# MultinetUp B2B Müşteri Kayıp Adayı Tahmini Projesi Dokümantasyonu

İçindekiler

[MultinetUp B2B Müşteri Kayıp Adayı Tahmini Projesi Dokümantasyonu 1](#_Toc181621065)

[1. Proje Özeti 2](#_Toc181621066)

[1.1 Amaç 2](#_Toc181621067)

[1.2 Kapsam 2](#_Toc181621068)

[1.3 Hedef Kullanıcılar 2](#_Toc181621069)

[2. Teknik Mimari 2](#_Toc181621070)

[2.1 Teknoloji Yığını 2](#_Toc181621071)

[3. Özellikler ve Fonksiyonlar 3](#_Toc181621072)

[3.1 Temel Özellikler 3](#_Toc181621073)

[3.2 Churn Hesaplama Kriterleri 3](#_Toc181621074)

[4. Veri İşleme ve Model 3](#_Toc181621075)

[4.1 Veri Önişleme 3](#_Toc181621076)

[4.2 Özellik Mühendisliği 3](#_Toc181621077)

[4.3 Model Özellikleri 4](#_Toc181621078)

[5. Kullanıcı Arayüzü 4](#_Toc181621079)

[5.1 Ana Ekran Bileşenleri 4](#_Toc181621080)

[5.2 Görselleştirmeler 4](#_Toc181621081)

[6. Kurulum ve Çalıştırma 4](#_Toc181621082)

[6.1 Gereksinimler 4](#_Toc181621083)

[6.2 Çevre Değişkenleri 4](#_Toc181621084)

[7. Güvenlik ve Performans 5](#_Toc181621085)

[7.1 Güvenlik Önlemleri 5](#_Toc181621086)

[7.2 Performans Optimizasyonları 5](#_Toc181621087)

[8. Test ve Kalite Kontrol 5](#_Toc181621088)

[8.1 Test Senaryoları 5](#_Toc181621089)

[9. Bakım ve Geliştirme 5](#_Toc181621090)

[9.1 Düzenli Bakım Görevleri 5](#_Toc181621091)

[9.2 Gelecek Geliştirmeler 6](#_Toc181621092)

[10. Uygulama Ekran Görüntüleri 6](#_Toc181621093)

## 1. Proje Özeti

### 1.1 Amaç

MultinetUp B2B Müşteri Kayıp Adayı Tahmini projesi şirketin kurumsal müşterilerinin churn (kayıp) riskini önceden tespit etmek ve proaktif önlemler alabilmek amacıyla geliştirilmiş bir makine öğrenmesi uygulamasıdır.

### 1.2 Kapsam

* Müşteri davranış analizi
* Churn risk tahmini
* Trend analizi ve görselleştirme
* Detaylı müşteri raporlama
* Real-time veri analizi

### 1.3 Hedef Kullanıcılar

* Satış ekipleri
* Müşteri ilişkileri yöneticileri
* İş analisti ekibi
* Üst yönetim

## 2. Teknik Mimari

### 2.1 Teknoloji Yığını

* **Backend:** Python 3.8.10
* **Frontend:** Streamlit
* **Veritabanı:** MS SQL Server
* **Veri Analizi:** Pandas, NumPy
* **Görselleştirme:** Altair
* **Loglama:** Loguru
* **Modelleme:** Scikit-learn (joblib ile serileştirilmiş), lightgbm

**2.2 Sistem Mimarisi**

ChurGuard/
├── app.py # Ana uygulama ve UI
├── churn\_functions.py # İş mantığı ve model fonksiyonları
├── models/ # Model dosyaları
│ ├── churn\_model\_\*.pkl
│ └── feature\_info\_\*.json
├── .env # Konfigürasyon
└── requirements.txt # Bağımlılıklar

## 3. Özellikler ve Fonksiyonlar

### 3.1 Temel Özellikler

* CRM numarası ve ürün grubu bazında müşteri analizi
* Gerçek zamanlı churn risk hesaplama
* Görsel trend analizi
* Detaylı müşteri metrikleri
* Özelleştirilmiş raporlama

### 3.2 Churn Hesaplama Kriterleri

1. Son 3 ay inaktiflik
2. Net ciro düşüşü (configüre edilebilir eşik)
3. KAG (Kart/Araç/Geceleme) sayısı düşüşü
4. Müşteri memnuniyet puanı analizi

## 4. Veri İşleme ve Model

### 4.1 Veri Önişleme

def preprocess\_data(df: pd.DataFrame):

 *# Sütun standardizasyonu*

 *# Veri tipi dönüşümleri*

 *# Eksik veri işleme*

 *# Kategorik değişken kodlama*

### 4.2 Özellik Mühendisliği

* Zaman bazlı özellikler
	+ Son aktivite tarihi
	+ İnaktif gün sayısı
	+ Kontrat süresi
* Finansal özellikler
	+ Net ciro değişimi
	+ İskonto oranları
	+ İade oranları
* Davranışsal özellikler
	+ Şikayet oranı
	+ Müşteri memnuniyeti
	+ Çağrı merkezi etkileşimleri

### 4.3 Model Özellikleri

* Kayıp olasılığı tahmini
* Feature importance analizi
* Model performans metrikleri
* Periyodik model güncelleme

## 5. Kullanıcı Arayüzü

### 5.1 Ana Ekran Bileşenleri

* Müşteri bilgi girişi (CRM ve Ürün Grubu)
* Trend grafikleri
* Risk göstergeleri
* Detaylı metrikler tablosu

### 5.2 Görselleştirmeler

* Net ciro trend grafiği
* KAG trend grafiği
* Risk gösterge paneli
* Metrik kartları

## 6. Kurulum ve Çalıştırma

### 6.1 Gereksinimler

pip install -r requirements.txt

### 6.2 Çevre Değişkenleri

DB\_HOST=<database\_host>

DB\_PORT=<port>

DB\_NAME=<database\_name>

DB\_USERNAME=<username>

DB\_PASSWORD=<password>

EPSILON=1e-10

INACTIVITY\_DAYS=90

REVENUE\_DECREASE=0.1

TRANSACTION\_DECREASE=0.1

**6.3 Çalıştırma**

streamlit run app.py

## 7. Güvenlik ve Performans

### 7.1 Güvenlik Önlemleri

* Veritabanı bağlantı şifreleme
* SQL injection koruması
* Hata yönetimi ve loglama
* Kullanıcı giriş doğrulama

### 7.2 Performans Optimizasyonları

* Veritabanı sorgu optimizasyonu
* Önbellek kullanımı
* Lazy loading
* Batch processing

## 8. Test ve Kalite Kontrol

### 8.1 Test Senaryoları

* Birim testleri
* Entegrasyon testleri
* UI testleri
* Performans testleri

## 9. Bakım ve Geliştirme

### 9.1 Düzenli Bakım Görevleri

* Model yeniden eğitimi
* Veritabanı bakımı
* Log temizleme
* Performans optimizasyonu

### 9.2 Gelecek Geliştirmeler

* Özelleştirilebilir dashboard
* Otomatik müşteri segmentasyonu
* API entegrasyonu
* Mobil uygulama desteği
* Çoklu dil desteği

## 10. Uygulama Ekran Görüntüleri



