

## Çözüm Mimarım

Bu projede temel amacım, kullanıcıların ruh haline, tarzına veya izleme niyetine göre doğal bir şekilde film önerisi alabileceği akıllı bir sohbet sistemi geliştirmektir.

Bunu başarmak için klasik anahtar kelime eşleşmelerinden çok daha öteye geçip, anlam tabanlı arama (semantic search) ve RAG (Retrieval-Augmented Generation) mimarisini birleştirdim.

### Proje Amacı


CineMate, bir kullanıcının “90’lar bilim kurgu”, “romantik ama çok duygusal olmasın”, ya da “yüksek puanlı gizemli bir film istiyorum” gibi doğal ifadelerini anlayarak uygun filmleri öneren bir asistan olarak tasarlandı.

Amaç sadece listelemek değil — aynı zamanda yapay zekâ desteğiyle, **insan gibi öneri yapan** bir sohbet deneyimi sunmak.

### Katman Teknoloji Görevi

<b>Arayüz</b> (Frontend)	 <b>Streamlit</b>	Kullanıcıyla etkileşim kuran web tabanlı sohbet arayüzü. Mesaj geçmişi, film kartları, posterler ve özetleri gösteriyor.
<b>Veri</b> Katmanı	 <b>MovieLens</b> <b>Dataset + TMDb API</b>	MovieLens veri setinden film ID, tür ve puan bilgileri alınır. TMDb API’den ise Türkçe film adları, özetleri ve poster görselleri çekilir.
<b>Vektörleştirme</b> Katmanı	 <b>SentenceTransformer</b> (all-MiniLM-L6-v2)	Filmlerin başlık ve tür bilgilerini anlam uzayında sayısal vektörlere dönüştürür. Bu sayede kullanıcı sorgusu ile filmler arasında anlam benzerliği ölçülebilir hale gelir.
<b>Anlam Arama</b> (Retrieval)	 <b>Cosine Similarity</b>	Kullanıcının doğal diliyle yazdığı sorgu (ör. “komik ama romantik”) embed edilerek, en benzer film vektörleri bulunur.
<b>Cevap Üretimi</b> (Generation)	 <b>OpenAI / OpenRouter</b> <b>API (Model: openai/gpt-oss-20b:free)</b>	RAG ile getirilen film sonuçları bağlam olarak modele gönderilir. Model, bu bağlama dayalı olarak <b>doğal Türkçe bir öneri mesajı</b> üretir.

Ortam  
Yönetimi

 **python-**  
**dotenv**

.env dosyasında gizli API anahtarlarını güvenli  
şekilde tutar.

## RAG (Retrieval-Augmented Generation) Mimarisi

RAG mimarisi bu projede şu şekilde çalışır:

1. **Kullanıcı sorgusu (prompt)** alınır.  
→ Örnek: “bilim kurgu ama biraz duygusal olsun”
2. **Retrieval aşaması:**
  - Kullanıcının girdiği metin SentenceTransformer modeliyle embedding’e dönüştürülür.
  - Bu embedding, MovieLens veri setindeki tüm film embedding’leriyle karşılaştırılır.
  - **Cosine similarity** ile en benzer filmler seçilir.
3. **Augmentation aşaması:**
  - Bulunan filmlerin başlık, tür, puan ve özet bilgileri **bağlam (context)** olarak hazırlanır.
  - Bu bağlam, kullanıcı sorgusuyla birlikte dil modeline (GPT tabanlı model) gönderilir.
4. **Generation aşaması:**
  - Model, bu bağlamdan yola çıkarak **doğal, Türkçe ve kişisel bir öneri yanıtı** üretir.
  - Önerilen filmler açıklamalı biçimde sohbet penceresinde kullanıcıya sunulur.

## Avantajlar

- Anlam bazlı öneri: yalnızca kelimelere değil, **niyete (intent)** odaklanır.
- RAG mimarisi sayesinde model, **dış veriden (MovieLens + TMDb)** beslenir.
- Türkçe destekli özet ve poster entegrasyonu ile daha doğal bir deneyim sunar.
- Streamlit üzerinden etkileşimli ve görsel açıdan zengin bir sohbet arayüzü.