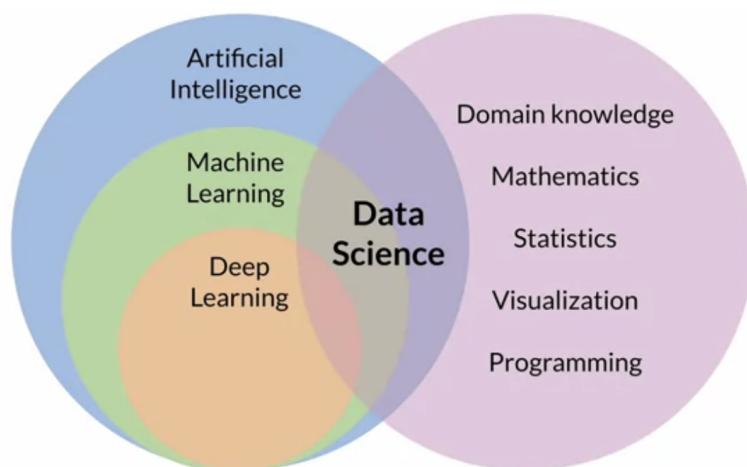


## WEEK 1:

Bu haftaya, pratik veri bilimi disiplinine kısa bir girişle başlanmış ve bulutta veri bilimi yapmanın faydalari tartışılmıştır. Pratik veri bilimi, veri bilimi ve makine öğrenimi becerilerinizi geliştirmenize, neredeyse her miktarda veriyle çalışmanız ve kullanım durumunuza en verimli şekilde uygulamanıza yardımcı olur. Veri bilimi projelerini buluta taşıyarak artık dizüstü bilgisayar, işlemci ve bellek gibi kaynak kısıtlamalarına bağlı kalınmamaktadır. Neredeyse her boyuttaki veride veri analizi yapılabilmektedir. Verileri dilimleyebilir ve veri dönüşümleri paralel olarak çalıştırılabilir. Model eğitimini hızlandırmak için CPU'dan GPU'ya geçilebilir ve bunların çoğu sadece birkaç tıklamayla yapılabilir. Bu modülde, metin verileriyle çalışılmıştır. Özellikle, ürün incelemelerinin duygusal analizi için çok sınıflı bir sınıflandırmaya odaklanılmıştır. Ve tabii ki, veri bilimi veri ile başlar. Bu modülde, bu metin sınıflandırma görevini uygulamak için kullanılacak veri kümesi paylaşılmıştır. Verileri merkezi bir depoya alma ve pratik veri bilimi ve makine öğrenimi araç takımından çeşitli araçlar kullanarak verileri keşfetmenin ne kadar kolay olduğu görülecektir. Ayrıca, interaktif sorgular kullanılarak verilerin daha fazla analiz edilmesi ve sonuçların görselleştirilmesi öğrenilecektir. Analiz, model geliştirme sürecinde gelecekteki görevler için önemli bilgiler ortaya çıkaracaktır.

### Practical Data Science:

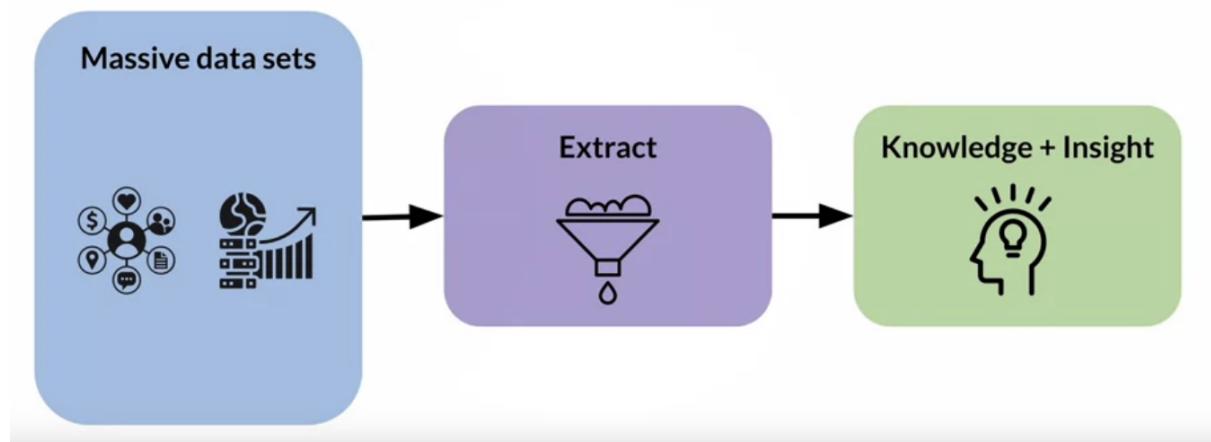
Bu modülde, ürün incelemelerinin duygusal analizi için çok sınıflı sınıflandırmaya odaklanılmıştır. Bu metin sınıflandırma görevini uygulamak için çalışacağımız veri kümesini nasıl içe aktaracağımızı, keşfedeceğimizi ve analiz edeceğimizi öğreneceğiz. Ancak kullanım durumu açıklamasına geçmeden önce, bulutta pratik veri biliminin arkasındaki kavramları kısaca tanımlamak ve uygulayacağımız veri bilimi ve makine öğrenimi kavramlarını ve kullanacağımız araç seti hakkında konuşmak istiyorum.



Başlangıç olarak yapay zeka, makine öğrenimi, derin öğrenme ve veri bilimi kavramlarını karşılaştırıralım. Yapay zeka veya AI, genellikle makinelerin insan davranışını taklit etmesini sağlayan bir teknik olarak tanımlanır. Yeni bir kavram değildir, yapay zeka alanı 1950'lardan

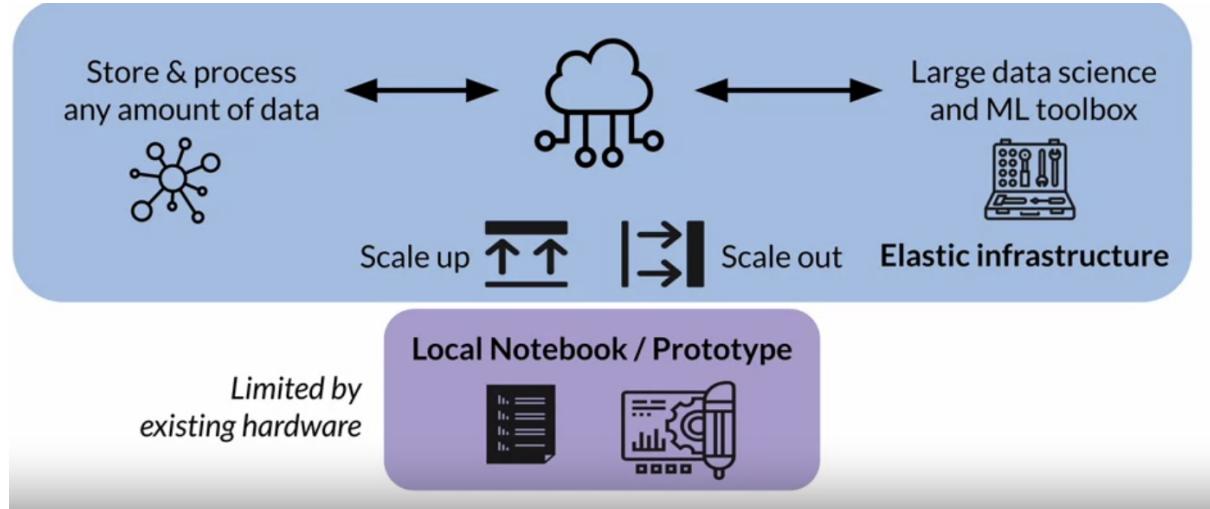
beri incelenmektedir. Makine öğrenimi veya ML, AI'in bir alt kümesidir ve açıkça programlanmadan verilerden öğrenme yeteneğine sahip istatistiksel yöntemler ve algoritmalar kullanır. Son olarak, derin öğrenme, makine öğreniminin başka bir alt kümesidir ve yapay sinir ağlarını kullanarak verilerden öğrenme sağlar. Eğer veri biliminde yeniyseniz, bu disiplinin tüm alanlara dokunduğunu göreceksiniz. Veri bilimi gerçekten iş ve alan bilgisiyle matematik, istatistik, veri görselleştirme ve programlama becerilerini birleştiren disiplinlerarası bir alandır.

## PRACTICAL DATA SCIENCE



### Peki practical data science nedir?

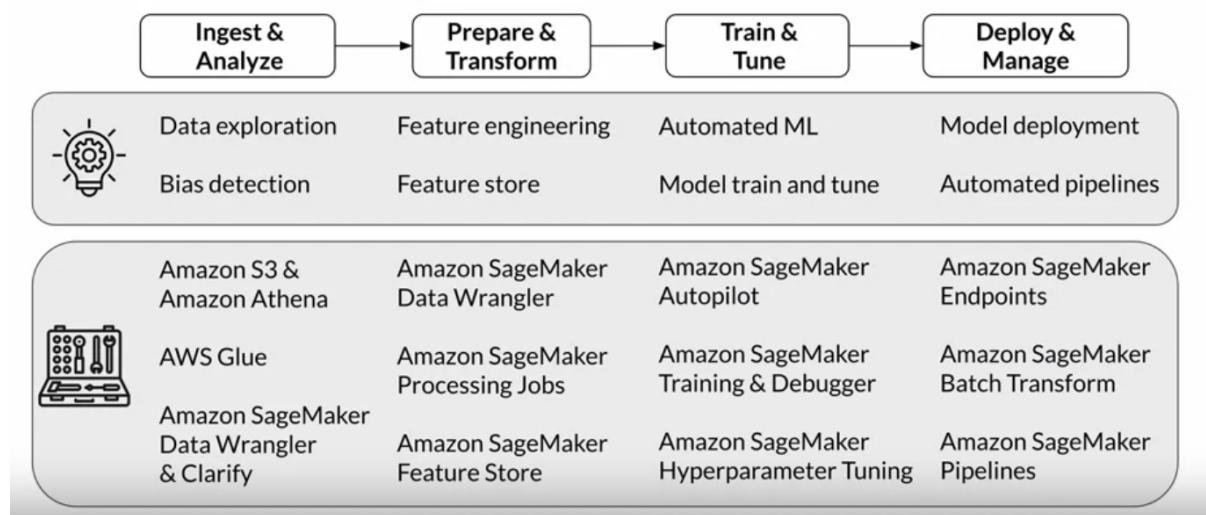
Pratik veri bilimi, veri bilimi ve makine öğrenimi becerilerinizi geliştirmenize, neredeyse her miktarda veriyle çalışmaya ve kullanım durumlarını en verimli şekilde uygulamanıza yardımcı olur. Bu, dizüstü bilgisayar gibi yerel bir geliştirme ortamında küçük düzenlenmiş veri kümesiyle çalışmaktan farklıdır. Pratik veri bilimi, sosyal medya kanallarından, mobil ve web uygulamalarından, kamu veya şirket içi veri kaynaklarından ve kullanım durumunuzdan bağımsız olarak, büyük veri kümelerini işlemeye yönelikir. Ve bu veriler sık sık düzensiz, hatalı veya hatta yetersiz belgelenmiş olabilir. Pratik veri bilimi, verileri analiz etmek ve temizlemek, ilgili özellikleri çıkarmak için araçlar sağlayarak bu sorunlarla başa çıkar. Ve bu süreç, veri biliminin en nihai amacı olan bilgi sıkıştırma ve büyük veri kümelerinden kavrayış elde etmeye (gaining insight) yönlendirir.



### Bulutta pratik veri bilimi ile ilgili fark nedir?

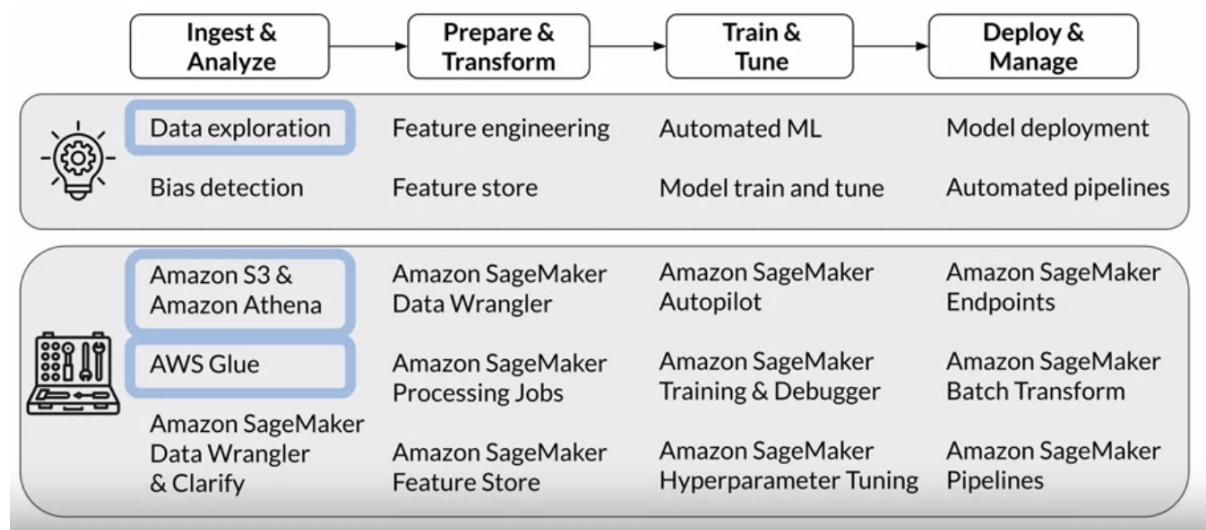
Eğer veri bilimi projelerini yerel bir bilgisayar üzerindeki notebook veya IDE ortamında geliştirirsınız, mevcut donanım kaynaklarıyla sınırlı kalırsınız. Örneğin, işlemcinizin ne kadar veri işleyebileceğini ve potansiyel olarak belleğe kaydedebileceğinizi dikkatlice takip etmek zorunda kalırsınız. Modelinizi eğitmek ve ayarlamak için ne kadar CPU işlem gücünüz olduğunu kontrol etmeniz gereklidir. Eğer daha fazlasına ihtiyacınız varsa, ek bilgisayar kaynakları satın almanız gereklidir. Bu süreç, hızlı bir şekilde gelişmenizi ve hareket etmenizi engeller. Bulutta veri bilimi projelerini geliştirmenin ve çalıştırmanın en büyük avantajlarından biri, bulutun sunduğu esneklik ve elastiklidir. Model eğitiminiz çok uzun sürüyor ve seçtiğiniz hesaplama örneğinin tüm CPU kaynaklarını tüketiyor olabilir. Daha fazla CPU kaynağına sahip bir hesaplama örneğini kullanabilir, hatta GPU tabanlı bir hesaplama örneğine geçebilirsiniz. Buna "dikey ölçeklendirme" (scaling up) denir. Örneğin, modelinizi tek bir CPU örneğinde eğitmek yerine, farklı hesaplama örneklerinde paralel olarak dağıtılmış model eğitimi yapmak isteyebilirsiniz. Buna "yatay ölçeklendirme" (scaling out) denir ve her iki senaryo da bulutta birkaç saniye içinde gerçekleştirilebilir. Model eğitimi tamamlandığında, örnekler de sonlandırılır. Bu, yalnızca kullandığınız kadar ödeme yapmanız anlamına gelir. Bu elastik altyapı, neredeyse her miktarda veriyi saklamanıza ve işlemenize olanak tanır, çünkü altyapı ihtiyacınız olan kaynaklara göre otomatik olarak ölçeklenir. Aynı zamanda, hızlı bir şekilde yeni veri kümeleri, yeni modeller, yeni kod veya yeni makine

öğrenimi kütüphanelerini deneyebilir ve başlangıçta herhangi bir yatırım yapmadan inovasyon yapabilirsiniz. Bulut aynı zamanda görevlerinizi mümkün olduğunda hızlı ve verimli bir şekilde gerçekleştirmek için seçebileceğiniz geniş bir veri bilimi ve makine öğrenimi araç kutusuyla birlikte gelir.



Gelecek haftalar boyunca çalışacağımız veri bilimi ve makine öğrenimi araç kutusuna bir göz atalım. Her tipik makine öğrenimi süreci, elbette, veriyle başlar. Veriyi alma ve analiz etme aşamasında, veriyi keşfedecek ve olası istatistiksel önyargıları analiz edeceğiz. Bu adım için kullanacağımız araç kutusuna göz attığımızda, veriyi almak, depolamak ve sorgulamak için Amazon Simple Storage Service (Amazon S3) ve Amazon Athena'yı kullanacağız. Veriyi şemasıyla birlikte kataloglayacak ve düzenleyecek AWS Glue'u kullanacağız. Verideki istatistiksel önyargıları tespit etmek için de Amazon SageMaker Data Wrangler ve Amazon SageMaker Clarify ile çalışmayı öğreneceğiz. Sonrasında, model geliştirme aşamasına geçeceğiz. Veriyi model eğitimi için hazırlamaya ve dönüştürmeye başlayacağız. Özellikle mühendisliği yapmayı öğrenecek ve bir özellik deposu kavramı ve faydaları hakkında bilgi sahibi olacağız. Yine, Amazon SageMaker hizmetinin bir parçası olan güçlü araçlarla çalışacağız. Model eğitimi ve ayarlama aşamasında, otomatik makine öğrenimini kullanmayı, birincil temel çizgisinin başarı oranını kontrol etmeyi ve kullanım durumuna yönelik en iyi model adaylarını belirlemeyi öğreneceğiz. Daha sonra özel model eğitimi ve ayarlama

yapacağımız. Yine, Amazon SageMaker hizmetinin bir parçası olan ek araçlarla çalışacağız. Model dağıtımını ve yönetimi aşamasında, farklı model dağıtım seçenekleri ve stratejilerini tartışmayı ve model geliştirmeyi otomatik bir süreç haline getirmeyi öğreneceğiz. Benzer şekilde, araç setimiz Amazon SageMaker temelli olacak.

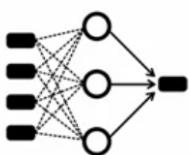


Bu hafta, veri keşfine başlayacağımız. Amazon S3 ve AWS Glue'u kullanarak veriyi nasıl içe aktarabileceğimizi ve kataloglayabileceğimizi, ardından Amazon Athena'yı kullanarak SQL sorguları ile veriyi nasıl keşfedebileceğimizi inceleyeceğiz.

## USE CASE AND DATA SCIENCE

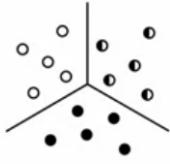
Önümüzdeki birkaç dakika içinde, üzerinde çalıştığımız kullanım durumları ve veri kümesi hakkında konuşacağız.

Popular ML Tasks and learning paradigms



Classification  
& Regression

*Supervised*



Clustering

*Unsupervised*



Image Processing

*Computer Vision*

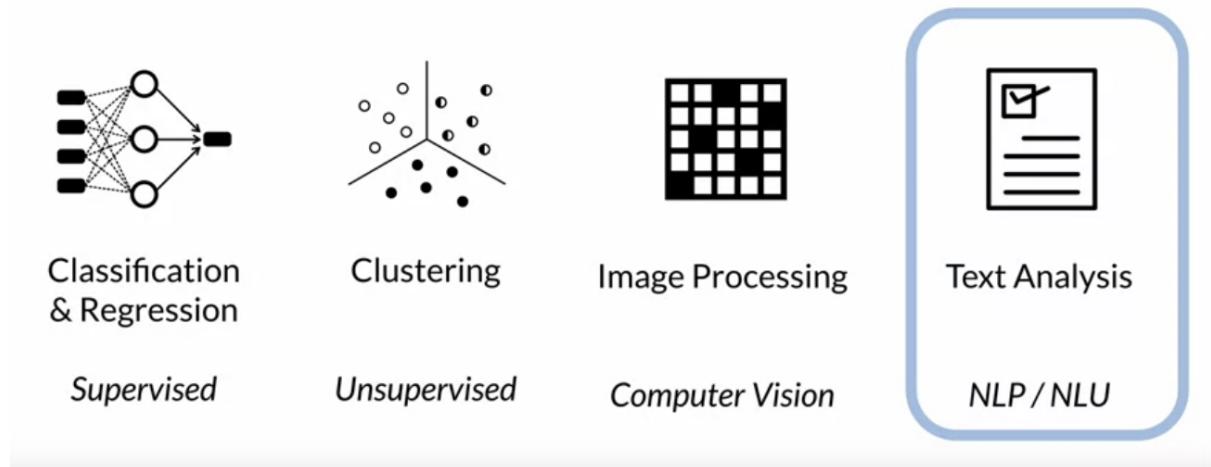


Text Analysis

*NLP / NLU*

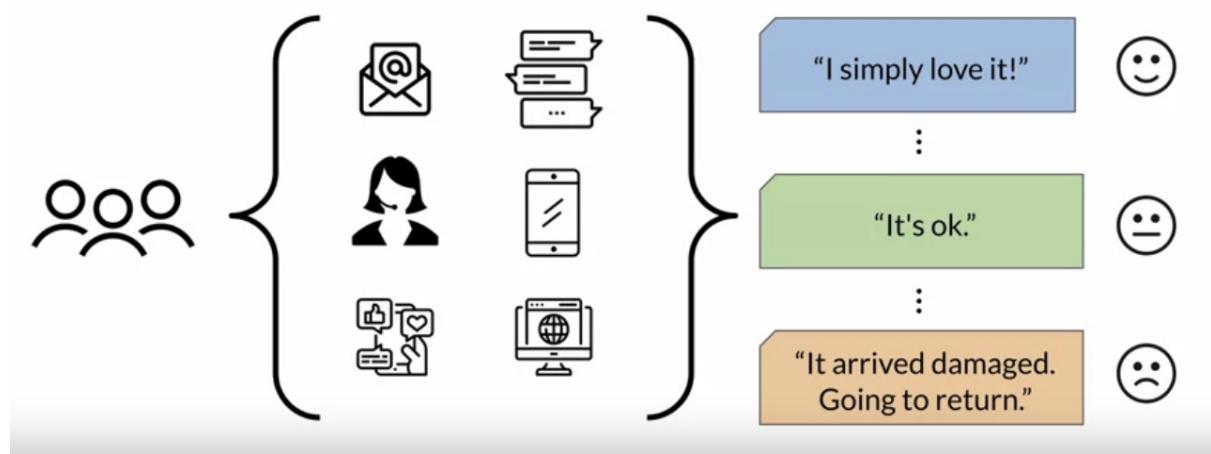
Popüler makine öğrenimi görevleri, sınıflandırma ve regresyon problemleri olup, bunlar denetimli öğrenme örneklerindendir. Denetimli öğrenmede, algoritmayı etiketli verilerle besleyerek öğrenirsiniz. Sınıflandırma ile hedef, girdi örneğini tanımlanmış bir sınıf'a atamaktır. Örneğin, aldığım bu e-posta spam mı yoksa spam değil mi? Buna karşılık, regresyon, bir dizi ilgili ve ilgisiz girdi değişkeni verilerek ev fiyatı gibi sürekli bir değeri tahmin etmek için istatistiksel yöntemler uygular. Diğer bir popüler görev ise kümelemedir ve kümeleme, denetimsiz öğrenmenin bir örneğidir. Burada veriler etiketlenmez ve dolayısıyla örnekler sağlanmaz. Kümeleme algoritması, verilerdeki desenleri bulmaya çalışır ve veri noktalarını farklı kümeler halinde grüplendirmeye başlar. Bir kümeleme kullanım durumu, pazarlama amaçları için farklı müşteri segmentlerini belirlemek olabilir. AI uygulamalarının alanlarına daha detaylı baktığımızda, görüntü işleme büyük bir görevdir ve daha geniş bir bilim alanı olan bilgisayarlı görünün bir parçasıdır. Görüntüleri köpek ve kedi resimleri olarak sınıflandırmamız, arabamızın sürücü destek sistemlerinin hız işaretleri ile ağaçları ayırt etmesine yardımcı olmak için görüntüdeki segmentleri belirlememiz veya bir görüntüdeki marka etiketlerini algılamamız gerekebilir. Bilisel / bilgisayarlı görüşün ardından, metin analizi son yıllarda endüstride popüleritesini ve araştırmasını geri kazanmıştır. Doğal Dil İşleme (NLP) veya Doğal Dil Anlama (NLU) alanlarında 1950'lardan beri çalışılmaktadır; ancak derin öğrenme ve yapay sinir ağları mimarilerindeki ilerlemeler sayesinde, nöral makine çevirileri (neural machine translations), duygusal analizi (sentiment

analysis), soru cevaplama (question answering) ve diğerleri gibi daha gelişmiş NLP görevlerini görmekteyiz.



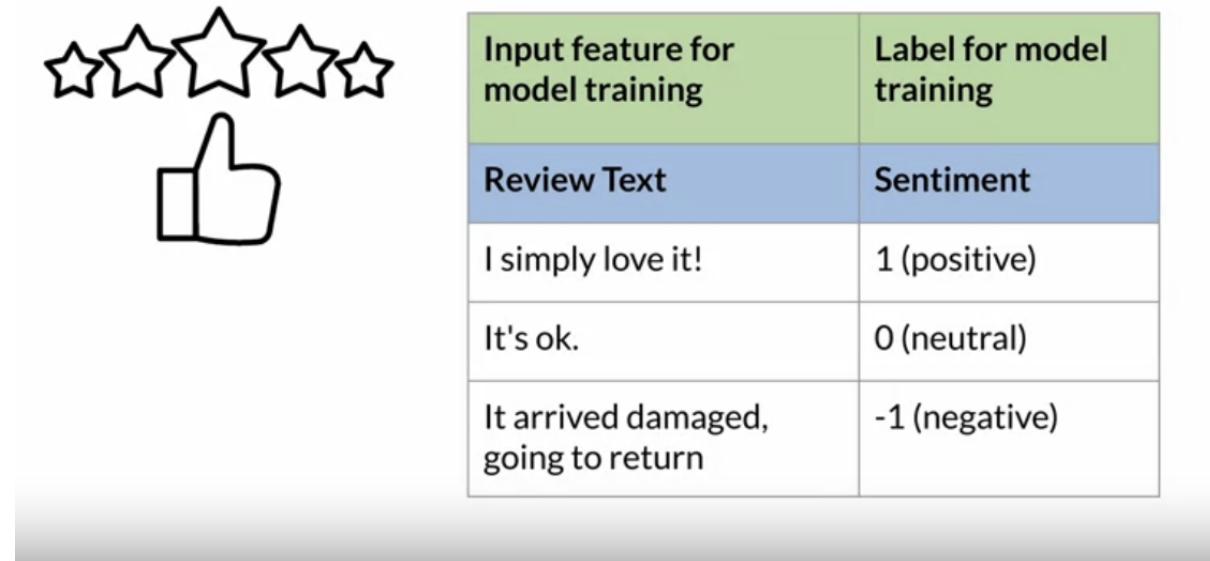
Ve bu kursa odaklandığımız görev metin analizi ve NLP alanı olacak ve bu alanda çok daha fazla bilgi edineceğiz

### **Multi-class Classification for Sentiment Analysis of Product Reviews**



Daha spesifik olarak, ürün incelemelerinin duygusal analizi için çok sınıflı sınıflandırma yapacağız. Varsayıyalım ki bir e-ticaret şirketinde çalışıyoruz ve çeşitli ürünlerin çevrimiçi olarak satıyoruz. Müşterilerimiz, ürün geri bildirimlerini çevrimiçi kanallarda bırakıyorlar. Bu geri bildirimler, e-posta göndererek, web sitemizdeki sohbet FAQ mesajları yazarak, destek merkezimize çağrı yaparak veya şirketimizin mobil uygulamasında, popüler sosyal ağlarda veya ortak web sitelerinde mesajlar paylaşarak olabilir. Bir işletme olarak, müşteri geri bildirimini mümkün olan en kısa sürede yakalamak ve pazar trendlerinde veya müşteri davranışlarında herhangi bir değişikliği tespit ederek olası ürün sorunları hakkında uyarı almak isteriz. Görevimiz, bu ürün incelemelerini girdi olarak alan bir Doğal Dil İşleme (NLP)

modeli oluşturmaktır. Ardından modeli kullanarak incelemelerin duygusunu olumlu, nötr veya olumsuz olarak üç sınıfa sınıflandıracağız. Örneğin, "I simply love it!!" gibi bir inceleme olumlu sınıfa sınıflandırılmalıdır.



Çok sınıflı sınıflandırma, denetimli bir öğrenme görevidir; bu nedenle, taksonomi sınıflandırıcı modelimizi doğru bir şekilde öğrenmesi için ürünleri ve ürün incelemelerini doğru duyu sınıflarına sınıflandırmak için örnekler sağlamamız gerekmektedir. Ürün incelemelerini keşfetmek için harika bir kaynak, e-ticaret siteleridir. İnceleme metnini model eğitimi için girdi özelliği olarak kullanabilir ve duyu durumunu model eğitimi için bir etiket olarak kullanabiliriz. Duyu durumu sınıfı genellikle model eğitimi için bir tamsayı değeri olarak ifade edilir; örneğin, olumlu duyu için 1, tarafsız duyu için 0 ve olumsuz duyu için -1 gibi.

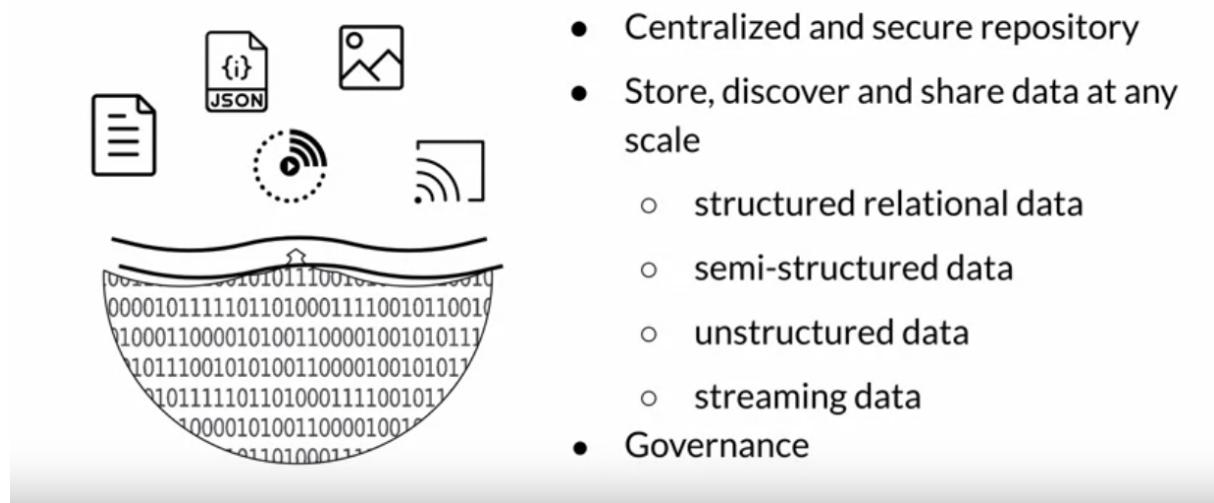
## Data Ingestion and Exploration

Bulut üzerinde veri bilimi yapmanın en büyük avantajlarından biri, neredeyse her miktarda veriyi depolayıp işleyebilmenizdir. Altyapı, verinizin boyutıyla esnek bir şekilde ölçeklendirilir. Üzerinde çalışacağımız ürün incelemeleri kullanım durumunu düşünelim. Hayal edelim ki e-ticaret şirketimiz, tüm çevrimiçi kanallardan müşteri geri bildirimlerini topluyor. Sosyal medya kanallarından gelen geri bildirimler, destek merkezi aramalarıyla yakalanan ve transkribe edilen geri bildirimler, gelen e-postalar, mobil uygulamalar ve web sitesi verileri ve çok daha fazlasını yakalamak zorundayız. Bunun için, farklı dosya formatlarını ele alabilen esnek ve elastik bir depoya ihtiyacımız vardır. Yapısallaştırılmış verilerle başa çıkabilmesi için CSV dosyaları gibi dosya formatlarından, aynı zamanda destek merkezi çağrı ses dosyaları gibi yapılandırılmamış verilere kadar. Ayrıca yeni veriler geldikçe depolama kapasitesini elastik olarak ölçeklemesi gerekmektedir.

## Ingest data into data lakes

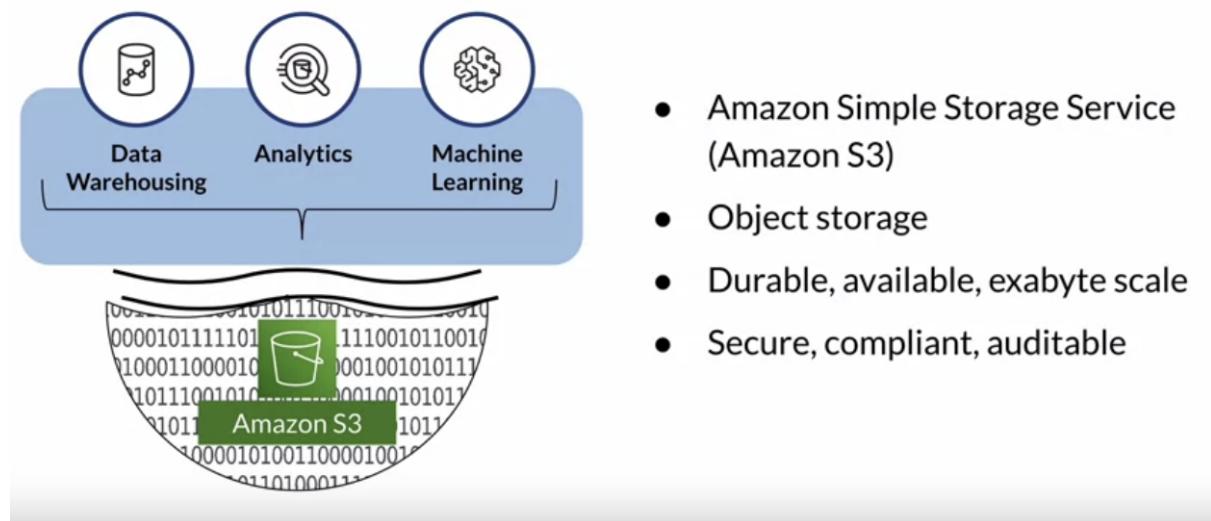
Bulut tabanlı veri gölleri bu sorunu çözmektedir. Bir veri gölünü, merkezi ve güvenli bir depo olarak düşünebilirsiniz, neredeyse her türde ve miktarda veriyi depolayabilir, keşfedebilir ve paylaşabilir. Verileri önceden herhangi bir veri dönüşümü olmadan ham biçiminde alabilirsiniz. Yapısal ilişkisel veriler olarak CSV veya TSV dosyaları biçiminde, yarı yapılandırılmış veriler olarak JSON veya XML dosyaları biçiminde veya yapılandırılmamış veriler olarak resimler, ses ve medya dosyaları biçiminde olsun, tüm veri türlerini depolayabilirsiniz. Ayrıca, sürekli olarak log dosyalarını besleyen bir uygulama veya sosyal medya kanallarından gelen akışlı veriler gibi veri akışlarını da veri gölüne alabilirsiniz. Bir veri gölü yönetilmelidir. Yeni verilerin herhangi bir zamanda gelmesi nedeniyle, yeni verileri keşfetmek ve kataloglamak için yöntemler uygulamanız gerekmektedir. Ayrıca verilere erişimi güvence altına almalı ve politika, veri güvenliği, gizlilik ve governance in place'e uygun olarak kontrol etmelisiniz. Bu governance in place ile birlikte, veri bilimi ve makine öğrenme ekplerine geniş ve çeşitli veri kümelerine erişim imkanı sağlayabilirsiniz.

“Governance in place” ifadesi, belirli bir sistemin, organizasyonun veya sürecin etkin ve düzenli bir şekilde yönetildiğini ifade eder. Bu, kuralların, politikaların ve süreçlerin uygulanması ve uyulması ile sağlanır. Özellikle veri bilimi ve veri yönetimi alanında “governance in place” ifadesi, verilerin doğru, güvenilir ve uygun bir şekilde yönetildiğini ve korunduğunu ifade eder. Bu, veri güvenliğinin sağlanması, veri gizliliği politikalarının uygulanması, veri erişimini kontrol altında tutma ve veri kalitesinin sürekli olarak izlenmesi gibi unsurları içerir. Veri gölleri ve büyük veri projelerinde, “governance in place” uygulanması, veri gölünde depolanan verilerin düzenli olarak kataloglanması, yeni verilerin izlenmesi ve belirli standartlara uygunluğunun sağlanması için önemlidir. Bu, verilerin doğru ve güvenilir bir şekilde kullanılmasını, uygun denetimlerin yapılmasını ve veri yönetiminin sorunsuz bir şekilde gerçekleştirilemesini sağlar.



## Data lakes on Amazon S3

Veri gölleri genellikle Amazon S3 gibi nesne depolama üzerine kurulur. Bu süreçte dosya ve blok depolamaya da aşina olabiliriz. Dosya depolama, verileri hiyerarsık dosya klasör yapısında düzenlenmiş bireysel dosyalar olarak saklar ve yönetir. Buna karşılık, blok depolama, verileri bloklar adı verilen bireysel parçalar olarak saklar ve yönetir. Her blok benzersiz bir kimlik alır, ancak bu blokla ilgili ekstra meta veri saklanmaz. Nesne depolamada ise veriler, verinin kendisi, nesnenin son olarak ne zaman değiştirildiği gibi ilgili meta veriler ve benzersiz bir kimlik içeren nesneler olarak saklanır. Nesne depolama, her türden büyüyen verileri saklamak ve geri almak için özellikle kullanışlıdır, bu nedenle veri gölleri için mükemmel bir temel oluşturur. Amazon S3, bulutta dayanıklı ve yüksek erişilebilirliğe sahip nesne depolamaya erişim sağlar. Sadece birkaç veri kümesi dosyasından exabaytlarca veriye kadar neredeyse her şeyi alabilirsiniz. AWS, ayrıca S3 üzerinde güvenli, uygun ve denetlenebilir bir veri gölü inşa etmenize yardımcı olacak ek araçlar ve hizmetler de sunar. Veri gölüğe sahip olduktan sonra, bu merkezi veri deposunu veri ambarı analitiği ve aynı zamanda makine öğrenimi için kullanabilirsiniz.



Veri gölü oluşturduktan sonra, bu merkezi veri deposunu veri ambarı analitiği ve aynı zamanda makine öğrenimi için kullanabilirsiniz. Şimdi, üzerinde çalışacağımız bazı ekstra araçlardan bahsedelim.

## AWS Data Wrangler

AWS Data Wrangler. AWS Data Wrangler, AWS profesyonel hizmetler ekibi üyeleri tarafından geliştirilen açık kaynaklı bir Python kütüphanesidir. Bu kütüphane, Pandas DataFrame'i AWS veri ile ilgili hizmetlerle bağlar. Pandas, çok popüler bir Python veri analizi ve manipülasyon aracıdır. AWS Data Wrangler, veri gölleri, veri ambarları veya AWS üzerindeki veri tabanlarından veri yükleme veya veri indirme işlemleri için soyutlanmış işlevler sunar. Kütüphaneyi PIP aracılığıyla yüklemek için "PIP install AWS wrangler" komutunu kullanabilirsiniz. İşte AWS Data Wrangler ile nasıl çalışacağınızı gösteren bir

örnek kod parçası: İlk olarak, kütüphaneyi ve Pandas'ı içeri aktarırınız. Örneğin, S3 veri gölünden CSV verilerini bir Pandas DataFrame'e okumak istiyorsanız, aşağıdaki komutu kullanarak S3 okuma CSV işlevini çalıştırabilirsiniz ve veri gölünüzüne S3 yolu sağlayabilirsiniz.

- Open source Python library
- Connects pandas DataFrames and AWS data services
- Load/unload data from
  - data lakes
  - data warehouses
  - databases

```
!pip install awswrangler

import awswrangler as wr
import pandas as pd

# Retrieving the data directly from Amazon S3
df = wr.s3.read_csv(
    path='s3://bucket/prefix/')
```



## Register Data with AWS Glue Data Catalog

Bu konu başlığında kullanacağınız diğer bir araç, AWS Glue Veri Kataloğu'dur. Bu veri kataloğu hizmeti, S3'te depolanan verileri kaydetmek veya kataloglamak için kullanılır. Bir mağazada envanter almak gibi, S3 veri gölünüzde veya kovasında (her bir nesnenin depolandığı bireysel konteyner) hangi verilerin saklandığını bilmelisiniz. Veri Kataloğu Hizmetini kullanarak, veriye bir referans oluşturursunuz, yani temelde S3 ile tablo eşleştirme yaparsınız. AWS Glue tablosu, AWS Glue veritabanının içinde oluşturulur ve yalnızca veri şeması gibi meta veri bilgilerini içerir. Önemli bir nokta, hiçbir verinin taşınmadığıdır. Tüm veriler S3 konumunuzda kalır. Verinin nerede bulunacağını ve veriyi sorgulamak için hangi şemanın kullanılacağını kataloglarsınız. Veriyi manuel olarak kaydetmek yerine, AWS Glue Crawler'ı da kullanabilirsiniz. Bir Crawler, belirli bir zaman dilimi içinde çalıştırılmak veya otomatik olarak yeni verileri bulmak için kurulabilir; bu süreçte veri şemasını çıkarır ve veri kataloğunu günceller.



Name	reviews
Database	dsoaws_deep_learning
Classification	csv
Location	s3://<bucket>/<prefix>

- Creates reference to data ("S3-to-table" mapping)
- Just metadata / schema stored in tables
- No data is moved
- AWS Glue Crawlers can be set up to automatically
  - infer data schema
  - update data catalog

Veriyi nasıl kaydedebilirsiniz? Bunun için yine yukarıda tanıttığım gibi AWS Data Wrangler aracını kullanabilirsiniz. İlk adım, bir AWS Glue Veri Kataloğu veritabanı oluşturmaktır. Bunun için burada gösterildiği gibi AWS Wrangler Python kütüphanesini içeri aktarın ve sonra catalog.create\_database işlevini çağırarak oluşturmak için bir veritabanı adı belirtin. AWS Data Wrangler ayrıca CSV verilerini AWS Glue Veri Kataloğu'nda kaydetmek için kullanabileceğiniz catalog.create\_CSV\_table adında bir kolaylık işlevi sunar. Bu işlev, yalnızca belirttiğiniz AWS Glue Veri Kataloğu tablosunda şemayı ve meta verileri depolar. Asıl veriler yine S3 kovada kalır.

The screenshot shows the AWS Glue Data Catalog interface on the left and a code editor on the right. The interface displays a table with the following data:

Name	reviews
Database	dsoaws_deep_learning
Classification	csv
Location	s3://<bucket>/<prefix>

The code editor contains the following Python script:

```
import awswrangler as wr

# Create a database in the
# AWS Glue Data Catalog
wr.catalog.create_database(
    name=...)

# Create CSV table (metadata only) in the
# AWS Glue Data Catalog
wr.catalog.create_csv_table(
    table=...,
    column_types=...,
    ...)
```

## Query Data with Amazon Athena

S3'te depolanan verilere Amazon Athena adlı bir araç kullanarak sorgu yapabilirsiniz. Athena, verilerinizi keşfetmek için standart SQL sorguları çalıştırmanızı sağlayan etkileşimli bir sorgu hizmetidir. Athena, altyapı kurmanıza gerek olmadan bu sorguları çalıştırmanızı sağlar ve sorgulamak istediğiniz veriler ne kadar büyük olursa olsun, basitçe SQL sorgunuzu yazabilir ve AWS Glue Veri Kataloğu'nda sağladığınız veri kümesi şemasına başvurabilirsiniz. Veri yüklenmez veya taşınmaz ve işte bir örnek SQL sorgusu: AWS Glue tarafından "reviews" adlı tabloya işaret eden ve S3'de depolanan veri kümesine işaret eden tüm ürün kategorilerini listeleyin. Bu sorguyu çalıştmak için yine daha önce tanıttığımız AWS Data Wrangler aracını kullanabilirsiniz. Çalışığınız Python ortamından Data Wrangler, Athena, read\_SQL\_query işlevini kullanın. Yazdığınız SQL ifadesini ve SQL sorgusunda burada başvuracağınız tablonun bulunduğu AWS Glue veritabanını gönderin. Yine bu veritabanı ve tablo yalnızca verilerinizin meta verilerini içerir. Veri hala S3'te bulunur ve bu Python komutunu çalıştırığınızda AWS Data Wrangler, bu SQL sorgusunu Amazon Athena'ya gönderecektir. Athena daha sonra belirlen veri kümesinde sorguyu çalıştırır ve sonuçları S3'e kaydeder ve ayrıca sonuçları burada gösterilen komutta belirlen şekilde bir Pandas DataFrame olarak döndürür.



Amazon  
Athena

- Query data in S3
- Using SQL
- No infrastructure to set up
- Schema lookup in AWS Glue Data Catalog
- No data to load

```
import awswrangler as wr
```

Python

```
# Create Amazon Athena S3 bucket
wr.athena.create_athena_bucket()
```

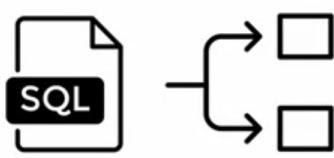
```
# Execute SQL query on Amazon Athena
df = wr.athena.read_sql_query(
    sql=...,
    database=...)
```



```
'SELECT product_category FROM reviews'
```

SQL

Bu basitlik göz önüne alındığında, bunun ne kadar özel olduğunu merak ediyor olabilirsiniz ve kabul etmeliyim ki burada gösterdiğim SQL sorgusu oldukça basitti. Ancak hayal edin, sadece gigabaytlar değil, potansiyel olarak terabaytlar veya petabaytlar düzeyinde verilere karşı çalıştmak için oldukça karmaşık analitik sorgular oluşturuyorsunuz. Athena'yı kullanarak, bu sorguyu desteklemek için herhangi bir hesaplama ve bellek kaynağı ile uğraşmanıza gerek yoktur, çünkü Athena otomatik olarak ölçeklendirilir ve sorgunuzu daha basit sorgulara böler ve verilerinizin karşısında paralel olarak çalıştırır. Athena, bu tam kullanım amacı için geliştirilmiş açık kaynaklı dağıtık bir SQL motoru olan Presto'ya dayanır ve veri kaynaklarından bağımsız olarak etkileşimli sorguları çalışma yeteneği sunar. Ve unutmayın, herhangi bir kurulum veya altyapı kurulumu gerekmez ve veri taşınması gerektirmez. Verinizi AWS Glue ile kaydedin ve Python ortamınızın rahatlığından Amazon Athena'yı kullanarak veri kümelerini keşfedin.



presto

- Complex analytical queries
- Gigabytes > Terabytes > Petabytes
- Scales automatically
- Runs queries in parallel
- Based on Presto
- No infrastructure setup / no data movement required

## Data Visualization

Verilerinizi görselleştirmek, verilerinizi aynı anda birden çok boyutta keşfetmenin en etkili yollarından biridir. Keşfetmekte olduğunuz verilere ve aradığınız ilişkilere bağlı olarak, kullandığınız görselleştirmeler farklı olabilir. Bu konu başlığı altında kullanacağınız bazı görselleştirmelere ve çalışacağınız araçlara bir göz atalım.

### Popular Python Data Analysis and Visualization Tools

Bu araç kutusuyla hızlı bir tanışma yapalım. Daha önce Python kullanarak verileri analiz etme, dönüştürme ve görselleştirme yaptıysanız, bu araçlar tanıdık görünecektir. Pandas ve NumPy, popüler açık kaynaklı Python kütüphaneleridir. Pandas, veri analizi ve veri manipülasyonu için kullanılırken, NumPy ise Python'da bilimsel hesaplamalar yapmak için kullanılır. Benzer şekilde, matplotlib ve Seaborn, görselleştirmeler oluşturmak için popüler Python kütüphaneleridir. Matplotlib, statik, animasyonlu ve etkileşimli görselleştirmeler oluşturmada yardımcı olur. Seaborn ise matplotlib'e dayanır ve istatistiksel veri görselleştirmeleri ekler.



`pip install pandas`



`pip install numpy`



`pip install matplotlib`



`pip install seaborn`

### How many reviews are in each sentiment class?

Bu araçları ürün değerlendirmelerini görselleştirmek için kullanacaksınız. Veri kümesini keşfetmek ve görselleştirmek amacıyla, basit şeylerden başlayarak, model eğitimi için kaç örnek veriye sahip olduğunuzu anlamak gibi, daha karmaşık iş sorularına cevap vermek gibi farklı hedefleriniz olabilir. İlkine başlayalım. Veri kümesinin kaç örnek içerdigini anlamak için, her duygusal sınıfın incelemelerin sayısını döndürecek bu eşit sorguyu Amazon Athena kullanarak çalıştırıralım. Sorgu sonucu, çubuk grafiğinde en uygun şekilde görselleştirilecektir. İşte görselleştirme kodu. Matplotlib kütüphanesini içe aktarıyorum. Basit

bir çubuk grafiği için, sorgu sonuçlarını içeren pandas veri çerçevesini kullanabilir ve ardından doğrudan plot bar fonksiyonunu çağırabilirsiniz. X ve Y eksen verileri olarak kullanılacak veri çerçevesi sütunlarını tanımlar ve çıktıya başlık gibi ek veriler ekleyebilirsiniz.

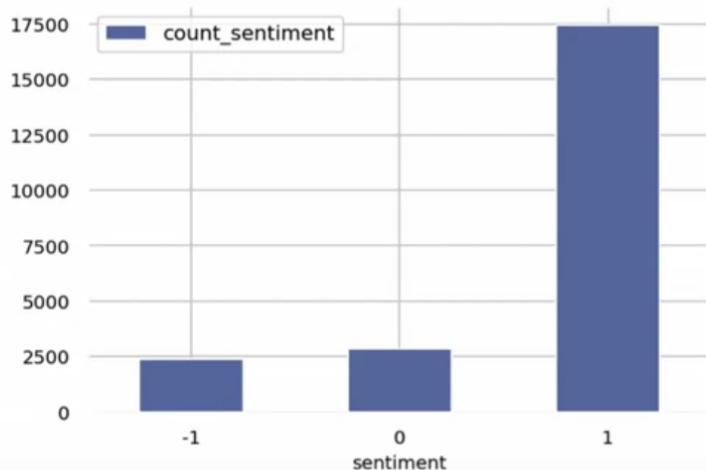
```
SELECT sentiment, COUNT(*) AS count_sentiment  
FROM dsoaws_deep_learning.reviews  
GROUP BY sentiment  
ORDER BY sentiment DESC, count_sentiment
```

SQL Query

```
import matplotlib.pyplot as plt  
chart = df.plot.bar(  
    x="sentiment",  
    y="count_sentiment")  
  
plt.xlabel("sentiment")  
plt.show(chart)
```

Python visualization code

Ve işiniz bittiğinde, show işlevini çağırabilirsiniz ve burada sonucu örnek bir çubuk grafiğinde (bar chart) görebilirsiniz. Çubuk grafiği aynı zamanda pozitif duygusal sınıfın (burada 1 numaralı) diğer sınıflardan çok daha fazla örnek içerdığını görmeyi kolaylaştırır. Temelde, duygusal sınıflar arasında dengesiz bir örnek dağılımınız bulunmaktadır. Bu durumu daha sonra nasıl ele alacağımızı göreceğiz.



## What is the distribution of reviews lengths? (*number of words*)

İşte başka bir ilginç görselleştirme. Bu sefer, yüzdelik dilimleri hesaplamayı ve veri dağılımlarını görselleştirmeyi göstereceğim. Bu amaçla, inceleme uzunluğunun veya inceleme başına kelime sayısının dağılımını hesaplayacağım. Buradaki SQL sorgusu oldukça basittir. İnceleme metnini "review body" sütununda boşluk karakterine göre bölerim, bu da bana bireysel kelimelerin bir listesini verir ve ardından kardinaliteyi hesaplarım, bu da bana kelime sayısını verir. Dağılımı çizmeden önce yüzdelik dilimleri hesaplamak istiyorum. İnceleme uzunluğunu içeren "panels" veri çerçevesi üzerinde belirtilen yüzdelik dilimleri hesaplamak için "describe" işlevini kullanabilirsiniz. İnceleme uzunluğunun dağılımını görselleştirmek için bir histogram seçiyorum. Histogramlar frekans dağılımlarını temsil eder. Her farklı değerin ne sıklıkta meydana geldiğini gösterirler. Bu durumda, inceleme uzunluğunu 100 aralığa bölmek istiyorum ve 100. yüzdelik dilime kırmızı bir işaretçi ekliyorum.

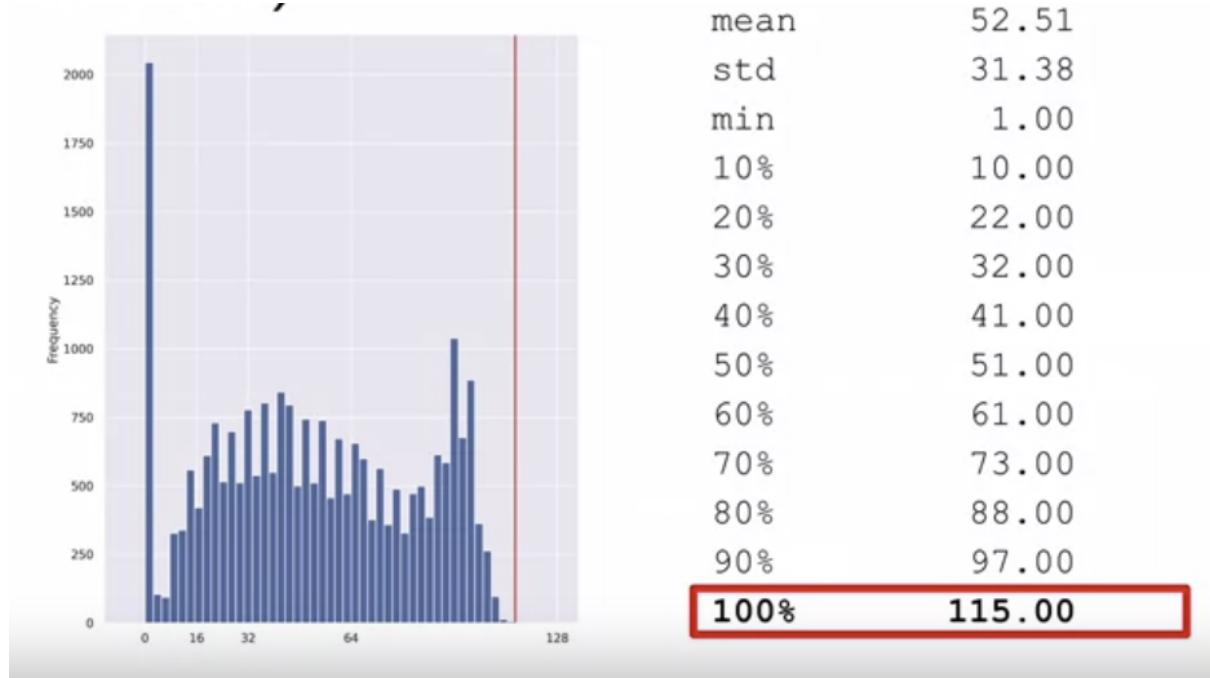
```
SELECT CARDINALITY(SPLIT(review_body, ' ')) AS num_words  
FROM dsoaws_deep_learning.reviews
```

SQL Query

```
summary = df["num_words"].describe()  
percentiles=[0.10, 0.20, 0.30, 0.40, 0.50, 0.60, 0.70, 0.80, 0.90, 1.00])  
  
df["num_words"].plot.hist(  
    xticks=[0, 16, 32, 64, 128, 256], bins=100,  
    range=[0, 256]).axvline(x=summary["100%"], c="red")
```

Python visualization code

Şimdi sonucumuza bir göz atalım. Sağ tarafta, yüzdelik dilim hesaplamalarının örnek sonuçlarını görebilirsiniz. En kısa incelemenin sadece bir kelime olduğunu görebilirsiniz ve eğer hatırlarsanız, tüm inceleme uzunluğunu 100 aralığa bölmüştüm ve burada mavi çubuklarla temsil edilen histogramda daha az aralık belirtirseniz, burada daha az çubuk görünür. X eksenin inceleme uzunluğunu gösterirken, Y eksenin ise frekanslarını temsil eder. Kırmızı renkte vurgulanan 100. yüzdelik dilime baktığınızda, tüm incelemelerin bu örnekte 115 kelime veya daha az olduğunu görebilirsiniz.



Bu bilgi özellikle metin sınıflandırma modeli oluştururken bize çok yardımcı olacaktır.

### Additional reading material

- <https://github.com/aws/aws-sdk-pandas>
- <https://aws.amazon.com/tr/glue/>
- <https://aws.amazon.com/tr/athena/>
- <https://matplotlib.org/>
- <https://seaborn.pydata.org/>
- <https://pandas.pydata.org/>
- <https://numpy.org/>

## QUIZ

1. Suppose you want to use Machine Learning to classify sentiments of product reviews. The model inputs will be the product reviews text and the output the predicted sentiment, positive or negative. In a typical ML workflow there are 3 main steps:

- I. Train and tune the machine learning model
- II. Deploy and monitor the trained model
- III. Prepare the data

What is the correct ordering of these steps?

- III, I, II
- I, III, II
- III, II, I
- I, II, III

2. Data lakes are enterprise storage solutions which can host virtually any amount of data. Are there any restrictions on the data types a data lake can store?

- Yes, data lakes only support unstructured and semi-structured data types.
- Yes, data lakes only support structured data.
- Yes, data lakes only support unstructured data.
- No, data lakes support all data types including structured relational data, semi-structured data, and unstructured data.

3. Amazon Simple Storage Service (Amazon S3) is a public cloud object storage service that allows users to store and retrieve any amount of data at any time.

Select the tool that you would use to register and discover data in Amazon S3.

- AWS Data Wrangler
- Amazon Athena
- Amazon Redshift
- AWS Glue

4. Amazon Athena provides great flexibility to run queries without adding any complexity to your project. Moreover, it is a very fast service and your queries return results in a matter of seconds, even on large datasets. Which of the following facts is **NOT true** about Amazon Athena?

- No infrastructure is needed to set up Amazon Athena.
- Amazon Athena does not require the AWS Glue Data Catalog to register and query S3 data.
- It is an interactive query service.
- It can be used to analyze data in Amazon S3 using standard SQL.

5. Within the AWS ecosystem, Data Wrangler is an agile service to load and unload data from data lakes and databases. What are other capabilities of this service? (Choose all that apply.)

It extends the power of the pandas library to AWS.

 **Correct**

Great Job. You can read more about AWS Data Wrangler [here ↗](#).

It connects pandas dataframes and other AWS services.

 **Correct**

Correct! You can read more about AWS Data Wrangler [here ↗](#).

It can be used for loading and unloading data from data lakes and databases.

 **Correct**

That's right! You can read more about AWS Data Wrangler [here ↗](#).

It extends the power of the NumPy library to AWS.

6. You work as a Machine Learning engineer in a company and are asked to develop algorithms to solve 3 tasks:

**Task 1:** You have a large dataset of unstructured text information. You are asked to convert/summarize this information into reports.

**Task 2:** A business has experienced a huge surge in the number of customers. To improve customer support and engagement, you are asked to build a chatbot.

**Task 3:** To simplify paperwork and time, you are asked to automate an employee expense system by building an image scanning system for expense receipts.

Which of these will you treat as Natural Language Processing (NLP) problems?

Task 1 only

Tasks 1 and 3

Tasks 1 and 2

All 3 tasks