BANKA MÜŞTERİSİ KAYIP ANALİZİ

Recep Furkan Koçyiğit

furkan.kocyigit@std.yildiz.edu.tr

Bilgisayar Mühendisliği Bölümü

Elektrik Elektronik Fakültesi, Yıldız Teknik Üniversitesi

Özet

Bu ödevde, makine öğrenmesi yöntemleri ile bir bankanın müşterilerinin kalıcı olup olmayacağını tespit etmek için bir sistem tasarlanmıştır. Ödev Python programlama dili kullanılarak yazılmıştır.

İstatiksel veri analizi yöntemleri kullanılarak veri analiz edilmiş, veri ön işlemi yöntemleri kullanılarak veri eğitim için hazır hale getirilmiştir. Tahmin modeli için K En Yakın Komşuluk, Karar Ağacı, Naive Bayes ve Yapay Sinir Ağları kullanılmıştır. Üzerinde çalışılan veri, kendi içerinde dağılım olarak dengesizlik gösterdiği için başarı ölçümü olarak F1 skoru tercih edilmiştir. En başarılı sonuç Karar Ağacı algoritmasından elde edilmiştir.

Giriş

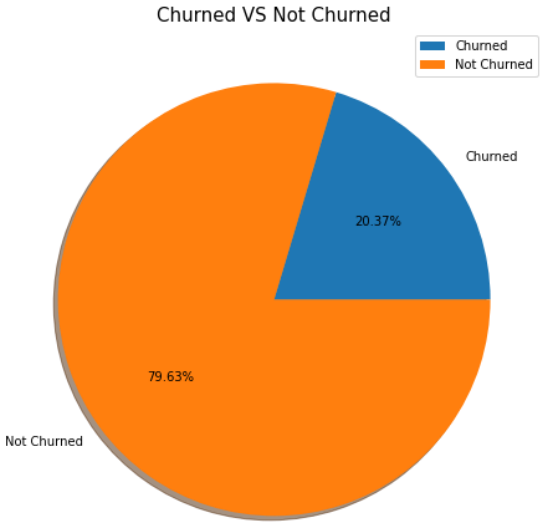
Bu ödevde bir bankanın verileri üzerinden müşteri kayıp analizi yapılmıştır. Müşteri kaybı, bir ürün veya hizmeti belirli bir zaman diliminde kullanmayı bırakan müşterilerin yüzdesini ifade etmektedir. Yeni müşteriler edinmek, mevcut müşterileri elde tutmaktan daha pahalıya mal olabileceğinden müşteri kayıp analizi şirketler için önemlidir.

Veri Kümesi

Ödev kapsamında kullanılan veri https://www.kaggle.com/datasets/gauravtopre/bank-customer-churn-dataset adresinden elde edilmiştir. Bu veride 10.000 adet müşteri bilgisi bulunmaktadır. Ödev kapsamında verilerin 400 tanesi eğitim, 300 tanesi validasyon ve 300 tanesi test için kullanılmıştır. Bu veriler seçilirken veri dağılımı göz önüne alınarak seçilmiştir.

Veri Kümesindeki Özelliklere Ait Bilgiler

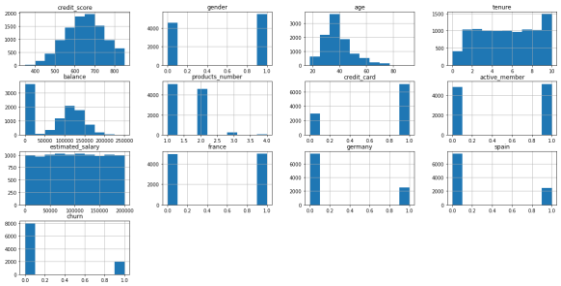
1. customer_id : Müşteri numarası(Integer). Özellik olarak kullanılmamıştır.
2. credit_score : Müşterinin bireysel kredi skoru(Integer).
3. country : Müşterinin yaşadığı ülke(String).
4. gender : Müşterinin cinsiyeti(String)
5. age : Müşterinin yaşı(Integer)
6. tenure : Müşterinin bankada hesabının kaç yıldır olduğu bilgisi(Integer).
7. balance : Müşterinin hesabında bulunan tutar(Float).
8. products_number : Müşterinin sahip olduğu ürün sayısı(Integer).
9. credit_card : Müşterinin kredi kartının olup/olmadığı bilgisi(Integer).
10. active_member : Müşterinin hesabının aktif olup/olmama bilgisi(Integer).
11. estimated_salary : Müşterinin tahmini maaşı(Float).
12. churn : Müşterinin banka müşteriliğinden ayrılıp/ayrılmama bilgisi(Integer).



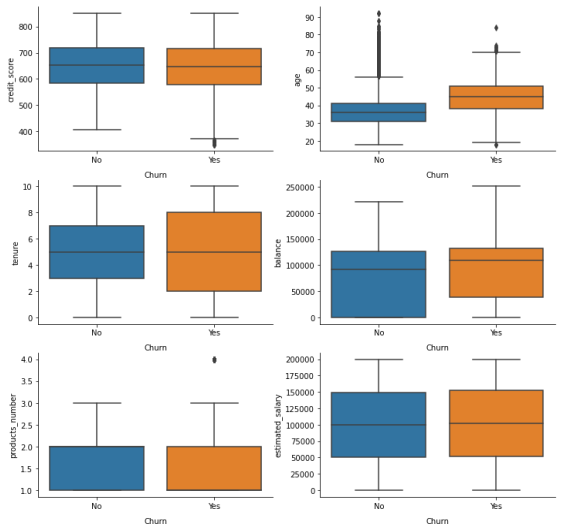
Verinin Dağılımı

Veri üstteki grafikte görülebileceği gibi eşit olarak dağılmamıştır. Eğitim, validasyon ve

test için örnek seçiminde bu dağılım korunmuştur. Kullanılan veriye ait özelliklerin kendi içerisinde dağılımları ise aşağıdaki gibidir:

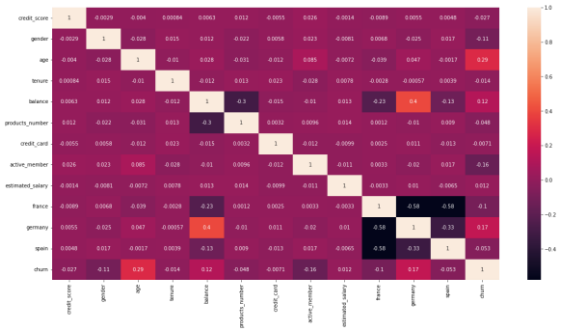


Yapılan Box-Plot analizine ait görüntü aşağıdaki gibidir:



Buradaki grafiklerden görülebileceği üzere ayrılan müşteriler kalanlara göre daha yaşlıdır. Bunun dışında ayrılan müşterilerin kalan müşterilere göre hesabında daha fazla miktarda parası bulunduğu gözlemlenmektedir. Kredi skoru, müşterinin bankada hesabı olma yılı, ürün sayısı ve tahmini maaşa bakıldığında ayrılan veya kalan müşteriler için bir fark gözlemlenmemektedir.

Veriye ait korelasyon matrisi aşağıdaki gibidir:



Korelasyon matrisine bakıldığında kredi skorunun, müşterinin bankada hesabı olma yılının, müşterinin kredi kartının olma bilgisinin ve müşterinin tahmini maaşının ile müşterinin ayrılıp ayrılmama bilgisi arasındaki ilişkinin çok düşük olduğu gözlenmiştir.

Sınıflandırma Modelleri

1. **Naive Bayes:** Bayes Teoremini kullanarak sınıflandırma yapar. Olasılıksal bir yaklaşım kullandığından verinin normalize edilmesine gerek yoktur.
2. **K En Yakın Komşuluk:** Tahminleme yaparken uzaklık kullandığından verinin normalize edilmesi gereklidir. Veri kümesine yeni eleman geldiğinde en yakın grubu bulmaya çalışır. Eğitim süreci yoktur. K değeri, çift bir sayı seçilmesinde test edilecek değerin gruplara eşit uzaklıkta çıkabilmesinden dolayı tek sayı seçilmelidir.
3. **Karar Ağacı:** Bilgi kazanımının en yüksek olduğu özelliği bulma üzerine kurulu bir algoritmadır. Özellik çıkarımına veya normalizasyona gerek yoktur. Önemli özellikler zaten seçilmiş olacaktır.
4. **Yapay Sinir Ağları:** İnsan beyninin çalışma prensibinden esinlenerek geliştirilmiştir. Normalizasyon işlemi gereklidir. Girdi katmanı, gizli katman ve çıktı katmanından oluşur. Girdi katmanından gelen değerler ile ağırlıklarının nokta çarpımının bir aktivasyon fonksiyonundan çıktısı sonucunda bir sonraki katmandaki girdi elde edilmiş olur.

Deneysel Analiz

Veri kümesinde ön işleme olarak “gender” sütunundaki değerler “Female” ve “Male” iken, bunlar 0 ve 1 olarak tutulmuştur. Bunun dışında “country” sütunundaki “France”, “Germany” ve “Spain” değerleri de One-Hot Encoding işlemi uygulanarak ayrı sütunlarda tutulmuştur. “customer_id” sütunu bir önem teşkil etmediğinden veriden çıkarılmıştır.

Naive Bayes algoritmasında en uygun hiper parametreleri seçmek için öncelikle model

eğitim kümesiyle eğitilmiş daha sonra “alpha” parametresi [0, 0.1, 0.01, …, 10^(-10)] arasındaki değerler için validasyon verisiyle “10-Fold Cross Validation” ile sınandığında en başarılı sonucu veren “alpha” değerinin 0.30 olduğu görülmüştür.

Karar ağacı için aşağıdaki parametreler arasında en iyi parametre aranmıştır:

```
params = {"max_features": ["auto", "sqrt", "log2"],
          "ccp_alpha": [0.1, .01, .001],
          "max_depth": [5, 6, 7, 8, 9, 10, 11, 12],
          "min_samples_split": [2, 3, 4],
          "min_samples_leaf": [1, 2],
          "criterion": ["gini", "entropy"]
          }
```

En iyi parametreler ise şu şekildedir:

'ccp_alpha': 0.01, 'criterion': 'entropy', 'max_depth': 11, 'max_features': 'auto', 'min_samples_leaf': 1, 'min_samples_split': 4

K en yakın komşuluk algoritmasında komşu sayısı için 1, 3, 5, 7, 9, 11 komşu değerleri için validasyon verisi ile test edilmiş ve en iyi sonuç komşu sayısı 3 iken elde edilmiştir.

Yapay sinir ağlarında ise model şu şekilde oluşturulmuştur:

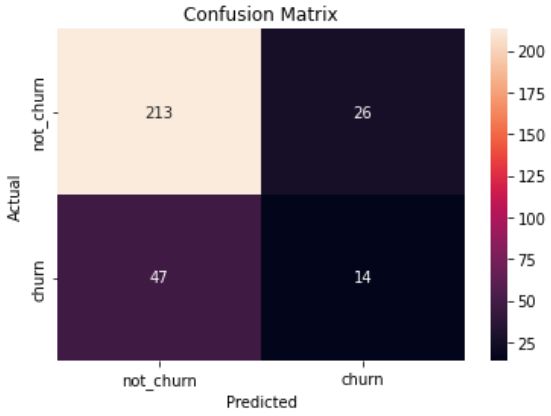
```
def create_model(activation = "sigmoid", hidden_layers=1):
  model = Sequential()
  model.add(Dense(12, activation=activation, input_shape=(12,)))

  for i in range(hidden_layers):
    model.add(Dense(36, activation=activation))

  model.add(Dense(1, activation="sigmoid"))
  model.compile(loss="binary_crossentropy", optimizer="adam", metrics=["accuracy"])

  return model
```

Girdi katmanı 12 adet girdi olduğu için 12 adet nörondan oluşmaktadır. Gizli katmanlarda ise 36 seçilmiştir. Çıktı katmanında ise tek bir çıktı olduğundan 1 seçilmiştir. Optimizasyon yöntemi olarak “Adam” seçilmiştir.

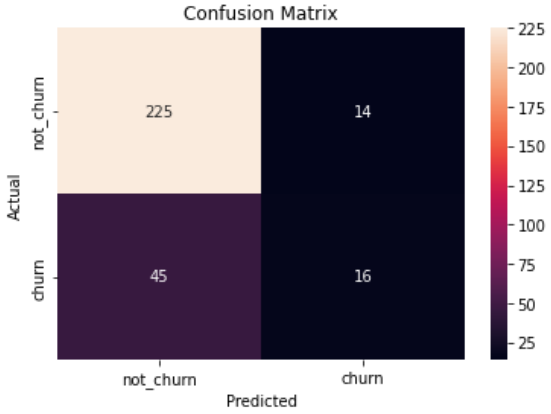
Parametre olarak şu parametreler arasında en iyi parametre aranmıştır:

```
hyperparameters = dict(
    epochs=[50, 100],
    batch_size=[16, 32, 64],
    hidden_layers=[1, 2, 3],
    activation=["relu", "sigmoid"],
)
```

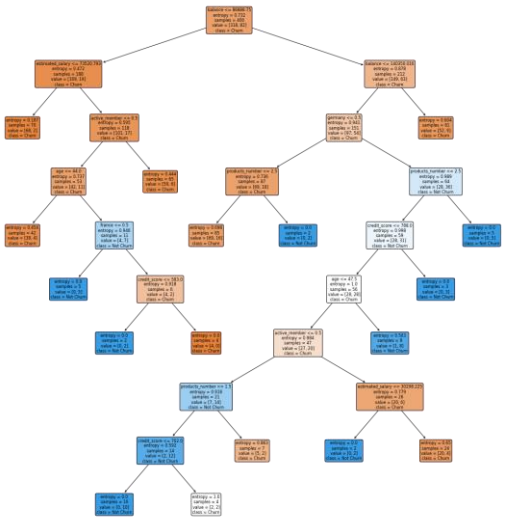
Naive Bayes için test verisi sonucunda karışıklık matrisi şu şekilde:



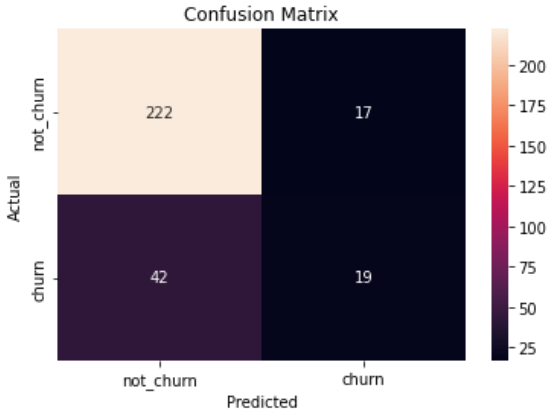
Karar ağacı için karışıklık matrisi şu şekilde:



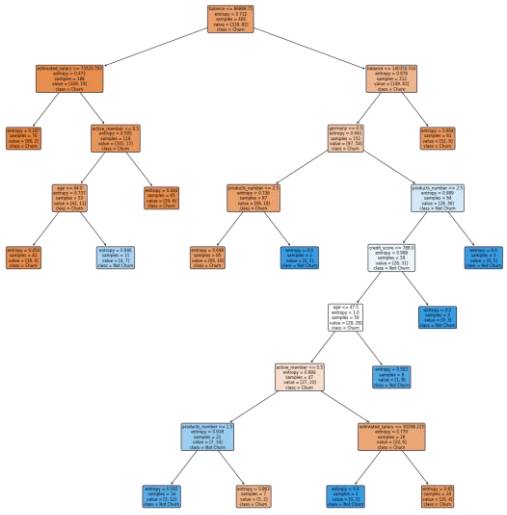
Oluşan karar ağacının görüntüsü ise şu şekilde:



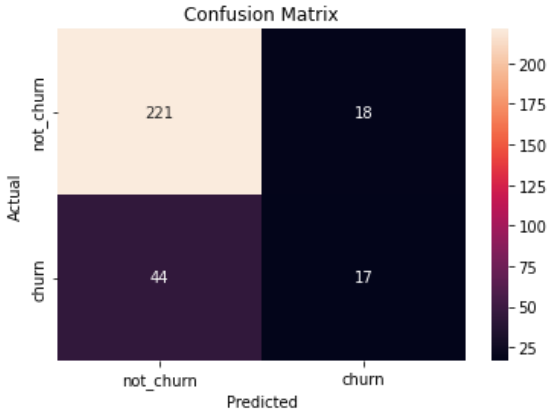
Karar ağacında en iyi parametrelerle eğitildikten sonra budama işlemi yapıldıktan sonraki karışıklık matrisi aşağıdaki gibidir:



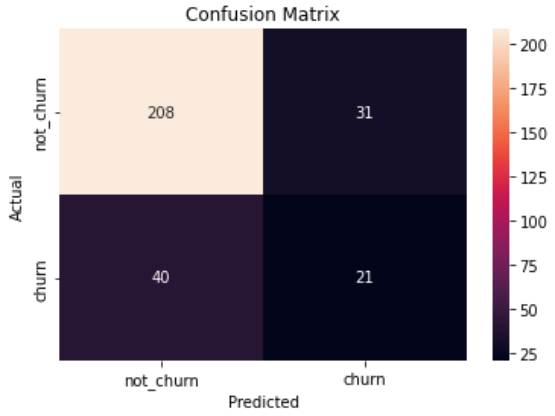
Budanan ağacın görüntüsü ise aşağıdaki gibidir:



K en yakın komşuluk için karışıklık matrisi şu şekilde:



Yapay sinir ağı için karışıklık matrisi şu şekilde:



Test verisi ile modellerin karşılaştırması şu şekilde:

Model	Accuracy	Precision	Recall	F1 Score
Naive Bayes	0.75667	0.58461	0.56036	0.56547
Decisition Tree	0.80333	0.68333	0.60186	0.61787
Decisition Tree Post Pruned	0.80333	0.68434	0.62017	0.63722
KNN	0.79333	0.65983	0.60169	0.61558
ANN	0.76333	0.62127	0.60727	0.61294

Tablodan da görülebileceği üzere en başarılı sonuç budanmış karar ağacından alınmıştır. Karar ağacını budamak accuracy de artışa neden olmasa da recall u arttırdığı için f1 score da artış meydana gelmiştir. En başarısız sonuç Naive Bayes algoritmasından alınmıştır. K en yakın komşuluk, yapay sinir ağı ve karar ağacı yakın bir başarıya sahiptir. Tüm modellerde gerçekte kaybedilen müşterilerin kaybedilmedi olarak yüksek sayıda tahmin edildiği gözlemlenmiştir.

Sonuç

En başarılı modelin başarısı tatmin edici değildir. Kaybedilmeyen 239 müşteriden 222 tanesi doğru bir şekilde tahmin edilmiştir ama 61 tane kaybedilen müşteriden sadece 19 tanesi doğru bir şekilde tahmin edilmiştir. Bunun sebebi örnek sayısının az olması, özelliklerin yetersiz olması ve verinin eşit dağılmaması olarak söylenebilir. Bu problemde asıl istenen kaybedilen müşterilerin tespiti olduğu için yapılan model tahminlemede yetersiz kalmıştır. Veride müşterinin kaybedilmesiyle düşük korelasyona sahip olan verilerden başka özellikler çıkarılabilirse başarı oranında artış meydana gelebilir. Bu 4 tahminleme modeli

dışında başka modellerin kullanılması da başarı oranında artış sağlayabilir. Toplamda 1000 tane veri kullanılması yerine tüm verinin kullanılması da başarı oranında artış sağlayabilir.