Navigation Records Veri Analiz Raporu ve Sonuçları (Task1)

1. Projeye Genel Bakış

Bu proje, rota bazlı seyahat verilerinde toplam mesafe ve süreyi hesaplamayı hedeflemektedir. Veride cihaz hataları, ağ arızaları, veya cihazın yeniden başlatılması gibi nedenlerle mesafe ölçümlerinde sıfırlanma gibi anormallikler ve veri kesintileri olabilir. Bu nedenle, analizde ve SQL sorgusunda anormal değerleri ele almak ve toplam mesafeyi doğru bir şekilde hesaplamak esastır.

2. Veri Tanımlaması

Elimizde navigation_records.csv adlı veri seti bulunmakta ve aşağıdaki üç sütundan oluşmaktadır:

•**route_id :** Rotanın benzersiz kimliği.

•**distance :** Rotanın o ana kadar katedilen kümülatif mesafesi.

•**recorded_at :** Mesafe kaydının alındığı tarih ve saat.

Verinin ilk incelemesi sonucunda, bazı kayıtların olağandışı veya hatalı olduğunu gözlemledik. Örneklerde olduğu gibi, rotalar sıfırlanmış mesafe değerleri ve eksik veri noktaları içerebilir .

Route_id	Recorded_at	Distance
100057	2020-02-20 02:22:58	0
151768	2020-11-03 00:51:12	0

Route_id	Recorded_at	Distance	Next_distance
100057	2020-02-20 02:24:45	34730988	347326
100057	2020-02-20 02:25:13	34736428	3474144

3. Veri Analizi ve Anormallik Tespiti

3.1 Anormallik Tespiti

Anomali analizi için, rota verilerinde sıfırlanmış veya tutarsız mesafe değerlerini incelemek üzere çeşitli analizler yapıldı:

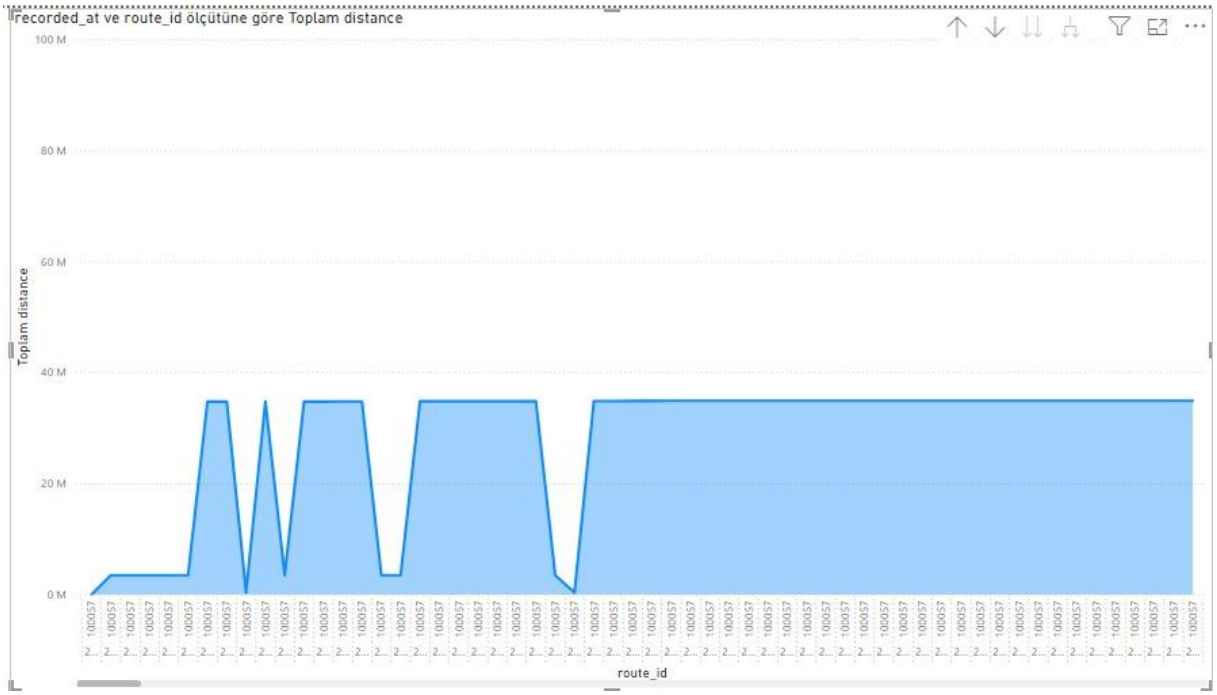
• Sıfırlanma Analizi: mesafe sütununda sıfırdan daha küçük veya önceki kayıttan düşük olan değerlere odaklandık. Bu durum genellikle rota sıfırlamalarına işaret eder.

• Tutarsız Değerler: Rotalarda mesafe değeri sıradışı artış veya azalış gösteriyorsa, bu tür değerler de anormal olarak sınıflandırıldı.

3.2 Veri Görselleştirmesi

Anormalliklerin görselleştirilmesi için şu grafikleri oluşturduk:

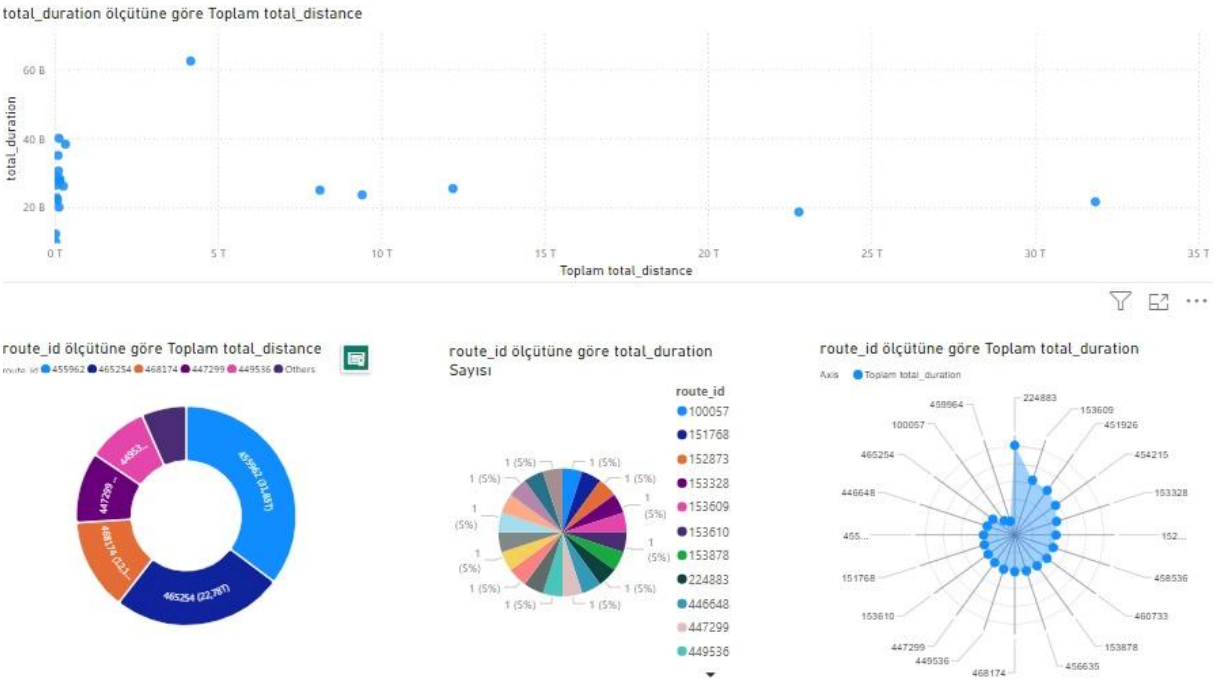
Bu görseller, toplam seyahat mesafesini hesaplarken hangi rotalarda ve kayıt noktalarında anormalliklerin bulunduğunu hızlıca gözlemlememize olanak tanıdı.



Şekil1

Power BI'da alan grafiği kullanarak veri setinizi görselleştirirken, x ekseninde recorded_at ve route_id, y ekseninde ise toplam mesafe (total distance) yer alarak, zamanla rotalarda katedilen toplam mesafeyi gözlemlediniz. Bu yaklaşım, verideki sıfırlanma ya da ani mesafe değişimleri gibi anormalliklerin kolayca tespit edilmesini sağladı. Alan grafiği, kümülatif mesafe değerlerinde ani düşüşler veya sıçramalar gibi beklenmedik paternlerin gözlemlenmesine olanak tanır; böylece ağ veya cihaz hataları gibi sorunları kolayca tespit edebildiniz. Bu grafikle elde edilen görsel, temizleme işlemlerine geçmeden önce verideki anomalileri belirlemek için oldukça yararlı bir araç olmuştur.

Veri setini düzenledikten sonra oluşturulan grafikler;



Şekil2

Total Duration ve Toplam Distance Dağılım Grafiği: Bu grafik, x ekseninde toplam mesafeyi (total distance) ve y ekseninde toplam süreyi (total duration) göstermektedir. Grafik, her rota için toplam mesafe ve süre ilişkisini görselleştirir. Grafik üzerinde yer alan kümelenmiş veri noktaları, belirli rotalarda kaydedilen mesafe ve sürelere dair genel bir bakış sunar.

Donut Chart – route_id'ye Göre Toplam Total Distance: Bu donut grafikte, her route_id için toplam mesafe oranları görselleştirilmiştir. Grafikte yer alan dilimlerin büyüklüğü, her rotanın toplam mesafeye olan katkısını gösterir. Örneğin, bazı route_idlerin toplam mesafeye daha büyük bir katkı sağladığı, yani daha uzun mesafeler katedildiği gözlemlenebilir. Bu grafik, rotalar arasındaki mesafe farklarını kıyaslamak için yararlıdır ve bazı rotaların olağan dışı uzun ya da kısa mesafeler kaydettiğini hızlıca fark etmenize yardımcı olur.

Pie Chart – route_id'ye Göre Total Duration Sayısı: Bu pasta grafikte, her route_id için toplam süre sayısı oranları gösterilmektedir. Her dilim, belirli bir rotanın toplam süre sayısına katkısını temsil eder. Grafikten, belirli route_idlerin toplam sürede daha fazla orana sahip olduğunu görebilirsiniz. Bu, bazı

rotalarda daha fazla veya daha az veri kaydının olduğunu gösterebilir ve bazı rotalarda eksik veri olabileceğine dair ipucu verebilir.

Radar Chart – route_id'ye Göre Toplam Total Duration: Bu radar grafikte, her route_id için toplam sürelere göre bir dağılım gösterilmiştir. Her bir route_id için toplam süre (total duration) değerleri, grafikte farklı eksenlerde yer almaktadır. Bu görselleştirme, rotalar arasındaki süre dağılımını karşılaştırmak için oldukça kullanışlıdır. Süre bakımından öne çıkan veya çok düşük kalan rotalar kolayca fark edilebilir, bu da bazı rotalarda anormal uzun veya kısa seyahat süresi olduğunu gösterir.

4.Oluşturulan Veri setleri;

Navigation_records.csv deki veriler baz alınarak yeni veri setleri elde edildi. Bu verileri python kodları yardımı ile oluşturuldu , bu kodları dizinin içerisindeki klasör adına göre ulaşabilirsiniz

4.1 navigation_records_analysis_anomaly_results.csv;

Route_id	Recorded_at	Distance	Next_distance
100057	2020-02-20 02:2	34730988	347326
100057	2020-02-20 02:2	34736428	3474144
100057	2020-02-20 02:2	34766536	3477801
100057	2020-02-20 02:2	34792212	3479923
100057	2020-02-20 02:2	3479923	348153
100057	2020-02-20 02:3	34904792	3490607
100057	2020-02-20 02:4	34935304	3494575
100057	2020-02-20 02:4	34950188	3495105

Bu veri setini oluştururken Mesafe farkını önceki satıra göre hesaplıyoruz. Eğer mesafe farkı negatifse burada bir anormallik olduğunu görüyoruz .

4.2 navigation_records_zero_anomaly_results.csv;

Route_id	Recorded_at	Distance
100057	2020-02-20 02:22:58	0
151768	2020-11-03 00:51:12	0
152873	2020-11-07 13:50:18	0
152873	2020-11-07 13:49:59	0
152873	2020-11-07 05:52:08	0
152873	2020-11-07 05:51:33	0
153328	2020-11-09 10:55:43	0
153328	2020-11-09 05:49:37	0
153328	2020-11-09 05:49:17	0

Bu veri setini oluştururken navigation_records.csv veri setindeki distance sütunundaki sıfır olan değerleri listeledik.

4.3 update_navigation.csv;

Route_id	Total_distance	Total_duration
100057	3127016168	12387
151768	8576522253	22185
152873	8420255834	28796
153328	11732152089	30617
153609	13543046921	40096
153610	7039764659	22854

Bu veri setinde anormallikleri tespit ettikten sonra , istenen gibi toplam yol ve toplam süreleri oluşturduk

5 SQL Sorgusu ile Anormallik tespiti

Bu kodda, SQL Server'daki navigation_records tablosundan mesafe verilerini analiz etmek için iki ana adım izlenmektedir. Kodun tamamında kullanılan adımları açıklayalım:

5.1 İlk Sorgu: distance = 0 Değerlerinin Tespiti

Bu sorgu, navigation_records tablosundaki distance değeri "0" olan tüm satırları getirir. Mesafe kaydının "0" olması, potansiyel olarak bir veri hatasını veya cihaz kaynaklı bir sorunu işaret edebilir. Bu tür veriler, analiz ve raporlama aşamalarında dikkate alınması gereken anormal durumlar olarak değerlendirilir.

sql

```
SELECT * FROM navigation_records WHERE distance = 0;
```

Bu sorgu ile elde edilen sonuçlar, veri kümesindeki sıfır mesafe değerlerini içerir ve “navigation_records_zero_anomaly_results.csv” dosyasına kaydedilmiştir.

5.2 İkinci Sorgu: Anormal Mesafe Değerlerinin Tespiti

İkinci sorgu, ardışık mesafe değerlerini kontrol ederek bir anormallik arar. LEAD fonksiyonu kullanılarak, distance değeri bir sonraki kayıtla karşılaştırılır ve daha büyük bir distance değerinin kendisinden sonra daha küçük bir değere dönmesi durumu tespit edilir. Bu tür durumlar, rota üzerindeki tutarsızlıkları gösterebilir ve sıklıkla ağ hataları, cihaz sıfırlamaları veya sensör hataları gibi teknik nedenlerden kaynaklanır.

sql

```
WITH AnomalyCheck AS (
    SELECT route_id , recorded_at , distance,
        LEAD(distance) OVER (ORDER BY recorded_at) AS next_distance
    FROM navigation_records
)SELECT *FROM AnomalyCheck WHERE distance > next_distance;
```

Bu sorgu ile tespit edilen anormal durumlar, “navigation_records_analysis_anomaly_results.csv” dosyasına kaydedilmiştir. Bu kayıtlar, rota verisindeki ardışık tutarsızlıkların detaylı analizi için kullanılabilir. Anormal mesafe değerlerini etkin bir şekilde filtreleyip, doğru toplam mesafe ve süreyi hesaplayacak bir algoritma geliştirilmiştir.

6 SQL Sorgusu ile total_distance ve total_duration değerlerini bulma

Bu SQL kodu, navigation_records tablosundaki her bir route_id için toplam mesafe ve süreyi hesaplamak üzere tasarlanmıştır. Aşağıda adım adım açıklamalar verilmiştir:

6.1 İlk Adım: Veriyi Sıralı Numaralandırma

ROW_NUMBER() fonksiyonu kullanılarak her bir route_id ve recorded_at alanına göre sıralı bir numara atanır. Bu işlem, rota verilerini zaman sırasına göre analiz edebilmek için gereklidir.

sql

```
WITH SortedData AS (
   SELECT
      route_id,
      recorded_at,
      distance,
      ROW_NUMBER() OVER(PARTITION BY route_id ORDER BY recorded_at) AS rn
   FROM navigation_records),
```

6.2 İkinci Adım: Mesafe Farkı Hesaplama

Bu adımda, LAG() fonksiyonu yardımıyla her bir kaydın mesafesi ile bir önceki kaydın mesafesi arasındaki fark hesaplanır. Böylece rotanın her bir noktada ne kadar değiştiği belirlenir.

sql

```
DistanceDiff AS (
    SELECT
        sd.route_id,
        sd.recorded_at,
        sd.distance,
        sd.distance - LAG(sd.distance) OVER (PARTITION BY sd.route_id ORDER BY sd.recorded_at) AS distance_diff
    FROM SortedData sd
),
```

6.3 Üçüncü Adım: Negatif Mesafe Farklarının Düzeltilmesi

Eğer bir mesafe farkı negatifse, bu durum mesafedeki sıfırlanmaları veya cihaz hatalarını gösterebilir. Negatif farklar, bu satırlarda mesafe değeri ile değiştirilir.

sql

```
AdjustedDistance AS (
    SELECT
        dd.route_id,
        dd.recorded_at,
        dd.distance,
        CASE
            WHEN dd.distance_diff >= 0 THEN dd.distance_diff
            ELSE dd.distance
        END AS adjusted_distance
    FROM DistanceDiff dd),
```

6.4 Dördüncü Adım: Kümülatif Mesafe Hesaplama

Bu adımda, her bir route_id için SUM() fonksiyonu kullanılarak kümülatif mesafe hesaplanır. Ayrıca, önceki kayıt zamanı prev_recorded_at olarak alınır, böylece iki zaman damgası arasındaki süre hesaplanabilir.

sql

```
CumulativeDistance AS (
    SELECT
        ad.route_id,
        ad.recorded_at,
        ad.distance,
        SUM(ad.adjusted_distance) OVER (PARTITION BY ad.route_id ORDER BY ad.recorded_at ROWS BETWEEN UNBOUNDED PRECEDING AND CURRENT ROW) AS cumulative_distance,
        LAG(ad.recorded_at) OVER (PARTITION BY ad.route_id ORDER BY ad.recorded_at) AS prev_recorded_at
    FROM AdjustedDistance ad
),
```

6.5 Beşinci Adım: Toplam Mesafe ve Süre Hesaplama

Son olarak, route_id başına en yüksek kümülatif mesafe total_distance olarak ve her iki kayıt arasındaki sürenin toplamı total_duration olarak hesaplanır.

Sql

```
FinalData AS (
    SELECT
        cd.route_id,
        MAX(cd.cumulative_distance) AS total_distance,
        SUM(DATEDIFF(SECOND, cd.prev_recorded_at, cd.recorded_at)) AS total_duration
    FROM CumulativeDistance cd
    GROUP BY cd.route_id
)
```

6.6 Nihai Sonuç;

Her route_id için toplam mesafe ve toplam süre total_distance ve total_duration olarak döndürülür ve CSV dosyasına kaydedilir.

Sql

```
SELECT route_id , total_distance , total_duration

FROM FinalData;
```

Sonuç, “update_navigation.csv” dosyasına kaydedilmiştir ve çıktı olarak görüntülenmiştir. Bu analiz ile rota bazında kümülatif mesafe ve toplam süre hesaplanmış, anormal değerler düzeltilmiştir.