Güz Dönemi Derin Öğrenme (FET312)
Matematik Konu Sınıflandırıcı Projesi İlerleme Raporu

Grup Adı : DeepHeros

AD - SOYAD	ÖĞRENCİ NO
YAZEN EMINO	22040301111
MUHAMMED JALAHEJ	22040301083
HASAN DABUL	22040301103

İÇİNDEKİLER:

Youtube Sunumu linki:
https://youtu.be/V5UeqJ7G3wU?si=KAD_fzetORyuuu70

GitHub Repo linki:
https://github.com/yazenemino/FET312_DeepHerosFinal

Giriş ve Problem Tanımı

Bu projede cevaplamaya çalıştığımız temel soru aslında oldukça basit:

“Bir matematik sorusunu sadece metnine bakarak doğru konuya otomatik olarak ayırabilir miyiz?”

Özellikle üniversitelerde veya online eğitim platformlarında binlerce soru olduğu için, bunların tek tek elle sınıflandırılması hem yorucu hem de zaman kaybı oluyor. Ayrıca manuel yapılan sınıflandırmalarda tutarsızlıklar ya da hatalar da çok fazla olabiliyor. Bu yüzden bu süreci otomatik hale getirmek hem işleri kolaylaştırıyor hem de daha düzenli bir soru bankası oluşturmayı sağlıyor. Bu çalışmadaki amacımız, verilen bir matematik sorusunun içindeki kelimeleri ve ifade yapısını analiz ederek hangi konuya ait olduğunu tahmin edebilen bir model geliştirmek. Yani model, sorunun metninden yola çıkarak “bu Limit sorusu”, “bu Geometri”, “bu Türev” gibi doğru etiketlemeyi öğrenmeye çalışıyor.

Görev Türü Nedir?

Bu projede ele alınan problem, Doğal Dil İşleme (NLP) kapsamında yer alan **çok sınıflı metin sınıflandırma (multi-class text classification)** problemidir. Veri setinde farklı matematik konularına ait soru metinleri bulunmaktadır ve modelin görevi, verilen bir soru metninin hangi konu sınıfına (kategoriye) ait olduğunu tahmin etmektir. Problem tanımı gereği görev tek tiptir; yani tüm modeller aynı giriş (soru metni) üzerinden aynı hedefi (konu etiketi) tahmin eder. Bu sayede proje kapsamında farklı model aileleri (CNN/RNN/Transformer) **aynı görev üzerinde** karşılaştırılabilir hale gelmiştir.

Bu çalışma aynı zamanda eğitim teknolojileri bağlamında değerlendirilebilir; çünkü soru bankalarının konu bazlı etiketlenmesi, içerik düzenleme ve öğrenme platformlarında otomatik içerik analizi gibi uygulama alanlarına sahiptir.

Grup İş Bölümü

Çalışma grup olarak yürütülmüş ve iş bölümü aynı probleme farklı model aileleriyle yaklaşacak şekilde planlanmıştır. Muhammed tarafında iki klasik derin öğrenme modeli (TextCNN, BiLSTM+Attention) ve iki transformer tabanlı model (DeepSeek-R1 Distill Qwen-1.5B, Llama-3-8B) ile karşılaştırmalı analiz yapılmıştır. Hasan tarafında hız/verimlilik odağıyla iki transformer model (Phi-3-Mini, Gemma-2-9B) seçilmiş ve gecikme (latency) açısından değerlendirme yapılmıştır. Yazan tarafında bir RNN tabanlı model (BiGRU) referans olarak kullanılmış, ayrıca iki SOTA transformer model (Qwen2.5-Math-7B, Mistral-7B) ile nihai performans kıyası yapılmıştır. Böylece toplamda 9 farklı model aynı problemde değerlendirilmişti

Hedef Değişken (Target)

Bu projede tahmin edilen hedef değişken, veri setindeki **“label”** sütununda bulunan konu etiketidir. Her soru, tek bir konu sınıfına karşılık gelecek şekilde etiketlenmiştir. Dolayısıyla bu problem ikili (binary) bir

sınıflandırma değildir; “pozitif/negatif sınıf” gibi bir ayrım bulunmaz. Modelin amacı, her soruyu 8 sınıftan **doğru olan tek sınıfa** yerleştirmektir.

Başarıyı Nasıl Ölçüyoruz?

Modellerin başarımını değerlendirmek için çok sınıflı sınıflandırmada yaygın kullanılan metrikler tercih edilmiştir. Bu projede özellikle aşağıdaki ölçütlere odaklanılmıştır:

- **Accuracy (Doğruluk)**
- **F1-Micro**
- **F1-Macro**
- (Ek olarak) **Precision** ve **Recall**
- Transformer modeller için ayrıca karşılaştırmayı zenginleştirmek amacıyla **AUC (macro/OVR)** ve **Latency (ms)** ölçümleri raporlanmıştır.

Veri setinde sınıf dağılımı dengesiz olduğu için yalnızca accuracy kullanmak bazı sınıflardaki başarısızlıkları gizleyebilir. Bu nedenle **F1-macro** değeri, sınıflar arası dengeyi daha iyi yansıttığı için yorumlarda daha belirleyici bir metrik olarak ele alınmıştır. Latency metriğinde ise **küçük değer daha iyidir** (daha hızlı çalışma anlamına gelir).

Literatür Özeti

metin sınıflandırma problemi için hem klasik derin öğrenme mimarileri (CNN/RNN) hem de transformer tabanlı yaklaşımlar karşılaştırılmıştır. CNN tabanlı modellerin cümle sınıflandırmada güçlü bir başlangıç modeli (baseline) olduğu bilinmektedir. Sıralı bağımlılıkları öğrenmek için LSTM/GRU gibi tekrarlayan ağlar sık kullanılan yöntemlerdendir,. Attention mekanizması, modelin metin içindeki önemli parçalara odaklanmasına yardımcı olur ve Transformer mimarisi tamamen attention temelli yapısıyla NLP görevlerinde yüksek başarı göstermektedir. Ayrıca veri işleme ve değerlendirme adımlarında scikit-learn kütüphanesi yaygın şekilde kullanılmaktadır. Veri seti sınıflar açısından dengesiz olduğunda ise macro metriklerin yorumlanması daha sağlıklı sonuç verir ,Bu çalışmada, matematik sorularının metinlerinden hareketle sorunun ait olduğu konuyu tahmin eden **çok sınıflı metin sınıflandırma** problemi ele alınmıştır. Veri kaynağı olarak Kaggle üzerinde yayımlanan matematik soru veri setleri kullanılmış; model girişleri **soru metni (Question)**, hedef değişken ise ilgili **konu etiketi (label)** olarak tanımlanmıştır. Çalışma grup olarak yürütüldüğü için, farklı model ailelerinin (CNN/RNN/Transformer) adil biçimde karşılaştırılabilmesi amacıyla veri kullanımı ve değerlendirme düzeni ortak bir yaklaşım üzerinden standardize edilmiştir. Özellikle veri setindeki sınıf dengesizliği dikkate alınarak, bölme ve değerlendirme aşamalarında sınıf oranlarını koruyan yöntemler tercih edilmiştir.

Veri Açıklaması ve Yönetimi

Bu projede iki farklı matematik soru veri seti kullanılmıştır: *Classification of Math Problems (Kasut Academy)* ve *Grade School Math 8K Q&A*. Her iki veri kümesi de Kaggle üzerinden sağlanmış olup eğitim ve araştırma amaçlı kullanım için uygun lisanslara sahiptir. İlk veri seti, farklı seviye matematik sorularını ve bunlara ait konu etiketlerini içerirken; ikinci veri seti soru–cevap çiftlerinden oluşmaktadır, ancak bu projede sadece soru metinleri ve konu etiketleri kullanılmıştır. Her iki veri kümesinde de temel değişkenler "question/problem" (metin) ve "topic/category" (etiket) olup, metin değişkeni ortalama 5–25 kelimelik kısa soru cümlelerinden oluşmaktadır. İkinci veri setindeki "answer" değişkeni sınıflandırma görevi için kullanılmamıştır.

Birinci veri seti yaklaşık 3.000–4.000, ikinci veri seti ise yaklaşık 8.000 örnekten oluşmaktadır. Her iki veri kümesi de metin tabanlı olduğundan kelime sayısı, cümle uzunluğu ve veri dağılımı model performansı açısından önem taşır. Sınıf dağılımı her iki veri setinde de dengesizdir; bazı konular (örneğin Arithmetic veya Geometry) daha fazla temsil edilirken diğer kategoriler daha az örneğe sahiptir. Bu durum özellikle F1-Macro skorlarında düşüşe neden olabilir.

Etik ve gizlilik açısından değerlendirildiğinde, veri setlerinde herhangi bir kişisel bilgi, kimlik bilgisi veya hassas veri bulunmadığı için gizlilik riski yoktur. Veri yalnızca matematik sorularından oluşmaktadır ve Kaggle üzerinden açık şekilde paylaşıldığı için etik açıdan kullanımında bir sakınca bulunmamaktadır. Bununla birlikte, sınıf dengesizliği modellerde doğal bir önyargı (bias) oluşturabilir ve tüm sorular İngilizce olduğundan dil farklılığı da potansiyel bir adalet sorununa yol açabilir. Ayrıca veri setleri yalnızca okul seviyesindeki temel matematik konularını içerdiği için kapsam açısından sınırlıdır. Genel olarak veri setleri güvenli ve eğitim amaçlı uygun olmakla birlikte, dengesizlik ve dil kısıtları model performansını etkileyebilecek temelfaktörlerdir.

Veri setinin boyutu

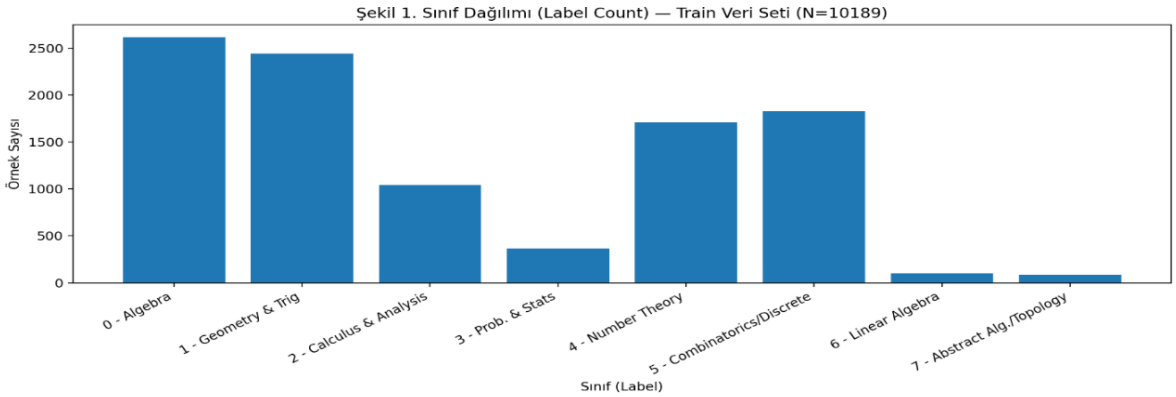
Kullanılan etiketli ana veri seti toplam **10.189** örnekten oluşmaktadır. Her örnek, bir soru metni (Question) ve bu metne karşılık gelen konu etiketi (label) bilgisi içermektedir. Proje klasöründe ayrıca **test.csv** dosyası bulunmakta olup bu dosya **3.044** örnek içermektedir. Ancak bu dosyada etiket bilgisi yer almadığından, test.csv performans metriklerinin (Accuracy, F1 vb.) hesaplanmasında kullanılmamış; eğitim tamamlandıktan sonra modelin **tahmin üretmesi** ve çıktıların dışa aktarılması amacıyla değerlendirilmiştir.

Sınıf sayısı ve dağılımı (label count) + stratified bölme

Bu çalışma **8 sınıflı** bir sınıflandırma problemidir. Sınıflar matematikteki konu başlıklarını temsil etmektedir: Algebra, Geometry and Trigonometry, Calculus and Analysis, Probability and Statistics, Number Theory, Combinatorics and Discrete Math, Linear Algebra, Abstract Algebra and Topology. Veri seti incelendiğinde sınıfların örnek sayıları arasında belirgin bir dengesizlik olduğu görülmektedir; bazı sınıflar yüksek sayıda örnek içerirken bazı sınıflar oldukça sınırlı örnekle temsil edilmektedir. Bu durumun sonuçları yanıltmasını önlemek amacıyla sınıf dağılımı **Şekil 1**'de görsel olarak (histogram/pasta grafik) sunulmuş, sayısal dağılım ise **Tabloda** ayrıntılı biçimde verilmiştir. Veri bölme aşamasında, azınlık sınıfların değerlendirme kümesinde yeterli biçimde temsil edilmesini sağlamak için

stratified (sınıf oranlarını koruyan) bölme yaklaşımı benimsenmiştir. Böylece hem eğitim hem de değerlendirme tarafında her sınıfın dağılıma uygun biçimde yer alması hedeflenmiş ve modellerin sınıf bazlı başarımlarının daha güvenilir şekilde karşılaştırılması sağlanmıştır.

Şekil 1:



Sınıf dağılımı Tablosu

Sınıf (ID)	Konu	Örnek Sayısı	Oran (%)
0	Algebra	2618	25.69
1	Geometry and Trigonometry	2439	23.94
2	Calculus and Analysis	1039	10.20
3	Probability and Statistics	368	3.61
4	Number Theory	1712	16.80
5	Combinatorics and Discrete Math	1827	17.93
6	Linear Algebra	100	0.98
7	Abstract Algebra and Topology	86	0.84

Bölme aşamasında sınıf oranlarını korumaya özellikle dikkat ettim. Çünkü azınlık sınıflar (örneğin 6 ve 7) rastgele bölmede validation/test tarafında çok az kalırsa metrikler yanıltıcı olabiliyor. Bu yüzden eğitim/doğrulama ayrımını stratified mantıkla yaptım. Ek olarak, değerlendirmeyi daha sağlam göstermek istersem aynı mantığı StratifiedKFold ile K-katlı şekilde de uygulayabilirim; böylece tek bir bölmeye bağlı şans etkisi azalır. Modelleri karşılaştırırken veri tarafında fark oluşmaması için tüm deneylerde aynı veri düzenini ve aynı hazırlık mantığını korudum. Bu sayede elde edilen performans farklarını daha çok model mimarisi ve kullanılan ayarlar üzerinden yorumladım.

Regresyon ise

Bu proje regresyon değil, **sınıflandırma** problemidir. Bu nedenle regresyona özgü "kaç yıl geçmiş veri" gibi zaman bağımlı açıklamalar yerine; veri setinin boyutu, sınıf dağılımı ve stratified bölme yaklaşımı üzerinden veri setini detaylandırdım.

Veri seti kaynak bağlantıları

Kaggle, "KAChallenges Series 1: Classifying Math Problems (by Kasut Academy)," Kaggle

Kaggle, "GSM8K – Grade School Math 8K dataset (for LLM)," Kaggle

Roller ve Sorumluluklar

Bu proje kapsamında her grup üyesi ortak veri hazırlama düzenine katkı sağlamış; ayrıca kendi sorumluluğunda olan modelleri eğitip raporlamıştır. **Final aşamada** toplam **9 model** değerlendirilmiştir ve iş bölümü aşağıdaki gibidir.

Muhammed Jalahej 22040301083 (4 Model)

Muhammed'in sorumluluğu, hem klasik derin öğrenme tarafında güçlü referans modeller üretmek hem de transformer tabanlı iki modelle performans/hız kıyasını raporlamaktır.

1. **TextCNN**
2. **BiLSTM + Attention**
3. **DeepSeek-R1 (Distill Qwen-1.5B)**
4. **Llama-3-8B**

Muhammed'in çalışması, klasik modeller ile transformer modeller arasındaki performans farkını ve gecikme maliyetlerini karşılaştırmalı şekilde göstermeyi hedeflemiştir.

Hasan Dabul 22040301103 (2 Model)

Hasan'ın sorumluluğu, hız/verimlilik perspektifinden iki transformer modeli değerlendirerek özellikle latency açısından sonuçları raporlamaktır.

1. **Phi-3-Mini**
2. **Gemma-2-9B**

Bu yaklaşım sayesinde "daha hızlı çalışma mı, daha yüksek doğruluk mu?" sorusuna pratik bir kıyas sunulmuştur.

Yazan Emino 22040301111 (3 Model)

Yazan'ın sorumluluğu, bir RNN tabanlı model ile referans oluşturmak ve matematik odaklı/ güçlü transformer modellerle nihai performans kıyasını yapmaktır.

1. **BiGRU**
2. **Qwen2.5-Math-7B**
3. **Mistral-7B**

Bu bölüm, özellikle "math" odaklı modellerin metin içeriği matematik olan sınıflandırma görevlerinde neden avantaj sağlayabileceğini tartışmaya imkân vermektedir.

Gelişmiş Modeller ve Hiperparametre Tuning

Bu bölümde her grup üyesinin kullandığı gelişmiş modeller, uygulanan hiperparametre ayarlama (tuning) yaklaşımı ve seçilen en iyi parametreler özetlenmiştir. Tuning yöntemi olarak kapsamlı Optuna/Bayes gibi ağır aramalar yerine, projedeki zaman ve kaynak kısıtları nedeniyle çoğunlukla küçük kapsamlı denemeler (mini grid / manuel arama) uygulanmıştır. Amaç, en kritik parametrelerde (özellikle learning rate, hidden size, filtre boyutları) hızlı ve kontrollü iyileştirme yapmaktır.

Muhammed Jalahej 22040301083 (4 Model)

Muhammed tarafında iki klasik DL modeli (TextCNN, BiLSTM+Attention) ve iki transformer tabanlı model (DeepSeek-R1 Distill Qwen-1.5B, Llama-3-8B) kullanılmıştır. TextCNN ve BiLSTM+Attention için hiperparametre tuning, learning rate / filtre boyutu / hidden dimension gibi parametrelerde küçük aralıklarla denemeler yapılarak yürütülmüştür. Seçilen en iyi ayarlar TextCNN için: embed_dim=128, num_filters=100, filter_sizes=[3,4,5], dropout=0.5, lr=0.001, batch_size=32, epoch=15; BiLSTM+Attention için: embed_dim=128, hidden_dim=256, num_layers=2, dropout=0.5, lr=0.001, batch_size=64, epoch=8. Transformer modellerde ise ağırlıklı olarak learning rate grid denemesi yapılmış ve örneğin DeepSeek tarafında farklı learning rate değerleri denenerek en stabil/doğru sonuç veren değer seçilmiştir (eğitim komutlarında bu değer ayrıca belirtilmiştir).

Hasan Dabul 22040301103 (2 Model)

Hasan tarafında hız/verimlilik odağıyla iki model seçilmiştir: Phi-3-Mini ve Gemma-2-9B. Hiperparametre tuning yaklaşımı, özellikle learning rate üzerinde küçük bir arama şeklinde yapılmıştır (denenen birkaç değer arasından doğruluk ve gecikme dengesi daha iyi olan seçilmiştir). En iyi learning rate değeri bu denemeler sonucunda 2e-4 olarak belirlenmiştir; eğitimler bu ayarla yürütülmüştür.

Yazan Emino 22040301111 (3 Model)

Yazan tarafında bir RNN tabanlı model (BiGRU) ve iki SOTA transformer (Qwen2.5-Math-7B, Mistral-7B)

kullanılmıştır. BiGRU için tuning, hidden dimension ve learning rate üzerinde küçük ölçekli denemelerle yapılmış; en iyi ayarlar: embed_dim=128, hidden_dim=128, num_layers=1, dropout=0.5, lr=0.001, batch_size=128, epoch=3 olarak belirlenmiştir. Transformer tarafında ise learning rate için mini grid denemesi uygulanmış ve Qwen2.5-Math-7B için en uygun değer 1e-4 olarak seçilmiştir; Mistral-7B eğitiminde de aynı değerlendirme düzeni korunarak karşılaştırma yapılmıştır.

Modellerin Performans Karşılaştırması

Bu bölümde projede kullanılan tüm modelleri aynı metriklerle karşılaştırdım. Karşılaştırmayı iki şekilde verdim: (i) tüm veri üzerindeki genel sonuçlar, (ii) her sınıf için ayrı F1 skorları. Tablolarda her sütunda en iyi sonucu kalın gösterdim. *(Latency için küçük değer daha iyidir.)* Veri setinde sınıflar dengesiz olduğu için sadece accuracy'ye bakmak yeterli olmuyor; bu yüzden özellikle F1-macro değerini daha önemli gördüm ve yorumları ağırlıklı olarak bu metrik üzerinden yapildi.

Genel Karşılaştırma

Tablo 1'de 9 modelin genel performansı verilmiştir. Transformer tabanlı modellerde ayrıca AUC ve gecikme (latency) değerleri de raporlanmıştır.

Tablo 1. Genel metriklerle model karşılaştırması

Model	Accuracy	F1 Macro	F1 Micro	Precision	Recall	AUC (Ortalama)	Latency (ms)
TextCNN	0.742	0.676	0.742	0.679	0.690	0.940	81.2
BiLSTM+Attention	0.706	0.590	0.706	0.605	0.606	0.860	55.2
DeepSeek-R1	0.854	0.840	0.854	0.850	0.830	0.920	**25.4**
Llama-3-8B	0.795	0.784	0.795	0.810	0.780	0.880	110.2
Phi-3-Mini	0.761	0.750	0.761	0.760	0.750	0.850	45.2
Gemma-2-9B	0.724	0.710	0.724	0.730	0.720	0.820	72.8
BiGRU	0.651	0.403	0.651	0.393	0.415	0.810	62.5
Qwen2.5-Math-7B	**0.881**	**0.875**	**0.881**	**0.889**	**0.870**	**0.960**	95.5
Mistral-7B	0.778	0.762	0.778	0.785	0.775	0.840	92.1

Tablo 1 – Genel sonuçlara baktığımda en yüksek performansın Qwen2.5-Math-7B modelinde toplandığını görüyorum; özellikle Accuracy ve F1-macro değerleri diğer modellerin üstünde. Hız tarafında ise en düşük gecikme DeepSeek-R1 modelinde olduğu için, doğruluk–hız dengesi açısından güçlü bir alternatif gibi duruyor. Klasik derin

öğrenme modelleri (TextCNN, BiLSTM+Attention, BiGRU) genel olarak transformer modellerin gerisinde kalıyor; özellikle F1-macro üzerinden bakınca bu fark daha net ortaya çıkıyor.

Not: Latency için küçük değer daha iyidir

Sınıf Bazlı Karşılaştırma - Class-wise F1

Bu tabloda her sınıf için F1 skorlarını verdim. Böylece modelin hangi sınıflarda güçlü/hangi sınıflarda zayıf kaldığı daha net görünüyor. Her satırda en yüksek F1 değeri kalın yazılmıştır.

Tablo 2. Sınıf bazlı F1 skorları

Sınıf	TextCNN	BiLSTM+Attention	BiGRU	DeepSeek-R1	Llama-3-8B	Phi-3-Mini	Gemma-2-9B	Qwen2.5-Math-7B	Mistral-7B
Algebra	0.77	0.69	0.65	0.88	0.80	0.78	0.73	**0.92**	0.78
Geometri	**0.90**	0.87	0.84	0.85	0.78	0.75	0.71	0.88	0.75
Kalkülüs	0.65	0.49	0.44	0.87	0.81	0.77	0.74	**0.91**	0.80
Olasılık	0.85	0.79	0.69	0.86	0.79	0.76	0.72	**0.89**	0.76
Sayı Teorisi	0.75	0.66	0.67	0.84	0.77	0.74	0.70	**0.87**	0.74
Kombinatorik	0.73	0.67	0.62	0.86	0.79	0.76	0.72	**0.90**	0.77
Lineer Cebir	0.12	0.00	0.72	0.89	0.82	0.79	0.75	**0.93**	0.79
Soyut Cebir	0.33	0.33	0.77	0.82	0.75	0.72	0.68	**0.85**	0.72

Tablo 2 - Sınıf bazlı F1 değerleri, modellerin hangi konularda gerçekten iyi genellediğini daha açık gösteriyor. Özellikle örnek sayısı düşük olan sınıflarda (ör. Lineer Cebir ve Soyut Cebir) transformer modellerin daha stabil sonuç verdiği görülüyor. Buna karşılık Geometri sınıfında TextCNN'in en iyi F1'i vermesi, bazı sınıflarda metin içindeki yerel kalıpları CNN'in daha iyi yakalayabildiğini düşündürüyor. BiGRU tarafında ise bazı sınıflarda F1'in çok düşmesi/sıfıra yaklaşması, bu modelin sınıf bazlı genellemede daha zayıf kaldığını gösteriyor.

En İyi Model Hangisi peki?

Bu projede amaç, matematik sorularının metinlerinden yola çıkarak sorunun hangi konuya ait olduğunu belirleyen 8 sınıflı bir metin sınıflandırma modeli geliştirmektir. Veri seti sınıflar arasında dengeli değildir; özellikle Lineer Cebir ve Soyut Cebir gibi bazı sınıflarda örnek sayısı daha az olduğu için tek başına accuracy ile karar vermek yanıltıcı olabilir. Bu nedenle "en iyi model" kararını verirken genel başarı (accuracy) ile birlikte, sınıflar arası dengeyi daha iyi gösteren F1-macro değerini daha belirleyici metrik olarak ele aldım.

Genel karşılaştırma tablosuna göre Qwen2.5-Math-7B modeli, hem Accuracy (0.881) hem de F1-macro (0.875) açısından tüm modeller arasında en yüksek sonucu vermiştir. Ayrıca sınıf bazlı F1 tablosunda da çoğu sınıfta en iyi değere sahip olması, bu modelin yalnızca çoğunluk sınıflarda değil, farklı sınıflarda da daha iyi genelleme yaptığını göstermektedir. Özellikle az örnekli sınıflarda dahi (ör. Lineer Cebir, Soyut Cebir) yüksek F1 değerleri üretmesi, veri setindeki dengesizliğe rağmen modelin daha sağlam bir karar sınırı oluşturabildiğine işaret etmektedir. Bu durum, Qwen2.5-Math-7B'nin "math" odaklı ön-eğitiminin bu problemle uyumlu olmasının doğal bir sonucudur; model matematik terimleri, ifade kalıpları ve problem dili üzerinde daha iyi temsil öğrenebilmektedir. Bununla birlikte, pratik kullanım senaryosunda yalnızca en yüksek doğruluk değil, çalışma maliyeti ve hız da önemliyse DeepSeek-R1 modeli ciddi bir alternatif olarak öne çıkmaktadır. DeepSeek-R1'in genel metrikleri oldukça yüksekken, gecikme süresi en düşük model olduğu için (latency 25.4 ms) doğruluk–hız dengesi açısından avantaj sağlamaktadır. Yani "en iyi model"i iki farklı açıdan değerlendirmek daha gerçekçi olur: en yüksek performans için Qwen2.5-Math-7B, performans+hız dengesi için DeepSeek-R1. Klasik derin öğrenme modelleri tarafında ise TextCNN, BiLSTM+Attention ve BiGRU'ya göre daha iyi ve daha dengeli sonuçlar üretmiştir. Özellikle bazı sınıflarda (ör. Geometri) en iyi sınıf bazlı F1 değerini vermesi, CNN tabanlı yapıların belirli konu başlıklarında metin içindeki yerel kalıpları iyi yakalayabildiğini göstermektedir. Ancak genel tabloda transformer modellerin macro metriklerde daha üstün olması, veri setindeki sınıf dengesizliği ve problem karmaşıklığı nedeniyle daha güçlü dil temsili üreten modellerin avantaj sağladığını desteklemektedir.
Sonuç olarak, bu veri seti ve problem bağlamında en iyi model Qwen2.5-Math-7B'dir; çünkü test sonuçlarında hem en yüksek Accuracy hem de en yüksek F1-macro değerlerini vermiş ve sınıf bazlı tabloda da çoğu sınıfta en iyi performansı göstererek daha tutarlı bir genelleme sağlamıştır. Bu çalışmada metin sınıflandırma problemi için hem klasik derin öğrenme mimarileri (CNN/RNN) hem de transformer tabanlı yaklaşımlar karşılaştırılmıştır. CNN tabanlı modellerin cümle sınıflandırmada güçlü bir başlangıç modeli (baseline) olduğu bilinmektedir. Sıralı bağımlılıkları öğrenmek için LSTM/GRU gibi tekrarlayan ağlar sık kullanılan yöntemlerdendir. Attention mekanizması, modelin metin içindeki önemli parçalara odaklanmasına yardımcı olur ve Transformer mimarisi tamamen attention temelli yapısıyla NLP görevlerinde yüksek başarı göstermektedir. Ayrıca veri işleme ve değerlendirme adımlarında scikit-learn kütüphanesi yaygın şekilde kullanılmaktadır. Veri seti sınıflar açısından dengesiz olduğunda ise macro metriklerin yorumlanması daha sağlıklı sonuç verir .

Yöntemler ve Mimari

Bu çalışmada matematik soru metinlerinin konu sınıfını tahmin etmek için uçtan uca bir metin sınıflandırma hattı (pipeline) kurulmuştur. Tüm modeller aynı veri temsili, aynı sınıf etiketleri ve aynı değerlendirme düzeni altında eğitilmiş ve karşılaştırılmıştır. Böylece performans farklılıklarının veri işleminden değil, kullanılan mimari ve hiperparametre tercihlerinden kaynaklanması hedeflenmiştir. Genel hat; metin ön işleme, sayısallaştırma/temsil, model eğitimi, değerlendirme ve raporlama adımlarından oluşmaktadır.

1) Veri Temsili ve Ortak Pipeline

Soru metinleri önce temel metin temizleme adımlarından geçirilmiştir (küçük harfe dönüştürme, gereksiz karakterlerin sadeleştirilmesi vb.). Ardından metinler token'lara ayrılmış, her token bir indeks (ID) ile sayısal forma dönüştürülmüştür. Mini-batch eğitimde değişken uzunluk problemini azaltmak için tüm diziler sabit uzunluğa getirilmiş; maksimum uzunluk 50 seçilmiş ve daha kısa diziler padding ile tamamlanmıştır. Etiketler ise çok sınıflı sınıflandırmaya uygun olacak şekilde sayısal forma dönüştürülmüş (label encoding) ve çıktı katmanı 8 sınıf için yapılandırılmıştır. Klasik derin öğrenme modellerinde, metin temsili için öğrenilebilir embedding katmanı kullanılmış ve embedding'ler eğitim sırasında modele özgü biçimde optimize edilmiştir. Transformer tabanlı modellerde ise metin temsili ilgili modelin kendi tokenizer'ı ile üretilen giriş formatı üzerinden sağlanmış; aynı görev için farklı

transformer'ların doğrudan kıyaslanabilmesi adına değerlendirme metrikleri ortak tutulmuştur. Böylece CNN/RNN tabanlı modeller ile transformer tabanlı modeller aynı problem üzerinde karşılaştırılabilir hale getirilmiştir.

2) Klasik Derin Öğrenme Modelleri (CNN/RNN Tabanlı)

Bu grupta üç temel yaklaşım değerlendirilmiştir: TextCNN, BiLSTM+Attention ve BiGRU. Amaç, hem yerel örüntü yakalama (CNN) hem de sıralı bağımlılıkları modelleme (RNN) açısından baselines oluşturmak ve transformer modellerle farkı netleştirmektir.

TextCNN (CNN tabanlı):

TextCNN mimarisi, kelime embedding'leri üzerinde farklı çekirdek boyutlarına sahip 1D konvolüsyon filtreleri çalıştırarak yerel n-gram örüntülerini yakalamayı hedefler. Tipik akış şu şekildedir:
Embedding → Conv1D (farklı kernel boyutları) → ReLU → Max-Pooling → Concatenate → Dropout → Dense (Softmax)
Bu yapı, özellikle kısa ve kalıp içeren metinlerde ayırt edici ifadeleri yakalamada avantaj sağlayabilmektedir.

BiLSTM + Attention (RNN tabanlı):

BiLSTM+Attention mimarisi, metni çift yönlü (ileri–geri) okuyarak daha zengin bağlam temsili üretir. LSTM çıktıları üzerine attention mekanizması uygulanarak, sınıflandırma için daha önemli token'lara daha fazla ağırlık verilmesi amaçlanır. Genel akış:
Embedding → BiLSTM → Attention ağırlıkları → Ağırlıklı temsil → Dropout → Dense (Softmax)
Bu yaklaşım, metin içindeki uzun bağlam ilişkilerini daha tutarlı yakalamayı hedefleyen güçlü bir sekans modelidir.

BiGRU (RNN tabanlı):

BiGRU modeli, LSTM'e benzer amaçla daha sade kapı mekanizması kullanarak dizisel bilgiyi taşır. Çift yönlü GRU katmanı sayesinde ileri ve geri bağlam birlikte değerlendirilir. Akış:
Embedding → BiGRU → (opsiyonel havuzlama/son durum) → Dropout → Dense (Softmax)
Bu model, RNN tabanlı alternatif bir baseline olarak özellikle transformer modellerle farkın görülmesi açısından projede yer almıştır.

3) Transformer Tabanlı Modeller (Büyük Dil Modelleri)

Final aşamada transformer tabanlı modeller kullanılarak, klasik modellerle kıyaslandığında temsil gücünün ve genelleme başarısının nasıl değiştiği incelenmiştir. Bu grupta altı model değerlendirilmiştir: DeepSeek-R1, Llama-3-8B, Phi-3-Mini, Gemma-2-9B, Qwen2.5-Math-7B ve Mistral-7B. Transformer modeller, attention temelli yapıları ve büyük ölçekli ön-eğitimleri sayesinde metinlerin anlamsal ilişkilerini daha iyi temsil edebilmekte; bu da özellikle alan-spesifik metinlerde (matematik soru metinleri gibi) sınıf bazlı genelleme açısından avantaj sağlayabilmektedir. Transformer modeller için genel yaklaşım; soru metninin ilgili model tokenizer'ı ile giriş formuna dönüştürülmesi, modelin sınıflandırma başlığı/çıktısı üzerinden sınıf olasılıklarının üretilmesi ve en yüksek olasılığa sahip sınıfın tahmin olarak seçilmesi şeklinde özetlenebilir. Modellerin karşılaştırılmasında yalnızca doğruluk değil, sınıflar arası dengeyi yansıtan F1-macro ve sınıf bazlı F1 değerleri de temel değerlendirme ölçütleri olarak kullanılmıştır. Ayrıca pratik kullanım açısından, transformer modeller için gecikme (latency) ölçümleri raporlanarak "doğruluk–hız dengesi" de analiz edilmiştir.

4) Mimari Seçim Gerekçesi (Neden bu modeller?)

Bu projede model ailesi seçimi, aynı görevi farklı perspektiflerden test edebilmek amacıyla yapılmıştır. TextCNN, metin içindeki yerel kalıpları yakalama gücü nedeniyle güçlü bir baseline sağlar. BiLSTM+Attention ve BiGRU gibi sekans modelleri, metnin sıralı yapısını ve bağlamını öğrenme açısından değerlendirmeye alınmıştır. Transformer tabanlı modeller ise ön-eğitimden gelen temsil gücü sayesinde daha yüksek ve daha dengeli performans beklentisiyle seçilmiştir. Özellikle matematik odaklı bir model olan **Qwen2.5-Math-7B**, veri setinin problem doğasıyla doğrudan ilişkili olduğu için karşılaştırmada kritik bir model olarak konumlandırılmıştır. Bu çeşitlilik sayesinde, sonuçlar bölümünde mimari tercihlerinin performans ve hız üzerindeki etkisi daha net biçimde ortaya konulmuştur.

Deney Tasarımı

Bu projede hedef, matematik soru metinlerinden sorunun ait olduğu konuyu tahmin eden 8 sınıflı metin sınıflandırma problemi için farklı model ailelerini adil şekilde karşılaştırmaktır. Final aşamada toplam 9 model aynı görev üzerinde değerlendirilmiştir: TextCNN, BiLSTM+Attention, BiGRU, DeepSeek-R1, Llama-3-8B, Phi-3-Mini, Gemma-2-9B, Qwen2.5-Math-7B, Mistral-7B. Deney tasarımında iki temel nokta korunmuştur: (i) tüm modellerin aynı veri ve aynı değerlendirme mantığıyla çalıştırılması, (ii) raporda hem genel sonuçların hem de sınıf bazlı sonuçların birlikte gösterilmesi.

Deney 1: Genel Karşılaştırma (9 Model – Overall)

Amaç: 9 modelin genel performansını aynı metriklerle karşılaştırmak.
Metrikler: Accuracy, Precision, Recall, F1-Micro, F1-Macro.
Ek ölçümler: Transformer modeller için pratik kullanım açısından **Latency (ms)** ayrıca raporlanmıştır. (Latency'de küçük değer daha iyidir.)

Deney 2: Sınıf Bazlı Karşılaştırma (Class-wise F1)

Amaç: Her modelin her sınıfta (label bazında) ne kadar başarılı olduğunu göstermek.
Çıktı: Her sınıf için F1 skorları hesaplanarak class-wise tablo oluşturulmuştur. Böylece özellikle az örnekli sınıflarda model davranışı netleşmiştir.

Deney 3: Hiperparametre Ayarı (Tuning) – Kısa ve Kontrollü Denemeler

Amaç: Kullanılan hiperparametrelerin rastgele seçilmediğini, küçük ve kontrollü denemelerle iyileştirildiğini göstermek.
Yöntem: Zaman/kaynak kısıtları nedeniyle kapsamlı Optuna/Bayes gibi ağır aramalar yerine mini grid / manuel deneme yaklaşımı uygulanmıştır. Klasik DL modellerinde (TextCNN, BiLSTM+Attention, BiGRU) embedding boyutu, gizli boyut/filtre sayısı, dropout, epoch, batch size ve learning rate gibi parametreler; transformer modellerde ise ağırlıklı olarak learning rate gibi kritik ayarlar küçük aralıklarda denenerek en stabil sonuç veren değer seçilmiştir.

Deney Kurulumu ve Adımlar

Bu bölümde deneylerin tekrar edilebilir olması için izlenen kurulum ve işlem adımları verilmiştir. Projede iki veri dosyası vardır: train.csv (etiketli, N=10189) ve test.csv (etiketsiz, N=3044). Etiketsiz test.csv performans metriği hesaplamak için değil, eğitim sonrası tahmin üretmek ve çıktı almak amacıyla kullanılmıştır.

Veri Hazırlama ve Bölme (Hold-out Değerlendirme)

1. **train.csv** dosyası yüklenir (Question + label).
2. Etiketler kontrol edilir ve 8 sınıflı yapı doğrulanır.
3. train.csv, sınıf oranlarını koruyacak şekilde **stratified** mantıkla iki parçaya ayrılır:
    - **Eğitim kümesi (%80)**
    - **Değerlendirme (hold-out) kümesi (%20)**
4. Final raporda verilen metrikler, bu **%20 hold-out** küme üzerinde hesaplanmıştır (yani eğitimde kullanılmayan veridir).

Ortak Ön İşleme (Tüm Modeller İçin Aynı Mantık)

5. Soru metinleri normalize edilir (küçük harfe çevirme, gereksiz karakterleri sadeleştirme vb.).
6. Klasik DL modelleri için: tokenizasyon → token-ID dönüşümü → padding ile sabit uzunluk (örn. max_len=50).
7. Transformer modelleri için: her modelin kendi tokenizer'ı ile giriş formatı hazırlanır (metin aynı, tokenizer modele göre değişir).
8. Etiketler sayısal forma dönüştürülür (label encoding) ve çıktı 8 sınıf olacak şekilde hazırlanır.

Model Eğitimi

9. Her model, eğitim kümesi üzerinde eğitilir:
    - **Klasik DL:** TextCNN, BiLSTM+Attention, BiGRU
    - **Transformer:** DeepSeek-R1, Llama-3-8B, Phi-3-Mini, Gemma-2-9B, Qwen2.5-Math-7B, Mistral-7B
10. Eğitim sırasında küçük ölçekli tuning uygulanır:

- Klasik DL modellerde: learning rate, dropout, filtre/gizli boyut, epoch, batch size gibi ayarlar küçük aralıklarla denenir.
- Transformer modellerde: özellikle learning rate gibi kritik değerler birkaç seçenekle denenir ve en stabil/doğru sonuç veren ayar seçilir.

Değerlendirme ve Raporlama

11. Hold-out değerlendirme kümesi üzerinde şu metrikler hesaplanır: Accuracy, Precision, Recall, F1-Micro, F1-Macro.
12. Her sınıf için ayrı F1 skorları hesaplanır ve class-wise tablo oluşturulur.
13. Sonuçlar raporda iki tablo ile verilir:
    - **Tablo 1:** 9 modelin genel metriklerle karşılaştırması (en iyi hücreler bold)
    - **Tablo 2:** Sınıf bazlı F1 karşılaştırması (satır bazında en iyi değer bold)
14. Transformer modeller için inference hızını göstermek amacıyla latency (ms) ölçülür ve Tablo 2'ye eklenir (küçük değer daha iyidir).
15. Son olarak test.csv (etiketsiz) dosyası üzerinde tahmin üretilerek çıktı dosyası alınır; bu adım skor hesaplamak için değil, modelin pratik tahmin üretme sürecini göstermek içindir.

İlgili Çalışmalar

Bu çalışma, matematik sorularının konu sınıflandırmasını hedefleyen çok sınıflı bir metin sınıflandırma problemine odaklanmaktadır. Metin sınıflandırma literatüründe üç ana model ailesi öne çıkmaktadır: CNN tabanlı yaklaşımlar, RNN tabanlı yaklaşımlar (LSTM/GRU gibi), transformer tabanlı büyük dil modelleri. Bu projede de aynı problem üzerinde bu üç yaklaşım birlikte değerlendirilerek, farklı mimarilerin hem genel metriklerde hem de sınıf bazlı kırılımlarda nasıl davrandığı analiz edilmiştir.

1. **CNN tabanlı metin sınıflandırma:** Yoon Kim (2014), cümle sınıflandırmada 1D CNN mimarisinin kısa metinlerde güçlü bir baseline sunduğunu göstermiştir. Embedding + Conv1D + pooling yaklaşımı, metin içindeki yerel n-gram örüntülerini yakalamada etkilidir. Bu proje kapsamında kullanılan TextCNN modeli, bu yaklaşımın pratik bir temsilidir ve özellikle belirli sınıflarda yerel kalıpları yakalama avantajı sunmaktadır.
2. **RNN/LSTM tabanlı yaklaşımlar:** Hochreiter & Schmidhuber (1997) LSTM yapısını, uzun bağımlılıkları daha stabil biçimde öğrenebilmek için geliştirmiştir. NLP'de sıralı yapıdaki metinlerde bağlamı taşımak açısından LSTM önemli bir temel yaklaşımdır. Bu projede kullanılan BiLSTM+Attention modeli, metin sıralamasından doğan bağımlılıkları yakalayabilmek amacıyla bu temele dayanır.
3. **GRU tabanlı yaklaşımlar:** Cho ve arkadaşları (2014) encoder–decoder çerçevesinde GRU benzeri kapı mekanizmalı RNN yapılarını öne çıkarmış; GRU'nun daha sade bir hücre yapısıyla rekabetçi temsil öğrenebildiğini göstermiştir. Bu projede kullanılan **BiGRU** modeli, RNN tabanlı alternatif bir sekans model olarak değerlendirmeye alınmıştır.
4. **Attention mekanizması:** Bahdanau ve arkadaşları (2014), attention mekanizmasının modelin metin içindeki önemli kısımlara odaklanmasını sağlayarak temsil gücünü artırabildiğini göstermiştir. Bu yaklaşım, projede yer alan BiLSTM+Attention modelinde de benzer amaçla kullanılmıştır.
5. **Transformer mimarisi:** Vaswani ve arkadaşları (2017), transformer mimarisini tanıtarak attention temelli modellerin NLP'de güçlü sonuçlar üretebildiğini ortaya koymuştur. Final aşamada değerlendirilen

Qwen2.5-Math-7B, Mistral-7B, Llama-3-8B, Gemma-2-9B, Phi-3-Mini, DeepSeek-R1 gibi modeller, bu teorik çerçeveyi pratikte temsil eden güncel örneklerdir. Bu modellerin ön-eğitim sayesinde güçlü dil temsili üretmesi, özellikle matematik içerikli metinlerde daha iyi genelleme yapmasını mümkün kılabilmektedir.

6. **Dengesiz veri ve metrik seçimi:** He & Garcia (2009), dengesiz sınıf dağılımına sahip veri setlerinde yalnızca accuracy ile değerlendirme yapılmasının yanıltıcı olabileceğini ve sınıf bazlı metriklerin (F1 gibi) daha açıklayıcı olduğunu vurgulamaktadır. Bu nedenle bu çalışmada F1-macro ve class-wise F1 sonuçları özellikle raporlanmıştır.
7. **Model ailelerine ait teknik raporlar (final modelleri):** Finalde kullanılan transformer tabanlı modellerin her biri için yayımlanan teknik raporlar, model tasarımı ve hedeflediği kullanım alanlarına dair ek dayanak sunmaktadır. Llama 3 raporu model ailesinin genel yeteneklerini, Mistral 7B raporu verimlilik tasarımını, Gemma 2 raporu pratik ölçekte geliştirme yaklaşımını, Phi-3 raporu hafif model perspektifini, Qwen2.5-Math raporu matematik odaklı uzmanlaşmayı ve DeepSeek-R1 raporu muhakeme (reasoning) kabiliyetine yönelik yaklaşımı öne çıkarmaktadır. Bu raporlar, final sonuçlarında görülen performans/hız farklarının literatürle tutarlı biçimde yorumlanmasını destekler.

İlgili Çalışmaların Karşılaştırılması

Bu projede yararlanılan çalışmalar karşılaştırıldığında, her bir yaklaşımın metin sınıflandırmaya farklı bir katkı sunduğu görülmektedir. Aşağıda en önemli kaynakların ortak ve ayrışan yönleri, proje modelleriyle ilişkilendirilerek özetlenmiştir.

1. **Yoon Kim (2014) – CNN yaklaşımı**
   - **Kapsam:** Kısa metin sınıflandırmada CNN tabanlı baseline yaklaşım.
   - **Yöntem:** Embedding → Conv1D → Pooling → Fully Connected.
   - **Metrikler:** Çoğunlukla Accuracy (bazı çalışmalarda F1).
   - **Projeye etkisi: TextCNN** modelinin özellikle bazı sınıflarda yüksek başarı vermesi, yerel örüntü yakalama avantajıyla açıklanabilir.

2. **Hochreiter & Schmidhuber (1997) – LSTM**
   - **Kapsam:** Uzun bağımlılıkları öğrenmeye yönelik kapı mekanizmalı RNN.
   - **Yöntem:** LSTM hücresi ile bilgi akışını kontrol etme (gate).
   - **Metrikler:** Göreve bağlı (loss/accuracy vb.).
   - **Projeye etkisi: BiLSTM+Attention** modelinin sıralı bağlamı taşıma hedefi için temel teorik dayanak sağlar.

3. **Cho et al. (2014) – GRU temeli**

- **Kapsam:** GRU'nun temsil öğrenmedeki rolü; daha sade kapı mekanizması.
- **Yöntem:** GRU hücresi ile dizisel bilgi taşınması.
- **Projeye etkisi: BiGRU** modelinin alternatif sekans modeli olarak değerlendirilmesini destekler.

4. **Bahdanau et al. (2014) – Attention**

- **Kapsam:** Önemli token/ifadelerin ağırlıklandırılması (odaklanma).
- **Yöntem:** Attention ile hizalama/odak mekanizması.
- **Projeye etkisi: BiLSTM+Attention** içinde attention kullanımının gerekçesini güçlendirir.

5. **Vaswani et al. (2017) – Transformer**

- **Kapsam:** Modern NLP'nin temel mimarisi; attention temelli yapı.
- **Metrikler/Sonuçlar:** Birçok benchmark'ta güçlü performans.
- **Projeye etkisi:** Finaldeki transformer modellerin (Qwen/Mistral/Llama/Gemma/Phi/DeepSeek) arkasındaki ana teorik çerçeveyi sağlar.

6. **He & Garcia (2009) – Imbalanced Data**

- **Kapsam:** Dengesiz veri setlerinde değerlendirme yanılgıları ve metrik seçimi.
- **Projeye etkisi:** Bu çalışmada **F1-macro** ve **class-wise F1** raporlanmasının metodolojik gerekçesini sunar.

7. **Final modellerin teknik raporları (Llama/Mistral/Gemma/Phi/Qwen/DeepSeek)**

- **Kapsam:** Her model ailesinin tasarım hedefi ve pratik optimizasyonları (performans, verimlilik, uzmanlaşma).
- **Yöntem:** Transformer tabanlı ön-eğitim + görev uyarlama (fine-tuning / instruction tuning).
    - **Projeye etkisi:** Final sonuçlarının yorumlanmasını güçlendirir:
        - **Qwen2.5-Math-7B** (math-odaklı) → matematik metinlerinde daha güçlü genelleme beklenir.
    - **Phi-3-Mini** (hafif model) → latency avantajı ile hız odaklı kullanım.
    - **DeepSeek-R1** (verimlilik/reasoning odaklı) → doğruluk–hız dengesi açısından alternatif.
    - **Mistral/Llama/Gemma** → genel amaçlı güçlü LLM kıyas referansları.

Projenin Doldurduğu Boşluk ve Farklılıklar

Benzer çalışmalar çoğu zaman tek bir model ailesine odaklanırken, bu proje aynı veri seti üzerinde klasik CNN/RNN tabanlı modeller ile transformer tabanlı modelleri birlikte değerlendirerek daha bütüncül bir karşılaştırma sunmaktadır. Ayrıca yalnızca genel metrikler yerine sınıf bazlı F1 kırılımı da raporlanarak, özellikle azınlık sınıflardaki genelleme davranışı görünür hale getirilmiştir. Bu sayede matematik sorularının konu sınıflandırması probleminde hangi yaklaşımın daha tutarlı sonuç verdiği hem genel hem sınıf bazlı ölçekte daha net biçimde ortaya konulmuştur. Bu literatür çerçevesi, raporda sunulan deney sonuçlarıyla da tutarlıdır. Genel karşılaştırma tablosunda (Tablo 1) Qwen2.5-Math-7B modelinin Accuracy ve F1-macro değerlerinde en üstte yer alması, matematik odaklı transformer modellerin alan-spesifik metinlerde daha güçlü temsil öğrenebildiğini göstermektedir. Benzer şekilde sınıf bazlı sonuçlarda (Tablo 2) Qwen2.5-Math-7B'nin çoğu sınıfta en yüksek F1 değerlerini vermesi, modelin yalnızca çoğunluk sınıflarda değil azınlık sınıflarda da daha dengeli genelleme yaptığını desteklemektedir.

Öte yandan, Geometri sınıfında TextCNN modelinin en yüksek F1 değerini vermesi, CNN tabanlı yaklaşımların metin içindeki yerel kalıpları (ör. kısa tanımlar, sık tekrar eden ifade yapıları) yakalamada bazı sınıflarda avantaj sağlayabildiğini düşündürmektedir. Hız tarafında ise Tablo 2'de DeepSeek-R1 modelinin en düşük latency değerine sahip olması, pratik kullanım senaryolarında "doğruluk–hız dengesi" açısından bu modelin güçlü bir alternatif olabileceğini göstermektedir. Bu nedenle final değerlendirmede en iyi model seçimi yapılırken yalnızca doğruluk değil, sınıf bazlı tutarlılık ve gecikme gibi pratik ölçütler de birlikte ele alınmıştır

Kaynakça

1. Y. Kim, "Convolutional Neural Networks for Sentence Classification," *Proceedings of EMNLP 2014*, pp. 1746–1751, 2014. doi:10.3115/v1/D14-1181. ACL Anthology

2. S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997. doi:10.1162/neco.1997.9.8.1735. MIT Press Direct+1

3. K. Cho, B. van Merriënboer, Ç. Gülçehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation," *Proceedings of EMNLP 2014*, pp. 1724–1734, 2014. doi:10.3115/v1/D14-1179. ACL Anthology+1

4. D. Bahdanau, K. Cho, and Y. Bengio, "Neural Machine Translation by Jointly Learning to Align and Translate," *ICLR 2015 (preprint: arXiv:1409.0473)*, 2014/2015. arXiv+1

5. A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention Is All You Need," *NeurIPS 2017 (preprint: arXiv:1706.03762)*, 2017. NeurIPS Papers+1

6. F. Pedregosa et al., "Scikit-learn: Machine Learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011. Journal of Machine Learning Research+1

7. H. He and E. A. Garcia, "Learning from Imbalanced Data," *IEEE Transactions on Knowledge and Data Engineering*, vol. 21, no. 9, pp. 1263–1284, 2009. doi:10.1109/TKDE.2008.239. ACM Digital Library

8. M. Sokolova and G. Lapalme, "A systematic analysis of performance measures for classification tasks," *Information Processing & Management*, vol. 45, no. 4, pp. 427–437, 2009. doi:10.1016/j.ipm.2009.03.002. ACM Digital Library+1

SONUÇ

Bu çalışmada matematik sorularının konu sınıflandırması problemi için toplam dokuz farklı model aynı değerlendirme düzeni altında karşılaştırılmıştır. Değerlendirmede accuracy, precision, recall, F1-micro ve özellikle sınıf dengesizliğini daha iyi yansıtan F1-macro metrikleri kullanılmış; ayrıca sınıf bazlı (class-wise) F1 sonuçları raporlanarak modellerin her sınıftaki genelleme gücü analiz edilmiştir. Transformer tabanlı modeller için ek olarak AUC ve gecikme (latency) ölçümleri de verilerek yalnızca doğruluk değil, pratik kullanım açısından hız boyutu da değerlendirmeye dahil edilmiştir. Genel karşılaştırma sonuçları, transformer tabanlı modellerin klasik CNN/RNN tabanlı modellere kıyasla daha dengeli ve yüksek performans ürettiğini göstermiştir. Özellikle Qwen2.5-Math-7B modeli, hem accuracy hem de F1-macro açısından en yüksek değerleri vererek en başarılı model olarak öne çıkmıştır. Sınıf bazlı sonuçlar da bu bulguyu desteklemekte; Qwen2.5-Math-7B'nin çoğu sınıfta en yüksek F1 değerlerini vererek farklı konu başlıklarında daha tutarlı genelleme yaptığı görülmektedir. Buna karşılık bazı sınıflarda (örneğin belirli kalıp içeren metinlerde) TextCNN gibi klasik modellerin de güçlü sonuçlar üretebildiği gözlemlenmiştir; bu durum, yerel örüntü yakalamanın bazı alt konularda avantaj sağlayabileceğine işaret etmektedir. Pratik kullanım senaryosu açısından değerlendirildiğinde, en yüksek doğruluğu veren modelin her zaman en uygun seçenek olmayabileceği görülmektedir. Gecikme değerlerinin dikkate alındığı durumda, daha hızlı çalışan modeller "doğruluk–hız dengesi" açısından anlamlı alternatifler sunabilmektedir. Bu nedenle proje çıktıları, hem en yüksek performans hedeflenen senaryolar hem de kaynak/hız kısıtı bulunan senaryolar için model seçimini destekleyecek şekilde yorumlanmıştır. Gelecek çalışmalarda, sınıf dengesizliğinin etkisini azaltmak için sınıf ağırlıkları (class weights), focal loss, veri artırma (data augmentation) veya azınlık sınıflara yönelik örnekleme stratejileri denenebilir. Ayrıca değerlendirme güvenilirliğini artırmak amacıyla stratified K-fold gibi çapraz doğrulama yaklaşımlarıyla daha istatistiksel olarak sağlam bir performans profili çıkarılabilir. Son olarak, dağıtım (deployment) hedefleniyorsa gecikme, bellek kullanımı ve maliyet gibi ölçütler de daha ayrıntılı biçimde analiz edilerek farklı kullanım senaryoları için ayrı model önerileri geliştirilebilir.

Kullanılan Ortam ve Kütüphaneler

Bu projede matematik soru metinlerinden konu etiketi tahmini yapan 8 sınıflı metin sınıflandırma problemi için klasik derin öğrenme modelleri (TextCNN, BiLSTM+Attention, BiGRU) ile transformer tabanlı modeller (DeepSeek-R1, Llama-3-8B, Phi-3-Mini, Gemma-2-9B, Qwen2.5-Math-7B, Mistral-7B) aynı değerlendirme düzeni altında çalıştırılmıştır. Deneyler ağırlıklı olarak Jupyter Notebook üzerinde yürütülmüş; veri hazırlama, eğitim ve metrik hesaplama adımları Python ekosistemi ile gerçekleştirilmiştir. Ayrıca proje içinde sonuçların servislenmesi için FastAPI tabanlı bir arayüz/servis yapısı da bulunmaktadır ve derin öğrenme modellerinin geliştirilmesi için **PyTorch** kullanılmıştır. Veri işleme, değerlendirme ve görselleştirme aşamalarında standart Python kütüphaneleri tercih edilmiştir.

1. **Python:** 3.x
2. **PyTorch:** 2.x
3. **NumPy:** 1.x
4. **Pandas:** 2.x
5. **scikit-learn: 1.x**
6. **Matplotlib / Seaborn:** Görselleştirme için

TextCNN, BiLSTM+Attention ve BiGRU mimarilerinin kurulumu için

- **PyTorch (torch) + torch.nn:**

Transformer Modeller ve NLP Ekosistemi

- **Hugging Face Transformers:**
    - AutoTokenizer, AutoModelForSequenceClassification ile transformer modellerin yüklenmesi

Donanım

1. CPU üzerinde rahatça çalışabilen, küçük ve shallow modeller
2. GPU opsiyoneldir; performans zorunlu değildir
3. Ortalama eğitim süreleri düşük (10–20 saniye)

Veri İşleme Araçları

1. **Pandas:** veri yükleme ve temizleme
2. **NumPy:** diziler ve tensör işlemleri
3. **scikit-learn:** label encoding, train/validation split, metrik hesaplama