

# PROLAB 3

## VERİ ANALİZİ VE KULLANICI İLİŞKİLENDİRMESİ

1. Çağla Gök  
Bilgisayar Mühendisliği  
Kocaeli Üniversitesi  
220201060

2. Melis Portakal  
Bilgisayar Mühendisliği  
Kocaeli Üniversitesi  
220201035

### Özetçe

Bu projede, bir kaynak dosyasından kullanıcılara ait olan verileri çekerek bu verileri analiz etmeyi ve kullanıcılar arasında benzer ilgi alanlarına göre eşleştirmeler yapmayı hedefledik.

**Anahtar Kelimeler** — veri çekme, ilgi alanı analizi, hash tablosu, algoritmalar, graf, BFS, DFS.

### I. GİRİŞ

Projede kaynak dosyadan çekilen verileri bir hash tablosuna atayıp bu verilerle kullanıcılar arasındaki ilişkiyi isterler doğrultusunda analiz ettik. Çok fazla veride analizi kısa sürede yapıp çıktı alabilmek için çeşitli algoritmalar araştırdık. Kullandığımız C# programlama dilinde yazdığımız bu çeşitli algoritmalar sayesinde kodumuzu hızlandırdık ve bu algoritmaları özümseyip kodumuza entegre etmeyi öğrendik. Kaynak dosyasından aldığımız verileri analiz edebilmek için araştırmalar yaptık. Araştırmalarımız sonucunda ilk defa bir dil işleme kütüphanesi kullanmayı öğrendik ve projemize bunu entegre edebildik.

### II. YÖNTEM

#### A. JSON VERİ ÇEKME

İlk olarak verilerimizin olduğu twitter\_data.json kaynak dosyamızdan verileri çektik. Bunu yaparken NuGet paketlerinden Newtonsoft.Json adlı paket sayesinde verileri kodumuza entegre edebildik.

#### B. VERİ ANALİZİ

Bu aşamada kaynak dosyamız olan twitter\_data.json dosyasından aldığımız veriler string tipindeydi. Biz de Kullanici adında bir class açtık ve aldığımız verileri kullanıcı classının içerisindeki değişkenlerimize gönderdik. Classımızın içerisinde string tipinde username, name, language, region gibi verileri tutarken int tipinde followers\_count ve following\_count gibi

verilerimizi tuttuk. Bu sayede kaynak dosyamızdan aldığımız verileri uygun şekilde parçalayarak kodumuzda istelere göre daha rahat kullanabildik.

#### C. İLGİ ALANI ANALİZİ

İlgi alanlarını analiz edebilmek için NuGets paket yöneticisinden indirdiğimiz ZemberekDotNet Dil İşleme kütüphanesini kullandık. Bu kütüphane sadece Türkçe dili için uygun olup yabancı kelimeleri saymamaktadır. Bu kütüphane sayesinde her kullanıcının attığı tweetlerin arasından ilgi alanı olan kelimeleri buluyor. Bu kelimeleri buldururken bazı istenmeyen bağlaçlar, edatlar, zamirler gibi kelimeleri de sayıyor. Bunları ayırtırmak için IsStopWord adlı sınıfımızda bu tarz kelimeleri ayıkliyoruz. IsVerbEndingWithMekMak adlı sınıfımızda ise fiil durumundaki kelimeleri ayıkliyoruz. Aynı zamanda bu kelime ayıklama işlemi kelimelerin köklerine inerek yapıyoruz. Çünkü kelimelere gelen ekler yüzünden ekli hallerini farklı kelime olarak saydırıyor. Bu adımları yaptıkları sonra ilgi alanlarımızı bulup sonraki aşamalarımıza geçiyoruz.

#### D. İLGİ ALANLARINA GRAF

Bu aşamada, kullandığımız interestGraph sınıfının içerisinde her kullanıcının ilgi alanı analiz edildikten sonra BFS algoritması kullanarak aynı ilgi alanlarına sahip olan kullanıcıları buluyoruz ve yazdırıyoruz. Bunu yaparken her yeni gelen kullanıcıyı ve ilgi alanını kuyruğa ekliyoruz ve sonra düğümlere kişiler eklenmiş oluyor. Ayrıca daha önce ziyaret edilenleri de kontrol ediyoruz, bu sayede kodumuz daha hızlı çalışmış ve boşa zaman kaybetmemiş oluyor. BFS algoritması kullanarak ilgi alanını kullanan kişileri bulduktan sonra çıktılarımızı oluşturduğumuz interest.txt dosyamızın içerisine yazdırıyoruz.

## E. BAĞLI LİSTE

KullanıcıEkleme sınıfında elimizle oluşturduğumuz bağlı listemiz yer almaktadır burada her yeni kullanıcıyı bağlı listenin sonuna ekliyoruz.

## F. HASH TABLOSU

Olusanhash sınıfımızda kendimiz oluşturduğumuz hash tablosu yer almaktadır. Burda kullandığımız denklem sayesinde gelen verileri hash tablomuzun içerisine atıp verileri buradan çekerek kullanıyoruz.

## G. BENZER TAKİPÇİ SAYISINA SAHİP KULLANICILAR

UserGraph sınıfımızın içerisinde kendi oluşturduğumuz graf yapısını kullanarak takipçi sayıları benzer olan kullanıcıları buldurmaya yönelik yazdığımız algoritma var. Bu algoritmada düğümlere yeni kullanıcılar ve onların takipçi sayıları eklenerek graf yapımızın içerisini oluşturmaktadır. BFS algoritmamıza da baslangicKullaniciAdi ve hedefTakipciSayisi göndererek başlatıyoruz. Burada algoritma ile kullanıcıları arıyor ve benzer takipçi sayısı görünce bu kişinin takipçi sayısının benzer olduğunu yazdırıyor. Bu classın içerisinde de yine txtler oluşturduk ve benzer şekilde txtlerimize çıktılarımızı yazdırıyoruz.

## H. BÖLGE VE DİL

BolgeVeDil sınıfımızın içerisinde belirli bir bölgeye göre kullanıcıları yazdırmaya çalıştık. Burada da BFS algoritmasını kullanarak düğümlere ve kenarlara yeni kullanıcıları ve onların bölgelerini ekledik.

## İ. DOSYALARA KAYIT

Her bir konsol çıktısını dosyalara detaylı bir şekilde aktardığımız bu projemizde, txt dosyalarına aktarabilmek için kullandığımız StreamWriter metodu sayesinde yazdırabildik. Her farklı veri için karışıklık olması durumuna karşı ayrı txt dosyaları açıp onlara ekledik. Bu txt dosyalarının adları şu şekildedir: benzer.txt, bolgevedil.txt, bolgevedil2.txt, graf.txt, tekkisi.txt, kayıt.txt, interest.txt.

## SONUÇ

Yaptığımız projede veri çekebilmek için yöntemler araştırdık. Daha çok kullanıcı ve daha çok veri imkanı olduğundan dolayı twitter api yerine json dosyasından veri çekmeye karar verdik. Verileri analiz edebilmek için kütüphaneler araştırdık ve dil işleme kütüphanesini öğrenip kodumuzda kullandık. Çektiğimiz verileri analiz ettikten sonra çeşitli algoritmaları araştırdık. Graf yapısını ve bfs algoritmasını kodumuzda kullandık. Böylece algoritmalarımız sayesinde veri sayımız oldukça fazla olmasına rağmen kodumuzun çıktısını hızlı şekilde alabildik. Aynı zamanda seçmiş olduğumuz c# dilinde kod yazmayı pekiştirmiş olduk. Takım çalışması ile birlikte kodlarımızı yazarak projemizi tamamladık.

Bu projede bizden istenen isterlerin çoğunu yaptık. Bu isterlerden yapmış olduklarımız şu şekildedir:

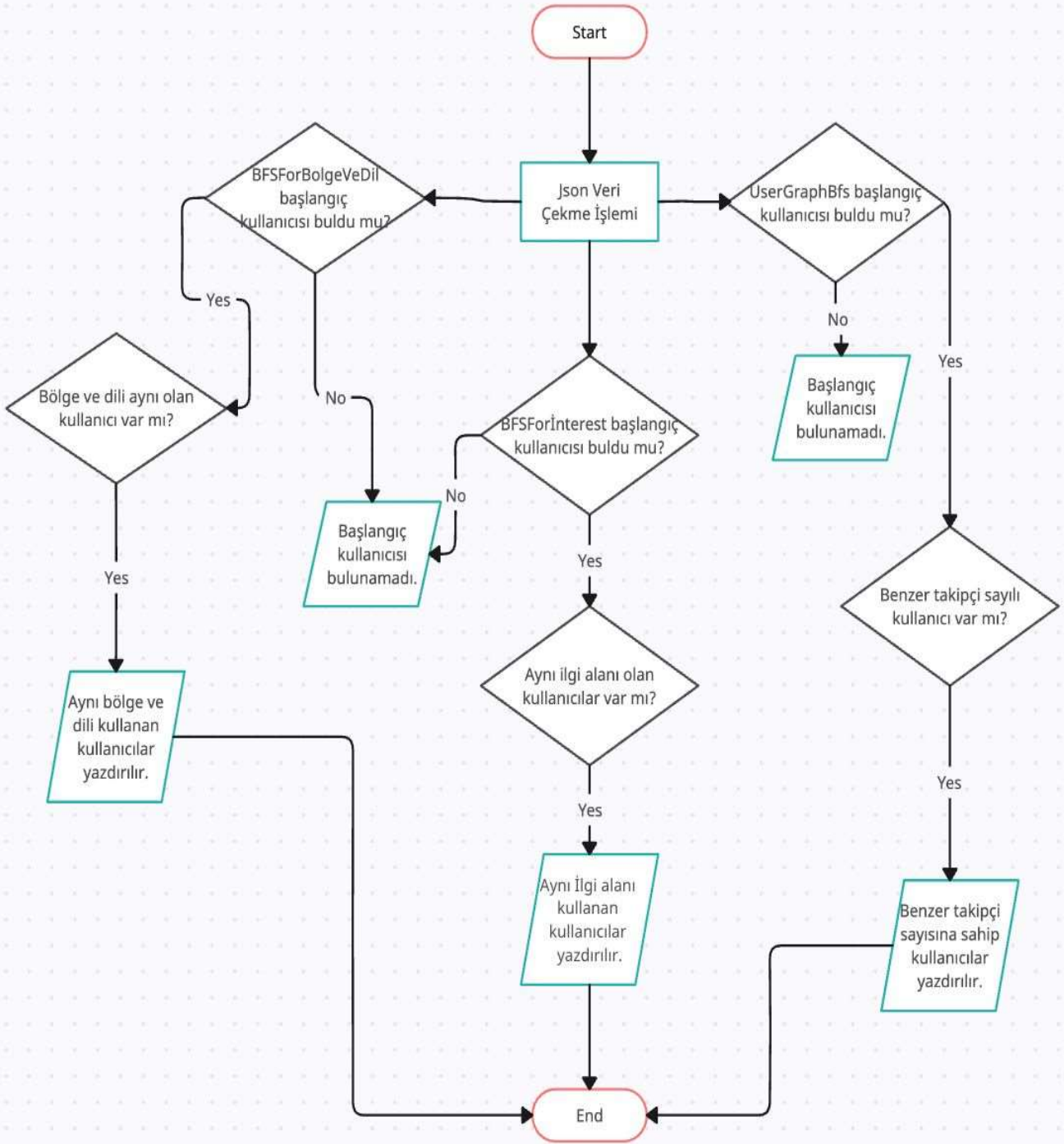
- Dosyalara kayıt işlemleri
- Veri çekme
- Hash tabloları ile organize etme
- Graf yazdırma
- Kullanıcıların ilgi alanlarının analizi
- Kullanıcılar arasındaki ilişki
- Bir kullanıcının aynı ilgi alanına sahip olan takip ettiği kişiler ve takipçileri

## III.KATKILAR

Bu projenin her bir kısmını birlikte yazdık. Daha önceki projelerdeki deneyimlerimiz doğrultusunda birlikte çalışınca projenin daha hızlı bir şekilde ilerlemesini ve ortaya çıkan sorunların çözümünde iki kişinin bir arada çalışmasının, tek bir kişiye kıyasla daha etkili olduğunu gösterdi. Bu süreçte birlikte hareket etmenin, yaratıcı düşünce süreçlerini güçlendirdiğini ve projenin kalitesini artırdığını fark ettik. Birbirimizin bakış açısından bakmayı öğrendik ve sorunlarımıza bu sayede daha hızlı çözümler ürettik. Kodu yazarken beyin fırtınası yapıp birbirimize güzel bilgi aktarımları yaptığımız verimli bir proje oldu.

## KAYNAKLAR




- [1] [Cytoscape.js tutorial demo](#)
- [2] [Hashing Nedir? \(Veri Yapıları\) - Örnek Üzerinde Anlatım - YouTube](#)
- [3] [BFS Algoritması \(Geniş Öncelikli Arama\) Konu Anlatımı - YouTube](#)
- [4] [DFS Algoritması Konu Anlatımı - YouTube](#)
- [5] [Graf Teorisi \(Graph Theory\) - YouTube](#)
- [6] [Hash Table \(Karım Tablosu, Özet Tablosu\) Veri Yapıları 22. Video - YouTube](#)
- [7] [5. Graph \(Graf veri yapısı\) | Mustafa Çağatay KIZILTAN](#)
- [8] [C#'ta JSON Kullanımı | by ismail kaşan | Medium](#)
- [9] [Big-Data-Project/HashtagAnalysis/src/HashtagAnalysis.java at master · Zhanelya/Big-Data-Project](#)
- [10] [Twitter-Data-Analysis/spark\\_app\\_A.py at master · fahad-qayyum/Twitter-Data-Analysis](#)
- [11] [ahmetaa/zemberek-nlp: NLP tools for Turkish.](#)
- [12] [NuGet Gallery | ZemberekDotNet.Tokenization 0.18.1](#)
- [13] [ZEMBEREK — Doğal Dil İşleme. Son günlerde fazlasıyla popüler olan... | by Uğur Özker | Medium](#)
- [14] [VeriYapilari/hafta08 - ikili arama agaclari/01 - Dizi ile - Copy/src/IkiliAramaAgaci.cpp at master · kayhanayar/VeriYapilari](#)
- [15] <https://medium.com/tapu-com-bak%C4%B1%C5%9F-a%C3%A7%C4%B1s%C4%B1/bfs-breath-first-search-geni%C5%9F-%C3%B6ncelikli-arama-algoritmas%C4%B1n%C4%B1-tan%C4%B1yal%C4%B1m-ec7050a41af>
- [16] <https://medium.com/@sadirgazada20/depth-first-search-dfs-c8658b7c2421>
- [17] <https://furkangulsen.medium.com/depth-first-search-algorithm-8dc6b280510b>








## BolgeVeDil

Sınıf

### Alanlar

-  dosyaYolu
-  edges
-  users

### Metotlar




-  AddUser
-  BFSForInterests
-  BolgeVeDil
-  GetNeighborN...
-  ProcessUserData

### İç içe Olan Türler






## interestGraph

Sınıf

### Alanlar

-  dosyaYolu
-  edges
-  users

### Metotlar

-  AddUser
-  BFSForInterests
-  GetNeighborN...
-  interestGraph
-  ProcessUserData

### İç içe Olan Türler






## olusanhash<TK...

Genel Sınıf

### Alanlar

-  DefaultCapacity
-  doluluk






### Metotlar

-  Add
-  GetBucketIndex
-  olusanhash
-  PrintKeyValueP...
-  TryGetValue









## UserGraph

Sınıf

### Alanlar

-  dosyaYolu
-  dosyaYolu1
-  dosyaYolu2
-  edges
-  users

### Metotlar









-  AddRelationship
-  AddUser
-  BFS
-  DisplayGraph
-  GetKomsuDug...
-  IsCloseEnough
-  TakipciVeTakipE...
-  UserGraph

### İç içe Olan Türler


## Kullanici

Sınıf

### Özellikler

-  followers
-  followers\_count
-  following
-  following\_count
-  GlobalInterest
-  Language
-  Languagesss
-  Name
-  Region
-  Regionsss
-  Tweets
-  Username

### Metotlar

-  Kullanici



## KullaniciEkleme

Sınıf

### Özellikler

-  Root

### Metotlar


-  DictionaryToLin...
-  SonaEkle

### İç içe Olan Türler

## Program

Sınıf

### Metotlar

-  Main