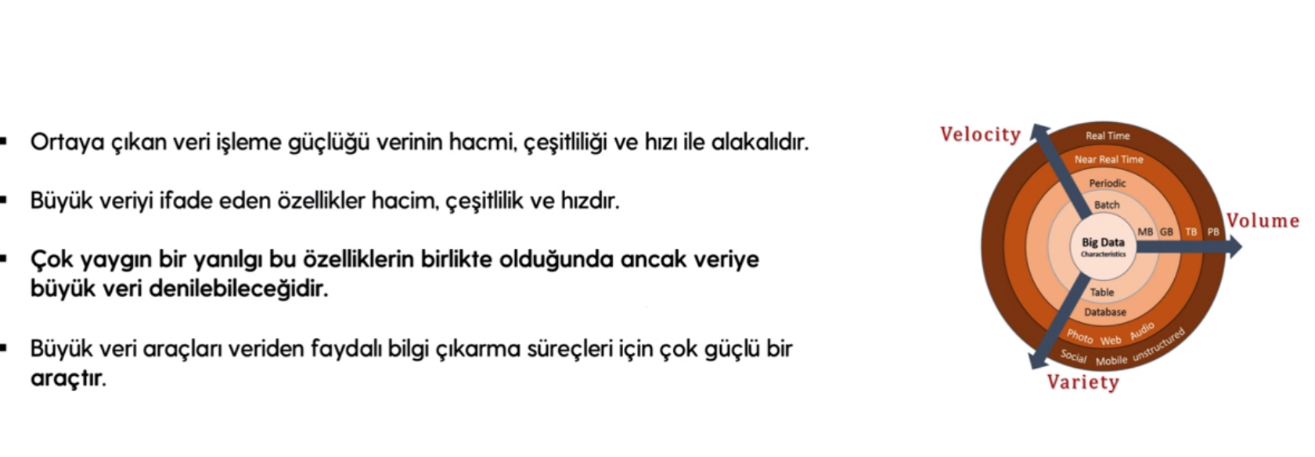
**Büyük Veri Nedir?**

**Geleneksel yöntemlerle işlenemeyen verilere Büyük Veri adı verilir.** Büyük Veri teknik olarak bir problemdir. Yani kabaca, veri boyutunun artması ve türünün çeşitlenmesi ile ilgili problemdir. Bu problemin çözümü birden fazla bilgisayarın bir araya gelerek tek bir bilgisayar gibi hareket etmesidir. Birlikte hareket edebilmesi içinde bir yazılım programlama modeli olması gerekmektedir. (MapReduce, RDD vb) Paralel ve dağıtık bir şekilde birden fazla bilgisayarın bir arada çalışabilmesini sağlayan yazılımlardır.

**Bu Problemin Çözülmesi Bizlere Ne Kazandırdı?**

Veri analitiği alanında yeni ufuklar açtı, hesaplama gücünün artmasıyla makine öğrenmesi algoritmalarının performansı arttı. Daha büyük miktar ve çeşitteki verilerin kullanılması ile veriden faydalı bilgi çıkarma süreci için çok önemli bir kaynak/araç sağlanmış oldu.

**Büyük Veri Bileşenleri**

****

**Apache Hadoop**

**Apache Hadoop** açık kaynak kodlu, güvenilir, ölçeklenebilir paralel hesaplama yazılım projesidir. Büyük veri denildiği zaman akla gelmesi gerek ilk bileşenlerden bir tanesidir.

. Büyük veri dünyasının temelini oluşturmaktadır.

. Geleneksel yöntemler ile etkin olarak işlenmesi mümkün olmayan verilerin işlenebilmesini sağlar.

. Bir bilgisayar kümesinin belirli bir işi yapmak için tek bir bilgisayar gibi birlikte hareket etmesini sağlamaktadır.

**Apache Hadoop Bileşenleri**

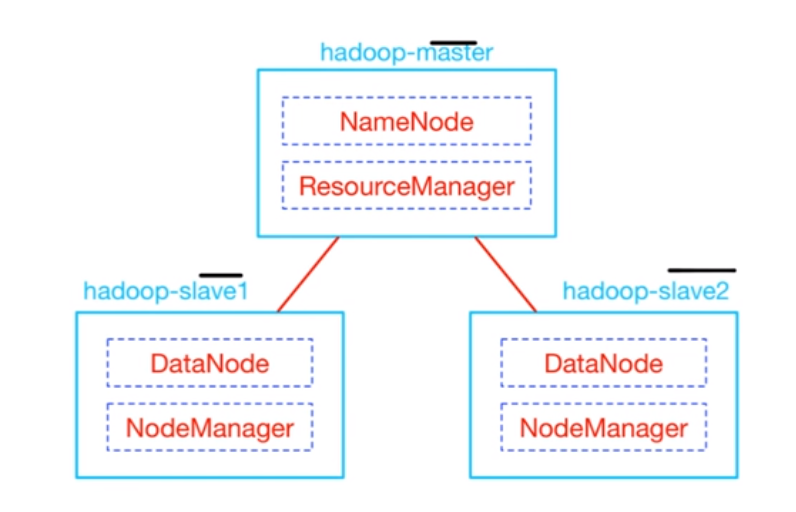
**. Hadoop Common:** Büyük veri teknolojileri, diğer bir ifadesi ile tüm hadoop modüllerini destekleyen ortak gereksinimdir.

**. HDFS (Hadoop Dağıtık Dosya Sistemi):** Yerel dosya sistemleri olan FAT32 ve NTFS gibi bir dosya sistemidir. Bu sistemlerden farkı büyük boyutlardaki veriyi dağıtık şekilde depolamaya ve kontrol etmeye imkan sağlamaktadır.

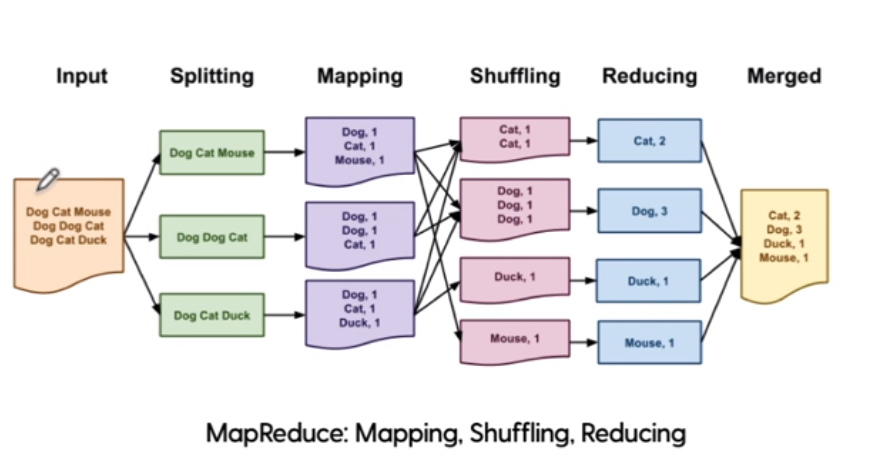
**. Hadoop YARN:** Kaynak yönetimi ve iş planlaması için kullanılan Apache Hadoop’un daha etkin bir şekilde kullanılmasına olanak sağlayan bir bileşendir.

**. Hadoop MapRedece:** Büyük veri dünyasının fonksiyonel anlamda temelini oluşturan bir bileşendir. Aynı ağdaki dağıtık bilgisayar kümeleri üzerinde büyük veri analizi yapılabilmesi için geliştirilmiş bir programlama modelidir.

**Apache Hadoop Küme Yapısı**



**MapReduce Süreci**



Yukarıdaki örnek yapı üzerinden **MapReduce** sisteminin nasıl çalıştığını değerlendireceğiz. Bir **Input** içerisinde bazı hayvan isimleri verilmiş ve görevimiz hangi hayvandan kaç tane olduğunu saymak istiyoruz. Buna yönelik olarak önce **Split** işlemi gerçekleşiyor ve her bir satır bölünüyor. Sonrasında her bir Split’te yer alan hayvanlar sayılmış. Bu aşama **Mapping** aşamasıdır. **Map** aşamasının görevi, HDFS üzerindeki girdi verilerini işlemektir. **Shuffle** basamağında ise Map sırasında yapılan işlemler biraz daha düzenli hale getirilecektir. **Shuffle ve Reduce** basamağında yapılan işlem, tanımlanmış olan hayvanların isimleri ve sayıları bir araya toplu şekilde getirilerek değerlendirme yapılmasıdır. Yani Map aşamasında gelen veriyi işlemek ve indirgemektir. Son basamak ise **Merge**’dir. Bu basamakta ise Reduce’de tamamlanan parça parça bölünmüş olan işlemlerin bir araya getirilmesidir. İşte tüm süreçlerin tamamına **MapReduce** adı verilir.

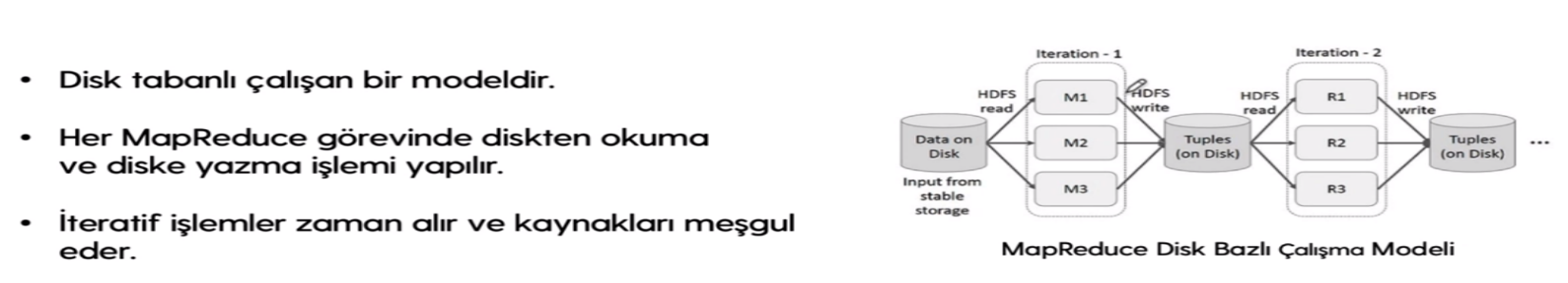
**Apache Hadoop Faydaları**

. Veri Saklama ve İşleme Gücü

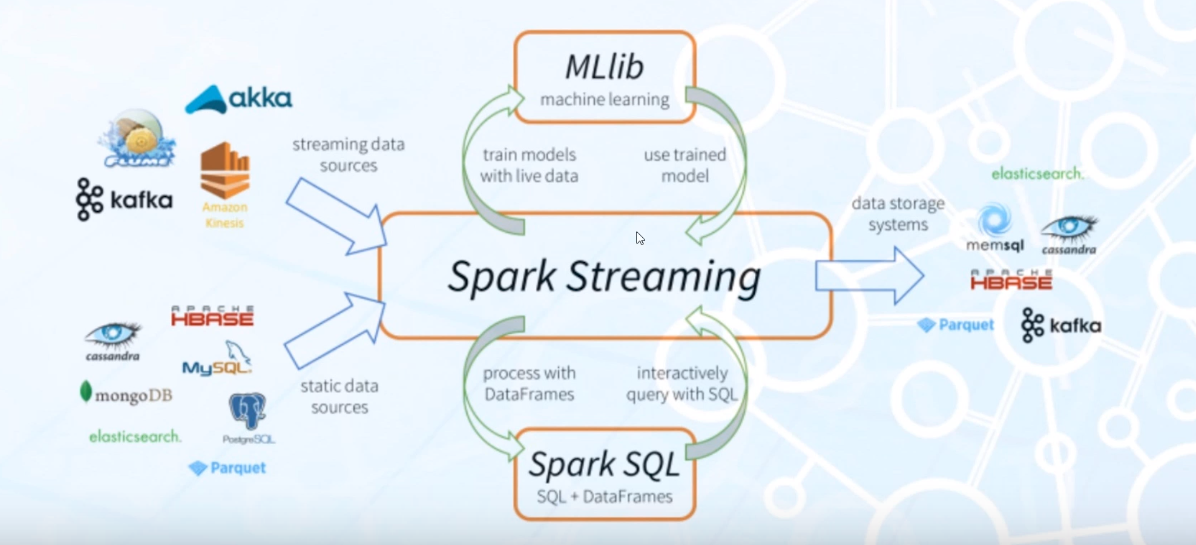
. Açık Kaynak

. Hız ve Esneklik

. Hata Toleransı ve Ölçeklenebilirlik



**Apache Spark**

****

Küme üzerinde hızlı ve genel amaçlı bilgi işleme sistemidir. MapReduce programlama modelinin alternatif olarak gelmiştir.

. MapReduce modelinde yer alan disk bazlı çalışma sisteminin yarattığı maliyetlerden dolayı çıkmıştır.

. Apache Hadoop’a göre 100 kat daha hızlı çalışmaktadır.

. Java, Scala, Python ve R ile uygulama geliştirilebilir.

. Genelleştiricidir. Spark SQL, Spark MLlib, Spark Streaming, GraphX aynı uygulamada kullanılabilir.

**Apache Spark Bileşenleri**

**. Spark Core** (Hafıza Yönetimi, Görevlerin Dağıtılması, Hata Kurtarma, Dosya Sistemlerine Erişim)

**. RDD’s** (MapReduce’nin alternatifidir. Apache Spark’ın programlama modeli verinin bellek içi tutularak paralel işlenmesini ifade etmektedir.)

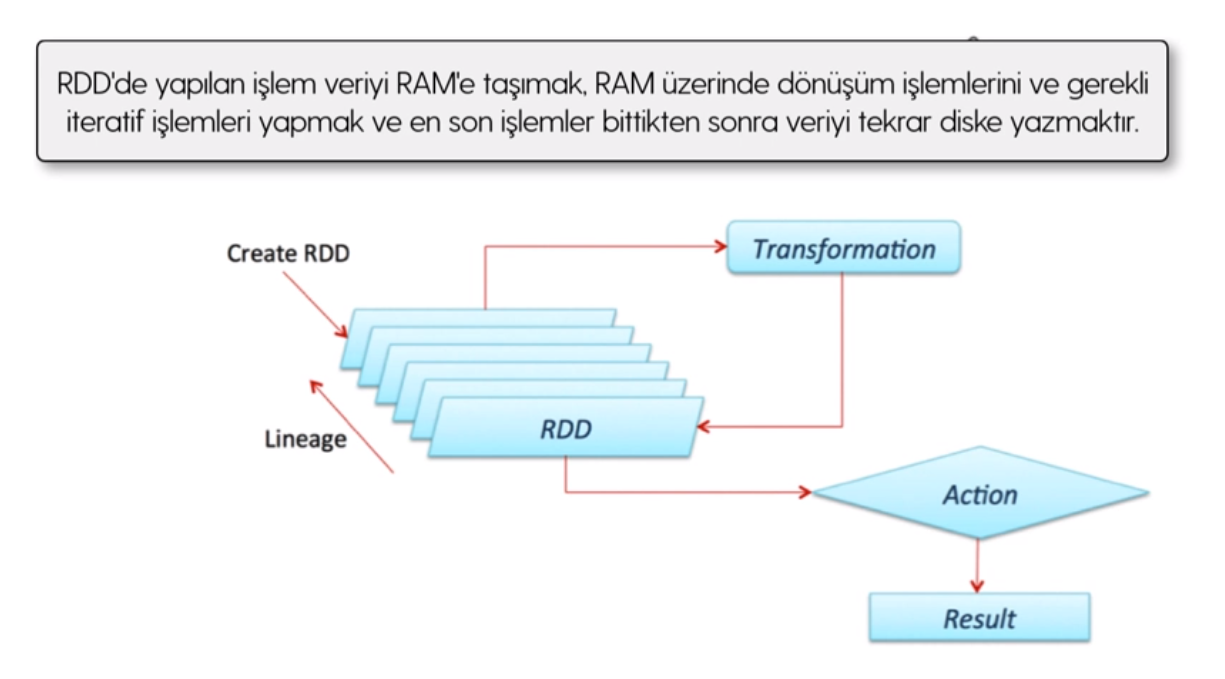
**. Spark SQL** (SQL ya da bir Dataframe API kullanarak Spark programlarında yapılandırılmış veriyi sorgulamaya yarar.)

**. Spark MLlib** (Ölçeklenebilir makine öğrenmesi kütüphanesidir. Apache Spark’ın bellek içi çalışma mantığından dolayı iteratif işlemler barındıran, makine öğrenmesi gibi işlemlerde bize çözüm sunmaktadır.)

**. Spark Streaming** (Akan verinin ölçeklenebilir, yüksek hacimli ve hata toleranslı bir şekilde işlenmesine olanak sağlayan temel Spark API’sine ait bir uzantıdır.)

**. GraphX** (Paralel tabanlı hesaplama işlemleri için kullanılan bir kütüphanedir.)

**RDD’s (Dayanıklı Dağıtık Data Setler)**

****

Verinin bellek içi tutularak paralel olarak işlenmesini ifade ediyor. RDD’s 3 aşamadan oluşmaktadır.

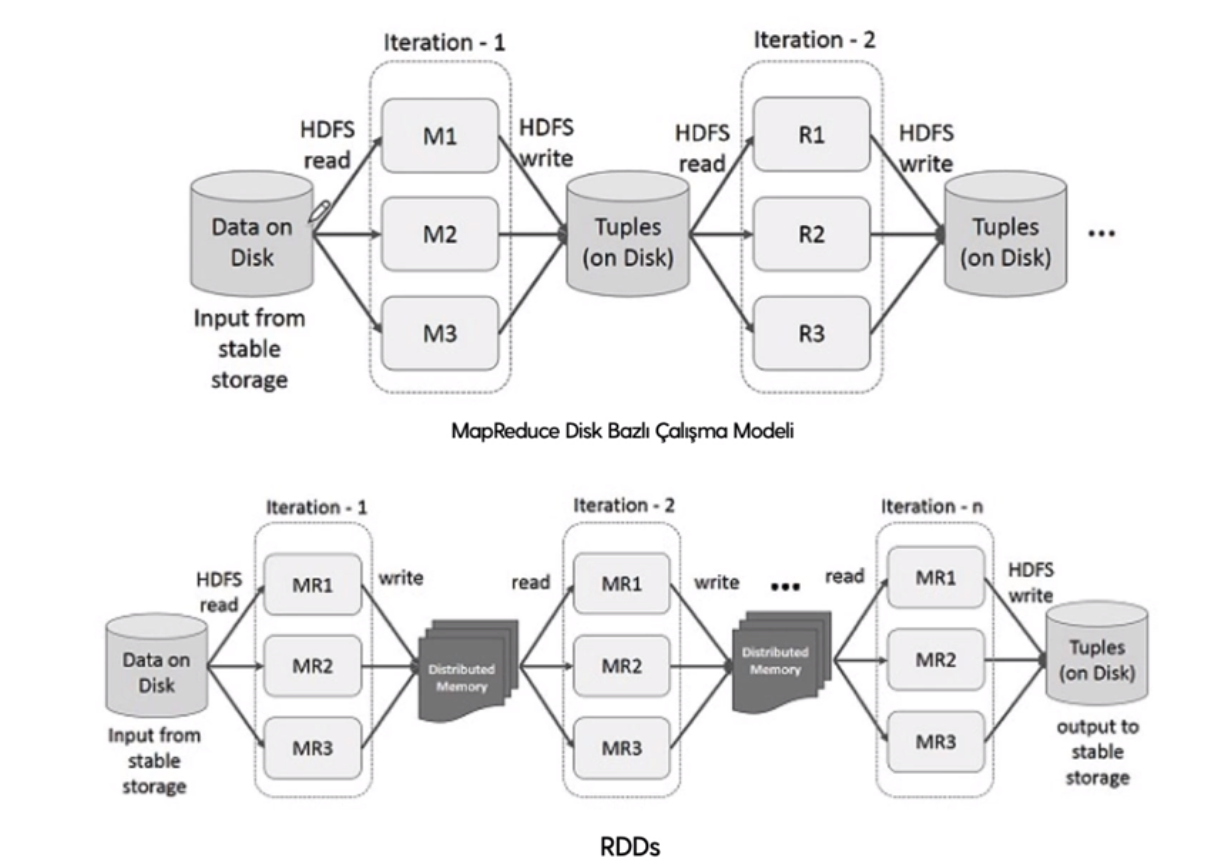
**.Create RDD’s, (RDD’leri oluşturmak)**

**.Transformation (Dönüşüm)**

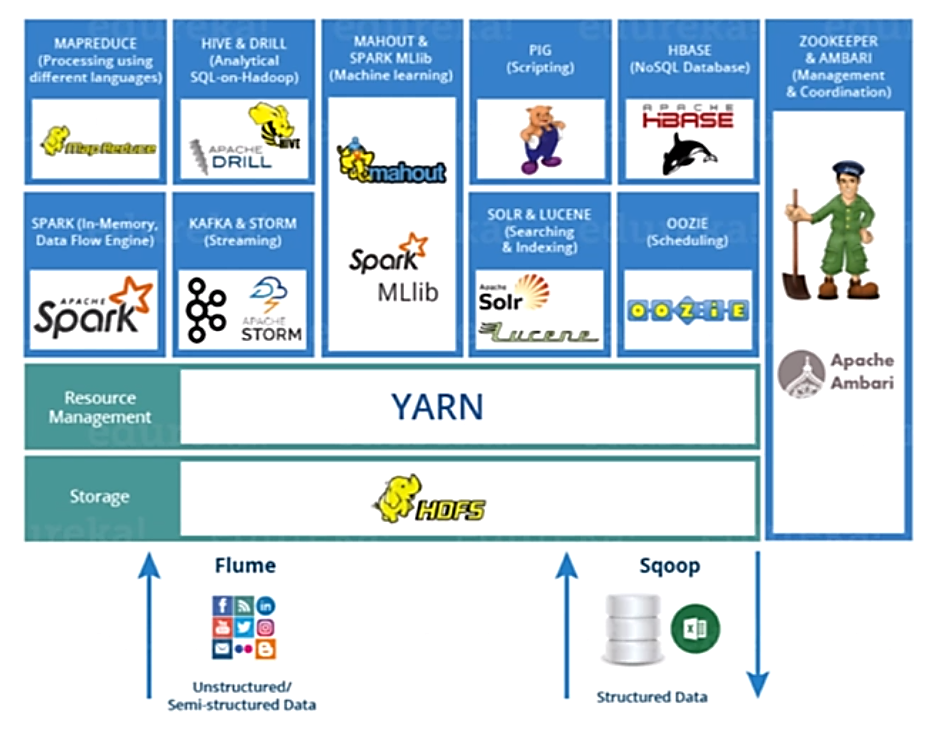
**. Action (Aksiyon)**

**Önemli Not:** Aksiyon basamağı tetiklenmeden uygulama işlemi (Execution) gerçekleşmemektedir. Bundan dolayı Spark’ın bu çalışma prensibi tembel çalışma (Lazy Evoliation) olarak adlandırılmaktadır.

**MapReduce vs RDDs**



**Ekosistemin Diğer Bazı Üyeleri**



**Windows Kurulum İşlemleri**

1. **Java Development Kit** indir, dosyayı **Yönetici Olarak** çalıştırma işlemini başlat ve kurulumu tamamla.

2. **Apache Spark** sitesine girerek güncel olan sürümü seçerek http başlığı altındaki linkten indirme işlemini gerçekleştiriyoruz. Daha sonra indirilen dosyayı ayıkladıktan sonra içindeki tüm dosya ve klasörleri kopyalayarak **C:** dizini altında yeni bir **Spark** klasörü oluşturarak içeresine atıyoruz.

3. **C:** dizini içerisinde **winutils** atında bir klasör oluşturup içerisine **bin** adında bir klasör daha oluşturuyoruz. Şimdi winutils adını verdiğimiz programın indirmemiz gerekmektedir. (Sourceforge.net) İndirilen klasörü oluşturduğumuz klasör altındaki bin içerisine yapıştırıyoruz.

4. İndirilen bazı dosyalar ile ilgili ortam değişkenleri ayarları yapmamız gerekmektedir. (Java, Hadoop) Bunun için **Bilgisayar 🡪Özellikler 🡪Gelişmiş Sistem Ayarları 🡪 Ortam Değişkenleri** dedikten sonra **Yeni** diyerek;

**. JAVA\_HOME** (Dizine Gözat diyerek Program Files içinden indirilen Java sürümünü seç.)

**. SPARK\_HOME** (C: dizini altında oluşturulan spark klasörünü seç.)

**. HADOOP\_HOME** (C: dizini altında oluşturulan winutils klasörünü seç.)

5. **Ortam Değişkenleri** sayfası içerisinde bulunan **Path** kısmına iki kere tıklayarak gelen yeni ekranda **Yeni** dedikten sonra;

**%JAVA\_HOME%\bin** ve **%SPARK\_HOME%\bin** diyoruz ve Enter ile kaydediyoruz.

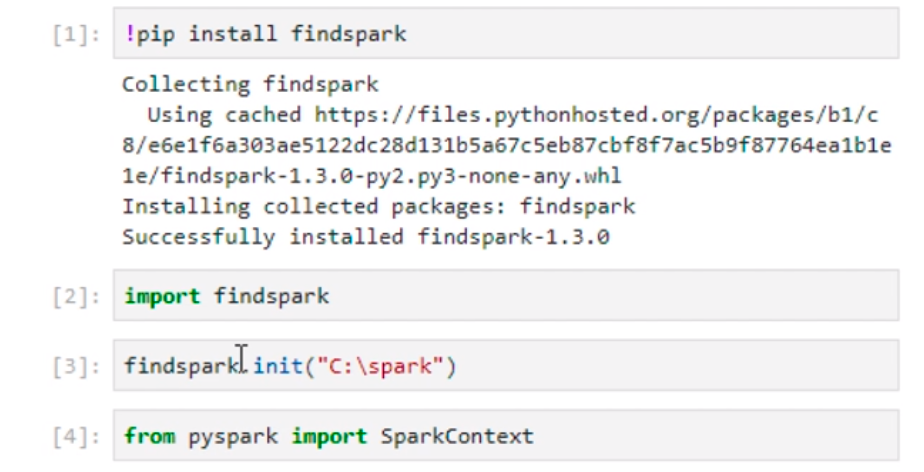
6. Tüm işlemleri yaptıktan sonra **Bilgisayar**’ımızı yeniden başlatıyoruz ve bağlantının yapılıp yapılmadığını kontrol etmiş olacağız.

7. **cmd Komut İstemi’ne** gelerek ilk olarak Spark’ın bulunduğu dosya konumuna gitmemiz gerekmektedir bunun için komut satırına **cd c: \spark** yazıyoruz sonra ise **spark-shell** diye çalıştırarak kontrol işlemini sağlıyoruz.

8. Son olarak Ortam Değişkenleri’ne Anaconda dizinini ekleyeceğiz ve kurulum işlemlerini tamamlayacağız.

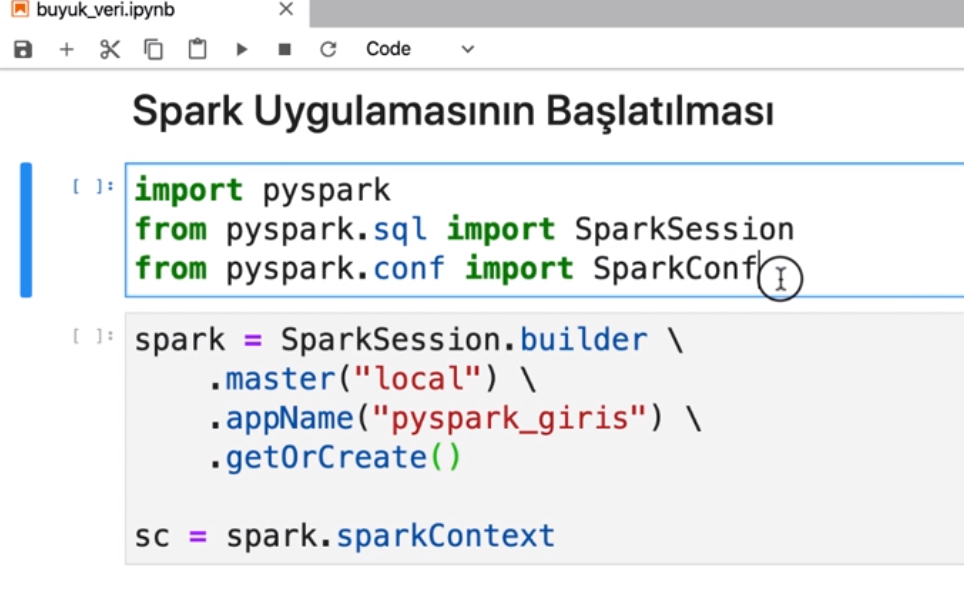
**Bilgisayar 🡪Özellikler 🡪Gelişmiş Sistem Ayarları 🡪 Ortam Değişkenleri** dedikten sonra **Path** içerisine gelerek **Yeni** diyerek sahip olduğumuz Kullanıcılar içerisinde bulunan **Anaconda3** dizininin dosya yolunu bu kısma yapıştırıyoruz. Daha sonra Anaconda3 içerisindeki **Script** klasörüne tıklayarak dosya yolunu alarak Path içerisine yapıştırıyoruz.

9. Şimdi de Anaconda uygulaması içerisinden JupyterLab tarafında Python ve Spark arasındaki bağlantıyı sağlayıp sağlayamadığımızı kontrol edeceğiz ve kurulumu tamamlayacağız. Anaconda içerisinden JupyterLab içerisine gelerek aşağıdaki işlemleri yapıyoruz.



**Büyük Veri Manipülasyonu ve Görselleştirmesi**

**Spark Uygulamasının Başlatılması**

****

İlk olarak J**upyterLab** içerisindeki kısmımıza gelerek yukarıdaki kısımların **import** işlemlerini yapıyoruz. Daha sonra çalışacağımız olan yeri, uygulamamıza bir isim vermek ve başlatma ile ilgili bilgiyi de tanımlayarak bağlantısı bilgisini ifade eden bir nesne oluşturarak çalıştırıyoruz.