110201001 Ramazan Güçlü-110201053 Feyyaz Özhan-110201035 Burak Utkucu

DURAK SİSTEMİ PROJESİ

1.Giriş

Akıllı Durak Sistemi (ADS), raylı sistem ve karayollarında, toplu taşıma uygulamalarında araçlarla duraklar arasında kurduğu akıllı bir otomasyon sayesinde, duraklarda bekleyen yolcuların gidecekleri güzergaha ait aracın ne zaman geleceği bilgisini öğrenmelerini sağlamaktadır. Ayrıca merkezde kurulan bir sistem odası sayesinde de toplu taşıma organizasyonunun sağlıklı bir şekilde gerçekleştirilmesine olanak tanımaktadır.

Bu sistem yardımıyla;

· Toplu ulaşım hatlarının hangi güzergâhlardan geçtiği,

· Otobüs hatlarına ait sefer zamanları,

· Toplu ulaşım aracının beklenen ya da belirtilen durağa tahmini olarak kaç dakika sonra geleceği,

· Belirtilen adrese en yakın duraklar öğrenilebilir.

Bizim bu projedeki amacımız ise küçük çaplı bir akıllı durak sistemini gerçeklemeye çalışmaktır. İki tane hatta ait (23 ve 24) tüm durakların listelenmesi, her duraktan geçen diğer hatların listelenmesi, otobüslerin sefer saatlerinin görüntülenmesi, koordinatları belirtilen duraklara harita üzerinde yakınlaşılması ve son olarak iki hatta ait güzergahların çizdirilmesi işlemlerini gerçeklemeye çalıştık.

Projemiz istediğimiz analizlerin gerçekleşmesini büyük oranda başarmıştır. En önemlisi projemiz, her durağın yanında bulunan ne zaman gelecek butonuna tıklayan kullanıcıya, otobüsün geçmiş verilerini kullanarak kaç dakika sonra geleceğini doğru olarak hesaplamıştır. Bunun dışındaki diğer analiz işlemlerini de başarmıştır. Örneğin durakların listelenmesi, sefer saatlerinin görüntülenmesi gibi işlemlerin hepsini gerçekleşmiştir.

2.Temel Bilgiler

3.Diğer Bölümler

3.1 Verileri Okuma

Akıllı durak sistemi projemiz her durakta bulunan ne zaman gelecek butonuna tıklandığında doğru sonuçlar üretmektedir. Bu işlemi sağlarken ilk olarak butona

tıklandığında otobüsün o duraktaki geçmiş verilerini kullanmalıdır. Otobüslerin her durak için geçmiş verilerini oluşturduk ve projenin klasörlerine ornek\_dev.txt olarak kaydettik. Bu. txt uzantılı dosyamızın içinde ID,GecKalma,Neden,Havadurumu,DOGS,Hatno,DurakNo verileri bulunmaktadır. ID her örnek veri için farklıdır. GecKalma ise Trafik, Ariza, Yolcuyoğunluğu ve Kaza olmak üzere 4 farklı veriden oluşmaktadır. Havadurumu ise Normal, Karlı ve Yağışlı olmak üzere 3 farklı veriden oluşmaktadır. DurakNo ise her hat için farklı değerlerden oluşmaktadır.

Dosyadan verileri okumak için StreamReader sınıfında bir nesne oluşturduk. Okuyacağı .txt uzantılı dosyanın yolunu da kod içinde belirttik. Okuduğumuz veriler üzerinde işlem yapabilmek için datatable türünde bir nesne daha oluşturduk. Bu nesne sayesinde okuduğumuz dosyanın içindeki yaklaşık 2000 tane veriyi tutmuş bulunmaktayız.

3.2.Naive Bayes Sınıflandırıcısı

Projemizde kullandığımız Naive Bayes sınıflandırıcı her durak için oluşturmuş olduğumuz verileri kullanarak bir değer döndürmektedir.Örneğin otobüsün geç kalma nendi arıza ,hava durumu da normal değerlerini içersin.Bu sınıflandırıcı sayesine ilk olarak oluşturduğumuz veriler arasında her durak için toplam on adet veri arasından nedeni arıza olanların sayısını bulacaktır.Ardından her sınıf ( projemizde zaman aralığı)İçin o sınıf koşullarını sağlayan arıza nedeni sayısını bulmaktadır.Bu sayıyı , toplam on adet veri arasından nedeni arıza olanların sayısına bölerek ilk çarpanı elde etmiş oluruz.

Her bir durak için bulunan on adet veri arasından hava durumu normal olan değerlerin sayısını bulacaktır.Ardından her sınıf için o sınıf koşullarını sağlayan normal hava durumu nedeni sayısını bulmaktadır. Bu sayıyı , toplam on adet veri arasından hava durumu normal olanların sayısına bölerek ikinci çarpanı elde etmiş oluruz.

Son olarak da her sınıf için o sınıftan kaç adet bulunuyorsa bu sayıyı ,her durak için bulunan veri seti sayısına bölerek üçüncü çarpan değeri de bulunmuş olur.

Üç çarpanı çarparak her sınıf için bir değer elde ederiz.Bu değerler arasından en büyük olanı seçtikten sonra otobüsün o durak için geç kalma dakikası sınıf değerine eşit olur.

3.3 Fonksiyonlar

deger\_uret(); Projenin global değişkenleri arasına, olası hava durumu değerleri(Normal, Karlı ve Yağışlı) ve olası nedenleri(Trafik, Ariza, Yolcu yoğunluğu Kaza) tutmak için iki adet dizi tanımladık. Bu fonksiyon ilk olarak Random sınıfını kullanarak rastgele değer üretir. Ardından belirtmiş olduğumuz iki dizi içinden rastgele oluşturduğu sayıya göre neden ve hava durumu değişkenlerini alır.

veri\_alma(); Diğer bölümler içinde bulunan verileri okuma işleminin gerçekleştiği yerdir.

algoritma1(); Öncelikle yedi adet değişken oluşturduk.

Sonuc ,x,y,z, eleman, satir\_say2, satir\_say3.

Örneğin otobüsün geç kalma nendi arıza ,hava durumu da normal değerlerini içersin.Bu fonksiyon sayesinde ilk olarak oluşturduğumuz veriler arasında her durak için toplam on adet veri arasından nedeni arıza olanların sayısını bulduk. Bu sayıyı satır\_say2 değişkenine atadık.Ardından her sınıf ( projemizde zaman aralığı)İçin o sınıf koşullarını sağlayan arıza nedeni sayısını bulduk.Bu sayıyı x değişkenine atadık.

Her bir durak için bulunan on adet veri arasından hava durumu normal olan değerlerin sayısını bulduk. Bu sayıyı satır\_say3 değişkenine atadık.Ardından her sınıf ( projemizde zaman aralığı)İçin o sınıf koşullarını sağlayan arıza nedeni sayısını bulduk.Bu sayıyı y değişkenine atadık.

Son olarak da her sınıf için o sınıftan kaç adet bulunuyorsa bu sayıyı z değişkenine atadık.

Sonuc değişkenini bulmak üzere ilk olarak x i satir\_say2 ye böldük.Ardından y yi satir\_say3 e böldük. Ve z yi eleman sayısına böldük.Son olarak bu değerleri birbiriyle çarparak her sınıf için geç alma olasılığını bulduk.

geckalmabul();Her otobüsün geç kalma dakikası çok fazla olmasından işlem kolaylığı için aralıklara böldük. Toplamda bir saatlik geç kalma ihtimalini göz önünde bulundurduğumuzdan her birinin büyüklüğü beş olmak üzere on iki adet aralık oluşturmuş olduk.

Projemizde bu her aralığın gelme ihtimali birbirinden farklı olacağındanon iki adet algoritma gerçekleştirdik ve her birini ayrı fonksiyonuna gönderdik.Bu on iki algoritmadan dönen sonuçları tutmak için sonuclar adında on iki elemanlı bir dizi oluşturduk.Bu fonksiyon sonuclar dizisindeki on iki adet sonuçtan en büyük olan değeri bulmaktadır.

saaticek();Kullanıcının tablodaki duraktan herhangi bir otobüs için ne zaman gelecek butonuna tıkladığında o satırda bulunan durak sırasını almaktadır.Bu sırayı kullanarak txt uzantılı dosyamızdan otobüsün o durakta olması gereken saati bulmaktadır.

geckalmabul();En yüksek olasılığa sahip olan sınıfı yukarıda bulmuştuk . Örneğin bu sınıfın aralığı 0-5 arasında ise geç kalma dakikasını üç olarak ayarlayacaktır. Bunun gibi her sınıf için bir adet geç kalma dakikası ayarlanacaktır.

geckalmaekle();Yukarıda bulmuş olduğumuz saatin üzerine geç kalma dakikasını ekleyerek otobüsün saat kaçta geleceğini hesaplar.

Kaynakça

http://www.kocyigityunus.com/category/google-maps/

http://www.kocyigityunus.com/google-maps-api-1-kurulum/

http://www.kocyigityunus.com/google-maps-api-2-marker-isaretci/

http://zxingnet.codeplex.com/

http://en.googlemaps.subgurim.net/

https://developers.google.com/maps/documentation/javascript/v2/v2tov3?hl=tr&csw=1

https://maps.google.com/

http://ulasimrehberi.kocaeli.bel.tr/