



**T.C.
GEBZE TEKNİK ÜNİVERSİTESİ**

Bilgisayar Mühendisliği Bölümü

**Makine Öğrenmesi Tabanlı Kayıp
Satışların İncelenme Sistemi**

Burak AKTEN

**Danışman
Yrd. Doç. Dr. Yakup Genç**

**Mayıs, 2019
Gebze, KOCAELİ**



**T.C.
GEBZE TEKNİK ÜNİVERSİTESİ**

Bilgisayar Mühendisliği Bölümü

**Makine Öğrenmesi Tabanlı Kayıp
Satışların İncelenme Sistemi**

Burak AKTEN

**Danışman
Yrd. Doç. Dr. Yakup GENÇ**

**Mayıs, 2019
Gebze, KOCAELİ**

Bu çalışma 30/05/2019 tarihinde aşağıdaki jüri tarafından Bilgisayar Mühendisliği Bölümü'nde Lisans Bitirme Projesi olarak kabul edilmiştir.

Bitirme Projesi Jürisi

| | | |
|--------------|---------------------------|--|
| Danışman Adı | Yrd. Doç. Dr. Yakup Genç | |
| Üniversite | Gebze Teknik Üniversitesi | |
| Fakülte | Mühendislik Fakültesi | |

| | | |
|------------|---------------------------|--|
| Jüri Adı | Doç. Dr. Mehmet GÖKTÜRK | |
| Üniversite | Gebze Teknik Üniversitesi | |
| Fakülte | Mühendislik Fakültesi | |

ÖNSÖZ

Bu projenin gerçekleştirilmesinde emeği geçenlere, projenin son halini almasında yol gösterici olan Sayın Yakup GENÇ hocama ve bu çalışmayı destekleyen Gebze Teknik Üniversitesi'ne içten teşekkürlerimi sunarım.

Ayrıca eğitimim süresince bana her konuda tam destek veren aileme ve bana hayatlarıyla örnek olan tüm hocalarıma saygı ve sevgilerimi sunarım.

Mayıs, 2019

Burak AKTEN

İÇİNDEKİLER

| | |
|---|------|
| ÖNSÖZ..... | iv |
| İÇİNDEKİLER | v |
| ŞEKİL LİSTESİ..... | vi |
| TABLO LİSTESİ | vii |
| KISALTMA LİSTESİ | viii |
| ÖZET | ix |
| SUMMARY | x |
| 1. GİRİŞ | 1 |
| 1.1 PROJE TANIMI | 2 |
| 1.2 PROJE TASARIM PLANI..... | 3 |
| 1.3 PROJENİN NEDEN VE AMAÇLARI..... | 4 |
| 1.4 BAŞARI KRİTERLERİ..... | 4 |
| 2 MALZEME VE YÖNTEM..... | 4 |
| 2.1 VERİ KÜMESİ VE ÖN İŞLEMLER..... | 4 |
| 2.2 YAZILIMSAL ARAÇLAR..... | 7 |
| 2.2.1 Numpy, Pandas, Scikit-tensor Kütüphaneleri..... | 7 |
| 2.2.2 Python Programlama Dili..... | 8 |
| 2.2.3 Visual Studio ve C# Programlama Dili..... | 9 |
| 2.2.4 Tensor Factorization..... | 8 |
| 2.3 Sistem Mimarisi..... | 11 |
| 3 BULGULAR..... | 12 |
| 4 SONUÇ..... | 14 |
| KAYNAKLAR..... | 16 |

ŞEKİL LİSTESİ

| | |
|--|----|
| Şekil 1 Makine Öğrenmesi Sınıflandırılması..... | 2 |
| Şekil 2 Proje Genel Yapı..... | 3 |
| Şekil 3 Proje Tasarım Planı..... | 3 |
| Şekil 4 Dört haftalık veri setinin ilişki(correlation) tablosu..... | 5 |
| Şekil 5 Ön işlemlerden sonraki ilişki (correlation) tablosu..... | 6 |
| Şekil 6 Tensor Factorization Yapısı..... | 10 |
| Şekil 7 Eşit boyutta test ve train datası olarak oluşturulan sistem..... | 13 |
| Şekil 8 Farklı boyuta test ve train datası olarak oluşturulan sistem..... | 13 |
| Şekil 9 Gerekli parametreden yapılan değişiklik sonucu Şekil 8'de belirtilen aynı data ile oluşturulan sistem..... | 14 |
| Şekil 10 Eğitim için bir haftalık datanın tümü ile oluşturulan sistem..... | 15 |
| Şekil 11 Bir haftalık veri ile oluşturulan sistem değerleri..... | 15 |

TABLO LİSTESİ

| | |
|---|----|
| Tablo 1 Eğitim için gerekli parametreler..... | 11 |
|---|----|

KISALTMA LİSTESİ

.csv : Dosya uzantı biçimi
.pdf : Dosya uzantı biçimi

ÖZET

Kayıp satışlar, bir firmanın sahip olduğu bir bayi/bayiler de yapmış olduğu satışlar da belli bir miktarda satışının yapılmasının beklenildiği bir ürünü veya ürünlerin beklenenden az satılmasıyla tespit edilen ürünlerdir. Çoğu firmada bu ürünlerin tespit edilerek bu ürünlere yönelik kampanya indirim vb. yöntemler ile bu ürünlerin satılmasını artırmak ister. Bu proje ile kayıp satışların tespitini makine öğrenmesi yöntemleri kullanarak yapılması amaçlanmıştır.

Projenin gerçekleşmesi için, oluşturulacak sistemin eğitiminde ve testinde kullanılacak çeşitli veri setlerine toplanmıştır. Toplanan bu veri de ürün-bayi ilişkisinin olmasına dikkat edilmiştir.

Veri toplama işlemleri bittikten sonra öğrenme aşaması için var olan veri üzerinde değişiklikler yapılmıştır. Sonrasında veri hazır olduğunda makine öğrenmesi teknikleri ile sistem oluşturulmuştur ve Web arayüzü ile kullanıcıdan gerekli bilgi ve veri sisteme girdi olması için istenilmiştir. Sistem aldığı inputlara göre sonuç dönebilmektedir.

SUMMARY

A product or products whose sales amount is less than a expected amount in a dealer of the company is treated as lost sales. Most companies want to identify these kind of products and increase the sales of these products through methods such as campaigns and discounts for these products. With this project, it is aimed to determine the loss sales by using machine learning techniques.

To achieve that, various dataset to be used in the training and testing of the system are collected. There should be a product-dealer relationship in this data. For this issue, the relationship has been paid attention when the data collected.

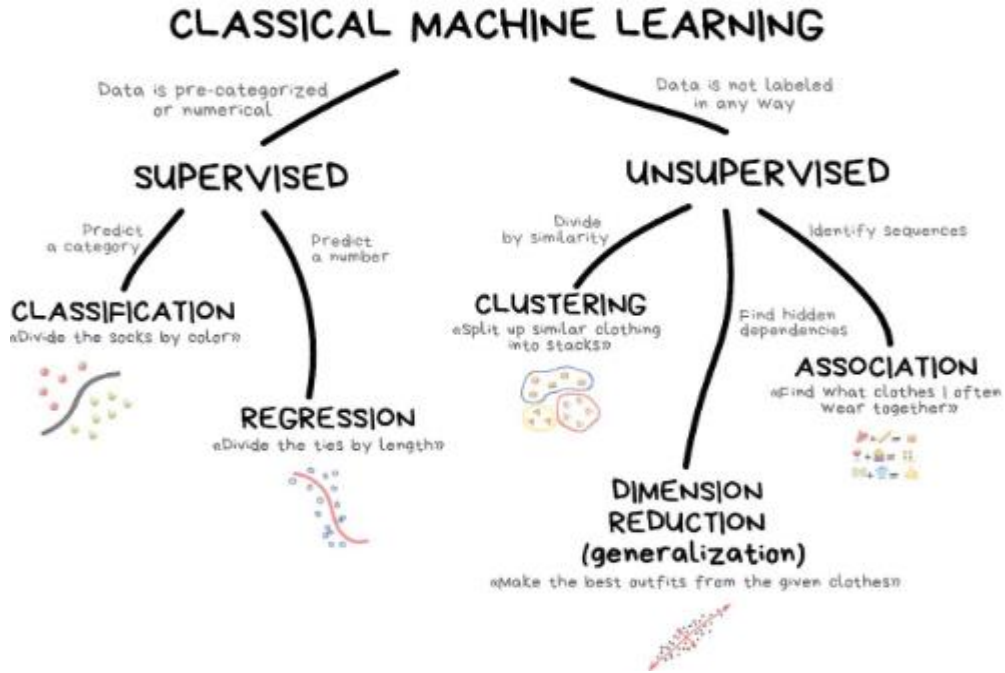
After the data collection process has been completed, changes have been made to the existing data for learning phase of the system. Then, when the data is ready, the system is trained with machine learning techniques and with Web interface the user of the system was required to have the necessary information and data entered into the system. Then the system can return results according to the inputs.

1 GİRİŞ

Çoğu firma kayıp satışların hesaplanmasının fiyatından/masraflarından dolayı değilde nasıl hesaplanacağı ve ne tür bir sistem kullanılacağı hakkında bilgisi olmadığından dolayı kayıp satışların belirlenmesi için gereken hesaplamayı yapamıyorlar. Kayıp satışların hesaplanması ile belirlenen tarih içerisinde firmanın ne kadar gelir kaybettiği de görülebilmektedir.

Bu kapsamda bu proje ile ticaret yapan firmalardan alışveriş verisi alınarak makine öğrenmesi yöntemleri ile bu veri üzerinde işlem yaptıktan sonra kayıp satışların hesaplanması ve raporlanması hedeflenmiştir.

Makine öğrenmesi, yazılım uygulamaların, açıkça programlamadan sonuçları tahmin etmede daha doğru olmalarını sağlayan bir algoritma kategorisidir. Makine öğreniminin temel öncülü girdi verilerini alabilen algoritmalar oluşturmak ve yeni veriler mevcut olduğunda çıktıları güncellerken çıktıyı tahmin etmek için istatistiksel analizi kullanmaktır. Makine öğrenme algoritmaları genellikle supervised veya unsupervised olarak kategorize edilir.[1] Supervised öğrenme, girdi değişkenlerinin (X) ve bir çıktı değişkeninin (Y) olduğu ve girdiden çıktıya eşleme işlevini öğrenmek için kullanılan bir algoritma türüdür ($Y = f(X)$). Unsupervised öğrenme, yalnızca girdi verilerinin (X) olduğu ve karşılık gelen çıktı değişkenlerinin olmadığı algoritma türüdür.[2]



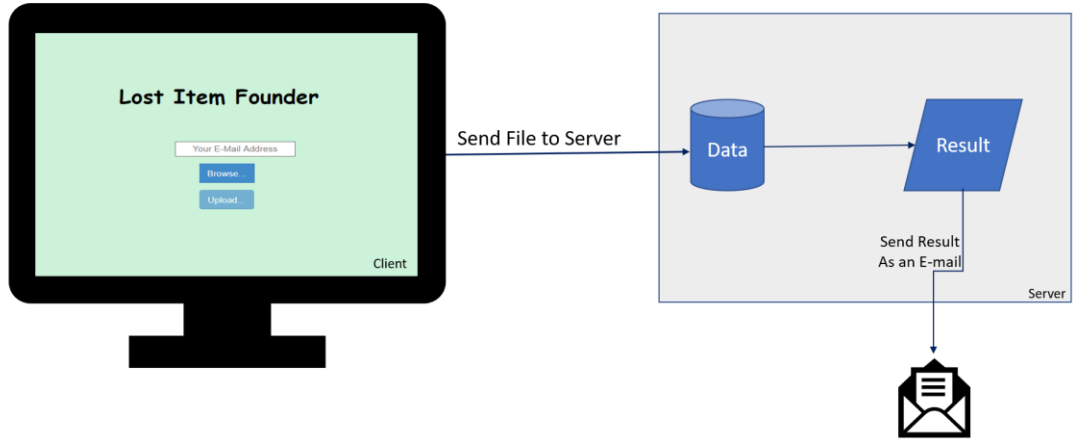
Şekil 1 Makine Öğrenmesi Sınıflandırılması

1.1 PROJE TANIMI

Bu projede ticaret yapan bir firmaya ait alışveriş verisi üzerinde istatistiksel bazı işlemler yaparak kayıp satışların bulunması hedeflenmiştir. Bunu yaparken makine öğrenmesi algoritmalarından olma “supervised öğrenme” yöntemi tabanlı “tensor factorization” gerçekleştirilmeye çalışılmıştır. Proje genel olarak üç kısımdan oluşmaktadır.

- İlk kısımda bir Web arayüzü gerçekleştirilmiştir. Bu arayüzden kullanıcı tarafından E-mail adresini yazması ve alışveriş datasını yüklenmesi istenmektedir.
- Kullanıcı istenilen bilgileri girdikten sonra bu bilgiler server tarafına iletilir ve burada yüklenen veri üzerinde oluşturulan sistem uygulanır. Sistem çıktı olarak kayıp satışların ID'lerini döner.

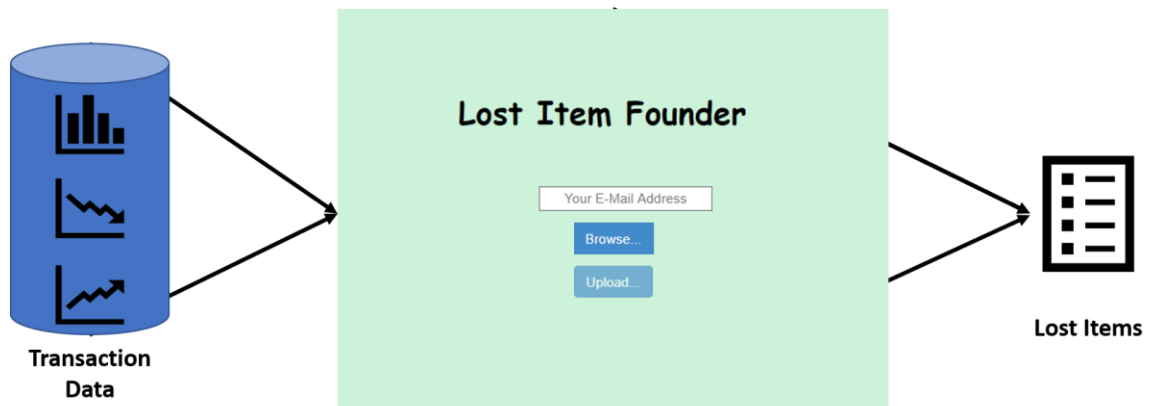
- Son kısımda ise elde edilen çıktıya göre bir .pdf formatında bir rapor oluşturulur ve mail yolu ile kullanıcıya gönderilir.



Şekil 2 Proje Genel Yapı

Genel olarak projeye bakıldığında **Hata! Başvuru kaynağı bulunamadı.** de de görüldüğü üzere sisteme ürün-bayi ilişkisi olan bir veri iletilir. Bu veri üzerinde oluşturulan sistem uygulanır ve kayıp satışlar bulunarak raporlanır.

1.2 PROJE TASARIM PLANI



Şekil 3 Proje Tasarım Planı

1.3 PROJENİN NEDEN VE AMAÇLARI

Çoğu firma kayıp satışların hesaplanmasının fiyatından/masraflarından dolayı değilde nasıl hesaplanacağı ve ne tür bir sistem kullanılacağı hakkında bilgisi olmadığından dolayı kayıp satışların belirlenmesi için gereken hesaplamayı yapamıyorlar. Kayıp satışların hesaplanması ile belirlenen tarih içerisinde firmanın ne kadar gelir kaybettiği de görülebilmektedir.

Bu kapsamda bu proje ile ticaret yapan firmalardan alışveriş verisi alınarak makine öğrenmesi yöntemleri ile bu veri üzerinde işlem yaptıktan sonra kayıp satışların hesaplanması ve raporlanması hedeflenmiştir.

1.4 BAŞARI KRİTERLERİ

- Oluşturulacak modelin train süresi maximum 5 saat olması bekleniyor. (100 ürün – 100 bayi için)
- Oluşturulan modelin haftalık veri için çalışma süresi en fazla 1 saat olması bekleniyor. (100 ürün – 100 bayi için)
- Sistemin Ortalama Kare Hata değeri (Mean Squared Error) %20 olması bekleniyor.

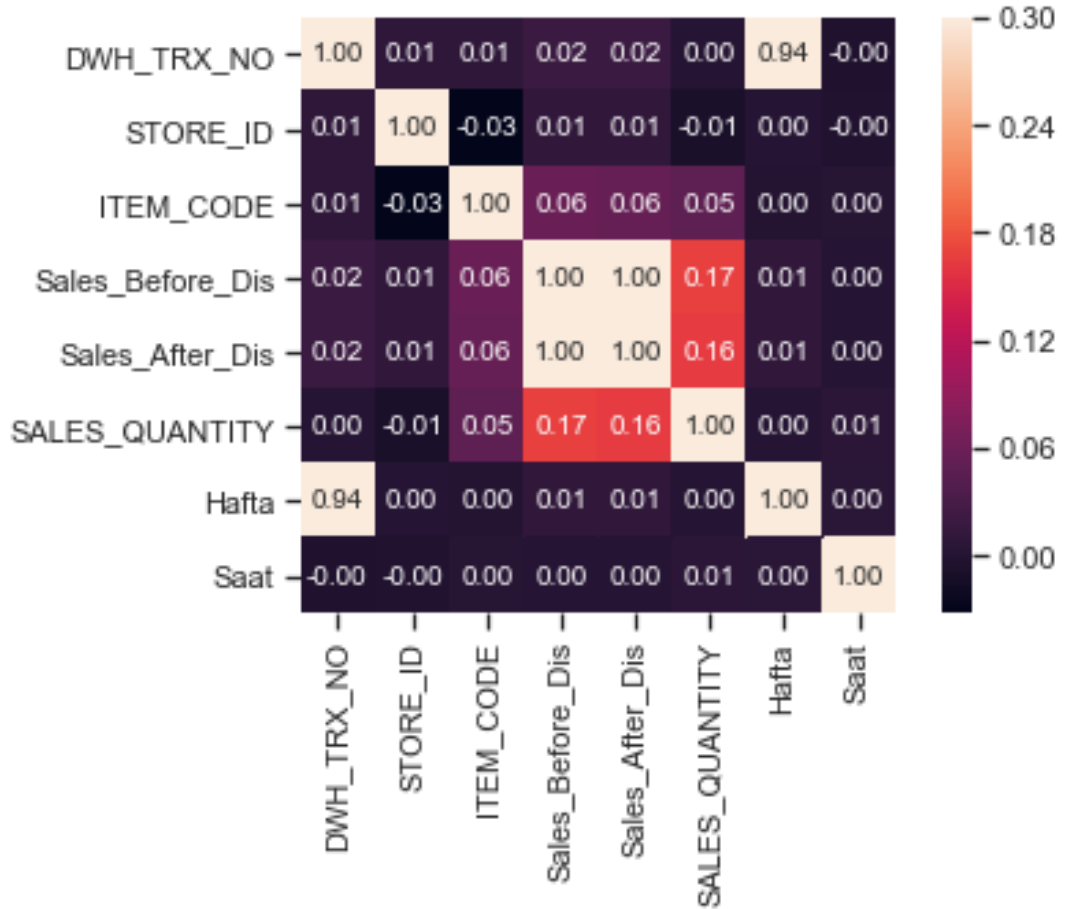
2 MALZEME VE YÖNTEM

2.1 VERİ KÜMESİ VE ÖN İŞLEMLER

Tasarlanan sistemi gerçeklemek için gerekli alışveriş verisi bir firmanın arşivinden temin edilmiştir. Temin edilen veri içerisinde ürün-bayi ilişkisi bulunmaktadır. Bu veri belirtilen firmanın 4 haftalık alışveriş verisini içermektedir. Her hafta ayrı ayrı olarak bir .csv uzantılı dosyaya kaydedilmiştir. Temin edilen bu verisinde aşağıdaki özellikler bulunmaktadır.

- DWH_TRX_NO
- ALISVERIS_TARIHI
- STORE_ID
- SALES_AMOUNT_BEFORE_DISCOUNT
- SALES_AMOUNT_AFTER_DISCOUNT
- SALES_QUANTITY

Öncelikle sahip olunan veri setindeki her bir kolonda bulunan değerlerin birbiri ile olan ilişkisini görmek adına 4 haftalık veri seti birleştirildi ve ilişki tablosu çıkarıldı. Bu tabloya göre herbir özelliğin bir biri ile olan ilişkisi görülmüş oldu. Ayrıca ek olarak veri setine “Saat” ve “Hafta” bilgisi “ALISVERIS_TARIHI” bilgisinden yararlanılarak eklendi.

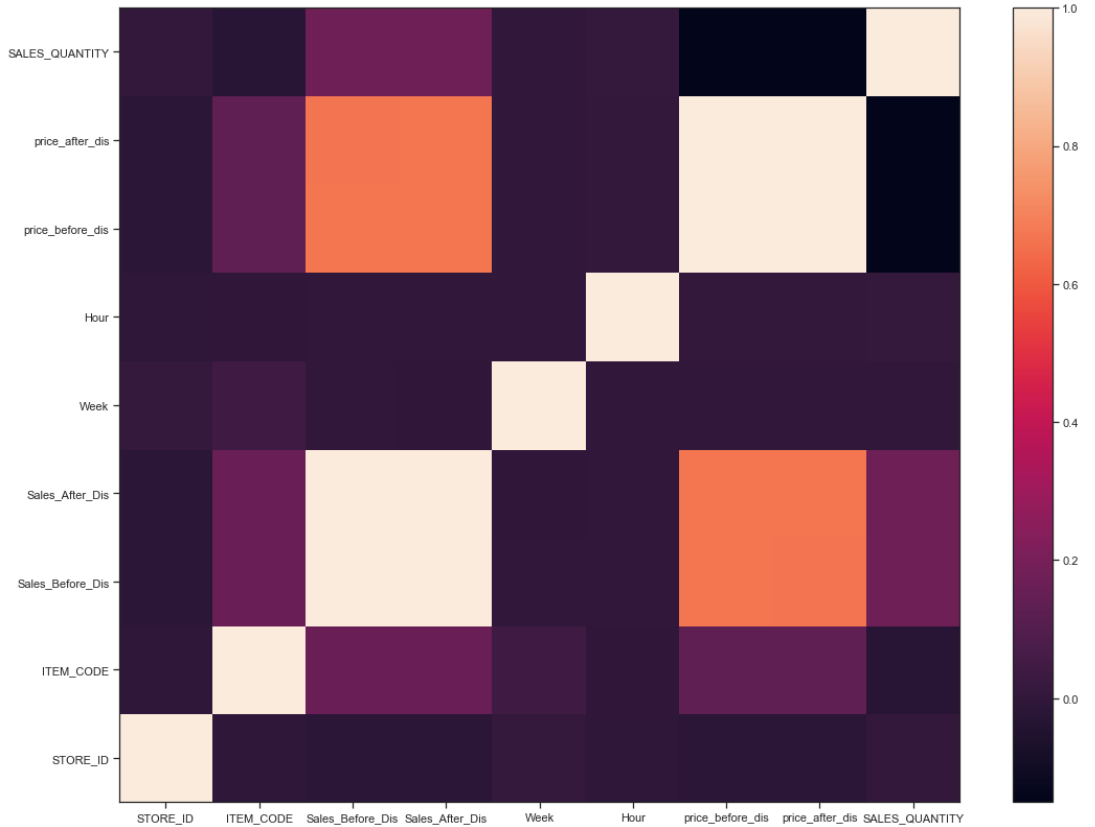


Şekil 4 Dört haftalık veri setinin ilişki(correlation) tablosu

Elimizde bulunan 4 haftalık alışveriş verisindeki kolonlarda bulunan değerlerin birbiri ile olan ilişki tablosunu çıkarttıktan sonra aşağıda belirtilen ön işlemler yapılarak veri seti zenginleştirilmeye ve daha kullanılabilir hale getirilmeye çalışıldı.

- “Sales_Before_Dis” ve “Sales_After_Dis” vektörlerindeki değerler “SALES_QUANTITY” vektörlerine bölünerek “price_after_dis” ve “price_before_dis” vektörleri elde edildi.
- “Store_ID” ve “ITEM_CODE” vektörleri büyük integer değerler olduğundan matematiksel işlemlerde işimizi kolaylaştırması açısından küçük değerlerle map edildi.
- **Hata! Başvuru kaynağı bulunamadı.**’de gösterilen ilişki tablosundan yararlanılarak veri setinde bulunan “DWH_TRX_NO” değerinin fazla etki etmeyeceği düşünülerek veri setinden çıkarıldı.

Bu ön işlemler bütün veri için yapıldı. Bu değişikliklerden sonra yeni bir hal alan veri setinin ilişki tablosu



Şekil 5 Ön işlemlerden sonraki ilişki (correlation) tablosu

Veri seti son halini aldıktan sonra test ve eğitim verisi olarak ikiye ayrıldı. Daha sonrasında da Tensor Factorization[3] olarak adlandırılan matematiksel yöntem implement edilerek veri setinin üzerinde uygulanıp bir sistem oluşturulmaya çalışıldı.

2.2 YAZILIMSAL ARAÇLAR

2.2.1 Numpy, Pandas, Scikit-tensor Kütüphaneleri

Her bir kütüphaneyi ayrı ayrı anlatacak olursak :

- **Numpy:** Python' da bilimsel hesaplamalarda kullanılan temel pakettir. Çok boyutlu diziler (array), çeşitli türetilmiş nesneler (maskelenmiş diziler ve matrisler gibi) ve bir sürü matematiksel , mantıksal, şekil manipülasyonu, sıralama, seçme, ayrık Fourier de dahil olmak üzere diziler üzerinde hızlı işlemler yapmamızı sağlayan Python kütüphanesidir. NumPy kullanılarak istatistik işlemleri ve simülasyonlarda yapılabilir. NumPy kütüphanesi C++ ile yazıldığından performans kaybı yaşanmaz. NumPy' da diziler homojen yapıdadır. Eğer elemanların hepsi `int` tipinde ise dizinin kendisinin de `int` tipinde olur.[4] Bu projede de verimizi tensor olacak numpy dişi içerisinde tuttuk ve bize sağlamış olduğu matematiksel kolaylıklardan yararlandık.
- **Pandas:** Python programlama dili için **yüksek performanslı**, kullanımı kolay **veri yapıları** ve **veri analiz araçları** sağlayan açık kaynaklı bir BSD lisanslı kütüphanedir. **Csv** ve **text** dosyalarını açmaya ve içerisinde bulunan verileri okuyarak istenen sonuca kolayca ulaşmak için kullanılmaktadır. Yani bir excel dosyasını açarak içerisinde bulunan bir sütunu veya satırı

seerek iřlemleri yapabiliriz. Numpy kütüphanesinde yapılan verilerin řekillendirilmesi daha detaylı bir biimde kullanılabilmektedir. Pandas hız konusunda optimize edilmiř ok hızlı bir řekilde iřlemleri yapabilmektedir.[5] Bu proje kapsamında bu kütüphaneyi verimizin bulunduėu .csv dosyasını okumak ve üzerinde bazı ön iřlemler yapmak için kullandık.

- **Scikit-tensor:** Multilinear ve tensör için bir Python modülüdür. Bu modül temel tensor operasyonlarını desteklemektedir[6] . Bizde Tensor Factorization olarak adlandırılan matematiksel fonksiyonu implment etmek için bu kütüphaneden yararlandık.

2.2.2 Python Programlama Dili

Python programlama dili, ilk sürümü Guido van Rossum tarafından 1991 yılında ortaya konulmuř genel amaçlı bir programlama dilidir. Python Yazılım Vakfı, Python'ın ana gerekleřimi olan C programlama dili gerekleřimini özgür ve açık kaynak kod mantıėı altında yürütmekte ve Python'ın fikirsel haklarını korumaktadır. Günümüzde Python, öğrenme kolaylıėı ve sahip olduėu geniř standart kütüphane sayesinde oldukça popülerleřmiř ve büyük kurumların da arasında olduėu yaygın bir kullanıcı kitlesine ulařmıřtır.[9]

Makine Öğrenmesi kütüphanleri dahil birok birok analistik iřlemler ile geliřtirilen projelerde genelde Python tercih ediliyor. Bundan dolayı bizde bu projenin eğitim kısmında, veri setini ayırma kısmında ve bazı diėer küçük kısımlarda hızlı sonuç alabilmek için bu dil tercih edilmiřtir.

2.2.3 Visual Studio ve C# Programlama Dili

Visual Studio, Microsoft tarafından üretilen ve konsollar, grafik kullanıcı arayüzleri, Windows formları, Web servisleri ya da Web uygulamaları oluşturmak için kullanılan bir IDE'dir. Visual Studio programı içerisinde yalnızca Microsoft Windows tarafından desteklenen yerel kodlar kullanılmaktadır. Visual Studio'nun bir nevi yazılım üretmek için yazılım olduğunu söyleyebiliriz. Visual Studio yazılımları elbette kendisi kodlamıyor ancak geliştiricilere ileri seviye özellikleri sayesinde yardımcı olarak daha kısa süre içerisinde programlarını hazırlama imkânı sunuyor. Visual Studio programını kullanarak bilgisayar yazılımları, web uygulamaları ve web servislerini çok daha hızlı ve kolay bir şekilde hazırlayabilirsiniz.[7]

C#, yazılım sektörü içerisinde en sık kullanılan iki yazılım dili olan C ve C++ etkileşimi ile türetilmiştir. Ayrıca C#, ortak platformlarda taşınabilir bir (portable language) programlama dili olan Java ile pek çok açıdan benzerlik taşımaktadır. En büyük özelliği ise .Net Framework platformu için hazırlanmış tamamen nesne yönelimli bir yazılım dilidir. Yani nesneler önceden sınıflar halinde yazılıdır. Yeni nesil programlama dili olan **C sharp programlama** konusunda yenilikleri de beraberinde getirmektedir. **C# sunucu** ve gömülü sistemleri için tasarlanmıştır. Microsoft tarafından geliştirilen **C# .NET** ortamında kullanılmak üzere geliştirilmiştir. Dilin tasarlanmasında Anders Hejlborg öncülük etmiştir. Zamanla gelişen bu programlama dili artık gerek masaüstü olsun gerek web uygulamaları

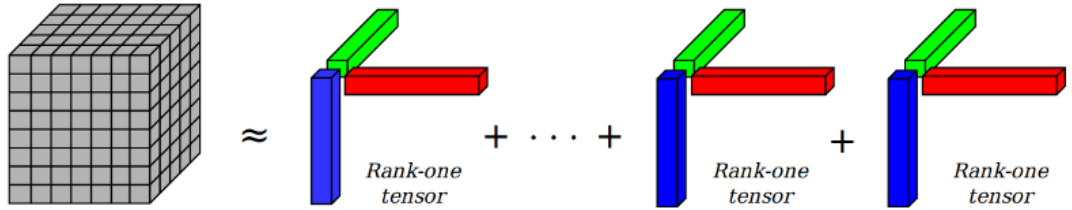
olsun çoğu yerde kullanılmaktadır.[8] Bizde bu projede kullanıcıdan email adresini ve veri setini yüklemesini istediğimiz Web kısımda Visual Studio ve C# kullandık.

2.2.4 Tensor Factorization

Multilinear cebirde bir tensor ayrışması (tensor fatorization) genellikle daha basit tensörler üzerinde etkili olan temel işlemlerin bir dizisi olarak bir tensörü ifade etmek için kullanılan bir yöntemdir. Birçok tensor ayrışması (tensor fatorization) bazı matris ayrışmalarını genelleştirir.[9] Temel tensor ayrışmaları şu şekildedir :

- Tensor Rank Decomposition
- Higher-order singular value decomposition
- Tucker Decomposition
- Tensor Trains
- Hierarchical Tucker Decomposition

Bizde bu proje kapsamında Tucker Decomposition yöntemini gerçeklemeye çalıştık.



Şekil 6 Tensor Factorization Yapısı

2.3 Sistem Mimarisi

Eğitim sürecinde, anlaşılan firmadan alınan veri çalışma ortamına atıldı ve bahsedilen ön işlemler yapıldıktan sonra implement edilen Tensor Factorization[9] yöntemini temsil eden fonksiyona input olarak verilerek eğitim işlemi gerçekleştirildi. Bu eğitim için yazılan fonksiyona üzerinde ön işlemler yapılmış verinin konumunun yanı sıra Tablo 1’de ki parametrelere de ihtiyaç duyulmuştur. Eğitim süresi veri setimizin büyüklüğü, gerekli parametreler arasında olan “k” değeri ve diğer parametrelerin değerine bağlı olarak değişiklik göstermektedir. Eğitimin sonunda elimizde benzer veri setini test edebileceğimiz bir model oluşmaktadır.

| Parametre | Tanım |
|-----------|--|
| “train” | Train file path |
| “test” | Test file path |
| “model” | Directory to save final model |
| “out” | File where the final result will be saved |
| “k” | Dimension of tensor |
| “reg” | Regularization for tensor facotrs |
| “regS” | Regularization for core tensor |
| “lr” | Initial learning rate for latent facotrs (default 0.1) |
| “lrS” | Initial learning rate for core tensor (default 0.1) |
| “maxEpo” | Max training epoch |
| “verbose” | Verbose or not |

Tablo 1 Eğitim için gerekli parametreler

Test kısmında ise iki farklı test işlemi gerçekleştirilebilir. İlkinde eğitim aşamasında oluşan modeli kullanarak testimizi gerçekleştirebiliriz. Ancak bu testi gerçekleştirebilmek için test için verilen veri seti train için verilen veri seti ile eş değer ve aynı ön işlemlere uğramış olması gerekmektedir. Eğer ki veri seti eğitim için kullanılan veri seti ile uyumsuz ise eğitim işlemi tekrar gerçekleştirilmeli ve yeni model oluşturulmalıdır ve test bu model üzerinde gerçekleştirilmelidir. İlk testimizde sonucumuzu saniyeler içinde alabilmekte iken ikinci test şeklinde eğitim durumuna ve veri setine bağlı olarak uzun süreler beklenebilmektedir.

3 BULGULAR

Projenin başlangıç aşamasında, makine öğrenmesi tarafında işlemler yapmadan önce, Android tarafında alışverişini alıp server'a iletecek bir uygulama yazılmaya çalışıldı fakat gönderilecek veri çok büyük veri olacağından mobil cihazda bulunması ve mobil cihazdan server'a iletilmesi sorun olduğundan bir Web uygulamasının yapılmasına karar verildi. Bu Web uygulamasına girdi olarak kullanıcı e-mail adresini ve test edilmesi istenilen alışveriş datasını almaktadır. Gönderilen alışveriş datası kişisel bilgisayardan seçilebilmektedir.

Sonrasında server kısmında Tensor Factorization olarak adlandırılan matematiksel fonksiyon Tablo 1 'de belirtilen parametreleri alacak şekilde implement edildi ve elimizde bulunan alışveriş verisi kullanılarak farklı parametre ve girdi değerleri ile farklı sistemler oluşturuldu. Oluşturulan sistemlerden bazıları aşağıdaki gibidir.

```

1 Xtest, core, U = main()

[INFO] ----- TF -----
[INFO] Data: Number of types for each feature = [70, 10809, 69, 69, 50, 13, 93, 93]
[INFO] Data: Training Size = 500. Testing Size = 500
[INFO] Settings: K = [5, 5, 5, 5, 5, 5, 5, 5]. reg = 0.01. regS = 0.01. lr = 0.0001. lrS = 0.001
[INFO] Iter 1/5. Time: 25.1
[INFO] Training Loss: 2.01 (Testing Loss: 2.01%). Testing MSE: 2.8850
[INFO] Iter 2/5. Time: 25.0
[INFO] Training Loss: 2.00 (Testing Loss: 2.00%). Testing MSE: 2.8851
[INFO] Iter 3/5. Time: 24.8
[INFO] Training Loss: 1.99 (Testing Loss: 1.99%). Testing MSE: 2.8851
[INFO] Iter 4/5. Time: 25.0
[INFO] Training Loss: 1.98 (Testing Loss: 1.98%). Testing MSE: 2.8851
[INFO] Iter 5/5. Time: 25.0
[INFO] Training Loss: 1.97 (Testing Loss: 1.97%). Testing MSE: 2.8851
[INFO] Total Time: 125.0s

```

Şekil 7 Eşit boyutta test ve train datası olarak oluşturulan sistem

Şekil 7’de gösterilen sistemin oluşması için içersinde 500 adet alışveriş verisi olan train ve test datası kullanıldı ve 5 iterasyon sonucu oluşan sistemin Ortalama Kare Hata değeri 2.8851 olarak hesaplanmıştır.

```

1 Xtest, core, U = main()

[INFO] ----- TF -----
[INFO] Data: Number of types for each feature = [70, 10809, 102, 102, 50, 13, 116, 116]
[INFO] Data: Training Size = 5000. Testing Size = 500
[INFO] Settings: K = [5, 5, 5, 5, 5, 5, 5, 5]. reg = 0.01. regS = 0.001. lr = 0.001. lrS = 0.0001
[INFO] Iter 1/5. Time: 245.5
[INFO] Training Loss: 0.48 (Testing Loss: 0.40%). Testing MSE: 2.8859
[INFO] Iter 2/5. Time: 244.6
[INFO] Training Loss: 0.48 (Testing Loss: 0.40%). Testing MSE: 2.8857
[INFO] Iter 3/5. Time: 245.1
[INFO] Training Loss: 0.48 (Testing Loss: 0.40%). Testing MSE: 2.8853
[INFO] Iter 4/5. Time: 245.5
[INFO] Training Loss: 0.48 (Testing Loss: 0.40%). Testing MSE: 2.8846
[INFO] Iter 5/5. Time: 245.1
[INFO] Training Loss: 0.48 (Testing Loss: 0.40%). Testing MSE: 2.8832
[INFO] Total Time: 1225.8s

```

Şekil 8 Farklı boyutta test ve train datası olarak oluşturulan sistem

Şekil 8’de gösterilen sistemde 5000 adet train için 500 adet test için alışveriş verisi kullanılmıştır. Diğer parametrelerde fazla oynama olmamıştır. Oluşan sistemin Ortalama Kare Hata değeri 2.8832 olarak hesaplanmıştır.

```

: 1 Xtest, core, U = main()

[INFO] ----- TF -----
[INFO] Data: Number of types for each feature = [70, 10809, 102, 102, 50, 13, 116, 116]
[INFO] Data: Training Size = 5000. Testing Size = 500
[INFO] Settings: K = [4, 4, 4, 4, 4, 4, 4, 4]. reg = 0.01. regS = 0.001. lr = 0.001. lrS = 1e-06
[INFO] Iter 1/5. Time: 21.5
[INFO] Training Loss: 0.84 (Testing Loss: 0.36%). Testing MSE: 2.8814
[INFO] Iter 2/5. Time: 21.7
[INFO] Training Loss: 0.84 (Testing Loss: 0.36%). Testing MSE: 2.8808
[INFO] Iter 3/5. Time: 21.5
[INFO] Training Loss: 0.84 (Testing Loss: 0.36%). Testing MSE: 2.8798
[INFO] Iter 4/5. Time: 21.3
[INFO] Training Loss: 0.84 (Testing Loss: 0.36%). Testing MSE: 2.8782
[INFO] Iter 5/5. Time: 21.9
[INFO] Training Loss: 0.84 (Testing Loss: 0.36%). Testing MSE: 2.8753
[INFO] Total Time: 107.9s

```

Şekil 9 Gerekli parametreden yapılan değişiklik sonucu Şekil 8'de belirtilen aynı data ile oluşturulan sistem

Şekil 9'da belirtitlen sistem için Şekil 8'de kullanılan veriri ile aynı boyutta veri kullanıldı ve Tablo 1'de belirtilen parametreler ile yapılan değişiklikler sonucu oluşan sistemin Ortalama Kare Hata değeri 2.8753 olarak hesaplanmıştır.

Veriler bu şekilde verildiğinde oluşan sistemlerden diğerlerine göre en iyi sistem Şekil 8'de belirtilen sistemdir.

4 SONUÇ

Test sonuçları ve proje süresince kazanılan deneyimler eşliğinde durumu değerlendirirsek oluşan oluşturulan sistemin elimizde bulunan veri setinin büyüklüğü ve çeşitliği ile doğru orantılı olmasına yanı sıra eğitim sırasındaki adım sayısı ve Tablo 1'de belirtilen parametrelere bağlı değişkenlik gösterdiğini görmüş olduk.

Bir önceki kısımda elimizdeki veriyi parça parça olarak incelediğimizde kazanmıl olduğumuz sonuçları inceledik. Şimdi ise birhaftalık veriyi eğitim için kullandığımızda sahip olduğumuz en iyi değeri inceleyeceğiz.

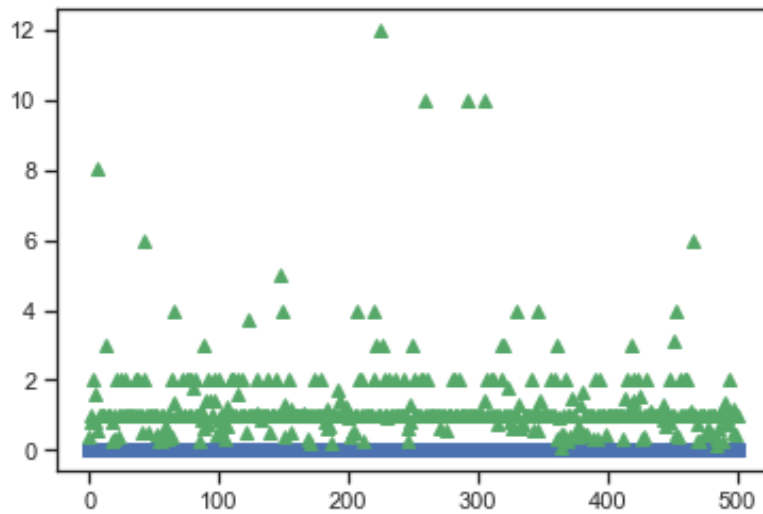

```

1 xtest, core, U = main()
[INFO] ----- TF -----
[INFO] Data: Number of types for each feature = [70, 10809, 2577, 2577, 50, 13, 812, 812]
[INFO] Data: Training Size = 1279084. Testing Size = 500
[INFO] Settings: K = [4, 4, 4, 4, 4, 4, 4, 4]. reg = 0.1. regS = 0.1. lr = 0.001. lrS = 0.0001
[INFO] Iter 1/5. Time: 4918.6
[INFO] Training Loss: 52708362.74 (Testing Loss: 15180.22%). Testing MSE: 2.8883
[INFO] Iter 2/5. Time: 4956.5
[INFO] Training Loss: 16764679098976.55 (Testing Loss: 4826993966.87%). Testing MSE: 2.8883
[INFO] Iter 3/5. Time: 4913.8
[INFO] Training Loss: 4928312612913712128.00 (Testing Loss: 1418991385986565.75%). Testing MSE: 2.8883
[INFO] Iter 4/5. Time: 4935.6
[INFO] Training Loss: 1340984678194298220445696.00 (Testing Loss: 386104911874224029696.00%). Testing MSE: 2.8883
[INFO] Iter 5/5. Time: 4944.8
[INFO] Training Loss: 338207308227397486730171908096.00 (Testing Loss: 97378818014682012459204608.00%). Testing MSE: 2.8883
[INFO] Total Time: 24669.3s

```

Şekil 10 Eğitim için bir haftalık datanın tümü ile oluşturulan sistem

Şekil 10 ‘da bir haftalık yani 1.279.084 adet alışveriş verisi bulunan veri seti eğitim için 500 adet ise test için kullanıldı ve Tablo 1’de belirtilen parametreler Şekil 9’da belirtilen sistem için verilen parametre değerleri ile benzer verildi. Oluşan sistemin Ortalama Kare Hata değeri 2.8883 olarak hesaplanmıştır. Bu sistemde de görüldüğü gibi ortalma kare hata değeri hep 2’nin üzerinde çıkmaktadır. Elimizde bulunan verinin karakteristik özellikleri ve implememnt edilen Tensor Factorization yönteminden kaynaklı olarak bu değer hep 2’nin üzerinde olmaktadır. Base olarak 2 değerini alırsak hata olarak %80 civarı olarak düşünebiliriz. Son olarak bu sstem ile tahminlemesi yapılan ürünler ve gerçek değerlerini Şekil 11’de ki gibi olmuştur.



Şekil 11 Bir haftalık veri ile oluşturulan sistem değerleri

KAYNAKLAR

- [1] Rouse, M. (2016). *What is machine learning (ML)? - Definition from WhatIs.com.* [online] SearchEnterpriseAI. Available at: <https://searchenterpriseai.techtarget.com/definition/machine-learning-ML> [Accessed 15 May 2019].
- [2] Brownlee, J. (2016). *Supervised and Unsupervised Machine Learning Algorithms.* [online] Machine Learning Mastery. Available at: <https://machinelearningmastery.com/supervised-and-unsupervised-machine-learning-algorithms/> [Accessed 15 May 2019].
- [3] Medium. (2019). *Tensor Decomposition in Python.* [online] Available at: <https://medium.com/@mohammadbashiri93/tensor-decomposition-in-python-f1aa2f9adbf4> [Accessed 31 Mar. 2019].
- [4] Numpy.org. *NumPy — NumPy.* [online] Available at: <https://www.numpy.org> [Accessed 17 May 2019].
- [5] Pandas.pydata.org. *Python Data Analysis Library — pandas: Python Data Analysis Library.* [online] Available at: <https://pandas.pydata.org> [Accessed 17 May 2019].
- [6] GitHub. *mnick/scikit-tensor.* [online] Available at: <https://github.com/mnick/scikit-tensor> [Accessed 17 May 2019].
- [7] Visual Studio. *Visual Studio IDE, Code Editor, Azure DevOps, & App Center - Visual Studio.* [online] Available at: <https://visualstudio.microsoft.com/?rr=https%3A%2F%2Fwww.google.com.tr%2F> [Accessed 17 May 2019].
- [8] GeeksforGeeks. *C# Programming Language - GeeksforGeeks.* [online] Available at: <https://www.geeksforgeeks.org/csharp-programming-language/> [Accessed 17 May 2019].
- [9] En.wikipedia.org. (n.d.). *Tensor decomposition.* [online] Available at: https://en.wikipedia.org/wiki/Tensor_decomposition [Accessed 17 May 2019].