T.C KOCAELİ ÜNİVERSİTESİ

BÜYÜK VERİ ANALİZİ

PREDICT FUTURE SALES

Muhammet Sercan Oruc

https://www.kaggle.com/sercanorc/161307008

Bilişim Sistemleri Mühendisliği Bölümü

Kocaeli Üniversitesi

161307008@kocaeli.edu.tr

Özet

Kaggle.com üzerinde sunulan Predict Future Sales başlığı adı altında açılan yarışma için bir çözüm önerisi geliştirildi. Yapılan Çalışmada 2013-2015 yılları arasında yer alan verilerden faydalanıldı[1].

Bu yarışmada, en büyük Rus yazılım firmalarından biri olan 1C Company tarafından sağlanan günlük satış verilerinden oluşan zorlu bir zaman serisi veri kümesiyle çalıştık. Yarışmada Önümüzdeki ay her ürün ve mağaza için toplam satışları tahmin etmeyi amaçlanmaktadır.

Abstract

A solution proposal has been developed for the competition under the title of Predict Future Sales on Kaggle.com. The study made use of the data between 2013-2015 [1].

In this competition, we worked with a challenging time series dataset consisting of daily sales data provided by 1C Company, one of the largest Russian software companies. The competition aims to estimate the total sales for each product and store in the next month.

Problem Tanımı

Predict Future Sales problemi için python programlama dili kullanarak verileri analiz ettikten sonra sales doğrultusunda her ay her ürün ve mağaza için toplam satışları tahmin etmemiz beklenmektedir

Veriseti bilgisi

Günlük geçmişte yapılmış satış verileri sağlanmaktadır. Görev, test seti için her mağazada satılan toplam ürün miktarını tahmin etmektir. Mağaza ve ürün listesinin her ay biraz değiştiği gözlenmiştir.

Dosya açıklamaları

sales_train.csv - eğitim seti. Ocak 2013'ten Ekim 2015'e kadar günlük geçmiş veriler.

test.csv - test kümesi. Bu mağazalar ve ürünler için Kasım 2015 satışlarını tahmin etmeniz gerekiyor.

sample_submission.csv - doğru formatta örnek bir gönderim dosyası.

items.csv - öğeler / ürünler hakkında tamamlayıcı bilgiler.

item_categories.csv - öğe kategorileri hakkında tamamlayıcı bilgiler.

shops.csv - mağazalar hakkında ek bilgiler.

Veri alanları

Id - test kümesi içindeki bir (Mağaza, Ürün) grubunu temsil eden bir Kimlik

shop_id - bir mağazanın benzersiz tanımlayıcısı

item_id - bir ürünün benzersiz tanımlayıcısı

item_category_id - öğe kategorisinin benzersiz tanımlayıcısı

item_cnt_day - satılan ürünlerin sayısı. Bu önlemin aylık miktarını tahmin ediyorsunuz

item_price - bir öğenin mevcut fiyatı

tarih - gg / aa / yyyy biçiminde tarih

tarih_block_num - kolaylık sağlamak için kullanılan ardışık bir ay numarası. Ocak 2013 0, Şubat 2013 1, ... Ekim 2015 33

item_name - öğenin adı

shop_name - dükkanın adı

item_category_name - öğe kategorisinin adı

Algoritmalar

- LinearRegression
- DecisionTreeClassifier

İcerik

Linear Regression: Basit doğrusal regresyon, iki sürekli (nicel) değişken arasındaki ilişkileri özetlememize ve incelememize izin veren istatistiksel bir yöntemdir: X olarak adlandırılan bir değişken, öngörücü, açıklayıcı veya bağımsız değişken olarak kabul edilir. Y olarak adlandırılan diğer değişken, yanıt, sonuç veya bağımlı değişken olarak kabul edilir. Basit doğrusal regresyon, sıfatı "basit" olarak alır, çünkü yalnızca bir tahmini değişkenin çalışması ile ilgilidir.

Decision Tree: Karar ağacı öğrenmesi (decision treelearning) yöntemi, makine öğrenmesi(machine learning) konularından birisidir. Literatürde karar ağacı öğrenmesinin alt yöntemleri olarak kabul edilebilecek sınıflandırma ağacı(classification tree) veya ilkelleştirme ağacı (regression tree,tahmin ağacı) gibi uygulamaları vardır. Karar ağacı öğrenmesinde, bir ağaç yapısı oluşturularak ağacın yaprakları seviyesinde sınıf etiketleri ve bu yapraklara giden ve başlangıçtan çıkan kollar ile de özellikler üzerindeki işlemeler ifade edilmektedir.

Analiz ve Yöntem

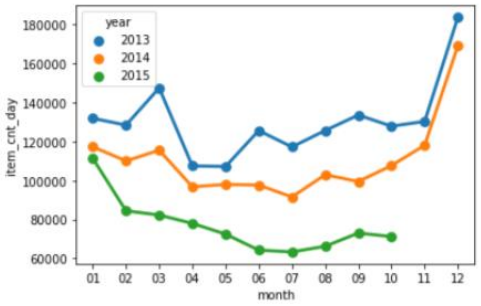
İlk olarak veri seti incelendiğinde train_set datasında negatif değerler olduğu gözlenmiştir.

```
train_set
```

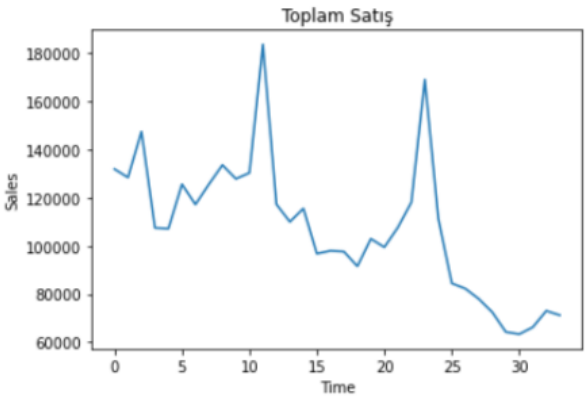
	date	date_block_num	shop_id	item_id	item_price	item_cnt_day
0	02.01.2013	0	59	22154	999.00	1.0
1	03.01.2013	0	25	2552	899.00	1.0
2	05.01.2013	0	25	2552	899.00	-1.0
3	06.01.2013	0	25	2554	1709.05	1.0
4	15.01.2013	0	25	2555	1099.00	1.0

Satılan ürün sayısı negatif olamayacağından, negatif ürün satırlarının kaldırılması ile analiz işlemi başlamıştır.

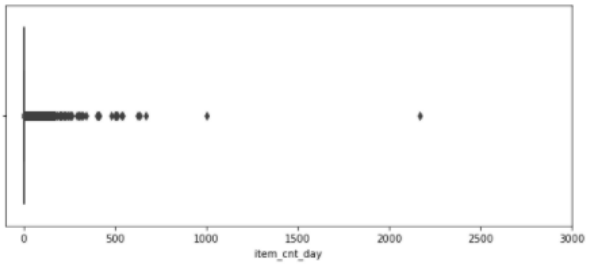
sales_train veri kümesi içindeki date değişkeninde belirtilen tarihleri ay ve yıla göre sınıflandırma işlemi yapıldı ve Yıllara göre aylık satış grafiği oluşturuldu

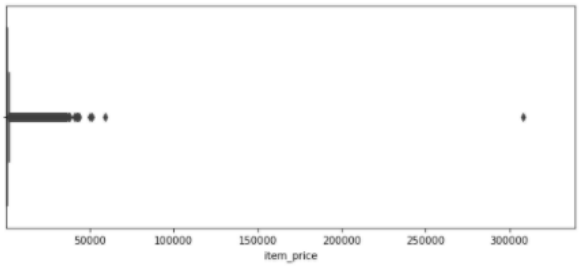


Ve date_block_num verisine göre Toplam satış grafiği yapıldı



Train veri seti tekrar incelendiğinde item_cnt_day ve item_price verilerinde ayrık veriler gözlemlenmiştir bunların olması model eğitimini olumsuz etkileyeceği için veri setinden kaldırılmıştır.





Aykırı değerlerin kaldırılması

```
#Aykırı değerlerin kaldırılması
train_set = train_set[(train_set.item_price < 100000 ) & (train_set.item_cnt_day < 800)]
```

Train_set verisinden ortalama aylık ürün fiyatı ve toplam satılan ürün sayısı analiz amaçlı elde edilmiştir

```
monthly_sales=train_set.groupby(["date_block_num","shop_id","item_id"])[
    "date","item_price","item_cnt_day"].agg({"date":["min",'max'],"item_price":"mean","item_cnt
_day":"sum"})
monthly_sales
```

Aşşağıdaki kod parçası ile oluşturulan grafikte

```
plt.figure(figsize=(3,32))
ax= sns.barplot(x.item_category_id, x.item_id, alpha=0.8)
plt.title("Items per Category")
plt.ylabel('# of items', fontsize=12)
plt.xlabel('Category', fontsize=12)
plt.show()
```

En çok 40 numaralı kategoride satış yapıldığı gözlemlenmiştir

Model

```
X_train, X_test, y_train, y_test = train_test_split(mounths,total_item_cnt_mounth_np, test_size = 2/3, random_state = 123, shuffle=2)
```

```
X_train = X_train.reshape(-1, 1)
X_test = X_test.reshape(-1, 1)
y_train = y_train.reshape(-1, 1)
y_test = y_test.reshape(-1, 1)
```

```
model = LinearRegression()
model.fit(X_train, y_train)
model.score(X_train,y_train)
```

```
from sklearn import tree
model_DTC = tree.DecisionTreeClassifier(criterion='gini')
```

+ Code + Markdown

```
model_DTC.fit(X_train, y_train)
model.score(X_train,y_train)
```

Aylar ve aylık toplam satış verisine göre modeli oluşturuyoruz

Sonuçlar

LinearRegression	**0.8148**
DecisionTree	**1.0**

Bu sonuçlar üzerinde çıkarımım Decision tree algroitmesi veri kümesini ezberlediği için 1 değeri veriyor Linerar regression ise 0.81 gibi gayet güzel bir sonuç çıkartıyor

2.Model

Train set verisi üzerinde ['year','date_block_num','month','item_price'] kolonları alınarak item_cnt_dayi tahmin etme çalışmasıdır.

Buradaki sonuçlarda algoritmalar

LinearRegressionScore 0.0006306616749353067

DTCscore 0.9040736142405338

Skorlarını vermektedir.

Linearregresion iki değişken arasındaki ilişkileri özetlememize ve incelememiz noktasında başarılı olan bir algoritma olduğu için başarısız sonuç vermesi gayet normaldir. Decisiontree ise 0.90 gibi oldukça yüksek bir skor vermektedir.

Teknolojiler

Proje kaggle linki :

https://www.kaggle.com/sercanorc/161307008

Projede tercih edilen programlama dili “Python” olmuştur.

Veri analizi için Numpy Pandas

Görselleştirme için Seaborn kütüphanesei

Decision ve LineaerRegression için Scikit-learn’in hazır olarak sunduğu kütüphanelerden faydalanılmıştır

Kaynakça

[1] '/kaggle/input/competitive-data-science-predict-future-sales/sales_train.csv'
[2]https://en.wikipedia.org/wiki/Decision_tree#See_also

[3] https://towardsdatascience.com/simple-and-multiple-linear-regression-in-python-c928425168f9