

## # Banka Mevzuat Uyum Sistemi - Calisma Mantigi ve Teknik Aciklama

Bu dokuman, sistemin nasıl çalıştığını, kullanılan teknolojileri ve "neden" kullanıldıklarını hiç bilmeyen birine anlatır gibi sade ve detaylı bir şekilde açıklar.

---

## 1. Temel Kavamlar: Ne Yapıyoruz?

**Amac:** Bir bankanın müşteri sözleşmeleri (örnegin Kredi Kartı Sözleşmesi) ile devletin belirlediği kurallar (Teblig/Mevzuat) arasındaki uyumu denetlemek.

**Sorun:** Sözleşmeler çok uzun (200+ sayfa) ve kurallar çok karmaşık (Teblig). İnsanlar yavaş okur ve gözden kaçırır.

**Cozum:** Bu dokumanları bilgisayara okutup, yapay zekaya denetletmek.

---

## 2. Adım Adım Çalışma Sureci (Pipeline)

Sistemin "Baslat" düğmesine bastığınızda arka planda şu işlemler sırasıyla gerçekleşir:

### A. Veri Hazırlığı ve "Chunking" (Parçalama)

Bilgisayarlar (ve Yapay Zeka modelleri), 200 sayfalık bir kitabı tek seferde hafızasında tutamaz. Tipki bir insanın kitabı sayfa sayfa veya paragraf paragraf okuması gerektiği gibi, biz de dokumanları küçük parçalara böleriz.

- **Chunking Nedir?** Uzun bir metni, anlam bütünlüğünü bozmadan küçük parçalara (örnegin her biri bir sözleşme maddesi olacak şekilde) bölmeye işlemidir.

- **Neden Yaptık?**

- Odaklanma:** Yapay zeka sadece ilgili maddeye odaklısanın, dikkati dağılmasın.
- Arama Başarısı:** "Kart aidati" diye arattığımızda, 200 sayfalık dosya yerine sadece aidatla ilgili paragrafi bulmak isteriz.

### B. İndeksleme: ChromaDB ve BM25

Parcaladığımız bu küçük metinleri (Chunk), daha sonra hızla bulabilmek için özel bir kütüphaneye (veritabanına) kaydederiz. Burada iki farklı teknoloji kullanıyoruz:

1. **ChromaDB (Vektor Veritabanı):**

- Nedir?** Kelimelerin "anlamını" sayılar (vektörler) çevirir.
- Ornek:** Siz "ek masraf" diye ararsınız, sistem bunun "ilave ücret" veya "komisyon" ile benzer anlamda geldiğini bilir ve onları bulur. Kelime aynı olmasa bile \*anlam\* aynıysa bulur.

2. **BM25 (Anahtar Kelime İndeksi):**

- Nedir?** Klasik "Google Araması" gibi çalışır. Kelimelerin birebir kendisini arar.
- Ornek:** "Madde 12/A" diye ararsak, vektor veritabanı bunu "Madde 12/B" ile karşılaştırabilir (sayilar birbirine benzerdir). Ancak BM25, tam olarak "12/A" yazan yeri bulur.

\*\*Teknik Detay: Türkçe için BM25 Stratejimiz (Neden Zemberek Kullanmadık?)\*\*

Türkçe, sondan eklemleli bir dildir (örn: \*Banka Bankalar Bankaların\*). Klasik arama motorlarında "Banka" diye aratınca "Bankaların" kelimesini bulmak için \*Stemming/Lemmatization\* (Kelime kokune inme - Zemberek vb.) yapılması önerilir.

Ancak biz bu projede \*\*Basit Tokenizasyon\*\* (sadece küçük harfe çevirme) kullandık.

\*\*Neden?\*\*

1. \*\*Hiz (Latency):\*\* Kok bulma islemi her sorguda sistemi yavaslatir. Hackathon'da hiz kritiktir.
2. \*\*Hibrit Gucu:\*\* BM25'in "eklerden dolayi kacirdigi" kelimeleri, zaten anlamdan yakalayan \*\*Vektor Arama (ChromaDB)\*\* tamamlamaktadir. Yani agir bir Turkce kutuphanesi kullanmadan, Hibrit Mimari sayesinde ayni basariyi cok daha hizli elde ettik.

## C. Hibrit Arama (Hybrid Search) ve RRF

Banka sozlesmesindeki bir maddeyi denetlemek icin, o maddeyle ilgili Mevzuat (Tebliğ) kuralini bulmamiz gereklidir.

- **Soru:** Banka sozlesmesindeki "Yillik uyelik bedeli 500 TL" maddesi icin hangi yasaya bakmaliyim?
- **Islem:** Sistem bu maddeyi hem ChromaDB'de (anlam) hem BM25'te (kelime) aratir.
- **Birlestirme (RRF - Reciprocal Rank Fusion):** Iki farkli arama sonucunu harmanlar. Hem anlam olarak en yakin, hem de kelime olarak en uyusan sonucları ust siraya tasir. Boylece dogru yasai bulma ihtimalimiz maksimuma cikar.

## D. Karar Motoru (Decision Engine): Once Kural, Sonra Zeka

Dogru yasai bulduk (Hibrit Arama tamamlandi). Simdi "Banka maddesi bu yasaya uygun mu?" sorusunu cevaplamlamaliyiz.

Bu motor, **her zaman ve kesinlikle sirasiyla** calisan 3 adimdan olusur:

1. **Adim: Kural Motoru (Rule Engine) - [Hesaplama & Uyari]**
  - **Yapi:** Sadece basit Regex (Kelime Arama) degildir. **Regex + Python Mantigi** birlikte calisir.
  - **Nasil Calisir?**
  1. **Veri Cikarma (Regex):** Metin icindeki sayiları ("5 TL"), oranları ("%10") ve anahtar kelimeleri ("Onaysiz") Regex ile cekip cikarir.
  2. **Mantiksal Kiyas (Python):** Cikarilan bu sayiları matematiksel olarak karsilastirir (`if banka\_ureti > yasa\_limiti`).
  - **Ornek:** Yasa "EFT max 2 TL" derken Banka "5 TL" demisse, Regex "2" ve "5" sayilarini bulur, Python kodu "5 > 2" islemi yapip hatayı yakalar.
  - **Kritik Nokta:** Kural motoru burada islemi bitirmez. Tespit ettiği hatayı bir "UYARI ETIKETI" (Warning Label) haline getirir ve bir sonraki asamaya (LLM'e) paslar.
2. **Adim: Buyuk Dil Modeli (LLM - Gemma) - [Akil Yurutme & Karar]**
  - Yapay zeka devreye girer ve kendisine sunulan **TUM** verileri okur:
  - Girdi 1: Banka Maddesi
  - Girdi 2: Ilgili Yasa (Mevzuat)
  - Girdi 3: **Kural Motorundan Gelen Uyari** ("Dikkat: Burada matematiksel limit asimi var!")
  - **Nihai Karar:** LLM, bu uc veriyi birlestirir. Kural motorunun uyarisini dikkate alarak ve yasanin sozel yorumunu da ekleyerek kararini verir: "UYUMSUZ (NOT\_OK)".
3. **Adim: Son Kontrol (Safety Override) - [Guvenlik Kilidi]**
  - LLM kararini verdikten sonra, devreye son bir guvenlik kodu girer.
  - **Neden?** Bazen LLM cok karmaşık, sinsi ifadeleri (orn: "imzayla sigortayı kabul etmis sayilirsınız") "Uyumlu" sanabilir.
  - Kod, bu tip yasakli kelime kaliplarını (Regex) kontrol eder. Eger LLM "Uyumlu" dese bile, kod bu yasagi gorurse karari ezer (**Overrule**) ve sonucu "UYUMSUZ" olarak degistirir.

**Ozet Siralama:** Hibrit Arama -> Kural Uyarisi -> LLM Karari -> Son Kontrol (Override). Bu sira asla degismez.

## E. Raporlama ve Chatbot

Tum maddeler tek tek bu surecten gectikten sonra:

- 1. Dashboard:** Sonuclar renkli grafiklerle ekran'a yansitilir. Kirmizi (Risk), Yesil (Uyumlu).
- 2. Otomatik Ozet:** Sistem sonuclarini analiz eder ve yoneticiye "Toplam 5 risk var, en kritik olani EFT ucreti" seklinde bir ozet yazar. Bu ozet Yapay Zeka (LLM) tarafindan degil, onceden hazırlanmis akilli sablonlar (Template) ile, cikan istatistiklere gore anlik uretilir.
- 3. Chatbot:** Tum analiz bittikten sonra, eger akliniza takilan bir sey olursa Chatbot'a sorabilirsiniz. Chatbot, hem orijinal yasayi hem de bizim analiz sonuclarimizi bilir. "Neden EFT maddesine riskli dedin?" diye sorarsaniz, size dayanagini (Madde 12/3 ve hesapladigi tutar farki) aciklar.

---

## Ozet: Neden 3 Farkli Teknoloji (Vektor, Kural, LLM) Kullaniyoruz?

- Tek basina **LLM** kullansaydik matematik hatasi yapabilirdi (LLM'ler sayı saymakta bazen zorlanır).
- Tek basina **Kural (Rule)** kullansaydik, "Musteri magdur edilemez" gibi yorumu açık cümleleri anlayamazdı.
- Tek basina **Vektor Arama** kullansaydik, sayıları (Madde 12 ile 13'u) karıştırabilirdi.

Hepsini birlestirerek (Hybrid Pipeline), **hem matematiksel kesinligi, hem hukuksal yorumlama yetenegini hem de dogru yasaya ulasma hizini** tek bir potada erittik.