



ROSSMANN MAĞAZA SATIŞLARI TAHMİNİ

Harun ÇATAL

Danışman: Prof. Dr. Şule Gündüz Öğüdücü

❖ ÖZET

- Projenin Açıklaması
- Projenin Amacı
- Projenin Kapsamı
- Proje Hakkında Teknik Bilgiler
- Veri Kümeleri
- Özellikler
- Ön Hazırlık
- Ekstra Özellik Türetme
- Veri Analizi
- Modeller
- Sonuç
- Görselleştirme
- Kaggle Sonuçları

❖ Projenin Açıklaması

- Proje bir Kaggle yarışmasıdır.
- Rossmann Avrupa'nın çeşitli ülkelerinde yayılmış olan 3000'den fazla mağazası olan kozmetik ve ilaç ticareti yapan bir şirkettir.



❖ Projenin Amacı

- 1 Ocak 2013 ile 31 Temmuz 2015 tarihleri arasında 1115 Rossmann mağazasına ait satış verilerini kullanarak 856 Rossmann mağazasının 17 Eylül 2015' e kadar olan satışlarını öngörmektir.
- Projenin amacı Rossmann şirketine sağlam bir satış tahmin modeli oluşturmaktır.
- Ayrıca mağaza müdürlerini doğru şekilde bilgilendirerek mağaza hakkında gerekli önlemler almalarını sağlamaktır.

❖ Projenin Kapsamı

- Farklı ön hazırlık metodlarını uygulama
- Uygun modelleri seçme
- Uygun modelleri veri kümesine uygulama
- Verileri görselleştirme
- Sonuçları yorumlama

❖ Proje Hakkında Teknik Bilgiler

- Problem Türü: Veriden tahmin yapma (Forecasting)
- Programlama Dili: Python 3.5
- Programlama Ekipmanları: Python Anaconda Spyder IDE
- Kullanılan Kütüphaneler: Pandas, Skilearn, Numpy, Matplotlib, Seaborn

❖ Veri Kümeleri

□ Projede kullanılmak üzere elimizde 3 adet veri kümesi vardır.

No	Veri Kümesi	Özellikler	Nitelik sayısı	Boyut(Satır)
1	Train	store, day of week, date, sales,customers, open, promo, state holiday, school holiday	9	1017210
2	Store	store, storetype, assortment, competition distance, competition open since month, promo2, promo2since week, promo2since year, promo interval	10	1115
3	Test	id, store, dayofweek, date, open, promo, state holiday, school holiday	8	41089

❖ Veri Kümeleri

- Train veri kümesindeki eksik bilgileri tamamlamak için Train ve Store veri kümelerine katma(join) işlemi uygulanmıştır.

❖ Özellikler

No	Özellik	Muhtemel Değerler
1	Store	1 ile 1115 arası
2	DayOfWeek	1,2,3,4,5,6,7
3	Date	01.01.2013 ile 31.07.2015 arası
4	Sales	0 ile 41551 arası
5	Customers	0 ile 7338 arası
6	Open	0(Kapalı), 1(Açık)
7	Promo	0(Promosyon yok), 1(Promosyon)
8	State Holiday	a: Resmi tatil b: Paskalya c: Noel 0: Tatil yok
9	School Holiday	0(Yok), 1(Var)
10	Store Type	a: En büyük b, c, d: En küçük
11	Assortment	a: Normal b: Ekstra c: Genişletilmiş

❖ Özellikler

No	Özellik	Muhtemel Değerler
12	Competition distance	20 ile 70860 arası
13	Competition open since month	1 ile 12 arası
14	Competition open since year	1900 ile 2015 arası
15	Promo2 (Uzun süreli promosyon)	0(Yok), 1(Var)
16	Promo2 since week	1 ile 50 arası
17	Promo2 since year	2009-2015
18	Promo interval	(jan, apr, jul, oct) (fab, may, aug, nov) (mar, jun, sept, dec)

❖ Ön Hazırlık

- ❑ Verideki boşlukların silinmesi
- ❑ Açık mağazaların veri kümesine alınması
- ❑ Satış yapan mağazaların satış yapılan günlerinin alınması
- ❑ Tarih değişkenini parçalayıp özellik olarak ekleme
- ❑ Haritalama işlemleri

❖ Ekstra Özellik Türetme

□ Süre Aralığının Anlamlandırılması

- CompetitionOpen, PromoOpen özelliklerinde

- 1 yıl 3 ay = 15, 3 yıl 3 ay = 39

❖ Ekstra Özellik Türetme

- Tarihin promosyon aralığında olduğunun kontrolü
 - PromoInterval= (jan, apr, jul, oct)
(fab, may, aug, nov)
(mar, jun, sept, dec)
- - isPromoMonth özelliği eklendi.

❖ Ekstra Özellik Türetme

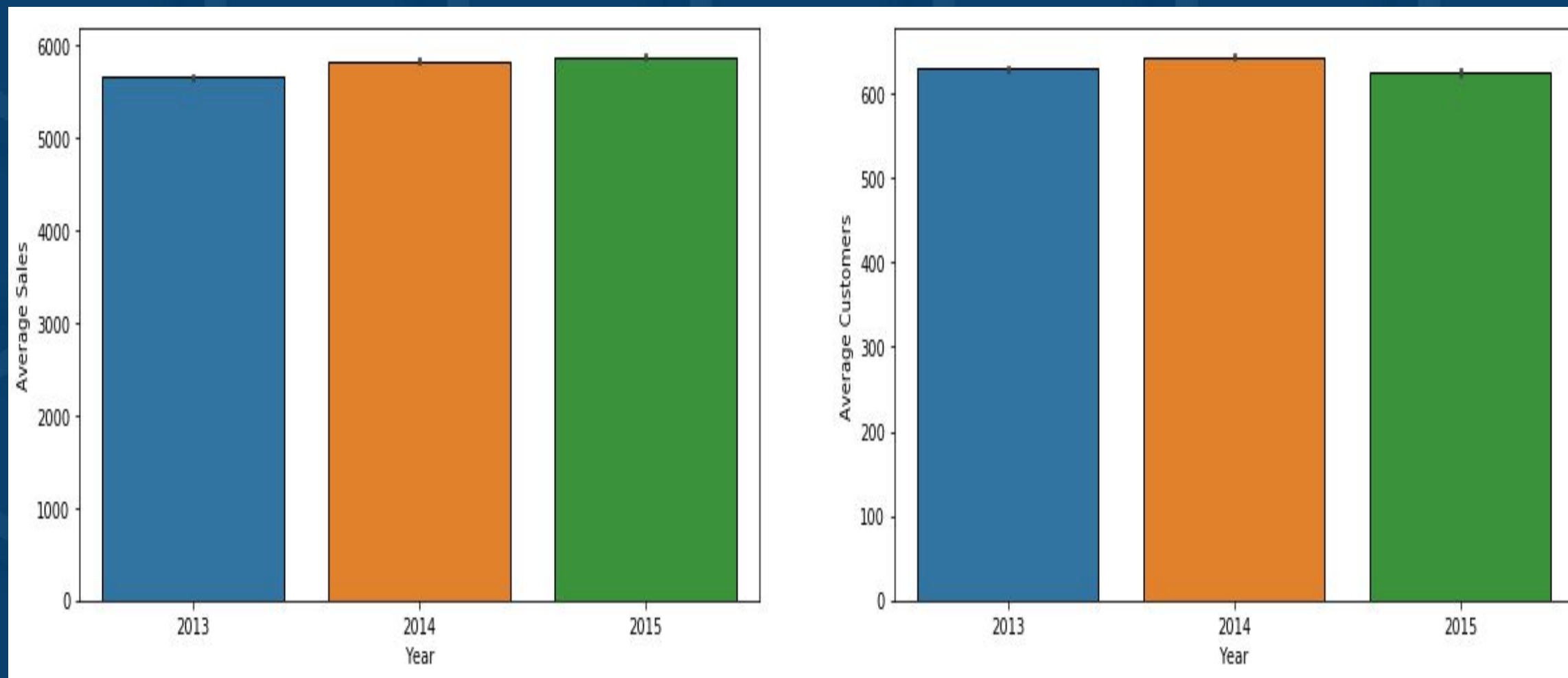
□ Haftalık, aylık ve yıllık satış miktarları

- Her mağazanın haftalık, aylık ve yıllık satış rakamları belirlenerek özellik olarak veri kümesine eklenmiştir.

1	WeekAvg,MonthAvg,YearAvg		
2	5235,4491,4527		
3	5235,4491,4527		
4	5235,4491,4527		
5	5235,4491,4527		
6	5235,4491,4527		
7	3876,4491,4527		
8	3876,4491,4527		
9	3876,4491,4527		
10	3876,4491,4527		
11	3876,4491,4527		

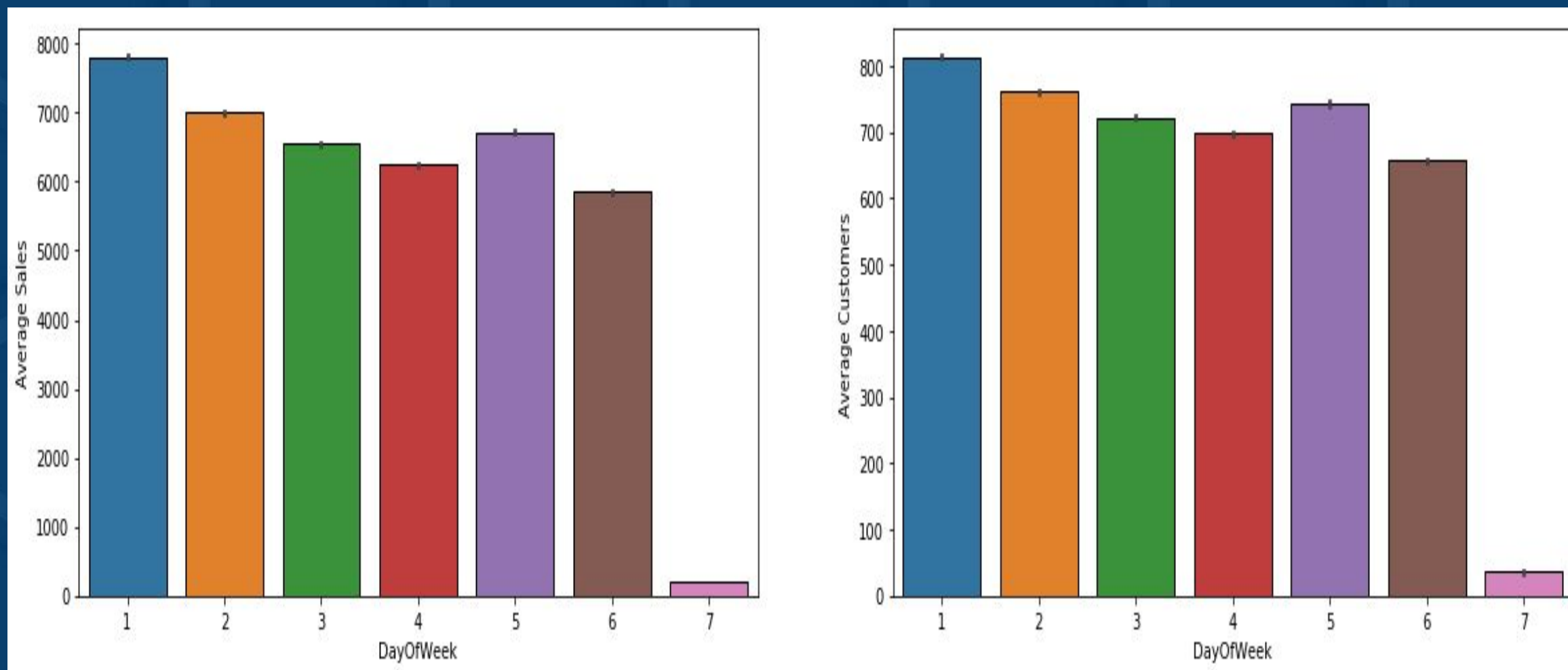
❖ Veri Analizi

□ Ortalama müşteri sayısı & Ortalama satış



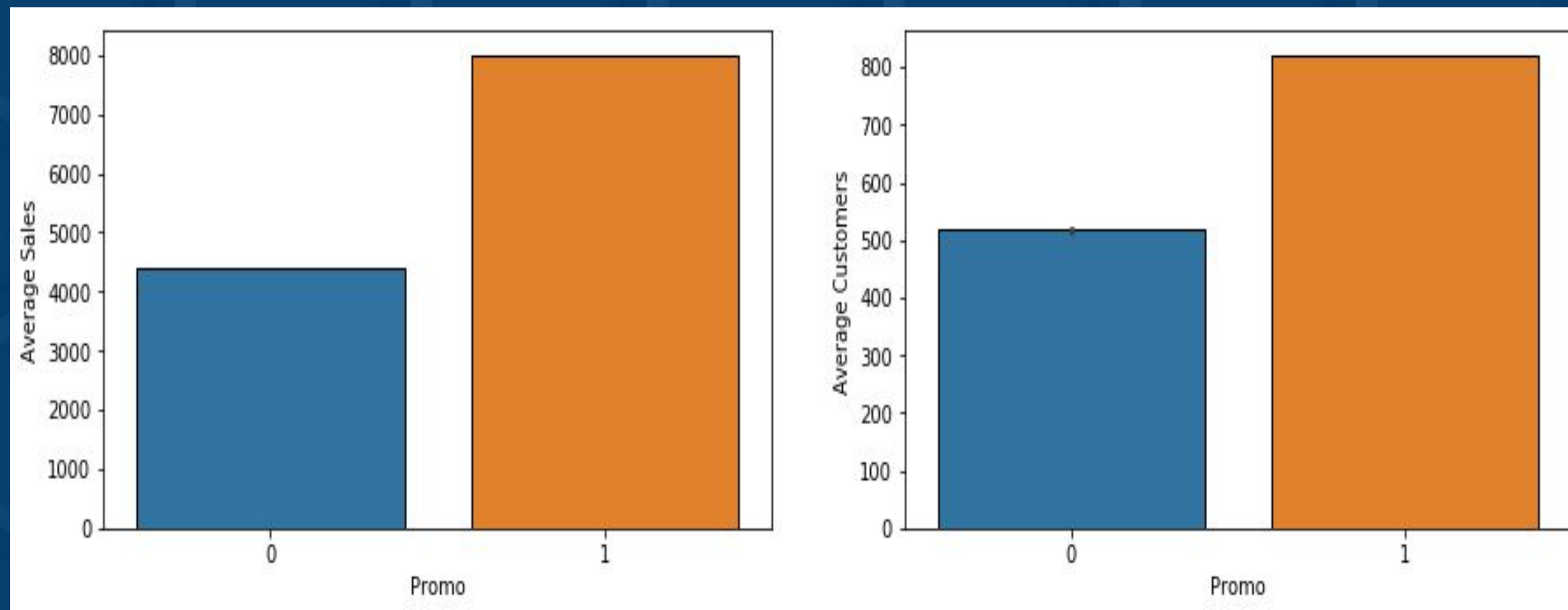
❖ Veri Analizi

□ "DayOfWeek" özelliğinin analizi



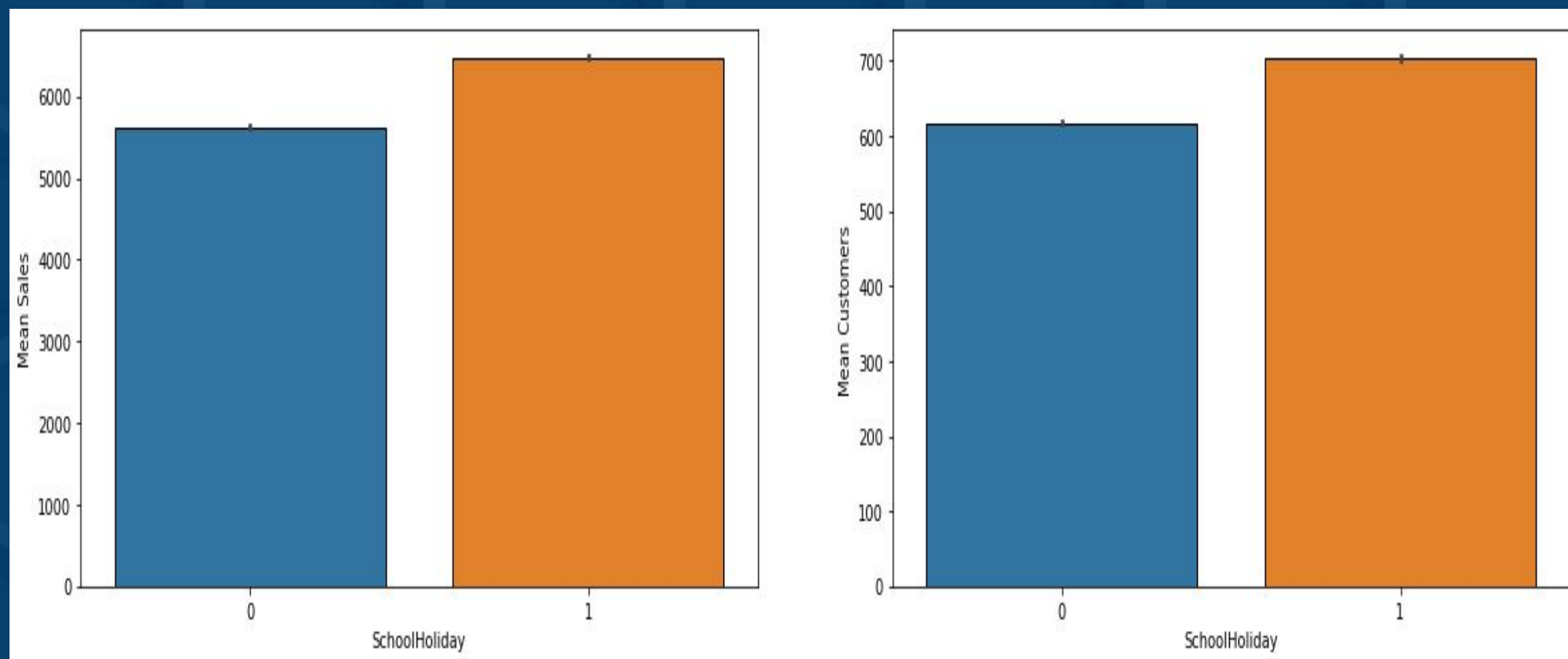
❖ Veri Analizi

□ “Promo” özelliğinin analizi



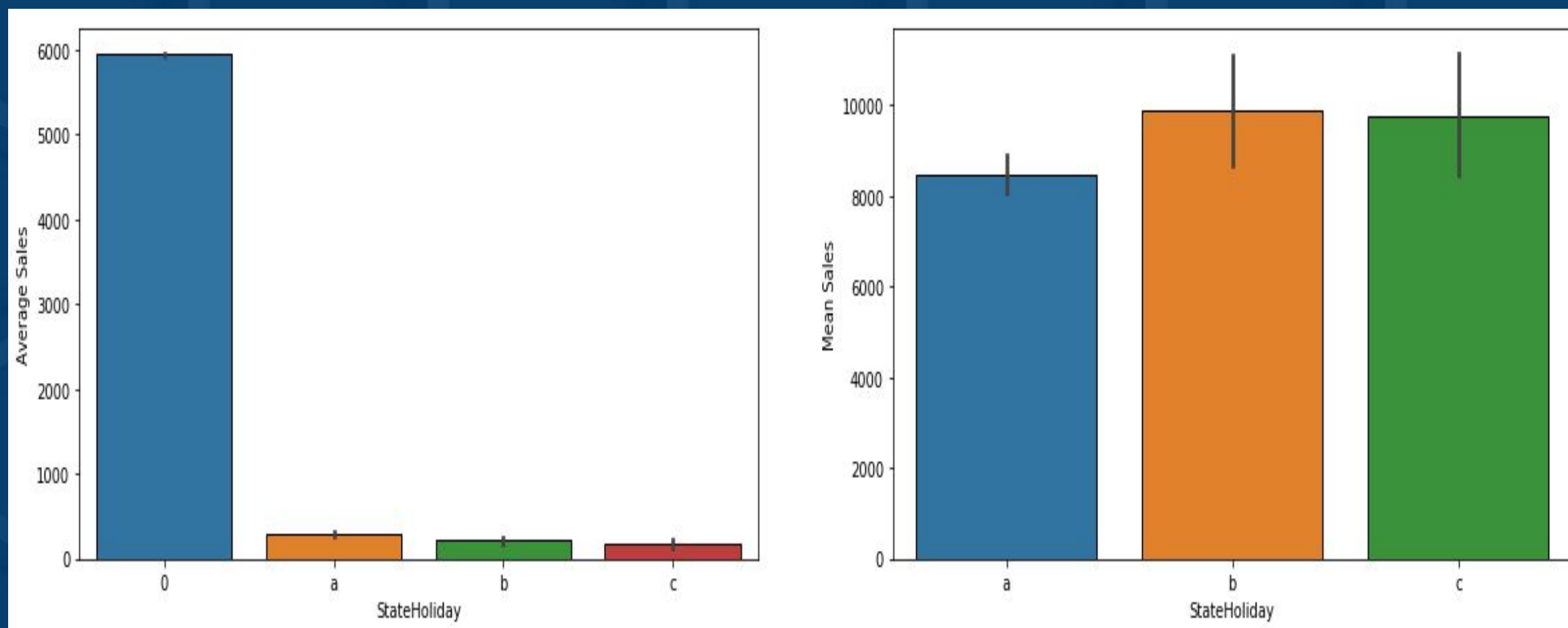
❖ Veri Analizi

□ “SchoolHoliday” özelliğinin analizi



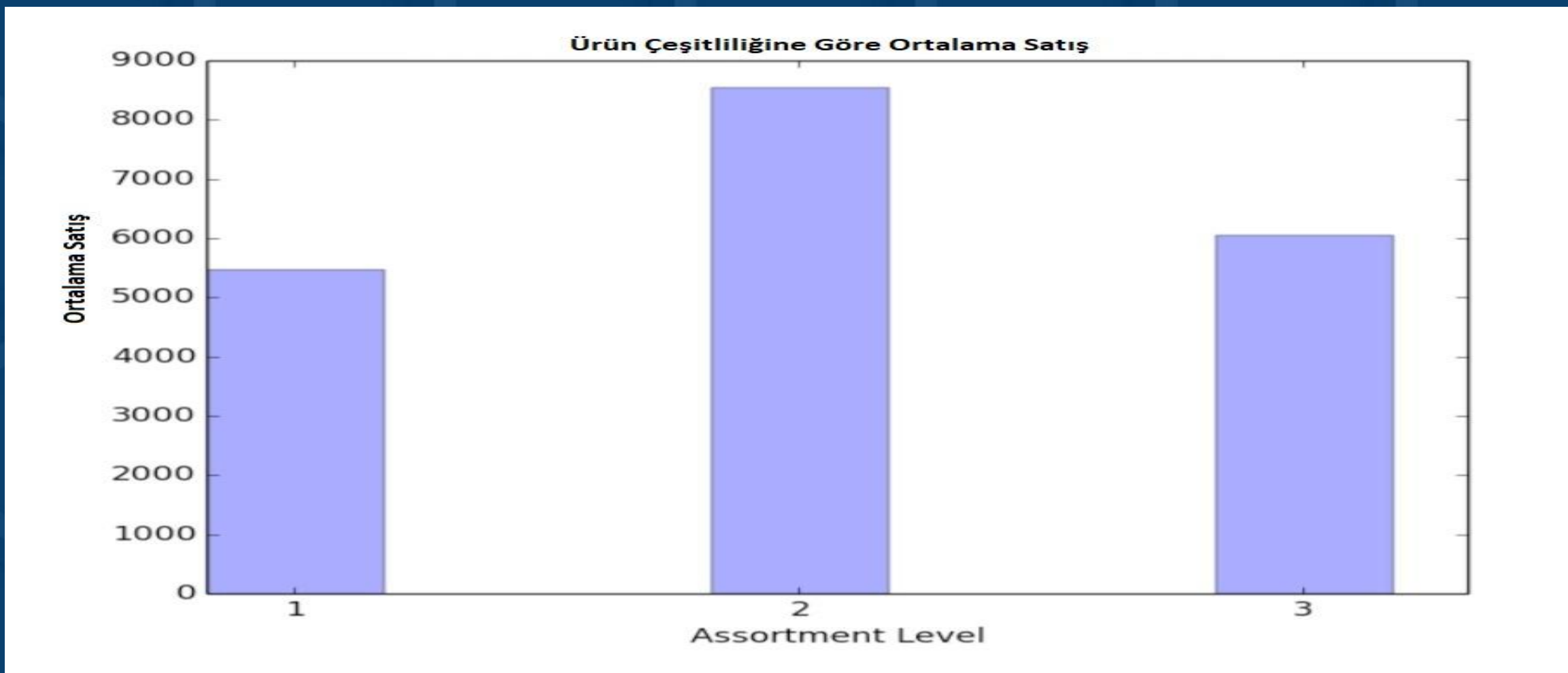
❖ Veri Analizi

□ “StateHoliday” özelliğinin analizi



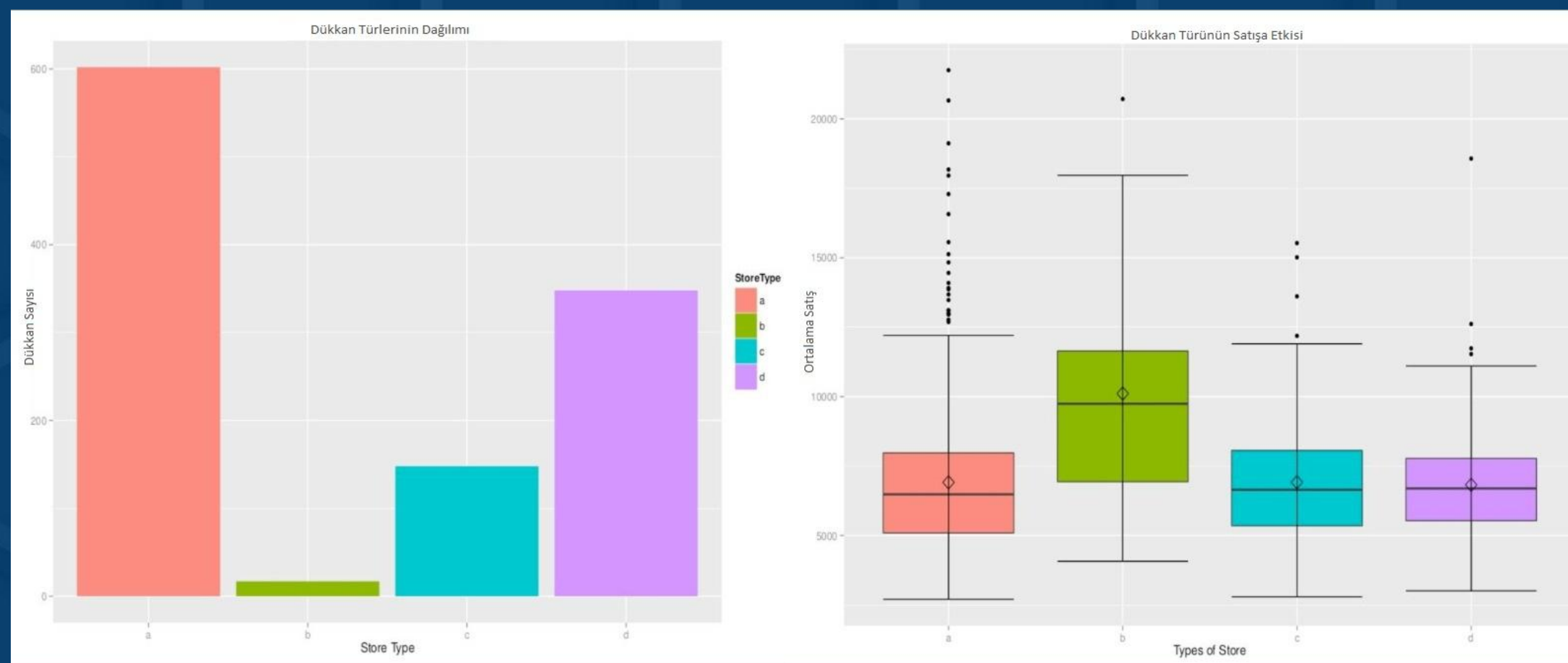
❖ Veri Analizi

□ “Assortment” özelliğinin analizi



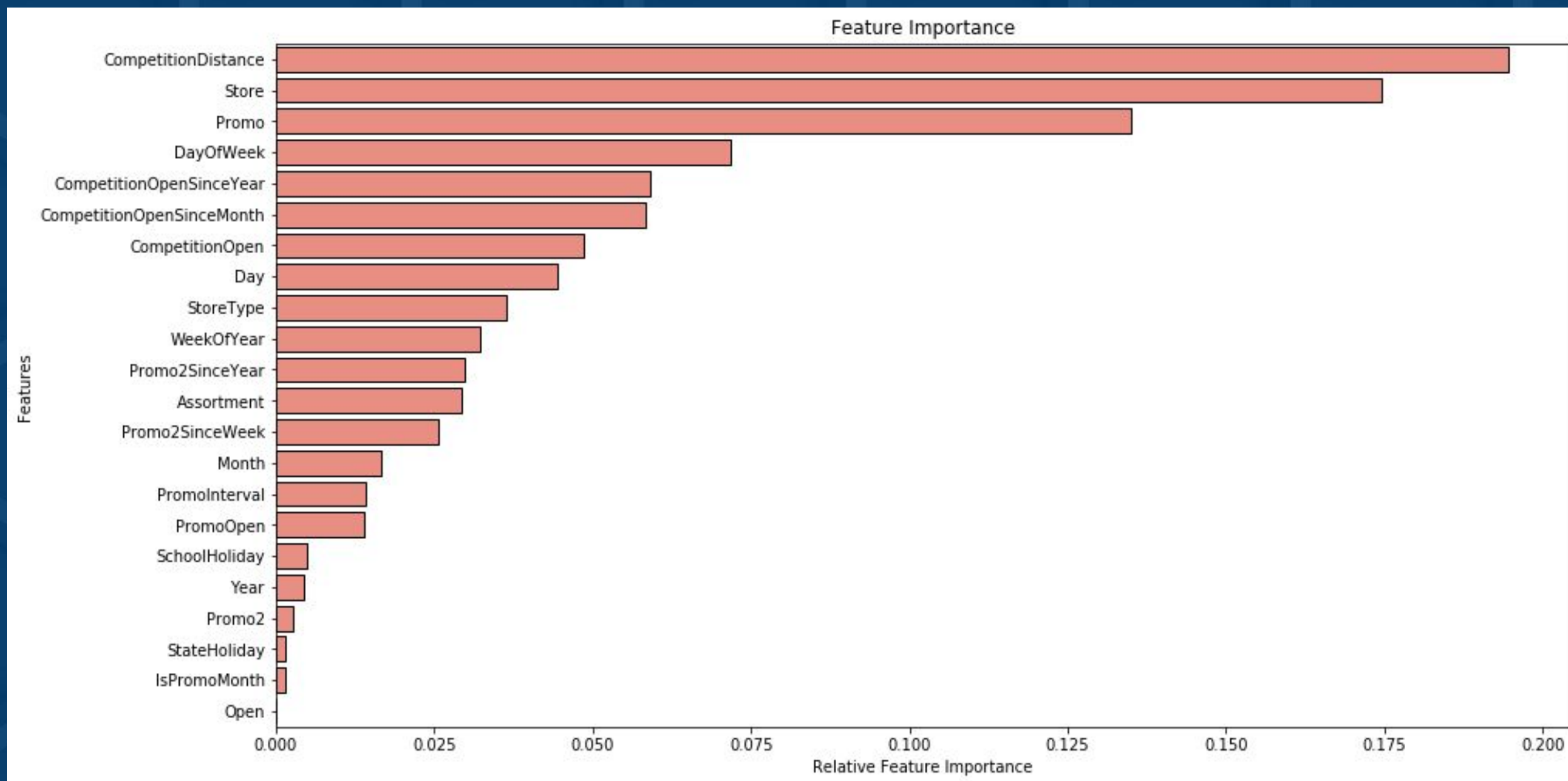
❖ Veri Analizi

□ “StoreType”



❖ Veri Analizi

□ Özelliklerin önemi tablosu



❖ Modellerin Uygulanması

□ Random Forest Regresyonu

- Random Forest algoritması regresyon esnasında birden fazla karar ağacı üreterek regresyon değerini yükseltmeyi hedefleyen bir algoritmadır.
- Bireysel olarak oluşturulan karar ağaçları bir araya gelerek karar ormanı oluşturur.
- Buradaki karar ağaçları bağlı olduğu veri kümesinden rastgele seçilmiş birer alt kümedir.

❖ Modellerin Uygulanması

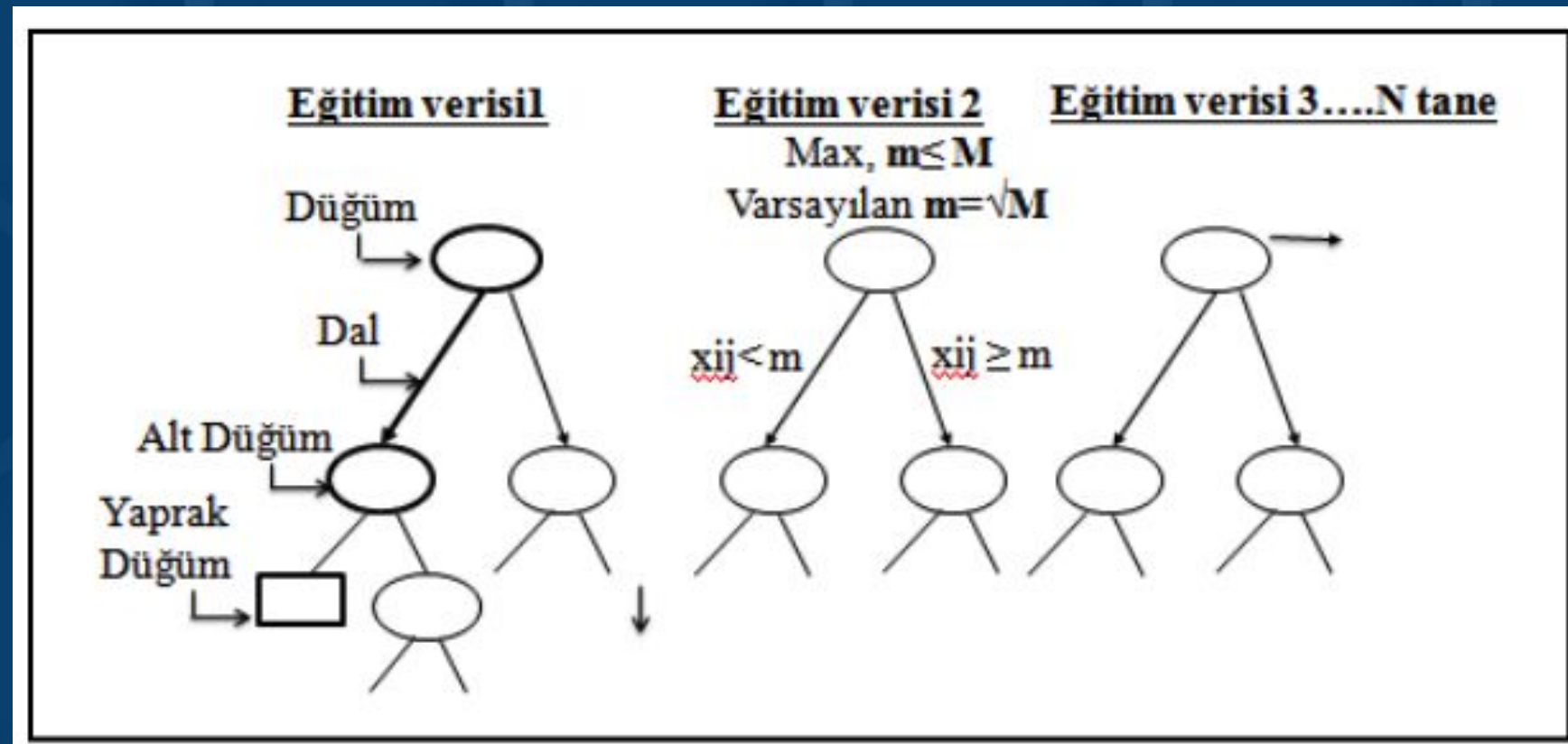
□ Random Forest Regresyonu

- Birçok alanda uygulanabilir. Oldukça hızlı ve doğru sonuçlar verir.
- İyi sonuçlar vermesinin nedeni oldukça büyük ağaçlar oluşturmastır.
- Mümkün olduğunca birbirinden farklı ağaçlar oluşturarak düşük korelasyon yapısında topluluklar oluşturur.
- Bagging, birçok bağımsız belirleyici/model/öğrenici inşa ettiğimiz ve bazı model ortalama teknikleri kullanarak bunları birleştiren basit bir toplama tekniğidir.

❖ Modellerin Uygulanması

□ Random Forest Regresyonu

- Birbirinden farklı olarak kurulan sınıflama ve regresyon ağaçları sonuca giden karar ormanını oluşturur. Karar ormanı oluşumu sırasında elde edilen sonuçlar bir araya gelerek en son tahmin yapılır.



❖ Modellerin Uygulanması

□ Random Forest Regresyonu

Ağaçlar oluştuktan sonra;

- Test özelliklerini alınır ve sonuçları tahmin etmek ve tahmin edilen sonucu saklamak için rastgele oluşturulmuş karar ağacının kurallarını kullanılır.
- Tahmin edilen her hedef için oylar hesaplanır.
- Rastgele Orman algoritmasından son tahmin olarak yüksek oy olan tahmin seçilir.

❖ Modellerin Uygulanması

□ Random Forest Regresyonu

Avantajları:

- Hem sınıflandırma hem regresyon görevlerinde kullanılıyor.
- Aşırı uyum bu tür projeler için en büyük sorunlardandır. Ama Random Forest' te yeteri kadar ağaç varsa bu sorun ortadan kalkar.
- Kategorik değerler için modellenenebilir.

❖ Modellerin Uygulanması

□ XGBoost Regresyonu

- Random Forest algoritmasındaki bagging yönteminde tahminler bağımsız olarak ele alınır.
- Bu modelde ele aldığımız boosting ise tahminlerin bagging yönteminin aksine sırayla yapılmaktadır.
- Bu nedenle bir gözlemin sonraki modellerde de görülme olasılığı eşit değildir.
- En yüksek hataya sahip olanlar en çok görünenlerdir. Bu yüzden gözlemler hata oranına göre seçilmektedir.

❖ Modellerin Uygulanması

□ XGBoost Regresyonu

- XGBoost algoritması sınıflandırma ve regresyon için oluşturulmuş bir makine öğrenmesi tekniğidir.
- XGBoost zayıf tahmin modellerinin biraraya gelmesiyle karar ağaçlarının oluşturmuş olduğu bir modeldir.

❖ Modellerin Uygulanması

□ XGBoost Regresyonu

- Modellerin amacı denetlenebilir bir kayıp fonksiyonu tanımlamaktır.
- Modelin hedefi bu kaybı en aza indirmektedir.
- Model kayıp fonksiyonunu sıfıra yaklaştıracak şekilde tahminlerini güncellemektedir.
- Kayıp = MSE = $\sum (y_i - y_i^p)$
 y_i = i. hedef değer, y_i^p = i. tahmin değeri, $L(y_i, y_i^p)$ kayıp fonksiyonudur.

❖ Sonuç

- Random Forest Regresyonu ve XGBoost Regresyonu sonucunda “Predict” veri kümesi elde edilir.
- Bu küme satış tahminlerinden oluşan kümedir.
- Bu kümedeki veriler test kümesindeki veriler ile kıyaslanılıp hata fonksiyonu hesaplanır.

❖ Sonuç

- Bu projede hata fonksiyonu olarak RMSPE(Root Mean Square Percentage Error) fonksiyonu kullanılmaktadır.

$$\text{RMSPE} = \sqrt{\frac{1}{n} \sum_{i=1}^n \left(\frac{y_i - \hat{y}_i}{y_i} \right)^2}$$

❖ Sonuç

- Random Forest Regresyonu sonucunda bir tahmin kümesi oluşturulmuştur.
- Bu tahmin kümesi ile test kümesinin verileri kullanılarak RMSPE hesaplanmıştır.
- Bu değer 0.1468 olarak bulunmuştur.

0	
0	3654.3
1	10913.9
2	6104
3	5111.9
4	5221.9
5	13527.5
6	7803
7	4514.8
8	9721.6
9	7747.9

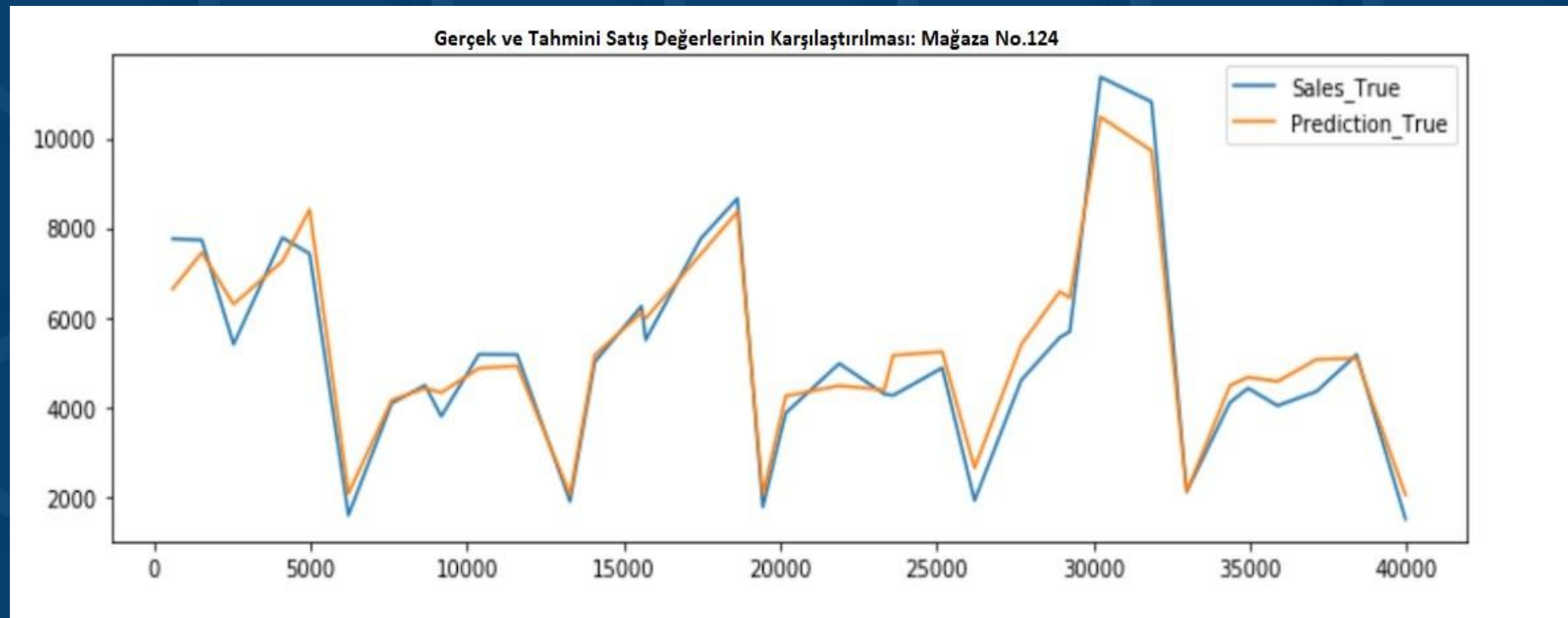
❖ Sonuç

- ❑ XGBoost regresyonu sonucu için de aynı şekilde RMSPE hesaplanmıştır.
- ❑ 0. değer $e^{8.2072}$ yani 3667.25, 1. değer 4755.84 olarak hesaplanmıştır.
- ❑ RMSPE 0.172421 olarak hesaplanmıştır.

0	
0	8.2072
1	8.46713
2	8.56448
3	9.04917
4	8.52179
5	8.9245
6	8.1747
7	8.94494
8	9.07072
9	8.57303

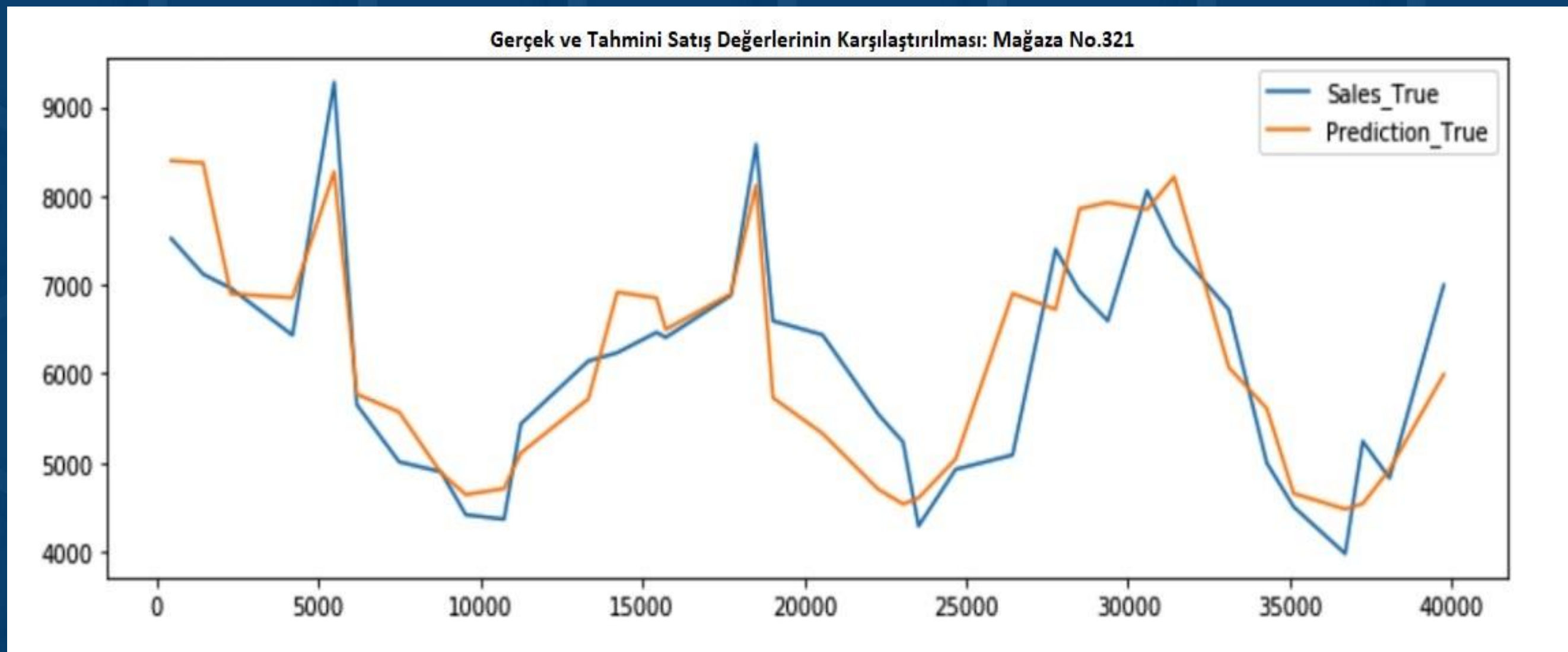
❖ Sonuçların Görselleştirilmesi

- Rastgele 3 mağaza seçilip gerçek satış değerleri ve modelin oluşturduğu tahmin verileri görselleştirilmiştir.
- 124 numaralı mağaza için gerçek satış ve tahmin grafiği



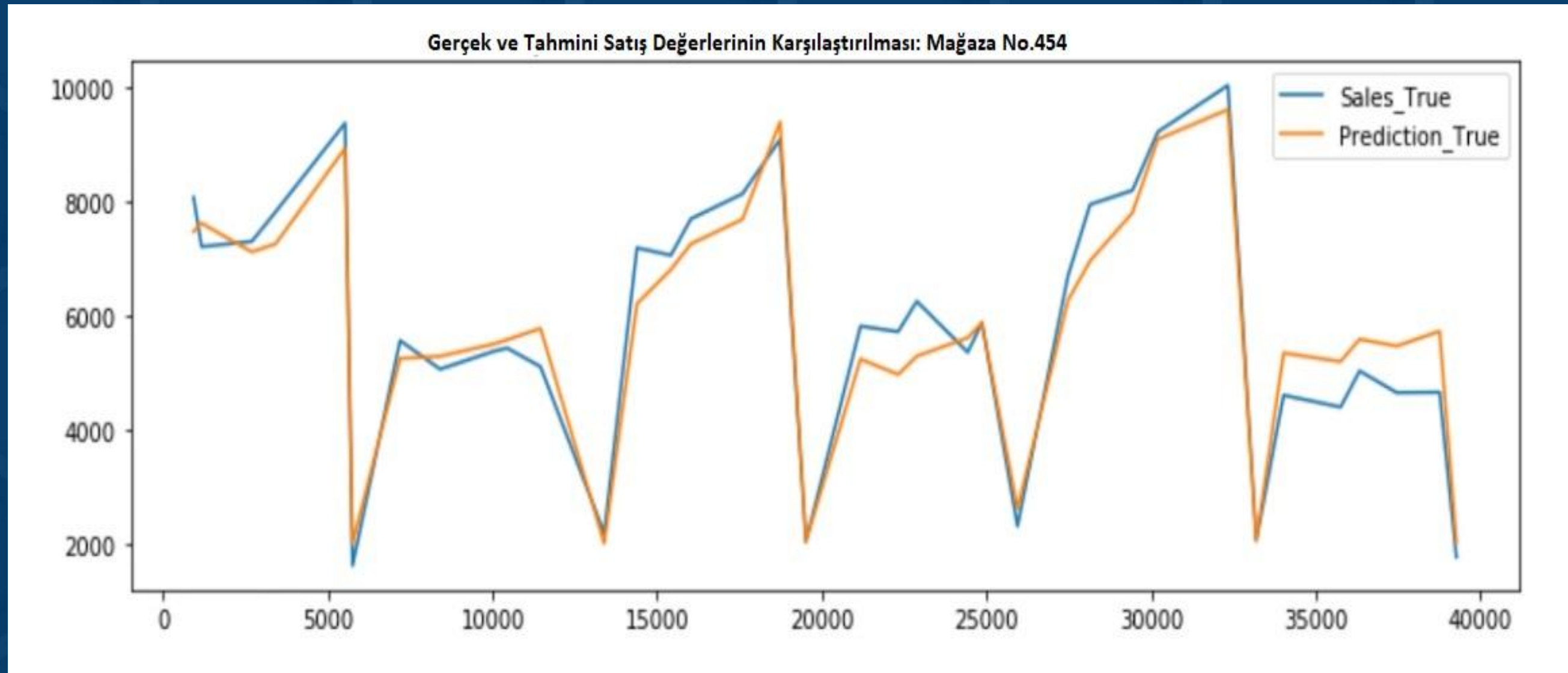
❖ Sonuçların Görselleştirilmesi

- 321 numaralı mağaza için gerçek satış ve tahmin grafiği



















❖ Sonuçların Görselleştirilmesi

- 454 numaralı mağaza için gerçek satış ve tahmin grafiği



❖ Kaggle Sonuçları

- ❑ Kaggle sonuçlarına göre birinci RMSPE' yi 0.10021 olarak bulmuştur.
- ❑ Bizim projemizde ise RMSPE Random Forest Regresyonu için 0.146836, XGBoost regresyonu için 0.172421 olarak bulunmuştur.

1	▲1	Gert		0.10021	19	3y
2	▲1	NimaShahbazi		0.10386	196	3y
3	▲10	Neokami Inc		0.10583	40	3y
4	▲16	Russ W		0.10621	126	3y
5	▲10	MIPT + PZAD		0.10763	195	3y
6	▲96	João N. Laia		0.10771	14	3y
7	▼6	SDNT	   	0.10784	289	3y
8	▲47	Evdilos_Ikaria		0.10817	239	3y
9	▲42	Too busy to compete	 	0.10826	200	3y
10	▲12	NaiveLearners	  	0.10839	367	3y

Dinlediğiniz için teşekkür ederim.

Harun Çatal