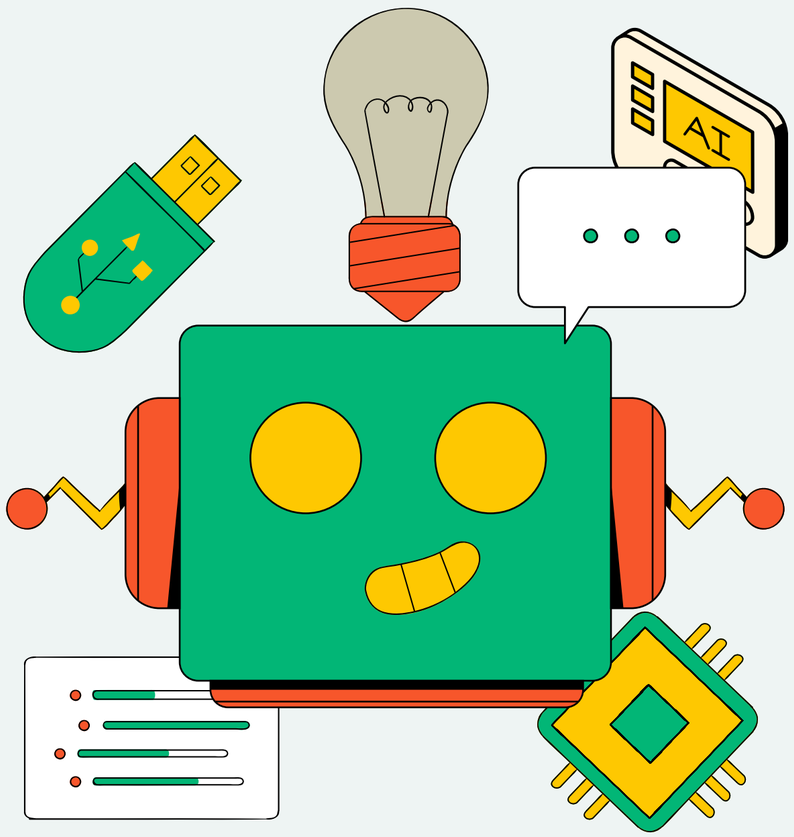
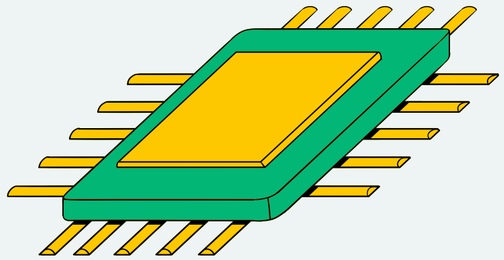


MÜŞTERİ KAYBI TAHMİNİ

HAZIRLAYAN:

SENA KİLCİOĞLU-
G210102052

SEVİLAY BAYRAM-
G220102352



PROJENİN AMACI

Bu proje, müşteri davranışlarını anlamak ve müşteri kaybını (churn) tahmin etmek için bir makine öğrenimi modeli geliştirmeyi amaçlıyor.

- Müşteri Kaybını Tahmin Etmek
- Müşteri Segmentasyonu
- Stratejik Karar Desteği

MÜŞTERİ KAYBINI TAHMİN ETMEK

Proje, mevcut müşteri verilerini analiz ederek hangi müşterilerin şirketten ayrılma olasılığının yüksek olduğunu belirlemeyi hedeflemektedir.

MÜŞTERİ SEGMENTASYONU

Cinsiyet, ödeme tercihleri, sipariş alışkanlıkları gibi faktörlere göre müşteri gruplarını inceleyerek daha iyi hedeflerin belirlenmesini sağlamaktadır.

STRATEJİK KARAR DESTEĞİ

Elde edilen tahminler ve analizler, pazarlama stratejilerini geliştirmek, sadakat programlarını iyileştirmek ve müşteri deneyimini artırmak için kullanılabilir.

BU PROJE SONUCUNDA ;

- Müşteri kaybını azaltarak şirketin gelirlerini artırabilir.
- Daha iyi müşteri anlayışı sayesinde hizmet kalitesini ve memnuniyeti artırır.
- Daha etkili pazarlama kampanyaları düzenlenmesini sağlar.
- Daha odaklı stratejiler geliştirilerek zaman ve kaynak tasarrufu sağlar.

UYGULAMA ADIMLARI

1-Verilerin elde edilmesi

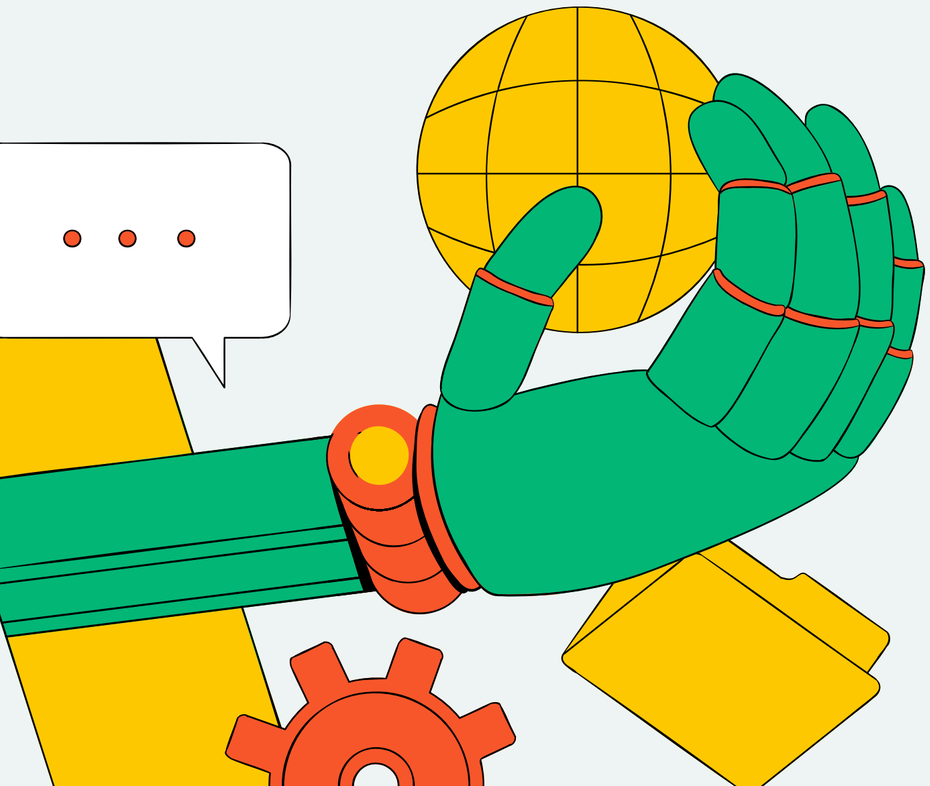
2- Verilerin incelenmesi

3-Veri açıklama ve hazırlama

4-Modelin eğitilmesi ve değerlendirilmesi

5-Modelin uygulanması

1-VERİLERİN ELDE EDİLMESİ



- WORLD BANK OPEN DATA
- EUROSTAT
- OECDSTAT
- KAGGLE
- GİTHUP

Bu platformalar üzerinden veri setleri taranarak proje konusuna en uygun veri seti seçilmiştir.

2-VERİLERİN İNCELENMESİ

-Müşteri ID
-Kayıp
-Kullanım süresi

-Tercih edilen cihaz
-Şehir kademesi
-Müşteri ve depo arası mesafe

-Ödeme şekli
-Cinsiyet
-Uygulamada geçirdiği süre

-Kayıtlı cihaz sayısı
-Önceki siparişte tercih edilen kategori
-Memnuniyet puanı

-Medeni hali
-Kayıtlı adres sayısı
-Şikayet var mı?

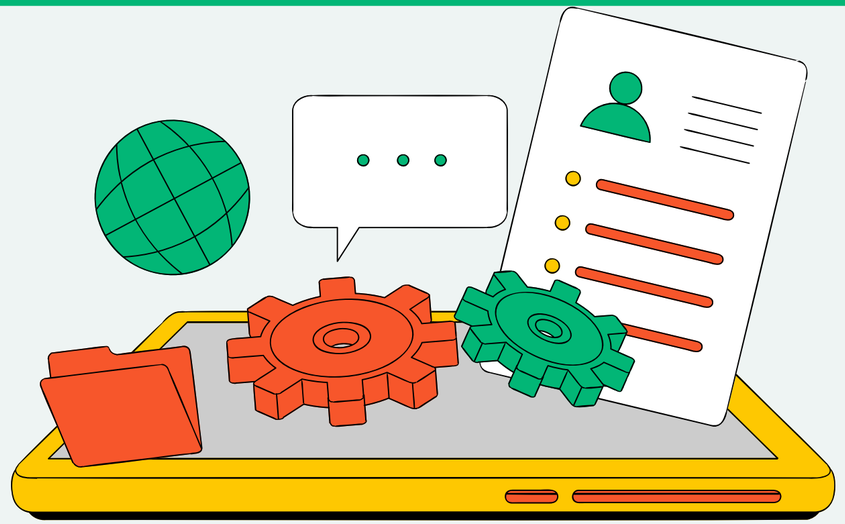
-Sipariş miktarı
-Kullanılan kupon sayısı
-Geçen ay verilen sipariş miktarı

-Son siparişten bu yana geçen gün sayısı
-İade tutarı

Variable	Discerption
CustomerID	5630 müşteri verisi ele alınmıştır.
Churn	Müşteri kaybı (0 veya 1)
Tenure	Kullanım süresi (0- 61 ay arasında değerler almıştır.)
PreferredLoginDevice	Tercih edilen cihaz (phone- mobil phone- computer)
CityTier	Şehir kademesi (1-2-3)
WarehouseToHome	Müşteri ve depo arası mesafe
PreferredPaymentMode	Ödeme şekli (UPI-CC -COD -E Wallet-DC)
Gender	Müşteri cinsiyet (Male- Female)
HourSpendOnApp	Müşterinin sitede harcadığı saat (0-5 saat)
NumberOfDeviceRegistered	Kayıtlı cihaz sayısı (1-6)
PreferedOrderCat	Müşterinin geçen ay tercih ettiği sipariş kategorisi (Laptop&Accessory, Mobile, Mobile Phone, Fashion, Grocery, Others)
SatisfactionScore	Memnuniyet derecesi (1-5)
MaritalStatus	Medeni hali (Single, Divorced, Married)
NumberOfAddress	Kayıtlı adres sayısı
Complain	Şikayet var mı? (0 veya 1)
OrderAmountHikeFromlastYear	Geçen yıl verdiği sipariş miktarı
CouponUsed	Geçen ay kullandığı kupon sayısı
OrderCount	Geçen ay verdiği sipariş sayısı
DaySinceLastOrder	Son siparişten bu yana geçen gün sayısı
CashbackAmount	İade tutarı

KÜTÜPHANELERİN YÜKLENMESİ

```python
# Gerekli kütüphaneler
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder, StandardScaler
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import confusion_matrix, classification_report, roc_auc_score, roc_curve
```



VERİ SETİNİN YÜKLENMESİ

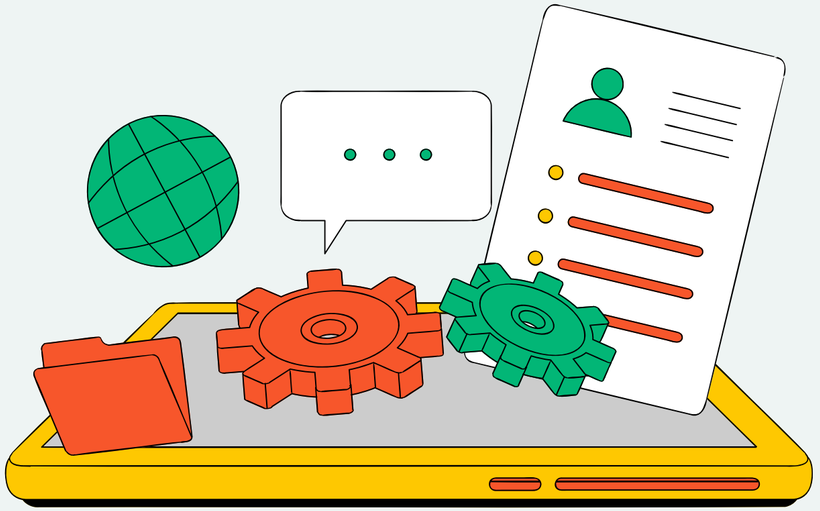
```
# Veriyi yükleme
file_path = 'C:/Users/HP/Desktop/4.0/E Commerce Dataset .xlsx'
excel_data = pd.ExcelFile(file_path)
data = excel_data.parse('E Comm')
```

excel_data.parse('E Comm') fonksiyonu, excel_data nesnesi üzerinde belirtilen 'E Comm' adlı sayfayı yükler ve bu sayfanın içeriğini bir pandas DataFrame'ine (data) dönüştürür. 'E Comm' burada, Excel dosyasındaki sayfanın adıdır.

VERİ SETİ ÇERÇEVESİ

Index	CustomerID	Churn	Tenure	PreferredLoginDevice	CityTier	WarehouseToHome	PreferredPaymentMode	Gender	HourSpendOn	NumberOfDeviceRe	PreferedOrderCat	SatisfactionSc	MaritalStatus	NumberOfAddress	Complain	OrderAmountHikeFrc	CouponUsed	OrderCount	DaySinceLastO
0	50001	1	4	Mobile Phone	3	6	Debit Card	Female	3	3	Laptop & Accessory	2	Single	9	1	11	1	1	5
1	50002	1	nan	Phone	1	8	UPI	Male	3	4	Mobile	3	Single	7	1	15	0	1	0
2	50003	1	nan	Phone	1	30	Debit Card	Male	2	4	Mobile	3	Single	6	1	14	0	1	3
3	50004	1	0	Phone	3	15	Debit Card	Male	2	4	Laptop & Accessory	5	Single	8	0	23	0	1	3
4	50005	1	0	Phone	1	12	CC	Male	nan	3	Mobile	5	Single	3	0	11	1	1	3
5	50006	1	0	Computer	1	22	Debit Card	Female	3	5	Mobile Phone	5	Single	2	1	22	4	6	7
6	50007	1	nan	Phone	3	11	Cash on Delivery	Male	2	3	Laptop & Accessory	2	Divorced	4	0	14	0	1	0
7	50008	1	nan	Phone	1	6	CC	Male	3	3	Mobile	2	Divorced	3	1	16	2	2	0
8	50009	1	13	Phone	3	9	E wallet	Male	nan	4	Mobile	3	Divorced	2	1	14	0	1	2
9	50010	1	nan	Phone	1	31	Debit Card	Male	2	5	Mobile	3	Single	2	0	12	1	1	1
10	50011	1	4	Mobile Phone	1	18	Cash on Delivery	Female	2	3	Others	3	Divorced	2	0	nan	9	15	8
11	50012	1	11	Mobile Phone	1	6	Debit Card	Male	3	4	Fashion	3	Single	10	1	13	0	1	0
12	50013	1	0	Phone	1	11	COD	Male	2	3	Mobile	3	Single	2	1	13	2	2	2
13	50014	1	0	Phone	1	15	CC	Male	3	4	Mobile	3	Divorced	1	1	17	0	1	0
14	50015	1	9	Mobile Phone	3	15	Credit Card	Male	3	4	Fashion	2	Single	2	0	16	0	4	7

```
# İlk 5 satıra göz atalım
print(data.head())
```



```
In [4]:
   ...: print(data.head())
   CustomerID  Churn  Tenure  ... OrderCount  DaySinceLastOrder  CashbackAmount
0       50001      1     4.0  ...         1.0                5.0          159.93
1       50002      1     NaN  ...         1.0                0.0          120.90
2       50003      1     NaN  ...         1.0                3.0          120.28
3       50004      1     0.0  ...         1.0                3.0          134.07
4       50005      1     0.0  ...         1.0                3.0          129.60

[5 rows x 20 columns]
```

GENEL İSTATİSTİKLER

data.describe(): Veri setindeki sayısal sütunlar için özet istatistikler sağlar (ortalama, medyan, standart sapma, min, max, çeyrekler vb.).

```
# Genel istatistikler
print(data.describe())
print(data.info())
```

```
In [5]:
   ...:
   ...: print(data.describe())
   ...: print(data.info())
         CustomerID        Churn  ...  DaySinceLastOrder  CashbackAmount
count   5630.000000  5630.000000  ...        5323.000000     5630.000000
mean   52815.500000     0.168384  ...           4.543491      177.223030
std     1625.385339     0.374240  ...           3.654433       49.207036
min    50001.000000     0.000000  ...           0.000000        0.000000
25%    51408.250000     0.000000  ...           2.000000      145.770000
50%    52815.500000     0.000000  ...           3.000000      163.280000
75%    54222.750000     0.000000  ...           7.000000      196.392500
max    55630.000000     1.000000  ...          46.000000      324.990000
```

GENEL İSTATİSTİKLER

data.info(): Veri setinin genel yapısı hakkında bilgi verir (sütun adları, veri türleri, eksik değerler vb.).

```
# Genel istatistikler
print(data.describe())
print(data.info())
```

```
#   Column                       Non-Null Count  Dtype
---  ------                       --------------  -----
0   CustomerID                   5630 non-null   int64
1   Churn                        5630 non-null   int64
2   Tenure                       5366 non-null   float64
3   PreferredLoginDevice         5630 non-null   object
4   CityTier                     5630 non-null   int64
5   WarehouseToHome              5379 non-null   float64
6   PreferredPaymentMode         5630 non-null   object
7   Gender                       5630 non-null   object
8   HourSpendOnApp               5375 non-null   float64
9   NumberOfDeviceRegistered     5630 non-null   int64
10  PreferedOrderCat             5630 non-null   object
11  SatisfactionScore            5630 non-null   int64
12  MaritalStatus                5630 non-null   object
13  NumberOfAddress              5630 non-null   int64
14  Complain                     5630 non-null   int64
15  OrderAmountHikeFromlastYear  5365 non-null   float64
16  CouponUsed                   5374 non-null   float64
17  OrderCount                   5372 non-null   float64
18  DaySinceLastOrder            5323 non-null   float64
```

EKSİK VERİLERİN KONTROLÜ

```
# Eksik veri kontrolü
missing_data = data.isnull().sum()
print(missing_data)
```

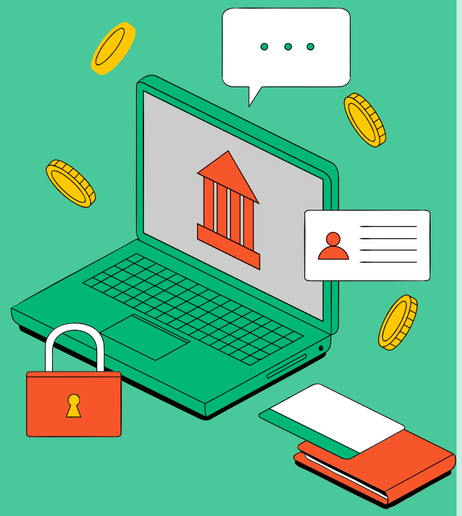
data.isnull() fonksiyonu, verinin her hücresinde eksik veri olup olmadığını kontrol etmek için kullanılır.

data.isnull().sum() fonksiyonu, her sütun için eksik (NaN) veri sayısını hesaplar.

```
In [6]:
   ...: missing_data = data.isnull().sum()
   ...: print(missing_data)
CustomerID                       0
Churn                            0
Tenure                         264
PreferredLoginDevice             0
CityTier                         0
WarehouseToHome                251
PreferredPaymentMode             0
Gender                           0
HourSpendOnApp                 255
NumberOfDeviceRegistered         0
PreferedOrderCat                 0
SatisfactionScore                0
MaritalStatus                    0
NumberOfAddress                  0
Complain                         0
OrderAmountHikeFromlastYear    265
CouponUsed                     256
OrderCount                     258
DaySinceLastOrder              307
CashbackAmount                   0
dtype: int64
```

EKSİK VERİLERİN KONTROLÜ

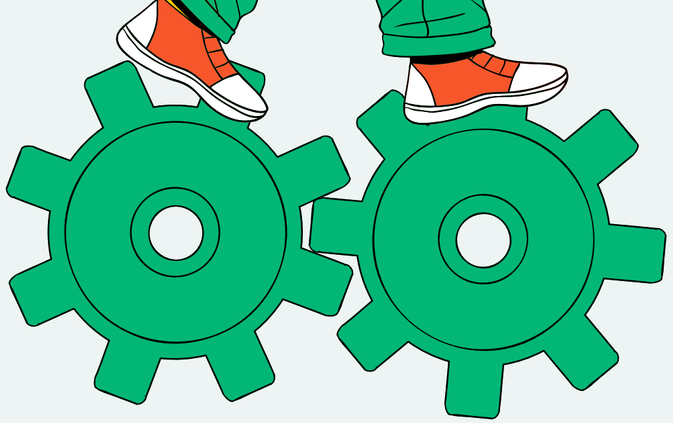
```
# Eksik verileri doldurma
data['Tenure'].fillna(data['Tenure'].median(), inplace=True)
data['HourSpendOnApp'].fillna(data['HourSpendOnApp'].mean(), inplace=True)
data['WarehouseToHome'].fillna(data['WarehouseToHome'].median(), inplace=True)
data['OrderAmountHikeFromlastYear'].fillna(data['OrderAmountHikeFromlastYear'].median(), inplace=True)
data['CouponUsed'].fillna(data['CouponUsed'].median(), inplace=True)
data['OrderCount'].fillna(data['OrderCount'].median(), inplace=True)
data['DaySinceLastOrder'].fillna(data['DaySinceLastOrder'].median(),inplace=True)
```



HourSpendOnApp sütunundaki verilerin ortama değeri; Tenure, WarehouseToHome, OrderAmountHikeFromlastYear ,CouponUsed, OrderCount, DaySinceLastOrder sütunlarındaki eksik değerleri medyan ile dolduruldu.

```
In [5]:
   ...: missing_data = data.isnull().sum()
   ...: print(missing_data)
CustomerID                       0
Churn                            0
Tenure                           0
PreferredLoginDevice             0
CityTier                         0
WarehouseToHome                  0
PreferredPaymentMode             0
Gender                           0
HourSpendOnApp                   0
NumberOfDeviceRegistered         0
PreferedOrderCat                 0
SatisfactionScore                0
MaritalStatus                    0
NumberOfAddress                  0
Complain                         0
OrderAmountHikeFromlastYear      0
CouponUsed                       0
OrderCount                       0
DaySinceLastOrder                0
CashbackAmount                   0
dtype: int64
```

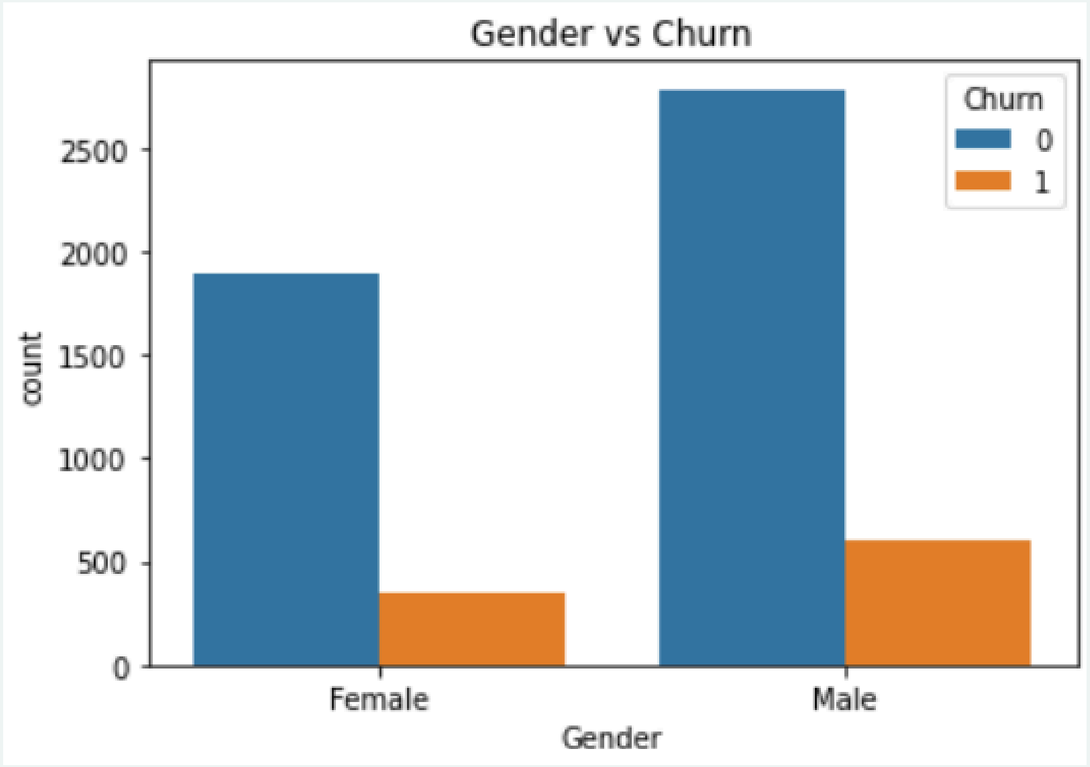
CİNSİYET VE MÜŞTERİ KAYBI ARASINDAKİ İLİŞKİ



```
# Cinsiyet ve müşteri kaybı (Churn) arasındaki ilişki
sns.countplot(data=data, x='Gender', hue='Churn')
plt.title("Gender vs Churn")
plt.show()
```

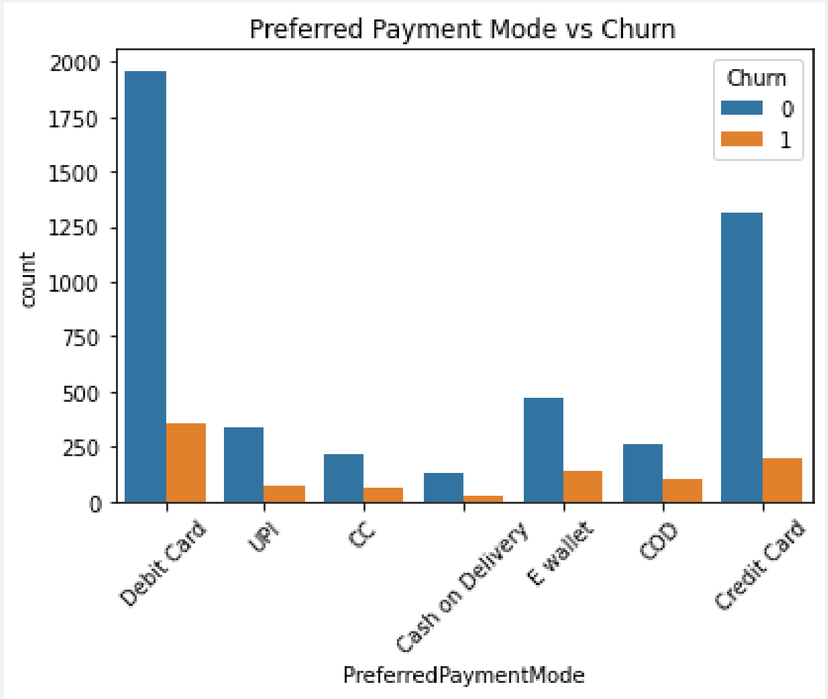
Erkek müşterilerin toplam sayısı daha fazla, ancak "Churn" (müşteri kaybı) yaşayan erkeklerin oranı, kadınlara kıyasla biraz daha yüksektir.

Müşteri kaybını azaltmak için erkek müşterilere yönelik stratejilere daha fazla odaklanılabilir.



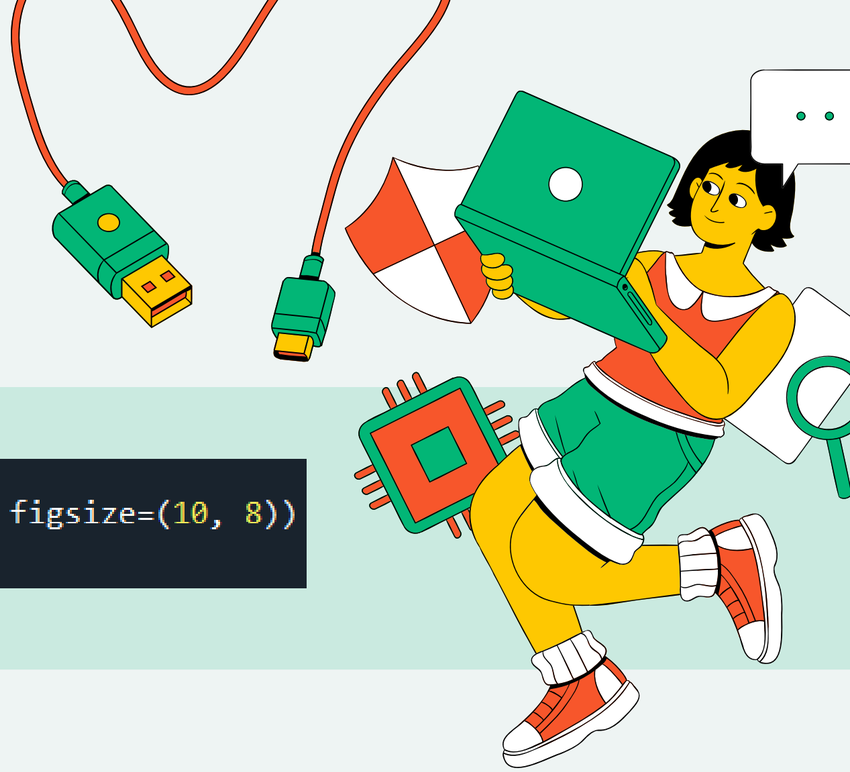
ÖDEME YÖNTEMİ-MÜŞTERİ KAYBI

```
# Tercih edilen ödeme yöntemi ve müşteri kaybı
sns.countplot(data=data, x='PreferredPaymentMode', hue='Churn')
plt.title("Preferred Payment Mode vs Churn")
plt.xticks(rotation=45)
plt.show()
```

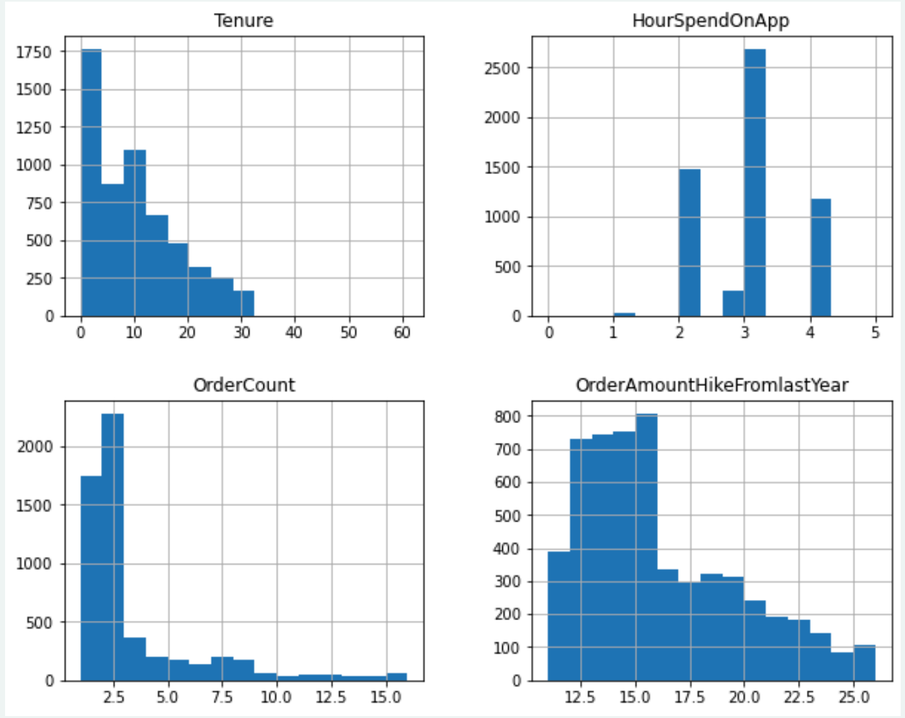


- Debit Card ve Credit Card kullanıcıları arasında müşteri sadakati daha yüksektir.
- COD kullanıcıları için müşteri kaybının daha yüksek olması, bu yöntemin daha az güvenilir ya da memnuniyet yaratmıyor olabileceğini gösterebilir.
- E-Wallet yönteminde kaybın yüksek olması, bu ödeme yöntemiyle ilgili kullanıcı deneyiminin iyileştirilmesi gerektiğine işaret edebilir.

SAYISAL DEĞİŞKENLERİN DAĞILIM GRAFİĞİ



```
# Sayısal değişkenlerin dağılım grafiği
data[['Tenure', 'HourSpendOnApp', 'OrderCount', 'OrderAmountHikeFromlastYear']].hist(bins=15, figsize=(10, 8))
plt.show()
```



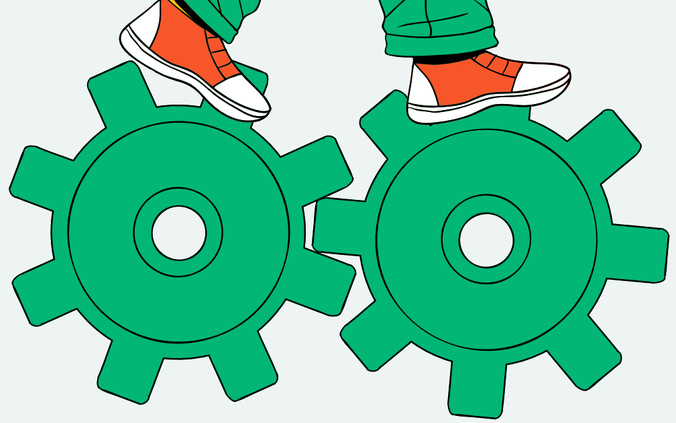
- Tenure ve OrderCount grafikleri müşteri sadakatinin düşük olduğunu ve birçok müşterinin az sayıda sipariş verdiğini işaret ediyor.
- HourSpendOnApp metriği, müşterilerin uygulamada geçirdiği sürenin nispeten odaklı olduğunu ve çoğunluğun 2–4 saat arasında yoğunlaştığını gösteriyor.
- OrderAmountHikeFromLastYear, müşterilerin geçen yıla kıyasla daha fazla harcama yapmaya eğilimli olduğunu, ancak artışların büyük çoğunluğunun sınırlı kaldığını gösteriyor.

KATEGORİK VERİLERİ ETİKETLEME

```
# Kategorik verileri etiketleme
label_encoder = LabelEncoder()
categorical_cols = ['PreferredLoginDevice', 'PreferredPaymentMode', 'Gender', 'MaritalStatus', 'PreferedOrderCat']
for col in categorical_cols:
    data[col] = label_encoder.fit_transform(data[col])
```

Bu kod, veri setindeki belirli kategorik sütunları (örneğin, PreferredLoginDevice, PreferredPaymentMode, Gender, MaritalStatus, PreferedOrderCat) etiketleme (label encoding) yöntemiyle sayısal verilere dönüştürür.

KORELASYON MATRİSİNİN OLUŞTURULMASI



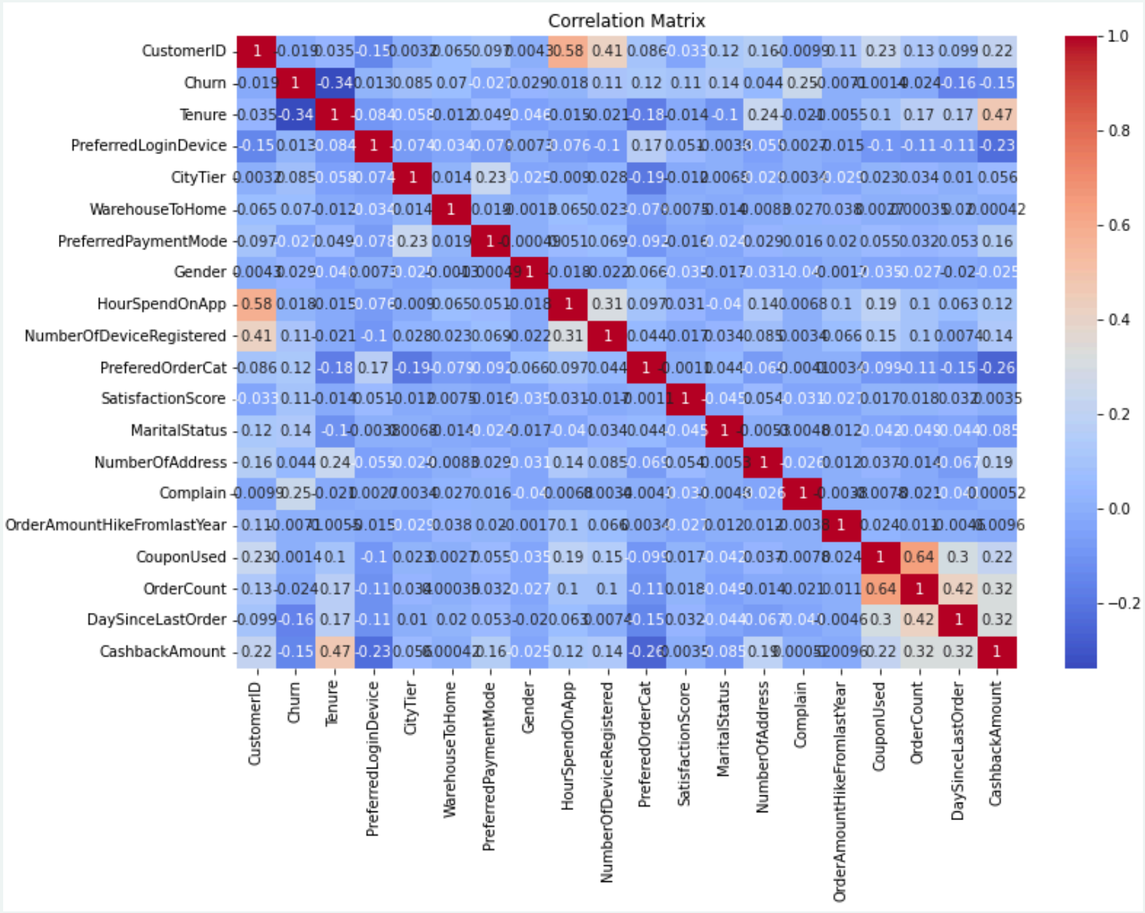
```
# Korelasyon matrisi
plt.figure(figsize=(12, 8))
sns.heatmap(data.corr(), annot=True, cmap='coolwarm')
plt.title("Correlation Matrix")
plt.show()
```

Korelasyon matrisi; değişkenler arasındaki doğrusal ilişkileri renkler ve korelasyon katsayılarıyla göstermektedir.

KORELASYON MATRİSİ

Korelasyon matrisi, bir veri setindeki değişkenler arasındaki doğrusal ilişkilerin derecesini ve yönünü gösteren bir tablodur. Korelasyon katsayısı genellikle –1 ile 1 arasında bir değer alır:

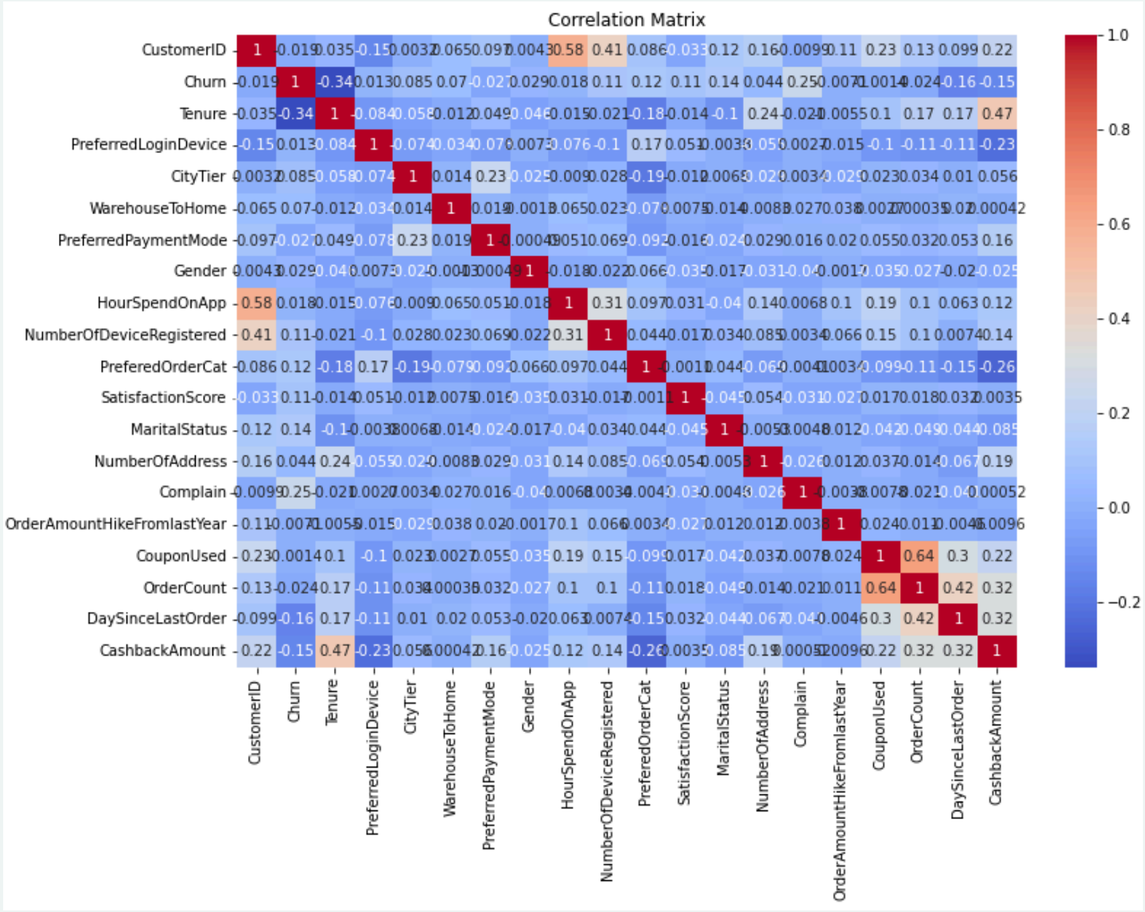
- **1'e yakın değerler:** Pozitif doğrusal ilişkiyi ifade eder. Bir değişken artarken diğeri de artar.
- **–1'e yakın değerler:** Negatif doğrusal ilişkiyi ifade eder. Bir değişken artarken diğeri azalır.
- **O'a yakın değerler:** Değişkenler arasında anlamlı bir doğrusal ilişki olmadığını gösterir.



KORELASYON MATRİSİ

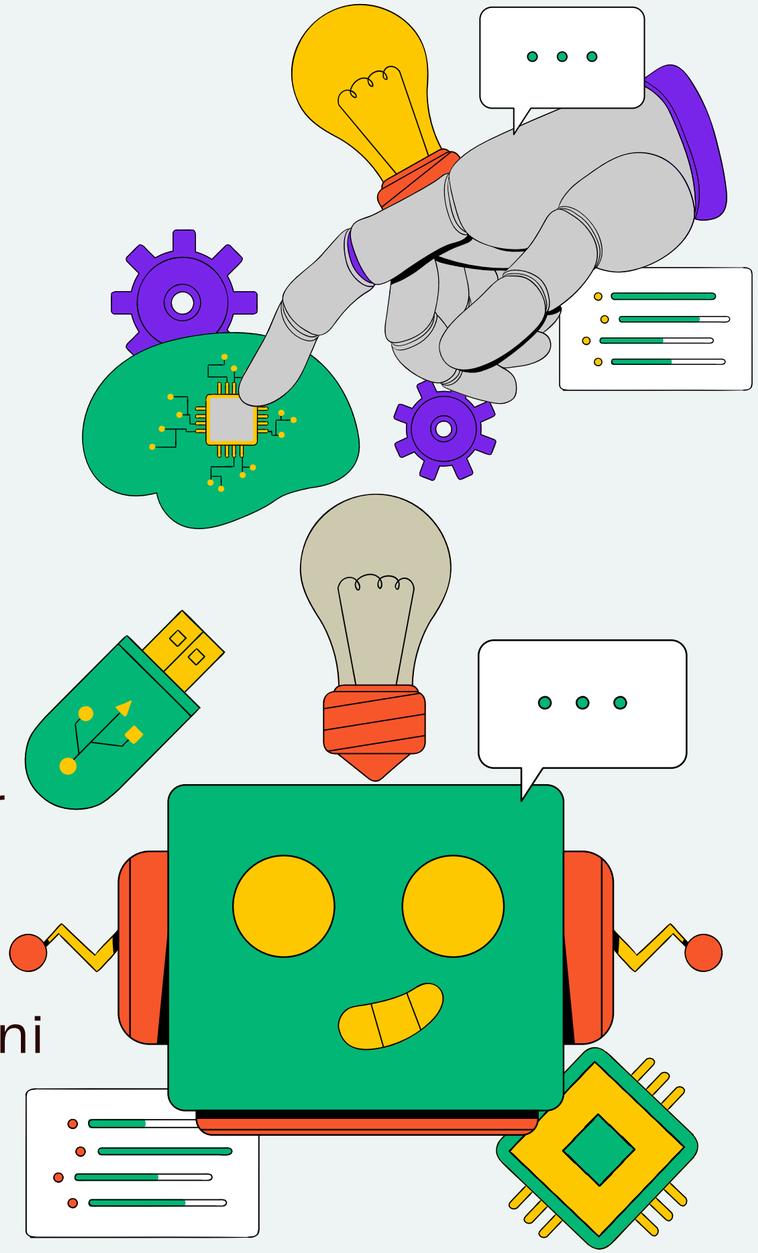
Bu grafiğe göre:

Churn ve Tenure: Daha uzun süre kalan müşterilerin kaybı azalıyor. Bu, müşteri sadakati artırmaya yönelik programların önemini gösterir.
OrderCount ve CashbackAmount: Cashback (kupon) kampanyalarının sipariş sayısını artırdığı düşünülebilir. Bu tür teşvikler daha fazla kullanılabilir.
Memnuniyet ve Şikayet: Müşteri memnuniyetini artırmak için şikayetlerin azaltılması ve çözüm süreçlerinin iyileştirilmesi önemlidir.



HEDEF VE ÖZELLİKLER

```
# Hedef ve özellikler
X = data.drop(columns=['CustomerID', 'Churn'])
y = data['Churn']
```



Bu kod, özellikler (X) ve hedef (y) değişkenlerini ayıran bir işlemi gerçekleştirir.

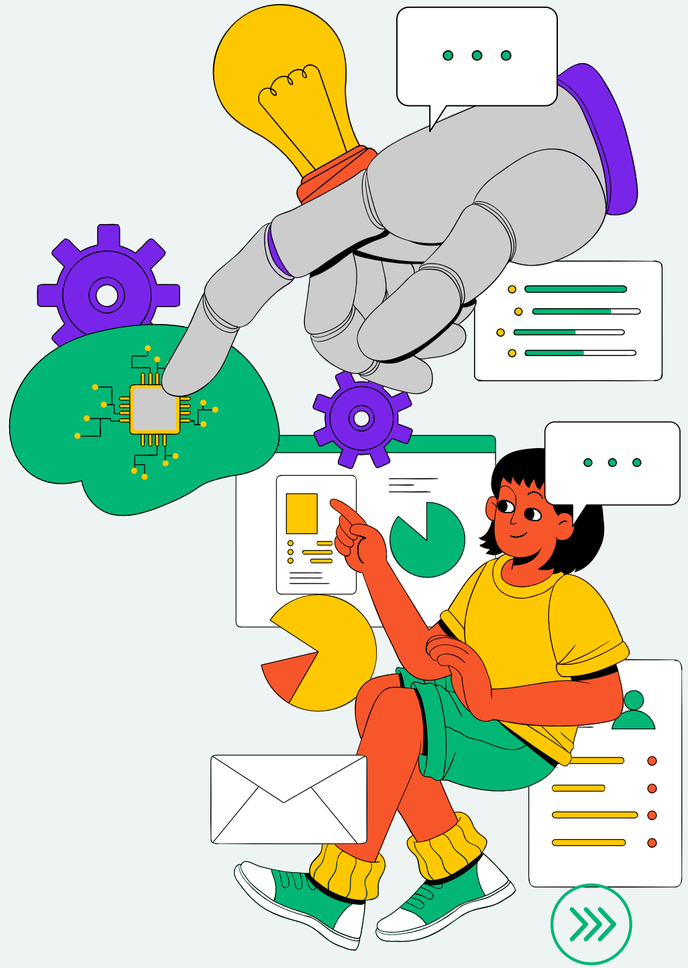
- **X:** Modelin eğitilmesi için kullanılacak olan özellikler (features) veri seti olur; burada 'CustomerID' ve 'Churn' dışındaki tüm sütunlar yer alır.
- **y:** Modelin tahmin etmeye çalışacağı hedef değişkeni (target) olur; burada 'Churn' sütunu kullanılır.

Bu ayrım, makine öğrenmesi modelinin özellikleri kullanarak hedefi tahmin etmesine olanak tanır.

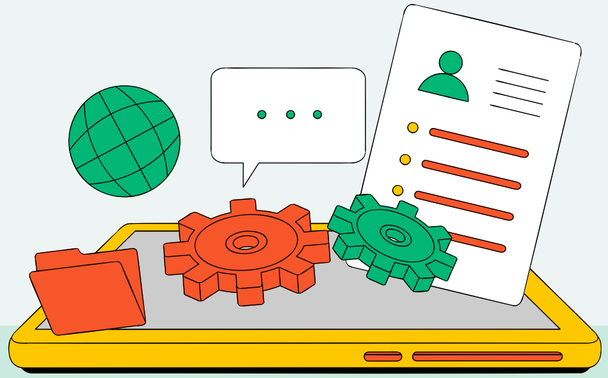
VERİYİ ÖLÇEKLEME

```
# Veriyi ölçekleme
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)
```

- **StandardScaler()** kullanılarak, her bir özelliğin ortalaması 0 ve standart sapması 1 olacak şekilde dönüştürülür.
- **fit_transform()** metodu, veriyi öğrenir ve dönüştürür, böylece algoritmaların daha doğru ve verimli çalışması sağlanır.
- Bu işlem, özellikle mesafeye dayalı algoritmalar ve bazı model türleri için gereklidir, çünkü veriler farklı ölçeklerde olduğunda modelin öğrenmesi zorlaşabilir.



VERİ SETİNİN AYRILMASI



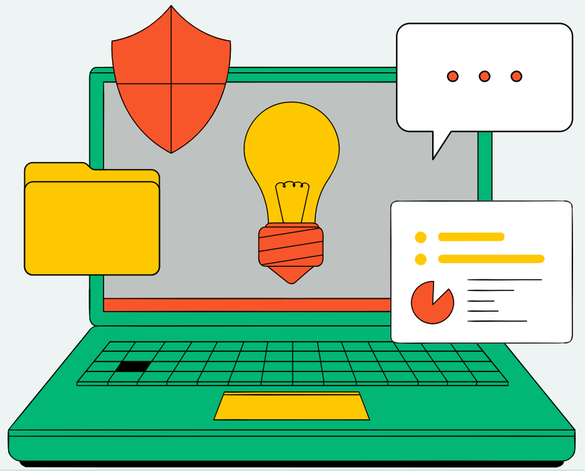
```
# Veriyi bölme
X_train, X_test, y_train, y_test = train_test_split(X_scaled, y, test_size=0.3, random_state=42, stratify=y)
```

X	DataFrame	(5630, 18)
X_scaled	Array of float64	(5630, 18)
X_test	Array of float64	(1689, 18)
X_train	Array of float64	(3941, 18)
y	Series	(5630,)
y_test	Series	(1689,)

Bu kod, veri setini eğitim ve test setlerine ayırır:

- **Eğitim seti (X_train, y_train)**, modelin eğitileceği veriyi içerir.
- **Test seti (X_test, y_test)**, modelin doğruluğunu test etmek için kullanılacak veriyi içerir.
- **test_size=0.3** parametresi, verilerin %30'unun test setine ayrılmasını sağlar.
- **stratify=y** parametresi, hedef değişkenin (y) sınıf dağılımını koruyarak veri setinin dengeli bir şekilde bölünmesini sağlar. Bu, özellikle sınıf dengesizliği olan verilerde önemlidir.

4-MODELİN EĞİTİLMESİ VE DEĞERLENDİRİLMESİ



```
# Model oluşturma ve eğitim
model = RandomForestClassifier(random_state=42)
model.fit(X_train, y_train)
```

Bu kodda, Random Forest sınıflandırma modelini oluşturuyoruz ve ardından eğitim verisiyle (X_train, y_train) modelin eğitimini başlatıyoruz:

- **RandomForestClassifier(random_state=42):** Random Forest modelini oluşturur ve rastgelelikin kontrol edilmesini sağlar.
- **model.fit(X_train, y_train)**: Modeli, eğitim verisiyle eğitir, yani modelin veriyi öğrenmesini sağlar.

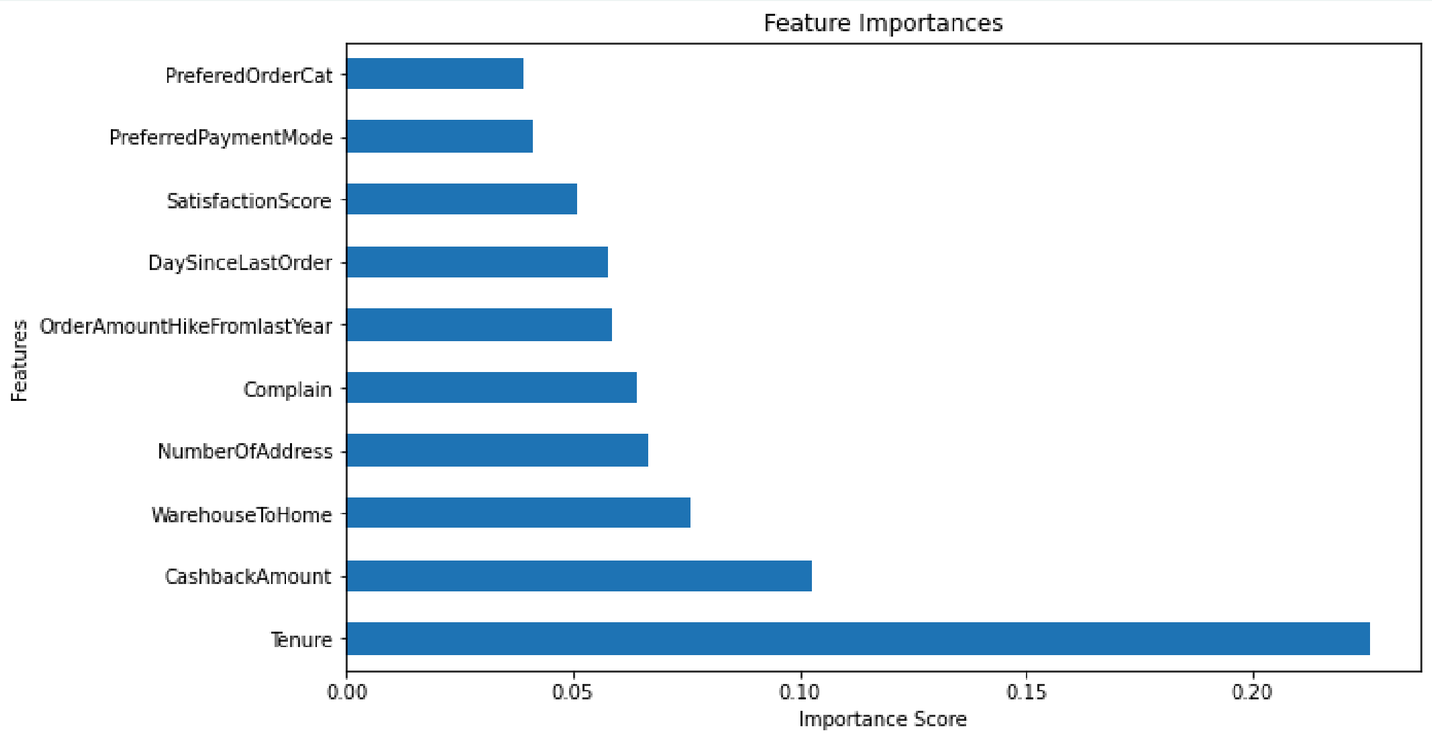
Bu aşama, modelin özellikler ile hedef arasındaki ilişkileri öğrenmesini ve bu ilişkileri daha sonra yeni veriler için tahmin yapmak amacıyla kullanmasını sağlar.

ÖNEM DÜZEYİNİN GÖRSELLEŞTİRİLMESİ

```python
# Özellik Önem Düzeyi Görselleştirme
feature_importances = pd.Series(model.feature_importances_, index=data.drop(columns=['CustomerID', 'Churn']).columns)
plt.figure(figsize=(10, 6))
feature_importances.nlargest(10).plot(kind='barh')
plt.title("Feature Importances")
plt.xlabel("Importance Score")
plt.ylabel("Features")
plt.show()
```

Bu görselleştirme, hangi özelliklerin modelin kararlarını en çok etkilediğini anlamanıza yardımcı olur ve modelin yorumlanabilirliğini artırır. Özellikle, daha düşük önem düzeyine sahip özellikler üzerinde daha fazla optimizasyon yapılabilir.

ÖZELLİK ÖNEM DÜZEYİNİN GÖRSELLEŞTİRİLMESİ



Grafik, bir makine öğrenimi modelinin özellik önem skorlarını göstermektedir. X ekseni önem skoru, Y ekseni ise modelin kullandığı özellikleri ifade ediyor.

1. En önemli özellik: "Tenure" (kullanım süresi) özelliği en yüksek önem skoruna sahip. Bu, modelin kararlarında en büyük etkiye sahip olduğunu gösteriyor.
2. Orta derecede önemli özellikler:
   - "CashbackAmount" (kupon/kampanya)
   - "WarehouseToHome" (depo-ev arası mesafe)
   - "NumberOfAddress" (adres sayısı)
3. Daha az önemli özellikler: "Complain" (şikayet), "DaySinceLastOrder" (son siparişten bu yana geçen gün sayısı), "PreferredOrderCat" (tercih edilen sipariş kategorisi) gibi diğer özellikler daha düşük önem skoru almış.

5-MODELİN UYGULANMASI

```
# Test verisi üzerinde tahmin yapma
y_pred = model.predict(X_test)
```

Bu kod, eğitim verisiyle eğitilmiş olan Random Forest modelinin, test verisi (X_test) üzerinde tahminler yapmasını sağlar.
Çıktı olarak elde edilen y_pred değerleri, modelin test verisi üzerindeki tahmin ettiği hedef değerlerdir. Bu tahminler daha sonra modelin başarısını ölçmek için kullanılabilir.

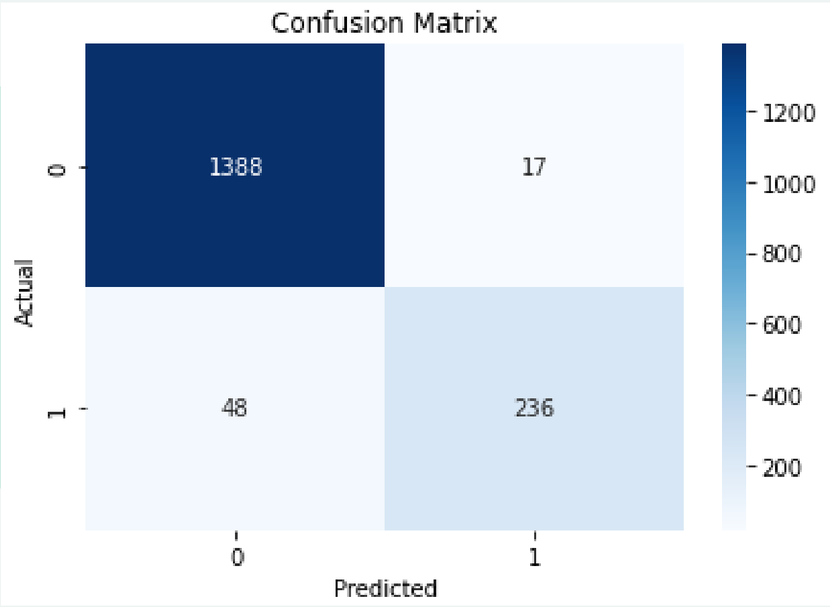
	0
0	0
1	0
2	0
3	1
4	0
5	0
6	0
7	0

CONFUSION MATRİX

```python
# Confusion Matrix
conf_matrix = confusion_matrix(y_test, y_pred)
sns.heatmap(conf_matrix, annot=True, fmt="d", cmap='Blues')
plt.title("Confusion Matrix")
plt.xlabel("Predicted")
plt.ylabel("Actual")
plt.show()
```

Bu kod, Confusion Matrix'i (Karışıklık Matrisi) hesaplar ve görselleştirir. Karışıklık matrisi, sınıflandırma modellerinin doğruluğunu değerlendirmek için yaygın olarak kullanılan bir araçtır.

Model, çoğu durumda doğru tahminler yapıyor, ancak yanlış negatif oranı dikkate alındığında, sınıf 1 için daha fazla iyileştirme yapılması gerekebilir. Özellikle, 48 yanlış negatif kayıpları, bu sınıfın yanlış tahminlerine sebep olabilir. Genel olarak, modelin performansı oldukça iyi gözüküyor.



RAPOR VE METRİKLER

```
# Rapor ve metrikler
print(classification_report(y_test, y_pred))
```

Bu satır, sınıflandırma raporunu yazdırır.

,

- "Churn = 0" (müşteri kaybı olmayan) sınıfı için model çok başarılı: Precision, recall ve F1-score çok yüksek.
- "Churn = 1" (müşteri kaybı olan) sınıfı için model yine iyi, ancak recall biraz daha düşük (%85). Bu, modelin bazı müşteri kaybı olanları kaçırdığı anlamına gelir.
- Genel Performans: Modelin genel doğruluğu çok iyi (%96), ancak modelin başarısı sınıf dengesine bağlıdır. "Churn = 0" sınıfı baskın olduğu için model bu sınıfı çok iyi tahmin etmiş. "Churn = 1" sınıfı daha az örnek içerdiği için modelin recall'u biraz daha düşük kalmış.

```
...:
...: print(classification_report(y_test, y_pred))
              precision    recall  f1-score   support

           0       0.97      0.99      0.98      1405
           1       0.93      0.83      0.88       284

    accuracy                           0.96      1689
   macro avg       0.95      0.91      0.93      1689
weighted avg       0.96      0.96      0.96      1689
```

ROC AUC SKORU

```
# ROC AUC Skoru
roc_auc = roc_auc_score(y_test, model.predict_proba(X_test)[:, 1])
print(f"ROC AUC Score: {roc_auc}")
```

```
ROC AUC Score: 0.9892737206155079
```

- **roc_auc_score():** Bu fonksiyon, Receiver Operating Characteristic (ROC) eğrisinin altındaki alanı (AUC) hesaplar. AUC, modelin tahminlerinin doğruluğunu gösteren bir metriktir. AUC değeri 0 ile 1 arasında değişir:
  - **AUC = 1:** Model mükemmel, her zaman doğru tahmin yapıyor.
  - **AUC = 0.5:** Model rastgele tahmin yapıyor (yani, modelin hiçbir tahmin gücü yok).
  - **AUC < 0.5**: Model, ters yönde tahmin yapıyor (yani, model aslında yanlış tahminler yapıyor).

- **ROC Eğrisi:**

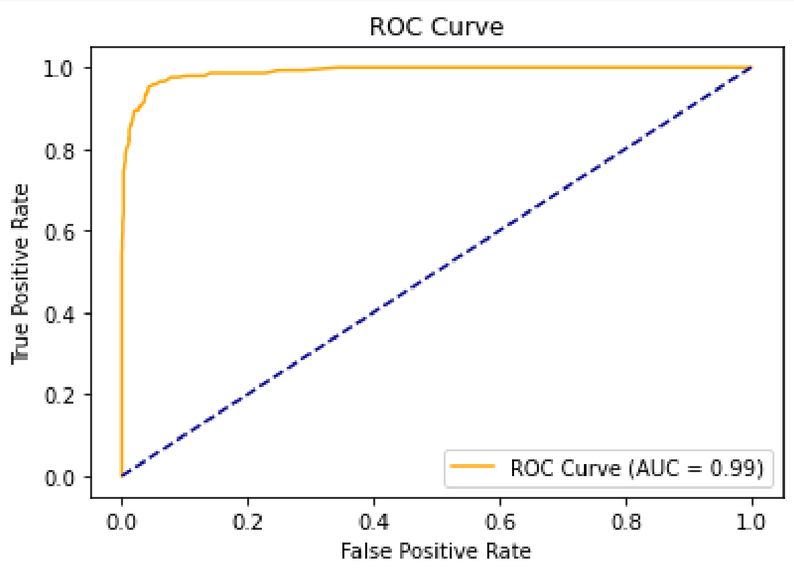
Eğri, gerçek pozitif oranını (TPR) yanlış pozitif oranına (FPR) karşı gösterir. Grafikteki sarı eğri, modelin sıralama yeteneğinin yüksek olduğunu gösteriyor.

- **AUC Değeri:**

AUC değeri 0.99, bu da modelin mükemmel bir ayrım yeteneğine sahip olduğunu gösteriyor. 1.0'a ne kadar yakınsa, modelin o kadar iyi performans gösterdiği anlamına gelir.

ROC EĞRİSİ

```
# ROC Eğrisi
fpr, tpr, thresholds = roc_curve(y_test, model.predict_proba(X_test)[:, 1])
plt.plot(fpr, tpr, color='orange', label=f'ROC Curve (AUC = {roc_auc:.2f})')
plt.plot([0, 1], [0, 1], color='darkblue', linestyle='--')
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title('ROC Curve')
plt.legend()
plt.show()
```



Eğrinin üst kısımda ve sola çok yakın olması, modelin yanlış pozitif oranını düşük tutarak yüksek bir doğru pozitif oranı elde ettiğini gösterir. Bu, modelin sınıf 1'i (pozitif sınıf) yüksek bir başarıyla doğru tahmin etme yeteneğine sahip olduğu anlamına gelir.

Genel olarak, modelin sınıflandırma başarısı oldukça yüksektir ve AUC değeri, modelin olumlu sonuçları doğru bir şekilde tanıma yeteneğinin mükemmel olduğunu göstermektedir. Ancak, yanlış pozitif oranın da göz önünde bulundurulması gereken bir faktör olduğunu unutulmamalıdır.

MODELİN BAŞARISININ DEĞERLENDİRİLMESİ

```python
# Modelin önemli metrikleri
from sklearn.metrics import f1_score, precision_score, recall_score

f1 = f1_score(y_test, y_pred)
precision = precision_score(y_test, y_pred)
recall = recall_score(y_test, y_pred)

print(f"F1 Score: {f1:.4f}")
print(f"Precision: {precision:.4f}")
print(f"Recall: {recall:.4f}")
```

```
F1 Score: 0.8790
Precision: 0.9328
Recall: 0.8310
```

MODELİN BAŞARISININ DEĞERLENDİRİLMESİ

1. F1 Skoru: 0.8889
0.8889, oldukça iyi bir F1 skoru olduğunu gösteriyor. Bu, modelin hem precision (kesinlik) hem de recall (duyarlılık) açısından iyi bir denge sağladığını ve genel olarak yüksek doğrulukla tahmin yaptığını gösterir. Yani model, hem yanlış pozitifleri hem de yanlış negatifleri düşük seviyelerde tutarak sağlam bir performans sergiliyor.

2. Precision: 0.9375
Yüksek bir precision değeri, modelin yanlış pozitif oranını düşük tuttuğunu gösterir. Başka bir deyişle, model Churn = 1 tahmininde genellikle doğru kararlar veriyor ve müşterileri yanlış bir şekilde kaybediyor (yanlış pozitif) oranı oldukça düşük.

3. Recall: 0.8451
Bu değer, modelin gerçek kayıp müşterilerin %15'ini kaçırdığı anlamına gelir (yanlış negatif). Bu, modelin daha fazla müşteri kaybını doğru tespit etmek için geliştirilebileceği anlamına gelebilir. Ancak, bu oran yine de oldukça iyi bir değerdir.

MODELİN BAŞARISININ DEĞERLENDİRİLMESİ

- Modelin precision değeri çok yüksek (0.9375), bu da modelin yanlış pozitifleri çok az yaparak doğru tahminler yaptığını gösterir.
- Recall değeri ise 0.8451 ile oldukça iyi, ancak daha fazla müşteri kaybını doğru tahmin etme kapasitesini artırmak adına bu değerin yükseltilmesi hedeflenebilir.
- F1 Skoru 0.8889, bu da hem precision hem de recall değerlerinin oldukça iyi bir dengeye sahip olduğunu gösterir.

DİNLEDİĞİNİZ İÇİN TEŞEKKÜRLER