

# BİTİRME PROJESİ A:

## Yeni Nesil Dijital Sağlık Asistanı: RAG Tabanlı Akıllı Beslenme Rehberi

221307036  
Tuğçe Gül  
Kocaeli Üniversitesi  
Bilişim Sistemler Mühendisliği

Bu çalışmada, büyük dil modelleri ve bilgi getirme temelli bir diyetisyen asistan sistemi sunulmaktadır. Geliştirilen sistem, hem kişiye uygun diyet listeleri oluşturmayı hem de beslenme ve sağlık ile ilgili sorulara doğal dilde yanıt vermemeyi amaçlamaktadır. Sistem mimarisinde, dil modelinden bağımsız olarak çalışan bir bilgi getirme bileşeni yer almaktadır. Bu bileşenin başarımı, Türkçe beslenme verilerinden oluşan bir test kümesi üzerinde Hit Rate at k ve Recall at k metrikleri kullanılarak değerlendirilmiştir. Deneyel sonuçlar, getirilen doküman sayısı arttıkça geri çağrıma oranının yükseldiğini ve düşük k değerlerinde dahi yüksek isabet oranlarının elde edildiğini göstermektedir. Elde edilen bulgular, bilgi getirme bileşeninin güvenli ve doğru diyet ve sağlık önerilerini sunulmasını desteklediğini ortaya koymaktadır.

*dietetisyen asistanı, bilgi getirme, büyük dil modelleri, beslenme, sağlık, doğal dil işleme*

### I. GİRİŞ

Günümüzde yapay zeka tabanlı sistemler, sağlık ve beslenme alanında karar destek aracı olarak giderek daha fazla kullanılmaktadır. Özellikle büyük dil modellerinin gelişimi, kullanıcıların doğal dilde soru sorabildiği ve kişiselleştirilmiş öneriler alabildiği akıllı asistanların ortaya çıkışmasını sağlamıştır. Ancak bu tür sistemlerin güvenilir ve doğru bilgi üretmesi, özellikle sağlık alanında büyük önem taşımaktadır. Bu çalışmada, beslenme ve sağlık alanına yönelik bir **dietetisyen asistan sistemi** geliştirilmiştir. Sistem, iki temel işlevi yerine getirmektedir: (i) kullanıcının özelliklerine göre diyet listeleri hazırlamak ve (ii) beslenme ve sağlık ile ilgili sorulara doğal dilde yanıt vermek.

Geliştirilen sistemde, dil modelinin bilgi uydurmasını azaltmak ve yanıtların güvenilirliğini artırmak amacıyla **bilgi getirme tabanlı bir mimari** benimsenmiştir. Bu mimaride, önceden yapılandırılmış beslenme dokümanları vektör uzayında temsil edilmekte ve kullanıcı sorgusuna en ilgili içerikler geri getirilmektedir.

### II. KULLANILAN KAYNAKLAR VE VERİ HAZIRLAMA SÜRECİ

Bu çalışmada kullanılan bilgi tabanı, Türkiye Beslenme Rehberi (TUBER) ve Halk Eğitim Merkezleri tarafından yayımlanan beslenme ve sağlık içeriği PDF dokümanlarından elde edilmiştir. Bu kaynaklar, bilimsel temelli beslenme ilkeleri ve hastalıklarda beslenme önerileri

icermeleri nedeniyle tercih edilmiştir. Kullanılan dokümanlar, bilgi getirme ve dil modeli destekli sistemde kullanılmak üzere ön işleme ve yapılandırma adımlarından geçirilmiştir.

#### A. PDF Dokümanlarından Metin Çkarımı

PDF belgelerinden metin çıkarımı işlemi PyMuPDF kütüphanesi kullanılarak gerçekleştirilmiştir. Belirli bir sayfa aralığı seçilerek yalnızca ilgili bölümler işlenmiş ve her sayfa üzerindeki metin blokları koordinat bilgileriyle birlikte elde edilmiştir. Bu yaklaşım, metnin belge üzerindeki görsel düzene uygun biçimde ayırtılmasını sağlamıştır.

#### B. Metin Temizleme ve Normalizasyon

Çıkarılan ham metinler, doğrudan kullanılmadan önce çeşitli temizleme ve normalizasyon işlemlerine tabi tutulmuştur. Bu işlemler kapsamında:

- Dokümanlara özgü tekrar eden başlık ve alt bilgi ifadeleri kaldırılmıştır.
- Sayfa numaraları ve ayırcı karakterler silinmiştir.
- Satır sonlarında hece bölünmesi nedeniyle parçalanan kelimeler birleştirilmiştir.
- Fazla boşluklar, satır sonları ve düzensiz biçimlendirmeler normalize edilmiştir.
- Madde işaretleri korunarak daha okunabilir bir yapı oluşturulmuştur.

Bu adımlar, metnin anlamsal bütünlüğünü koruyarak daha temiz ve tutarlı veri elde edilmesini amaçlamaktadır.

#### C. Renk Tabanlı Chunk Oluşturma Stratejisi

Bu çalışmada kullanılan chunk oluşturma yöntemi, PDF dokümanları üzerinde yapılan **renkli vurgulara dayalı** bir stratejiye dayanmaktadır. Dokümanlar önceden manuel olarak işaretlenmiş ve bu işaretlemeler, metnin semantik yapısını koruyacak şekilde chunk oluşturmak için kullanılmıştır.

Renklerin sistem içindeki işlevleri şu şekildedir:

- Sarı vurgular:** Bölüm başlığı olarak kabul edilmiştir. Sarı ile işaretlenen metin, aktif içerik toplama modunu başlatan bir başlık görevi görmektedir.
- Renksiz metinler:** Eğer aktif bir sarı başlık tanımlıysa, bu metinler ilgili başlığın içeriği olarak toplanmıştır.

- Mavi vurgular:** Not veya açıklama niteliğinde değerlendirilmiş ve ana içerikten ayrı bir yapı altında saklanmıştır.
- Kırmızı vurgular:** Tablo başlığı veya tablo referansı olarak ele alınmıştır. Bu çalışmada tabloların içeriği çıkarılmamış, yalnızca tablo başlıklarını meta bilgi olarak kaydedilmiştir.

Bu yöntem sayesinde metinler, sabit uzunluklu parçalara bölünmek yerine belgenin anlamlı bölümlerine göre ayrıstırılmıştır.

#### D. İki Sütunlu Sayfa Yapısının Düzenlenmesi

Kaynak dokümanların bir kısmı iki sütunlu sayfa düzenine sahiptir. Bu tür düzenler, metnin yanlış sırayla çıkarılmasına neden olabilmektedir. Bu problemi gidermek amacıyla her sayfa, yatay eksende iki sütuna ayrılmıştır. Sol ve sağ sütunlardaki metin blokları ayrı ayrı yukarıdan aşağı sıralanmış ve daha sonra doğru okuma sırasını yansıtacak şekilde birleştirilmiştir.

Bu yaklaşım, chunk'ların bağlam bütünlüğünü koruyarak daha doğru bilgi getirme sonuçları elde edilmesini sağlamıştır.

#### E. Tablo İçeriklerinin Hariç Tutulması

Beslenme dokümanlarında yer alan tablolar genellikle yoğun sayısal veri içermektedir. Bu tür içerikler, anlamsal arama ve embedding süreçlerinde gürültü oluşturabilmektedir. Bu nedenle, tablo bölgeleri tespit edilmiş ve bu bölgelerle kesişen metin blokları içerik toplama aşamasında dışlanmıştır. Böylece yalnızca açıklayıcı ve metinsel bilgiler chunk'lara dahil edilmiştir.

#### F. Anlamsal Paragraf Bazlı Chunking

Chunk oluşturma sürecinde, aynı başlık altındaki metinler tek bir uzun metin halinde birleştirilmemiştir. Bunun yerine, her temizlenmiş metin bloğu bağımsız bir paragraf olarak saklanmıştır. Bu yaklaşım, anlamsal paragraf bazlı chunking olarak değerlendirilmiştir.

Bu yöntemin avantajları şunlardır:

- Her paragrafin bağımsız bir bilgi birimi olarak temsil edilmesi
- Embedding ve retrieval aşamalarında daha hassas eşleşmelerin yapılabilmesi
- Gereksiz uzun bağlantıların önlenmesi

Elde edilen chunk'lar, başlıklar altında yapılandırılmış JSON formatında saklanarak sonraki embedding ve Chroma veritabanı aşamalarında kullanılmak üzere hazır hale getirilmiştir.

### III. SİSTEM MODÜLLERİ

Geliştirilen diyetisyen asistan sistemi, kullanıcı ihtiyaçlarına göre yapılandırılmış iki ana modülden oluşmaktadır. Bu modüler, hem beslenme ve sağlık ile ilgili soruların yanıtlanması hem de kişiye özel diyet listelerinin hazırlanmasını hedeflemektedir. Her iki modülde de bilgi getirme tabanlı yaklaşım benimsenmiş ve dil modeli, yalnızca ilgili bağlam bilgisile desteklenmiştir

#### A. Modül 1: Beslenme ve Sağlık Sorularına Yanıt Verme

Bu modül, kullanıcıların beslenme ve sağlık alanında doğal dilde sordukları sorulara doğru, güvenilir ve anlaşılır yanıtlar üretmeyi amaçlamaktadır. Sağlık alanının hassas yapısı göz önünde bulundurularak, yalnızca güvenilir kaynaklardan elde edilen bilgiler kullanılmıştır.

Kullanıcı tarafından girilen soru, öncelikle anlamsal temsil oluşturmak amacıyla vektör uzayına dönüştürülmektedir. Bu temsil, Chroma vektör veritabanında saklanan beslenme rehberi dokümanları ile karşılaştırılarak en ilgili bilgi parçalarının bulunmasını sağlamaktadır. Elde edilen bağlam, dil modeline aktarılmakta ve modelin yalnızca bu bağlam çerçevesinde yanıt üretmesi sağlanmaktadır.

Bu yaklaşım sayesinde, dil modelinin bağlam dışına çıkması ve uydurma bilgi üretmesi engellenmeye çalışılmıştır. Ayrıca sistem, sağlık alanına özgü güvenlik kuralları doğrultusunda, kullanıcıyı yanıltabilecek veya risk oluşturabilecek önerilerden kaçınacak şekilde yapılandırılmıştır.

#### B. Modül 2: Kişiye Özel Diyet Listesi Hazırlama

Bu modül, kullanıcının bireysel özelliklerine dayalı olarak diyet listeleri oluşturmayı hedeflemektedir. Kullanıcıdan alınan yaş, boy, kilo, yaşam tarzı ve hedef gibi bilgiler, diyet planının temel girdilerini oluşturmaktadır.

Diyet listesi hazırlanırken, ilgili beslenme kuralları ve rehber ilkeler bilgi getirme bileşeni aracılığıyla sistem tarafından belirlenmektedir. Ardından dil modeli, bu bilgileri kullanarak özgün, bazlı, uygulanabilir ve güvenli bir diyet planı oluşturmaktadır. Diyet listeleri, bilimsel beslenme ilkelerine uygun olacak şekilde tasarlanmış ve aşırı kısıtlayıcı önerilerden kaçınılmıştır.

Bu modül, yalnızca genel beslenme önerileri sunmakta; tıbbi teşhis veya tedavi yerine gelecek yönlendirmeler yapmamaktadır. Böylece sistem, kullanıcı sağlığını ön planda tutan bir karar destek aracı olarak konumlandırılmıştır.

#### C. Modüllerin Ortak Güvenlik ve Bilgi Kullanım Yaklaşımı

Her iki modülde de ortak olarak, bilgi getirme bileşeni ile desteklenmiş bir dil modeli kullanımı benimsenmiştir. Bu yaklaşım, sistemin doğruluğunu artırmakta ve sağlık alanında kritik olan güvenilirlik gereksimini karşılamaktadır. Dil modelinin yanıtları, yalnızca önceden yapılandırılmış ve doğrulanmış beslenme dokümanlarına dayandırılmıştır.

### IV. SORU-CEVAP MODÜLÜ İÇİN YAKLAŞIMLAR VE DEĞERLENDİRME

Bu bölümde, geliştirilen sistemin **soru-cevap modülü** için kullanılan farklı yaklaşımlar ve bu yaklaşımın performans değerlendirme süreci ele alınmaktadır. Soru-cevap modülü, kullanıcıların beslenme ve sağlık alanındaki sorularına doğru, güvenilir ve bağlama dayalı yanıtlar üretmesini hedeflemektedir.

Değerlendirme sürecinde, beslenme rehberleri ve sağlık içerikleri dikkate alınarak hazırlanmış **35 sorudan oluşan bir test kümesi** kullanılmıştır. Bu sorular,コレsterol, yağ türleri, besin grupları, kalp-damar sağlığı ve genel beslenme ilkeleri gibi farklı konu başlıklarını kapsayacak şekilde seçilmiştir.

Her soru için beklenen konu başlıkları tanımlanarak, sistemin doğru bilgiyi geri getirme ve kullanma başarısı ölçülmüştür. Soru-cevap modülü üç farklı yaklaşım altında incelenmiştir:

- A. yalnızca bilgi getirme bileşeninin değerlendirildiği retriever-only.
- B. yalnızca büyük dil modeli kullanılarak yanıt üretimi,
- C. bilgi getirme destekli dil modeli yaklaşımı (RAG) ve

Bu yaklaşımlar, sistemin doğruluk, bağlam uyumu ve güvenilirlik açısından nasıl davranışını karşılaştırmalı olarak incelemek amacıyla kullanılmıştır. Deneysel değerlendirme, yalnızca soru-cevap modülü üzerinde gerçekleştirilmiş olup, diyet listesi oluşturma modülü bu kapsam dışında tutulmuştur.

#### **A. Retriever-Only (Sadece Bilgi Getirme) Yaklaşımı**

Bu bölümde, soru-cevap modülü kapsamında kullanılan bilgi getirme (retriever) bileşeninin performansı değerlendirilmiştir. Değerlendirme süreci, yalnızca dil modeli kullanılmadan gerçekleştirilmiş ve sistemin doğru bağlamı bulma başarısı ölçülmüştür.

Deneysel, beslenme ve sağlık alanına yönelik **35 sorudan oluşan bir test kümesi** üzerinde yürütülmüştür. Test kümesindeki her soru için beklenen konu başlıklarları (expected topics) tanımlanmış ve geri getirilen dokümanların bu konularla örtüşüp örtüşmediği incelenmiştir.

Bilgi getirme performansı, farklı **k** değerleri için aşağıdaki metrikler kullanılarak değerlendirilmiştir:

- **Hit Rate@k:** İlk  $k$  sonuç içerisinde en az bir ilgili dokümanın bulunup bulunmadığını gösterir.
- **Recall@k:** Beklenen konu başlıklarının ne kadarının geri getirilen sonuçlar içerisinde yer aldığıni ölçer.

Test kümesindeki tüm sorular için beklenen konu bilgisi mevcut olduğundan, değerlendirme sürecinde herhangi bir soru dışlanmamıştır.

k değeri	Soru Sayısı	Hit Rate@k	Recall@k
3	35	0.886	0.686
5	35	0.971	0.757
10	35	0.971	0.814

Elde edilen sonuçlar, bilgi getirme bileşeninin yüksek bir başarıyla çalıştığını göstermektedir. Özellikle  $k=5$  ve  $k=10$  değerlerinde **Hit Rate@k** oranının %97 seviyesine ulaşması, sistemin çoğu soru için en az bir ilgili dokümanı başarıyla geri getirdiğini ortaya koymaktadır.

Ayrıca,  $k$  değeri arttıkça **Recall@k** oranının yükselmesi, daha fazla bağlam kullanıldığında beklenen konuların daha büyük bir kısmının kapsandığını göstermektedir. Bu durum, RAG mimarisinde kullanılacak bağlam sayısının cevap kalitesi üzerinde doğrudan etkili olduğunu desteklemektedir. Retriever-only değerlendirmesi, bilgi getirme bileşeninin yeterli doğrulukta çalıştığını ve dil modeli ile birlikte kullanıldığından (RAG yaklaşımı) güvenilir bir soru-cevap sistemi oluşturulabileceğini göstermektedir.

#### **B. LLM Only Yaklaşımı**

Metrik / Model	Gemma-2:2B	Llama-3:1:8B	Gemma-3:4B	Qwen-2.5:3B
Exact Match (EM)	0.4571	0.2857	0.3143	0.2571
F1 Skoru	0.2793	0.2491	0.2142	0.2068
Must-Have OK	0.2571	0.2286	0.3143	0.2571
Must-Have Recall	0.5548	0.4976	0.5786	0.5214
Must-Not Violation	0.0000	0.0000	0.0857	0.0286
Citation OK	1.00	1.00	1.00	1.00
Hallucination Rate	0.2965	0.1115	0.2389	0.2551
Supported Answer Ratio	0.7035	0.8885	0.7611	0.7449

LLM-only deney sonuçları incelendiğinde, modellerin performansının genel olarak sınırlı kaldığı ve özellikle kural uyumu gerektiren sorularda belirgin zorluklar yaşandığı görülmektedir. En yüksek F1 skoruna sahip model **Llama-3:1:8B** olmuş olsa da (%33), bu değer sistematik ve güvenilir bir performans için yeterli değildir. Buna karşın **Gemma-3:4B**, Exact Match (EM) ve Must-Have metriklerinde daha yüksek değerlere ulaşmış, ancak düşük Must-Not Violation oranı bazı durumlarda kural ihlallerinin göz ardı edildiğini göstermiştir.

Daha küçük modeller olan **Gemma-2:2B** ve **Qwen-2.5:3B**, özellikle Must-Have Recall ve F1 metriklerinde belirgin biçimde daha düşük performans sergilemiştir. Qwen modelinin düşük EM ve F1 değerleri, bağlam desteği olmadan rehber temelli bilgileri tutarlı şekilde üretemediğini göstermektedir.

Worst-case analizleri incelendiğinde, birçok soruda F1 skorunun **0.0** olduğu görülmektedir. Bu durum, LLM-only yaklaşımında modellerin bazı sorulara ya tamamen alakasız cevaplar verdiği ya da zorunlu bilgilerin tamamını kaçırdığını göstermektedir. Özellikle Must-Have Recall değerlerinin sıfırlandığı örnekler, modelin rehber bilgisine erişimi olmadığından kritik diyet kurallarını atladığını ortaya koymaktadır.

#### **C. LLM+RAG Yaklaşımı**

Bu bölümde, bilgi getirme (retriever) bileşeni ile büyük dil modeli (LLM) birlikte kullanılarak oluşturulan **RAG (Retrieval-Augmented Generation)** mimarisinin performansı değerlendirilmiştir. Değerlendirmeler, farklı **K** değerleri (3, 5 ve 10) için gerçekleştirilmiş ve çeşitli LLM modelleri karşılaştırılmıştır. K değeri, cevap üretimi sırasında LLM'e sağlanan geri getirilmiş belge sayısını ifade etmektedir.

Performans değerlendirmesinde aşağıdaki metrikler kullanılmıştır:

- **EM (Exact Match):** Üretilen cevabin, beklenen cevapla birebir eşleşme oranını gösterir.
- **F1:** Üretilen cevap ile beklenen cevap arasındaki örtüşmenin harmonik ortalamasını ifade eder.
- **Must-Have OK:** Cevapta bulunması gereken kritik bilgilerin sağlanması oranını gösterir.

- **Must-Have Recall:** Beklenen zorunlu bilgilerin ne kadarının kapsandığını ölçer.
- **Must-Not Violation:** Cevapta bulunmaması gereken bilgilerin ihlal edilme oranını ifade eder.
- **Cite OK:** Üretilen cevapların kaynaklara uygun şekilde dayandırılıp dayandırılmadığını gösterir.
- **Hallucination:** Modelin desteklenmeyen veya uydurma bilgi üretme oranını ifade eder.
- **Supported Ratio:** Cevapların, geri getirilen belgeler tarafından desteklenme oranını gösterir.

### 1) $K = 3$ için LLM + RAG Model Karşılaştırması

Metrik / Model	gemma2:2b	llama3:1:8b	gemma3:4b	qwen2.5:3b
EM	0.4571	0.2857	0.3143	0.2571
F1	0.2793	0.2491	0.2142	0.2068
Must-Have OK	0.2571	0.2286	0.3143	0.2571
Must-Have Recall	0.5548	0.4976	0.5786	0.5214
Must-Not Violation	0.0000	0.0000	0.0857	0.0286
Cite OK	1.00	1.00	1.00	1.00
Hallucination	0.2965	0.1115	0.2389	0.2551
Supported Ratio	0.7035	0.8885	0.7611	0.7449

$K=3$  için elde edilen sonuçlar, sınırlı sayıda belge kullanıldığında modellerin genel olarak **orta seviyede cevap doğruluğu** sunduğunu göstermektedir. **llama3:1:8b** modeli, düşük **hallucination** oranı ve yüksek **supported ratio** değeri ile dikkat çekmeye, ürettiği cevapların büyük ölçüde kaynak belgeler tarafından desteklendiğini göstermektedir. Buna karşın **gemma2:2b**, daha yüksek EM ve F1 değerlerine rağmen daha yüksek halüsinsiyon oranına sahiptir.

### 2) $K = 5$ için LLM + RAG Model Karşılaştırması

Metrik / Model	gemma2:2b	llama3:1:8b	qwen2.5:3b	gemma3:4b
EM	0.4000	0.4286	0.2857	0.2571
F1	0.2779	0.2709	0.1989	0.1969
Must-Have OK	0.3143	0.2857	0.2571	0.2857
Must-Have Recall	0.5595	0.5310	0.5048	0.5452
Must-Not Violation	0.0000	0.0000	0.0286	0.0857
Cite OK	1.00	1.00	1.00	1.00
Hallucination	0.2578	0.1232	0.2048	0.1917
Supported Ratio	0.7422	0.8768	0.7952	0.8083

Bu tabloda,  $K = 5$  için LLM + RAG yaklaşımında kullanılan modellerin performansları **dikey formatta** sunulmuştur. Bu gösterim, tablonun rapor sayfasına yatay taşıma olmadan yerleştirilmesini **sağlamaktadır**. Sonuçlar incelendiğinde **llama3:1:8b** modelinin düşük halüsinsiyon oranı ve yüksek desteklenme oranı ile güvenilir cevap üretimi açısından öne çıktıığı, **gemma2:2b** modelinin ise F1 skoru bakımından daha başarılı olduğu görülmektedir.

### 3) $K = 10$ için LLM + RAG Model Karşılaştırması

Metrik / Model	gemma2:2b	llama3:1:8b	gemma3:4b	qwen2.5:3b
EM	0.4000	<b>0.4857</b>	0.3714	0.4000
F1	0.2730	0.2719	0.2147	0.1949
Must-Have OK	0.3143	<b>0.3429</b>	<b>0.3429</b>	0.3143
Must-Have Recall	0.5690	0.5690	<b>0.6143</b>	0.5500
Must-Not Violation	0.0286	0.0000	0.0857	0.0286
Cite OK	1.00	1.00	1.00	1.00
Hallucination	0.1841	<b>0.0525</b>	0.2090	0.1620
Supported Ratio	0.8159	<b>0.9475</b>	0.7910	0.8380

$K=10$  için elde edilen sonuçlar, daha fazla bağlam kullanıldığında sistem performansının belirgin şekilde arttığını göstermektedir.

Özellikle **llama3:1:8b** modeli, en yüksek **EM**, **Supported Ratio** ve en düşük **Hallucination** oranı ile en başarılı model olarak öne çıkmıştır.

Bu durum, daha fazla belgeyle desteklenen cevapların hem doğruluk hem de güvenilirlik açısından avantaj sağladığını göstermektedir.

### 4) SONUÇ VE KARŞILAŞTIRMA

Elde edilen deneysel sonuçlar, sistem performansının kullanılan yaklaşım ve geri getirilen belge sayısı ( $K$  değeri) ile doğrudan ilişkili olduğunu göstermektedir. **Retriever-only**, **LLM-only** ve **LLM + RAG** yaklaşımı birlikte değerlendirildiğinde, her bir yaklaşımın güçlü ve sınırlı yönleri daha net biçimde ortaya çıkmaktadır.

Retriever-only yaklaşımında,  $K$  değerinin artmasıyla birlikte **Hit Rate@k** ve **Recall@k** metriklerinde belirgin bir artış gözlemlenmiştir. Özellikle  $K=5$  ve  $K=10$  senaryolarında, sistemin ilgili belgeleri geri getirme başarısı oldukça yükselmiş; bu durum bilgi getirme bileşeninin tek başına dahi güçlü bir kapsama sağlayabildiğini göstermiştir. Ancak bu yaklaşımın herhangi bir üretim adımı bulunmadığından, elde edilen başarı yalnızca belge bulunabilirliği ile sınırlı kalmıştır.

**LLM-only** yaklaşımında ise modellerin dil üretim kabiliyeti öne çıkmakla birlikte, **bağlamsal destek eksikliği** nedeniyle EM ve F1 gibi doğruluk temelli metriklerin görece düşük kaldığı görülmüştür. Ayrıca bazı modellerde halüsinsiyon

oranlarının yükseldiği ve cevapların kaynaklarla yeterince desteklenemediği gözlemlenmiştir. Bu durum, yalnızca LLM kullanımının özellikle klinik veya kural yoğun alanlarda güvenilirlik açısından sınırlı kaldığını ortaya koymaktadır. LLM + RAG yaklaşımı için elde edilen sonuçlar birlikte değerlendirildiğinde ise, **K değerinin artmasıyla sistem performansında genel ve tutarlı bir iyileşme** olduğu görülmektedir. K=3 senaryosunda sınırlı bağlam nedeniyle modeller zaman zaman yüzeysel veya eksik cevaplar üretirken, K=5 seviyesinde cevapların içerik kapsamı genişlemiş, Must-Have Recall ve Supported Answer Ratio gibi metriklerde artış sağlanmıştır. K=10 senaryosunda ise modele sağlanan bağlam miktarının artması, cevapların geri getirilen belgelerle daha güçlü şekilde ilişkilendirilmesini sağlamış ve özellikle **halüsinasyon oranlarında belirgin bir düşüş** gözlemlenmiştir.

Bu bağlam artışı, EM ve Must-Have OK gibi doğruluk odaklı metriklerde iyileşme sağlarken, cevapların kaynaklarla desteklenme oranının da en yüksek seviyelere ulaşmasına katkı sağlamıştır. Bununla birlikte, bazı modellerde K değerinin çok yükselmesi durumunda bağlam fazlalığına bağlı olarak gereksiz veya dolaylı bilgilerin cevaplara eklenmesi de gözlemlenmiştir.

Genel olarak sonuçlar, **en dengeli ve güvenilir performansın LLM + RAG yaklaşımı ile elde edildiğini**, Retriever-only yaklaşımının güçlü bir bilgi kapsamı temeli sunduğunu ve LLM-only yaklaşımının ise bağlam desteği olmadan klinik doğruluk açısından yetersiz kaldığını göstermektedir. Bu bulgular, RAG tabanlı sistemlerde K değerinin modele ve kullanım senaryosuna özgü olarak dikkatli şekilde belirlenmesi gerektiğini açıkça ortaya koymaktadır.

## V. MODÜL 2 — DİYET OLUŞTURMA SÜRECİ (PLANLAMA MODÜLÜ)

Bu modülün amacı, kullanıcı profili ve rehber tabanlı beslenme bilgisini birleştirerek **kural-uyumlu (constraint-driven)** bir diyet planı üretmektir. Sistem, öncelikle ilgili hastalık/condition için rehber dokümanlarından elde edilen kanıtları kullanarak beslenme kurallarını çıkarır ve bu kuralları standart bir temsil formattına dönüştür. Ardından bu kurallar, besin havuzu (food pool) üzerindeki ögün/etiket bilgileriyle eşleştirilerek günlük ve haftalık düzeyde planlama yapılır.

Planlama süreci yalnızca “öneri üretme” yaklaşımıyla sınırlı değildir; modül aynı zamanda **sayısal kısıtları (ör. haftada 2–3 kez)** ve **mantiksal kısıtları (ör. balık–süt ürünü çakışması, öğle/akşam tek ana yemek)** doğrudan plan üzerinde uygular. Böylece oluşturulan program, hem hedef enerji aralığına yaklaşacak şekilde optimize edilir hem de condition'a özgü kuralların ihlal edilmemesi amaçlanır. Son aşamada, üretilen plan belirlenen metriklerle değerlendirilir; model bazlı karşılaştırma raporları oluşturulur ve istenirse plan kullanıcıya PDF formatında sunulur.

### A. Rehber Tabanlı Diyet Kurallarının Üretilmesi (QA Summary)

Bu aşamada amaç, belirli bir hastalık/condition (domain) için beslenme rehberlerinde yer alan bilgileri kullanarak, diyet

planlamasında doğrudan uygulanabilecek **kısa, standart ve kanıt-temelli beslenme kurallarını** üretmektir. Bu süreç, bilgi getirme (retrieval) ve büyük dil modeli (LLM) çıktılarının birlikte kullanıldığı **RAG (Retrieval-Augmented Generation)** yaklaşımı ile gerçekleştirilmiştir. Üretilen kurallar, daha sonraki etiketleme ve planlama adımlarında kullanılmak üzere yapılandırılmış bir özet dosyası (qa\_summary.json) halinde saklanmaktadır.

#### 1) Amaç ve Tasarım İlkeleri

QA Summary üretim sürecinin temel amacı, rehber metinlerinden elde edilen bilgilerin **serbest metin** olarak kalmasını engellemek ve bunları **kontrollü, karşılaştırılabilir ve tekrar kullanılabilir** kural ifadelerine dönüştürmektedir. Bu nedenle sistem, her soru için çıktılarını:

- tek satır halinde olmasını,
- en fazla belirli sayıda kısa ifade içermesini,
- ifadelerin yalnızca geri getirilen kanıtlara dayanmasını,
- modelin yeni bilgi uydurmasını (hallucination) engellemesini

zorunlu kılacak şekilde tasarlanmıştır.

Bu tasarım, farklı LLM modellerinin çıktılarının daha sonra **nesnel olarak karşılaştırılabilmesini** ve planlama modülünde **deterministik kurallar** gibi kullanılabilmesini sağlamaktadır.

#### 2) Soru Şablonları ve Alan Kapsamının Belirlenmesi

Her domain için sorulacak sorular, diyetisyen rehberlerinde sıkça vurgulanan besin gruplarını kapsayacak şekilde önceden tanımlanmıştır. Bu sorular; genel beslenme ilkeleri, protein kaynakları, yağ türleri, lif, tuz/sodyum ve rafine şeker gibi başlıklar altında yapılandırılmıştır. Böylece sistem, yalnızca genel öneriler değil, planlama açısından kritik olan **besin bazlı kuralları** da ayrı ayrı ele alabilmektedir.

Sayısal bilgi gerektiren sorular için sistem otomatik olarak ek soru varyasyonları üretmektedir. Bu varyasyonlar, ilgili besinin **tüketim sıklığı** ve **miktarı** gibi sayısal kısıtların rehberlerden yakalanabilmesini amaçlamaktadır. Bu yaklaşım, ilerleyen aşamalarda haftalık ve günlük kısıtların plan üzerinde uygulanmasına doğrudan katkı sağlamaktadır.

#### 3) Kanıt Getirme ve Katı Filtreleme Mekanizması

Her soru için öncelikle ChromaDB üzerinde vektör tabanlı bir arama gerçekleştirilmekte ve belirlenen sayıda (TOP-K) aday metin parçası elde edilmektedir. Ancak bu adaylar doğrudan kullanılmamaktadır. Sistem, domain'e özgü politika bilgilerini (anchor, alias, domain marker) kullanarak **katı birfiltreleme süreci** uygular.

Bu filtrelere sırasında:

- tablo formatındaki metinler,
- domain dışı veya rakip domain'lere ait içerikler,
- yalnızca belirsiz bağlam içeren parçalar elenmektedir. Böylece LLM'e iletilen kanıtlar, insan uzman gözüyle de **ilgili ve bağlamsal olarak tutarlı** kabul edilebilecek metinlerle sınırlıdır. Bu yaklaşım, özellikle benzer terminolojiye sahip farklı hastalık rehberlerinin karışmasını engellemek amacıyla geliştirilmiştir.

#### **4) Kanıt Özette ve İpuçu (Hint) Çıkarımı**

Filtrelenen metin parçalarından, her bir parça için sınırlı uzunlukta **kanıt özetleri (evidence snippets)** oluşturulmaktadır. Bu özetler hazırlanırken:

- sayısal ifadeler (g, mg, %, kez vb.),
- öneri ve kısıtlama belirten nitel ifadeler öncelikli olarak seçilmektedir.

Elde edilen bu özetler ve ipuçları, büyük dil modeline doğrudan ham metin olarak değil, **kontrollü bir bağlam** olarak sunulmaktadır. Böylece modelin uzun ve dağınık metinler üzerinden çıkarım yapması yerine, rehberin özünü yansıtan kritik cümlelere odaklanması sağlanmaktadır.

#### **5) İnsan Gözüyle Değerlendirme ve Model Seçimi**

QA Summary üretim sürecinde farklı LLM modelleri ile karşılaştırmalı denemeler gerçekleştirilmiştir. Yapılan manuel incelemelerde, Gemma-3:4B modelinin ürettiği kural cümlelerinin dil akıcılığı, bağlam tutarlılığı ve rehber metni yansıtma başarısı açısından daha dengeli olduğu gözlemlenmiştir. Bu nedenle, takip eden planlama ve değerlendirme aşamalarında temel model olarak Gemma-3:4B tercih edilmiştir.

Bu değerlendirme, yalnızca otomatik metriklerde değil, diyetisyen rehberlerini okuyan bir insanın beklediği ifade biçimine olan yakınlığa dayanmaktadır. Böylece sistem çıktılarının yalnızca sayısal olarak değil, pratik uygulanabilirlik açısından da uygun olması hedeflenmiştir.

#### **6) QA Summary Çıktısının Planlama Modülündeki Rolü**

Bu aşamanın sonunda elde edilen qa\_summary.json dosyası, planlama modülünün geri kalan adımları için **temel bilgi kaynağı** niteliğindedir. Dosya; her soru için üretilen kural metnini ve bu kuralların dayandığı rehber parça kimliklerini içermektedir. Böylece ilerleyen aşamalarda yapılan etiketleme, sayısal kısıt çıkarımı ve diyet planı üretimi, doğrudan bu kanıt-temelli özetler üzerine inşa edilmektedir.

#### **B. QA Summary'den Yapılandırılmış Etiket/Kural Çıkarımı**

Bu aşamada amaç, Modül 2'nin A bölümünde üretilen qa\_summary.json çıktısındaki doğal dil kural cümlelerini işleyerek, planlama motorunun doğrudan kullanabileceğini **yapılardırılmış bir etiket ve kural temsiline** dönüştürmektedir. Bu dönüşüm süreci tamamen Python tabanlı yürütülmekte; LLM kullanılmadan (LLM-free) kural çıkarımı yapılmaktadır. Böylece sistemin karar mekanizması daha deterministik hale getirilmekte ve farklı modellerin ürettiği metinlerin aynı şemaya göre karşılaştırılabilmesi sağlanmaktadır.

Bu bölümde sistem, tag\_dicts.json dosyasındaki tag sözlüğünü referans alarak otomatik bir alias indeksi oluşturur ve QA cevaplarını madde madde (clause) analiz eder. Her clause için; (i) ilgili tag'lerin bulunması, (ii) tag'in bağlamına göre "prefer/limit/avoid" niyetinin belirlenmesi, (iii) sayısal ifadelerden haftalık/günlük kısıtların çıkarılması ve (iv) her kuralın dayandığı kanıt cümlesinin "rag\_evidence" alanında saklanması adımları uygulanır. Son aşamada çıktı, kalite kontrolleri ve tutarlılık düzeltmeleri (vocab filtresi, kanıt doğrulama, çakışan tag gruplarının tekilleştirilmesi, numeric

constraint normalizasyonu) ile temizlenerek domain bazlı labels.json dosyasına kaydedilir. Böylece planlama modülünün devamındaki adımlar, serbest metin yerine doğrulanmış ve standartlaştırılmış kurallar üzerinden yürütültür.

#### **1) Girdi–Çıktı ve Dosya Organizasyonu**

Bu adımda sistemin kullandığı giriş dosyaları ve üretilicek çıktı yapısı tanımlanmıştır. Girdi olarak, her domain için daha önce üretilmiş qa\_summary.json dosyası ve tag sözlüğünü içeren tag\_dicts.json kullanılmaktadır. Kod, BASE\_DIR altında domain klasörlerini otomatik olarak keşfeder (ör. her condition için ayrı klasör) her domain'i sırayla işlemektedir. Çıktılar proje kök dizini altında labels/<domain>/python\_only/ klasörüne yazılmakta; burada ana çıktı labels.json, süreç özeti ise report.json olarak saklanmaktadır. Ayrıca tüm domain çalışmalarının genel bir indeksi \_index.json dosyasında tutulmaktadır. Bu yapı, hem deney tekrarlanabilirliğini artırmakta hem de model bazlı farklı QA çıktılarının sonraki modüllerde düzenli biçimde kullanılmasını sağlamaktadır.

#### **2) Tag Sözlüğü ve Alias İndeksi Oluşturma**

Etiketleme sürecinin temel gereksinimi, serbest metin içindeki ifadeleri tag'lerle eşlestirebilmektir. Bu amaçla sistem, tag\_dicts.json içindeki tag açıklamalarını kullanarak otomatik bir alias indeksi oluşturur. Alias üretimi; (i) açıklama içinde parantezle verilen eş anlamlılar, (ii) açıklamadaki anlamlı kelime token'ları ve (iii) Türkçe eklerden kaynaklanan varyasyonları yakalamak için uygulanan basit "stem-lite" yaklaşımıyla desteklenmiştir. Buna ek olarak, doğruluğu artırmak için bazı kritik tag'ler için manuel anahtar kelime zenginleştirmesi (TAG\_KEYWORD\_MAP) uygulanmaktadır; bu adım yeni tag icat etmez, yalnızca mevcut tag'lerin alias kapsamını genişletir.

Alias eşlemenin verimli yapılabilmesi için iki katmanlı bir strateji izlenmektedir: Çok kelimeli alias'lar (ör. "kirmizi et") doğrudan substring eşleşmesi ile yakalanırken, tek kelimeli alias'lar için geniş kapsamlı bir regex indeksi üretilmektedir. Ancak tek kelimeli alias'lar içinde çok genel/çakışmaya açık olanlar (ör. "et") birden fazla tag'e gidiyorsa regex listesine alınmayarak yanlış eşleşmeler azaltılmaktadır. Buna karşılık, sistemin kaçırılmaması gereken kritik terimler PRIORITY\_ALIASES listesi ile güvence altına alınmıştır. Bu tasarım, yüksek kapsam (recall) ile düşük yanlış eşleşme (precision) arasında denge kurmayı hedefler

#### **3) QA Cevaplarının Clause'lara Ayrılması**

qa\_summary.json içindeki cevaplar tek satır olup ";" ayıracıyla çoklu kural içerdiginden, etiketleme öncesinde bu metinler işlenebilir parçalara ayrılmaktadır. Sistem önce ";" ayıracını bullet formata dönüştürerek her maddeyi ayrı bir satır olarak ele alır. Ardından satır uzunluğu çok fazla olan ifadeler, noktalama işaretleri üzerinden cümlelere bölünür. Çok kısa ve anlam taşımayan parçalar (örn. 3-4 kelimedenden kısa olanlar) filtrelenerek gürültü azaltılır. Böylece her clause, tek bir öneri/kısıt ifadesini temsil eden daha temiz bir metin birimi haline gelir.

**4) Tag Eşleme Mekanizması (Span Tabanlı Yaklaşım)**  
Her clause üzerinde tag eşleme işlemi, alias indeksi kullanılarak yapılır. Bu eşleme yalnızca “tag bulundu/bulunmadı” şeklinde değil, aynı zamanda eşleşmenin metin içindeki konumu (span: başlangıç-bitiş indexi) ile birlikte tutulur. Çok kelimeli alias’lar önce yakalanarak daha spesifik eşleşmelerin korunması sağlanır; ardından tek kelimeli alias regex’i uygulanır. Span bilgisinin tutulmasının temel nedeni, özellikle sayısal kısıt çıkarımı sırasında sayısal ifade ile en ilgili tag’ın “yakınlık” temelli bağlanabilmesidir. Bu sayede, aynı clause içinde birden fazla tag geçtiğinde sayısal değerlerin yanlış tag’e bağlanması olasılığı azaltılmış olur.

**5) Niyet (Intent) Sınıflandırma: Prefer / Limit / Avoid**  
Tag tespitinden sonra her tag için sistem, ilgili clause içindeki ifadenin niyetini (prefer/limit/avoid) belirler. Bunun için üç ayrı regex kümesi kullanılır: tercih/öneri sinyalleri (PREFER\_PAT), sınırlama sinyalleri (LIMIT\_PAT) ve kaçınma/yasak sinyalleri (AVOID\_PAT). Ayrıca Türkçedeki olumsuzluk yapıları (NEGATION\_PAT) kontrol edilerek “tercih edilmemeli” gibi ifadelerin yanlışlıkla “prefer” sınıfına düşmesi engellenir.

Intent tespiti tüm cümle üzerinden değil, tag’ın bulunduğu span çevresindeki belirli token penceresi üzerinden yapılır (intent\_in\_window). Böylece cümlede farklı bölgelerde geçen çelişkili ifadelerin niyet tespitini bozması azaltılır. Ek olarak, tag\_dicts.json açıklamalarından çıkarılan tag sınıfı (risk/positive/neutral) ile bir koruma mekanizması uygulanır: “risk” sınıfındaki tag’ların yanlışlıkla “prefer” olması engellenir (gerekirse limit’e düşürülür), “positive” sınıfındaki tag’ların de doğrudan “avoid” yapılması yemdirilir. Bu adım, kuralların semantik tutarlığını güçlendirmektedir.

#### **6) X yerine Y” Yapısının Çıkarımı (Hybrid NLP: Regex + Stanza)**

Beslenme rehberlerinde sıkça geçen “X yerine Y tercih edilmeli” kalıbı, basit niyet tespitinden daha güçlü bir yönlendirme içerir. Bu nedenle sistem, clause içinde “yerine” yapısını ayrıca analiz eder. Basit cümlelerde regex ile “X yerine Y” çiftleri çıkarılır. Ancak cümle ağır bağlaçlar içeriyorsa (ör. “ancak”, “fakat”, “koşuluyla”, “aksi halde”), yalnızca regex ile yanlış bölme riski artacağından Stanza bağımlılık çözümleyici **on-demand** şekilde devreye sokulur (lazy init). Bu tasarım, hem doğruluk hem de çalışma verimliliği açısından dengelidir. Elde edilen çiftlerde genel mantık şudur: X tarafı çoğunlukla azaltılacak/kaçınılacak (avoid/limit), Y tarafı ise önerilecek (prefer) olarak işaretlenir. Bununla birlikte, X veya Y tarafı tag sınıfına göre tekrar kontrol edilerek (risk/positive) aşırı sert veya hatalı sınıflamalar yemdirilir. Bu bölüm, rehber dilindeki “yerine tercih” yönlendirmelerinin planlama kurallarına daha doğru taşınmasını sağlar.

#### **7) Sayısal Kısıt Çıkarımı (Numeric Constraints)**

Planlama için kritik bileşenlerden biri, rehberlerde geçen sayısal önerilerin (örn. “haftada 2–3 kez”, “günde 5 g”,

“%20–35”) yapılandırılmış kısıtlara dönüştürülmesidir. Sistem, bu amaçla numeric sinyalleri tespit ederek clause’u sayısal segmentlere böler ve her segmentten bir constraint üretmeye çalışır. Ünite tespiti (gram, mg, yüzde, kez, porsiyon vb.) ve dönem tespiti (gün/hafta/ay → period\_days) segment içinden çıkarılır. “en az / en fazla” gibi ifadeler min-max ayrimını belirlemeye kullanılır; aralık ifadeleri (2–3, %20–35) ayrı regex’lerle yakalanır.

Bu aşamanın en önemli tasarım kararı, **segment-local tag binding** yaklaşımıdır: Sayısal segment içinde geçen tag eşleşmeleri dışına çıkmadan tag seçilir. Böylece aynı clause içinde birden fazla tag geçtiğinde sayısal değerlerin yanlış tag’e bağlanma riski azaltılır. Eğer segment makro düzeyde bir kavram içeriyorsa (sodyum, trans, GI, lif, KH vb.), proximity yerine “macro hint” ile daha doğru tag’e bağlama yapılır. Segment içinde açık tag bulunamazsa fuzzy window yöntemiyle alias/stem benzerliğine dayalı bir tahmin denenir. Buna rağmen tag bulunamazsa segment numeric\_constraints alanına yazılır; bunun yerine “unmatched\_numeric” olarak recommendations listesine atılır. Bu karar, planlama motoruna yanlış kısıt enjekte edilmesini önlemek amacıyla alınmıştır.

#### **8) Doğrulama, Temizleme ve Tutarlılık Sağlama (Validation + Fix)**

Çıkarılan tüm tag ve constraint’ler, çıktı kalitesini garanti etmek için çok aşamalı doğrulama ve düzeltme adımlarından geçirilir. Öncelikle vocab\_set dışındaki tag’ler temizlenir. Ardından rag\_evidence alanına eklenen her alıntıın gerçekten ilgili tag’i destekleyip desteklemediği, alias/stem tabanlı hızlı bir doğrulama (quote\_supports\_tag) ile kontrol edilir; desteklenmeyen kanıtlar çıkarılır. Kanıtların aşırı tekrarını engellemek için (i) aynı alıntı üzerinde tutulabilecek tag sayısı ve (ii) tag başına kanıt sayısı sınırlanır.

Numeric constraints tarafından birimler normalize edilir ve geçersiz birimler elenir. Numeric constraint’e bağlanan tag’lerin en az “limit” sınıfında bulunması sağlanır (numeric bir kısıt varsa sistem bu tag’ı kontrol altına almak zorundadır). Daha sonra kanıt kapsamına göre tag listeleri yeniden inşa edilir; böylece kanıtı olmayan tag’ler nihai listeden düşer. Son olarak çakışan tag grupları için mutex uygulanır (örn. sodyum\_düşük/orta/yüksek aynı anda kalamaz), “prefer-limit-avoid” kümeleri arasında hiyerarşi korunur (avoid > limit > prefer) ve engine\_rules alanı sadece doğrulanmış tag’lerle eşleşecek şekilde temizlenir.

Bu doğrulama adımları sonucunda her domain için labels.json çıktısı hem semantik açıdan tutarlı, hem de planlama motorunun doğrudan kullanabileceği şekilde standartlaştırılmış olur. Ek olarak üretilen report.json dosyası, kaç clause işlendiği ve hangi düzeltmelerin yapıldığı gibi süreç bilgilerini kayıt altına alarak deneyin izlenebilirliğini artırır.

#### **C. Diyet Planı Üretimi ve LLM Audit Patch Entegrasyonu (Weekly/Monthly Planner)**

Bu modülün amacı, önceki aşamalarda oluşturulan **NLP tabanlı labels.json kurallarını** temel olarak kullanıcından alınan profil bilgilerine göre **haftalık veya aylık beslenme planı** üretmektir. Plan üretimi, (i) kullanıcıya özgü enerji ihtiyacını hesaplayan bir “nutritional math” katmanı, (ii) besin havuzundan kural uyumlu seçim yapan bir planlayıcı, (iii) sayısal kısıtları (özellikle haftalık tüketim sınırları) zorunlu kılan bir enforcement mekanizması ve (iv) opsiyonel olarak LLM ile QA tabanlı kontrol yapıp yalnızca eksik kısıtları “patch” olarak ekleyen bir denetim adımından oluşur. Çıktı hem terminalde görüntülenebilmekte hem de istenirse PDF raporu olarak üretilibilmektedir.

### 1) Kural Kaynağı ve LLM Audit Patch Yaklaşımı

Planlama sürecinde Esas alınan temel kural kaynağı, NLP tabanlı etiketleme modülünden elde edilen labels.json dosyasıdır. Bu dosya, sistemin deterministik davranışmasını sağlamak amacıyla **değiştirilmeyen ana referans** olarak kabul edilmektedir. Büyük dil modeli bu aşamada kuralları yeniden üretmez; yalnızca QA Summary içerisinde bulunmasına rağmen labels.json içerisinde eksik kalmış olan kural ve özellikle sayısal kısıtları tamamlamak için kullanılır. LLM, bu amaçla yalnızca “**patch**” formatında çıktı üretir. Patch çıktısı, mevcut etiketleri silmeden veya değiştirmeden, eksik olan numeric\_constraints, prefer, limit ve avoid etiketlerini ekler. Bu yaklaşım, LLM kaynaklı halüsinsiyon riskini azaltırken, rehber bilgisi ile yapılandırılmış kural seti arasındaki tutarlılığı artırmaktadır. Revize edilen etiketler, orijinal dosyanın üzerine yazılmadan ayrı bir dizinde saklanmakta ve planlama aşaması bu revize kurallar üzerinden yürütülmektedir.

### 2) Kullanıcı Profili ve Enerji İhtiyacı Hesaplama

Planlama süreci, kullanıcının alınan demografik ve antropometrik bilgilerle başlar. Cinsiyet, yaşı, boyu, kilo ve fiziksel aktivite düzeyi kullanılarak Bazal Metabolizma Hızı (BMR) Mifflin–St Jeor formülü ile hesaplanır. Ardından aktivite katsayısı uygulanarak toplam günlük enerji ihtiyacı (TDEE) elde edilir. Kullanıcının belirlediği kalori deficitinin oranı bu değere uygulanarak günlük hedef kalori miktarı belirlenir. Bu hedef, planın sayısal temelini oluşturarak günün seçimlerinin kalori açısından yönlendirilmesini sağlar.

### 3) Besin Havuzu ve Öğün Yapısının Standartlaştırılması

Besin havuzu, farklı JSON formatlarını destekleyecek şekilde normalize edilerek tek tip bir yapı haline getirilir. Her besin için enerji değeri, etiketler, kaçınılması gereken tag'ler ve uygun olduğu öğün bilgileri standartlaştırılır. Öğün yapısı sabitlenmiş bir sıraya göre tanımlanır: sabah, ara\_1, öğle, ara\_2, ara\_3, akşam ve yatarken. Bu sabit yapı, hem plan üretiminin hem de değerlendirme aşamasındaki karşılaştırmaları kolaylaştırmaktadır.

### 4) Plan Üretim Mekanizması ve Kural Uyumlu Seçim

Diyet planı, her gün için ayrı ayrı oluşturulmakta ve besinler heuristik bir skorlamaya dayalı olarak seçilmektedir. Bu

skorlamada, prefer etiketleri pozitif, limit etiketleri negatif etki yaratırken, avoid etiketleri kesin olarak elenmektedir. Ayrıca haftalık sayısal kısıtları henüz karşılanmamış etiketler için ek bir ödül mekanizması uygulanarak sistemin haftalık hedeflere yönelik yapılması sağlanmaktadır.

Planlama sırasında gerçek hayatı beslenme alışkanlıklarını yansıtmak amacıyla bazı yapısal kurallar zorunlu tutulmuştur. Türk kahvaltısı sabah öğünü için teşvik edilmekte, balık ve süt ürünlerinin aynı öğünde yer olması engellenmekte ve öğle ile akşam öğünlerinde yalnızca tek ana yemek bulunmasına izin verilmektedir. Ara öğünlerin boş bırakılmaması da sistem tarafından kontrol edilmektedir.

### 5) Haftalık Sayısal Kısıtların Zorlanması Uygulanması

Rehberlerde yer alan “haftada 2–3 kez” gibi sayısal ifadeler, planlama sürecinde yalnızca bilgilendirici olarak kalmamakta, doğrudan enforce edilmektedir. period\_days = 7 olarak tanımlanan sayısal kısıtlar, haftalık bazda izlenir. Eğer bir etiket için maksimum sınır aşılmışsa ilgili besinler sonraki seçimlerde yasaklanır. Minimum sınır karşılanmamışsa, sistem hafta içinde uygun öğünlere bu etiketi taşıyan besinleri eklemeye çalışır. Bu yaklaşım, planın rehberlerle uyumlu olmasını pratik düzeyde garanti altına almaktadır.

### 6) Porsiyonlandırma ve Kalori Ölçekleme

Besinlerin miktar bilgileri farklı birimlerde ifade edildiğinden, her besin için otomatik bir porsiyon modeli çıkarılmaktadır. Gram, adet, dilim, kepçe veya kaşık gibi birimler normalize edilmekte ve besinin ölçülebilir olup olmadığına göre porsiyon aralıkları belirlenmektedir. Bu model sayesinde her öğünün ve günün toplam kalori değeri hesaplanabilmekte ve hedef kalori ile karşılaşırılabilmektedir.

### 7) Çıktı Sunumu ve PDF Oluşturma

Planlama sürecinin sonunda sistem, QA Summary'ye dayalı olarak Türkçe bir beslenme açıklaması üretir ve ardından diyet planını kullanıcıya sunar. Çıktı hem terminal üzerinden görüntülenebilir hem de istenirse PDF formatında oluşturulabilir. PDF çıktısında her gün ayrı bir sayfa olarak sunulmakta, yalnızca açıklama ve plan yer almaktır; teknik etiket listeleri kullanıcının gizlenmektedir. Bu tasarım, çıktının doğrudan danişanlara sunulabilecek nitelikte olmasını amaçlamaktadır.

### 8) Otomatik Değerlendirme ve Model Karşılaştırması

Üretilen planlar, kural uyumu ve sayısal doğruluk açısından otomatik olarak değerlendirilmektedir. Günlük kalori sapmaları, kaçınılması gereken besin ihlalleri, zorunlu ara öğünlerin durumu ve haftalık sayısal kısıtların başarı oranı gibi metrikler hesaplanır. Bu sonuçlar her model için ayrı JSON raporları halinde saklanır ve farklı LLM modellerinin planlama başarımı karşılaştırmalı olarak analiz edilir. Böylece sistem, yalnızca plan üretmeye kalmayıp, model bazlı performans değerlendirmesine de olanak tanımaktadır.

### 9) Modellerin Karşılaştırmalı Analizi (Kolesterol Senaryosu)

Bu çalışmada kolesterol senaryosu için diyet planı üretiminde farklı büyük dil modelleri kullanılarak, model seçiminin plan çıktısı üzerindeki etkisi karşılaştırılmış olarak incelenmiştir. Tüm modeller aynı besin havuzu, aynı NLP tabanlı kural seti (labels.json) ve aynı planlama algoritması ile çalıştırılmıştır. Büyük dil modelleri bu aşamada doğrudan plan üretmemekte; yalnızca QA Summary içeriğini denetleyerek mevcut kural setinde eksik kalan sayısal veya mantıksal kurallar varsa “patch” formatında eklemeler yapmaktadır. Bu nedenle karşılaştırma, esas olarak LLM denetim aşamasının planlama çıktısına dolaylı etkisini değerlendirmeyi amaçlamaktadır. Elde edilen sonuçlara göre Gemma-2:2B, Qwen-2.5:3B, Gemma-3:4B, Llama-3.1:8B ve Mistral-7B modellerinin tamamı için plan çıktısı metrikleri büyük ölçüde aynı kalmıştır. Günlük hedef kaloriye olan ortalama mutlak sapma tüm modellerde yaklaşık 273 kcal düzeyinde ölçülülmüş, hedef kalorinin  $\pm 10\%$  aralığında kalınan gün oranı ise %28.57 olarak hesaplanmıştır. Bu durum, hedef kaloriye yakınsama performansının modelden bağımsız olarak planlama ve porsiyon ölçekleme mekanizmasına bağlı olduğunu göstermektedir.

Kural ihlalleri açısından değerlendirildiğinde, tüm modellerde kaçınılmazı gereken besinlere ilişkin sınırlı sayıda ihlal gözlenmiş, ancak balık ve süt ürünlerinin aynı öğrenci yer almaması gibi kritik kombinasyon kuralları tüm planlarda başarıyla korunmuştur. Zorunlu ara öğünlerin boş kalma metriğinin tüm modellerde yüksek olması ise, ara öğünlerin plan şemasında tanımlı olmasına rağmen aday besin seçimi ve ölçüm katmanında sistematik bir sorun bulunduğuuna işaret etmektedir.

Haftalık sayısal kısıt değerlendirmelerinde bazı modeller için “ok=false” sonucu elde edilmesine rağmen, planların içerik açısından rehberlere uygun olduğu görülmektedir. Bu durum, haftalık balık, yumurta ve baklagıl gibi besin gruplarına ilişkin kısıtların planlayıcı tarafından doğru şekilde uygulanmasına rağmen, NLP tabanlı kural çıkarması sürecinde aynı rehber bilgisinin birden fazla kez yakalanması sonucu ortaya çıkan tekrarlı sayısal kısıtların değerlendirme aşamasında ayrı kurallar gibi ele alınmasından kaynaklanmaktadır. Dolayısıyla haftalık başarısızlık raporları, planlama ya da model hatasından ziyade, sayısal kısıtların otomatik çıkarımı sonrasında gerekli olan tekilleştirme ve birleştirme adıminin eksikliğini yansımaktadır.

Sonuç olarak kolesterol senaryosunda model karşılaştırması, kullanılan büyük dil modelinin planın sayısal ve kural uyumlu yapısını doğrudan belirlemediğini, plan kalitesinin büyük ölçüde kural setinin yapısı ve planlama algoritmasının heuristiklerine bağlı olduğunu göstermiştir. Büyük dil modelleri arasında belirgin farklar daha çok QA açıklama metinlerinin dili ve üslubunda ortaya çıkmış, ancak planın nicel performans metrikleri üzerinde anlamlı bir farklılık gözlenmemiştir. Bu bulgular, sistemin deterministik bir planlama yaklaşımı benimsedigini ve LLM’lerin bu mimaride daha çok denetleyici ve destekleyici bir rol üstlendiğini ortaya koymaktadır.

## VI. GENEL SONUÇ VE DEĞERLENDİRME

Bu çalışma kapsamında geliştirilen sistem, beslenme rehberlerinden ve tıbbi metinlerden elde edilen bilgileri NLP tabanlı etiketleme yapısı ile sayısal ve mantıksal kurallara

dönüştürerek, kişiye özel diyet planları üretebilen hibrit bir mimari sunmaktadır. Sistem, deterministik bir planlama çemberdeği ile büyük dil modellerini (LLM) destekleyici ve denetleyici bir katman olarak birlikte kullanarak, hem tekrar edilebilir hem de açıklanabilir çıktılar üretmeyi amaçlamaktadır.

Elde edilen sonuçlar, diyet planlarının temel doğruluğunun ve kural uyumunun büyük ölçüde kural setinin kalitesine, sayısal kısıtların doğru tanımlanmasına ve planlama algoritmasının heuristiklerine bağlı olduğunu göstermektedir. Büyük dil modelleri bu mimaride doğrudan plan üreticisi olarak değil, daha çok bilgi denetimi, eksik kural tespiti ve doğal dilde açıklama üretimi görevlerini üstlenmiştir. Modeller arasında yapılan karşılaştırmalar, plan çıktılarının nicel performans metrikleri açısından büyük ölçüde benzer olduğunu, farklılıkların daha çok QA açıklamalarının dili ve ifade biçiminde ortaya çıktığını göstermiştir.

Sistem genelinde tespit edilen en önemli sınırlıklardan biri, NLP tabanlı bilgi çıkarımı sonucunda aynı rehber bilgisinin farklı cümle yapılarıyla birden fazla kez yakalanabilmesidir. Bu durum özellikle haftalık sayısal kısıtların değerlendirilmesinde, gerçekte doğru uygulanan kuralların analiz katmanında “başarısız” olarak işaretlenmesine yol açabilmektedir. Bu bulgu, planlama hatasından ziyade, kural çıkarımı sonrasında uygulanması gereken daha güçlü bir semantik tekilleştirme ve çelişki çözümleme mekanizmasına ihtiyaç olduğunu ortaya koymaktadır.

Soru (QA) modülü açısından bakıldığından, sistemin mevcut hali rehber metinlerinden beslenme önerilerini başarıyla özetleyebilmekte ve diyet planlarının gerekliliklerini kullanıcıya açıklayabilmektedir. Bununla birlikte, QA modülünün ilerleyen aşamalarda kaynak metinlere referans verebilen, farklı rehberler arasındaki çelişkileri işaretleyebilen ve güven seviyesine göre cevaplarını derecelendirebilen bir yapıya evrilmesi mümkündür. Ayrıca QA çıktılarının, doğrudan kural setine bağlanarak hangi cümlenin hangi sayısal veya mantıksal kuralı tetiklediğinin izlenebilir hale getirilmesi, sistemin şeffaflığını ve klinik güvenilirliğini artıracaktır.

Diyetisyen (planlama) modülü açısından değerlendirildiğinde ise sistemin güçlü yönü, haftalık ve günlük kısıtları aynı anda ele alabilen, ögün bazlı mantık kurallarını uygulayabilen ve besin karışımalarını otomatik olarak engellebilen yapısıdır. Buna karşın, ara öğünlerin doldurulması, kalori hedeflerine daha sıkı yakınsama sağlanması ve kullanıcı tercihleri ile klinik kısıtlar arasında daha esnek bir denge kurulması geliştirmeye açık alanlar olarak öne çıkmaktadır. Ayrıca planlama çıktılarının uzun vadeli çeşitlilik, alışkanlık tekrarı ve sürdürülebilirlik açısından değerlendirilmesi, sistemin gerçek hayatı kullanımını güçlendirecektir.

Genel olarak değerlendirildiğinde, bu çalışma LLM’lerin doğrudan karar verici olmadığı, ancak tıbbi bilgi işleme ve açıklama üretimi süreçlerinde güvenli bir yardımcı katman olarak konumlandırıldığı bir diyet planlama yaklaşımının uygulanabilirliğini göstermektedir. Gelecekte yapılacak geliştirmelerle birlikte sistemin hem klinik doğruluğu hem de kullanıcı deneyimi açısından daha güçlü, daha açıklanabilir ve daha ölçülebilir bir yapıya kavuşması beklenmektedir.

önerilen mimarının genellenebilirliğini test etmek açısından değerli olacaktır.

## VII. GELECEK ÇALIŞMALAR

Bu çalışma, LLM destekli ve kural tabanlı hibrit bir diyet planlama sisteminin uygulanabilirliğini ortaya koymakla birlikte, sistemin doğruluğunu, ölçeklenebilirliğini ve klinik geçerliliğini artırmaya yönelik çeşitli geliştirme alanları barındırmaktadır. Gelecek çalışmalarında öncelikli olarak ele alınabilecek konular aşağıda özetlenmektedir.

İlk olarak, NLP tabanlı bilgi çıkarım sürecinin daha gelişmiş bir semantik normalizasyon katmanı ile desteklenmesi hedeflenmektedir. Mevcut yapıda, aynı beslenme önerisinin farklı ifade biçimleriyle birden fazla sayısal kısıt olarak sisteme aktarılması mümkün değildir. Bu durum, haftalık kural değerlendirmelerinde gereksiz çelişkilere ve yanlış negatif sonuçlara yol açabilmektedir. Gelecekte, kural çıkarımı sonrasında çalışan bir “semantik tekilleştirme ve çelişki çözümleme” modülü ile aynı anlama gelen kuralların otomatik olarak birleştirilmesi ve önceliklendirilmesi planlanmaktadır.

İkinci olarak, QA (soru-cevap) modülünün yalnızca özetleme yapan bir yapıdan, kaynak metinlere açık referanslar verebilen ve farklı rehberler arasındaki bilgi tutarlığını değerlendirebilen bir sisteme dönüştürülmesi hedeflenmektedir. Bu kapsamında, her üretilen beslenme kuralının hangi rehber cümlelerinden türetildiğinin izlenebilir hale getirilmesi ve kullanıcuya güven düzeyi ile birlikte sunulması önemli bir geliştirme alanı olarak önemli bir parçadır. Diyet planlama modülü açısından bakıldığından, mevcut deterministik planlama yaklaşımının, uzun vadeli çeşitlilik ve sürdürülebilirlik kriterlerini daha iyi gözeten optimizasyon yöntemleriyle desteklenmesi planlanmaktadır. Özellikle kalori hedeflerine daha hassas yakınsama sağlayan, ögünler arası dağılımı dinamik olarak ayarlayabilen ve kullanıcının önceki planlara verdiği geri bildirimleri öğrenebilen adaptif algoritmalar gelecekteki çalışmaların önemli bir parçası olabilir.

Buna ek olarak, sistemin kişiselleştirme kapasitesinin artırılması hedeflenmektedir. Kullanıcı tercihleri, kültürel beslenme alışkanlıkları, ekonomik erişilebilirlik ve bireysel toleranslar gibi faktörlerin kural setine daha derinlemesine entegre edilmesi, sistemin gerçek hayatı kullanımını güçlendirecektir. Bu bağlamda, kullanıcı geri bildirimlerine dayalı yarı-denetimli öğrenme yaklaşımıları değerlendirilebilir.

Son olarak, sistemin klinik geçerliliğinin değerlendirilmesi amacıyla gerçek kullanıcılar veya diyetisyenler ile yapılacak pilot çalışmalar önemli bir gelecek adımıdır. Uzman geri bildirimleri doğrultusunda kural setlerinin ve planlama stratejilerinin iyileştirilmesi, sistemin tıbbi karar destek sistemi olarak güvenilirliğini artıracaktır. Ayrıca farklı hastalık profilleri için (örneğin diyabet, hipertansiyon, obezite) benzer değerlendirme senaryolarının uygulanması,

**IEEE conference templates contain guidance text for composing and formatting conference papers. Please ensure that all template text is removed from your conference paper prior to submission to the conference. Failure to remove template text from your paper may result in your paper not being published.**

We suggest that you use a text box to insert a graphic (which is ideally a 300 dpi TIFF or EPS file, with all fonts embedded) because, in an MSW document, this method is somewhat more stable than directly inserting a picture.

To have non-visible rules on your frame, use the MSWord “Format” pull-down menu, select Text Box > Colors and Lines to choose No Fill and No Line.