Telecom Churn Analysis

Muhammed Enes GÜLSOY
Computer Engineering
Yildiz Technical University
Istanbul, Turkey
enes.gulsoy@std.yildiz.edu.tr

***Abstract*—Bu döküman, telekomünikasyon müşterilerinin davranışlarını analiz edilmesi için kullanılan veri madenciliği modellerinin incelemesini içerir ve bu modellerin çıkarımlarını listeler.**

***Index Terms*—classification, deep-learning, model, knn, perceptron, precision, recall, confusion-matrix, roc, undersampling**

I. Giriş

Bir telekomünikasyon şirketi için halihazırda mevcut olan abonelerini elinde tutmak, yeni abone kazanmaktan daha kolaydır. Aynı şekilde, bir müşteri ayrılmaya karar vermişken onu kalmaya ikna etmek, müşteri ayrıldıktan sonra onu geri dönmeye ikna etmekten daha kolaydır. Hiçbir firma müşteri kaybetmekten hoşlanmaz.

Müşterilerin caymasının firmalara bedeli ağırdır.[3] Bu nedenle, müşteri davranışlarının analiz edilip, cayma ihtimali yüksek olan abonelerin tahmin edilmesi, ve ayrılmamaları için ikna edilmeleri gerekmektedir.

Bu projede, Telco firmasının müşterilerine ait bilgileri içeren veriseti[1] içerisindeki müşteri bilgileri analiz edilip, müşterilerin cayıp caymayacağı tahmin edilecektir. Elde edilen sonuçlar analiz edilerek, hangi özelliklere sahip müşterilerin caymaya yatkın olduğu ortaya çıkartılacaktır.

II. Veri Setinin Özellikleri

Veri setinde her bir satır bir müşteriyi temsil etmektedir. Toplamda 7043 müşteri bulunmaktadır. Kolonlarda ise müşterilere ait bilgiler yer almaktadır. Toplamda 21 tane feature bulunmaktadır. Churn attribute'u, tahmin etmeye çalıştığımız sınıftır.

A. Attributes (Features)

Verisetindeki featureları inceleyecek olursak;

- CustomerID: Müşteriye ait benzersiz tanımlayıcı.
- gender: Müşterinin cinsiyeti. [Female, Male]
- SeniorCitizen: Müşterinin yaşlı mı genç mi olduğu bilgisi. Müşteri emekliyse 1, değilse 0 değerini alır. [0, 1]
- Partner: Müşterinin partneri olup olmadığı. [Yes, No]
- Dependents: Müşteriye bağımlı olan başka bireylerin olup olmadığı. [Yes,No]
- tenure: Müşterinin hizmet aldığı ay sayısı.
- PhoneService: Müşteri telefon hizmeti alıyor mu? [Yes, No]
- MultipleLines: Müşterinin birden fazla telefon hattı var mı ? [Yes, No, No phone service]
- InternetService: Müşteri internet hizmeti alıyor mu? [DSL, Fiber optic, No]
- OnlineSecurity: Müşteri online güvenlik hizmeti alıyor mu? [Yes, No, No internet service]
- OnlineBackup: Müşteri online yedekleme hizmeti alıyor mu? [Yes, No, No internet service]
- DeviceProtection: Müşteri cihaz koruma hizmeti alıyor mu? [Yes, No, No internet service]
- TechSupport: Müşteri teknolojik destek hizmeti alıyor mu? [Yes, No, No internet service]
- StreamingTV: Müşteri TV yayını hizmeti alıyor mu? [Yes, No, No internet service]
- StreamingMovies: Müşteri film yayını hizmeti alıyor mu? [Yes, No, No internet service]
- Contract: Müşterinin kontrat süresi. [Month-to-month, One year, Two year]
- PaperlessBilling: Müşteri paperless billing tercih ediyor mu? [Yes, No]
- PaymentMethod: Müşterinin tercih ettiği ödeme yöntemi. [Electronic check, Mailed check, Bank transfer (automatic), Credit card (automatic)]
- MontlyCharges: Müşteriye aboneliği süresince fatura edilmiş aylık tutar.
- TotalCharges: Müşteriye aboneliği süresince fatura edilmiş toplam tutar.
- Churn: Müşterinin taşıma yapıp yapmadığı bilgisi. [Yes, No]

B. Veri Dağılımı

Şekil 1'de görüldüğü üzere, müşterilerin cinsiyet dağılımı hemen hemen birbirine eşittir. Müşterilerin %26.6'sı churn etmiştir. Hangi cinsiyetteki müşterilerin ne kadar caymaya yatkın olduğuna ise Şekil 2'den ulaşılabilir. Cayma işlemi ile cinsiyetler arasında bir korelasyon görülmemektedir. İki cinsiyetteki aboneler de benzer davranışlar sergilemiştir.

Şekil 3'de görülebileceği üzere, cayan abonelerin %75'i taahhüt vermemiş olup, %13'ü bir yıl, %3'ü ise iki yıl taahhüt vermiştir. Aydan aya ödeme yapan aboneler caymaya daha yatkındır.

Cayan müşterilerin çoğunluğu ödeme yöntemi olarak elektronik çek tercih etmiştir. Ödeme yöntemi olarak kredi kartıyla otomatik ödeme talimatı ve posta tercih eden aboneler caymaya daha az yatkındır. (Bkz. Şekil 4)

Müşterilerin çoğu fiber optik servisini seçmiştir, ayrıca fiber servisini seçen abonelerin yüksek bir cayma oranına sahip

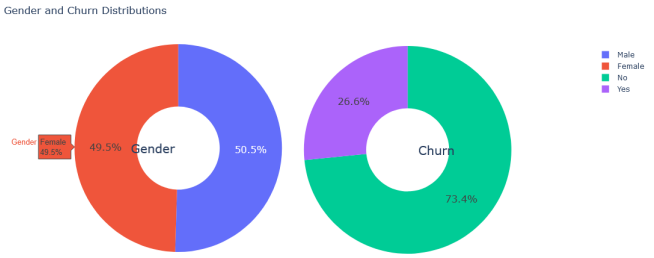

Fig. 1. Müşterilerin cinsiyet dağılımı ve cayma oranları

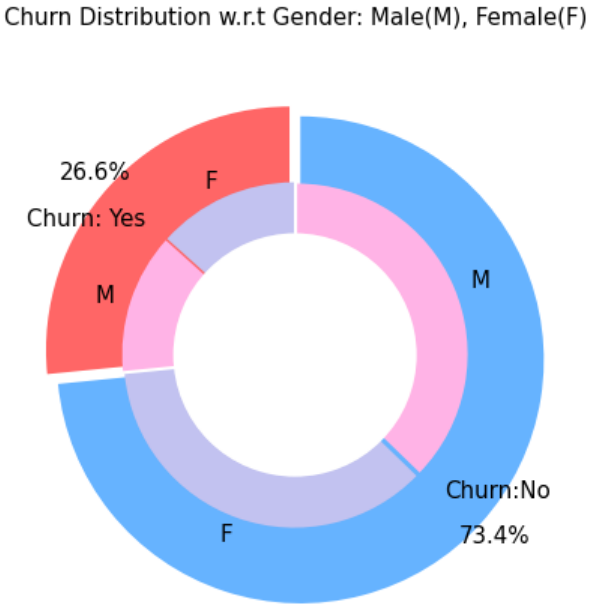

Fig. 2. Müşterilerin cinsiyet dağılımı beraberinde cayma oranları

olduğu görülmektedir. (Bkz. Şekil 5) Müşterilerin bu tip internet hizmetinden memnun olmadığı düşünülebilir. DSL tipinde servis alan müşterilerin cayma oranının, fiber abonelerine göre daha az olduğu görülmektedir.

Kendisine bağımlı bireyler olmayan abonelerin caymaya daha yatkın olduğu tespit edilmiştir. (Bkz. Şekil 6)

Partneri olmayan aboneler caymaya daha yatkın olarak görünmektedir. (Bkz. Şekil 7)

Emekli abonelerin genele oranla daha az olduğu Şekil 8'de görülebilir. Cayan abonelerin yaklaşık %25'i emeklidir.

Online Security ve Tech Support hizmeti alan abonelerin caymaya daha az meyilli olduğu Şekil 9 ve Şekil 10'de gözlemlenebilir.

Yeni müşteriler caymaya daha yatkındır.(Bkz. Şekil 11)

Şekil 12'de, churn ile korelasyonu yüksek olan attribute'lar Contract, tenure, OnlineSecurity, TechSupport olarak görünmektedir.

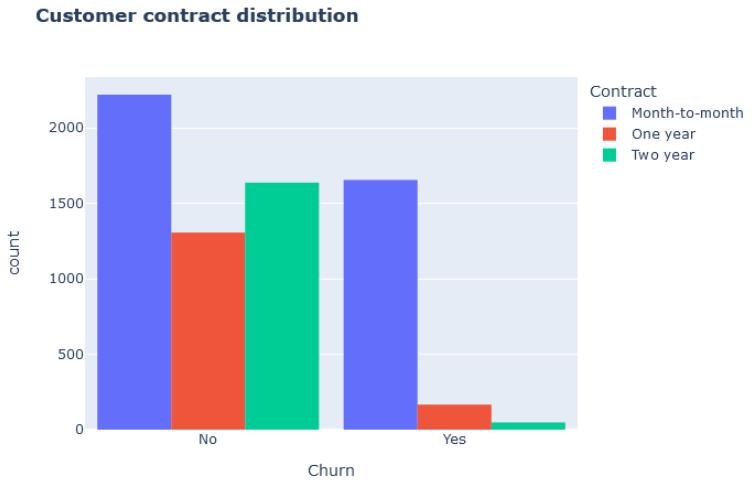

Fig. 3. Taahhüt süresi ve cayma ilişkisi

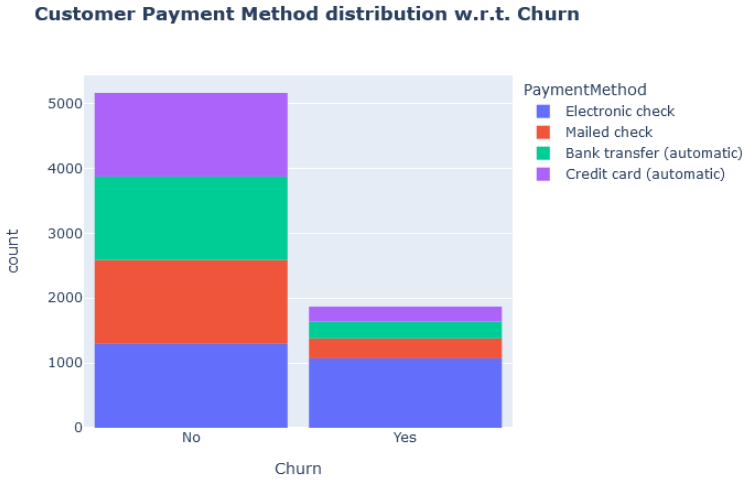

Fig. 4. Ödeme yöntemi ve cayma ilişkisi

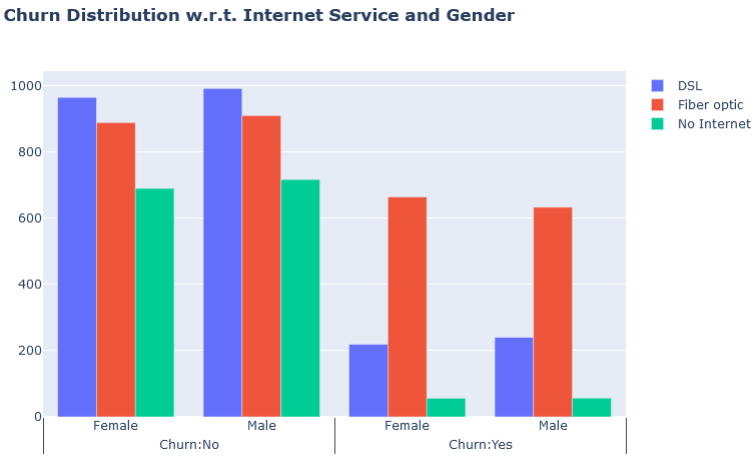

Fig. 5. Churn bazında, seçilen internet servisi ve cinsiyet arasındaki ilişki

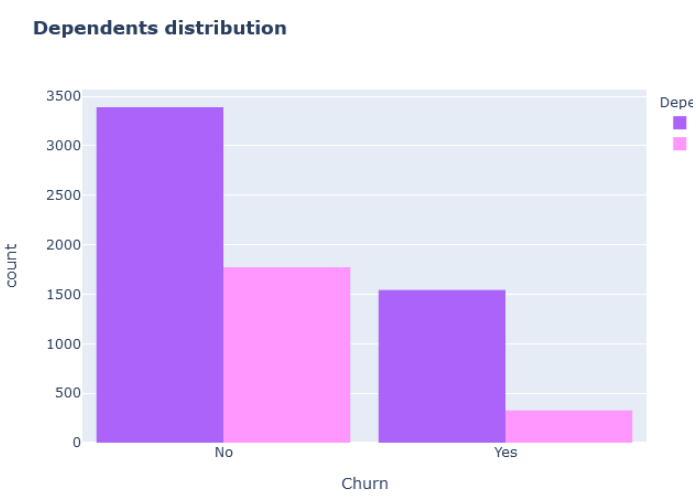

Fig. 6. Churn ve müşteriye bağımlı olan kişi ilişkisi

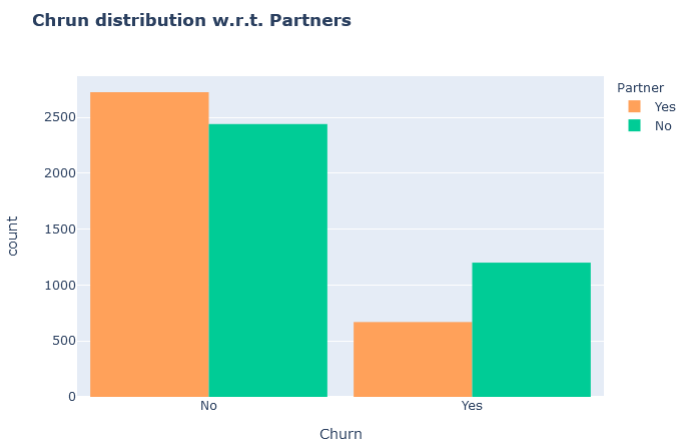

Fig. 7. Churn ve müşterinin partneri olup olmaması arasındaki ilişki

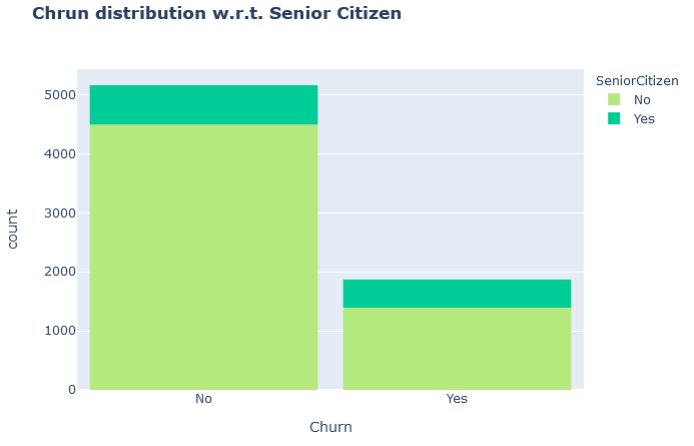

Fig. 8. Emeklilik durumu ve churn arasındaki ilişki

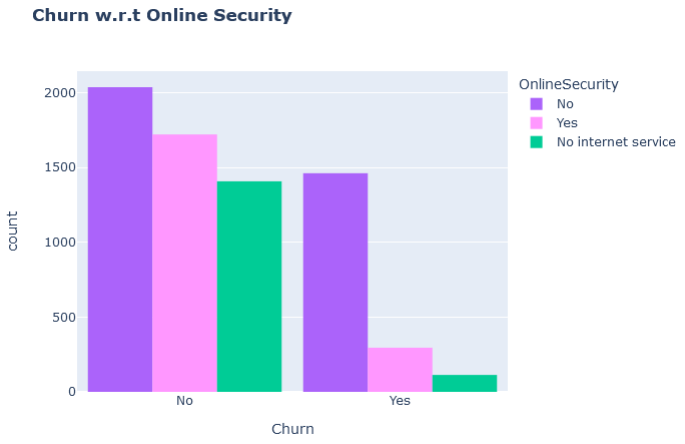

Fig. 9. Online Security hizmeti ile churn ilişkisi

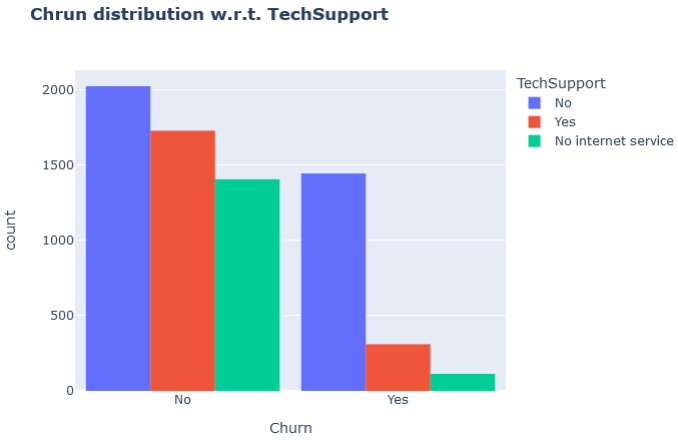

Fig. 10. Tech Support ile churn ilişkisi

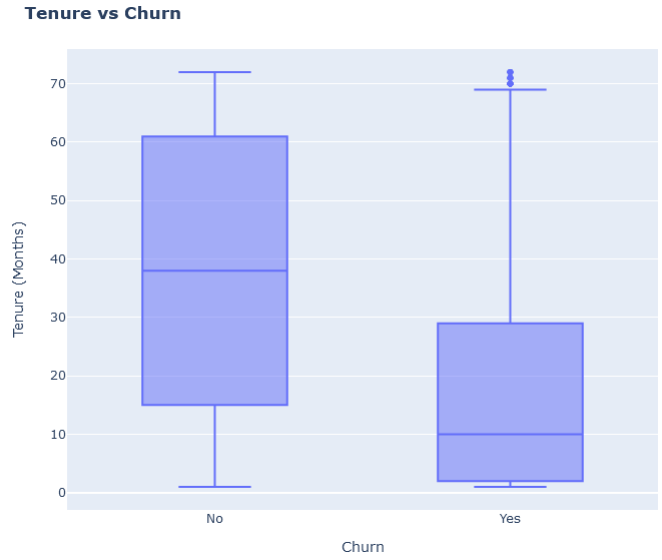

Fig. 11. Abone olunan süre ve churn ilişkisi

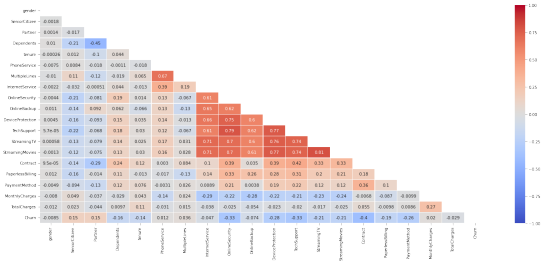

Fig. 12. Attribute'lar arasındaki korelasyon

III. Veri Setinin Hazırlanışı

Kodlama için colab ortamı tercih edilmiştir.[2]

İlk olarak, eksik veriye sahip tuplelar tespit edilmiştir. 11 tane müşterinin TotalCharges ve tenure bilgisi eksiktir. Genele vurulduğunda az bir sayı olduğundan, eksik verileri doldurmak yerine komple atmak tercih edilmiştir.

Sonrasında kategorik değerler kodlanarak sayısal değere çevrilmiştir. Örneğin, gender kolonu için Female/Male yerine 0/1 gibi.

7043 müşteriden eksik veriye sahip olanlar atıldıktan sonra, geriye kalan 7032 müşteriden 1869'u caymış, 5163'ü ise caymamıştır. Veri imbalanced olarak dağıldığı için, bu şekilde modele verildiği zaman precision ve recall değerleri düşük çıkmaktadır. Model, çoğunluk olan sınıfı öğrenebilmekte, ancak azınlık olan sınıfta hata yapmaktadır. Bu sorunu çözmek için undersampling yöntemi uygulanmıştır. Caymamış 5163 müşteri içerisinden rastgele 1869 kişi seçilmiş, sonuç olarak 1869 churned tuple ve 1869 not churned tuple elde edilmiştir.

Sonrasında bu verinin %70'i train, %30'u test set olarak ayrılmış ve bu ayırma işleminde stratify fonksiyonu kullanılmıştır. (Böylece sınıf dağılımları train ve test kümelerinde dengeli olacaktır.) Son olarak numerik kolonlar için ('tenure', 'MonthlyCharges', 'TotalCharges') scaling uygulanmıştır.

IV. Kullanılan Modeller

Bu bölümde, sınıfı bilinmeyen müşterilerin churn edip etmeyeceğini tahmin etmek için eğitilen modellerin başarıları kıyaslanacaktır.

A. Decision Tree

Modelin eğitim sonucu verdiği sonuçlar için Şekil 13 ve Şekil 14'yi inceleyiniz. Accuracy, Recall, Precision değerleri 0.68 olarak ölçülmüştür.

B. Random Forest

Modelin eğitim sonucu verdiği sonuçlar için Şekil 15, Şekil 16 ve Şekil 17'yi inceleyiniz. Accuracy 0.75, not churned sınıfının precision değeri 0.76, recall değeri 0.73; churned sınıfının precision değeri 0.74, recall değeri 0.77 olarak ölçülmüştür.

Tasarımı gereği birden çok ağaç kullanarak sınıflandırma yapan RF, Decision tree'den daha iyi bir sonuç vermiştir.

```
[ ]  clf = DecisionTreeClassifier().fit(X_train, y_train)

     print('Accuracy of Decision Tree classifier on test set: {:.2f}'
          .format(clf.score(X_test, y_test)))

     Accuracy of Decision Tree classifier on test set: 0.68
```

```
     ypredict = clf.predict(X_test)
     print(classification_report(y_test, ypredict))
```

	precision	recall	f1-score	support
0	0.68	0.68	0.68	561
1	0.68	0.68	0.68	561
accuracy			0.68	1122
macro avg	0.68	0.68	0.68	1122
weighted avg	0.68	0.68	0.68	1122

Fig. 13. Decision Tree Classifier Accuracy, Precision, Recall Değerleri

```
     plt.figure(figsize=(4,3))
     sns.heatmap(confusion_matrix(y_test, ypredict),
                 annot=True,fmt = "d",linecolor="k",linewidths=3)

     plt.title(" Decision Tree CONFUSION MATRIX",fontsize=14)
     plt.show()
```

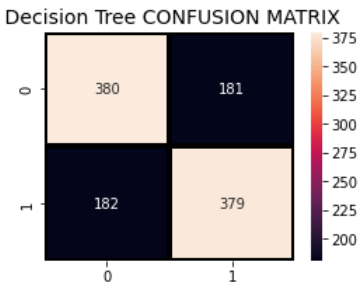

Fig. 14. Decision Tree Classifier Confusion Matrix

C. Adaboost

Şekil 18'de görülebileceği üzere, accuracy değeri 0.75'tir.

D. Logistic Regression

Şekil 19'i inceleyiniz. Accuracy 0.76 olarak ölçülmüştür. Precision ve recall değerleri de 0.74 üzerindedir.

E. KNN

Şekil 20'i inceleyiniz. Accuracy 0.68, precision 0.68, recall ise 0.66 üzerindedir.

F. Perceptron

Şekil 21'i inceleyiniz. Accuracy 0.68 idir. Not churned sınıfı için recall 0.47, precision 0.84; churned sınıfı için recall 0.91, precision 0.63 olarak ölçülmüştür.

```
randomforestmodel = RandomForestClassifier(n_estimators=500 , oob
                                random_state =50, max_features
                                max_leaf_nodes = 30)
randomforestmodel.fit(X_train, y_train)

# Make predictions
prediction_test = randomforestmodel.predict(X_test)
print (metrics.accuracy_score(y_test, prediction_test))
```

```
0.7522281639928698
```

```
plt.figure(figsize=(4,3))
sns.heatmap(confusion_matrix(y_test, prediction_test),
                annot=True,fmt = "d",linecolor="k",linewidths=3)

plt.title(" RANDOM FOREST CONFUSION MATRIX",fontsize=14)
plt.show()
```

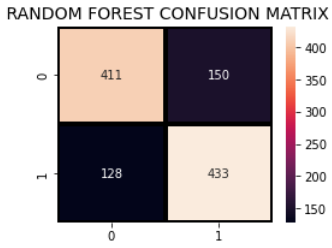

Fig. 15. Random Forest Accuracy ve Confusion Matrix

```
print(classification_report(y_test, prediction_test))
```

	precision	recall	f1-score	support
0	0.76	0.73	0.75	561
1	0.74	0.77	0.76	561
accuracy			0.75	1122
macro avg	0.75	0.75	0.75	1122
weighted avg	0.75	0.75	0.75	1122

Fig. 16. Random Forest Precision, Recall Değerleri

G. SVM

Şekil 22'i inceleyiniz. Accuracy 0.74, precision 0.74, recall 0.73 üzerindedir.

H. Diğer Modeller

Voting classifier accuracy değeri 0.76, Gaussian classifier accuracy değeri ise 0.75 olarak ölçülmüştür. (Bkz. Şekil 23)

```
y_rfpred_prob = randomforestmodel.predict_proba(X_test)[:,1]
fpr_rf, tpr_rf, thresholds = roc_curve(y_test, y_rfpred_prob)
plt.plot([0, 1], [0, 1], 'k--' )
plt.plot(fpr_rf, tpr_rf, label='Random Forest',color = "r")
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title('Random Forest ROC Curve',fontsize=16)
plt.show();
```

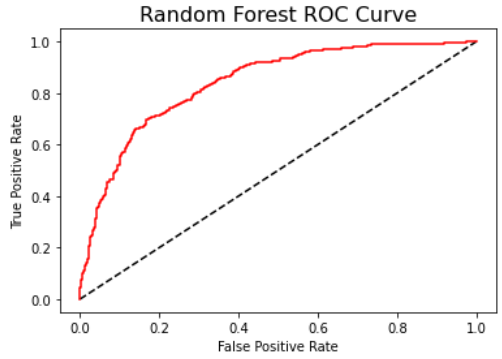

Fig. 17. Random Forest ROC Curve

```
a_model = AdaBoostClassifier()
a_model.fit(X_train,y_train)
a_preds = a_model.predict(X_test)
print("AdaBoost Classifier accuracy")
metrics.accuracy_score(y_test, a_preds)
```

```
AdaBoost Classifier accuracy
0.7513368983957219
```

Fig. 18. Adaboost Accuracy

```
[59] lr_model = LogisticRegression()
     lr_model.fit(X_train,y_train)
     accuracy_lr = lr_model.score(X_test,y_test)
     gyy = lr_model.predict(X_test)
     print("Logistic Regression accuracy is :",accuracy_lr)
```

```
Logistic Regression accuracy is : 0.7602495543672014
```

```
print(classification_report(y_test, gyy))
```

	precision	recall	f1-score	support
0	0.77	0.74	0.75	561
1	0.75	0.78	0.77	561
accuracy			0.76	1122
macro avg	0.76	0.76	0.76	1122
weighted avg	0.76	0.76	0.76	1122

Fig. 19. Logistic Regression Accuracy, Precision, Recall Değerleri

```
knn_model = KNeighborsClassifier (n_neighbors = 3)
knn_model.fit(X_train,y_train)
predicted_y22 = knn_model.predict(X_test)
accuracy_knn = knn_model.score(X_test, y_test)
print("KNN accuracy:",accuracy_knn)
```

```
KNN accuracy: 0.6871657754010695
```

```
print(classification_report(y_test, predicted_y22))
```

```
              precision    recall  f1-score   support

           0       0.70      0.66      0.68       561
           1       0.68      0.72      0.70       561

    accuracy                           0.69      1122
   macro avg       0.69      0.69      0.69      1122
weighted avg       0.69      0.69      0.69      1122
```

Fig. 20. KNN Accuracy, Precision, Recall Değerleri

```
[57] from sklearn.linear_model import Perceptron
     p = Perceptron()

     p.fit(X_train,y_train)
     predictpercpt= p.predict(X_test)
     print("Perceptron accuracy")
     accuracy_score (y_test, predictpercpt)
```

```
Perceptron accuracy
0.6898395721925134
```

```
print(classification_report(y_test, predictpercpt))
```

```
              precision    recall  f1-score   support

           0       0.84      0.47      0.60       561
           1       0.63      0.91      0.75       561

    accuracy                           0.69      1122
   macro avg       0.74      0.69      0.67      1122
weighted avg       0.74      0.69      0.67      1122
```

Fig. 21. Perceptron Accuracy, Precision, Recall Değerleri

V. Sonuç

Modeller genel olarak %60'ın üzerinde accuracy, precision ve recall değeri vermiştir. Uygun bir model ile, bir firma hangi müşterisinin kendisini terk edeceğini, hangisinin ise kendisinde aboneliğe devam etmeye meyilli olduğunu tahmin edebilir ve bu sayede müşteri kaybetmeyerek kârını artırabilir.

K-Fold kullanılarak başarı daha da artırılabilir. Undersampling yerine sentetik veri üretilerek veri sayısı fazlalaştırılabilir. Türlü metotlar ile eğitim sürecine yön vermek mümkündür.

```
[52] #support vector machines
     svm = SVC(random_state=213)
     svm.fit(X_train, y_train)

     print('Accuracy of SVM classifier')
     sonuy = svm.predict(X_test)
     accuracy_score (y_test, sonuy)
```

```
Accuracy of SVM classifier
0.749554367201426
```

```
print(classification_report(y_test, sonuy))
```

```
              precision    recall  f1-score   support

           0       0.76      0.73      0.75       561
           1       0.74      0.76      0.75       561

    accuracy                           0.75      1122
   macro avg       0.75      0.75      0.75      1122
weighted avg       0.75      0.75      0.75      1122
```

Fig. 22. Support Vector Machine Accuracy, Precision, Recall Değerleri

```
from sklearn.ensemble import VotingClassifier
clf1 = GradientBoostingClassifier()
clf2 = LogisticRegression()
clf3 = AdaBoostClassifier()
eclf1 = VotingClassifier(estimators=[('gbc', clf1), ('lr',
eclf1.fit(X_train, y_train)
predictions = eclf1.predict(X_test)
print("Final Accuracy Score ")
print(accuracy_score(y_test, predictions))
```

```
Final Accuracy Score
0.7638146167557932
```

```
gnb = GaussianNB()
gnb.fit(X_train, y_train)
print('Accuracy of GNB classifier on training set: {:.2f}'
     .format(gnb.score(X_train, y_train)))
print('Accuracy of GNB classifier on test set: {:.2f}'
     .format(gnb.score(X_test, y_test)))
```

```
Accuracy of GNB classifier on training set: 0.75
Accuracy of GNB classifier on test set: 0.75
```

Fig. 23. Voting Classifier and Gaussian Classifier Accuracy Değerleri

References

[1] blastchar. *Telco Customer Churn*. https://www.kaggle.com/datasets/blastchar/telco-customer-churn. Accessed: 2022-05-14.

[2] M. Gülsoy. *Colab Script*. https://github.com/enzo11183/Customer_Churn_Prediction, https://colab.research.google.com/drive/1jEkHsNL5_Xexr64Ixn0ol6juzipMwf4U?usp=sharing. [Online; accessed 22-May-2022].

[3] *What is customer churn? How to measure & prevent it*. https://www.qualtrics.com/uk/experience-management/customer/customer-churn/. Accessed: 2022-05-20.