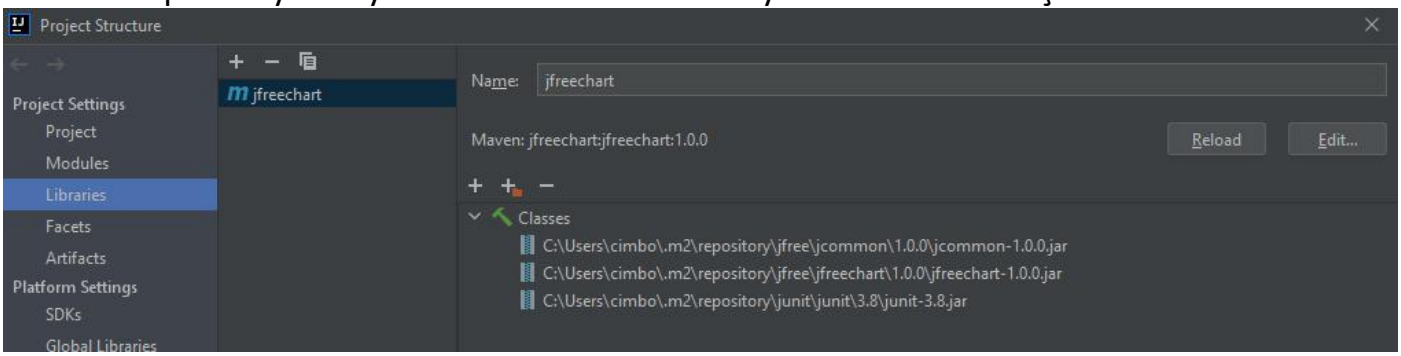
BIL3013 Veri Madenciliğine Giriş - Final

Öğrenci Adı: Eyüb Salih Özdemir
Numarası: 2018280059
Proje dili: Java (via IntelliJ IDEA)

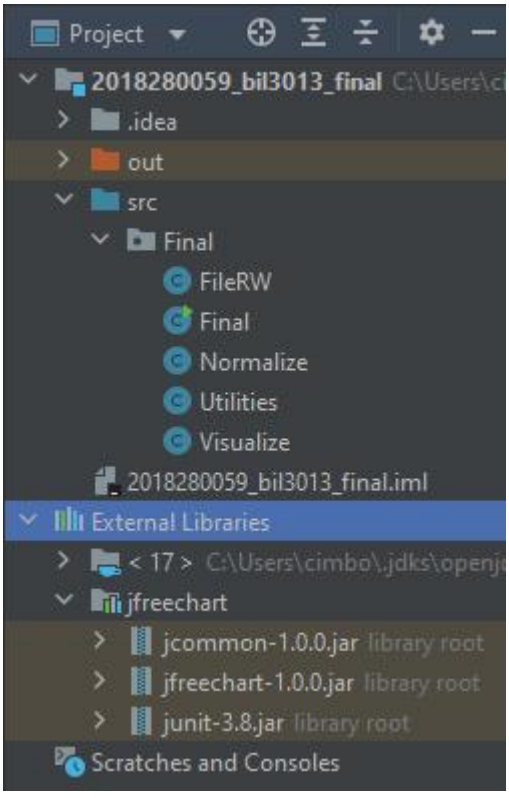
Gereklilikler

- Final-data.txt dosyası “C:” dizininde olmalıdır. (Final.java - 46. ve 47. satırlarda değiştirilebilir)

- Görselleştirme için **JFreeChart (v1.0.0)** kütüphanesi (programdaki tek hazır kütüphane) kullanılmıştır. IntelliJIDEA kullanılarak Maven için yüklenmiştir. “.jar” dosyalarını içeren Maven “repository” dosyası teslim edilen RAR dosyasına dahil edilmiştir.



Ön Bilgiler



Final.java - Main Class

FileRW.java:
- *readFile*(): Datayı okur.
- *writeResults*(): “sonuc.txt” dosyalarını yazar.

Normalize.java:
- *zScore*(): Final-data’nın normalize edilmiş halini döndürür.

Visualize.java:
- *int SHAPE_SIZE, IMAGE_WIDTH, IMAGE_HEIGHT, Color BACKGROUND_COLOR*;
- *visualize*(): Küme değerlerini serilere ekler ve yaratılan tabloyu PNG olarak kaydeder.
- *createChart*(): Tabloyu yaratır ve verilen dataset(series)’e göre özelliklerini belirler.

Utilties.java:
- *analyzePerformance*(): WCSS, BCSS and Dunn Index değerlerini hesaplar ve bir liste olarak döndürür.
- *euclideanD*(): Verilen iki List<List<Double>> listeleri arasındaki Euclidean Distance’ı hesaplar ve döndürür.
- *getCategoryInput*(): Görselleştirilecek kategorilerin input olarak alımını yapar.

```
//
boolean debug = false;
int times_run = 20; // how many times do you want the program to run, if debug = true
```

Debug=False durumunda:

- Program tek sefer çalışır.
- Verilen “cluster count (K value)” ve görselleştirme için kategori değerlerine göre, tek bir PNG dosyası ve tek bir “sonuc.txt” dosyasını masaüstüne kaydeder.
- Cluster boyutlarını, iterasyon sayısını ve istatistik değerlerini(WCSS, BCSS, Dunn) ön inceleme olması için konsola yazdırır.

Debug=True durumunda:

- Hard code şeklinde belirlenen “times_run” sayısınca çalışır.
- Her çalıştırma için farklı “cluster count (K value)” ve görselleştirme için kategori değerleri **sormaz**. Tüm çalıştırmalar aynı girdileri kullanır.
- Çoklu PNG ve TXT çıktılarını, uygun isimler altında klasörler ile masaüstüne kaydeder.
- **Her çalıştırmadaki** cluster boyutlarını, iterasyon sayılarını ve istatistik değerlerini ön inceleme olması için konsola yazdırır.

Bunlardan daha önemli birkaç bilgi main fonksiyon başlangıcında, ve kod bloklarını açıklayan kısa ek açıklamalar program boyunca verilmiştir.

Diğer Açıklamalar

```
Map<Integer, Color> colorMap = new HashMap<Integer, Color>();
colorMap.put(0, Color.RED);
colorMap.put(1, Color.BLUE);
colorMap.put(2, Color.GREEN);
colorMap.put(3, Color.MAGENTA);
colorMap.put(4, Color.PINK);
colorMap.put(5, Color.BLACK);
colorMap.put(6, Color.CYAN);

// set 7 different colors of series to be able to use later on
for (int i = 0; i < 7; i++) {
    renderer.setSeriesPaint(i, colorMap.get(i));
    renderer.setSeriesLinesVisible(i, visible: false);
}
```

Program toplamda 7’ye kadar farklı küme destekliyor. Visualize.createChart()’ta bulunan colorMap’e farklı renk değerleri eklenerek bu sayı artırılabilir. Daha fazlası renk karmaşası yaratacağı için bu projeye eklemeye gerek görmedim.

Hem incelemede hem de herhangi bir sorun yaşanması durumunda kolaylık olsun diye şu senaryolarda çalıştırılan program ve konsol çıktılarını hazır olarak, teslim edilen RAR dosyasında veriyorum:

Senaryo 1:

- debug = false
- Cluster size: 2
- Categories: Theatre - Picnic

Senaryo 2:

- debug = true, times_run = 20
- Cluster size: 3
- Categories: Religious - Nature

Senaryo 3:

- debug = true, times_run = 20
- Cluster size: 4
- Categories: Nature - Shopping