DAĞITIK MAKİNE ÖĞRENMESİ PLATFORMLARI

DR.ÖĞR.ÜYESI BETÜL AY

MAKİNE ÖĞRENMESİ

- Makine öğrenmesi, çeşitli veri kümelerinin işlenmesi ve analizinin gerçekleşmesi için gerekli algoritmaları, yöntemleri, geliştirme süreçlerini inceleyen bir bilgisayar bilimidir.
- Makine öğrenmesinin temel amacı geçmiş deneyimleri kullanarak gelecek kararları vermektir.
- Bilgisayar bilimcisi Tom Mitchell, makine öğrenmesini şu şekilde tanımlamaktadır:

  "Eğer bir bilgisayar programı, bir T görevini, P performansında yaparak, deneyiminde E artış sağlıyorsa, o bilgisayar programı, T görevinde, P performansıyla, E deneyimlerinden öğreniyordur denilir".
- Makine öğrenmesinin basit bir örneği doküman filtrelemedir. Spor ya da ekonomi konularını içeren binlerce metin belgeleri gözlemlenerek, doküman filtreleme yeni metin belgelerini sınıflandırmayı öğrenir.

  Bir T görevinde, dokümanlar ekonomi ve spor konularına göre ayrılarak sınıflandırılır. Bir P performansında bilgisayar programı, sınıflandırılmış dokümanları bu görevde çalıştırır ve E deneyimlerinde sınıflandırılmış dokümanların doğruluk yüzdelerini hesaplayarak performansını değerlendirir.

BÜYÜK VERİ İLE MAKİNE ÖĞRENMESİ ZORLUKLARI

- Tek bir makine üzerinde çalışan geleneksel makine öğrenmesi genellikle küçük ve kontrollü veri setleri kullanmaktadır.
- Öğrenme algoritmalarının eğitim veri boyutu arttıkça sınıflandırma ve kümeleme algoritmalarından elde edilen doğruluk oranı büyük ölçüde artar.
- Fakat, tek bir makine küçük veri kümeleri ile çalışır ve yüksek hesaplama gücü, depolama kapasitesi ve ağ trafiği gerektiren daha geniş veri kümeleri için yüksek performans sunamaz.
- Büyük verilerin işlenmesi ve analizinde var olan makine öğrenmesi algoritmalarının uygulanması oldukça zordur ve bu algoritmaların tek bir makine üzerinde uygulaması günlerce hatta haftalarca sürmesi zaman ve maliyet açısından dezavantaj oluşturmaktadır.

BÜYÜK VERİ İLE MAKİNE ÖĞRENMESİ ZORLUKLARI

- Tek bir makine üzerinde çalışan geleneksel makine öğrenmesi genellikle küçük ve kontrollü veri setleri kullanmaktadır.
- Öğrenme algoritmalarının eğitim veri boyutu arttıkça sınıflandırma ve kümeleme algoritmalarından elde edilen doğruluk oranı büyük ölçüde artar.
- Fakat, tek bir makine küçük veri kümeleri ile çalışır ve yüksek hesaplama gücü, depolama kapasitesi ve ağ trafiği gerektiren daha geniş veri kümeleri için yüksek performans sunamaz.
- Büyük verilerin işlenmesi ve analizinde var olan makine öğrenmesi algoritmalarının uygulanması oldukça zordur ve bu algoritmaların tek bir makine üzerinde uygulaması günlerce hatta haftalarca sürmesi zaman ve maliyet açısından dezavantaj oluşturmaktadır.

BÜYÜK VERİ İLE MAKİNE ÖĞRENMESİ ZORLUKLARI

- Tek bir merkezi işlem birimi üzerinde büyük veriyi işleme aşağıdaki nedenlerden dolayı etkisiz bir yöntemdir ve internet üzerinde arama motorları, e-mail, sosyal ağlar ve diğer servislerin yaygınlaşması ile ilgi odağı haline gelen veri koruma (data privacy) hususunda veri güvenliğini risk altına sokmaktadır:
  1. Tek bir merkezi veri tabanında, veri setinin depolama maliyeti, veri setinin daha küçük parçalara bölünerek elde edilen küçük veri setlerinin depolama maliyetinin toplamından daha büyüktür.
  2. Merkezi bir veritabanının işlenmesi için gerekli hesaplama maliyeti, verilerin daha küçük parçalar halinde analiz edilerek hesaplama maliyetlerinin toplamından daha büyüktür.
  3. Ağ üzerinden çok büyük boyutlara sahip verilerin transferi çok fazla zaman alır ve finansal maliyet gerektirir.
  4. Tıbbi ve finansal kayıtlar gibi özel verilerin korunması son derece önemlidir.

BÜYÜK VERİ İLE VERİ MADENCİLİĞİ SORUNLARI

Büyük veri ile veri madenciliği sorunları veri, model ve sistem seviyesinde aşağıda verilmiştir:

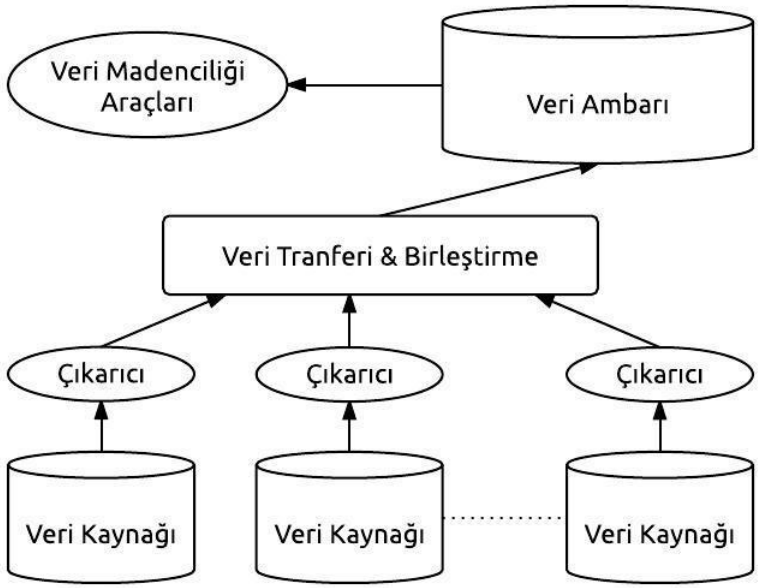
- Veri seviyesinde, farklı lokasyonlarda depolanan büyük veri sıklıkla heterojen, belirsiz ve eksik veri içerir. Bu yüzden, güvenli veri koruma ve bilgi paylaşım protokolleri geliştirmek büyük bir sorundur.
- Model seviyesinde, farklı veri kaynaklarından global modeller üretmek anahtar sorundur. Bu durum, dağıtık veri kaynakları arasında model ilişkileri için tasarlanan algoritmaların dikkatlice analiz edilmesini gerektirmektedir.
- Sistem seviyesinde, büyük veri madenciliği dayanıklılık (robustness) ve ölçeklenebilirliği sağlamak amacıyla veri kaynakları ve modeller arasında bazı önemli ilişkileri dikkate almasını gerektirir.

DAĞITIK VERİ ANALİZİ

- Tek bir merkezi sistemde, farklı lokasyonlardan toplanan dağıtık büyük verinin işlenmesi çok büyük hesaplama gücü ve depolama maliyeti gerektirmektedir.
- Diğer bir açıdan, veri setleri daha küçük veri setlerine bölünüp çeşitli sunuculara dağıtıldığında, küçük veri setlerinin analiz maliyetlerinin toplamı merkezi bir veri setinin depolama ve hesaplama maliyetinden daha az olacaktır.
- Büyük dağıtık bilginin işlenmesi ölçeklenebilir makine öğrenmesi sorunlarını da birlikte getirir. Bu sorun, dağıtık makine öğrenmesi gibi ölçeklenebilir makine öğrenmesi çözümlerine talebi arttırmaktadır.
- Dağıtık makine öğrenmesi, dağıtık veriden etkili bir şekilde yararlı bilgi çıkarımını hedeflemektedir.

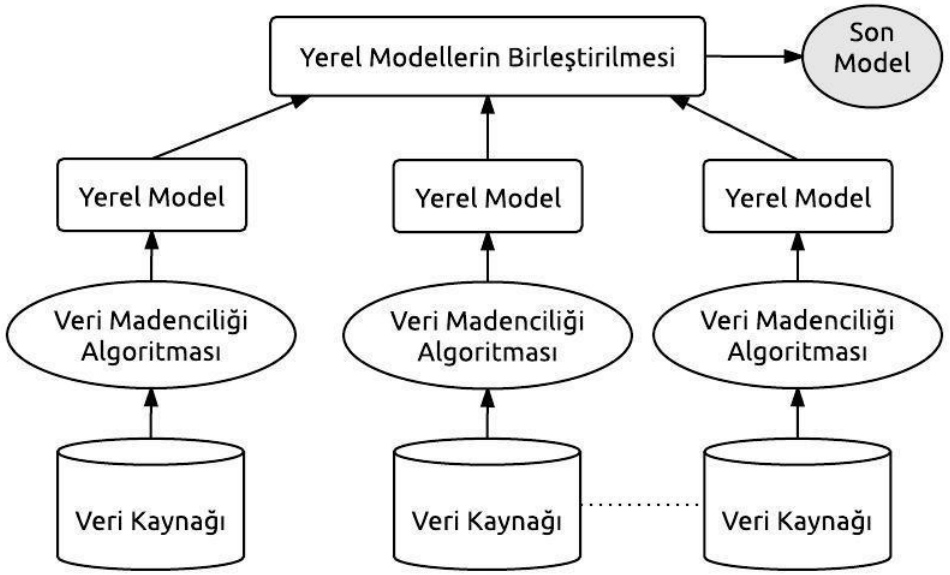
GELENEKSEL VERİ AMBARI TABANLI MİMARİ

- Her bir veri kaynağından çıkarılan veriler bir sonraki merkezi veri madenciliği uygulaması için veri ambarına transfer edilmektedir.
- Bu merkezi yaklaşım, dağıtık kaynakların kullanım eksikliği, gizlilik sorunları ve uzun tepki süresi nedeniyle veri madenciliği algoritmalarını çoğu dağıtık veri madenciliği uygulaması için kullanışsız hale getirir.



DAĞITIK VERİ MADENCİLİĞİ MİMARİSİ

- Seçilen veri madenciliği uygulaması her bir veri kaynağına uygulanır ve elde edilen yerel modeller toplanarak son model elde edilir.
- Dağıtık veri madenciliğinin amacı, hesaplama, depolama ve iletişim yetenekleri gibi özelliklere göre dağıtık veri kaynakları üzerinde veri madenciliği algoritmalarını gerçekleştirmektir.
- Özellikle uygulama daha geniş sayıda veri kaynağı gerektirdiğinde bu yaklaşım zaman ile değişen veriyi analiz etmek için daha ölçeklenebilir ve pratiktir.

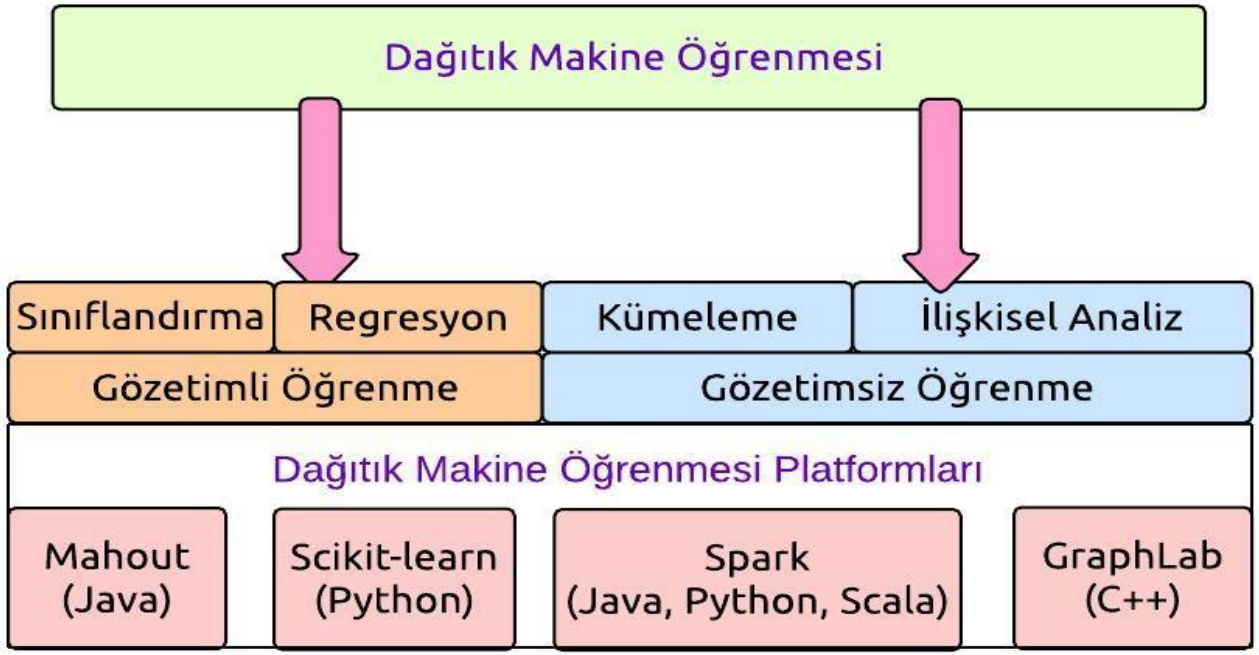


DAĞITIK VERİ MADENCİLİĞİ/MAKİNE ÖĞRENMESİ PRENSİPLERİ

Çoğu dağıtık makine öğrenmesi aşağıdaki temel prensip üzerine çalışmaktadır:

- Her bir veri kaynağına belirlenen merkezi bir algoritma uygulanır.
- Her alt veri kaynağı merkezi algoritma sonuçlarını global veri kaynağına gönderir.
- Bütün yerel modeller toplanarak global bir model elde edilir.
- Ana veri kaynağı global modeli her bir alt veri kaynağına gönderir ve her bir veri kaynağı global modele göre algoritma sonuçlarını günceller.

DAĞITIK MAKİNE ÖĞRENMESİ PLATFORMLARI (FRAMEWORKS)



MAHOUT

- Mahout, Apache Yazılım Lisansı altında lisanlanmıştır ve dağıtık bir makine öğrenmesi platformudur.
- Mahout, sınıflandırmadan ortak filtreleme (collaborative filtering) ve kümelemeye kadar pek çok uygulamayı çoklu makineler üzerinde paralel bir şekilde çalıştırmaktadır.
- Bu algoritmaları gerçekleştiren kullancı dostu RapidMiner ve Weka gibi pek çok makine öğrenmesi platformları vardır. Fakat, bu geleneksel platformlar hafıza yetersizliği nedeniyle büyük veriyi işleyememektedir.

MAHOUT

- Mahout' un amacı, böl-birleştir paradigması olarak adlandırılan MapReduce ile çalışan Hadoop sistemi üzerinde kullanılan ölçeklenebilir makine öğrenmesi kütüphanelerini barındıran bir platform sunmaktır.
- Bu şekilde geniş veri kümelerini işlemek için tüm görev, alt görevlere bölünür ve alt kümeler tekrardan birleştirilir.
- Mahout kütüphanesi aynı zamanda, Hadoop sistemi haricinde MapReduce paradigmasını kullanan diğer işleme sistemlerinde de kullanılabilir.

SCIKIT-LEARN

- Phyton dilinde yazılmış ve BSD licence altında lisanslanmış Scikit-learn, açık kaynaklı makine öğrenmesi kütüphanelerinin bir kümesidir ve aynı zamanda popüler bir akademik araştırma alanıdır.
- İyi dökümante edilmiş, kullanımı kolay ve fonksiyonel API sağlayan Scikit-learn [39], sınıflandırma, regresyon, boyutsal azaltma ve kümeleme gibi pek çok algoritmayı gerçekleştirebilmektedir
- Scikit-learn, popüler dağıtık makine öğrenmesi algoritmalarının uygulaması için NumPy ve SciPy Python modüllerini içerir

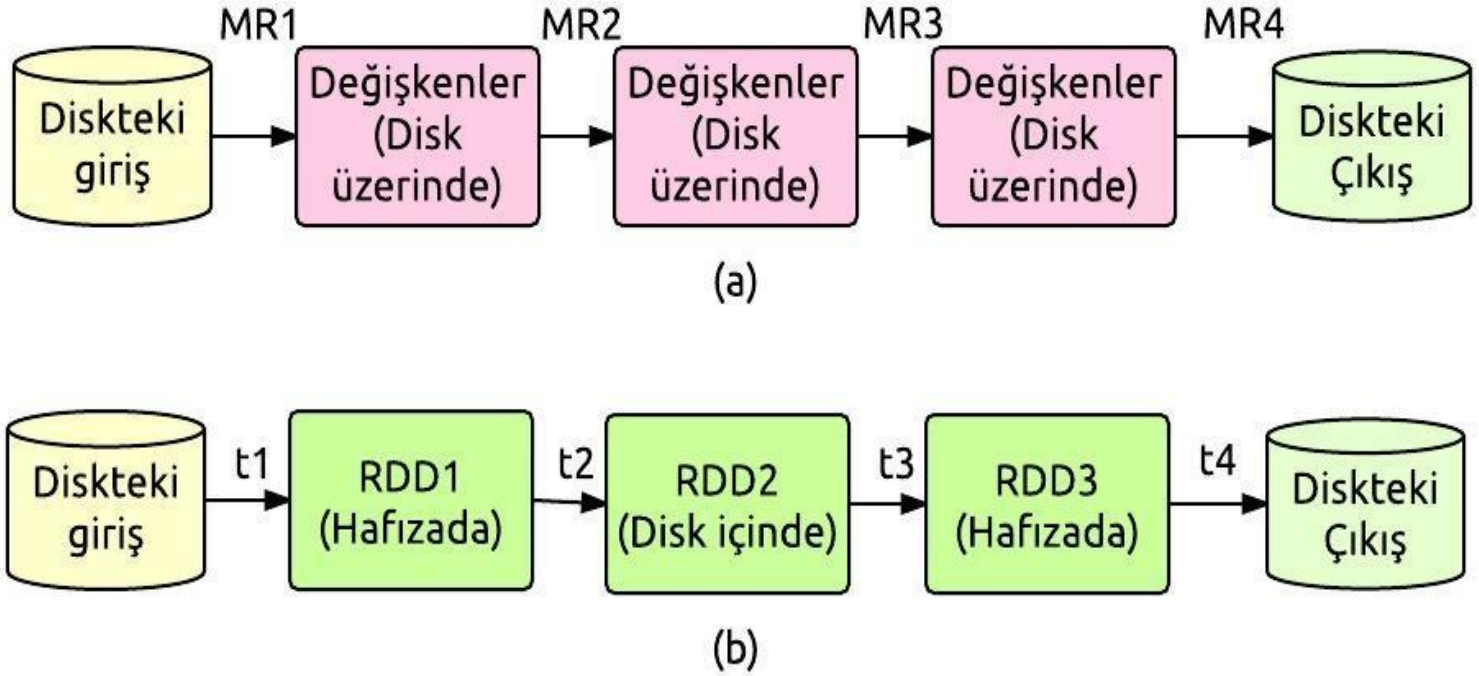
SPARK

- Mahout gibi Apache Yazılım Lisansı altında lisanlan ve Berkeley AMPLab' da geliştirilen hafıza tabanlı dağıtık makine öğrenmesi platformudur.
- Hadoop' a benzer dağıtık veri ile çalışmak için Java, Pyton and Scala dillerinde yazılmış yüksek seviyeli API' ler sağlamaktadır ve Apache Hadoop için sunulmuş tek hafıza işleme çözümüdür
- Spark 4 küme modunda çalışmayı destekler; Standalone, Amazon EC2, Apache Mesos ve Hadoop Yarn. Bunun haricinde istenirse, Spark kümesi test amaçlı lokal modunda tek bir sunucu üzerinde de çalıştırılabilir.

SPARK

- Standalone deploy mode, Spark bir deploy script kümesini kullanarak özel bir küme üzerinde çalıştırılabilir. Ek olarak, bütün Spark işlemleri Standalone lokal modda aynı Java Sanal Makinesi (JVM) işleminde çalıştırılabilir.
- Amazon EC2, Spark kümeleri Amazon EC2 üzerinde başlatılabilir, yönetilebilir ve kapatılabilir ve kullanıcının kümesi üzerinde Spark, Shark ve HDFS kurulumu otomatik olarak gerçekleştirilir.
- Apache Mesos, Spark ve diğer platformlar arasında dinamik olarak kaynak paylaşımı sağlar. Kolay ve etkili açık kaynak küme platformudur.
- Hadoop Yarn, Uygulama master' ında çalışmak için Spark sürücülerine izin verir ve genellikle Hadoop 2 olarak adlandırılır.

MAHOUT VE SPARK



SPARK

- Dağıtık bir hesaplama platformu olan Spark, RDD içeren programlama kümesi için veri katmanı kullanarak var olan küme hesaplama platformlarından daha güçlü ve kullanışlı iterative hesaplamalara sahiptir
- Sparkın son sürümleri ile birlikte, bir veritabanı (3 yıl önce geliştirilen bir proje Shark SQL yerine Spark SQL), bir makine öğrenmesi kütüphanesi (MLLib) ve bir graf motoru (GraphX) gibi zengin araçlar sunulmaktadır.

GRAPHLAB

- C++ dilinde yazılmış ve Apache Yazılım Lisansı altında lisanlanmış GraphLab [58], graf tabanlı dağıtık bir makine öğrenmesi platformudur.
- Büyük veri üzerinde iterative hesaplamalar asekron olarak daha yüksek performans sergilemektedir. Sekron hesaplamada bir işlem ilgili hesaplamalarını tamamlayana kadar diğer işlemleri beklediğinde, tüm çalışma zamanı bu gecikmeler ile birlikte artmaktadır. Bu problemin üstesinden gelmek ve pek çok iterative makine öğrenme algoritmalarını daha kısa sürede uygulamak için, GraphLab asekron ve graf-paralel hesaplama sunmaktadır.

PLATFORMLARIN KARŞILAŞTIRILMASI

	Mahout	Scikit-Learn	Spark	GraphLab
Kütüphane	Java	Phyton	Java, Pyton, Scala	C++
Model üzerinde iterative hesaplama	Sekron	Sekron	Sekron	Sekron & Asekron
Hata Toleransı	Görevler Master tarafından tekrar başlatılır.	Hata toleransına odaklanmamıştır	Kalan görevler otomatik olarak tekrardan yapılandırılır.	Hata toleransı yoktur fakat dağıtık GraphLab' da, kayıp veri son chechpoint ile tekrardan iyileştirilir.
Avantajlar	Populer makine öğrenmesi algoritmaları için kullanımı kolay ve etkili bir platformdur	Büyük veri kümleri için hızlı ve ölçeklenebilirdir, çapraz-doğrulama ve standart veri kümeleri sunar	RDD olarak adlandırılan yeni bir hafıza katmanı sunar, hızlı ve esnek iterative hesaplama sağlar.	Uygun tutarlılık seçenekleri ile her paralel çalışma için sıralı tutarlılığı (sequential consistency) garantiler, asekron ve graf paralel hesaplama sağlar
Dezavantajlar	Doğru kümeleme sonuçları için iş bellek yapılandırma parametrelerini ayarlamak için dikkat gerektirir.	İstatistiklere daha az odaklanmıştır, araçlar ve kütüphaneler dağınıktır, komut satırlı arayüze sahiptir	Metin dosyaları yavaşça kaydedilir, çoklu bilgisayarlar arasında dağıtık veriyi takip etmek zordur, Spark üzerinde Pig kullanmak kullanışsız bir yöntemdir	Doğal grafiklerin bölünmesi için verimsiz ve kötü performans sergiler
Önerilen Algoritmalar	Naïve Bayes, HMM, logistic regression, random forest, k-Means, fuzzy k-Means, canopy, spectral clustering, latent drichlet allocation, collaborative filtering	SVM, nearest neighbors, random forest, SVR, ridge regression, k-Means, spectral clustering, PCA, future selection, grid search, preprocessing	linear SVM and logistic regression, classification and regression tree, k means clustering, singular value decomposition, inear regression Naive Bayes, basic statistics, feature transformations	collaborative filtering, k-Means, PageRank, latent drichlet allocation, Jacobi method for linear solver,distributed dual decomposition, Image-Stitching, graph analytics