İSTANBUL TOPKAPI ÜNİVERSİTESİ

Veri Madenciliği

Tüketici Şikâyetlerinden Sorun Türü Tahmini: CFPB Consumer Complaints Verisiyle Açıklanabilir Metin Madenciliği Uygulaması

AVENGERS

Barış Yasin ŞAHİN	22040301029	barisyasinsahin@stu.topkapi.edu.tr
Musa ULUĞ	21040301044	musaulug @stu.topkapi.edu.tr
Salih İmran BÜKER	22040301062	salihimranbuker @stu.topkapi.edu.tr
Yusuf YENİGÜN	22040301052	yusufyenigun @stu.topkapi.edu.tr

https://github.com/barisYasi/Veri_Madencilik_proje

https://github.com/mUsaulug/Veri_Madenciligi_Proje/tree/main

https://github.com/mersieS/veri_madenciligi_proje

https://github.com/YusufYenigun/Veri-Madenciligi-Vize-.git

1) Problem Tanımı

İş / Bilimsel Soru
Consumer Financial Protection Bureau (CFPB) tarafından yayımlanan tüketici finans şikâyetleri veri setinde, her şikâyet kaydı için ürün türü (mortgage, kredi kartı, borç tahsili vb.), şikâyet konusu (issue/sub_issue), şirketin verdiği yanıt türü, yanıtın zamanında olup olmaması, şikâyet metninin uzunluğu ve yanıt süresi gibi yapısal özelliklere bakarak, tüketicinin bu şikâyeti resmî olarak "dispute" (itiraz) edip etmeyeceğini önceden tahmin etmeyi hedefliyoruz.

İş sorusunu şöyle özetliyoruz:

"Verilen yapısal özellikler kullanılarak, bir şikâyetin consumer_disputed? = Yes olup olmayacağı önceden tahmin edilebilir mi? Bu tahmin, müşteri ilişkileri ve şikâyet yönetimi süreçlerini iyileştirmek için anlamlı bir sinyal üretebilir mi?"

Görev Türü
- Görev türü: İkili Sınıflandırma (Binary Classification)
- Etiketler: Yes ve No
- Pozitif sınıf: Yes (müşterinin firmaya resmî itiraz yaptığı durum)

Hedef Değişken(ler)
- Hedef değişken: consumer_disputed?
- Değerler: "Yes" ve "No"
- Pozitif sınıf: Yes (müşterinin firmaya resmî itiraz yaptığı durum)
- Etki alanı: Tüketici finansı, şikâyet yönetimi, müşteri memnuniyeti ve risk yönetimi

Başarı Kriterleri
Veri seti dengesiz (≈%80 No / %20 Yes) olduğu için yalnızca accuracy yeterli değil. Başarı kriterlerimizi şöyle tanımlıyoruz:

Baseline (Dummy model - herkese "No" tahmin eden):

- ROC AUC ≈ 0,50
- Recall(Yes) = 0,00

Vize aşaması hedefleri:

- Baseline'a göre ROC AUC > 0,60 elde etmek
- Pozitif sınıf için Recall(Yes) ≥ 0,60 seviyesine çıkmak

Final aşaması hedefleri (ileriki plan):

- ROC AUC'yi 0,65+ seviyesine taşımak
- Recall(Yes) / Precision(Yes) arasında anlamlı bir denge kurmak (F1)

2) Proje Yönetimi

Kilometre Taşları ve Zaman Çizelgesi
Proje zaman çizelgesini haftalık kilometre taşları ile aşağıdaki gibi planladık:

1. Hafta (20–27 Ekim):

- Veri seti arayışı ve proje konusunun netleştirilmesi
- Kaggle üzerinden "US Consumer Finance Complaints" veri setinin seçimi
- Sorumlu: Tüm grup

2. Hafta (3–10 Kasım):

- Veri ön incelemesi (df.info, eksik değer analizi, hedef dağılımı)
- Temel EDA: ürün dağılımı, hedef değişken dağılımı, tarih alanlarının kontrolü
- Sorumlu: Barış (ortak kod iskeleti), tüm grup gözden geçirme

3. Hafta (11–17 Kasım):

- Ortak veri ön işleme pipeline'ının tasarlanması (imputasyon, One-Hot, feature engineering)
- Ortak train/test split yapısının belirlenmesi
- Sorumlu: Barış

4. Hafta (18–24 Kasım):

- Her grup üyesinin kendi base modellerini geliştirmesi (en az 2 model)
- Farklı feature set ve feature selection tekniklerinin denenmesi
- Sorumlu: Her üye kendi notebook'undan sorumlu

5. Hafta (25–30 Kasım – Vize Teslim):

- Base model sonuçlarının tablo halinde toplanması
- Bu proje özeti (Project Outline Report) dokümanının yazılması
- ALMS'e rapor + bireysel Jupyter Notebook dosyalarının yüklenmesi
- Sınıfta imza ve fiziksel rapor teslimi

Final aşaması için (vize sonrası): Daha ileri model geliştirme, tuning ve metin tabanlı özellikler (TF-IDF) ile genişletme planlanmaktadır.

Roller ve Sorumluluklar

Barış Yasin Şahin

- Temel feature set'in belirlenmesi
- DummyClassifier, Logistic Regression ve Decision Tree base modelleri
- Sınıf dengesizliği stratejilerinin (class_weight) denenmesi
- SelectKBest(f_classif) ile feature selection pilotu
- Ortak veri ön işleme pipeline'ının tasarımı

Yusuf Yenigün

- Genişletilmiş feature set (sub_product, sub_issue, submitted_via, state vb.) tasarımı
- Decision Tree ve Logistic Regression modellerinin bu geniş set üzerinde denenmesi
- SelectFromModel(DecisionTree) ile embedded feature selection

Musa Uluğ

- Alternatif feature set (özellikle product, issue, submitted_via, complaint_length_words vb.)
- Complement Naive Bayes ve KNN base modellerinin geliştirilmesi
- SelectKBest(chi^2) ile kategorik odaklı feature selection
- KNN'in büyük veri setlerindeki hesaplama maliyetini azaltmak için alt küme stratejisi

Salih İmran Büker

- Ortak feature set üzerinde LinearSVC ve Logistic Regression (C=0,5) modelleri
- mutual_info_classif tabanlı SelectKBest ile k=50 ve k=200 senaryoları
- Farklı feature selection seviyelerinin performansa etkisini karşılaştırma

3) İlgili Çalışmalar (Mini Literatür İncelemesi)

Bu proje, Kaggle ve akademik literatürde sıklıkla kullanılan CFPB şikâyet veri seti üzerinde, özellikle dispute davranışının tahmini ve sınıf dengesizliği temalarını birleştirmektedir.

Kısa Literatür Özeti

CFPB veri seti üzerinde yapılan Kaggle çalışmaları:

- Çoğunlukla şikâyet metninin sınıflandırılması (NLP odaklı)
- Ürün veya firma bazlı şikâyet yoğunluğunun analizi
- Temel gradient boosting / random forest modelleri ile skorlama

Sınıf dengesizliği literatürü:

- Fraud detection, churn prediction gibi problemlerde class_weight, oversampling (SMOTE), undersampling tekniklerinin etkinliği

Projemizin Farklılıkları

Bizim projemiz, literatürden şu yönleriyle ayrılmayı hedefliyor:

1. Yapısal odak: İlk aşamada yalnızca yapısal sütunlar (product, issue, response, zamanlama, metin uzunluğu) üzerine odaklanıp consumer_disputed? etiketi için açıklanabilir ve basit base modeller kurmak.
2. Sistematik karşılaştırma: Her grup üyesinin farklı feature selection tekniği ve farklı model ailesi kullanarak, hangi yaklaşımın sınıf dengesizliği altında daha dengeli performans verdiğini sistematik biçimde kıyaslaması.
3. Pilot yaklaşımı: Vize aşamasında elde edilen sonuçları, final aşamasında metin tabanlı özelliklerle (TF-IDF vb.) genişletmek üzere "pilot deney" olarak kullanmak.

Not: Final raporda 2–4 kaynağı IEEE formatında referanslandırmayı planlıyoruz (CFPB veri tabanı, Kaggle veri seti sayfası, ilgili blog/makale).

4) Veri Tanımı ve Yönetimi

Veri Seti

- Adı: US Consumer Finance Complaints
- Kaynak:
- CFPB'nin kamuya açık şikâyet veri tabanı
- Kaggle üzerinden erişilen türetilmiş veri seti (kaggle.com/datasets/kaggle/us-consumer-finance-complaints)
- Kullanım hakkı: Eğitim/araştırma amaçlı açık veri olarak paylaşılmaktadır. Nihai lisans detayları, Kaggle sayfası ve CFPB kullanım şartları üzerinden kontrol edilecektir.

Veri Şeması

- Satır sayısı: 555.957
- Sütun sayısı: 18
- Sütun türleri:
- Çoğu sütun object (kategorik/metinsel)
- complaint_id: int64
- date_received ve date_sent_to_company: tarih tipine dönüştürülmektedir

Önemli sütunlar:

Sütun Türü	Sütun Adları
Ürün ve konu	product, sub_product, issue, sub_issue
Müşteri açıklaması	consumer_complaint_narrative
Firma yanıtı	company_response_to_consumer, company_public_response, timely_response
Süreç bilgileri	date_received, date_sent_to_company, submitted_via
Hedef değişken	consumer_disputed?

Boyut ve Sınıf Dengesi

- Toplam gözlem: 555.957
- Hedef dağılımı:
- No: 443.823 (%79,83)
- Yes: 112.134 (%20,17)

Veri seti belirgin şekilde sınıf dengesiz; bu durum modelleme ve değerlendirme aşamalarında özel olarak ele alınacaktır.

Veri Erişim Planı

- Veri seti, Kaggle üzerinden CSV dosyası olarak indirilip her grup üyesinin yerel ortamında `consumer_complaints.csv` adıyla saklanmaktadır.
- İsteğe bağlı olarak GitHub reposunda küçük bir örnek subset (sample.csv) tutulacaktır.
- Veri güncellenmeyecektir; proje boyunca aynı snapshot kullanılacaktır.

Etik, Gizlilik, Önyargı

- Veri, CFPB tarafından önceden anonimleştirilmiş ve kamuya açık hale getirilmiş durumdadır.
- Yine de state ve zipcode gibi sütunlar dolaylı olarak sosyo-ekonomik ve bölgesel farkları yansıtabilir; bu nedenle model sonuçlarının belirli alt gruplar üzerinde sistematik önyargı üretip üretmediği final aşamasında değerlendirilecektir.
- Proje, yalnızca eğitim/araştırma amaçlıdır; ticari veya gerçek operasyonel karar alma sürecinde kullanılmayacaktır.

5) Keşifsel Veri Analizi (Exploratory Data Analysis)

Veri Kalitesi Kontrolleri
Genel inceleme:

- `df.info()` ile her sütunun tipleri ve non-null sayıları incelenmiştir.
- Veri setinin boyutu: 555.957 satır × 18 sütun

Eksik değer analizi:

- `consumer_complaint_narrative`, `tags`, `company_public_response`, `consumer_consent_provided` gibi sütunlarda çok yüksek oranda eksik değer bulunmaktadır.
- `sub_product` ve `sub_issue` sütunlarında sırasıyla yaklaşık %28 ve %62 civarında eksik değer vardır.
- `state` ve `zipcode` sütunlarında ise eksik oranı %1'in altındadır.

Yinelenen kayıtlar ve aşırı uç değerler:

- Vize aşamasında özellikle `response_days` için aşırı uçlar incelenmiş, negatif değerler 0'a kliplenmiştir.
- Tam duplicate kayıt kontrolü final aşamasında yapılacaktır.

Dağılımlar ve Denge
Hedef değişken dağılımı:

- Bar grafikleri ile sınıf dengesizliği görselleştirilmiştir (%80 No / %20 Yes)

Product sütununda en çok şikâyet gelen ürünler:

4. Mortgage
5. Debt collection
6. Credit reporting
7. Credit card
8. Bank account or service

Response_days değişkeni:

- Histogram çizilerek yanıt süresinin çoğunlukla 0–4 gün aralığında kümelendiği gözlemlenmiştir.
- Az sayıda da olsa 100+ gün süren yanıtlar bulunmaktadır.

Özellik–Hedef İlişkileri (Plan)
Vize aşamasında temel dağılımlar incelenmiş, final aşamasında ise şunlar analiz edilecektir:

- `product × consumer_disputed?` oranları
- `company_response_to_consumer` kategorilerine göre dispute oranları
- `response_days` ve `complaint_length_words` için kutu grafikleri (dispute olup olmamasına göre)

Görselleştirme Planı
Vize aşamasında kullanılanlar:

- Hedef dağılımı için bar grafikleri
- Ürün, issue ve yanıt türleri için bar grafikleri
- response_days ve complaint_length_words için histogram / boxplot

Final aşamasında planlanlar:

- Karışıklık matrisleri
- ROC eğrileri
- Feature importance grafikleri

6) Veri Hazırlama Planı

Temizleme

9. Tarih sütunları:

- `date_received`, `date_sent_to_company` sütunlarının datetime tipine dönüştürülmesi
- Parse edilemeyen değerlerin NaT yapılması

10. Response_days türetilirken:

- Negatif değerlerin 0'a çekilmesi
- Uç değerlerin (ör. 365+ gün) gerekirse winsorization veya log-transform ile dengelenmesi (final aşamasında düşünülebilir)

11. Duplicate kayıtlar:

- Gerekirse duplicate kayıtların tespit edilip kaldırılması

İmputasyon Stratejisi

Sayısal sütunlar (complaint_length_words, response_days):

- Eksik değerleri median ile doldurma

Kategorik sütunlar (product, issue, company_response_to_consumer, timely_response, genişletilmiş set için sub_product, sub_issue, submitted_via, state):

- Eksik değerleri en sık görülen kategori ile doldurma

Önemli not: İmputasyon işlemleri, sızıntıyı önlemek için mutlaka Pipeline içinde ve yalnızca train verisi üzerinden fit edilecektir.

Dönüşümler

Kategorik değişkenler:

- OneHotEncoder(handle_unknown="ignore") ile One-Hot Encoding

Sayısal değişkenler:

- Vize aşamasında ölçeklendirme yapılmamıştır
- Final aşamasında Logistic Regression / LinearSVC için StandardScaler eklenmesi değerlendirilecektir

Pipeline yapısı:

- Tüm dönüşümler ColumnTransformer içinde tanımlanmış
- Her model için tekrar kullanılabilen preprocessor objesi ile uygulanmaktadır

Özellik Mühendisliği

Şu anda yapılan özellikler:

Şu anda yapılan / planlanan özellik mühendisliği adımları:

- response_days: date_sent_to_company - date_received farkından türetilen yanıt süresi.
- complaint_length_words ve complaint_length_chars: Şikâyet metninden türetilen basit metin uzunluğu ölçüleri.
- Final aşamasında potansiyel ek özellikler:
    - Yoğun saat/gün bilgisi (ör. hafta sonu şikâyetleri için farklı davranış)
    - Ürün/kategori kombinasyonlarının frekansına dayalı etkileşim özellikleri.

Final aşamasında potansiyel ek özellikler:

- Yoğun saat/gün bilgisi (ör. hafta sonu şikâyetleri için farklı davranış)
- Ürün/kategori kombinasyonlarının frekansına dayalı etkileşim özellikleri

Özellik Seçimi ve Boyut İndirgeme

Her grup üyesi farklı feature selection yaklaşımı denemektedir:

Filter yöntemleri:

- Barış: SelectKBest(f_classif, k=100)
- Musa: SelectKBest(chi², k=50)
- Salih: SelectKBest(mutual_info_classif, k=50 ve k=200)

Embedded yöntemler:

- Yusuf: SelectFromModel(DecisionTreeClassifier, threshold="median")

Not: Boyut indirgeme için PCA benzeri yöntemler şu aşamada zorunlu görülmemektedir; ancak One-Hot sonrası feature uzayı fazla büyürse final aşamasında değerlendirilebilir.

7) Modelleme Planı

Baseline Modeller
Dumb Baseline (Referans Model)

Model: DummyClassifier(strategy="most_frequent")

Açıklama: Tüm gözlemlere "No" tahmini yaparak, neden yalnızca accuracy'ye bakmanın sorunlu olduğunu göstermek için referans modeli

Beklenen performans:

- Accuracy ≈ 0,80
- ROC AUC = 0,50
- Recall(Yes) = 0,0

Basit Baseline

Model: Sınıf dengesizliği dikkate alınmadan eğitilen Logistic Regression

Açıklama: class_weight parametresi kullanmayan temel model

Beklenen performans:

- Accuracy ≈ 0,80
- Pozitif sınıfı neredeyse hiç yakalamama

Amaç: Bu iki baseline, "class_weight kullanan" modellerle karşılaştırma için temel alınacaktır.

Model Ailesi	Örnekler	Kullanım Gerekçesi	Sorumlu Kişi
Doğrusal Modeller	LogisticRegression, LinearSVC	• Yorumlanabilirlik (özellik katsayıları üzerinden) • Yüksek boyutlu sparse veri ile uyumlu olmaları • Hızlı eğitim ve tahmin	• Barış: Logistic Regression (class_weight="balanced") + SelectKBest(f_classif) • Salih: Logistic Regression (C=0.5, balanced) ve LinearSVC (balanced) + mutual_info_classif
Ağaç Tabanlı Modeller	DecisionTreeClassifier	• Non-lineer ilişkileri yakalayabilme • Feature importance üzerinden açıklanabilirlik • Kategorik değişkenlerle doğal uyum	• Barış: Temel feature set üzerinde Decision Tree (balanced) • Yusuf: Geniş feature set + Decision Tree (balanced) + SelectFromModel
Olasılıksal Modeller	ComplementNB	• Çok kategorili, One-Hot'lanmış özelliklerde hızlı baseline • Şikâyet metnine dair türetilmiş özelliklerle uyumlu • Sınıf dengesizliğinde iyi performans	• Musa: Complement Naive Bayes (tam feature set) + chi² tabanlı SelectKBest
Mesafe Tabanlı Modeller	KNeighborsClassifier	• Benzer şikâyet örneklerine göre tahmin yapan sezgisel yöntem • Non-parametrik yaklaşım • Lokal yapıları yakalama	• Musa: KNN (k=5, weights="distance") *Not: Hesaplama maliyeti nedeniyle eğitimde 50K, testte 20K örneklik stratified alt kümeler kullanılıyor*

Hiper-Parametre Ayarlama

Vize Aşaması (Mevcut)

Hiper-parametreler ağırlıklı olarak mantıklı başlangıç değerleriyle elle seçilmiştir:

- Decision Tree: max_depth, min_samples_leaf, min_samples_split
- Logistic Regression: C, max_iter, class_weight
- KNN: n_neighbors, weights
- LinearSVC: C, class_weight

Final Aşaması (Plan)

Seçilen birkaç model için:

- Grid Search / Random Search ile hiper-parametre aralıklarının sistematik taranması
- Stratified k-fold CV kullanarak ortalama ± standart sapma performanslarının raporlanması

Sınıf Dengesizliği Stratejisi

Şu Anda Kullanılan Stratejiler

Class weight:

- Logistic Regression, Decision Tree, LinearSVC için `class_weight="balanced"` kullanımı
- Azınlık sınıfının (Yes) öğrenme sürecinde daha fazla ağırlık almasını sağlama

Planlanan Stratejiler (Final Aşaması)

Eşik ayarı (Threshold Tuning):

- ROC eğrisi ve Precision–Recall eğrisi üzerinden pozitif sınıf için farklı karar eşiklerinin test edilmesi

Yeniden örnekleme:

- Gerekirse SMOTE ve/veya random undersampling ile eğitim verisinde sınıfları kısmen dengeleme
- Veri boyutu ve runtime göz önüne alınarak bu adım final aşamasına bırakılmıştır