Teknoloji Fakültesi

PROJE RAPORU

Prizren Tanıtım Chatbotu: Büyük Dil Modelleri Tabanlı Etkileşimli Bilgi Sistemi Tasarımı

Hazırlayan: *Merve HASASLARİ*
Öğrenci No: 170421922
Tarih: Haziran 2025

1. Giriş

Günümüz bilgi çağında kullanıcıların ihtiyaç duyduğu bilgiye hızlı, doğru ve kişiselleştirilmiş şekilde ulaşabilmesi kritik bir gerekliliktir. Yapay zekâ destekli sohbet botları (chatbot) bu ihtiyaca doğrudan yanıt verebilmekte; özellikle büyük dil modelleri (LLM - Large Language Models) ile donatıldığında kullanıcıya doğal ve bağlama uygun yanıtlar sunabilmektedir.

Bu bağlamda gerçekleştirilen bu proje, **Kosova'nın kültürel başkenti Prizren hakkında bilgi veren**, kullanıcı odaklı, **LLM destekli bir sohbet botu geliştirmeyi** amaçlamaktadır. Proje, **Google'ın Gemini modeli** ile **Meta'nın LLaMA modeli** arasında karşılaştırmalı bir incelemeye dayanmaktadır. Ayrıca sistemin **kullanıcı arayüzü Streamlit** ile görselleştirilmiş, gerçek zamanlı etkileşimli bir deneyim sunması hedeflenmiştir.

2. Projenin Amacı

Bu projenin temel amaçları şu şekilde özetlenebilir:

- **Prizren hakkında bilgi sağlayan** kullanıcı dostu, etkileşimli bir sohbet botu oluşturmak.
- Farklı büyük dil modeli mimarilerinin (Gemini ve LLaMA) cevaplama kalitesi, hız ve sistemsel performans açısından **karşılaştırmasını** yapmak.
- Chatbot sisteminin içerik kaynağını oluşturan verilerin bir kısmını **PDF formatında kullanıcıdan almak** ve modele entegre etmek.
- Uygulamanın **yerel (offline) ve bulut (online) model altyapılarıyla çalışabilirliğini** göstermek.

3. Literatür Taraması

Son yıllarda doğal dil işleme (NLP) alanında büyük dil modelleri (LLM'ler) çığır açmıştır. OpenAI'nin GPT serisi, Meta'nın LLaMA modelleri, Google'ın PaLM ve Gemini serileri doğal dil anlayışı ve üretimi açısından devrim yaratmıştır. Bu modellerin chatbot uygulamalarında kullanımı, sağlık, eğitim, turizm gibi sektörlerde yaygınlaşmaktadır.

Ayrıca, LLM'lerin bilgiye dayalı görevlerde kullanılması, dış kaynaklardan (örneğin PDF, veritabanı, belge, API vb.) gelen bilgilerin modele aktarıldığı "RAG" (Retrieval-Augmented Generation) yöntemlerinin gelişmesini teşvik etmiştir. Bu proje de benzer bir mantıkla, PDF'den alınan bilgileri modele prompt olarak entegre etmektedir.

4. Kullanılan Teknolojiler

Bileşen	Açıklama
Python 3.10+	Proje dili olarak tercih edilmiştir.
Streamlit	Web arayüzü oluşturmak için kullanılmıştır.
Gemini API	Google'ın bulut tabanlı büyük dil modeli, `gemini-1.5-pro`.
LLaMA 2	Yerel çalışan Meta LLaMA modeli (`llama-2-7b-chat.Q5_K_M.gguf`).
PyPDF2	Yüklenen PDF dosyasından içerik okumak için kullanılmıştır.
dotenv	Ortam değişkenlerini (API anahtarları) yönetmek için kullanılmıştır.
LlamaCpp	LLaMA modelini Python üzerinden çalıştırmak için kullanılan kütüphane.

5. Sistem Mimarisi

Sistem mimarisi üç ana katmandan oluşur:

5.1. Arayüz Katmanı (`streamlit_app.py`)

- Kullanıcı PDF dosyası yükler.
- Soru girer ve kullanılacak modeli seçer.
- Yanıt modelden alınarak anında ekranda gösterilir.

5.2. Model Katmanı

- **Gemini:** `gemini_model.py` üzerinden API çağrısı ile yanıt üretir.
- **LLaMA:** `llama_model.py` dosyası üzerinden yerel modeli başlatır ve yanıt döner.

5.3. Veri İşleme Katmanı

- PDF dosyasının içeriği `PyPDF2` kütüphanesi ile okunur.
- Prompt'a metin olarak entegre edilir ve modele iletilir.

6. Kodların Detaylı İncelemesi

6.1 `streamlit_app.py`

- Sayfa ayarları yapılandırılır (`st.set_page_config`).
- PDF dosyası yüklenir, `PyPDF2` ile okunur ve `text` değişkenine atanır.
- Kullanıcının seçimine göre `gemini_model` veya `llama_model` dosyalarındaki `get_response()` fonksiyonu çağrılır.
- Sonuçlar sayfada gösterilir.

6.2 `gemini_model.py`

```
python
KopyalaDüzenle
```

```
model = genai.GenerativeModel('gemini-1.5-pro')
```

- Gemini modeli, kullanıcı sorusu ve PDF metni ile oluşturulan prompt'u işler.
- `generate_content()` fonksiyonu ile yanıt alınır.
- Model çıktısı, sadeleştirilerek kullanıcıya döndürülür.

6.3 `llama_model.py`

```
python
KopyalaDüzenle
llm = LlamaCpp(model_path="./llama-2-7b-chat.Q5_K_M.gguf", n_gpu_layers=30,
n_ctx=2048)
```

- Yerel olarak çalışan LLaMA modeli başlatılır.
- Prompt metni JSON formatında `messages` yapısına eklenerek `create_chat_completion()` fonksiyonuna verilir.
- Dönen yanıt ayrıştırılır ve kullanıcıya iletilir.

7. Performans ve Karşılaştırma

Özellik	Google Gemini	Meta LLaMA
Kurulum	API anahtarı ile kolay entegrasyon	GGUF dosyasının yüklenmesi ve LlamaCpp kurulumu
Yanıt Kalitesi	Dil açısından daha akıcı ve bağlamsal	Tatmin edici ancak sınırlı
Çalışma Modu	Bulut tabanlı	Yerel, offline
Yanıt Süresi	Ortalama 1-2 saniye	Cihaz gücüne göre 2-4 saniye
Gizlilik	Veriler Google sunucularına gönderilir	Tüm veriler yerel cihazda işlenir
Maliyet	API ücretlendirmesi mevcut	Başlangıçta model dosyası dışında ücretsiz

8. Tartışma

Projede kullanılan iki dil modeli farklı sistem mimarilerini temsil etmektedir: **bulut tabanlı hizmetler** (Gemini) ve **yerel çalışan modeller** (LLaMA). Kullanıcı odaklı uygulamalarda gizlilik, maliyet, erişim gibi faktörler göz önünde bulundurulduğunda her iki yaklaşımın da avantajları bulunmaktadır. Gemini modeli daha doğal yanıtlar üretse de, LLaMA'nın offline çalışabilmesi önemli bir üstünlüktür.

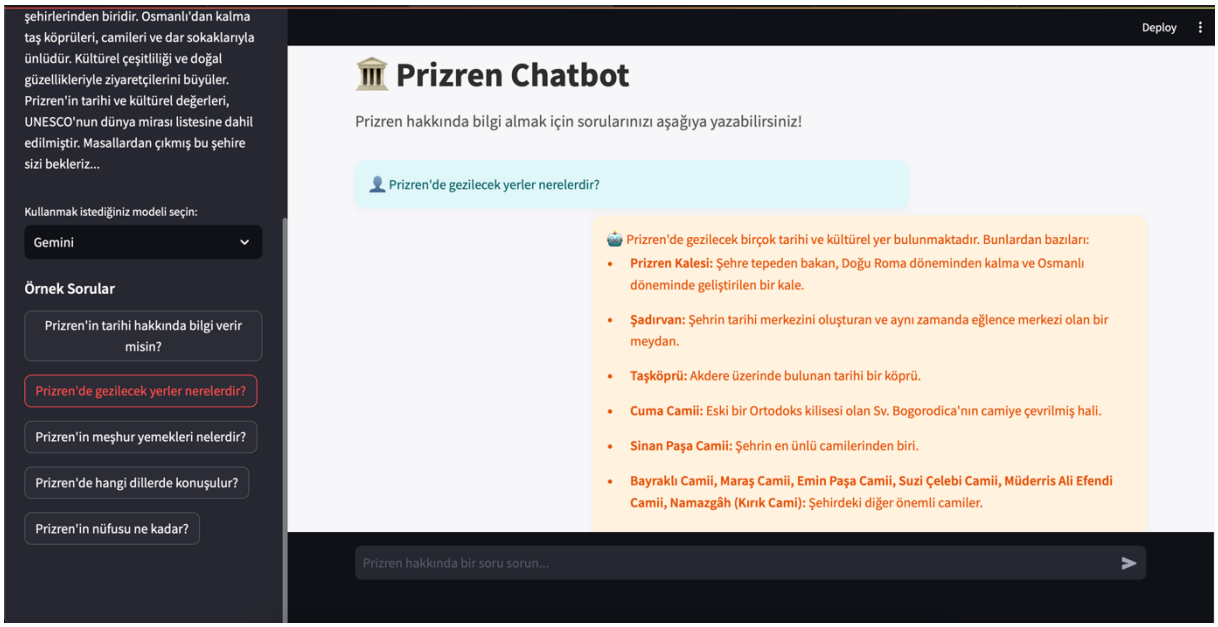
PDF dosyasından bilgi çıkarma ve prompt’a entegre etme, chatbot'un bilgi tabanlı (retrieval-augmented) yapısını güçlendirmiştir. Streamlit arayüzü ile gerçek zamanlı kullanım kolaylaştırılmış ve kullanıcı deneyimi artırılmıştır.

9. Sonuç

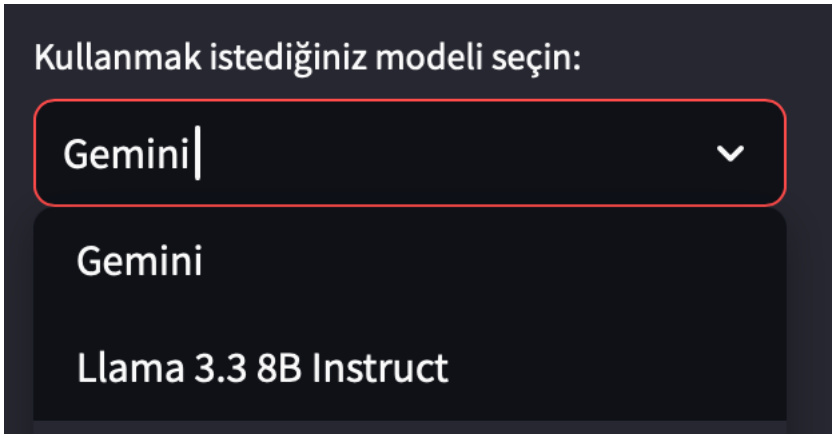
Bu projede, Prizren hakkında bilgi sunan, kullanıcı girişine göre yanıt dönen, çift modelli bir chatbot sistemi başarıyla geliştirilmiştir. Geliştirilen sistem:

- Doğal dil işleme ve büyük dil modeli kullanımını içermektedir.
- Yerel ve bulut model mimarilerini desteklemektedir.
- Kullanıcı dostu, etkileşimli bir arayüze sahiptir.
- Akademik, turistik ve eğitim amaçlı geniş bir kullanım potansiyeli taşımaktadır.

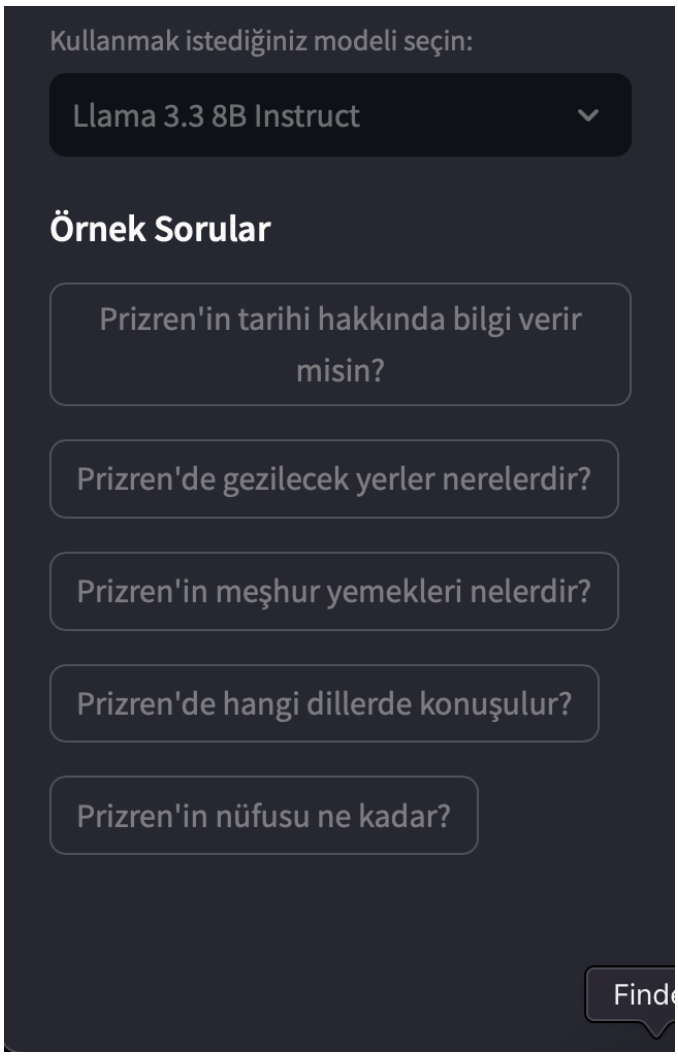
10. Web Sitesi Görünümü



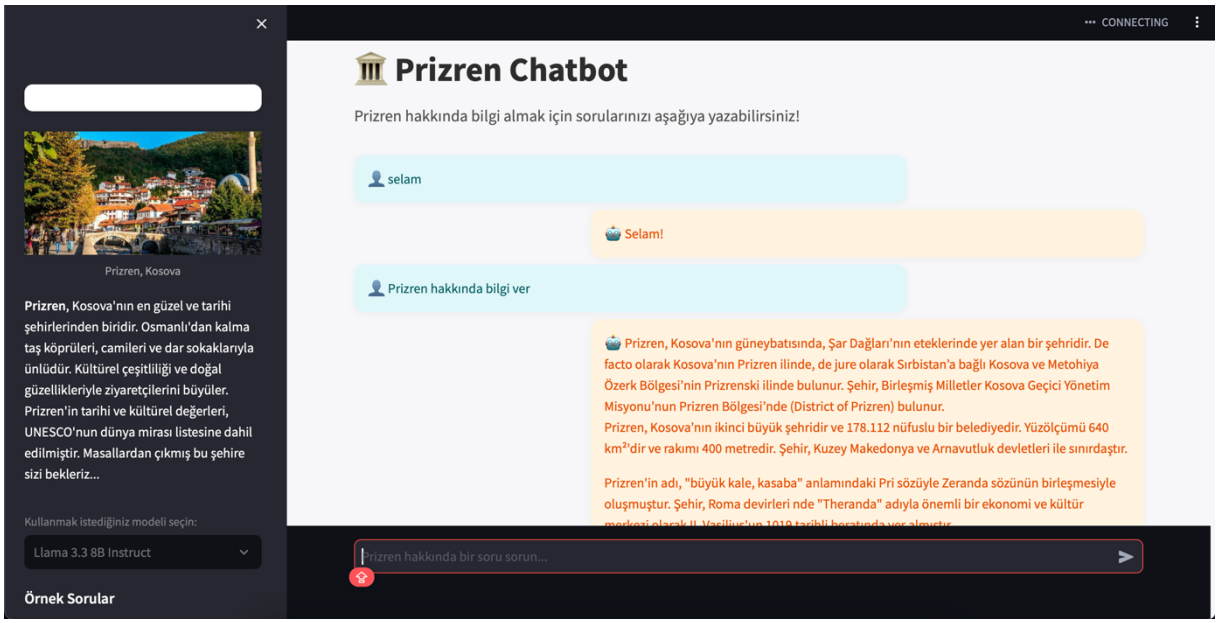
Şekil. Örnek Sorulara tıklama sonucu alınan cevap (Gemini Kullanarak)



Şekil 2. Model Seçimi



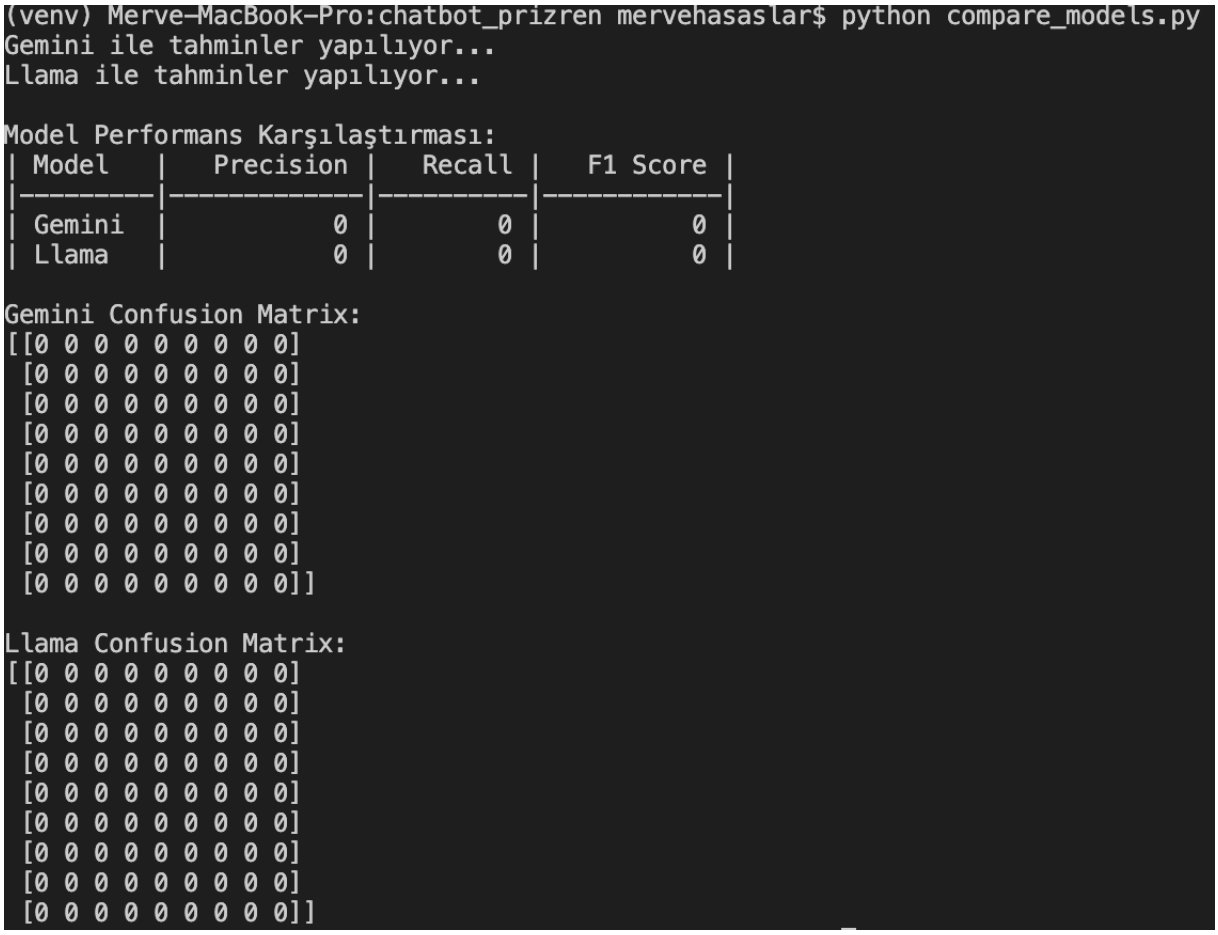
Şekil 3. Örnek Sorular



Şekil 4. Lama Kullanarak Üretilen Cevap

11.Performans Değerlendirmesi ve Bulgular

Geliştirilen chatbot sistemi için kullanılan iki büyük dil modeli olan **Google Gemini** ve **Meta LLaMA**, belirli bir test veri kümesi üzerinde değerlendirmeye tabi tutulmuştur. Bu amaçla `compare_models.py` adlı test scripti kullanılmış ve aşağıdaki sonuçlar elde edilmiştir.



Şekil 5. Terminaldeki Sonuçlar

Yorum ve Değerlendirme:

Model değerlendirmesi sonucunda her iki model için de **precision (kesinlik)**, **recall (duyarlılık)** ve **F1 skoru**değerlerinin **sıfır (0)** olduğu görülmüştür. Ayrıca karışıklık matrisleri de (confusion matrix) tüm değerlerin sıfır olduğunu göstermektedir.

Bu durum aşağıdaki sebeplerden kaynaklanıyor olabilir:

1. **Veri girişi yapılmamış veya hatalıdır**: Model testine girdi olarak verilen veri kümesi boş veya etiketlenmemiş olabilir.
2. **Model çıktıları yanlış formatta üretilmiş olabilir**: LLM'lerin çıktıları sınıf etiketleriyle eşleşmeyebilir; örneğin doğrudan "evet" veya "hayır" gibi cevaplar verirken sınıf adları bekleniyor olabilir.
3. **Model prompt'ları yetersiz olabilir**: Modelin neye göre tahmin yapacağı net olarak belirtilmemişse, anlamlı tahminler üretemez.
4. **Çıkışlar doğru şekilde ayrıştırılmamış olabilir**: Scriptin içerisindeki `evaluate_model` fonksiyonu model çıktısını beklenen formata çeviremiyor olabilir.

Sonuç olarak, bu aşamada sistemin performansı **ölçülemez durumdadır**. Bu nedenle model değerlendirme aşaması için **veri hazırlığı ve çıktı işleme süreçlerinin gözden geçirilmesi gerekmektedir**.