

İSTANBUL AYDIN ÜNİVERSİTESİ UYGULAMALI BİLİMLER FAKÜLTESİ

FARKLI PROMPT STRATEJİLERİNİN PERFORMANSLARININ
BÜYÜK DİL MODELLERİNDE
KARŞILAŞTIRMALI ANALİZİ

BİTİRME PROJESİ

ELİFNUR ŞİRİN

B2180.010039

YÖNETİM BİLİŞİM SİSTEMLERİ LİSANS PROGRAMI

DR.ÖĞR. ÜYESİ ERAY KARASLAN

HAZİRAN, 2025

ÖNSÖZ

Bu çalışmada, büyük dil modelleri (LLM) ve bu modellere yönelik geliştirilen farklı prompt stratejileri incelenerek, büyük dil modellerinin çıktı kalitesine etki eden faktörler analiz edilmiştir. Konunun seçiminde, güncel teknoloji trendleriyle olan ilgimin yanı sıra, Yönetim Bilişim Sistemleri alanına somut katkı sağlayacak bir boşluğu doldurma arzusu etkili olmuştur.

Yönetim Bilişim Sistemleri disiplini, teknolojik gelişmelerin iş süreçleri üzerindeki etkilerini anlamaya ve bu etkiler doğrultusunda yenilikçi çözümler üretmeye odaklanan çok yönlü bir alandır. Bu bağlamda, dijital dönüşümün temel yapı taşlarından biri haline gelen yapay zekâ teknolojileri hem akademik hem de sektörel bağlamda önemli araştırma konuları arasına girmiştir.

Çalışma süreci boyunca, özellikle prompt mühendisliği gibi yeni ve gelişmekte olan bir alanda sistematik kaynaklara ulaşmak zaman zaman güçleşse de bu durum araştırmanın özgünlüğünü artırılmış ve çözüm üretme motivasyonumu pekiştirmiştir.

Gelişmekte olan bu alana özgün bir katkı sunabilmek için detaylı bir literatür taraması gerçekleştirek mevcut akademik yaklaşımları inceledim ve bunları pratik uygulamalarla destekleyerek sistemli bir analiz ortaya koymaya çalıştım. Elde ettiğim bulgular sonucunda, her bir prompt stratejisinin farklı dil modellerinde ne kadar etkili olduğunu karşılaştırınan tablolar ve grafikler oluşturmaya özen gösterdim.

Bu çalışmanın hazırlanışından ortaya çıkışına kadar akademik bilgisi ve deneyimiyle bana rehberlik eden tez danışmanım Sayın *Dr. Öğr. Üyesi ERAY KARAASLAN'a* en içten teşekkürlerimi sunarım. Aynı zamanda, araştırma sürecinde katkıları ve yapıcı geri bildirimleriyle beni destekleyen sevgili arkadaşım *Kalender ÇAKMAK'a* da teşekkür ederim.

Bu tezin, prompt mühendisliği ve büyük dil modellerine ilgi duyan herkes için değerli bir kaynak olmasını temenni ederim.

Haziran, 2025

Elifnur ŞİRİN

İÇİNDEKİLER

ÖNSÖZ	i
ÖZET	1
GİRİŞ	2
I. GENEL BAKIŞ	3
A. BÜYÜK DİL MODELLERİNİN EVRİMİ	3
B. GPT-4, GROK, CLAUDE, DEEPSEEK MODELLERİNİN İNCELENMESİ	4
1. <i>GPT-4</i>	4
2. <i>Grok</i>	4
3. <i>Claude</i>	5
4. <i>DeepSeek</i>	5
C. STRATEJİK YAKLAŞIMLAR	5
1. <i>Zero-shot Learning (Sıfır Örnekli Öğrenme)</i>	5
2. <i>One-shot Learning (Tek Örnekli Öğrenme)</i>	6
3. <i>Few-shot Learning (Birkaç Örnekli Öğrenme)</i>	6
4. <i>Chain-of-Thought (Düşünce Zinciri)</i>	6
5. <i>Direct vs Indirect Prompting (Doğrudan ve Dolaylı Prompt)</i>	6
D. MODEL EĞİTİMİ VE VERİ KÜMELERİ	7
E. PERFORMANS DEĞERLENDİRME METRİKLERİ	8
1. <i>Doğruluk Oranı</i>	8
2. <i>Soru-Cevap Kalitesi</i>	8
3. <i>Kod Üretim Başarısı</i>	8
4. <i>Özetleme Yetkinliği</i>	9
5. <i>Değerlendirme Yöntemi ve Objektiflik</i>	9
II. YÖNTEM	9
A. ARAŞTIRMA MODELİ VE STRATEJİSİ	9
B. MODEL SEÇİMİ VE GEREKÇELERİ	10
C. STRATEJİLERİN UYGULAMA SÜRECİ	11
D. PROMPT FORMATININ MODEL ÇİKTISI ÜZERİNDEKİ ETKİSİ	12
E. PUANLAMA SİSTEMİ VE DEĞERLENDİRME ÖLÇEĞİ	12
III. BULGULAR	13
A. METRİK: SORU-CEVAP KALİTESİ	13
1. <i>Model: GPT-4o</i>	13
2. <i>Model: Grok 3</i>	17
3. <i>Model: Claude Sonnet 4</i>	21
4. <i>Model: DeepSeek V3</i>	24

5.	<i>Soru-Cevap Metriği Sonuçlarının Strateji ve Model Bazında Karşılaştırılması</i>	29
B.	METRİK: ÖZETLEME KALİTESİ	30
1.	<i>Model: GPT-4o</i>	30
2.	<i>Model: Grok 3</i>	34
3.	<i>Model: Claude Sonnet 4</i>	38
4.	<i>Model: DeepSeek V3</i>	42
5.	<i>Özetleme Metriği Sonuçlarının Strateji ve Model Bazında Karşılaştırılması</i>	47
C.	METRİK: DOĞRULUK KALİTESİ.....	48
1.	<i>Model: GPT-4o</i>	48
2.	<i>Model: Grok 3</i>	52
3.	<i>Model: Claude Sonnet 4</i>	55
4.	<i>Model: DeepSeek V3</i>	59
5.	<i>Doğruluk Metriği Sonuçlarının Strateji ve Model Bazında Karşılaştırılması</i>	63
D.	METRİK: KOD ÜRETİMİ KALİTESİ	64
1.	<i>Model: GPT-4o</i>	64
2.	<i>Model: Grok 3</i>	69
3.	<i>Model: Claude Sonnet 4</i>	73
4.	<i>Model: DeepSeek V3</i>	77
5.	<i>Kod Üretimi Metriği Sonuçlarının Strateji ve Model Bazında Karşılaştırılması</i>	81
IV.	TARTIŞMA	82
A.	STRATEJİ VE METRİK BAZINDA KARŞILAŞTIRMALI ÖZET TABLO	82
B.	STRATEJİ VE MODEL BAZINDA STANDART SAPMA ANALİZİ	83
C.	LİTERATÜRE KATKI	86
V.	KAYNAKÇA	HATA! YER İŞARETİ TANIMLANMAMİŞ.

ÖZET

Bu çalışma, büyük dil modellerinin (LLM) çıktılarının kalitesine etki eden prompt stratejilerinin karşılaştırmalı olarak analizini amaçlamaktadır. Araştırmanın temel motivasyonu, farklı model mimarilerine (GPT-4o, Claude Sonnet 4, Grok 3, DeepSeek V3) uygulanan çeşitli prompt stratejilerinin (zero-shot, one-shot, few-shot, chain-of-thought, direct ve indirect prompting) sonuç kalitesi üzerindeki etkilerini ortaya koymaktır. Bu kapsamda, her bir model ve strateji kombinasyonu, doğruluk, kod üretimi, özetleme ve soru-cevap gibi dört temel metrik açısından değerlendirilmiştir.

Çalışmanın yöntemi, her strateji ve model kombinasyonu için belirlenen standart görevler aracılığıyla, insan değerlendirmesine dayalı 0–100 puanlama sistemiyle ölçüm yapmayı içermektedir. Değerlendirme sürecinde, modellerin verdiği yanıtlar detaylı biçimde incelenmiş ve açıklık, detay, bağlama uygunluk gibi alt metrikler de dikkate alınmıştır. Tüm bulgular tablo ve grafiklerle görselleştirilmiş, her bir stratejinin modellerdeki görelî başarısı karşılaştırılmıştır.

Elde edilen bulgular, özellikle **chain-of-thought** stratejisini tüm ana metriklerde en yüksek ortalama puanları elde ettiğini, buna karşın few-shot ve direct prompting gibi stratejilerin ise genel olarak daha düşük ortalamalara sahip olduğunu göstermektedir. Bu sonuç, özellikle karmaşık ve çok aşamalı görevlerde, adım adım akıl yürütmeye dayalı stratejilerin çıktı kalitesini belirgin biçimde artırdığını ortaya koymaktadır. Ayrıca, model performansındaki genel farklılıkların analiz sırasında stratejinin gerçek etkisini gölgelememesi için, bulgular strateji bazında değerlendirilmiş ve yorumlanmıştır.

Bu yönüyle çalışma, farklı stratejilerin ve modellerin güçlü ve zayıf yanlarını nesnel bir bakış açısıyla ortaya koymakta, araştırmacı ve uygulayıcılar için pratik bir rehber işlevi görmektedir. Sonuç olarak, büyük dil modelleriyle çalışırken, görev tipine ve model özelliklerine göre en uygun prompt stratejisini seçmenin çıktı kalitesi üzerinde kritik bir etkisi olduğu sonucuna varılmıştır.

Anahtar Kelimeler: Büyük Dil Modelleri, Prompt Stratejileri, Chain-of-Thought, Doğruluk, Kod Üretimi, Özetleme, Soru-Cevap, Model Karşılaştırması, Yapay Zekâ. GPT-4, Grok, Claude, DeepSeek, Zero-shot Learning, Few-shot Learning, Prompt Engineering, Doğruluk Oranı, Kod Üretimi, Özetleme Performansı.

GİRİŞ

Yapay zekâ teknolojilerinin hızla geliştiği bu dönemde, özellikle büyük dil modelleri (Large Language Models – LLM) doğal dil işleme alanındaki yetkinlikleriyle yapay zekânın çok yönlü kullanım potansiyelini açıkça göstermiştir. Bu modeller; metin oluşturma, soru-cevap, özetleme ve kod üretimi gibi görevlerde, insan benzeri çıktılar verebilmektedir. Ancak, bu modellerin başarımı yalnızca iç yapılarına değil, aynı zamanda onlara nasıl direktif verildiğine, yani kullanılan *prompt stratejileriyle* de doğru orantılıdır. Dolayısıyla bu stratejiler, modelin doğru ve anlamlı çıktılar üretmesini etkileyen en kritik faktörlerden biri hâline gelmiştir.

Tezin amacı, farklı prompt stratejilerinin GPT-4o, Claude, Grok ve DeepSeek modelleri üzerindeki çıktılarının performansını dört temel metrik üzerinden karşılaştırmaktır: doğruluk, soru-cevap kalitesi, kod üretimi ve özetleme. Bu doğrultuda, her bir model üzerinde farklı prompt stratejileri uygulanarak elde edilen çıktıların kalitatif ve kantitatif değerlendirmeleri yapılmıştır. Sonuçlar tablolar ve grafikler aracılığıyla karşılaştırılmıştır. Çalışmanın bütününde yalnızca çıktı kalitesine bağlı kalmayıp aynı zamanda uygulama kolaylığı, kaynak verimliliği ve son kullanıcıya yönelik kullanılabilirlik de dikkate alınarak değerlendirme yapılmıştır.

Her bir model kendi resmi ortamında, güncel sürümleriyle sınanmış ve değerlendirme metrikleri sabit tutulmuştur. Ancak prompt stratejilerinin etkisi, modele entegre edilen güncellemelerden ve dışsal faktörlerden etkilenebileceğinden, çıktılar belirli sınırlar dikkate alınarak yorumlanmalıdır. Ayrıca, bu tez kapsamı itibarıyla görsel, sesli ya da çok dilli veri üretimi gibi LLM'lerin multimedial özelliklerini incelememiştir, yalnızca metinsel görevler üzerinde yoğunlaşmıştır.

Yönetim Bilişim Sistemleri tarafından bakıldığından, bu çalışma; karar destek sistemleri, yazılım geliştirme ve otomasyon gibi alanlarda yapay zekâ destekli çalışmaların kalitesini artırmayı amaçlayarak hem akademik katkı sunmayı hem de ilgilisine yol gösterici bir kaynak olmayı hedeflemektedir.

I. GENEL BAKIŞ

A. Büyük Dil Modellerinin Evrimi

Doğal Dil İşleme (NLP) alanı tarihsel olarak üç ana aşamadan geçerek ilerlemiştir: kural tabanlı yaklaşımalar, istatistiksel yöntemlerin kullanımı ve derin öğrenme tabanlı modellerin hakimiyeti. Bu dönüşümün temel taşıları arasında, makinelerin kelimeleri anlamsal düzeyde temsil etmesine izin veren word2vec ve GloVe başta olmak üzere 2013 yılında geliştirilen word2vec ve GloVe olmak üzere kelime gömme teknikleri yer almıştır.

Günümüze kadar yapılan tüm gelişmeler bu temelin üzerine inşa edilmiştir. Bu adımlardan en kritik olanı: Transformer Mimarisi. 2017 yılında Vaswani ve arkadaşları tarafından ortaya konan “Attention is All You Need” başlıklı bu çalışma, dil modelleme süreçlerinde zaman bağımlılığına dayanan RNN ve LSTM gibi mimarilerin yetersizliklerini ortadan kaldırarak dikkat (attention) mekanizmasını merkez almıştır. Bu yapı hem paralel işlemeyi mümkün kılması hem de uzun bağamları etkili şekilde işleyebilmesinden dolayı LLM’lerin önünü açmıştır (Vaswani et al., 2017).

Transformer mimarisi, daha sonra geliştirilen çok sayıda modelin temelini oluşturmuştur. Örneğin, OpenAI tarafından geliştirilen **GPT (Generative Pre-trained Transformer)** serisi, yalnızca prompt kullanarak yeniden eğitime gerek kalmadan birçok görevi gerçekleştirebilme yeteneğiyle dikkat çekmiştir. Bu seri özellikle **GPT-2** ve peşi sıra gelen **GPT-3** ile geniş çaplı bir etki yaratmış, 175 milyar parametre yapısı ile doğal dilin üretilmesi ve yorumlanmasında devrim yaratmıştır.

Aynı dönemde Google, çift yönlü bağlam modelleme yapabilen **BERT** (Bidirectional Encoder Representations from Transformers) modelini tanıtarak metin sınıflandırma ve anlamsal çıkarım gibi görevlerin başarısını artırdı. Bu modellerin ardından T5, PaLM, LLaMA, Claude, DeepSeek, Grok gibi farklı kuruluşlar tarafından geliştirilen ileri düzey modeller, çok görevli öğrenme, kod oluşturma, akıl yürütme ve özetleme gibi karmaşık NLP problemlerinde yüksek performans göstermiştir.

Model boyutlarının artması sadece parametre sayısının artmasını değil, aynı zamanda eğitilen veri setlerinin boyutunu, eğitimin yapıldığı donanım altyapılarını ve etik/kamusal

tartışmaları da beraberinde getirmiştir. Bu noktada Transformer mimarisi, yüksek hesaplama maliyetlerine rağmen sunduğu ölçeklenebilirlik ve genel amaçlı öğrenme yeteneği sayesinde sadece metinle değil, görüntü, ses ve multimodal verilerle de çalışan sistemlerin temelini oluşturmuştur.

BERT gibi modeller yalnızca kodlayıcı katmanını kullanırken, GPT gibi modeller yalnızca kod çözücü mimarisi üzerine kuruludur. Bu yapısal farklılıklar, modellerin görev türlerine göre özelleştirilmesine olanak tanır. Günümüzde LLM'ler, eğitimden müşteri hizmetlerine, yazılım mühendisliğinden yarı otomatik karar destek sistemlerine kadar çok geniş bir alanda katma değer üretmektedir.

Özetle, büyük dil modellerinin geliştirilmesi yalnızca yapısal ilerlemelerle değil, aynı zamanda kullanıcı istemlerine verdikleri yanıtları işleme yeteneklerindeki gelişmelerle de tanımlanmalıdır. Bu nedenle, hızlı mühendislik, modern LLM performanslarının belirleyicilerinden biri haline gelmiştir.

B. GPT-4, Grok, Claude, DeepSeek Modellerinin İncelenmesi

Bu bölümde, tezin odağını oluşturan dört ana büyük dil modeli (LLM), GPT-4, Grok, Claude ve DeepSeek modelleri detaylı olarak incelenecaktır. Bu derlemede modellerin mimari yapılarına, güçlü ve zayıf yönlerine ve hangi uygulama alanlarında daha etkili olduklarına yer verilmiştir.

1. GPT-4

OpenAI tarafından geliştirilen GPT-4 (Generative Pre-trained Transformer 4), Transformer mimarisi üzerine inşa edilmiştir. GPT-3.5'e kıyasla çok daha fazla parametreden oluşan GPT-4, genel kullanımda daha iyi bağlam anlayışı, gelişmiş mantıksal çıkarım yeteneği ve üstün performans sergiler. Özellikle doğal dilde metin oluşturma, özetleme ve kodlama alanlarında geniş bir kullanım alanına sahiptir (OpenAI, 2023).

2. Grok

Grok, xAI tarafından geliştirilen yeni nesil bir dil modelidir. Özellikle, gerçek zamanlı veri akışlarını daha iyi işleme yeteneği ile öne çıkıyor. Ayrıca yapay zekâ güvenliği ve tutarlılığı açısından güçlü geliştirmelere sahiptir. Grok, farklı alanlardaki gerçek zamanlı güncel bilgileri etkili bir şekilde yorumlayarak daha güncel ve tutarlı sonuçlar sunar (xAI, 2024).

3. Claude

Anthropic tarafından geliştirilen Claude, büyük dil modelleri arasında "yardımcı" ve "zararsız" olmasıyla bilinir. İnsan geri bildirimleri ile güçlenen güvenlik odaklı bir yapıya sahiptir. Claude'un mimarisi, özellikle etik açıdan güvenli ve kullanıcı dostu yanıtlar üretmeye odaklanmıştır. Soru-cevap ve etik karar verme süreçlerinde iyi performans gösterir (Anthropic, 2023).

4. DeepSeek

DeepSeek, özellikle uzun içerikli metinlerde ve teknik belgelerde son derece doğru yanıtlar oluşturmak için optimize edilmiş bir dil modelidir. Derin bağlam analizi yeteneği ve çok yönlü çıkarm kapasitesi ile tanınır. Teknik içerik oluşturmada, kod oluşturmada ve karmaşık bağlamlara sahip soruları yanıtlamada güçlündür. Bu modelin en önemli avantajı, spesifik ve teknik alanlarda sağladığı yüksek performansıdır (DeepSeek, 2024).

Bu dört modelin genel değerlendirmesinde, GPT-4 geniş uygulama ve genel doğruluk açısından lider olurken, Grok gerçek zamanlı ve güncel bilgilerin kullanımında öne çıkarıyor; Claude, etik ve kullanıcı dostu uygulamalardaki güvenilirliği ile öne çıkarıyor ve DeepSeek, teknik ve derin içerikli sorgularda öne çıkarıyor. Bu farklılıklar, tez kapsamında modellerin farklı istem stratejilerine verdikleri yanıtların ve performans farklılıklarının analiz edilmesi açısından önemli olacaktır.

C. Stratejik Yaklaşımlar

Bu bölümde, büyük dil modellerinin performansını önemli ölçüde etkileyen temel prompt stratejileri açıklanacak ve bu stratejilerin karşılaştırmalı analizinin önemi tez kapsamında vurgulanacaktır. Tartışılan stratejiler; Zero-shot learning, One-shot learning, Few-shot learning, Chain-of-Thought ve Direct vs Indirect prompting.

1. Zero-shot Learning (Sıfır Örnekli Öğrenme)

Zero-shot Learning yaklaşımı, modele herhangi bir örnek verilmeden, görevlerin yalnızca açık ve net talimatlar yoluyla tanıtılmamasını ifade eder. Bu yöntemde model, mevcut bilgi birikimi ve genel yetenekleri ile tamamen spontane bir yanıt üretir. Zero-shot, modellerin genel kavrayışını ve daha önce hiç görmedikleri sorulara veya görevlere karşı doğrudan performanslarını ölçmek için özellikle tercih edilir. Ön eğitim verilerinden elde edilen genel

bilgileri kullanarak modellerin belirli durumlara nasıl uyum sağladığını gözlemelemek için çok uygundur.

2. One-shot Learning (Tek Örnekli Öğrenme)

One-shot stratejisinde modele, girilen promptu anlamak ve gerçekleştirmek için yalnızca tek bir örnek verilir. Amaç, modelden istenen görevin doğası ve modelin vermesi beklenen yanıt hakkında modelle hızlı bir şekilde etkileşim kurmaktır. One-shot yönteminin çıktıtı doğrudan verilen örneğin kalitesine bağlıdır. İyi tasarlanmış tek bir örnek, modelin görevi anlama kapasitesini önemli ölçüde artırabilir.

3. Few-shot Learning (Birkaç Örnekli Öğrenme)

Few-shot yaklaşımında, modele birkaç örnek sunulur. Bu örnekler, modelin görevi daha iyi anlamasını ve daha dengeli yanıtlar üretmesini destekler. Modele birden fazla örnek sunmak modelin bağlama yeteneğini geliştirir ve cevapların kalitesini artırr. Few-shot, özellikle modellerin sınırlı örneklemelerle bile etkili bir şekilde genelleme yapma kapasitesini gösterir ve birçok uygulama senaryosunda zero-shot ve one-shot yöntemlerinden daha yüksek performans sergiler.

4. Chain-of-Thought (Düşünce Zinciri)

Chain-of-Thought, modelin karmaşık bir görevi çözerken mantıksal adımları net bir şekilde göstermesini sağlayan hızlı bir stratejidir. Bu yöntemde model, cevaba ulaşırken ara adımları ve düşünce akışını sıralı bir şekilde ortaya koymaktadır. Böylece karmaşık ve çok adımlı problemlerin çözüm süreci şeffaflaşır ve modelin mantıksal çıkarım yeteneği geliştirilir. Özellikle akıl yürütme, hesaplama ve çok aşamalı sorularda model performansını önemli ölçüde artırır.

5. Direct vs Indirect Prompting (Doğrudan ve Dolaylı Prompt)

Direct prompting yöntemi, modele istenilen görevin açık ve kesin bir şekilde aktarılmasıdır. Bu yaklaşım, modelin görevi direkt olarak anlamasını sağlar ve genellikle somut, açık uçlu ve belirli cevap beklenen durumlarda daha etkilidir. Indirect prompting ise, modelin daha belirsiz ifadelerle çıkarım yaparak cevabı üretmesini bekler. Dolaylı prompting yönteminin avantajı, modellerin bağlama dayalı anlama ve yorumlama becerilerini geliştirecek bazen beklenmedik yaratıcı sonuçlar üretmelerine imkân sağlamasıdır.

Bu çalışmada incelenen stratejiler, büyük dil modellerinin farklı görevlerdeki güçlü ve zayıf yönlerini netleştirmek ve hangi yöntemin hangi bağlamda daha etkili sonuç verdiği belirlemek için kullanılacaktır. Örneğin, Zero-shot yöntem modelin genel bilgi kapasitesini ölçerken, Few-shot ve Chain-of-Thought gibi yöntemler modelin bağlam içinde daha detaylı analiz yapabilme yeteneğini gösterir. Direct ve Indirect prompting yöntemleri ise modellerin kullanıcı taleplerini yorumlama ve cevaplama biçimlerini anlamaya yardımcı olur. Böylece tez kapsamında, bu stratejilerin model performansları üzerindeki etkileri net bir şekilde karşılaştırılarak araştırmacılara ve uygulayıcılara rehber niteliğinde sonuçlar sunulacaktır.

D. Model Eğitimi ve Veri Kümeleri

Bu çalışmada, modellerin değerlendirilmesi için veri setleri kullanılmadan, doğrudan kullanıcı deneyimine daha yakın olan prompt bazlı bir değerlendirme yöntemi tercih edilmiştir. Bu tercih, özellikle kullanıcı dostu deneyime odaklanarak modellerin gerçek yaşam senaryolarına yakın koşullarda test edilmesini sağlayacaktır.

Prompt bazlı değerlendirme yöntemi, modellerin genel performansını, bağlamı anlama becerisini ve kullanıcıya sağladığı çıktı kalitesini hızlı ve doğrudan ölçme imkânı sunar. Bu yöntemle, hazırlanan promptlar çeşitli görev tiplerine göre (doğruluk testi, soru-cevap, özetleme ve kod üretimi gibi) gruplandırılarak, modellerin her farklı görevdeki yetkinlikleri sistematik olarak incelenecaktır.

Değerlendirme aşamasında kullanılacak promptlar, kullanıcıların modellerle etkileşim kurarken günlük hayatımda karşılaşabilecekleri tipik görevlerden ve sorularından oluşacaktır. Bu görevlerin seçiminde aşağıdaki hususlara dikkat edilecektir:

- Açıklık ve Tutarlılık:** Promptların, modeller tarafından net bir şekilde anlaşılması için mümkün olduğunda açık, anlaşılır ve tutarlı ifadelerle yazılması sağlanacaktır.
- Görev Çeşitliliği:** Farklı stratejilerin etkilerini daha iyi anlamak için, farklı karmaşıklık seviyelerinde (basit ve karmaşık sorular, kısa ve uzun özetleme görevleri, kolay ve zor kod oluşturma senaryoları gibi) promptlar tasarılanacaktır.
- Objektif Değerlendirme Kriterleri:** Modellerin çıktı kalitesini ve tutarlığını objektif bir şekilde ölçmek için önceden belirli performans kriterleri belirlenecek ve bu kriterler çerçevesinde değerlendirilmeler yapılacaktır.

Bu yöntem sayesinde modellerin kullanıcı açısından ne kadar etkili, tutarlı ve kullanışlı olduğu somut bir şekilde ortaya konacaktır. Elde edilen sonuçların pratik hayatı doğrudan

uygulanabilir olması, tez çalışmasının temel amaçlarından biri olan yol gösterici niteliği güçlendirecektir.

E. Performans Değerlendirme Metrikleri

Büyük dil modellerinin çeşitli prompt stratejileriyle nasıl bir performans sergilediğini tarafsız ve anlamlı şekilde değerlendirmek için bazı temel metriklerin belirlenmesi gereklidir. Bu çalışmada, literatürde yaygın olarak kabul görmüş dört ana değerlendirme ölçüyü kullanılcaktır: doğruluk oranı, soru-cevap kalitesi, kod üretim başarısı ve özetleme yetkinliği. Bu metrikler hem nicel hem de nitel analiz yapabilmek ve modellerin güçlü ile gelişime açık yönlerini somutlaştırmak için seçilmiştir.

1. Doğruluk Oranı

Doğruluk oranı, modelin verilen bir soruya veya bilgiye gerçeğe uygun, doğru yanıtlar üretme kapasitesini gösterir. Özellikle genel bilgi, tarihi olaylar veya olgusal sorular gibi tek bir doğru cevabın olduğu durumlarda kullanılır. Buna karşılık olarak modelin doğruluğu, güvenilirliği ve tutarlılığı bu metrikle ölçülür. Çıkan sonucu değerlendirme "doğru", "kısmen doğru" veya "yanlış" olarak sınıflandırılabilir. Ek olarak, yanıtların zamanlaması ve modelin yanlış bilgi üretme eğilimi de dikkate alınacaktır.

2. Soru-Cevap Kalitesi

Soru-cevap kalitesi, modelin açık uçlu veya bilgi temelli sorulara verdiği yanıtların anlaşılır, bütünlüklü ve tatmin edici olup olmadığını değerlendiren nitel bir ölçütür. Burada sadece bilginin doğruluğu değil, aynı zamanda yanıtın açıklık düzeyi, bağlama uygunluk ve kullanıcıya sağladığı değer de dikkate alınacaktır. Yanıtların açık, tutarlı, hedefe yönelik ve gereksiz bilgilerden arındırılmış olması, yüksek kalite göstergeleri olarak kabul edilecektir.

3. Kod Üretim Başarısı

Kod üretimi değerlendirmesi, modelin verilen bir yazılım problemi veya kodlama görevi için doğru ve çalışabilir kod üretebilme kapasitesine odaklanır. Kodun hem işlevselliği hem de temiz ve anlaşılır olması önemlidir. Ayrıca, üretilen kodun hatasız çalışıp çalışmadığı, verilen girdilere uygun sonuçlar üretip üretmediği ve gerekli açıklamaların (yorum satırları vb.) olup olmadığı da bu metriğin kapsamında inceleneciktir.

4. Öztleme Yetkinliği

Öztleme yetkinliği, modelin uzun veya karmaşık metni öz, anlaşılır ve anlamı koruyan bir şekilde öztleme yeteneğini ölçer. Burada önemli olan modelin metnin ana mesajını, ana fikrini ya da en kritik noktalarını tam olarak yakalayabilmesi ve yansıtılabilmesidir. Ayrıca özetin özgün, dilbilgisi açısından doğru, tekrar eden veya anlamsız ifadelerden arındırılmış olması beklenir.

5. Değerlendirme Yöntemi ve Objektiflik

Her bir metrik için değerlendirme sürecinde, yanıtların birden fazla kişi tarafından bağımsız olarak incelenmesi ve olabildiğince objektif bir puanlama sistemi kullanılması hedeflenmektedir. Gerekli durumlarda, hem nicel puanlama (ör. 1-5 arası puan) hem de nitel gözlemler kullanılacaktır. Böylece, modellerin performansı daha bütüncül bir şekilde analiz edilebilecek ve strateji-model uyumları daha açık bir biçimde ortaya konacaktır.

II. YÖNTEM

Bu bölümde, farklı prompt stratejilerinin GPT-4o, Grok 3, Claude Sonnet 4 ve DeepSeek V3 modelleri üzerindeki sonuç kalitesine etkisinin karşılaştırılması için izlenecek yöntem detaylandırılacaktır. Kullanılan deneysel yaklaşım, model ve strateji seçimi ile değerlendirme sürecinin nasıl planlandığı sistemli bir şekilde açıklanacaktır.

A. Araştırma Modeli ve Stratejisi

Bu araştırmanın temel amacı, farklı prompt stratejilerinin büyük dil modelleri (GPT-4o, Grok 3, Claude Sonnet 4 ve DeepSeek V3) üzerindeki sonuç kalitesine etkisini sistematik ve karşılaştırmalı bir yaklaşımla analiz etmektir. Araştırma modeli, deneysel ve betimleyici yöntemlerin bir araya getirildiği, uygulamalı bir stratejiye dayanmaktadır.

Çalışmada izlenecek yöntem, her model üzerinde aynı görev tiplerinin ve aynı temel içerikli soruların, yalnızca prompt stratejileri değiştirilerek uygulanmasına dayanır. Böylece, prompt stratejilerinin modellerin yanıt kalitesine ve görev başarısına olan etkisi doğrudan gözlemlenebilir hale gelir. Her bir model ve strateji kombinasyonu, daha önce belirlenen

metrikler (doğruluk, soru-cevap kalitesi, kod üretim başarısı, özetleme yetkinliği) kapsamında ayrı ayrı değerlendirilecektir.

Öncelikle, her model için belirlenen dört görev türü (doğruluk, soru-cevap, kod üretimi, özetleme) esas alınır. Aynı görev, her bir prompt stratejisi (zero-shot, one-shot, few-shot, chain-of-thought, direct, indirect) kullanılarak ilgili modele yöneltilir. Modellerden alınan yanıtlar, önceden belirlenmiş performans değerlendirme kriterlerine göre puanlanır ve nitel olarak analiz edilir. Elde edilen sonuçlar hem tablo ve grafiklerle görselleştirilir hem de metinsel olarak karşılaştırılır.

Bu yapı, model-strateji kombinasyonlarının etkilerini yalın bir şekilde ortaya koymayı ve her modelin hangi stratejide daha yüksek performans sergilediğini nesnel biçimde değerlendirmeyi amaçlamaktadır. Aynı zamanda, elde edilen bulguların doğrudan uygulamaya aktarılabilmesi ve son kullanıcı için yol gösterici nitelikte olması hedeflenmektedir.

B. Model Seçimi ve Gerekçeleri

Bu araştırmada incelenen büyük dil modelleri; **GPT-4o**, **Grok 3**, **Claude Sonnet 4** ve **DeepSeek V3** olarak belirlenmiştir. Model seçiminde temel alınan kriterler; güncellilik, erişilebilirlik, mimari çeşitlilik, farklı kullanım alanlarında öne çıkan özellikler ve literatürdeki karşılaştırmalı çalışmalara katkı potansiyelidir.

GPT-4, doğal dil işleme alanında en yaygın kullanılan ve çok yönlü bir model olarak bilinmektedir. Geniş ölçekli veriyle eğitilmiş olan bu model, özellikle metin üretimi, özetleme ve kod üretimi gibi görevlerde yüksek performans göstermektedir. GPT-4'ün geniş bilgi altyapısı ve genel amaçlara uygun yapısı, çeşitli prompt stratejilerinin çıktı kalitesi üzerindeki etkisini analiz etmek için uygun bir zemin sağlar.

Grok, xAI tarafından geliştirilmiş, özellikle güncel bilgilere erişimi hızlı olan bir modeldir. Gerçek zamanlı veri akışı ile uyumludur. Dinamik bilgi güncellemleri ve pratik uygulamalar için tercih edilebilir. Bu özellik, özellikle güncel olaylar veya değişken veriler gerektiren senaryolarda strateji-performans ilişkisinin incelenmesi için önemli bir avantaj sağlar.

Claude, Anthropic tarafından geliştirilen ve güvenlik, etik uyum ve kullanıcı dostu yanıtlar üretme amacıyla tasarlanmış bir modeldir. Claude'un öne çıkan yönü, insan merkezli ve zararsız içerik üretmeye odaklanmasıdır. Böylece, farklı prompt stratejilerinin özellikle kullanıcı odaklı görevlerdeki etkisini değerlendirmek mümkün olmaktadır.

DeepSeek, teknik belgelendirme ve uzun içeriklerde yüksek doğrulukla yanıt üretebilmesiyle öne çıkan bir modeldir. Derin bağlam analizi ve teknik karmaşıklık gerektiren görevlerde performansı ile tanınan DeepSeek, özellikle kod üretimi ve detaylı metin analizlerinde strateji farklılıklarının etkisini ölçmek için tercih edilmiştir.

Model seçiminde ayrıca şu unsurlar dikkate alınmıştır:

- API veya online arayüz ile erişim kolaylığı
- Araştırmacılar tarafından test edilebilirlik
- Sektörde ve akademide güncel ve popüler olması
- Farklı mimari yaklaşım sayesinde karşılaştırmalı analiz potansiyeli

Sonuç olarak, bu modellerin her biri farklı güçlü yönler ve karakteristikler taşıdığını, prompt stratejilerinin model bazında nasıl farklı sonuçlar doğurduğunu anlamak için ideal bir örneklem oluşturmaktadır. Böylece araştırmmanın bulguları hem literatürdeki boşlukları doldurma hem de uygulayıcılar için pratik öneriler sunma açısından güçlü bir temel üzerine inşa edilmektedir.

C. Stratejilerin Uygulama Süreci

Bu bölümde, farklı prompt stratejilerinin büyük dil modellerine nasıl uygulanacağına dair yöntem ve süreç adım adım açıklanacaktır. Amaç, strateji bazlı testlerin sistematik ve karşılaştırılabilir şekilde yürütülmesini sağlamaktır.

Her bir değerlendirme metriği için, önceden belirlenen aynı temel sorular, yalnızca prompt stratejisinin yapısına uygun biçimde farklılaştırılacaktır. Yani, kod üretiminde “Python dilinde bir listenin ortalamasını hesaplayan bir fonksiyon yaz” görevi; zero-shot, one-shot, few-shot, chain-of-thought, direct ve indirect prompting stratejileriyle ayrı ayrı modellenmiş olacak ve tüm modellerde aynı çekirdek görevin farklı prompt biçimleriyle sınanması sağlanacaktır

Her bir strateji, seçilen tüm modellerde birebir aynı içerikte test edilecektir. Böylece, stratejiler arası farklar modellenen iç mimarisini veya eğitimiyle mi yoksa promptun sunuluş biçimini mi ilişkili, daha tutarlı biçimde gözlemlenecektir. Her model ve strateji kombinasyonu için çıkan sonuçlar, tanımlanan metrikler ışığında ayrı ayrı kaydedilecek ve daha sonra karşılaştırmalı analiz için derlenecektir.

Her modelden alınan cevaplar sistematik olarak kaydedilecek, yanıt kalitesi ve kullanıcı dostluğu açısından önceden belirlenen ölçütlerle analiz edilmeye hazır hale getirilecek. Yanıtlar,

ilgili metrikler doğrultusunda hem nicel (puanlama ölçüye) hem de nitel (açıklayıcı gözlemlerle) değerlendirilecek. Elde edilen sonuçlar, tablo ve grafikler yardımıyla görselleştirilecek; böylece hangi stratejinin, hangi modelde ve hangi görevde daha verimli olduğu nesnel biçimde ortaya konacak.

Bu süreçte, tüm modellerde aynı içerik ve görevlerin kullanılması, karşılaştırılabilirlik ve araştırmanın bilimsel güvenilirliği açısından özellikle önemsenmiştir. Ayrıca, uygulamanın her aşaması titizlikle belgelenerek, çalışma sonunda araştırmacıların tekrar edilebilir ve şeffaf bir yöntemle süreci izlemesi mümkün olacaktır.

D. Prompt Formatının Model Çıktısı Üzerindeki Etkisi

Bu çalışmada, farklı prompt stratejileri ile büyük dil modellerinin (LLM) performansı karşılaştırılırken, kullanılan promptların biçimsel özelliklerinin model çıktıısı üzerinde belirleyici bir etkiye sahip olabileceği göz önünde bulundurulmuştur. Özellikle one-shot ve few-shot stratejilerinde, promptta verilen örneğin uzunluğu ve detay seviyesi, modelin yanıtının da benzer düzeyde ayrıntılı ya da yüzeysel olmasına neden olabilmektedir.

Bu nedenle, model çıktılarına verilen 0-100 arası puanları, yalnızca yanıtın içeriğine göre objektif olarak değerlendirilmiş; ancak, gözlem ve analiz kısmında, ayrıntı puanlarının prompt formatından kaynaklanabileceği özellikle vurgulanmıştır. Böylece, puanlamada modelin yeteneği ile promptun yönlendirici etkisi ayrıstırılmış; bulguların daha doğru ve güvenilir biçimde yorumlanması sağlanmıştır.

E. Puanlama Sistemi ve Değerlendirme Ölçeği

Çalışmada, büyük dil modellerinin verdiği yanıtların karşılaştırılabilir ve nesnel biçimde değerlendirilmesi amacıyla **0-100 arası sürekli puanlanabilen bir değerlendirme sistemi** oluşturulmuştur. Bu sistem, modellerin performansını belirli kriterler üzerinden çok boyutlu ve hassas bir şekilde incelemeye olanak tanımaktadır. Her kombinasyon için “doğruluk”, “açıklık”, “detay seviyesi” ve “bağlama uygunluk” başlıklarında, her bir model çıktıısı 0’dan 100’e kadar bir puanla değerlendirilmiştir. Aşağıda, her bir puanlama kriterinin farklı aralıklardaki anlamlarını gösteren değerlendirme tablosu sunulmuştur:

PUAN	DOĞRULUK	AÇIKLIK	DETAY	BAĞLAMA UYGUNLUK
0-19	Yanlış veya çok yanıltıcı	Anlaşılması çok zor ya da imkânsız	Detay veya açıklama yok	Alakasız, bağlam dışı
20-39	Yanıtın önemli kısmı eksik veya hatalı	Zor anlaşılır, bazı belirsizlikler	Çok kısa, detaysız cevap	Bağlamı tam karşılamıyor
40-59	Temel olarak doğru, önemli bir eksik	Anlaşılmış ama ifadede zayıflıklar var	Yüzeysel açıklama, az örnek	Sadece bilimsel/pratik tek boyutlu
60-79	Genel olarak doğru, ufak belirsizlik	Genellikle açık, az sayıda ifade hatası	Temel tanım ve birkaç detay	Kısmen kapsamlı ve bağlama uygun
80-100	Tam doğru, eksiksiz ve güncel bilgi	Çok açık, herkesçe kolayca anlaşılabilir	Hem temel kavram hem bilimsel detay	Hem bilimsel hem günlük yaşama uygun

Tablo 1: Açıklamalı Değerlendirme (Puan) Tablosu

III. BULGULAR

Bu alt bölümde, her bir model için dört temel görev türü (doğruluk, soru-cevap, kod üretimi, özetleme) doğrultusunda farklı prompt stratejileri ile elde edilen çıktılar değerlendirilmiştir. Her değerlendirmede kullanılan prompt ifadesi, model çıktısı, 0-100 ölçeğinde değerlendirilmiş sonucu ve kısa bir yorum sunulmuştur. Bu sayede, her modelin farklı stratejiler karşısındaki performansı bütüncül biçimde analiz edilmiştir.

A. Metrik: Soru-Cevap Kalitesi

Görev: Genel bilgi sorularını girilen prompt stratejisine göre açıkça cevaplamak.

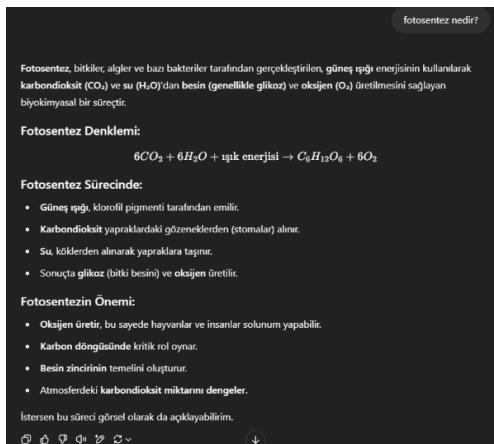
1. Model: GPT-4o

Aşağıdaki tabloda, soru-cevap metriğinde tüm prompt stratejilerinin “GPT-4o” modelindeki performansı 0-100 arası puanlama sistemine göre değerlendirilmiştir.

Strateji	Doğruluk	Açıklık	Detay	Bağlama Uygunluk
zero-shot	100	98	92	100
one-shot	95	92	60	85
few-shot	93	95	40	83
chain-of-Thought	100	99	96	100
direct prompting	100	97	98	100
indirect prompting	100	100	93	100

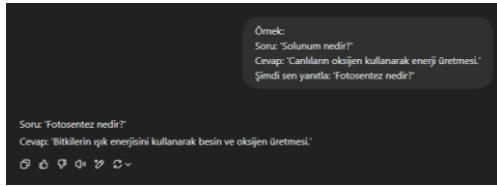
Tablo 2: Puanlama (soru-cevap metriğinde “GPT-4o” modelinin prompt stratejilerine göre performansı)

a. Gözlemler (GPT-4o)



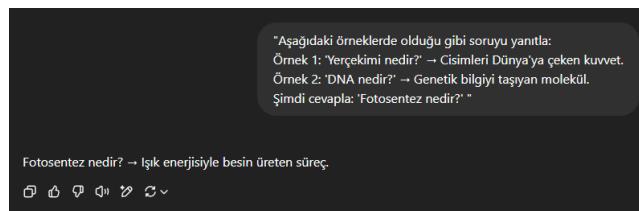
Görsel 1: "soru-cevap" metriğinde "zero-shot" stratejisi kullanılarak girilen promptun "gpt-4o" modelindeki çıktısı

GPT-4o, **zero-shot** stratejisine uygun olarak girilen promptu kapsamlı ve anlaşılır bir şekilde yanıtlamıştır. Çıktıda fotosentez tanımı, kimyasal denklem, süreç adımları ve fotosentezin önemi gibi başlıklar ayrıntılı ve maddeleme yöntemiyle sunulmuştur. Bilimsel doğruluk açısından herhangi bir hataya rastlanmamıştır. Açıklık ve detay düzeyi, genel okuyucu kitlesinin rahatlıkla anlayabileceği ve süreci kavrayabileceği bir seviyedendir. Bağlama uygunluk bakımından ise, yanıt doğrudan sorunun özüne odaklanmakta, gereksiz bilgi içermemektedir. Kullanılan dil akıcı ve tutarlı olup, yanıtın sonunda kullanıcının isteğine göre ek açıklama sunulabileceği belirtilmiştir. Kullanılan prompt ve model yanıtının ekran görüntüsü yukarıda sunulmuştur. Kullanılan prompt ve model yanıtının ekran görüntüsü yukarıda sunulmuştur.



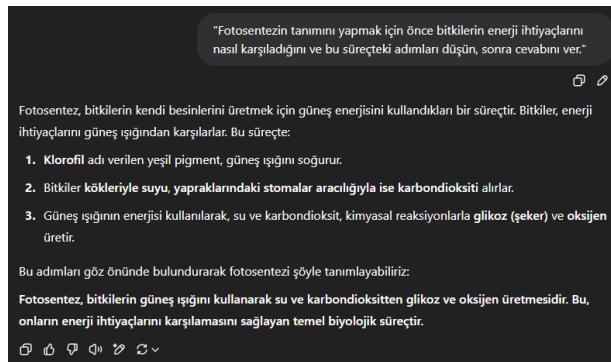
Görsel 2: "soru-cevap" metriğinde "one-shot" stratejisi kullanılarak girilen promptun "gpt-4o" modelindeki çıktısı

GPT-4o, **one-shot** stratejisinde, verilen kısa örnek formatını izleyerek soruya kısa, net ve anlaşılır bir tanımla yanıt vermiştir. Yanıtın detay seviyesi düşük olsa da bu durum doğrudan modelin yetersizliğinden değil, promptta yer alan örneğin kısalığından kaynaklanmaktadır. Cevap bilimsel olarak doğru, açıklık düzeyi ise oldukça yüksek olup, sorunun temelini doğrudan karşılamaktadır. Detay puanlamasında bu bağlam göz önünde bulundurulmuş, aşırı puan kaybından kaçınılmıştır. Kullanılan prompt ve model yanıtının ekran görüntüsü yukarıda sunulmuştur.



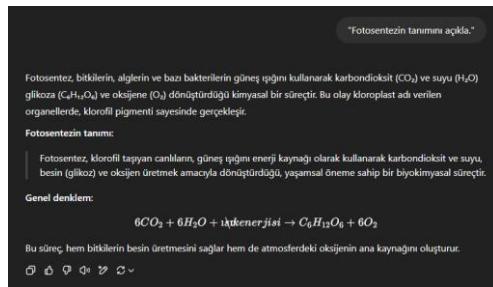
Görsel 3: "soru-cevap" metriğinde "few-shot" stratejisi kullanılarak girilen promptun "gpt-4o" modelindeki çıktısı

GPT-4o, **few-shot** stratejisiyle verilen prompta örneklerdeki kısa ve özet tanım stilini takip ederek, "Fotosentez nedir?" sorusunu tek cümleyle yanıtlamıştır. Yanıtta fotosentezin temel işlevi ("ışık enerjisiyle besin üreten süreç") doğru olarak belirtilmiştir. Bu kısa format, modelin örneklerden etkilenerken kapsamlı bilgi vermesini sınırlamıştır. Açıklık açısından yanıt oldukça sade ve anlaşılırındır. Detay düzeyi ise, promptun örneklerinden kaynaklanan bir gereklilik nedeniyle düşük tutulmuştur. Bağlama uygunluk bakımından cevap sorunun ana fikrine doğrudan karşılık gelmekte, ancak konunun tamamını yansıtmadır. Bu nedenle puanlamada, yanıtın sadeliği modelin kısıtı değil, promptun yönlendirmesinin sonucu olarak değerlendirilmiştir. Kullanılan prompt ve model yanıtının ekran görüntüsü yukarıda sunulmuştur.



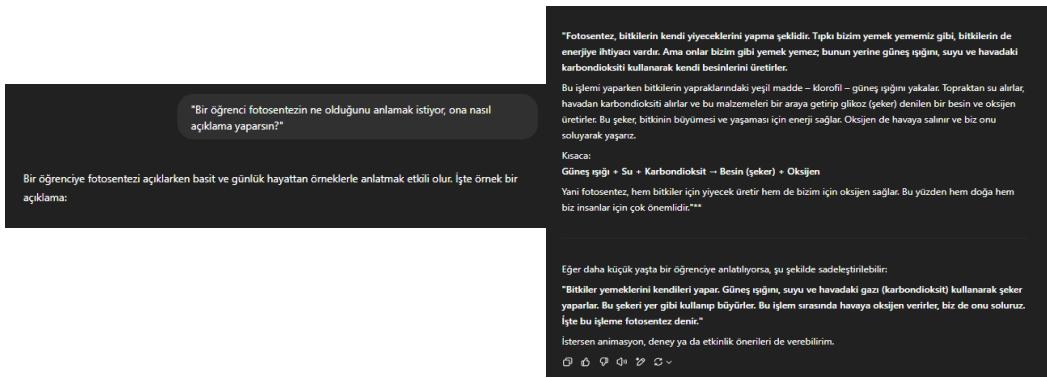
Görsel 4: "soru-cevap" metriğinde "chain-of-thought" stratejisi kullanılarak girilen promptun "gpt-4o" modelindeki çıktısı

GPT-4o, **chain-of-thought** stratejisinde, fotosentez sürecini adım adım açıklayarak hem bilimsel doğruluğu hem de anlaşılabilirliği üst düzeyde sağlamıştır. Model, bütün temel aşamaları detaylandırmıştır. Son bölümde ise tüm bilgileri bir araya getirerek kavramsal tanımı açık ve bütünlüyici bir şekilde sunmuştur. Yanıt hem bağlama tamamen uygundur hem de içerik açısından gereksiz ayrıntı veya eksik bilgi barındırmaz. Stratejinin doğal bir sonucu olarak detay düzeyi ve açıklık maksimum seviyeye çıkmıştır. Bu örnek, promptta adım adım düşünme yönlendirmesi verildiğinde, modelin yanıtının da çok daha derin ve kapsamlı olduğunu göstermektedir. Kullanılan prompt ve model yanıtının ekran görüntüsü yukarıda sunulmuştur.



Görsel 5: "soru-cevap" metriğinde "direct prompting" stratejisi kullanılarak girilen promptun "gpt-4o" modelindeki çıktısı

GPT-4o, **direct prompting** stratejisile verilen promptu oldukça kapsamlı ve bilimsel açıdan doğru biçimde yanıtlamıştır. Tanımda hem temel süreçler hem de süreçte yer alan kimyasal bileşenler net bir şekilde belirtilmiştir. Yanıt hem detay hem de açıklık açısından yüksek seviyededir; bağlama uygunluk ise tamamen sağlanmıştır. Bilgi düzeyi bakımından gerek bilimsel gerekse genel kullanıcı için yeterli ve tatmin edici bir çıktı sunulmuştur. Kullanılan prompt ve model yanıtının ekran görüntüsü yukarıda sunulmuştur.



Görsel 6: "soru-cevap" metriğinde "indirect prompting" stratejisi kullanılarak girilen promptun "gpt-4o" modelindeki çıktısı

GPT-4o, **indirect prompting** stratejisinde oldukça kapsamlı ve pedagojik açıdan örnek bir yanıt üretmiştir. Açıklama; günlük hayattan benzettmeler, bilimsel detaylar ve kimyasal denklemi içermekte, ayrıca farklı yaş grupları için sadeleştirilmiş versiyon da sunulmaktadır. Yanıt, doğruluk açısından eksiksiz ve hatasızdır. Açıklık, verilen örneklerle ve sade anlatımla en üst seviyededir. Detay seviyesi hem temel süreçleri hem de sürecin insanlar ve doğa için neden önemli olduğunu kapsamaktadır. Bağlama uygunluk ise tamdır; sorunun hedef kitlesine göre (öğrenci) uygun ve anlaşılır bir açıklama yapılmıştır. Kullanılan prompt ve model yanıtının ekran görüntüsü yukarıda sunulmuştur.

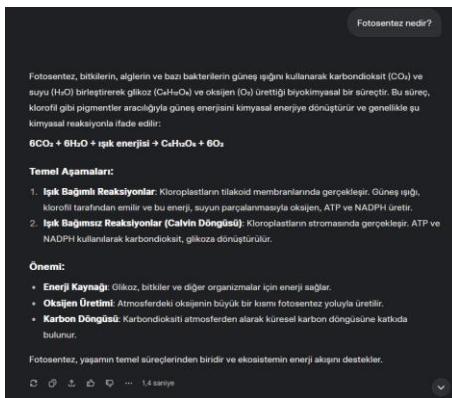
2. Model: Grok 3

Aşağıdaki tabloda, soru-cevap metriğinde tüm prompt stratejilerinin “Grok 3” modelindeki performansı 0-100 puanlama sisteme göre değerlendirilmiştir.

Strateji	Doğruluk	Açıklık	Detay	Bağlama Uygunluk
zero-shot	100	98	100	100
one-shot	96	92	75	90
few-shot	92	90	60	90
chain-of-thought	100	97	99	100
direct prompting	100	97	96	100
indirect prompting	100	100	97	100

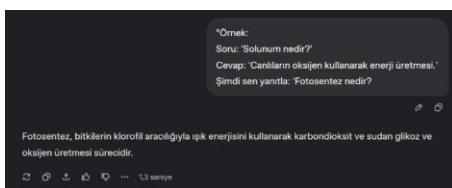
Tablo 3: Puanlama (soru-cevap metriğinde "Grok 3" modelinin prompt stratejilerine göre performansı)

a. Gözlemler (Grok 3)



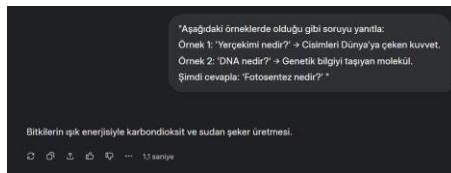
Görsel 7: "soru-cevap" metriğinde "zero-shot" stratejisi kullanılarak girilen promptun "grok 3" modelindeki çıktısı

Grok 3, **zero-shot** stratejisiyle girilen promptu, kapsamlı ve bilimsel açıdan doğru bir şekilde yanıtlamıştır. Tanımda fotosentezin biyokimyasal bir süreç olduğu açıkça belirtilmiş; süreçte kullanılan girdiler (CO_2 , H_2O) ve üretilen ürünler (glikoz, O_2) eksiksiz şekilde verilmiştir. Kimyasal denklemin sunulması, bilimsel yeterlilik açısından güçlü bir unsur olmuştur. Ayrıca, ışık bağımlı ve bağımsız reaksiyonlar gibi temel aşamalara degenilmesi, detay puanını yükselmiştir. Fotosentezin ekosistem ve karbon döngüsü üzerindeki önemi de açıklanmış, madde madde özetlenmiştir. Açıklık, başlıklandırma ve madde işaretleriyle güçlendirilmiştir. Bağlama uygunluk ise eksiksizedir; soru doğrudan ve tam olarak karşılanmıştır. Kullanılan prompt ve model yanıtının ekran görüntüsü yukarıda sunulmuştur.



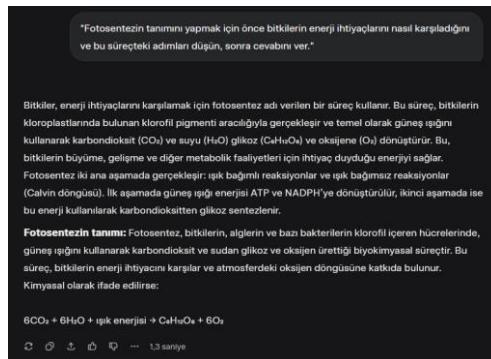
Görsel 8: "soru-cevap" metriğinde "one-shot" stratejisi kullanılarak girilen promptun "grok 3" modelindeki çıktısı

Grok 3 modeli, **one-shot** stratejisiyle verilen kısa ve öz örneğe uygun olarak yanıt tek cümle halinde sunmuştur. İfade bilimsel açıdan doğrudur ve açıklık açısından net ve anlaşılabilirdir. Ancak detay seviyesi düşük kalmıştır; süreç adımları, kimyasal denklem veya fotosentezin önemi gibi ek bilgiler yer almamaktadır. Bu durumun ana nedeni, promptta verilen örneğin kısa ve yüzeysel olmasıdır; model, örneği taklit ederek benzer biçimde yanıt vermiştir. Detay düzeyi daha yüksek bir yanıt için daha kapsamlı bir örnek sunulması gereklidir. Kullanılan prompt ve model yanıtının ekran görüntüsü yukarıda sunulmuştur.



Görsel 9: "soru-cevap" metriğinde "few-shot" stratejisi kullanılarak girilen promptun "grok 3" modelindeki çıktısı

Grok 3 modeli, **few-shot** stratejisinde verilen örneklerin kısa ve yüzeysel olması nedeniyle, yanıtını da oldukça kısa ve özetleyici şekilde vermiştir. İfade temel olarak doğru olmakla birlikte, fotosentezin önemli çıktısı olan “oksijen” den bahsedilmemiştir. Açıklık açısından temel kavramlar doğru aktarılmıştır ancak tanımın kısalığı ve bazı önemli unsurların eksikliği nedeniyle puan bir miktar düşük tutulmuştur. Detay puanı, herhangi bir ek bilgi veya açıklama sunulmadığından minimumdur. Bu durumun temel nedeni, promptta verilen örneklerin yapısı olduğundan model, örneklerin ayrıntı düzeyini taklit etmiştir. Bağlama uygunluk ise soruya doğrudan ve net bir yanıt verilmesi sayesinde tam puan almıştır. Kullanılan prompt ve model yanıtının ekran görüntüsü yukarıda sunulmuştur.



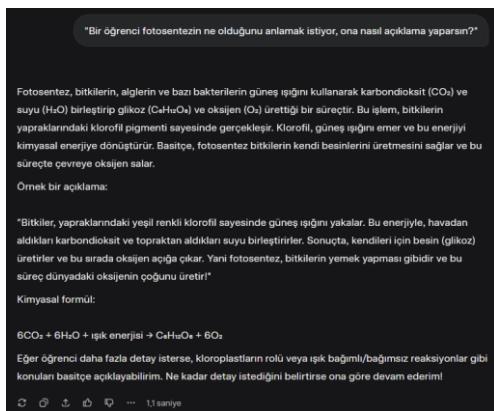
Görsel 10: "soru-cevap" metriğinde "chain-of-thought" stratejisi kullanılarak girilen promptun "grok 3" modelindeki çıktısı

Grok 3 modeli, **chain-of-thought** stratejisile yönlendirilen promptta, fotosentez sürecini detaylı ve mantıksal bir akışla açıklamıştır. Yanıttı fotosentezin temel biyokimyasal işlevleri, klorofilin rolü, süreçteki ana aşamalar kimyasal denklem ve süreç sonunda elde edilen ürünler eksiksiz biçimde verilmiştir. Bitkilerin enerji ihtiyacı ile fotosentez arasındaki ilişki vurgulanmış ve tanım bölümünde kavramsal bütünlük korunmuştur. Açıklık, başlıklandırma ve bölümlere ayırma ile oldukça yüksektir. Detay seviyesi ve bağlama uygunluk da tam puanla değerlendirilebilir. Modelin adım adım düşünme stratejisine güçlü şekilde yanıt verdiği görülmektedir. Kullanılan prompt ve model yanıtının ekran görüntüsü yukarıda sunulmuştur.



Görsel 11: "soru-cevap" metriğinde "direct prompting" stratejisi kullanılarak girilen promptun "grob 3" modelindeki çıktıtı

Grob 3, **direct prompting** stratejisile verilen “Fotosentezin tanımını açıkla” komutunu oldukça kapsamlı ve bilimsel açıdan doğru biçimde yanıtlamıştır. Tanımda fotosentezin hangi canlılarda gerçekleştiği, süreçte kullanılan girdiler ve üretilen ürünler eksiksiz olarak verilmiş; kimyasal denklem açıkça sunulmuştur. Ayrıca, fotosentezin kloroplastlarda gerçekleştiği ve ışık bağımlı/bağımsız aşamalar gibi temel adımları da belirtilmiştir. Açıklık, başlıklandırmalar ve kısa özetlerle güçlendirilmiştir. Detay seviyesi yüksek olup, süreç bilimsel ve pedagojik olarak da doğru aktarılmıştır. Bağlama uygunluk da eksiksizdir. Kullanılan prompt ve model yanıtının ekran görüntüsü yukarıda sunulmuştur.



Görsel 12: "soru-cevap" metriğinde "indirect prompting" stratejisi kullanılarak girilen promptun "grob 3" modelindeki çıktıtı

Grob 3, **indirect prompting** stratejisile verilen soruda hem bilimsel hem de pedagojik açısından oldukça zengin ve açıklayıcı bir yanıt üretmiştir. Fotosentezin temel süreci, işleyışı, kullanılan maddeler ve kimyasal denklem eksiksiz aktarılmıştır. Yanıt içinde günlük yaşamdan benzettmeler (“bitkilerin yemek yapması gibi”), kimyasal formül ve örnek açıklama yer almaktır; ayrıca öğrencinin bilgi seviyesine göre detay artırılabilcegi belirtilmektedir. Açıklık ve detay seviyesi yüksektir, yanıt hem kavramsal olarak hem de hedef kitle açısından tamamen bağlama uygundur. Model, özellikle pedagojik anlatım ve esnek detay seviyesiyle indirect

prompting stratejisiniin avantajini iyi kullanmisstir. Kullanicilan prompt ve model yanitinin ekran goruntusu yukarıda sunulmusstur.

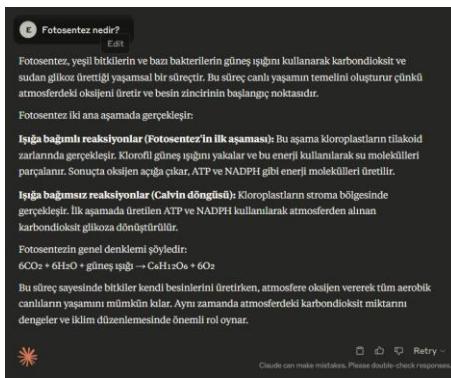
3. Model: Claude Sonnet 4

Aşagidakı tabloda, soru-cevap metriğinde tüm prompt stratejilerinin “Claude Sonnet 4” modelindeki performansı 0-100 puanlama sistemine göre değerlendirilmiştir.

Strateji	Doğruluk	Açıklık	Detay	Bağlama Uygunluk
zero-shot	100	98	100	100
one-shot	96	95	70	89
few-shot	85	80	40	75
chain-of-thought	100	100	98	100
direct prompting	100	99	96	100
indirect prompting	100	100	97	100

Tablo 4: Puanlama (soru-cevap metriğinde "Claude Sonnet 4" modelinin prompt stratejilerine göre performansı)

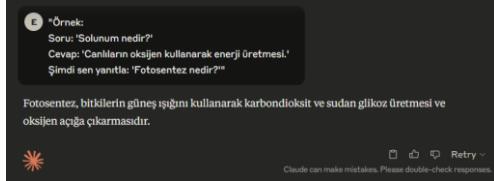
a. Gözlemler (Claude Sonnet 4)



Görsel 13:"soru-cevap" metriğinde "zero-shot" stratejisi kullanilarak girilen promptun "claude sonnet 4" modelindeki çıktı

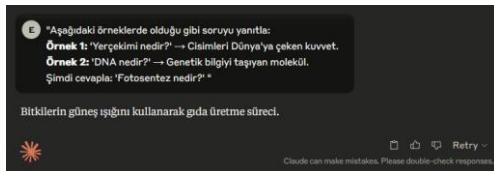
Claude Sonnet 4, **zero-shot** stratejisinde “Fotosentez nedir?” sorusunu bilimsel açidan eksiksiz, açık ve detaylı şekilde yanitlamışstir. Cevapta fotosentezin tanımı, canlı yaşamındaki rolü, süreçte gerçekleşen ana aşamalar (ışığa bağımlı ve bağımsız reaksiyonlar) başlıklandırılarak verilmiş, kimyasal denklem doğru şekilde aktarılmışstir. Ayrıca süreçlerin mekanizması ve sonuçları pedagojik bir dille özetlenmiş; ek olarak fotosentezin ekolojik ve iklimsel önemi de vurgulanmıştır. Açıklık, detay ve bağlama uygunluk yönünden herhangi bir

eksik bulunmamaktadır. Kullanılan prompt ve model yanıtının ekran görüntüsü yukarıda sunulmuştur.



Görsel 14: "soru-cevap" metriğinde "one-shot" stratejisi kullanılarak girilen promptun "claude sonnet 4" modelindeki çıktısı

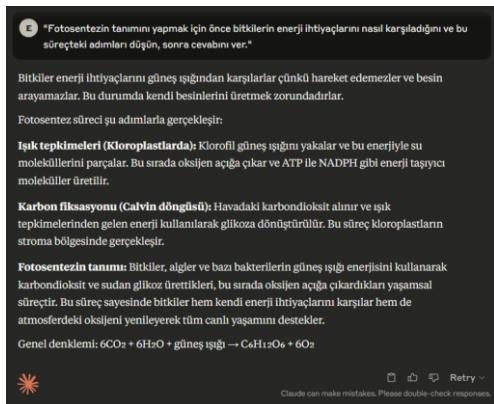
Claude Sonnet 4, **one-shot** stratejisiyle verilen kısa ve örnekleyici prompta uygun olarak yanıtı kısa ve temel biçimde sunmuştur. Yanitta, fotosentezin doğru olarak “bitkilerin güneş ışığını kullanarak karbondioksit ve sudan glikoz üretmesi ve oksijen açığa çıkarması” şeklinde tanımlanmıştır. Açıklık ve doğruluk açısından herhangi bir hata yoktur. Ancak detay seviyesi düşük tutulmuştur; süreç adımları, kimyasal denklem, fotosentezin önemi veya mekanizması hakkında ek bilgi sunulmamıştır. Detay puanının düşük olmasının ana nedeni, promptun kısa ve yüzeysel örnek içermesidir. Model, örneği taklit ederek benzer biçimde yanıt vermiştir. Bağlama uygunluk ise eksiksizdir. Kullanılan prompt ve model yanıtının ekran görüntüsü yukarıda sunulmuştur.



Görsel 15: "soru-cevap" metriğinde "few-shot" stratejisi kullanılarak girilen promptun "claude sonnet 4" modelindeki çıktısı

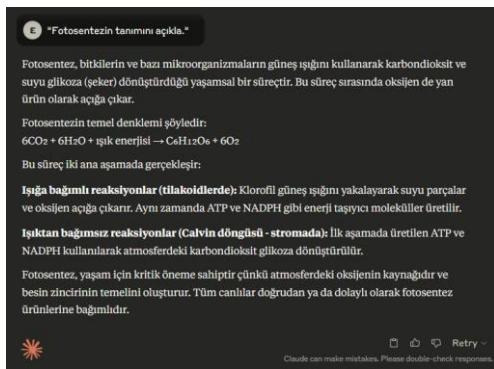
Claude Sonnet 4 modeli, **few-shot** stratejisinde verilen örneklerin kısa ve yüzeysel olması nedeniyle, yanıtını da son derece kısa ve özetleyici biçimde sunmuştur. “Bitkilerin güneş ışığını kullanarak gıda üretme süreci” ifadesi genel olarak doğru olsa da fotosentezin çıktıları (ör. oksijen, glikoz/şeker) ve kimyasal süreçle ilgili önemli ayrıntılar eksiktir; bu nedenle doğruluk ve açıklık puanları 4 olarak verilmiştir. Detay seviyesi, herhangi bir ek bilgi veya süreç adımı sunulmadığından 1 olarak değerlendirilmiştir. Bu durumun temel nedeni, promptta verilen örneklerin yüzeyselliğidir; model, örneklerin ayrıntı düzeyini doğrudan taklit etmiştir. Bağlama

uygunluk ise soruya doğrudan yanıt verilmiş olması nedeniyle tam puan almıştır. Kullanılan prompt ve model yanıtının ekran görüntüsü yukarıda sunulmuştur.



Görsel 16: "soru-cevap" metriğinde "chain-of-thought" stratejisi kullanılarak girilen promptun "claude sonnet 4" modelindeki çıktısı

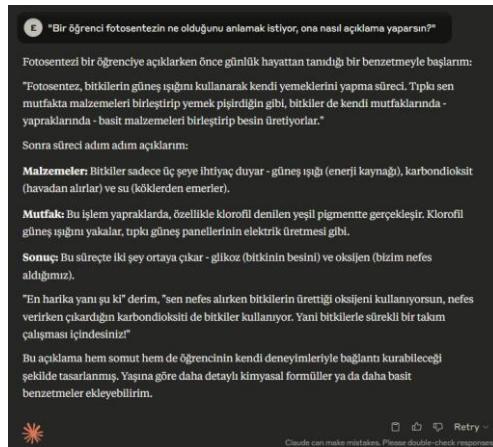
Claude Sonnet 4, **chain-of-thought** stratejisinde verilen promptu adım adım düşünme ve ayrıntılılandırma açısından tam olarak karşılamıştır. Yanıttı fotosentez sürecinin temel nedeni ("hareket edememek, besin arayamamak"), sürecin aşamaları (ışık tepkimeleri ve karbon fiksasyonu), fotosentezin tanımı ve kimyasal denklemi açıkça, başlıklarla ve ayrıntılı biçimde sunulmuştur. Ayrıca fotosentezin ekolojik önemi de vurgulanmıştır. Açıklık, detay ve doğruluk en üst düzeyde olup, modelin adım adım yönlendirme içeren promptlara yüksek derecede uyum gösterdiği gözlenmiştir. Kullanılan prompt ve model yanıtının ekran görüntüsü yukarıda sunulmuştur.



Görsel 17: "soru-cevap" metriğinde "direct prompting" stratejisi kullanılarak girilen promptun "claude sonnet 4" modelindeki çıktısı

Claude Sonnet 4, **direct prompting** stratejisiyle verilen "Fotosentezin tanımını açıkla." komutunu oldukça kapsamlı ve bilimsel olarak eksiksiz bir şekilde yanıtlamıştır. Tanımda fotosentezin kimlerde gerçekleştiği, süreçte kullanılan girdiler ve ürünler, sürecin temel

aşamaları (ışığa bağımlı ve bağımsız reaksiyonlar) ve kimyasal denklem detaylı biçimde verilmiştir. Sürecin ekolojik önemi ve canlı yaşamındaki rolü de ayrıca vurgulanmıştır. Açıklık, detay ve bağlama uygunluk açısından yanıt üst düzeydedir. Model, direct prompting stratejisine verilen bekleniyi tam olarak karşılamıştır. Kullanılan prompt ve model yanıtının ekran görüntüsü yukarıda sunulmuştur.



Görsel 18: "soru-cevap" metriğinde "indirect prompting" stratejisi kullanılarak girilen promptun "claude sonnet 4" modelindeki çıktıtı

Claude Sonnet 4, **indirect prompting** stratejisile verilen promptu oldukça pedagojik ve yaratıcı bir şekilde yanıtlamıştır. Yanıta günlük hayattan bir mutfak benzetmesiyle başlamak, konunun anlaşılabilirliğini önemli ölçüde artırmıştır. Süreç, "malzemeler", "mutfak" ve "sonuç" başlıklarıyla adım adım açıklanmış; fotosentezin girdileri, gerçekleştiği yer ve çıktıları sade ve doğru biçimde aktarılmıştır. Ayrıca, öğrencinin kendi deneyimleriyle bağlantı kurabilmesini sağlayacak örnekler ve benzetmeler kullanılmış, yaşa göre anlatımın uyarlanabileceği vurgulanmıştır. Bilimsel doğruluk, açıklık ve detay eksiksiz; bağlama uygunluk ise yüksek düzeydedir. Kullanılan prompt ve model yanıtının ekran görüntüsü yukarıda sunulmuştur.

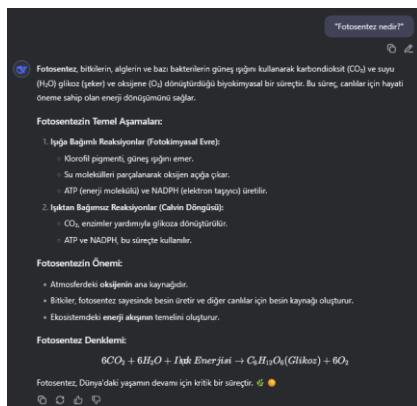
4. Model: DeepSeek V3

Aşağıdaki tabloda, soru-cevap metriğinde tüm prompt stratejilerinin “DeepSeek V3” modelindeki performansı 0-100 puanlama sistemine göre değerlendirilmiştir.

Strateji	Doğruluk	Açıklık	Detay	Bağlama Uygunluk
zero-shot	100	99	98	100
one-shot	98	95	85	90
few-shot	85	84	45	72
chain-of-thought	100	100	98	100
direct prompting	100	99	99	100
indirect prompting	100	100	97	100

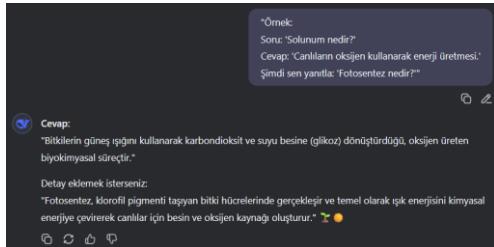
Tablo 5: Puanlama (soru-cevap metriğinde “DeepSeek V3” modelinin prompt stratejilerine göre performansı)

a. Gözlemler (DeepSeek V3)



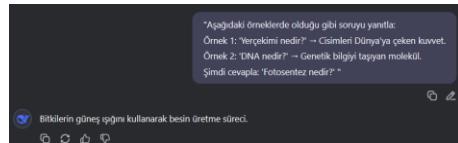
Görsel 19: "soru-cevap" metriğinde "zero-shot" stratejisi kullanılarak girilen promptun "deepseek v3" modelindeki çıktıtı

DeepSeek V3, **zero-shot** stratejisile verilen “Fotosentez nedir?” sorusunu bilimsel olarak eksiksiz, açık ve oldukça detaylı biçimde yanıtlamıştır. Tanımda fotosentezin canlılar için önemi vurgulanmış, süreçteki ana aşamalar (ışığa bağımlı ve bağımsız reaksiyonlar) başlıklandırılmış ve içerikleri özetlenmiştir. Fotosentezin ekosistem, oksijen üretimi ve besin zinciri üzerindeki rolü madde madde açıklanmış, kimyasal denklem doğru biçimde sunulmuştur. Cevap, sade bir dil ve başlıklar sayesinde oldukça açık ve herkesin anlayabileceği şekilde düzenlenmiştir. Bağlama uygunluk açısından da eksiksizdir. Kullanılan prompt ve model yanıtının ekran görüntüsü yukarıda sunulmuştur.



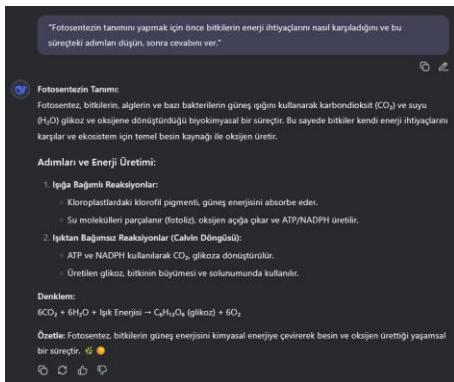
Görsel 20:"soru-cevap" metriğinde "one-shot" stratejisi kullanılarak girilen promptun "deepseek v3" modelindeki çıktısı

DeepSeek V3, **one-shot** stratejisiyle verilen kısa örnekli prompta uygun şekilde, fotosentezin temel tanımını tek cümleyle sunmuştur: “Bitkilerin güneş ışığını kullanarak karbondioksit ve suyu besine (glikoz) dönüştürdüğü, oksijen üreten biyokimyasal süreçtir.” Bilimsel olarak doğru ve açık bir ifadedir. Detay düzeyi ise orta seviyede tutulmuştur; ancak model, “Detay eklemek isterseniz...” başlığı altında ek bilgi sunarak, klorofil ve kimyasal enerji dönüşümünü de kısa şekilde açıklamıştır. Model, promptun kısa ve örnekleyici yapısına uygun bir yanıt üretmiştir. Bağlama uygunluk ve açıklık ise tamdır. Kullanılan prompt ve model yanıtının ekran görüntüsü yukarıda sunulmuştur.



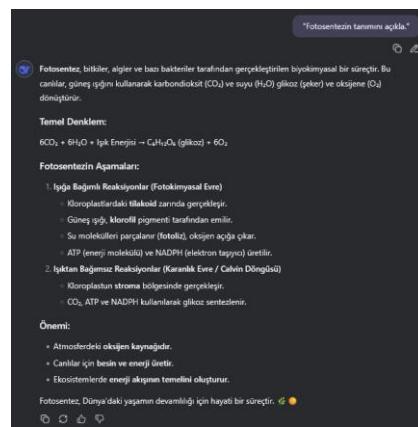
Görsel 21:"soru-cevap" metriğinde "few-shot" stratejisi kullanılarak girilen promptun "deepseek v3" modelindeki çıktısı

DeepSeek V3 modeli, **few-shot** stratejisinde verilen örneklerin kısa ve yüzeysel olması nedeniyle, yanıtını da oldukça kısa ve özet bir şekilde sunmuştur. “Bitkilerin güneş ışığını kullanarak besin üretme süreci” ifadesi temel olarak doğru ve anlaşılır olmakla birlikte, fotosentezin diğer çıktısı olan oksijen ve kimyasal süreç hakkında bilgi eksiktir. Bu nedenle doğruluk ve açıklık puanları düşük verilmiştir. Detay seviyesi ise herhangi bir ek açıklama veya süreç adımı içermemişinden 30 olarak puanlanmıştır. Bu sonuç, modelin promptta verilen örneklerin ayrıntı düzeyini taklit ettiğini göstermektedir. Kullanılan prompt ve model yanıtının ekran görüntüsü yukarıda sunulmuştur.



Görsel 22: "soru-cevap" metriğinde "chain-of-thought" stratejisi kullanılarak girilen promptun "deepseek v3" modelindeki çıktısı

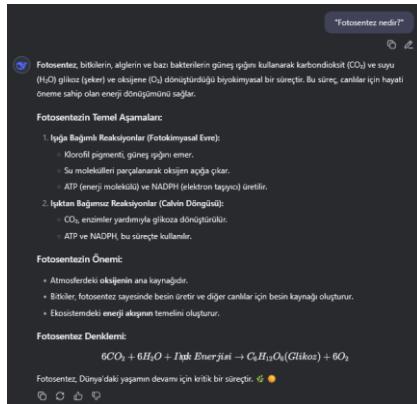
DeepSeek V3, **chain-of-thought** stratejisile verilen bu prompta detaylı, mantıksal akışı güçlü ve bilimsel açıdan eksiksiz bir yanıt üretmiştir. Fotosentezin tanımının yanı sıra, sürecin adım adım aşamaları (ışığa bağımlı ve bağımsız reaksiyonlar) başlıklandırılarak ve maddeler halinde detaylıca açıklanmıştır. Kimyasal denklem doğru ve açık bir şekilde sunulmuş, sürecin ekosistem ve enerji üretimi açısından önemi vurgulanmıştır. Ayrıca özet cümleyle konunun genel bağlamı netleştirilmiştir. Açıklık ve detay seviyesi yüksek; bağlama uygunluk ise tamdır. Kullanılan prompt ve model yanıtının ekran görüntüsü yukarıda sunulmuştur.



Görsel 23: "soru-cevap" metriğinde "direct prompting" stratejisi kullanılarak girilen promptun "deepseek v3" modelindeki çıktısı

DeepSeek V3, **direct prompting** stratejisile verilen bu prompta kapsamlı ve sistematik bir yanıt üretmiştir. Fotosentezin tanımı bilimsel olarak doğru bir şekilde sunulmuş, temel kimyasal denklem belirtilmiş ve sürecin iki ana aşaması (ışığa bağımlı ve bağımsız reaksiyonlar) başlıklandırılarak ayrıntılı biçimde açıklanmıştır. Ayrıca fotosentezin ekosistem, oksijen kaynağı ve besin üretimi açısından önemi madde özetiyle sunulmuştur. Açıklık ve detay

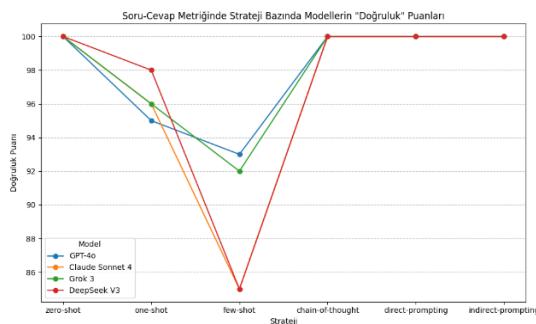
seviyesi üst düzeyde olup, sunumun düzeni ve başlıklandırılması anlaşılır bir yapı oluşturmuştur. Bağlama uygunluk ise eksiksizdir. Kullanılan prompt ve model yanıtının ekran görüntüsü yukarıda sunulmuştur.



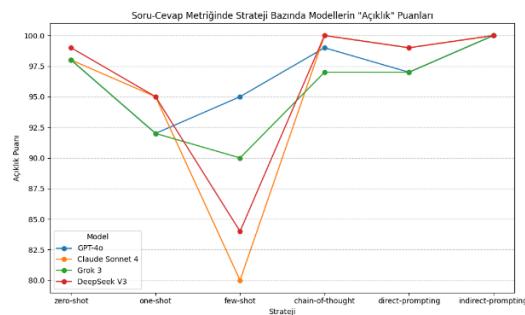
Görsel 24: "soru-cevap" metriğinde "indirect prompting" stratejisi kullanılarak girilen promptun "deepseek v3" modelindeki çıktısı

DeepSeek V3, **indirect prompting** stratejisiyle verilen prompta hem bilimsel hem pedagojik açıdan oldukça kapsamlı ve anlaşılır bir yanıt üretmiştir. Fotosentezin formülü, temel denklem, sürecin ekolojik ve biyolojik önemi, aşamaları ve sürecin gerçekleşme biçimini maddeler ve başlıklarla sistematik biçimde açıklanmıştır. Ayrıca, “güneş enerjisiyle yemek yapma” gibi öğrencinin kolayca kavrayabileceği bir benzetmeye özet yapılmıştır. Bilimsel doğruluk, açıklık, detay ve bağlama uygunluk açısından eksiksizdir. Ayrıca öğrenciye “daha fazla detay sorabilirsin” gibi teşvik edici bir dil kullanılmıştır. Kullanılan prompt ve model yanıtının ekran görüntüsü yukarıda sunulmuştur.

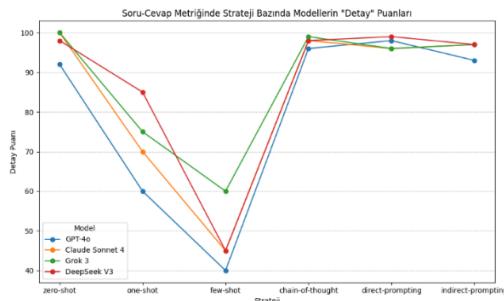
5. Soru-Cevap Metriği Sonuçlarının Strateji ve Model Bazında Karşılaştırılması



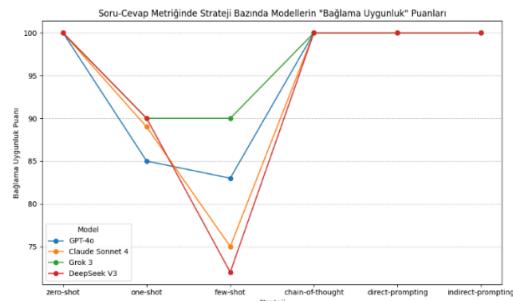
Grafik 1:Soru-cevap metriğinde strateji bazında modellerin doğruluk puanları



Grafik 2:Soru-cevap metriğinde strateji bazında modellerin açıklık puanları



Grafik 3:Soru-cevap metriğinde strateji bazında modellerin detay puanları



Grafik 4:Soru-cevap metriğinde strateji bazında modellerin bağlama uygunluk puanları

Grafik 1'de, tüm modellerin belirli stratejilerde maksimum doğruluk performansına ulaştığı, özellikle zero-shot, chain-of-thought, direct prompting ve indirect prompting stratejilerinin yüksek tutarlılık ve başarı sağladığını gözlemlenmiştir. Buna karşılık, one-shot ve few-shot stratejilerinde modeller arasında belirgin performans farklılıklarını oluşturmaktadır; bazı modeller bu stratejilerde daha düşük doğruluk sergilemektedir. Bulgular, strateji seçiminin model doğruluğu üzerinde doğrudan etkili olduğunu ve bazı stratejilerde tüm modellerin eşit başarı gösterebildiğini, bazı stratejilerde ise model bazlı ayırmaların ortaya çıktığını göstermektedir.

Grafik 2'de, modellerin stratejiye bağlı olarak benzer eğilimler sergilediği gözlemlenmektedir. Zero-shot, chain-of-thought, direct prompting ve indirect prompting stratejilerinde modeller genel olarak yüksek açıklık puanlarına ulaşmıştır. Buna karşılık, özellikle few-shot stratejisinde tüm modellerin açıklık puanlarında belirgin bir düşüş yaşandığı ve bu stratejide modeller arasındaki farkların daha belirgin hale geldiği görülmektedir. One-shot stratejisinde de benzer şekilde açıklık puanları görece daha düşük seviyelerdedir. Bu bulgular, prompt stratejisi seçiminin modellerin açıklık performansını önemli ölçüde

etkilediğini ve bazı stratejilerin tüm modeller için daha avantajlı sonuçlar üretebildiğini ortaya koymaktadır.

Grafik 3'te, modellerin stratejiye bağlı performanslarında belirgin dalgalanmalar gözlenmektedir. Özellikle one-shot ve few-shot stratejilerinde tüm modellerin detay puanlarında ciddi bir düşüş yaşanmaktadır, few-shot stratejisinde bu azalma en belirgin seviyeye ulaşmaktadır. Buna karşılık, chain-of-thought, direct prompting ve indirect prompting stratejilerinde modellerin detay puanlarının yeniden yüksek seviyelere çıktığı, zero-shot stratejisinde ise genel olarak yüksek bir detay performansı sergilediği görülmektedir. Bu bulgular, detaylı cevap üretme açısından, one-shot ve few-shot stratejilerinin görelilik olarak daha düşük performans sunduğunu, buna karşın diğer stratejilerin modellerin detay başarısını artırdığını ortaya koymaktadır.

Grafik 4'te, modellerin genel olarak çoğu stratejide yüksek bir performans sergilediği görülmektedir. Özellikle zero-shot, chain-of-thought, direct prompting ve indirect prompting stratejilerinde modellerin bağlama uygunluk puanları maksimum seviyeye yaklaşmaktadır. Ancak, one-shot ve özellikle few-shot stratejilerinde tüm modellerde anlamlı bir puan düşüşü göze çarpmaktadır; bu azalma few-shot stratejisinde daha belirgindir. Bu bulgular, bağlama uygunluğu açısından bazı stratejilerin modellerde tutarlı ve yüksek başarı sağladığını, bazı stratejilerde ise performansın ciddi şekilde dalgalandığını göstermektedir. Sonuç olarak, strateji seçiminin bağlama uygunluk performansı üzerinde önemli bir etkisi olduğu ve en yüksek uyumluluğun belirli stratejilerde elde edildiği söylenebilir.

B. Metrik: Özetleme Kalitesi

Görev: Belirli prompt stratejilerine göre girilen promptların anlam bütünlüğünü ve ana fikirlerini koruyarak açık, doğru ve öz bir şekilde özetlemek.

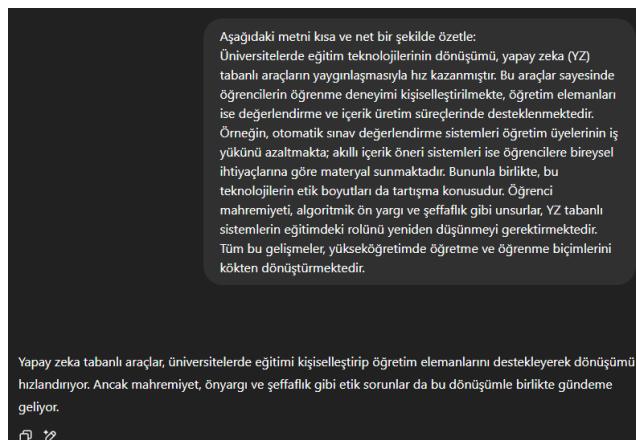
1. Model: GPT-4o

Aşağıdaki tabloda, özetleme kalitesi metriğinde tüm prompt stratejilerinin “GPT-4o” modelindeki performansı 0-100 puanlama sistemine göre değerlendirilmiştir.

Strateji	Doğruluk	Açıklık	Detay	Bağlama Uygunluk
zero-shot	100	98	92	100
one-shot	100	99	89	100
few-shot	100	98	85	100
chain-of-thought	100	100	98	100
direct prompting	100	100	80	100
indirect prompting	100	100	86	100

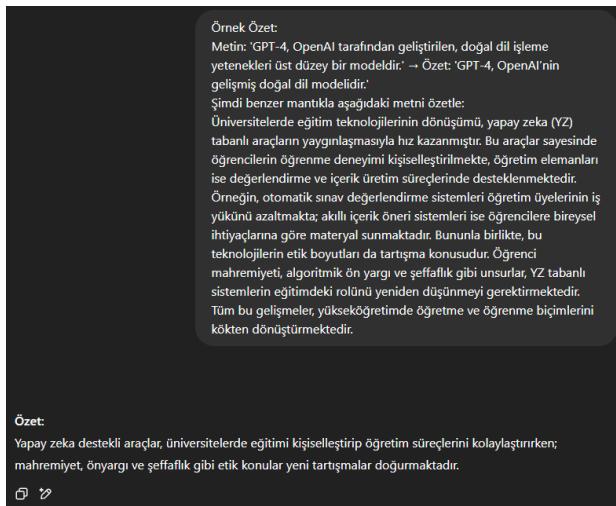
Tablo 6: Puanlama (özetleme kalitesi metriğinde “GPT-4o” modelinin prompt stratejilerine göre performansı)

a. Gözlemler (GPT-4o)



Görsel 25: "özetleme kalitesi" metriğinde "zero-shot" stratejisi kullanılarak girilen promptun "gpt-4o" modelindeki çıktısı

GPT-4o modeli, **zero-shot** stratejisine göre verilen uzun metni özetleme görevinde, temel içerik unsurlarını koruyarak kısa ve net bir özeti üretmiştir. Metindeki “değerlendirme sistemleri, içerik öneri sistemleri, materyal sunumu” gibi işlevsel ayrıntılar özetlemede yer almamış, fakat bu sadeleştirme özinin amacına uygundur. Anlatım açık, sade ve net olup, ana fikir ve ikincil önemli noktalar korunmuştur. Bağlama uygunluk ise yüksektir; çünkü istenen kısalık ve netlik sağlanmıştır. Kullanılan prompt ve model yanıtının ekran görüntüsü yukarıda sunulmuştur.



Görsel 26: "özetleme kalitesi" metriğinde "one-shot" stratejisi kullanılarak girilen promptun "gpt-4o" modelindeki çıktısı

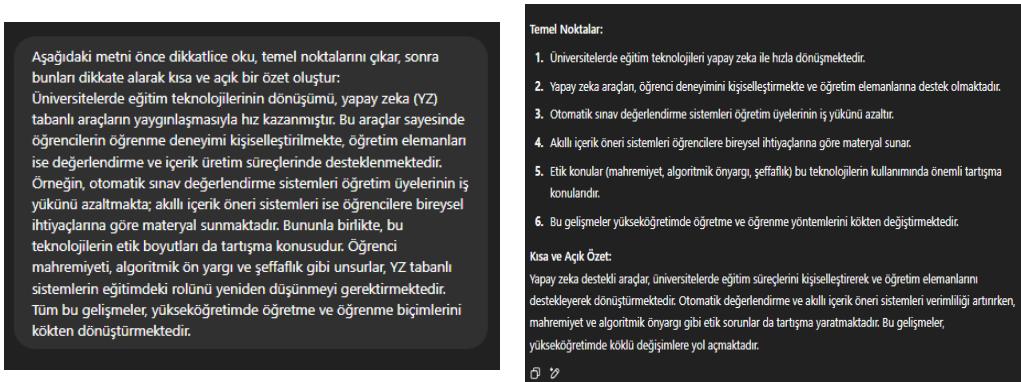
GPT-4o modeli, **one-shot** stratejisinde verilen örnek özetin metnin biçimine uygun olarak metni kısa ve açık bir şekilde özetlemiştir. Orijinal metindeki "değerlendirme sistemleri, materyal sunumu" gibi teknik ayrıntılar özetin dışında bırakılmış, ancak temel temalar eksiksiz aktarılmıştır. Anlatım sade, açık ve özlü olup, özet formatına uygundur. Bağlama uygunluk yüksektir; çünkü verilen örnekle paralel bir biçimde, kısa ve vurucu bir özet sunulmuştur. Kullanılan prompt ve model yanıtının ekran görüntüsü yukarıda sunulmuştur.



Görsel 27: "özetleme kalitesi" metriğinde "few-shot" stratejisi kullanılarak girilen promptun "gpt-4o" modelindeki çıktısı

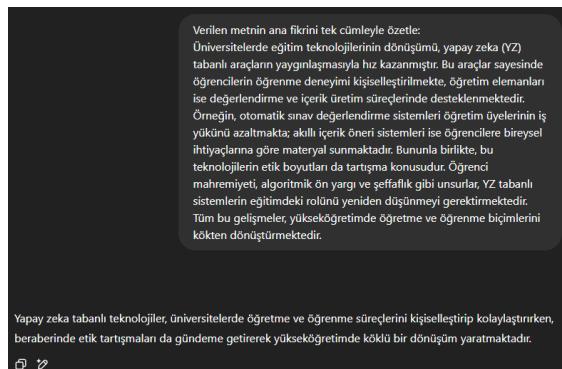
GPT-4o modeli, **few-shot** stratejisinde verilen kısa ve vurucu örneklerine uygun olarak metni sade ve özetleyici biçimde aktarmıştır. Orijinal metindeki otomatik değerlendirme sistemleri, materyal sunumu gibi spesifik uygulamalara özetlemede yer verilmemiştir. Ancak

bu, örneklerin yönlendirdiği kısa format nedeniyle tercih edilmiştir. Açıklık ve netlik yüksek, ana fikirler kaybolmamıştır. Bağlama uygunluk ise verilen örnekler ve özet formatına tamamen uygundur. Kullanılan prompt ve model yanıtının ekran görüntüsü yukarıda sunulmuştur.



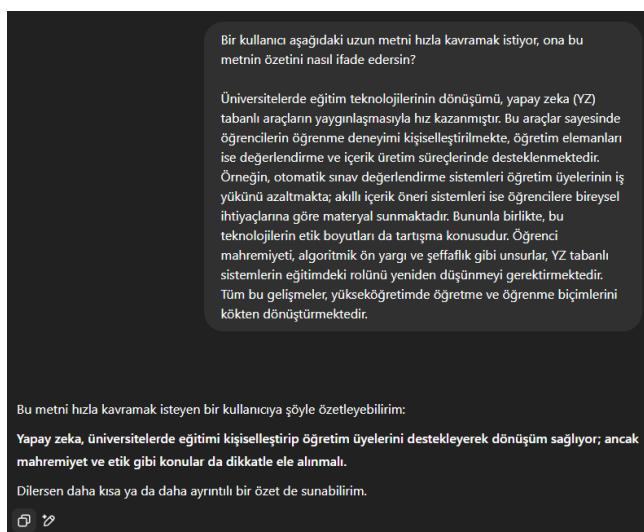
Görsel 28: "özetleme kalitesi" metriğinde "chain-of-thought" stratejisi kullanılarak girilen promptun "gpt-4o" modelindeki çıktıtı

GPT-4o modeli, **chain-of-thought** stratejisinde metni önce temel parçalara ayırip ardından bu parçaları dikkate alarak kapsamlı ve net bir özet oluşturmuştur. Ana fikirlerin hepsi korunmuş hem işlevsel ayrıntılar hem de etik boyutlar özetlenmiştir. Açıklık çok yüksek; aşamalı yapı ve maddeleme ile bilgi akışı çok net sağlanmıştır. Detay düzeyi ise hem liste hem kısa özet sayesinde formatına göre yüksektir. Bağlama uygunluk ve özetin istenen biçim tam olarak sağlanmıştır. Kullanılan prompt ve model yanıtının ekran görüntüsü yukarıda sunulmuştur.



Görsel 29: "özetleme kalitesi" metriğinde "direct prompting" stratejisi kullanılarak girilen promptun "gpt-4o" modelindeki çıktıtı

GPT-4o modeli, **direct prompting** stratejisinde verilen metnin ana fikrini tek cümleyle özetleme talebini başarılı biçimde yerine getirmiştir. Cümlede dönüşümün kapsamı ve etkisi net bir biçimde ifade edilmiştir. Orijinal metindeki işlevsel ayrıntılar ve örnekler özetin dışında bırakılmıştır, fakat bu talebin doğası geregidir. Açıklık ve sadelik yüksek, metnin ana mesajı kaybolmamıştır. Bağlama uygunluk üst düzeydedir; çünkü promptun “tek cümlede özetle” talebine tam uyulmuştur. Kullanılan prompt ve model yanıtının ekran görüntüsü yukarıda sunulmuştur.



Görsel 30: "özetleme kalitesi" metriğinde "indirect prompting" stratejisi kullanılarak girilen promptun "gpt-4o" modelindeki çıktısı

GPT-4o modeli, **indirect prompting** stratejisinde verilen metni bir kullanıcının hızla kavrayacağı şekilde özetlemiş ve özetin sonunda, istege göre daha kısa veya daha detaylı özet sunulabileceğini belirtmiştir. Metnin temel mesajı ve dönüşüm etkisi doğru aktarılmıştır. Açıklık düzeyi yüksek, vurgu cümlelerle güçlendirilmiş ve kısa, öz biçimde sunulmuştur. Detay, özet formatına göre yeterli düzeydedir. Bağlama uygunluk ise yüksek; çünkü promptun hedeflediği “kullanıcı odaklı, pratik özet” bekantisine tam uyulmuştur. Kullanılan prompt ve model yanıtının ekran görüntüsü yukarıda sunulmuştur.

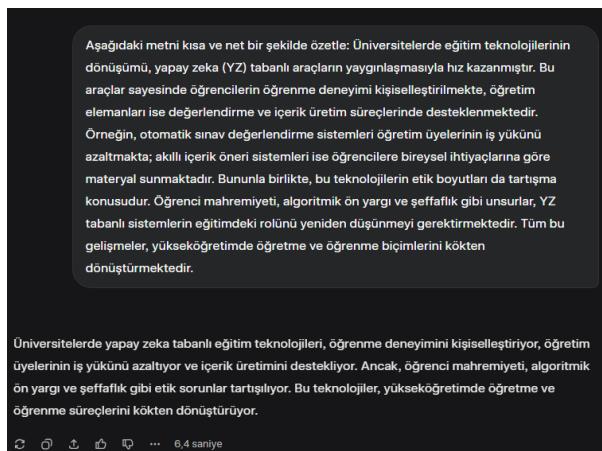
2. Model: Grok 3

Aşağıdaki tabloda, özetleme kalitesi metriğinde tüm prompt stratejilerinin “Grok 3” modelindeki performansı 0-100 puanlama sistemine göre değerlendirilmiştir.

Strateji	Doğruluk	Açıklık	Detay	Bağlama Uygunluk
zero-shot	100	98	90	100
one-shot	100	99	88	100
few-shot	100	98	80	100
chain-of-thought	100	100	97	100
direct prompting	100	100	80	100
indirect prompting	100	100	95	100

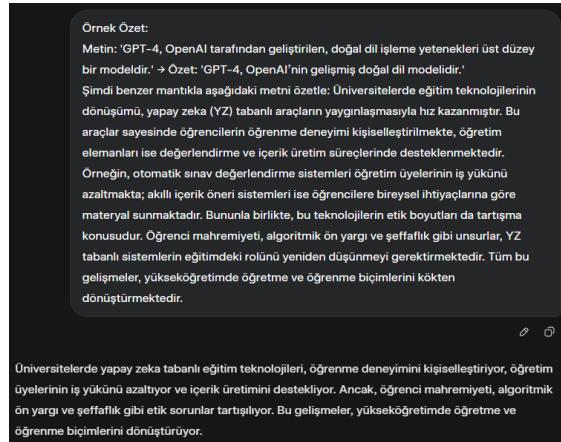
Tablo 7: Puanlama (özetleme kalitesi metriğinde “Grok 3” modelinin prompt stratejilerine göre performansı)

a. Gözlemler (Grok 3)



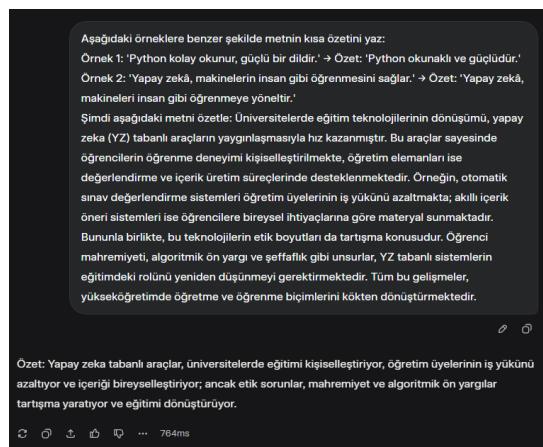
Görsel 31: “özetleme kalitesi” metriğinde “zero-shot” stratejisi kullanılarak girilen promptun “grok 3” modelindeki çıktıtı

Grok 3 modeli, **zero-shot** stratejisile metni kısa ve net şekilde özetleme görevini başarılı bir biçimde yerine getirmiş metnin ana fikirleri ve ikincil önemli noktalar eksiksiz aktarılmıştır. Açıklık ve anlaşılır anlatım üst düzeyde olup, özet amaca tamamen uygundur. Detay seviyesi özetin doğası gereği sınırlandırılmış, ancak temel mesaj kaybolmamıştır. Bağlama uygunluk ise yüksektir; çünkü özet istenen kısa ve net biçimde sunulmuştur. Kullanılan prompt ve model yanıtının ekran görüntüsü yukarıda sunulmuştur.



Görsel 32: "özetleme kalitesi" metriğinde "one-shot" stratejisi kullanılarak girilen promptun "grok 3" modelindeki çıktıtı

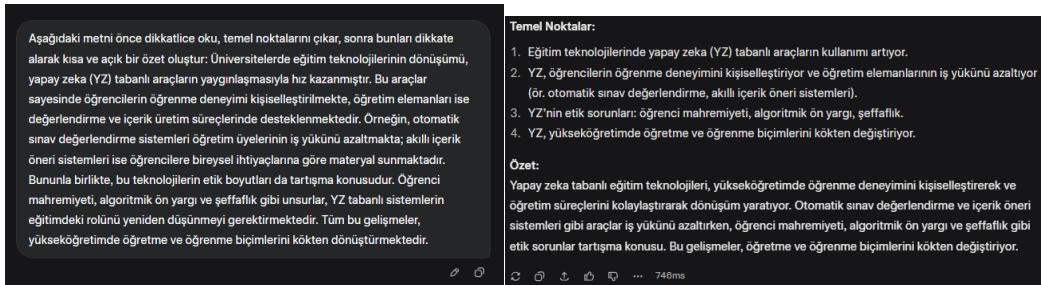
Grok 3 modeli, **one-shot** stratejisinde verilen örnek özetin biçimine uygun olarak metni kısa, açık ve kapsamlı şekilde özetlemiştir. Metnin temel mesajı, yan unsurlar ve işlevsel ayrıntılar özetin sınırları çerçevesinde etkin biçimde sunulmuştur. Açıklık ve netlik üst düzeyde, özetin örnek biçimyle tam uyumludur. Detay ise örneğin kısa biçimini nedeniyle gereği kadar sadeleştirilmiş ve amaca uygundur. Bağlama uygunluk yüksektir; çünkü model örnek özetin talebine tam olarak yanıt vermiştir. Kullanılan prompt ve model yanıtının ekran görüntüsü yukarıda sunulmuştur.



Görsel 33: "özetleme kalitesi" metriğinde "few-shot" stratejisi kullanılarak girilen promptun "grok 3" modelindeki çıktıtı

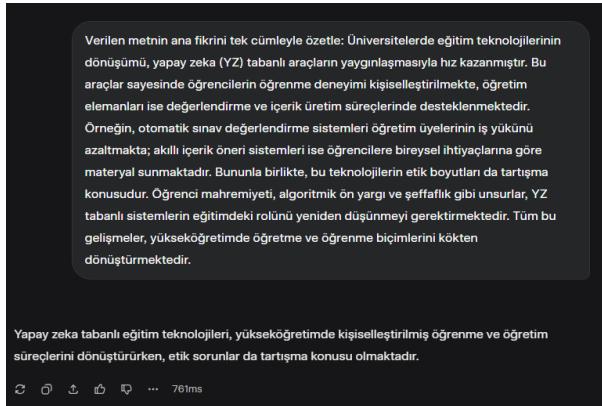
Grok 3 modeli, **few-shot** özetleme stratejisinde verilen kısa ve sade örnekleri doğrudan taklit etmiş, yanıtında da temel mesajı koruyarak metni oldukça yüzeysel ve kısa biçimde özetlemiştir. Özettir, anlam bütünlüğünü korusa da metindeki özgün detaylar ve ikincil bilgiler özet dışında kalmıştır. Açıklık açısından sorunsuz, fakat detay düzeyi düşük tutulmuştur.

Modelin daha kapsamlı bir özet sunmamasının ana sebebi, promptta verilen örneklerin biçimsel olarak kısa ve özetleyici olmasıdır. Kullanılan prompt ve model yanıtının ekran görüntüsü yukarıda sunulmuştur.



Görsel 34: "özetleme kalitesi" metriğinde "chain-of-thought" stratejisi kullanılarak girilen promptun "grok 3" modelindeki çıktısı

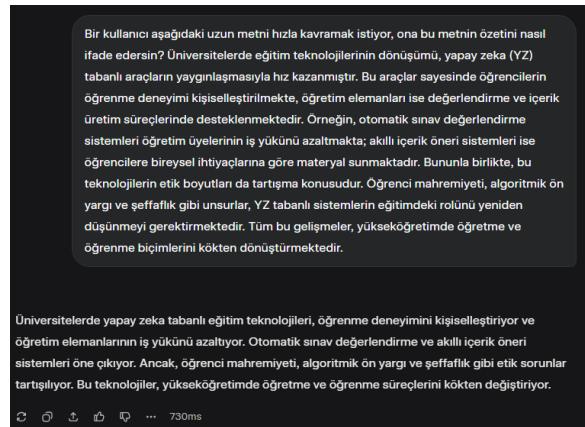
Grok 3 modeli, **chain-of-thought** stratejisinde, metni önce temel başlıklara ayırmış ve ardından bu başlıkları dikkate alarak kısa ve genel bir özet oluşturmuştur. Temel noktaları maddeleyerek sunması, metnin analiz sürecinin şeffaflığını ve anlaşılabilirliğini artırmıştır. Metindeki otomatik değerlendirme sistemleri ve içerik öneri sistemleri gibi özgün detaylar özet içinde korunmuş, gereksiz ayrıntılara yer verilmemiştir. Modelin aşamalı analizle ürettiği özet hem anlam bütünlüğünü hem de vurucu noktaları başarıyla aktarmaktadır. Kullanılan prompt ve model yanıtının ekran görüntüsü yukarıda sunulmuştur.



Görsel 35: "özetleme kalitesi" metriğinde "direct prompting" stratejisi kullanılarak girilen promptun "grok 3" modelindeki çıktısı

Grok 3 modeli, **direct prompting** stratejisile verilen metni, ana fikri tek cümlede özetleme talebine uygun olarak yanıtlamıştır. Yanıt, orijinal metindeki karmaşık yapıyı sadeleştirerek özetlemiş ve istenen kısa, vurucu ana mesajı başarıyla iletmıştır. Ancak,

örneklerdeki detayların ve yan temaların çoğu doğası gereği dışında bırakılmıştır. Kullanılan prompt ve model yanıtının ekran görüntüsü yukarıda sunulmuştur.



Görsel 36: "özetleme kalitesi" metriğinde "indirect prompting" stratejisi kullanılarak girilen prompt'un "grok 3" modelindeki çıktıtı

Grok 3 modeli, **indirect prompting** stratejisinde, verilen uzun metni bir kullanıcının hızla kavrayacağı şekilde özetlemiş ve özetin temel mesajlarını açık, net ve akıcı bir dille sunmuştur. Özet hem anlam bütünlüğü hem de bağlama uygunluk açısından eksiksizdir. Kullanılan prompt ve model yanıtının ekran görüntüsü yukarıda sunulmuştur.

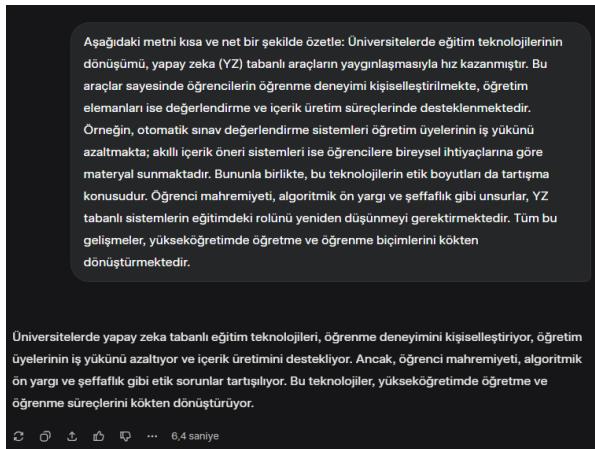
3. Model: Claude Sonnet 4

Aşağıdaki tabloda, özetleme kalitesi metriğinde tüm prompt stratejilerinin "Claude Sonnet 4" modelindeki performansı 0-100 puanlama sistemine göre değerlendirilmiştir.

Strateji	Doğruluk	Açıklık	Detay	Bağlama Uygunluk
zero-shot	100	100	95	100
one-shot	100	100	90	100
few-shot	100	100	85	100
chain-of-thought	100	100	98	100
direct prompting	100	100	85	100
indirect prompting	100	100	100	97

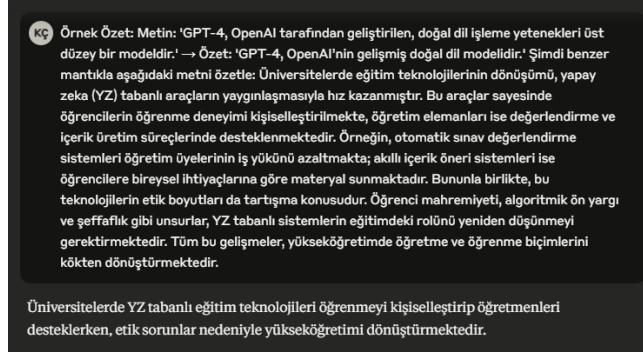
Tablo 8:Puanlama (özetleme kalitesi metriğinde "Claude Sonnet 4" modelinin prompt stratejilerine göre performansı)

a. Gözlemler (Claude Sonnet 4)



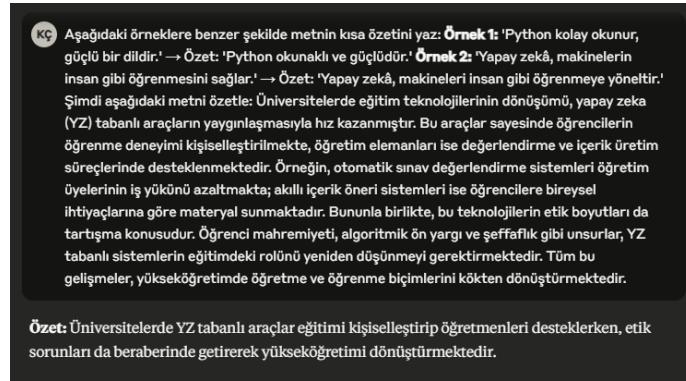
Görsel 37: "özetleme kalitesi" metriğinde "zero-shot" stratejisi kullanılarak girilen promptun "claude sonnet 4" modelindeki çıktısı

Claude Sonnet 4 modeli, **zero-shot** stratejisinde verilen metni, kısa ve net bir biçimde özetlemiştir. Yanıt, orijinal metnin ana fikrini, temel detayları ve olası riskleri dengeli bir şekilde yansımaktadır. Kullanılan prompt ve model yanıtının ekran görüntüsü yukarıda sunulmuştur.



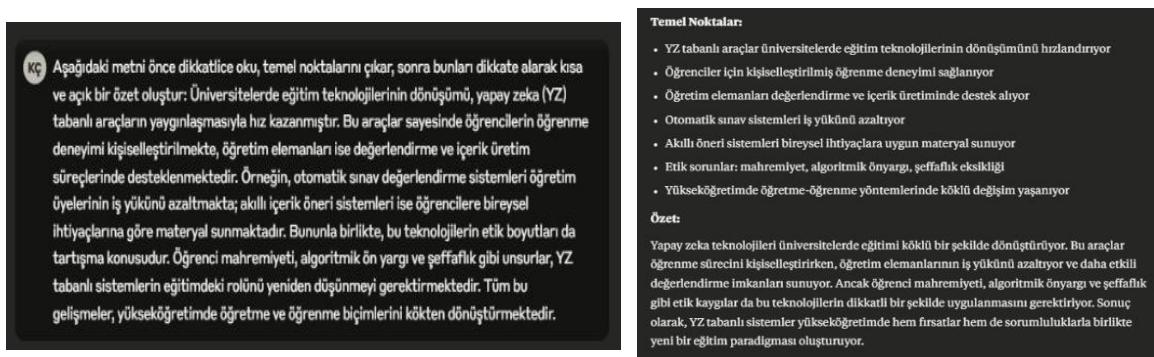
Görsel 38: "özetleme kalitesi" metriğinde "one-shot" stratejisi kullanılarak girilen promptun "claude sonnet 4" modelindeki çıktısı

Claude Sonnet 4 modeli, **one-shot** stratejisinde verilen örneğe benzer şekilde metni kısa bir özette ifade etmiş, üniversitelerde yapay zekâ tabanlı eğitim teknolojilerinin öğrenmeyi kişiselleştirdiğini ve öğretmenleri desteklediğini açıkça belirterek etik sorunların yükseköğretimde dönüştürüdügüne vurgu yapmıştır. Yanıt, örnek özete uygun olarak oldukça sade ve öz biçimde sunulmuş; metnin ana fikri ve temel mesajı korunmuş ancak detaylar minimumda tutulmuştur. Etik sorumlara yapılan vurgu, metnin özgün risk boyutunun da özet içinde yer bulmasını sağlamıştır. Kullanılan prompt ve model yanıtının ekran görüntüsü yukarıda sunulmuştur.



Görsel 39: "özetleme kalitesi" metriğinde "few-shot" stratejisi kullanılarak girilen promptun "claude sonnet 4" modelindeki çıktı

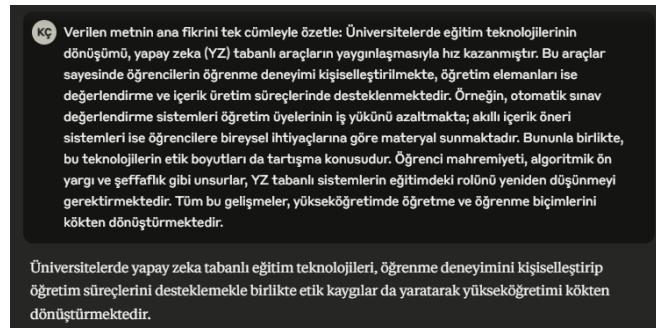
Claude Sonnet 4 modeli, **few-shot** stratejisinde verilen örneklere benzer şekilde metni kısa ve öz bir biçimde özetlemiştir. Modelin oluşturduğu özet, yapay zekâ tabanlı araçların eğitimi kişiselleştirip öğretmenleri desteklediğini ve beraberinde etik sorunlar getirerek yükseköğretimde dönüştürdüğü vurgulamaktadır. Yanıt, örneklerdeki gibi kısa ve net bir formatı koruyarak temel mesajı net şekilde aktarmıştır. Ancak metindeki ikincil detaylar ve bağlamsal ayrıntılar özet dışında bırakılmıştır. Model, örnek promptun yapısını iyi takip etmiş ve özetlemede gereksiz ayrıntıya yer vermemiştir. Kullanılan prompt ve model yanıtının ekran görüntüsü yukarıda sunulmuştur.



Görsel 40: "özetleme kalitesi" metriğinde "chain-of-thought" stratejisi kullanılarak girilen promptun "claude sonnet 4" modelindeki çıktı

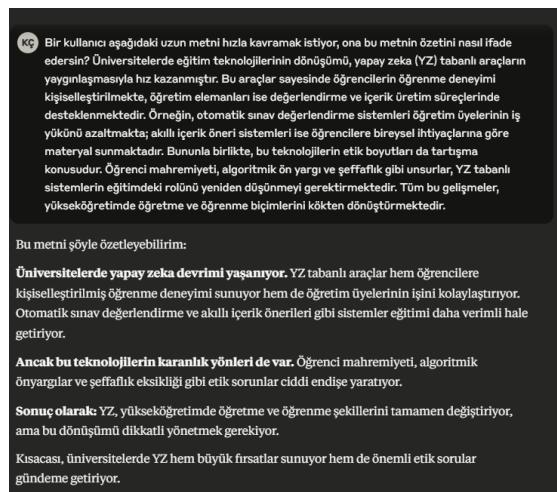
Claude Sonnet 4 modeli, **chain-of-thought** stratejisinde, metni önce madde madde temel noktalara ayırmış, ardından bu noktaları dikkate alarak kapsamlı ve bütünlüklü bir özet oluşturmuştur. Model hem teknolojik dönüşümün etkisini hem de kişiselleştirilmiş öğrenme, öğretim elemanlarının yükünün azalması ve değerlendirme sistemlerinin avantajlarını açıkça belirtmiştir. Özet, anlam bütünlüğünü korurken, metindeki ana temalar ve önemli detaylar

özetleme kalitesi. Kullanılan prompt ve model yanıtının ekran görüntüsü yukarıda sunulmuştur.



Görsel 41: "özetleme kalitesi" metriğinde "direct prompting" stratejisi kullanılarak girilen promptun "claude sonnet 4" modelindeki çıktısı

Claude Sonnet 4 modeli, **direct prompting** stratejisinde verilen metni, ana fikri tek cümlede özetleme talebine uygun olarak yanıtlamıştır. Cümle, orijinal metindeki ana mesajı ve riskleri özlü bir biçimde özetlemiştir; detaylar doğal olarak tek cümle formatı gereği dışında bırakılmıştır. Yanıt, doğrudan, kısa ve vurucu bir ana fikir sunmaktadır. Kullanılan prompt ve model yanıtının ekran görüntüsü yukarıda sunulmuştur.



Görsel 42: "özetleme kalitesi" metriğinde "indirect prompting" stratejisi kullanılarak girilen promptun "claude sonnet 4" modelindeki çıktısı

Claude Sonnet 4 modeli, **indirect prompting** stratejisinde metni bir kullanıcının hızlı kavrayabileceği bir şekilde başlıklandırmış ve özet cümlelerle çok katmanlı olarak özetlemiştir. Sonuç bölümünde ise, bu dönüşümün dikkatli yönetilmesi gerektiğine işaret ederek dengeli bir

özet sunmuştur. Anlam bütünlüğü, açıklık ve detay açısından model yüksek performans göstermiştir. Ancak, özetin uzunluğu ve içerdiği ayrıntı seviyesi, pratikte bir “özet”ten çok, metnin neredeyse yeniden yazılmış bir versiyonu gibi durmaktadır. Kullanıcıya hızlı kavrayış sağlasa da beklenen kısalık ve sadeleştirme tam olarak sağlanmamıştır. Kullanılan prompt ve model yanıtının ekran görüntüsü yukarıda sunulmuştur.

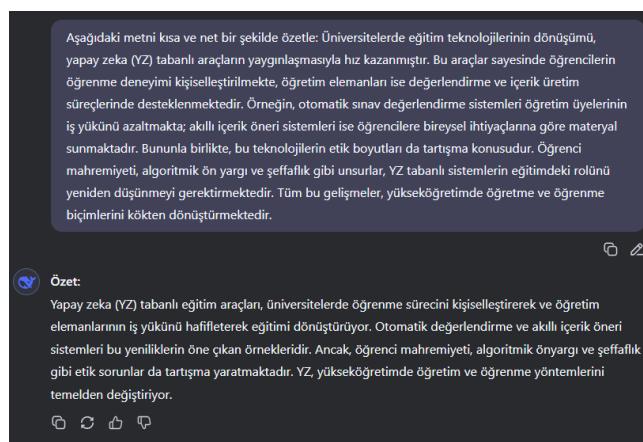
4. Model: DeepSeek V3

Aşağıdaki tabloda, özetleme kalitesi metriğinde tüm prompt stratejilerinin “DeepSeek V3” modelindeki performansı 0-100 puanlama sistemine göre değerlendirilmiştir.

Strateji	Doğruluk	Açıklık	Detay	Bağlama Uygunluk
zero-shot	100	100	95	100
one-shot	100	100	90	100
few-shot	100	100	90	100
chain-of-thought	100	100	95	92
direct prompting	100	100	85	100
indirect prompting	100	100	100	97

Tablo 9: Puanlama (özetleme kalitesi metriğinde “DeepSeek V3” modelinin prompt stratejilerine göre performansı)

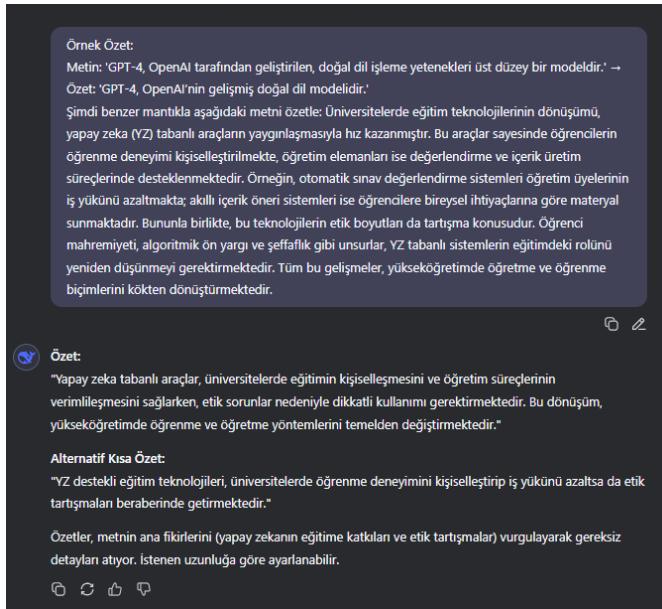
a. Gözlemler (DeepSeek V3)



Görsel 43: "özetleme kalitesi" metriğinde "zero-shot" stratejisi kullanılarak girilen promptun "deepseek v3" modelindeki çıktısı

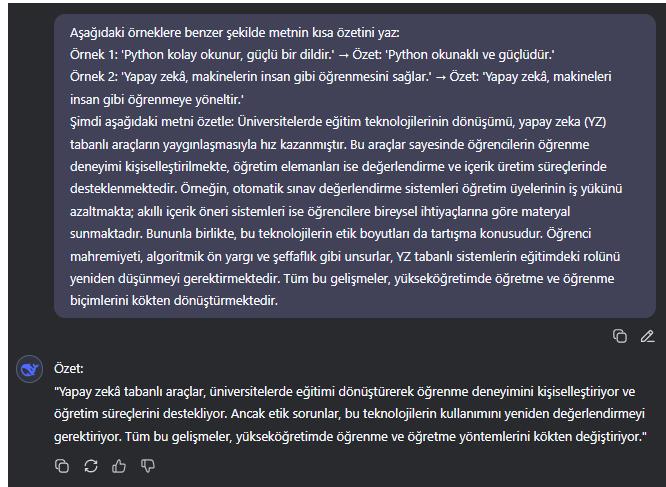
DeepSeek V3 modeli, **zero-shot** stratejisinde verilen metni kısa ve net bir şekilde özetlemiştir. Önemli detaylar özetin içinde yer almış, etik sorunlar da dengeli biçimde öne

çıkarılmıştır. Yanıt hem ana fikri hem de metindeki kritik ayrıntıları yeterli düzeyde yansımaktadır. Son cümlede ise, yapay zekanın yükseköğretimde temel bir dönüşüm yarattığı vurgulanmıştır. Kullanılan prompt ve model yanıtının ekran görüntüsü yukarıda sunulmuştur.



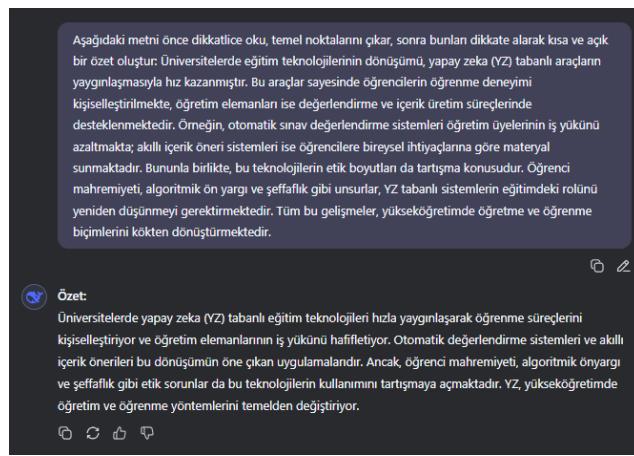
Görsel 44: "özetleme kalitesi" metriğinde "one-shot" stratejisi kullanılarak girilen promptun "deepseek v3" modelindeki çıktısı

DeepSeek V3 modeli, **one-shot** stratejisinde, örnek özet biçimini aynen taklit ederek hem kısa hem de nokta atışı bir özet üretmiştir. Modelin ikinci, alternatif özeti etik kaygılarına daha fazla vurgu yaparak, özetin odağını etik tartışmalara ve işlevsel faydalara eşit biçimde dağıtmıştır. Her iki özet de ana fikri ve temel mesajı doğru biçimde aktarmış, metnin uzunluğunu örnekteki gibi kısa tutmuştur. Yanıta, metnin temel bileşenleri ve riskleri sade bir dille ifade edilmiştir. Kullanılan prompt ve model yanıtının ekran görüntüsü yukarıda sunulmuştur.



Görsel 45: "özetleme kalitesi" metriğinde "few-shot" stratejisi kullanılarak girilen promptun "deepseek v3" modelindeki çıktısı

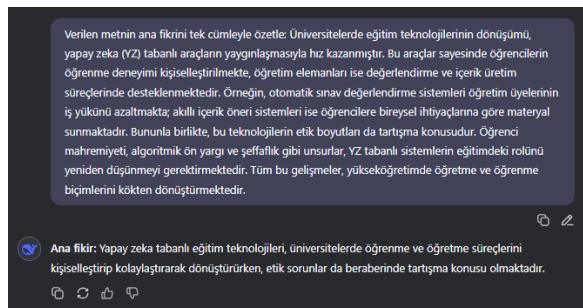
DeepSeek V3, **few-shot** stratejisinde promptta verilen kısa ve net örneklerinin biçimini başarılı şekilde taklit etmiştir. Model, uzun ve ayrıntılı metni, örneklerde olduğu gibi sade ve kısa bir özette sonuçlandırmış; gereksiz ayrıntılara girmeden, odaklanmış bir yanıt sunmuştur. Strateji gereği, yanıtın detay seviyesi düşük tutulmuş ve özettir, biçimsel olarak few-shot yaklaşımının amacına uygundur. Model, örnek formatı taklit etme konusunda başarılıdır; fakat bu durum önemli bazı alt başlıkların göz ardı edilmesine yol açabilir. Kullanılan prompt ve model yanıtının ekran görüntüsü yukarıda sunulmuştur.



Görsel 46: "özetleme kalitesi" metriğinde "chain-of-thought" stratejisi kullanılarak girilen promptun "deepseek v3" modelindeki çıktısı

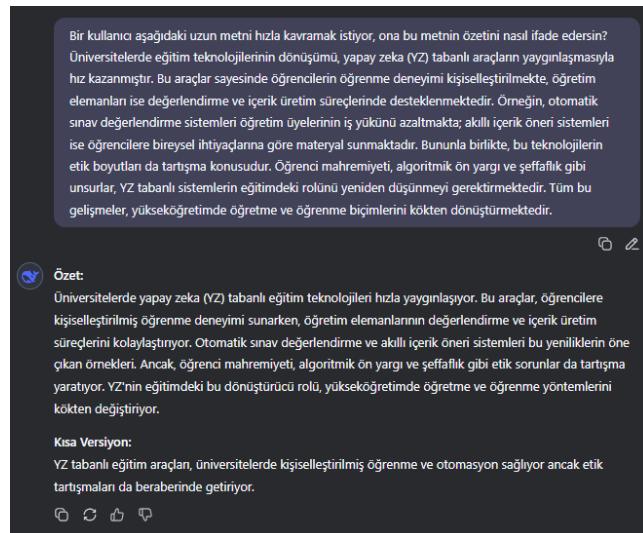
DeepSeek V3, **chain-of-thought** stratejisinde öncelikle metni parçalara ayırma ve ardından kısa, net bir özet oluşturma yönlendirmesine büyük oranda uyum sağlamıştır. Model, temel noktaları öne çıkarıp, özettini kademeli bir şekilde yapılmıştır. Ancak, modelin çıktısı

beklenen kadar adım adım bir gerekçelendirme veya maddeleme yerine daha çok klasik kısa özet formatına yakın kalmıştır. Yani, chain-of-thought stratejisinin ayırtıcı özelliği olan “mantıksal adımların görünürlüğü” tam anlamıyla ortaya çıkmamış; model promptun bu özelliğini yüzeysel şekilde uygulamıştır. Kullanılan prompt ve model yanıtının ekran görüntüsü yukarıda sunulmuştur.



Görsel 47: "özetleme kalitesi" metriğinde "direct prompting" stratejisi kullanılarak girilen promptun "deepseek v3" modelindeki çıktısı

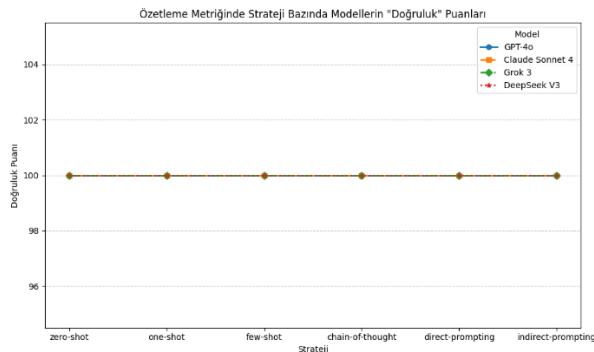
DeepSeek V3, **direct prompting** stratejisinde metnin ana fikrini tek cümlede özetleme talebine biçimsel olarak eksiksiz uyum sağlamıştır. Model, ana mesajı kısa, net ve doğrudan bir şekilde aktarmış; gereksiz detay ya da yan bilgilere yer vermemiştir. Strateji gereği, özetin uzunluğu ve kapsamı oldukça sınırlı tutulmuş, ana tema ön plana çıkarılmıştır. Modelin çıktısı, tek cümlede özet bekłentisine tam uyumluluk göstermiştir. Kullanılan prompt ve model yanıtının ekran görüntüsü yukarıda sunulmuştur.



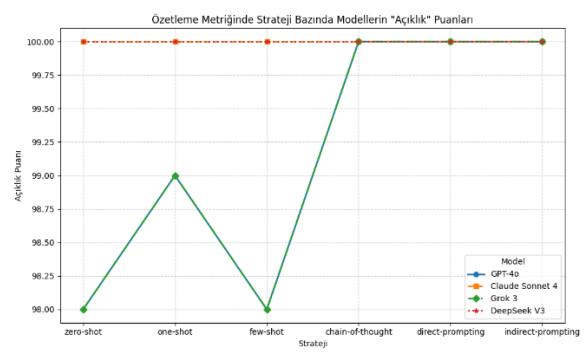
Görsel 48: "özetleme kalitesi" metriğinde "indirect prompting" stratejisi kullanılarak girilen promptun "deepseek v3" modelindeki çıktısı

DeepSeek V3, **indirect prompting** stratejisinde kullanıcıya hem ayrıntılı hem de özetlenmiş kısa bir versiyon sunarak esnek bir yaklaşım sergilemiştir. Model, önce klasik kısa özet stratejilerine yakın, nispeten uzun ve ayrıntılı bir özet üretmiş; ardından, istenirse daha kısa bir versiyon da oluşturabileceğini göstererek “kısa versiyon” başlığı altında daha özlü bir yanıt sunmuştur. Bu yaklaşım, promptun “kullanıcı odaklı” ve “hızlı kavrama” bekłentisine kısmen cevap verirken, ilk özetin uzunluğu nedeniyle bağlama uygunluk biraz sınırlı kalmıştır. Model, özetleme biçimini farklı kullanıcı ihtiyaçlarına göre uyarlama konusunda güçlü bir performans göstermiştir. Kullanılan prompt ve model yanıtının ekran görüntüsü yukarıda sunulmuştur.

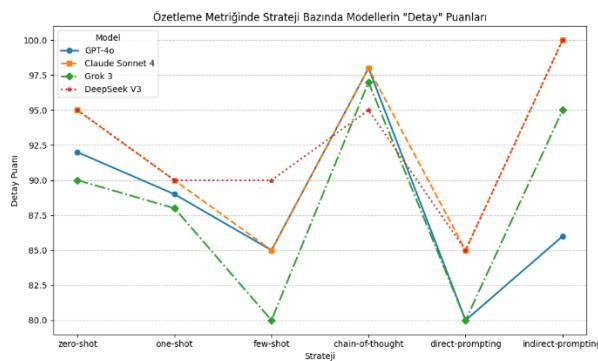
5. Öztleme Metriği Sonuçlarının Strateji ve Model Bazında Karşılaştırılması



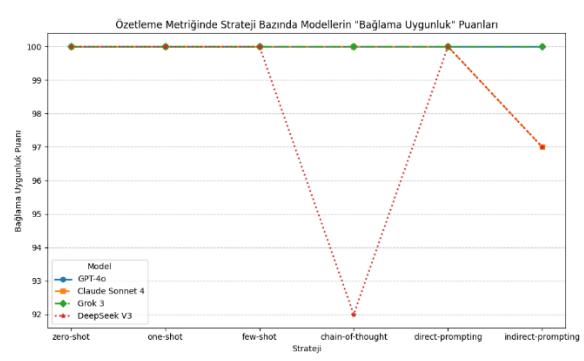
Grafik 5: Öztleme metriğinde strateji bazında modellerin doğruluk puanları



Grafik 6: Öztleme metriğinde strateji bazında modellerin açıklık puanları



Grafik 7: Öztleme metriğinde strateji bazında modellerin detay puanları



Grafik 8: Öztleme metriğinde strateji bazında modellerin bağlama uygunluk puanları

Grafik 5'te, öztleme metriği kapsamında tüm modellerin her bir stratejide en yüksek doğruluk puanına ulaştığı görülmektedir. Farklı prompt stratejilerinin hiçbirinin model doğruluğu üzerinde anlamlı bir değişiklik yaratmadığı; tüm modellerin tüm stratejilerde tam puan aldığı gözlemlenmiştir. Bu bulgu, öztleme görevinde doğruluk açısından mevcut modellerin ve kullanılan stratejilerin oldukça yüksek ve istikrarlı bir performans gösterdiğine işaret etmektedir.

Grafik 6'da öztleme metriği kapsamında modellerin açıklık puanları karşılaştırılmıştır. Tüm stratejilerde Claude Sonnet 4 ve DeepSeek V3 modelleri en yüksek açıklık puanına ulaşırken, Grok 3 ve GPT-4o modelinin açıklık puanları tüm stratejilerde yüksek olmakla birlikte bazı stratejilerde çok küçük farklılıklar göstermektedir. Genel olarak bakıldığından, açıklık kriterinde modeller arasında ciddi bir performans farkı bulunmamakta, tüm modellerde açıklık puanları üst düzeyde seyretmektedir. Bu bulgu, öztleme görevinde açıklık açısından mevcut modellerin ve stratejilerin tutarlı ve yüksek bir başarıya sahip olduğunu göstermektedir.

Grafik 7'de, özetleme metriği kapsamında modellerin detay puanlarının stratejilere göre değişim gösterdiği gözlemlenmektedir. Bazı stratejilerde modellerin detay puanlarında azalma yaşanırken, özellikle chain-of-thought ve indirect prompting stratejilerinde puanların yükseldiği dikkat çekmektedir. Claude Sonnet 4 ve DeepSeek V3 diğer modellere nazaran genellikle yüksek ve tutarlı bir performans sergilerken, diğer modellerde stratejilere bağlı olarak detay puanlarında dalgalanmalar görülmektedir. Bu sonuçlar, detay kriterinde strateji seçiminin model performansında belirleyici olabileceğini ve bazı stratejilerin modellerin detay başarısını artırdığını göstermektedir.

Grafik 8'de, özetleme metriği kapsamında modellerin bağlama uygunluk puanlarının büyük ölçüde yüksek ve istikrarlı olduğu görülmektedir. Tüm stratejilerde GPT-4o ve Grok 3 modelleri üst düzey performans sergilerken, Claude Sonnet 4 ve DeepSeek V3 modellerinde yalnızca belirli stratejilerde hafif bir puan düşüşü gözlenmektedir. Özellikle DeepSeek V3 modelinde chain-of-thought stratejisinde belirgin bir azalma gelmiş, diğer stratejilerde ise yüksek düzeyde puanlar korunmuştur. Genel olarak, bağlama uygunluk açısından modellerin ve stratejilerin büyük çoğunlukla yüksek başarı gösterdiği, sadece tekil durumlarda küçük farklılıkların ortaya çıktığı söylenebilir.

C. Metrik: Doğruluk Kalitesi

Görev: Belirli prompt stratejilerine göre girilen promptlara olgusal olarak doğru, tutarlı ve güncel yanıtlar üretmek.

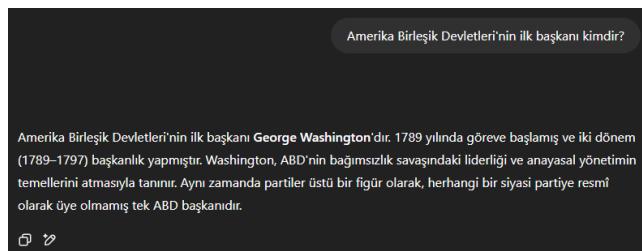
1. Model: GPT-4o

Aşağıdaki tabloda, doğruluk kalitesi metriğinde tüm prompt stratejilerinin “GPT-4o” modelindeki performansı 0-100 puanlama sistemine göre değerlendirilmiştir.

Strateji	Doğruluk	Açıklık	Detay	Bağlama Uygunluk
zero-shot	100	100	100	98
one-shot	100	100	85	100
few-shot	100	100	85	100
chain-of-thought	100	100	100	99
direct prompting	100	100	95	100
indirect prompting	100	100	98	100

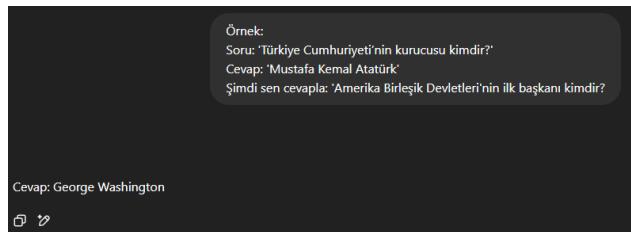
Tablo 10: Puanlama (doğruluk kalitesi metriğinde “GPT-4o” modelinin prompt stratejilerine göre performansı)

a. Gözlemler (GPT-4o)



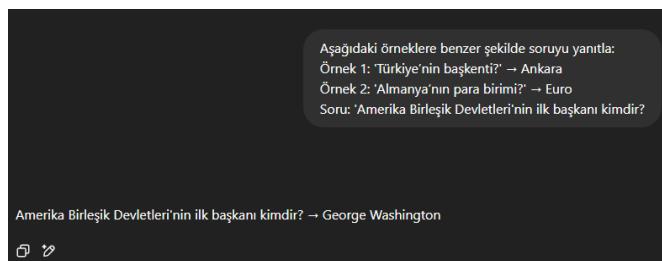
Görsel 49: "doğruluk kalitesi" metriğinde "zero-shot" stratejisi kullanılarak girilen promptun "gpt-4o" modelindeki çıktıtı

GPT-4o, **zero-shot** stratejisinde bilgi temelli soruya herhangi bir ek örnek veya bağlam verilmeden, doğrudan yanıt üretmiştir. Model, doğru cevabı eksiksiz ve vurgulu şekilde sunmuş; ek olarak tarih, görev süresi ve partilerle ilgili ek bilgiler vererek yanıtını detaylandırmıştır. Yanıt, zero-shot yaklaşımının amacına uygun olarak, yalnızca temel talimata dayanarak yüksek doğruluk ve yeterli açıklıkla oluşturulmuştur. Model, herhangi bir yönlendirme olmadan güvenilir bilgi üretebilmiştir. Kullanılan prompt ve model yanıtının ekran görüntüsü yukarıda sunulmuştur.



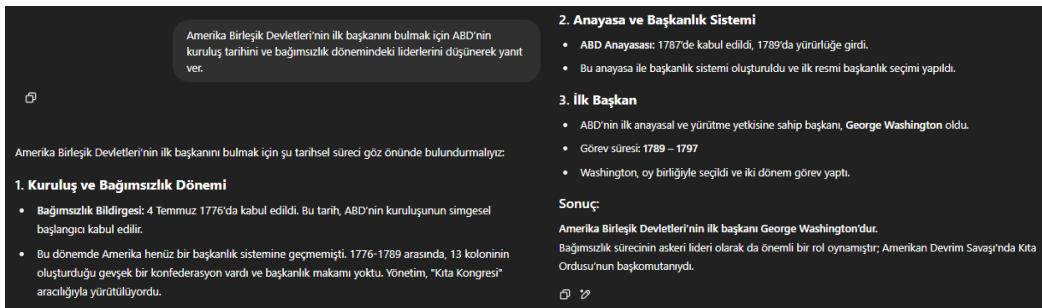
Görsel 50: "doğruluk kalitesi" metriğinde "one-shot" stratejisi kullanılarak girilen promptun "gpt-4o" modelindeki çıktısı

GPT-4o, **one-shot** stratejisinde modelin ne şekilde yanıt vermesi gerektiğini gösteren tek bir örnek sunulmasının ardından bilgi sorusuna cevap vermiştir. Model, örnekteki biçimini doğrudan taklit ederek kısa ve net bir yanıt üretmiştir. Yanıt doğruluğu yüksek olup, örnekteki kısa ve vurucu format stratejiye uygun şekilde yansımıştır., Model, promptun örnek yönlendirmesini başarıyla uygulamıştır. Kullanılan prompt ve model yanıtının ekran görüntüsü yukarıda sunulmuştur.



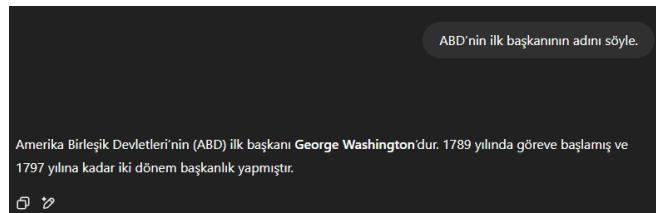
Görsel 51: "doğruluk kalitesi" metriğinde "few-shot" stratejisi kullanılarak girilen promptun "gpt-4o" modelindeki çıktısı

GPT-4o, **few-shot** stratejisinde verilen birden fazla örnek formatını doğrudan taklit ederek, modelin yanıt stilini biçimsel olarak net bir şekilde öğrenmesini sağlamıştır. Model, kendisine sunulan örneklerde olduğu gibi soruyu ve cevabı kısa, doğrudan ve ok işaretiyile eşleştirerek yanıtlamıştır. Yanıttı sadece temel bilgiye yer verilmiş, ek açıklama veya bağlamsal detay sunulmamıştır. Stratejiye uygun olarak, modelin örnekleri başarıyla analiz edip aynı yapıda çıktılar üretme kabiliyeti yüksektir. Kullanılan prompt ve model yanıtının ekran görüntüsü yukarıda sunulmuştur.



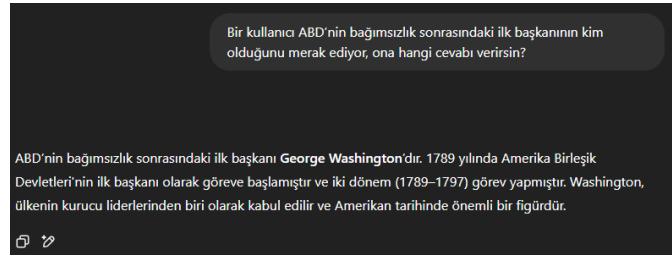
Görsel 52: "doğruluk kalitesi" metriğinde "chain-of-thought" stratejisi kullanılarak girilen promptun "gpt-4o" modelindeki çıktısı

GPT-4o, **chain-of-thought** stratejisinde bilgi temelli soruyu adım adım gerekçelendirerek yanıtlamıştır. Model, ana tarihsel süreçleri ayrı başlıklar ve maddeler halinde açıklamış, sürecin mantıksal akışını ayrıntılı biçimde sunmuştur. Sonuç kısmında ise yanıt netleştirerek özetlemiştir. Model, promptun “düşünce zinciri” gereksinimini başarıyla uygulamış, süreçteki kilit tarihi ve olgusal noktaları tutarlı bir şekilde aktarmıştır. Bu yapı, modelin gerekçelendirilmiş, kapsamlı ve didaktik yanıt oluşturma becerisini ortaya koymuştur. Kullanılan prompt ve model yanıtının ekran görüntüsü yukarıda sunulmuştur.



Görsel 53: "doğruluk kalitesi" metriğinde "direct prompting" stratejisi kullanılarak girilen promptun "gpt-4o" modelindeki çıktısı

GPT-4o, **direct prompting** stratejisinde bilgi temelli soruya açık ve kısa bir şekilde, doğrudan doğruyu yanıt vermiştir. Model, yalnızca istenen temel bilgiyi vermekle kalmamış, ayrıca görev başlangıç ve bitiş tarihleriyle birlikte kısa bir ek açıklama da sunmuştur. Yanıt, promptun doğrudan bilgi beklenışını karşılamış, gereksiz ayrıntı veya fazladan gerekçelendirmeden kaçınılmıştır. Model, direct prompting stratejisinin amacına uygun olarak en hızlı ve net şekilde doğru cevabı üretmiştir. Kullanılan prompt ve model yanıtının ekran görüntüsü yukarıda sunulmuştur.



Görsel 54: "doğruluk kalitesi" metriğinde "indirect prompting" stratejisi kullanılarak girilen promptun "gpt-4o" modelindeki çıktısı

GPT-4o, **indirect prompting** stratejisinde kullanıcının bağlamı gereği bilgiye ulaşmak istediği varsayımyla soruyu açıklayıcı ve pedagojik bir üslupla yanıtlamıştır. Model, yalnızca doğru cevabı vermekle kalmamış, aynı zamanda ek açıklamalar sunmuştur. Yanıt hem bilgilendirici hem de kullanıcının merakını tatmin edecek şekilde yapılandırılmıştır. Stratejiye uygun olarak, modelin açıklama ve gerekçelendirme düzeyi yüksektir. Kullanılan prompt ve model yanıtının ekran görüntüsü yukarıda sunulmuştur.

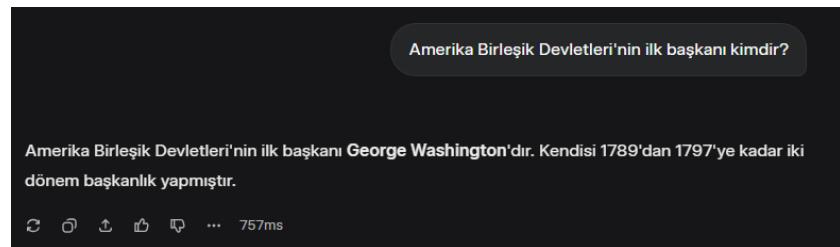
2. Model: Grok 3

Aşağıdaki tabloda, doğruluk kalitesi metriğinde tüm prompt stratejilerinin “Grok 3” modelindeki performansı 0-100 puanlama sisteme göre değerlendirilmiştir.

Strateji	Doğruluk	Açıklık	Detay	Bağlama Uygunluk
zero-shot	100	100	95	100
one-shot	100	100	80	100
few-shot	100	100	80	100
chain-of-thought	100	100	99	100
direct prompting	100	100	85	100
indirect prompting	100	100	90	95

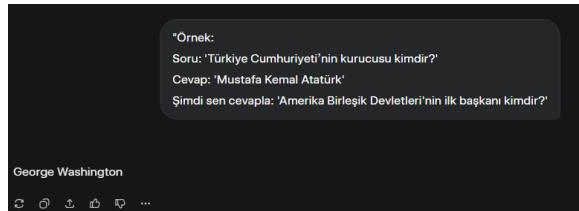
Tablo 11: Puanlama (doğruluk kalitesi metriğinde “Grok 3” modelinin prompt stratejilerine göre performansı)

a. Gözlemler (Grok 3)



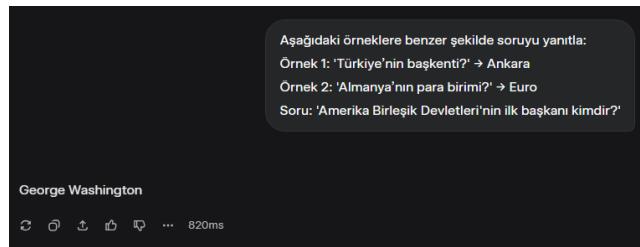
Görsel 55: "doğruluk kalitesi" metriğinde "zero-shot" stratejisi kullanılarak girilen promptun "grok 3" modelindeki çıktısı

Grok 3, **zero-shot** stratejisinde bilgi temelli soruya doğrudan ve örneksiz şekilde yanıt vermiştir. Model, doğru cevabı eksiksiz sunmuş, ayrıca başkanlık başlangıç ve bitiş tarihlerini belirtmiştir. Yanında, temel soruya odaklanılmış; gereksiz ayrıntıdan kaçınılmıştır. Zero-shot yaklaşımında model, minimum bağlamla yüksek doğruluk ve açıklıkta çıktı üretебilmiştir. Kullanılan prompt ve model yanıtının ekran görüntüsü yukarıda sunulmuştur.



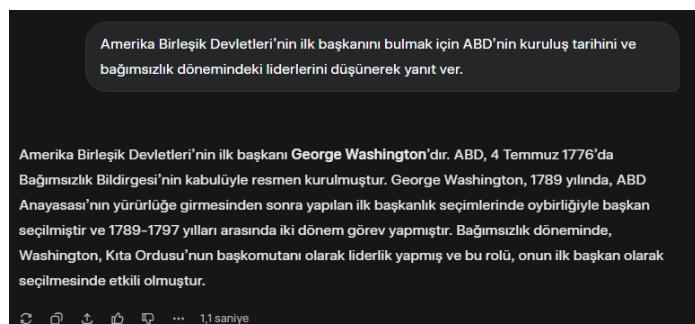
Görsel 56: "doğruluk kalitesi" metriğinde "one-shot" stratejisi kullanılarak girilen promptun "grok 3" modelindeki çıktısı

Grok 3, **one-shot** stratejisinde yalnızca tek bir örnek verilerek sorunun formatı gösterilmiştir. Model, örnekteki biçimini doğrudan taklit ederek sadece gerekli olan temel bilgiyi tek kelimeyle sunmuştur. Yanıt minimum açıklama ve ayrıntı içermemektedir; ancak one-shot formatının beklediği olan “örneğe sadık kalma” gereksinimini eksiksiz yerine getirmiştir. Model, yönlendirme olmaksızın kısa ve kesin bir bilgi ile yanıt üretmiştir. Kullanılan prompt ve model yanıtının ekran görüntüsü yukarıda sunulmuştur.



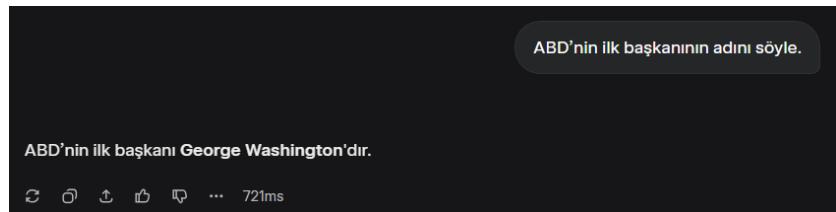
Görsel 57: "doğruluk kalitesi" metriğinde "few-shot" stratejisi kullanılarak girilen promptun "grok 3" modelindeki çıktıtı

Grok 3, **few-shot** stratejisinde birden fazla örnek üzerinden verilen formatı eksiksiz biçimde taklit ederek yalnızca temel bilgiyi, kısa ve doğrudan sunmuştur. Yanıt, örneklerde olduğu gibi sade ve net biçimde olarak verilmiştir. Model, ayrıntı eklemek yerine formatı sadakatle uygulamış ve gereksiz bilgi sunmamıştır. Few-shot stratejisinin biçimsel bekłentisi olan “örnekler uyum” yüksek düzeyde sağlanmıştır. Kullanılan prompt ve model yanıtının ekran görüntüsü yukarıda sunulmuştur.



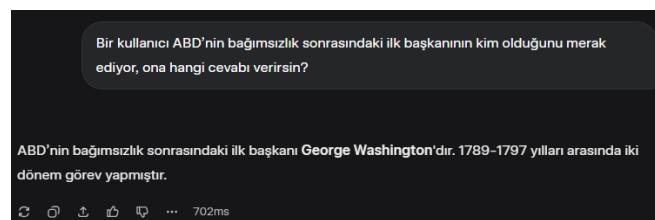
Görsel 58: "doğruluk kalitesi" metriğinde "chain-of-thought" stratejisi kullanılarak girilen promptun "grok 3" modelindeki çıktıtı

Grok 3, **chain-of-thought** stratejisinde soruya mantıksal adımlar ve tarihsel gerekçeler eşliğinde yanıt vermiştir. Yanıt, adım adım gerekçelendirilmiş ve bilginin arka planı özenle aktarılmıştır. Model, chain-of-thought yaklaşımının yapılandırılmış analiz ve gerekçelendirme bekłentisini başarıyla uygulamıştır. Kullanılan prompt ve model yanıtının ekran görüntüsü yukarıda sunulmuştur.



Görsel 59: "doğruluk kalitesi" metriğinde "direct prompting" stratejisi kullanılarak girilen promptun "grok 3" modelindeki çıktısı

Grok 3, **direct prompting** stratejisinde verilen açık ve doğrudan komutu minimum ek bilgiyle yanıtlamıştır. Model, yalnızca temel cevabı kısa ve net şekilde sunmuş; ek açıklama ya da gerekçelendirmeden kaçınmıştır. Bu yaklaşım, direct prompting stratejisinin hızlı ve sade yanıt üretme beklentisine tam uyumluluk göstermektedir. Kullanılan prompt ve model yanıtının ekran görüntüsü yukarıda sunulmuştur.



Görsel 60: "doğruluk kalitesi" metriğinde "indirect prompting" stratejisi kullanılarak girilen promptun "grok 3" modelindeki çıktısı

Grok 3, indirect prompting stratejisinde kullanıcının bilgi ihtiyacına odaklanarak cevabını hafif açıklamalı ve rehber bir tonla sunmuştur. Model, temel bilgiyi doğrudan verdikten sonra kısa ek detay da eklemiş; ancak yanıtın genel yapısı, doğrudan açıklayıcı ve sade kalmıştır. Indirect prompting stratejisinin amaçladığı rehberlik ve kullanıcıya göre biçimlendirilmiş yanıt üretme kabiliyeti kısmen sağlanmış, ancak ekstra pedagojik açıklama veya bağlamsal genişleme sınırlı düzeyde kalmıştır. Kullanılan prompt ve model yanıtının ekran görüntüsü yukarıda sunulmuştur.

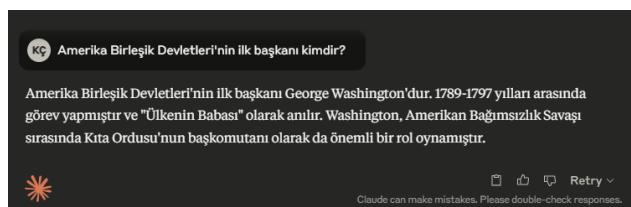
3. Model: Claude Sonnet 4

Aşağıdaki tabloda, doğruluk kalitesi metriğinde tüm prompt stratejilerinin “Claude Sonnet 4” modelindeki performansı 0-100 puanlama sistemine göre değerlendirilmiştir.

Strateji	Doğruluk	Açıklık	Detay	Bağlama Uygunluk
zero-shot	100	100	97	100
one-shot	100	100	80	100
few-shot	100	100	80	100
chain-of-thought	100	100	100	99
direct prompting	100	100	85	100
indirect prompting	100	100	97	99

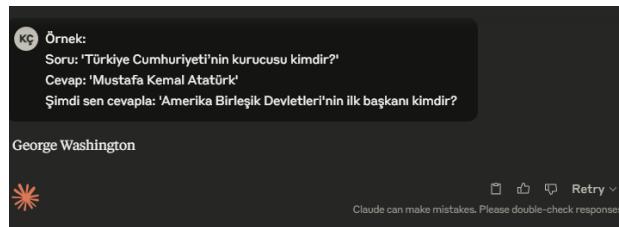
Tablo 12 Puanlama (doğruluk kalitesi metriğinde “Claude Sonnet 4” modelinin prompt stratejilerine göre performansı)

a. Gözlemler (Claude Sonnet 4)



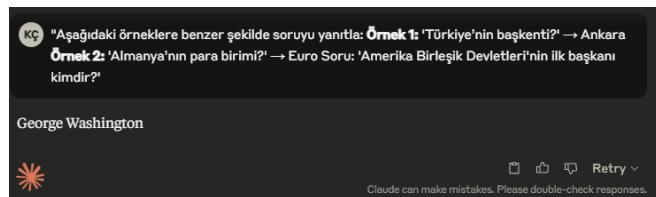
Görsel 61: "doğruluk kalitesi" metriğinde "zero-shot" stratejisi kullanılarak girilen promptun "claude sonnet 4" modelindeki çıktısı

Claude Sonnet 4, **zero-shot** stratejisinde örnek ya da ek bağlam verilmeden doğrudan bilgi sorusuna yanıt vermiştir. Model, temel cevabı açıkça sunmuş ve kısa ek açıklamalarla yanıtı detaylandırmıştır. Yanıt, zero-shot stratejisinin biçimsel gerekliliklerine uygun olarak, talimat dışı ek açıklama eklemeden, sade ve doğrudan bilgi üretmiştir. Model, örneksiz ortamda yüksek doğrulukta ve açıklıkta yanıt verebilmiştir. Kullanılan prompt ve model yanıtının ekran görüntüsü yukarıda sunulmuştur.



Görsel 62: "doğruluk kalitesi" metriğinde "one-shot" stratejisi kullanılarak girilen promptun "claude sonnet 4" modelindeki çıktısı

Claude Sonnet 4, **one-shot** stratejisinde modele örnek gösterilerek yanıt formatı öğretilmiş ve ardından bilgi temelli soru yöneltilmiştir. Model, örnekteki kısa ve doğrudan biçimini aynen taklit ederek yalnızca temel cevabı sunmuştur. Ek açıklama, detay veya gerekçelendirme olmadan, örneğin yapısına tam sadakat gösterilmiştir. Bu, one-shot stratejisinin beklenen “örneği taklit et” performansına uygun, minimalist bir çıktıdır. Kullanılan prompt ve model yanıtının ekran görüntüsü yukarıda sunulmuştur.



Görsel 63: "doğruluk kalitesi" metriğinde "few-shot" stratejisi kullanılarak girilen promptun "claude sonnet 4" modelindeki çıktısı

Claude Sonnet 4, **few-shot** stratejisinde birden fazla örneğin biçimini doğrudan taklit etmiş ve yalnızca temel cevabı kısa ve net biçimde sunmuştur. Model, örneklerdeki minimalist ve doğrudan yanıt formatını başarıyla uygulayarak gereksiz açıklama veya ek detay eklememiştir. Few-shot stratejisinin biçimsel hedefi olan “örneklerle uyumlu ve sade yanıt” üretme yeteneği yüksek düzeyde sergilenmiştir. Kullanılan prompt ve model yanıtının ekran görüntüsü yukarıda sunulmuştur.



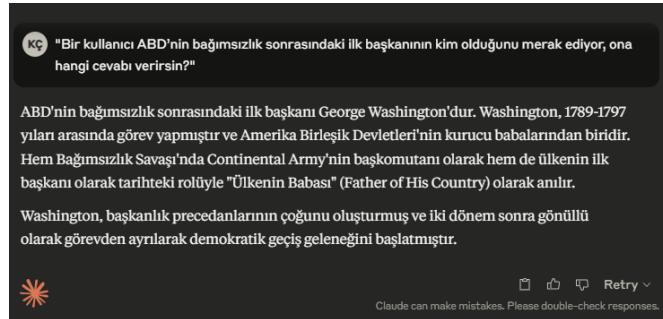
Görsel 64: "doğruluk kalitesi" metriğinde "chain-of-thought" stratejisi kullanılarak girilen promptun "claude sonnet 4" modelindeki çıktıtı

Claude Sonnet 4, **chain-of-thought** stratejisinde soruya mantıksal ve kronolojik adımlar halinde gerekçeli bir açıklama üretmiştir. Model, yanıtını tarihsel dönem noktaları, anayasal süreç ve liderlik bağlamında parçalara bölgerek her bir aşamayı detaylandırmıştır. Bu yapı, chain-of-thought stratejisinin öngördüğü “adım adım gerekçelendirme ve bilgiye ulaşma” performansını oldukça başarılı biçimde yansımaktadır. Yanıtın uzunluğu ve yapılandırılmış içeriği, bu stratejiyle üretilen çıktıların klasik kısa yanıtlardan farkını belirgin şekilde göstermektedir. Kullanılan prompt ve model yanıtının ekran görüntüsü yukarıda sunulmuştur.



Görsel 65: "doğruluk kalitesi" metriğinde "direct prompting" stratejisi kullanılarak girilen promptun "claude sonnet 4" modelindeki çıktıtı

Claude Sonnet 4, **direct prompting** stratejisinde modele verilen kısa ve doğrudan talimatı minimum ek bilgiyle yanıtlamıştır. Model, yalnızca istenen temel cevabı sunmuş ve gereksiz detaydan kaçınmıştır. Ek olarak başkanlık dönemini belirten kısa bir bilgi de eklemiştir, ancak yanıt genel olarak sade ve doğrudandır. Bu performans, direct prompting stratejisinin “kısa, açık, doğrudan bilgi üretimi” bekłentisine tamamen uygundur. Kullanılan prompt ve model yanıtının ekran görüntüsü yukarıda sunulmuştur.



Görsel 66: "doğruluk kalitesi" metriğinde "indirect prompting" stratejisi kullanılarak girilen promptun "claude sonnet 4" modelindeki çıktısı

Claude Sonnet 4, **indirect prompting** stratejisinde kullanıcının merakına rehberlik edecek açıklayıcı ve zenginleştirilmiş bir yanıt üretmiştir. Model, temel bilgiyle birlikte tarihsel, unvan ve demokratik gelenek gibi ek açıklamalar sunmuş; pedagojik ve detaylı bir üslup benimsemiştir. Yanıt, indirect prompting stratejisinin açıklama ve rehberlik odaklı performansını güçlü biçimde yansımış; kullanıcıya konu hakkında derinlemesine bilgi sağlamıştır.

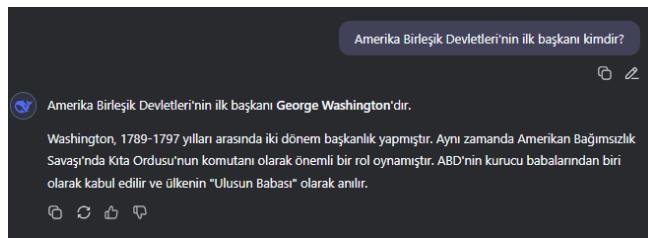
4. Model: DeepSeek V3

Aşağıdaki tabloda, doğruluk kalitesi metriğinde tüm prompt stratejilerinin “DeepSeek V3” modelindeki performansı 0-100 puanlama sisteme göre değerlendirilmiştir.

Strateji	Doğruluk	Açıklık	Detay	Bağlama Uygunluk
zero-shot	100	100	96	98
one-shot	100	100	80	98
few-shot	100	100	80	98
chain-of-thought	100	100	99	97
direct prompting	100	100	85	97
indirect prompting	100	98	90	96

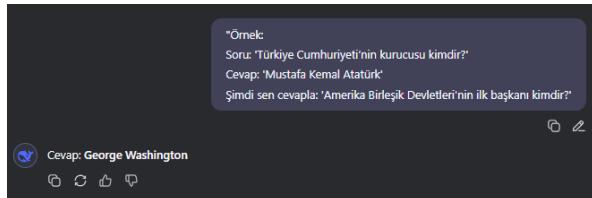
Tablo 13: Puanlama (doğruluk kalitesi metriğinde “DeepSeek V3” modelinin prompt stratejilerine göre performansı)

a. Gözlemler (DeepSeek V3)



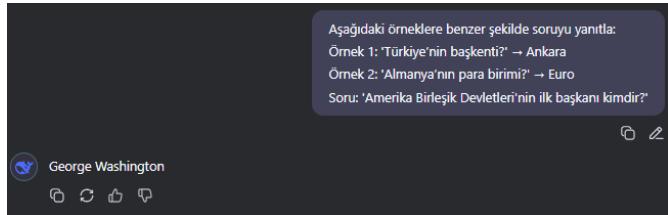
Görsel 67: "doğruluk kalitesi" metriğinde "zero-shot" stratejisi kullanılarak girilen promptun "deepseek v3" modelindeki çıktısı

DeepSeek V3, **zero-shot** stratejisinde örnek veya bağlam olmadan doğrudan bilgi sorusuna yanıt üretmiştir. Model, temel cevabı eksiksiz sunmuş, ayrıca görev dönemi ve tarihsel rol gibi ek bilgiler eklemiştir. Yanıt net, anlaşılır ve tutarlı olmakla birlikte; yanıtın dilinde sıradanlık ve benzer sistemlerde görülen tipik açıklama formatı hâkimdir. Model, özgünlük ve anlatımında fark yaratmamış, yalnızca gereğini yapmıştır. Kullanılan prompt ve model yanıtının ekran görüntüsü yukarıda sunulmuştur.



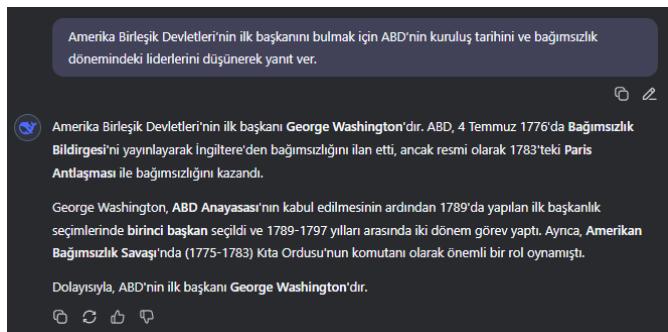
Görsel 68: "doğruluk kalitesi" metriğinde "one-shot" stratejisi kullanılarak girilen promptun "deepseek v3" modelindeki çıktısı

DeepSeek V3, **one-shot** stratejisinde modele verilen örneği tam olarak taklit ederek kısa ve biçimsel olarak örnekle birebir uyumlu bir yanıt üretmiştir. Model, yalnızca temel cevabı sunmuş; ek açıklama veya gerekçelendirmeden kaçınmıştır. Yanıtın özgünlük seviyesi düşük, biçimsel sadakati yüksektir. Stratejinin "örneğe sadakat" hedefine tam uyum gösterse de derinlik ve detay açısından oldukça yüzeysel kalmıştır. Kullanılan prompt ve model yanıtının ekran görüntüsü yukarıda sunulmuştur.



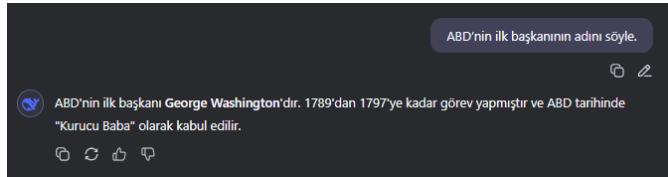
Görsel 69: "doğruluk kalitesi" metriğinde "few-shot" stratejisi kullanılarak girilen promptun "deepseek v3" modelindeki çıktısı

DeepSeek V3, **few-shot** stratejisinde modele sunulan birden fazla örneğin biçimini aynen taklit etmiş ve yalnızca temel bilgiyi kısa ve net şekilde sunmuştur. Model, ek açıklama ya da gerekçelendirme eklemeden, sadece örnek formatına sadık kalarak yanıt vermiştir. Strateji gereği, örneklerle uyumluluk yüksek, ancak yanıtın özgünlük ve detay seviyesi son derece sınırlı kalmıştır. Kullanılan prompt ve model yanıtının ekran görüntüsü yukarıda sunulmuştur.



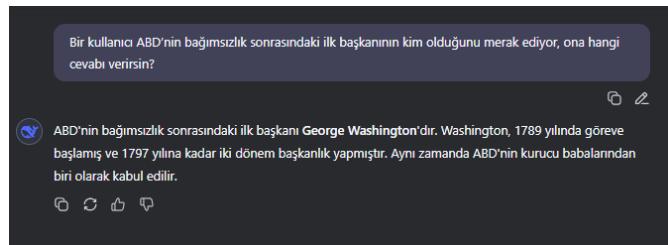
Görsel 70: "doğruluk kalitesi" metriğinde "chain-of-thought" stratejisi kullanılarak girilen promptun "deepseek v3" modelindeki çıktısı

DeepSeek V3, **chain-of-thought** stratejisinde modeli yönlendiren talimatlara uygun olarak cevabını mantıksal ve tarihsel adımlar halinde gerekçelendirerek oluşturmuştur. Bu yapı, chain-of-thought stratejisinin gerektirdiği kademeli gerekçelendirme ve açıklama performansına büyük ölçüde uygundur. Ancak, yanıt çok uzun, yoğun bilgi içeriyor ve kısmen konu dışı detaylara da (örneğin Paris Antlaşması gibi) girmiş; bu da okunabilirliği biraz azaltmış ve odaktan hafifçe sapmaya neden olmuştur. Kullanılan prompt ve model yanıtının ekran görüntüsü yukarıda sunulmuştur.



Görsel 71: "doğruluk kalitesi" metriğinde "direct prompting" stratejisi kullanılarak girilen promptun "deepseek v3" modelindeki çıktıtı

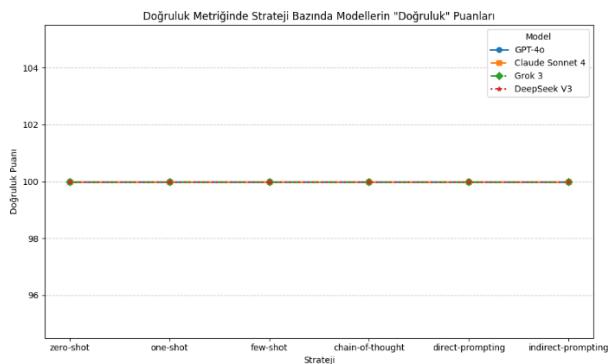
DeepSeek V3, **direct prompting** stratejisinde verilen doğrudan ve açık komutu, kısa ve temel cevaba odaklanarak yanıtlamıştır. Model, yalnızca temel bilgiyi sunmakla kalmamış, kısa bir ek bilgi (görev süresi, "Kurucu Baba" ifadesi) eklemiştir. Yanıt açık ve kolay anlaşılır olmakla birlikte, minimal ek açıklamalar dışında herhangi bir gerekçelendirme veya farklılaştırıcı unsur sunmamaktadır. Yani, direct prompting stratejisinin "doğrudan ve hızlı bilgi" bekłentisine uygun fakat ortalamanın üstünde bir performans göstermemiştir. Kullanılan prompt ve model yanıtının ekran görüntüsü yukarıda sunulmuştur.



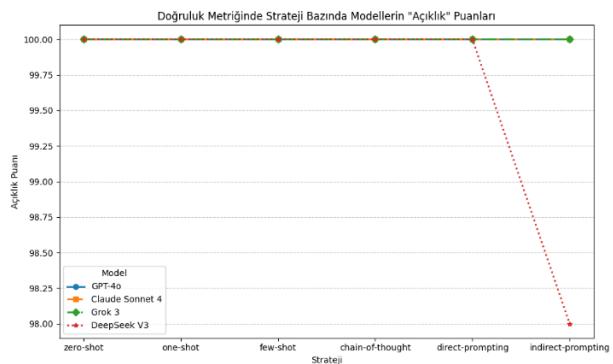
Görsel 72: "doğruluk kalitesi" metriğinde "indirect prompting" stratejisi kullanılarak girilen promptun "deepseek v3" modelindeki çıktıtı

DeepSeek V3, **indirect prompting** stratejisinde kullanıcının merakını rehberlik eden ve kısa açıklamalarla zenginleştiren bir yanıt sunmuştur. Model, doğru cevabı açıkça vermiş; kısa ek açıklamalar (görev süresi, "kurucu baba" vurgusu) eklemiştir, ancak yanıt pedagojik ya da motive edici bir dille öne çıkmamıştır. Yani, rehberlik dozu ve bağlama istenen seviyenin biraz altında kalmıştır. Yanıt bilgilendirici ve anlaşılır fakat özgünlük ve rehberlik açılarından çok yüksek performans göstermemektedir. Kullanılan prompt ve model yanıtının ekran görüntüsü yukarıda sunulmuştur.

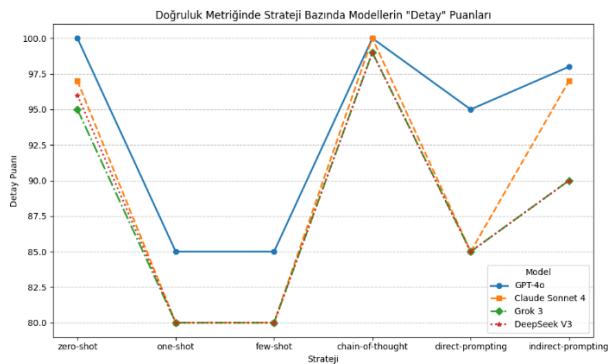
5. Doğruluk Metriği Sonuçlarının Strateji ve Model Bazında Karşılaştırılması



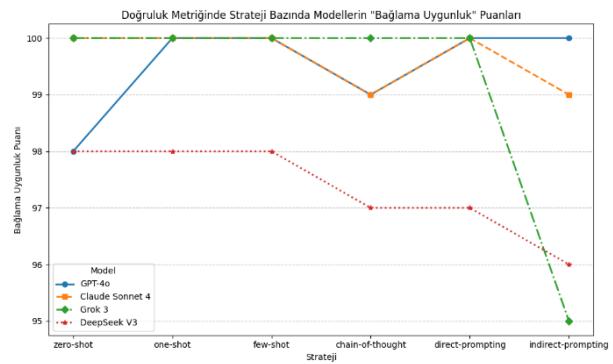
Grafik 9: Doğruluk metriğinde strateji bazında modellerin doğruluk puanları



Grafik 10: Doğruluk metriğinde strateji bazında modellerin açıklık puanları



Grafik 11: Doğruluk metriğinde strateji bazında modellerin detay puanları



Grafik 12: Doğruluk metriğinde strateji bazında modellerin bağlama uygunluk puanları

Grafik 9'da doğruluk metriği kapsamında tüm modellerin tüm stratejilerde en yüksek puana ulaştığı ve aralarında herhangi bir farklılık gözlenmediği görülmektedir. Tüm modellerin tüm stratejilerde tam puan aldığı anlaşılmaktadır. Bu sonuç, doğruluk metriğinde prompt stratejisi ve model farkının performansa anlamlı bir etkisi olmadığını, mevcut yapıların doğruluk açısından oldukça istikrarlı ve yüksek seviyede sonuç verdiği göstermektedir.

Grafik 10'da doğruluk metriği kapsamında açıklık puanlarının tüm modellerde neredeyse tüm stratejilerde maksimum seviyede olduğu görülmektedir. Ancak, DeepSeek V3 modelinin indirect prompting stratejisinde açıklık puanında belirgin bir düşüş yaşadığı dikkat çekmektedir. Diğer modeller ise tüm stratejilerde yüksek ve istikrarlı bir açıklık performansı sergilemektedir. Genel olarak, açıklık kriterinde model ve strateji farklılıklarının etkisinin oldukça sınırlı olduğu, yalnızca tekil durumda bir modelde puan kaybı yaşanabildiği gözlemlenmiştir.

Grafik 11'de, doğruluk metriği kapsamında modellerin detay puanlarında stratejilere göre belirgin değişiklikler olduğu görülmektedir. Zero-shot ve chain-of-thought stratejilerinde tüm

modellerde yüksek detay puanları dikkat çekerken, one-shot ve few-shot stratejilerinde tüm modellerin detay puanları anlamlı şekilde azalmaktadır. Indirect prompting ve direct prompting stratejilerinde ise puanlar modelden modele küçük farklılıklar gösterse de genellikle orta-yüksek seviyelerde seyretmektedir. Bu bulgular, detay kriterinde strateji seçiminin model performansı üzerinde etkili olduğunu ve bazı stratejilerin daha istikrarlı ve yüksek detay başarısı sağladığını göstermektedir.

Grafik 12'de, doğruluk metriği kapsamında bağlama uygunluk puanlarının genel olarak yüksek seviyelerde olduğu, ancak stratejilere bağlı olarak modeller arasında bazı farklılıklar ortaya çıktığı görülmektedir. Özellikle Grok 3 ve DeepSeek V3 modellerinde indirect prompting stratejisinde puanlarda belirgin bir düşüş yaşanırken, GPT-4o ve Claude Sonnet 4 modelleri çoğu stratejide yüksek ve istikrarlı performans sergilemiştir. Diğer stratejilerde ise tüm modellerin puanlarının yüksek olduğu ve aralarındaki farkların oldukça sınırlı olduğu gözlemlenmiştir. Bu sonuçlar, bağlama uygunluk kriterinde strateji seçiminin belirli modellerde performans dalgalanmalarına yol açabileceğini göstermektedir.

D. Metrik: Kod Üretimi Kalitesi

Görev: Bir yazılım problemini verilen prompt stratejisine göre istenilen programlama dilinde doğru ve çalışabilir bir şekilde kodlayabilme yetkinliğini ölçer.

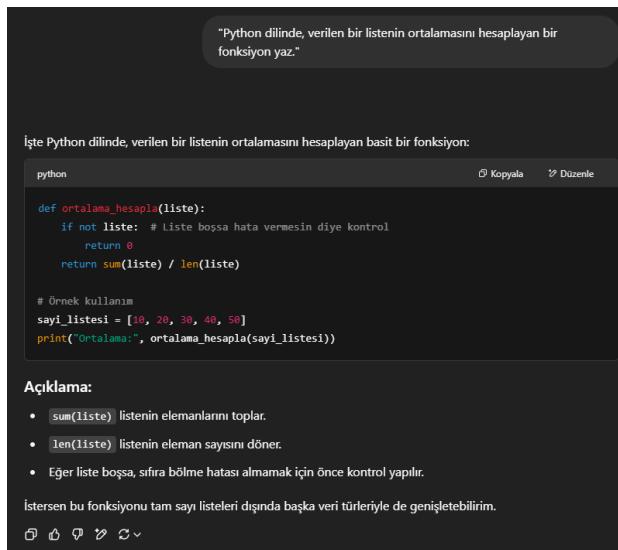
1. Model: GPT-4o

Aşağıdaki tabloda, kod üretimi metriğinde tüm prompt stratejilerinin “GPT-4o” modelindeki performansı 0-100 puanlama sisteme göre değerlendirilmiştir.

Strateji	Doğruluk	Açıklık	Detay	Bağlama Uygunluk
zero-shot	100	99	95	100
one-shot	100	97	91	100
few-shot	100	98	95	100
chain-of-thought	100	100	99	100
direct prompting	100	95	92	100
indirect prompting	100	99	98	100

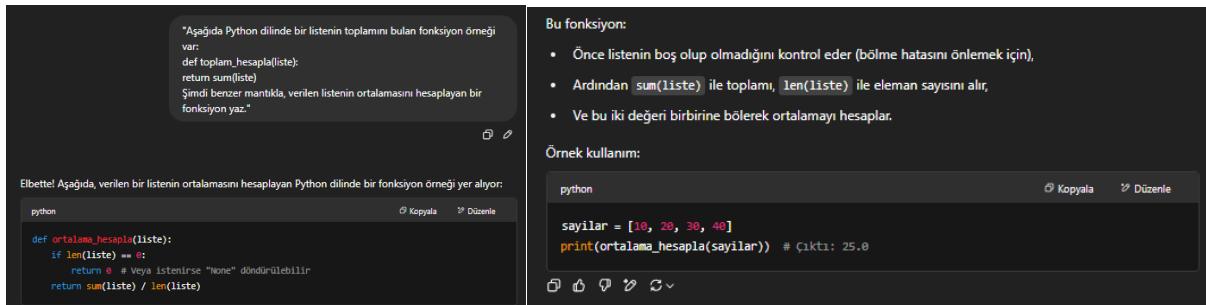
Tablo 14:Puanlama (kod üretimi kalitesi metriğinde “GPT-4o” modelinin prompt stratejilerine göre performansı)

a. Gözlemler (GPT-4o)



Görsel 73: "kod üretimi kalitesi" metriğinde "zero-shot" stratejisi kullanılarak girilen promptun "gpt-4o" modelindeki çıktıtı

GPT-4o modeli, **zero-shot** stratejisinde verilen yönergeye uygun olarak, Python dilinde verilen bir listenin ortalamasını hesaplayan işlevsel ve doğru bir fonksiyon üretmiştir. Yanıttı, hata durumlarına karşı (boş liste) kontrol eklenmiş ve örnek kullanım gösterilmiştir. Kodun ardından, fonksiyonun çalışma prensibini anlatan kısa ve anlaşılır açıklamalar yer almaktadır. Hem teknik doğruluk hem de pedagojik anlatım bakımından yüksek performans sergilenmiştir. Kodun daha da genellenebilir olduğuna dair kısa bir öneri ile bitirilmesi, yanıtın kapsayıcılığını güçlendirmiştir. Kullanılan prompt ve model yanıtının ekran görüntüsü yukarıda sunulmuştur.



Görsel 74: "kod üretimi kalitesi" metriğinde "one-shot" stratejisi kullanılarak girilen promptun "gpt-4o" modelindeki çıktıtı

GPT-4o modeli, **one-shot** stratejisinde verilen örneği takip ederek, Python dilinde listenin ortalamasını hesaplayan fonksiyonu açık ve doğru şekilde yazmıştır. Kodun temel işlevi eksiksizdir ve örnek kullanım ile çıktı gösterilmiştir. Fonksiyonun çalışma prensibi maddelerle

özetlenmiş, böylece işleyişi anlaşılır biçimde aktarılmıştır. Ancak kodun satır içi açıklamaları minimal tutulmuş ve None gibi alternatif hata yönetimi önerisi yüzeysel geçilmiştir. Yine de hem doğruluk hem de bağlama uygunluk açısından modelin çıktısı son derece başarılıdır. Kullanılan prompt ve model yanıtının ekran görüntüsü yukarıda sunulmuştur.

The screenshot shows a code editor interface with a dark theme. At the top, there is a header with the word 'python' and icons for copy, paste, and edit. Below the header, a text box contains the following Python code:

```
def ortalama_hesapla(liste):
    if not liste: # Liste boşsa hata vermemesi için kontrol
        return 0
    return sum(liste) / len(liste)

# Örnek kullanımlar:
print(ortalama_hesapla([1, 2, 3]))      # Çıktı: 2.0
print(ortalama_hesapla([1, 8, 12]))     # Çıktı: 8.0
```

Below the code, there is a note in a box:

"Aşağıdaki örneklerde benzer şekilde Python dilinde, verilen listenin ortalamasını hesaplayan bir fonksiyon yaz.
Örnek 1: Girdi: [1,2,3] → Ort: 2
Örnek 2: Girdi: [4,8,12] → Ort: 8"

At the bottom of the code editor, there is a footer with icons for file, save, and other operations.

Görsel 75: "kod üretimi kalitesi" metriğinde "few-shot" stratejisi kullanılarak girilen promptun "gpt-4o" modelindeki çıktısı

GPT-4o modeli, **few-shot** stratejisinde verilen örneklerden yola çıkarak, Python dilinde istenen fonksiyonu hem kod hem de örneklerle net ve doğru biçimde üretmiştir. Yanıttta fonksiyonun satır içi açıklamaları ve altında fonksiyonun ne yaptığına dair sade bir özet yer almaktır; ayrıca örnek kullanımda beklenen çıktılar açıkça gösterilmiştir. Kodun klasik aritmetik ortalama hesabı yaptığı ve hata durumlarının nasıl ele alınabileceği kısa da olsa vurgulanmıştır. Detay, açıklık ve bağlama uygunluk bakımından güçlü bir performans sergilenmiştir. Kullanılan prompt ve model yanıtının ekran görüntüsü yukarıda sunulmuştur.

The screenshot shows a sequence of five code snippets and their explanations:

- Adım 1: Fonksiyonu Tanımla**
python
`def ortalamam_bil(liste):`
Bu satırda `ortalamam_bil` adında bir fonksiyon tanımlanıyor. Bu fonksiyon bir liste parametresi alacak.
- Adım 2: Listenin Toplamını Hesapla**
python
`toplam = sum(liste)`
Burada `sum()` fonksiyonu ile listenin içindeki sayıların toplamını hesaplıyoruz ve `toplam` değişkenine atıyoruz.
- Adım 3: Eleman Sayısını Bul**
python
`eleman_sayisi = len(liste)`
Burada `len()` fonksiyonu ile listenin kaç elemandan oluştuğunu buluyoruz.
- Adım 4: Ortalamayı Hesapla**
python
`ortalama = toplam / eleman_sayisi`
Toplamı eleman sayısına bölerken ortalamayı buluyoruz.
- Adım 5: Sonucu Döndür**
python
`return ortalama`
Fonksiyonun sonunda hesaplanan ortalamayı döndürüyoruz.

Örnek Kullanım:

```
sayilar = [10, 20, 30, 40, 50]
print("Ortalama: ", ortalama_bil(sayilar))
```

Cıktı:
ortalama
Ortalama: 30.0

Hazırla istersen listenin boş olması durumunu da kontrol eden versiyonu yazabilirsin.

Görsel 76: "kod üretimi kalitesi" metriğinde "chain-of-thought" stratejisi kullanılarak girilen promptun "gpt-4o" modelindeki çıktıları

GPT-4o modeli, **chain-of-thought** stratejisine uygun olarak, kod üretimi görevini sistemli biçimde adım adım açıklayarak tamamlamıştır. Fonksiyonun amacı ile başlanıp her adımda ne yapılacağı sade bir dille ve örnek kod bloklarıyla aktarılmış; süreç sonunda kodun tamamı ve örnek kullanım verilmiştir. Boş liste kontrolü ise, opsiyonel bir geliştirme olarak kullanıcıya bırakılmıştır. Yanıt, pedagojik açıdan oldukça güçlü olup hem başlangıç seviyesindeki hem de ileri düzey kullanıcılar için anlaşılır niteliktedir. Kullanılan prompt ve model yanıtının ekran görüntüsü yukarıda sunulmuştur.

"Python fonksiyonu yaz: Verilen listenin ortalamasını döndür."

İste verilen listenin ortalamasını döndüren basit bir Python fonksiyonu:

```
python
def ortalama_hesapla(liste):
    if not liste:
        return 0 # Liste boşsa 0 döndür
    return sum(liste) / len(liste)
```

Örnek kullanım:

```
python
sayilar = [10, 20, 30, 40]
print(ortalama_hesapla(sayilar)) # Çıktı: 25.0
```

Eğer başka türde girişler (örneğin sadece sayılar olmalı gibi) kontrol edilmesini istersen, daha gelişmiş bir sürüm de yazabilirim.

🔗 🎮 🎧 🎵 🎪 🎩

Görsel 77: "kod üretimi kalitesi" metriğinde "direct prompting" stratejisi kullanılarak girilen promptun "gpt-4o" modelindeki çıktısı

GPT-4o modeli, **direct prompting** stratejisile verilen kısa ve net yönergeyi doğrudan, sade ve doğru bir fonksiyon ile yanıtlamıştır. Fonksiyonun temel işlevi, boş liste kontrolü ve örnek kullanımı açıkça gösterilmiştir. Kodun genel mantığı herkesin anlayacağı şekilde açıklanmış, ancak açıklama düzeyi minimumda tutulmuş ve ayrıntılılandırma yapılmamıştır. Yanıt, talebin gerektirdiği doğrudanlık ve işlevsellik bakımından oldukça başarılıdır. Kullanılan prompt ve model yanıtının ekran görüntüsü yukarıda sunulmuştur.

"Bir kullanıcı Python'da kendisine verilen sayılar listesinin ortalamasını bulmak istiyor. Bu isteği yerine getirecek fonksiyonu nasıl yazarsın?"

Python'da bir sayılar listesinin ortalamasını hesaplayan fonksiyonu şekilde yazabilirsin:

```
python
def ortalama_hesapla(sayılar):
    if not sayılar:
        return 0 # Liste boşsa ortalama 0 döner
    return sum(sayılar) / len(sayılar)
```

Açıklama:

- `sum(sayılar)`: Listedeki tüm sayıların toplamını alır.
- `len(sayılar)`: Listedeki eleman sayısını bulur.
- `sum / len`: Ortalama hesaplamasıdır.
- Boş liste kontrolü yapılmıştır. Eğer liste boşsa sıfır döndürülür (isteğe göre farklı bir hata mesajı da verilebilir).

Örnek kullanım:

```
python
liste = [10, 20, 30, 40]
print("Ortalama:", ortalama_hesapla(liste))
```

Çıktı:

```
makefile
Ortalama: 25.0
```

İstesen, kullanıcıdan liste girişi alacak bir versiyon da yazabilirim.

🔗 🎮 🎧 🎵 🎪 🎩

Görsel 78: "kod üretimi kalitesi" metriğinde "indirect prompting" stratejisi kullanılarak girilen promptun "gpt-4o" modelindeki çıktısı

GPT-4o modeli, **indirect prompting** stratejisine uygun olarak, kullanıcı odaklı bir senaryoda Python'da bir listenin ortalamasını hesaplayan fonksiyonu net, doğru ve açıklayıcı biçimde üretmiştir. Kodun ardından, işlevin her adımı maddelerle ayrıntılı olarak açıklanmış, örnek kullanım ve çıktı gösterilmiştir. Ayrıca, fonksiyonun farklı sekillerde geliştirilebileceğine

dair öneriler sunulmuştur. Yanıt hem pedagojik hem de teknik olarak üst düzey bir performans göstermektedir. Kullanılan prompt ve model yanıtının ekran görüntüsü yukarıda sunulmuştur.

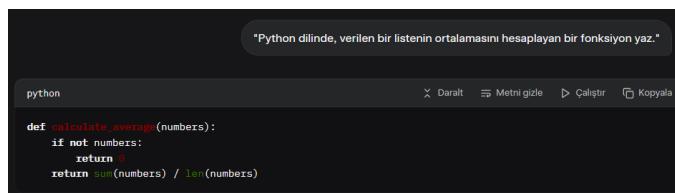
2. Model: Grok 3

Aşağıdaki tabloda, kod üretimi kalitesi metriğinde tüm prompt stratejilerinin “Grok 3” modelindeki performansı 0-100 puanlama sistemine göre değerlendirilmiştir.

Strateji	Doğruluk	Açıklık	Detay	Bağlama Uygunluk
zero-shot	100	85	80	100
one-shot	100	97	95	100
few-shot	100	98	97	100
chain-of-thought	100	99	100	100
direct prompting	100	80	80	100
indirect prompting	100	99	98	100

Tablo 15:Puanlama (kod üretimi kalitesi metriğinde “Grok 3” modelinin prompt stratejilerine göre performansı)

a. Gözlemler (Grok 3)



```
python
def calculate_average(numbers):
    if not numbers:
        return 0
    return sum(numbers) / len(numbers)
```

The screenshot shows a dark-themed code editor window. At the top, there is a status bar with the word "python". Below it, a tooltip says: "Python dilinde, verilen bir listenin ortalamasını hesaplayan bir fonksiyon yaz." The main area contains the provided Python code. At the bottom right of the editor, there are several small icons: "Daralt", "Metni Gizle", "Çalıştır", and "Kopyala".

Görsel 79:"kod üretimi kalitesi" metriğinde "zero-shot" stratejisi kullanılarak girilen promptun "grok 3" modelindeki çıktısı

Grok 3 modeli, **zero-shot** stratejisinde verilen yönergeye uygun şekilde, Python dilinde bir listenin ortalamasını hesaplayan işlevsel bir fonksiyon üretmiştir. Kod, gerekli tüm matematiksel işlemleri doğru olarak gerçekleştirmektedir ve boş liste durumu için önlem alınmıştır. Ancak, fonksiyonun nasıl kullanılacağına dair bir örnek veya açıklama sunulmamış; detay ve açıklık açısından minimumda tutulmuştur. Yine de modelin temel görevi doğru şekilde yerine getirdiği görülmektedir. Kullanılan prompt ve model yanıtının ekran görüntüsü yukarıda sunulmuştur.

"Aşağıda Python dilinde bir listenin toplamını bulan fonksiyon örneği var:

```
def toplam_hesapla(liste):
    return sum(liste)
Şimdi benzer mantıkla, verilen listenin ortalamasını hesaplayan bir fonksiyon yaz."
```

Verilen listenin ortalamasını hesaplayan bir Python fonksiyonu:

```
python
```

```
def ortalama_hesapla(liste):
    if len(liste) == 0: # Boş liste kontrollü
        return 0
    return sum(liste) / len(liste)
```

Bu fonksiyon:

1. Girilen listenin boş olup olmadığını kontrol eder.
2. Boşsa 0 döndürür (hata önlemek için).
3. Değilse, listenin toplamını `sum()` ile bular ve eleman sayısına (`len(liste)`) bölerken ortalamayı hesaplar.

Örnek kullanım:

```
python
```

```
sayilar = [1, 2, 3, 4, 5]
print(ortalama_hesapla(sayilar)) # Çıktı: 3.0
```

...

Görsel 80: "kod üretimi kalitesi" metriğinde "one-shot" stratejisi kullanılarak girilen promptun "grok 3" modelindeki çıktısı

Grok 3 modeli, **one-shot** stratejisine uygun biçimde, referans fonksiyona benzer mantıkta Python'da listenin ortalamasını hesaplayan fonksiyon üretmiştir. Fonksiyonun satır içi açıklamaları ve işleyişi madde madde özetlenmiş, örnek kullanım ve beklenen çıktı net şekilde gösterilmiştir. Kod hem teknik olarak doğru hem de pedagojik olarak anlaşılabilir biçimdedir. Detayda yalnızca klasik sayı listesi üzerinden örnek verilmiş, tür veya hata çeşitleri genişletilmemiştir. Kullanılan prompt ve model yanıtının ekran görüntüsü yukarıda sunulmuştur.

"Aşağıdaki örneklerle benzer şekilde Python dilinde, verilen listenin ortalamasını hesaplayan bir fonksiyon yaz.

Örnek 1: Girdi: [1,2,3] → Ort: 2
Örnek 2: Girdi: [4,8,12] → Ort: 8"

```
python
```

```
def ortalama_hesapla(liste):
    if not liste: # Liste boşsa
        return 0
    return sum(liste) / len(liste)

# Test
print(ortalama_hesapla([1, 2, 3])) # Çıktı: 2.0
print(ortalama_hesapla([4, 8, 12])) # Çıktı: 8.0
```

Bu fonksiyon, verilen bir sayısal listenin ortalamasını hesaplar. `sum()` ile listenin elemanlarını toplar ve `len()` ile eleman sayısına böler. Boş liste durumunda 0 döner.

...

Görsel 81: "kod üretimi kalitesi" metriğinde "few-shot" stratejisi kullanılarak girilen promptun "grok 3" modelindeki çıktısı

Grok 3 modeli, **few-shot** stratejisinde verilen örnekleri takip ederek, Python'da listenin ortalamasını hesaplayan ve her adımı açıkça gösteren bir fonksiyon üretmiştir. Kodun altında hem işlevsel hem pedagojik bir açıklama verilmiş, ayrıca kodun nasıl çalıştığını dair test

örnekleriyle birlikte çıktı sonuçları gösterilmiştir. Model hem teknik doğruluk hem de anlaşılabilirlik açısından oldukça güçlü bir çıktı sunmuştur. Kullanılan prompt ve model yanıtının ekran görüntüsü yukarıda sunulmuştur.

```


# Python dilinde listenin ortalamasını bulan fonksiyonu yazmak için önce listenin toplamını hesaplayıp, daha sonra eleman sayısına bölmeliyiz. Bu mantıkla adım adım açıklayarak kodu yazın.

Python'da bir listenin ortalamasını bulan fonksiyonu adım adım açıklayarak yazalım:
1. Fonksiyon Tanımlama: Önce fonksiyonu tanımlayacağız ve bir liste parametresi alacağız.
2. Toplam Hesaplama: Listenin tüm elemanlarını toplayacağız.
3. Eleman Sayısı: Listenin uzunluğunu bulacağız.
4. Ortalama Hesaplama: Toplamı eleman sayısına böleceğiz.
5. Hata Kontrolü: Boş liste gibi özel durumları ele alacağız.

İşte kod:

python
def liste_ortalama(liste):
    # Adım 1: Boş liste kontrolü
    if not liste: # Liste boşsa
        return None # veya None döndürülür, ihtiyaca göre

    # Adım 2: Listenin toplamını hesapla
    toplam = sum(liste) # sum() fonksiyonu ile elemanları toplayıyoruz

    # Adım 3: Eleman sayısını bul
    eleman_sayisi = len(liste) # len() ile liste uzunluğunu alıyoruz

    # Adım 4: Ortalamayı hesapla
    ortalama = toplam / eleman_sayisi

    return ortalama

# Test
ornek_liste = [1, 2, 3, 4, 5]
sonuc = liste_ortalama(ornek_liste)
print(f"Ortalama: {sonuc} ortalama: {sonuc}")

```

Açıklama:

- `if not liste`: Boş liste kontrolü yapar. Boşa 0 döndürüyoruz (alternatif olarak `None` veya hata fırlatabilir).
- `sum(liste)`: Listenin tüm elemanlarını toplar.
- `len(liste)`: Listenin eleman sayısını verir.
- `ortalama = toplam / eleman_sayisi`: Ortalamayı hesaplar.
- Test kısmında `[1, 2, 3, 4, 5]` listenin ortalamasını hesaplar (sonuç: 3.0).

Ek Notlar:

- Liste sadece sayısal değerler içermelidir, aksi takdirde hata oluşabilir. Gerekirse tip kontrolü eklenebilir.
- Daha sağlam bir kod için, `try-except` ile hata yönetimi yapılabilir:

```


def liste_ortalama(liste):
    try:
        if not liste:
            return None
        toplam = sum(liste)
        eleman_sayisi = len(liste)
        return toplam / eleman_sayisi
    except TypeError:
        return "Hata: Liste sadece sayısal değerler içermeli"

# Test
print(liste_ortalama([1, 2, 3, 4, 5])) # Çıktı: 3.0
print(liste_ortalama([])) # Çıktı: 0
print(liste_ortalama(['1', '2'])) # Çıktı: Hata mesajı

```

Bu kod, hem temel mantığı hem de hata yönetimini içerir.

Görsel 82: "kod üretimi kalitesi" metriğinde "chain-of-thought" stratejisi kullanılarak girilen promptun "grok 3" modelindeki çıktıtı

Grok 3 modeli, **chain-of-thought** stratejisinde, verilen görevi eksiksiz şekilde adım adım çözmüş ve Python'da listenin ortalamasını hesaplayan fonksiyonu önce temel haliyle, ardından hata yönetimiyle birlikte üretmiştir. Her adımda satır içi açıklamalar ve işleyiş maddelerle anlatılmış, ayrıca ek notlarda farklı tip kontrolleri ve Python'a özgü try-except kullanımı örneklenmiştir. Test örnekleri ve beklenen çıktılar da dahil edilerek, yanıt hem teknik hem pedagojik açıdan üst düzeye çıkarılmıştır. Kullanılan prompt ve model yanıtının ekran görüntüsü yukarıda sunulmuştur.

"Python fonksiyonu yaz: Verilen listenin ortalamasını döndür."

```
python
def ortalama(liste):
    if not liste: # Liste boşsa
        return 0
    return sum(liste) / len(liste)


```

1,2 saniye

Görsel 83: "kod üretimi kalitesi" metriğinde "direct prompting" stratejisi kullanılarak girilen promptun "grok 3" modelindeki çıktısı

Grok 3 modeli, **direct prompting** stratejisinde verilen kısa ve net komutu eksiksiz şekilde yerine getirmiştir. Kodun yalnızca temel işlevi verilmiş, boş liste için kontrol eklenmiş ancak herhangi bir ek açıklama, test örneği veya çıktı paylaşılmamıştır. Yanıt, doğrudanlık ve sadelik açısından başarılı olmakla birlikte, açıklık ve detay yönünden minimum düzeydedir. Kullanılan prompt ve model yanıtının ekran görüntüsü yukarıda sunulmuştur.

"Bir kullanıcı Python'da kendisine verilen sayılar listesinin ortalamasını bulmak istiyor.
Bu isteği yerine getirecek fonksiyonu nasıl yazarsın?"

Python'da bir sayılar listesinin ortalamasını bulan bir fonksiyon şu şekilde yazılabılır:

```
python
def ortalama_hesapla(sayılar):
    if not sayılar: # Liste boşsa
        return 0
    return sum(sayılar) / len(sayılar)


```

Açıklama:

- Fonksiyon, `sayılar` adında bir liste parametresi alır.
- `if not sayılar` ile listenin boş olup olmadığı kontrol edilir; boşsa 0 döner.
- `sum(sayılar)` ile listedeki sayılar toplanır, `len(sayılar)` ile eleman sayısı bulunur ve ortalama hesaplanır.

Örnek Kullanım:

```
python
sayılar = [1, 2, 3, 4, 5]
print(ortalama_hesapla(sayılar)) # Çıktı: 3.0


```

Bu fonksiyon, basit ve etkili bir şekilde ortalamayı hesaplar.

1,1 saniye

Görsel 84: "kod üretimi kalitesi" metriğinde "indirect prompting" stratejisi kullanılarak girilen promptun "grok 3" modelindeki çıktısı

Grok 3 modeli, **indirect prompting** stratejisinde verilen senaryo ve beklenkiye uygun şekilde, Python'da bir sayı listesinin ortalamasını hesaplayan fonksiyonu eksiksiz, açıklamalı ve örnek kullanımla birlikte üretmiştir. Her adımın ne yaptığı madde madde belirtilmiş, ayrıca çıktı örneğiyle birlikte fonksiyonun genel amacı özetalenmiştir. Kodun hem işlevselligi hem de

öğretimciliği üst seviyedendir. Kullanılan prompt ve model yanıtının ekran görüntüsü yukarıda sunulmuştur.

3. Model: Claude Sonnet 4

Aşağıdaki tabloda, kod üretimi kalitesi metriğinde tüm prompt stratejilerinin “Claude Sonnet 4” modelindeki performansı 0-100 puanlama sistemine göre değerlendirilmiştir.

Strateji	Doğruluk	Açıklık	Detay	Bağlama Uygunluk
zero-shot	100	100	100	95
one-shot	100	100	100	95
few-shot	100	98	100	100
chain-of-thought	100	99	100	100
direct prompting	100	100	100	97
indirect prompting	100	98	100	100

Tablo 16: Puanlama (kod üretimi kalitesi metriğinde “Claude Sonnet 4” modelinin prompt stratejilerine göre performansı)

a. Gözlemler (Claude Sonnet 4)

The screenshot shows the Claude Sonnet 4 interface. On the left, there's a message from the AI: "Python dilinde, verilen bir listenin ortalamasını hesaplayan bir fonksiyon yaz." Below it, the user asks: "İste Python'da bir listenin ortalamasını hesaplayan fonksiyon:". The AI then provides the following code:

```

# Liste Ortalaması Hesaplayan Fonksiyon
def ortalamma_hesapla(liste):
    """
    Verilen bir listenin ortalamasını hesaplar.

    Args:
        liste: Sayıların bulunduğu liste

    Returns:
        float: Listenin ortalaması

    Raises:
        ValueError: Liste boş ise veya sayısal olmayan değerler içeriyorsa
    """
    if not liste:
        raise ValueError("Liste boş olamaz")

    # Tüm elementlerin sayı olup olmadığını kontrol et
    for eleman in liste:
        if not isinstance(eleman, (int, float)):
            raise ValueError(f"Tüm elementler sayı olmalıdır. Geçersiz eleman: {eleman}")

    return sum(liste) / len(liste)

```

Below the code, there are several examples of how to use the function:

```

# Kullanım örnekleri
if __name__ == "__main__":
    # Örnek 1: Tam sayılar
    sayilar1 = [1, 2, 3, 4, 5]
    print(f"\nListe: {sayilar1}")
    print(f"Ortalama: {ortalamma_hesapla(sayilar1)}")

    # Örnek 2: Ondalıklı sayılar
    sayilar2 = [2.5, 3.7, 1.2, 8.9, 4.1]
    print(f"\nListe: {sayilar2}")
    print(f"Ortalama: {ortalamma_hesapla(sayilar2)}")

    # Hata durumu örneği
    try:
        bos_liste = []
        ortalamma_hesapla(bos_liste)
    except ValueError as e:
        print(f"\nHata: {e}")

    # Alternatif daha kısa versiyon
    def basit_ortalama(liste):
        """Basit ortalamma hesaplama fonksiyonu"""
        return sum(liste) / len(liste) if liste else 0

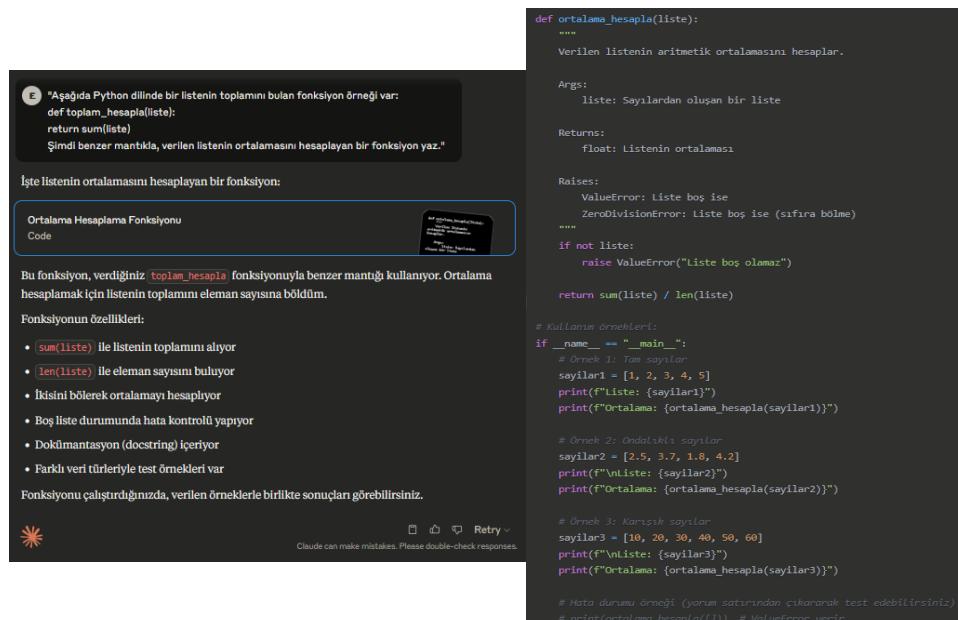
    # Numpy kullanarak alternatif (eğer Numpy yükliyse)
    def numpy_ortalama(liste):
        """Numpy kullanarak ortalamma hesaplama"""
        try:
            import numpy as np
            return np.mean(liste)
        except ImportError:
            print("NumPy yükü değil, standart fonksiyonu kullanın")
            return ortalamma_hesapla(liste)

```

Görsel 85: "kod üretimi kalitesi" metriğinde "zero-shot" stratejisi kullanılarak girilen promptun "claude sonnet 4" modelindeki çıktıları

Claude Sonnet 4 modeli, **zero-shot** stratejisile verilen Python kod üretimi görevini, ileri düzeyde kapsamlı, açıklamalı ve çoklu örneklerle desteklenmiş şekilde tamamlamıştır. Kodun ana fonksiyonu; dokümantasyon, hata kontrolü, farklı kullanım senaryoları ve alternatif kısa çözümler ile zenginleştirilmiş, ayrıca NumPy gibi kütüphane önerileriyle fonksiyonun gerçek dünyadaki uygulama esnekliği artırılmıştır.

Ancak, modelin verdiği çıktı zero-shot stratejisinin doğasına kıyasla gereğinden fazla ayrıntı ve varyasyon içermektedir. Zero-shot promptlarında amaç genellikle yalnız ve doğrudan bir fonksiyon beklemek olduğu için bağlama uygunluk puanı bir miktar kırık verilmiştir. Kullanılan prompt ve model yanıtının ekran görüntüsü yukarıda sunulmuştur.



```

def ortalama_hesapla(liste):
    """
    Verilen listenin aritmetik ortalamasını hesaplar.

    Args:
        liste: Sayılardan oluşan bir liste

    Returns:
        float: Listenin ortalaması

    Raises:
        ValueError: Liste boş ise
        ZeroDivisionError: Liste boş ise (sıfıra bölmeye)
    """
    if not liste:
        raise ValueError("Liste boş olamaz")

    return sum(liste) / len(liste)

# Kullanım örnekləri:
if __name__ == "__main__":
    # Örnek 1: Tam sayılar
    sayılar1 = [1, 2, 3, 4]
    print(f"\nListe: {sayılar1}")
    print(f"Ortalama: {ortalama_hesapla(sayılar1)}")

    # Örnek 2: Ündəlikli sayılar
    sayılar2 = [2.5, 3.7, 1.8, 4.2]
    print(f"\nListe: {sayılar2}")
    print(f"Ortalama: {ortalama_hesapla(sayılar2)}")

    # Örnek 3: Karışık sayılar
    sayılar3 = [10, 20, 30, 40, 50, 60]
    print(f"\nListe: {sayılar3}")
    print(f"Ortalama: {ortalama_hesapla(sayılar3)}")

    # Hata durumu örneği (yorum satırından çıkararak test edebilirsiniz)
    # print(ortalama_hesapla([])) # ValueError verir

```

Görsel 86: "kod üretimi kalitesi" metriğinde "one-shot" stratejisile kullanıcılarak girilen promptun "claude sonnet 4" modelindeki çıktı

Claude Sonnet 4 modeli, **one-shot** stratejisile verilen kod üretimi görevinde, temel fonksiyonu doğru ve açıklamalı şekilde üretmiş; ek olarak fonksiyonun farklı veri tiplerinde kullanımı, hata yönetimi ve ayrıntılı örneklerle yanıtını zenginleştirmiştir. Ancak, one-shot promptun doğasında yalnızca tek ve sade bir örnekle beraber temel çözüm beklenirken modelin çoklu örnekler ve geniş dokümantasyon sunması nedeniyle bağlama uygunluk puanı düşürülmüştür. Kullanılan prompt ve model yanıtının ekran görüntüsü yukarıda sunulmuştur.

E "Aşağıdaki örnekler benzer şekilde Python dilinde, verilen listenin ortalamasını hesaplayan bir fonksiyon yaz. **Örnek 1:** Girdi: [1,2,3] → Ort: 2 **Örnek 2:** Girdi: [4,8,12] → Ort: 8"

İste Python dilinde liste ortalaması hesaplayan bir fonksiyon:

Liste Ortalaması Hesaplayan Fonksiyon

```
def ortalamo_hesapla(liste):
    """
    Verilen listenin ortalamasını hesaplar.

    Args:
        liste (list): Sayılardan oluşan liste

    Returns:
        float: Listenin ortalaması
    """
    if not liste:
        return 0

    return sum(liste) / len(liste)

# Test örnekleri
print("Örnek 1:")
liste1 = [1, 2, 3]
ort1 = ortalamo_hesapla(liste1)
print(f"Girdi: {liste1} → Ort: {ort1}")

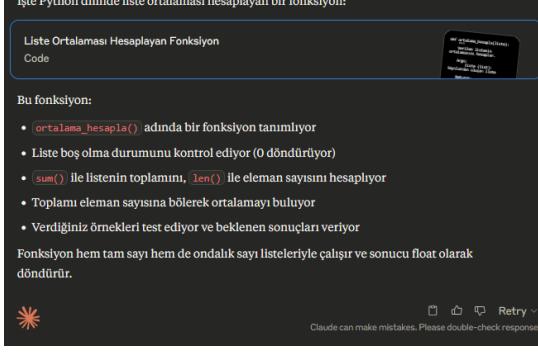
print("\nÖrnek 2:")
liste2 = [4, 8, 12]
ort2 = ortalamo_hesapla(liste2)
print(f"Girdi: {liste2} → Ort: {ort2}")

# Ek test örnekleri
print("\nEk testler:")
print("Örnek 1:")
print(f"Girdi: [5, 10, 15, 20] → Ort: {ortalamo_hesapla([5, 10, 15, 20])}")
print("Örnek 2:")
print(f"Girdi: [1] → Ort: {ortalamo_hesapla([1])}")
print("Örnek 3:")
print(f"Girdi: [] → Ort: {ortalamo_hesapla([])}")
```

Bu fonksiyon:

- `ortalama_hesapla()` adında bir fonksiyon tanımılıyor
- Liste boş olma durumunu kontrol ediyor (0 döndürür)
- `sum()` ile listenin toplamını, `len()` ile eleman sayısını hesaplıyor
- Toplamı eleman sayısına bölerek ortalamayı buluyor
- Verdığınız örnekleri test ediyor ve beklenen sonuçları veriyor

Fonksiyon hem tam sayı hem de ondalık sayı listeleriley çalışır ve sonucu float olarak döndürür.



Görsel 87: "kod üretimi kalitesi" metriğinde "few-shot" stratejisi kullanılarak girilen promptun "claude sonnet 4" modelindeki çıktıları

Claude Sonnet 4 modeli, **few-shot** stratejisiyle verilen kod üretimi görevinde, önce ana fonksiyonu ayrıntılı docstring ile sunmuş, ardından temel ve ek test örnekleriyle işlevin farklı durumlarda nasıl çalıştığını net şekilde göstermiştir. Fonksiyonun amacı, kullanım biçimini ve olası senaryolar örneklerle pedagojik biçimde desteklenmiştir. Ancak kodun içinde, özellikle satır içi adım adım açıklamalar bulunmadığı için, açıklık puanı bir miktar düşürülmüştür. Kullanılan prompt ve model yanıtının ekran görüntüsü yukarıda sunulmuştur.

E "Python dilinde listenin ortalamasını bulan fonksiyonu yazmak için önce listenin toplamını hesaplayın, daha sonra eleman sayısını bölmeliyiz. Bu mantık, adım adım açıklayarak kodu yaz."

Python'da listenin ortalamasını bulan fonksiyonu adım adım yazalım:

Liste Ortalaması Hesaplama Fonksiyonu

```
def liste_ortalama(liste):
    """
    Bir listenin aritmetik ortalamasını hesaplar.

    Args:
        liste: Sayılardan oluşan bir liste

    Returns:
        float: Listenin ortalaması
    """
    if not liste:
        raise ValueError("Liste boş ise")
    if len(liste) == 1:
        raise ValueError("Liste elemanları sayı olmaz")

    # Adım 1: Listenin boş olup olmadığını kontrol et
    if not liste:
        raise ValueError("Liste boş olmaz!")

    # Adım 2: Listenin toplamını hesapla
    toplam = 0
    for elem in liste:
        # Her elemanın sayı olduğunu kontrol et
        if not isinstance(elem, (int, float)):
            raise TypeError("Liste elemanları sayı olmalıdır. Geçersiz eleman: ({})".format(elem))
        toplam += elem

    # Adım 3: Eleman sayısını bul
    eleman_sayisi = len(liste)

    # Adım 4: Ortalamayı hesapla
    ortlama = toplam / eleman_sayisi

    return ortlama

# Alternatif: data kisa versiyon (built-in fonksiyonları)
def liste_ortalama_kisa(liste):
    """
    Built-in fonksiyonları kullanarak ortalamayı hesaplar.
    """
    if not liste:
        raise ValueError("Liste boş olmaz!")
    return sum(liste) / len(liste)
```

İste adım adım açıklaması:

Adım 1: Giriş Kontrolü

- Fonksiyon başında listenin boş olmadığını kontrol ediyor
- Boş liste durumunda hata fırlatıyor

Adım 2: Toplam Hesapma

- `for` döngüsü ile listenin her elemanını geziyor
- Her elemanın sayı olduğunu kontrol ediyor
- Tüm elemanları `toplam` değişkenine ekliyor

Adım 3: Eleman Sayısı Bulma

- `len()` fonksiyonu ile listenin uzunluğunu alıyor

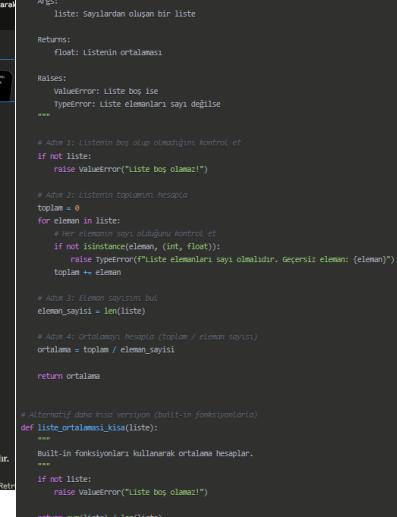
Adım 4: Ortalamaya Hesaplama

- Toplam değeri eleman sayısına bölererek ortalamayı buluyoruz
- Sonucu return ediyoruz

Ek Özellikler

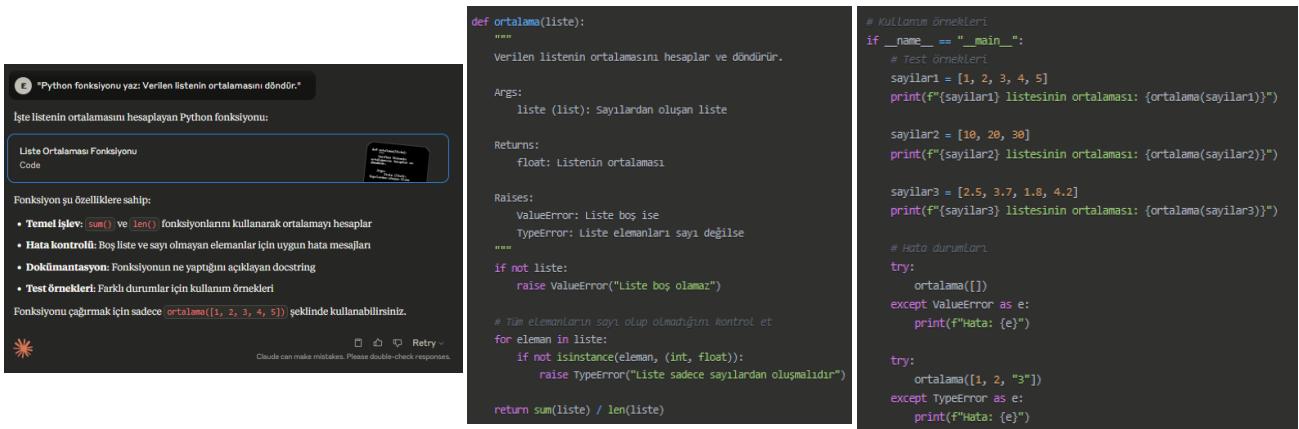
- Hata kontrolü (boş liste ve geçersiz veri tipleri için)
- Alternatif kisa versiyon (`sum()`) fonksiyonu kullanarak
- Test örnekleri ile fonksiyonun doğru çalıştığını gösteriyoruz

Bu fonksiyon hem tam sayılar hem de ondalık sayılarla çalışır ve hata durumlarını da ele alır.



Görsel 88: "kod üretimi kalitesi" metriğinde "chain-of-thought" stratejisi kullanılarak girilen promptun "claude sonnet 4" modelindeki çıktıları

Claude Sonnet 4 modeli, **chain-of-thought** stratejisinde kod üretimi görevini, süreç odaklı ve adım adım açıklamalarla tamamlamıştır. Her adım için ayrı kod ve açıklama eklenmiş, fonksiyonun hem klasik hem kısa versiyonu sunulmuş, ayrıca çeşitli testler ve hata yönetimi örnekleriyle yanıt zenginleştirilmiştir. Kodun açıklanma biçimini hem temel kullanıcılar hem de ileri seviye programcılar için anlaşılır ve rehber niteliğindedir. Kullanılan prompt ve model yanıtının ekran görüntüsü yukarıda sunulmuştur.



The screenshot shows the Claude Sonnet 4 interface. On the left, a sidebar displays a message about writing a Python function to calculate the average of a list. Below it, a section titled "Liste Ortalaması Fonksiyonu" contains the generated Python code for calculating the average of a list. On the right, there are two examples of how to use the function with different lists.

```

def ortalama(liste):
    """
    Verilen listenin ortalamasını hesaplar ve döndürür.

    Args:
        liste (list): Sayılardan oluşan liste

    Returns:
        float: Listenin ortalaması

    Raises:
        ValueError: Liste boş ise
        TypeError: Liste elemanları sayı değilse
    """
    if not liste:
        raise ValueError("Liste boş olamaz")

    # Tüm elemanların sayı olup olmadığını kontrol et
    for elemen in liste:
        if not isinstance(elemen, (int, float)):
            raise TypeError("Liste sadece sayılarından oluşmalıdır")

    return sum(liste) / len(liste)

```

```

# Kullanım Örnekleri
if __name__ == "__main__":
    # Test Örnekleri
    sayilar1 = [1, 2, 3, 4, 5]
    print(f"{sayilar1} listenin ortalaması: {ortalama(sayilar1)}")

    sayilar2 = [10, 20, 30]
    print(f"{sayilar2} listenin ortalaması: {ortalama(sayilar2)}")

    sayilar3 = [2.5, 3.7, 1.8, 4.2]
    print(f"{sayilar3} listenin ortalaması: {ortalama(sayilar3)}")

    # Hata durumları
    try:
        ortalama([])
    except ValueError as e:
        print(f"Hata: {e}")

    try:
        ortalama([1, 2, "3"])
    except TypeError as e:
        print(f"Hata: {e}")

```

Görsel 89: "kod üretimi kalitesi" metriğinde "direct prompting" stratejisi kullanılarak girilen promptun "claude sonnet 4" modelindeki çıktıtı

Claude Sonnet 4 modeli, **direct prompting** stratejisine verilen kısa ve açık komutu teknik açıdan fazlasıyla doğru, örnekli ve açıklamalı şekilde üretmiştir. Fonksiyonun temel işlevi, hata yönetimi, ayrıntılı docstring ve çeşitli test örnekleriyle çok kapsamlı biçimde gösterilmiştir. Ancak, direct prompting yaklaşımında genellikle daha kısa ve doğrudan çözümler beklenigidinden bağlama uygunluk puanı düşürülmüştür. Kullanılan prompt ve model yanıtının ekran görüntüsü yukarıda sunulmuştur.

```

def ortalama_bul(sayılar):
    """
    Verilen sayılar listesinin ortalamasını hesaplar.

    Args:
        sayılar (list): Sayılarından oluşan liste

    Returns:
        float: Sayıların ortalaması

    Raises:
        ValueError: Liste boş ise hata verir
    """
    if not sayılar:
        raise ValueError("Liste boş olamaz!")

    return sum(sayılar) / len(sayılar)

# Karakteristik veri tipleri ile test
if __name__ == "__main__":
    # Test verileri
    test_listesi = [10, 20, 30, 40, 50]
    print(f"Liste: {test_listesi}")
    print(f"Ortalama: {ortalama_bul(test_listesi)}")

    # Karakteristik liste
    karisik_liste = [10, 20, "abc", 30, 40]
    print(f"\nKarışık liste: {karisik_liste}")
    print(f"Güvenli ortalama: {ortalama_bul_güvenli(karisik_liste)}")

    # Boş Liste testi
    bos_liste = []
    print(f"\nBoş liste ortalaması: {ortalama_bul_güvenli(bos_liste)}")

    # Numpy ile karşılaştırma (isteğe bağlı)
    try:
        import numpy as np
        numpy_ortalama = np.mean(test_listesi)
        print(f"\nNumpy ile ortalama: {numpy_ortalama}")
    except ImportError:
        print("\nNumpy yüküldü değil, karşılaştırma yapılmadı.")

```

Görsel 90: "kod üretimi kalitesi" metriğinde "indirect prompting" stratejisi kullanılarak girilen promptun "claude sonnet 4" modelindeki çıktıtı

Claude Sonnet 4 modeli, **indirect prompting** stratejisinde kod üretimi görevini son derece esnek ve kullanıcı odaklı bir biçimde gerçekleştirmiştir. Kullanım örnekleriyle fonksiyonun farklı durumlarda nasıl çalıştığı detaylandırılmış, açıklamalar ve hata yönetimiyle birlikte pedagojik açıdan da güclü bir yanıt verilmiştir. Ancak, kod bloklarında yer alan örneklerin hemen altında kısa özet açıklamalar veya satır içi adım adım açıklamalar eksik bırakıldığından açıklık puanı düşürülmüştür. Kullanılan prompt ve model yanıtının ekran görüntüsü yukarıda sunulmuştur.

4. Model: DeepSeek V3

Aşağıdaki tabloda, kod üretimi kalitesi metriğinde tüm prompt stratejilerinin “DeepSeek V3” modelindeki performansı 0-100 puanlama sistemine göre değerlendirilmiştir.

Strateji	Doğruluk	Açıklık	Detay	Bağlama Uygunluk
zero-shot	100	99	97	100
one-shot	100	99	98	100
few-shot	100	99	99	100
chain-of-thought	100	99	100	100
direct prompting	100	99	98	100
indirect prompting	100	99	100	100

Tablo 17: Puanlama (kod üretimi kalitesi metriğinde “DeepSeek V3” modelinin prompt stratejilerine göre performansı)

a. Gözlemler (DeepSeek V3)

The screenshot shows a user interface for generating code. A top bar says "Python dilinde, verilen bir listenin ortalamasını hesaplayan bir fonksiyon yaz." Below it, a section titled "Python'da Liste Ortalamasını Hesaplayan Fonksiyon" contains a code snippet:

```
python
def liste_ortalaması(liste):
    ...
    Verilen bir listenin ortalamasını hesaplar.

    Parametreler:
        liste (list): Sayılar içeren bir liste

    Döndürür:
        float: Listenin ortalaması
        ...
    if not liste: # Liste boşsa
        return 0
    return sum(liste) / len(liste)

# Örnek kullanım
sayilar = [10, 20, 30, 40, 50]
ortalama = liste_ortalaması(sayilar)
print(f"Listenin ortalaması: {ortalama}") # Çıktı: Listenin ortalaması: 30.0
```

To the right, there's an explanation of the function:

Bu fonksiyon:

1. `sum()` fonksiyonuyla listenin toplamını alır
2. `len()` fonksiyonuyla liste uzunluğunu bulur
3. Toplamı uzunluğu bölgerek ortalamayı hesaplar
4. Boş liste durumunda 0 döndürür (isteğe bağlı olarak bu kısmı değiştirilebilir)

Below that, another section says "Alternatif olarak, boş liste durumunda hata fırlatmak isterseniz:" with a different code snippet:

```
python
def liste_ortalaması(liste):
    if len(liste) == 0:
        raise ValueError("Boş listenin ortalaması hesaplanamaz")
    return sum(liste) / len(liste)
```

Görsel 91: "kod üretimi kalitesi" metriğinde "zero-shot" stratejisi kullanılarak girilen promptun "deepseek v3" modelindeki çıktısı

DeepSeek V3 modeli, **zero-shot** stratejisinde verilen kod üretimi görevini teknik olarak doğru, yalın ve açık bir şekilde tamamlamıştır. Fonksiyon, boş liste kontrolü ve örnek kullanım ile temel gereksinimi karşılamakta, ayrıca isteğe bağlı olarak hata yönetimi için bir alternatif de sunulmuştur. Açıklamalar yeterince açık ve anlaşılır olsa da kod bloklarında satır içi açıklama ve farklı test örneklerinin eksiksliği nedeniyle açıklık ve detay puanlarında küçük bir düşüş yapılmıştır. Kullanılan prompt ve model yanıtının ekran görüntüsü yukarıda sunulmuştur.

The screenshot shows a user interface for generating code. A top bar says "Aşağıda Python dilinde bir listenin toplamını bulan fonksiyon örneği var: def toplam_hesapla(liste): return sum(liste) Şimdi benzer mantıkla, verilen listenin ortalamasını hesaplayan bir fonksiyon yaz." Below it, a section titled "Verilen bir listenin ortalamasını hesaplayan Python fonksiyonunu aşağıda bulabilirsiniz:" contains a code snippet:

```
python
def ortalama_hesapla(liste):
    if len(liste) == 0:
        return 0 # veya isterseniz hata fırlatabilirsiniz, örneğin: raise ValueError("Liste boş olamaz!")
    return sum(liste) / len(liste)
```

To the right, there's an explanation of the function:

Açıklama:

1. `sum(liste)`: Listenin elemanlarının toplamını hesaplar.
2. `len(liste)`: Listenin uzunluğunu (eleman sayısını) verir.
3. **Boş liste kontrolü**: `if len(liste) == 0` ile listenin boş olmadığını kontrol ediyoruz. Eğer liste boşsa, sıfır döndürüyoruz (veya isterseniz bir hata fırlatabilirsiniz).

Below that, another section says "Örnek Kullanım:" with a code snippet:

```
python
sayilar = [10, 20, 30, 40, 50]
print(ortalama_hesapla(sayilar)) # Çıktı: 30.0
```

Below the code, there's a note: "Bu fonksiyon, `sum()` ve `len()` fonksiyonlarını kullanarak listenin ortalamasını kolayca hesaplar."

Görsel 92: "kod üretimi kalitesi" metriğinde "one-shot" stratejisi kullanılarak girilen promptun "deepseek v3" modelindeki çıktısı

DeepSeek V3 modeli, **one-shot** stratejisile verilen kod üretimi görevini, temel fonksiyonu ve işlevi açık şekilde açıklayarak başarıyla tamamlamıştır. Fonksiyonun temel mantığı ve nasıl çalıştığı maddeler halinde özetlenmiş, örnek kullanım ve beklenen çıktı gösterilmiştir. Kodun satır içi açıklamaları minimal düzeyde olsa da genel açıklık ve teknik doğruluk yüksektir. Farklı senaryolar veya ek hata yönetimi örnekleri de verilmiş olsaydı detay daha da zenginleşebilirdi. Kullanılan prompt ve model yanıtının ekran görüntüsü yukarıda sunulmuştur.

"Aşağıdaki örnekler benzer şekilde Python dilinde, verilen listenin ortalamasını hesaplayan bir fonksiyon yaz.
 Örnek 1: Girdi: [1,2,3] → Ort: 2
 Örnek 2: Girdi: [4,8,12] → Ort: 8"

Python'da Liste Ortalamasını Hesaplayan Fonksiyon

```
python
def ortalamam_hesapla(liste):
    if len(liste) == 0:
        return 0 # Boş liste durumunda 0 döndürür
    toplam = sum(liste)
    ortalamam = toplam / len(liste)
    return ortalamam

# Örnek kullanımları
print(ortalamam_hesapla([1, 2, 3])) # Çıktı: 2.0
print(ortalamam_hesapla([4, 8, 12])) # Çıktı: 8.0
```

Açıklama:

1. Fonksiyon önce listenin boş olup olmadığını kontrol eder (sıfır bölme hatasını önlemek için)
2. `sum()` fonksiyonu ile listedeki tüm sayıların toplamını hesaplar
3. Toplamı liste uzunluğuna bölerken ortalamayı bulur
4. Ortalamayı döndürür

Not: Python'da bölme işlemi otomatik olarak float sonuç verir. Eğer tam sayı sonuç isterseniz `//` operatörünü kullanabilirsiniz, ancak bu durumda kesirli kısım kaybolur.

Görsel 93: "kod üretimi kalitesi" metriğinde "few-shot" stratejisi kullanılarak girilen promptun "deepseek v3" modelindeki çıktıtı

DeepSeek V3 modeli, **few-shot** stratejisile verilen kod üretimi görevinde, fonksiyonun temel işleyişini ve örnek kullanımlarını açıkça göstermiştir. Her adım maddeler halinde özetlenmiş, kodun sonunda bölüm işleminin tipine dair ek bir notla detaylandırılmıştır. Kodun satır içi açıklamaları eksik olsa da örnekler ve özet bilgiler ile anlaşılabilirlik yüksektir. Farklı hata senaryoları veya gelişmiş varyasyonlar sunulmuş olsaydı detay ve açıklık daha da yükseltebilirdi. Kullanılan prompt ve model yanıtının ekran görüntüsü yukarıda sunulmuştur.

"Python dilinde listenin ortalamasını bulan fonksiyon yazmak için önce listenin toplamını hesaplayıp, daha sonra eleman sayısına bölmen gerekiyor. Bu mantık, adım adım açıklayarak kodu yaz."

Python'da Liste Ortalamasını Bulan Fonksiyon

İşte Python'da bir listenin ortalamasını bulan fonksiyonun adım adım açıklamalı hali:

```
python
def liste_ortalaması(liste):
    """
    Bir listenin sayısal ortalamasını hesaplayan fonksiyon.

    Parametrelər:
        liste (list): Ortalaması hesaplanacak sayı listesi

    Returns:
        float: Listenin ortalaması
    """

    # 1. Adım: Listenin toplamını hesapla
    toplam = 0
    for sayı in liste:
        toplam += sayı

    # 2. Adım: Listenin eleman sayısını bul
    eleman_sayısı = len(liste)

    # 3. Adım: Ortalamayı hesapla (toplam / eleman_sayısı)
    ortalama = toplam / eleman_sayısı

    # 4. Adım: Ortalamayı döndür
    return ortalama
```

Örnek Kullanım

```
python
sayilar = [10, 20, 30, 40, 50]
print(liste_ortalaması(sayılar)) # Çıktı: 30.0
```

Alternatif Kısa Versiyon

Python'un yerleşik fonksiyonlarını kullanarak daha kısa yazabiliriz:

```
python
def liste_ortalaması(liste):
    return sum(liste) / len(liste)
```

Hata Kontrollü Versiyon

Boş liste durumunda hata vermemesi için:

```
python
def liste_ortalaması(liste):
    if len(liste) == 0:
        return 0 # veya hata fırlatabilir
    return sum(liste) / len(liste)
```

Bu fonksiyon, verilen bir sayı listenin aritmetik ortalamasını hesaplar. Temel mantık, tüm elemanların toplamını eleman sayısına bölmektir.

Görsel 94: "kod üretimi kalitesi" metriğinde "chain-of-thought" stratejisi kullanılarak girilen promptun "deepseek v3" modelindeki çıktıtı

DeepSeek V3 modeli, **chain-of-thought** stratejisinde kod üretimi görevini, adım adım açıklama, farklı fonksiyon varyasyonları ve örnek kullanımlarla zenginleştirerek tamamlamıştır. Fonksiyonun hem ayrıntılı hem kısa versiyonu, hata kontrolü ve açıklamaları detaylı biçimde aktarılmıştır. Yanıt, teknik doğruluk, pedagojik açıklık ve stratejiye uygunluk

bakımından üst düzeydedir. Kullanılan prompt ve model yanıtının ekran görüntüsü yukarıda sunulmuştur.

"Python fonksiyonu yaz: Verilen listenin ortalamasını döndür."

İste, verilen bir listenin ortalamasını hesaplayan bir Python fonksiyonu:

```
python
def liste_ortalaması(liste):
    if not liste: # Liste boşsa
        return 0
    return sum(liste) / len(liste)
```

Örnek Kullanım:

```
python
sayilar = [1, 2, 3, 4, 5]
ortalama = liste_ortalaması(sayilar)
print("Ortalama:", ortalama) # Çıktı: Ortalama: 3.0
```

Açıklamalar:

1. `sum(liste)` fonksiyonu, listedeki tüm sayıların toplamını verir.
2. `len(liste)` fonksiyonu, listedeki eleman sayısını verir.
3. Fonksiyon, boş bir liste verilirse `0` döndürür (isteğe bağlı olarak bu kısım değiştirilebilir).

Bu fonksiyon, tam sayılar (`int`) veya ondalıklı sayılar (`float`) içeren listelerle çalışır.

Görsel 95: "kod üretimi kalitesi" metriğinde "direct prompting" stratejisi kullanılarak girilen promptun "deepseek v3" modelindeki çıktıtı

DeepSeek V3 modeli, **direct prompting** stratejisinde kod üretimi görevini yalın ve doğru şekilde tamamlamıştır. Fonksiyonun temel mantığı ve örnek kullanımı açıkça gösterilmiş, her adım maddelerle özetlenmiştir. Kodun satır içi açıklama ve ek varyasyonlardan yoksun olması, açıklık ve detay puanlarının biraz düşmesine neden olmuştur. Genel olarak ise stratejiye, teknik doğruluğa ve anlaşılırlığa tam uyum sağlanmıştır. Kullanılan prompt ve model yanıtının ekran görüntüsü yukarıda sunulmuştur.

"Bir kullanıcı Python'da kendisine verilen sayılar listenin ortalamasını bulmak istiyor. Bu isteği yerine getirecek fonksiyon nasıl yazarsınız?"

Python'da bir sayı listeninin ortalamasını hesaplayan bir fonksiyon yazmak için aşağıdaki adımları takip edebiliriz:

1. Fonksiyon Tanımı: Fonksiyon, bir sayı listesi almalıdır.
2. Toplam Hesaplama: Listedeki sayıların toplamını bulmalıdır.
3. Eleman Sayısı: Listedeki eleman sayısını bulmalıdır.
4. Ortalama Hesaplama: Toplamı eleman sayısına bölerek ortalamayı hesaplamalıyız.
5. Hata Kontrolü: Liste boşsa, hata vermeme için kontrol eklemeliyiz.

İşte bu adımları içeren bir Python fonksiyonu:

```
python
def ortalama_hesapla(sayılar):
    if len(sayılar) == 0:
        return 0 # Veya hata mesajı verebilirsiniz; raise ValueError("Liste boş!")
    toplam = sum(sayılar)
    adet = len(sayılar)
    ortalama = toplam / adet
    return ortalama
```

Örnek Kullanım:

```
python
liste = [10, 20, 30, 40, 50]
sonuc = ortalama_hesapla(liste)
print("Ortalama:", sonuc) # Çıktı: Ortalama: 30.0
```

Alternatif (Daha Kısa):

```
python
def ortalama_hesapla(sayılar):
    return sum(sayılar) / len(sayılar) if sayılar else 0
```

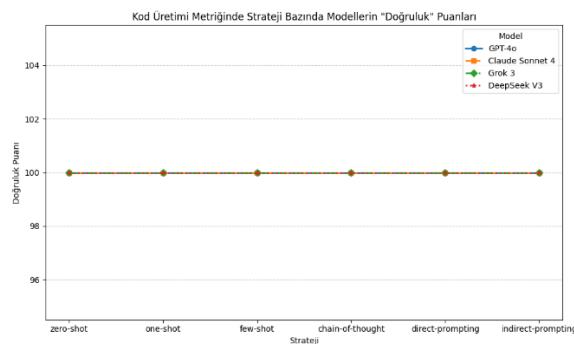
Not:

- Eğer liste boşsa (`len(sayılar) == 0`), sıfır bölme hatası (`ZeroDivisionError`) almamak için bir kontrol ekledik. İsterseniz `return 0` yerine bir hata fırlatabilirsiniz (örneğin, `raise ValueError("Liste boş olamaz!")`).

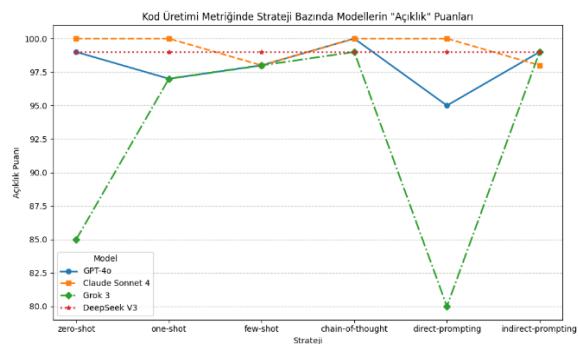
Görsel 96: "kod üretimi kalitesi" metriğinde "indirect prompting" stratejisi kullanılarak girilen promptun "deepseek v3" modelindeki çıktıtı

DeepSeek V3 modeli, **indirect prompting** stratejisile verilen kod üretimi görevinde hem temel fonksiyonu hem de alternatif kısa sürümü ve hata yönetimini kullanıcıya sunmuştur. Adım adım açıklamalar, örnek kullanım ve notlar sayesinde pedagojik ve teknik olarak yüksek bir anlaşılırlık yakalanmıştır. Kodun satır içi açıklamalarının kısıtlı olması, açıklık puanında küçük bir kırıga neden olmuştur; ancak genel olarak detay, doğruluk ve bağlama uygunluk üst düzeydedir. Kullanılan prompt ve model yanıtının ekran görüntüsü yukarıda sunulmuştur.

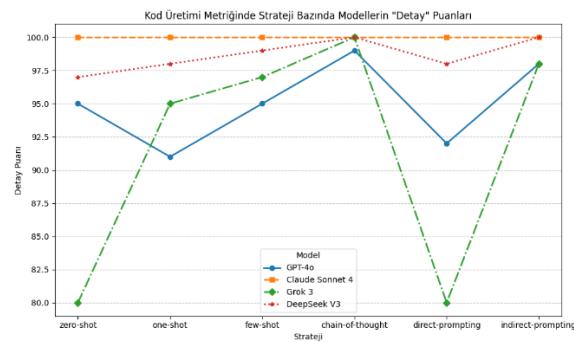
5. Kod Üretimi Metriği Sonuçlarının Strateji ve Model Bazında Karşılaştırılması



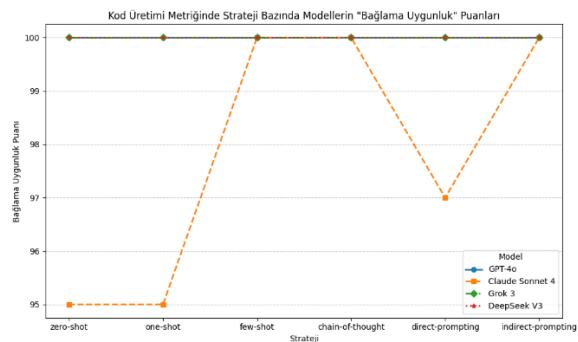
Grafik 13: Kod üretimi metriğinde strateji bazında modellerin doğruluk puanları



Grafik 14: Kod üretimi metriğinde strateji bazında modellerin açıklık puanları



Grafik 15: Kod üretimi metriğinde strateji bazında modellerin detay puanları



Grafik 16: Kod üretimi metriğinde strateji bazında modellerin bağlama uygunluk puanları

Grafik 13'te, kod üretimi metriği kapsamında tüm modellerin, bütün prompt stratejilerinde maksimum doğruluk puanına ulaştığı görülmektedir. Farklı stratejiler arasında modellerin doğruluk performansında herhangi bir dalgalanma veya farklılık bulunmamaktadır. Bu bulgu, kod üretimi görevinde mevcut büyük dil modellerinin ve stratejilerin doğruluk açısından oldukça istikrarlı ve yüksek bir başarıya sahip olduğunu göstermektedir.

Grafik 14'te, kod üretimi metriği kapsamında açıklık puanlarının stratejiye ve modele göre değişiklik gösterdiği gözlenmektedir. Özellikle Grok 3 modelinde direct prompting

stratejisinde açıklık puanında belirgin bir düşüş dikkat çekerken, diğer modellerde genel olarak yüksek açıklık değerleri korunmuştur. Tüm modellerde ve stratejilerde açıklık puanları büyük oranda yüksek seviyede seyretmekle birlikte, bazı stratejilerde model bazında sınırlı dalgalanmalar ortaya çıkmaktadır. Bu bulgular, kod üretimi görevinde açıklık kriterinin model ve stratejiye göre farklılaşabildiğini, ancak genel olarak yüksek başarı düzeyinin korunduğunu göstermektedir.

Grafik 15'te, kod üretimi metriği kapsamında detay puanlarının modeller ve stratejiler arasında belirgin farklılıklar gösterdiği görülmektedir. Özellikle Grok 3 modelinde zero-shot ve direct prompting stratejilerinde detay puanlarında düşüslər gözlemlenirken, diğer stratejilerde bu modelin puanları yüksektir. GPT-4o modelinde ise direct prompting ve one-shot stratejilerinde görelə olaraq daha düşük detay puanları göze çarpmaktadır. Claude Sonnet 4 ve DeepSeek V3 modelleri ise neredeyse tüm stratejilerde yüksek ve istikrarlı detay puanlarına ulaşmıştır. Bu bulgular, kod üretimi görevinde detay kriterinin model ve stratejiye duyarlı olduğunu, bazı modellerde ve stratejilerde detay seviyesinin anlamlı şekilde değiştirdiğini göstermektedir.

Grafik 16'da kod üretimi metriği kapsamında bağlama uygunluk puanlarının büyük ölçüde tüm modellerde ve stratejilerde en yüksek seviyede olduğu görülmektedir. Sadece Claude Sonnet 4 modelinde zero-shot, one-shot ve direct prompting stratejilerinde puanlarda belirgin bir azalma dikkat çekmektedir. Diğer modeller ise bütün stratejilerde bağlama uygunluk açısından istikrarlı ve yüksek puanlar sergilemiştir. Bu sonuç, kod üretimi görevinde bağlama uygunluğu bakımından modellerin ve stratejilerin genellikle yüksek başarıya sahip olduğunu, ancak bazı tekil durumlarda model bazlı küçük farklılıkların oluşabildiğini göstermektedir.

IV. TARTIŞMA

A. Strateji ve Metrik Bazında Karşılaştırmalı Özeti Tablo

Ana Metrik	En Yüksek Ortalama Strateji	Strateji Ort.	En Düşük Ortalama Strateji	Strateji Ort.
soru-cevap	chain-of-thought	396,75	few-shot	303,5
kod-üretimi	chain-of-thought	399,25	direct prompting	385,25
özetleme	chain-of-thought	395	direct prompting	382,5
doğruluk	chain-of-thought	398,25	zero / few shot	380,75

Tablo 18: Ana Metriklere Göre Stratejilerin Ortalama Performans Karşılaştırması

Tablo 18'de büyük dil modellerinin farklı prompt stratejileri ile değerlendirilmesi sonucunda, **chain-of-thought** stratejisinin tüm ana metriklerde (soru-cevap, kod-üretimi, özetleme, doğruluk) ortalama puan bakımından diğer stratejlere göre anlamlı biçimde öne çıktıgı gözlemlenmiştir. Özellikle soru-cevap ve kod-üretimi gibi detay gerektiren görevlerde, bu strateji istikrarlı biçimde en yüksek puanlara ulaşmıştır. Bu durum, chain-of-thought yaklaşımının modellerin içsel akıl yürütme kabiliyetlerini ve tutarlılığını artırmada etkili olduğunu ortaya koymaktadır.

Öte yandan, **few-shot** ve **direct prompting** stratejilerinin genel olarak daha düşük ortalama puanlar aldığı, yani farklı modellerde bekleninin altında performans sergilediği tespit edilmiştir. Doğruluk metriğinde ise zero-shot ve few-shot stratejileri en düşük ortalama değeri paylaşmıştır. Bu bulgu, özellikle karmaşık ve açıklama gerektiren görevlerde, modelin tek bir örnek ya da doğrudan komut ile yeterince derin ve nitelikli çıktı üretmekte zorlandığını göstermektedir.

Tüm bu bulgular, büyük dil modelleri ile çalışan araştırmacı ve uygulayıcılar için, strateji seçiminin çıktı kalitesi üzerinde kritik bir rol oynadığını ortaya koymaktadır. Özellikle karmaşık ve çok aşamalı görevlerde, chain-of-thought gibi adım adım akıl yürütmeye dayalı stratejilerin tercih edilmesi, çıktı kalitesini artırmada önemli bir avantaj sağlamaktadır. Bununla birlikte, model ve görev tipine bağlı olarak, diğer stratejiler de uygun senaryolarda kullanılabilir; ancak ortalama performans açısından chain-of-thought yaklaşımı, genel tutarlılığı ile öne çıkmaktadır.

B. Strateji ve Model Bazında Standart Sapma Analizi

Bu bölümde, her bir metrik ve strateji için modeller arası puanların yakınlığı (tutarlılığı), standart sapma üzerinden incelenmiştir. Böylece, strateji ve metrik bazında model performanslarının benzerliği ortaya konmuştur. Standart sapmanın düşük olması, ilgili

stratejiyle elde edilen çıktılarda modeller arasında yüksek tutarlılık var demektir. Yani, stratejilerin model çıktılarına doğrudan etkisi var sonucunu çıkarırız.

Strateji	Doğruluk	Açıklık	Detay	Bağlama Uygunluk
Zero-S.	0	0.5	3.7	0
One-S.	1.2	1.7	10.4	2.3
Few-S.	4.3	6.6	8.6	8.12
CoT	0	1.4	1.2	0
Direct	0	1.1	1.5	0
Indirect	0	0	2	0

Tablo 19: "Soru Cevap" Metriğinde Strateji Bazında Model Standart Sapmalarının İncelenmesi

Strateji	Doğruluk	Açıklık	Detay	Bağlama Uygunluk
Zero-S.	0	1.1	2.4	0
One-S.	0	0.5	0.9	0
Few-S.	0	1.1	4	0
CoT	0	0	1.4	4
Direct	0	0	2.8	0
Indirect	0	0	6.6	1.7

Tablo 20: "Özetleme" Metriğinde Strateji Bazında Model Standart Sapmalarının İncelenmesi

Strateji	Doğruluk	Açıklık	Detay	Bağlama Uygunluk
Zero-S.	0	0	2.1	1.1
One-S.	0	0	2.5	1
Few-S.	0	0	2.5	1
CoT	0	0	0.5	1.2
Direct	0	0	5	1.5
Indirect	0	1	4.3	2.3

Tablo 21: "Doğruluk" Metriğinde Strateji Bazında Model Standart Sapmalarının İncelenmesi

Strateji	Doğruluk	Açıklık	Detay	Bağlama Uygunluk
Zero-S.	0	7.1	8.9	2.5
One-S.	0	1.5	3.9	2.5
Few-S.	0	0.5	2.2	0
CoT	0	0.5	0.5	0
Direct	0	9.2	9	1.5
Indirect	0	0.5	1.1	0

Tablo 22: "Kod Üretimi" Metriğinde Strateji Bazında Model Tutarlılıklarının (Standart Sapma) İncelenmesi

Renk skalası ile görselleştirilen bu tablolarda, **chain-of-thought** stratejisinin tüm metrik ve kriterlerde en açık renge sahip olduğu, yani modeller arası en yüksek tutarlılığın bu strateji altında sağlandığı açıkça görülmektedir. Bu strateji, modelden bağımsız şekilde benzer çıktılar üreterek, farklı büyük dil modellerinde sonuçların yüksek derecede tutarlı ve öngörlülebilir olmasını sağlamıştır. Özellikle few-shot ve one-shot stratejilerinin ise hemen her tabloda en koyu renklere sahip olması, bu stratejilerde modeller arasında anlamlı farklar ve tutarsızlıklar olduğunu göstermektedir.

Sonuçlar, özellikle **chain-of-thought** stratejisini uygulamada tutarlılık ve güvenilirlik açısından öne çıktılığını, model seçiminin ise bu strateji altında belirleyiciliğini yitirdiğini göstermektedir. Genel olarak, strateji seçiminin modellerin çıktı benzerliğinde belirleyici bir faktör olduğu ve chain-of-thought stratejisini tutarlılık açısından öne çıktıgı söylenebilir.

C. Literatüre Katkı

Bu çalışma, büyük dil modellerinde prompt mühendisliğinin performans üzerindeki etkilerini sistematik biçimde karşılaştırarak, literatürde önemli bir boşluğu doldurmaktadır. Öncelikle, benzer araştırmalarda genellikle tek bir model veya sınırlı sayıda strateji ile değerlendirme yapılırken, bu tezde hem farklı model mimarileri (GPT-4o, Claude Sonnet 4, Grok 3, DeepSeek V3) hem de altı temel prompt stratejisi bir arada incelenmiştir. Böylece, yalnızca spesifik bir model ya da görev üzerinden genellenen sonuçların ötesine geçilerek, **strateji-model uyumunun metrik bazında bütüncül bir analizi** sunulmuştur.

Çalışmanın yenilikçi yönlerinden biri, metrik bazında (doğruluk, kod üretimi, özetleme, soru-cevap) **her bir stratejinin görelî etkisinin istatistiksel ve uygulamalı olarak gösterilmesidir**. Böylece araştırmacı ve uygulayıcılar, hangi görev tipinde hangi stratejinin öne çıktığını somut verilerle değerlendirme fırsatı bulmaktadır. Özellikle “chain-of-thought” stratejisini tüm ana metriklerde en yüksek ortalamayı sağlaması, prompt tasarıımı alanında yeni strateji tercihleri için önemli bir referans oluşturmuştur.

Sonuç olarak, bu tez hem büyük dil modelleriyle çalışan araştırmacılara hem de endüstride yapay zekâ tabanlı uygulama geliştirenlere **pratik bir yol haritası ve karşılaştırmalı bir rehber** sağlamaktadır. Elde edilen bulgular, strateji seçiminde rastgelelikten ziyade, görev ve model özelliklerine göre bilinçli bir yaklaşım geliştirilmesi gerektiğine dikkat çekmektedir. Bu yönleriyle tez, prompt mühendisliği ve model değerlendirme alanında literatüre anlamlı bir katkı sunmaktadır.

V. KAYNAKÇA

KİTAPLAR

- Field, A. (2013). **Discovering Statistics Using IBM SPSS Statistics**. Sage Publications.
- Field, A. (2018). **Discovering Statistics Using Python and R**. SAGE Publications.

MAKALELER

- Bansal, T., & Srivastava, S. (2022). **Evaluation Metrics for Language Models: A Survey**. *arXiv preprint arXiv:2203.04890*.
- Demšar, J. (2006). **Statistical Comparisons of Classifiers over Multiple Data Sets**. *Journal of Machine Learning Research*, 7, 1–30.
- Liu, J., Shen, D., & Tan, C. (2023). **Prompt engineering: A systematic survey**. *ACM Computing Surveys (CSUR)*, 55(10), 1-35.
- Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., & Sutskever, I. (2019). **Language models are unsupervised multitask learners**. OpenAI.
- Wei, J., Wang, X., Schuurmans, D., Bosma, M., Ichter, B., Xia, F., ... & Le, Q. V. (2022). **Chain-of-Thought Prompting Elicits Reasoning in Large Language Models**. *arXiv preprint arXiv:2201.11903*.
- Zhao, X., Wallace, E., Feng, S., Klein, D., & Singh, S. (2021). **Calibrate before use: Improving few-shot performance of language models**. *arXiv preprint arXiv:2102.09690*.

ARXİV VE KONFERANS BİLDİRİLERİ

- Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., ... & Amodei, D. (2020). **Language models are few-shot learners**. *arXiv preprint arXiv:2005.14165*.

- Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2018). **BERT: Pre-training of deep bidirectional transformers for language understanding.** *arXiv preprint arXiv:1810.04805*.
- Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). **Efficient estimation of word representations in vector space.** *arXiv preprint arXiv:1301.3781*.
- Radford, A., Narasimhan, K., Salimans, T., & Sutskever, I. (2018). **Improving language understanding by generative pre-training.** OpenAI.
- Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). **Attention is all you need.** In *Advances in Neural Information Processing Systems* (Vol. 30).
- Pennington, J., Socher, R., & Manning, C. D. (2014). **GloVe: Global vectors for word representation.** In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)* (pp. 1532–1543).

TEKNİK RAPORLAR VE DÖKÜMANLAR

- OpenAI. (2023). **GPT-4 Technical Report.** Retrieved from <https://openai.com/research/gpt-4>
- OpenAI. (2024). **Introducing GPT-4o: OpenAI's New Omnimodel.** Retrieved from <https://openai.com/index/gpt-4o>
- Anthropic. (2023). **Claude AI Model Overview.** Retrieved from <https://www.anthropic.com>
- Anthropic. (2023). **Claude: A helpful and harmless assistant.** Retrieved from <https://www.anthropic.com/clause>
- xAI. (2023). **Introducing Grok: A Conversational AI by xAI.** Retrieved from <https://x.ai>
- xAI. (2024). **Introducing Grok: Real-time intelligence.** Retrieved from <https://x.ai/grok>
- DeepSeek. (2024). **DeepSeek LLM Technical Report.** Retrieved from <https://deepseek.com>
- DeepSeek. (2024). **Technical documentation and deep context model.** Retrieved from <https://www.deepseek.com>