

**EN KISA YOLUN BULUNMASI ALGORİTMASI**

YAPAY ZEKA FİNAL PROJESİ

YUSUF TERZIOGLU

18MY93014

© YUSUF TERZIOGLU

Starbucks'tan çıkan insanları yolcu kabul edelim, bir taksicinin de en yakın yolcuyu bulmasını sağlayalım. Taksicinin konumu bir sabit olarak alalım , buradaki yolcuların koordinaatlarını taksiye yapılan istek olarak belirleyelim. Taksiciye de onun konumuna en yakın olan yolcuları gösterelim.

Uygulamanın kısaca yapılış amacı buna hizmet ediyor. Yani şu şekilde , eğer bir yazılımcıysanız ve bu şekilde bir webservise ihtiyacınız varsa, yani yolcuların konumu ve taksicinin konumu eğer elinizdeyse bu backend mantığında yazılmış servisi kullanıp istediğiniz verilere kolayca erişebilirsiniz.

Kısım 1

Öncelikle math kütüphanesini kullanacağımız için bir import işlemi yapmamız gerekiyor şu şekilde ;

**from** **math** **import** sin, cos, sqrt, atan2, radians

Sonrasında pandas adında bir kütüphane kullanacağız elimizdeki veriyi kodun içine aktarabilmek için onuda hemen import edelim ;

**import** **pandas** **as** **pd**

**Sonrasında eklediğimiz dosyayı bir değişkene aktaralım ;**

df = pd.read\_csv(‘directory.csv’)

Bu işlemi de bitirdikten sonra sürücümüzün koordinatlarını verelim ;

driver\_cordinates = [{“lat":40.912522,"long":42.874365}]

Az önce içeriye aldığımız dosyada dünyada ki bütün starbucksların konumu bulunmaktaydı. Şu anda Türkiye’de bulunduğumuz için bu dosyayı filtreleyip sadece türkiyede bulunan koordinatları alıyoruz ;

df=df[df[‘Country']=='TR'].reset\_index()

Sonrasında skala’mızı daha da küçültebilmek için konum alanımızı istanbul’a indirgiyoruz;

df = df[df[‘City']=='Istanbul'].reset\_index()

Ve bu filtrelenmiş datanın içindeki koordinat değerlerini alıyoruz;

df = df[[‘Latitude','Longitude']]

Ve mesafe adında boş bir array oluşturuyouz;

mesafe = []

Sürücümüzün koordinatlarını bildiriyoruz;

driver\_cordinates = [{“lat":41.0429528,"long":28.9970722}]

Sonrasında bu aldığımız verileri bir döngüye sokup uzunluğu kadar döndürüyoruz;

**for** i **in** range (0,len(df)):

Elimizdeki verilerin koordinatlarını almıştık , bunları kilometre cinsine çevirebilmek için sabit bir sayımız var ;

R = 6373.0

Öncelikle taksicinin koordinatlarını km cinsine çeviriyoruz;

lat1 = radians(driver\_cordinates[0]['lat'])

lon1 = radians(driver\_cordinates[0][‘long'])

Sonrasında filtrelediğimiz datayı;

lat2 = radians(df['Latitude'][i])

lon2 = radians(df['Longitude'][i])

Ardından datadan gelen km cinsindeki veriyi taksicinin kilometresinden çıkartıyoruz;

dlon = lon2 - lon1

dlat = lat2 - lat1

Sonrasında gerekli matematik işlemlerimizi yaparak arasındaki mesafeyi bulmuş oluyoruz;

a = sin(dlat / 2)\*\*2 + cos(lat1) \* cos(lat2) \* sin(dlon / 2)\*\*2

c = 2 \* atan2(sqrt(a), sqrt(1 - a))

Sonra bulduğumuz bu uzaklığı bir değere atıyoruz ve bu değerimizin adı distance ;

distance = R \* c

yakinlik = {"passengerId":i+1,"km between driver and passenger":round(distance,10)}

mesafe.append(yakinlik)

Sonrasında bu mesafe datamızın içindeki minimum değerleri alıyoruz ;

df\_mesafe = pd.DataFrame(mesafe)

min\_distance = df\_mesafe['km between driver and passenger’].min()

En kısa yol bulunduktan sonra bunu ekrana bastırıyoruz ;

df\_mesafe = df\_mesafe[df\_mesafe['km between driver and passenger']==min\_distance]

print(df\_mesafe)

Burada ekrana bastırmak yerine bu olay bir webservise çevrilebilir. Bir apiye istek halinde veriler işlenerek response döndürülebilir.