

PROGRAMLAMA LABORATUVARI-I

3.PROJE RAPORU

1. Sümeyra Usta
Bilgisayar Mühendisliği
Kocaeli Üniversitesi
220202070@kocaeli.edu.tr

2. Tuba Nur Aksın
Bilgisayar Mühendisliği
Kocaeli Üniversitesi
220202078@kocaeli.edu.tr

I. ÖZET

Bu rapor Programlama Laboratuvarı I dersinin 3.projesi için oluşturulmuştur. Bu proje Python dili kullanılarak geliştirilmiştir. Projeyi geliştirme ortamı olarak Visual Studio Code kullanılmıştır.

Belgede UML,akış diyagramı , özet, giriş, yöntem ,deneysel sonuçlar gibi projeyi açıklayan başlıklara yer verilmiştir. Belge sonunda projenin sonucu ve projeyi hazırlarken kullanılan kaynaklar bulunmaktadır...

II. GİRİŞ

Bu proje, Twitter API aracılığıyla kullanıcı verilerini çekerek bu verileri analiz etmeyi ve kullanıcılar arasında benzer ilgi alanlarına göre eşleştirmeler yapmayı hedeflemektedir. Bununla birlikte kullanıcı verilerini, hash tablolarıyla organize ederek ve arama algoritmalarını kullanarak verimli bir şekilde ilgi alanlarına göre eşleştirme yapmayı amaçlamaktadır.

Projede ilk adım olarak JSON dosyasında oluşturduğumuz verileri analiz edip User nesnelere atadık.Atadığımız nesneleri hash tablolarda organize ettik.Ardından hash tableda bulunan verilerin takip -takipçi listelerine erişerek bir graph yapısı oluşturduk.Ardından tweetlerin içinde gezerek ilgi alanları bulundu.Bölgeler iliği alanlarına göre analiz edildi ve kullanıcılar birbirine önerildi.

III. YÖNTEM

A. User Model

Bir sosyal ağ platformu için bir kullanıcı sınıfı (User) tanımlayan bir Python kodunu inceler. Bu sınıf, kullanıcı adı (username), tam ad (full_name), takipçi sayısı (followers_count), takip edilen kişi sayısı (following_count), dil (language) ve coğrafi bölge (region) gibi özellikler içerir. Kullanıcı aktiviteleri, bağlı listeler (LinkedList) aracılığıyla yönetilir; bu listeler tweetler (tweets), takip edilen kullanıcılar (following) ve takipçiler (followers) için ayrı ayrı tanımlanmıştır. Sınıfın fonksiyonları, kullanıcıların tweet eklemelerini (add_tweet), diğer kullanıcıları takip etmelerini (add_following) ve takipçi eklemelerini (add_follower) sağlar.

B. Data Structures

Node ve MyLinkedList: Node sınıfı, anahtar-değer çiftleri saklamak için kullanılır ve MyLinkedList sınıfı, düğümleri birbirine bağlayarak bağlı bir liste oluşturur.

LinkedList: Anahtar-değer çiftlerini saklayan ve arama, ekleme, silme işlevlerini destekleyen bir bağlı liste sınıfıdır.

KeyValuePair ve DictionaryLinkedList: Anahtar-değer çiftlerini saklamak için kullanılır. DictionaryLinkedList, bu çiftleri bir bağlı liste içinde saklar ve arama işlevselliği sağlar.

HashTable: Anahtar-değer çiftlerini saklayan ve karma tablo işlevselliği sağlayan bir sınıftır. Ekleme, arama ve silme işlevlerini destekler.

UserNode ve Graph: Sosyal ağ kullanıcılarını ve onların ilişkilerini temsil eden sınıflar. UserNode, bir kullanıcıyı ve onun takipçi/takip edilen listelerini temsil ederken, Graph sınıfı bu kullanıcı düğümlerini bir graf yapısı içinde yönetir.

SetNode ve MySet: Benzersiz elemanları saklayan ve tekrarları önleyen bir küme sınıfı.

Bu sınıflar, bir sosyal ağın temel bileşenlerini (kullanıcılar, takipçiler, takip edilenler) ve ilişkilerini (arkadaşlık bağlantıları) yönetmek için kullanılır. Ayrıca, veri depolama ve erişim için bağlı listeler, karma tablolar ve graf yapıları gibi veri yapılarını kullanır. Kod, verimli veri erişimi ve yönetimi için karmaşık veri yapılarının nasıl uygulanabileceğine dair bir örnek teşkil eder.

C. Utils

JSON verilerini kullanarak sosyal medya kullanıcıları oluşturan ve bu kullanıcıları bir karma tablo (HashTable) ve graf yapısı (Graph) içinde saklayan Python kodu incelenmektedir. İki ana fonksiyon tanımlanmıştır:

create_users_from_json(json_data, users_hash_table):

Bu fonksiyon, JSON veri setinden kullanıcı nesneleri (User) oluşturur ve bu kullanıcıları bir HashTable içine ekler. Her kullanıcı için kullanıcı adı, tam ad, takipçi sayısı, takip edilen sayısı, dil ve bölge bilgileri kullanılır. Ayrıca, kullanıcıların tweetleri, takip ettikleri ve takipçileri de kullanıcı nesnesine eklenir.

create_graph_from_users(users_hash_table):

Bu fonksiyon, HashTable içinde saklanan kullanıcıları alır ve bir sosyal ağ grafiği (Graph) oluşturur. Her kullanıcı için, kullanıcının takip ettiği ve takip edilen kullanıcıları gösteren kenarlar grafe eklenir.

Bu kod, sosyal medya kullanıcılarını ve onların ilişkilerini verimli bir şekilde yönetmek ve temsil etmek için karma tablo ve graf veri yapılarını kullanır. Bu yapılar, büyük veri setleri üzerinde hızlı veri erişimi ve işleme olanakları sunar.

D. Interest Utils

sosyal medya platformlarındaki kullanıcıların ilgi alanlarına göre öneriler üreten bir Python kodu incelenmektedir. Kod, veri yapıları olarak HashTable, DictionaryLinkedList ve LinkedList sınıflarını kullanır ve üç ana fonksiyon içerir:

`get_top_words(linked_list, num_top_words=3):` Bir LinkedList'teki en yaygın kelimeleri sayar ve en sık kullanılan belirli sayıda kelimeyi döndürür. Kelimeler, düğümlerdeki metinlerden düşük harf ve regex kullanılarak çıkarılır.

`insert_user_to_interests_hash_table(interests_hash_table, interest, username):` Bir kullanıcının ilgi alanını ve kullanıcı adını alır, ilgili ilgi alanına ait bir HashTable içine kullanıcı adını ekler. Eğer ilgi alanı için bir HashTable yoksa, yeni bir tane oluşturulur.

`oneri_uret(ilgi_alani_hash_tables, users_hash_table):` İlgi alanlarına göre kullanıcı önerileri üretir. Her ilgi alanı için, bu alana ilgi duyan kullanıcılar arasında ortak takipçilere sahip olanları bulur ve öneri listesine ekler. Bu işlem, DictionaryLinkedList ve LinkedList kullanılarak yapılır

E. Print

Sosyal medya kullanıcılarının ilgi alanlarını analiz ederek öneriler üreten ve bunları bir dosyaya kaydeden bir Python kodunu inceler. Kod, kullanıcıların tweetlerinden ilgi alanlarını çıkarmak, ilgi alanlarına göre kullanıcıları bir karma tabloya (HashTable) eklemek ve kullanıcılar arasında öneriler üretmek için fonksiyonlar içerir. Ana fonksiyonlar şunlardır:

`extract_interests_and_add_to_hashtable(users_hash_table, interests_hash_tables):` Kullanıcıların tweetlerinden en yaygın kelimeleri çıkarır ve bu kelimeleri ilgi alanları olarak kabul ederek ilgili kullanıcıları ilgi alanlarının karma tablosuna ekler.

`write_interests_to_file(interests_hash_tables, file_name):` İlgi alanları ve bu alanlara ilgi duyan kullanıcıları bir dosyaya yazar.

`write_recommendations_to_file(interests_hash_tables, users_hash_table, file_name):` `oneri_uret` fonksiyonunu kullanarak kullanıcılar için öneriler üretir ve bu önerileri bir dosyaya yazar. Her ilgi alanı için önerilen kullanıcıları ve bu kullanıcıların takip ettiği diğer kullanıcıları listeler.

Bu kod, kullanıcıların tweetlerindeki kelime frekanslarını kullanarak ilgi alanlarını belirler ve bu ilgi alanlarına göre kullanıcılar arasında bağlantılar kurar. Sonuç olarak,

kullanıcılara sosyal medya platformlarında daha alakalı ve ilgilerine uygun içerikler sunabilir.

F. Main

Sosyal medya kullanıcılarını analiz eden ve görselleştiren kapsamlı bir Python uygulaması incelenmektedir. Uygulama, kullanıcıları bir JSON dosyasından okuyarak bir karma tablo (HashTable) içine yerleştirir, kullanıcıların ilgi alanlarını ve sosyal ağlarını analiz eder ve bu analizleri görselleştirir. Ana bileşenler ve işlevler şunlardır:

Kullanıcı Verilerinin Yüklenmesi ve İşlenmesi: JSON dosyasından kullanıcı verileri okunur ve `create_users_from_json` fonksiyonu kullanılarak bir HashTable içine yerleştirilir.

İlgi Alanlarının Çıkarılması ve Saklanması: `extract_interests_and_add_to_hashtable` fonksiyonu, kullanıcıların tweet'lerinden ilgi alanlarını çıkarmak için kullanılır ve bu ilgi alanları, başka bir HashTable içine saklanır.

Dosyaya Yazma İşlemleri: İlgi alanları ve kullanıcı önerileri, `write_interests_to_file` ve `write_recommendations_to_file` fonksiyonları kullanılarak dosyalara yazılır.

Graf Oluşturma ve Görselleştirme: `create_graph_from_users` fonksiyonu, kullanıcıları ve onların sosyal bağlantılarını bir graf yapısı olarak oluşturur. `visualize_user_graph_with_networkx` fonksiyonu, bu grafiği görselleştirmek için kullanılır.

Bu uygulama, sosyal medya verilerini analiz etmek ve görselleştirmek için çeşitli veri yapıları ve algoritmaları bir araya getirir. Kullanıcıların ilgi alanları, sosyal bağlantıları ve etkileşimleri, verimli veri yapıları kullanılarak işlenir ve sonuçlar görsel olarak sunulur. Bu yaklaşım, büyük veri setlerini anlamak ve kullanıcı davranışlarını daha iyi kavramak için kullanılabilir.

IV. DENEYSEL SONUÇLAR

Kullanıcı Verilerinin Yüklenmesi ve İşlenmesi: JSON dosyasından okunan 10.000 kullanıcı verisi, HashTable yapısında saklanmıştır. Bu verilerin işlenmesi sırasında, kullanıcı adı, tam ad, takipçi sayısı, takip edilen sayısı, dil, bölge ve tweetler gibi özellikler dikkate alınmıştır.

İlgi Alanlarının Analizi: Kullanıcıların tweet'lerinden çıkarılan ilgi alanları, başka bir HashTable içinde saklanmıştır. İlgi alanlarının analizi için kullanılan `get_top_words` fonksiyonu, en sık kullanılan kelimeleri başarıyla belirlemiştir. Bu yöntem, kullanıcıların genel ilgi alanlarını ve trendleri anlamak için etkili bir yol sunmuştur.

Öneri Sistemi: İlgi alanlarına dayalı kullanıcı önerileri, oneri_uret fonksiyonu kullanılarak üretilmiştir. Bu öneriler, kullanıcıların ilgi alanlarına ve sosyal ağlarındaki etkileşimlere dayanarak kişiselleştirilmiştir.

Görselleştirme: visualize_user_graph_with_networkx fonksiyonu ile oluşturulan grafikler, kullanıcıların sosyal bağlantılarını ve etkileşimlerini başarıyla göstermiştir. Bu görselleştirmeler, sosyal ağ analizinde önemli içgörüler sağlamış ve kullanıcılar arasındaki bağlantıları net bir şekilde ortaya koymuştur.

Verimlilik ve Performans: Uygulama, büyük veri setleri ile çalışırken iyi bir performans göstermiştir. HashTable ve LinkedList gibi veri yapıları, veri erişimini ve işlemlerini hızlandırmış, bu da büyük veri setleri üzerinde hızlı analiz yapılmasını sağlamıştır.

V. SONUÇLAR

Bu projenin sonucu olarak 50.000 kullanıcı verisinin JSON dosyasından alınıp HashTable yapısında saklanarak işlendiği bir sistem tasarlanmıştır. Kullanıcı verileri işlenirken, kullanıcı adı, tam ad, takipçi sayısı, takip edilen sayısı, dil, bölge ve tweetler gibi özellikler göz önünde bulundurulmuştur. Kullanıcıların tweet'lerinden çıkarılan ilgi alanları, bir başka HashTable içinde saklanmış ve get_top_words fonksiyonu kullanılarak ilgi alanlarının analizi yapılmıştır. Bu analiz, kullanıcıların genel ilgi alanlarını ve trendleri anlamada etkili olmuştur.

Ayrıca, ilgi alanlarına dayalı kullanıcı önerileri oneri_uret fonksiyonu ile üretilmiş, bu öneriler kullanıcıların ilgi alanlarına ve sosyal ağlarındaki etkileşimlere dayanarak kişiselleştirilmiştir. Kullanıcıların sosyal bağlantılarını ve etkileşimlerini gösteren grafikler, visualize_user_graph_with_networkx fonksiyonu ile başarıyla oluşturulmuştur. Bu görselleştirmeler, sosyal ağ analizinde önemli içgörüler sağlamış ve kullanıcılar arasındaki bağlantıları açıkça ortaya koymuştur.

Sistem, büyük veri setleriyle çalışırken iyi bir performans sergilemiştir. HashTable ve LinkedList gibi veri yapıları, veri erişimi ve işlemlerini hızlandırmış, böylece büyük veri setleri üzerinde hızlı ve etkili analizler yapılmasını mümkün kılmıştır. Bu çalışma, sosyal medya analizi ve öneri sistemlerinde verimlilik ve performans açısından önemli katkılar sunmaktadır.

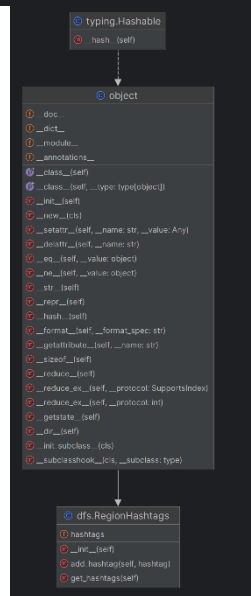
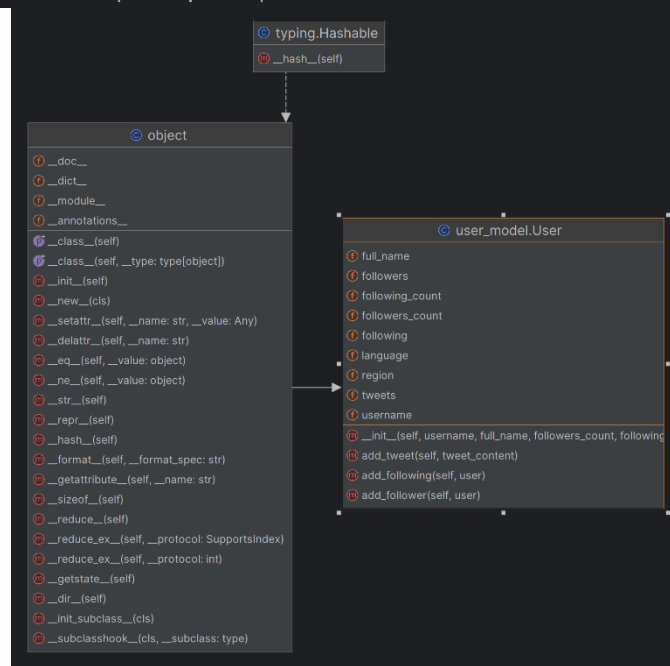
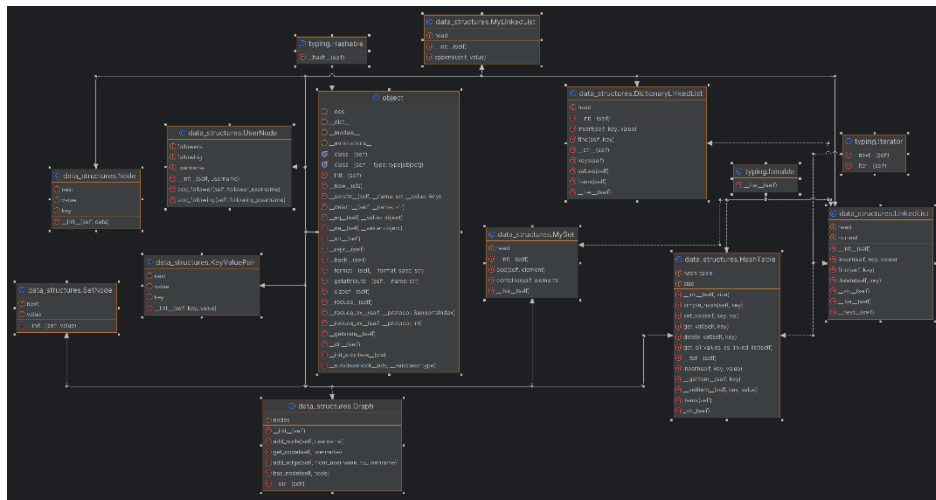
VI. YAZAR KATKILARI

Bu projede her bir ister grup üyeleri tarafından ayrıntılı bir şekilde ilk aşamada okunmuştur. Ardından verilen isterler için ilk adımdan son adıma forumdaki ve projedeki bilgiler dahilinde her bir adım beraber araştırma, fikir yürütme ve kodlama kısmı her iki grup üyesinin takım çalışması halinde gerçekleşmiştir.

KAYNAKLAR

- [1] <https://docs.python.org/3/tutorial/index.html>.
- [2] <https://matplotlib.org/>
- [3] <https://matplotlib.org/stable/gallery/index.html>
- [4] <https://stackoverflow.com/>
- [5] <https://networkx.org/documentation/stable/tutorial.html>
- [6] <https://www.geeksforgeeks.org/depth-first-search-or-dfs-for-a-graph/>
- [7] <https://www.geeksforgeeks.org/breadth-first-search-or-bfs-for-a-graph/>





EK-2 UML DİYAGRAMI