BANDIRMA ONYEDİ EYLÜL ÜNİVERSİTESİ

MÜHENDİSLİK VE DOĞA BİLİMLER FAKÜLTESİ

BİLGİSAYAR MÜHENDİSLİĞİ

Veri Madenciliği Dersi

Projesi

Kaggle Gelecekteki Satışları Tahmin Ediyor

HAZIRLAYAN

Ali Yılmaz-191522010

DANIŞMAN

Dr. Ögr. Üyesi Buket Toptaş

2023

BANÜ 2022-2023 Eğitim Öğretim Yılı Veri Madenciliği (BLM4146) Proje Raporu		
Öğrenci Adı	Ali	
Öğrenci Soyadı	Yılmaz	
Öğrenci Numarası	191522010	

1. Giriş

1.1. Seçilen proje konusunun önemi nedir?

Gelecekteki satışları tahmin etmek, bir işletmenin şu anda ve gelecekteki performansı hakkında bilgi sahibi olmasını sağlar. Bu bilgi sayesinde işletme, stok yönetimi, üretim planlaması ve finansal tahminler gibi alanlarda daha bilinçli kararlar alabilir.

Örneğin, eğer bir işletme gelecekteki satışları doğru tahmin edebiliyorsa, müşteri talebini karşılamak için yeterli stok miktarını elinde bulundurabilecek ve gereksiz depolama maliyetlerine maruz kalmayacaktır. Bu sayede, işletme stoklarını daha etkin bir şekilde yönetebilir ve bu da üretim süreçlerini daha verimli hale getirebilir. Ayrıca, doğru satış tahminleri, bir işletmenin üretim programlarını planlamasına ve kaynaklarını etkin bir şekilde kullanmasına yardımcı olabilir. Bu sayede, işletme üretim süreçlerinde daha az kaynak harcayarak daha fazla ürün üretebilir.

Son olarak, gelecekteki satışlar hakkında bilgi sahibi olmak, bir işletmenin finansal stratejisini planlamasına ve bütçeleme ve yatırımlar gibi konularda daha bilinçli kararlar almasına yardımcı olabilir. Örneğin, eğer bir işletme gelecekteki satışları doğru tahmin edebiliyorsa, bütçesini daha doğru bir şekilde oluşturabilir ve yatırımlarını daha etkin bir şekilde yönetebilir. Bu sayede, işletme daha sağlıklı ve uzun vadeli bir büyüme hedefi oluşturabilir. Bu nedenlerden dolayı, gelecekteki satışları tahmin etmek önemlidir ve bu konuda yapılacak olan çalışmaların doğruluğu ve hassasiyeti önemlidir.

1.2. Bu konu neden seçilmiştir?

Bu konu, gelecekteki satışları tahmin etmek, bir işletmenin şu anda ve gelecekteki performansı hakkında bilgi sahibi olmak ve bu bilgi sayesinde işletme, stok yönetimi, üretim planlaması ve finansal tahminler gibi alanlarda daha bilinçli kararlar alınması nedeniyle seçilmiştir.

1.3. Kaggle Gelecekteki Satış Tahmininin Literatürdeki Yeri

1.3.1. LSTM (Uzun Kısa Süreli Bellek)

Yazarlar, LSTM ağlarının tahmin problemlerinde iyi performans gösterdiğinden, bilginin korunmasını ve hatırlanmasını gerektirdiğinden bahsetmektedir, bu da LSTM için kullanılacak doğal model seçimi olmasının nedenidir [1]. Bu makalede LSTM, raporda açıklanacaktır.

1.3.2. Sınıflandırma ve Regresyon Ağacı Yöntemi

Bu çalışmada kesikli ve sürekli verilerin analizinde kullanılabilen sınıflandırma ve regresyon ağacı yöntemi Çanakkale ilinde zeytinyağı tüketici anketinden elde edilen veriler ile anlatılmaya çalışılmıştır. [2] Bu makalede yöntemler bahsedilecektir.

2. Kullanılan Veri Setinin Temini

Bu veri seti, Rus yazılım şirketinin sunduğu "Predict Future Sales" adlı bir Kaggle yarışmasından alınmıştır. Bir sonraki ayda her ürün ve mağaza için toplam satışlar tahmin edilecektir.[3]

2.1. Dosya açıklamaları

- **sales_train.csv** - eğitim seti. Ocak 2013'ten Ekim 2015'e kadar günlük tarihsel veriler.
- **test.csv** - test seti. Kasım 2015 için bu mağaza ve ürünlerin satışlarını tahmin etmeniz gerekiyor.
- **items.csv** - ürünler/ürünler hakkında ek bilgiler.
- **item_categories.csv** - ürün kategorileri hakkında ek bilgi.
- **shop.csv** - mağazalar hakkında ek bilgi.

2.2. Veri Alanları

2.2.1. ID - test kümesi içindeki bir (Mağaza, Öğe) grubunu temsil eden bir Kimlik

2.2.2. shop_id – bir mağazanın ID'si

2.2.3. item_id – bir ürünün ID'si

2.2.4. item_category_id – ürün kategori ID'si

2.2.5. **item_cnt_day** - satılan ürün sayısı

2.2.6. **item_price** - bir ürünün geçerli fiyatı

2.2.7. **date** - gg/aa/yyyy biçimindeki tarih

2.2.8. **date_block_num** - kolaylık sağlamak için kullanılan ardışık bir ay numarası.

2.2.9. **item_name** - ürünün adı

2.2.10. **shop_name** – mağazanın adı

2.2.11. item_category_name - ürün kategorisinin adı

3. Kullanılan Veri Yönteminin Teorik Bilgisi

3.1. Karar Ağacı

Karar ağacı, bir kurum veya kuruluş tarafından tercihlerin, risklerin, kazançların ve hedeflerin anlaşılmasına yardımcı olan bir teknik türüdür. Aynı zamanda birçok önemli yatırım sahalarında uygulanabilen, birbiriyle bağlantılı şans olaylarıyla ilgili olarak çıkan çeşitli karar noktalarını incelemek için kullanılan bir karar destek aracıdır. Yalnızca koşullu kontrol ifadeleri içeren bir algoritmayı görüntülemenin bir yoludur.

Karar ağacı, bir hedefe ulaşma olasılığı en yüksek olan stratejiyi belirlemeye yardımcı olmak için kullanılan bir yöntemdir. Özellikle karar analizinde olmak üzere karmaşık sorunların araştırmasında yaygın olarak kullanılmaktadır. Ayrıca makine öğrenmesinde kullanılan yaygın bir araçtır.[4]

3.1.1. Sınıflandırma Ağaçları

Bir kişinin harcamalarından eğitim düzeyinin tahmini gibi, hedef kümeyi çeşitli sınıflardan birisine yerleştirmeyi amaçlayan ve sınıf tanımı yapan problemlerdir.[5]

3.1.2. Regresyon Ağaçları

Regresyon, iki ya da daha çok değişken arasındaki ilişkinin fonksiyonel halini veren bir tahmin edici veri madenciliği tekniğidir. Yani, değişkenler arasındaki ilişkinin matematiksel bir ifadesidir.

Sürekli değerleri tahmin etmek için başvurulan regresyon modellerinde, girdiler ile çıktılar arası bağ kurabilecek bir fonksiyon oluşturup, tahmin başarısı eniyilemeye çalışılmaktadır. Değişkenler arası ilişkiyi doğrusal varsayarak, bir bağımsız değişkene sahip modellere basit doğrusal regresyon modeli, birden fazla bağımsız değişkene sahip modellere çoklu doğrusal regresyon modeli olarak adlandırılır.[6]

3.2. Zaman Serisi Tahmini

Zaman Serisi Tahmini (Time Series Forecasting), bir makine öğrenimi modelidir ve belirli eylemler için en iyi zamanlamayı tahmin etmek için kullanılır.

Tarihsel verileri kullanır ve tarihsel verilerdeki kalıpları belirler. Örneğin, bir araç üreticisi, stoklamanın ne zaman gerekli olduğunu tahmin etmek için geçmiş verileri, zaman serisi modeliyle analiz edebilir. Benzer şekilde, bir perakendeci yeni bir ürünün piyasaya sürülmesini planlamak için bu modeli kullanabilir.[7]

3.3. LSTM (Uzun Kısa Süreli Bellek)

Uzun kısa süreli bellek derin öğrenme alanında kullanılan yapay bir yinelemeli sinir ağı (RNN) mimarisidir. Standart ileri beslemeli sinir ağlarının aksine, LSTM'nin geri bildirim bağlantıları vardır. Yalnızca anlık veriyi (resim gibi) değil, veri dizilerini (konuşma veya video gibi) de işleyebilir. Örneğin, LSTM bölümlenmemiş, bağlı el yazısı tanıma, konuşma tanıma ve ağ trafiğinde anomali veya IDS'lerde (saldırı tespit sistemleri) tespiti gibi görevler için geçerlidir.

LSTM ağları, zaman serisi verilerine dayanarak sınıflandırmak, işlemek ve tahminler yapmak için çok uygundur, çünkü bir zaman serisindeki önemli olaylar arasında bilinmeyen süreli gecikmeler olabilir. LSTM'ler, geleneksel RNN'leri eğitirken karşılaşılabilecek patlayan ve yok olan gradyan problemleriyle başa çıkmak için geliştirilmiştir.[8]

3.4.RNN (Tekrarlayıcı Sinir Ağı)

Yinelemeli sinir ağı, düğümler arası bağların zamansal bir dizi doğrultusunda yönlü çizge oluşturduğu bir yapay sinir ağı çeşididir. Yaygın olarak İngilizce kısaltması olan RNN olarak anılır. İleri beslemeli sinir ağından türetilen RNN yöntemi, bir iç durum belleği kullanarak değişik uzunluktaki dizileri işleyebilir. Bu sayede yazı tanıma ve konuşma tanıma gibi problemlere uygulanabilir.[9]

4. Projenin Uygulanması ve Performans analizi

Bu projenin uygulanmasında, karşılaştırma için iki yöntem kullanılmıştır. Bu iki yöntem karar ağaçları ve uzun kısa süreli bellektir. Karar ağaçlarının uygulanmasında LSTM'den daha kolay olduğu anlaşılmıştır.

4.1. LSTM Kodunun Açıklanması

4.1.1. Verilerin Okunması

```
clc; clear;

data_items = readtable('data/items.csv');
data_shops = readtable('data/shops.csv');
```

```
data_cats = readtable('data/item_categories.csv');
data_train = readtable('data/sales_train.csv');
data_test = readtable('data/test.csv');

data_train = head(data_train,1000000)
```

Bu kod bloğunda veri setleri, değişkenlere atanır ve data_train'nin ilk bir milyon verisi gösterilir.

4.1.2. Verilerin Gruplanması

```
[G, ID] = findgroups(data_train(:, 3:4));
sale_counts = splitapply(@(month, sale){count_sales(month, sale)}, data_train(:, 2),
data_train(:, 6), G);
sale_counts = cell2mat(sale_counts);
sales = [ID table(sale_counts)];
head(sales)
```

Bu MATLAB kodu, "data_train" değişkeninde bulunan satış verilerine dayalı olarak çeşitli işlemler yapıyor. İlk olarak, "findgroups" fonksiyonu kullanılarak veri kümesinde satırların gruplandırılması sağlanıyor. Bu gruplandırma, 3. ve 4. sütunlardaki değerlere göre yapılıyor (bu sütunlar, mağaza ID ve ürün ID sütunlarını temsil ediyor).

Sonra, "splitapply" fonksiyonu kullanılarak veri kümesi üzerinde bir işlem uygulanıyor. Bu işlem, "count_sales" adında bir fonksiyon tarafından yapılıyor ve bu fonksiyonun açıklaması verilmemiş. Ancak, fonksiyonun içinde "month" ve "sale" değişkenleri kullanılıyor. Bu değişkenler, "data_train" değişkeninde bulunan 2. ve 6. sütunları temsil ediyor (bu sütunlar, tarih ve satış adedi sütunlarını temsil ediyor).

Sonuç olarak, "sale_counts" değişkenine bir cell dizisi atanıyor ve bu dizi, gruplandırılmış veri kümesine uygulanan işlemlerin sonuçlarını içerir. Bu cell dizisi, "cell2mat" fonksiyonu kullanılarak bir matris haline dönüştürülüyor. Son olarak, "sales" değişkenine bir tablo atanıyor ve bu tablo, gruplandırılmış veri kümesinin ID'leri ve uygulanan işlemlerin sonuçlarını içerir. Bu tablo, "head" fonksiyonu kullanılarak gösteriliyor.

4.1.3. Verilerin Standardizasyonu

```
mu = mean(sale_counts.')';
sig = std(sale_counts.')';
sales_train_standardized = (sale_counts - mu) ./ sig;
sales_train_standardized(isnan(sales_train_standardized)) = 0
```

Bu MATLAB kodu, öncelikle "sale_counts" değişkeninde bulunan verilere dayalı

olarak birkaç istatistiksel ölçüm hesaplıyor. Bu ölçümler, ortalama ("mu") ve standart sapma ("sig") değerleridir. Bu değerler, "sale_counts" değişkeninin her bir sütunu için ayrı ayrı hesaplanır.

Sonra, "sale_counts" değişkeninin her bir elemanı için bir standartlaştırma işlemi uygulanır. Bu işlem, her elemanın ortalama değerden çıkarılması ve bu değerin standart sapma değerine bölünmesiyle gerçekleştirilir. Bu işlem sonucu, "sales_train_standardized" değişkenine bir matris atanır.

Son olarak, "sales_train_standardized" değişkeninde bulunan "NaN" değerleri (Not-a-Number, sayı değil) sıfır değerine dönüştürülür. Bu, "isnan" fonksiyonu kullanılarak yapılır. Bu fonksiyon, verilen matriste "NaN" değerlerini bulur ve bunları "true" değerine dönüştürür. Bu "true" değerleri kullanarak, "sales_train_standardized" değişkeninde bulunan "NaN" değerleri sıfır değerine dönüştürülür.

```
x_train = sales_train_standardized(:, 1:end-1);
y_train = sales_train_standardized(:, 2:end);
```

Bu MATLAB kodu, "sales_train_standardized" değişkeninde bulunan verileri iki farklı değişkene atar. Öncelikle, "x_train" değişkenine "sales_train_standardized" değişkenindeki tüm sütunların bir kısmı atanır. Bu kısım, 1. sütun ile sütun sayısının 1 eksiğine kadar olan tüm sütunları temsil eder. Bu değişken, veri kümesinin girdi verilerini (input features) temsil eder.

Sonra, "y_train" değişkenine "sales_train_standardized" değişkenindeki tüm sütunların bir kısmı atanır. Bu kısım, 2. sütun ile sütun sayısına kadar olan tüm sütunları temsil eder. Bu değişken, veri kümesinin hedef verilerini (target values) temsil eder.

Bu kod bloğu, veri kümesinin girdi verileri ve hedef verileri olarak iki ayrı değişkene atanmasını sağlar. Bu, daha sonra veri kümesini eğitim ve test olarak bölmek için kullanılabilir.

4.1.4. LSTM Ağ Mimarisinin Tanımlanması

```
group_height = height(ID);

num_inputs = group_height;
num_responses = group_height;
num_hidden_units = 75;

layers = [ ...
    sequenceInputLayer(num_inputs)
    lstmLayer(num_hidden_units)
    fullyConnectedLayer(num_responses)
```

```
    regressionLayer];

options = trainingOptions('adam', ...
    'ExecutionEnvironment',"gpu", ...
    'MaxEpochs',200, ...
    'InitialLearnRate',0.05, ...
    'LearnRateSchedule','piecewise', ...
    'MiniBatchSize',20, ...
    'LearnRateDropPeriod',50, ...
    'LearnRateDropFactor',0.5, ...
    'VerboseFrequency', 25, ...
    'Plots','training-progress' );
```

Bu MATLAB kodu, bir derin öğrenme modeli oluşturmak için gerekli olan bazı yapılandırmaları yapar. Bu model, LSTM (Long Short-Term Memory) yapısına sahip bir yapay sinir ağıdır ve veri kümesinde zaman serisi verilerini tahmin etmeyi amaçlar. İlk olarak, "ID" tablosunun satır sayısı "group_height" değişkenine atanır. Bu değişken, veri kümesinin gruplandırılmış haliyle ilgili bir ölçümdür.

Sonra, model için gerekli olan bazı yapılandırmalar yapılır. Bu yapılandırmalar arasında, girdi verilerinin sayısı ("num_inputs"), hedef verilerinin sayısı ("num_responses"), gizli katmanların birim sayısı ("num_hidden_units") gibi değişkenler bulunur.

Daha sonra, modelin katmanları tanımlanır. Bu katmanlar arasında, girdi katmanı, LSTM katmanı, tam bağlı katman ve regresyon katmanı bulunur. Bu katmanlar, "layers" değişkenine atanır.

Son olarak, eğitim seçenekleri ("options") tanımlanır. Bu seçenekler arasında, eğitim algoritmasının türü ("adam"), eğitimin gerçekleştirileceği ortam ("ExecutionEnvironment"), eğitimin maksimum epoch sayısı ("MaxEpochs"), başlangıç öğrenme hızı ("InitialLearnRate"), öğrenme hızının nasıl düşürüleceği ("LearnRateSchedule"), mini batch büyüklüğü ("MiniBatchSize"), öğrenme hızının düşürüleceği epoch sayısı ("LearnRateDropPeriod"), öğrenme hızının düşürüleceği faktör ("LearnRateDropFactor") gibi değişkenler yer alır. Bu seçenekler, "options" değişkenine atanır.

4.1.5. LSTM Ağının Eğitilmesi

```
if isfile("model\net.mat")
    load net
else
    net = trainNetwork(x_train, y_train, layers, options);
    save('model\net.mat', 'net', '-v7.3');
```

```
end
```

Bu MATLAB kodu, daha önce eğitilmiş bir yapay sinir ağı modeli var mı yok mu kontrol eder. Eğer model varsa, bu model "net" değişkenine yüklenir. Eğer model yoksa, veri kümesi kullanılarak yeni bir model oluşturulur ve bu model "net" değişkenine atanır.

Model oluşturma işlemi, "trainNetwork" fonksiyonu kullanılarak gerçekleştirilir. Bu fonksiyon, veri kümesinin girdi verileri ("x_train"), hedef verileri ("y_train"), modelin katmanları ("layers") ve eğitim seçenekleri ("options") gibi argümanları alır.

Model oluşturulduktan sonra, "save" fonksiyonu kullanılarak model dosyaya kaydedilir. Bu dosya, "model\net.mat" adresinde bulunur ve "net" değişkenini içerir. "-v7.3" seçeneği, modelin daha sonra MATLAB versiyonlarında da açılabilmesini sağlar.

4.1.6. Gelecek Zaman Adımlarının Tahmin Edilmesi

```
% Initialize Net.
net = predictAndUpdateState(net, x_train);
[net, y_pred] = predictAndUpdateState(net, y_train(:, end));

% Predict next months sales
[net, pred] = predictAndUpdateState(net, y_pred);

pred = pred .* sig + mu;
pred_r = round(pred);

result = addvars(ID, sale_counts, pred_r);
```

Bu MATLAB kodu, daha önce eğitilmiş olan yapay sinir ağı modelini kullanarak veri kümesinde bulunan zaman serisi verilerini tahmin etme işlemini yapar. Bu tahminler, "pred" değişkenine atanır.

İlk olarak, model "predictAndUpdateState" fonksiyonu kullanılarak önceden eğitilmiş verilere göre güncellenir. Bu fonksiyon, modeli ("net") ve girdi verilerini ("x_train") alır ve güncellenmiş modeli ve tahmin verilerini döndürür. Bu işlemler, ilk iki satırda gerçekleştirilir.

Sonra, model "predictAndUpdateState" fonksiyonu kullanılarak, veri kümesinde bulunan son hedef veriye göre tahmin yapılır. Bu tahmin, "y_pred" değişkenine atanır.

Sonra, model "predictAndUpdateState" fonksiyonu kullanılarak, önceki tahmin verilerine göre bir sonraki ayın satış verilerini tahmin etme işlemi gerçekleştirilir. Bu tahminler, "pred" değişkenine atanır.

Daha sonra, tahmin edilen veriler standartlaştırma işleminin tersine çevrilir. Bu işlem,

tahmin verilerinin standart sapma değerine çarpılması ve ortalama değer eklenmesiyle gerçekleştirilir. Bu işlemler, "pred" değişkeninin "sig" ve "mu" değişkenleriyle çarpılıp toplamasıyla gerçekleştirilir.

Son olarak, tahmin verileri yuvarlanır ve "pred_r" değişkenine atanır. Bu yuvarlama işlemi, "round" fonksiyonu kullanılarak gerçekleştirilir.

Son olarak, tahmin verileri ve gerçek veriler "result" tablosuna eklenir. Bu tablo, "ID" tablosu ve "sale_counts" ve "pred_r" değişkenlerinin birleştirilmesiyle oluşturulur. "addvars" fonksiyonu kullanılarak bu işlem gerçekleştirilir.

4.1.7. Sonuç

```
result_sorted = sortrows(result, "pred_r", "descend");
head(result_sorted(:, [1 2 4]))
```

Bu kod parçacığı, result adlı bir veri kümesinin satırlarını "pred_r" adlı sütun değerlerine göre büyükten küçüğe sıralar. Daha sonra, bu sıralanmış veri kümesinin ilk 5 satırını seçer ve sadece 1., 2. ve 4. sütunlarını gösterir.

sortrows fonksiyonu, veri kümesinin satırlarını belirtilen sütun değerlerine göre sıralamaya yarar. Örnekte, "pred_r" sütunu kullanılarak satırlar sıralanmıştır. "descend" argümanı, sıralamanın büyükten küçüğe yapılmasını belirtir.

head fonksiyonu, veri kümesinin ilk n (varsayılan olarak n=5) satırını gösterir. Bu örnekte, ilk 5 satır seçilmiştir ve result_sorted adlı sıralanmış veri kümesinden seçilir.

	shop_id	item_id	pred_r
1	0	30	0
2	0	31	0
3	0	32	0
4	0	33	0
5	0	35	0
6	0	36	0
7	0	40	0
8	0	42	0

Şekil 1 pred_r sonucu

4.2. Classification Decision Tree Kodlarının Açıklanması

4.2.1. Test Veriyle Tahmin

```
data_test = addvars(data_test(:, 2:3), 34.* ones(height(data_test), 1), 'Before', 'shop_id');
```

```
data_test.Properties.VariableNames = {'date_block_num', 'shop_id', 'item_id'};
[sale_cnt, score, cost] = predict(c_tree, data_test);
```

Bu kod parçacığında, data_test adlı bir veri kümesine yeni bir sütun eklenir. Bu sütun, data_test veri kümesinin satır sayısı kadar 34 değerlerinden oluşan bir vektörle doldurulur. Bu yeni sütun, data_test veri kümesinin "shop_id" sütununun öncesine eklenir ve "shop_id" olarak adlandırılır.

Bu işlemi yapmak için, addvars fonktiyonu kullanılır. Bu fonksiyon, veri kümesine yeni sütunlar eklemek için kullanılır. data_test(:, 2:3) ifadesi, data_test veri kümesinin sadece 2. ve 3. sütunlarını seçer. Daha sonra, bu seçilen sütunlarla birlikte, yeni sütun olarak verilen 34 değerlerinden oluşan vektör eklenir. Bu yeni sütun, "Before" argümanı ile "shop_id" sütununun öncesine eklenir.

Daha sonra, data_test veri kümesinin sütun isimleri yeniden atanır. Önceki isimleri kullanılmaz ve sütunlar sırasıyla "date_block_num", "shop_id" ve "item_id" olarak adlandırılır.

Son olarak, predict fonksiyonu kullanılarak data_test veri kümesi üzerinde tahminler yapılır. Bu fonksiyon, bir model ve bir veri kümesi alır ve veri kümesi üzerinde tahminler yapar. Bu örnekte, c_tree adlı bir model kullanılmıştır ve data_test veri kümesi kullanılarak tahminler yapılmıştır. Tahminler, sale_cnt değişkenine atanır ve ayrıca, score ve cost değişkenlerine de değerler atanır.

Tahmini satışlar gruplarla birleştirilir:

```
data_test_result = addvars(data_test, sale_cnt);
```

4.2.2. Tahmin Değerinin Hesaplanması

```
% Tahmin edilen en yüksek değerlere dayalı olarak maksimum puanların hesaplanması
max_prediction_scores = max(score, [], 2);
avg_prediction_score = mean(max_prediction_scores);
```

Bu kodda, **score** veri kümesinin maksimum değerleri alınıp ortalaması hesaplanmıştır. Bu ortalama değer, tahmini değeri gösterir.

4.3.Regression Decision Tree

4.3.1. Test Veriyle Tahmin

```
data_test = addvars(data_test(:, 2:3), 34.* ones(height(data_test), 1), 'Before', 'shop_id');
data_test.Properties.VariableNames = {'date_block_num', 'shop_id', 'item_id'};
[sale_cnt, score, cost] = predict(c_tree, data_test);
```

4.3.2. Tahmini Satışlarla Grupları Birleştirme

```
data_test_result = addvars(data_test, sale_cnt);
```

5. Sonuçlar ve Değerlendirme

5.1. Karar Ağaçları Tahmin Sonuçları

	date_block_num	shop_id	item_id	sale_cnt
1	34	5	5037	3.2857
2	34	5	5320	2.1667
3	34	5	5233	3
4	34	5	5232	3
5	34	5	5268	1.6364
6	34	5	5039	3.2857
7	34	5	5041	1.2162
8	34	5	5046	1.2162

Şekil 2: Sınıflandırma Ağacı Tahmin Sonucu

Sınıflandırma ağacının ortalama tahmin değeri 0.838287 olarak hesaplanmıştır

	date_block_num	shop_id	item_id	sale_cnt
1	34	5	5037	3.2857
2	34	5	5320	2.1667
3	34	5	5233	3
4	34	5	5232	3
5	34	5	5268	1.6364
6	34	5	5039	3.2857
7	34	5	5041	1.2162
8	34	5	5046	1.2162

Şekil 3: Regresyon Ağacı Tahmin Sonucu

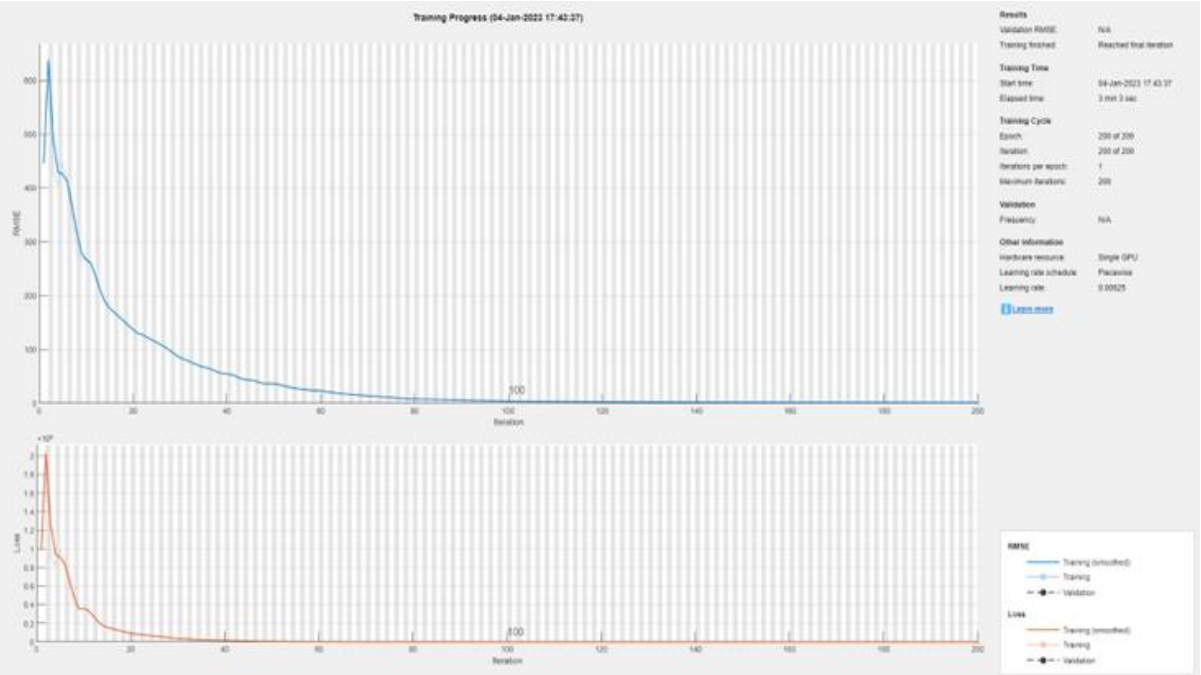
BANÜ 2022-2023 Eğitim Öğretim Yılı Veri Madenciliği (BLM4146) Proje Raporu		
Öğrenci Adı	Ali	
Öğrenci Soyadı	Yılmaz	
Öğrenci Numarası	191522010	

5.2. LSTM Sonuçları

	shop_id	item_id	pred_r
1	0	30	0
2	0	31	0
3	0	32	0
4	0	33	0
5	0	35	0
6	0	36	0
7	0	40	0
8	0	42	0

Şekil 4: LSTM Tahmin Sonucu

Epoch	Iteration	Time Elapsed (hh:mm:ss)	Mini-batch RMSE	Mini-batch Loss	Base Learning Rate
1	1	00:00:03	445.99	99453.0	0.0500
25	25	00:00:27	113.31	6419.6	0.0500
50	50	00:00:50	38.11	726.0	0.0500
75	75	00:01:11	10.62	56.4	0.0250
100	100	00:01:35	3.93	7.7	0.0250
125	125	00:01:57	2.23	2.5	0.0125
150	150	00:02:19	1.64	1.4	0.0125
175	175	00:02:40	1.42	1.0	0.0063
200	200	00:03:03	1.24	0.8	0.0063



Şekil 5: LSTM ile Veri Setinin Eğitilmesi

BANÜ 2022-2023 Eğitim Öğretim Yılı Veri Madenciliği (BLM4146) Proje Raporu		
Öğrenci Adı	Ali	
Öğrenci Soyadı	Yılmaz	
Öğrenci Numarası	191522010	

6. Kaynakça

[1] "A comparative study between LSTM and ARIMA for sales forecasting in retail", Ajla Elmasdotter, Carl Nyströmer, 06, June 2018.

[2] https://tarekoder.org/2012konya/203-209.pdf

[3]https://www.kaggle.com/competitions/competitive-data-science-predict-future-sales

[4] https://tr.wikipedia.org/wiki/Karar_a%C4%9Fac%C4%B1

[5] https://bilgisayarkavramlari.com/2012/04/11/karar-agaci-ogrenmesi-decision-tree-learning/

[6] https://www.datasciencearth.com/python-uygulamasi-ile-karar-agaclari/

[7] https://blog.turhost.com/veri-madenciligi-nedir/

[8] https://tr.wikipedia.org/wiki/Uzun_k%C4%B1sa_s%C3%BCreli_bellek

[9] https://tr.wikipedia.org/wiki/Yinelemeli_sinir_a%C4%9F%C4%B1

MATLAB

- https://www.mathworks.com/help/deeplearning/ug/long-short-term-memory-networks.html
- https://www.mathworks.com/help/deeplearning/ug/time-series-forecasting-using-deep-learning.html
- https://www.mathworks.com/help/deeplearning/ref/trainnetwork.html
- https://www.mathworks.com/help/deeplearning/ref/trainingoptions.html
- https://www.mathworks.com/help/stats/fitctree.html
- https://www.mathworks.com/help/matlab/ref/findgroups.html
- https://www.mathworks.com/help/matlab/ref/splitapply.html
- https://www.mathworks.com/help/deeplearning/ref/nnet.cnn.layer.fullyconnectedlayer.html
- https://www.mathworks.com/help/deeplearning/ref/regressionlayer.html
- https://www.mathworks.com/help/deeplearning/ref/nnet.cnn.layer.lstmlayer.html
- https://www.mathworks.com/help/stats/compactclassificationtree.predict.html
- https://www.mathworks.com/help/deeplearning/ref/predictandupdatestate.html