Taksi Varış Zamanı / Taksi Ücreti Tahmini

Arman Kuyucu[1], Kübranur Aysöndü[2], Adil Abdullayev[3]

[1,2] *Bilgisayar Mühendisliği, Kocaeli Üniversitesi*
Kabaoğlu Mahallesi, Baki Komsuoğlu Bulvarı No:515, Umuttepe, 41001 İzmit/Kocaeli
[1]190201099@kocaeli.edu.tr
[2]210201104@kocaeli.edu.tr
[3]160201097@kocaeli.edu.tr

***Abstract*— Büyük veri analizi, büyük veri setlerinin işlenmesi ve analiz edilmesidir. Büyük veri analizi, genellikle büyük veri setlerinden anlamlı bilgi çıkarmak için kullanılır. Büyük veri analizi teknikleri, makine öğrenmesi, veri madenciliği, doğal dil işleme ve benzeri yöntemleri içerir. New York Şehri Taksi ve Limuzin Komisyonunun yayınladığı etiketli bir veri seti kullanılmıştır. Çalışmada Rastgele Orman, Lineer Regresyon makine öğrenme algoritmaları kullanılarak taksi varış zamanı ve taksi ücreti tahmini yapılmıştır. Deneysel Sonuçlara bakılacak olursa Lineer Regresyon algoritması en iyi sonucu veren algoritma olmuştur.**

***Keywords*— Büyük Veri Analizi, Apache Spark, Yapay Zeka, Makine Öğrenmesi.**

I. Giriş

Günümüzde teknolojinin hızlı gelişimi ve mobil uygulamaların yaygınlaşması ile birlikte, taksi hizmetleri de popülerliğini korumaktadır. Ancak, taksi yolculukları sırasında belirsizlikler sıklıkla yaşanır. Özellikle, tahmini varış zamanı ve taksi ücreti hakkında bilgi sahibi olmak yolcular için oldukça önemlidir. Tahmini varış zamanı bilgisi yolcuların zamanlarını daha etkili bir şekilde planlamalarına yardımcı olurken, taksi ücreti tahmini ise yolcuların yolculuk maliyetleri hakkında önceden bilgi sahibi olmalarına olanak tanır. Bu bağlamda, bu rapor deep learning tekniklerinin kullanılarak taksi varış zamanı ve taksi ücreti tahmini yapabilen bir uygulama geliştirilmesini amaçlamaktadır. Taksi varış zamanı ve taksi ücreti tahmini, her iki durumda da doğru tahminler yapmak oldukça önemlidir. Doğru tahminler, yolculuk planlaması ve maliyet tahmini gibi konularda kullanıcıya fayda sağlayabilir. Bu nedenle, bu sorunlara deep learning yöntemleri uygulayarak çözümler üretebiliriz. Bu raporda, taksi varış zamanı ve taksi ücreti tahmini için bir deep learning uygulamasının nasıl tasarlandığı ve performansının nasıl değerlendirildiği anlatılacaktır.

Projede sürekli değerler olan yolculuk süresi ve taksi varış zamanı tahmin edilmiştir yani bir regresyon tahmini yapılmıştır. Kullanılan veri setinde[2] New York Şehrindeki taksi yolculuklarına ait bilgiler yer almaktadır. Veri seti test ve eğitim verileri ayrı olarak ikiye ayrılmıştır. Bu projede 3 milyondan fazla satır ve 17 sütun bulunmaktadır. Veri setinin içeriği şekil 1'de verilmiştir. Bu veri setinde yer alan bilgiler arasında seyahat başlangıç ve bitiş zamanları,ödeme tipi, taksi ücretleri ve diğer öznitelikler bulunmaktadır. Veri setindeki sütunların tiplerini gösteren şeması ise Şekil 2'de verilmiştir.

	0	1	2	3	4
summary	count	mean	stddev	min	max
passenger_count	2995023	1.3625321074328978	0.8961199745510026	0.0	9.0
trip_distance	2995023	3.436198813831972	42.09135148963798	0.0	62359.52
RatecodeID	2995023	1.4974395856058536	6.4747666839879425	1.0	99.0
PULocationID	2995023	166.4399308452723	64.06784692466815	1	265
DOLocationID	2995023	164.46508290587417	69.92720379904178	1	265
payment_type	2995023	1.2230961164572025	0.5020600425963035	1	4
fare_amount	2995023	18.308272437307277	17.893249013700697	-900.0	1160.1
extra	2995023	1.569119779714551	1.795131358511999	-7.5	12.5
mta_tax	2995023	0.4880263356909147	0.10463915729013075	-0.5	53.16
tip_amount	2995023	3.3591933985158047	3.84055152209937	-96.22	380.8
tolls_amount	2995023	0.5202259615387964	2.021864729714521	-65.0	196.99
improvement_surcharge	2995023	0.9817241470265604	0.18538655715750696	-1.0	1.0
total_amount	2995023	26.969763197075103	22.26908296748362	-751.0	1169.4
congestion_surcharge	2995023	2.274231283031883	0.771845410867694	-2.5	2.5
airport_fee	2995023	0.10740860754658646	0.3556511325197914	-1.25	1.25

Şekil 1. Ön İşlemden Önceki Veri Setinin İçeriği

```
df.printSchema()
```

```
root
 |-- VendorID: long (nullable = true)
 |-- tpep_pickup_datetime: timestamp (nullable = true)
 |-- tpep_dropoff_datetime: timestamp (nullable = true)
 |-- passenger_count: double (nullable = true)
 |-- trip_distance: double (nullable = true)
 |-- RatecodeID: double (nullable = true)
 |-- store_and_fwd_flag: string (nullable = true)
 |-- PULocationID: long (nullable = true)
 |-- DOLocationID: long (nullable = true)
 |-- payment_type: long (nullable = true)
 |-- fare_amount: double (nullable = true)
 |-- extra: double (nullable = true)
 |-- mta_tax: double (nullable = true)
 |-- tip_amount: double (nullable = true)
 |-- tolls_amount: double (nullable = true)
 |-- improvement_surcharge: double (nullable = true)
 |-- total_amount: double (nullable = true)
 |-- congestion_surcharge: double (nullable = true)
 |-- airport_fee: double (nullable = true)
```

Şekil 2. Veri Setindeki Sütunların Tiplerini Gösteren Şeması

Projede Rastgele Ağaç ve Lineer Regresyon olmak üzere 2 adet makine öğrenmesi modeli kullanılmıştır.

II. YÖNTEM

A. Kullanılan Makine Öğrenme Algoritmaları

1) *Rastgele Ağaç (Random Forest)(RF) :* Random Forest algoritması, denetimli sınıflandırma algoritmalarından biridir. Hem regresyon hem de sınıflandırma problemlerinde kullanılmaktadır. Algoritma, birden fazla karar ağacı üreterek sınıflandırma işlemi esnasında sınıflandırma değerini yükseltmeyi hedefler. Random forest algoritması birbirinden bağımsız olarak çalışan birçok karar ağacının bir araya gelerek aralarından en yüksek puan alan değerin seçilmesi işlemidir. Ağaç sayısı arttıkça kesin bir sonuç elde etme oranı artmaktadır. Karar ağaçları algoritması ile arasındaki temel fark, Random Forest algoritmasında kök düğümü bulma ve düğümleri bölme işleminin rastgele olmasıdır. Random forest algoritması, elinde yeterli miktarda ağaç varsa aşırı öğrenme sorununu azaltır. Az oranda bir veri hazırlığına ihtiyaç duyar [3].

2) *Lineer Regresyon (Doğrusal Regresyon):* Lineer regresyon, makine öğrenmesi alanında kullanılan bir algoritmadır ve bir bağımlı değişkenin (çıktı) bağımsız değişkenler (girdiler) ile ilişkisini modellemek için kullanılır. Bu model, girdi değişkenleri ve çıktı değişkenleri arasındaki doğrusal ilişkiyi öğrenmeye çalışır. Doğrusal korelasyon ve basit doğrusal regresyon, iki değişken arasındaki doğrusal ilişkiyi inceleyen istatistiksel yöntemlerdir. Burada şu farklılığı vurgulamakta fayda var: Korelasyon, iki değişkenin ne kadar ilişkili olduğunu gösterirken, doğrusal regresyon, iki değişken arasındaki ilişkiye dayanarak birinin değerini diğerinden tahmin etmeyi sağlayan bir denklem (model) oluşturmayı içerir. Doğrusal regresyon, bir dizi noktaya en uygun düz çizgiyi veya hiper düzlemi bulmak için kullanılmaktadır. Bir diğer ifadeyle doğrusal regresyon, en uygun düz çizgi (regresyon çizgisi olarak da bilinir) kullanarak bağımlı değişken (Y) ile bir veya daha fazla bağımsız değişken (X) arasında bir ilişki kurar.

B. Veri Önişleme

Yolculuk süresinin hesaplanabilmesi için yolcu alış ve bırakış sürelerinin tipinin dönüştürülmesi işlemini gerçekleştirdik.

```
df = df.withColumn('tpep_pickup_datetime', col('tpep_pickup_datetime').cast('timestamp'))
```

```
df = df.withColumn('tpep_dropoff_datetime', col('tpep_dropoff_datetime').cast('timestamp'))
```

Daha sonra null değerleri tespit ettik ve her satırda kaç tane olduğunu yazdırdık.

```
for c in df.columns:
  print("`{:s}` satırındaki null değer sayısı = {:d}".format(c, df.where(col(c).isNull()).
```

```
`VendorID` satırındaki null değer sayısı = 0
`tpep_pickup_datetime` satırındaki null değer sayısı = 0
`tpep_dropoff_datetime` satırındaki null değer sayısı = 0
`passenger_count` satırındaki null değer sayısı = 71743
`trip_distance` satırındaki null değer sayısı = 0
`RatecodeID` satırındaki null değer sayısı = 71743
`store_and_fwd_flag` satırındaki null değer sayısı = 71743
`PULocationID` satırındaki null değer sayısı = 0
`DOLocationID` satırındaki null değer sayısı = 0
`payment_type` satırındaki null değer sayısı = 0
`fare_amount` satırındaki null değer sayısı = 0
`extra` satırındaki null değer sayısı = 0
`mta_tax` satırındaki null değer sayısı = 0
`tip_amount` satırındaki null değer sayısı = 0
`tolls_amount` satırındaki null değer sayısı = 0
`improvement_surcharge` satırındaki null değer sayısı = 0
`total_amount` satırındaki null değer sayısı = 0
`congestion_surcharge` satırındaki null değer sayısı = 71743
`airport_fee` satırındaki null değer sayısı = 71743
```

Bizim veri setimizde 3 milyon civarında satır olduğundan ve yalnızca 71743 satır (%2,33) null değer içerdiğinden null değer olan satırlar çıkarılmıştır.

df = df.dropna()

Toplam ücretin negatif olduğu sütunların görüntülenmesi ve çıkarılması işlemlerini gerçekleştirdik.

df = df.filter(df.total_amount >= 0)
df = df.filter(df.fare_amount >=0)

C. Öznitelik Mühendisliği

1) Gereksiz Özniteliklerin Çıkarılması

Store_and_fwd_flag sütunu verinin araba hafızasında depolanıp daha sonra gönderilip gönderilmediği bilgisini içeriyor. Bu sütun hem tek kategorik sütun olması hem de bizim amacımız için anlamsız olduğundan silinmiştir. VendorID sütunu ise veriyi kaydeden sağlayıcının id'sini gösterdiğinden silinmiştir.

df = df.drop('store_and_fwd_flag', 'VendorID')

2) Yeni Öznitelik Eklenmesi

Burada yolculuk süresinin hesaplanması ve sıfır veya negatif değerlerin silinmesi işlemlerini gerçekleştirdik. Yolculuk süresi hesaplandıktan sonra taksinin hızını hesaplayabileceğiz.

df = df.withColumn('trip_duration', col('tpep_dropoff_datetime').cast('long') - col('tpep_pickup_datetime').cast('long'))

Daha sonra ise taksinin hızı hesaplanarak yeni bir sütun olarak eklenmiştir.

df = df.withColumn('speed', round(col('trip_distance') / (col('trip_duration') / 3600), 2))

Veri ön işlemden sonraki istatistiksel özellikler Şekil 3'de verilmiştir.

	0	1	2	3	4
summary	count	mean	stddev	min	max
passenger_count	2968767	1.362914637625654	0.8974650834139217	0.0	9.0
trip_distance	2968767	3.443046092872817	42.27485540425921	0.0	62359.52
RatecodeID	2968767	1.4989546165125118	6.4931298283036405	1.0	99.0
PULocationID	2968767	166.4704454745017	64.06645752278516	1	265
DOLocationID	2968767	164.48519200058476	69.9019222163139	1	265
payment_type	2968767	1.2047937746545956	0.45757000527078556	1	4
fare_amount	2968767	18.624366462575704	17.47427806391104	0.0	1160.1
extra	2968767	1.5901115446244227	1.7832219135557128	0.0	12.5
mta_tax	2968767	0.4963185052919311	0.05340525081901216	0.0	53.16
tip_amount	2968767	3.387903628005493	3.8382684202657207	0.0	380.8
tolls_amount	2968767	0.5283384987794155	2.019691296183337	0.0	196.99
improvement_surcharge	2968767	0.9985671829415812	0.03281785390279815	0.0	1.0
total_amount	2968767	27.402347984799786	21.72864152195286	0.0	1169.4
congestion_surcharge	2968767	2.3103657174847334	0.6619098826585396	0.0	2.5
airport_fee	2968767	0.10982463426735746	0.35386351142941863	0.0	1.25
trip_duration	2968767	941.2717124651413	2591.527288185206	1	601751
speed	2968767	13.379227797262553	341.5969481806966	0.0	544752.0

Şekil 3. Veri Ön İşlemden Sonraki İstatistiksel Özellikler

III. DENEYSEL SONUÇLAR

Bu çalışma kapsamında taksi varış zamanı ve ücretini Lineer Regresyon ve Rastgele Orman algoritmalarını kullanarak tahmin ettik.

Alınan sonuçlar aşağıdaki tablolarda gösterilmiştir.

Yolculuk Süresi Tahmin Sonuçları

MODEL	TRAIN MSE	TRAIN RMSE	TRAIN R2
Lineer Regression	0.0099	0.0999	0.9999

MODEL	TEST MSE	TEST RMSE	TEST R2
Lineer Regression	0.0096	0.0982	0.9999
Random Forest	1056778.1472	1027.9971	0.8381

Toplam Ücret Tahmin Sonuçları

MODEL	TRAIN MSE	TRAIN RMSE	TRAIN R2
Lineer Regression	0.0939	0.3064	0.9998

MODEL	TEST MSE	TEST RMSE	TEST R2
Lineer Regression	0.0932	0.3054	0.9998
Random Forest	14.3844	3.7926	0.9696

features	prediction	trip_duration
(17,[0,2,3,4,5,6,...	4.036061380554031	4
[1.0,0.97,1.0,132...	132.03113931874861	132
[1.0,0.02,5.0,1.0...	4.036061380554031	4
[1.0,0.41,1.0,48....	631.011950968429	631
[1.0,2.9,1.0,263....	727.008259422075	727
[1.0,3.76,1.0,249...	1155.9917628243054	1156
[1.0,8.02,1.0,138...	922.0007609685433	922
[1.0,2.38,1.0,142...	471.01810354568585	471
[1.0,17.51,1.0,13...	1476.979419216184	1477
[3.0,0.95,1.0,79....	225.02756313321814	225

features	prediction	total_amount
(17,[0,2,3,4,5,6,...	119.83726555386747	120.3
[1.0,0.97,1.0,132...	7.066820852021671	6.55
[1.0,0.02,5.0,1.0...	119.95381729067246	120.3
[1.0,0.41,1.0,48....	14.877611231144472	15.0
[1.0,2.9,1.0,263....	23.47495857879943	23.6
[1.0,3.76,1.0,249...	35.437695433152726	35.5
[1.0,8.02,1.0,138...	49.54630630114151	50.2
[1.0,2.38,1.0,142...	21.303532544541298	21.38
[1.0,17.51,1.0,13...	69.55379608644174	69.75
[3.0,0.95,1.0,79....	16.48150302124689	16.5

IV. Sonuçlar

Geliştirdiğimiz proje kullanıcının taksi yolculuğunun tahmini varış zamanı veya taksi ücretini doğru bir şekilde tahmin etmesine yardımcı olacaktır. Bu proje, kullanıcıların taksi yolculuğunu daha iyi planlamalarına ve daha iyi bir bütçe yapmalarına yardımcı olacaktır. Projemiz gerçek zamanlı taksi verileri kullanılarak test edilmiştir. Sonuç olarak Spark ve Yapay zeka teknolojilerinin birleşimi, New York'taki taksi seyahatleri için tahmini varış zamanı ve ücreti sağlamak gibi karmaşık görevleri bile gerçekleştirmek için kullanılabilir. Bu proje taksi sektörüne de ayrıca birçok fayda sağlamaktadır. Bu sonuçlar, modelimizin gerçek hayatta kullanılabilecek kadar iyi performans gösterdiğini gösteriyor. Bu çalışmada, taksi varış zamanı ve taksi ücreti tahmini yapmak için bir deep learning modeli geliştirdik. Modelimizin test seti üzerindeki performansı oldukça başarılı oldu ve gerçek hayatta kullanılabilecek kadar iyi performans gösterdi. Ancak, modelimizin doğruluğunu artırmak için daha fazla çalışma yapılabilir. Bu çalışma, deep learning modellerinin gerçek hayatta kullanılabilirliğini göstermesi açısından önemli bir örnektir.

Kaynaklar

[1] https://www.nyc.gov/site/tlc/about/tlc-trip-record-data.page

[2] https://spark.apache.org/. Apache Spark. (2021). Retrieved

[3] https://databricks.com/. Databricks. (2021). Retrieved

[4] https://pandas.pydata.org/. Pandas. (2021). Retrieved

[5] Towards Data Science, "Linear Regression: Understanding the Theory", Towards Data Science, https://towardsdatascience.com/linear-regression-understanding-the-theory-7e53ac2831b5

[6] https://www.geeksforgeeks.org/linear-regression-python-implementation/ GeeksforGeeks, "Linear Regression (Python Implementation)", GeeksforGeeks,

[7] https://www.coursera.org/learn/machine-learning Andrew Ng, "Machine Learning", Stanford University (Online Course), https://www.coursera.org/learn/machine-learning

[8] https://machinelearningmastery.com/linear-regression-for-machine-learning/ Jason Brownlee, "Linear Regression for Machine Learning", Machine Learning Mastery, https://machinelearningmastery.com/linear-regression-for-machine-learning/