



**YENİLİKÇİ PROBLEM ÇÖZÜMLERİNDE YAPAY ZEKA TABANLI
KARAR DESTEK SİSTEMİ OLUŞTURMA**

Seher SOLMAZ

**YÜKSEK LİSANS TEZİ
BİLİŞİM SİSTEMLERİ ANA BİLİM DALI**

**GAZİ ÜNİVERSİTESİ
BİLİŞİM ENSTİTÜSÜ**

AĞUSTOS 2024

ETİK BEYAN

Gazi Üniversitesi Bilişim Enstitüsü Tez Yazım Kurallarına uygun olarak hazırladığım bu tez çalışmada;

- Tez içinde sunduğum verileri, bilgileri ve dokümanları akademik ve etik kurallar çerçevesinde elde ettiğimi,
 - Tüm bilgi, belge, değerlendirme ve sonuçları bilimsel etik ve ahlak kurallarına uygun olarak sunduğumu,
 - Tez çalışmada yararlandığım eserlerin tümüne uygun atıfta bulunarak kaynak gösterdiğimi,
 - Kullanılan verilerde herhangi bir değişiklik yapmadığımı,
 - Bu tezde sunduğum çalışmanın özgün olduğunu,
- bildirir, aksi bir durumda aleyhime doğabilecek tüm hak kayıplarını kabullendiğimi beyan ederim.

İmza

Seher SOLMAZ

08/08/2024

YENİLİKÇİ PROBLEM ÇÖZÜMLERİNDE YAPAY ZEKA TABANLI KARAR DESTEK SİSTEMİ OLUŞTURMA

(Yüksek Lisans Tezi)

Seher SOLMAZ

GAZİ ÜNİVERSİTESİ

BİLİŞİM ENSTİTÜSÜ

Ağustos 2024

ÖZET

Yapay zeka teknolojisinde sürekli devam eden gelişmeler neticesinde, günümüzde hemen hemen her alanda uygulanabilir hale gelmiştir. Bu kapsamda hukuki araştırmalar, karar tahmini, dava yönetimi ve diğer hukuki süreçlerin daha etkili ve verimli hale getirilmesini sağlamaktadır. Bu tez çalışmasında, Kamu Denetçiliği Kurumu tarafından kurumun internet sitesinde yayınlanan kamu erişime açık kişisel verilerden arındırılmış verilerle karar destek sistemi oluşturulmuştur. Sistem karar tahmini ve taslak karar metni üretimi olmak üzere iki kısımdan oluşmaktadır. İlk kısım olan karar tahmini için denetimli makine öğrenme modelleri (Karar Ağacı, Destek Vektör Makinesi, Naïve Bayes, K-en Yakın Komşu, Lojistik Regresyon ve XGBoost) ve derin öğrenme modelleri (Evrişimsel Sinir Ağları, İleri Beslemeli Sinir Ağları, Mükerrer Birimli Sinir Ağı, Uzun ve Kısa Vade Hafızalı Sinir Ağı ve Çift Yönlü LSTM) kullanılmıştır. Modeller eğitim sırasında, karar türü ve karar metninde yer alan “BAŞVURANIN İDDİA VE TALEPLERİ” kısmında bulunan metinlerle eğitilmiş diğer kısımlar modellere gösterilmemiştir. Makine öğrenmesi modelleri için metinler TF-IDF vektörlerine derin öğrenmesi modelleri için ise metinler tokenlere dönüştürülerek eğitilmiştir. Elde edilen sonuçlar ile farklı modellerin performansları karşılaştırmalı ve detaylı bir şekilde incelenmiştir. Test veri seti üzerinde Ombudsmanlık kararlarını en başarılı şekilde tahmin eden makine öğrenmesi modeli SVM, derin öğrenme modeli ise GRU ve LSTM olduğu gözlemlenmiştir. Taslak karar metni oluşturmak için ytu-ce-cosmos/turkish-gpt2-large modeli ile fine-tuning işlemi yapılmıştır. Fine-tuning işleminde model karar türü, şikayet konusu, şikayet alt konusu ve karar metninde yer alan “KARAR” kısmında bulunan metinlerle eğitilmiş diğer kısımlar gösterilmemiştir. Model metinlerle eğitilerek taslak karar metni oluşturması sağlanmıştır. Karar tahmininde SVM modeli ile %61, GRU ve LSTM modelleri ile %64 F1 skoru değerine ulaşılmıştır. Fine-tuning işleminde eğitim kaybı 0,26 ve doğrulama kaybı 0,23 olan başarılı bir model elde edilmiştir. Oluşturulan karar destek sistemi, hukuk alanında daha verimli karar verme sürecine yönelik önemli bir potansiyele sahiptir.

Bilim Kodu : 92432

Anahtar Kelimeler : Yapay zeka, makine öğrenmesi, derin öğrenme, metin sınıflandırma, metin üretme, karar destek sistemi

Sayfa Adedi : 57

Danışman : Prof. Dr. Mahir DURSUN

CREATING AN AI-BASED DECISION SUPPORT SYSTEM WITH INNOVATIVE PROBLEM SOLUTIONS

(M. Sc. Thesis)

Seher SOLMAZ

GAZİ UNIVERSITY
INSTITUTE OF INFORMATICS

Ağustos 2024

ABSTRACT

Due to continuous advancements in artificial intelligence technology, it has become applicable in almost every field today. In this context, AI enables legal research, decision prediction, case management, and other legal processes to be conducted more effectively and efficiently. In this thesis, a decision support system was created using data published by the Ombudsman's Office on its website, which is free of personal data and open to public access. The system consists of two parts: decision prediction and draft decision text generation. For the first part, decision prediction, supervised machine learning models (Decision Tree, Support Vector Machine, Naïve Bayes, K-Nearest Neighbor, Logistic Regression, and XGBoost) and deep learning models (Convolutional Neural Networks, Feed Forward Neural Networks, Gated Recurrent Unit Neural Networks, Long Short-Term Memory Networks, and Bidirectional LSTM) were employed. During training, the models were provided with the decision type and the texts in the "applicant's claims and demands" section of the decision text, while other parts were excluded. For machine learning models, the texts were transformed into TF-IDF vectors, and for deep learning models, the texts were tokenized. The results obtained and the performance of the different models were compared and analyzed in detail. It was observed that SVM was the most successful machine learning model in predicting Ombudsman decisions on the test dataset, while GRU and LSTM were the most successful deep learning models. Fine-tuning was performed using the ytu-cecosmos/turkish-gpt2-large model to generate draft decision texts. During fine-tuning, the model was trained with the decision type, complaint subject, complaint sub-subject, and the text in the "decision" section, excluding other parts. The model was trained to generate a draft decision text. In decision prediction, an F1 score of 61% was achieved with the SVM model, while the GRU and LSTM models achieved an F1 score of 64%. In the fine-tuning process, a successful model was obtained with a training loss of 0.26 and a validation loss of 0.23. The decision support system holds significant potential for more efficient decision-making in the field of law.

Science Code : 92432

Key Words : Artificial intelligence, machine-learning, deep learning, text classification, text generation, decision support system

Page Number : 57

Supervisor : Assist. Prof. Dr. Mahir DURSUN

TEŐEKKÜR

Öncelikle Tez danışmanım Sayın Prof. Dr. Mahir DURSUN'a, tez konusunun belirlenmesinden sonuca ulaşana kadar verdiği rehberlik, bilgi ve destek için sonsuz teşekkürlerimi sunarım. Kendisinin engin bilgi birikimi ve sabrı bu çalışmanın tamamlanmasını mümkün kıldı. Eğitim hayatım boyunca bana her zaman inanan ve destek olan aileme, özellikle de sevgili annem, babam, eşim Şahin SOLMAZ'a ve bana sevgisi ile güç veren oğlum Mehmet Akif SOLMAZ'a, sonsuz sevgi ve teşekkürlerimi sunarım. Onların manevi desteğı ve teşviki bu sürecin en önemli parçalarından biri olmuştur.

Son olarak, bu çalışmanın gerçekleşmesine katkı sağlayan Kamu Denetçiliğı Kurumu'na teşekkür ederim.

İÇİNDEKİLER

	Sayfa
ÖZET	iv
ABSTRACT.....	v
TEŞEKKÜR.....	vi
İÇİNDEKİLER	vii
ÇİZELGELERİN LİSTESİ.....	ix
ŞEKİLLERİN LİSTESİ.....	x
RESİMLERİN LİSTESİ.....	xi
SİMGELER VE KISALTMALAR.....	xii
1. GİRİŞ.....	1
2. VERİ SETİNİN HAZIRLANMASI.....	7
2.1. Veri Toplama.....	7
2.2. Veri Önleme	8
2.2.1. Metinlerin Küçük Harfe Dönüştürülmesi.....	10
2.2.2. Noktalama İşaretlerinin ve Özel Karakterlerin Temizlenmesi.....	11
2.2.3. Gereksiz Boşlukların Kaldırılması.....	11
2.2.4. Durdurma Kelimelerinin Kaldırılması	11
3. KARAR TAHMİNİ	13
3.1. Makine Öğrenmesi Modelleri	13
3.1.1. Karar ağacı	19
3.1.2. Destek vektör makineleri	19
3.1.3. Naïve bayes	20
3.1.4. K-en yakın komşu	21

3.1.5. Lojistik regresyon.....	21
3.1.6. XGBoost.....	21
3.2. Derin Öğrenme Modelleri	22
3.2.1. Evrişimsel sinir ağları	24
3.2.2. İleri beslemeli sinir ağlar.....	26
3.2.3. Mükerrer birimli sinir ağı.....	27
3.2.4. Uzun ve kısa vade hafızalı sinir ağı	28
3.2.5. Çift yönlü LSTM.....	28
3.3. Sınıflandırma Ölçütleri	29
3.3.1. Kesinlik	29
3.3.2. Duyarlılık	30
3.3.3. F1-Skor.....	30
3.3.4. Makro ortalama	30
3.3.5. Ağırlıklı ortalama	31
3.3.6. Doğruluk	31
4. TASLAK KARAR METNİ.....	33
4.1. İnce Ayar	35
4.2. Metin Üretimi	36
5. DENEYSEL SONUÇLAR	39
5.1. Karar Tahmini Uygulaması	39
5.2. Taslak Karar Metni Oluşturma Uygulaması	44
6. SONUÇ VE ÖNERİLER.....	47
KAYNAKLAR	51
ÖZGEÇMİŞ	57

ÇİZELGELERİN LİSTESİ

Çizelge	Sayfa
Çizelge 2.1. Başvuru konusuna göre karar sayısı	8
Çizelge 2.2. Karar türüne göre dosya dağılımı.....	10
Çizelge 5.1. Makine öğrenmesi modelleri karar tahmini sonuçları.....	40
Çizelge 5.2. Derin öğrenme modelleri ret karar tahmini sonuçları	42
Çizelge 5.3. CNN modeli karar tahmini sonuçları	43
Çizelge 5.4. GRU modeli performans değerleri.....	43
Çizelge 5.5. Fine-tuning kayıp değerleri.....	45

ŞEKİLLERİN LİSTESİ

Şekil	Sayfa
Şekil 3.1. Makine öğrenmesi modelleri ile karar tahmini uygulaması akış şeması.....	15
Şekil 3.2. Veri setinin ayrılması.....	16
Şekil 3.3. TF-IDF değerleri.....	18
Şekil 3.4. Destek vektör makinesi.....	20
Şekil 3.5. Lojistik regresyon	21
Şekil 3.6. Derin öğrenme modelleri ile karar tahmini uygulaması akış şeması.....	23
Şekil 3.7. Evrişimsel sinir ağları	24
Şekil 3.8. İleri beslemeli sinir ağları	26
Şekil 3.9. Mükerrer birimli sinir ağı	27
Şekil 3.10. Uzun ve kısa vade hafızalı sinir ağı.....	28
Şekil 3.11. Çift yönlü LSTM	29
Şekil 4.1. Taslak karar metni oluşturma uygulaması akış şeması.....	34
Şekil 5.1. Makine öğrenmesi modellerinin F1-skoru.....	41

RESİMLERİN LİSTESİ

Resim	Sayfa
Resim 2.1. Kamu denetçiliği kurumu kararlar bilgi bankası.....	7
Resim 2.2. Kamu denetçiliği kurumu örnek karar metni.....	9
Resim 4.1. Örnek karar metni karar alanı	33



SİMGELER VE KISALTMALAR

Bu çalışmada kullanılmış simgeler ve kısaltmalar, açıklamaları ile birlikte aşağıda sunulmuştur.

Kısaltmalar	Açıklamalar
BiLSTM	Çift Yönlü LSTM
CNN	Evrişimsel Sinir Ağları
DDİ	Doğal Dil İşleme
DT	Karar Ağacı
FFNN	İleri Beslemeli Sinir Ağları
GRU	Mükerrer Birimli Sinir Ağı
GPT	Generative Pre-trained Transformer
IDF	Ters Döküman Frekansı
KNN	K-en Yakın Komşu
LR	Lojistik Regresyon
LSTM	Uzun ve Kısa Vade Hafızalı Sinir Ağı
SVM	Destek Vektör Makineleri
TF	Terim Frekansı
TF-IDF	Terim Frekansı -Ters Döküman Frekansı
OCR	Optik Karakter Tanıma
XGBoost	Aşırı Gradyan Artırma (Extreme Gradient Boosting)
YZ	Yapay Zeka

1. GİRİŞ

Hızlı bir ivme ile gelişmekte olan Yapay Zeka (YZ), hemen hemen tüm çalışma alanlarına uygulanabilir hale gelmiş ve etkin bir şekilde kullanılmaya başlanmıştır. YZ'de meydana gelen gelişmeler diğer çalışma alanlarını etkilediği gibi hukuk alanını da etkilemiştir. Hukuk alanında çalışmalar günümüzde popüler olmasına rağmen çalışmalar daha eskiye dayanmaktadır. Alan Turing'in yapmış olduğu çalışma[1] YZ'nin hukuk alanında kullanımına dair doğrudan bir çalışma olmasada hukuki karar vermede potansiyeneline ilişkin temeller oluşturmuştur. Hukuki karar alma süreçlerini modelleme, destekleme ve hukuki davaların sonuçlarını tahmin etme, 1970'lerden beri YZ ve Hukuk'un merkezi konuları olmuştur[2].

YZ'nin hukuk alanındaki kullanıma ilişkin çalışmaların arttığı gözlemlenmektedir[3]. Bu artıştaki en önemli etkenlerden birisi ise hukuk alanında, çok büyük miktarda verilerin bulunmasıdır. Bu verilerin analiz edilerek, anlamlı ve kullanılabilir bilgiye dönüştürülmesi ihtiyaç haline gelmiştir. Her yıl artan dosya sayıları ve veriler göz önüne alındığında, yıllık bazda çok büyük ölçekte artışın olduğu gözlemlenmektedir. Bu denli büyük ölçekteki veriler ile çalışmak hukuk alanında çalışanlara çok fazla yük getirmektedir. Bu yükün azaltılması için hukukda YZ'nin kullanımı önem kazanmaktadır.

Collenette ve diğerleri, yaptıkları çalışmada[4] dünya genelindeki adalet sistemleri, benzer davalara benzer şekilde davranılması gereken temel bir ilkeyi vurgulamaları gerektiğini, böylece hukuki kararlarda tutarlılık ve öngörülebilirlik sağlanabileceğini, bu ilkeyi gerçekleştirmek için bir davanın sonucunu belirleyen kişi benzer davalarda geçmişte nasıl kararlar alındığının farkında olmasının ve iyi bir neden olmadıkça bu kararları takip etmesi gerektiğini belirtmişlerdir.

Benzer konularda farklı hukukçular tarafından yapılan değerlendirmelerde, gittikçe artan veri sayısı ve değerlendirme için ayrılabilen süre yetersizliği nedeniyle farklı kararların oluşması kaçınılmazdır. Hatta farklı zamanlarda aynı konuda farklı değerlendirmelere dahi rastlanabilmektedir. Bu farklılık/çelişkilerin en aza indirilebilmesi ve karar metnininin daha kısa sürede hazırlanabilmesi yapay zeka teknolojilerinden yararlanılması

gerekmektedir. Böylece etkin çözümünde standardizasyon sağlanarak vatandaşların devlete olan bağlılık ve güven duygusunun artırılması katkı sağlayabilir.

MarketsandMarkets'e göre[5], küresel hukuki YZ yazılım pazarının 2025 yılı için tahmini değeri 2,7 milyar ABD dolarıdır. Bu, 2024 yılında tahmin edilen 2,35 milyar ABD dolarından %14,9'luk bir artış anlamına gelmektedir[6]. YZ'nin hukuk alanında pazar payının her geçen gün artması neticesinde; avukat, hakim, savcı ve hukuk profesyonellerinin iş ve işlemlerinin daha da kolaylaşması tahmin edilmektedir. Efe, yapmış olduğu çalışmada[7] YZ teknolojisinin hız, kolaylık ve ekonomiklik noktalarındaki etkisi nedeniyle hukukçuların daha yüksek katma değerli görevlere odaklanmalarına yardımcı olmak için önemli bir potansiyele sahip olduğunu belirtmiştir.

YZ hukuk alanında, metinlerin ve raporların özetlenmesi[8], mahkemeye ikna edici bir argüman sunulması[9], güncel mevzuat ve içtihat takibi[10], benzer davaları eşleştirme [11] ve karar verme mekanizmalarının yerini almadan yargıca tavsiyede bulunabilme[12][13] işlemlerini gerçekleştirebilir[14]. Dosyaların incelenmesi aşamasında, dosyaya karşı önyargılı[15] olunabilir bu nedenle benzer vakalara farklı kararlar verilebilmesi durumunun önüne de geçebilir.

Yılmaz yapmış olduğu çalışmada[3], YZ'nin yargı uygulamasında; hakime yardımcı yapay zeka, karar taslaklarını hazırlayan yapay zeka ve karar veren yapay zeka olarak üç şekilde kullanılabileceğini ön görmektedir.

Karar tahmini çalışmalarının ilk aşamalarında, çalışmalar mevcut vakaların matematiksel ve istatistiksel analizine odaklanmakta, nasıl tahmin yapılacağına dair herhangi bir metodoji ortaya koymamaktadır[16][17][18].

Yassine ve diğerleri yapmış oldukları çalışmada[19], YZ; Hukukçular tarafından gerçekleştirilen yargısal işlemlere yardımcı olduğu, avukatlar veya yargıçlar gibi ilgili bilgileri veya belgeleri sağlayabileceği ve uygun kararlar önerisinde bulunabileceğini belirtmişlerdir.

Turan ve diğerleri yapmış oldukları çalışmada[20], Harvard Hukuk Fakültesinde okuyan 4 öğrenci derin öğrenme, makine öğrenmesi ve doğal dil işleme (DDİ) modellerini kullanarak

Evisort isimli arama motorunu oluşturdıklarını ve Evisort ile hukuk sözleşmelerinin hızlı bir şekilde taslaklarını oluşturup, sonrasında yönetilmesini sağladığını ve Evisort sayesinde avukatların hayatlarından 10 yıl tasarruf ettiklerini belirtmişlerdir.

Gulyamov ve Bakhramova yapmış oldukları çalışmada[21], “hukuk alanında kullanılan bir yapay zekâ yazılımı olan eBrevi’in hukuk alanında kullanılan belgelerin daha hızlı ve etkili bir şekilde incelenmesine olanak sağladığını, belgelerin ilgili bölümlerini tanımlamak için makine öğrenimini kullandıklarını, şirketin web sitesinde yer alan açıklamaya göre 50'den fazla belgeyi bir dakikadan kısa sürede analiz edebildiğini ve manuel inceleme sürecinden yüzde 10 daha doğru sonuç verdiğini” belirtmişlerdir.

İlişkili Çalışmalar

Lawlor yapmış olduğu çalışmada[22] bilgisayarların bir gün yargı kararlarının sonuçlarını analiz ve tahmin edebilecek hale geleceğini öngörmüştür[23]. Bu öngöründe olduğu gibi 1970 yılında yapılan çalışmada[24] hukukun yapay zekadan faydalanabileceğine ilişkin fikir tartışılmıştır. Günümüzde ise oldukça popüler olan yargı kararının tahminine ilişkin literatürde bir çok uluslararası çalışma [25]-[30] bulunmaktadır.

Amerika Birleşik Devletleri Yüksek Mahkemesi'nin davranışını genelleştirilmiş, örneklem dışı bir bağlamda tahmin etmek için yapılan çalışmada [25] makine öğrenmesi random forest classifier kullanarak, 28 000 dava sonucuna göre geliştirilen modelden %71,9 doğruluk tespit etmişlerdir.

Uçtan Uca Derin Öğrenme Sinir Ağları kullanarak Tayland Yüksek Mahkemesi'ndeki ceza davaları için bir tahmin modeli öneren çalışmada [26], model yasal yorumlama sürecini taklit etmektedir. Yapılan performans testinden sonra, modelin Naïve Bayes ve Destek Vektör Makineleri (SVM) gibi geleneksel metin sınıflandırma tekniklerinden daha yüksek F1 sağlayabildiğini tespit etmişlerdir.

Hakimlerin yeni ceza davası dilekçeleri hakkındaki kararlarını tahmin etmek için regresyon analizinin kullanan çalışmada[29], 2010-2020 döneminde İsveç Yüksek Mahkemesi ve altı İstinaf Mahkemesine sunulan 3 915 adet yeni ceza davası dilekçeleri incelenmiştir. Bu veriler, daha sonra 2021 yılında dilekçelerle ilgili kararları tahmin etmek için kullanılan bir

regresyon modelinin temelini oluşturmuşlardır. Regresyon modeli, yasal temsile erişim ve suç türü temelinde, 2021 yılında verilen kararların %100'ünü doğru bir şekilde tahmin etmişlerdir.

Avrupa İnsan Hakları Mahkemesi yargı kararlarının otomatik tahmini üzerine yapılan çalışmada[30] geçmişteki davalara dayanarak gelecekteki davalar için karar tahmininde bulunmanın ortalama doğruluk değerini %58 ile %68 arasında değiştiğini tespit etmişlerdir.

Kur'an ayetleri ve hadislerini alarak Suudi mahkemelerindeki hukukçular için tasarlanmış derin öğrenme tabanlı adli karar destek sistemi çalışmasında[31], sınıflandırma aşaması %95,8'lik bir doğruluk ve F-skoru elde ederken, bilgi alma aşaması %79,8'lik bir doğruluk ve %79,3'lük bir F-skoru elde etmişlerdir.

Türkçe hukuk metinleri üzerine yapay zekanın kullanılarak karar tahmininde bulunması konusunda literatürde kısıtlı sayıda çalışma bulunmakta iken taslak karar metni oluşturma konusunda herhangi bir çalışma bulunmamaktadır. Türkçe'nin morfolojik yapısı diğer dillerden farklıdır. Sonradan eklemeli bir dil olması nedeni ile sona eklenen ek, kökün anlamını değiştirmektedir. Bu nedenle metinlerin yapay zeka ile anlamlandırılmasında bazı sorunlara sebep olmaktadır. Başka problemler ise hukuki belgelerde kullanılan terminoloji normal metinlere göre anlamsal olarak daha karmaşık ve uzun cümlelerden oluşmaktadır. Bu durum yapay zeka kullanılarak hukuki metinler üzerinde yapılan çalışmaların performans ve başarısını önemli derece etkilemektedir. Bu zorlukları aşarak yapay zeka kullanılarak Türkçe hukuki metinlerinde karar tahmini üzerine değerli çalışmalar bulunmaktadır.

Türkiye'deki yüksek mahkeme kararlarının tahminine ilişkin çalışmada [32] Karar Ağaçları (DT), Rastgele Ormanlar (RF), SVM ve derin öğrenme yöntemlerini kullanarak %93'e ulaşan doğruluk değerleri ve 0,87'ye ulaşan F1 skorları elde etmişlerdir.

Türk üst mahkemelerinin dava metinlerini kullanarak karar tahmin yöntemlerinin geliştirilmesi için yapılan çalışmada[33] İleri Beslemeli Sinir Ağları (FFNN), kelime temsilleri ve metinlerden Temel Bileşenler Analizi (PCA) ile çıkarılan öznitelikler kullanılarak eğitimler yapmışlar ve karar tahmini için %85,4 Makro F1 skoruna ulaşmışlardır.

GRU ve LSTM BiLSTM derin öğrenme modellerini kullanarak Yargıtay davalarının sonuçlarını tahmin edildiği çalışmada [34] yapılan deneyler sonucunda en başarılı karar tahmini yapan modelin GRU olduğunu tespit etmişlerdir. GRU modeli ile karar tahmininde %96,8 doğruluk skoruna ulaşmışlardır.

Yüksek mahkeme kararından oluşan 90 bin dosya sayısını içeren çalışmada [35] verilerini çeşitli sınıflandırıcılarla, sınıflandırma deneyleri yapmış. BERT modellerinin diğer sınıflandırıcılardan büyük bir farkla daha iyi performans gösterdiğini ve F1 skorunda yaklaşık %2 artışa yol açtığını tespit etmiştir.

Tezin Literatüre Katkısı

Bu tezin amacı, yapay zeka algoritmaları kullanılarak yargı uygulamaları için örnek karar destek sistemi oluşturmaktır. Oluşturulan sistem hukuk sisteminde daha verimli karar verme sürecine yönelik önemli bir potansiyele sahiptir. Oluşturulan sistem karar tahmini ve taslak karar metin oluşturma olmak üzere iki kısımdan oluşmaktadır. Çalışmada Kamu Denetçiliği Kurumu (Ombudsmalık) web sayfasında yayınlanan kişisel verilerden arındırılmış veriler ile veri seti oluşturulmuştur. Veri seti ile eğitilen makine öğrenmesi ve derin öğrenme modelleriyle karar tahmininde bulunulmuştur. Fine-tuning (ince ayar) işleminde oluşturulan veri seti ile eğitim yapılarak taslak karar metni üretimi (text generation) yapılmıştır. Karar tahmini için denetimli altı farklı makine öğrenme ve beş farklı derin öğrenme modelleri kullanılmıştır. Elde edilen sonuçlar ile farklı modellerin performansları karşılaştırmalı ve detaylı bir şekilde incelenmiştir. Test veri seti üzerinde Ombudsmanlık kararlarını en başarılı şekilde tahmin eden makine öğrenmesi modeli SVM, derin öğrenme modeli ise GRU ve LSTM olduğu gözlemlenmiştir. Taslak karar metni oluşturmak için ytu-ce-cosmos/turkish-gpt2-large modeli ile ince ayara işleme yapılmıştır. Karar tahmininde SVM modeli ile %61, GRU ve LSTM modeli ile %64 F1 skoru değerine ulaşılmıştır. İnce ayar işleminde eğitim kaybı 0,26 ve doğrulama kaybı 0,23 olan başarılı bir model elde edilmiştir.

Bu tez altı bölümden oluşmaktadır ve şu şekilde düzenlenmiştir. İkinci bölümde veri setinin hazırlanması ve yöntemleri açıklanmıştır. Üçüncü bölümde karar tahmini ve yöntemleri açıklanmıştır. Dördüncü bölümde taslak karar metni oluşturma ve kullanılan teknikler

açıklanmıştır. Beşinci bölümde deneysel sonuçlara, altıncı bölümde ise çalışmanın sonuçlarına yer verilmiştir.



2. VERİ SETİNİN HAZIRLANMASI

Ülkemizde Ombudsmanlık Kurumu'na ulaşan vatandaş şikâyet ve talepleri, alanlarında yetişmiş uzmanlar veya bu işle ilgili geçici görevli personel (hakim/savcı/avukat ..vb) tarafından titizlikle incelenmekte ve en kısa sürede yanıtlanmaya çalışılmaktadır. Ombudsmanlık kurumuna gelen şikâyet başvuruları türü, kapsamı ve konusuna göre dosya bazlı olarak incelenmektedir. İnceleme işlemi tamamlandıktan sonra şikâyetle ilgili karar verilmektedir. Kararın neticesi, ilgili idare ve başvurana bildirilerek vatandaş ile idare arasındaki sorunları çözülmektedir.

Bu bölümde, modellerimizi eğitmek için Ombudsmanlık kararlarını veri setine dönüştürüyoruz. Dönüştürülen tüm kararlar Ombudsmanlık web sayfasında çevrimiçi olarak sunulan verilerden elde edilmiştir. Bu çalışmada, çalışma zamanında mevcut olan 10 600 dosya kararı kullanılmıştır. Kararların veri setine dönüştürülmesi için uygulanan veri toplama ve ön işleme adımları aşağıda detaylı olarak açıklanacaktır.

2.1. Veri Toplama

Bu çalışmada, çalışma zamanında, Resim 2.1.'de gösterilen ombudsmanlık web sayfası kararlar bilgi bankasında var olan 10 600 karar, geliştirilen yazılım ile indirilmiştir. İndirilen kararlar; şikâyet numarası, şikâyet konusu, şikâyet alt konusu ve karar tipi kolonlarına ayrılarak .csv formatında kaydedilmiştir. Kararlar Optical Character Recognition - Optik Karakter Tanıma (OCR) işlemine tabi tutulmuştur.



Başvuru No	Karar Tarihi	Karar Türü	Karar Konusu	Başvuru Konu İdare
2024/874	09/05/2024	Ret Kararı	Dağıtıcı kadrosuna atanma talebi hakkındadır.	POSTA VE TELGRAF TESKİLATI ANONİM ŞİRKETİ GENEL MÜDÜRLÜĞÜ
2024/812	03/04/2024	Kısmen Tavsiye Kısmen Ret Kararı	Cezai nakil talebine ilişkindir.	CEZA VE TEVKİFİLERİ GENEL MÜDÜRLÜĞÜ
2024/797	30/04/2024	Tavsiye Kararı	Faiz talebi hakkındadır.	SOSYAL GÜVENLİK KURUMU BAŞKANLIĞI
2024/795	30/05/2024	Tavsiye Kararı	Başvurunun idesi yapılan primlerine faiz işletilmesi talebi hakkındadır.	SOSYAL GÜVENLİK KURUMU BAŞKANLIĞI
2024/763	19/04/2024	Ret Kararı	Unvan Değişikliği Talebi Hakkında	SAĞLIK BAKANLIĞI
2024/760	14/05/2024	Ret Kararı	Koruyucu giyim ve donanım malzemesi talebine ilişkindir.	TOPRAK MAHSULLERİ OFİSİ GENEL MÜDÜRLÜĞÜ
2024/753	24/05/2024	Tavsiye Kararı	Afsetzedelere yapılan destek ödemeleri hakkındadır.	AFET VE ACİL DURUM YÖNETİMİ BAŞKANLIĞI HATAY İL AFET VE ACİL DURUM MÜDÜRLÜĞÜ
2024/729	06/05/2024	Ret Kararı	Meslek kodu değişikliği hakkındadır.	SAĞLIK BAKANLIĞI
2024/705	16/05/2024	Kısmen Tavsiye Kısmen Ret Kararı	Cezai nakil talebi hakkındadır.	ADALET BAKANLIĞI

Resim 2.1. Kamu denetçiliği kurumu kararlar bilgi bankası

2.2. Veri Önışleme

Ombudsmanlık web sitesinde kararlar; başvuru numarası, karar türü ve başvuru konusuna göre tablolarla sunmaktadır. Bu tablolarda yer alan karar konu bilgisine göre sınıflandırılma işlemleri yapılmıştır. Çizelge 2.1’de, başvuru konusuna göre karar dağılımı gösterilmektedir.

Çizelge 2.1. Başvuru konusuna göre karar sayısı

Başvuru Konusu	Karar Sayısı
Kamu personel rejimi	4058
Adalet, milli savunma ve güvenlik	1552
Eğitim-öğretim, gençlik ve spor	1150
Çalışma ve sosyal güvenlik	1146
Ekonomi, maliye ve vergi	725
Mahalli idarelerce yürütölen hizmetler	590
Mülkiyet hakkı	219
Sağlık	207
Orman, su, çevre ve şehircilik	194
Engelli hakları	154
Enerji, sanayi, gümrük ve ticaret	147
Ulaştırma, basın ve iletişim	140
Sosyal hizmetler	90
İnsan hakları	82
Nüfus, vatandaşlık, mülteci ve sığınmacı hakları	77
Bilim, sanat, kültür ve turizm	38
Gıda, tarım ve hayvancılık	37
Çocuk Hakları	33
Ailenin korunması	33

için ise “KARAR” kısmında yer alan metinler alınarak yeni bir alan olarak kaydedilmiştir.

Karar türü alanı, metin verileri içerdiği için yeni bir KARAR_ID alanı eklenmiş ve {0: Kısmen Tavsiye Kısmen Ret Kararı, 1: Tavsiye Kararı, 2: Ret Kararı ,3: Kısmi Tavsiye Kararı} etiketleme işlemi yapılmıştır. Boş ve hatalı olan alanların olduğu sütunlar datasetinin içerisinden çıkarılmıştır. Boş ve hatalı kısımlar datasetinden çıkarıldığında 10 600 karardan, 9 674 karar kullanılabilir hale getirilmiştir. Kısmi ret kararından 2 adet karar olduğu ve veri setinde dengesizlik oluşturmaması için ret kararı olarak etiketlenmiştir.

Yapılan düzenlemelerden sonra veri setinde yer alan karar türü dağılımı Çizelge 2.2’de gösterilmiştir. Çizelgede yer alan verilere göre karar türü dağılımının sayısal olarak dengesiz olduğu karar türüne göre en fazla verinin olduğu karar türü ret kararı iken en az veriyi barındıran karar türü kısmi tavsiye kararı olarak gözlemlenmiştir.

Çizelge 2.2. Karar türüne göre dosya dağılımı

No	Karar Türü	Dosya Sayısı
1	Ret Kararı	4 343
2	Tavsiye Kararı	3 757
3	Kısmen Tavsiye Kısmen Ret Kararı	1 377
4	Kısmi Tavsiye Kararı	197

Veri setinin içinde yer alan metinler aşağıdaki ön işlem adımlarına tabi tutulmuştur:

- Metinlerin küçük harfe dönüştürülmesi (Lowercase).
- Noktalama işaretlerinin ve özel karakterlerin temizlenmesi (Punctuation).
- Gereksiz boşlukların kaldırılması.
- Durdurma kelimelerinin kaldırılması (Stopword).

2.2.1. Metinlerin küçük harfe dönüştürülmesi

Metinlerin küçük harfe dönüştürülmesi (Lowercase) işlemi, metin içinde bulunan tüm harflerin küçük harfe dönüştürülmesi işlemidir. Bu işlem verinin tutarlı bir formatta olmasını sağlamaktadır. Veri setlerinin homojen bir yapıda olmasını sağlayarak daha etkili bir analiz

yapılmasına yardımcı olmaktadır. Python’da veya diğer birçok programlama dilinde lower() fonksiyonu kullanılarak metinde yer alan büyük harfler küçük harfe dönüştürülür.

2.2.2. Noktalama işaretlerinin ve özel karakterlerin temizlenmesi

Noktalama işaretlerinin ve özel karakterlerin temizlenmesi (Punctuation) işlemi, noktalama işaretleri ve özel karakterlerin metin içinden kaldırılması işlemidir. Bu işlem metinlerin daha düzenli ve homojen yapıda olmasını sağlamaktadır. Noktalama işaretleri ve özel karakterler genellikle kelime sayımı, metin analizi veya modellerin eğitimi esnasında gereksiz gürültüye neden olabilir ve analiz işlemini zorlaştırabilirler. Bu nedenle, oluşturulan veri setlerinin ön işleme aşamasında noktalama ve özel işaretlerin kaldırılması gerekmektedir.

2.2.3. Gereksiz boşlukların kaldırılması

Gereksiz boşlukların kaldırılması işlemi, metin içinde fazladan veya yanlış yerleştirilmiş boşlukların metinlerin içinden kaldırılması işlemidir. Metin içindeki gereksiz boşlukları kaldırmak için; yan yana gelen birden fazla boşluk tek boşlukla değiştirilir, metnin başında/sonunda yer alan boşluklar kaldırılır ve metinlerdeki boşlukluklar kelimeler arasında uygun bir şekilde düzenlenerek gerçekleştirilir. Bu işlemler metin verilerinin daha temiz, düzenli ve analiz için uygun hale getirilmesini sağlamaktadır.

2.2.4. Durdurma kelimelerinin kaldırılması

Durdurma kelimelerinin (Stopword) kaldırılması işlemi, metin verilerinde bulunan yaygın ve anlamsal olarak düşük değere sahip kelimelerin kaldırılması veya filtrenilmesi işlemidir. Stopword’ler cümlelerde sıklıkla geçerler, örneğin “ve”, “ama”, “ise”, “veya”, vb. ancak anlamsal açıdan içeriğe katkıda bulunmadıkları ve metin analizi sırasında gereksiz gürültüye neden oldukları için metin içinde filtreleme işlemi yapılmaktadır. Bazı durumlarda, stopword’ler metinlerin anlamlarını etkileyebilir bu nedenle analiz yapılacak metinlerin içeriği ve amaçları göz önünde bulundurularak stopword işlemi yapılmalıdır.

Kelimenin köküne inildiğinde örneğin “yalnızlaştırıldığını” kelimesinde “yanlız” kökü elde edilmektedir. Bu durumda cümlenin anlam bütünlüğünü bozulmakta ve anlam karmaşasına sebep olmaktadır. Bu durumlar göz önünde bulundurulduğu için kök bulma (stemming) ve lemmatization veri ön işleme adımları uygulanmamıştır.



3. KARAR TAHMİNİ

Dünya genelinde sürekli değişen yasal ortam, yasa ve yönetmeliklerin sürekli güncellenmesini gerektirmektedir. Karar vericilerin bu dinamik ortamda yollarını bulabilmeleri için gelişmelerden haberdar olmaları ve yeni yasal çerçevelere ilişkin bilinçli kararlar alabilmeleri gerekmektedir. Bu sadece karar alma sürecini kolaylaştırmakla kalmaz, aynı zamanda yasal kararlarda tutarlılığı da teşvik eder. Metin tanıma araştırmaları çok sayıda pratik uygulama ortaya koymuş olsa da, literatürdeki odak noktası öncelikle İngilizce olmuştur. Bu durum, hukuk alanında, metin sınıflandırma tekniklerinin Ural-Altay dil ailesine ait olan ve farklı gramer yapılarına sahip Türkçe hukuk metinlerine doğrudan uygulanmasının önünde bir engel teşkil etmektedir.

Hukuki söylemdeki argümanların tanımlanması, sınıflandırılması ve analiz edilmesinin argüman madenciliği alanının başlangıcından bu yana önemli bir araştırma alanı olmuştur [36]. Hukuki sınıflandırmada, mevcut vaka kararlarına dayanarak, davayla ilgili olarak olası verilebilecek kararın tahmini sınıflandırma uygulamasından bir tanesidir [37]-[39].

Bu çalışma kapsamında, Ombudsmanlık tarafından daha öncesinde verilen kararlar ile modellerimizi denetimli bir şekilde eğittik. Eğitilen makine öğrenmesi ve derin öğrenme modelleri ile Ombudsmanlık tarafından verilen kararlar (ret kararı, tavsiye kararı, kısmen tavsiye kısmen ret kararı ve kısmi tavsiye kararı) tahmin edilmiştir. Karar tahmini için eğitilen modeller ve modellerin değerlendirilmesi için kullanılan teknik ve yöntemler ilgili bölümlerde açıklanmıştır.

Çalışmada geliştirilen tüm kodlar Python dilinde kodlanmıştır. Tüm kodlar Google Colab pro platformu üzerinde koşturulmuştur.

3.1. Makine Öğrenmesi Modelleri

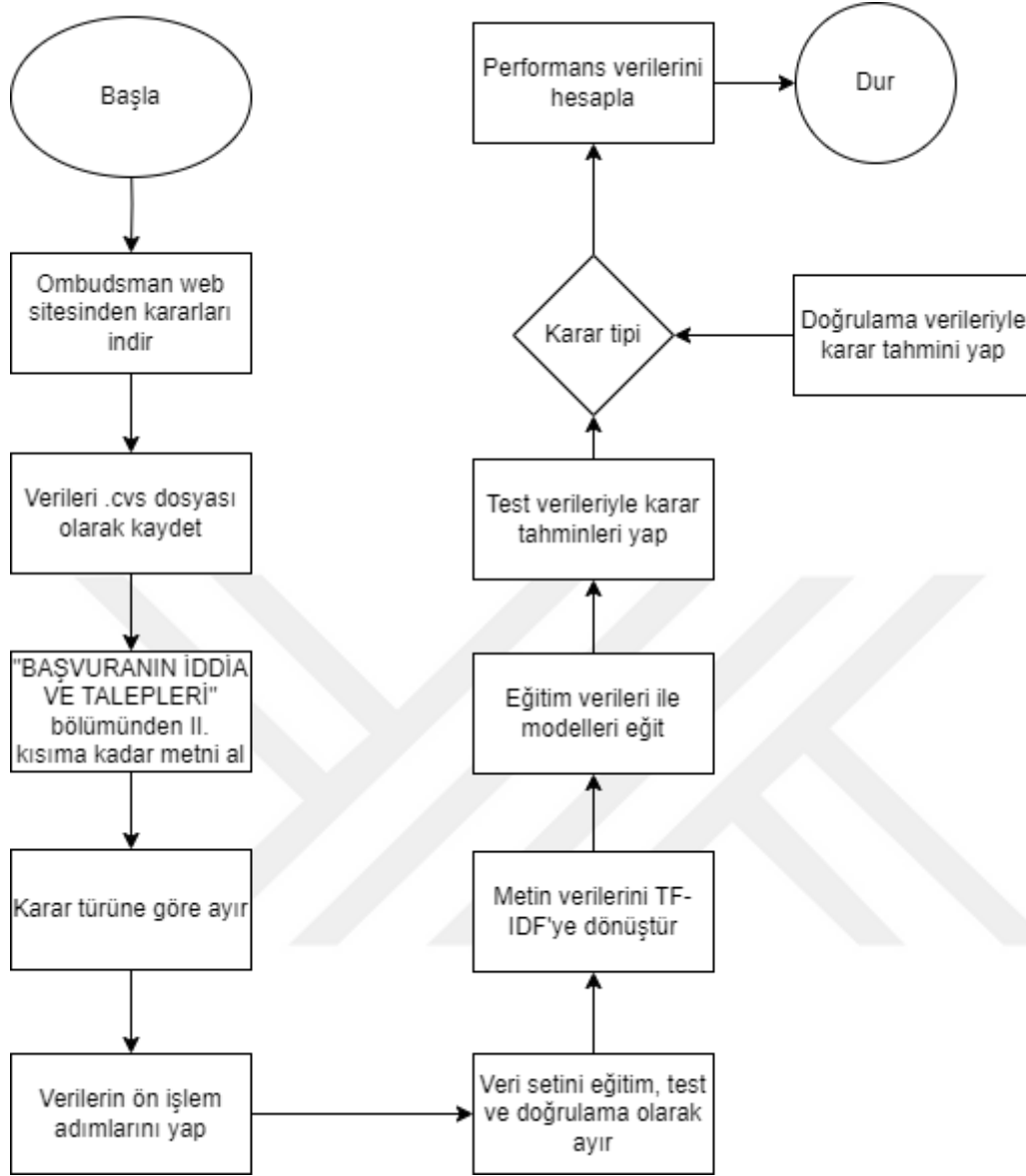
Makine öğrenmesi, sürekli gelişen ve her geçen gün pratik uygulamalarda karşımıza çıkmaktadır. Hukuk verilerine dayalı karar tahmini süreçlerini optimize etmek ve otomatikleştirmek için güçlü araçlar sunmaktadır. Bu kapsamda Kamu Denetçiliği Kurumu kararlarını tahmin etmek için makine öğrenmesi modelleri ile hukuki metin sınıflandırma işlemi yapılmıştır. Bu modeller gözetimli öğrenme modeli olan DT, SVM, Naïve Bayes,

KKN, LR ve XGBoost'dur. Modeller eğitim verilerini ile eğitilmişlerdir. Modeller ilgili kısımlarda detaylı olarak açıklanmıştır.

Çalışmanın bu kısmında Ombudsmanlık kararları ile makine öğrenmesi modelleri denetimli bir şekilde eğitilerek karar tahmini yapılmıştır.

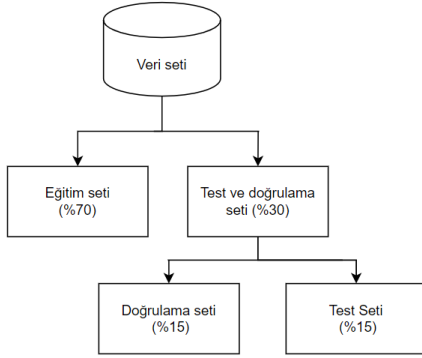
Çalışmada Ombudsmanlık web sayfasında kamuya açık olarak yayınlanan kararlar veri seti olarak kullanılmıştır. Çalışmanın aşamaları Şekil 3.1'de görülmektedir. Web sayfasından indirilen kararlar .csv formatında kaydedilmiştir. Karar metninde yer alan "BAŞVURANIN İDDİA VE TALEPLERİ" 'nden başlayarak II. Bölüme kadar olan kısım metin olarak alınmıştır. Metinler karar türlerine göre {0: Kısmen Tavsiye Kısmen Ret Kararı, 1: Tavsiye Kararı, 2: Ret Kararı, 3: Kısmi Tavsiye Kararı} olarak sınıflandırılmıştır. Veri setine ön işlem adımları uygulanmıştır. Fakat durdurma kelimelerini filtreleme işlemi cümlenin anlam yapısını bozacağı için uygulanmamıştır. Veri seti üç bölüme ayrılmıştır. Derlenen modeller eğitim verisi içinde yer alan karar tipi ve başvuranın iddia ve talepleri kısmında yer alan verilerle eğitilmiştir.

Test veri kümesi ile karar tahmininde yapılmıştır. Modellerin performans değerlendirmeleri hesaplanmıştır. Uygulanan işlem adımlarının nasıl işlendiği ile ilgili bilgiler akış sırasına göre açıklanmıştır.



Şekil 3.1. Makine öğrenmesi modelleri ile karar tahmini uygulaması akış şeması

Çalışmada “.csv” formatında dönüştürülmüş olan karar veri seti eğitim, test ve doğrulama bölümlerine ayrılmıştır. Şekil 3.2’de gösterilmektedir. Veri setinin bölümlere ayrılmasında sklearn.model_selection kütüphanesinde yer alan train_test_split sınıfı kullanılmıştır. Eğitim, test ve doğrulama olarak şekilde görüldüğü gibi %70-%15-%15 oranında veriler ayrılmıştır.



Şekil 3.2. Veri setinin ayrılması

Veri setine ön işlem adımları ve TF-IDF dönüştürme işlemleri uygulanmıştır. Ön işlem uygulamaları farklı Python standart kütüphanesi olan “re” ve “string” kütüphaneleri kullanılmıştır.

Metin verilerini makine öğrenme modelleri için uygun bir formata yani sayısal veri olan terim frekansı-ters doküman frekansı (TF-IDF) vektörüne dönüştürülmüştür. Bu dönüştürme işlemi ile metin içerisindeki kelimelerin ne kadar sık geçtiği terim frekansı (TF) ve tüm metinler içinde ne kadar sık yer aldığına ters doküman frekansı (IDF) bakılarak, o kelimenin o metin için ne kadar önemli olduğu tespit edilmiştir.

TF-IDF dönüşümünde `sklearn.feature_extraction.text` kütüphanesinde yer alan `TfidfVectorizer` sınıfı kullanılmıştır. `max_features=1 000` parametresi ile en fazla 1 000 özelliği (kelimeyi) kullanacak şekilde vektörizeri sınırlandırılmıştır. TF-IDF vektörleri oluşturulurken, en yüksek skora sahip 1 000 kelimenin seçilmesi sağlanmıştır.

Bir kelimenin metin içerisinde ne kadar sık geçtiğini Terim TF ile tespit edebilmek için kullanılan matematiksel hesabı Eş. (3.1)'de görülmektedir. $TF(t,d)$ kelime t 'nin belge d 'de terim frekansını, $n(t,d)$ belge d 'de kelime t 'nin toplam görülme sayısını ve $N(d)$ Belge d 'deki toplam kelime sayısını ifade etmektedir.

$$TF(t,d) = n(t,d) / N(d) \quad (3.1)$$

Bir kelimenin tüm dokümanlar içinde ne kadar sık yer aldığını IDF ile tespit etmek için kullanılan matematiksel hesabı Eş. (3.2)'de görülmektedir. $IDF(t)$ kelime t 'nin ters belge

frekansını, N tüm dokümanların toplam sayısını ve $n(t)$ kelime t 'nin tüm dokümanlarda görülme sayısını ifade etmektedir. Nadir görülen kelimeler, daha sık görülen kelimelere göre daha önemli kabul edilmektedir.

$$\text{IDF}(t) = \log(N / n(t)) \quad (3.2)$$

TF-IDF değeri yukarıdaki iki metrik birleştirilerek hesaplanmaktadır. Kullanılan matematiksel hesabı Eş. (3.3)'de görülmektedir. TF-IDF(t,d) kelime t 'nin belge d 'deki TF-IDF değerini, TF(t,d) kelime t 'nin belge d 'de terim frekansını ve IDF(t) kelime t 'nin ters belge frekansını temsil etmektedir.

$$\text{TF-IDF}(t,d) = \text{TF}(t,d) * \text{IDF}(t) \quad (3.3)$$

Örneğin;

Belge 1: "savunma hakkı"

Belge 2: "bilgi edinme hakkı" için TF-IDF işlem adımları detaylı olarak aşağıda açıklanmıştır.

TF (Term Frequency) Hesaplaması

Belge 1 için "savunma" ve "hakkı" terimlerinin TF'si:

$$\text{TF}(\text{savunma}, \text{Belge 1}) = 1 / 2 = 0,5$$

$$\text{TF}(\text{hakkı}, \text{Belge 1}) = 1 / 2 = 0,5$$

Belge 2 için "bilgi", "edinme" ve "hakkı" terimlerinin TF'si:

$$\text{TF}(\text{bilgi}, \text{Belge 2}) = 1 / 3 \approx 0,33$$

$$\text{TF}(\text{edinme}, \text{Belge 2}) = 1 / 3 \approx 0,33$$

$$\text{TF}(\text{hakkı}, \text{Belge 2}) = 1 / 3 \approx 0,33$$

IDF (Inverse Document Frequency) Hesaplaması

$$\text{IDF}(\text{savunma}) = \log(2 / 1) = \log(2) \approx 0,301$$

$$\text{IDF}(\text{hakkı}) = \log(2 / 2) = \log(1) = 0$$

$$\text{IDF}(\text{bilgi}) = \log(2 / 1) = \log(2) \approx 0,301$$

$$\text{IDF}(\text{edinme}) = \log(2 / 1) = \log(2) \approx 0,301$$

TF-IDF Hesaplaması

Belge 1 için TF-IDF:

$$\text{TF-IDF}(\text{savunma, Belge 1}) = 0,5 * 0,301 \approx 0,1505$$

$$\text{TF-IDF}(\text{hakkı, Belge 1}) = 0,5 * 0 = 0$$

Belge 2 için TF-IDF:

$$\text{TF-IDF}(\text{bilgi, Belge 2}) = 0,33 * 0,301 \approx 0,0993$$

$$\text{TF-IDF}(\text{edinme, Belge 2}) = 0,33 * 0,301 \approx 0,0993$$

$$\text{TF-IDF}(\text{hakkı, Belge 2}) = 0,33 * 0 = 0$$

Şekil 3.3'te her bir terim için TF-IDF değerleri gösterilmiştir:

Belge 1:

"savunma" terimi için TF-IDF $\approx 0,1505$

"hakkı" terimi için TF-IDF = 0

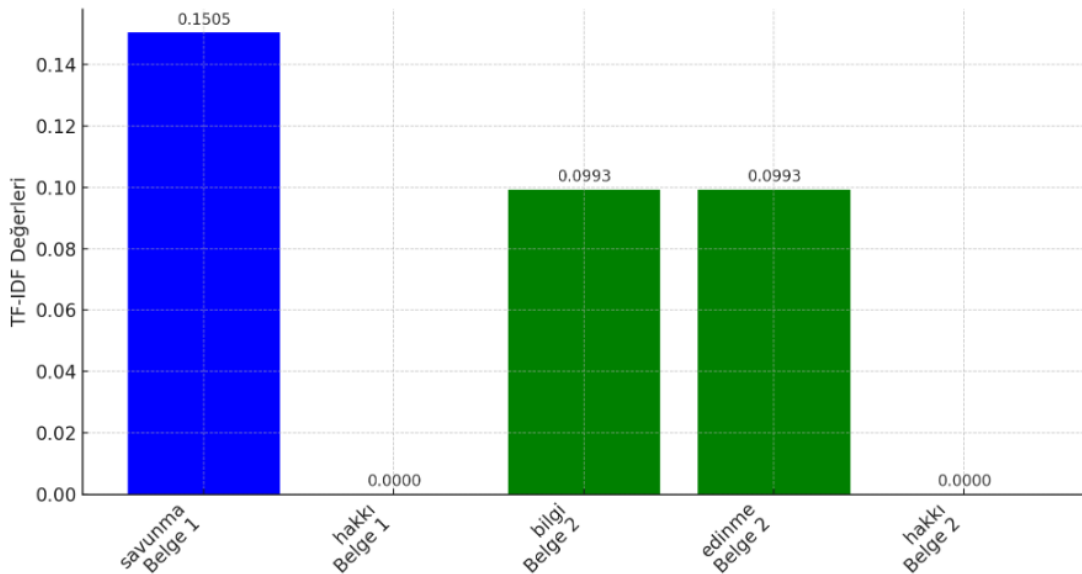
Belge 2:

"bilgi" terimi için TF-IDF $\approx 0,0993$

"edinme" terimi için TF-IDF $\approx 0,0993$

"hakkı" terimi için TF-IDF = 0

Şekil 3.1'de terimlerin belgelerdeki önemini karşılaştırarak hangi terimlerin belirleyici olduğu gösterilmektedir. "savunma", "bilgi" ve "edinme" terimleri yüksek TF-IDF değerlerine sahipken, "hakkı" terimi her iki belgede de geçtiği için TF-IDF değeri 0 olmuştur.



Şekil 3.3. TF-IDF değerleri

Modeller eğitildikten sonra test verileri üzerinden karar tahmini yapılmıştır. Modellerin performansları DENEYSEL SONUÇLAR başlığında altında sunulmuştur.

3.1.1. Karar ağacı

Karar ağaçları (DT), metin sınıflandırma işlemlerinde sıkça kullanılan bir modelleme yöntemidir.

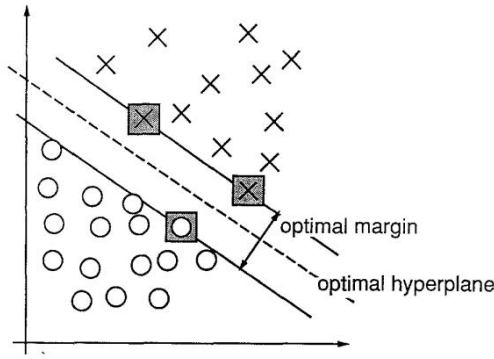
Bu yöntem, karar verme süreçlerini görselleştirmek ve analiz etmek için kullanılır[40]. Rus mahkeme kararlarının yarı yapılandırılmış verilerinin (yaklaşık 30 milyon) işlenmesi ve analizinin sonuçlarının açıklanması için karar ağaçları kullanılmıştır[19][41].

Karar ağaçları, verileri bir dizi karar düğümü ve sonuç yaprakları aracılığıyla sınıflandırır. Karar ağacının yaprakları sınıf adlarıdır, diğer düğümler ise her olası sonuç için bir dal içeren öznitelik tabanlı testleri temsil eder. Bir nesneyi sınıflandırmak için ağacın kökünden başlar, testi değerlendirir ve sonuca uygun dalı alır. Süreç bir yaprakla karşılaşılan kadar devam eder[42].

3.1.2. Destek vektör makineleri

Destek Vektör Makineleri (SVM), gözetimli öğrenme algoritmalarından biridir ve sınıflandırma problemlerinde kullanılabilir. Düzlem üzerindeki noktaları ayırmak için bir doğru veya hiper düzlem çizerek çalışırlar. Bu doğrunun, iki sınıfının noktaları için de maksimum uzaklıkta olmasını amaçlar. Optimal marjın, sınıflar arasındaki ayırıcı çizginin her iki sınıfa ait noktalardan da mümkün olduğunca uzakta olması gerektiren bir prensiptir. Bu sayede, karar sınırının daha genellenebilir ve gürültüye karşı daha dirençli olması sağlanmaktadır. Marj ne kadar geniş ise sınıflandırma o kadar başarılı olur[43]. Şekil 3.2'de gri karelerle işaretlenmiş destek vektörleri, iki sınıf arasındaki en büyük ayırma marjını tanımlar [44].

Optimal hiperplan, bu prensibe göre bulunan ve iki sınıfı en iyi şekilde ayıran hiper düzlemi temsil eder. Bu hiper düzlem, SVM algoritmasının karar verme aşamasında kritik rol oynar ve yeni verilerin hangi sınıfa ait olduğunu belirlemede kullanılır.



Şekil 3.4. Destek vektör makinesi

3.1.3. Naïve bayes

Naïve Bayes, olasılıksal sınıflandırma algoritmalarından biridir ve adını Bayes teoreminden alır. Metin sınıflandırma için yaygın olarak kullanılmaktadır[45]. Bu algoritma her bir özelliğin sınıfa katkısının diğer özelliklerden bağımsız olduğu varsayılmaktadır. Algoritmanın denklemleri Eş. (3.4) ve (3.5)'te gösterilmektedir[46]. Eş. (3.4)'te $P(c | x)$: Yeni veri örneği "x" için sınıf "c"nin arka olasılığı (posterior probability), yani "x" göz önüne alındığında "c" sınıfına ait olma olasılığını, $P(c)$: Sınıf "c"nin ön olasılığı (prior probability), yani verideki tüm örnekler arasında "c" sınıfına ait olma olasılığını, $P(x | c)$: "x" veri örneğinin, sınıf "c"nin özelliklerine sahip olma olasılığını, yani "c" sınıfı geçerliyse "x" özelliklerinin gerçekleşme olasılığını ve $P(x)$: "x" veri örneğinin gerçekleşme olasılığını ifade etmektedir.

$$P(c | x) = (P(c) * P(x | c)) / P(x) \quad (3.4)$$

Eş. (3.4)'te kritik nokta, $P(x | c)$ hesaplamasıdır. Naïve Bayes, özelliklerin birbirinden bağımsız olduğunu varsaymaktadır. Bu varsayım her zaman doğru olmayabilir ancak hesaplamayı basitleştirir. Bağımsızlığı varsayarak, $P(x | c)$ 'nin hesaplanması Eş. (3.5)'te gösterilmektedir. x_1, x_2, \dots, x_n : verinin sahip olduğu n adet özelliğini ve $P(x_i | c)$: "i"inci özelliğin, sınıf "c"nin özelliklerine sahip olma olasılığını ifade etmektedir.

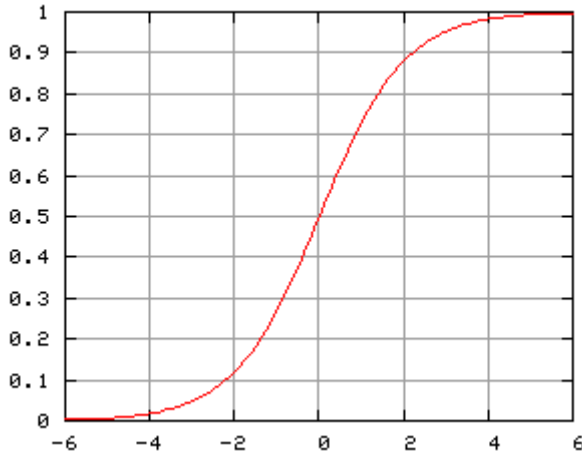
$$P(x | c) = P(x_1 | c) * P(x_2 | c) * \dots * P(x_n | c) \quad (3.5)$$

3.1.4. K-en yakın komşu

K-En Yakın Komşu (KNN) algoritması, sınıflandırma ve regresyon problemlerinde yaygın olarak kullanılmaktadır. KNN algoritması, eğitim aşamasında veri noktaları için belirli bir model oluşturmaz. Bunun yerine, sınıflandırma veya tahmin yaparken eğitim veri setinden en yakın K komşusunun (benzer özelliklere sahip veri noktaları) sınıfına veya değerine dayanarak tahmin etmektedir[47].

3.1.5. Lojistik regresyon

Lojistik regresyon (LR), denetimli öğrenme kategorisinde yer alan ve sınıflandırma problemlerinde kullanılan bir modeldir. Bu model, bir veya daha fazla bağımsız değişken (özellik) ile bir bağımlı değişken (sınıf) arasındaki ilişkiyi modellemek için kullanılmaktadır[48]. LR, logistik fonksiyon adı verilen bir fonksiyonu kullanır. Bu fonksiyon, bağımsız değişkenlerin bir lineer kombinasyonlarını alır ve 0 ile 1 arasında bir olasılık değeri döndürür. Bu olasılık değeri, veri noktasının belirli bir sınıfa ait olma olasılığını temsil etmektedir. Şekil (3.3)'te[49] gösterilen y değerleri 0'dan küçük veya 1'den büyük olamaz.



Şekil 3.5. Lojistik regresyon

3.1.6. XGBoost

XGBoost (Aşırı Gradyan Artırma), karar ağacı temelli ve eğim artırma yöntemini kullanan yüksek performanslı bir makine öğrenmesi algoritmasıdır[50]. 2016 yılında Tianqi Chen ve Carlos Guestrin tarafından geliştirilmiş ve hızla en popüler makine öğrenmesi

algoritmalarından biri haline gelmiştir. Özellikle büyük veri setlerinde yüksek performans gösterir ve Kaggle gibi veri bilimi yarışmalarında sıklıkla kullanılır. XGBoost, gradyan artırma (gradient boosting) algoritmasının optimize edilmiş bir versiyonudur ve çeşitli iyileştirmelerle daha hızlı ve daha hassas sonuçlar sağlamaktadır[51].

XGBoost, hatayı minimize etmek için zayıf öğrenen (genellikle küçük karar ağaçları) topluluğunu iteratif olarak oluşturur. Her zayıf öğrenen, önceki modellerin hatalarını düzeltmeye çalışmaktadır. Eş. (3.6)'da XGBoost deklemini gösterilmektedir. Denklemden $f(x)$: yeni bir veri noktası "x" için nihai tahmini, $f_0(x)$: ilk zayıf öğrenenin tahminini, η_j : "j"inci zayıf öğrenenin öğrenme oranını ve $h_j(x)$: "j"inci zayıf öğrenenin tahminini ifade etmektedir.

$$f(x) = f_0(x) + \sum \eta_j * h_j(x) \quad (3.6)$$

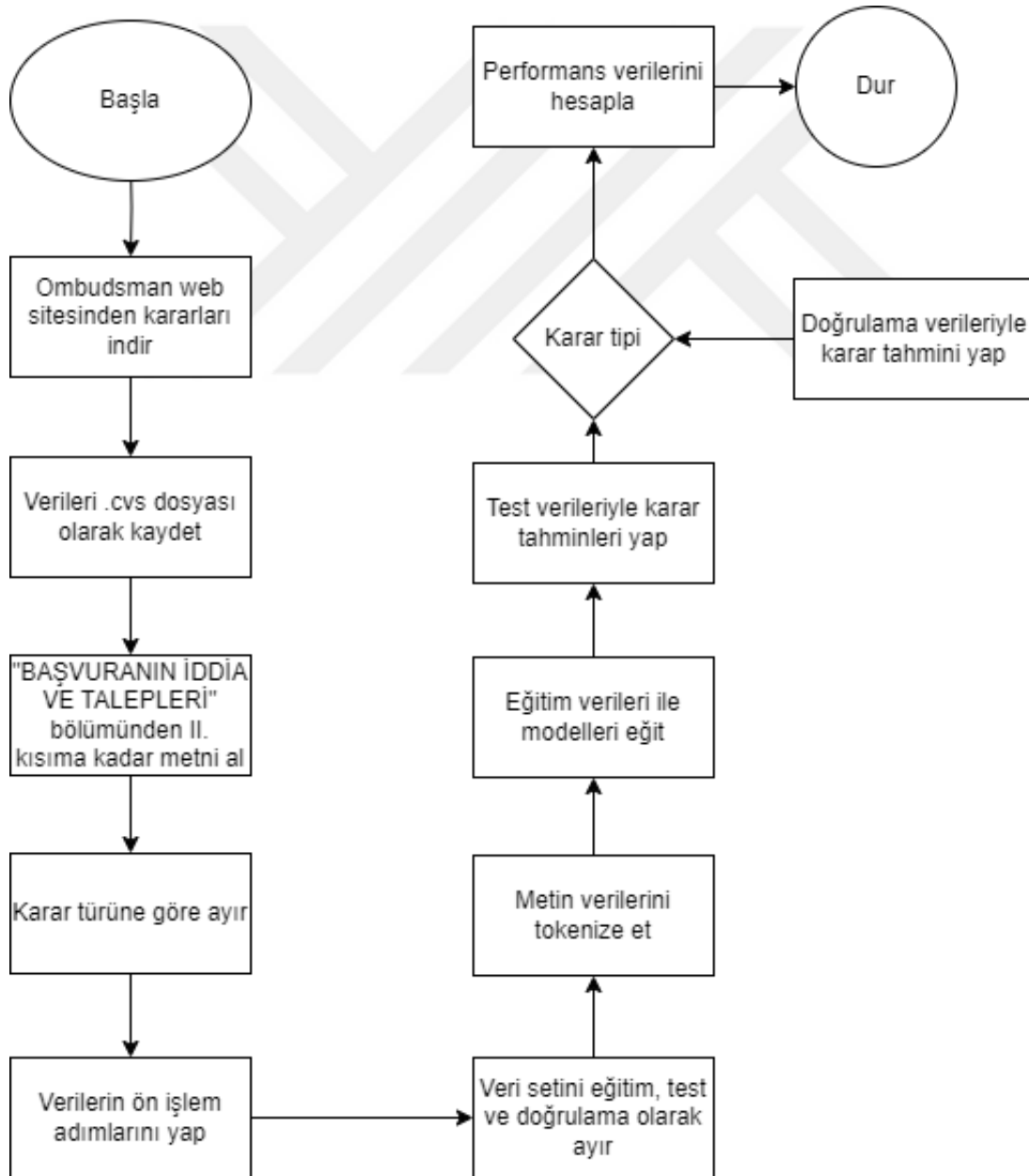
3.2. Derin Öğrenme Modelleri

Derin öğrenme, yapay zekânın bir alt dalıdır ve insan beyninden esinlenerek geliştirilmiş yapay sinir ağları kullanarak karmaşık problemleri çözmeye odaklanmaktadır. Bu ağlar, çok katmanlı bir yapıya sahiptir ve her katman, veriyi daha da soyutlayarak ve analiz ederek çalışmaktadır. Hukuki metinlerin sınıflandırılmasında etkili bir araçtır. Özellikle büyük ve karmaşık hukuki veri setlerinde derin öğrenme modelleri, manuel yöntemlerle elde edilemeyecek derecede yüksek doğruluk elde edebilmektedir[52].

Bu kapsamda Kamu Denetçiliği Kurumu kararlarını tahmin etmek için derin öğrenme modelleri ile hukuki metin sınıflandırma işlemi yapılmıştır. Karar tahmini için eğitilen modeller CNN, FFNN, GRU, LSTM ve BiLSTM'dir. Modeller eğitim verileri ile eğitilmişlerdir. Modeller ilgili kısımlarda detaylı olarak açıklanmıştır.

Çalışmanın bu kısmına makine öğrenmesi modellerinde kullanılan veri setinin aynısı kullanılmıştır. Çalışmanın aşamaları Şekil 3.6'da görülmektedir. Veri seti eğitim, test ve doğrulama bölümlerine ayrılmıştır. Veri setinin bölümlere ayrılmasında sklearn.model_selection kütüphanesinde yer alan train_test_split sınıfı kullanılmıştır. Eğitim, test ve doğrulama olarak %80-%10-%10 oranında veriler ayrılmıştır.

Metin verilerini derin öğrenme modelleri için uygun bir formata tokenization işlemi ile dönüştürülmüştür. Tokenization işlemi, metin verilerini sayısal vektörlere dönüştürmek için kullanılmıştır. Tokenizer ile her kelimeye bir indeks atılarak kelime sözlüğü oluşturulmuştur. Metinlerin tokenize edilmesinde tensorflow.keras.preprocessing.text kütüphanesinde yer alan Tokenizer sınıfı kullanılmıştır. Tokenize işleminde max_words = 10 000 en fazla 10 000 kelime ve max_len = 100 ile her bir örneği en fazla 100 kelime ile sınırlandırılmıştır.



Şekil 3.6. Derin öğrenme modelleri ile karar tahmini uygulaması akış şeması

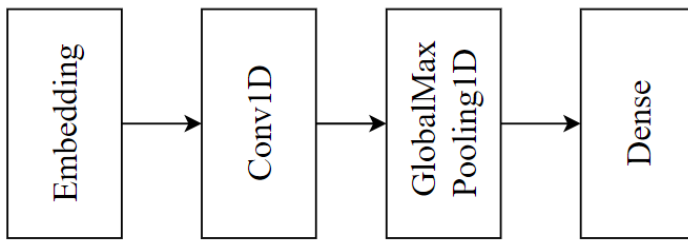
Deneysel çalışmalarda, derin öğrenmenin farklı modelleri ve katman yapıları kullanılarak karar tahmini yapılmıştır. Kullanılan katmanlar ve aktivasyon fonksiyonları ilgili modelin başlığı altında açıklanmıştır.

Modellerin karar tahminlerine ilişkin performans metrikleri hesaplanmıştır. Değerleri DENEYSEL SONUÇLAR başlığında altında sunulmuştur.

3.2.1. Evrişimsel sinir ağları

Evrişimsel sinir ağları (CNN), derin öğrenme alanında yaygın olarak kullanılan bir yapay sinir ağı modelidir. Geleneksel olarak görüntü işleme alanında başarılı olan CNN, hukuki metinlerin sınıflandırılması ve analizinde kullanılan en popüler ve yüksek performans sunan modellerden biri olduğu belirtilmiştir[53][54].

CNN modeli katmanlarını oluştururken keras.layers kütüphanesinde yer alan Embedding, Conv1D, GlobalMaxPooling1D ve Dense sınıfı kullanılmıştır. Modelin katmanlarını sırasıyla tanımlamak için keras.models kütüphanesinde yer alan Sequential sınıfı kullanılmıştır. Şekilde 3.7’de görüldüğü gibi Embedding + Conv1D + GlobalMax Pooling1D + Dense katmanlarından oluşan bir mimari modellenmiştir.



Şekil 3.7. Evrişimsel sinir ağları

Embedding Katmanı; kelime dağarcığını temsil eden gömme matrisi oluşturularak, kelimeleri belirli bir boyuttaki vektörlerle temsil edilmesini ve sinir ağı tarafından daha kolay işlenebilmesi sağlanmıştır. Eş. (3.7)’de gösterilen denklemde, E kelimelerin gömülü vektörlerinin matrisini, W_e kelimelerin gömülme matrisini ve x girdi metnini ifade etmektedir. Embedding katmanı ile kelimelerin gömülme matrisini (W_e) girdi metni (x) ile çarpılarak kelimeleri sayısal vektörlere dönüştürülmüştür. Kelimelerin gömülme matrisi, her

kelimeyi gerçek sayılardan oluşan bir vektörle temsil etmektedir. Bu vektörler ile, kelimelerin anlamsal ilişkileri ve özellikleri kodlanmıştır. Çıktı matrisi (E) vektörleri, sinir ağının sonraki katmanları tarafından işlenerek metin sınıflandırma işlemi için kullanılmıştır.

$$E = W_e * x \quad (3.7)$$

Bir boyutlu evrişim katmanı olan Conv1D katmanı kullanılarak; Metin verileri üzerinde evrişim işlemleri gerçekleştirilmiştir. Çalışmada 7 860 filtre kullanılmıştır ve 4 boyutlu bir pencereyi gezdirerek evrişim işlemi uygulanmıştır. Eş. (3.8)'de matematiksel denklem gösterilmiştir. y_i : i. çıktı değerini, f: Aktivasyon fonksiyonunu (örneğin ReLU, sigmoid), m: Filtre boyutunu (kernel size), w_{ij} : Filtre matrisinin (kernel) i. satır ve j. sütunundaki değeri, x_i : Girdi sinyallerinin i. değerini ve b_i : i. bias değerini temsil etmektedir.

$$y_i = f(\sum_{j=1}^m w_{ij} * x_{i+j-1} + b_i) \quad (3.8)$$

GlobalMaxPooling1D katmanı kullanılarak evrişim katmanından gelen çıktının boyutunu küçültülerek her özellik haritasından en yüksek değer alınıp tek boyutlu bir vektöre dönüştürülmüştür. Eş. (3.9)'da matematiksel denklem gösterilmiştir. y_i : i. çıktı değeri, x_{i1} : Girdi sinyallerinin i. özelliğinin 1. Değeri, x_{i2} : Girdi sinyallerinin i. özelliğinin 2. Değeri, ..., x_{in} : Girdi sinyallerinin i. özelliğinin n. değerini temsil etmektedir.

$$y_i = \max(x_{i1}, x_{i2}, \dots, x_{in}) \quad (3.9)$$

Evrişim katmanlarından gelen çıktıları işleyerek belirli bir boyuttaki gizli katmanlarda yer alan nöronları beslemek için Dense katmanı kullanılmıştır. Bu çalışmada 64 nöronlu bir dense katmanı tanımlanmıştır. Katmandaki her nöron, bir önceki katmandan gelen tüm nöronlarla bağlantılıdır. Katmanın aktivasyon fonksiyonu olarak ReLU kullanılmıştır. ReLU, doğrusal olmayan bir aktivasyon fonksiyonudur ve $x < 0$ için 0, $x \geq 0$ için x değerini almaktadır. Çıkış katmanındaki nöron sayısı, sınıf sayısına eşittir. Katmanın aktivasyon fonksiyonu olarak sigmoid kullanılmıştır. Sigmoid, 0 ile 1 arasında bir değer üreten bir aktivasyon fonksiyonudur. Bu değer, bir sınıfın olasılık değerini temsil etmektedir. Eş. (3.10)'da matematiksel denklem gösterilmiştir. y_i : i. çıktı değeri (sınıfın olasılık değeri), σ : aktivasyon fonksiyonu (sigmoid), n: girdi vektörünün boyutuna (bir önceki katmandan gelen nöron sayısına), w_{ij} : i. çıktı ve j. girdi arasındaki ağırlık, x_j : Girdi vektörünün j. Değeri ve

b_i : i . çıktı için bias(modelin öğrenme algoritması tarafından otomatik olarak belirlenen sabit değeri) değerlerini temsil etmektedir.

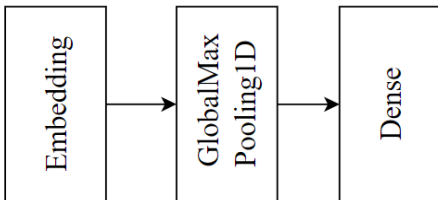
$$y_i = \sigma(\sum_{j=1}^n w_{ij} * x_j + b_i) \quad (3.10)$$

Modeli derlemek için Adaptive Moment Estimation (Adam) optimizasyon algoritmasını kullanılmıştır. Modeli eğitmek için eğitim verileri ve eğitim etiketleri kullanılmıştır. Modelin eğitim verilerinin üzerinden 10 kez geçmesini sağlanarak yeterince öğrenmemesi veya aşırı öğrenmesinin önüne geçilmiştir. Her adımda 32 örnek üzerinden güncelleme yapılarak Modelin daha yavaş öğrenmesinin önüne geçilmiştir.

3.2.2. İleri beslemeli sinir ağları

İleri beslemeli sinir ağları (FFNN), basit ve temel yapay sinir ağı mimarilerinden biridir. Hukuki metinlerin sınıflandırılmasında ve çeşitli akademik çalışmalarda kullanılmıştır[55]. Bu sinir ağları, girdi katmanından başlayıp, bir veya daha fazla gizli katmandan geçerek çıktı katmanına ulaşan ve bir geri besleme (feedback) döngüsü içermeyen ağlardır. Bu ağlar, özellikle metin sınıflandırma gibi görevlerde kullanılabilir.

FFNN modeli katmanlarını oluştururken keras.layers kütüphanesinde yer alan Embedding, GlobalMaxPooling1D ve Dense sınıfı kullanılmıştır. Modelin katmanlarını sırasıyla tanımlamak için keras.models kütüphanesinde yer alan Sequential sınıfı kullanılmıştır. Şekilde 3.8'de görüldüğü gibi Embedding + GlobalMax Pooling1D + Dense katmanlarından oluşan bir mimari modellenmiştir.



Şekil 3.8. İleri beslemeli sinir ağları

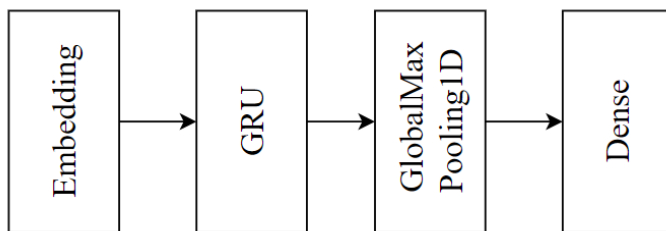
Embedding katmanı ile kelimeler yoğun vektörlere dönüştürülmüştür. GlobalMaxPooling1D katmanı, Embedding katmanının çıktısının en yüksek değerini alır ve değeri düzleştirir. Bu, her kelimenin en önemli özelliklerinin alınmasını sağlar ve giriş vektörünü sabit boyuta indirger.

Dense katmanları, 64 nöronlu ve ReLU aktivasyon fonksiyonuna sahip tam bağlantılı katman. ReLU, doğrusal olmayan aktivasyon fonksiyonudur ve modelin öğrenme kapasitesini artırmaktadır. Sınıf sayısı kadar nöronlu ve softmax aktivasyon fonksiyonuna sahip tam bağlantılı katman. Çıkış katmanı, çok sınıflı sınıflandırma için kullanılır. Softmax, her sınıfa bir olasılık değeri atamaktadır. Dense katmanları, düzleştirilmiş veriler üzerinde tam bağlantılı nöronlar aracılığıyla işlemler yapmaktadır.

3.2.3. Mükerrer birimli sinir ağı

Mükerrer birimli sinir ağı (GRU), yapay sinir ağlarının bir türüdür ve uzun vadeli bağımlılıkları modellemek için özel olarak tasarlanmıştır. Bu, özellikle zaman serileri ve DDİ gibi alanlarda oldukça faydalıdır. GRU'lar, geri bildirim döngüleri ve kapı mekanizmaları kullanarak önceki adımlardan gelen bilgileri hatırlayabilir ve işleyebilirler. Bu sayede, uzun metinlerde anlamı korumada ve uzun vadeli tahminlerde oldukça başarılıdır. GRU'lar ve diğer RNN türevlerinin hukuki kararların tahmin edilmesinde nasıl kullanıldığı incelenmiş ve GRU'lar, metinlerdeki uzun vadeli bağımlılıkları yakalayarak daha doğru tahminler yapabildiğini tespit etmişlerdir[56].

GRU modeli katmanlarını oluştururken keras.layers kütüphanesinde yer alan Embedding, GRU, Dense ve GlobalMaxPooling1D sınıfı kullanılmıştır. Modelin katmanlarını sırasıyla tanımlamak için keras.models kütüphanesinde yer alan Sequential sınıfı kullanılmıştır. Şekilde 3.9'de görüldüğü gibi Embedding + GRU+ GlobalMax Pooling1D + Dense katmanlarından oluşan bir mimari modellenmiştir.



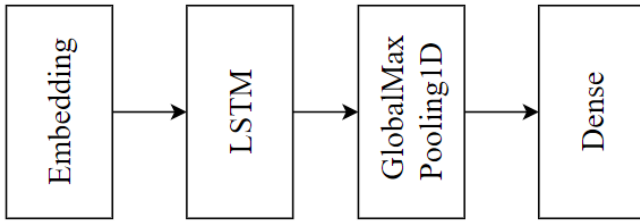
Şekil 3.9. Mükerrer birimli sinir ağı

GRU katmanı, embedding vektörlerini alır ve her zaman adımında gizli durum vektörleri üretir. Metin dizilerinde uzun vadeli bağımlılıkları yakalamak için kullanılır. GRU, kapı mekanizmaları kullanarak bilgiyi filtrelemektedir.

3.2.4. Uzun ve kısa vade hafızalı sinir ağı

Uzun ve kısa vade hafızalı sinir ağı (LSTM), RNN'nin bir türüdür ve özellikle uzun süreli bağımlılıkları öğrenme konusundaki zorlukları aşmak için geliştirilmiştir. LSTM'ler, metin verilerinde, özellikle sıralı veri yapılarında kullanılmaktadır. Hukuk metinlerinin analizi ve sınıflandırılmasında LSTM'lerin kullanımı oldukça yaygındır. Yapılan çalışmalarda [55][56] LSTM'lerin hukuki metinlerin analizi ve sınıflandırmasında, potansiyelini ve etkinliğini göstermektedir.

LSTM modeli katmanlarını oluştururken keras.layers kütüphanesinde yer alan Embedding, LSTM, Dense ve GlobalMaxPooling1D sınıfı kullanılmıştır. Modelin katmanlarını sırasıyla tanımlamak için keras.models kütüphanesinde yer alan Sequential sınıfı kullanılmıştır. Şekilde 3.10'da görüldüğü gibi Embedding + LSTM+ GlobalMax Pooling1D + Dense katmanlarından oluşan bir mimari modellenmiştir.



Şekil 3.10. Uzun ve kısa vade hafızalı sinir ağı

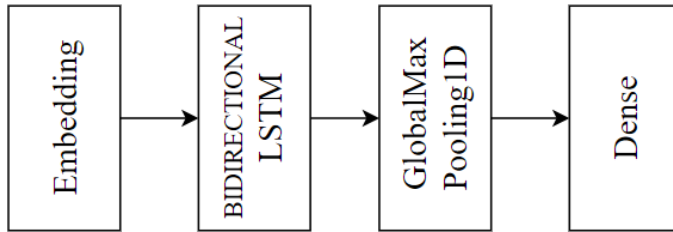
LSTM katmanı, Embedding vektörlerini alır ve her zaman adımında gizli durum vektörleri üretir. Bu katman, metin dizisindeki uzun vadeli bağımlılıkları yakalamak için kullanılır. LSTM, kapı mekanizmaları kullanarak bilgiyi filtrelemektedir.

3.2.5. Çift Yönlü LSTM

Çift yönlü LSTM (BiLSTM), LSTM modelinin bir türevidir. LSTM'ler gibi, BiLSTM'ler de uzun vadeli bağımlılıkları modellemek için özel olarak tasarlanmıştır. Fakat BiLSTM'ler,

metni iki yönden işleme özelliğine sahiptir. İki yönlü bilgi akışını kullanarak sekans verilerinin daha zengin ve kapsamlı bir temsiline olanak tanır. Bu sayede, metnin hem öncesindeki hem de sonrasındaki bağlamlar dikkate alınarak daha kapsamlı bir analiz yapılmasını sağlar. BiLSTM, metin verilerinde sıralı bağımlılıkları yakalamak için hem geçmiş hem de gelecek bilgilere bakmaktadır.

BiLSTM modeli katmanlarını oluştururken keras.layers kütüphanesinde yer alan Embedding, LSTM, Dense, GlobalMaxPooling1D ve Bidirectional sınıfı kullanılmıştır. Modelin katmanlarını sırasıyla tanımlamak için keras.models kütüphanesinde yer alan Sequential sınıfı kullanılmıştır. Şekilde 3.11'de görüldüğü gibi Embedding + BIDIRECTIONAL LSTM+ GlobalMax Pooling1D + Dense katmanlarından oluşan bir mimari modellenmiştir.



Şekil 3.11. Çift yönlü LSTM

Bidirectional LSTM Katmanı, embedding vektörlerini alır ve her zaman adımında çift yönlü gizli durum vektörleri üretir. Bu katman iki LSTM katmanını içerir: biri ileri yönde, diğeri geri yönde çalışır. Metin dizisindeki uzun vadeli bağımlılıkları her iki yönde yakalamaktadır.

3.3. Sınıflandırma Ölçütleri

Modellerin değerlendirilmesi için veri setinden ilk başta ayrılan test verileri kullanılarak kesinlik performans değerleri hesaplanmıştır.

3.3.1. Kesinlik

Modelin pozitif olarak tahmin ettiği örneklerin ne kadarının gerçekten pozitif olduğunu ölçerek kesinlik (precision) değeri hesaplanmıştır. Eş (3.11)'de matematiksel metriği

gösterilmiştir. TP (True Positive): Doğru pozitif tahminlerin sayısı. Modelin doğru bir şekilde pozitif olarak tahmin ettiği örneklerin sayısı. FP (False Positive): Yanlış pozitif tahminlerin sayısı. Modelin pozitif olarak tahmin ettiği ancak gerçekte negatif olan örneklerin sayısı ifade etmektedir.

$$\text{Kesinlik} = TP / (TP+FP) \quad (3.11)$$

3.3.2. Duyarlılık

Gerçek pozitiflerin, tüm pozitiflerin oranı hesaplanmıştır. Yani, gerçekte pozitif olan tüm örnekler arasından modelin doğru bir şekilde kaçını tahmin ettiğini ifade duyarlılık (recall) değeri hesaplanmıştır. Eş (3.12)'de matematiksel metriği gösterilmiştir. TP: Doğru pozitif tahminlerin sayısı. Modelin doğru bir şekilde pozitif olarak tahmin ettiği örneklerin sayısı. FN (False Negative): Yanlış negatif tahminlerin sayısı. Modelin negatif olarak tahmin ettiği ancak gerçekte pozitif olan örneklerin sayısını ifade etmektedir.

$$\text{Duyarlılık} = TP / (TP+FN) \quad (3.12)$$

3.3.3. F1-Skor

F1-skoru, kesinlik ve duyarlılık metriklerinin harmonik ortalaması olarak bulunabilmektedir. F1-skoru, bu iki metriğin dengesini sağlamaktadır. . Eş (3.13)'te matematiksel metriği gösterilmiştir. Kesinlik değeri, modelin ne kadar kesin olduğunu, duyarlılık değeri ise modelin ne kadar kapsamlı olduğunu ifade etmektedir.

$$\text{F1 Skoru} = 2 \times (\text{kesinlik} \times \text{duyarlılık}) / (\text{kesinlik} + \text{duyarlılık}) \quad (3.13)$$

3.3.4. Makro Ortalama

Makro ortalama (Macro average), sınıf bazında hesaplanan metrik değerlerinin her bir sınıfın örnek sayısına bağlı olmadan doğrudan sınıf başına hesaplanması yapılmıştır. Makro ortalama, her sınıfın metrik değerini hesaplar ve ardından bu değerlerin ortalamasını alarak sınıf başına ağırlıklandırma yapılmadan her sınıfın katkısının eşit olarak değerlendirir. Eş (3.14)'te matematiksel metriği gösterilmiştir. N, toplam sınıf sayısını, *Metrick* her sınıf için

hesaplanan metrik değerlerini ifade etmektedir.

$$\text{Macro Avg} = 1/N \sum_{k=1}^N \text{Metric}_k \quad (3.14)$$

3.3.5. Ağırlıklı Ortalama

Ağırlıklı ortalama (Weighted average), sınıf bazında hesaplanan metrik değerlerinin her bir sınıfın örnek sayısına göre ağırlıklandırılması ve ardından bu ağırlıklandırılmış metriklerin toplanmasıyla elde edilen bir ortalama değeri hesaplanmıştır. Ağırlıklı ortalama, her bir sınıfın örnek sayısının metrik sonuçlarına olan etkisini göz önünde bulundurarak bir metriğin genel performansını hesaplamaktadır. Eş (3.15)'te matematiksel metriği gösterilmiştir. N, toplam sınıf sayısını, Support_k her bir sınıfın örnek sayısını ve Metric_k her sınıf için hesaplanan metrik değerlerini ifade etmektedir.

$$\text{Weig.Avg} = \sum_{k=1}^N (\text{Support}_k + \text{Metric}_k) / \sum_{k=1}^N \text{Support}_k \quad (3.15)$$

3.3.6. Doğruluk

Doğruluk (Accuracy) sınıflandırma yüzdesi hesaplanmıştır. Eş (3.16)'da matematiksel metriği gösterilmiştir. Doğruluk metriği, doğru tahmin edilen örneklerin sayısını, toplam örnek sayısına bölerek hesaplanmaktadır. Bu değer, modelinizin tahminleri ile gerçek değerler arasındaki ortalama hatayı göstermektedir.

$$\text{Doğruluk} = (\text{Doğru Tahminler} / \text{Toplam Örnekler}) \times 100 \quad (3.16)$$



4. TASLAK KARAR METNİ

Hukuki metinleri kullanarak metin oluşturma, son yıllarda ilgi çeken bir araştırma alanıdır. Bu alandaki çalışmalar, yapay zeka modellerini kullanarak var olan veriler ile yasal belgelerin analiz edilmesi, taslak karar metni oluşturulması potansiyelini incelemişlerdir[57][58].

Bu çalışmada yapay zeka alanında kullanılan dil modellerinden biri olan GPT-2'yi kullanarak taslak karar metni oluşturulmuştur. İşlem adımları Şekil 5.1'de gösterilmiştir. GPT-2 ile fine-tuning işlemi yapılmıştır. Fine-tuning, transformer tabanlı öğrenmede önceden eğitilmiş(pre-train) bir modelin, belirli bir görev veya veri kümesi üzerinde daha iyi performans göstermesi için optimize etme işlemi içermektedir. Bu işlem, modelin önceden öğrendiği genel bilgi ve becerileri kullanarak, yeni bir alana/göreve daha hızlı ve kolay bir şekilde uyum sağlamasına olanak tanımaktadır. Fine-tuning işlemi için Ombudsmanlık kararlarından oluşan veri seti kullanılmıştır. Bu çalışmada geliştirilen kodlar çalışmada geliştirilen tüm kodlar Python dilinde kodlanmıştır. Tüm kodlar Google Colab pro platformu üzerinde koşturulmuştur.

Fine-tuning işlemi ile eğitilen dil modeli ile taslak karar metni üretimi (text generation) yapılmıştır. Yeterli RAM ve işlemci kapasitesine sahip olunmadığı için; Resim 4.1'de gösterilen Ombudsmanlık karar metni içinde bulunan "KARAR" kısmındaki metinler kullanılmıştır. Model şikayet konu, şikayet alt konu, karar tipi ve karar alanı metni verileri ile eğitilmiştir. Eğitilen dil modeli taslak karar metni oluşturulmuştur. Metin üretimi için gerçekleştirilen işlem adımları ilgili kısımlarda anlatılmıştır.

VII. KARAR

Açıklanan gerekçelerle BAŞVURUNUN KABULÜNE;

Başvuranın hasta nakil ambulansında görevlendirilmesi işleminin ivedilikle geri alınması, söz konusu görevlendirmenin, Ambulanslar Ve Acil Sağlık Araçları İle Ambulans Hizmetleri Yönetmeliğinin 4 üncü ve 7 nci maddesi hükümlerine uygun olarak yeniden yapılması gerektiği hususunda MERSİN İL SAĞLIK MÜDÜRLÜĞÜ MUT DEVLET HASTANESİ BAŞHEKİMLİĞİNE TAVSİYEDE BULUNULMASINA,

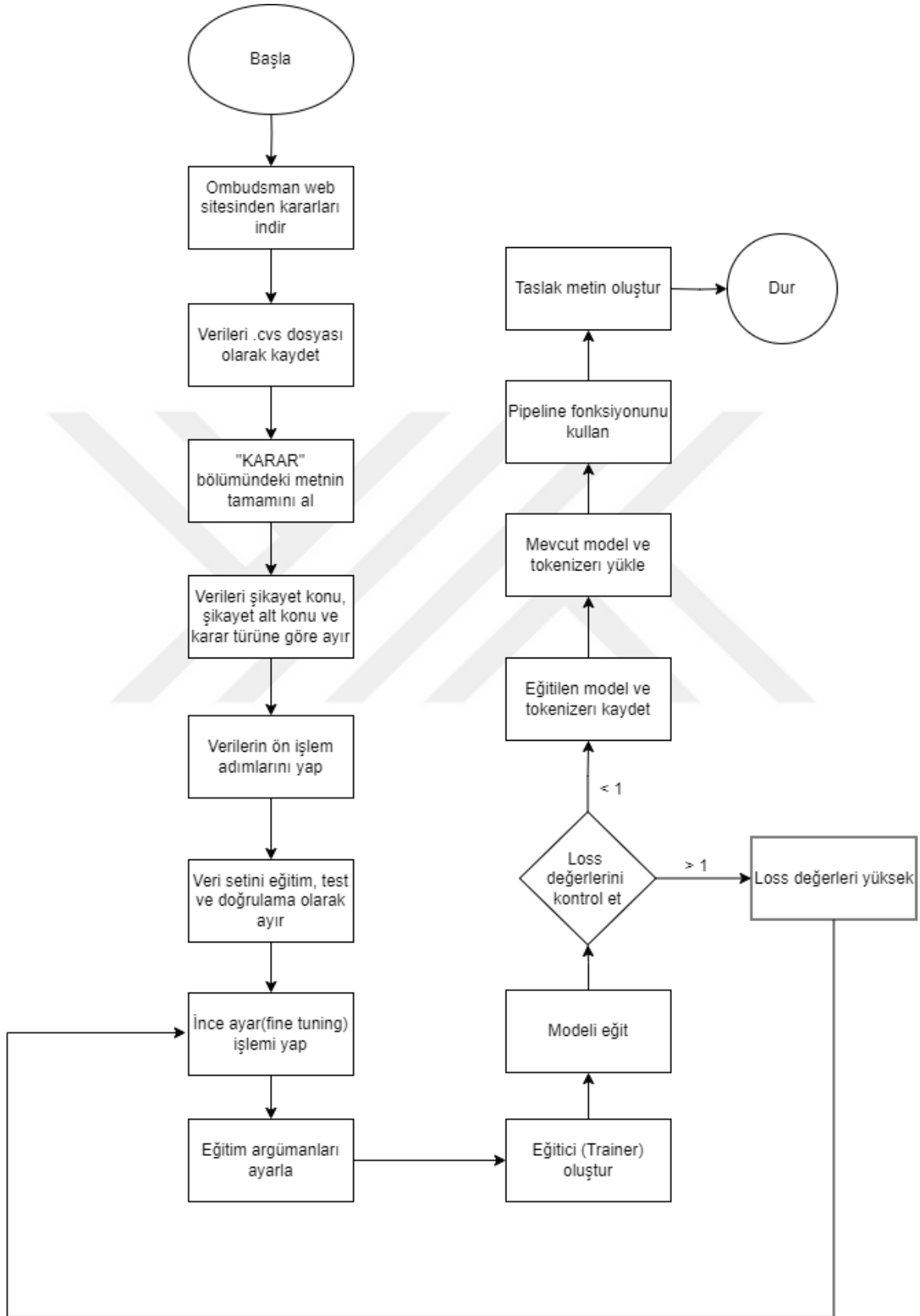
6328 sayılı Kamu Denetçiliği Kurumu Kanununun 20 nci maddesinin üçüncü fıkrası uyarınca; MERSİN İL SAĞLIK MÜDÜRLÜĞÜ MUT DEVLET HASTANESİ BAŞHEKİMLİĞİ tarafından bu karar üzerine tesis edilecek işlemin otuz gün içinde Kurumumuza bildirilmesinin zorunlu olduğuna,

Kararın BAŞVURANA ve MERSİN İL SAĞLIK MÜDÜRLÜĞÜ MUT DEVLET HASTANESİ BAŞHEKİMLİĞİNE tebliğine,

Türkiye Cumhuriyeti Kamu Başdenetçisince karar verildi.

Şeref MALKOÇ
Kamu Başdenetçisi

Resim 4.1. Örnek karar metni karar alanı



Şekil 4.1. Taslak karar metni oluşturma uygulaması akış şeması

4.1. İnce Ayar

İnce ayar (Fine-tuning), önceden eğitilmiş bir modelin belirli bir görev için optimize edilmesidir. Transformer tabanlı modellerde yaygın olarak kullanılan bu yöntem, geniş veri kümeleri üzerinde önceden eğitilmiş modellerin daha küçük ve özel veri kümeleri üzerinde yeniden eğitilmesiyle gerçekleştirilir. Bu süreç, modelin genel bilgi birikimini korurken, belirli bir alan veya görev için gerekli olan özel bilgileri öğrenmesini sağlamaktadır.

Hukuk alanında, transformer tabanlı öğrenme ve fine-tuning yöntemlerinin kullanımı, büyük metin veri kümelerinden anlamlı bilgiler çıkarmak, yargı kararlarını tahmin etmek ve taslak karar metni oluşturmak gibi çeşitli uygulamalar için oldukça faydalıdır.

Transformer tabanlı öğrenme, DDİ alanında oldukça popüler hale gelen bir derin öğrenme yaklaşımıdır. Transformer mimarisi, 2017 yılında Google Research tarafından geliştirilmiştir [59]. Bu mimari, RNN mimarisinden farklı olarak, metinleri işlerken dikkat mekanizması adı verilen yeni bir teknik kullanmaktadır. Dikkat mekanizması, metindeki en önemli kelimelere modelin odaklanmasını sağlamaktadır. Bunun sonucunda model, metnin anlamını daha iyi anlayarak daha doğru sonuçlar üretebilmektedir.

Bu çalışmada Hugging Face'de yer alan GPT-2 dil modeli mimarisini temel alarak geliştirilmiş olan ytu-ce-cosmos/turkish-gpt2-large modeli kullanılmıştır. Yıldız Teknik Üniversitesi Elektrik Elektronik Fakültesi Bilgisayar Mühendisliği Bölümü'nden bir araştırma ekibi tarafından geliştirilen Türkçe bir dil modelidir. Bu model geniş bir metin korpusu üzerinde türkçe dil yapısına göre eğitilmiş olup 750 milyon parametre içermektedir[61]. Veri seti ytu-ce-cosmos/turkish-gpt2-large tokenizer'ı ile tokenize edilmiştir. Geçersiz veriler (none, null vb.) eğitimin dışında tutulmuştur.

Eğitim argümanları ayarlanarak, modelin nasıl eğitileceğini ve değerlendirilmesi gerektiğini belirlenmiştir. Her bir parametre, eğitim sürecinin verimliliğini, hızını ve doğruluğunu etkileyen önemli hiperparametrelerden oluşmaktadır. Eğitim sürecini ve performansı optimize parametreler yapılan deneme ve test sonucunda tespit edilmiştir.

Örneğin, öğrenme oranı $2e-5$ belirlenerek modelin daha yavaş fakat daha hassas bir şekilde öğrenmesi sağlanmıştır. output_dir parametresi ile model ve eğitim çıktılarının (örneğin,

kontrol noktası (checkpoint) kaydedileceği dizin belirlenmiştir. Eğitim süreci boyunca ve sonunda oluşturulan model dosyaları bu dizine kaydedilmiş olup eğitim süreci uzun olduğu için RAM çökmesi, bellek yetersizliği vb. durumlarda eğitim tamamlanmadığında kontrol noktası ile işleme devam edilerek zaman kaybı yaşanmasının önüne geçilmiştir.

Eğitim (Trainer) nesnesini oluşturarak model eğitim sürecini başlatmaya hazır hale getirilmiştir. Eğitim nesnesi, modelin eğitim ve değerlendirme süreçlerini yönetmektedir. Bu sayede daha az kod yazarak karmaşık eğitim işlemlerinin gerçekleştirilmesi sağlanmaktadır. Eğitim nesnesi, Hugging Face Transformers kütüphanesinde model eğitimini basitleştirmek ve hızlandırmak için kullanılan yüksek seviyeli bir API'dir. Eğitim döngüsünü, doğrulama sürecini, checkpoint almayı, erken durdurmayı ve daha fazlasını yönetmektedir.

Eğitim nesnesi, oluşturulduktan sonra model eğitilmiştir. Model karar türü, şikayet konu, şikayet alt konu ve KARAR kısmında yer alan metinlerle eğitilmiştir. Zaman kaybının önüne geçmek ve eksik kalan eğitimi tamamlamak için; eğer bir kontrol noktası varsa eğitime o noktadan itibaren devam edilmiştir.

Eğitim tamamlandıktan sonra loss (kayıp) değerleri kontrol edilmiştir. Kontrol sırasında eğitim ve doğrulama kayıplarının 1 'in altında olup olmadığı izlenmiştir. Çalışmada eğitim kaybı: 0.26 ve doğrulama kaybı: 0.23 olduğu görülmüştür. Değerlere modelimizin öğrendiği tespit edilerek model ve tokenizer kaydedilmiştir.

4.2. Metin Üretimi

Metin üretimi (Text generation), metin oluşturmak için yapay zekanın kullanımını içermektedir. Metin üretimi, bir derin öğrenme modeli kullanarak, belirli bir konu veya başlık altında insan benzeri metinler oluşturma sürecinin tamamını kapsamaktadır. Bu metinler, haber makaleleri, blog yazıları, kurgusal hikayeler ve hatta hukuk metinleri gibi çeşitli formatlarda olabilir. Metin üretimi, derin öğrenme ve makine öğrenmesi gibi alanlarda meydana gelen gelişmelerle mümkün hale gelmiştir. Son yıllarda en yaygın kullanılan yöntemler arasında Transformer mimarisi ve özellikle GPT (Generative Pre-trained Transformer) modelleri bulunmaktadır.

Hukuk alanında metin üretimi, hem pratik hem de teorik açıdan büyük bir potansiyele sahiptir [60]. Örneğin, karar tahmini, hukuki belgelerin özetlenmesi, dava dosyalarının analiz edilmesi ve taslak karar metinlerinin oluşturulmasını kapsamaktadır.

Çalışmanın bu kısmında fine-tuning işlemi ile eğitilerek kaydedilen model ve tokenizer kullanılmıştır. Taslak karar metni üretimi için modele öğretilen KARAR kısmı metinleri kullanılmıştır. Veri seti oluşturulan tokenizer ile tokenize edilmiştir.

Transformers kütüphanesi ile, Hugging Face'de bulunan önceden eğitilmiş dil modeli cosmos/turkish-gpt2-large erişim sağlanmıştır. Pipeline ile cosmos/turkish-gpt2-large dil modeline API ile erişim sağlanmıştır. Bu API sayesinde çok az kodla kompleks görevlerin gerçekleştirilmesi kolaylaştırılmıştır.

Taslak metin üretimi için fonksiyon geliştirilmiştir. Bu fonksiyon, belirli filtre kriterlerine (şikayet konu, şikayet alt konu ve karar türü) uyan veri çerçevesi satırlarını seçer, bu satırlardan metinleri birleştirerek bir giriş metni oluşturur ve ardından bu metni kullanarak yeni metin üretmektedir. Üretilen metin, belirtilen kriterleri karşılıyorsa döndürülür; aksi takdirde, uygun bir hata mesajı döndürülmektedir.

Geliştirilen uygulama kullanıcı tarafından girilen şikayet konu, şikayet alt konu ve karar türüne göre taslak metin üretimi yapılmaktadır.



5. DENEYSEL SONUÇLAR

Makine öğrenmesi ve derin öğrenme modelleri ile karar tahmini çalışması Google Colab Pro platformu üzerinde “Sklearn”, “xgboost” ve “Tensorflow.Keras” kütüphaneleri kullanılarak geliştirilmiştir. Taslak karar metni oluşturma çalışması Hugging Face’de yer alan GPT-2 dil modeli mimarisini temel alarak geliştirilmiş olan ytu-ce-cosmos/turkish-gpt2-large modeli kullanılmıştır. Bu tezde veri kümesi eğitim, test ve doğrulama olarak bölünmüştür. Bölümle işleminin oranları kullanılan modeller ve işlemlere göre farklılık göstermektedir.

Deneysel çalışmalar sonucunda, karşılaştırması yapılan çalışmalarda ret kararı veri kümesinde modellerin daha yüksek başarı gösterdiği görülmüştür. İçerik olarak karar türlerinin bir birinden farklı olması ve kararlarında kendi içlerinde şikayet konu ve şikayet alt konusuna göre birbirinden farklı olarak metinler içermesi sonucu etkilediği gözlemlenmiştir. Ret kararında yer alan örnek sayısının diğerlerinden daha fazla olması bu sonucu ortaya çıkardığı düşünülmektedir. Veri setinde yer alan örnek sayısı arttıkça başarı ihtimali yükselmektedir.

5.1. Karar Tahmini Uygulaması

Ombudsmanlık kararlarından oluşan veri setinde, daha önceden verilmiş kararlar yer almaktadır. Kararlar eğitim, test ve doğrulama olarak üç kısma ayrılmıştır. Eğitim sırasında modellere gösterilmeyen kararlar ile karar tahmini yapılmış bu kapsamda performans metrikleri hesaplanmıştır.

Ombudsmanlık kararlarını en başarılı şekilde tahmin eden makine öğrenmesi modeli SVM, derin öğrenme modeli ise GRU olduğu gözlemlenmiştir. Karar tahmininde SVM modeli ile %61, GRU ve LSTM modeli ile %64 F1 skoru değerine ulaşılmıştır.

5.1.1 Makine öğrenmesi modellerinin uygulaması

Makine öğrenmesi modelleri DT, SVM, Naïve Bayes, KKN, LR ve XGBoost denetimli şekilde eğitilerek karar tahmini yapılmıştır. Ombudsmanlık kararlarından oluşan test verisi ile elde edilen karar tahmini sonuçları Çizelge 5.1’de verilmiştir. SVM modeli en yüksek

doğruluk ve F1-skoru'na sahip iken DT modeli en düşük değere sahiptir. SVM modelinin doğru tahmin oranının diğer modellerden daha iyi olduğunu göstermektedir. Makine öğrenmesi modellerinin F1-skoru Şekil 5.1'de gösterilmiştir.

Doğruluk değerine göre DT modeli haricinde diğer modellerin birbirlerine yakın performans gösterdiği tespit edilmiştir.

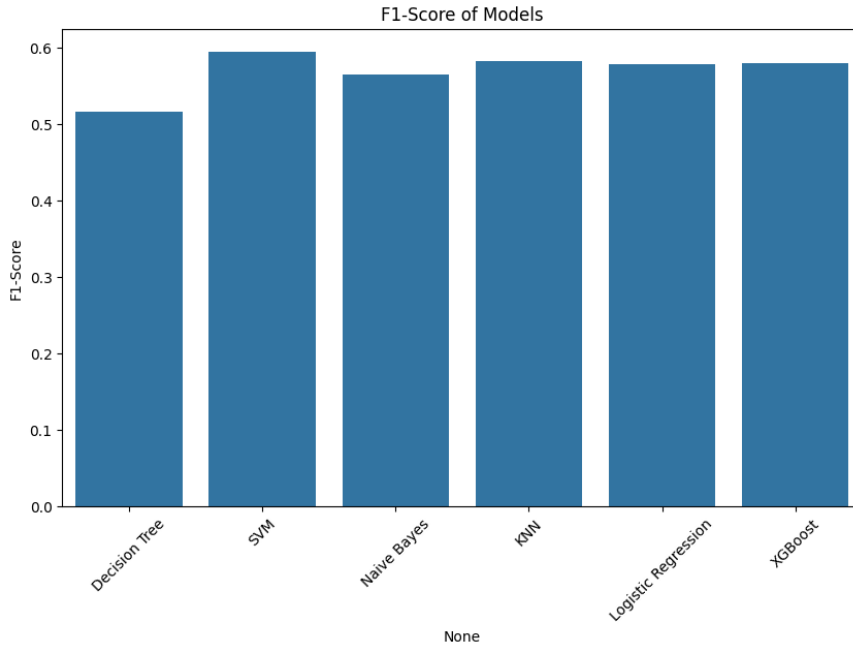
Tüm modeller için DT modeli hariç kesinlik değerleri %57 ile %63 arasında değiştiği gözlemlenmiştir. Bu, modellerin pozitif tahminlerde nispeten daha başarılı olduklarını göstermektedir.

Tüm modeller için duyarlılık değerleri de %58 ile %61 arasında değiştiği gözlemlenmiştir. Bu, modellerin gerçekte pozitif olan örneklerin bir kısmını kaçırma ihtimali olduğunu göstermektedir.

Çizelge 5.1. Makine öğrenmesi modelleri karar tahmini sonuçları

	Doğruluk	Kesinlik	Duyarlılık	F1-Skor
DT	0,49	0,49	0,49	0,49
SVM	0,62	0,63	0,62	0,61
Naïve Bayes	0,59	0,59	0,59	0,57
KKN	0,58	0,57	0,58	0,57
LR	0,60	0,61	0,60	0,59
XCBoost	0,61	0,61	0,61	0,60

Tüm modeller, DT modeli hariç F1-Skor değerleri de %57 ile %61 arasında değiştiği gözlemlenmiştir. Bu da doğruluk, kesinlik ve duyarlılık'tan elde ettiğimiz değerleri desteklemektedir.



Şekil 5.1. Makine öğrenmesi modellerinin F1-skoru

5.1.2 Derin öğrenme modellerinin uygulaması

Derin öğrenme modelleri CNN, FFNN, GRU, LSTM ve BiLSTM ile eğitilerek karar tahmini yapılmıştır. Ret kararlarından oluşan test verisi ile elde edilen karar tahmini sonuçları Çizelge 5.2’de verilmiştir.

CNN modeli, kesinlik ve duyarlılık değerleri arasında iyi bir denge göstermektedir. Bunun sonucunda F1 skoruna olumlu şekilde etkilemektedir. CNN, doğru pozitifleri tespit etmede oldukça etkili olduğu tespit edilmiştir.

FFNN modeli, CNN’e benzer şekilde biraz daha düşük değerlere sahip. Kesinlik ve duyarlılık değerleri yakın, bu da F1 skorunu biraz daha düşük bir değere çekiyor. Stabil performans gösteriyor fakat CNN’den biraz daha az etkili olduğu gözlemlenmektedir.

GRU modeli, en yüksek duyarlılık değerine sahip olan model, bu da çoğu doğru pozitif yakaladığını göstermektedir. Kesinlik değeri duyarlılık değerine göre biraz daha düşük, bu da biraz daha fazla yanlış pozitif olduğunu göstermektedir. Ancak yüksek duyarlılık değeri, F1 skoru değerini daha yüksek yapıyor, bu da modelin genel performansın iyi olduğunu göstermektedir.

LSTM modeli, GRU'ya benzer, ancak duyarlılığı biraz daha düşük ve kesinliği biraz daha yüksek performansa sahiptir. F1 skoru GRU ile aynı, bu da LSTM'nin de oldukça etkili bir model olduğunu göstermektedir.

BiLSTM modeli, Dengeli bir yaklaşım göstermektedir, ancak diğer modellerle karşılaştırıldığında biraz daha düşük değerlere sahip olduğu gözlemlenmektedir. Kesinlik ve duyarlılık ile F1 skoru LSTM ve CNN'den daha düşüktür. Bu değerler BiLSTM'nin daha az etkili olduğunu göstermektedir, fakat yine de makul bir performans sunmaktadır.

Çizelge 5.2. Derin öğrenme modelleri ret karar tahmini sonuçları

	Kesinlik	Duyarlılık	F1-Skor
CNN	0,62	0,63	0,63
FFNN	0,62	0,63	0,62
GRU	0,60	0,69	0,64
LSTM	0,61	0,68	0,64
BiLSTM	0,60	0,64	0,62

CNN modeli kullanılarak test verisi ile elde edilen sonuçlar Çizelge 5.3'te verilmiştir. Kısmen Tavsiye Kısmen Ret Kararı için kesinlik ve duyarlılık performansı nispeten düşüktür. F1 skoru da ortalamanın altındadır. Modelin bu karar türünü doğru şekilde tahmin etmede zorlandığını göstermektedir.

Tavsiye Kararı için kesinlik ve duyarlılık performansı nispeten yüksektir. F1 skoru da ortalamanın üzerindedir. Bu, modelin bu sınıfı doğru şekilde tahmin etmede iyi olduğunu göstermektedir.

Ret Kararı için hassasiyet ve geri çağırma nispeten yüksektir. F1 skoru da ortalamanın üzerindedir. Bu, modelin bu sınıfı doğru şekilde tahmin etmede daha başarılı olduğunu göstermektedir. Kısmi Tavsiye Kararının yeterli veri olmadığı için modelin doğru şekilde tahmin etme yeteneği sınırlı kalmıştır. Örnek sayısı arttıkça modelin karar tahmininde daha başarılı performanslar elde ettiği tespit edilmiştir.

Çizelge 5.3. CNN modeli karar tahmini sonuçları

	Kesinlik	Duyarlılık	F1-Skore	Örnek Sayısı
Kısmen tavsiye kısmen ret kararı	0,64	0,47	0,54	135
Tavsiye kararı	0,58	0,64	0,61	357
Ret kararı	0,62	0,63	0,63	450
Kısmi tavsiye kararı	0,50	0,22	0,31	25

Derin öğrenme modelleri içinde daha yüksek performans gösteren modellerden biri olan GRU modelinin genel performans değerleri Çizelge 5.4'te gösterilmiştir. Bu değerlere göre modelin doğruluğu %59, yani model tüm veri setindeki örneklerin %59'unu doğru olarak sınıflandırmaktadır.

Makro ortalama, kesinlik doğru pozitif tahminlerinin oranı. %55.9, modelin sınıflandırdığı pozitif örneklerin ne kadarının gerçekten pozitif olduğunu göstermektedir. Duyarlılık %47.4, bu değer düşük olması, modelin gerçek pozitif örneği kaçırdığını (yanlış negatif) göstermektedir. F1 skoru %49.8, bu değer düşük olması, modelin genel performansının nispeten düşük olduğunu göstermektedir.

Ağırlıklı ortalama, kesinlik değerine göre modelin kesinlik oranı %58.7 olarak tespit edilmiştir. Modelin genel olarak daha yüksek kesinlik ile çalıştığını göstermektedir. Modelin duyarlılık oranı %59 olup, modelin genel olarak gerçek pozitifleri yakalama oranını göstermektedir. F1 skoru, modelin performansının ağırlıklı olarak %58.5 dengelendiğini göstermektedir.

Modelin performans değerlerini artırmak için daha fazla veri ile eğitilmesi gerektiğini göstermektedir.

Çizelge 5.4. GRU modeli performans değerleri

	Kesinlik	Duyarlılık	F1-Skor
Doğruluk	0,59	0,59	0,59

Makro Ortalama	0,55	0,47	0,49
Ağırlıklı Ortalama	0,58	0,59	0,58

5.2. Taslak Karar Metini Oluşturma Uygulaması

Metin üretiminde, İngilizce dil modellerine göre Türkçe kısıtlı sayıda dil modeli bulunmaktadır. Bu kısıtlara rağmen Hugging Face’de yer alan GPT-2 dil modeli mimarisini temel alarak geliştirilmiş olan ytu-ce-cosmos/turkish-gpt2-large modeli ile fine-tuning işlemi yapılmıştır. Çalışma Google Colab Pro’da 51 GB RAM kapasitesi kısıtı ile gerçekleştirildiği için daha büyük dil modelleri üzerinde çalışma yapılamamıştır. O nedenle fine-tuning işlemi ytu-ce-cosmos/turkish-gpt2-large modeli ile yapılmıştır.

Eğitim süreci yaklaşık olarak 90 saat sürmüştür. Eğitim sürecinin uzun olmasından dolayı sistem çökmeleri, kaynakların cevap vermemesi sorunları ile karşılaşmıştır. Her hata veya sorunda eğitim sürecini yeniden başlatmamak için model çalışmayı durdurduğunda işleme kontrol noktaları ile devam edilmiştir.

ytu-ce-cosmos/turkish-gpt2-large modeli ile eğitilen veri setinin kayıp değerleri Çizelge 5.5’te gösterilmiştir. Eğitim sırasında model, tüm veri kümesinden üç defa geçerek (3 epoch) veri kümesi üzerinde üç defa eğitilmiştir. Kayıp değerleri 1’in üzerinde yada altında olmasına göre değerlendirilmektedir. 1’in altındaki değerler modelin iyi bir şekilde öğrendiğini gösterirken çok düşük değerler modelin aşırı öğrenme (overfitting) yaptığını göstermektedir.

Eğitim kaybı, modelin eğitim verisi üzerinde ne kadar hata yaptığını göstermektedir. 0.26 değeri, modelin eğitim verisi üzerinde oldukça düşük bir hata oranına sahip olduğunu göstermektedir. Modelin eğitim verilerini iyi öğrendiğini göstermektedir.

Doğrulama kaybı, modelin eğitim sırasında görmediği doğrulama verisi üzerinde ne kadar hata yaptığını gösterir. Bu, modelin genel performansını göstermektedir. 0.23 değeri, modelin doğrulama verisi üzerinde de düşük bir hata oranına sahip olduğunu göstermektedir. Bu, modelin iyi genelleme yapabildiğini ve aşırı öğrenme (overfitting) yapmadığını göstermektedir.

Kayıp değerlerine göre, modelin eğitim süreci oldukça başarılı görünmektedir. Eğitim kaybı ve doğrulama kaybı değerleri düşük ve birbirine yakın olarak tespit edilmiştir. Bu durumda modelin hem eğitim verisinde hem de doğrulama verisinde iyi performans gösterdiğini ve genelleme yeteneğinin iyi olduğunu göstermektedir. Sonuç olarak bu değerler eğitim sürecinin doğru bir şekilde yapıldığını ve modelin kullanılabilir bir durumda olduğunu göstermektedir.

Çizelge 5.5. Fine-tuning kayıp değerleri

Eğitim Kaybı	Doğrulama Kaybı
0,26	0,23

Fine-tuning işlemi yapılarak eğitilmiş olan model ve tokenizer taslak karar metni üretmek için uygulamaya yüklenmiştir. Geliştirilen fonksiyon ile kullanıcı tarafından girilen şikayet konu, şikayet alt konu ve karar türüne uyan satırlar seçilerek metinler birleştirilmiştir. Birleştirilen metin kullanarak yeni bir taslak 'KARAR' alanı metni üretilmiştir.



6. SONUÇ VE ÖNERİLER

Bu tezde, karar destek sistemi geliştirilmiştir. Ombudsmalık web sayfasında kişisel verilerden arındırılmış kamu erişimine açık kararlardan veri seti oluşturulmuştur. Oluşturulan veri seti ile modeller eğitilerek uygulama oluşturulmuştur. Uygulama iki kısımdan oluşmaktadır. İlk kısımda sınıflandırma işlemi yapılarak karar tahmini yapılmıştır. Karar tahmininde altı farklı makine öğrenme tekniği ve beş farklı derin öğrenme modeli kullanılmıştır. Oluşturulan veri setine makine öğrenmesi algoritmalarından DT, SVM, Naïve Bayes, KKN, LR ve XCBoost modelleri uygulanmıştır. Derin öğrenme algoritmalarından CNN, FFNN, GRU, LSTM ve BiLSTM modelleri uygulanmıştır.

Uygulamanın ikinci kısmında taslak karar metni oluşturulmuştur. Veri seti Ombudsmanlık web sayfasında yayınlanan kararlardan oluşturulmuştur. Taslak karar metni oluşturmak için Hugging Face kütüphanesinde yer alan ytu-ce-cosmos/turkish-gpt2-large dil modeli ile fine-tuning işlemi yapılmıştır. Fine-tuning sırasında model, metin üretimi (text generation) için eğitilerek metin üretme yetenekleri de optimize edilmiştir. Bu sayede model, eğitildiği karar metinlerinin tarzında metinler üretebilir hale getirilmiştir. Fine-tuning işlemi sonucunda oluşturulan model ve tokenizer kaydedilmiştir. Oluşturulan model ve tokenizer ile şikayet konu, şikayet alt konu ve karar tipine göre taslak karar metni üretimi yapılmıştır.

Veri seti, ön işleme tabi tutulmuştur. Karar tahmini kısmında, “BAŞVURANIN İDDİA VE TALEPLERİ” alanında yer alan metinlere göre karar tahmini yapılmıştır. Bu kısımda yer alan metinlerin içerisinde durdurma kelimelerinin filtrelenmesi metnin anlamlarını etkilediği için karar tahmininde veri ön işleme adımı olan durdurma kelimelerinin filtrelenmesi işlemi yapılmamıştır. Taslak karar metni oluşturma işleminde “KARAR” alanında yer alan metinlere göre metin oluşturulmuştur. Modelin, ağırlığı yüksek kelimelere yönelmesi için durdurma kelimeleri veri setinden filtrelenmiştir.

Ön işlemler neticesinde veri setinde yer alan veriler, karar tahmini ve taslak karar oluşturma için hazır hale getirilmiştir. modellerinin metin verileri ile eğitilebilmesi için veri setinde yer alan metinler derin öğrenme modellerinde ise tokenize, makine öğrenmesinde TF-IDF vektörlerine dönüştürülmüştür. Modeller karar tipi ve “BAŞVURANIN İDDİA VE TALEPLERİ” alanında yer alan metinler ile eğitilmiştir.

Taslak karar metni üretimi için ön işlem adımları uygulanan veri setindeki metinlere tokenize işlemi uygulanmıştır. Fine-tuning işleminde şikayet konu, şikayet alt konu, karar tipi ve “KARAR” alanında yer alan metinler ile model eğitilmiştir.

Karar tahmininde, makine öğrenmesi modelleri ile elde edilen sonuçlar ile farklı modellerin performansları karşılaştırmalı ve detaylı bir şekilde incelenmiştir. Test veri seti üzerinde Ombudsmanlık kararlarını en başarılı şekilde tahmin eden model SVM olarak gözlenmiştir. DT modeli ise karar tahmininde diğer modellere göre daha başarısız olduğu tespit edilmiştir. SVM modeli ile karar tahmininde %61 F1 skoruna ulaşılmıştır.

Ret kararı tahmininde, derin öğrenme modelleri ile elde edilen sonuçlar ile farklı modellerin performansları karşılaştırmalı ve detaylı bir şekilde incelenmiştir. Test veri seti üzerinde Ombudsmanlık kararlarını en başarılı şekilde tahmin eden model GRU olarak gözlenmiştir. Diğer Modeller,LSTM, CNN ve FFNN makul performans gösterebilir biraz daha az etkili olmuştur. BiLSTM en düşük değerlere sahip olsada dengeli bir performans göstermiştir. GRU ve LSTM modeli ile karar tahmininde %64 F1 skoruna ulaşılmıştır.

Karar türlerine göre performans değerleri karşılaştırıldığında veri setinde örnekleme daha yüksek olan karar türlerinin performans değerlerinin yüksek olduğu gözlenmektedir. Deneyle esnasında veri setinde şikayet konusuna göre dağılımın fazla olması karar tahminini etkilediği gözlemlenmiştir. Karar metninde yer alan “BAŞVURANIN İDDİA VE TALEPLERİ” alanının şikayet konusu ve başvuran talebine göre farklılıkların fazla olması modellerin performansını etkilediği gözlemlenmiştir.

Taslak karar metni oluşturmak için yapılan fine-tuning işleminin sonucunda modelin eğitim verilerini iyi öğrendiğini, iyi genelleme yapabildiğini ve aşırı öğrenme (overfitting) yapmadığı tespit edilmiştir. Kayıp değerlerine göre, modelin eğitim süreci oldukça başarılı olduğu ve modelin kullanılabilir bir durumda olduğu gözlemlenmiştir. Fine-tuning işlemi sonucunda elde edilen eğitim kaybı değeri 0,26 doğrulama değeri ise 0,23 olarak tespit edilmiştir.

Fine-tuning işleminde girdi olarak sağlanan metin miktarı arttığında metin üretimi çeşitliliğinin arttığı gözlemlenmiştir.

Ombudsmanlık web sayfasında benzer kararların yayınlanmaması ve karar dağılımında oluşan sayısal dengesizlik modellerin eğitimini ve test sonucu etkilediği gözlemlenmiştir. Kararlar şikayet konusuna göre ayrılmaktadır. Şikayet konularındaki farklılıklar karar türünü etkilediği gibi metin içeriğininide etkilediği gözlemlenmiştir.

Gelecekteki çalışmalarda, karar tahmini uygulamalarında veri setinde yer alan örneklerin sayısı artırıldığında, karar türüne ve şikayet konusuna göre dengeli dağılım sağlandığında daha iyi sonuçlar elde edilebileceği tahmin edilmektedir.

Bu çalışmada taslak karar metni oluşturma işlemi Google Colab Pro ortamında yüksek kapasiteli 51 GB RAM ve 225 GB disk sınırlılıkları altında geliştirilmiştir. Bu nedenle GPT-3, LLaMA, vb. gibi büyük dil modelleri kullanılamamıştır. Daha güçlü makinelerle çalışarak güncel büyük dil modelleri ile daha iyi sonuçlar elde edileceği tahmin edilmektedir.



KAYNAKLAR

1. Turing, A. M. (1950). Computing machinery and intelligence. *Mind*, LIX (236), 433-460.
2. Buchanan, B. G., & Headrick, T. E. (1970). Some speculation about artificial intelligence and legal reasoning. *Stan. L. Rev.*, 23, 40.
3. YILMAZ, O. G. (2021). Yargı Uygulamasında Yapay Zekâ Kullanımı-Yapay Zekâ Hâkim Cübbesini Giyebilecek mi? *Adalet Dergisi*, (66), 379-415.
4. Collenette, J., Atkinson, K., & Bench-Capon, T. (2023). Explainable AI tools for legal reasoning about cases: A study on the European Court of Human Rights. *Artificial Intelligence*, 317, 103861.
5. İnternet: MarketsandMarkets, "Legal AI Software Market - Global Forecast to 2025. URL: <https://www.marketsandmarkets.com/PressReleases/ai-in-marketing.aspx>. Son Erişim Tarihi: 2024-03-20
6. İnternet: MarketsandMarkets, "Legal AI Software Market - Global Forecast to 2024. URL: <https://www.marketsandmarkets.com/Market-Reports/legal-ai-software-market-88725278.html>. Son Erişim Tarihi: 2024-03-20
7. Efe, A. (2022). Yargısal ve Hukuki Süreçlerde Yapay Zekâ Kullanan Araçlar Üzerine Bir Araştırma. *Bilgi Yönetimi*, 5(1), 92-117
8. G. Pike, G. H. (2018). AI in legal research: Casetext and LexisNexis battle it out. *Information Today*, 35(9), 16-17.
9. Atkinson, K., Bench-Capon, T., & Bollegala, D. (2020). Explanation in AI and law: Past, present and future. *Artificial Intelligence*, 289, 103387.
10. Von Lucke, J., Fitsilis, F., & Etscheid, J. (2022). Using artificial intelligence for legislation-thinking about and selecting realistic topics. *EGOV-CeDEM-ePart*, 32-42.
11. Tong, S., Yuan, J., Zhang, P., & Li, L. (2024). Legal Judgment Prediction via graph boosting with constraints. *Information Processing & Management*, 61(3), 103663.
12. Aletras, N., Tsarapatsanis, D., Preoțiu-Pietro, D., & Lampos, V. (2016). Predicting judicial decisions of the European Court of Human Rights: A natural language processing perspective. *PeerJ computer science*, 2, e93.
13. Pople, J., (1990). Legal expert systems: The inadequacy of a rule-based approach. In: Proceedings of the Thirteenth Australian Computer Science Conference (ACSC13). Monash University, Melbourne, 7-9.
14. Almuzaini, H. A., & Azmi, A. M. (2023). TaSbeeb: A judicial decision support system based on deep learning framework. *Journal of King Saud University-Computer and Information Sciences*, 35(8), 101695.
15. Smoliński, P. R., & Brycz, H. (2024). Individual differences in inaccurate versus accurate economic judgment and decision making. *Metacognitive approach. Personality and Individual Differences*, 219, 112500.

16. Long, S., Tu, C., Liu, Z., & Sun, M. (2019). Automatic judgment prediction via legal reading comprehension. In *Chinese Computational Linguistics: 18th China National Conference, CCL 2019, Kunming, China, October 18–20, 2019, Proceedings 18* (pp. 558-572). Springer International Publishing.
17. Kort, F. (1957). Predicting Supreme Court decisions mathematically: A quantitative analysis of the “right to counsel” cases. *American Political Science Review*, 51(1), 1-12.
18. Nagel, S. S. (1963). Applying correlation analysis to case prediction. *Tex. L. Rev.*, 42, 1006.
19. Yassine, S., Esghir, M., & Ibrihich, O. (2023). Using Artificial Intelligence Tools in the Judicial Domain and the Evaluation of their Impact on the Prediction of Judgments. *Procedia Computer Science*, 220, 1021-1026.
20. Turan, T., Kemaloğlu, N., & Küçüksille, E. (2020). Hukuk'ta Yapay Zeka: Çalışmalar ve Gelecek Öngörülleri. *Mehmet Akif Ersoy Üniversitesi Fen Bilimleri Enstitüsü Dergisi*, 11(2), 246-255.
21. Gulyamov, S., & Bakhramova, M. (2022). Digitalization of international arbitration and dispute resolution by artificial intelligence. *World Bulletin of Management and Law*, 9, 79-85.
22. Lawlor, R. C. (1963). What computers can do: Analysis and prediction of judicial decisions. *American Bar Association Journal*, 337-344
23. Aletras, N., Tsarapatsanis, D., Preoțiu-Pietro, D., & Lampos, V. (2016). Predicting judicial decisions of the European Court of Human Rights: A natural language processing perspective. *PeerJ computer science*, 2, e93.
24. B. G. Buchanan and T. E. Headrick, “Some speculation about artificial intelligence and legal reasoning,” *Stanford Law Review*, vol. 23, pp. 40–62, 05 1970.
25. D. M. Katz, M. J. Bommarito, and J. Blackman, “A general approach for predicting the behavior of the supreme court of the United States,” *PloS one*, vol. 12, no. 4, p. e0174698, 2017.
26. K. Kowsrihawat, P. Vateekul, and P. Boonkwan, “Predicting judicial decisions of criminal cases from Thai Supreme Court using bi-directional GRU with attention mechanism,” 2018 5th Asian Conference on Defense Technology (ACDT), pp. 50–55, 2018.
27. A. Y. Ikram and L. Chakir, “Arabic text classification in the legal domain,” in 2019 Third International Conference on Intelligent Computing in Data Sciences (ICDS). IEEE, 2019, pp. 1–6.
28. Chen, H., Wu, L., Chen, J., Lu, W., & Ding, J. (2022), A comparative study of automated legal text classification using random forests and deep learning, *Information Processing & Management*, 59(2), 102798.
29. Lidén, M. (2024). Can criminal justice be predicted? Using regression analysis to predict judges' decisions on petitions for new criminal trials. *Science & Justice*, 64(1), 43-49.

30. B. G. Buchanan and T. E. Headrick, "Some speculation about artificial intelligence and legal reasoning," *Stanford Law Review*, vol. 23, pp. 40–62, 05 1970.
31. Almuzaini, H. A., & Azmi, A. M. (2023). TaSbeeb: A judicial decision support system based on deep learning framework. *Journal of King Saud University-Computer and Information Sciences*, 35(8), 101695.
32. E. Mumcuoğlu, C. E. Öztürk, H. M. Ozaktas, and A. Koç, "Natural language processing in law: Prediction of outcomes in the higher courts of Turkey," *Information Processing & Management*, vol. 58, no. 5, p. 102684, 2021.
33. Aras, A. C., Öztürk, C. E., & Koç, A. (2022, May). Feedforward Neural Network Based Case Prediction in Turkish Higher Courts. In 2022 30th Signal Processing and Communications Applications Conference (SIU) (pp. 1-4). IEEE.
34. Öztürk, C. E., Özçelik, Ş. B., & Koç, A. (2022, May). Predicting Outcomes of the Court of Cassation of Turkey with Recurrent Neural Networks. In 2022 30th Signal Processing and Communications Applications Conference (SIU) (pp. 1-4). IEEE.
35. Akça, O. (2023). Natural Language Processings in Legal Domain: Classification of Turkish Legal Texts, Yüksek Lisans Tezi, Marmara Üniversitesi Fen Bilimleri Enstitüsü, İstanbul.
36. Habernal, I., Faber, D., Recchia, N., Bretthauer, S., Gurevych, I., Spiecker genannt Döhmann, I., & Burchard, C. (2023). Mining legal arguments in court decisions. *Artificial Intelligence and Law*, 1-38.
37. Zhu, Q., Wei, K., Ding, L., Lai, K.K., (2017). Court judgment decision support system based on medical text mining. *In: WHICEB*, pp. 2.
38. Chen, B., Li, Y., Zhang, S., Lian, H., He, T., (2019). A deep learning method for judicial decision support. In: 2019 IEEE 19th International Conference on Software Quality, Reliability and Security Companion (QRS-C). IEEE, pp. 145–149.
39. Luo, B., Feng, Y., Xu, J., Zhang, X., Zhao, D., (2017). Learning to predict charges for criminal cases with legal basis. *arXiv preprint arXiv:1707.09168*.
40. Demirović, E., Lukina, A., Hebrard, E., Chan, J., Bailey, J., Leckie, C., ... & Stuckey, P. J. (2022). Murtree: Optimal decision trees via dynamic programming and search. *Journal of Machine Learning Research*, 23(26), 1-47.
41. Metsker, O., Trofimov, E., Petrov, M., & Butakov, N. (2019). Russian court decisions data analysis using distributed computing and machine learning to improve lawmaking and law enforcement. *Procedia Computer Science*, 156, 264-273.
42. Quinlan, J. R. (1986). Induction of decision trees. *Machine learning*, 1, 81-106.
43. Alparslan, G., & DURSUN, M. (2023). Konvolüsyonel Sinir Ağları Tabanlı Türkçe Metin Sınıflandırma. *Bilişim Teknolojileri Dergisi*, 16(1), 21-31.
44. Vapnik, V. (1995). Support-vector networks. *Machine learning*, 20, 273-297.

45. Zhang, W., & Gao, F. (2011). An improvement to naive bayes for text classification. *Procedia Engineering*, 15, 2160-2164.
46. Lewis, D. D. (1998, April). Naive (Bayes) at forty: The independence assumption in information retrieval. In *European conference on machine learning* (pp. 4-15). Berlin, Heidelberg: Springer Berlin Heidelberg.
47. Dudani, S. A. (1976). The distance-weighted k-nearest-neighbor rule. *IEEE Transactions on Systems, Man, and Cybernetics*, (4), 325-327.
48. Agresti, A. (2012). *Categorical data analysis* (Vol. 792). John Wiley & Sons.
49. İnternet:Wikipedia,"Logistic regression" URL: https://simple.wikipedia.org/wiki/Logistic_regression, Son Erişim Tarihi: 2024-06-13
50. Chen, T., & Guestrin, C. (2016, August). Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining* (pp. 785-794).
51. Kelle, A. C., & Yüce, H. (2022). MQTT Trafikinde DoS Saldırılarının Makine Öğrenmesi ile Sınıflandırılması ve Modelin SHAP ile Yorumlanması. *Journal of Materials and Mechatronics: A*, 3(1), 50-62.
52. Tran, V., Le Nguyen, M., Tojo, S., & Satoh, K. (2020). Encoded summarization: summarizing documents into continuous vector space for legal case retrieval. *Artificial Intelligence and Law*, 28, 441-467.
53. Wei, F., Qin, H., Ye, S., & Zhao, H. (2018, December). Empirical study of deep learning for text classification in legal document review. In *2018 IEEE International Conference on Big Data (Big Data)* (pp. 3317-3320). IEEE.
54. Abimbola, B., de La Cal Marin, E., & Tan, Q. (2024). Enhancing Legal Sentiment Analysis: A Convolutional Neural Network–Long Short-Term Memory Document-Level Model. *Machine Learning and Knowledge Extraction*, 6(2), 877-897.
55. Zhong, H., Xiao, C., Tu, C., Zhang, T., Liu, Z., & Sun, M. (2020). How does NLP benefit legal system: A summary of legal artificial intelligence. *arXiv preprint arXiv:2004.12158*.
56. Chalkidis, I., Androutsopoulos, I., & Aletras, N. (2019). Neural legal judgment prediction in English. *arXiv preprint arXiv:1906.02059*.
57. Phelps, T., & Ashley, K. (2022). " Alexa, Write a Memo": The Promise and Challenges of AI and Legal Writing. *Legal Writing: J. Legal Writing Inst.*, 26, 329.
58. Callister, P. D. (2020). Law, artificial intelligence, and natural language processing: a funny thing happened on the way to my search results. *Law Libr. J.*, 112, 161.
59. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). Attention is all you need. *Advances in neural information processing systems*, 30.

60. Ebibli, Ö. F. (2022). Hukuk Açısından Yapay Zekânın İncelenmesi. TC İstanbul Üniversitesi Sosyal Bilimler Enstitüsü Sosyal Yapı Sosyal Değişme Anabilim Dalı Yüksek Lisans Tezi.
61. Kesgin, H. T., Yuce, M. K., Dogan, E., Uzun, M. E., Uz, A., Seyrek, H. E., ... & Amasyali, M. F. (2024). Introducing cosmosGPT: Monolingual Training for Turkish Language Models. arXiv preprint arXiv:2404.17336.







GAZİLİ OLMAK AYRICALIKTIR.