



Son Rapor

Apache Flink ile Nesnelerin İnternetinde Toplanan Verilerin Analizi

Proje Adı

131180009 Bekir AYDIN

131180059 Soner ŞEKERCİ

BM496 BİLGİSAYAR PROJESİ II

Ocak 2018

İçindekiler

1. Projenin Gerçekleştirilmesi	2
1.1. Proje Özeti	2
1.2. Proje Hedefi	2
1.3. Projede Mevcut Aşama	2
1.3.1. Tasarım Kararları	2
1.3.1.1. Sınıfların Oluşturulması	2
1.3.1.2. Pencerelemler oluşturulması	3
1.3.1.3. Pencere Üzerindeki İşlemler	3
1.3.1.4. Harita (Map) Fonksiyonu	4
1.3.1.5. Modelin Kurulması	4
1.3.1.6. Veritabanı Tasarımı	5
1.3.1.7. İstisnai Durumların İdaresi	5
2. TEST	5
2.1. Test Edilen Özellikler	5
2.2. Test Sonuçları	6
2.2.1. Zaman Penceresi	6
2.2.2. Pencere Üzerindeki İşlemler	7
2.2.3. Harita (Map) Fonksiyonu	8
2.2.3.1. Tahmin İşlemi	8
2.2.3.2. Model Kullanımı	8
2.2.4. Modelin Doğruluğu	9
3. Durum Değerlendirmesi	12
3.1. Test Durumları	12
3.2. Projenin Gerçekleştirilmesi	12
3.3. Projenin Sonuçlandırılması	12
4. Projeye Yönelik Gelecek Çalışmaları	12
5. Referanslar	13

1. PROJENİN GERÇEKLEŞTİRİLMESİ

1.1. Proje Özeti

Trafik günümüzün en büyük problemlerinden biridir. Trafiğe çıkacak insanlar trafiğin durumunu önceden bilmek isterler. Bu konuda Nesnelerin İnterneti büyük fayda sağlamaktadır. Gerçek zamanlı akan verinin Apache Flink ile analizi sonucu geleceğe dair durum tahmini amaçlanmıştır.

1.2. Proje Hedefi

Bu projenin hedefi sensörler aracılığıyla toplanan trafik verisi üzerinde makine öğrenmesi algoritmaları kullanılarak geleceğe dair yol durumu analizi amaçlanmaktadır. Analiz yapılırken mevcut zamanın 5 dakika sonrasındaki hız ve ortalama süre değerlerinin tahmini amaçlanmaktadır. Projenin hedefi doğrultusunda yapılan analizin bize trafik durumunu göstermesi beklenmektedir. Bu tahminler doğrultusunda insanlar trafik durumunu önceden görüp kendi rotalarını belirleyebilir.

1.3. Projede Mevcut Aşama

Akan veri başarılı bir şekilde pencereler halinde gruplanıp üzerinde uygulanmak istenen makine öğrenmesi algoritmaları denenmiştir. Ortaya çıkan sonuçlar doğrultusunda yolun yakın gelecekteki durumu modellenmiştir. Gerçek durumun ne kadar iyi tahmin edilebildiği ise aşağıda ayrıntılı bir şekilde anlatılacaktır.

1.3.1. Tasarım Kararları

1.3.1.1. Sınıfların Oluşturulması

Temel olarak kullanılan üç farklı sınıf bulunmaktadır. Bunlardan ilki verinin sistemle ilk bulunduğu anda oluşturulan Reading sınıfıdır. Gelen veri ayrıştırılıp haritalanarak Reading sınıfı şeklinde formatlanmaktadır. Diğer bir sınıf ise modele konulacak özellikleri barındıran Test sınıfıdır. Bu sınıf pencereleme işlemi tamamlandıktan sonra, oluşan penceredeki sınıfların üzerinde hızların, ortalama sürelerin ve araç sayılarının farkları gibi işlemleri de içererek modele girecek nihai Test sınıfı olarak ortaya çıkmaktadır. Son kullanılan sınıf Model sınıfı olup Test sınıfının tahmin işlemlerini gerçekleştirecek modeli içinde barındırmaktadır. Model sınıfına ilerde değinilecektir. Sınıflar ve içerikleri aşağıdaki şekillerde gösterilmiştir.

```
case class Reading(Status: String,
  avgTime: Int,
  avgSpeed : Int,
  extID : Int,
  medianTime : Int,
  times : String,
  vehicleCount : Int,
  normID : Int,
  reportID : Int
)
```

Şekil 1. Reading Sınıfı

```
case class Test(aAvgTime : Int,aAvgSpeed:Int,aVehicleCount : Int,aHour: Int,aMinute : Int,
  bAvgTime : Int,bAvgSpeed:Int,bVehicleCount : Int,bHour: Int,bMinute : Int,
  cAvgTime : Int,cAvgSpeed:Int,cVehicleCount : Int,cHour: Int,cMinute : Int,
  f21AvgTime : Int,f21AvgSpeed : Int,f21VehicleCount : Int,
  f32AvgTime : Int,f32AvgSpeed : Int,f32VehicleCount : Int |
)
```

Şekil 2. Test Sınıfı

1.3.1.2. Pencerelelerin oluşturulması

Sistemde zaman karakteristiği olay zamanlı olacak şekilde belirlenmiştir. Bundan dolayı gelen veride kendisinin oluşturulma süreside bulunmaktadır. Bu şekilde gelen veriden zaman çıkarılacak ve 5 dakika fazlalıkla imzası da (watermark) oluşturulup sisteme Flink yardımı ile verilecektir. Daha sonra zaman damgası (time stamp) ve imzası (watermark) belirlenen veriler üzerinden pencereleri oluşturma işlemine geçilmektedir. Verisetinde veri 5 dakikalık zaman aralığıyla gelmektedir. Bizde pencerenin boyutunu 15 ve pencere kaydırma zamanının 5 olarak belirleyerek verileri 5 dakika aralıklarla üçerli olarak grupladık. Gruplanan bu pencereler üzerinden daha sonra test edilecek sınıfın değerleri elde edilecektir.

```
Wrappers(Reading(OK,50,86,690,50,2014-09-30T12:25:00,10,28023979,158355), Reading(OK,50,86,690,50,2014-09-30T12:30:00,14,28024414,158355), Reading(OK,47,91,690,47,2014-09-30T12:35:00,9,28024863,158355))
Wrappers(Reading(OK,50,86,690,50,2014-09-30T12:30:00,14,28024414,158355), Reading(OK,47,91,690,47,2014-09-30T12:35:00,9,28024863,158355), Reading(OK,51,84,690,51,2014-09-30T12:40:00,13,28025298,158355))
Wrappers(Reading(OK,47,91,690,47,2014-09-30T12:35:00,9,28024863,158355), Reading(OK,51,84,690,51,2014-09-30T12:40:00,13,28025298,158355), Reading(OK,52,82,690,52,2014-09-30T12:45:00,15,28025742,158355))
Wrappers(Reading(OK,51,84,690,51,2014-09-30T12:40:00,13,28025298,158355), Reading(OK,52,82,690,52,2014-09-30T12:45:00,15,28025742,158355), Reading(OK,54,79,690,54,2014-09-30T12:50:00,9,28026117,158355))
```

Şekil 3. Zamana Bağlı Gruplama

Yukarıda 12:25:00'dan 12:25:50'ye kadar olan verilerin beşer dakika aralıklarla üçlü gruplanması gösterilmektedir.

1.3.1.3. Pencere Üzerindeki İşlemler

Burada yapılan işlemler sonucu Test sınıfından bir örnek oluşturulması amaçlanmaktadır. Eksik

veriler ya da pencerelerin düzgün oluşmaması gibi sorunlar Test Sonuçları 2.2.1. kısmında ayrıntılı olarak anlatılmaktadır. Pencere üzerindeki üçlü Reading grubu içerisinde istenilen özellikler gerekli işlemler sonucu çıkartılmaktadır.

1.3.1.4. Harita (Map) Fonksiyonu

Sistemin temelinde akan veri üzerinde makine öğrenmesi ve öğrenilen model üzerinde test işlemleri yatmaktadır. Apache Flink'in bize sağladığı makine öğrenme kütüphanesindeki fonksiyonların hepsi yığın (batch) mantığıyla çalışacak şekilde oluşturulmuştur yani öğrenme için gereken bir yığın (DataSet) ve test için ise oluşturulmuş başka bir yığın (DataSet) istemektedir. Ancak biz akan veri (Stream) üzerinde işlemlerimizi gerçekleştirmekteyiz. Dolayısıyla fonksiyonlar girdi olarak DataSet isterken bizim elimizde test için DataStream oluşmaktadır. Bu sorunu aşmak için akış (stream) üzerinde harita (map) fonksiyonu kullanılmaktadır.

Buradaki temel mantık akış şeklinde gelen ve Test sınıfına dönüştürülmüş veriyi tek satırlık küçük bir DataSet'e dönüştürmek ve tahmin işlemini bu DataSet üzerinden gerçekleştirmektir. Harita fonksiyonunun içinde oluşturulan mini DataSet'i tahmin (predict) işlemine sokmak amacıyla Model sınıfından bir örnek bulunmaktadır. Burada ortaya çıkan bir diğer husus ise akış üzerinde oluşan her pencereyi tahmin etmek için modelin harita fonksiyonunun içinde oluşturulmasıdır ancak aynı modeli defalarca örneklemek gereksizdir ve yük oluşturmaktadır. Bunu aşmak amacıyla harita fonksiyonuna CheckpointedFunction özelliği eklenmiştir. Bu sayede model içeren bir ListState tanımlanmış ve örneklenen modelin hafızada tutularak defalarca üretilmemesi sağlanmıştır.

1.3.1.5. Modelin Kurulması

Eğitim için elimizdeki veri Test sınıfı üzerinde tahmin işlemi uygulanabilecek şekilde sokulmuş ve tahmin edilmek istenen gelecek zamana dair bilgi eklenerek Dataset[LabeledVector] elde edilmiştir. Model içerisinde Multiple Linear Regression öğrenme metodu ve veri ön işleme teknikleri kullanmak amacıyla pipeline sistemi kurulmuştur. Daha sonra bu pipeline üzerinden eğitim işlemi gerçekleştirilmiştir. Ayrıca yukarıda bahsedilen Harita Fonksiyonu içerisinde kullanılmak üzere Predict isimli bir fonksiyonu bulunmaktadır. Bu fonksiyon temel olarak tahmin işlemini gerçekleştirmektedir. Harita fonksiyonunda da bahsedildiği gibi DataSet üzerinde ön işlemler ve eğitim gibi durumlar sadece bir kere çağırılarak gereksiz işlem

kaçınılacaktır. Sadece her pencere için Predict fonksiyonu defalarca işleme alınacaktır.

1.3.1.6. Veritabanı Tasarımı

Sistemde ayrıntılı bir veri tabanına ihtiyaç duyulmamaktadır. Ortaya çıkan analiz sonuçları csv formatında kaydedilecektir.

1.3.1.7. İstisnai Durumların İdaresi

Eğer gelen veri üzerinde boş veya tutarsız kolon varsa bu durumlar yakalanıp kontrol edilecektir. Kolondaki veri boş veya gürültülü ise o ve ona bağlı olan veriler mantık çerçevesinde düzeltilmeye çalışılacaktır. Bu işlemlere örnek olarak ortalama alma veya bir önceki veriyi kullanma gibi işlemler gösterilebilir. Ancak elimizde bulunan veri setinde eksik ya da bozuk olarak nitelendirilebilecek durumlarda bir önceki verinin değerleri kullanılmaktadır. Bu sebeple bizde aynı yaklaşımı kullanarak penceredeki eksik değerleri ona en yakın bir önceki verinin değerlerini atayarak sistemde o anda bir değişiklik yokmuş gibi davranmasını istedik.

2. TEST

2.1. Test Edilen Özellikler

Test edilen özellikler;

- Zaman penceresi (Time Window)
- Pencere üzerindeki işlemler
- Eğitim ve test sınıfları
- Map fonksiyonu
- Modelin doğruluğu

Zaman penceresi: Sisteme akan verinin olay zamanında (event time) gerçekleşmesine bağlı olarak timestamp ve watermarkların atanması ile pencereler belli zaman aralıkları ile oluşturulmaktadır. Kullanılacak test sınıfının özelliklerine göre pencere boyutu değişiklik gösterir.

Pencere üzerindeki işlemler: Oluşturulan pencere üzerinde test sınıfını oluşturmak amaçlı penceredeki eksik verilerin doldurulması ve gerekli özelliklerin çıkarılması bu bölümde gerçekleşmektedir.

Eğitim ve test sınıfları: Farklı biçimlerdeki eğitim ve test sınıflarının oluşturulması ve oluşturulan sınıf çeşitlerinin tahmin üzerindeki etkisi.

Map fonksiyonu: İşlemler tamamlandıktan sonra ortaya çıkan test sınıfının üzerinde tahmin (predict) işlemi uygulanır. Bunu yaparken kullanılan model bir kere oluşturulmalı ve oluşturulan model hafızada kalıp test işlemi için kullanılmalıdır.

Modelin Doğruluğu: Sonucu eniyileme amacıyla modelin parametreleri değiştirilerek en iyi model bulunmaya çalışılacaktır.

2.2. Test Sonuçları

2.2.1. Zaman Penceresi

Ham veriyi zaman penceresi (Time Window) olarak üçlü gruplandığımızda yani beşer dakikalık aralıklarla 15 dakikalık pencere oluşturma işleminde aşağıdaki durumlar oluşmuştur.

Durum 1: Gelen verileri üçerli gruplamaya çalıştığımızda resimdeki gibi hatalı olarak sonuçlar gelmektedir. Resmin sonunda görünen 1 rakamı penceredeki eleman sayısını göstermektedir.

```
TimeWindow{start=1402252200000, end=1402253100000}(0,1)
TimeWindow{start=1402252500000, end=1402253400000}(0,1)
TimeWindow{start=1402252800000, end=1402253700000}(0,1)
TimeWindow{start=1402253100000, end=1402254000000}(0,1)
```

Şekil 4. Zaman Penceresi Durum1

Bu durum watermark (ıslak imza) kullanılarak aşılmaya çalışılmıştır.

Durum 2: Watermark kullanıldığı halde eksik veri nedeniyle penceredeki eleman eksikliği gösterilmiştir.

```
TimeWindow{start=1402283400000, end=1402284300000}(0,3)
TimeWindow{start=1402283700000, end=1402284600000}(0,2)
TimeWindow{start=1402283100000, end=1402284000000}(0,3)
TimeWindow{start=1402284000000, end=1402284900000}(0,1)
```

Şekil 5. Zaman Penceresi Durum2

Durum 3: Watermark eklenen durumda penceredeki eleman sayısı 3 olarak sorun giderilmiştir.

```
TimeWindow{start=1402252200000, end=1402253100000}(0,3)
TimeWindow{start=1402252500000, end=1402253400000}(0,3)
TimeWindow{start=1402252800000, end=1402253700000}(0,3)
TimeWindow{start=1402253100000, end=1402254000000}(0,3)
```

Şekil 6. Zaman Penceresi Durum3

2.2.2. Pencere Üzerindeki İşlemler

Zaman penceresi yapılırken ayrıca eksik veri doldurulması ve özellik çıkarılması yapılmaktadır. Bu sayede eğitim verisi kodun pencere oluşturmaya evresinde oluşturulmaktadır. Test için gelen ham veri (gerçek zamanlı akan veri) yine bu pencereleme işleminden geçmekte ve test edilmeye hazır hale gelmektedir.

İlk olarak hız tahmini adına oluşturulan üçlü pencere aşağıdaki şekilde görüldüğü gibidir.

```
55,78,8,51,84,5,47,91,3,-4,6,-3,-4,7,-2,91.0
51,84,5,47,91,3,57,75,2,-4,7,-2,10,-16,-1,75.0
47,91,3,57,75,2,50,86,9,10,-16,-1,-7,11,7,86.0
57,75,2,50,86,9,54,79,13,-7,11,7,4,-7,4,79.0
```

Şekil 7. Üçlü pencerede hız

Sonrasında zaman tahmini adına yaptığımız üçlü pencere bulunmaktadır.

```
55,78,8,51,84,5,47,91,3,-4,6,-3,-4,7,-2,57.0
51,84,5,47,91,3,57,75,2,-4,7,-2,10,-16,-1,50.0
47,91,3,57,75,2,50,86,9,10,-16,-1,-7,11,7,54.0
57,75,2,50,86,9,54,79,13,-7,11,7,4,-7,4,54.0
```

Şekil 8. Üçlü pencerede zaman

Saat ve dakika bilgileri özellik (feature) olarak eklenmiş şekilde zaman tahmini için eğitim verisi aşağıda bulunmaktadır. Dördüncü, dokuzuncu, on dördüncü kolon saat, beşinci, onuncu, on beşinci kolonlar dakika olarak alınmıştır.

```
51,84,6,11,50,49,87,9,11,55,45,95,8,12,0,-2,3,3,-4,8,-1,45.0
49,87,9,11,55,45,95,8,12,0,45,95,8,12,0,-4,8,-1,0,0,0,45.0
45,95,8,12,0,45,95,11,12,10,45,95,11,12,10,0,0,3,0,0,0,51.0
```

Şekil 9. Saat ve Dakika Kullanılarak Zaman

Saat ve dakika bilgileri özellik (feature) olarak eklenmiş şekilde hız tahmini için eğitim verisi aşağıda bulunmaktadır. Dördüncü, dokuzuncu, on dördüncü kolon saat, beşinci, onuncu, on beşinci kolonlar dakika olarak alınmıştır.

```
55,78,8,7,55,51,84,5,8,0,47,91,3,8,5,-4,6,-3,-4,7,-2,75.0
51,84,5,8,0,47,91,3,8,5,57,75,2,8,10,-4,7,-2,10,-16,-1,86.0
47,91,3,8,5,57,75,2,8,10,50,86,9,8,15,10,-16,-1,-7,11,7,79.0
57,75,2,8,10,50,86,9,8,15,54,79,13,8,20,-7,11,7,4,-7,4,79.0
```

Şekil 10. Saat ve Dakika Kullanılarak Hız

Pencereler başarılı bir şekilde oluşturulmuştur.

2.2.3. Map Fonksiyonu

2.2.3.1. Tahmin İşlemi

Sonuç değerinin (label) doğruluğundan bağımsız olarak oluşturulan pencereler üzerinde tahmin (predict) işlemi gerçekleştirilebilmiştir.

Üçlü pencere kullanılarak hız ve zaman tahminleri:

```
(DenseVector(49.0, 87.0, 0.0, 44.0, 97.0, 0.0, 44.0, 97.0, 0.0, -5.0, 10.0, 0.0, 0.0, 0.0, 0.0),89.0328976830197)
(DenseVector(44.0, 97.0, 0.0, 44.0, 97.0, 0.0, 54.0, 79.0, 2.0, 0.0, 0.0, 0.0, 10.0, -18.0, 2.0),87.03944559434449)
(DenseVector(44.0, 97.0, 0.0, 54.0, 79.0, 2.0, 53.0, 81.0, 3.0, 10.0, -18.0, 2.0, -1.0, 2.0, 1.0),84.53370100889813)
(DenseVector(54.0, 79.0, 2.0, 53.0, 81.0, 3.0, 49.0, 87.0, 1.0, -1.0, 2.0, 1.0, -4.0, 6.0, -2.0),83.25217472699987)
```

Şekil 11. Üçlü pencerede hız tahmini

```
(DenseVector(43.0, 100.0, 0.0, 59.0, 72.0, 1.0, 59.0, 72.0, 1.0, 16.0, -28.0, 1.0, 0.0, 0.0, 0.0),48.55563288758699)
(DenseVector(59.0, 72.0, 1.0, 59.0, 72.0, 1.0, 59.0, 72.0, 0.0, 0.0, 0.0, 0.0, 0.0, -1.0),46.11806290609811)
(DenseVector(59.0, 72.0, 1.0, 59.0, 72.0, 0.0, 58.0, 74.0, 1.0, 0.0, 0.0, -1.0, -1.0, 2.0, 1.0),46.286362926580004)
(DenseVector(59.0, 72.0, 0.0, 58.0, 74.0, 1.0, 48.0, 89.0, 2.0, -1.0, 2.0, 1.0, -10.0, 15.0, 1.0),47.49939052473537)
```

Şekil 12. Üçlü pencerede zaman tahmini

```
(DenseVector(45.0, 95.0, 6.0, 14.0, 25.0, 51.0, 84.0, 8.0, 14.0, 30.0, 48.0, 89.0, 10.0, 14.0, 35.0, 6.0, -11.0, 2.0, -3.0, 5.0, 2.0),50.9058488211416,50.0)
(DenseVector(51.0, 84.0, 8.0, 14.0, 30.0, 48.0, 89.0, 10.0, 14.0, 35.0, 50.0, 86.0, 6.0, 14.0, 40.0, -3.0, 5.0, 2.0, -3.0, -4.0),51.05453943888035,51.0)
(DenseVector(48.0, 89.0, 10.0, 14.0, 35.0, 50.0, 86.0, 6.0, 14.0, 40.0, 51.0, 84.0, 4.0, 14.0, 45.0, 2.0, -3.0, -4.0, 1.0, -2.0, -2.0),51.68354376490198,48.0)
(DenseVector(50.0, 86.0, 6.0, 14.0, 40.0, 51.0, 84.0, 4.0, 14.0, 45.0, 48.0, 89.0, 6.0, 14.0, 50.0, 1.0, -2.0, -2.0, -3.0, 5.0, 2.0),51.85917396185555,52.0)
(DenseVector(51.0, 84.0, 4.0, 14.0, 45.0, 48.0, 89.0, 6.0, 14.0, 50.0, 52.0, 82.0, 8.0, 14.0, 55.0, -3.0, 5.0, 2.0, 4.0, -7.0, 2.0),52.54501330676039,54.0)
```

Şekil 13. Saat ve Dakika kullanılarak zaman tahmini

```
(DenseVector(59.0, 72.0, 1.0, 20.0, 30.0, 59.0, 72.0, 0.0, 20.0, 35.0, 58.0, 74.0, 1.0, 20.0, 40.0, 0.0, 0.0, -1.0, -1.0, 2.0, 1.0),73.32753373956719,89.0)
(DenseVector(59.0, 72.0, 0.0, 20.0, 35.0, 58.0, 74.0, 1.0, 20.0, 40.0, 48.0, 89.0, 2.0, 20.0, 45.0, -1.0, 2.0, 1.0, -10.0, 15.0, 1.0),82.46183921504286,102.0)
(DenseVector(58.0, 74.0, 1.0, 20.0, 40.0, 48.0, 89.0, 2.0, 20.0, 45.0, 42.0, 102.0, 1.0, 20.0, 50.0, -10.0, 15.0, 1.0, -6.0, 13.0, -1.0),88.29889994129044,102.0)
(DenseVector(48.0, 89.0, 2.0, 20.0, 45.0, 42.0, 102.0, 1.0, 20.0, 50.0, 42.0, 102.0, 0.0, 20.0, 55.0, -6.0, 13.0, -1.0, 0.0, 0.0, -1.0),88.81993538761769,75.0)
```

Şekil 14. Saat ve Dakika kullanılarak hız tahmini

2.2.3.2. Model Kullanımı

Modelin her pencere için baştan oluşturulması işlem bakımından yorucu ve gereksizdir. Tek bir modelin hafızada tutulup her pencere için aynı model üzerinden test işlemi başarı ile

gerçekleştirilmiştir.

2.2.4. Modelin Doğruluğu

Üçlü pencerede süre tahmini için bulunan parametreler:

setIteration = 10

setStepSize = 0.0003

ConvergenceThreshold = default

Gelen tahmin sonucunun kısa bir kesiti aşağıdaki gibidir. Vector ve tahmin değerleri bulunmaktadır.

```
(DenseVector(43.0, 100.0, 0.0, 59.0, 72.0, 1.0, 59.0, 72.0, 1.0, 16.0, -28.0, 1.0, 0.0, 0.0, 0.0),46.9931828253904)
(DenseVector(59.0, 72.0, 1.0, 59.0, 72.0, 1.0, 59.0, 72.0, 0.0, 0.0, 0.0, 0.0, 0.0, -1.0),44.61704803066145)
(DenseVector(59.0, 72.0, 1.0, 59.0, 72.0, 0.0, 58.0, 74.0, 1.0, 0.0, 0.0, -1.0, -1.0, 2.0, 1.0),44.783150572133685)
(DenseVector(59.0, 72.0, 0.0, 58.0, 74.0, 1.0, 48.0, 89.0, 2.0, -1.0, 2.0, 1.0, -10.0, 15.0, 1.0),45.985681035294824)
(DenseVector(58.0, 74.0, 1.0, 48.0, 89.0, 2.0, 42.0, 102.0, 1.0, -10.0, 15.0, 1.0, -6.0, 13.0, -1.0),48.4282751153322)
```

Şekil 15. Parametresi bulunan üçlü süre tahmini

Üçlü pencerede hız tahmini için bulunan parametreler:

setIteration = 10

setStepSize = 0.000015

ConvergenceThreshold = default

Gelen tahmin sonucunun kısa bir kesiti aşağıdaki gibidir. Vector ve tahmin değerleri bulunmaktadır.

```
(DenseVector(49.0, 87.0, 0.0, 44.0, 97.0, 0.0, 44.0, 97.0, 0.0, -5.0, 10.0, 0.0, 0.0, 0.0, 0.0),89.0328976830197)
(DenseVector(44.0, 97.0, 0.0, 44.0, 97.0, 0.0, 54.0, 79.0, 2.0, 0.0, 0.0, 0.0, 10.0, -18.0, 2.0),87.03944559434449)
(DenseVector(44.0, 97.0, 0.0, 54.0, 79.0, 2.0, 53.0, 81.0, 3.0, 10.0, -18.0, 2.0, -1.0, 2.0, 1.0),84.53370100889813)
(DenseVector(54.0, 79.0, 2.0, 53.0, 81.0, 3.0, 49.0, 87.0, 1.0, -1.0, 2.0, 1.0, -4.0, 6.0, -2.0),83.25217472699987)
(DenseVector(53.0, 81.0, 3.0, 49.0, 87.0, 1.0, 45.0, 95.0, 1.0, -4.0, 6.0, -2.0, -4.0, 8.0, 0.0),86.12836324145339)
```

Şekil 16. Parametresi bulunan üçlü hız tahmini

Üçlü pencerede hız tahmini için bulunan parametreler:

setIteration = 30

setStepSize = 0.45

ConvergenceThreshold = default

Gelen tahmin sonucunun kısa bir kesiti aşağıdaki gibidir. Vector ve tahmin değerleri bulunmaktadır.

```
(DenseVector(45.0, 95.0, 6.0, 14.0, 25.0, 51.0, 84.0, 8.0, 14.0, 30.0, 48.0, 89.0, 10.0, 14.0, 35.0, 6.0, -11.0, 2.0, -3.0, 5.0, 2.0),50.9058488211416,50.0)
(DenseVector(51.0, 84.0, 8.0, 14.0, 30.0, 48.0, 89.0, 10.0, 14.0, 35.0, 50.0, 86.0, 6.0, 14.0, 40.0, -3.0, 5.0, 2.0, 2.0, -3.0, -4.0),51.05453943888035,51.0)
(DenseVector(48.0, 89.0, 10.0, 14.0, 35.0, 50.0, 86.0, 6.0, 14.0, 40.0, 51.0, 84.0, 4.0, 14.0, 45.0, 2.0, -3.0, -4.0, 1.0, -2.0, -2.0),51.68354376490198,48.0)
(DenseVector(50.0, 86.0, 6.0, 14.0, 40.0, 51.0, 84.0, 4.0, 14.0, 45.0, 48.0, 89.0, 6.0, 14.0, 50.0, 1.0, -2.0, -2.0, -3.0, 5.0, 2.0),51.85917396185555,52.0)
(DenseVector(51.0, 84.0, 4.0, 14.0, 45.0, 48.0, 89.0, 6.0, 14.0, 50.0, 52.0, 82.0, 8.0, 14.0, 55.0, -3.0, 5.0, 2.0, 4.0, -7.0, 2.0),52.54501330676039,54.0)
```

Şekil 17. Saat ve Dakika Kullanılarak üçlü süre tahmini

Üçlü pencerede hız tahmini için bulunan parametreler:

setIteration = 10

setStepSize = 0.45

ConvergenceThreshold = default

Gelen tahmin sonucunun kısa bir kesiti aşağıdaki gibidir. Vector ve tahmin değerleri bulunmaktadır.

```
(DenseVector(59.0, 72.0, 1.0, 20.0, 30.0, 59.0, 72.0, 0.0, 20.0, 35.0, 58.0, 74.0, 1.0, 20.0, 40.0, 0.0, 0.0, -1.0, -1.0, 2.0, 1.0),73.32753373956719,89.0)
(DenseVector(59.0, 72.0, 0.0, 20.0, 35.0, 58.0, 74.0, 1.0, 20.0, 40.0, 48.0, 89.0, 2.0, 20.0, 45.0, -1.0, 2.0, 1.0, -10.0, 15.0, 1.0),82.46183921504286,102.0)
(DenseVector(58.0, 74.0, 1.0, 20.0, 40.0, 48.0, 89.0, 2.0, 20.0, 45.0, 42.0, 102.0, 1.0, 20.0, 50.0, -10.0, 15.0, 1.0, -6.0, 13.0, -1.0),88.29889994129044,102.0)
(DenseVector(48.0, 89.0, 2.0, 20.0, 45.0, 42.0, 102.0, 1.0, 20.0, 50.0, 42.0, 102.0, 0.0, 20.0, 55.0, -6.0, 13.0, -1.0, 0.0, 0.0, -1.0),88.81993538761769,75.0)
```

Şekil 18. Saat ve Dakika Kullanılarak üçlü hız tahmini

Test sonuçlarındaki tahminler ile asıl verimizdeki değerleri Standart Sapma kullanarak karşılaştırdığımızda saat ve dakikayı eklediğimiz üçlü zaman modeli yaklaşık 8 değerlik sapma göstermektedir.

```
1387.620420619033
1447.9517432546431
1508.2830658902533
StandartSapma:7.927491895008611
```

Şekil 19. Üçlü Zaman için Standart Sapma

Polynomial Feature 3. derecen kullanarak yaptığımız yine üçlü zaman tahmini modeli daha iyi sonuç vermektedir. Ulaşılmak istenen labela olan uzaklık ortalama 4-5 civarlarında olmaktadır. Şekil 20. de gösterilen resimde virgülden önceki değerler tahmin değerleri olup, yanındaki değerler karşılaştırmak amacıyla kullandığımız verilerin değerleridir.

(56.590810744103855,68.0)
(58.73122158548911,62.0)
(53.969717702679134,49.0)
(49.1735740386884,50.0)
(46.60984269263683,50.0)
(48.194376552873955,47.0)
(48.36674064530042,45.0)
(48.54119483276059,51.0)
(51.22240118364909,48.0)
(50.93572924955338,50.0)
(51.11052082907494,51.0)
(52.308323203494126,48.0)
(51.70017138054535,52.0)
(53.578190739192024,54.0)
(51.72819952834342,45.0)
(47.8414897897214,52.0)
(47.48238042354502,56.0)
(49.8317281518462,56.0)
(48.26641484649596,54.0)
(47.86702861526986,45.0)
(47.3032795404491,44.0)
(48.94061532354728,45.0)
(51.35753283283662,45.0)
(52.203874352114944,46.0)
(52.950968078303596,46.0)
(53.61000909989406,45.0)
(50.428296901037065,53.0)
(49.257131344300056,53.0)
(48.51673119441427,55.0)
(47.809147858517974,37.0)
(45.70000576292188,42.0)
(49.450565500035616,48.0)
(54.90314703704801,42.0)
(51.34059724768856,42.0)
(49.717699694873104,47.0)
(53.73326907910744,46.0)
(53.24038686826487,46.0)
(52.55717892663867,50.0)

Şekil 20. Polynomial Feature kullanarak zaman tahmini

Dizi Ortalaması: 50.712381404665
En Küçük Sayı: 45.70000576292188
En Büyük Sayı: 58.73122158548911
Standart Sapma: 2.8608657929124
Varyans (Standart Sapma): 8.184553085056
Populasyonun Standart Sapması: 2.8229718576588
Varyans (Populasyonun Standart Sapması): 7.9691701091335

Şekil 21. Polynomial Feature ile Sapma değerleri

3. DURUM DEĞERLENDİRMESİ

3.1. Test Durumları

Dokümanın Test kısmında bahsedildiği üzeri Reading, Test ve Model sınıfları başarılı bir şekilde çalışmaktadır. Pencereleme işlemindeki sorunlar giderilmiş ve beşer dakikalık aralıklarla on beşer dakikalık üçlü pencere oluşturma işlemi zamanlama (time stamp) ve imzalar (watermark) yardımıyla gerçekleştirilmiştir. Bunların yanı sıra modelin eğitim ve test işlemleri için parametre bulma ve kontrol etme işlemleri gerçekleştirilmiştir.

Genel olarak bir yolun beş dakika sonraki süre tahmini, o yolun hız tahmininden daha verimli görünmektedir. Hızda oluşabilen ani sıçramaları yakalamak biraz daha zor olduğundan üçlü gruplandırılmış zaman tahmini kullandığımız özelliklere (feature) daha iyi tepki vermektedir.

3.2. Projenin Gerçekleştirilmesi

Proje gerçekleştirilirken Apache Flink dokümanı baz alınmıştır [1]. Özellikleri veri setinden alıp ekstra olarak kendimiz özellik ekledik. Mantık çerçevesinde hız, süre, araç sayısı farklarını alarak ve verinin kendi değerleriyle oluşturduğumuz sistem, beş dakika sonrasının tahminini yapmaktadır.

3.3. Projenin Sonuçlandırılması

Projede başarılı bir şekilde model kuruldu, pencereleme ve haritalama işlemleri gerçekleştirildi. Model, bir makine öğrenme algoritması olan Multiple Linear Regression kullanılarak oluşturuldu ve akan veri üzerinde doğru sonuçları vermesi adına parametreler düzeltildi. Modele ek özellikler çıkartıldı. Bu projede Apache Flink'i ve yapabildiklerini öğrendik, uygulamaya döktük.

4. PROJEYE YÖNELİK GELECEK ÇALIŞMALARI

Projenin gelecekte geliştirilme imkânı bulunmaktadır. Öncelikle daha iyi özelliklere ihtiyaç bulunmaktadır. Elimizde bulunan veriler ile çıkarılmış şu anki özellikler yeterli düzeydedir ama daha iyi sonuçlar için iyileştirilmeleri gerekmektedir. Projenin zaman kısıdı ve sahip olduğumuz donanımlar belirli bir yere kadar bizi ilerletmektedir. PolynomialFeature kullanarak yaptığımız deneyler yüksek CPU gücü ve zaman gerektirmektedir. Parametreleri doğru bulmak bu yüzden zor olmaktadır. Her ne kadar bazı sorunlarla karşılaşmış olsak da bu proje ile geleceğe ufak bir adım atmış olmak, gelecek çalışmalarımız için önayak olacaktır diyebiliriz.

5. REFERANSLAR

1. <https://ci.apache.org/projects/flink/flink-docs-release-1.4/>