YBS Ansiklopedi

www.YBSAnsiklopedi.com

Cilt 8, Sayı 1,Haziran 2020

Müşteri Kayıp Tahmini (Customer Churn Prediction)

Fahrulnisa Sema ÖZBAŞ

*Antalya Bilim University, Computer Engineering, Turkey

Özet

Bu makale müşteri kaybının ne olduğu, müşteri kaybı tahmininin elde edilmesi sırasında OptiScorer motorunun nasıl kullanılacağı ve bu tahmin verisinin işlenmesi ile ilgilidir. OptiScorer, makine öğrenmesi metotlarını kullanarak veri kümelerini öğrenen ve tahmin geliştiren bir yapay zeka motorudur. Üzerinde çalışılan verinin okunması ve performans testi Python kodlama ortamında ayrıca düğümler arasında ilişkilendirmeler yapılarak verinin işlenmesi, yorumlanması, görselleştirmesi, raporlanmasını sağlayan bir veri analiz platformu olan Knime'da yapılmıştır. Yazının bir diğer amacı OptiScorer motorunda bulunan otomatik makine öğrenmesi adımlarının öncesinde ve sonrasında, Python ve Knime'da yapılan adımları da göstermektir.

***Anahtar Kelimeler**:Müşteri Kaybı Tahmini,Python,Knime, Makine Öğrenmesi*

Summary

In this paper, the concept of customer churn analysis and how to use the OptiScorer engine during the prediction of the customer churn is going to be explained. During the data loading, performance tests were performed with Python coding environment together with the Knime, which is a data analysis platform that provides data processing, interpretation, visualization and reporting by making connections between nodes. Another purpose of the article is to show how the automated machinelearning steps in OptiScorer is connected to the problem, before and after the engine via Python and Knime.

***Keywords**:Customer Churn Prediction,Python,Knime,Machine Learning*

1. Giriş ve Tanım

Müşteri kaybı tahmininin nerelerde kullanıldığını anlamak için, ne olduğunu bilmek gerekir. Müşteri kaybı veya literatürdeki diğer bir adıyla müşteri yıpranması(customer attrition) müşterilerin bir markayı terk etme, müşterisi olmayı bırakma eğilimidir [1]. Örnek olarak, bir kullanıcı e-ticaret sitesini bir aydır ziyaret etmediğinde muhtemelen işletme tarafından kaybedilmiştir. Şirketlerin yeni müşteri kazanma maliyeti eldeki müşteriyi tutma maliyetinden daha yüksektir [2]. Bu durum banka, sigorta ve telekomünikasyon şirketleri gibi müşterileri kolayca kaybetmesi muhtemel şirketlerin müşteri kayıp tahmini ve analizi yapmasını ve stratejik adımları buna göre atmasını gerekli kılmaktadır. Bu stratejik adımlara kaybedilme ihtimali yüksek müşterileri tespit etmek ve bu müşterinin sadakatini arttıracak müşteriye özel olarak kampanyalar sunma örnek olarak verilebilir. Bu projede, telekomünikasyon sektöründe müşteri kayıp

tahmini OptiScorer motoru ile gerçekleştirilip, alınan sonuçlar performans testine tabi tutulmuş, seçilen veri üzerindeki başarı oranı hesaplanarak sunulmuştur.

2. Müşteri Kayıp Tahmininin Kullanım Yerleri ve Amaçları

2.1. Telekomünikasyon Sektöründe Müşteri Kayıp Tahmini

Telekomünikasyon sektörü çağrı, e-posta, mesaj gibi hizmetler sunar. Teknolojinin gelişmesiyle hızla büyümekte ve rekabetçi hale gelmektedir. Müşteri tahmininin telekomünikasyon sektöründe kullanılmasının amacı, firma müşterilerinin, şirketin sunduğu hizmet ve ürünleri terk etmeden önce bu durumun farkına varmak ve müşteri kayıplarını önleyici faaliyetlerde bulunmaktır [3]. Müşterilerin hangi koşullarda ne tür sorunlarla karşılaştığını belirleyerek bu sorunların, müşterilerin firmanın sunduğu hizmetlerden duyduğu memnuniyetsizliği ortadan kaldırarak, rakip firmalardan hizmet alma ihtimallerini azaltmak amaçlanmaktadır [4]. Yapılan araştırmalara göre müşteri kayıp analizinin en çok kullanıldığı sektör telekomünikasyondur [5]. Bu projede de telekomünikasyon müşteri kaybı verisi üzerinde çalışılmıştır.

2.2. Reklam Sektöründe Müşteri Kayıp Tahmini

Reklam sektöründe hedef kitleyi belirlemek ve analiz etmek önemlidir. Hedef kitlenin kadın, erkek veya çocuk olması reklamın içeriğini ve tasarımını etkilerken, kullanılan yapay zekâ ile belirli bir yaş grubunun veya cinsiyetin hangi reklam başlığına tıklayacağı tahmin edilebilir. Yapılan tahminler sonucunda da oluşturulacak olan reklam içeriğine yön verilebilir [6].

3.OptiScorer Kullanımı ve Performans Değerlendirmesi

Projeye ilk olarak veri bulunarak başlanmıştır, ardından veriler düzenlenerek OptiScorer motoruna yüklenmiş ve alınan sonuçlar üzerinde Python ve Knime kullanılarak değerlendirmeler yapılmıştır. Makalede her biri açıklanacaktır.

3.1. Veriyi anlama ve hazırlama

Bu projede Kaggle'dan alınan Telco Customer Churn isimli veri kümesi kullanılmıştır. Veri kümesine https://www.kaggle.com/blastchar/telco-customer-churn adresinden ulaşılabilir. Her satır bir müşteriyi temsil eder, her sütun müşterinin özelliklerini içerir. Bu veri kümesi customerID, gender, Phone Service, Internet Service gibi bilgiler içeren 21 sütun (değişken) ve 7043 satır (müşteri) içerir. Tablo 1'de 17 si kategorik ve 4ü sayısal veri içeren değişkenlerin veri tipleri gösterilmektedir.

Tablo 1 Değişkenler ve veri tipleri

Değişken	Veri Tipi
CustomerID	Nominal
gender	Nominal
SeniorCitizen	Nümerik
Partner	İkili
Dependents	İkili
Tenure	Nümerik
PhoneService	İkili
MultipleLines	Nominal
InternetService	Nominal
OnlineSecurity	Nominal
OnlineBackup	Nominal
DeviceProtection	Nominal
TechSupport	Nominal
StreamingTV	Nominal
StreamingMovies	Nominal
Contract	Nominal
PaperlessBilling	İkili
PaymentMethod	Nominal
MonthlyCharges	Nümerik
TotalCharges	Nümerik
Churn	İkili

OptiScorer kullanımı için ilk sütunda ID gibi benzersiz değerler ve son sütunda eğitim sonucunda test edilecek değerler yerleştirilmelidir ve bunlar nümerik olmalıdır.

3.2. OptiScorer

Seçilen veri kümesi olan Telco Customer Churn'de ilk sütun benzersiz değerlerden oluşmakta ve test edilecek değer olan Churn değişkeni son sütunda bulunmaktadır ancak son sütundaki değerler OptiScorer kullanımı için sayısal değerlere dönüştürülmelidir. Bu dönüşüm işlemi Python'da bulunan Pandas kütüphanesi kullanılarak yapılmıştır [7].

```
import pandas as pd
data = pd.read_csv("WA_Fn-UseC_-Telco-Customer-Churn (1).csv")
df = pd.DataFrame(data)
pd.DataFrame(data)["Churn"].replace(('Yes', 'No'), (1, 0), inplace=True)
```

Yukarıdaki veri yükleme ve sayısal veriye dönüştürme aşmalarının ardından, veri kümesi, makine öğrenme adımlarında herhangi bir ezberleme olma ihtimaline karşı, eğitim ve test kümesi olarak %70'e %30 oranında ayırılır,

```
train=df.sample(frac=0.7,random_state=200)
test=df.drop(train.index)
```

Test verisi olarak ayırılan veri kümesindeki son sütun çıkarılmalıdır, bu adım makine öğrenmesi tahminlerinin test edilmesi amacıyla yapılmaktadır. Bununla birlikte Test Etiketlerinde test değerlerinin orijinal hali, tahmin değerlerinin doğruluğunu kontrol etmek için tutulmalıdır.

```
test_label = test
```

```
test= test.iloc[:, :-1]
```

Eğitim, Test ve Test Etiketleri Excel dosyaları oluşturulur.

```
test_label.to_csv("test_label.csv", index=False)
test.to_csv("test.csv", index=False)
train.to_csv("train.csv", index=False)
```

Ardından Eğitim ve Test dosyaları OptiScorer'a yüklenir. Otomatik makine öğrenmesi metotlarını kullanarak veri kümelerini öğrenen ve tahmin geliştiren bir yapay zekâ yazılımı olan OptiScorer bu tahmin değerlerini içeren bir dosya oluşturur. OptiScorer herhangi bir özel ayara gerek olmaksızın kendi seçtiği bir algoritmayı kullanır ancak kullanıcı dilerse farklı algoritmaları seçerek OptiScorer'ı çalıştırabilir. Projede, incelenmek üzere OptiScorer yazımı tarafından sunulan tüm algoritmalar denenmiştir. Aşağıda Eğitim ve Test verilerinin yüklenmesi aşamasının ekran görüntüsü paylaşılmıştır.

Şekil 1 Eğitim ve Test dosyalarının sisteme yüklenmesi

3.3. Makine Öğrenmesi Algoritma Skorları

Projenin bu kısmında farklı algoritmaların seçilen veri kümesi üzerindeki performanslarına bakılacaktır. İlk olarak Test Etiketi dosyası ve sistemden alınan tahmin verisi okunur ve ardından performanslarına bakılır.

```
import pandas as pd
from sklearn.metrics import r2_score
def ML_Scores(csv_file):
  test2 = pd.read_csv('test_label.csv', index_col='customerID')
  test3 = pd.read_csv(csv_file, index_col='idx')  # from optiscore
  y_pred = test3['score']
  realValue = test2['Churn']
```

Skorlama için hata ölçüm yöntemlerinden olan r2_score fonksiyonu kullanılmıştır. r2 sistematik hata ile yüklü bir istatistik olup, sistematik hata düzeyi, yükseldikçe veya örnek büyüklüğü arttıkça azalmaktadır [8]. Tablo 2, OptiScorer'dan alınan tahmin verilerinin r2_score fonksiyonu uygulanarak elde edilen hata düzeylerini göstermektedir.

```
score = r2_score(realValue.loc[y_pred.index], y_pred)
```

3.4. Hata Matrisi(Confusion Matrix)

Tablo 2 Algoritma Performans Skorları

Algoritma	Score
Decision Tree Algoritması	-0.3536565116577466
Light GBM Algoritması	0.303531195996138
Rondom Forest Algoritması	0.1874886641255693
Support Vector Algoritması	-0.09692160429168539
KNN Algoritması	-0.09692160429168539
Linear Algoritması	0.28077583130930195
XGBoost Algoritması	0.3253903438259522

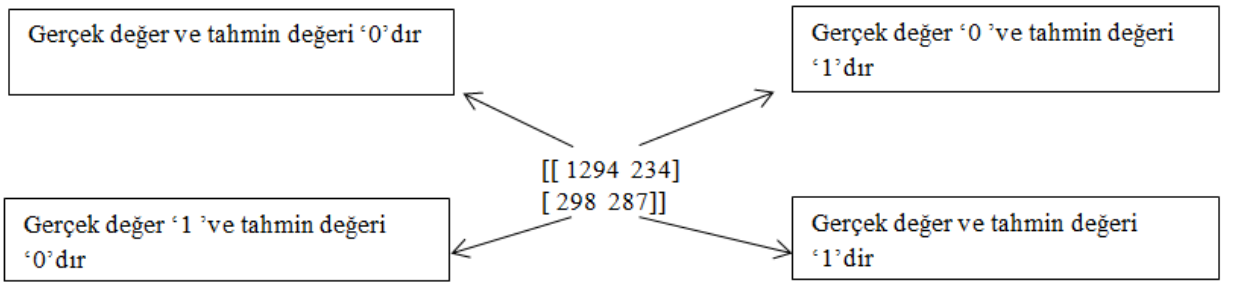
Hata matrisi makine öğrenmesinde kullanılan algoritmaların performansını yani tahminlerin doğruluğunu değerlendirmek için tahmin ve gerçek değerlerin karşılaştırıldığı bir ölçüm aracıdır. Projenin bu kısmında hata matrisi (confusion matrix) ve doğruluk oranı Sklearn kütüphanesi kullanılarak hesaplanmıştır [9].

```
from sklearn.metrics import confusion_matrix
from sklearn.metrics import accuracy_score
```

Hata matrisinde tahmin edilen değerler gerçek değerlerle tam olarak eşleşmelidir. Bundan dolayı 0,5'den büyük değerler 1 küçük değerler ise 0 olacak şekilde eşik belirlenir.

```
for i in df["score"]:
  if i >= 0.5:
      df.replace(to_replace=i, value=1, inplace=True)
  else:
      df.replace(to_replace=i, value=0, inplace=True)
```

Tablo 3'ün daha iyi anlaşılması adına Şekil 2'de hata matrisindeki satırlar ve Formül 1'de doğruluk skoru aşağıda açıklanmıştır. [10].



Şekil 2 Hata Matrisi

3.4.1. Doğruluk(Accurancy)

$$\text{Doğruluk(Accurancy)} = \frac{\#doğru\ sınıflandırılan\ değerler}{\#\ tüm\ sınıflandırılan\ değerler}$$

Formül 1 Doğruluk hesaplanması

Doğruluk, etiketleri doğru tahmin edilen test gözlemlerinin toplam test gözlemlerinin sayısına bölünmesidir. Bu oran, eğitilen algoritmanın gerçek hayattaki kullanımı sırasında yapacağı tahminlerin ne kadar doğru olacağını belirtir [11].

Tablo 3 Hata matrisi ve doğruluk skoru

Algoritma	**Hata Matrisi**	**Doğruluk Skoru**
Decision Tree Algoritması	[[1294 234] [298 287]]	0.7482252721249408
Light GBM Algoritması	[[1438 236] [154 285]]	0.8154283009938476
Rondom Forest Algoritması	[[1336 228] [256 293]]	0.7709417889256981
Support Vector Algoritması	[[1417 438] [175 83]]	0.709891150023663
KNN Algoritması	[[1359 435] [233 86]]	0.6838618078561287
Linear Algoritması	[[1438 240] [154 281]]	0.8135352579271179
XGBoost Algoritması	[[1458 256] [134 265]]	0.8154283009938476

Tablo 3 makine öğrenmesi algoritmalarının yukarıda hesaplama yöntemleri açıklanan hata matrisi ve doğruluk değerleridir.

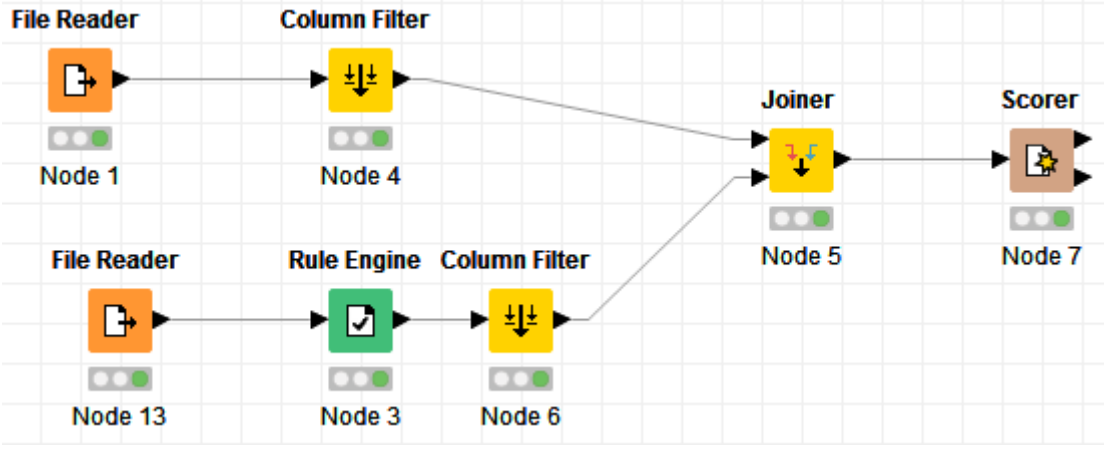
Tablo 4'teki Precision doğru olduğu bilinen gözlemlerin doğru olarak tahmin edilmişlerinin bütün doğru olarak tahmin edilmişlere oranıdır. Recall ya da Sensitivity, sınıflar içerisinde doğru olduğu bilinen gözlemlerin doğru olarak tahmin edilenlerinin bütün doğru olduğu bilinen gözlemlere oranıdır. F score sınıflandırma algoritmalarını karşılaştırırken çok sık kullandığımız bir ölçüdür. Specificity, sınıflar içerisinde yanlış olduğu bilinen gözlemlerin yanlış olarak tahmin edilenlerinin bütün yanlış olduğu bilinen gözlemlere oranıdır [12].

Tablo 4 Sınıflandırma Raporu

	precision	recall	f1-score	support
0.0	0.81	0.85	0.83	1528
1.0	0.55	0.49	0.52	585
accuracy			0.75	2113
macro avg	0.68	0.67	0.67	2113
weighted avg	0.74	0.75	0.74	2113

3.5. Knime ile Hata Matrisi

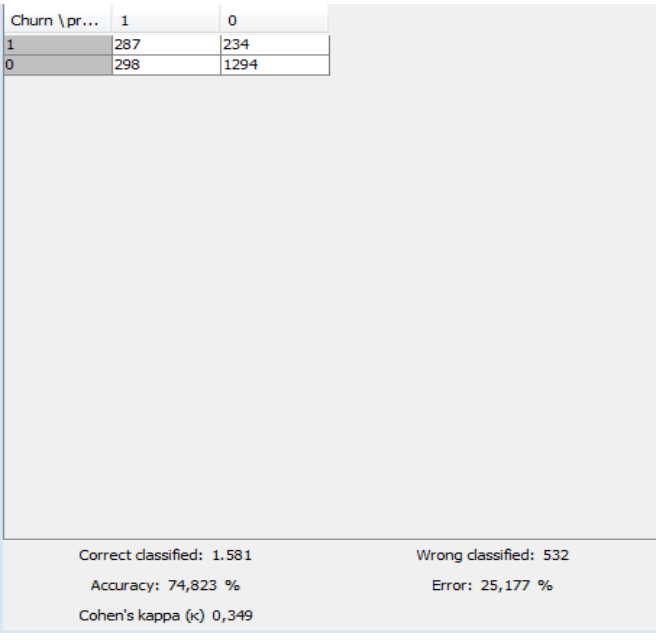
Knime'da aşağıdaki Şekil 3'deki operatörler kullanılarak hata matrisi değerine ulaşılabilir.



Şekil 3 Knime'da Modelleme

Ayırılan %30'luk test verisi ve OptiScorer'dan indirilen tahmin verileri, Şekil 3'de gösterilen File Reader operatörlerine yüklenir, Rule Engine operatörü ile 0,5'den büyük değerler '1' küçük değerler ' 0' olarak belirlenir. Joiner operatörü

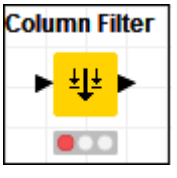
ile aynı ID'ler eşlenir. Scorer operatörü ile hata matrisine ulaşılır. Buradaki sonuçlar Python'da hesaplanan hata matrisleriyle aynıdır. Şekil 4'te Knime'dan Decision Tree algoritmasının hata matrisi ekran görüntüsü gösterilmiştir.



Şekil 4 Decision Tree algoritması hata matrisi ve doğruluk skoru

3.6. Ek Bilgiler

Veride gereksiz sütunlar varsa çıkarılırsa tahmin başarısı arttırılabilir. Knime'da aşağıda Şekil 5'te gösterilen Column Filter kullanılarak veri kümesindeki müşterilerin faturalarını hangi yolla ödediği bilgilerinin yer aldığı PaymentMethod sütunu çıkarılmış ve sonucun değişmediği görülmüştür. Buradan müşteri kayıplarının ödeme yöntemleriyle ilgisi olmadığı sonucu çıkarılabilir.



Şekil 5 Column Filter

4. Sonuç

Projede OptiScorer'da veri kümesinin eğitilmesinin ardından farklı algoritmalar(K-NearestNeighbors, Linear ,XGBoost, LightGBM, Random, Forest, Support Vector, Decision Tree) kullanılarak elde edilen tahmin verileri baz alınarak başarı oranı en yüksek olan algoritma saptanmıştır. Bu saptama Python kodlama ortamında ve Knime analiz platformunda yapılmış olup regresyon analizi ve hata matrisi ile yapılmıştır. Elde edilen sonuçlara göre en uygun algoritmanın %82 doğruluk oranıyla LightGBM olduğu anlaşılmıştır. Verilerdeki bazı değişkenlerin gerekli olmayabileceği ve veri kümesinden çıkarılmalarının tahmin sonuçlarını etkilemeyeceği gösterilmiştir.

Çalışmanın şirketlerin müşteri ilişkileri yönetimi, pazarlama ve reklam birimlerinde değerlendirilmesi tavsiye edilmektedir. Makale telekomünikasyon sektörü haricinde üyelik sistemi ile çalışan diğer sektörlerde de müşteri kayıplarının önceden tahmini ile müşteri kaybının azaltılmasını sağlayacaktır.

Çalışmaya devam edilmesi durumunda müşteri kayıp tahmini mantığının, insanların genetik özelliklerini analiz etme ve hastalık riski oranının tahmin edilmesinde kullanılması planlanmaktadır.

Referanslar

[1] https://www.kdnuggets.com/2019/05/churn-prediction-machine-learning.html
[2]Oğuz Kaynar,Fatih Murat Tuna,Yasin Görmez,Mehmet Ali Deveci, Makine Öğrenmesi Yöntemleriyle Müşteri Kaybı Analizi, 2017
[3] Tuğba Şimşek Gürsoy,Customer churn analysis in telecommunication sector,Cilt/Vol:39, Sayı/No:1, 2010
[4] Ramis Başkal, Telekomünikasyon sektöründe müşteri segmentasyonu ve müşteri kayıp analizi, 2019
[5] Fatma Önay Koçoğlu, Tuncay Özcan, Ş. Alp Baray, Veri madenciliğinde ayrılan müşteri üzerine literatür araştırması, 2016
[6] https://medium.com/@optiwisdom/yapay-zeka-ile-reklam-verece%C4%9Finiz-kitleyi-belirleyin-a1181e774bc9
[7] https://optiwisdom.web.app/home/analysis
[8]Alptekin Günel, Regresyon denkleminin başarısını ölçmede kullanılan belirleme katsayısı, 2003

[9] https://scikit-learn.org/stable/modules/generated/sklearn.metrics.accuracy_score.html
[10] https://www.geeksforgeeks.org/confusion-matrix-machine-learning/
[11] http://www.analytichouse.com/siniflandirma-ve-confusion-matrix/
[12]Çağrı Aksu, Sınıflandırma ve Confusion Matrix, 2019