BULUT ÜZERİNDE DAĞITIK DOKÜMAN SINIFLANDIRMA VE KÜMELEME
YÜKSEK LİSANS TEZİ

Selen GÜRBÜZ

Bilgisayar Mühendisliği Anabilim Dalı
Danışman: Yrd. Doç. Dr. Galip AYDIN

ARALIK-2015

T.C.
FIRAT ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

BULUT ÜZERİNDE DAĞITIK DOKÜMAN SINIFLANDIRMA VE KÜMELEME

YÜKSEK LİSANS TEZİ

Selen GÜRBÜZ

121129105

Tezin Enstitüye Verildiği Tarih : 08.12.2015
Tezin Savunulduğu Tarih : 28.12.2015

Tez Danışmanı : Yrd. Doç. Dr. Galip AYDIN
Diğer Jüri Üyeleri : Prof. Dr. Ali KARCI
Yrd. Doç. Dr. Taner TUNCER

ARALIK-2015

ÖNSÖZ

Dokümanları sınıflandırmada amaç, bir dokümanın özelliklerine bakılarak önceden belirlenmiş belli sayıdaki kategorilerden hangisine dahil olacağını belirlemektir. Doküman kümeleme algoritmaları da bir doküman grubu içerisinde yer alan benzer dokümanların bulunmasına çalışılmasıdır.

Bu tezde metin madenciliği işlemlerinden metin sınıflandırma ve metin kümeleme yöntemleri incelenmiş ve bunlar dağıtık sunucular üzerinde çalışan MapReduce teknolojisi yardımıyla büyük sayıda doküman üzerinde uygulanmıştır. Amacımız pahalı olmayan sunuculardan oluşturulacak bir küme (cluster) üzerinde kurulacak dağıtık sistemde koşturulacak doküman kümeleme algoritmalarını kullanarak büyük sayıda Türkçe dokümanı analiz edebilmektir. Türkçe dilinde dağıtık doküman analizi çalışmalarının azlığı sebebiyle bu tez çalışmasının literatüre katkı sağlayacağı düşünülmektedir.

Özellikle veri miktarının büyümesiyle zorlaşan doküman analizi problemine açık kaynaklı yazılımlar kullanılarak bir çözüm geliştirilmiş ve Türkçe dokümanların analizi yapılarak da ulusal bilgi birikimine katkı sağlanmıştır.

Bu tez çalışması boyunca ilgi ve yardımlarını esirgemeyen danışman hocam, **Yrd. Doç. Dr. Galip AYDIN**'a teşekkür ve şükranlarımı sunarım. Ayrıca hayatım boyunca bana destek olan aileme çok teşekkür ederim.

Selen GÜRBÜZ

ELAZIĞ-2015

İÇİNDEKİLER

ÖZET

Büyük veriler geleneksel sistemlerin işleyemeyeceği kadar yüksek hacimli ve karmaşık veriler olup bunların bilinen veri tabanı veya dosya sistemleri üzerinde saklanması ve geleneksel algoritmalarla işlenmesi zordur. Dolayısıyla büyük verileri işlemek ve saklamak için geleneksel hesaplama yöntemleri yerine yeni teknolojiler kullanılmaktadır. Farklı kaynaklar tarafından üretilen çok büyük sayıda dokümanın içerikleri ile ilgili otomatik çıkarsamaların yapılması, alan ve konularının belirlenmesi, özetlerinin çıkarılması veya örüntü keşfi gibi doküman analizi konuları bilim insanlarının çözmeye çalıştığı konulardır.

Bu tez çalışmasında Türkçe bilimsel makalelerden oluşan bir veri seti üzerinde çalıştırılan dağıtık sınıflandırma ve kümeleme algoritmaları ile Büyük Veri analizleri yapılmaya çalışılmıştır.

Dağıtık Makine Öğrenmesi algoritmaları çalıştırabilmek için Apache Mahout ve Apache Spark kullanılmıştır. Dağıtık doküman sınıflandırma ve kümeleme yapılabilmesi için gerekli olan sunucular Google Cloud, Amazon AWS ve Microsoft Azure bulut altyapıları üzerinde çalıştırılmıştır.

Anahtar Kelimeler: Büyük Veri, Dağıtık Hesaplama, Bulut Bilişim

SUMMARY

DISTRIBUTED DOCUMENT CLASSIFICATION AND CLUSTERING ON CLOUD

Big data is described as large and complex data sets which can not be stored and processed using traditional databases, file systems and algorithms.Therefore new technologies are being utilized to store and process these big data sets. Automatic content extraction, field and topic discovery, summarization or pattern recognition over very large document sets which are produced by various sources are the subjects of many current research.

In this thesis, Big Data analysis, namely distributed classification and clustering algorithms are applied to a large data sets consisting Turkish scientific articles, To be able to run distributed Machine Learning algorithms Apache Mahout and Apache Spark are used.

The servers needed for distributed classification and clustering algorithms are deployed on the Google Cloud, Amazon AWS and Microsoft Azure cloud computing infrastructures.

Keywords: Big Data, Distributed Computing, Cloud Computing

ŞEKİLLER LİSTESİ

TABLOLAR LİSTESİ

1. GİRİŞ

Bilişim dünyasındaki hızlı büyüme beraberinde her alanda üretilen verinin miktarının da çok hızlı bir şekilde artmasına sebep olmuştur. İnternetin yaygınlaşması ve Web 2.0 ile beraber sosyal medya ve kullanıcılar arasında paylaşımın artması, çevrimiçi yayın yapan gazete, dergi, blog gibi kaynaklar aracılığıyla yayınlanan bilgi miktarını daha önce görülmedik boyutlara çıkarmıştır. Benzer şekilde bilim dünyasında da önceki yıllarla karşılaştırılmayacak sayılarda yayınlar yapılmakta ve dokümanlar üretilmektedir.

Ancak veri miktarındaki artış bu verileri anlamlandırabilecek, ne ile ilgili olduğunu anlayabilecek, ait oldukları sınıfları bulabilecek algoritmaların geliştirilmesi mümkün olduğunda kıymetli olabilmektedir. İşlenemeyen ve bilgiye dönüştürülemeyen veriden faydalanılamaz.

Bilgisayar Bilimlerinin Veri Madenciliği, Makine Öğrenmesi gibi dalları veri analizi ile ilgili önemli yöntemler ve algoritmalar sunmaktadır [1]. Ancak bu algoritmalar geleneksel olarak çok büyük miktarda veriyi işlemek için tasarlanmamışlardır. Bir algoritmanın işleyebileceği veri miktarı donanımsal kaynakların kapasitesi ile yakından ilgilidir. Çok büyük analizler yapabilmek için işlem ve depolama kapasitesi yüksek bilgisayarlara ihtiyaç duyulmaktadır [2]. Ancak yüksek kapasiteli bilgisayarlar yüksek maliyetleri nedeniyle çok az sayıda bulunabilmekte ve bunlara araştırmacıların önemli bir kısmının erişim imkanı olamamaktadır [3].

Geleneksel hesaplama yöntemleriyle analiz edilemeyecek boyutlardaki verileri ifade etmek için Büyük Veri (Big Data) kavramı kullanılmaktadır [4]. Son yılların en popüler konularından birisi olan Büyük Veri, Bulut Bilişim, Paralel Hesaplama, MapReduce gibi teknolojileri kullanarak kolaylıkla bulunabilecek, pahalı olmayan bilgisayarlar üzerinde çok büyük verilerin analizine olanak sağlamaktadır [5].

Bulut Bilişim donanımsal ve yazılımsal kaynakların servis olarak sunularak geleneksel istemci sunucu mimarilerine göre çok daha yüksek ölçeklenebilirliğe ulaşılmasını sağlayan modern bir yaklaşımdır [6]. Bulut Bilişim sayesinde ortalama özellikli sunucular üzerinde çok sayıda sanal sunucu çalıştırılabilmekte yüksek kaynakların daha optimize kullanılması sağlanmaktadır [7].

Doküman analizinde çok kullanılan metin madenciliği metinden yüksek kaliteli bilginin çıkartımı işlemi olarak tanımlanmıştır. Metin madenciliği işlemleri arasında metin sınıflandırma, metin kümeleme, konu çıkartımı, duygu analizi, doküman özetleme ve ilişki keşfi gibi konular bulunmaktadır [8].

2. DAĞITIK SİSTEMLER

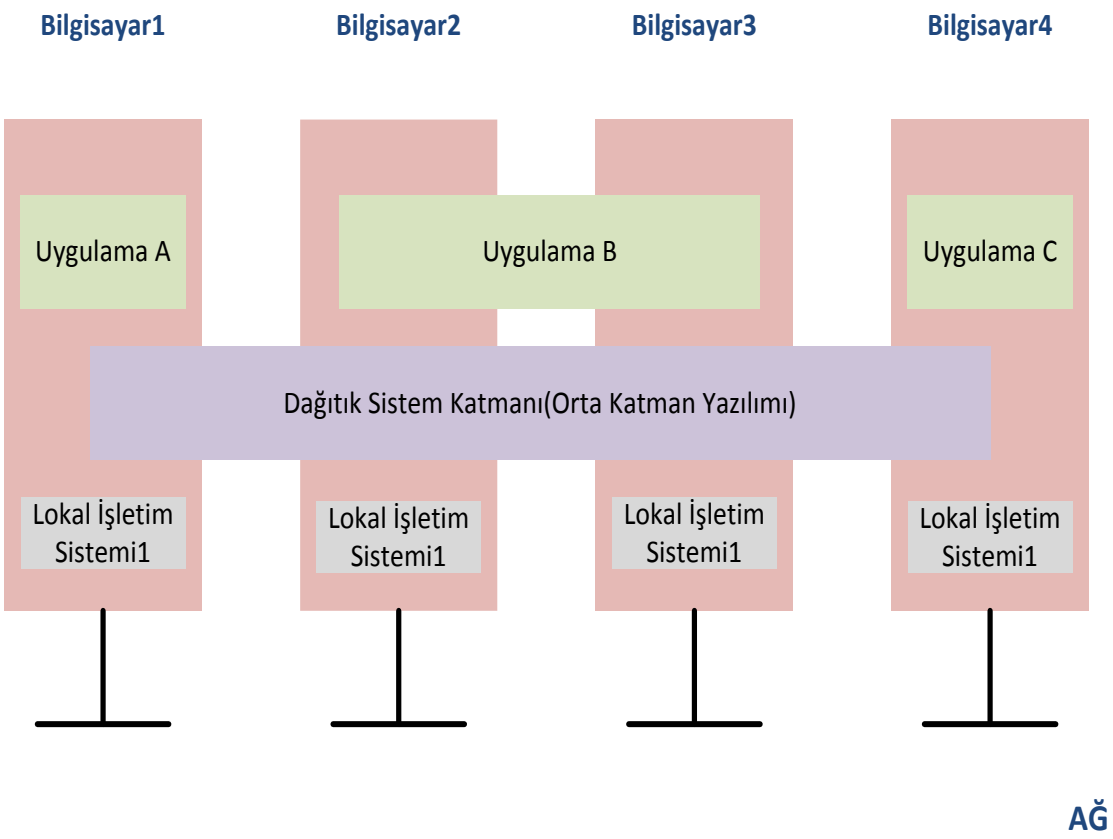
Dağıtık sistemlerin birçok tanımı bulunmasına rağmen en genel haliyle kullanıcılarına tek bir bilgisayarmış gibi görünen ancak birbirinden bağımsız bilgisayarlardan oluşan sistem olarak tanımlanabilir [9].

Dağıtık sistemlerin bazı temel özellikleri şunlardır: bu sistemler çeşitli bilgisayarların çoğunlukla kullanıcılardan gizli olarak nasıl haberleştikleri arasındaki ayrımı yapabilmektedir. Başka bir önemli özellik de kullanıcılar ve uygulamalar bir dağıtık sistem ile tutarlı ve değişmeyen bir yolla, nerede ve ne zaman olursa olsun karşılıklı etkileşimde bulunabilirler.

Prensipte, dağıtık sistemlerin geliştirilebilir ve ölçeklenebilir olması istenir. Bu karakteristik özellik, bağımsız bilgisayarlara sahip olmanın ve bu bilgisayarların bir bütün olarak sisteme dahil olmalarının kullanıcılardan gizlenmesinin doğrudan bir sonucudur. Bazı kısımları geçici olarak kullanım dışı olsa bile, dağıtık sistemler normal ve sürekli bir şekilde etkin durumdadır. Kullanıcılar ve uygulamalar sistemdeki değişim ve onarımları görmemeli ve daha fazla kullanıcı ve uygulamaya hizmet edebilmek için yeni kaynakların eklendiği gösterilmemelidir.

Tek sistem görünümünü sunmanın yanında, heterojen bilgisayar ve ağları desteklemek için, dağıtık sistemler genellikle bir yazılım katmanı vasıtasıyla düzenlenir. Bu yazılım mantıksal olarak kullanıcı ve uygulamalardan oluşan yüksek seviyeli bir katman ile işletim sistemleri ve orta katman yazılımı denilen temel iletişim hizmetleri içeren katmanlardan oluşur. İdeal bir dağıtık sistemin özellikleri arasında heterojenlik, ölçeklenebilirlik, güvenli olma ve hata toleransı bulunur.

.



Şekil 2.1. Dağıtık Sistem Mimarisi [9]

Ağ üzerinde birbiri ile bağlantılı dört bilgisayar ve üç uygulama için, uygulamaB Bilgisayar2 ve Bilgisayar3 arasında dağıtılmış olup her uygulama aynı arayüzü sunmaktadır. Donanım ve işletim sistemlerindeki farklılıklar her uygulamadan gizlenir [10].

2.1. OpenStack

Rackspace Cloud ve NASA tarafından geliştirilmiş olan ve bulut bilişim altyapısı olarak kullanılan OpenStack platformu ile Hadoop uygulamaları etkin bir şeklide yapılabilmektedir. Bulut bilişim teknolojilerinden olan veri depolama ve hesaplama gibi teknolojileri sunabilmesinin yanında açık kaynaklı olması büyük avantaj sağlamaktadır. OpenStack bir Hizmet olarak Altyapı (IaaS – Infrastructure as a Service) yazılımı olup kurulumu ile beraber aşağıdaki servisler sunulur [11].

- Compute(Nova)

Kendisine ait sanallaştırması olmadığından misafir sistem ara katmanı olan sanallaştırma yöneticisi (hypervisor) ile doğrudan konuşan servis Nova'dır. Kullanıcıdan gelen sanal makine (instance) taleplerini sanallaştırma yöneticisine iletir. Sanal makine için

gerekli olan tüm hafıza, disk, ağ tanımlamalarını gerçekleştirir. Oluştur, başlat, durdur gibi yönetimleri yapar.

- Ağ yönetimi (Quantum, Neutron)

Sanal makineye IP tahsisi yapılmasından sorumludur. Kullanıcıdan gelen IP istekleri ve filtreleme isteklerini gerçekleştirir. Büyük ölçekli ve yedekli bulut kurmaya imkan tanımaktadır.

- Kimlik Servisi (Keystone)

Kullanıcı ve sistem politikalarını merkezi olarak denetlemenin yanında kullanıcı-sistem etkileşimi, izin ve erişim yönetimi yapar. RESTFul API'den gelen doğrulama isteklerini karşılar. Sanal makinelere erişim için kullanılan PEM dosyalarının yönetimini yapar.

- İmaj Servisi (Glance)

Disk yönetiminin gerçekleşmesi ve sanal makine kopyasının alınması, yüklenmesi, gibi işlemler yapar. Arka planda OpenStack'ın Nesne Depolama (Object Storage) servisini kullanır. Sınırsız sayıda yedek alabilmesi ve programlanabilir olması büyük avantajdır.

- Nesne Depolama (Swift)

Herhangi bir sanal makine kurulumu gereksinimi olmaksızın nesne depolama görevini yapabilir. Büyük ölçekli ve yedeklidir ayrıca ölçeklenebilirdir. Veri depolama sistemini dağıtık olarak sağlar.

- Blok Depolama (Cinder)

Gerek sistemde var olan gerekse yeni açılmak üzere olan sanal makinelere ek disk alanı sağlar. Kullanıcılar arayüz üzerinden disk oluşturma, disk kaldırma gibi işlemleri gerçekleştirebilirler. Performansa ihtiyaç duyan servisler için bir RAW disk alanı sağlar.

- Kullanıcı Arayüzü (Horizon)

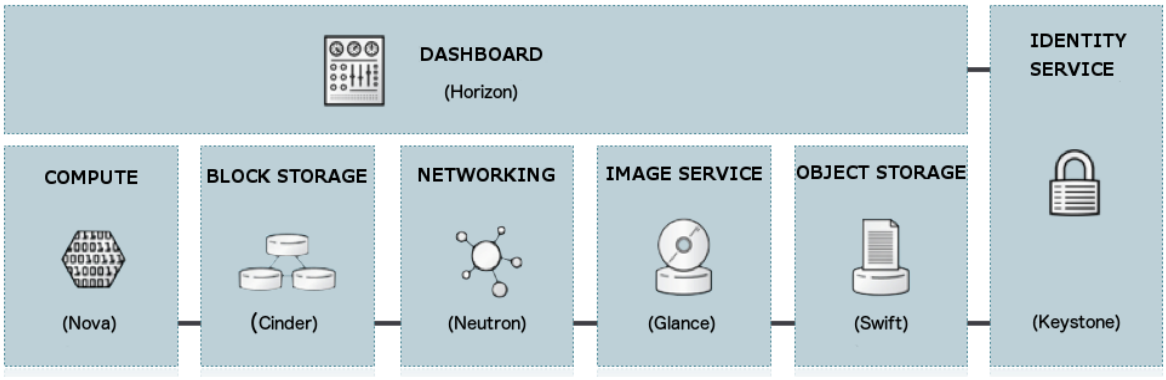
OpenStack'taki tüm aktif olan bileşenlerin yönetimini yapmaktadır. OpenStack'ın web arayüzü olup arayüzden API yönetimi ve kullanıcı yönetimi sağlanır. Sanal makine oluşturma, sonlandırma da bu servis ile yapılır.

- Orkestrasyon (Heat)

Otomatik büyüme ve ölçeklendirme için şablon tabanlı çalışmaktadır.

- Ölçme (Ceilometer)

OpenStack servislerini monitörleme ve ölçme işlemlerini gerçekleştirir.

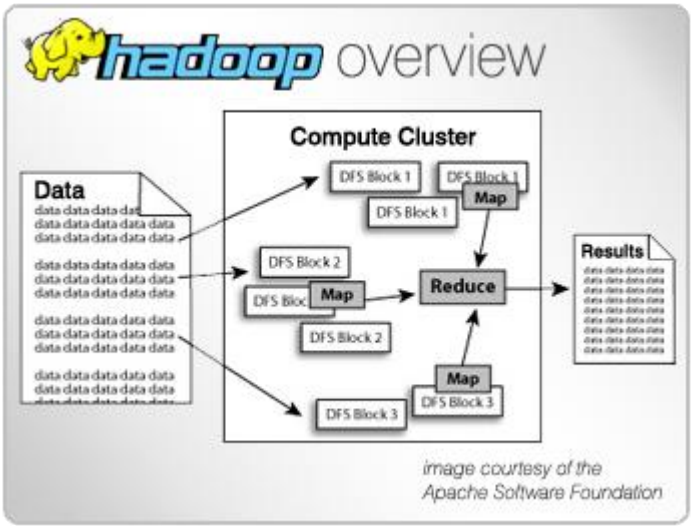


Şekil 2.2. OpenStack Servisleri [12]

2.2.Hadoop

Hadoop terabaytlarla ifade edilen ya da daha büyük verileri işlemek amacı ile oluşturulmuş bir kütüphanedir. Hadoop Distributed File System (HDFS) adı verilen dağıtık dosya sistemine sahip olup Java ile geliştirilmiştir. Hadoop birçok firma ve akademik grup tarafından büyük verileri analiz etmek amacı ile kullanılmaktadır.

Güvenilir, ölçeklenebilir ve veriyi düşük bütçeler ile işleyebilir olması Hadoop'u popüler kılmaktadır. Veriler analiz edilirken tek makine yerine birden fazla makineye dağıtıldığından işin tamamlanma süresi minimuma inmektedir. Hadoop'ta veriler kümelerde saklanır.



Şekil 2.3. Hadoop Verilerinin Kümede Saklanması [13]

Hadoop, tek bir sunucu üzerinde çalışabildiği gibi, kendi CPU ve hafıza birimi bulunan binlerce sunucusu olan bir küme üzerinde de çalışabilir.

Hadoop, Common, HDFS, YARN ve MapReduce olmak üzere dört modülden oluşur.

Tablo 2.1. Hadoop Modülleri

Common	Tüm Hadoop modüllerini destekleyen ortak modül
HDFS	Dağıtık dosya sistemi
YARN	Kaynak yönetimi ve iş zamanlaması yapan kütüphane
MapReduce	Dağıtık, paralel hesaplama kütüphanesi

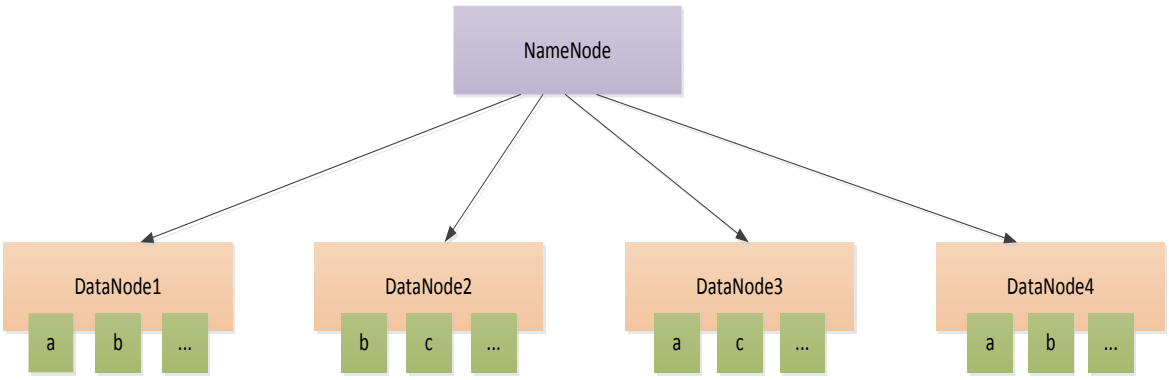
Hadoop, GFS (Google File System) tabanlı geliştirilmiş olan bir Hadoop Dağıtık Dosya Sistemine (HDFS) sahiptir [14]. HDFS dosyaların bir küme üzerinde dağıtılmalarından sorumludur.

Hadoop sisteminde veriler bloklara bölünür. Her bir bloğun büyüklüğü varsayılan olarak 64 MB olarak ayarlanmıştır. Farklı blok büyüklüğü dfs.block.size parametresiyle ayarlanabilir. Verilerin bloklara ayrılması 64 MB'lık veri büyüklüğü elde edilen satırdan sonra yeni bir blok oluşturarak gerçekleştirilir [15].

2.3. HDFS

Hadoop'ta uygulama yaparken kullanılan dosya sistemi HDFS (Hadoop File System) olarak adlandırılmıştır. Bir adet Namenode ve buna bağlı olan birden çok Datanode'dan oluşan HDFS'de master görevini Namenode yapmaktadır. Namenode tarafından gönderilen komutlar Datanode'lar tarafından alınır ve buna göre işlemler yapılır.

Dağıtık bir dosya sistemine sahip olan HDFS ile birden fazla sunucunun diski bir araya gelerek tek bir sanal disk oluşturulur. HDFS'de verilerin bir kopyası birden fazla düğümde kayıt altına alındığı için hata anında veri kaybı yaşama durumu olmamaktadır.



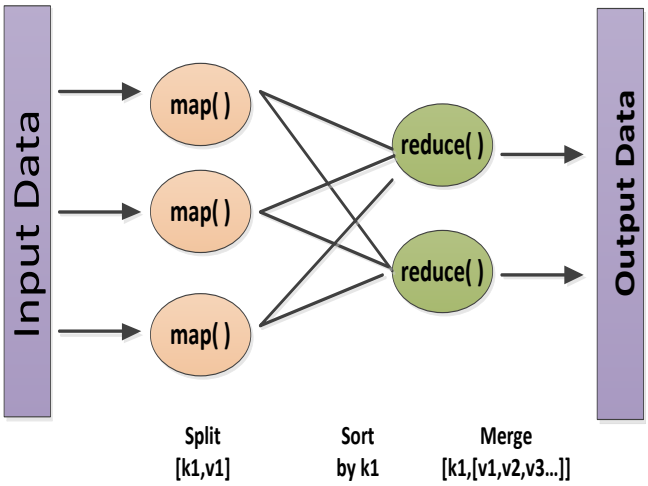
Şekil 2.4. HDFS'de Verilerin Saklanması

Şekilde görüldüğü üzere b verisi DataNode1, DataNode2 ve DataNode4 olmak üzere 3 ayrı düğüm üzerinde saklanmıştır.

2.4.MapReduce

MapReduce büyük verileri işlemek için geliştirilmiş dağıtık bir programlama modelidir [16]. Dağıtık bir sistemde veriler MapReduce ile analiz edilirken iki fonksiyon kullanılır. Bunlardan birincisi map fonksiyonu diğeri de reduce fonksiyonudur [17]. Kolay bir programlama çerçevesi vardır. Sadece map ve reduce fonksiyonlarını geliştirmemiz yeterlidir.

MapReduce modelinin ilk adımı elemanları bir anahtar ile eşleştirip bir sonraki işleme hazır hale getirir. İkinci adımda iteratör ile aynı anahtar için bir değerler listesi alır ve bunları biriktirip, filtreleyip, örnekleyip azaltır. Sonuçlarını da HDFS veya HBASE NoSQL veritabanına yazar [18].



Şekil 2.5. MapReduce [19]

2.5.Bulut Bilişim

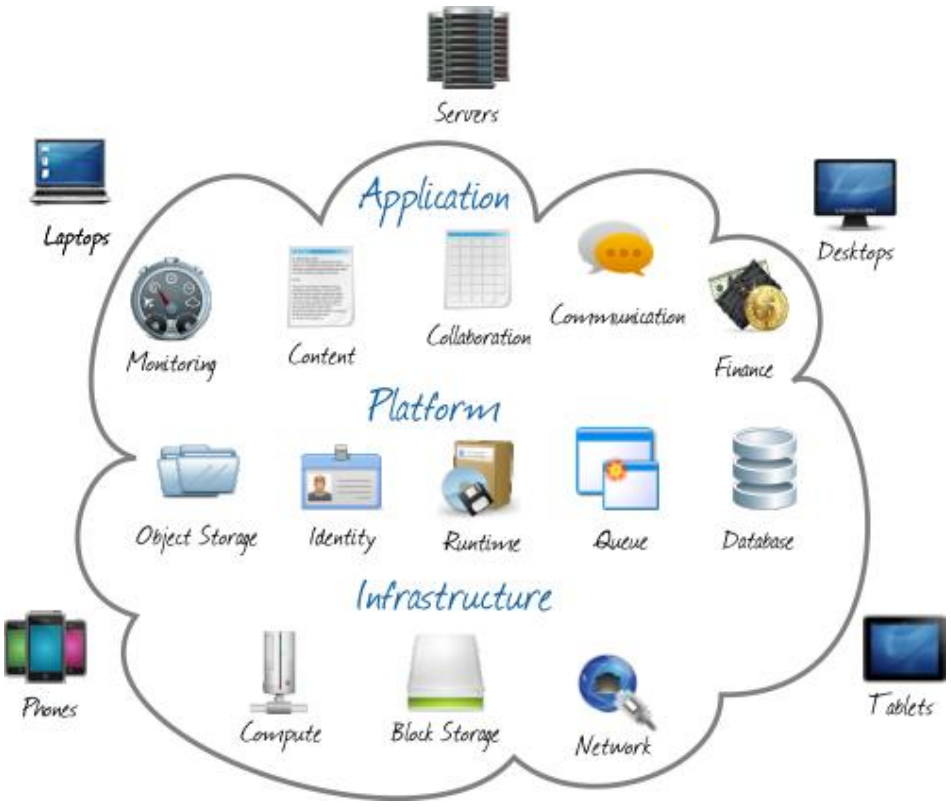
Kurulum gereksinimi olmadan her yerde çalışma desteği sunan bulut bilişim hizmeti, bilgiyi çevrimiçi olarak dağıtırken aynı zamanda gerekli yazılımları da paylaşarak mevcut olan hizmetlerin bir ağ üzerinden kullanılmasını sağlar. Elektronik posta bulut bilişim için tipik bir örnek olup evlerde veya işyerlerinde bir yapılandırma olmaksızın bu

hizmetten faydalanılmaktadır. Bunun dışında bulut depolama hizmetine örnek olarak Dropbox, GoogleDrive, iCloud verilebilir.

Kullandığın kadar öde mantığıyla çalışan çeşitli ticari bulutlar mevcut olup bunlara örnek olarak Google Cloud, Amazon EC2, GoGrid, FlexiScale verilebilir. Buna ek olarak, OpenStack, Eucalyptus, OpenNebula, ve Nimbus gibi açık kaynaklı bulut bilişim yazılımları kullanarak özel bulutlar oluşturmak mümkündür [20].

2.5.1.Bulut Servisleri

Bulut bilişim, servis olarak yazılım (SaaS - Software as a Service), servis olarak platform (PaaS - Platform as a Service) ve servis olarak altyapı (IaaS - Infrastructure as a Service) olmak üzere üç şekilde sunulmaktadır.



Şekil 2.6. Bulut Servisleri [21]

2.5.1.1.Servis Olarak Yazılım (SaaS - Software as a Service)

Servis olarak yazılım (SaaS) içerisindeki uygulamaların bir satıcı veya servis sağlayıcı tarafından barındırıldığı ve müşterilerine internet gibi tipik bir ağ üzerinden sunulan bir yazılım dağıtım modelidir. Servis olarak yazılım için, yazılımın servis olarak sunulması da denilebilir.

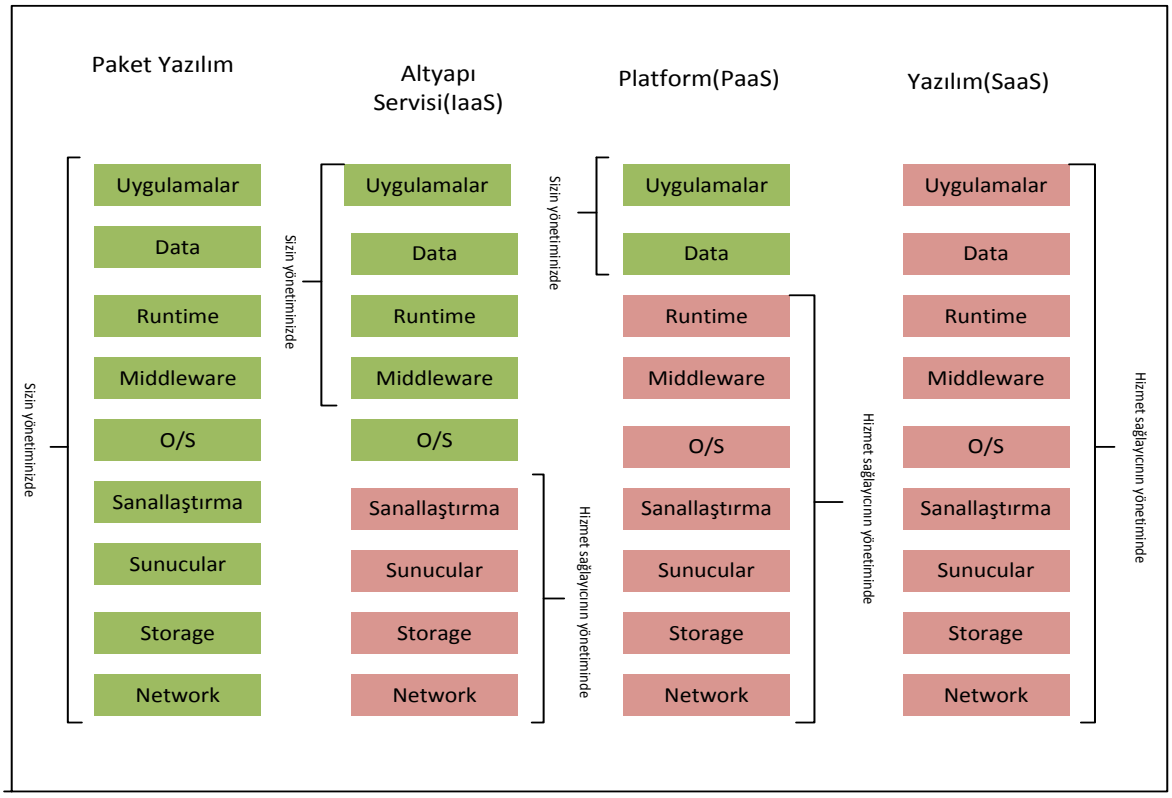
Bir yazılıma ihtiyaç duyulduğu zaman bunun altyapı ve lisanslaması için herhangi bir maliyet olmaksızın kullanım olanağı sağlamaktadır. Dolayısıyla donanım, kurulum, iletişim, güvenlik gibi maliyetlere ek ücret ödenmez.

2.5.1.2.Servis Olarak Platform (PaaS - Platform as a Service)

Servis olarak platform hizmeti kendi bir platform ile birlikte ihtiyaç duyulan yazılımın da kullanıcıya internet üzerinden sunulduğu bulut servisidir. Adından da anlaşılacağı üzere veri tabanları, işletim sistemleri gibi platformlar hazır olarak sunulur. Servis olarak platforma örnek olarak Amazon Elastic Beanstalk verilebilir**.**

2.5.1.3.Servis Olarak Altyapı (IaaS - Infrastructure as a Service)

Bulut bilişimin en temel hizmeti de denilebilecek olan servis olarak altyapı, bulut hizmetinin kullanıcılara sanal sunucu oluşturulup sunulmasıdır. Altyapı bakımından esneklik sağlamasının yanında kaynaklar ihtiyaca göre arttırılıp azaltılabilir. Servis olarak altyapı hizmetinde donanımsal bazı özellikler ile ağ sistemleri sağlanır. Donanımsal bazı özellikler sağlandığı için kullanıcı bu altyapı üzerinde makinelerini başlatır. Böylelikle ihtiyaç duyulan donanımsal altyapı servis olarak sunulur.



Şekil 2.7. Iaas, PaaS, SaaS [22]

2.5.2.Bulut Kurulum Modelleri

2.5.2.1.Genel Bulut

Bir servis sağlayıcı tarafından işletilen altyapının veya çalıştırılan yazılımların internet üzerinden herkese açık şekilde kiralanması biçiminde çalışan modeldir. Amazon AWS, Microsoft ve Google gibi genel bulut sağlayıcıları kendi altyapılarını işletir ve internet aracılığıyla erişim sunarlar [23].

2.5.2.2.Topluluk Bulutu

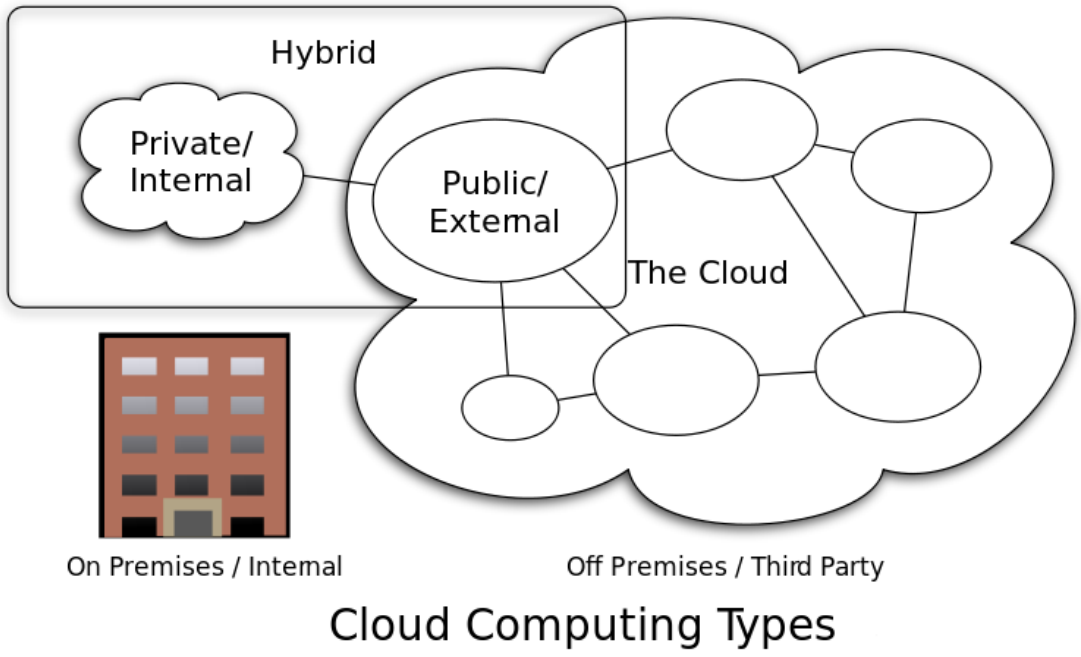
Topluluk bulutu, oluşturulan bulut altyapısının belirli bir topluluk ile paylaşılmasıdır. Aynı amaç için veya aynı güvenlik gereksinimlerine sahip olan firmalar veya devlet kuruluşlarının bir araya gelerek kullandığı bulut modelidir. Dahili olarak yönetilebildiği gibi üçüncü parti tarafından da yönetilebilir.

2.5.2.3.Hibrit(karma)Bulut

Karma bulut iki ya da daha fazla bulutun birleşimidir. Birleştirilen bulutlar kendi özelliklerini kaybetmeden bağlanırlar.

2.5.2.4.Özel Bulut

Özel bulut tek bir organizasyon için kullanılan buluttur.



Şekil 2.8. Bulut Kurulum Modelleri [24]

2.6.Büyük Veri

Büyük veriler, mevcut bilgi sistemlerinin işleyemeyeceği kadar geniş ve karmaşık veri kümeleridir ve yüksek hacimlerinin yanında, yüksek veri üretim hızı ve yüksek veri değişkenliğine sahiptir [25].

Hemen her alanda bilişim teknolojileri kullanılan dünyada sayısal veri üretimi, yapılan işlerin ve tüm etkinliklerin bir yan ürünü haline gelmiş durumdadır. Örneğin, yüksek çözünürlüklü (HD) bir video'nun tek bir saniyesi, 1 tam sayfa metnin veri hacminin tam 2000 katı veri üretmektedir.

Büyük verilerin kaynaklarına birkaç örnek verilirse.

- CERN' deki LHC deneyleri, atom altı parçacıkları saniyede 600 milyon kez çarpıştırarak, 150 milyon almaç (sensor) aracılığı ile 25 petabayt işlenecek veri (toplam verilerin on binde biri) üretir.
- SDSS (Sloan Digital Sky Survey) gözlemleri 2000 yılından beri 140 terabayt veri toplamıştır. 2016'da yerini alacak olan LSST (Large Synoptic Survey Telescope) ise aynı miktardaki veriyi 5 günde toplamaya başlayacaktır.
- NASA'nın iklim modelleme merkezi (NCSS) süper bilgisayar kümesinde 32 petabayt veri işlenmektedir.
- Özel sektörde ise ebay'in veri ambarları 40 petabayt veri işlemekte, Amazon' un veritabanları 25 petabaytı bulmakta, küresel düzeyde iş dünyasının veri hacmi her 1.2 yılda iki katına çıkmaktadır [26].

Büyük verinin en önemli beş özelliği Tablo 2.2'de gösterilmiştir.

Tablo 2.2. Büyük veri bileşenleri

Çeşitlilik (variety)	Farklı kaynaklardan, yapısal ve yapısal olmayan çok çeşitli veriler üretilmektedir.
Hız (velocity)	Hızla artan veri ile işlem sayısı ve çeşitliliğinin de aynı hızda artması ve sistemin de buna göre tasarlanması gerekir.
Hacim (volume)	Gittikçe büyümekte olan verinin işlenmesi, saklanması, arşivlenmesi ile nasıl başa çıkacağının da planlanması gerekir.
Doğrulama (verification)	Veri akışı sırasında doğru katmandan güvenlik seviyesinde doğru kişiler tarafından izlenip görünür veya gizli kalması gerekmektedir.
Değer (value)	Veriden anlamlı bilgiye ulaşabilmek önemlidir.

3. MAKİNE ÖĞRENMESİ

Makine öğrenmesi mevcut verilerden çıkarımlar yapıp bu çıkarımlardan tahmin yapılmasıdır. Makine öğrenmesine örnek olarak sınıflandırma, yüz tanıma, coğrafi koordinat kümeleme, arama motorları verilebilir.

Makine öğrenmesinde veriler eğitim verisi ve test verisi olmak üzere ikiye ayrılır. Eğitim verisinde algoritma bu veriden çıkarımlarda bulunur ve bir model şekillendirir. Test verisinde ise şekillenmiş olan modelin gerçeğe yakınlığının ne düzeyde olduğu tespit edilir ve buna göre tahminlerde bulunulur.

Makine öğrenmesi yöntemleri temel olarak iki çeşit öğrenmeye dayanır. Bunlar, gözetimli ve gözetimsiz öğrenmedir. Gözetimli öğrenme, bilgisayarın var olan veriden eğitilerek, gelecek olan diğer verilerin durumunu tahmin etmeye dayanır. Gözetimsiz öğrenme ise, var olan veriden örüntüler çıkarmaya dayalı olarak geliştirilmiştir.

Makine öğrenmesi bilgisayar bilimleri, mühendislik ve istatistiğin kesişiminde bulunur [27].

Makine öğrenmesi algoritmalarını kategorize edilmiş hali Tablo 3.1'deki gibidir.

Tablo 3.1. Makine öğrenmesi algoritmaları

Gözetimsiz	Gözetimli
• **Kümeleme ve Boyut İndirgeme** o SVD o PCA o K-Means • **Birliktelik Analizi** o Apriori o FP-Growth • **Hidden Markov Modeli**	• **Regresyon** o Liner o Polinomal • **Decision Tree** • **Random Forest** • **Sınıflandırma** o KNN o Logistic Regression o Naive Bayes o SVM

3.1.Sınıflandırma

Sınıflandırma, makine öğrenmesi ve istatistikte, bir dizi yeni gözlemin, belli eğitim seti verisine bağlı olarak hangi kategoride üyelikleri olduğunun belirlenmesidir. Bireysel gözlemler, çeşitli açıklayıcı değişkenler olarak bilinen, ölçülebilir özellikler kümesi içinde analiz edilir. Bu özellikler çeşitli şekillerde kategorize edilebilir. Bazı algoritmalar ayrık veri açısından sadece çalışmak ve gerçek değerli ya da tamsayı değerli veri gruplarının bölümlendirilmesini gerektirir. Örneğin "spam" ya da "spam değil" sınıflara belirli bir e-posta atama ya da hastanın gözlenen özelliklerine göre tarif edilmesi, belirli bir hasta için bir teşhis atama gibi.

Genel olarak sınıflandırma uygulayan bir algoritma sınıflayıcı olarak bilinir. Sınıflayıcı terimi bazen de bir sınıflandırma algoritması tarafından uygulanan matematiksel fonksiyon olup bir kategoriye giriş verisi atayan anlamına gelir. Doğru tanımlanmış gözlem kümesi eğitiminin yapıldığı makine öğrenme terminolojisinde sınıflandırma, denetimli öğrenme, yani öğrenme örneği olarak kabul edilir.

Sınıflandırma, örüntü tanıma gibi belli bir giriş değeri için çıkış değeri atayan daha genel bir sorun örneğidir.

3.1.1.Sınıflandırma Algoritmaları

Sınıflandırma, veri madenciliğinin görevlerinden biri olup birçok uygulama çoğunlukla sınıflandırma yapmayı gerektirir. En yaygın sınıflandırma algoritmaları decision tree, naive bayes, KNN, SVM, ID3, C4.5'tir.

Aşağıda bu tez çalışmasında kullanılan Naive Bayes algoritması detaylı olarak incelenmiştir.

3.1.1.1.Naive Bayes

Naive Bayes sınıflandırma metodu Bayes kuralına dayalıdır [28]. Naive Bayes ile normalizasyonsuz olasılıklar üzerinden sınıflandırma yapılabilir. Bayes Teoremi yalnızca kategorik veri özelliklerinde kullanılabilmekte iken nümerik değerlere sahip özelliklere uygulamak için örneklerin Gauss dağılımına sahip oldukları varsayılır.

Bayes sınıflandırma prosedürleri genel popülasyon içindeki farklı gruplar ile ilişkili alt popülasyonların göreli boyutları hakkında herhangi bir mevcut bilgi alarak doğal bir yol sağlamaktadır. Bayes hesaplamaları maliyet açısından yüksek olma eğiliminde

olduklarından, alternatif hesaplamalar geliştirilmiştir. Bazı Bayes işlemleri grup üyeliği olasılıklarını hesaplama içerir. Bu her yeni gözlem için tek bir grup-etiket basit veri analizi olan daha bilgilendirici sonuç sağlamak olarak görülebilir.

Bayes Teoremi rastgele bir A olayı ile rastgele bir B olayı için koşullu olasılıklar ve marjinal olasılıklar ilişkisidir. Yani:

$$P(A|B)=\frac{P(B|A)P(A)}{P(B)} \quad \textbf{(3.1)}$$

Bu denklemde P(A) 'ya A için marjinal olasılık denir. P(A|B), B için A'nın koşullu olasılığıdır. P(B|A), A için B'nin koşullu olasılığıdır. P(B) ise B için marjinal olasılıktır [29].

Naive Bayes, metin sınıflandırmada kullanılan bir metin madenciliği yöntemidir. Naive Bayes Sınıflandırma'da sınıflar ve örnek verilerin hangi sınıfa ait olduğu önceden bilinir.

Öğreticili makine öğrenmesi dediğimiz bu yöntem ile sonuca karar verilmiş olur. Burada yapılan işlem daha önceden tanımlanmış olan sınıflara örüntüyü sınıflandırmaktır. Buradaki her örüntü de nicelik kümesi tarafından temsil edilmektedir.

Nicelik kümesi aşağıdaki gibi ifade edilir [30]

x(i), i=1,2,….,L ise

$x= [x(1),x(2),…….,x(L)\}]^T \in R^L$

$x \in R^L$ ise ve S de ayrıştırılacak sınıflar kümesi:

$P(S_i|x) *p(x)=p(x|S_i)*P(S_i)$ olur ve:

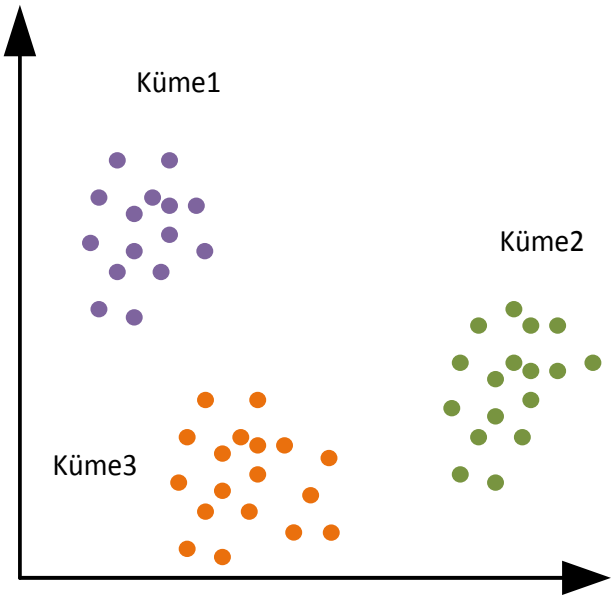
$$p(x)=\sum_{i=1}^{L} p(x/S_i)P(S_i) \quad \textbf{(3.2)}$$

Naive Bayes sınıflandırma, hastalık teşhisi, genetik araştırmalar ve metin ayrıştırma gibi birçok alanda kullanılmaktadır.

3.2.Kümeleme

Gözetimsiz öğrenme tekniklerinden olan kümeleme bir nesnenin özelliklerine göre kendisine benzer olan gruba dahil edilmesidir. Sınıflandırma yeni gelen ve bilinen bir verinin belli bir grup ile kıyaslanarak hangi gruba daha çok benzerlik gösteriyor ise o gruba dahil edilme işi idi fakat kümelemede gelen verinin bu gruplara ayrılma işi yapılır.

Aynı kümedeki elemanlar benzer olup veriler herhangi bir sınıfa dahil değildir. Yani verilerin sınıfları bilinmez. Bazı uygulamalarda kümeleme sınıflandırmanın bir önişlemi gibi görev alır [31].



Şekil 3.1. Kümeleme

Kullanılacak olan kümeleme algoritmasının seçimi, veri tipine ve amaca bağlıdır [32]. Kümeleme yöntemleri sıralanırsa bunlar [33] :

En çok kullanılan kümeleme algoritmaları şunlardır:

- K-Means
- Mean-Shift
- Spectral Clustering
- Affinity Propagation
- Hierarchical Clustering
- DBSCAN
- Gaussian Mixtures

Bu algoritmaların ölçeklenebilir olması, farklı büyüklükteki kümeleri belirleyebilir olması, yüksek boyutlu olması ve farklı tipteki özellikleri destekliyor olması gerekmektedir.

Aşağıda bu tez çalışmasında kullanılan K-Means kümeleme algoritması detaylı olarak incelenmiştir.

3.2.1. K-Means Kümeleme

K-means yöntemi, kümeleme problemini çözen en basit denetimsiz öğrenme yöntemleri arasında yer alır [34]. K-means yönteminde küme merkezi belirlendikten sonra merkezin dışındaki verilerin mesafelerine göre kümelendirilmesi yapılır. Bu kümelendirmeye göre yeni merkez belirlenip herhangi bir değişim olmayıncaya kadar aynı adımlara devam edilir.

K-means'te her veri sadece bir kümeye ait olabilir. K küme sayısı N de veri sayısı olmak üzere k tane nesnenin rastgele seçimi yapıldıktan sonra diğer nesneler küme merkezine uzaklıkları dikkate alınıp benzer oldukları kümeye dahil edilir. K-means kümeleme yapılırken uzaklık belirlemek için genellikle Öklit mesafesi kullanılır.

Öklit mesafesine göre iki nokta arasındaki uzaklık bulunurken :

$$d_{ab=\|a-b\|=\sqrt{(a_1-b_1)^2 +(a_2-b_2)^2 +\cdots+(a_n-b_n)^2}} \quad \textbf{(3.3)}$$

denklemi kullanılır.

Benzerlik ise genellikle uzaklığın tersi olup

$$s_{ab} = \frac{1}{1+\|a-b\|} \quad \textbf{(3.4)}$$

denklemi ile hesaplanır.

Kümeleme ile benzerliği yüksek olan ve sınıf bilgisi olmayan verilerin gruplama işlemi yapılır. Her gruplama ve küme merkezinin hesaplanması işlemi bir iterasyon içerisinde gerçekleştirilir. Algoritmanın iterasyon sayısı çoğunlukla J maliyet fonksiyonu ile belirlenir.

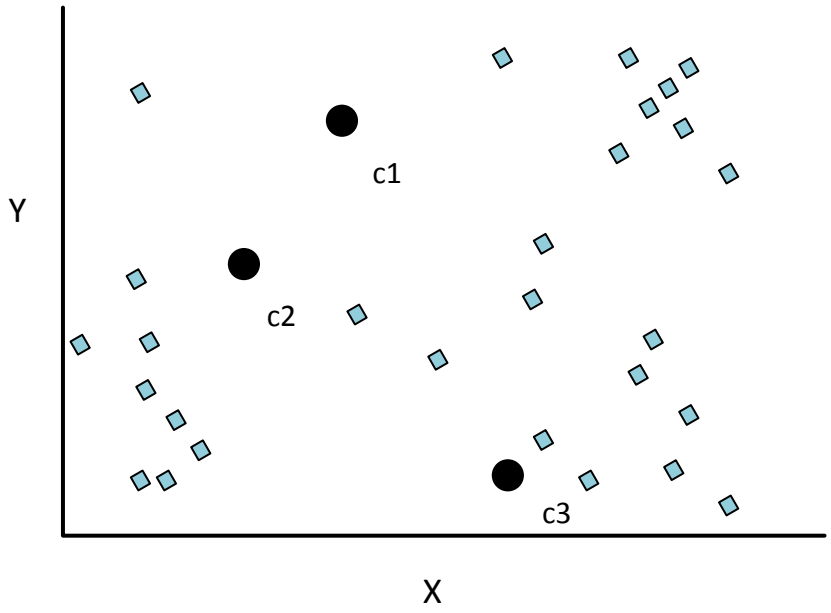
$$J = \sum_{i=1}^{K} \sum_{k} \|x_k - c_i\|^2 \quad \textbf{(3.5)}$$

Burada maliyet fonksiyonunun, tüm küme merkezi değerleri en uygun konumda olduğunda minimum değerde olması istenir [35].

Sınıf sayısı belirlendikten sonra küme merkezi (centroid) değeri hesaplanır. Küme merkezinin nesnelere olan uzaklığı da hesaplandıktan sonra en kısa uzaklığa göre gruplandırma yapılır. Nesne yer değiştirdiğinde algoritma sonlanır. Kolay uygulanabilir ve büyük veri kümeleri için hızlı çalışabilir.

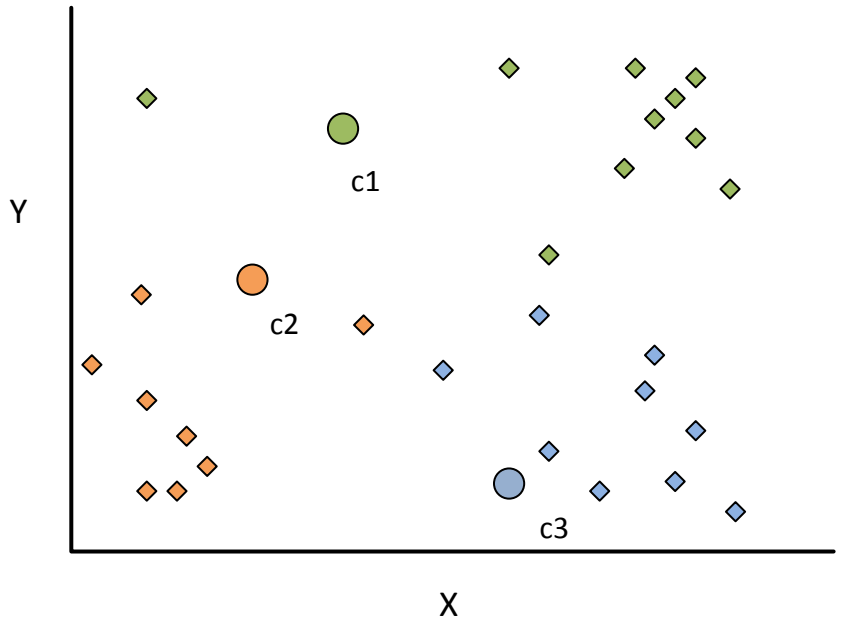
Aşağıda K-Means kümeleme adımlarını şekiller üzerinde gösteren bir örnek verilmiştir [36]:

Öncelikle X-Y koordinat düzleminde rastgele olmak üzere 3 adet küme merkezi c1, c2, c3 olarak belirlensin.



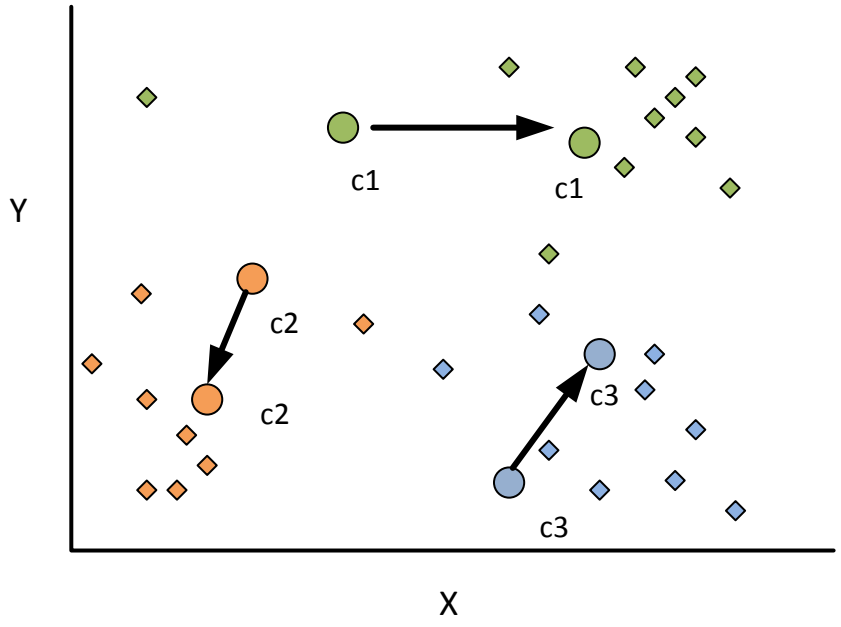
Şekil 3.2. Küme merkezleri

Her örnek kendisine en yakın olan merkezin olduğu kümeye dahil edilir.



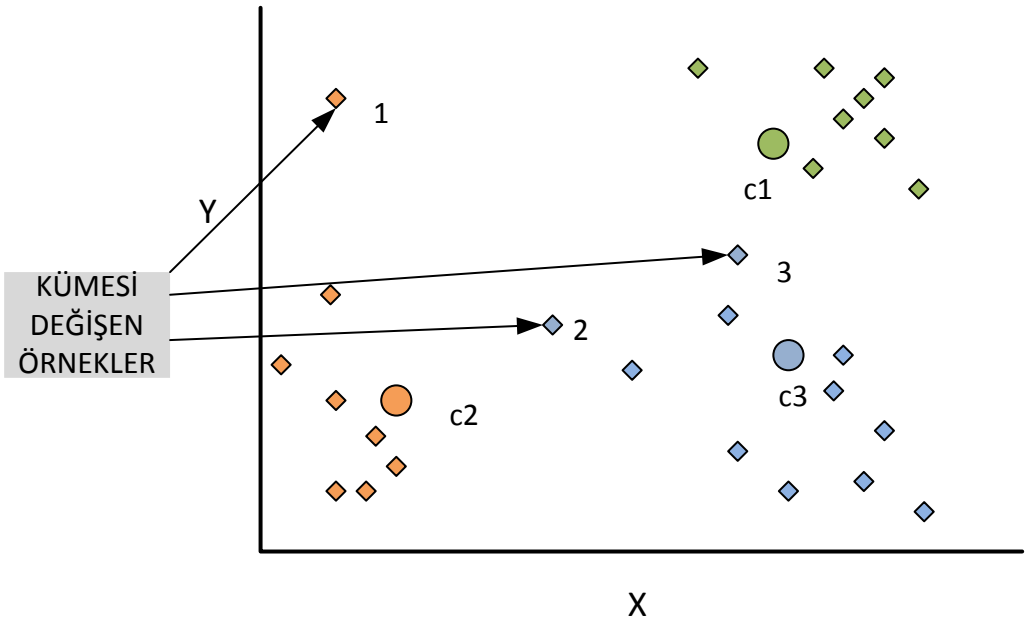
Şekil 3.3. Kümeye en yakın merkez belirleme

Daha sonra merkezler kendi kümelerinin merkezine taşınır.c_1



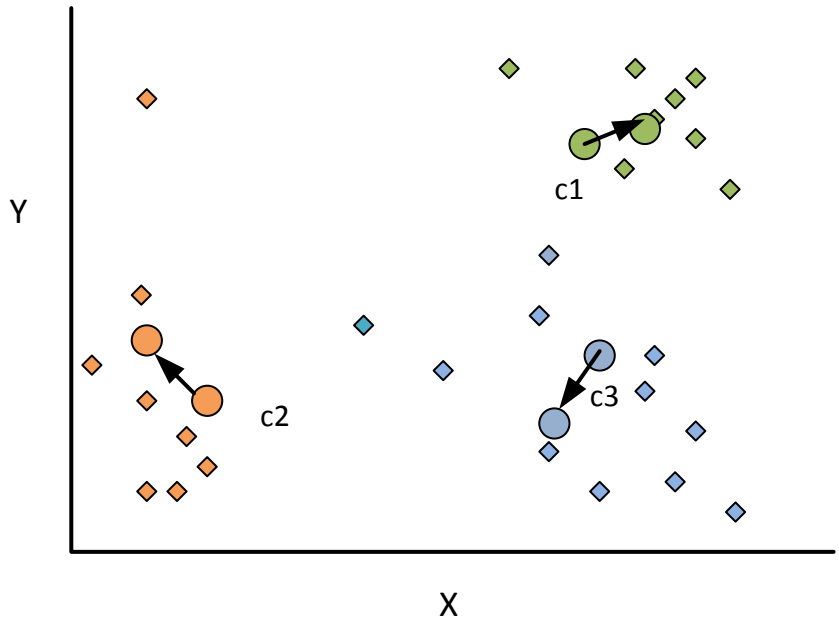
Şekil 3.4. Küme merkezlerini taşıma

Her örnek tekrardan kendisine en yakın olan merkezin olduğu kümeye dahil edilir. Burada bazı örneklerin kümesinin değiştiği görülür.



Şekil 3.5. Kümesi değişen örnekler

Merkezler kendi kümelerinin merkezi olan noktalara taşınır.



Şekil 3.6. Küme merkezlerinin taşınması

Basit bir örnek ile K=2 için k-means algoritması gerçekleştirilirse [37]:

Tablo 3.2. K=2 için k-means algoritması

Nesne	**Değişken1**	**Değişken2**
1	1.0	1.0
2	1.5	2.0
3	3.0	4.0
4	5.0	7.0
5	3.5	5.0
6	4.5	5.0
7	3.5	4.5

Başlangıç durumuna getirme: İki küme (cluster) için gelişigüzel olmak üzere küme merkezleri (centroids) seçilsin.

m1= (1.0 ,1.0) ve m2= (5.0,7.0) olsun.

Tablo 3.3. Küme merkezlerinin seçilmesi

Nesne	**Değişken1**	**Değişken2**
1	**1.0**	**1.0**
2	1.5	2.0
3	3.0	4.0
4	**5.0**	**7.0**
5	3.5	5.0
6	4.5	5.0
7	3.5	4.5

Tablo 3.4. Grupların elde edilmesi

	Nesne	**Ana vektör**
Grup1	1	(1.0 ,1.0)
Grup2	4	(5.0,7.0)

İkinci adıma geçildiğinde {1,2,3} ve {4,5,6,7} içeren iki adet küme elde edilmiş olur. Yeni küme merkezleri hesaplanır.

Tablo 3.5. Yeni küme merkezlerinin hesaplanması

Nesne	**Küme Merkezi1 (Centroid1)**	**Küme Merkezi2 (Centroid2)**
1	0	7.21
2 (1.5, 2.0)	1.12	6.10
3	3.61	3.61
4	7.21	0
5	4.72	2.5
6	5.31	2.06
7	4.30	2.92

m_1=$(\frac{1}{3}(1.0 + 1.5 + 3.0), (\frac{1}{3}(1.0 + 2.0 + 4.0))$=(1.83,2.33)

m_2=$(\frac{1}{4}(5.0 + 3.5 + 4.5 + 3.5), (\frac{1}{4}(7.0 + 5.0 + 5.0 + 4.5))$=(4.12, 5.38) olarak bulunur.

Daha sonra belirlenen k= 2'ye göre seçilen merkezler ile öklit uzaklık hesaplaması yapılır.

$$d(m_1, 2) = \sqrt{|1.0 - 1.5|^2 + |1.0 - 2.0|^2} = 1.12$$

$$d(m_2, 2) = \sqrt{|5.0 - 1.5|^2 + |7.0 - 2.0|^2} = 6.10$$

Bu küme merkezleri kullanılarak her bir nesne için tabloda görüldüğü gibi öklit uzaklığına göre hesaplama yapılır.

Tablo 3.6. Yeni kümelerin belirlenmesi

Nesne	**Küme Merkezi1 (Centroid1)**	**Küme Merkezi2 (Centroid2)**
1	1.57	5.38
2	0.47	4.28
3	**2.04**	**1.78**
4	5.64	1.84
5	3.15	0.73
6	3.78	0.54
7	2.74	1.08

Bu durumda yeni kümeler {1,2} ve {3,4,5,6,7} olur. Yeni küme merkezleri de;

$$m_1 = (1.25,1.5)\ ve\ m_2 = (3.9,5.1)$$

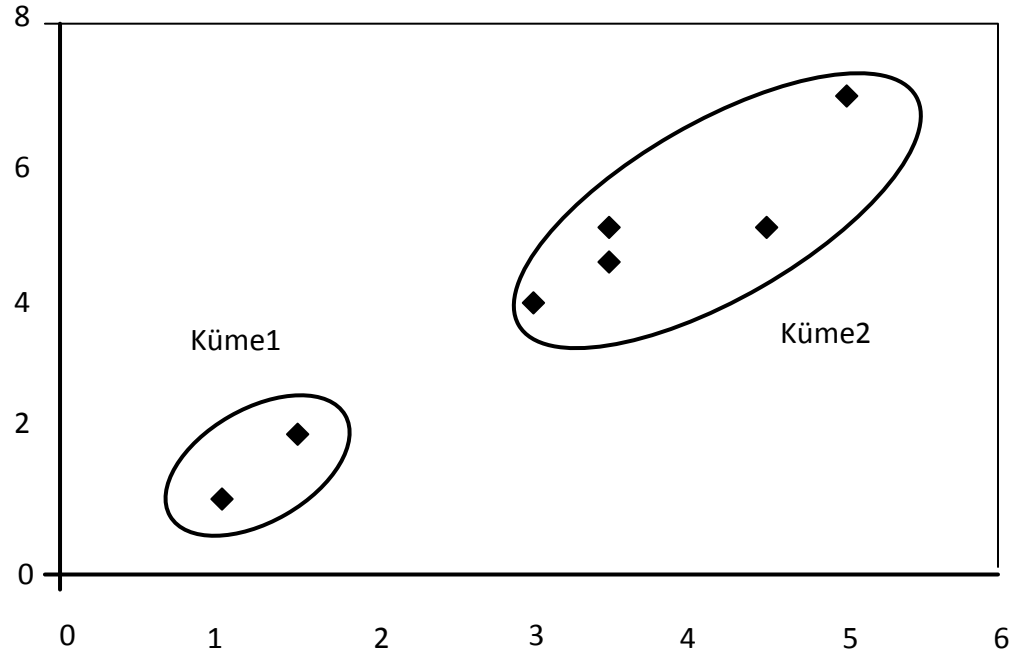
dir.

Buna göre kümede herhangi bir değişim gözlenmemiştir. Algoritma burada sonlanır ve neticede {1,2} ve {3,4,5,6,7} olmak üzere iki küme(cluster) elde edilmiş olur.

Tablo 3.7. Algoritma sonucu yeni kümeler

Nesne	**Küme Merkezi1 (Centroid1)**	**Küme Merkezi2 (Centroid2)**
1	0.56	5.02
2	0.56	3.92
3	3.05	1.42
4	6.66	2.20
5	4.16	0.41
6	4.78	0.61
7	3.75	0.72

Sonuç olarak tüm bu değerler ile küme grafiksel olarak gösterildiğinde Şekil'deki gibi Küme1 ve Küme2 elde edilmiş olur.



Şekil 3.7. Küme1 ve Küme2'nin Elde Edilmesi

Aynı adımlar ve hesaplamalar K=3, $m_1 = 1,\ m_2 = 2\ ve\ m_3 = 3$ için tekrarlandığında:

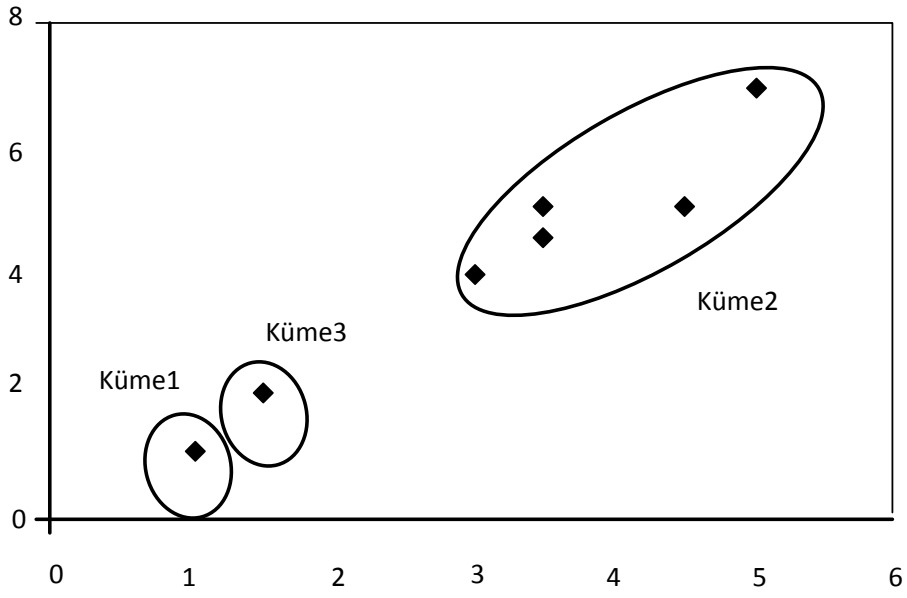
Tablo 3.8. K=3 için k-means algoritması

Nesne	$m_1 = 1$	$m_2 = 2$	$m_3 = 3$	Küme(cluster)
1	0	1.11	3.61	1
2	1.12	0	2.5	2
3	3.61	2.5	0	3
4	7.21	6.10	3.61	3
5	4.72	3.61	1.12	3
6	5.31	4.24	1.80	3
7	4.30	3.20	0.71	3

Tablo 3.9. K=3 için yeni küme merkezlerinin belirlenmesi

Nesne	m_1 (1.0, 1.0)	m_2 (1.5, 2.0)	m_3 (3.9, 5.1)	Küme(cluster)
1	0	1.11	5.02	1
2	1.12	0	3.92	2
3	3.61	2.5	1.42	3
4	7.21	6.10	2.20	3
5	4.72	3.61	0.41	3
6	5.31	4.24	0.61	3
7	4.30	3.20	0.72	3

K=3 için kümeleme yapıldığında Şekil'deki Küme1, Küme2 ve Küme3 elde edilir.



Şekil 3.8. Küme1, Küme2 ve Küme3

4. BULUT ÜZERİNDE DOKÜMAN SINIFLANDIRMA VE KÜMELEME UYGULAMALARI

Bu bölümde makine öğrenmesi algoritmaları kullanarak doküman sınıflandırma ve kümeleme uygulamalarının gerçekleştirilmesi için gerekli programların kurulumu ve üzerinde çalışılacak platformlar anlatılmıştır. Bu platformlar ve teknolojiler kullanılarak sınıflandırma ve kümeleme algoritmaları çalıştırılmış ve sistemin performans ölçümleri verilmiştir.

Hadoop, Mahout, Spark gibi teknolojiler ile sınıflandırma ve kümelemenin nasıl yapıldığı, ne kadar başarılı olduğu ve birbirine göre üstün yönleri açıklanmıştır. Google Cloud, Amazon EC2 ve Microsoft Azure bulutlarında oluşturulan Hadoop kümeleri üzerinde, testler gerçekleştirilmiştir. Makine öğrenmesi algoritmaları ile büyük veri üzerinde analizler yapılırken sağladığı esneklik ve faydalardan ötürü Bulut Bilişim kullanılmıştır.

4.1. Uygulama Ortamı

Çalışma yapılırken kullanılmış olan tüm platformlar ve gerekli olan altyapının oluşturulması bu bölümde anlatılıp gerçekleştirilen kurulumlar ve adımları hakkında bilgi verilmiştir.

4.1.1.Apache Hadoop Kurulumu

MapReduce ile yazılan programların çalıştırılabilmesi için öncelikle Hadoop kurulumunun gerçekleştirilmiş olması gerekir. Bu bölümde Hadoop 2.2.0 versiyonunun Ubuntu 13.10 üzerinde kurulum adımları verilmiştir [38].

Hadoop Java tabanlı açık kaynaklı bir kütüphane olup öncelikle java kurulumunun gerçekleştirilmiş olması gerekir.

```
sudo apt-get install python-software-properties

sudo add-apt-repository ppa:webupd8team/java

sudo apt-get update

sudo apt-get install oracle-java7-installer
```

Java kurulumu /usr/lib/jvm/java-7-oracle/ şeklinde belirtilen yola kurulmuş olur. Daha sonra https://hadoop.apache.org/ sitesinden Hadoop indirilip açılır.

Hadoop indirilip açıldıktan sonra Hadoop'un kullanacağı dizinler oluşturulur.

```
sudo mkdir -p /var/hadoop/tmp
sudo mkdir -p /var/hadoop/dfs/data
sudo mkdir -p /var/hadoop/dfs/name
sudo mkdir -p /var/hadoop/mapred/local
sudo mkdir -p /var/hadoop/mapred/system
```

Tüm makinelerde aynı tutulmak üzere Hadoop için özel bir grup ve kullanıcı oluşturulur.

```
sudo addgroup hadoop
sudo adduser --ingroup hadoop hduser
```

Bu kullanıcı grubunu bir listeye eklemek kolaylık sağlayacağından aşağıdaki komut kullanılır:

```
sudo nano /etc/sudoers
```

Bu dosyaya %hadoop ALL=(ALL) ALL satırı eklenir.

Parola ve diğer bilgiler girildikten sonra klasöre erişim için izin verilir:

```
sudo chown -R hduser:hadoop /var/hadoop/
sudo chown -R hduser:hadoop /usr/local/hadoop
```

Bu aşamadan sonra ssh ayarları yapılandırılır ve işlem sonunda ssh localhost komutu ile ayarları test edilip exit komutu ile ssh'tan çıkılır.

```
ssh-keygen -t rsa -P ""
cat ~/.ssh/id_rsa.pub >> ~/.ssh/authorized_keys
ssh localhost
```

Başlangıç değişkenleri ayarları yapılıp, makinenin IP adresine ifconfig ile erişimden sonra ağ ayarları da yapılır ve host dosyasına IP adresleri için daha sonra kullanılmak üzere isim verilir.

Bu aşamadan sonra kurulumun tamamlanabilmesi için Hadoop ayarları yapılır. /usr/local/hadoop/etc/hadoop/ konumunda bulunan 4 adet XML dosyasının ayarları yapılır.

/usr/local/hadoop/etc/hadoop/core-site.xml

/usr/local/hadoop/etc/hadoop/hdfs-site.xml

/usr/local/hadoop/etc/hadoop/yarn-site.xml

/usr/local/hadoop/etc/hadoop/mapred-site.xml

Burada belirtilen tüm Hadoop modülleri için ayarlar yapıldıktan sonra:

/usr/local/hadoop/etc/hadoop/hadoop-env.sh konumunda

export JAVA_HOME=${JAVA_HOME} satırı bulunur ve

export JAVA_HOME="/usr/lib/jvm/java-7-oracle"

export HADOOP_COMMON_LIB_NATIVE_DIR=${HADOOP_PREFIX}/lib/native

export HADOOP_OPTS="-Djava.library.path=$HADOOP_PREFIX/lib"

ile değiştirilir.

Son olarak hdfs namenode –format komutu ile NameNode'a format atılır ve kurulum tamamlanmış olur.

4.1.2.Apache Mahout Kurulumu

Öncelikle kurulumu yapmak için bir proje yönetim aracı olan ve projeleri belli bir standart içinde tutmak için projelerin iskeletini oluşturan Apache Maven denilen yapı indirilir [39].

sudo apt-get install maven

Apache Mahout'un sitesine girerek indirilmek istenen Mahout sürümü indirilir.

cd /home/selen/Documents

Yeni bir klasör oluşturulur.

mkdir mahout

Oluşturulan klasörün içine indirilen Apache Mahout klasörüne cd mahout ile gidilir ve mvn install komutu çalıştırılır.

4.1.3.Apache Spark Kurulumu

Spark kurulumu sırası ile aşağıda verilen adımlardan oluşmaktadır [40].

- Spark kurulumu için ilk olarak Spark indirilir:

  wget http://www.eu.apache.org/dist/spark/spark-1.2.0/spark-1.2.0.tgz

- Daha sonra indirilen dosya tar vxzf spark-1.2.0.tgz

  cd spark-1.2.0 komutu ile açılır.

- Spark paketini build etmek için "git" kurulumu gerçekleştirilir.

  sudo yum install git (CentOS için)

- Spark paketini build etmek için sbt/sbt -Pyarn -Phadoop-2.4 assembly

  veya

mvn -Dhadoop.version=2.4.1 -DskipTests clean package

komutu kullanılır. Paketi build etmek için biri "sbt" diğeri "maven" olmak üzere iki yöntem mevcuttur.

- Eğer paket build edilmek istenmiyorsa, herhangi bir build operasonu gerektirmeyen pre-build paket indirilebilir.
  wget http://www.eu.apache.org/dist/spark/spark-1.2.0/spark-1.2.0-bin-hadoop2.4.tgz
  komutu pre-build paket için indirme işlemini gerçekleştirir.
- Spark shell'i çalıştırmak için Spark kurulum lokasyonunda ./bin/spark-Shell

komutu çalıştırılır.

4.2.Apache Mahout

Apache Mahout, Hadoop platformu üzerinde, dağıtık bir şekilde ölçeklenebilir makine öğrenmesi algoritmalarını sunan Apache tarafından tasarlanmış olan bir projedir.

4.2.1. Apache Mahout ile Metin Sınıflandırma

Bu bölümde Mahout ile metinlerin nasıl sınıflandırılacağı konusu üzerinde durulmuştur. Metin sınıflandırma yöntemlerini doğru bir şekilde kullanarak Apache Mahout ile iş akışları oluşturulabilir, kullanılabilir ve bir dizi belge sınıflandırılabilir.

Apache Mahout'tan alınan sınıflandırma sonuçları, bilgi erişim ve analizi için farklı araçlarda kullanılabilir. Metin sınıflandırmanın genel olarak hedefi önceden oluşturulmuş kurallara göre nesneleri sınıflara atamaktır [41]. Örneğin, e-postaları 'spam' ya da 'spam değil' şeklinde sınıflandırmak veya kitapları 'tarih' ya da 'kurgu' sınıflarından birisine atamak gibi.

Mahout ile bir metni sınıflandırırken algoritma olarak Naive Bayes kullanılabilir.

Sınıflandırma kuralı olarak :

c_{map}=argmax $[log\ \hat{P}\ (c) + \sum_{1 \le k \le n_d} log\ \hat{P}\ (t_k/c)\ ]$ kullanılır. **(4.1)**

Bu kural basitçe yorumlanırsa; her bir koşullu $log\ \hat{P}\ (t_k/c)$ parametresi, c sınıfı için t_k 'nın nasıl iyi bir belirleyici olduğunu gösterir. Öncelikli $log\ \hat{P}\ (c)$, c sınıfı için akrabalık derecesini belirtir. Terim ağırlıklarının ve birincil log'ların toplamı dökümanın

ne kadarlık bir bölümünün sınıfa ait olduğunu ispatlamak içindir. Bunun için çok örneği olan bir sınıf seçilir.

4.2.2. Apache Mahout ile Metin Sınıflandırma Uygulaması

Bu bölümde Apache Mahout ile metin sınıflandırma için bir örnek uygulama verilmiştir. Elimizde 20 farklı gruba ait veriler olsun ve bunları sınıflandırmak için de Naive Bayes kullanılıyor olsun. [42].

Bir çalışma dizini oluşturma ve veri kopyalama yaparken:

Komut: mkdir 20news -all

cp –R 20news -bydate/*/* 20news -all

Burada 20news-all adında bir dizin oluşturulur ve tüm verilerin bu dizine kopyalama işlemi yapılır.

Metin dosyaları Mahout tarafından kullanılabilecek Sequence dosya formatına dönüştürülür:

Komut: ./bin/mahout seqdirectory -i 20news---all -o 20news -seq

Bu komutta '-i' belirtilen giriş dosya/dizinini '-o' çıktı dosya/dizinini gösterir.

Daha sonra sequence dosyaları, vektörlere dönüştürülür:

Komut: ./bin/mahout seq2sparse \

-i 20news-seq \

-o 20news-vectors -lnorm -nv -wt tfidf

Burada,

-lnorm: log normalize, -nv: named vectors ve –wt: weight (ağırlık) anlamına gelir. Ağırlık türü olarak tfidf kullanılmıştır.

Rastgele bölünmüş olan eğitim verisi ve test verisi oluşturma:

Komut: ./bin/mahout split\

-i 20news -vectors/tfidf -vectors\

--trainingOutput 20news -train -vectors\

--testOutput 20news -test -vectors\

--randomSelectionPct 20 --overwrite --sequenceFiles

-xm sequential

Burada,

Tablo 4.1. Mahout komut vektörleri (1)

--overwrite -ow	Eğer mevcut ise, işi çalıştırmadan önce çıktı dizininin üzerine yaz
--sequenceFiles -seq	Girdi dosyaları ardışıl dosyalar ise ayarla. Varsayılan yanlıştır.
--method <method> -xm <method>	Kullanılacak yürütme yöntemi: <method>= ardışıl ya da mapreduce Varsayılan mapreduce.

Verilerin %80'i eğitim verisi %20'si de test verisi olarak ayarandığında **[42]:**

Naive Bayes modelini eğitirken:

Komut: ./bin/mahout trainnb\
-i 20news -train -vectors -el\
-o model\
-li labelindex\
-ow

Burada,

Tablo 4.2. Mahout komut vektörleri (2)

--labelIndex <labelIndex> -li <labelIndex>	labelIndex'i depolamak için yol
--overwrite -ow	Eğer varsa,iş çalıştırılmadan önce çıktı dizinini overwrite etmek

Bu Naive Bayes sınıflandırıcıyı eğitmek için asıl komuttur. Komut girdi olarak daha önceden seçilen eğitim verilerini alır ve çıktı olarak bir model verir.

Eğitim setinde kendini test ederken:

Komut: ./bin/mahout testnb\
-i 20news-train-vectors\
-m model\
-l labelindex\
-ow -o 20news-testing

Burada,

Tablo 4.3. Mahout komut vektörleri (3)

--model <model> -m <model>	Eğitim süresince oluşturulan modelin yolu
--labelIndex <labelIndex> -l <labelIndex>	Label index'in yerini işaret eden yol
--overwrite -ow	Varsa eğer, işi yürütmeden önce çıktı dizini overwrite etmek
--output <output> -o <output>	Çıktı için dizin yol adı

Bu komut, sınıflandırıcı modeli kullanılarak önceden oluşturulan eğitim verilerini test eder. Burada model eğitimi için kullanılan aynı eğitim seti üzerinde hemen hemen %100 doğruluk beklenir.

Gizleme seti üzerinde test:

Komut: ./bin/mahout testnb\

-i 20news-test-vectors\

-m model\

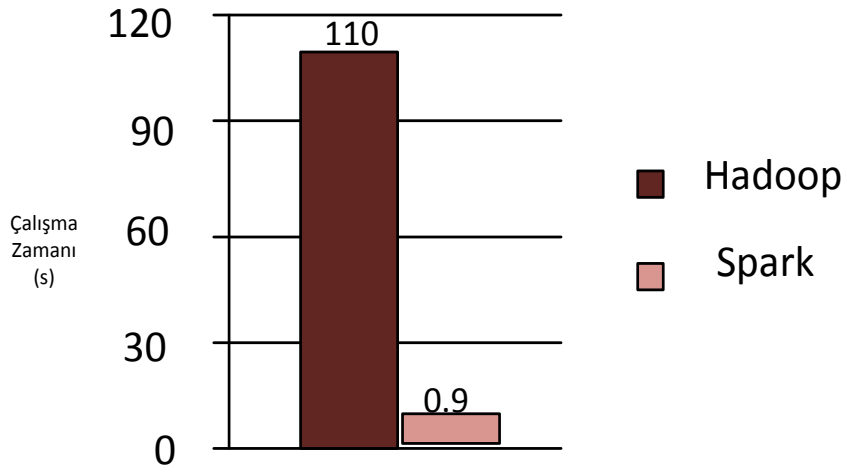
-l labelindex\

-ow -o 20news-testing $c

Öğrenme deneyimlerine dayanarak, sınıflandırıcı gerçek test verilerine uygulanır. Bu komutun çıktısı da sınıflandırma doğruluğunu gösterir .

4.3.Apache Spark

Spark, Apache tarafından büyük veri işlemek amaçlı geliştirilmiş bir başka projedir. Hadoop'tan daha hızlı çalışıyor olması sunmuş olduğu büyük avantajlardandır.



Şekil 4.1. Hadoop-Spark Karşılaştırma [43]

Spark, Java, Scala veya Python programlama dilleri kullanılarak uygulama yazılmasına imkan tanır. Paralel uygulamaları oluşturmak için yaklaşık 80 tane üst düzey operatör sunmaktadır. Hadoop'ta, Mesos'da, bağımsız olarak ya da bulutta çalışabilir. MLlib adı verilen ölçeklenebilir bir makine öğrenmesi kütüphanesine sahiptir.

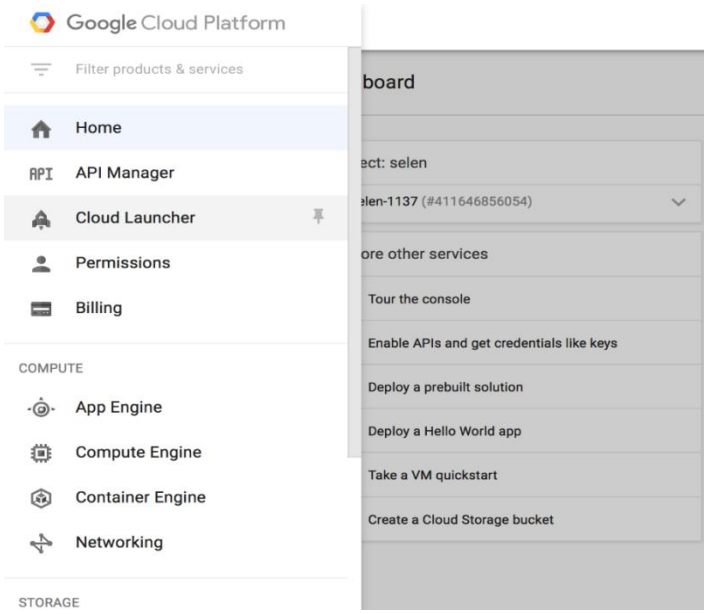
4.4.Google Cloud

Google Cloud platformu Google arama motoru, Youtube gibi sitelerin sunucu altyapısı hizmetlerini kullanıcıya sunan bir bulut platformudur [44]. İster basit ister karmaşık olsun uygulamalar bu platformda gerçekleştirilebilir.

Şekil 4.2. Google Cloud Hizmetleri [45]

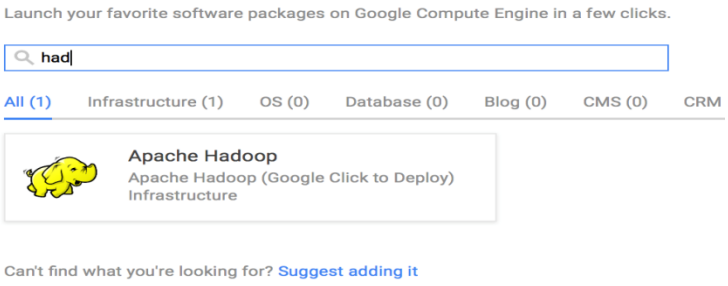
4.4.1.Google Cloud Platformu üzerinde Hadoop Kümesi Oluşturma

Öncelikle cloud.google.com'dan hesap oluşturulup giriş yapıldıktan sonra Cloud Launcher'a tıklanır.



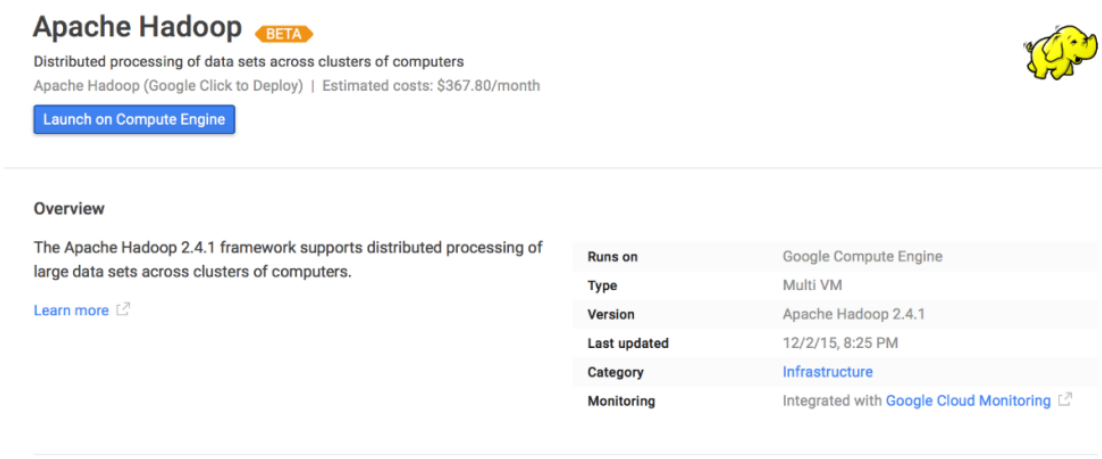
Şekil 4.3. Google Cloud kullanıcı hesabı özet sayfası

Çıkan ekrandan Hadoop seçilir. Deploy butonuna tıklandığında kümenin özellikleri ekrana gelir.



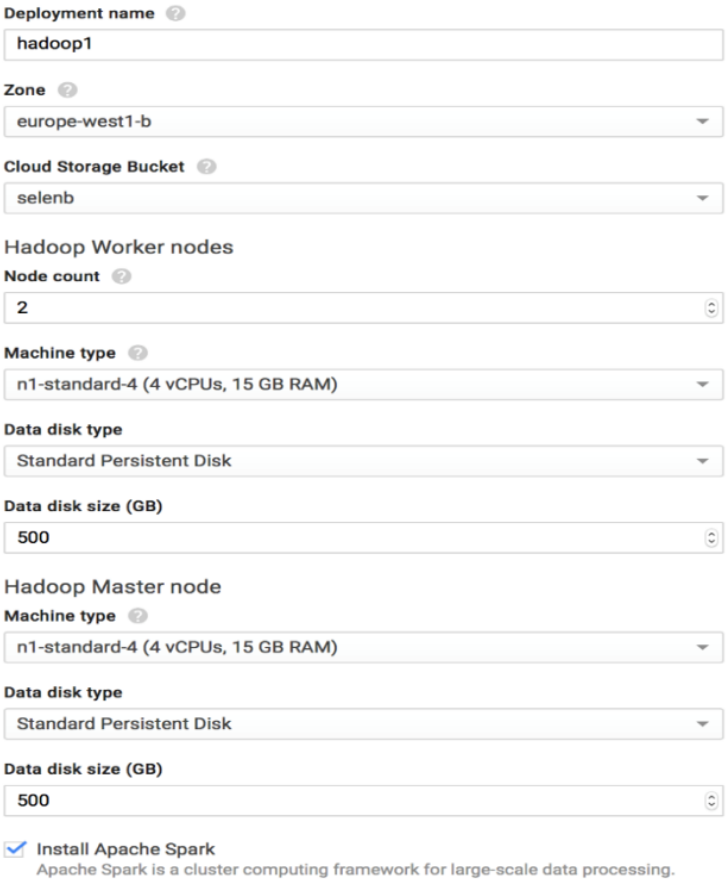
Şekil 4.4. Google Cloud "Click to Deploy" sayfasında kurulabilecek yazılımlardan Hadoop seçimi

Burada çıkan ekrandan da Launch on Compute Engine seçilir.



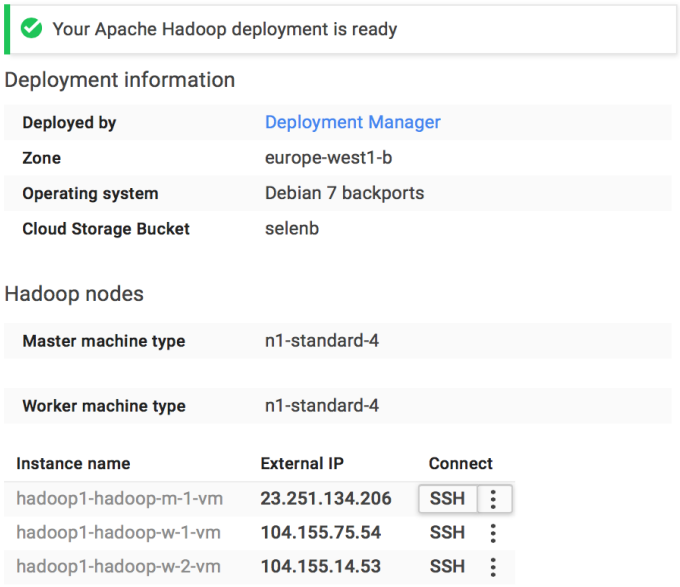
Şekil 4.5. Hadoop Kurulum ekran

Gerekli bilgiler girilir.



Şekil 4.6. Hadoop Kümesi kurulum ekranı

Kümenin özellikleri belirlendikten sonra küme oluşumu tamamlanır.



Şekil 4.7. Kurulumu tamamlanan Hadoop kümesi

4.5.Microsoft Azure

Microsoft Azure teknolojisi Windows Server işletim sisteminin özelleştirilmiş bir sürümü diyebileceğimiz büyük bir bulut platformudur [46]. Microsoft Azure ile birlikte gelen hizmetlerden web barındırma hizmeti web uygulamalarını kurumsal düzeyde oluşturmayı sağlar. Java, C#, ASP.Net gibi dil seçeneklerinin mevcut olduğu bir hizmettir. Dil özgürlüğü sağlamasının yanında birçok web uygulamasını da çalıştırmaya imkan sağlar. Ölçeklendirme özelliği Azure için önemli bir özelliktir ve ister elle ister otomatik olarak ölçeklendirme imkanı tanır [47].

Azure'un sanal makine hizmeti ile varolan bir imaj dosyasından sanal makine oluşturulabilir veya kullanıcı kendi sanal makinesini başlatabilir. Çok kısa sürede bir SQL Server çalıştırılabilir. Giriş çıkış işlemlerini hızlı yapmak adına SSD depolama kullanılır.

Azure mobil hizmetlerini birçok firma kullanmakta olup iOS, Android, Mac, Windows Phone ve Windows için aynı altyapıyı kullanabilen bir uygulama oluşturmak mümkündür [48].

ActiveDirectory özelliği de Azure için bir diğer önemli özellik olup bu özellik ile kullanıcı kimlikleri doğrulanır ve çok kısa kodlar ile ilgili kullanıcı ilgili uygulamaya kolaylıkla giriş yapabilir. Azure ile birlikte gelen popüler çözümler,

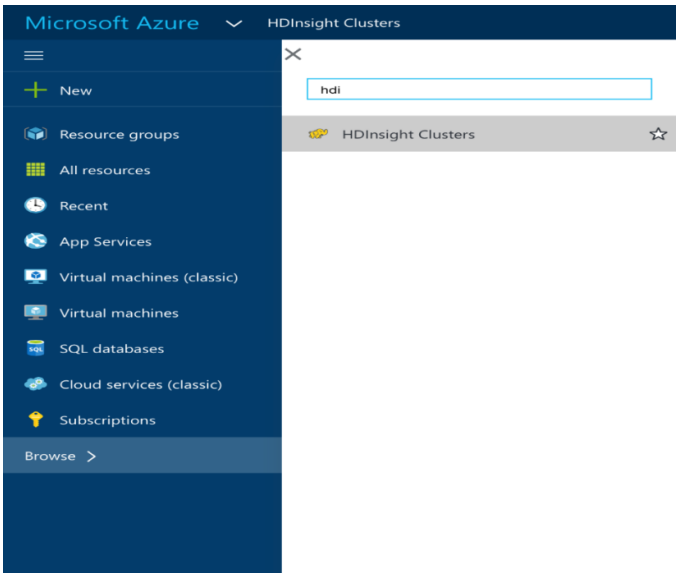
- Modern web sitelerini ve web uygulamalarını saniyeler içinde dağıtıp ölçeklendirme,

- Windows Server ve Linux'u dakikalar içinde çalıştırma,
- Hizmet olarak yürütülen SQL ilişkisel veritabanı,
- Bulut tabanlı güçlü tahmine dayalı analiz,
- Tüm mobil uygulamalar için arka uç oluşturma ve barındırma,
- Windows işlemci uygulamalarını bulutta dağıtıp, istenilen cihazda çalıştırma [49].

Şekil 4.8. Azure Servis Platformları [50]

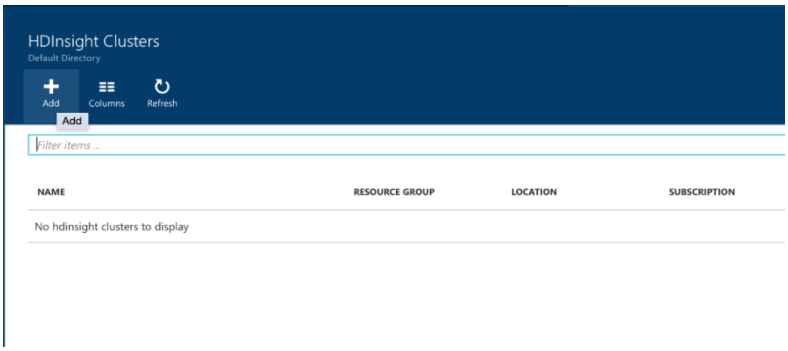
4.5.1.Azure Platformu'nda Sunucuya Erişim [Office1]

Microsoft, Azure platfromunda bir çok bulut bilişim hizmeti sunulur. HDInsight ta bunlardan biridir. HDInsight kullanılarak çok hızlı bir şekilde Hadoop kümesi oluşturulabilir. HDInsight'a hızlı bir şekilde ulaşabilmek için browse ekranından HDInsight seçilir.



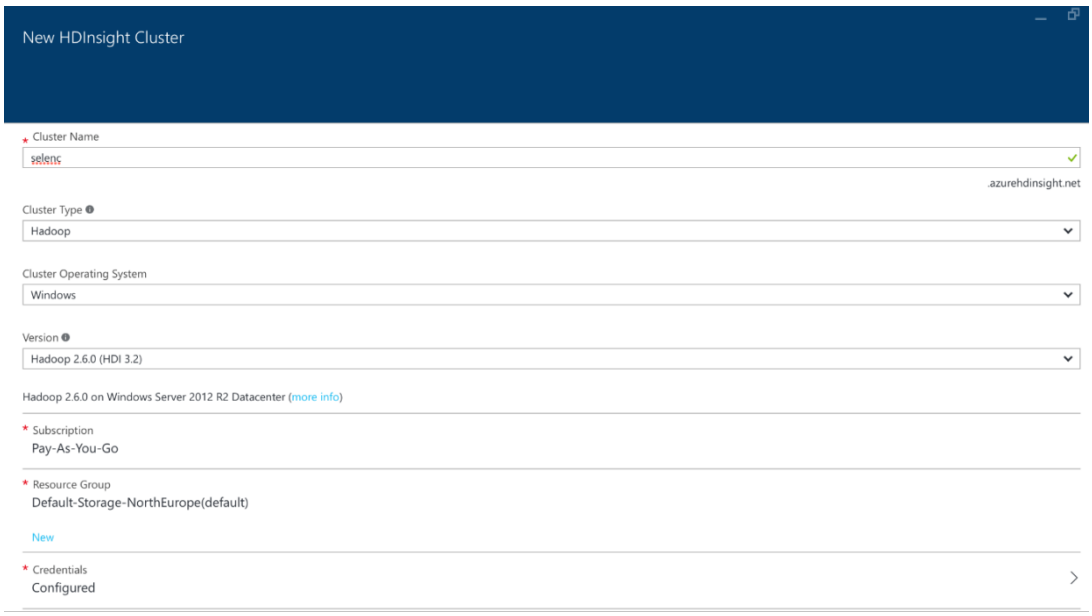
Şekil 4.9. Azure Browse Ekranı

Çıkan ekranda varsa oluşturulan kümeler görüntülenir. Yeni küme eklemek için add butonuna tıklanır.



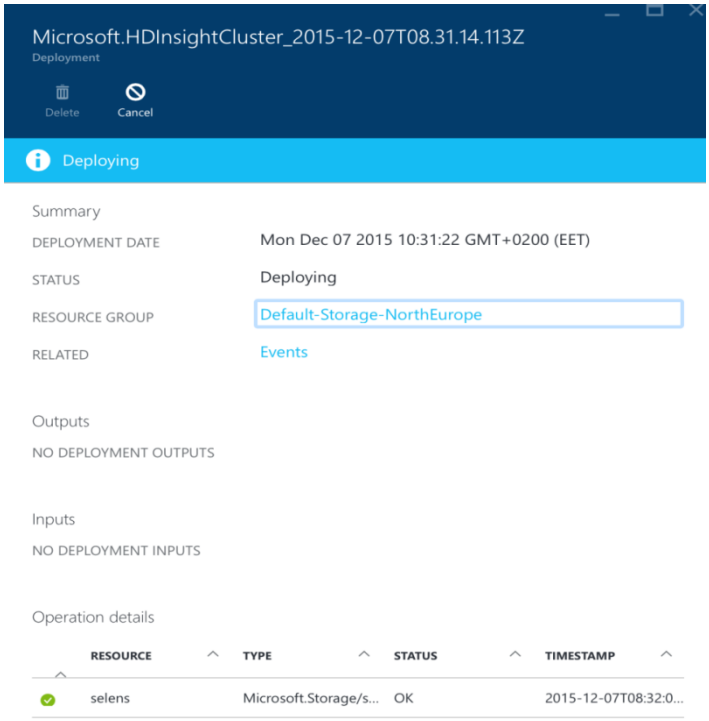
Şekil 4.10. Ekle butonu

HDInsight oluşturuken küme için, kümenin işletim sistemini, Hadoop versiyonu ödeme şekli gibi bilgiler istenmektedir. Bu sayfada ihtiyaca göre şekildeki gibi girilmiştir.



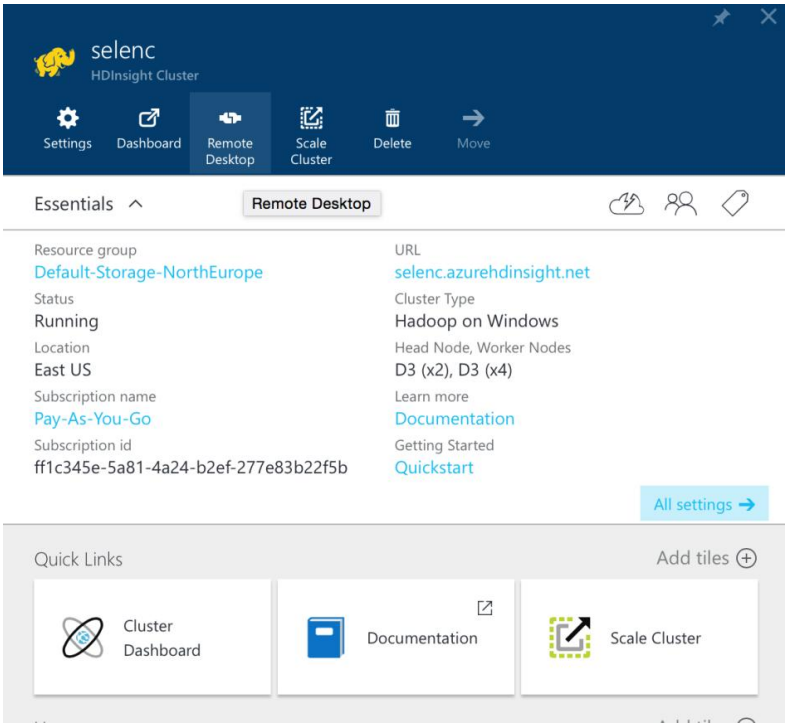
Şekil 4.11. Bilgilerin girilmesi

Deploy tuşuna basıldığında kümenin durumu gerçek zamanlı olarak görülebilir.



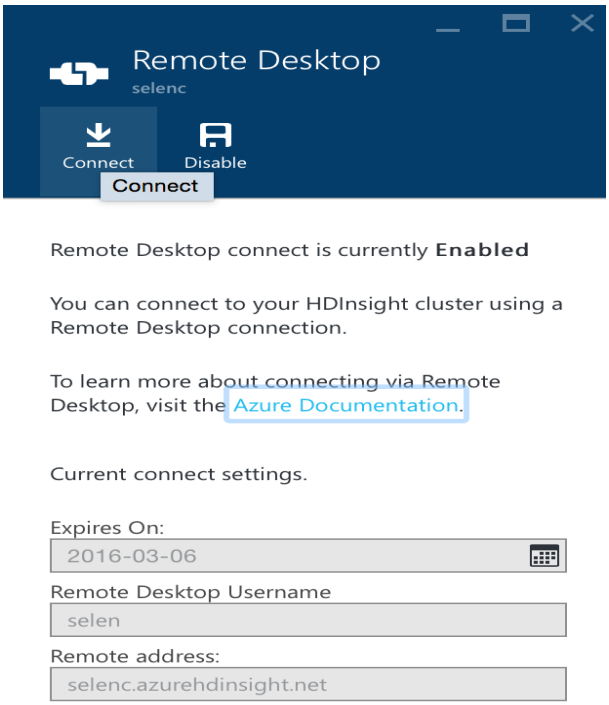
Şekil 4.12. İşlemin başlaması

İşlem bittikten sonra küme erişime hazır hale gelir. HDInsight master bilgisayarına ulaşmak için remote desktop butonuna tıklanır. Azure HDInsight sadece master bilgisayara erişime izin verir diğer bilgisayarlara erişim olmaz.



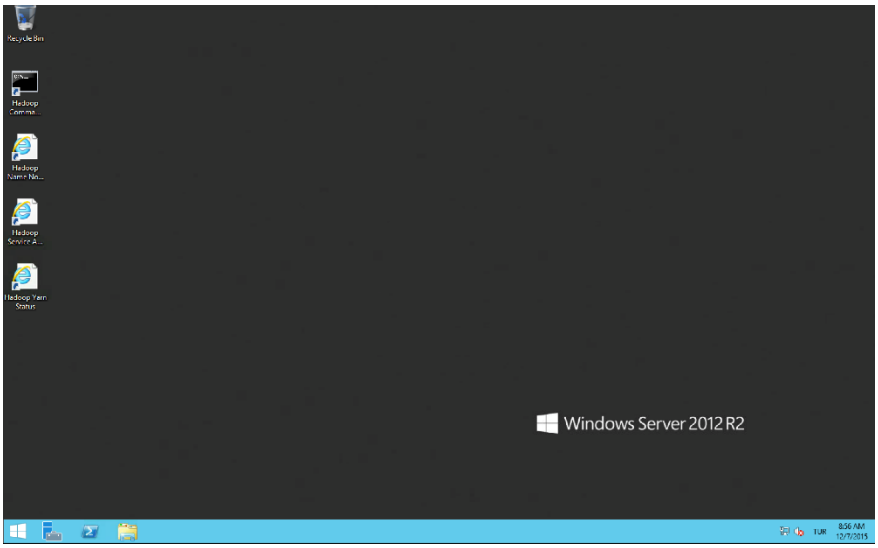
Şekil 4.13. Ekle butonu

Çıkan ekrandan Connect butonuna tıklanır. Bu ekranda kurulacak olan uzak masaüstü bağlantısının protokolleri de görülebilir.



Şekil 4.14. Bağlan butonu

İndirilen RDP dosyasına çift tıklanarak sunucuya erişilir.

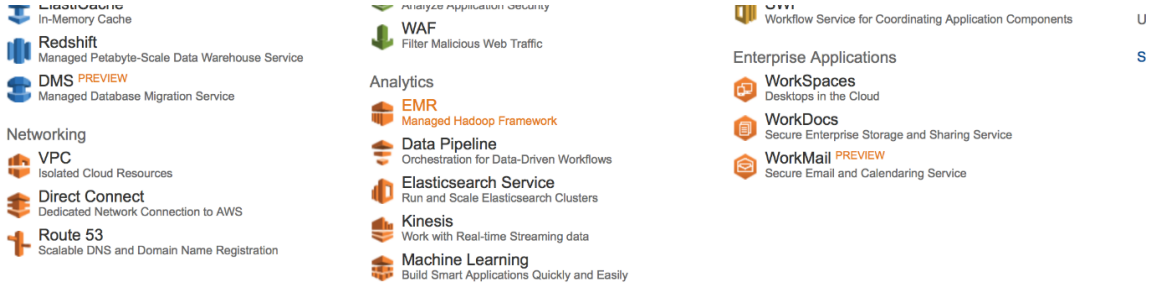


Şekil 4.15. Sunucuya erişimin sağlanması

4.6.Amazon AWS

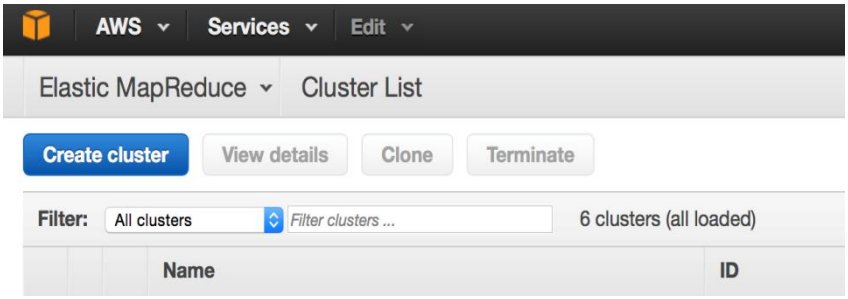
4.6.1.Amazon EC2 Cloud Üzerinde Hadoop Kümesi Oluşturma

Amazon'da Azure gibi bir çok servise hizmet sağlar. Hadoop kümesi oluşturmak için kullanılan servis olan EMR ile yeni bir küme oluşturmak için Console ekranından EMR seçilir.



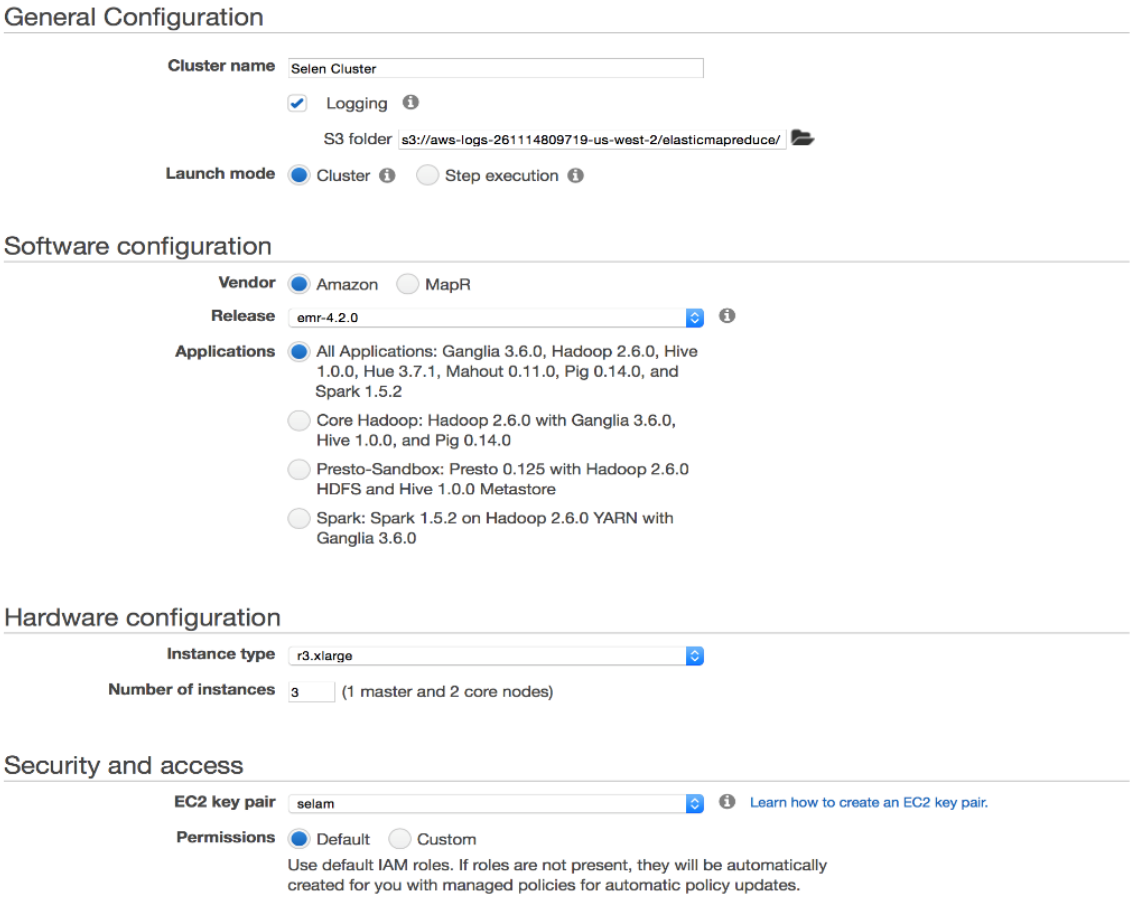
Şekil 4.16. Amazon konsol ekranı

Çıkan ekranda var olan kümeler görülebilir. Yeni küme oluşturmak için Create Cluster butonuna tıklanır.



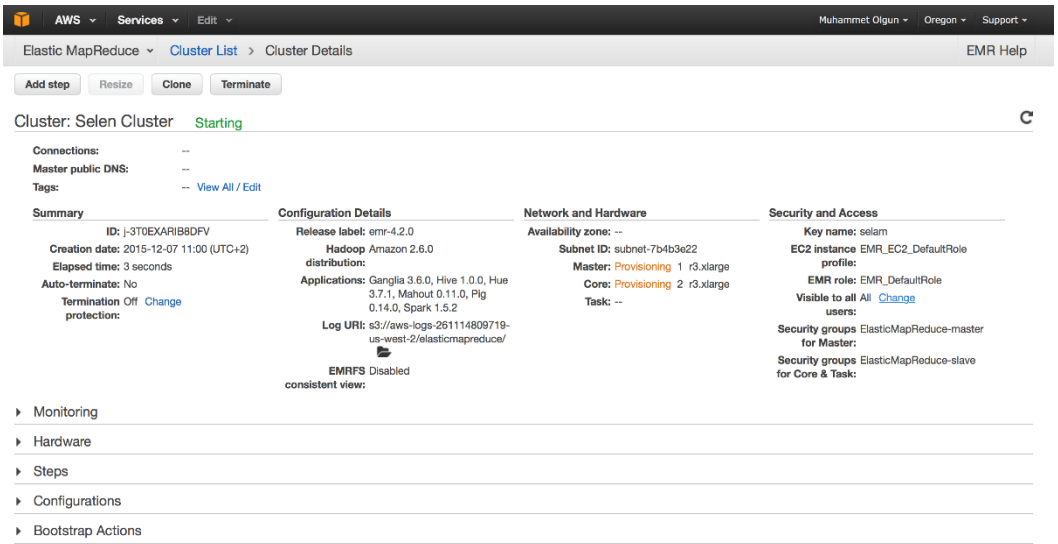
Şekil 4.17. Küme oluşturma

Çıkan ekranda EMR hizmeti oluşturulacak olan kümenin ayarlarını istemektedir. Oluşturulacak olan kümede hangi teknolojilerin kurulu olacağı, kümenin kaç bilgisayardan oluşacağı ve özellikleri ve giriş bilgileri bu ekranda girilir.



Şekil 4.18. Bilgilerin girilmesi

Gerekli bilgiler girildikten sonra küme deploy edilir. Oluşturulan kümenin durumu gerçek zamanlı olarak görülebilir.



Şekil 4.19. Küme bilgileri

Küme oluşturulduğunda key pair ile kümeye giriş yapılır.

4.7.Zemberek

Açık kaynak kodlu doğal dil işleme kütüphanesi Zemberek bir OpenOffice ve LibreOffice eklentisi olup java ile geliştirilmiştir. Platform bağımsızdır ve Zemberek kullanılarak kelime kök ve gövdeleri, edatlar, bağlaçlar gibi işlemler otomatik bir şekilde yapılmış olur. Zemberek ile karmaşık çözümleme işlemleri daha rahat yapılmaktadır [51].

4.8.Apache Mahout ve Apache Apark ile Naive Bayes Sınıflandırma Testleri

Yukarıda anlatılan platformlarda Hadoop kümeleri kurulumları yapıldıktan sonra ilk olarak Google Cloud üzerinde, Tübitak Dergipark'tan çekilmiş olan akademik makalelerin PDF dosyalarından metinlerin çıkarılması işlemi gerçekleştirilip, bu metinlerin hangi dilden olduğu tespit edilmiştir. Metinler öncelikle MongoDB NoSQL veritabanında kaydedilmiştir. PDF dosyalarından metin çıkarımı için Apache Tika kullanılmıştır..

Çıkarılmış olan içerik doğal dil işleme kütüphanesi olan zemberekten geçirilerek gereksiz kelimeler (stop-word) atılmıştır.

MongoDB'de tutulan metinler Hadoop üzerinde işlenebilmek için HDFS'e atılmıştır. TÜBİTAK Dergipark sitesinde kategorilerin her birisi bir sınıf olarak kabul edilmiş ve böylece beş sınıf elde edilmiştir. Sınıflandırılmamış dergiler veri setine eklenmemiştir. HDFS'e veriler atılırken her yayın kendi ismindeki klasörün altına kopyalanmıştır. HDFS'te var olan bir klasörün altına engineering, law, life, social, medicine olmak üzere beş adet sınıfı temsil eden klasörler ve onların altında da o sınıfa ait yayınlar atılmıştır. Bu şekilde aşağıdaki komutlar çalıştırılarak Naive Bayes sınıflandırma yapılmıştır:

```
./mahout seqdirectory -i /academic -o /academic-seq
./mahout seq2sparse -i /academic-seq -o /academic-vectors -lnorm -nv -wt tfidf
./mahout split -i /academic-vectors/tfidf-vectors --trainingOutput /academic-train-vectors --testOutput /academic-test-vectors --randomSelectionPct 40 --overwrite --sequenceFiles -xm sequential
./mahout trainnb -i /academic-train-vectors -el -o /academic-model -li /academic-labelindex -ow -c
./mahout testnb -i /academic-test-vectors -m /academic-model -l /academic-labelindex -ow -o /academic-testing –c
```

Sınıflandırma sonucuna bakıldığında toplam verinin %40'ı olan 19066 adet verinin teste girdiği, 28599 tane de eğitim verisinin olduğunu görülmektedir. Bununla beraber %89 doğruluk oranı ile programın çalıştığı gözlenmiştir.

```
=======================================================
Summary
-------------------------------------------------------
Correctly Classified Instances          :      17116    89.7724%
Incorrectly Classified Instances        :       1950    10.2276%
Total Classified Instances              :      19066

=======================================================
Confusion Matrix
-------------------------------------------------------
a     b     c     d     e     <--Classified as
2927  64    89    14    185    |  3279   a     = engineering
4     1     2     0     34     |  41     b     = law
81    213   5473  251   381    |  6399   c     = life
60    80    56    4879  364    |  5439   d     = medicine
17    17    9     29    3836   |  3908   e     = social

=======================================================
Statistics
-------------------------------------------------------
Kappa                                       0.8611
Accuracy                                   89.7724%
Reliability                                60.8491%
Reliability (standard deviation)            0.4638
Weighted precision                          0.9222
Weighted recall                             0.8977
Weighted F1 score                           0.9064

15/11/23 23:14:38 INFO MahoutDriver: Program took 52452 ms (Minutes: 0.8742)
```

Bu bölümde, Amazon Cloud üzerinde 3 adet düğümden oluşan bir Hadoop kümesi üzerinde Apache Mahout kullanılarak Naive Bayes Sınıflandırma yapılması anlatılmıştır. Komutlar çalıştırılana kadarki verilerin içeriklerinin çıkarılması, zemberekten geçirilmesi, her yayının isminin olduğu klasöre atılması gibi önceki bölümde anlatılanlar ile aynı olmak üzere aşağıdaki komutlar çalıştırılarak sonuç elde edilmiştir:

```
mahout seq2sparse -i /file.seq -o /spark-sparse -lnorm -nv -wt tfidf
mahout split -i /spark-sparse/tfidf-vectors --trainingOutput /spark-training --testOutput /spark-test --randomSelectionPct 40 --overwrite --sequenceFiles -xm sequential
mahout spark-trainnb -i /spark-training -o /spark-model -ow
mahout spark-testnb -i /spark-test -m /spark-model
```

Sonuca bakıldığında %91 doğru sınıflandırma oranı tespit edilmiş oldu.

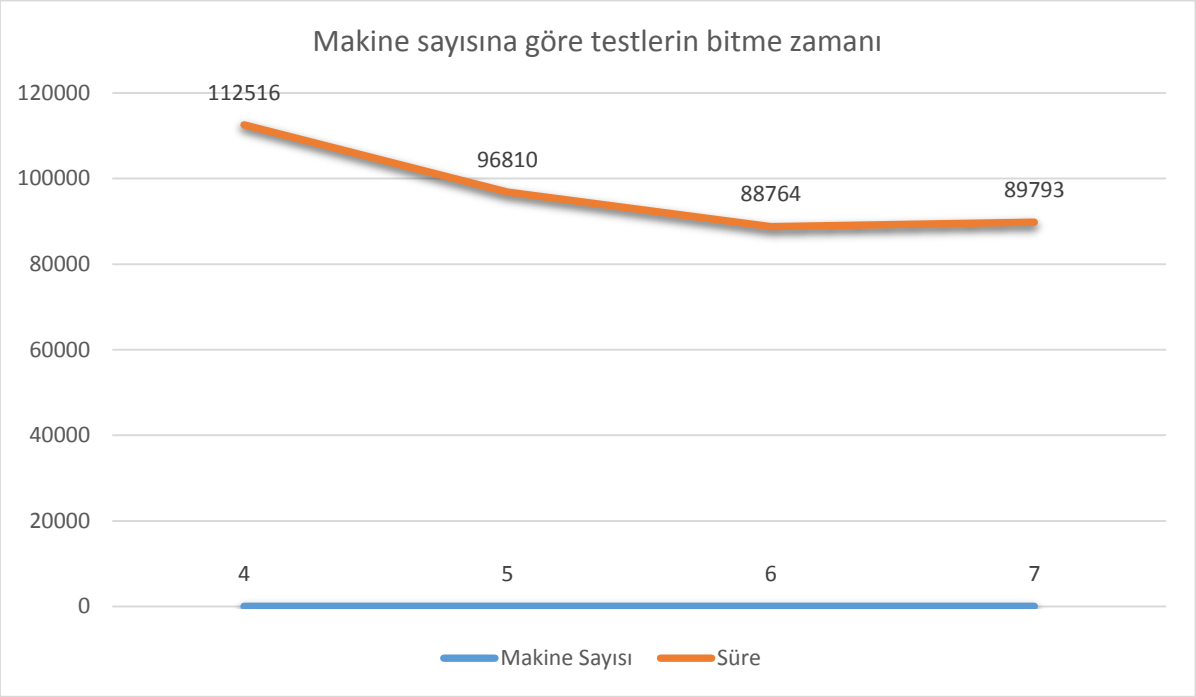
```
=======================================================
Summary
-------------------------------------------------------
Correctly Classified Instances: 17445 91.4836%
Incorrectly Classified Instances: 1624 8.5164%
Total Classified Instances: 19069

=======================================================
Confusion Matrix
-------------------------------------------------------
d      c      a      b      e      <--Classified as
4941      85      98      10      326       |  5460       d      = medicine
214      5770      106      19      299       |  6408       c      = life
17      102      2986      9      163       |  3277       a      = engineering
0      1      4      9      29       |  43       b      = law
43      16      39      44      3739       |  3881       e      = social

=======================================================
Statistics
-------------------------------------------------------
Kappa:  0.8832
Accuracy:  91.4836%
Reliability:  64.8216%
Reliability (std dev):  0.4268
Weighted precision:  0.9218
Weighted recall:  0.9148
Weighted F1 score:  0.9166
```

Tablo 4.4. Beş kategori ile sınıflandırma sonucu testlerin bitme süresi

Kümedeki Makine Sayısı	Testlerin Bitme Süresi (ms)
4	114577
5	109275
6	90721
7	99973

3 düğümden oluşan küme üzerinde Apache Mahout ile Naive Bayes Sınıflandırma yapılıp yüksek başarı oranı elde edilmiştir. Aynı şekilde Google Cloud üzerinde bu kez makine sayıları değiştirilerek testler yapılmıştır. 4 makine için testlerin bitme süresi 114577ms iken makine sayısı arttıkça testlerin bitme süresinin kısaldığı görülmüştür. Tablo 4.5'teki

değerler ile makine sayısına göre testlerin bitme zamanı grafiği oluşturulduğunda Şekil 4.21'deki değişim gözlenmiştir .

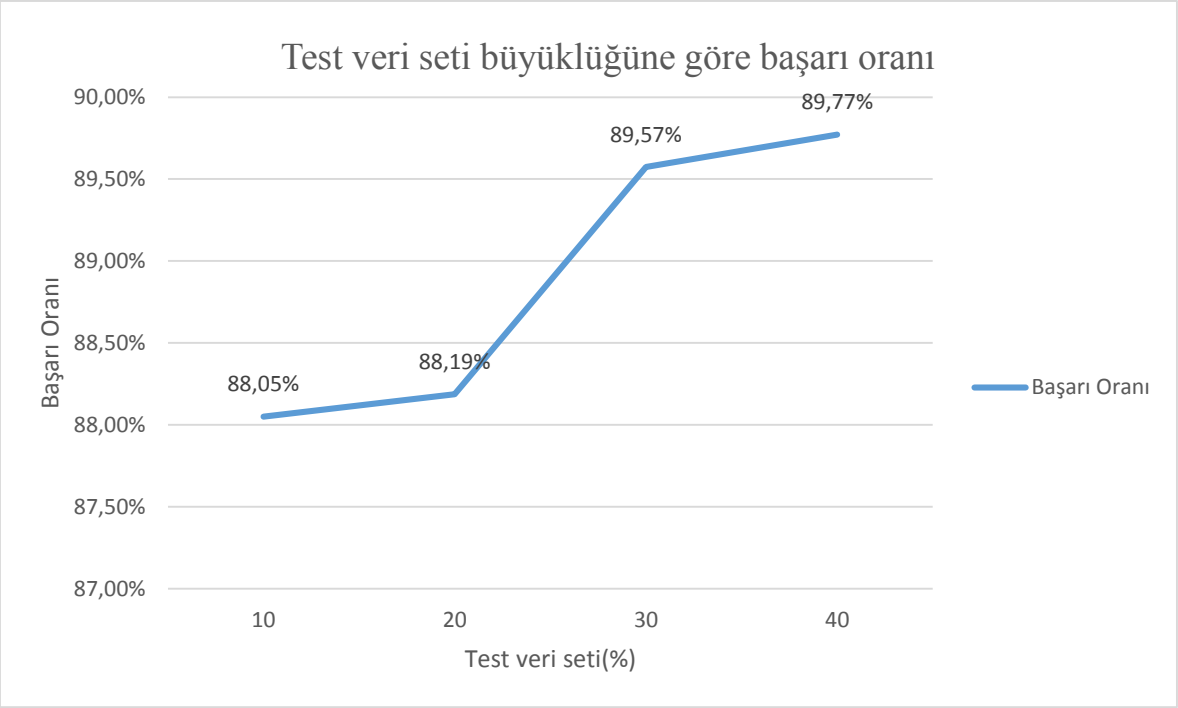


Şekil 4.20. Beş kategori ile sınıflandırma sonucu testlerin bitme süresi grafiği

Tablo 4.5. Test veri sayısına göre sınıflandırma başarı tablosu

Test veri seti (%)	**Başarı Oranı (%)**
10	88.0504
20	88.1862
30	89.574
40	89.7724

Apache Mahout kullanılarak, test veri setinin tüm veri setine oranı %40 iken sınıflandırma yapıldığında %89.7724 başarı oranı elde edilmişti. Test veri seti yüzdesi değiştirildiğinde başarı oranının da değiştiği gözlenmiştir. Örneğin Tablo 4.6'da test veri seti %10 iken başarı oranı %88.050 olmuş, %20 iken başarı oranı %88.1862 ve %30 iken %89.574 şeklinde test veri seti oranı arttıkça başarı oranının da arttığı sonucuna ulaşılmıştır.

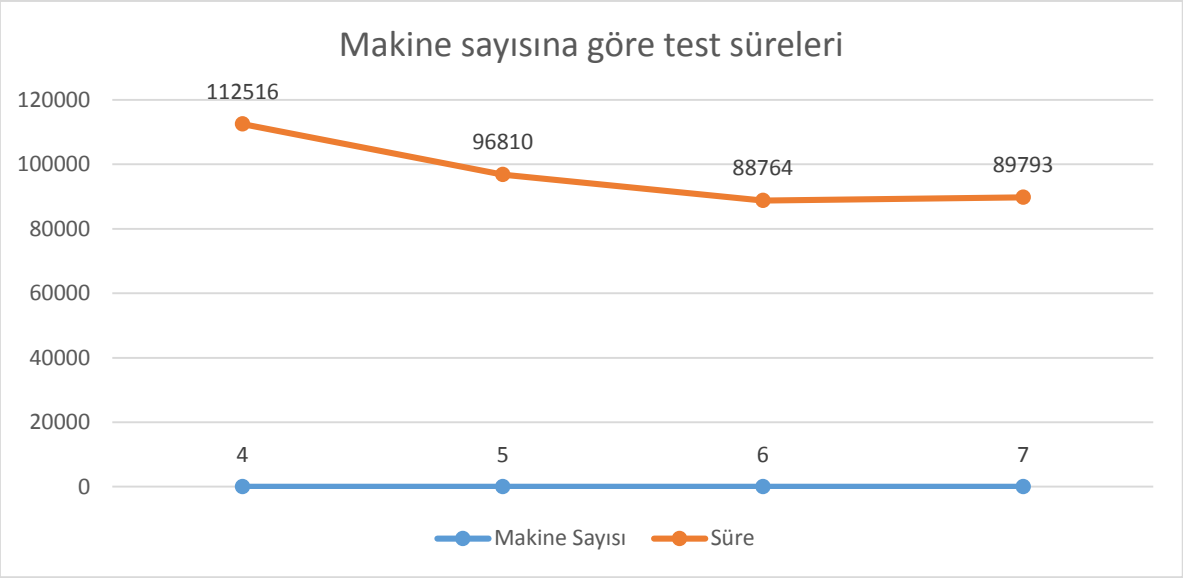


Şekil 4.21. Test veri sayısına göre sınıflandırma başarı grafiği

Tablo 4.6. Spark beş kategori ile sınıflandırma sonucu testlerin bitme süresi

Makine Sayısı	**Testlerin Bitme Süresi**
4	112516ms
5	96810ms
6	88764ms
7	89793ms

Amazon Cloud üzerinde 3 düğümden oluşan küme ile, Apache Spark kullanılarak Naive Bayes Sınıflandırma yapılıp %91.4836 başarı oranı elde edilmiştir. Ayrıca Amazon Cloud üzerinde oluşturulan kümedeki makine sayıları değiştirilerek testler yapılmıştır. Aşağıdaki grafikte görüldüğü gibi 4,5,6 ve 7 düğümden oluşan kümeler kullanılarak yapılan sınıflandırma testleri süreleri, makine sayısı arttıkça azalmaktadır.



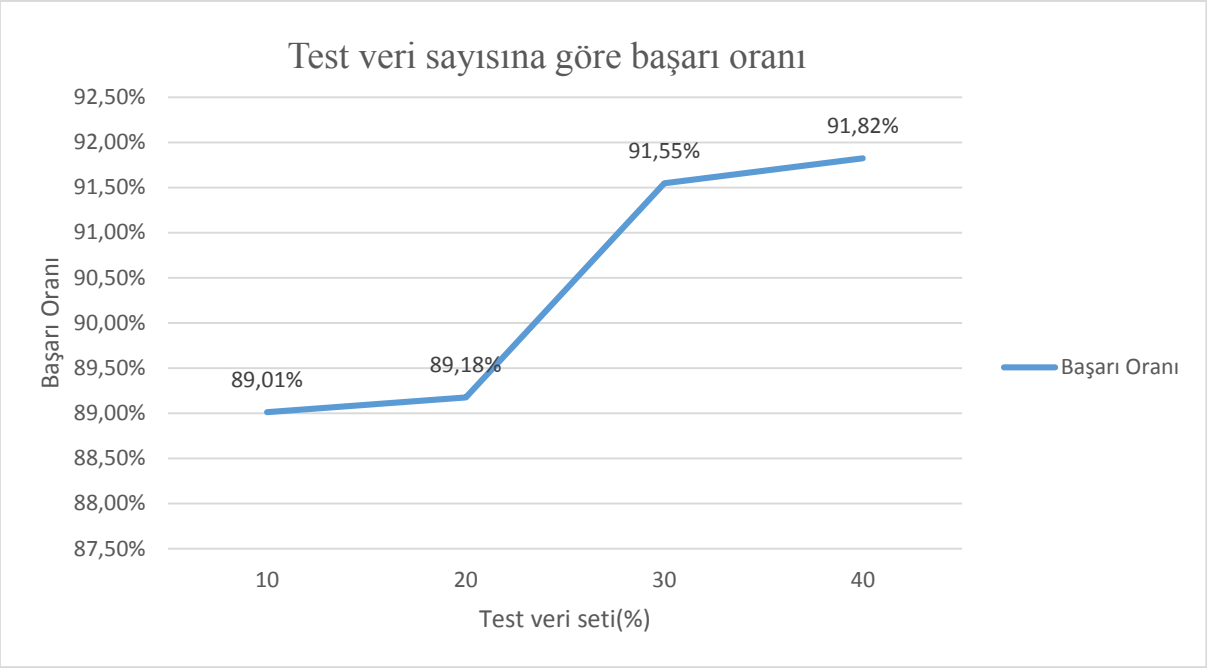
Şekil 4.22. Apache Spark beş kategori ile sınıflandırma süreleri

Apache Spark için test veri setinin toplam veri setine oranına göre başarıya bakılırsa:

Tablo 4.7. Apache Spark test veri sayısına göre sınıflandırma başarı tablosu

Test veri seti(%)	**Başarı Oranı(%)**
10	89.012
20	89.176
30	91.546
40	91.824

Apache Spark kullanılarak, 3 makineden oluşan bir küme üzerinde, test veri setinin toplam veri setine oranı değiştirildiğinde sınıflandırma başarısının değişimi için testler yapılmıştır. Buna göre, test veri seti yüzdesi arttıkça başarı oranının da aynı şekilde arttığı gözlenmiştir. Şekil 4.24'te görüldüğü üzere %10 test veri seti için %89.01 olan başarı oranı %20 test veri seti için %89.18 olmuştur.



Şekil 4.23. Spark test veri sayısına göre sınıflandırma başarı grafiği

Naive Bayes algoritması kullanılarak Apache Mahout için Google Cloud üzerinde Apache Spark için ise Amazon Cloud üzerinde oluşturulan kümeler ile sınıflandırma testleri yapılmış ve sonuçlar kıyaslandığında Apache Spark'ın Apache Mahout'a göre daha yüksek başarım gösterdiği gözlemlenmiştir. Mahout için %89 olarak elde edilen başarı oranının Spark için %91 olduğu görülmüştür.

4.9. Apache Mahout ile K-Means Kümeleme

Azure'da 6 makina ile çalıştırılmış olan k-means kümeleme algoritmasında aşağıdaki komutlar girildikten sonra elde edilen sonuca göre 10 iterasyonda 20 küme bulunmuş oldu.

./bin/mahout seq2sparse -i /file.seq -o /kmeans-sparse --maxDFPercent 85 --namedVector

./bin/mahout kmeans -i /kmeans-sparse/tfidf-vectors -c /kmeans-clusters -o /kmeans -dm org.apache.mahout.common.distance.EuclideanDistanceMeasure -x 10 -k 20 -ow --clustering

./bin/mahout clusterdump -i /kmeans/clusters-10-final/ -o /output -d /kmeans-sparse/dictionary.file-0 -dt sequencefile -b 100 -n 20 --evaluate -dm org.apache.mahout.common.distance.EuclideanDistanceMeasure -sp 0 --pointsDir /kmeans/clusteredPoints/

N küme sayısı c de kümenin merkezi olmak üzere çıkan sonuçlarda örneğin CL-38933 numaralı kümenin kelimelerine bakılarak eğitim bilimleri olduğu söylenebilir. Her kelimenin karşısındaki değer kümenin merkezine olan uzaklığı belirtmektedir.

öğrenci => 15.890060659754646
olmak => 14.33436497534627
öğretmen => 13.859277880849532
eğitim => 12.881112660021367

Kümeleme sonucuna bakıldığında Inter-Cluster Density ve Intra Cluster Density değerleri görülmektedir. Bu değerler kümelerin homojenlik-heterojenlik durumunu ve kümelerin birbirinden uzaklığını ifade eder. Kümeleme bu şekilde değerlendirildiğinde eğer küme daha homojense ve küme merkezleri birbirinden daha uzak kümeler elde ediliyorsa bu iyi kümeye işaret etmektedir [52].

Inter-Cluster Density: 0.42671156033106467
Intra-Cluster Density: 0.5857023526744929
CDbw Inter-Cluster Density: 0.0
CDbw Intra-Cluster Density: 66.25203156378485
CDbw Separation: 89762.75810470397

5. SONUÇ

Bu tez çalışmasında Bulut Bilişim konusu ele alınıp dağıtık makine öğrenmesi algoritmaları büyük verilere uygulanarak akademik makalelerin sınıflandırılması ve kümelenmesi amaçlanmıştır. Bunun için açık kaynak kodlu yazılımlar incelenip yeni ve popüler teknolojiler ile bunların veriler üzerine uygulanması yöntemleri üzerinde durulmuştur. Bulut üzerinde verileri işlemek için Apache Mahout ve Apache Spark teknolojileri ile uygulamalar gerçekleştirilmiştir.

Sağlık, sosyal, mühendislik, hukuk ve tıp olmak üzere beş kategoriye ayrılmış olan dokümanların Mahout ve Spark üzerinde otomatik olarak Naive Bayes algoritması ile sınıflandırılıp yine aynı verilerin K-Means algoritması ile kümelenmesi gerçekleştirilmiştir. Yapılan sınıflandırma ile kategorilerin tahmini ve performans anlamında yüksek başarı oranı elde edilmiştir.

Makine öğrenmesi algoritmaları incelenerek sınıflandırma işlemi için Naive Bayes kullanılmış olup Mahout ve Spark için performans karşılaştırılması yapılmıştır. Geleneksel yöntemler ile kümelenemeyecek kadar büyük olan verilerin kümeleme işleminin dağıtık olarak yapılabilirliği gösterilmiştir.

KAYNAKLAR

[1] Kantardzic, M. (2003). Data Mining: Concepts, Models, Methods, and Algorithms. New York: Wiley, a.g.e., s.125.

[2] Han, J. ve Kamber, M. , "Data Mining Concepts and Techniques 2nd ed.", Morgan Kauffmann Publishers Inc, 348 (Ağustos 2001).

[3] Zhijie Xu, Laisheng Wang, Jiangin Luo ve Jiangin Zhang, 'A Modified Clustering Algorithm for Data Mining', IGRSS'05 Proceedings 2005 IEEE Int., Vol.2, (2005), s.741

[5] Oswaldo T, Pjotr P, Marc S & Ritsert C. J., "Big data, but are we ready?", Nature Reviews Genetics 12, 224 (March 2011).

[6] Hoffa, C., Mehta, G., Freeman, T., Deelman, E., Keahey, K., Berriman, B., & Good, J. (2008, December). "On the use of cloud computing for scientific workflows". In eScience, 2008. eScience'08. IEEE Fourth International Conference on (pp. 640-645). IEEE.

[7] Ostermann, S., Iosup, A., Yigitbasi, N., Prodan, R., Fahringer, T., & Epema, D. (2010). "A performance analysis of EC2 cloud computing services for scientific computing." In Cloud Computing (pp. 115-131). Springer Berlin Heidelberg.

[8] Adsız, A. "Metin madenciliği", Ahmet Yesevi Üniversitesi Bilişim Sistemleri ve Mühendislik Fakültesi, Kazakistan, 17-19 (2006).

[9] Tanenbaum, Andrew S., and Maarten Van Steen. Distributed systems. Prentice-Hall, 2007.

[10] Available from :http://csis.pace.edu/~marchese/CS865/Lectures/Chap1/Chapter1a.htm,15 Aralık 2015.

[11] Wen, Xiaolong, et al. "Comparison of open-source cloud management platforms: OpenStack and OpenNebula." Fuzzy Systems and Knowledge Discovery (FSKD), 2012 9th International Conference on. IEEE, 2012.

[12] OpenStack Services. Available from: http://docs.openstack.org

[13] Official Hadoop Web Site. [cited 2015 17.12.2015]; Available from: http://hadoop.apache.org/

[14] Haines S. "Big Data Analysis with MapReduce and Hadoop",(2013).

[15] Gunarathne, T., Wu, T. L., Qiu, J., & Fox, G. (2010, November). MapReduce in the Clouds for Science. In Cloud Computing Technology and Science (CloudCom), 2010 IEEE Second International Conference on (pp. 565-572). IEEE

[16] White, Tom. Hadoop: The definitive guide. " O'Reilly Media, Inc.", 2012

[17] Dean, J., & Ghemawat, S. (2008). MapReduce: simplified data processing on large clusters. Communications of the ACM, 51(1), 107-113

[18] Haines S. "Big Data Analysis with MapReduce and Hadoop",(2013).

[19] Ho, R. Pragmatic Programming Techniques. http://horicky.blogspot.com.tr/ 2008/11/hadoop-mapreduce-implementation.html , 15 Aralık 2015 .

[20] Juve, Gideon, et al. "Scientific workflow applications on Amazon EC2." E-Science Workshops, 2009 5th IEEE International Conference on. IEEE, 2009.

[21] Cloud Computing. Available from: https://en.wikipedia.org/wiki/ Cloud_computing

[22] IAAS, PAAS, SAAS ve Windows Azure. Available from : http://daron.yondem.com/tr/post/IAAS_PAAS_SAAS_ve_Windows_Azure

[**23]** "Defining "Cloud Services" and "Cloud Computing"".IDC. 2008-09-23. Retrieved 2010-08-22.

[24] Cloud Computing Types. Available from : https://commons.wikimedia.org/wiki/ File:Cloud_computing_types.svg

[25] Lynch,C.,(2008). Big data: How do your data grow?. Nature 455, 28-29.

[26] BigData. Available from: www.deu.edu.tr/userweb/yilmaz.goksen/BigData.ppt

[27] Harrington, Peter. Machine learning in action. Manning, 2012.

[28] SOLMAZ, Ramazan, Mücahid GÜNAY, and Ahmet ALKAN. "Fonksiyonel Tiroit Hastalığı Tanısında Naive Bayes Sınıflandırıcının Kullanılması."

[29] Bayes Teoremi. Available from: https://tr.wikipedia.org/wiki/Bayes_teoremi

[30] Naive Bayes Sınıflandırıcı. Available from: http://tr.wikipedia.org/wiki/ Naive_Bayes_sınıflandırıcı

[31] Clustering Data Without Distance Functions, Ramkumar G.D. - Swami A., IEEE Bulletin ofthe Technical Committee on Data Engineering, Vol.21 No.l, March 1998 : 9-14.

[32] Özekes, Serhat. "Veri madenciliği modelleri ve uygulama alanları." (2003).

[33] Data Mining Concepts and Techniques, Han, J.-Kamber, M., Morgan Kaufmann Publishers, 1st Ed., San Francisco, USA, 2000.

[34] Çalışkan, Sibel Kırmızıgül, and İbrahim Soğukpınar. "K× KNN: K-MEANS VE K EN YAKIN KOMŞU YÖNTEMLERİ İLE AĞLARDA NÜFUZ TESPİTİ." *EMO Yayınları* (2008): 120-124.

[35] Orhan, U. Makine Öğrenmesi. http://bmb.cu.edu.tr/uorhan/DersNotu/Ders03.pdf , 17 Aralık 2015.

[36] B. Diri, "Makine Öğrenmesi Ders Notları," Makine Öğrenmesi Ders Notları.

[37] Teknomo, Kardi. "K-means clustering tutorial." *Medicine* 100.4 (2006): 3.

[38] Hadoop Kurulumu. [cited 2015 18.12.2015]; Available from: http://blog.bahadir.me/big-data/ubuntu-13-10-uzerine-hadoop-2-2-0-kurulumu/

[39] Hadoop ve Mahout ile BigData İşleme. [cited 2015 18.12.2015] http://koddit.com/veri-madenciligi/hadoop-ve-mahout-ile-big-data-isleme/

[40] Spark Kurulumu. [cited 2015 18.12.2015];Available from: http://bigdataakademi.com/author/bigdataakademi/

[41] Mahout. **[**cited 2015 17.12.2015]; Available from: http://mahout.apache.org

[42] Perera, Srinath. *Hadoop MapReduce Cookbook*. Packt Publishing Ltd, 2013.

[43] Spark. [cited 2015 18.12.2015]; Available from: http://spark.apache.org/

[44] GoogleCloud. **[**cited 2015 18.12.2015];Available from: https://cloud.google.com/ why-google

[45] Google Cloud Hizmetleri [cited 2015 17.12.2015];Available from:http://microsoft-news.com/

[46] Microsoft Azure. [cited 2015 18.12.2015]; Available from: https://tr.wikipedia.org/wiki/Microsoft_Azure

[47] Microsoft Azure Hizmetler. [cited 2015 18.12.2015];Available from: https://azure.microsoft.com/tr-tr/services/

[48] Microsoft Azure Nedir. [cited 2015 18.12.2015];Available from: http://www.webtekno.com/microsoft/microsoft-azure-nedir-h3972.html

[49] Microsoft Azure. [cited 2015 18.12.2015];Available from: https://azure.microsoft.com/tr-tr/

[50] Azure Services Platforms. [cited 2015 18.12.2015];Available from: https://news.microsoft.com

[51] Zemberek. [cited 2015 18.12.2015]; Available from: https://code.google.com/p/zemberek/.

[52] Anil, Robin, Ted Dunning, and Ellen Friedman. *Mahout in action*. Shelter Island: Manning, 2011.

ÖZGEÇMİŞ

Selen GÜRBÜZ

Cumhuriyet Mah. Arif Nihat Asya Sok.No:15/10, ELAZIĞ

Cep (539) 3508667

E-posta:selen.esen1809@gmail.com

EĞİTİM BİLGİLERİ

2012-	**Fırat Üniversitesi, Elazığ**
	Yüksek Lisans, Bilgisayar Mühendisliği
2007-2012	**Fırat Üniversitesi, Elazığ**
	Lisans, Bilgisayar Mühendisliği
2001-2005	**Elazığ Anadolu Lisesi**

İŞ DENEYİMİ VE STAJLAR

20.07.09-15.08.09	**Türkiye Radyo Televizyon Kurumu, ANKARA**
	Stajyer- Yazılım
19.07.10-14.08.10	**FONET Yazılım, ANKARA**
	Stajyer-Yazılım

BİLGİSAYAR BİLGİSİ

MapReduce, Spark, Shark

Java, JSP, Servlets

C/C++/Java/C#/Bash scripting

JavaScript; OpenLayers and ExtJS Libraries

YABANCI DİL İngilizce (Orta)