

# BOOKSTORE CHATBOT

**Dersin Adı:** Üretken Yapay Zeka Yardımı ile Chatbot Geliştirme Temelleri

**Hazırlayan:** Mustafa Ata Abatay

## 1.Giriş:

Bu proje, e-ticaret tabanlı kitap satış platformları için geliştirilmiştir. Geleneksel kural tabanlı botların esneklik sorununu ve saf Büyük Dil Modellerinin (LLM) "halüsinasyon" (yanlış bilgi üretme) riskini ortadan kaldırmayı hedefler. Sistem, kullanıcı niyetini anlamak için Makine Öğrenmesi, kesin veriler (Fiyat/Stok) için yapısal veritabanı sorguları ve karmaşık prosedürler (İade/Değişim) için RAG teknolojisini birleştirir.

## 2.Sistem Mimarisi:

Sistem gelen mesajı işlemek için 3 katmanlı bir mimari kullanır.

### 2.1. Niyet Analizi (Intent Classification) ve Slot Masking:

Kullanıcının mesajı işlenmeden önce, sistemde geliştirilen Slot Masking yöntemini kullanır. Bu yöntem, modelin özel isimleri (örn: "Nutuk", "Sefiller") ezberlemesini (overfitting) engeller.

- **Süreç:** Kullanıcı "Nutuk ne kadar?" yazdığında, sistem bunu arka planda *"PH\_KITAP ne kadar?"* şablonuna dönüştürür.
- **Algoritma:** Vektörleştirme için TF-IDF (Term Frequency-Inverse Document Frequency) ve sınıflandırma için Logistic Regression modeli kullanılmıştır.

### 2.2. Veri Yönetimi ve Hibrit Yönlendirme:

Sınıflandırılan niyet (intent) sonucuna göre sistem iki farklı veri kaynağına yönelir:

1. **Yapısal Veri (Structured Data):** "ask\_price", "ask\_stock" gibi niyetlerde, Python Pandas kütüphanesi ile kitap\_envanteri.xlsx dosyasından anlık ve kesin veri çekilir.
2. **Yapısal Olmayan Veri (Unstructured Data):** return\_product (İade) niyetinde ise RAG sistemi devreye girer.

### 2.3. RAG (Retrieval-Augmented Generation) Sistemi:

İade politikaları gibi yoruma dayalı ve uzun metinler için LangChain ve Google Gemini modelleri kullanılmıştır.

- **Vektör Veritabanı:** iade.pdf dokümanı parçalanarak (chunking) ChromaDB üzerinde vektörel olarak saklanmaktadır.
- **LLM Konfigürasyonu:** Cevapların tutarlılığı için Gemini 2.5 Flash modeli kullanılmış ve **Temperature** değeri **0.0** olarak ayarlanmıştır. Bu sayede modelin yaratıcılığı kısıtlanarak sadece dokümana sadık kalması sağlanmıştır.

## 3. VERİ SETİ VE ÖN İŞLEME

Projenin başarısındaki en kritik faktör, gerçek dünya koşullarını simüle eden eğitim verisidir.

### 3.1. Veri Üretimi:

Modelin dayanıklılığını artırmak amacıyla, sadece temiz veriler değil, kasıtlı olarak bozulmuş veriler de eğitim setine eklenmiştir. dataset\_deneme.py modülü kullanılarak:

- **Yazım Hataları (Typos):** Rastgele harf silme veya değiştirme (Örn: "kiatp", "fiyta").
- **Dolgu Kelimeler (Filler Words):** "Hocam", "Acaba", "Bi baksana" gibi günlük konuşma kalıpları.
- **Büyük/Küçük Harf Varyasyonları:** İnternet diline uygun olarak tamamen küçük harfli veri üretimi gerçekleştirilmiştir.

Toplamda 2207 satırlık veri seti oluşturulmuştur.

## 4. DENEYSEL SONUÇLAR VE PERFORMANS ANALİZİ

Model seçimi aşamasında **Naive Bayes** ve **Logistic Regression** algoritmaları karşılaştırılmıştır.

### 4.1. Model Karşılaştırması

Yapılan testler sonucunda Logistic Regression modelinin, özellikle gürültülü verilerde daha kararlı olduğu gözlemlenmiştir.

```
Naive Bayes Accuracy: 0.9305135951661632
Naive Bayes F1 Score: 0.9314069240085378
Naive Bayes Precision: 0.9453340506046891
Naive Bayes Recall: 0.9305135951661632
Logistic Regression Accuracy: 0.9501510574018127
Logistic Regression F1 Score: 0.9498731039982183
Logistic Regression Precision: 0.9551098318546059
Logistic Regression Recall: 0.9501510574018127
```

Şekil 1: Naive Bayes ve Logistic Regression modellerinin performans karşılaştırması.

## 5. SONUÇ

Bu projede geliştirilen Kitap Dünyası Asistanı, hibrit mimarisi sayesinde hem kural tabanlı sistemlerin kesinliğini hem de yapay zekanın esnekliğini başarıyla birleştirmiştir. Slot Masking tekniği ile ezberleme sorunu çözülmüş, RAG entegrasyonu ile halüsinasyon riski minimize edilmiştir.