Yıldız Teknik Üniversitesi
Elektrik-Elektronik Fakültesi
Bilgisayar Mühendisliği Bölümü

BLM4120

Büyük Veri İşleme Ve Analizi

Gr: 1

Doç. Dr. Mehmet Sıddık AKTAŞ

Dönem Projesi

İsim: Berke ÖZGEN

No: 20011602

E-posta: berke.ozgen1@std.yildiz.edu.tr

İsim: Muhammet Ali ŞEN

No: 20011701

E-posta: ali.sen@std.yildiz.edu.tr

1. Hedef

Projemizde, IMDb film incelemelerinden oluşan bir veri seti üzerinde Hadoop kullanarak 2 sanal makine ile çeşitli hesaplamaları yapmayı hedefledik. JSON formatında olan veri setimiz, 6 parça dosyadan oluşup, aşağıdaki verilen bağlantıda yer almaktadır. Veri seti üzerinde gerçekleştireceğiz işlemler şunlardır:

1. Her kullanıcının yaptığı toplam inceleme sayısı
2. Her filmin ortalama puanı
3. Her filmin minimum ve maksimum puanları
4. Her filmin verilen bir tarih aralığında yapılan incelemelerinin puan ortalamaları
5. Her filmin puanlarının standart sapmaları

Veri seti: https://www.kaggle.com/datasets/ebiswas/imdb-review-dataset

2. Kullandığımız Teknolojiler

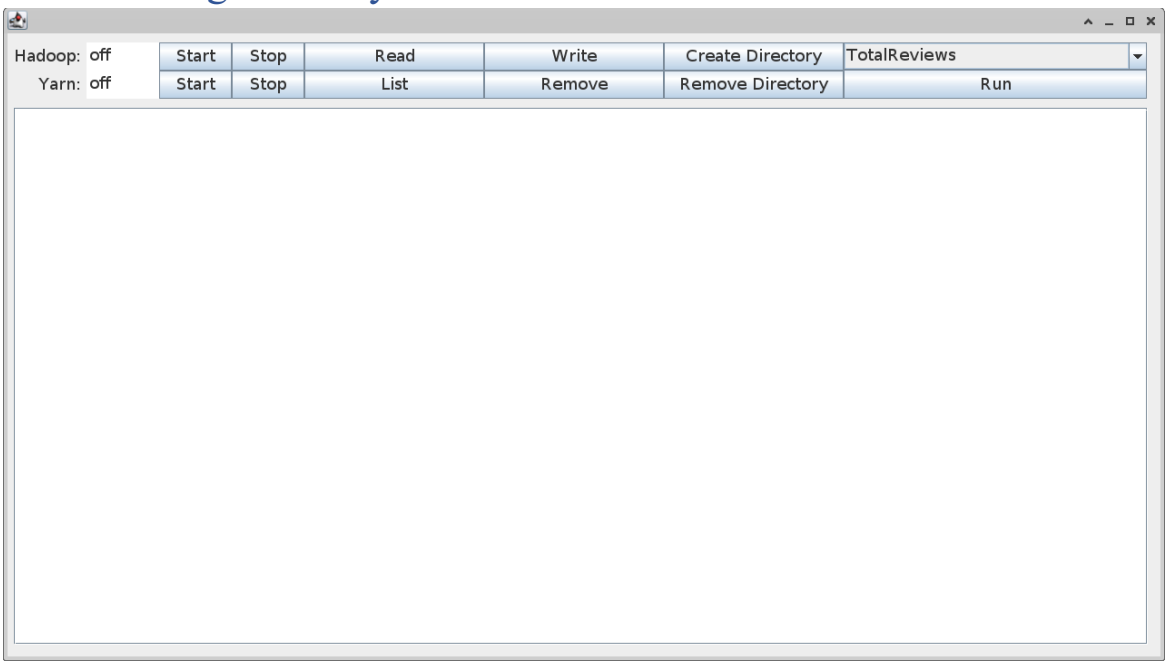
Projemizi, toplam 2 Node'da olacak şekilde, Microsoft Azure platformunda bir makine Master + Slave, diğer makine ise Slave olacak şekilde oluşturduk. Sanal makinelerimizde Linux İşletim Sistemini kullandık. Hadoop işlemlerini yapmak için Java dilini, GUI işlemleri için Swing kütüphanesini kullandık. Gui tasarımını Eclipse İDE'sinin Window Builder eklentisi ile gerçekleştirdik. Programımızı test etmek için RDP protokolü ile sanal makinemize erişim sağladık.

3. Kullanım Senaryosu

Programımız Hadoop kurulu bir makinede HDFS dosya işlemleri ve hadoop işlevlerini gerçekleştirmek için dizayn edilmiştir. Yarattığımız arayüz ile şunlar yapılabilmektedir;

1. Hadoop ve Yarn servislerini Aktif/Pasif duruma getirmek,
2. Hadoop Dosya Sisteminde;
    a. Okuma,
    b. Yazma,
    c. Dosya Listeleme,
    d. Dosya Silme,
    e. Klasör Oluşturma,
    f. Klasör Silme,
3. Hadoop İşlemlerini gerçekleştirmek
4. Terminal çıktısının görütülemek,
5. Hadoop işleminin çıktısını tablo şeklinde göstermek,

3.1. Program Arayüzü



Yazılan programımız sade bir arayüze sahiptir.

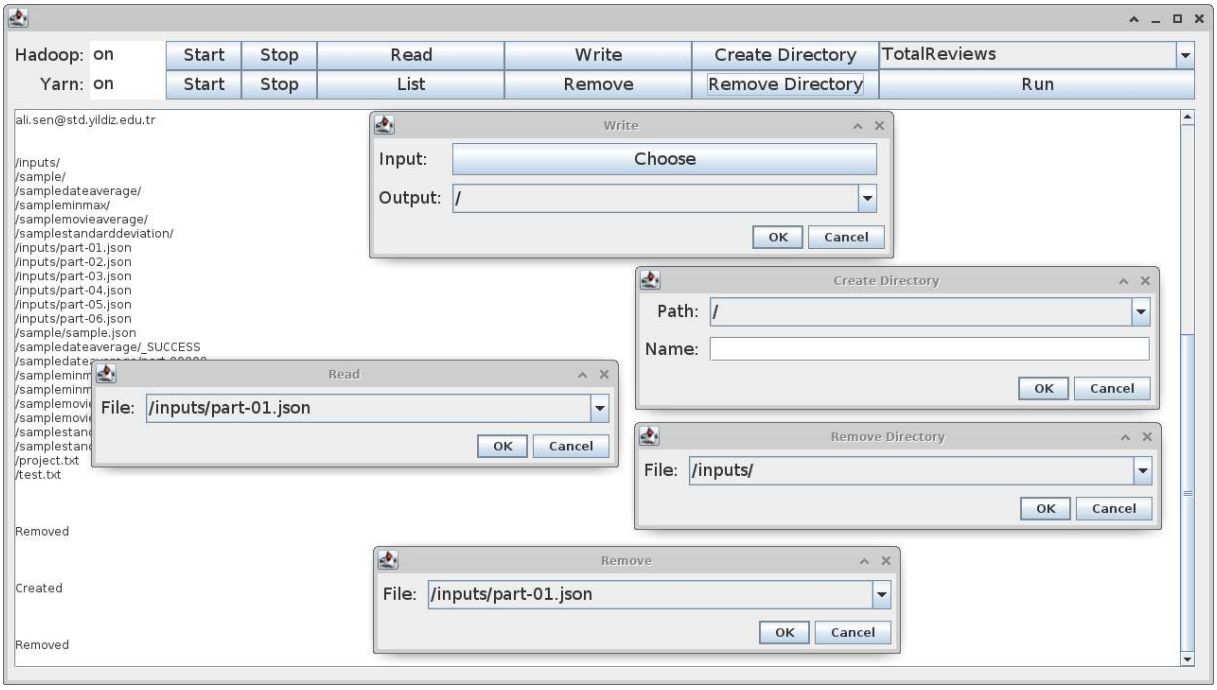
-Hadoop ve Yarn operatörlerinin başlama, durdurma,

-dosya okuma, yazma, listeleme, silme

-dizin oluşturma, silme

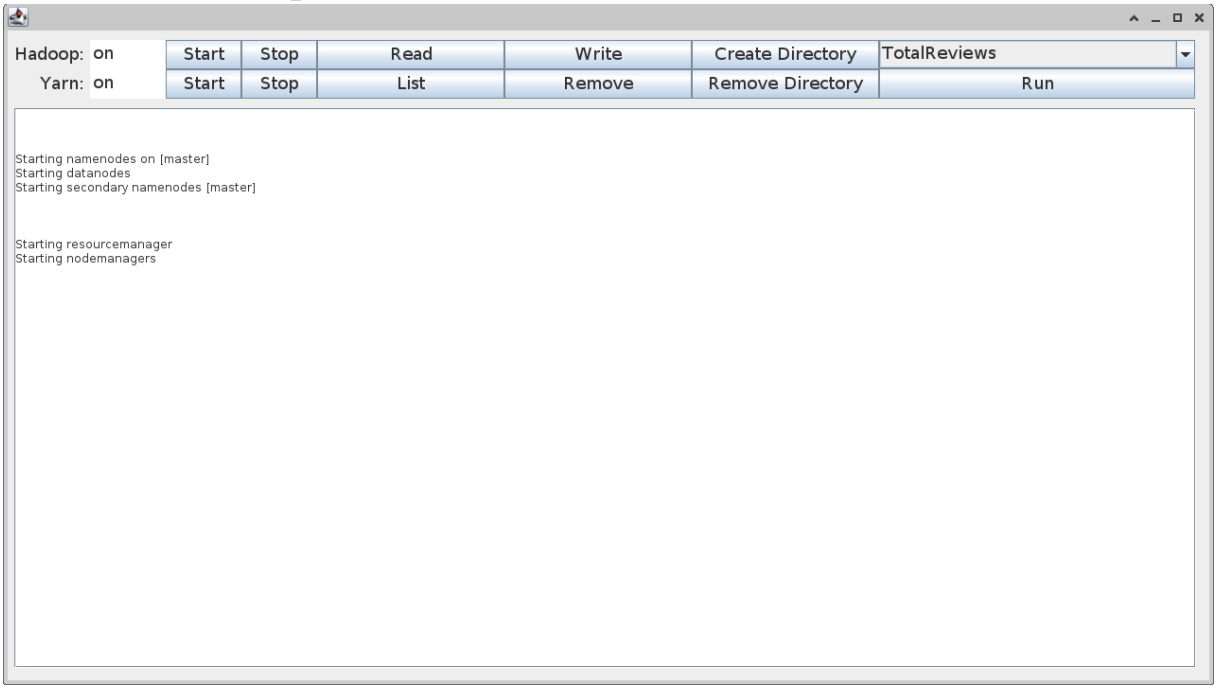
-hangi metod/işlem kullanılacaksa onu seçme (dropdown menu)

-programı çalıştırma menüleri yer almaktadır.



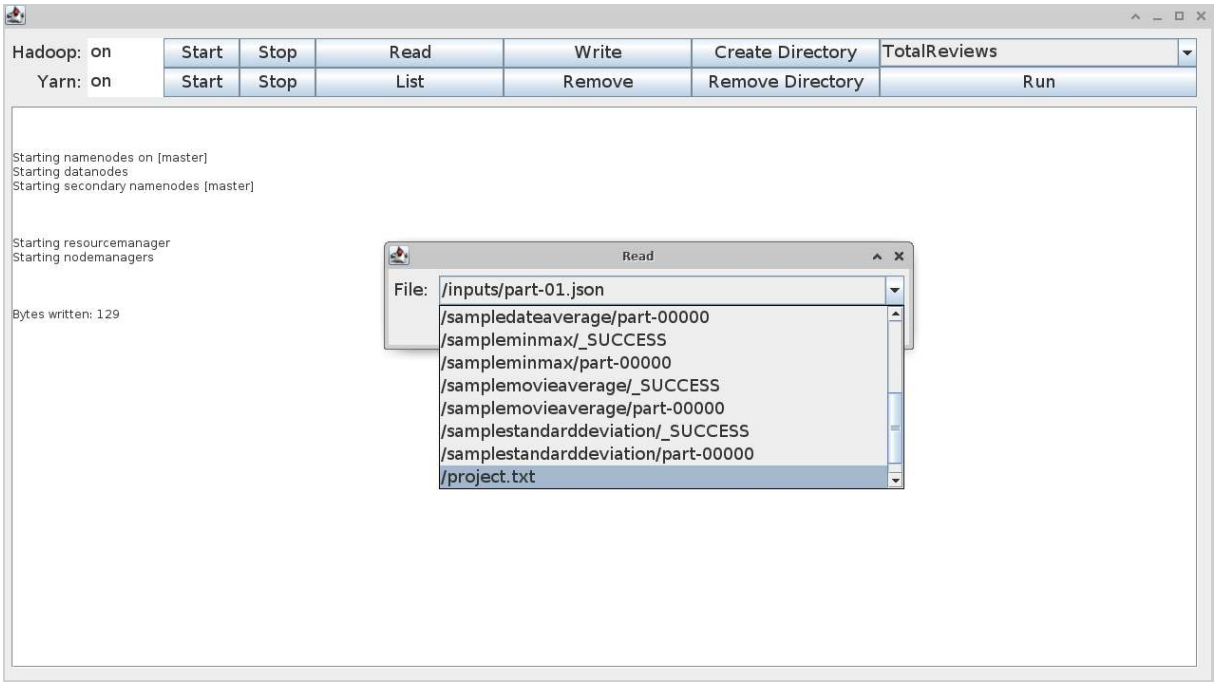
Menülere ait dialog pencereleri görseldeki gibidir.

3.1.1 Start – Stop



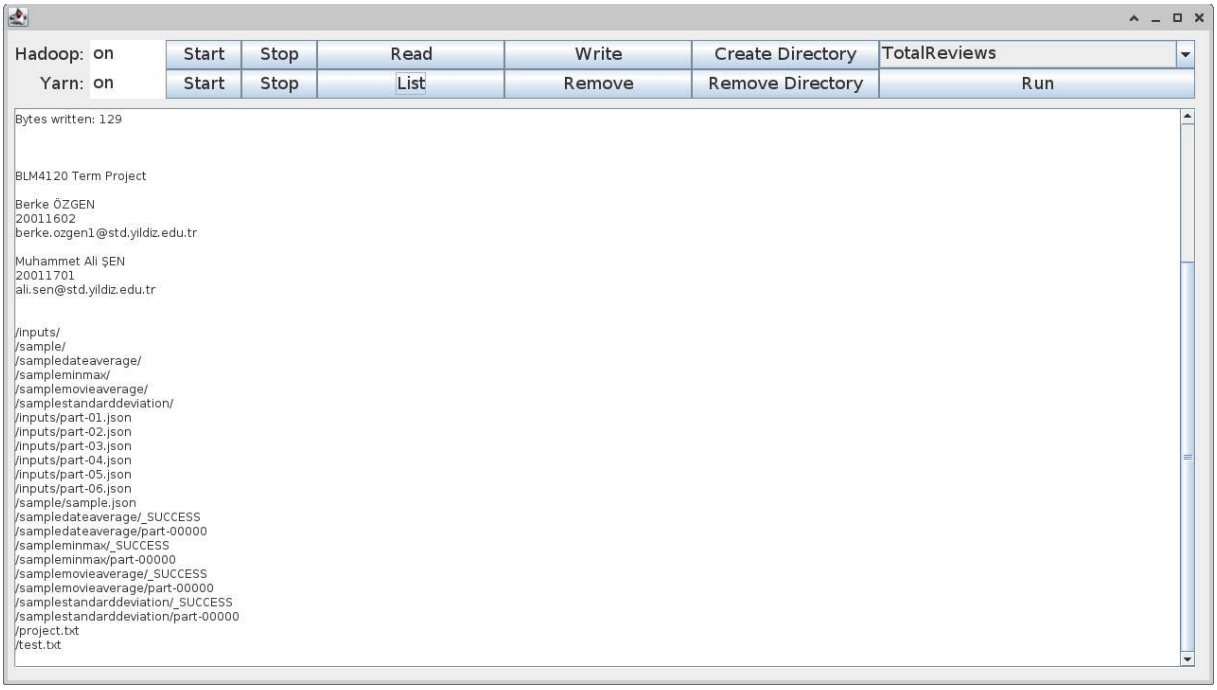
Hadoop ve Yarn servisleri Start ile çalıştırılır. Güncel durumu Hadoop ve Yarn metin alanlarının sağında ‘on/off’ şeklinde görülür.

3.1.2 Read



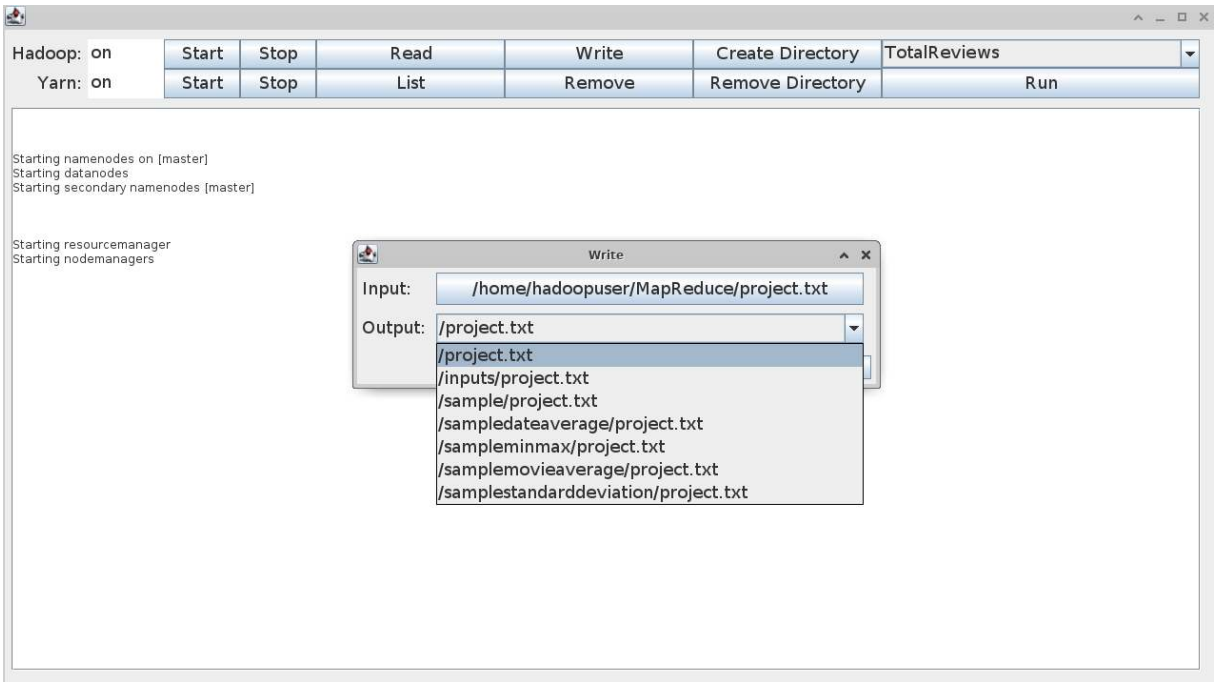
Read menüsü ile karşımıza çıkan file açılır menüsünden üzerinden hadoop sistemimizde yer alan bir dosya içeriği ekrana yazdırılabilir.

3.1.3 List

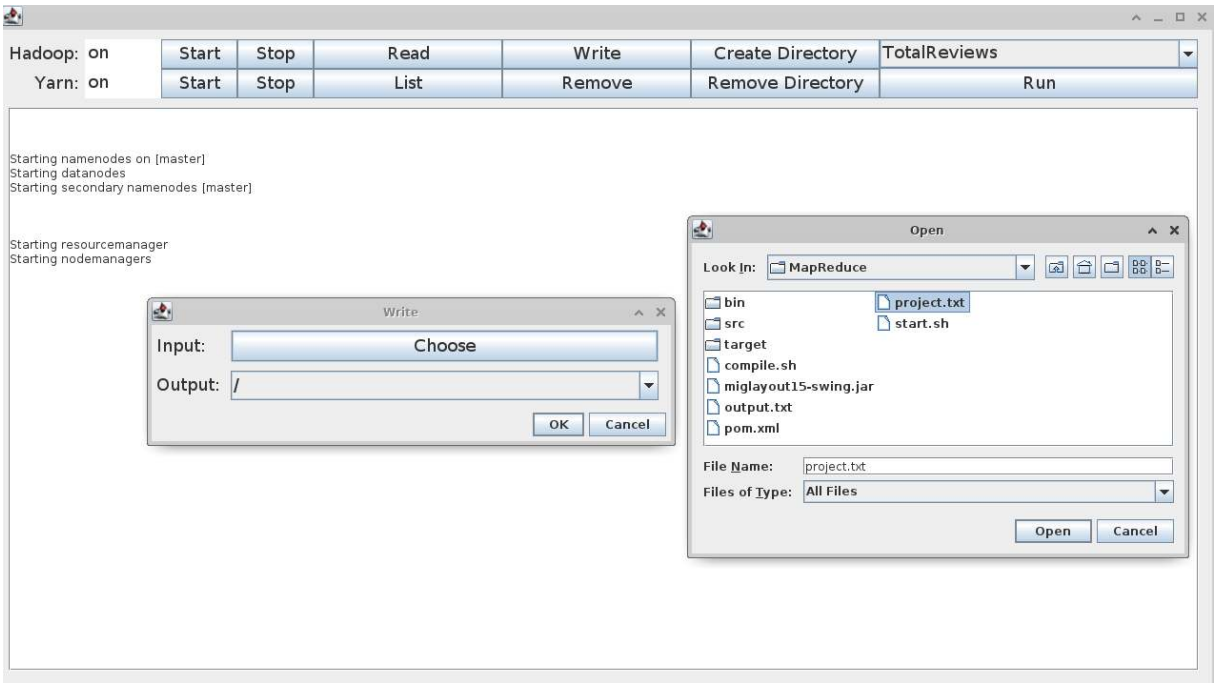


List menüsü tıklanınca üzerinde çalışılabilinecek büyük veriler terminal ekranımızda sıralanır.

3.1.4 Write

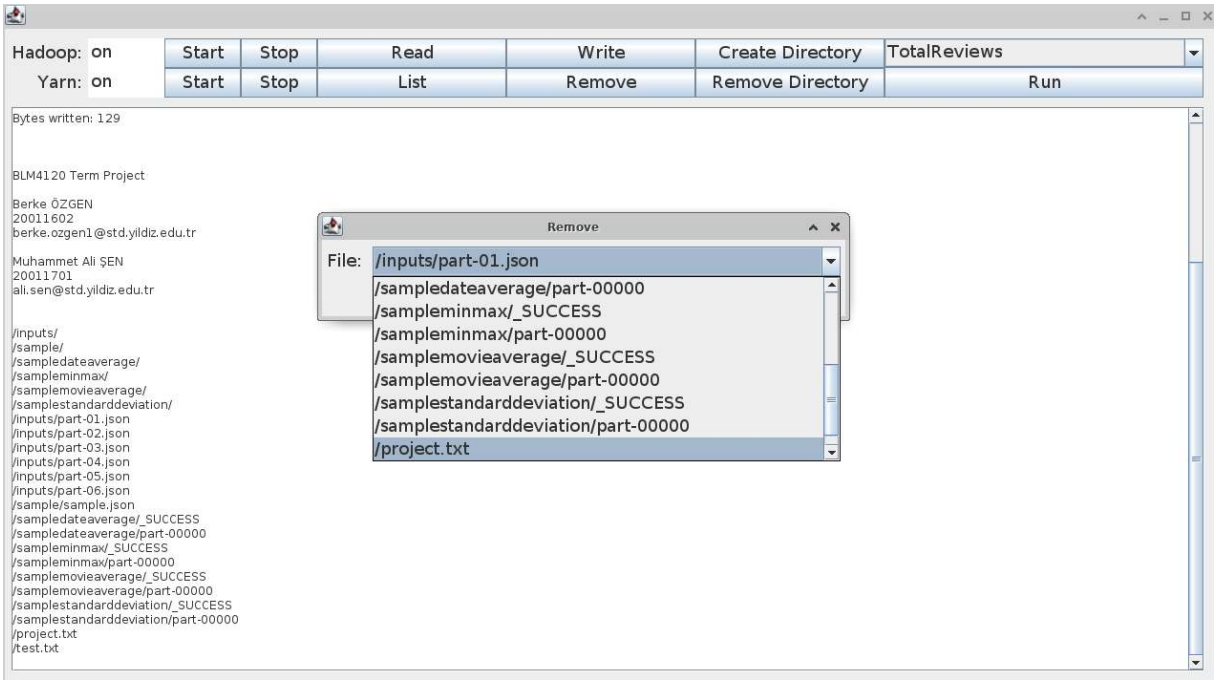


Yapılan analiz sonucu oluşacak çıktı hangi dizinde hangi dosyaya yazılması için menüden dosya seçilir.



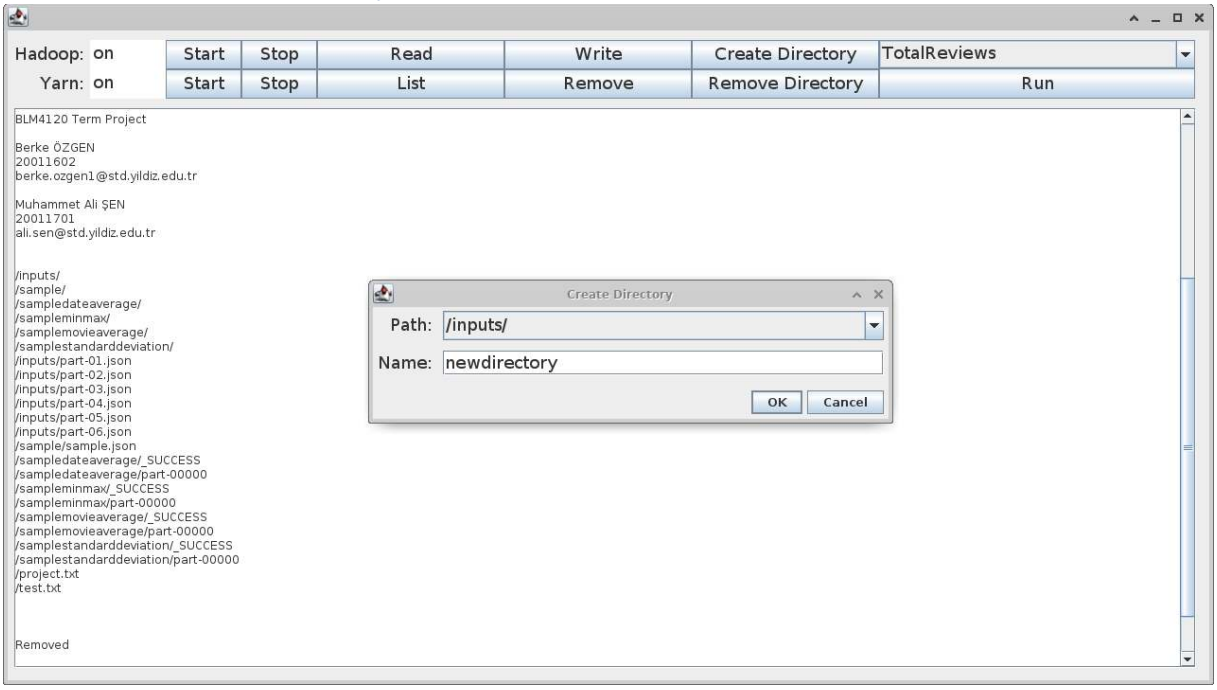
Var olan bir dosya seçilebilmesi için input kısmından Choose sekmesi tıklanarak hedef yazma dosyası seçilebilir.

3.1.5 Remove



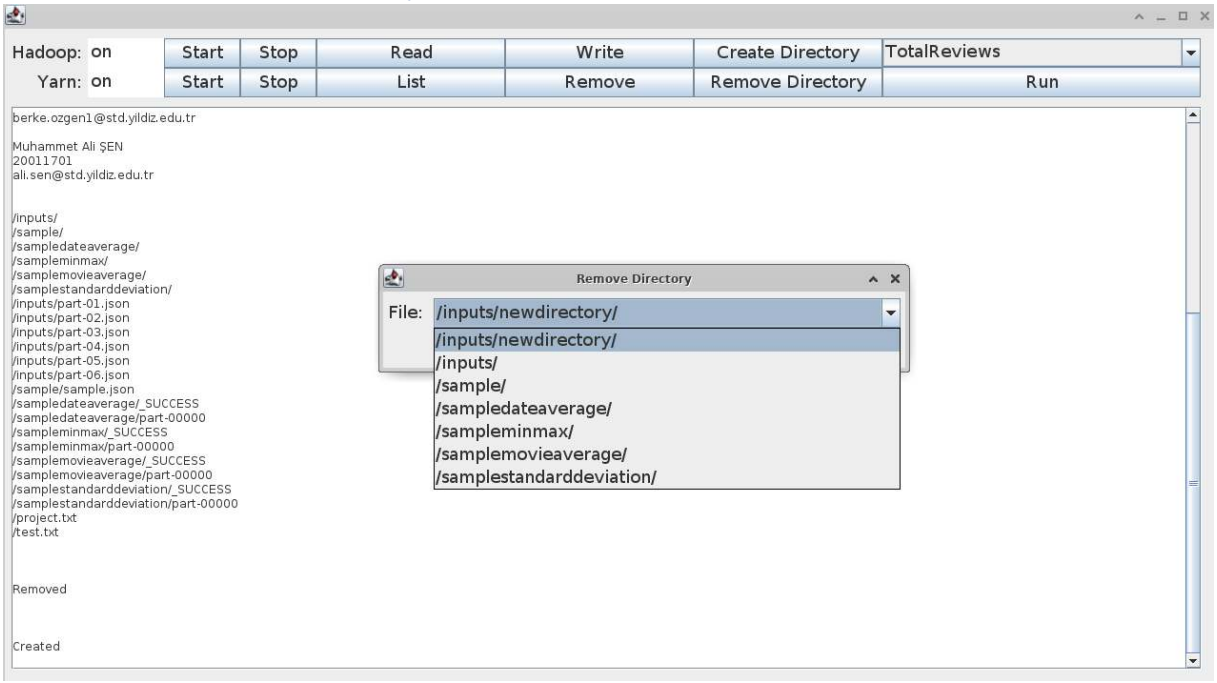
İhtiyaç kalmayan ve artık silinmesi istenen dosyalar seçilerek sanal bilgisayardan silinebilir. Bilgisayarımızda yer açılması için bu şekilde silme işlevi eklenmiştir.

3.1.6 Create Directory



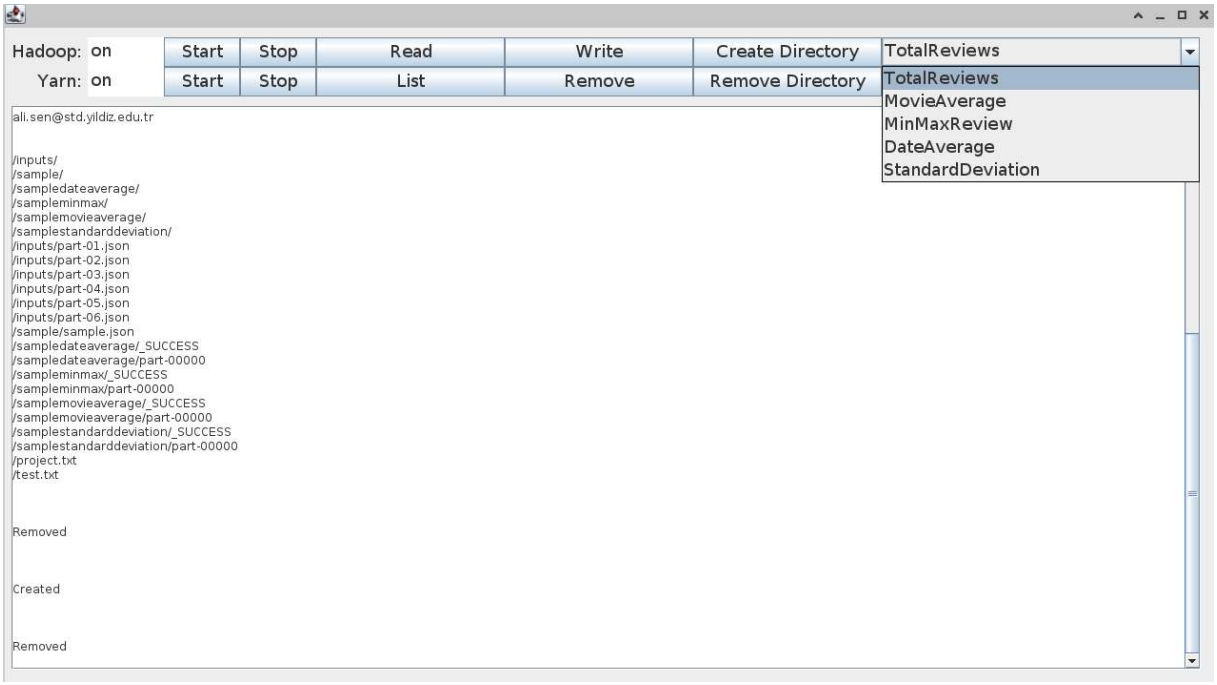
Sanal bilgisayarımız üzerinde klasör oluşturma ve çalışmalarımızı dizinleyebilme imkanı için yeni klasör olşuturulabilinir.

3.1.7 Remove Directory



İhtiyaç kalmayan klasör ve dizinler silinebilinir.

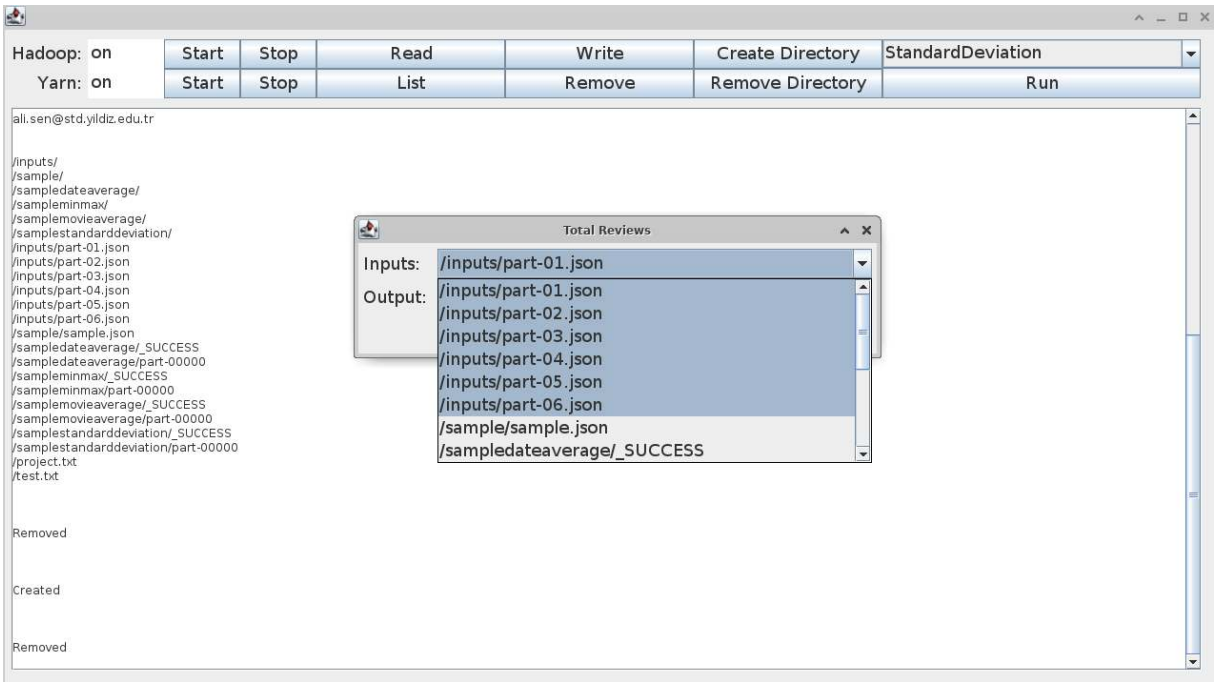
3.1.8 Jobs



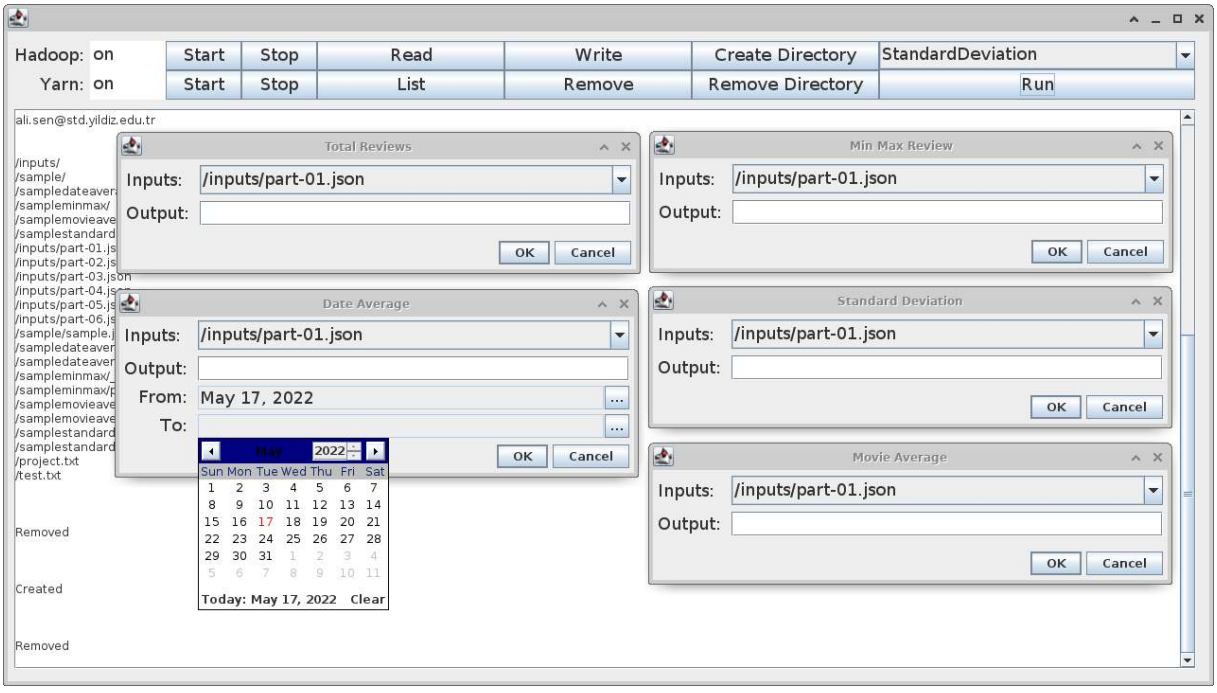
Programımız 5 genel iş yüküne sahiptir. Üzerinde çalışılması istenen büyük veri için yapılması istenen işlerden;

1. TotalReviews: Her kullanıcının yaptığı toplam inceleme sayısı
2. MovieAverage: Her filmin ortalama puanı
3. MinMaxReview: Her filmin minimum ve maksimum puanları
4. DateAvearge: Her filmin verilen bir tarih aralığında yapılan incelemelerinin puan ortalamaları
5. StandartDeviation: Her filmin puanlarının standart sapmaları

İstenileni yapılabilir.

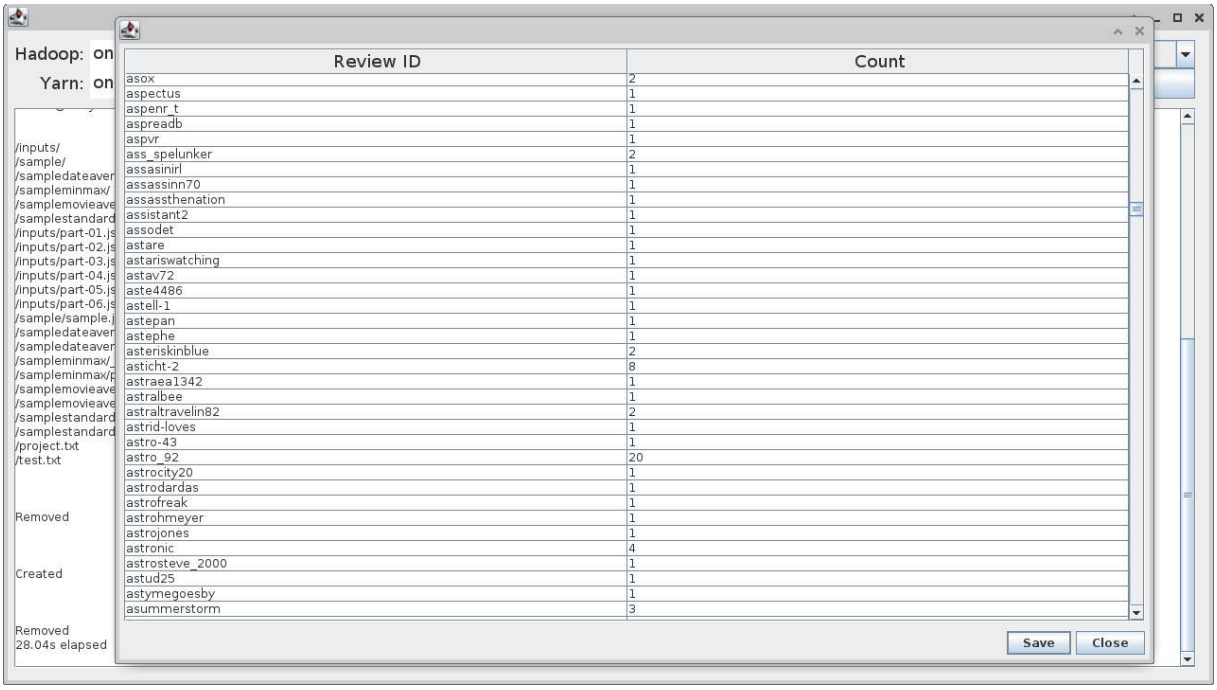


Yapılması istenen işlem seçildikten sonra hangi büyük veri üzerinde bu işlemin yapılacağı belirtilmelidir.



İşlemlere ait dialog pencereleri görseldeki gibidir. İşlemlerin farklı girdi değerleri olabilme durumu göz önüne alınarak farklı dialog pencereleri oluşturulmuştur.

3.2 Result



İstenen çalışma bittikten sonra çıktımız ekranımızda görseldeki gibi görülür.

4. Projenin Çalıştırılması

4.1 Projenin Gereksinimleri

Projemiz gerekli işlemleri gerçekleştirebilmesi için maven ve hadoop servislerinin yüklü olması gereklidir.

4.2 Projenin Çalıştırılma Aşamaları

Projenin çalıştırılabilmesi için aşağıda yer alan adımların sırasıyla gerçekleştirilmesi gereklidir.

1. mvn install clean compile assembly:single
2. hadoop jar target/MapReduce-jar-with-dependencies.jar

5. Karşılaşılan Zorluklar

Büyük veri analiz işlemlerinde kullanılan Hadoop Teknolojisinde çok yeni olmamız nedeniyle karşılaştığımız birçok sorunda internetten yardım alarak çözümler üretmeye çalıştık. Örnek olarak hadoop kurulumu aşamasında dökümantasyonu takip ettik. JSON veri tipini işlemek için Hadoop tarafından sunulan bir kütüphane olmaması nedeniyle, internetten bulunan XML formatındaki dosyaları işleyen bir kütüphane değiştirilerek JSON formatını desteklemesini sağladık.

Akabinde Java kütüphanelerinin kullanımı sırasında örnek kodlamalardan yardım aldık. GUI tasarımına Hadoop işlemlerini entegre ederken güçlük çektik ancak bunun çözümü için hazır kütüphanelerden (Swing vb.) destek aldık.

Ayrıca tek node ve daha fazla node üzerinde çalışılarak performans farkının görülebilmesi işlemi configuration (yarn, hadoop) dosyaları üzerinde bir hayli düzenleme yapılmasını gerekli kılmıştır. Bunların araştırılması ve doğru şekilde ayarlanabilmesi çok araştırma yaptık.

Üzerinde çalıştığımız büyük veriyi birden fazla node ile analiz edebilmek için sanal cihazların (Microsoft Azure) kurulum ve kullanımı konusunda herhangi bir deneyimimizin olmaması nedeniyle birçok engelle karşılaştık. Yine sanal cihazların kurulum ve efektif kullanımında karşılaşılan bu teknik sorunlar için internet araştırmalarıyla çözüm ürettik.

Node sayımız toplamda 2 olması ve bunlardan birinin de hem master hem de slave olarak görev alması nedeniyle performans farkını sadece saniyeler cinsinden görebildik.

6. Gerçekleme

Öncelikle Java ile Hadoop ve Yarn servislerini yönetmeyi gerçekledik. Her bir dosya ve klasör işlemi için yazdığımız metotları içeren java sınıflarını oluşturduk. Yapacağımız her Map Reduce işlemi için, bir driver mapper ve reducer sınıflarını oluşturduk. Bunlar sayesinde Büyük Veri Analizi işlemlerini map ve reduce metotları ile gerçekleştirdik. Arayüz tasarımında Window Builder eklentisini kullanarak sürükle bırak yöntemiyle hızlı ve efektif bir tasarım dizayn ettik. Yapılacak işlemler için ayrı ayrı pencereler oluşturarak kullanışlı ve modüler bir tasarım düzenlemeye çalıştık. Kullanım kolaylığını sağlamak için dosya işlemlerini içeren pencerelerde dosya sistemlerinden okuma yaparak kullanıcıya var olan dosya ve klasörler arasından seçim imkanı sunduk.

1. TotalReviews
    a. Mapper:
        i. Key: İncelemenin ID numarası
        ii. Value: 1
    b. Reducer:
        i. Key: İncelemenin ID numarası
        ii. Value: Value değerlerinin toplamı
2. MovieAverage
    a. Mapper:
        i. Key: Film ismi
        ii. Value: Filme verilen puan
    b. Reducer:
        i. Key: Film ismi
        ii. Value: Filme verilen toplam puanların inceleme sayısına oranı
3. MinMaxReview
    a. Mapper:
        i. Key: Film ismi
        ii. Value: Minimum ve Maksimum alanları verilen puana eşit olan MinMaxTuple objesi
    b. Reducer:
        i. Key: Film ismi
        ii. Value: Puanlar arasında gezinti yaparak Maks ve Min değerlerinden oluşan MinMaxTuple objesi
4. DateAverage
    a. Mapper:

i. Key: Kullanıcıdan alınan tarih aralığında yapılan incelemelerin Film ismi
ii. Value: Filme verilen puan

b. Reducer:
i. Key: Film ismi
ii. Value: Filme verilen toplam puanların inceleme sayısına oranı

5. StandardDeviation
a. Mapper:
i. Key: Film ismi
ii. Value: Filme verilen puan
b. Reducer:
i. Key: Film ismi
ii. Value: Filmin puanlarının standart sapması

6. Performans Analizi

Part-01.json Dosyası 1.1GB **Total Reviews**	**Tek Node İş Bitirme Süresi**	**İki Node İş Bitirme Süresi**
1. Deneme	114.33s	102.21s
2. Deneme	119.29s	102.27s
3. Deneme	125.09s	85.09s

7. Sonuç & Değerlendirme

Yaptığımız bu projede birçok yeni teknoloji ve bu teknolojilerin kullanımı konusunda çok fazla deneyim kazandık. Derslerimizde büyük veri analiz işlemlerinin nasıl yapılması gerektiğini ve nasıl gerçekleştirildiğini teorik olarak görmüştük. Ancak pratik olarak da, elimizde yer alan büyük veriyi açık kaynak kodlu kütüphaneler ve servisleri ücretsiz şekilde kullanarak birden fazla bilgisayar üzerinde koşturabilmeyi ve analizini mümkün kılabilmeyi öğrenmiş olduk. Artık büyük veri üzerinde daha efektif ve kolay şekilde basit map-reduce işlemleri ile analiz yöntemlerini deneyimlemiş olarak büyük verileri üzerinde istatistiki çalışmaları yapabiliyoruz.

Bu işlemleri kolaylaştırmak için tasarladığımız arayüz kullanımı kolay ve istenen işlemleri gerçekleştirebilmeyi mümkün kılacak şekilde tasarlanmıştır. Ayrıca terminal çıktısını ekrana yansıtarak ile karşılaşılan hata çeşitleri ve nerelerde karşılaşıldığı bug-fixi kolaylaştırmak için dizayn edilmiştir. Java ile hazırlanan programımızın arayüzü de ileride geliştirmeye açık şekilde yeni menüler eklenebilecek şekildedir.