YAZILIM LABORATUVARI 1 PROJE 2

TARIK GÖREN

Bilgisayar Mühendisliği 200202022@kocaeli.edu.tr

ÖMER ARAN

Bilgisayar Mühendisliği 190202012@kocaeli.edu.tr

ÖZET

Bu proje ile amacımız thread ve multithread kavramlarını öğrenmek, uygulamak ve büyük verilerle çalışırken dikkat etmemiz gereken noktaları keşfetmek aynı zamanda tecrübe edinmekti. Yaklaşık bir milyon verinin bulunduğu bir şikayet listesini önce csv formatında localimize alarak ardından tercih ettiğimiz java dilinin bir kütüphanesi olan common-csv kütüphanesiyle ayıklamak gereksiz sütunları yok edip yeni csv dosyasına yazdırıp ardından bu şikayet listesindeki değerleri java arayüzü olan swing ile masaüstü uygulama haline getirerek isterlere göre listeleme ve filtreleme yapmaktı. Multihtreadler ise burada devreye giriyor. Milyonlarca veriyi kendi belirlediğimiz thread sayısı ile işlemek inanılmaz bir hız katmalı ve zamandan kazandırıp kaliteli bir hizmet sunmaya yardımcı olmalıdır. Temel amacımız multithtread, ayıklama ve masaüstü uygulama geliştirmekti.

I. Giriş

Bu projede backend kısmında java dilini, arayüz tasarımında Java Swing'i verileri ayıklamak için ve gerektiğinde tekrar yazdırmak için common-csv kütüphanesini ve işlemleri hızlandırmak için threadleri kullandık.

Bu proje kapsamında bizden aşağıdaki amaçların gerçekleştirilmesi istenmiştir:

- Elde edilen tabloda 6 farklı sütun bulunmalıdır: Product (Ürün), Issue (Konu), Company (Şirket), State, Complaint ID, Zip Code.
- Null değer içeren kayıtlar bulunmamalıdır.
- Kayıtlardaki noktalama işaretleri kaldırılmalıdır
- Kayıtlardaki stop word'ler kaldırılmalıdır (nltk kütüphanesi kullanılabilir)

II. İSTERLER

- Verilen veri seti istenen şekilde yeniden düzenlenmelidir.
- Düzenlenmiş veri setindeki kayıtlar arasında benzerlik kontrolü yapılmalıdır. Kontrol sırasında mutlaka multithreading kullanılmalıdır. Multithreading için kullanılacak thread sayısı uygulama arayüzünden girilmelidir
- Her thread'in çalışma zamanı ve tüm thread'ler için toplam çalışma zamanı bilgileri uygulama arayüzünde gösterilmelidir.
- İstenilen sütun ya da sütunlar arasındaki girilen benzerlik oranı (threshold) ve üzerinde benzerliğe sahip kayıtlar masaüstü uygulamasında gösterilmelidir..
- Uygulamanızı sunmak üzere basit bir arayüz geliştirmeniz istenmektedir. Bu arayüz aşağıdaki isterleri içermelidir:
 - Benzerlik oranının (Threshold değeri) seçilebileceği / girilebileceği bir araç
 - Benzerliklerinin araştırılması istenen sütun veya sütunların seçilebileceği bir araç.
 - Kaç tane thread kullanılacağının seçilebileceği / girilebileceği bir araç,
 - Her bir thread'in çalışma zamanını ve toplam çalışma zamanını gösteren araçlar
 - Sonuçların açıkça ekranda gösterilebileceği bir araç.
- Uygulamada aşağıdaki ve benzer senaryolardan elde edilen sonuçlar ekranda gösterilmelidir:
 - Senaryo 1: Ürün (Product) sütununda ekranda gösteriniz.
 - Senaryo 2: Aynı ürünler (Product) için (Issue) içeren Şirket (Company) isimlerini ekranda gösteriniz.
 - Senaryo 3: 'Complaint Id' = 3198084 olan şikayet kaydı için benzerlikteki konuları (is-

sue) içeren kayıtları ekranda gösteriniz.Senaryo 4: 5 Thread ile Konular(Issue) sütununda kayıtları ekranda gösteriniz

III. YÖNTEM

A. Başlangıç

Öncelikle bizden istenen programları ve programların içerisinde kullanacağımız kütüphaneleri entegre ettik. İsterleri gerçekleştirmeye başladık. Bizden istenen ilk isterden başladık. Common-CSV dokümanları ve ilgili eğitim videoları ile ayıklama yapacağımız verileri oluşturduğumuz maven projesinin test kısmında denemeler yaparak complaint(kaggle'den alınan veri kümesinin complaintlerden oluşması üzerine class ismi olarak bunu uygun gördük.) verilerini csv dosyamızdan çekmeyi denedik.

B. Geliştirme Ortamı

Projenin backend tarafı Java'nın Maven mimari proje tipi ile geliştirilirken arayüz tarafı Swing-GUI ile geliştirilmiştir. Ayıklama aracı olarak Common-csv ve Java csv okuma-yazma komutları kullanılmıştır.

Macos-Windows sistemlerde geliştirilip IDE olarak Intellij Idea Ultimate Version ve Netbeans 15 kullanılmıştır. Projenin versiyon kontrolü tamamen github üzerinden yapılmıştır. IDE'lerimizde yazdığımız kodları ayıklama ve senaryo gerçeklemelerini sağladıktan sonra çalıştırıp önlem açısından debug ve testler yapılmıştır.

IV. PSUEDO KOD

Bir ayıklama ve multithread projesinin yalancı kodu senaryoları gerçekleşme sırasına uygun bir durumda açıklayarak gerçekleşir.

Uygulama Çalıştırılır:

Arayüz(swing) açılır: seçebileceğimiz tablar ve senaryolar

senaryo 1: sectik

gerekli verileri girip getir butonuna bas

button similarity classının similarity1 fonksiyonunu

çağırıp

ayıklanmış veriyi load et

load olan veriyi istenen kolon ve benzerlik oranına göre çalıştır

şartları sağlayanları bir listeye at

listeyi döndür

arayüzde table'da sayı ve kolonları listele

senaryo1 işi biter:

senaryo2 çalıştır:

gir thread sayısı

gir benzerlik oranı

gir talep edilen kolon

gir aynı olanlar

butona bas: getir 2

arka planda similarity2 fonksiyonu çalışır

ayıklanan verileri bir listeye ata

listedeki verileri giriş koşullarına uygun olanları al

yeni listeye at

liste tamamlanınca threadleri kapat

listeyi döndür

arayüzde thread ve veri listeleme bölümleri var

bu bölümlerde elindeki verileri sergile

senaryo 2: bitti.

senaryo 3:

gir thread sayısı

gir benzerlik oranı

gir talep edilen kolon

gir aynı olanlar

butona bas: getir 3

arka planda similarity3 fonksiyonu çalışır

ayıklanan verileri bir listeye ata

arayüze döndür

senaryo 3: bitti

senaryo 4:

gir thread sayısı

gir benzerlik oranı

gir talep edilen kolon

gir aynı olanlar

butona bas: getir 4

arka planda similarity4 fonksiyonu çalışır

ayıklanan verileri bir listeye ata

senaryo 4: bitti

V. Sonuçlar

Proje amacı yerine getirilerek multithreading ile verileri daha hızlı işleme, ayıklama, atama ve şartları sağlayanları arayüze döndürerek baştan sona end to end bir masaüstü uygulaması geliştirdik. Multithread kavramının temel amacı ve gerekliliği hakkında bilgi edindik. Geçmiş yıllarda da kullandığımız üzere javanın masaüstü app geliştirmek maksadıyla sunmuş olduğu swing gui ile projemizin önuç kısmını senaryoları gerçekleştirmek için ve aynı zamanda kullanıcıya daha iyi bir deneyim sunmak amacıyla combobox'lar, textfieldlar ve sade tasarım ile geliştirdik. Verileri işleme kısmı bir kez olacağı için projenin main yerine test kısmında bir class içerisinde parse adındaki class ile ayıklayıp yeni bir csv dosyasına attık bu şekilde her seferinde yeniden parse işlemi yapmak zorunda kalmadık. Bir yazılım hizmeti sunmanın çeşitli aşamaları olduğunu bu aşamaların hepsinin projenin sunumunda ve kullanıcı deneyimi kısmında önemini aynı zamanda zamandan ve maliyetten kazanmanın multithreadler aaracılığı ile yapılabileceği ve milyonlarca veriye sahip şirketlerin bu verileri işlerken ya da efektif bir veri kullanımı sağlamak adına girişimlerde bulunurken maliyeti en aza indirme uğraşlarının yazılım kısmında çok büyük avantaj sağlayabileceğini gördük.

KAYNAKLAR

[1] Multihreading dokümanları

https://www.geeksforgeeks.org/multithreading-in-java/

https://www.tutorialspoint.com/java/java_multithreading.htm

https://medium.com/@sinanselimoglu/ java-8-streams-paralel-i%CC%87%C5% 9Flemler-3-a01233c8fd9f

https://www.tutorialspoint.com/java/java_multithreading.htm

https://medium.com/s%C4%B1f% C4%B1rdan-i%CC%87leri-d%C3% BCzeye-java-e%C4%9Fitim-serisi/ multithreaded-programlama-1-k%C4%B1s% C4%B1m-40904a219a18 [2] Veri Seti Düzenleme

https://commons.apache.org/proper/commons-csv/

https://www.baeldung.com/ apache-commons-csv

https://www.callicoder.com/ java-read-write-csv-file-apache-commons-csv/

https://commons.apache.org/proper/commons-csv/user-guide.html

https://stackoverflow.com/questions/35319544/removing-stopwords-java

[3] Swing Arayüz dokümanları https://www.javatpoint.com/java-swing

https://docs.oracle.com/javase/7/docs/api/javax/swing/package-summary.html

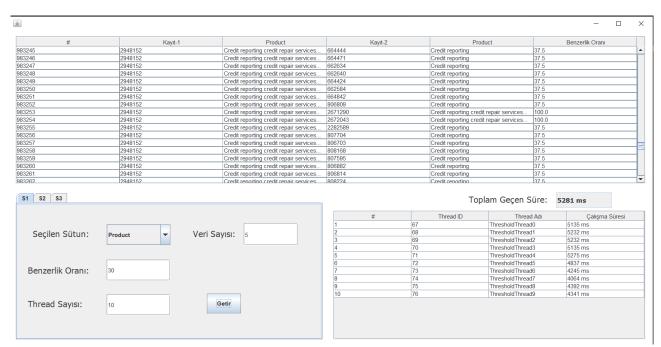
https://docs.oracle.com/javase/7/docs/api/javax/swing/ComboBoxModel.html

http://www.baskent.edu.tr/~tkaracay/etudio/ders/prg/java/ch27/JavaSwing.pdf

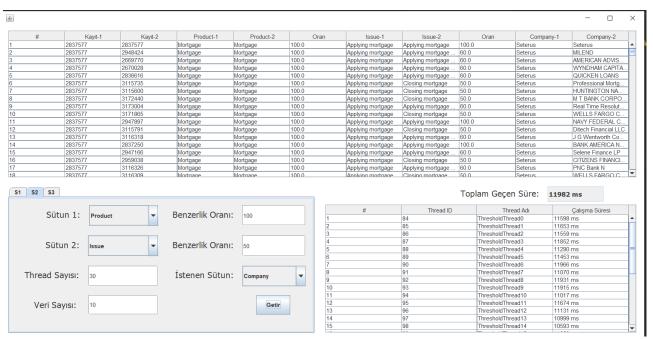
[4] Spring Framework Doküman https://spring.io/projects/spring-boot

https://www.baeldung.com/learn-spring-course

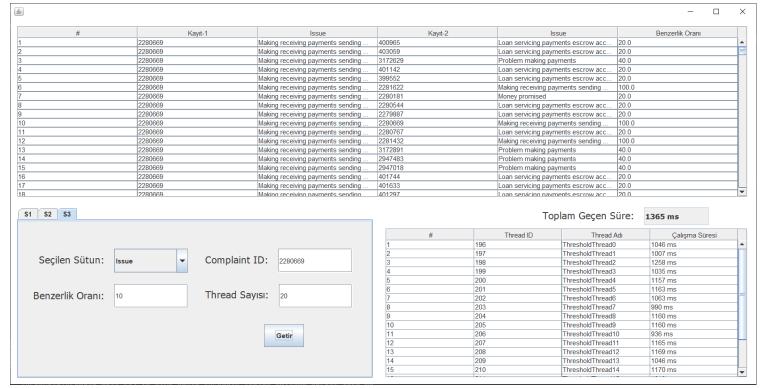
EKRAN GÖRÜNTÜLELRİ



Şekil 1: Senaryo 1



Şekil 2: Senaryo 2



Şekil 3: Senaryo 3