



YILDIZ TEKNİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
BİLGİSAYAR MÜHENDİSLİĞİ YÜKSEK LİSANS PROGRAMI
KOLLEKTİF ÖĞRENME
2025-2026
2.ÖDEV PROJE RAPORU

ÖDEV KONUSU : Tez Yılı Tahmininde Kolektif Öğrenme

ÖĞRENCİ ADI-SOYADI : ZEYNEP AZİLİ - 25501963
ESRA SOLTAN - 25501811

I. Giriş ve Proje Özeti

Bu çalışmanın amacı, Türkçe akademik tezlerin başlık ve özet metinlerinden yola çıkarak, tezin yayınlandığı yılı tahmin eden bir makine öğrenmesi modeli geliştirmektir. Tez metinleri, içerdikleri terminoloji ve akademik dil açısından zamanla değişim gösterdiğinden, bu problem bir metin sınıflandırma problemi olarak ele alınmıştır. Çalışma kapsamında, tekil sınıflandırıcılar yerine, genelleme yeteneği daha yüksek olan kolektif öğrenme (ensemble learning) yöntemleri (**Bagging, Random Subspace ve Random Forest**) kullanılmıştır.

2. Veri Seti Tanımlama

Çalışmada HuggingFace platformunda yer alan [umutertugrul/turkish-academic-theses-dataset](#) veri seti kullanılmıştır. Veri seti üzerinde aşağıdaki ön işleme adımları uygulanmıştır:

- **Filtreleme:** 2001 ve 2024 yılları (dahil) arasındaki tezler filtrelenmiştir. (Veri seti üzerinde 2025 yılına dair tez bulunamadığı için 2024 yılına kadar olan tezler çalışma kapsamında kullanılmıştır.)
- **Örnekleme (Sampling):** Sınıf dengesizliğini önlemek amacıyla, her bir yıl için rastgele 500 adet tez seçilmiştir. Toplam veri sayısı $24 * 500 = 12.000$ adettir.
- **Veri Bölme İşlemi (Hold Out Yöntemi):** Her yıl için seçilen 500 tezin %50'si train (250 eğitim), %50'si test (250 test) kümesi olarak ayrılmıştır. Böylece yıl bazında dengeli ve rastgele bir veri seti oluşturulmuş olup, toplamda 6.000 eğitim, 6.000 test verisine ulaşılmıştır.

3. Veri Ön İşleme (Data Preprocessing)

Modelin başarısını doğrudan etkileyebilecek olan veri kalitesini artırmak ve algoritmaların ihtiyaç duyduğu formatı sağlamak amacıyla aşağıdaki veri ön işleme adımları uygulanmıştır:

Proje kapsamında veri manipülasyonu için pandas ve numpy, veri setine erişim için ise Hugging Face datasets Kütüphaneleri kullanılmıştır. İlgili veri seti kaynağından çekildikten sonra, veri işleme kolaylığı sağlama amacıyla Pandas DataFrame formatına dönüştürülmüştür. Veri setinin yapısal bütünlüğünü doğrulamak adına ilk 5 gözlem (head) ve veri tipleri (dtypes) incelenmiş; sütunların beklenen formatta olup olmadığı kontrol edilmiştir.

Ham veri seti içerisinde yazar, danışman, enstitü ve konu gibi modelin yıl tahminleme problemine doğrudan katkısı olmayacağı düşünülen meta veriler bulunmaktadır. Çalışmanın odak noktası metin madenciliği olduğu için, veri seti sadeleştirilerek yalnızca aşağıdaki üç temel değişken analiz kapsamına alınmıştır:

- **title_tr:** Tezin Türkçe başlığı.
- **abstract_tr:** Tezin Türkçe özeti.
- **year:** Tezin yayınlandığı yıl (Hedef değişken).

Ham verideki year (yıl) değişkeni ile sayısal işlem yapılabilmesi için, tip dönüşümü uygulanmıştır. Hatalı veya eksik (null) veri içeren satırların olup olmadığı kontrol edilmiş, modelin gürültülü veriden öğrenmesini engellemek adına gerekli temizlik işlemleri yapılmıştır.

3.1 Veri setinin genel yapısı

sampled_df.head()					
	year	title_tr	abstract_tr	subject	year_numeric
641625	2001	Nilpotent grupların integral grup halkalarında...	ÖZET NİLPOTENT GRUPLARIN INTEGRAL GRUP HALKALAR... ...	Matematik = Mathematics	2001
429635	2001	İzmit Klinik ve Tehlikeli Atıklar Yakma Tesisi...	İZMİT KLİNİK VE TEHLİKELİ ATIKLAR YAKMA TESİSİ...	Çevre Mühendisliği = Environmental Engineering	2001
234017	2001	Polipirol ile modifiye edilmiş karbon elektrot...	POLİPIROL İLE MODİFYİYE EDİLMİŞ KARBON ELEKTROT...	Kimya = Chemistry	2001
313410	2001	Diagnostik laparoskopisi yapılan tüberküloz peri...	ÖZET Tüberküloz tarih boyunca insanlığı en çok... ...	Gastroenteroloji = Gastroenterology ; Göğüs Ha...	2001
508998	2001	Asarcık ağızı	...	Türk Dili ve Edebiyatı = Turkish Language and Literature	2001

Veri setinin genel yapısı .head (fonksiyonu) ile incelenmiştir.

3.2 Tez Konusu Sınıf Sayısı

subject	count
Eğitim ve Öğretim = Education and Training	849
İşletme = Business Administration	580
Ziraat = Agriculture	368
Türk Dili ve Edebiyatı = Turkish Language and Literature	291
Tarih = History	256
...	...
Genetik = Genetics ; Moleküler Tip = Molecular Medicine ; Nöroloji = Neurology	1
Deniz Bilimleri = Marine Science ; Denizcilik = Marine ; Kazalar = Accidents	1
Biyoteknoloji = Biotechnology ; Kimya Mühendisliği = Chemical Engineering ; Polimer Bilim ve Teknolojisi = Polymer Science and Technology	1
Eğitim ve Öğretim = Education and Training ; Tarih = History ; Türk İnkılap Tarihi = History of Turkish Revolution	1
Halk Bilimi (Folklor) = Folklore ; Sahne ve Görüntü Sanatları = Performing and Visual Arts ; Türk Dili ve Edebiyatı = Turkish Language and Literature	1

1232 rows x 1 columns

En yüksek sınıf sayısına ait tez alanının “Eğitim ve Öğretim” olduğu tespit edilmiştir.

3.3 Eksik Değer Analizi

```
sampled_df.isnull().sum()
```

	0
year	0
title_tr	0
abstract_tr	0
subject	5
year_numeric	0

dtype: int64

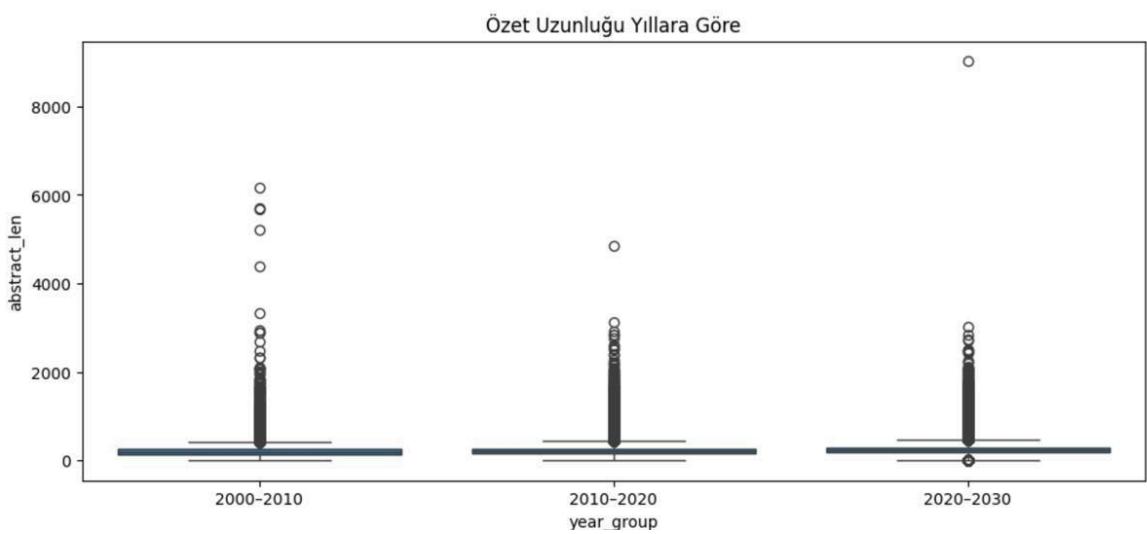
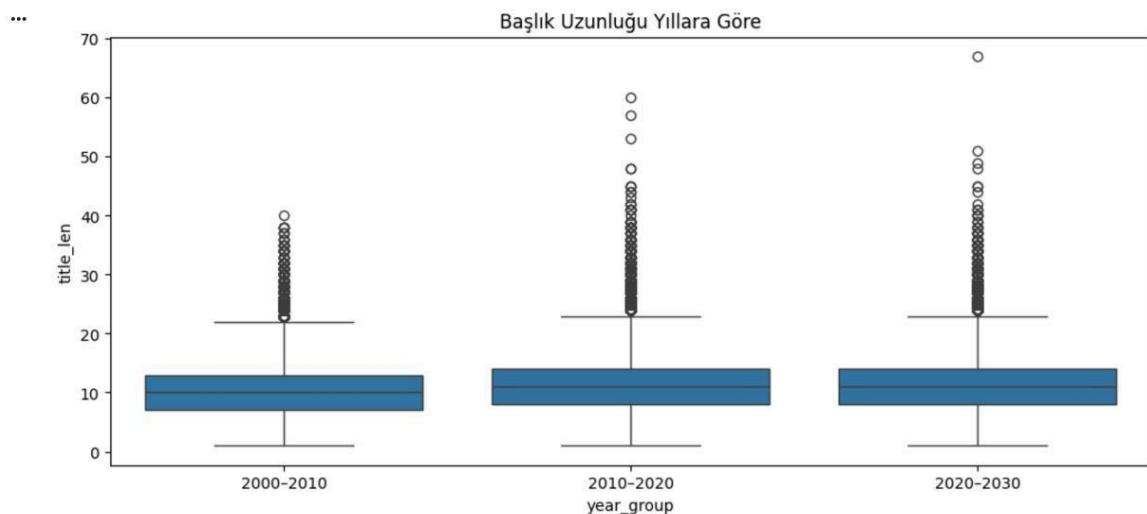
Veri setindeki sütunlarda eksik (null) değer olup olmadığı kontrol edilmiştir ve Subject sütununda 5 adet eksik veri olduğu görülmüştür.

3.4 Başlık Uzunluğu ve Özeti Uzunluğu Yıllara Göre Scatter Plot Grafiği

```
df_clean["year_group"] = pd.cut(df_clean["year_numeric"],
                                bins=[2000, 2010, 2020, 2030],
                                labels=["2000-2010", "2010-2020", "2020-2030"])

/tmpp/ipython-input-707336673.py:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user\_guide/indexing.html#returning-a-view-versus-a-copy
df_clean["year_group"] = pd.cut(df_clean["year_numeric"],
```



Tez başlıkları ve özetleri 10'ar yıllık gruplara ayrılmış ve bu periyotlarda veri dağılımlarının farklılıklarını gözlemlenmiştir.

3.5 İstatistiksel Anlamlılık Testi (ANOVA)

```
# ANOVA testi → ortalama uzunluklar yıllara göre farklı mı?  
from scipy.stats import f_oneway  
  
f_title = f_oneway(  
    *[df_clean.loc[df_clean["year_group"]==grp,"title_len"] for grp in df_clean["year_group"].unique()])  
)  
f_abs = f_oneway(  
    *[df_clean.loc[df_clean["year_group"]==grp,"abstract_len"] for grp in df_clean["year_group"].unique()])  
)  
  
print("\nANOVA Title:", f_title)  
print("ANOVA Abstract:", f_abs)
```

ANOVA Title: F_onewayResult(statistic=np.float64(2628.6061693293664), pvalue=np.float64(0.0))

ANOVA Abstract: F_onewayResult(statistic=np.float64(4850.332370292906), pvalue=np.float64(0.0))

Yıllara göre tezlerin başlık uzunlukları ya da özet uzunlukları arasında istatistiksel olarak anlamlı farklılık olup olmadığını belirlemek için ANOVA testi uygulanmıştır.

H0: “Yıllara göre ortalama metin uzunlukları arasında fark yoktur”. Elde edilen sonuçlarda hem title_len hem de abstract_len için p-value ≈ 0 bulunmuştur ($p < 0.05$). H0 red. Metin uzunluklarının yıllar arasında anlamlı farklılık gösterdiği sonucuna ulaşılmıştır.

3.6 Spearman Korelasyon Analizi

```
df_clean[["year_numeric","title_len","abstract_len"]].corr(method="spearman")
```

	year_numeric	title_len	abstract_len
year_numeric	1.000000	0.098038	0.15737
title_len	0.098038	1.000000	0.22385
abstract_len	0.157370	0.223850	1.00000

0.40 Güçlü pozitif — yıl arttıkça metin uzuyor 0.20–0.40 Zayıf ama anlamlı < 0.15 Neredeyse ilişki yok

Yıl (year_numeric) ile metin (title_len, abstract_len) uzunlukları arasındaki ilişki Spearman korelasyon katsayısı ile incelenmiştir. Pozitif ancak düşük korelasyon değerleri, yıllar ilerledikçe tez metinlerinin kısmen uzama eğiliminde olduğunu, ancak tek başına yıl tahmininde yeterli olmayacağı göstermiştir.

4. Vektör Temsilleri (Embedding)

Makine öğrenmesi algoritmalarının metinsel veriyi işleyebilmesi için, verilerin matematiksel vektör uzayına taşınması gerekmektedir. Bu kapsamda, Türkçe metinler için özel olarak eğitilmiş ve anlamsal ilişkileri koruma başarısı yüksek olan [ytu-ce-cosmos/turkish-e5-large](#) dil modeli kullanılmıştır. Bu model aracılığıyla:

- **Tez başlıkları (Title),**

- **Tez özetleri** (Abstract),
- **Birleştirilmiş Temsili (Tab)** (*Tab=Title + Abstract*)
ayrı ayrı işlenerek, her bir öznitelik için 1024 boyutlu vektör temsilleri elde edilmiştir.

5. Yöntem

Bu çalışmada, tez yılı tahminlemesi problemi bir sınıflandırma (classification) problemi olarak ele alınmıştır. Tekil öğrenicilerin (base learners) genellikle yüksek varyans ve aşırı öğrenme (overfitting) eğilimi göstermesi nedeniyle, raporun giriş kısmında da belirtildiği üzere bu dezavantajları minimize eden ve genelleme yeteneği daha yüksek olan **Kolektif Öğrenme (Ensemble Learning)** yöntemleri tercih edilmiştir.

5.1 Model

Çalışma kapsamında kullanılan temel algoritmalar aşağıda açıklanmıştır:

5.1.1 Bagging (Bootstrap Aggregating)

Bagging yöntemi, eğitim veri setinden yerine koymalı örneklemeye (bootstrap sampling) yaparak çok sayıda alt eğitim kümesi oluşturur. Her bir alt küme üzerinde bağımsız bir **Karar Ağacı (Decision Tree)** eğitilir. Modelin nihai tahmini, tüm ağaçların ürettiği tahminlerin çoğunluk oylaması (majority voting) ile belirlenir. Bu yöntem, özellikle veri setindeki gürültüye karşı direnci artırmak ve model varyansını düşürmek amacıyla kullanılmıştır.

5.1.2 Random Subspace

Standart Bagging yöntemi veri satırlarını örneklerken, Random Subspace yöntemi öznitelik uzayında (feature space) çeşitlilik sağlar. Modelimizdeki vektörlerin tamamını kullanmak yerine, her bir temel öğrenici (base learner) için özniteliklerin rastgele bir alt kümesi seçilmiştir. Bu yaklaşım, özellikler arasındaki korelasyonu düşürür.

5.1.3 Random Forest (Rassal Orman)

Random Forest, Bagging yönteminin geliştirilmiş ve daha güçlü bir versiyonudur. Bagging'den farklı olarak, karar ağaçlarının oluşturulması sırasında her bir düğümde (node) en iyi bölmeyi ararken tüm öznitelikleri değerlendirmek yerine, rastgele seçilen bir öznitelik alt kümelerini kullanır. Bu işlem, ağaçlar arasındaki korelasyonu (bağımlılığı) düşürür ve kolektif yapının çeşitliliğini (diversity) artırır. Yüksek boyutlu metin verilerinde (embedding vektörleri gibi) Random Forest genellikle üstün performans sergilemektedir.

5.2 Hiper-Parametre Optimizasyonu

Kurulan modellerin performansını artırmak ve veri setine en uygun kombinasyonları bulmak için her algoritmada **Randomized Search CV** hiper parametre optimizasyonları kullanılmıştır. Bunun sebebi Grid Search CV'den daha hızlı olması ve verilen aralıkta rastgele deneme yapmasıdır.

5.2.1 Random Forest Algoritması

Aşağıdaki parametre uzayında rastgele örnekleme yöntemiyle hiper-parametre optimizasyonu gerçekleştirilmiştir:

- **Ağaç Sayısı (n_estimators):** randint (50,121)
- **Ağaç Derinliği (max_depth):** [10, 20]
- **Yaprak Başına Minimum Örnek (min_samples_leaf):** randint(5,10)

Bu sayede hem modelin aşırı öğrenmesi (overfitting) engellenmiş hem de hesaplama maliyeti düşürüülerek en iyi genellemeye performansına ulaşılması hedeflenmiştir.

Ayrıca **n_jobs=2** olarak atanmış olup, sürecin hızlanması sağlanmıştır.

n_jobs değeri	Anlamı
<code>n_jobs=1</code>	:Sadece 1 CPU çekirdeği kullanır → en yavaş ama en stabil
<code>n_jobs=2</code>	:2 çekirdek paralel çalışır → 2 kat hızlanabilir
<code>n_jobs=-1</code>	:Kullanılabilir tüm çekirdekleri kullanır → maksimum hız, ama RAM ve ısınma artar

```
===== Abstract için Random Forest RandomizedSearch (HIZLI) =====
Fitting 2 folds for each of 8 candidates, totalling 16 fits

[Abstract-RandomForest] En iyi CV accuracy (sub-train): 0.0688
[Abstract-RandomForest] En iyi parametreler: {'max_depth': 20, 'min_samples_leaf': 10, 'n_estimators': 113}

[Abstract - Random Forest TEST SONUÇLARI]
Accuracy : 0.0838
MAE      : 5.8805
RMSE     : 7.8057
```

Kombinasyonlardan en iyi Abstract Train seti Accuracy değeri 0.0688 iken Test seti Accuracy değeri 0.0838'dir.

MAE değeri 5.88 olup, yaklaşık olarak 6 yıllık bir hata yaptığı görülmektedir.

5.2.1 Bagging Algoritması

Bagging algoritması için temel öğrenici olarak Karar Ağaçları (Decision Tree) kullanılmış ve aşağıdaki hiper-parametreler optimize edilmiştir:

- "**n_estimators": randint(50, 121)** : 50 ile 120 arasında rastgele tamsayı değerleridir.
- "**estimator__max_depth": [10, 20]** : Her iterasyondaki farklı ağaç sayısıdır.
- min_samples_leaf=5** : Aşırı öğrenmeyi engellemek için bir yaprak düğümünde bulunan örnek sayısı
- max_samples=0.7** : Her bir ağaç, eğitim veri setindeki örneklerin

	(satırların) rastgele %70'i seçilerek eğitilmiştir.
max_features=0.7	: Her bir ağaç, eğitim veri setindeki özniteliklerin (sütunların) rastgele %70'i seçilerek eğitilmiştir.
n_jobs=1	: Sadece tek CPU kullanır.
n_iter=8	: Rastgele 8 iterasyon yapar.
cv=2	: Veri setinin ikiye bölünmesidir.

```
===== TaB için Bagging RandomizedSearch (HIZLI) =====
Fitting 2 folds for each of 8 candidates, totalling 16 fits

[TaB-Bagging] En iyi CV accuracy (sub-train): 0.0663
[TaB-Bagging] En iyi parametreler: {'estimator__max_depth': 20, 'n_estimators': 79}

[TaB - Bagging TEST SONUÇLARI]
Accuracy : 0.0763
MAE      : 6.1100
RMSE     : 7.9960
```

Kombinasyonlardan en iyi TAB Train seti Accuracy değeri 0.0663 iken Test seti Accuracy değeri 0.0763'tür.

MAE değeri 6.11 olup, yaklaşık olarak 6 yıllık bir hata yaptığı görülmektedir.

5.2.3 Random Subspace Algoritması

Random Subspace algoritması, her bir temel öğrenicinin (karar ağaçının) eğitim verisindeki özniteliklerin (features) yalnızca rastgele bir alt kümesini kullanarak eğitilmesi prensibine dayanır. Bu yaklaşım, ağaçlar arasındaki korelasyonu düşürerek modelin varyansını azaltmayı hedefler. Çalışmada, BaggingClassifier yapılandırılarak max_features parametresinin 1.0'dan küçük değerler alması sağlanmış ve özellik uzayında çeşitlilik elde edilmiştir. Temel öğrenici olarak aşırı öğrenmeyi önlemek adına derinliği sınırlandırılmış (max_depth=20) Karar Ağaçları kullanılmış ve modelin en iyi konfigürasyonunu belirlemek için aşağıdaki hiper-parametreler optimize edilmiştir:

"n_estimators": randint(50, 121)	: 50 ile 120 arasında rastgele tamsayı değerleridir.
"max_features": [0.3, 0.5, 0.8]	: Her bir ağaçın eğitilirken sütunların sadece belirli bir oranını (%30, %50 veya %80) görmesini sağlayarak çeşitliliği artırır.
max_depth=20	: karar ağaçının inebileceği max derinliktir.
min_samples_leaf=5	: Bir yaprak düğümünde bulunan örnek sayısıdır.
test_size=0.6	: Hiper Parametre aramasında eğitim kümelerinin %40'ı kullanılmıştır.
max_samples=1.0	: Her ağaçın, train setinin tamamının kullanılması
n_jobs=2	: CPU'nun çift çekirdekle paralel çalışmasıdır.

```
=====
Abstract için Random Subspace RandomizedSearch (HIZLI)
Fitting 2 folds for each of 8 candidates, totalling 16 fits

[Abstract-RS] En iyi CV accuracy (sub-train): 0.0708
[Abstract-RS] En iyi parametreler: {'max_features': 0.8, 'n_estimators': 71}

[Abstract - Random Subspace TEST SONUÇLARI]
Accuracy : 0.0797
MAE      : 6.0072
RMSE     : 7.9138
```

Kombinasyonlardan en iyi Abstract Train seti Accuracy değeri 0.0708 iken Test seti Accuracy değeri 0.0797'dir.

MAE değeri 6.00 olup, yaklaşık olarak 6 yıllık bir hata yaptığı görülmektedir.

6.Bulgular

Çalışma kapsamında oluşturulan üç farklı veri temsilinin (Başlık, Özeti ve Tab) ve uygulanan üç farklı kolektif öğrenme algoritmasının (Bagging, Random Forest, Random Subspace) performansları, Doğruluk (Accuracy) ve Ortalama Mutlak Hata (MAE) metrikleri üzerinden görselleştirilmiştir.

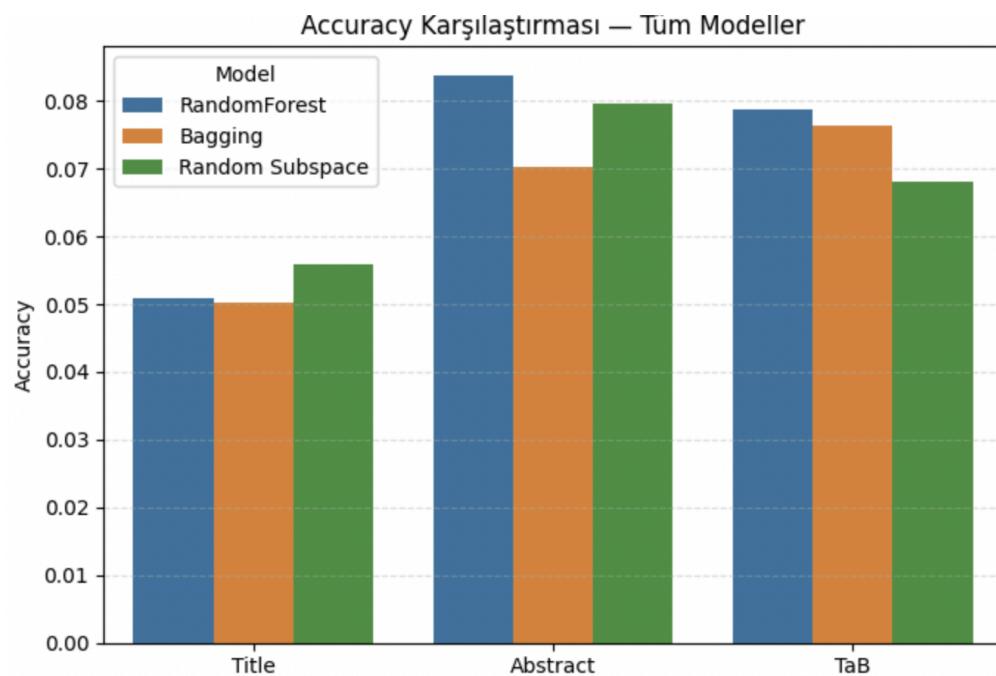
Şekil 1: Model Doğruluk (Accuracy) Oranlarının Karşılaştırılması

== TOPLU PERFORMANS TABLOSU ==						
	Temsil	Model	Accuracy	MAE	RMSE	\
0	Title	RandomForest	0.050833	7.465500	9.339102	
1	Abstract	RandomForest	0.083833	5.880500	7.805714	
2	TaB	RandomForest	0.078667	6.322333	8.305159	
3	Title	Bagging	0.050167	7.318167	9.181022	
4	Abstract	Bagging	0.070167	6.141500	8.036490	
5	TaB	Bagging	0.076333	6.110000	7.995999	
6	Title	Random Subspace	0.055833	7.181167	9.055045	
7	Abstract	Random Subspace	0.079667	6.007167	7.913775	
8	TaB	Random Subspace	0.068167	6.078333	7.917828	

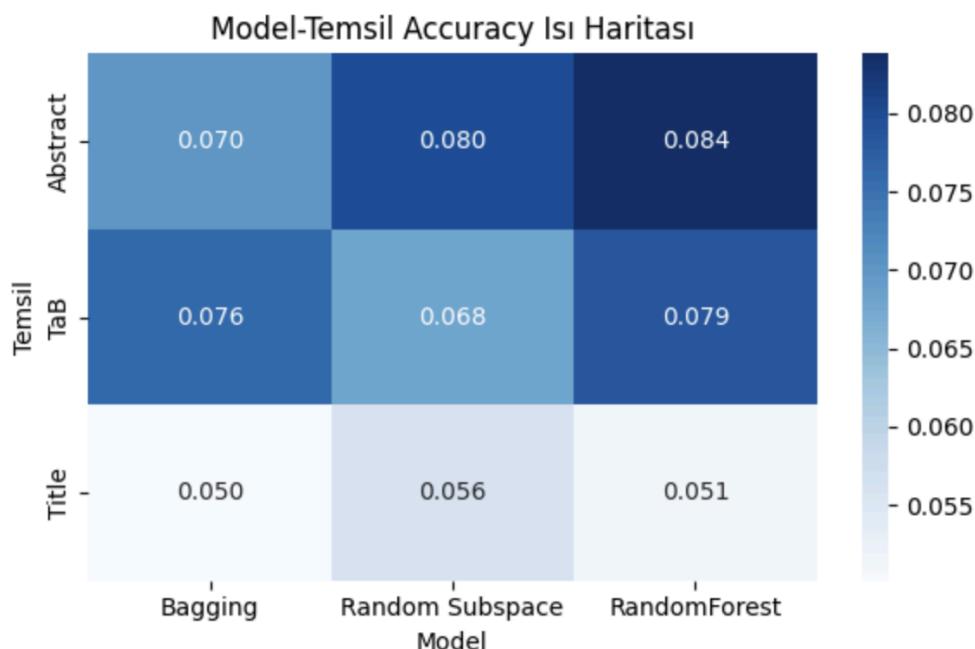
Elde edilen doğruluk oranlarını gösteren grafik incelendiğinde, veri setinin kalitesinin model başarısı üzerindeki belirleyici etkisi net bir şekilde görülmektedir:

EN İYİ MODEL : RandomForest
 EN İYİ TEMSİL : Abstract
 Accuracy Skoru : 0.0838

Şekil 2- Tüm Modellerin Accuracy Karşılaştırması



Şekil 3- Model - Temsil Accuracy Isı Haritası



Random Forest, Bagging ve Random Subspace'den daha iyi sonuç verdi çünkü iki teknigi birlikte kullanarak hem bootstrap hem feature rassalligi uygular. Böylece hem varyansı hem korelasyon etkisini azaltır ve daha genellenebilir bir tahmin gücü sağlar. Yıl tahmini için abstract metinlerinin yalnızca title'a göre daha belirgin bir zaman bilgisi taşıdığı görülmüştür. Tab birleşimi ise boyut genişlediği için gürültü artışına neden olmuş ve performansı düşürmüştür.

7. Öneriler

- **Yılların Kümelenmesi:** Modelin başarısını artırmak için, tezlerin tekil yıllar (2021, 2022 vb.) yerine **5'er yıllık dönemler (kümeler)** halinde (Örn: 2000-2005 Dönemi) sınıflandırılması önerilmektedir. Bu sayede sınıf içi varyans azalacak ve doğruluk oranı yükselecektir.
- **Regresyon Yaklaşımı:** Problemin kategorik sınıflandırma yerine, yıllar arasındaki sıralı ilişkiyi koruyan **Regresyon** modelleri ile çözülmesi hata payını düşürebilir.
- **Veri Zenginleştirme:** Uç yıllardaki (2000 başları ve günümüz) veri sayısının artırılması veya sentetik veri üretimi (oversampling) ile daha genellenebilir ve yüksek doğruluk oranına sahip sonuçlar elde edilebilir.
- **Embedding :** Embedding yöntemi değiştirilerek farklı varyasyonlar denenebilir.

8.Ek Notlar

- Çalışma süresince Unsupervised Learning olan K-Means yöntemi ile 24 sınıfından ilişkili 8 sınıfı indirgeyerek bir sınıflandırma yapıldı ve başarı oranı 5 kat artarak 0.35 bulundu.

En iyi RF val accuracy: 0.35166666666666667

En iyi RF parametreler: {'n_estimators': 300, 'max_depth': None, 'min_samples_leaf': 5}

- Hiper-parametre optimizasyon yöntemlerinden Optuna denenmiş olup, RAM'e aşırı yükleme olduğundan dolayı çalışma nihayete erdirilememiştir.
- Embedding'te farklı bir yöntem olan weighted mean fusion hibrit modeli kullanılarak title ve abstract farklı ağırlıklandırma yapılarak da başarının arttığı gözlemlenmiş, denenmiştir.

Tüm Sonuçlar:

	Temsil	Model	Accuracy	MAE	RMSE
0	Title	Bagging	0.061000	6.839000	8.741281
1	Abstract	Bagging	0.092167	5.667000	7.554006
2	Fusion	Bagging	0.080500	6.037667	7.962684
0	Title	Random Subspace	0.059833	7.094667	8.991125
1	Abstract	Random Subspace	0.087833	5.890667	7.799167
2	Fusion	Random Subspace	0.079333	6.011500	7.926611
0	Title	Random Forest	0.058667	6.926333	8.797500
1	Abstract	Random Forest	0.089500	5.409667	7.215077
2	Fusion	Random Forest	0.080500	5.739500	7.565327

- Classifier yerine Regressor modeli kullanıldı ancak başarının düşüğü ve çalışma zamanının arttığı gözlemlendi.

==== Accuracy Pivot ===				
Model	Bagging-Regressor	Random Forest (Regressor)	Random Subspace (Regressor)	
Temsil				
Abstract	0.053333		0.051833	0.049167
Tab	0.054500		0.056000	0.052833
Title	0.045500		0.045000	0.044667
==== MAE Pivot ===				
Model	Bagging-Regressor	Random Forest (Regressor)	Random Subspace (Regressor)	
Temsil				
Abstract	5.002089		4.998874	5.019181
Tab	5.060882		5.057419	5.069118
Title	5.656654		5.656953	5.669574
==== RMSE Pivot ===				
Model	Bagging-Regressor	Random Forest (Regressor)	Random Subspace (Regressor)	
Temsil				
Abstract	5.965566		5.957531	5.965866
Tab	6.021226		6.017288	6.009154
Title	6.594760		6.590890	6.606068