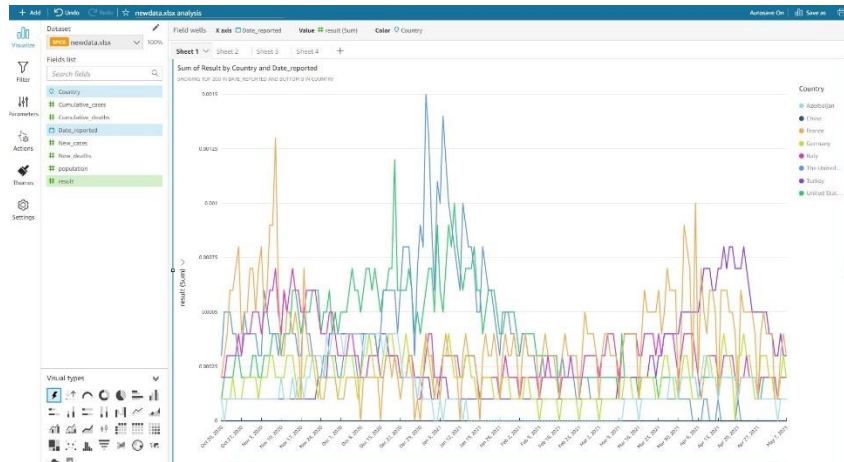
```

1 SELECT "country/region" AS COUNTRY,
2       SUM(most_cases) AS MOST_CASES
3 FROM "corona-analysis".glue_output
4 GROUP BY "country/region"
5 ORDER BY "country/region" ASC

```

Edit/Preview data
Confirm query

Şekil 4.40. SQL Query sayfası

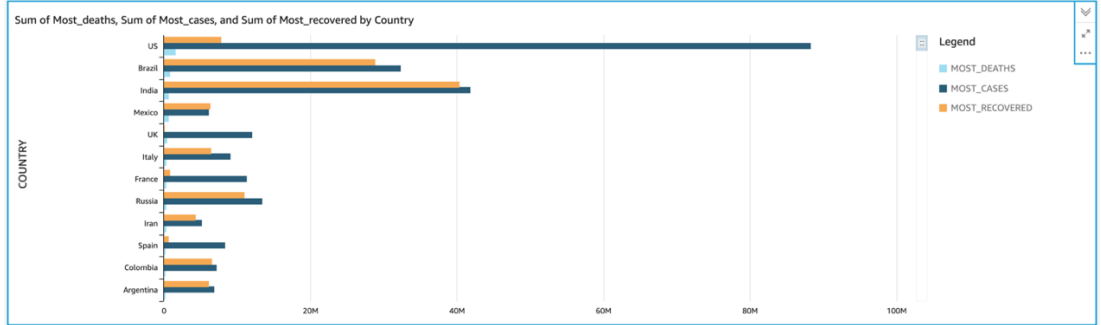


Şekil 4.41. QuickSight'ta oluşturulan bir grafik

Amazon'un QuickSight hizmeti kullanılarak, parquet formatındaki veri grafiklere görselleştirilmiştir. Amazon S3'teki bucket'a korona veri kümesini günlük ekleme zorunluluğu yoktur. Şekil 4.42'deki gibi dokuz günlük veri birer dosya halinde ya da tek bir dosya içinde eklenebilir. Bazı veriler tek tek bazıları toplu halde eklendiğinde de crawler onları bütünleşik hale getirerek tablosunu çıkarır ve yeni eklenen verilerle beraber mevcut kataloğu günceller. Şekil 4.43'te yenilenen verinin grafiklerden biri verilmiştir.

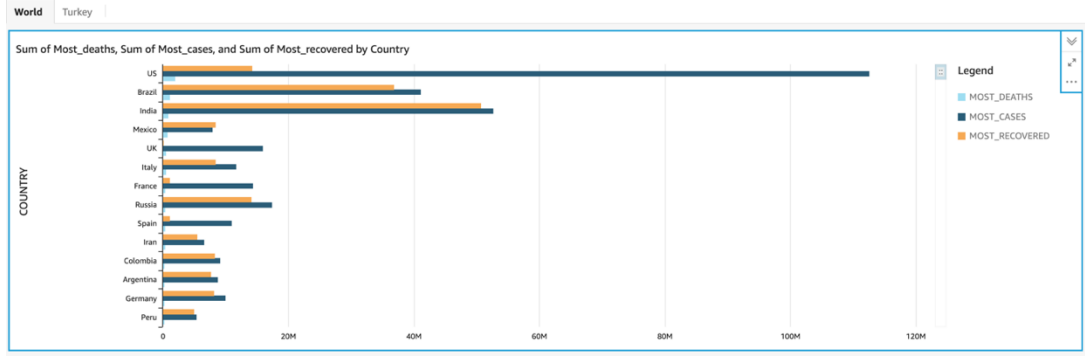
| Name | Type | Last modified | Size | Storage class |
|---|------|--|----------|---------------|
| covid_19_data_2021_01_01.csv | csv | January 24, 2021, 08:35:09 (UTC+00:00) | 49.9 KB | Standard |
| covid_19_data_2021_01_02.csv | csv | January 24, 2021, 09:46:57 (UTC+00:00) | 55.9 KB | Standard |
| covid_19_data_2021_01_03.csv | csv | January 28, 2021, 11:45:48 (UTC+00:00) | 55.9 KB | Standard |
| covid_19_data_2021_01_04.csv | csv | January 28, 2021, 11:45:48 (UTC+00:00) | 56.0 KB | Standard |
| covid_19_data_2021_01_05.csv | csv | January 28, 2021, 11:45:48 (UTC+00:00) | 56.0 KB | Standard |
| covid_19_data_2021_01_06.csv | csv | January 28, 2021, 11:55:05 (UTC+00:00) | 56.0 KB | Standard |
| covid_19_data_2021_01_07.csv | csv | January 28, 2021, 11:55:05 (UTC+00:00) | 56.0 KB | Standard |
| covid_19_data_2021_01_08.csv | csv | January 28, 2021, 11:55:06 (UTC+00:00) | 54.5 KB | Standard |
| covid_19_data_2021_01_09.csv | csv | January 28, 2021, 11:55:06 (UTC+00:00) | 56.0 KB | Standard |
| covid_19_data_2021_01_10.csv | csv | January 28, 2021, 11:55:06 (UTC+00:00) | 56.0 KB | Standard |
| covid_19_data_2021_01_11_2021_01_19.csv | csv | January 29, 2021, 10:45:15 (UTC+00:00) | 501.3 KB | Standard |

Şekil 4.42. Birer dosyada eklenen 9 günlük covid19 veri kümesi

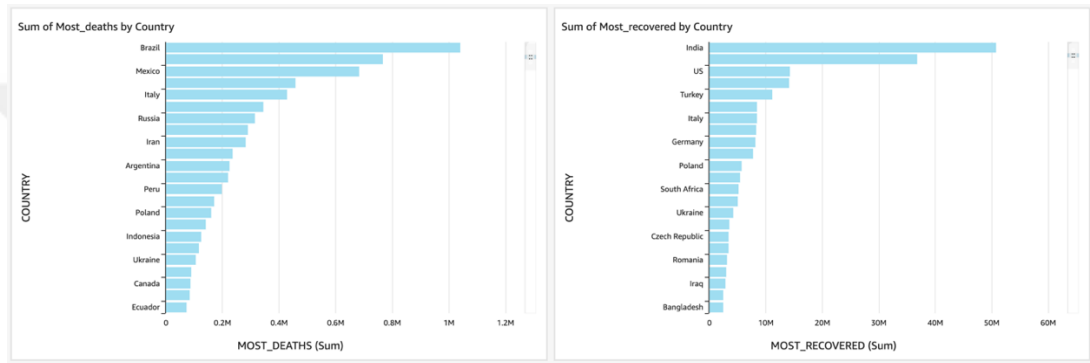


Şekil 4.43. 01.01.2021 – 19.01.2021 tarihinde korona virüs sebebiyle en çok ölüm, vaka ve iyileşme sayılarının ülkelere göre grafiği

2020 Ocak – 2021 Ocak tarihleri arasında kalan döneme ait tüm covid19 veri kümesi tek dosya halinde S3'e eklenmiştir. Crawler yeni verileri öncekilerle birlikte tarayarak mevcut kataloğu güncellemiştir. Şekil 4.44 ve Şekil 4.45'te yenilenmiş covid19 verilerine ilişkin bazı grafikler gösterilmiştir.



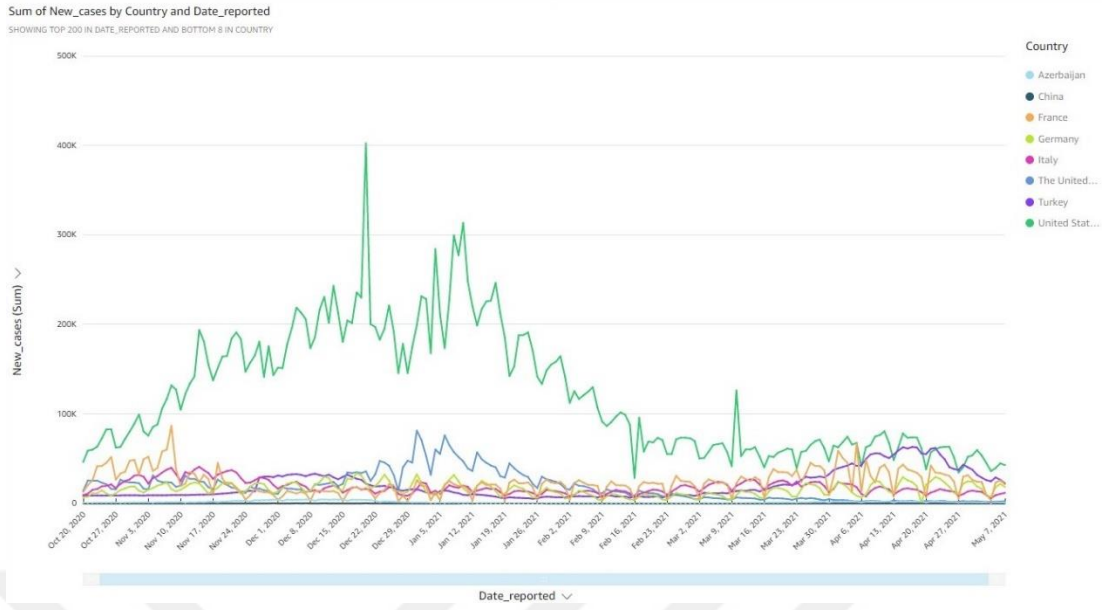
Şekil 4.44. 2020 Ocak – 2021 Ocak tarihleri arasında korona virüs sebebiyle en çok ölüm, vaka ve iyileşme sayılarının ülkelere göre grafiği



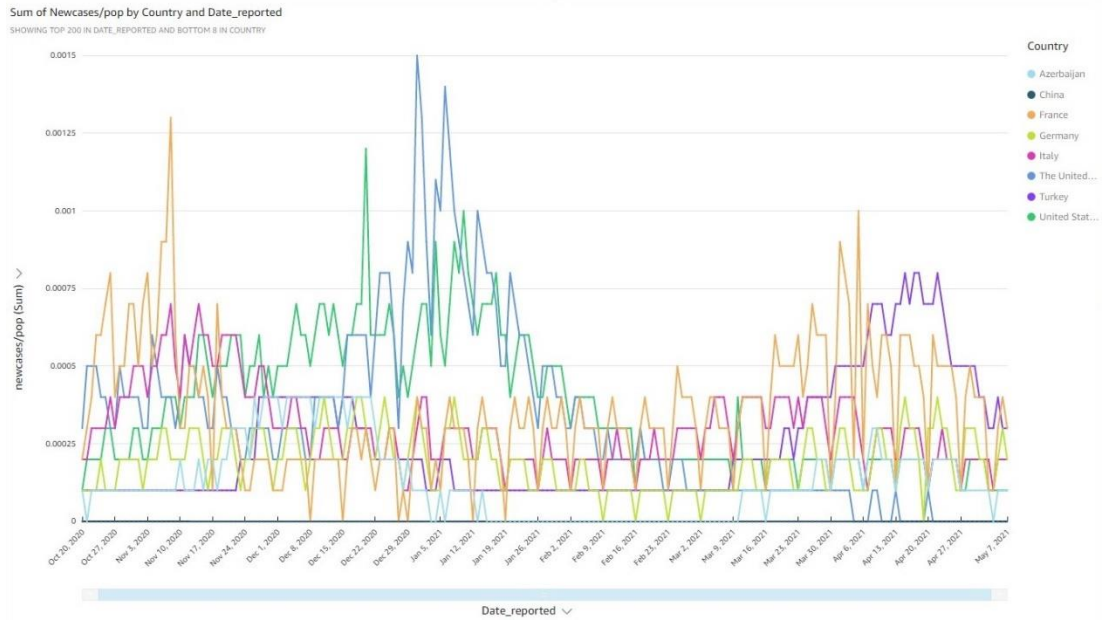
Şekil 4.45. 2020 Ocak – 2021 Ocak tarihleri arasında korona virüs sebebiyle en çok ölüm ve iyileşme sayılarının ülkelere göre grafikleri

2021 yılının ocak ayından 2021 yılının mayıs ayına kadar olan zaman dilimine ait veriler her gün otomatik olarak internet üzerinden AWS Lambda ile çekilerek S3'e yüklenmiştir. S3'e veri eklendikçe bu veri kümesi de önceki verilere katılarak işlenmiştir. Daha kolay okunabilmesi için sekiz ülke seçilerek o ülkelere ait veriler görselleştirilmiştir.

Şekil 4.46'da bu ülkelerdeki günlük yeni vaka sayılarının grafiği verilmiştir. Ancak ülkelerin nüfusları aynı olmadığı için vaka sayıları tek başına yeterince anlam ifade etmeyecektir. Ülkelerin pandemiye karşı mücadelesinde ne durumda olduğuyla ilgili yorum yapabilmek için bu sayılardan daha fazlasına ihtiyaç duyulacaktır. Bu yüzden veri kümesine bu ülkelerin her birine ait nüfus bilgileri tek tek eklenerek Şekil 4.47'de görüldüğü gibi vaka sayılarının ülkelerin nüfuslarına oranları ile oluşturulan yeni bir grafik daha çizilmiştir.

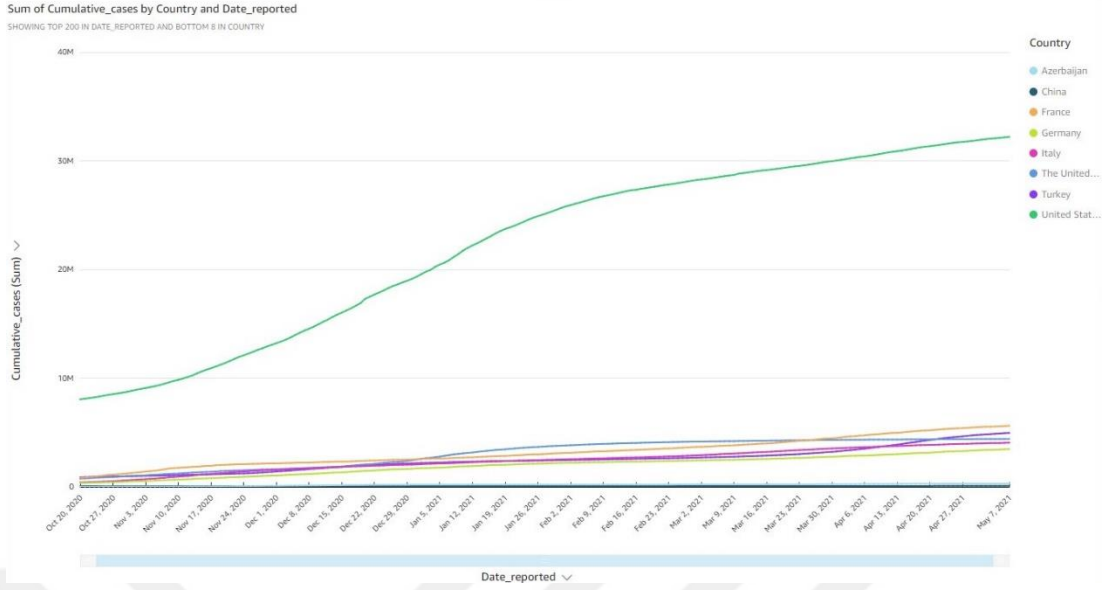


Şekil 4.46. 2020 Ocak – 2021 Mayıs tarihleri arasında sekiz ülkeye ait yeni vaka sayılarının grafiği

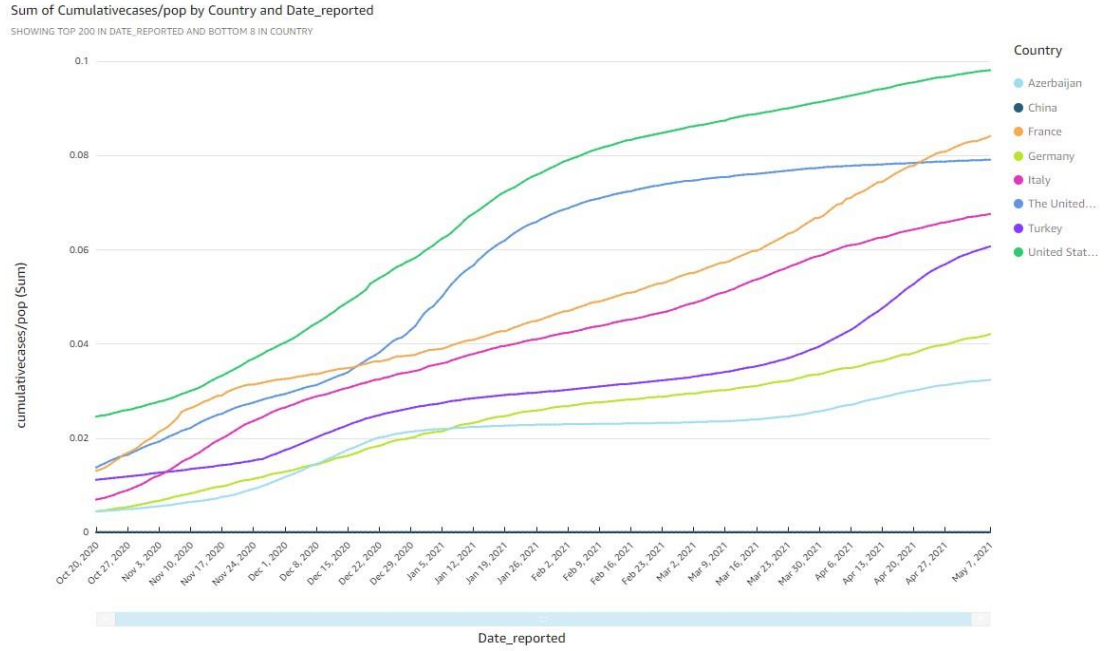


Şekil 4.47. 2020 Ocak – 2021 Mayıs tarihleri arasında sekiz ülkeye ait yeni vaka sayılarının ülkelerin popülasyonuna oranını gösteren grafik

Ülkelerin yeni vaka sayılarını ve bu sayıların nüfuslarına oranlarını gösteren grafikler incelendikten sonra aynı işlemler bir de toplam vaka sayıları ile yapılmıştır. Şekil 4.48’de ülkelerin o tarihe kadar olan toplam vaka sayılarına ait grafik verilmiştir. Şekil 4.49’da ise bu sayıların ülkelerin nüfusuna oranı görülmektedir. Hastalığa tekrar yakalananlar ihmal edilirse, Amerika’da yaşayan insanların 10’da 1’inin hastalığı geçirdiği yorumu yapılabilir.



Şekil 4.48. 2020 Ocak – 2021 Mayıs tarihleri arasında sekiz ülkeye ait vaka sayılarının kümülatif toplamını gösteren grafik



Şekil 4.49. 2020 Ocak – 2021 Mayıs tarihleri arasında sekiz ülkeye ait vaka sayılarının kümülatif toplamının ülkelerin popülasyonuna oranını gösteren grafik

4.7. Makine Öğrenmesi Algoritmasının Uygulanması

Amazon SageMaker, makine öğrenmesi modellerinin hızlıca oluşturulup kullanılabilmesini sağlayan bir servistir [62].

Bu çalışmada zaman serileri makine öğrenmesi yöntemi kullanılmıştır.

Makine öğrenmesi modellerinin eğitilmesi için öncelikle büyük ölçüde verinin yönetilmesi ve verinin eğitilmesi için en uygun algoritmanın seçilmesi gerekmektedir. Eğitim sırasında hesaplamaların yapılabilmesi ve makine öğrenmesi modelinin bir üretim ortamında uygulanması gerekir. Amazon SageMaker bu işlemleri daha sade hale getirmektedir. İçerdiği seçenekler arasından projeye uygun olan algoritmayı seçmek yeterlidir [62].

Makine öğrenmesi algoritmalarını uygulamadan önce, yeni bir veri seti çalışmaya eklenmiştir. Çalışmada kullanılan birinci veri setinde iyileşme sayılarının bulunması, grafikler üretmek için tercih sebebi olmuştur ancak güncel verilerin eklenmemesi nedeniyle kullanımına devam edilmemiştir. Şekil 4.50 ve Şekil 4.51’de ilk ve son satırları gösterilen veri seti ise, WHO (World Health Organization) tarafından sağlanan kaynaklardan elde edilmiştir. İndirilen internet sitesi adresi:

<https://data.humdata.org/dataset/coronavirus-covid-19-cases-and-deaths>

| | A | B | C | D | E | F | G |
|----|---------------|-------------|-----------|------------------|------------|-------------------|---|
| 1 | Date_reported | Country | New_cases | Cumulative_cases | New_deaths | Cumulative_deaths | |
| 2 | 1/3/2020 | Afghanistan | 0 | 0 | 0 | 0 | |
| 3 | 1/4/2020 | Afghanistan | 0 | 0 | 0 | 0 | |
| 4 | 1/5/2020 | Afghanistan | 0 | 0 | 0 | 0 | |
| 5 | 1/6/2020 | Afghanistan | 0 | 0 | 0 | 0 | |
| 6 | 1/7/2020 | Afghanistan | 0 | 0 | 0 | 0 | |
| 7 | 1/8/2020 | Afghanistan | 0 | 0 | 0 | 0 | |
| 8 | 1/9/2020 | Afghanistan | 0 | 0 | 0 | 0 | |
| 9 | 1/10/2020 | Afghanistan | 0 | 0 | 0 | 0 | |
| 10 | 1/11/2020 | Afghanistan | 0 | 0 | 0 | 0 | |

Şekil 4.50. Veri kümesinin kolon isimleri ve ilk satırları

| | | | | | | |
|--------|-----------|----------|----|-------|---|------|
| 116355 | 4/24/2021 | Zimbabwe | 27 | 38045 | 1 | 1556 |
| 116356 | 4/25/2021 | Zimbabwe | 19 | 38064 | 0 | 1556 |
| 116357 | 4/26/2021 | Zimbabwe | 22 | 38086 | 1 | 1557 |
| 116358 | 4/27/2021 | Zimbabwe | 16 | 38102 | 3 | 1560 |
| 116359 | 4/28/2021 | Zimbabwe | 62 | 38164 | 5 | 1565 |
| 116360 | 4/29/2021 | Zimbabwe | 27 | 38191 | 0 | 1565 |
| 116361 | 4/30/2021 | Zimbabwe | 44 | 38235 | 2 | 1567 |
| 116362 | 5/1/2021 | Zimbabwe | 22 | 38257 | 0 | 1567 |
| 116363 | 5/2/2021 | Zimbabwe | 3 | 38260 | 1 | 1568 |
| 116364 | 5/3/2021 | Zimbabwe | 21 | 38281 | 2 | 1570 |
| 116365 | 5/4/2021 | Zimbabwe | 12 | 38293 | 3 | 1573 |
| 116366 | 5/5/2021 | Zimbabwe | 34 | 38327 | 1 | 1574 |
| 116367 | 5/6/2021 | Zimbabwe | 30 | 38357 | 0 | 1574 |
| 116368 | 5/7/2021 | Zimbabwe | 41 | 38398 | 1 | 1575 |
| 116369 | | | | | | |

Şekil 4.51. Veri kümesinin son satırları

Tarih, ülke ismi, yeni vaka sayısı, toplam vaka sayısı, yeni ölüm sayısı ve toplam ölüm sayısı olmak üzere 6 kolondan oluşan veri seti, Ocak 2020’den günümüze kadar her güne ve ülkeye ait veriyi içermektedir. Otomatik olarak çalışan ve her gün güncel

veriyi alıp sahip olduğu veriye ekleyerek işlem yapabilecek bir sistem geliştirilmesi için AWS Lambda üzerinde Şekil 4.52 ve Şekil 4.53'teki fonksiyon oluşturulmuştur ve Şekil 4.54'te gösterildiği gibi her gün çalışmak üzere ayarlanmıştır.

```

1 const converter = require('json-2-csv');
2 const https = require('https');
3 const md5 = require('md5');
4
5 exports.handler = (event, context, callback) => {
6
7   const options = {
8     "method": "GET",
9     "hostname": "worldometers.p.rapidapi.com",
10    "port": null,
11    "path": "/api/coronavirus/all/",
12    "headers": {
13      "x-rapidapi-key": "36f764b412e54811c4ee6f9fc7018ac873n7530f1c840w",
14      "x-rapidapi-host": "worldometers.p.rapidapi.com",
15      "useQueryString": true
16    }
17  };
18
19  const req = https.request(options, function (res) {
20    const chunks = [];
21
22    res.on("data", function (chunk) {
23      chunks.push(chunk);
24    });
25
26    res.on("end", function () {
27      const body = Buffer.concat(chunks);
28      //console.log(body.toString());
29      var objectValue = JSON.parse(body.toString());
30      var last_update = objectValue["last_update"];
31      var data = objectValue["data"];
32
33      var filename = last_update.slice(0, 10);
34      last_update = last_update[5] + last_update[6] + '/' + last_update[8] + last_update[9] + '/' + last_update[0] + last_update[1] + last_update[2] + last_update[3]
35      console.log('deneme' + last_update)
36      data.forEach(function(entry) {
37

```

Şekil 4.52. Lambda fonksiyonunun birinci kısmı

```

36
37
38  data.forEach(function(entry) {
39    entry["New Cases"] = entry["New Cases"].replace(new RegExp(',', 'g'), '').replace('+', '');
40    entry["Total Cases"] = entry["Total Cases"].replace(new RegExp(',', 'g'), '');
41    entry["New Deaths"] = entry["New Deaths"].replace(new RegExp(',', 'g'), '').replace('+', '');
42    entry["Total Deaths"] = entry["Total Deaths"].replace(new RegExp(',', 'g'), '');
43  });
44  if (data[0]) {
45    data.splice(0, 1);
46  }
47
48  var options = {
49    keys: [{ field: 'last_update', title: 'Date_reported' }, { field: 'Country', title: 'Country' }, { field: 'New Cases', title: 'New_cases' }, { field: 'Total Cases', title: 'Cumulative_cases' }, { field: 'New Deaths', title: 'New_Deaths' }, { field: 'Total Deaths', title: 'Cumulative_Deaths' }],
50    excludeKeys: ['Critical', 'Population']
51  };
52
53  converter.json2csv(data, options, (err, csv) => {
54    if (err) {
55      throw err;
56    }
57
58    // print CSV string
59    //console.log(csv);
60
61    var s3 = new AWS.S3();
62    var params = {
63      Bucket: 'corona-bucket-ter',
64      Key: 'data/' + filename + '.csv',
65      Body: csv
66    };
67    s3.putObject(params, function (err, data) {
68      if (err) console.log(err, err.stack); // an error occurred
69      else console.log(data); // successful response
70    });
71  });

```

Şekil 4.53. Lambda fonksiyonunun ikinci kısmı

The screenshot shows the 'Add trigger' configuration page in the AWS Lambda console. The 'Trigger configuration' section is expanded, showing the following settings:

- Provider:** EventBridge (CloudWatch Events)
- Rule:** Create a new rule
- Rule name:** dailyUpdate
- Rule description:** (empty field)
- Rule type:** Schedule expression
- Schedule expression:** rate(1 day)
- Enable trigger:**

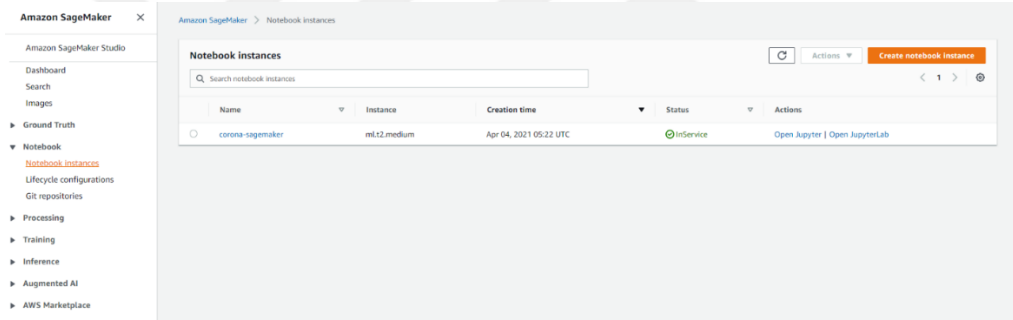
Şekil 4.54. Lambda fonksiyonunu zamanlayıcı ile çalıştırma

Güncel verinin elde edilmesi için kullanılan api worldometer internet sitesine aittir ve adresi şudur:

https://rapidapi.com/ronanren/api/worldometers?endpoint=apiendpoint_999cd491-a3b0-4765-abe0-af3670f76e6f

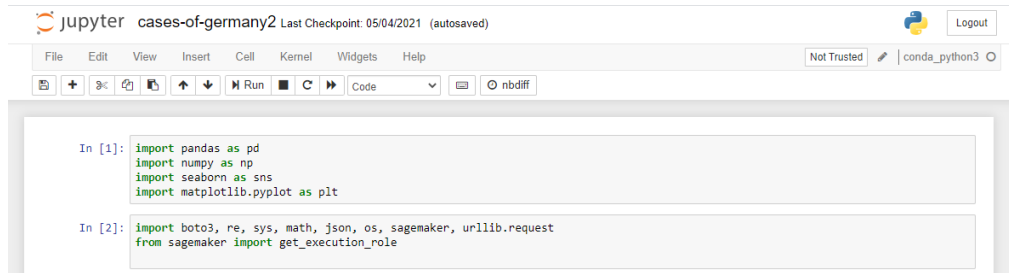
Veri seti üzerinde çalıştırılan crawler'ın ürettiği meta veriye göre yeni bir job daha çalıştırılarak ülkelere ait veriler ayrı birer dosyaya aktarılmıştır. Elde edilen bu veriler üzerinde makine öğrenimi gerçekleştirilmiştir.

1. Adım: Bir Amazon SageMaker örneği ve ardından not defteri örneği oluşturulmuştur ve Şekil 4.55'te gösterilmiştir. Ayrıca Amazon SageMaker'ın S3'teki verilere erişebilmesi için yeni bir IAM rolü oluşturularak SageMaker örneğine tanımlanmıştır.



Şekil 4.55. SageMaker'daki oluşturulan not defteri

2. Adım: Jupyter'i açıp conda_python3 tipinde yeni bir kod sayfası oluşturulmuştur. Şekil 4.56'daki gibi, gerekli kitaplıkların içe aktarılması, verinin hazırlanması, makine modelinin eğitilmesi ve uygulanması için ihtiyaç duyulacak tüm ortam değişkenleri tanımlanmıştır.



Şekil 4.56. Kütüphanelerin tanımlanması

Depolanacak veri için daha önce oluşturulan S3 paketi kullanılmıştır.

3. Adım: Şekil 4.57’de gösterildiği gibi veri SageMaker’a yüklenerek bir veri çerçevesine eklenmiştir.

```
In [29]: try:
urlib.request.urlretrieve ("https://corona-bucket-tez.s3.eu-west-2.amazonaws.com/total-output/turkey-cases/part-00000-422ab5c
print('Success: downloaded cvd19tr.csv.')
except Exception as e:
print('Data load error: ',e)

try:
model_data = pd.read_parquet('./part-00000-422ab5ce-275a-40cb-9607-8049c487b1ed-c000.snappy.parquet')
print('Success: Data loaded into dataframe.')
except Exception as e:
print('Data load error: ',e)
```

Şekil 4.57. Verinin veri çerçevesine eklenmesi

Şekil 4.58’de görüleceği üzere, verinin doğru şekilde alındığını test etmek için veri görüntülenmiştir.

```
In [3]: print(model_data)

   Date_reported  Country  New_cases  Cumulative_cases  New_deaths  \
0      1/3/2020  Afghanistan      0           0           0
1      1/4/2020  Afghanistan      0           0           0
2      1/5/2020  Afghanistan      0           0           0
3      1/6/2020  Afghanistan      0           0           0
4      1/7/2020  Afghanistan      0           0           0
...
116362  5/3/2021  Zimbabwe      21          38281           2
116363  5/4/2021  Zimbabwe      12          38293           3
116364  5/5/2021  Zimbabwe      34          38327           1
116365  5/6/2021  Zimbabwe      30          38357           0
116366  5/7/2021  Zimbabwe      41          38398           1

   Cumulative_deaths
0           0
1           0
2           0
3           0
4           0
...
116362          1570
116363          1573
116364          1574
116365          1574
116366          1575

[116367 rows x 6 columns]
```

Şekil 4.58. Verinin görüntülenmesi

4. Adım: Verinin tarih özelliğinin formatı makine öğrenmesinin işlemesine uygun olmadığı için format değiştirilmiştir. Şekil 4.59’da gösterilmiştir.

```
In [5]: model_data['Date_reported'] = pd.to_datetime(model_data['Date_reported'], format = '%m/%d/%Y')

In [6]: print(model_data)

   Date_reported  Country  New_cases  Cumulative_cases  New_deaths  \
0      2020-01-03  Afghanistan      0           0           0
1      2020-01-04  Afghanistan      0           0           0
2      2020-01-05  Afghanistan      0           0           0
3      2020-01-06  Afghanistan      0           0           0
4      2020-01-07  Afghanistan      0           0           0
...
116362  2021-05-03  Zimbabwe      21          38281           2
116363  2021-05-04  Zimbabwe      12          38293           3
116364  2021-05-05  Zimbabwe      34          38327           1
116365  2021-05-06  Zimbabwe      30          38357           0
116366  2021-05-07  Zimbabwe      41          38398           1

   Cumulative_deaths
0           0
1           0
2           0
3           0
4           0
...
116362          1570
116363          1573
116364          1574
116365          1574
116366          1575

[116367 rows x 6 columns]
```

Şekil 4.59. Tarih özelliğinin formatının değiştirilmesi

5. Adım: Makine öğrenmesinin veriyi işleyebilmesi için verinin tüm özelliklerinin sayısal olması gerektiği Genel Bilgiler Bölümü'nde belirtilmişti. Şekil 4.60'ta görüldüğü gibi verinin sayısal yapıda olmayan özellikleri sayısal değerlere çevrilmiştir.

```
In [7]: from sklearn.preprocessing import OrdinalEncoder
ord_enc = OrdinalEncoder()
model_data["country-name"] = ord_enc.fit_transform(model_data[["Country"]])
model_data["date"] = ord_enc.fit_transform(model_data[["Date_reported"]])
model_data = model_data.drop(["Country", "Date_reported"], axis=1)

In [8]: print(model_data)

   New_cases  Cumulative_cases  New_deaths  Cumulative_deaths \
0           0                0           0                0
1           0                0           0                0
2           0                0           0                0
3           0                0           0                0
4           0                0           0                0
...         ...                ...         ...                ...
116362      21                38281         2                1570
116363      12                38293         3                1573
116364      34                38327         1                1574
116365      30                38357         0                1574
116366      41                38398         1                1575

   country-name  date
0           0.0  0.0
1           0.0  1.0
2           0.0  2.0
3           0.0  3.0
4           0.0  4.0
...         ...  ...
116362      235.0  486.0
116363      235.0  487.0
116364      235.0  488.0
116365      235.0  489.0
116366      235.0  490.0

[116367 rows x 6 columns]
```

Şekil 4.60. Özelliklerin sayısal değerlere çevrilmesi

6. Adım: Makine öğrenmesi çalıştırılmadan önce verinin sıralaması karıştırılarak eğitim ve test için ayrılır. Bu işlemin yapıldığı kod Şekil 4.61'dedir.

```
In [26]: X = model_data.drop(["New_cases"], axis = 1)
y = model_data["New_cases"]
#Convert to numpy arrays
X = np.array(X).astype('float32')
y = np.array(y).astype('float32')

from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X,y, test_size=0.25)
```

Şekil 4.61. Verinin makine öğrenmesi için eğitim ve test verisine ayrılması

7. Adım: IAM rolle ilgili işlem yapılmış ve kaydedilecek sonuç için alt dosya oluşturulmuştur. Şekil 4.62'de gösterilmiştir.

```
In [51]: sagemaker_session = sagemaker.Session()
bucket = "corona-bucket-tez"
prefix = "linear-learner" #prefix S3 bucket'ta bir alt dosya

role = sagemaker.get_execution_role()
print(role)

arn:aws:iam::105049160776:role/service-role/AmazonSageMaker-ExecutionRole-20210404T062242
```

Şekil 4.62. IAM rol işlemi ve hedef klasör oluşturma

8. Adım: Şekil 4.63'te kod ile lineer öğrenicinin veriyi anlayabilmesi için veri kümesi RecordIO formatına dönüştürülmüştür [70].

```
In [52]: #datasetin RecordIO formatına donusturulmesi gerekir
import io
import sagemaker.amazon.common as smac
import os

buf = io.BytesIO()
smac.write_numpy_to_dense_tensor(buf, X_train, y_train)
buf.seek(0)

#Egitim datasi yukleme
#S3'e yuklenecek egitim datasi icin dosya
key = 'linear-train-data'
#Upload training data to S3
boto3.resource('s3').Bucket(bucket).Object(os.path.join(prefix, 'train', key)).upload_fileobj(buf)
s3_train_data = 's3://{}/train/{}'.format(bucket, prefix, key)
print('uploaded training data location: {}'.format(s3_train_data))

##Test datasi yukleme
buf = io.BytesIO()
smac.write_numpy_to_dense_tensor(buf, X_test, y_test)
buf.seek(0)

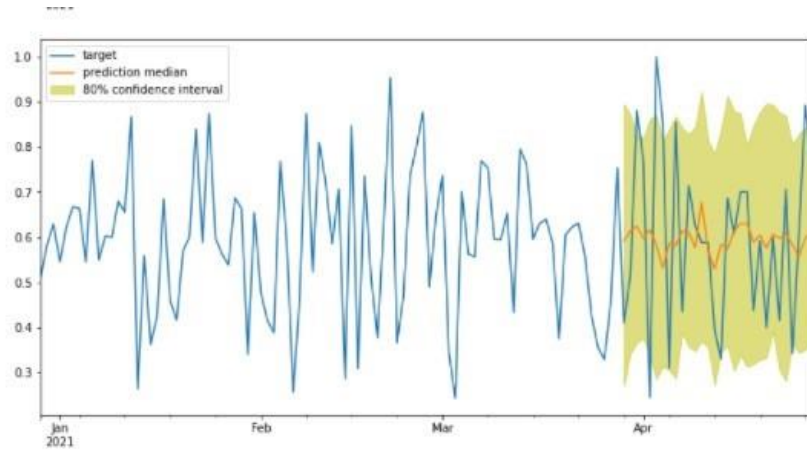
#test data icin alt dosya
key = 'linear-test-data'
boto3.resource('s3').Bucket(bucket).Object(os.path.join(prefix, 'test', key)).upload_fileobj(buf)
s3_test_data = 's3://{}/test/{}'.format(bucket, prefix, key)
print('uploaded training data location: {}'.format(s3_test_data))

output_location = 's3://{}/output'.format(bucket, prefix)
print('Training artifacts will be uploaded to: {}'.format(output_location))

uploaded training data location: s3://corona-bucket-tez/linear-learner/train/linear-train-data
uploaded training data location: s3://corona-bucket-tez/linear-learner/test/linear-test-data
Training artifacts will be uploaded to: s3://corona-bucket-tez/linear-learner/output
```

Şekil 4.63. Veri kümesinin RecordIO formatına dönüştürülmesi

9. Adım: Hazırlanan veri kümesi üzerinde zaman serisi algoritmaları uygulanmış ancak başarılı sonuç alınamamıştır. Daha sonra veri kümesi, içerdiği dalgalanmalardan arındırılmak üzere bir dizi işlemden geçirildikten sonra zaman serisi algoritmaları tekrar uygulanmış ve Şekil 4.64'teki gibi sonuç alınmıştır.



Şekil 4.64. Zaman serisi algoritmasının sonucu

10. Adım: Üretilen tahminler ile gerçek veri örtüşmediği için bir hazırlanan veri kümesi Şekil 4.65'te görüldüğü gibi Amazon Forecast'e yüklenerek eğitim başlatılmıştır.

Dataset import details

Dataset import name
The name can help you distinguish this dataset import from other imports on your dataset detail page.

coviddata

The dataset import name must have 1 to 63 characters. Valid characters: a-z, A-Z, 0-9, and _.

Select time zone Info
Select a time zone for your dataset.

Do not use time zone

Data location info
The location is the path to the file in your S3 bucket that contains your data.

s3://covrona-bucket-tez/data.csv

Your files must be in CSV format.

IAM role info
Dataset groups require permissions from IAM to read your dataset files in S3. Choose or create a role using this control.

Enter a custom IAM role ARN

Custom IAM role ARN
arn:aws:iam::YourAccountID:role/YourRole

Şekil 4.65. Modelin eğitilmesi

Tahmin yöntemi seçimi Amazon Forecast'in otomatik seçeneği yapmıştır. Şekil 4.66'da seçilebilecek tüm yöntemler gösterilmektedir.

Predictor details

Algorithm selection info
An algorithm is used to train your predictor.

Automatic (AutoML)
Let Amazon Forecast choose the right algorithm for your dataset.

Manual
Explore the algorithms and choose one.

Algorithm
The algorithm that you want Amazon Forecast to use to train your predictor.

CNN-QR
An Amazon proprietary convolutional neural network algorithm. Useful for 100+ time series and flexible with all forms of related time series and item metadata.

DeepAR+
An Amazon proprietary recurrent neural network algorithm. Useful for 100+ time series with forward-looking related time series and item metadata.

Prophet
A Bayesian structural time series model. Useful for time series with strong seasonal effects and several seasons of historical data.

NPTS
Non-Parametric Time Series. An Amazon proprietary algorithm useful for sparse and intermittent time series.

ARIMA
Autoregressive Integrated Moving Average. A statistical algorithm useful for datasets with fewer time series.

ETS
Exponential Smoothing. A statistical algorithm useful for datasets with fewer time series and seasonality.

Forecast dimensions - optional
Item id is used in training by default. Select additional keys you would like to use to generate a forecast. These keys are fields in your dataset.

Choose a forecast dimension

Number of backtest windows - optional info
This is the number of times that the algorithm splits the input data for use in training and evaluation.

1

Backtest window offset - optional info
This is the point in the dataset where you want to split the data for model training and evaluation.

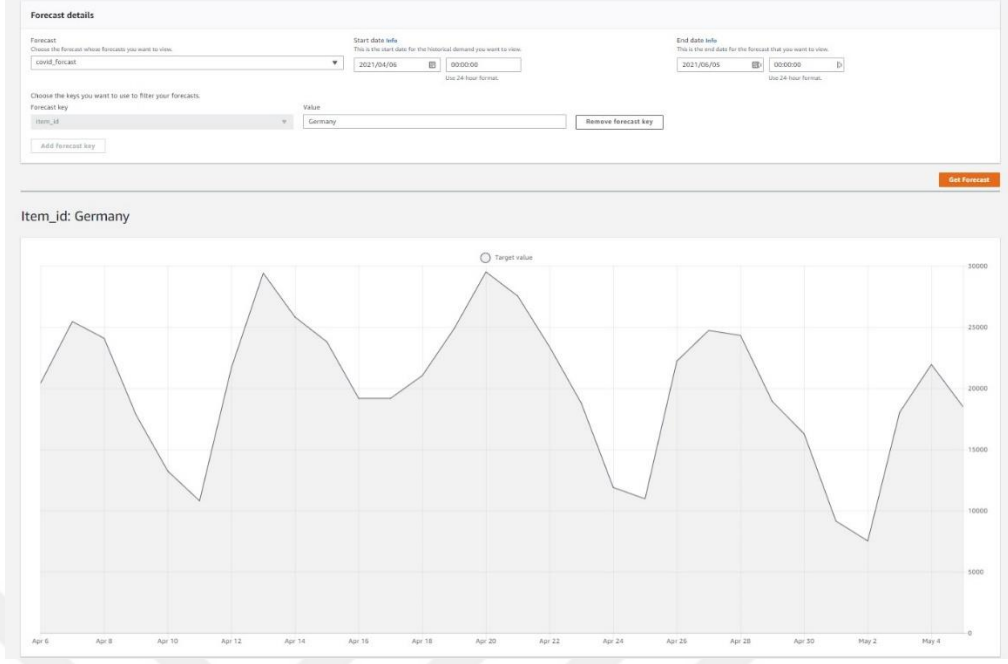
10

Forecast type - optional info
Specify the quantiles used to create forecasts and evaluate predictors. Choose up to 5 quantiles between 0.01 and 0.99 (by increments of 0.01). You can also include the mean forecast with 'mean'.

| Forecast types | Value | |
|-----------------|-------|--------|
| Forecast type 1 | 0.10 | Remove |
| Forecast type 2 | 0.50 | Remove |
| Forecast type 3 | 0.90 | Remove |

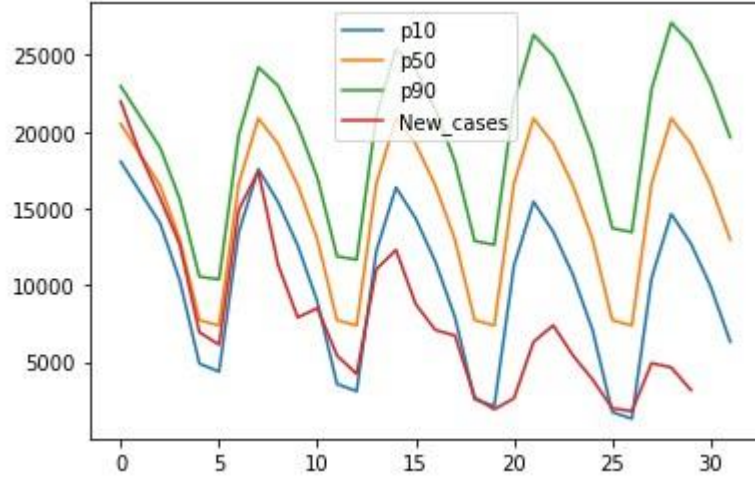
Şekil 4.66. Forecast yöntemleri

11. Adım: Tahmin motorunun çalışması bittiğinde tahmin edilmek istenen tarih aralığı Şekil 4.67'deki gibi seçilerek sonuç grafiği gözlemlenmiştir. Makine öğrenmesi algoritmalarında genelde eğitim verisi toplam verinin %70'i, test verisi %30'u olarak ayrılır. Zaman serilerinde ise bu aralık seçilirken hafta, ay ya da mevsim süresi seçilmesi tercih edilir. Bu örnekte bir aylık zaman dilimi seçilmiştir.



Şekil 4.67. Belirtilen tarih aralığına ait tahmin

12. Adım: Makine öğrenmesi sonucunda üretilen tahminler ile gerçek veriler Şekil 4.68’de aynı grafikte gösterilmiştir.



Şekil 4.68. Tahminler ve gerçek veri grafiği

30 günlük zaman diliminin ilk yarısında gerçek sayılardan yüksek olsa dahi gerçeğe daha yakın tahminler üretildiği gözlenmiştir. Sonraki yarıda ise vaka sayılarının artması beklenirken düşüş yaşandığı görülmektedir. Bunun nedeni, o tarihte aşılmalara hızlanması ve aşılana kişi sayısının artmasına bağlı olarak vaka sayılarının azalması olmuştur. Tahminlerin yakınlık değeri Şekil 4.69’daki gibi hesaplanmıştır.


```
Out[225]: 0.6062465879815337
```

Şekil 4.69. Tahminlerin yakınlık değeri

4.8. Makine Öğrenmesi ile Üretilen Tahminlerin Değerlendirilmesi

Üzerinde çalışılan veri kümesi, ülkelerin her güne ait yeni vaka, toplam vaka, yeni ölüm ve toplam ölüm sayılarını içermektedir. Bu veri üzerinde çalıştırılan makine öğrenmesi algoritmalarının ürettiği tahminlerin doğruluk değeri yaklaşık olarak 0.6 bulunmuştur. Daha yüksek doğruluk değerleri elde etmek için, ülkelerin pandemiyle mücadele için yaptıkları kapanma tarihleri, okul ve iş yerlerine gidilme/gidilmeme durumları, seyahat yasakları, aşılama hızları gibi bilgilerin olduğu bir veri kümesi temin etmek etkili olacaktır.

5. SONUÇLAR VE ÖNERİLER

Bu çalışmada Bulut Bilişim ve Makine Öğrenmesi ile ilgili araştırma yapılmış ve iki teknolojinin kullanıldığı, otomatik olarak internet üzerinden elde ettiği güncel veri ile beslenen ve bu veri üzerinde makine öğrenmesi gerçekleştirip sonuçlarını sunan bir sistem geliştirilmiştir. Korona virüse ilişkin tüm ülkelerin bilgilerini içeren veri kümeleri üzerinde çalışılmıştır. Bu veri kümeleri, bulut bilişim sağlayıcılarından Amazon Web Servis'in hizmetleri kullanılarak depolanmış ve ETL işlemleri (ayıklama, dönüştürme ve yükleme) yapılmıştır. Tüm ülkelerin günlük vaka, ölüm ve iyileşme sayılarını içeren bu veriden, "en çok vaka görülen ülkeler", "vaka sayılarının nüfusa oranları" gibi veriler çıkarılarak grafiklerle gösterilmiştir. Daha sonra ayıklanmış ve isteğe göre biçimlendirilmiş bu veriler üzerinde, zaman serileri algoritmaları çalıştırılarak, vaka sayılarına dair tahminler üretilmiştir.

Üretilen tahminlerin doğruluk oranları beklenenin altında çıkmıştır. Daha sonra makine öğrenmesi işlemi baştan yürütülerek, verinin eğitim verisi ve test verisi olarak ikiye ayrıldığı adımda bu ayrımın rasgele yapılması yerine tarihe göre yapılması denenmiştir. Yani Eğitim verisi olarak pandeminin yaklaşık olarak ilk 1 yıllık verisi, test verisi olarak da sonraki 1 aylık verisi etiketlenmiştir. Bu şartlar altında öğrenme gerçekleştirildiğinde tahminlerin doğruluk oranı artmıştır.

Çalışmanın Genel Bilgiler bölümünde de bahsedildiği gibi makine öğrenmesinin verimli çalışabilmesi ve doğruluk değeri yüksek tahminler üretebilmesi için, üzerinde çalıştığı verinin özellikleri çok önemlidir. Çalışmada kullanılan veride ülke ismi, tarih, yeni vaka, toplam vaka, yeni ölüm ve toplam ölüm bilgileri yer alması makine öğrenmesinin verimli çalışması için yeterli olmamıştır. Her bölgeye ait o tarihte okulların açık olup olmaması, günlük yaşam kısıtlamaları, seyahat yasakları, aşılama hızı, işletmelerin çalışır durumda olması gibi bilgilere sahip olunması durumunda makine öğrenmesi algoritmasının daha doğru tahminler üretilbileceği öngörülmektedir. Çalışmada 1 aylık veriye dair üretilen tahminlerin ilk 15 günü gerçeğe çok yakın bulunmuş, diğer 15 günde ise tahmin edilenden daha düşük vaka

sayısı gözlenmiştir. Bu gözlem süreci, aşılmanın hızlandığı ve aşılana insan sayısının hızla arttığı döneme denk gelmektedir.

Büyük veriyi depolamak ve üzerinde işlemler yapmak için Amazon Web Servis'in hizmetleri yeterli ve performanslı bulunmuştur. Diğer Bulut Bilişim sağlayıcılarına göre daha kapsamlı hizmetleri olduğu ve veriyle yapılmak istenebilecek birçok işlemin yapılabilmesine olanak sağladığı görülmüştür. Fiyatlandırmasının da makul olduğu göz önünde bulundurularak, akademik ve bireysel çalışmalarda bu hizmetlerden faydalanılması önerilir.

Bu çalışmadaki veri kümesinin özelliklerinin yetersiz olması nedeniyle üretilen tahminlerde yüksek başarı elde edilememiş olsa da geliştirilmesi amaçlanan, otomatik olarak güncel veriyi temin eden, depolayan ve veriyi işleyen sistem başarıyla tamamlanmıştır. Bu sistem, üzerinde birkaç basit değişiklik yapılarak belirlenen herhangi bir alanda kullanılmak üzere düzenlenebilir. İnsan müdahalesi gerektirmeyen bu sistemin kurulumu sırasında dikkat edilmesi gereken en önemli noktalar şöyle sıralanabilir:

- Başlangıçta kullanılmak üzere uygun veri kümesinin bulunması
- Güncel verinin sağlanacağı api'nin bulunması
- İki verinin birbiriyle tutarlı olması
- Maliyetin düzenli olarak gözlenmesi

İleriki çalışmalarda, bu çalışmada oluşturulan sisteme ek olarak birden fazla veri kaynağının sistemi beslemesi, makine öğrenmesi algoritmalarından çıkan sonuçların dosyalanması, belirlenen durumlarda sistemin uyarı göndermesi, arttırılması/azaltılması istenen özelliğe göre sistemin öneri üretmesi gibi özellikler geliştirilmesi hedeflenmektedir. Böylece, daha kapsamlı ve daha çok alanda fayda sağlanabilecek bir sistem elde edilebilir.

KAYNAKLAR

- [1] Duru N., Eker A. G., Kat O., Ildıran A., Makine Öğrenmesi ile Film Başarı Tahmini, *2018 3rd International Conference on Computer Science and Engineering (UBMK)*, Sarajevo, Bosnia and Herzegovina, 20-23 September 2018.
- [2] Kökcü B. N., Köse R. D., Bulut F., Amasyalı M. F., Kolektif Öğrenme Algoritmalarıyla Çocuklarda Obezite Hastalığına Yakalanma Olasılıklarının Hesaplanması, *Akıllı Sistemlerde Yenilikler ve Uygulamaları Sempozyumu*, İzmir, Türkiye, 9 Ekim 2014.
- [3] Bayer H., Çoban T., Web İstatistiklerinde Makine Öğrenmesi Algoritmaları ile Kritik Parametre Tespiti, *Electronic Journal of Vocational Colleges*, 2015, 5(4), 23-41.
- [4] Okal G., Meme Kanseri Riskinin Termal Görüntüleme ve Makine Öğrenmesi ile Saptanması, Yüksek Lisans Tezi, Pamukkale Üniversitesi, Fen Bilimleri Enstitüsü, Denizli, 2019, 656094.
- [5] Rar P. R., A Survey Of Iot Cloud Platforms, *Future Computing And Informatics Journal*, 2017, 1(1-2), 35-46.
- [6] Internet of Things (IoT) connected devices installed base worldwide from 2015 to 2025, <https://www.statista.com/statistics/471264/iot-number-of-connected-devices-worldwide/>, (Ziyaret tarihi: 10 Şubat 2021).
- [7] Veri Gölü ve Veri Ambarı Arasındaki Fark Nedir?, https://www.linkedin.com/pulse/veri-g%C3%B6lü-ve-veri-ambar%C3%BC-data-lake-dl-ve-ambar%C4%B1-warehouse-dwh-fark-kapukaya/?trk=public_profile_article_view, (Ziyaret tarihi: 10 Şubat 2021).
- [8] Tayaksi C., Ada E., Kazançoğlu Y., Bulut Üretim: İşlemler Yönetiminde Yeni Bir Bulut Bilişim Modeli, *Ege Akademik Bakış Dergisi*, 2016, 16(Özel), 71-84.
- [9] Karaman İ., Bulut Bilişim Tabanlı Gerçek Zamanlı Yüz Tanıma Sistemi, Yüksek Lisans Tezi, Kırıkkale Üniversitesi, Fen Bilimleri Enstitüsü, Kırıkkale, 2019, 572912.
- [10] İnternetin tarihi, https://tr.wikipedia.org/wiki/%C4%B0nternetin_tarihi (Ziyaret tarihi: 10 Şubat 2021).
- [11] Kahraman O., Bulut Bilişimin Gelişimi ve Microsoft Azure Ortamında Örnek Bir Web Uygulaması, Yüksek Lisans Tezi, Altınbaş Üniversitesi, Fen Bilimleri Enstitüsü, İstanbul, 2019, 611389.

- [12] Ağır M., Yerel Yönetimlerde Bulut Bilişim Teknolojisi: Durum ve Gelecek Değerlendirmesi, Yüksek Lisans Tezi, Aksaray Üniversitesi, Sosyal Bilimler Enstitüsü, Aksaray, 2019, 582079.
- [13] The NIST Definition of Cloud Computing, <http://faculty.winthrop.edu/domanm/csci411/Handouts/NIST.pdf>, (Ziyaret tarihi: 10 Şubat 2021).
- [14] Carstensen J., Morgenthal J. Golden B., *Cloud Computing: Assessing The Risks*, 1st ed., IT Governance Publishing, United Kingdom, 2012.
- [15] Oyucu S., Polat H., M2M ve Iot Platformları Üzerinde Prototip Uygulama Geliştirme, *Türkiye Bilişim Vakfı Bilgisayar Bilimleri ve Mühendisliği Dergisi*, 2018, 9(2), 11-20.
- [16] Bulut Bilişim Nedir, <https://www.natro.com/blog/bulut-bilisim-nedir/>, (Ziyaret tarihi: 10 Şubat 2021).
- [17] Rouse, M., Cloud Computing, <https://searchcloudcomputing.techtarget.com/definition/cloud-computing>, (Ziyaret tarihi: 10 Şubat 2021).
- [18] Topaloğlu M., Özkişi H., Tekkanat E., *Bulut Bilişim*, 1. Baskı, Seçkin Yayıncılık, İstanbul, 2017.
- [19] Microsoft Azure Documantation, What is a private cloud?, <https://azure.microsoft.com/en-us/overview/what-is-a-private-cloud/>, (Ziyaret tarihi: 10 Şubat 2021).
- [20] Microsoft Azure Documantation, What is a hybrid cloud?, <https://azure.microsoft.com/en-gb/overview/what-is-hybrid-cloud-computing/#:~:text=A%20hybrid%20cloud%E2%80%94sometimes%20called,to%20be%20shared%20between%20them.>, (Ziyaret tarihi: 10 Şubat 2021).
- [21] Bulut Bilişim Mimarisi, <https://www.sys.com.tr/tr/bulut-bilisim-mimarisi/>, (Ziyaret tarihi: 10 Şubat 2021).
- [22] Bulut Bilişim, https://tr.wikipedia.org/wiki/Bulut_bili%C5%9Fim, (Ziyaret tarihi: 10 Şubat 2021).
- [23] Microsoft Azure Documantation, What is SaaS, <https://azure.microsoft.com/en-us/overview/what-is-saas/>, (Ziyaret tarihi: 10 Şubat 2021).
- [24] Demircan B., Endüstriyel Tabanlı Sistem İçin Bulut Bilişim Tabanlı Nesnelerin İnterneti Uygulaması, Yüksek Lisans Tezi, Balıkesir Üniversitesi, Fen Bilimleri Enstitüsü, Balıkesir, 2019, 560625.
- [25] Ghosh M., Xu L., Gupta I., Resource Management: Performance Assuredness in Distributed Cloud Computing via Online Reconfigurations, Editors:

- Campbell R. H., Kamhoua C. A., Kwiat K. A., *Assured Cloud Computing*, 1st ed., Wiley-IEEE Pres, 160-236, 2018.
- [26] Bulut Bilişim'in Hizmet Modelleri Nelerdir? IaaS, PaaS ve SaaS Nedir?, <https://radore.com/blog/bulut-bilisimin-hizmet-modelleri-nelerdir-iaas-paas-ve-saas-nedir.html>, (Ziyaret tarihi: 10 Şubat 2021).
- [27] Sazak, T. ve Albayrak, Y., Nesnelerin İnterneti (IoT) Üzerine Ortam Verilerini Toplayan ve Uzaktan Takibini Sağlayan Bir Sistem Tasarımı, *19. Akademik Bilişim Konferansı -- AB 2017*, Aksaray, Türkiye, 8-10 Şubat 2017.
- [28] Hongyan W., Zengliang M., Yong W., Enyu Z., The Model of Big Data Cloud Computing Based on Extended Subjective Logic, *2020 IEEE International Conference on Power, Intelligent Computing and Systems (ICPICS)*, Shenyang, China, 28-30 July 2020.
- [29] Yazır S., Türkiye'de Bulut Bilişimin Teknolojik Gelişimi ve Bulut Platformu Üzerinde Örnek Bir Kişisel Web Uygulamasının Sunulması, Yüksek Lisans Tezi, Necmettin Erbakan Üniversitesi, Fen Bilimleri Enstitüsü, Konya, 2018, 493831.
- [30] Arslan M.Ş., Bulut Bilişim Uygulamaları ile Veritabanlarına Erişim: Ulakbim ve Üniversite Kütüphaneleri İçin Bir Model Önerisi, Yüksek Lisans Tezi, Atatürk Üniversitesi, Sosyal Bilimler Enstitüsü, Erzurum, 2019, 583010.
- [31] Çebi G., Homomorphic Encryption In Cloud Computing, Master's Thesis, Bahçeşehir Üniversitesi, *Graduate School Of Natural And Applied Sciences Cyber Security*, İstanbul, 2019, 586562.
- [32] Şanlı, O., Bulut Bilişim, Akademik Bilişim Konferansı. <http://ab.org.tr/ab11/bildiri/34.pdf>, (Ziyaret tarihi: 10 Şubat 2021).
- [33] Wyld, D., The Cloudy Future Of Government It: Cloud Computing and The Public Sector Around The World. *International Journal of Web & Semantic Technology (IJesT)*, 2010, **1**(1), 1-20.
- [34] Semilof M., What is cloud migration? An introduction to moving to the cloud, TechTarget, <https://searchcloudcomputing.techtarget.com/definition/cloud-migration>, (Ziyaret tarihi: 10 Şubat 2021).
- [35] Tonta, Y., Bilgi Toplumu ve Bilgi Teknolojisi, *Türk Kütüphaneciliği*, 1999, **13**(4), 363- 375.
- [36] Aydın H., Üniversite Kütüphaneleri ve Bulut Bilişim: Kavramsal Bir Model Önerisi, Doktora Tezi, Hacettepe Üniversitesi, Sosyal Bilimler Enstitüsü, Ankara, 2017, 463184.
- [37] Hossain, S., What is IBM Bluemix?, IBM, <https://www.ibm.com/blogs/cloud-computing/2014/04/21/ibm-codename-bluemix/>, (Ziyaret tarihi: 10 Şubat 2021).

- [38]. What is Azure?, <https://azure.microsoft.com/en-us/overview/what-is-azure/>, (Ziyaret tarihi: 10 Şubat 2021).
- [39] Google Cloud, Ürün ve Hizmetler, <https://cloud.google.com/products/?hl=tr>, (Ziyaret tarihi: 10 Şubat 2021).
- [40] Bulut Bilişim İçin Amazon Web Services, <https://aws.amazon.com/what-is-aws>, (Ziyaret tarihi: 10 Şubat 2021).
- [41] Alkan M., Makine Öğrenimi Nedir?, Türkiye'nin Endüstri 4.0 Platformu, <https://www.endustri40.com/makine-ogrenimi-nedir/>, (Ziyaret tarihi: 10 Şubat 2021).
- [42] Şafak H., Makine Öğrenmesi Nedir?, Medium, <https://medium.com/t%C3%BCrkiye/makine-%C3%B6%C4%9Frenmesi-nedir-20dee450b56e>, (Ziyaret tarihi: 10 Şubat 2021).
- [43] Makine Öğrenmesi Nedir?, <https://azure.microsoft.com/tr-tr/overview/what-is-machine-learning-platform/>, (Ziyaret tarihi: 10 Şubat 2021).
- [44] Proente, Makine Öğrenimi Nedir?, Proente Otomasyon, <https://proente.com/makine-ogrenimi-nedir/>, (Ziyaret tarihi: 10 Şubat 2021).
- [45] Diri B., Makine Öğrenmesine Giriş, Yıldız Teknik Üniversitesi, <https://docplayer.biz.tr/6389888-Makine-ogrenmesine-giris-machine-learning-ml.html>, (Ziyaret tarihi: 10 Şubat 2021).
- [46] Koyum A., Afşin E., Derin Öğrenme ile İki Boyutlu Optik Karakter Tanıma, *Türkiye Bilişim Vakfı Bilgisayar Bilimleri ve Mühendisliği Dergisi*, 2017, **10**(1), 11-14.
- [47] Aladağ E., Makine Öğrenmesi Nedir?, <https://www.emrealadag.com/makine-ogrenmesi-nedir.html>, (Ziyaret tarihi: 10 Şubat 2021).
- [48] S. E. Seker, C. Mert, K. Al-Naami, N. Ozalp, and U. Ayan, Time Series Analysis On Stock Market For Text Mining Correlation Of Economy News, *International Journal of Social Sciences and Humanity Studies*, 2013, **6**(1), 69–91.
- [49] Covid-19 Salgınına Yönelik Zaman Serisi Verileri ile Prophet Model Kullanarak Makine Öğrenmesi Temelli Vaka Tahminlemesi, *Avrupa Bilim ve Teknoloji Dergisi*, 2020, Sayı 19, 827-835.
- [50] Taylor S. J., Letham B., Prophet: Forecasting At Scale, Facebook Research, <https://research.fb.com/blog/2017/02/prophet-forecasting-at-scale/>, (Ziyaret tarihi: 10 Şubat 2021).
- [51] Kıran B., Zaman Serileri Analizi, İstanbul Üniversitesi Uzaktan Eğitim Uygulama ve Araştırma Merkezi, http://auzefkitap.istanbul.edu.tr/kitap/ekonometri_ue/zamanserilerianalizi.pdf, (Ziyaret tarihi: 10 Şubat 2021).

- [52] AWS Lambda Developer Guide, AWS, <https://docs.aws.amazon.com/lambda/latest/dg/lambda-dg.pdf#welcome>, (Ziyaret tarihi: 10 Şubat 2021).
- [53] Amazon S3, AWS, <https://aws.amazon.com/tr/s3/>, (Ziyaret tarihi: 10 Şubat 2021)
- [54] AmazonS3 SSS, AWS, <https://aws.amazon.com/tr/s3/faqs/>, (Ziyaret tarihi: 10 Şubat 2021), (Ziyaret tarihi: 10 Şubat 2021).
- [55] Amazon S3 Nedir ve Bucket Nasıl Oluşturulur?, AWS, <http://www.diabolikss.com/amazon-s3-nedir-ve-bucket-nasil-olusturulur/>, (Ziyaret tarihi: 10 Şubat 2021).
- [56] Amazon Simple Storage Service User Guide, AWS, <https://docs.aws.amazon.com/AmazonS3/latest/userguide/s3-userguide.pdf#Welcome>, (Ziyaret tarihi: 10 Şubat 2021).
- [57] What is AWS Glue?, AWS, <https://docs.aws.amazon.com/glue/latest/dg/what-is-glue.html>, (Ziyaret tarihi: 10 Şubat 2021).
- [58] AWS Glue: How It Works, AWS, <https://docs.aws.amazon.com/glue/latest/dg/how-it-works.html>, (Ziyaret tarihi: 10 Şubat 2021).
- [59] AWS Glue Concepts, AWS, <https://docs.aws.amazon.com/glue/latest/dg/components-key-concepts.html>, (Ziyaret tarihi: 10 Şubat 2021).
- [60] AWS Glue Components, AWS, <https://docs.aws.amazon.com/glue/latest/dg/components-overview.html>, (Ziyaret tarihi: 10 Şubat 2021).
- [61] Converting Semi-Structured Schemas to Relational Schemas, AWS, <https://docs.aws.amazon.com/glue/latest/dg/schema-relationalize.html>, (Ziyaret tarihi: 10 Şubat 2021).
- [62] Build, Train, and Deploy a Machine Learning Model with Amazon SageMaker, AWS, <https://aws.amazon.com/getting-started/hands-on/build-train-deploy-machine-learning-model-sagemaker/> (Ziyaret tarihi: 10 Şubat 2021).
- [63] What Is Amazon SageMaker?, AWS, <https://docs.aws.amazon.com/sagemaker/latest/dg/whatis.html>, (Ziyaret tarihi: 10 Şubat 2021).
- [64] What Is Amazon SageMaker?, AWS, <https://docs.aws.amazon.com/sagemaker/latest/dg/sagemaker-dg.pdf#whatis>, (Ziyaret tarihi: 10 Şubat 2021).

- [65] Amazon Forecast, AWS, <https://aws.amazon.com/forecast/>, (Ziyaret tarihi: 10 Şubat 2021).
- [66] What Is Amazon Forecast?, AWS, <https://docs.aws.amazon.com/forecast/latest/dg/forecast.dg.pdf#what-is-forecast>, (Ziyaret tarihi: 10 Şubat 2021).
- [67] What Is Amazon QuickSight?, AWS, <https://docs.aws.amazon.com/quicksight/latest/user/amazon-quicksight-user.pdf#welcome> (Ziyaret tarihi: 10 Şubat 2021).
- [68] Parquet, databricks, <https://databricks.com/glossary/what-is-parquet#:~:text=Back%20to%20glossary,like%20CSV%20or%20TSV%20files.>, (Ziyaret tarihi: 10 Şubat 2021).
- [69] Dosya Formatından Apache Parquet Formatına Dönüştürme, Medium, <https://medium.com/baybaynakit/dosya-format%C4%B1ndan-apache-parquet-format%C4%B1na-d%C3%B6n%C3%BCrme-aeff16ad0015>, (Ziyaret tarihi: 10 Şubat 2021).
- [70] Using AWS SageMaker's Linear Learner to Solve Regression Problems, AWS, <https://towardsdatascience.com/using-aws-sagemakers-linear-learner-to-solve-regression-problems-36732d802ba6>, (Ziyaret tarihi: 10 Şubat 2021).

KİŞİSEL YAYIN VE ESERLER

- [1] Duru N., **Sağlam F.Z.**, Bulut Tabanlı Sunucusuz Bir ETL Çözümü “AWS Glue”, *3. Uluslararası Erciyes Bilimsel Araştırmalar Kongresi*, Kayseri, 9 Mayıs 2020



ÖZGEÇMİŞ

Fatma Zehra Sağlam lise eğitimini Hüseyin Avni Sözen Anadolu Lisesi'nde tamamladı. 2012 yılında başladığı Kocaeli Üniversitesi Mühendislik Fakültesi Bilgisayar Mühendisliği Bölümü'nden 2016 yılında mezun oldu. 2017 yılında Kocaeli Üniversitesi Fen Bilimleri Enstitüsü Bilgisayar Mühendisliği Anabilim Dalı'nda tezli yüksek lisans programına başladı. 2016'dan itibaren özel sektörde bilgisayar mühendisi olarak çalışmaya devam etmektedir.

